
Galaxy Code Documentation

Release

Galaxy Team

May 13, 2015

1	Contents	3
1.1	Galaxy API Documentation	3
1.2	lib	40
2	Indices and tables	367
	Python Module Index	369

Galaxy is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research.

- *Accessible*: Users without programming experience can easily specify parameters and run tools and workflows.
- *Reproducible*: Galaxy captures information so that any user can repeat and understand a complete computational analysis.
- *Transparent*: Users share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis.

Two copies of the Galaxy code documentation are published by the Galaxy Project

- **Galaxy-Dist**: This describes the code in the [most recent official release](#) of Galaxy.
- **Galaxy-Central**: Describes the [current code in the development branch](#) of Galaxy. This is the latest checkin, bleeding edge version of the code. The documentation should never be more than an hour behind the code.

Both copies are hosted at [ReadTheDocs](#), a publicly supported web site for hosting project documentation.

If you have your own copy of the Galaxy source code, you can also generate your own version of this documentation:

```
$ cd doc
$ make html
```

The generated documentation will be in `doc/build/html/` and can be viewed with a web browser. Note that you will need to install Sphinx and a fair number of module dependencies before this will produce output.

For more on the Galaxy Project, please visit the [project home page](#).

Contents

1.1 Galaxy API Documentation

1.1.1 Background

In addition to being accessible through a web interface, Galaxy can also be accessed programmatically, through shell scripts and other programs. The web interface is appropriate for things like exploratory analysis, visualization, construction of workflows, and rerunning workflows on new datasets.

The web interface is less suitable for things like

- Connecting a Galaxy instance directly to your sequencer and running workflows whenever data is ready.
- Running a workflow against multiple datasets (which can be done with the web interface, but is tedious).
- When the analysis involves complex control, such as looping and branching.

The Galaxy API addresses these and other situations by exposing Galaxy internals through an additional interface, known as an Application Programming Interface, or API.

1.1.2 Quickstart

Log in as your user, navigate to the API Keys page in the User menu, and generate a new API key. Make a note of the API key, and then pull up a terminal. Now we'll use the `display.py` script in your `galaxy/scripts/api` directory for a short example:

```
% ./display.py my_key http://localhost:4096/api/histories
Collection Members
-----
#1: /api/histories/8c49be448cfe29bc
    name: Unnamed history
    id: 8c49be448cfe29bc
#2: /api/histories/33b43b4e7093c91f
    name: output test
    id: 33b43b4e7093c91f
```

The result is a Collection of the histories of the user specified by the API key (you). To look at the details of a particular history, say #1 above, do the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc
Member Information
-----
state_details: {'ok': 1, 'failed_metadata': 0, 'upload': 0, 'discarded': 0, 'running': 0, 'setting_m
```

```
state: ok
contents_url: /api/histories/8c49be448cfe29bc/contents
id: 8c49be448cfe29bc
name: Unnamed history
```

This gives detailed information about the specific member in question, in this case the History. To view history contents, do the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc/contents
Collection Members
-----
#1: /api/histories/8c49be448cfe29bc/contents/6f91353f3eb0fa4a
    name: Pasted Entry
    type: file
    id: 6f91353f3eb0fa4a
```

What we have here is another Collection of items containing all of the datasets in this particular history. Finally, to view details of a particular dataset in this collection, execute the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc/contents/6f91353f3eb0fa4a
Member Information
-----
misc_blurb: 1 line
name: Pasted Entry
data_type: txt
deleted: False
file_name: /Users/yoplait/work/galaxy-stock/database/files/000/dataset_82.dat
state: ok
download_url: /datasets/6f91353f3eb0fa4a/display?to_ext=txt
visible: True
genome_build: ?
model_class: HistoryDatasetAssociation
file_size: 17
metadata_data_lines: 1
id: 6f91353f3eb0fa4a
misc_info: uploaded txt file
metadata_dbkey: ?
```

And now you've successfully used the API to request and select a history, browse the contents of that history, and then look at detailed information about a particular dataset.

For a more comprehensive Data Library example, set the following option in your galaxy.ini as well, and restart galaxy again:

```
admin_users = you@example.org
library_import_dir = /path/to/some/directory
```

In the directory you specified for 'library_import_dir', create some subdirectories, and put (or symlink) files to import into Galaxy into those subdirectories.

In Galaxy, create an account that matches the address you put in 'admin_users', then browse to that user's preferences and generate a new API Key. Copy the key to your clipboard and then use these scripts:

```
% ./display.py my_key http://localhost:4096/api/libraries
Collection Members
-----

0 elements in collection

% ./library_create_library.py my_key http://localhost:4096/api/libraries api_test 'API Test Library'
```



```

Response
-----
/api/libraries/f3f73e481f432006
  name: api_test
  id: f3f73e481f432006

% ./display.py my_key http://localhost:4096/api/libraries
Collection Members
-----
/api/libraries/f3f73e481f432006
  name: api_test
  id: f3f73e481f432006

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006
Member Information
-----
synopsis: None
contents_url: /api/libraries/f3f73e481f432006/contents
description: API Test Library
name: api_test

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents
Collection Members
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591f61ddda595d2c3670
  name: /
  type: folder
  id: 28202595c0d2591f61ddda595d2c3670

% ./library_create_folder.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents 28202595c0d2591f61ddda595d2c3670
Response
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
  name: api_test_folder1
  id: 28202595c0d2591fa4f9089d2303fd89

% ./library_upload_from_import_dir.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents 28202595c0d2591fa4f9089d2303fd89
Response
-----
/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
  name: 2.bed
  id: e9ef7fdb2db87d7b
/api/libraries/f3f73e481f432006/contents/3b7f6a31f80a5018
  name: 3.bed
  id: 3b7f6a31f80a5018

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents
Collection Members
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591f61ddda595d2c3670
  name: /
  type: folder
  id: 28202595c0d2591f61ddda595d2c3670
/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
  name: /api_test_folder1
  type: folder
  id: 28202595c0d2591fa4f9089d2303fd89
/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b

```

```
name: /api_test_folder1/2.bed
type: file
id: e9ef7fdb2db87d7b
/api/libraries/f3f73e481f432006/contents/3b7f6a31f80a5018
name: /api_test_folder1/3.bed
type: file
id: 3b7f6a31f80a5018

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
Member Information
-----
misc_blurb: 68 regions
metadata_endCol: 3
data_type: bed
metadata_columns: 6
metadata_nameCol: 4
uploaded_by: nate@...
metadata_strandCol: 6
name: 2.bed
genome_build: hg19
metadata_comment_lines: None
metadata_startCol: 2
metadata_chromCol: 1
file_size: 4272
metadata_data_lines: 68
message:
metadata_dbkey: hg19
misc_info: uploaded bed file
date_uploaded: 2010-06-22T17:01:51.266119
metadata_column_types: str, int, int, str, int, str
```

Other parameters are valid when uploading, they are the same parameters as are used in the web form, like 'link_data_only' and etc.

The request and response format should be considered alpha and are subject to change.

1.1.3 API Design Guidelines

The following section outlines guidelines related to extending and/or modifying the Galaxy API. The Galaxy API has grown in an ad-hoc fashion over time by many contributors and so clients SHOULD NOT expect the API will conform to these guidelines - but developers contributing to the Galaxy API SHOULD follow these guidelines.

- API functionality should include docstring documentation for consumption by readthedocs.org.
- Developers should familiarize themselves with the HTTP status code definitions <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>. The API responses should properly set the status code according to the result - in particular 2XX responses should be used for successful requests, 4XX for various kinds of client errors, and 5XX for the errors on the server side.
- If there is an error processing some part of request (one item in a list for instance), the status code should be set to reflect the error and the partial result may or may not be returned depending on the controller - this behavior should be documented.
- API methods should throw a finite number of exceptions (defined in [exceptions Package](#)) and these should subclass *MessageException* and not paste/wsgi HTTP exceptions. When possible, the framework itself should be responsible catching these exceptions, setting the status code, and building an error response.

- Error responses should not consist of plain text strings - they should be dictionaries describing the error and containing the following:

```
{
  "status_code": 400,
  "err_code": 400007,
  "err_msg": "Request contained invalid parameter, action could not be completed.",
  "type": "error",
  "extra_error_info": "Extra information."
}
```

Various error conditions (once a format has been chosen and framework to enforce it in place) should be spelled out in this document.

- Backward compatibility is important and should be maintained when possible. If changing behavior in a non-backward compatible way please ensure one of the following holds - there is a strong reason to believe no consumers depend on a behavior, the behavior is effectively broken, or the API method being modified has not been part of a tagged dist release.

The following bullet points represent good practices more than guidelines, please consider them when modifying the API.

- Functionality should not be copied and pasted between controllers - consider refactoring functionality into associated classes or short of that into Mixins (http://en.wikipedia.org/wiki/Composition_over_inheritance) or into Managers ([managers Package](#)).
- API additions are more permanent changes to Galaxy than many other potential changes and so a second opinion on API changes should be sought. (Consider a pull request!)
- New API functionality should include functional tests. These functional tests should be implemented in Python and placed in *test/functional/api*. (Once such a framework is in place - it is not right now).
- Changes to reflect modifications to the API should be pushed upstream to the BioBlend project if possible.

Longer term goals/notes.

- It would be advantageous to have a clearer separation of anonymous and admin handling functionality.
- If at some point in the future, functionality needs to be added that breaks backward compatibility in a significant way to a component used by the community - a “dev” variant of the API will be established and the community should be alerted and given a timeframe for when the old behavior will be replaced with the new behavior.
- Consistent standards for range-based requests, batch requests, filtered requests, etc... should be established and documented here.

1.1.4 API Controllers

Galaxy offers the following API controllers:

annotations Module

API operations on annotations.

```
class galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesHistory, galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin, galaxy.web.base.controller.UsesStoredWorkflowMixin, galaxy.model.item_attrs.UsesAnnotationsMixin

    create (trans, *args, **kwargs)
```

```
    delete (trans, *args, **kwargs)
    index (trans, *args, **kwargs)
    undelete (trans, *args, **kwargs)

class galaxy.webapps.galaxy.api.annotations.HistoryAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'history_annotations'
    tagged_item_id = 'history_id'

class galaxy.webapps.galaxy.api.annotations.HistoryContentAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'history_content_annotations'
    tagged_item_id = 'history_content_id'

class galaxy.webapps.galaxy.api.annotations.WorkflowAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'workflow_annotations'
    tagged_item_id = 'workflow_id'
```

authenticate Module

API key retrieval through BaseAuth Sample usage:

```
curl -user zipzap@foo.com:password http://localhost:8080/api/authenticate/baseauth
```

Returns:

```
{ "api_key": "baa4d6e3a156d3033f05736255f195f9"
}
```

```
class galaxy.webapps.galaxy.api.authenticate.AuthenticationController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    get_api_key (trans, *args, **kwargs)
        def get_api_key( self, trans, **kwd ) * GET /api/authenticate/baseauth
            returns an API key for authenticated user based on BaseAuth headers

            Returns api_key in json format
            Return type dict
            Raises ObjectNotFound, HTTPBadRequest
```

configuration Module

API operations allowing clients to determine Galaxy instance's capabilities and configuration settings.

```
class galaxy.webapps.galaxy.api.configuration.ConfigurationController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    EXPOSED_ADMIN_OPTIONS = ['library_import_dir', 'user_library_import_dir', 'allow_library_path_paste', 'allow_use
    EXPOSED_USER_OPTIONS = ['enable_unique_workflow_defaults', 'ftp_upload_site', 'ftp_upload_dir', 'wiki_url', 'supp
```

get_config_dict (*config*, *is_admin=False*)

index (*trans*, **args*, ***kwargs*)

GET /api/configuration Return an object containing exposable configuration settings.

Note: a more complete list is returned if the user is an admin.

dataset_collections Module

class `galaxy.webapps.galaxy.api.dataset_collections.DatasetCollectionsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesHistoryMixin`, `galaxy.web.base.controller.UsesLibraryMixinItems`

create (*trans*, **args*, ***kwargs*)

•**POST /api/dataset_collections:** create a new dataset collection instance.

Parameters **payload** (*dict*) – (optional) dictionary structure containing: * **collection_type**: dataset colltion type to create. * **instance_type**: Instance type - ‘history’ or ‘library’. * **name**: the new dataset collections’s name * **datasets**: object describing datasets for collection

Return type dict

Returns element view of new dataset collection

index (*trans*, **args*, ***kwargs*)

show (*trans*, **args*, ***kwargs*)

datasets Module

datatypes Module

API operations allowing clients to determine datatype supported by Galaxy.

class `galaxy.webapps.galaxy.api.datatypes.DatatypesController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`

index (*trans*, **args*, ***kwargs*)

GET /api/datatypes Return an object containing upload datatypes.

mapping (*trans*, **args*, ***kwargs*)

GET /api/datatypes/mapping Return a dictionary of class to class mappings.

sniffers (*trans*, **args*, ***kwargs*)

GET /api/datatypes/sniffers Return a list of sniffers.

extended_metadata Module

API operations on annotations.

class `galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesExtendedMetadataMixin`, `galaxy.web.base.controller.UsesHistoryMixin`, `galaxy.web.base.controller.UsesLibraryMixinItems`, `galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin`, `galaxy.web.base.controller.UsesStoredWorkflowMixin`

create (*trans*, **args*, ***kwargs*)

```
delete (trans, *args, **kwargs)
```

```
index (trans, *args, **kwargs)
```

```
undelete (trans, *args, **kwargs)
```

```
class galaxy.webapps.galaxy.api.extended_metadata.HistoryDatasetExtendMetadataController (app)  
    Bases: galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController
```

```
    controller_name = 'history_dataset_extended_metadata'
```

```
    exmeta_item_id = 'history_content_id'
```

```
class galaxy.webapps.galaxy.api.extended_metadata.LibraryDatasetExtendMetadataController (app)  
    Bases: galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController
```

```
    controller_name = 'library_dataset_extended_metadata'
```

```
    exmeta_item_id = 'library_content_id'
```

folder_contents Module

API operations on the contents of a library folder.

```
class galaxy.webapps.galaxy.api.folder_contents.FolderContentsController (app)  
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibrary  
           galaxy.web.base.controller.UsesLibraryMixinItems, galaxy.web.base.controller.UsesHistory
```

Class controls retrieval, creation and updating of folder contents.

```
build_path (trans, folder)
```

Search the path upwards recursively and load the whole route of names and ids for breadcrumb building purposes.

Parameters

- **folder** – current folder for navigating up
- **type** – Galaxy LibraryFolder

Returns list consisting of full path to the library

Type list

```
create (self, trans, library_id, payload, **kwd)
```

• **POST /api/folders/{encoded_id}/contents** create a new library file from an HDA

Parameters **payload** – dictionary structure containing:

Returns a dictionary containing the id, name, and ‘show’ url of the new item

Return type dict

Raises `ObjectAttributeInvalidException`, `InsufficientPermissionsException`, `ItemAccessibilityException`, `InternalServerError`

```
index (trans, *args, **kwargs)
```

GET /api/folders/{encoded_folder_id}/contents

Displays a collection (list) of a folder's contents (files and folders). Encoded folder ID is prepended with 'F' if it is a folder as opposed to a data set which does not have it. Full path is provided in response as a separate object providing data for breadcrumb path building.

Parameters

- **folder_id** (*encoded string*) – encoded ID of the folder which contents should be library_dataset_dict
- **kwd** (*dict*) – keyword dictionary with other params

Returns dictionary containing all items and metadata

Type dict

Raises MalformedId, InconsistentDatabase, ObjectNotFound, InternalServerError

show (*trans, *args, **kwargs*)

GET /api/folders/{encoded_folder_id}/

update (*trans, *args, **kwargs*)

PUT /api/folders/{encoded_folder_id}/contents

folders Module

API operations on library folders.

class galaxy.webapps.galaxy.api.folders.**FoldersController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibraryMixinItems*

create (*self, trans, encoded_parent_folder_id, **kwd*)

*POST /api/folders/{encoded_parent_folder_id}

Create a new folder object underneath the one specified in the parameters.

Parameters

- **encoded_parent_folder_id** (*an encoded id string (should be prefixed by 'F')*) – the parent folder's id (required)
- **name** (*str*) – the name of the new folder (required)
- **description** (*str*) – the description of the new folder

Returns information about newly created folder, notably including ID

Return type dictionary

Raises RequestParameterMissingException

delete (*self, trans, id, **kwd*)

- **DELETE /api/folders/{id}** marks the folder with the given *id* as *deleted* (or removes the *deleted* mark if the *undelete* param is true)

Note: Currently, only admin users can un/delete folders.

Parameters

- **id** (*an encoded id string*) – the encoded id of the folder to un/delete
- **undelete** (*bool*) – (optional) flag specifying whether the item should be deleted or undeleted, defaults to false:

Returns detailed folder information

Return type dictionary

Raises ItemAccessibilityException, MalformedId, ObjectNotFound

get_permissions (*trans*, **args*, ***kwargs*)

•GET /api/folders/{id}/permissions

Load all permissions for the given folder id and return it.

Parameters

- **encoded_folder_id** (*an encoded id string*) – the encoded id of the folder
- **scope** (*string*) – either ‘current’ or ‘available’

Returns dictionary with all applicable permissions’ values

Return type dictionary

Raises ObjectNotFound, InsufficientPermissionsException

index (*trans*, **args*, ***kwargs*)

*GET /api/folders/ This would normally display a list of folders. However, that would be across multiple libraries, so it’s not implemented.

set_permissions (*trans*, **args*, ***kwargs*)

def set_permissions(self, trans, encoded_folder_id, **kwd): *POST /api/folders/{encoded_folder_id}/permissions

Parameters

- **encoded_folder_id** (*an encoded id string*) – the encoded id of the folder to set the permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: set_permissions
- **add_ids** [] (*string or list*) – list of Role.id defining roles that should have add item permission on the folder
- **manage_ids** [] (*string or list*) – list of Role.id defining roles that should have manage permission on the folder
- **modify_ids** [] (*string or list*) – list of Role.id defining roles that should have modify permission on the folder

Return type dictionary

Returns dict of current roles for all available permission types.

Raises RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException

show (*self*, *trans*, *id*, ***kwd*)

*GET /api/folders/{encoded_folder_id}

Displays information about a folder.

Parameters **id** (*an encoded id string (has to be prefixed by ‘F’)*) – the folder’s encoded id (required)

Returns dictionary including details of the folder

Return type dict

update (*trans*, **args*, ***kwargs*)
PUT /api/folders/{encoded_folder_id}

forms Module

API operations on FormDefinition objects.

class `galaxy.webapps.galaxy.api.forms.FormDefinitionApiController` (*app*)
Bases: `galaxy.web.base.controller.BaseApiController`

create (*trans*, **args*, ***kwargs*)
POST /api/forms Creates a new form.

index (*trans*, **args*, ***kwargs*)
GET /api/forms Displays a collection (list) of forms.

show (*trans*, **args*, ***kwargs*)
GET /api/forms/{encoded_form_id} Displays information about a form.

ftp_files Module

API operations on remote files.

class `galaxy.webapps.galaxy.api.ftp_files.FTPFilesApiController` (*app*)
Bases: `galaxy.web.base.controller.BaseApiController`

index (*trans*, **args*, ***kwargs*)
GET /api/ftp_files/
Displays remote files.

Parameters

- **target** (*str*) – target to load available datasets from, defaults to ftp possible values: ftp, userdir
- **format** – requested format of data, defaults to flat possible values: flat, jstree, ajax

Returns list of available files

Return type *list*

genomes Module

class `galaxy.webapps.galaxy.api.genomes.GenomesController` (*app*)
Bases: `galaxy.web.base.controller.BaseApiController`

RESTful controller for interactions with genome data.

index (*trans*, **args*, ***kwargs*)
GET /api/genomes: returns a list of installed genomes

show (*trans*, **args*, ***kwargs*)
GET /api/genomes/{id}
Returns information about build <id>

`galaxy.webapps.galaxy.api.genomes.get_id` (*base*, *format*)

group_roles Module

API operations on Group objects.

```
class galaxy.webapps.galaxy.api.group_roles.GroupRolesAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    delete (trans, *args, **kwargs)
        DELETE /api/groups/{encoded_group_id}/roles/{encoded_role_id} Removes a role from a group

    index (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id}/roles Displays a collection (list) of groups.

    show (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id}/roles/{encoded_role_id} Displays information about a group role.

    update (trans, *args, **kwargs)
        PUT /api/groups/{encoded_group_id}/roles/{encoded_role_id} Adds a role to a group
```

group_users Module

API operations on Group objects.

```
class galaxy.webapps.galaxy.api.group_users.GroupUsersAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    delete (trans, *args, **kwargs)
        DELETE /api/groups/{encoded_group_id}/users/{encoded_user_id} Removes a user from a group

    index (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id}/users Displays a collection (list) of groups.

    show (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id}/users/{encoded_user_id} Displays information about a group user.

    update (trans, *args, **kwargs)
        PUT /api/groups/{encoded_group_id}/users/{encoded_user_id} Adds a user to a group
```

groups Module

API operations on Group objects.

```
class galaxy.webapps.galaxy.api.groups.GroupAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create (trans, *args, **kwargs)
        POST /api/groups Creates a new group.

    index (trans, *args, **kwargs)
        GET /api/groups Displays a collection (list) of groups.

    show (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id} Displays information about a group.

    update (trans, *args, **kwargs)
        PUT /api/groups/{encoded_group_id} Modifies a group.
```

histories Module

API operations on a history.

See also:

`galaxy.model.History`

class `galaxy.webapps.galaxy.api.histories.HistoriesController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesHistoryMixin`, `galaxy.web.base.controller.UsesTagsMixin`, `galaxy.web.base.controller.ExportsHistoryMixin`, `galaxy.web.base.controller.ImportsHistoryMixin`

archive_download (*trans*, **args*, ***kwargs*)

`export_download(self, trans, id, jeha_id) * GET /api/histories/{id}/exports/{jeha_id}:`

If ready and available, return raw contents of exported history. Use/poll “PUT /api/histories/{id}/exports” to initiate the creation of such an export - when ready that route will return 200 status code (instead of 202) with a JSON dictionary containing a `download_url`.

archive_export (*trans*, **args*, ***kwargs*)

`export_archive(self, trans, id, payload) * PUT /api/histories/{id}/exports:`

start job (if needed) to create history export for corresponding history.

Parameters `id` (*str*) – the encoded id of the history to undelete

Return type dict

Returns object containing url to fetch export from.

citations (*trans*, **args*, ***kwargs*)

create (*trans*, *payload*)

•**POST /api/histories:** create a new history

Parameters `payload` (*dict*) – (optional) dictionary structure containing: * `name`: the new history’s name * `history_id`: the id of the history to copy * `archive_source`: the url that will generate the archive to import * `archive_type`: ‘url’ (default)

Return type dict

Returns element view of new history

delete (*self*, *trans*, *id*, ***kwd*)

•**DELETE /api/histories/{id}** delete the history with the given `id`

Note: Stops all active jobs in the history if purge is set.

Parameters

- `id` (*str*) – the encoded id of the history to delete
- `kwd` (*dict*) – (optional) dictionary structure containing:
 - **payload: a dictionary itself containing:**
 - * `purge`: if True, purge the history and all of its HDAs

Return type dict

Returns an error object if an error occurred or a dictionary containing: * `id`: the encoded id of the history, * `deleted`: if the history was marked as deleted, * `purged`: if the history was purged

index (*trans*, *deleted*=*'False'*)

- **GET /api/histories:** return undeleted histories for the current user
- **GET /api/histories/deleted:** return deleted histories for the current user

Note: Anonymous users are allowed to get their current history

Parameters **deleted** (*boolean*) – if True, show only deleted histories, if False, non-deleted

Return type *list*

Returns list of dictionaries containing summary history information

show (*trans*, *id*, *deleted*=*'False'*)

- **GET /api/histories/{id}:** return the history with `id`
- **GET /api/histories/deleted/{id}:** return the deleted history with `id`
- **GET /api/histories/most_recently_used:** return the most recently used history

Parameters

- **id** (*an encoded id string*) – the encoded id of the history to query or the string `'most_recently_used'`
- **deleted** (*boolean*) – if True, allow information on a deleted history to be shown.

Return type dictionary

Returns detailed history information from `galaxy.web.base.controller.UsesHistoryMixin.get_h`

undelelete (*self*, *trans*, *id*, ***kwd*)

- **POST /api/histories/deleted/{id}/undelelete:** undelelete history (that hasn't been purged) with the given `id`

Parameters **id** (*str*) – the encoded id of the history to undelelete

Return type *str*

Returns `'OK'` if the history was undeleleted

update (*self*, *trans*, *id*, *payload*, ***kwd*)

- **PUT /api/histories/{id}** updates the values for the history with the given `id`

Parameters

- **id** (*str*) – the encoded id of the history to undelelete
- **payload** (*dict*) – a dictionary containing any or all the fields in `galaxy.model.History.to_dict()` and/or the following:
 - **annotation**: an annotation for the history

Return type *dict*

Returns an error object if an error occurred or a dictionary containing any values that were different from the original and, therefore, updated

history_contents Module

API operations on the contents of a history.

class `galaxy.webapps.galaxy.api.history_contents.HistoryContentsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesHistoryMixin`, `galaxy.web.base.controller.UsesLibraryMixin`, `galaxy.web.base.controller.UsesLibraryMixinItems`, `galaxy.web.base.controller.UsesTagsMixin`

create (*self*, *trans*, *history_id*, *payload*, ***kwd*)

• **POST** `/api/histories/{history_id}/contents/{type}` create a new HDA by copying an accessible LibraryDataset

Parameters

- **history_id** (*str*) – encoded id string of the new HDA’s History
- **type** (*str*) – Type of history content - ‘dataset’ (default) or ‘dataset_collection’.
- **payload** (*dict*) – dictionary structure containing:
 - copy from library (for type ‘dataset’): ‘source’ = ‘library’ ‘content’ = [the encoded id from the library dataset]
 - copy from history dataset (for type ‘dataset’): ‘source’ = ‘hda’ ‘content’ = [the encoded id from the HDA]
 - copy from history dataset collection (for type ‘dataset_collection’) ‘source’ = ‘hdca’ ‘content’ = [the encoded id from the HDCA]
 - create new history dataset collection (for type ‘dataset_collection’) ‘source’ = ‘new_collection’ (default ‘source’ if type is ‘dataset_collection’ - no need to specify this)
 - ‘collection_type’ = For example, “list”, “paired”, “list:paired”. ‘name’ = Name of new dataset collection. ‘element_identifiers’ = Recursive list structure defining collection.

Each element must have ‘src’ which can be ‘hda’, ‘ldda’, ‘hdca’, or ‘new_collection’, as well as a ‘name’ which is the name of element (e.g. “forward” or “reverse” for paired datasets, or arbitrary sample names for instance for lists). For all src’s except ‘new_collection’ - a encoded ‘id’ attribute must be included with element as well. ‘new_collection’ sources must defined a ‘collection_type’ and their own list of (potentially) nested ‘element_identifiers’.

..note: Currently, a user can only copy an HDA from a history that the user owns.

Return type dict

Returns dictionary containing detailed information for the new HDA

delete (*self*, *trans*, *history_id*, *id*, ***kwd*)

• **DELETE** `/api/histories/{history_id}/contents/{id}` delete the HDA with the given *id*

Note: Currently does not stop any active jobs for which this dataset is an output.

Parameters

- **id** (*str*) – the encoded id of the history to delete
- **purge** (*bool*) – if True, purge the HDA
- **kwd** (*dict*) – (optional) dictionary structure containing:
 - **payload: a dictionary itself containing:**
 - * **purge:** if True, purge the HDA

Note: that payload optionally can be placed in the query string of the request. This allows clients that strip the request body to still purge the dataset.

Return type *dict*

Returns an error object if an error occurred or a dictionary containing: * **id:** the encoded id of the history, * **deleted:** if the history was marked as deleted, * **purged:** if the history was purged

index (*self, trans, history_id, ids=None, **kwd*)

- **GET /api/histories/{history_id}/contents** return a list of HDA data for the history with the given *id*

Note: Anonymous users are allowed to get their current history contents

If *ids* is not given, *index* returns a list of *summary* objects for every HDA associated with the given *history_id*.

If *ids* is given, *index* returns a *more complete* json object for each HDA in the *ids* list.

Parameters

- **history_id** (*str*) – encoded id string of the HDA's History
- **ids** (*str*) – (optional) a comma separated list of encoded *HDA* ids
- **types** (*str*) – (optional) kinds of contents to index (currently just dataset, but dataset_collection will be added shortly).

Return type *list*

Returns dictionaries containing summary or detailed HDA information

See also:

`_summary_hda_dict()` and `galaxy.web.base.controller.UsesHistoryDatasetAssociationMixi`

show (*self, trans, id, history_id, **kwd*)

- **GET /api/histories/{history_id}/contents/{id}** return detailed information about an HDA within a history

Note: Anonymous users are allowed to get their current history contents

Parameters

- **ids** – the encoded id of the HDA to return
- **history_id** (*str*) – encoded id string of the HDA's History

Return type dict**Returns** dictionary containing detailed HDA information**See also:**

`galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin.get_hda_dict()`

update (*self*, *trans*, *history_id*, *id*, *payload*, ***kwd*)

- **PUT /api/histories/{history_id}/contents/{id}** updates the values for the HDA with the given *id*

Parameters

- **history_id** (*str*) – encoded id string of the HDA's History
- **id** (*str*) – the encoded id of the history to undelete
- **payload** (*dict*) – a dictionary containing any or all the fields in `galaxy.model.HistoryDatasetAssociation.to_dict()` and/or the following:
 - **annotation**: an annotation for the HDA

Return type dict

Returns an error object if an error occurred or a dictionary containing any values that were different from the original and, therefore, updated

item_tags Module

API operations related to tagging items.

class `galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesTagsM`

create (*trans*, **args*, ***kwargs*)

delete (*trans*, **args*, ***kwargs*)

index (*trans*, **args*, ***kwargs*)

show (*trans*, **args*, ***kwargs*)

update (*trans*, **args*, ***kwargs*)

class `galaxy.webapps.galaxy.api.item_tags.HistoryContentTagsController` (*app*)

Bases: `galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController`

controller_name = 'history_content_tags'

tagged_item_class = 'HistoryDatasetAssociation'

tagged_item_id = 'history_content_id'

class `galaxy.webapps.galaxy.api.item_tags.HistoryTagsController` (*app*)

Bases: `galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController`

controller_name = 'history_tags'

```
    tagged_item_class = 'History'
    tagged_item_id = 'history_id'
class galaxy.webapps.galaxy.api.item_tags.WorkflowTagsController(app)
    Bases: galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController

    controller_name = 'workflow_tags'
    tagged_item_class = 'StoredWorkflow'
    tagged_item_id = 'workflow_id'
```

job_files Module

API for asynchronous job running mechanisms can use to fetch or put files related to running and queued jobs.

```
class galaxy.webapps.galaxy.api.job_files.JobFilesAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController
```

This job files controller allows remote job running mechanisms to read and modify the current state of files for queued and running jobs. It is certainly not meant to represent part of Galaxy's stable, user facing API.

Furthermore, even if a user key corresponds to the user running the job, it should not be accepted for authorization - this API allows access to low-level unfiltered files and such authorization would break Galaxy's security model for tool execution.

create (*self*, *trans*, *job_id*, *payload*, ***kwargs*)

- **POST /api/jobs/{job_id}/files** Populate an output file (formal dataset, task split part, working directory file (such as those related to metadata)). This should be a multipart post with a 'file' parameter containing the contents of the actual file to create.

Parameters

- **job_id** (*str*) – encoded id string of the job
- **payload** (*dict*) – dictionary structure containing:: 'job_key' = Key authenticating
'path' = Path to file to create.

..note: This API method is intended only for consumption by job runners, not end users.

Return type dict

Returns an okay message

index (*self*, *trans*, *job_id*, ***kwargs*)

- **GET /api/jobs/{job_id}/files** Get a file required to staging a job (proper datasets, extra inputs, task-split inputs, working directory files).

Parameters

- **job_id** (*str*) – encoded id string of the job
- **path** (*str*) – Path to file.
- **job_key** (*str*) – A key used to authenticate this request as acting on behalf of a job runner for the specified job.

..note: This API method is intended only for consumption by job runners, not end users.

Return type *binary*

Returns contents of file

jobs Module

API operations on a jobs.

See also:

`galaxy.model.Jobs`

class `galaxy.webapps.galaxy.api.jobs.JobController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesHistory`, `galaxy.web.base.controller.UsesLibraryMixinItems`

create (*trans*, **args*, ***kwargs*)

index (*trans*, *state=None*, *tool_id=None*, *history_id=None*)

• **GET /api/jobs:** return jobs for current user

Parameters *state* (*string or list*) – limit listing of jobs to those that match one of the included states. If none, all are returned.

Valid Galaxy job states include: ‘new’, ‘upload’, ‘waiting’, ‘queued’, ‘running’, ‘ok’, ‘error’, ‘paused’, ‘deleted’, ‘deleted_new’

Parameters

- **tool_id** (*string or list*) – limit listing of jobs to those that match one of the included tool_ids. If none, all are returned.
- **history_id** (*string*) – limit listing of jobs to those that match the history_id. If none, all are returned.

Return type *list*

Returns list of dictionaries containing summary job information

inputs (*trans*, **args*, ***kwargs*)

`show(trans, id) * GET /api/jobs/{job_id}/inputs`

returns input datasets created by job

Parameters *id* (*string*) – Encoded job id

Return type dictionary

Returns dictionary containing input dataset associations

outputs (*trans*, **args*, ***kwargs*)

`show(trans, id) * GET /api/jobs/{job_id}/outputs`

returns output datasets created by job

Parameters *id* (*string*) – Encoded job id

Return type dictionary

Returns dictionary containing output dataset associations

search (*trans*, *payload*)

• **POST /api/jobs/search:** return jobs for current user

Parameters **payload** (*dict*) – Dictionary containing description of requested job. This is in the same format as a request to POST /apt/tools would take to initiate a job

Return type *list*

Returns list of dictionaries containing summary job information of the jobs that match the requested job run

This method is designed to scan the list of previously run jobs and find records of jobs that had the exact some input parameters and datasets. This can be used to minimize the amount of repeated work, and simply recycle the old results.

show (*trans*, *id*)

• **GET /api/jobs/{job_id}:** return jobs for current user

Parameters

- **id** (*string*) – Specific job id
- **full** (*boolean*) – whether to return extra information

Return type dictionary

Returns dictionary containing full description of job data

lda_datasets Module

API operations on the library datasets.

class `galaxy.webapps.galaxy.api.lda_datasets.LibraryDatasetsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesVisual`

delete (*trans*, **args*, ***kwargs*)

`delete(self, trans, encoded_dataset_id, **kwd):` * DELETE /api/libraries/datasets/{encoded_dataset_id}

Marks the dataset deleted or undeleted based on the value of the undelete flag. If the flag is not present it is considered False and the item is marked deleted.

Parameters **encoded_dataset_id** (*an encoded id string*) – the encoded id of the dataset to change

Returns dict containing information about the dataset

Return type dictionary

download (*self*, *trans*, *format*, ***kwd*)

• **GET /api/libraries/datasets/download/{format}**

• **POST /api/libraries/datasets/download/{format}** Downloads requested datasets (identified by encoded IDs) in requested format.

example: GET localhost:8080/api/libraries/datasets/download/tbz?ld_ids%255B%255D=a0d8

Note: supported format values are: 'zip', 'tgz', 'tbz', 'uncompressed'

Parameters

- **format** (*string*) – string representing requested archive format
- **ld_ids** [] (*an array*) – an array of encoded ids

Return type file

Returns either archive with the requested datasets packed inside or a single uncompressed dataset

Raises `MessageException`, `ItemDeletionException`, `ItemAccessibilityException`, `HTTPBadRequest`, `OSError`, `IOError`, `ObjectNotFound`

load (*trans*, **args*, ***kwargs*)

load(self, trans, ****kwd**): * POST /api/libraries/datasets Load dataset from the given source into the library. Source can be:

user directory - root folder specified in galaxy.ini as “\$user_library_import_dir”

example path: `path/to/galaxy/$user_library_import_dir/user@example.com/{user can browse everything here}` the folder with the user login has to be created beforehand

(admin)import directory - root folder specified in galaxy ini as “\$library_import_dir”

example path: `path/to/galaxy/$library_import_dir/{admin can browse everything here}`

(admin)any absolute or relative path - option allowed with “allow_library_path_paste” in galaxy.ini

Parameters

- **encoded_folder_id** (*an encoded id string*) – the encoded id of the folder to import dataset(s) to
- **source** (*str*) – source the datasets should be loaded form
- **link_data** (*bool*) – flag whether to link the dataset to data or copy it to Galaxy, defaults to copy while linking is set to True all symlinks will be resolved `_once_`
- **preserve_dirs** (*bool*) – flag whether to preserve the directory structure when importing dir if False only datasets will be imported
- **file_type** (*str*) – file type of the loaded datasets, defaults to ‘auto’ (autodetect)
- **dbkey** (*str*) – dbkey of the loaded genome, defaults to ‘?’ (unknown)

Returns dict containing information about the created upload job

Return type dictionary

show (*self*, *trans*, *id*, ***kwd*)

- **GET /api/libraries/datasets/{encoded_dataset_id}**: Displays information about the dataset identified by the encoded ID.

Parameters **id** (*an encoded id string*) – the encoded id of the dataset to query

Returns detailed dataset information from base controller

Return type dictionary

See also:

`galaxy.web.base.controller.UsesLibraryMixinItems.get_library_dataset`

show_roles (*trans*, **args*, ***kwargs*)

show_roles(self, trans, id, **kwd): * GET /api/libraries/datasets/{encoded_dataset_id}/permissions

Displays information about current or available roles for a given dataset permission.

Parameters

- **encoded_dataset_id** (*an encoded id string*) – the encoded id of the dataset to query
- **scope** (*string*) – either ‘current’ or ‘available’

Return type dictionary

Returns either dict of current roles for all permission types or dict of available roles to choose from (is the same for any permission type)

show_version (*trans*, **args*, ***kwargs*)

show_version(self, trans, encoded_dataset_id, encoded_ldda_id, **kwd): * GET /api/libraries/datasets/:encoded_dataset_id/versions/:encoded_ldda_id

Displays information about specific version of the library_dataset (i.e. ldda).

Parameters

- **encoded_dataset_id** (*an encoded id string*) – the encoded id of the dataset to query
- **encoded_ldda_id** (*an encoded id string*) – the encoded id of the ldda to query

Return type dictionary

Returns dict of ldda’s details

update_permissions (*trans*, **args*, ***kwargs*)

def update(self, trans, encoded_dataset_id, **kwd): *POST /api/libraries/datasets/{encoded_dataset_id}/permissions

Parameters

- **encoded_dataset_id** (*an encoded id string*) – the encoded id of the dataset to update permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: make_private, remove_restrictions, set_permissions
- **access_ids** [] (*string or list*) – list of Role.name defining roles that should have access permission on the dataset
- **manage_ids** [] (*string or list*) – list of Role.name defining roles that should have manage permission on the dataset
- **modify_ids** [] (*string or list*) – list of Role.name defining roles that should have modify permission on the library dataset item

Return type dictionary

Returns dict of current roles for all available permission types

Raises RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException

libraries Module

API operations on a data library.

class `galaxy.webapps.galaxy.api.libraries.LibrariesController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`

create (*self, trans, payload, **kwd*)

• **POST /api/libraries:** Creates a new library. Only `name` parameter is required.

Note: Currently, only admin users can create libraries.

Parameters **payload** (*dict*) – dictionary structure containing:: ‘name’: the new library’s name (required) ‘description’: the new library’s description (optional) ‘synopsis’: the new library’s synopsis (optional)

Returns detailed library information

Return type dict

Raises `ItemAccessibilityException`, `RequestParameterMissingException`

delete (*self, trans, id, **kwd*)

• **DELETE /api/libraries/{id}** marks the library with the given `id` as *deleted* (or removes the *deleted* mark if the *undelete* param is true)

Note: Currently, only admin users can un/delete libraries.

Parameters

- **id** (*an encoded id string*) – the encoded id of the library to un/delete
- **undelete** (*bool*) – (optional) flag specifying whether the item should be deleted or undeleted, defaults to false:

Returns detailed library information

Return type dictionary

See also:

`galaxy.model.Library.dict_element_visible_keys`

Raises `ItemAccessibilityException`, `MalformedId`, `ObjectNotFound`

get_permissions (*trans, *args, **kwargs*)

• **GET /api/libraries/{id}/permissions**

Load all permissions for the given library id and return it.

Parameters

- **encoded_library_id** (*an encoded id string*) – the encoded id of the library
- **scope** (*string*) – either ‘current’ or ‘available’
- **is_library_access** (*bool*) – indicates whether the roles available for the library access are requested

Returns dictionary with all applicable permissions' values

Return type dictionary

Raises ObjectNotFound, InsufficientPermissionsException

index (*self*, *trans*, ***kwd*)

•**GET /api/libraries:** Returns a list of summary data for all libraries.

Parameters **deleted** (*boolean (optional)*) – if True, show only deleted libraries, if False show only non-deleted

Returns list of dictionaries containing library information

Return type *list*

See also:

galaxy.model.Library.dict_collection_visible_keys

set_permissions (*trans*, **args*, ***kwargs*)

def set_permissions(self, trans, encoded_dataset_id, *kwd*):** *POST
/api/libraries/{encoded_library_id}/permissions

Parameters

- **encoded_library_id** (*an encoded id string*) – the encoded id of the library to set the permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: remove_restrictions, set_permissions
- **access_ids** [] (*string or list*) – list of Role.id defining roles that should have access permission on the library
- **add_ids** [] (*string or list*) – list of Role.id defining roles that should have add item permission on the library
- **manage_ids** [] (*string or list*) – list of Role.id defining roles that should have manage permission on the library
- **modify_ids** [] (*string or list*) – list of Role.id defining roles that should have modify permission on the library

Return type dictionary

Returns dict of current roles for all available permission types

Raises RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException

set_permissions_old (*trans*, *library*, *payload*, ***kwd*)
* old implementation for backward compatibility *

POST /api/libraries/{encoded_library_id}/permissions Updates the library permissions.

show (*self*, *trans*, *id*, *deleted='False'*, ***kwd*)

•**GET /api/libraries/{encoded_id}:** returns detailed information about a library

•**GET /api/libraries/deleted/{encoded_id}:** returns detailed information about a deleted library

Parameters

- **id** (*an encoded id string*) – the encoded id of the library
- **deleted** (*boolean*) – if True, allow information on a `deleted` library

Returns detailed library information

Return type dictionary

See also:

`galaxy.model.Library.dict_element_visible_keys`

Raises `MalformedId`, `ObjectNotFound`

update (*trans, *args, **kwargs*)

- **PATCH /api/libraries/{encoded_id}** Updates the library defined by an `encoded_id` with the data in the payload.

Note: Currently, only admin users can update libraries. Also the library must not be *deleted*.

param id the encoded id of the library

type id an encoded id string

param payload (required) dictionary structure containing:: 'name': new library's name, cannot be empty 'description': new library's description 'synopsis': new library's synopsis

type payload dict

returns detailed library information

rtype dict

raises `ItemAccessibilityException`, `MalformedId`, `ObjectNotFound`, `RequestParamerInvalidException`, `RequestParamerMissingException`

library_contents Module

API operations on the contents of a data library.

class `galaxy.webapps.galaxy.api.library_contents.LibraryContentsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesLibrary`, `galaxy.web.base.controller.UsesLibraryMixinItems`, `galaxy.web.base.controller.UsesHistory`

create (*self, trans, library_id, payload, **kwd*)

- **POST /api/libraries/{library_id}/contents:** create a new library file or folder

To copy an HDA into a library send `create_type` of 'file' and the HDA's encoded id in `from_hda_id` (and optionally `ldda_message`).

Parameters

- **library_id** (*str*) – encoded id string of the library where to create the new item
- **payload** (*dict*) – dictionary structure containing:
 - `folder_id`: the parent folder of the new item
 - `create_type`: the type of item to create ('file', 'folder' or 'collection')

- **from_hda_id:** (optional) the id of an accessible HDA to copy into the library
- **ldda_message:** (optional) the new message attribute of the LDDA created
- **extended_metadata:** (optional) **dict-dictionary** containing any extended metadata to associate with the item
- **link_data_only:** (optional) either ‘copy_files’ (default) or ‘link_to_files’
- **upload_option:** (optional) one of ‘upload_file’ (default), ‘upload_directory’ or ‘upload_paths’
- **server_dir:** (optional) only if upload_option is ‘upload_directory’
- **filesystem_paths:** (optional) only if upload_option is ‘upload_paths’ and the user is an admin

Return type dict

Returns a dictionary containing the id, name, and ‘show’ url of the new item

delete (*self*, *trans*, *library_id*, *id*, ***kwd*)

•**DELETE /api/libraries/{library_id}/contents/{id}** delete the LibraryDataset with the given *id*

Parameters

- **id** (*str*) – the encoded id of the library dataset to delete
- **kwd** (*dict*) – (optional) dictionary structure containing:
 - **payload:** a dictionary itself containing:
 - * **purge:** if True, purge the LD

Return type dict

Returns an error object if an error occurred or a dictionary containing: * **id:** the encoded id of the library dataset, * **deleted:** if the library dataset was marked as deleted, * **purged:** if the library dataset was purged

index (*self*, *trans*, *library_id*, ***kwd*)

•**GET /api/libraries/{library_id}/contents:** Returns a list of library files and folders.

Note: May be slow! Returns all content traversing recursively through all folders.

See also:

`galaxy.webapps.galaxy.api.FolderContentsController.index` for a non-recursive solution

Parameters **library_id** (*string*) – encoded id string of the library

Returns

list of dictionaries of the form: * **id:** the encoded id of the library item * **name:** the ‘library path’

or relationship of the library item to the root

- **type:** ‘file’ or ‘folder’
- **url:** the url to get detailed information on the library item

Return type *list*

Raises MalformedId, InconsistentDatabase, RequestParameterInvalidException, InternalServerError

show (*self*, *trans*, *id*, *library_id*, ***kwd*)

• **GET /api/libraries/{library_id}/contents/{id}** Returns information about library file or folder.

Parameters

- **id** (*str*) – the encoded id of the library item to return
- **library_id** (*str*) – encoded id string of the library that contains this item

Returns detailed library item information

Return type dict

See also:

galaxy.model.LibraryDataset.to_dict() and *galaxy.model.LibraryFolder.dict_element_v1*

update (*self*, *trans*, *id*, *library_id*, *payload*, ***kwd*)

• **PUT /api/libraries/{library_id}/contents/{id}** create a ImplicitlyConvertedDatasetAssociation

See also:

galaxy.model.ImplicitlyConvertedDatasetAssociation

Parameters

- **id** (*str*) – the encoded id of the library item to return
- **library_id** (*str*) – encoded id string of the library that contains this item
- **payload** (*dict*) – dictionary structure containing:: 'converted_dataset_id':

Return type None

Returns None

metrics Module

API operations for for querying and recording user metrics from some client (typically a user's browser).

class *galaxy.webapps.galaxy.api.metrics.MetricsController* (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

create (*trans*, *payload*)

• **POST /api/metrics:** record any metrics sent and return some status object

Note: Anonymous users can post metrics

Parameters **payload** (*dict*) – (optional) dictionary structure containing: * metrics: a list containing dictionaries of the form:

** namespace: label indicating the source of the metric ** time: isoformat datetime when the metric was recorded ** level: an integer representing the metric's log level ** args: a json string containing an array of extra data

Return type dict

Returns status object

debugging = None

set to true to send additional debugging info to the log

page_revisions Module

API for updating Galaxy Pages

```
class galaxy.webapps.galaxy.api.page_revisions.PageRevisionsController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.SharableIt
           galaxy.model.item_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin
```

create (*self*, *trans*, *page_id*, *payload* ****kwd**)

•**POST /api/pages/{page_id}/revisions** Create a new revision for a page

Parameters

- **page_id** – Add revision to Page with ID=page_id
- **payload** – A dictionary containing:: ‘title’ = New title of the page ‘content’ = New content of the page

Return type dictionary

Returns Dictionary with ‘success’ or ‘error’ element to indicate the result of the request

index (*self*, *trans*, *page_id*, ****kwd**)

•**GET /api/pages/{page_id}/revisions** return a list of Page revisions

Parameters **page_id** – Display the revisions of Page with ID=page_id

Return type *list*

Returns dictionaries containing different revisions of the page

pages Module

API for updating Galaxy Pages

```
class galaxy.webapps.galaxy.api.pages.PagesController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.SharableIt
           galaxy.model.item_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin
```

create (*self*, *trans*, *payload*, ****kwd**)

•**POST /api/pages** Create a page and return dictionary containing Page summary

Parameters **payload** – dictionary structure containing:: ‘slug’ = The title slug for the page URL, must be unique ‘title’ = Title of the page ‘content’ = HTML contents of the page ‘annotation’ = Annotation that will be attached to the page

Return type dict

Returns Dictionary return of the Page.to_dict call

delete (*self*, *trans*, *id*, ***kwd*)

•**DELETE /api/pages/{id}** Create a page and return dictionary containing Page summary

Parameters *id* – ID of page to be deleted

Return type dict

Returns Dictionary with ‘success’ or ‘error’ element to indicate the result of the request

index (*self*, *trans*, *deleted=False*, ***kwd*)

•**GET /api/pages** return a list of Pages viewable by the user

Parameters *deleted* – Display deleted pages

Return type *list*

Returns dictionaries containing summary or detailed Page information

show (*self*, *trans*, *id*, ***kwd*)

•**GET /api/pages/{id}** View a page summary and the content of the latest revision

Parameters *id* – ID of page to be displayed

Return type dict

Returns Dictionary return of the Page.to_dict call with the ‘content’ field populated by the most recent revision

provenance Module

API operations provenance

class `galaxy.webapps.galaxy.api.provenance.BaseProvenanceController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesHistory`

create (*trans*, **args*, ***kwargs*)

delete (*trans*, **args*, ***kwargs*)

index (*trans*, **args*, ***kwargs*)

show (*trans*, **args*, ***kwargs*)

class `galaxy.webapps.galaxy.api.provenance.HDAProvenanceController` (*app*)

Bases: `galaxy.webapps.galaxy.api.provenance.BaseProvenanceController`

controller_name = ‘history_content_provenance’

provenance_item_class = ‘HistoryDatasetAssociation’

provenance_item_id = ‘history_content_id’

class `galaxy.webapps.galaxy.api.provenance.LDDAProvenanceController` (*app*)

Bases: `galaxy.webapps.galaxy.api.provenance.BaseProvenanceController`

controller_name = ‘ldda_provenance’

provenance_item_class = ‘LibraryDatasetDatasetAssociation’

provenance_item_id = ‘library_content_id’

quotas Module

API operations on Quota objects.

```
class galaxy.webapps.galaxy.api.quotas.QuotaAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controllers.admin.AdminActions, galaxy.actions.admin.AdminActions, galaxy.web.base.controller.UsesQuotaMixin, galaxy.web.params.QuotaParamParser

    create (trans, *args, **kwargs)
        POST /api/quotas Creates a new quota.

    delete (trans, *args, **kwargs)
        DELETE /api/quotas/{encoded_quota_id} Deletes a quota

    index (trans, *args, **kwargs)
        GET /api/quotas GET /api/quotas/deleted Displays a collection (list) of quotas.

    show (trans, *args, **kwargs)
        GET /api/quotas/{encoded_quota_id} GET /api/quotas/deleted/{encoded_quota_id} Displays information about a quota.

    undelete (trans, *args, **kwargs)
        POST /api/quotas/deleted/{encoded_quota_id}/undelete Undeletes a quota

    update (trans, *args, **kwargs)
        PUT /api/quotas/{encoded_quota_id} Modifies a quota.
```

request_types Module

API operations on RequestType objects.

```
class galaxy.webapps.galaxy.api.request_types.RequestTypeAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create (trans, *args, **kwargs)
        POST /api/request_types Creates a new request type (external_service configuration).

    index (trans, *args, **kwargs)
        GET /api/request_types Displays a collection (list) of request_types.

    show (trans, *args, **kwargs)
        GET /api/request_types/{encoded_request_type_id} Displays information about a request_type.
```

requests Module

API operations on a sample tracking system.

```
class galaxy.webapps.galaxy.api.requests.RequestsAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    index (trans, *args, **kwargs)
        GET /api/requests Displays a collection (list) of sequencing requests.

    show (trans, *args, **kwargs)
        GET /api/requests/{encoded_request_id} Displays details of a sequencing request.

    update (trans, *args, **kwargs)
        PUT /api/requests/{encoded_request_id} Updates a request state, sample state or sample dataset transfer status depending on the update_type
```

```
v = ('REQUEST', 'request_state')
```

roles Module

API operations on Role objects.

```
class galaxy.webapps.galaxy.api.roles.RoleAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create (trans, *args, **kwargs)
        POST /api/roles Creates a new role.

    index (trans, *args, **kwargs)
        GET /api/roles Displays a collection (list) of roles.

    show (trans, *args, **kwargs)
        GET /api/roles/{encoded_role_id} Displays information about a role.
```

samples Module

API operations for samples in the Galaxy sample tracking system.

```
class galaxy.webapps.galaxy.api.samples.SamplesAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    index (trans, *args, **kwargs)
        GET /api/requests/{encoded_request_id}/samples Displays a collection (list) of sample of a sequencing
        request.

    k = 'SAMPLE_DATASET'

    update (trans, *args, **kwargs)
        PUT /api/samples/{encoded_sample_id} Updates a sample or objects related ( mapped ) to a sample.

    update_type_values = ['sample_state', 'run_details', 'sample_dataset_transfer_status']

    update_types = <galaxy.util.bunch.Bunch object>

    v = ['sample_dataset_transfer_status']
```

search Module

API for searching Galaxy Datasets

```
class galaxy.webapps.galaxy.api.search.SearchController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.SharableIt

    create (trans, *args, **kwargs)
        POST /api/search Do a search of the various elements of Galaxy.
```

tool_data Module

```
class galaxy.webapps.galaxy.api.tool_data.ToolData (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    RESTful controller for interactions with tool data
```

index (*trans*, *args, **kwargs)
GET /api/tool_data: returns a list tool_data tables:

show (*trans*, *args, **kwargs)

tool_shed_repositories Module

class galaxy.webapps.galaxy.api.tool_shed_repositories.**ToolShedRepositoriesController** (*app*)
Bases: *galaxy.web.base.controller.BaseAPIController*

RESTful controller for interactions with tool shed repositories.

exported_workflows (*trans*, *args, **kwargs)
GET /api/tool_shed_repositories/{encoded_tool_shed_repository_id}/exported_workflows
Display a list of dictionaries containing information about this tool shed repository's exported workflows.

Parameters **id** – the encoded id of the ToolShedRepository object

get_latest_installable_revision (*trans*, *args, **kwargs)
POST /api/tool_shed_repositories/get_latest_installable_revision Get the latest installable revision of a specified repository from a specified Tool Shed.

Parameters **key** – the current Galaxy admin user's API key

The following parameters are included in the payload. :param tool_shed_url (required): the base URL of the Tool Shed from which to retrieve the Repository revision. :param name (required): the name of the Repository :param owner (required): the owner of the Repository

import_workflow (*trans*, *args, **kwargs)
POST /api/tool_shed_repositories/import_workflow
Import the specified exported workflow contained in the specified installed tool shed repository into Galaxy.

Parameters

- **key** – the API key of the Galaxy user with which the imported workflow will be associated.
- **id** – the encoded id of the ToolShedRepository object

The following parameters are included in the payload. :param index: the index location of the workflow tuple in the list of exported workflows stored in the metadata for the specified repository

import_workflows (*trans*, *args, **kwargs)
POST /api/tool_shed_repositories/import_workflows
Import all of the exported workflows contained in the specified installed tool shed repository into Galaxy.

Parameters

- **key** – the API key of the Galaxy user with which the imported workflows will be associated.
- **id** – the encoded id of the ToolShedRepository object

index (*trans*, *args, **kwargs)
GET /api/tool_shed_repositories Display a list of dictionaries containing information about installed tool shed repositories.

install_repository_revision (*trans*, *args, **kwargs)
POST /api/tool_shed_repositories/install_repository_revision Install a specified repository revision from a specified tool shed into Galaxy.

Parameters **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool_shed_url (required): the base URL of the Tool Shed from which to install the Repository :param name (required): the name of the Repository :param owner (required): the owner of the Repository :param changeset_revision (required): the changeset_revision of the RepositoryMetadata object associated with the Repository :param new_tool_panel_section_label (optional): label of a new section to be added to the Galaxy tool panel in which to load

tools contained in the Repository. Either this parameter must be an empty string or the tool_panel_section_id parameter must be an empty string or both must be an empty string (both cannot be used simultaneously).

Parameters

- **(optional)** (*shed_tool_conf*) – id of the Galaxy tool panel section in which to load tools contained in the Repository. If this parameter is an empty string and the above new_tool_panel_section_label parameter is an empty string, tools will be loaded outside of any sections in the tool panel. Either this parameter must be an empty string or the tool_panel_section_id parameter must be an empty string of both must be an empty string (both cannot be used simultaneously).
- **(optional)** – Set to True if you want to install repository dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – Set to True if you want to install tool dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – The shed-related tool panel configuration file configured in the “tool_config_file” setting in the Galaxy config file (e.g., galaxy.ini). At least one shed-related tool panel config file is required to be configured. Setting this parameter to a specific file enables you to choose where the specified repository will be installed because the tool_path attribute of the <toolbox> from the specified file is used as the installation location (e.g., <toolbox tool_path=”../shed_tools”>). If this parameter is not set, a shed-related tool panel configuration file will be selected automatically.

install_repository_revisions (*trans, *args, **kwargs*)

POST /api/tool_shed_repositories/install_repository_revisions Install one or more specified repository revisions from one or more specified tool sheds into Galaxy. The received parameters must be ordered lists so that positional values in tool_shed_urls, names, owners and changeset_revisions are associated.

It’s questionable whether this method is needed as the above method for installing a single repository can probably cover all desired scenarios. We’ll keep this one around just in case...

Parameters **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool_shed_urls: the base URLs of the Tool Sheds from which to install a specified Repository :param names: the names of the Repositories to be installed :param owners: the owners of the Repositories to be installed :param changeset_revisions: the changeset_revisions of each RepositoryMetadata object associated with each Repository to be installed :param new_tool_panel_section_label: optional label of a new section to be added to the Galaxy tool panel in which to load

tools contained in the Repository. Either this parameter must be an empty string or the tool_panel_section_id parameter must be an empty string, as both cannot be used.

Parameters

- **tool_panel_section_id** – optional id of the Galaxy tool panel section in which to load tools contained in the Repository. If not set, tools will be loaded outside of any sections in the tool panel. Either this parameter must be an empty string or the `tool_panel_section_id` parameter must be an empty string, as both cannot be used.
- **(optional) (shed_tool_conf)** – Set to True if you want to install repository dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – Set to True if you want to install tool dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – The shed-related tool panel configuration file configured in the “tool_config_file” setting in the Galaxy config file (e.g., galaxy.ini). At least one shed-related tool panel config file is required to be configured. Setting this parameter to a specific file enables you to choose where the specified repository will be installed because the `tool_path` attribute of the <toolbox> from the specified file is used as the installation location (e.g., <toolbox tool_path=”../shed_tools”>). If this parameter is not set, a shed-related tool panel configuration file will be selected automatically.

repair_repository_revision (*trans*, *args, **kwargs)

POST /api/tool_shed_repositories/repair_repository_revision Repair a specified repository revision previously installed into Galaxy.

Parameters **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool_shed_url (required): the base URL of the Tool Shed from which the Repository was installed :param name (required): the name of the Repository :param owner (required): the owner of the Repository :param changeset_revision (required): the changeset_revision of the RepositoryMetadata object associated with the Repository

reset_metadata_on_installed_repositories (*trans*, *args, **kwargs)

PUT /api/tool_shed_repositories/reset_metadata_on_installed_repositories

Resets all metadata on all repositories installed into Galaxy in an “orderly fashion”.

Parameters **key** – the API key of the Galaxy admin user.

show (*trans*, *args, **kwargs)

GET /api/tool_shed_repositories/{encoded_tool_shed_repository_id} Display a dictionary containing information about a specified tool_shed_repository.

Parameters **id** – the encoded id of the ToolShedRepository object

```
galaxy.webapps.galaxy.api.tool_shed_repositories.get_message_for_no_shed_tool_config()
```

tools Module

users Module

API operations on User objects.

class galaxy.webapps.galaxy.api.users.**UserAPIController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesTagsMixin, galaxy.web.base.controller.CreatesUsersMixin, galaxy.web.base.controller.CreatesApiKeysMixin*

anon_user_api_value (*trans*)
Returns data for an anonymous user, truncated to only usage and quota_percent

api_key (*trans*, **args*, ***kwargs*)
POST /api/users/{encoded_user_id}/api_key Creates a new API key for specified user.

create (*trans*, **args*, ***kwargs*)
POST /api/users Creates a new Galaxy user.

delete (*trans*, **args*, ***kwargs*)

index (*trans*, **args*, ***kwargs*)
GET /api/users GET /api/users/deleted Displays a collection (list) of users.

show (*trans*, **args*, ***kwargs*)
GET /api/users/{encoded_user_id} GET /api/users/deleted/{encoded_user_id} GET /api/users/current
Displays information about a user.

undelete (*trans*, **args*, ***kwargs*)

update (*trans*, **args*, ***kwargs*)

visualizations Module

Visualizations resource control over the API.

NOTE!: this is a work in progress and functionality and data structures may change often.

class galaxy.webapps.galaxy.api.visualizations.**VisualizationsController** (*app*)
Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesVisualizations, galaxy.web.base.controller.SharableMixin, galaxy.model.item_attrs.UsesAnnotations*

RESTful controller for interactions with visualizations.

create (*trans*, **args*, ***kwargs*)
POST /api/visualizations creates a new visualization using the given payload

POST /api/visualizations?import_id={encoded_visualization_id} imports a copy of an existing visualization into the user's workspace

index (*trans*, **args*, ***kwargs*)
GET /api/visualizations:

show (*trans*, **args*, ***kwargs*)
GET /api/visualizations/{viz_id}

update (*trans*, **args*, ***kwargs*)
PUT /api/visualizations/{encoded_visualization_id}

workflows Module

API operations for Workflows

class galaxy.webapps.galaxy.api.workflows.**WorkflowsAPIController** (*app*)
Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesStore, galaxy.web.base.controller.UsesHistoryMixin, galaxy.model.item_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin*

cancel_workflow_invocation (*trans*, **args*, ***kwargs*)
DELETE /api/workflows/{workflow_id}/usage/{usage_id} Cancel the specified workflow invocation.

Parameters

- **workflow_id** (*str*) – the workflow id (required)
- **usage_id** (*str*) – the usage id (required)

Raises exceptions.MessageException, exceptions.ObjectNotFound

create (*trans*, **args*, ***kwargs*)

POST /api/workflows

Run or create workflows from the api.

If installed_repository_file or from_history_id is specified a new workflow will be created for this user. Otherwise, workflow_id must be specified and this API method will cause a workflow to execute.

:param installed_repository_file The path of a workflow to import. Either workflow_id, installed_repository_file or from_history_id must be specified :type installed_repository_file str

Parameters

- **workflow_id** (*str*) – An existing workflow id. Either workflow_id, installed_repository_file or from_history_id must be specified
- **parameters** (*dict*) – If workflow_id is set - see _update_step_parameters()
- **ds_map** (*dict*) – If workflow_id is set - a dictionary mapping each input step id to a dictionary with 2 keys: 'src' (which can be 'ldda', 'ld' or 'hda') and 'id' (which should be the id of a LibraryDatasetDatasetAssociation, LibraryDataset or HistoryDatasetAssociation respectively)
- **no_add_to_history** (*str*) – If workflow_id is set - if present in the payload with any value, the input datasets will not be added to the selected history
- **history** (*str*) – If workflow_id is set - optional history where to run the workflow, either the name of a new history or "hist_id=HIST_ID" where HIST_ID is the id of an existing history. If not specified, the workflow will be run a new unnamed history
- **replacement_params** (*dict*) – If workflow_id is set - an optional dictionary used when renaming datasets
- **from_history_id** (*str*) – Id of history to extract a workflow from. Either workflow_id, installed_repository_file or from_history_id must be specified
- **job_ids** (*str*) – If from_history_id is set - optional list of jobs to include when extracting a workflow from history
- **dataset_ids** (*str*) – If from_history_id is set - optional list of HDA 'hid's corresponding to workflow inputs when extracting a workflow from history
- **dataset_collection_ids** (*str*) – If from_history_id is set - optional list of HDCA 'hid's corresponding to workflow inputs when extracting a workflow from history
- **workflow_name** (*str*) – If from_history_id is set - name of the workflow to create when extracting a workflow from history

delete (*trans*, **args*, ***kwargs*)

DELETE /api/workflows/{encoded_workflow_id} Deletes a specified workflow Author: rpark

copied from galaxy.web.controllers.workflows.py (delete)

import_new_workflow_deprecated (*trans*, **args*, ***kwargs*)

POST /api/workflows/upload Importing dynamic workflows from the api. Return newly generated workflow id. Author: rpark

currently assumes payload['workflow'] is a json representation of a workflow to be inserted into the database

Deprecated in favor to POST /api/workflows with encoded 'workflow' in payload the same way.

import_shared_workflow_deprecated (*trans*, **args*, ***kwargs*)

POST /api/workflows/import Import a workflow shared by other users.

Parameters **workflow_id** (*str*) – the workflow id (required)

Raises exceptions.MessageException, exceptions.ObjectNotFound

index (*trans*, **args*, ***kwargs*)

GET /api/workflows

Displays a collection of workflows.

Parameters **show_published** (*boolean*) – if True, show also published workflows

show (*trans*, **args*, ***kwargs*)

GET /api/workflows/{encoded_workflow_id}

Displays information needed to run a workflow from the command line.

workflow_dict (*trans*, **args*, ***kwargs*)

GET /api/workflows/{encoded_workflow_id}/download Returns a selected workflow as a json dictionary.

workflow_invocation_step (*trans*, **args*, ***kwargs*)

GET /api/workflows/{ workflow_id }/usage/{ usage_id }/steps/{ step_id }

Parameters

- **workflow_id** (*str*) – the workflow id (required)
- **usage_id** (*str*) – the usage id (required)
- **step_id** (*str*) – encoded id of the WorkflowInvocationStep (required)
- **payload** – payload containing update action information for running workflow.

Raises exceptions.MessageException, exceptions.ObjectNotFound

workflow_invocation_step_update (*trans*, **args*, ***kwargs*)

PUT /api/workflows/{ workflow_id }/usage/{ usage_id }/steps/{ step_id } Update state of running workflow step invocation - still very nebulous but this would be for stuff like confirming paused steps can proceed etc....

Parameters

- **workflow_id** (*str*) – the workflow id (required)
- **usage_id** (*str*) – the usage id (required)
- **step_id** (*str*) – encoded id of the WorkflowInvocationStep (required)

Raises exceptions.MessageException, exceptions.ObjectNotFound

workflow_request (*trans*, **args*, ***kwargs*)

POST /api/workflows/{encoded_workflow_id}/usage

Schedule the workflow specified by *workflow_id* to run.

workflow_usage (*trans*, **args*, ***kwargs*)

GET /api/workflows/{ workflow_id }/usage Get the list of the workflow usage

Parameters **workflow_id** (*str*) – the workflow id (required)

Raises exceptions.MessageException, exceptions.ObjectNotFound

workflow_usage_contents (*trans*, **args*, ***kwargs*)
GET /api/workflows/{workflow_id}/usage/{usage_id} Get detailed description of workflow usage

Parameters

- **workflow_id** (*str*) – the workflow id (required)
- **usage_id** (*str*) – the usage id (required)

Raises exceptions.MessageException, exceptions.ObjectNotFound

1.2 lib

1.2.1 fpconst Module

1.2.2 galaxy Package

galaxy Package

Galaxy root package – this is a namespace package.

app Module

config Module

Universe configuration builder.

```
class galaxy.config.Configuration (**kwargs)
    Bases: object

    check ()

    deprecated_options = ('database_file',)

    ensure_tempdir ()

    get (key, default)

    get_bool (key, default)

    guess_galaxy_port ()

    is_admin_user (user)
        Determine if the provided user is listed in admin_users.

        NOTE: This is temporary, admin users will likely be specified in the database in the future.

    resolve_path (path)
        Resolve a path relative to Galaxy's root.

    sentry_dsn_public
        Sentry URL with private key removed for use in client side scripts, sentry server will need to be configured to accept events

exception galaxy.config.ConfigurationError
    Bases: exceptions.Exception

class galaxy.config.ConfiguresGalaxyMixin
    Shared code for configuring Galaxy-like app objects.
```

galaxy.config.configure_logging(*config*)

Allow some basic logging configuration to be read from ini file.

```
galaxy.config.get_database_engine_options(kwargs, model_prefix='')

```

Allow options for the SQLAlchemy database engine to be passed by using the prefix “database_engine_option”.

galaxy.config.resolve_path(*path*, *root*)

If 'path' is relative make absolute by prepending 'root'

Subpackages

actions Package

admin Module	Contains administrative functions
---------------------	-----------------------------------

```
class galaxy.actions.admin.AdminActions
```

Bases: object

Mixin for controllers that provide administrative functionality.

datatypes Package

assembly Module velvet datatypes James E Johnson - University of Minnesota for velvet assembler tool in galaxy

```
class galaxy.datatypes.assembly.Amos (**kwd)
```

Bases: `galaxy.datatypes.data.Text`

Class describing the AMOS assembly file

```
file_ext = 'afg'
```

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number

sniff (*filename*)

Determines whether the file is an amos assembly file format Example:

```
{CTG  
iid:1  
eid:1  
seq:  
CCTCTCCTGTAGAGTTCAACCGA-GCCGGTAGAGTTTTATCA  
.br/>qlt:  
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD  
.br/>{TLE  
src:1027  
off:0  
clr:618,0  
gap:  
250 612  
.br/>}  
}
```

```
class galaxy.datatypes.assembly.Roadmaps (**kwd)
```

Bases: `galaxy.datatypes.data.Text`

Class describing the Sequences file generated by velveth

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of lines

sniff (filename)

Determines whether the file is a velveth produced RoadMap:: 142858 21 1 ROADMAP 1
ROADMAP 2 ...

class galaxy.datatypes.assembly.Sequences (**kwd)

Bases: *galaxy.datatypes.sequence.Fasta*

Class describing the Sequences file generated by velveth

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of lines

sniff (filename)

Determines whether the file is a velveth produced fasta format The id line has 3 fields separated by tabs:
sequence_name sequence_index category:

```
>SEQUENCE_0_length_35 1 1
GGATATAGGGCCAACCACTCAACGGCCTGTCTT
>SEQUENCE_1_length_35 2 1
CGACGAATGACAGGTCACGAATTTGGCGGGGATTA
```

class galaxy.datatypes.assembly.Velvet (**kwd)

Bases: *galaxy.datatypes.images.Html*

allow_datatype_change = False

composite_type = 'auto_primary_file'

file_ext = 'html'

generate_primary_file (dataset=None)

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of lines

regenerate_primary_file (dataset)

cannot do this until we are setting metadata

set_meta (dataset, **kwd)

binary Module Binary classes

class galaxy.datatypes.binary.Ab1 (**kwd)

Bases: *galaxy.datatypes.binary.Binary*

Class describing an ab1 binary sequence file

display_peek (dataset)

file_ext = 'ab1'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

set_peek (dataset, is_multi_byte=False)

class galaxy.datatypes.binary.Bam (**kwd)

Bases: *galaxy.datatypes.binary.Binary*

Class describing a BAM binary file

column_dataprovider (*args, **kwargs)

data_sources = {'index': 'bigwig', 'data': 'bai'}

```

dataproviders = {'chunk64': <function chunk64_dataprovider at 0x7fc0780c7668>, 'id-seq-qual': <function id_seq_qual_dataprovider at 0x7fc0780c7668>}
dataset_content_needs_grooming (file_name)
    See if file_name is a sorted BAM file
dict_dataprovider (*args, **kwargs)
display_peek (dataset)
file_ext = 'bam'
genomic_region_dataprovider (*args, **kwargs)
genomic_region_dict_dataprovider (*args, **kwargs)
groom_dataset_content (file_name)
    Ensures that the Bam file contents are sorted. This function is called on an output dataset after the content
    is initially generated.
header_dataprovider (*args, **kwargs)
id_seq_qual_dataprovider (*args, **kwargs)
init_meta (dataset, copy_from=None)
line_dataprovider (*args, **kwargs)
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', bam_index (FileParameter): BAM Index
regex_line_dataprovider (*args, **kwargs)
samtools_dataprovider (*args, **kwargs)
    Generic samtools interface - all options available through settings.
set_meta (dataset, overwrite=True, **kwd)
    Creates the index for the BAM file.
set_peek (dataset, is_multi_byte=False)
sniff (filename)
track_type = 'ReadTrack'

class galaxy.datatypes.binary.Bcf (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class describing a BCF file
    file_ext = 'bcf'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    sniff (filename)

class galaxy.datatypes.binary.BigBed (**kwd)
    Bases: galaxy.datatypes.binary.BigWig
    BigBed support from UCSC.
    data_sources = {'data_standalone': 'bigbed'}
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

class galaxy.datatypes.binary.BigWig (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Accessing binary BigWig files from UCSC. The supplemental info in the paper has the binary details:
    http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btq351v1

```

```
data_sources = {'data_standalone': 'bigwig'}

display_peek (dataset)

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

set_peek (dataset, is_multi_byte=False)

sniff (filename)

track_type = 'LineTrack'

class galaxy.datatypes.binary.Binary (**kwd)
    Bases: galaxy.datatypes.data.Data
    Binary data

    display_data (trans, dataset, preview=False, filename=None, to_ext=None, size=None, off-
        set=None, **kwd)

    get_mime ()
        Returns the mime type of the datatype

    static is_ext_unsniffable (ext)

    static is_sniffable_binary (filename)

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    static register_sniffable_binary_format (data_type, ext, type_class)

    static register_unsniffable_binary_ext (ext)

    set_peek (dataset, is_multi_byte=False)
        Set the peek and blurb text

    sniffable_binary_formats = [{'ext': 'bam', 'type': 'bam', 'class': <class 'galaxy.datatypes.binary.Bam'>}, {'ext':
    unsniffable_binary_formats = ['ab1', 'asn1-binary', 'h5', 'scf']

class galaxy.datatypes.binary.GeminiSQLite (**kwd)
    Bases: galaxy.datatypes.binary.SQLite
    Class describing a Gemini Sqlite database

    display_peek (dataset)

    file_ext = 'gemini.sqlite'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', tables (ListParameter): Database Tables,

    set_meta (dataset, overwrite=True, **kwd)

    set_peek (dataset, is_multi_byte=False)

    sniff (filename)

class galaxy.datatypes.binary.GenericAsn1Binary (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class for generic ASN.1 binary format

    file_ext = 'asn1-binary'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

class galaxy.datatypes.binary.H5 (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class describing an HDF5 file
```



```

    display_peek (dataset)
    file_ext = 'h5'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    set_peek (dataset, is_multi_byte=False)
class galaxy.datatypes.binary.SQLite (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class describing a Sqlite database
    dataproviders = {'chunk64': <function chunk64_dataprovider at 0x7fc0780c7668>, 'base': <function base_dataprovider at 0x7fc0780c7668>}
    display_peek (dataset)
    file_ext = 'sqlite'
    init_meta (dataset, copy_from=None)
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', tables (ListParameter): Database Tables,
    set_meta (dataset, overwrite=True, **kwd)
    set_peek (dataset, is_multi_byte=False)
    sniff (filename)
    sqlite_dataprovider (*args, **kwargs)
class galaxy.datatypes.binary.Scf (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class describing an scf binary sequence file
    display_peek (dataset)
    file_ext = 'scf'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    set_peek (dataset, is_multi_byte=False)
class galaxy.datatypes.binary.Sff (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Standard Flowgram Format (SFF)
    display_peek (dataset)
    file_ext = 'sff'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    set_peek (dataset, is_multi_byte=False)
    sniff (filename)
class galaxy.datatypes.binary.TwoBit (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class describing a TwoBit format nucleotide file
    display_peek (dataset)
    file_ext = 'twobit'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

```

```
set_peek (dataset, is_multi_byte=False)
```

```
sniff (filename)
```

```
class galaxy.datatypes.binary.Xlsx (**kwd)  
    Bases: galaxy.datatypes.binary.Binary
```

Class for Excel 2007 (xlsx) files

```
file_ext = 'xlsx'
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
```

```
sniff (filename)
```

checkers Module

```
galaxy.datatypes.checkers.check_binary (name, file_path=True)
```

```
galaxy.datatypes.checkers.check_bz2 (file_path)
```

```
galaxy.datatypes.checkers.check_gzip (file_path)
```

```
galaxy.datatypes.checkers.check_html (file_path, chunk=None)
```

```
galaxy.datatypes.checkers.check_image (file_path)
```

```
galaxy.datatypes.checkers.check_zip (file_path)
```

```
galaxy.datatypes.checkers.is_bz2 (file_path)
```

```
galaxy.datatypes.checkers.is_gzip (file_path)
```

chrominfo Module

```
class galaxy.datatypes.chrominfo.ChromInfo (**kwd)
```

```
    Bases: galaxy.datatypes.tabular.Tabular
```

```
file_ext = 'len'
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
```

coverage Module Coverage datatypes

```
class galaxy.datatypes.coverage.LastzCoverage (**kwd)
```

```
    Bases: galaxy.datatypes.tabular.Tabular
```

```
file_ext = 'coverage'
```

```
get_track_resolution (dataset, start, end)
```

```
get_track_window (dataset, data, start, end)
```

Assumes we have a numpy file.

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
```

data Module

```
class galaxy.datatypes.data.Data (**kwd)
```

```
    Bases: object
```

Base class for all datatypes. Implements basic interfaces as well as class methods for metadata.

```

>>> class DataTest( Data ):
...     MetadataElement( name="test" )
...
>>> DataTest.metadata_spec.test.name
'test'
>>> DataTest.metadata_spec.test.desc
'test'
>>> type( DataTest.metadata_spec.test.param )
<class 'galaxy.datatypes.metadata.MetadataParameter'>

```

CHUNKABLE = False

add_composite_file (*name*, ***kwds*)

add_display_app (*app_id*, *label*, *file_function*, *links_function*)

Adds a display app to the datatype. *app_id* is a unique id label is the primary display label, e.g., display at 'UCSC' *file_function* is a string containing the name of the function that returns a properly formatted display *links_function* is a string containing the name of the function that returns a list of (*link_name*,*link*)

add_display_application (*display_application*)

New style display applications

after_setting_metadata (*dataset*)

This function is called on the dataset after metadata is set.

allow_datatype_change = True

as_display_type (*dataset*, *type*, ***kwd*)

Returns modified file contents for a particular display type

base_dataprovider (**args*, ***kwargs*)

before_setting_metadata (*dataset*)

This function is called on the dataset before metadata is set.

chunk64_dataprovider (**args*, ***kwargs*)

chunk_dataprovider (**args*, ***kwargs*)

clear_display_apps ()

composite_files = {}

composite_type = None

convert_dataset (*trans*, *original_dataset*, *target_type*, *return_output=False*, *visible=True*, *deps=None*, *set_output_history=True*)

This function adds a job to the queue to convert a dataset to another type. Returns a message about success/failure.

copy_safe_peek = True

data_sources = {}

dataprovider (*dataset*, *data_format*, ***settings*)

Base dataprovider factory for all datatypes that returns the proper provider for the given *data_format* or raises a *NoProviderAvailable*.

dataproviders = {'chunk64': <function chunk64_dataprovider at 0x7fc0780c7668>, 'base': <function base_dataprovider at 0x7fc0780c7668>}

dataset_content_needs_grooming (*file_name*)

This function is called on an output dataset file after the content is initially generated.

display_data (*trans, data, preview=False, filename=None, to_ext=None, size=None, offset=None, **kwd*)

Old display method, for transition - though still used by API and test framework. Datatypes should be very careful if overriding this method and this interface between datatypes and Galaxy will likely change.

TODO: Document alternatives to overriding this method (data providers?).

display_info (*dataset*)

Returns formatted html of dataset info

display_name (*dataset*)

Returns formatted html of dataset name

display_peek (*dataset*)

Create HTML table, used for displaying peek

find_conversion_destination (*dataset, accepted_formats, datatypes_registry, **kwd*)

Returns (target_ext, existing converted dataset)

generate_auto_primary_file (*dataset=None*)

get_composite_files (*dataset=None*)

get_converter_types (*original_dataset, datatypes_registry*)

Returns available converters by type for this dataset

get_display_application (*key, default=None*)

get_display_applications_by_dataset (*dataset, trans*)

get_display_label (*type*)

Returns primary label for display app

get_display_links (*dataset, type, app, base_url, target_frame='_blank', **kwd*)

Returns a list of tuples of (name, link) for a particular display type. No check on 'access' permissions is done here - if you can view the dataset, you can also save it or send it to a destination outside of Galaxy, so Galaxy security restrictions do not apply anyway.

get_display_types ()

Returns display types available

get_max_optional_metadata_filesize ()

get_mime ()

Returns the mime type of the datatype

get_raw_data (*dataset*)

Returns the full data. To stream it open the file_name and read/write as needed

get_visualizations (*dataset*)

Returns a list of visualizations for datatype.

groom_dataset_content (*file_name*)

This function is called on an output dataset file if dataset_content_needs_grooming returns True.

has_dataprovider (*data_format*)

Returns True if *data_format* is available in *dataproviders*.

has_resolution

init_meta (*dataset, copy_from=None*)

is_binary = True

matches_any (*target_datatypes*)

Check if this datatype is of any of the target_datatypes or is a subtype thereof.

```

max_optional_metadata_filesize

static merge (split_files, output_file)
    Merge files with copy.copyfileobj() will not hit the max argument limitation of cat. gz and bz2 files are
    also working.

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    dictionary of metadata fields for this datatype:

missing_meta (dataset, check=[], skip=[])
    Checks for empty metadata values, Returns True if non-optional metadata is missing Specifying a list of
    'check' values will only check those names provided; when used, optionality is ignored Specifying a list
    of 'skip' items will return True even when a named metadata value is missing

primary_file_name = 'index'

remove_display_app (app_id)
    Removes a display app from the datatype

repair_methods (dataset)
    Unimplemented method, returns dict with method/option for repairing errors

set_max_optional_metadata_filesize (max_value)

set_meta (dataset, overwrite=True, **kwd)
    Unimplemented method, allows guessing of metadata from contents of file

set_peek (dataset, is_multi_byte=False)
    Set the peek and blurb text

set_raw_data (dataset, data)
    Saves the data on the disc

supported_display_apps = {}

track_type = None

validate (dataset)
    Unimplemented validate, return no exceptions

writable_files

write_from_stream (dataset, stream)
    Writes data from a stream

class galaxy.datatypes.data.DataMeta (name, bases, dict_)
    Bases: type

    Metaclass for Data class. Sets up metadata spec.

class galaxy.datatypes.data.GenericAsn1 (**kwd)
    Bases: galaxy.datatypes.data.Text

    Class for generic ASN.1 text format

    file_ext = 'asn1'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.data.LineCount (**kwd)
    Bases: galaxy.datatypes.data.Text

    Dataset contains a single line with a single integer that denotes the line count for a related dataset. Used for
    custom builds.

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

```

```
class galaxy.datatypes.data.Newick (**kwd)
    Bases: galaxy.datatypes.data.Text

    New Hampshire/Newick Format

    file_ext = 'nhx'

    get_visualizations (dataset)
        Returns a list of visualizations for datatype.

    init_meta (dataset, copy_from=None)

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    sniff (filename)
        Returning false as the newick format is too general and cannot be sniffed.

class galaxy.datatypes.data.Nexus (**kwd)
    Bases: galaxy.datatypes.data.Text

    Nexus format as used By Paup, Mr Bayes, etc

    file_ext = 'nex'

    get_visualizations (dataset)
        Returns a list of visualizations for datatype.

    init_meta (dataset, copy_from=None)

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    sniff (filename)
        All Nexus Files Simply puts a '#NEXUS' in its first line

class galaxy.datatypes.data.Text (**kwd)
    Bases: galaxy.datatypes.data.Data

    count_data_lines (dataset)
        Count the number of lines of data in dataset, skipping all blank lines and comments.

    dataproviders = {'chunk64': <function chunk64_dataprovider at 0x7fc0780c7668>, 'base': <function base_dataprovider at 0x7fc0780c7668>}
    estimate_file_lines (dataset)
        Perform a rough estimate by extrapolating number of lines from a small read.

    file_ext = 'txt'

    get_mime ()
        Returns the mime type of the datatype

    line_class = 'line'
        Add metadata elements

    line_dataprovider (*args, **kwargs)
        Returns an iterator over the dataset's lines (that have been 'strip'ed) optionally excluding blank lines and
        lines that start with a comment character.

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    regex_line_dataprovider (*args, **kwargs)
        Returns an iterator over the dataset's lines optionally including/excluding lines that match one or more
        regex filters.

    set_meta (dataset, **kwd)
        Set the number of lines of data in dataset.
```

set_peek (*dataset*, *line_count=None*, *is_multi_byte=False*, *WIDTH=256*, *skipchars=[]*)
Set the peek. This method is used by various subclasses of Text.

set_raw_data (*dataset*, *data*)
Saves the data on the disc

classmethod split (*input_datasets*, *subdir_generator_function*, *split_params*)
Split the input files by line.

write_from_stream (*dataset*, *stream*)
Writes data from a stream

`galaxy.datatypes.data.get_file_peek` (*file_name*, *is_multi_byte=False*, *WIDTH=256*,
LINE_COUNT=5, *skipchars=[]*)
Returns the first LINE_COUNT lines wrapped to WIDTH

```
## >>> fname = get_test_fname('4.bed') ## >>> get_file_peek(fname) ## 'chr22 30128507
31828507 uc003bnx.1_cds_2_0_chr22_29227_f0 +
```

`galaxy.datatypes.data.get_test_fname` (*fname*)
Returns test data filename

genetics Module rgenetics datatypes Use at your peril Ross Lazarus for the rgenetics and galaxy projects

genome graphs datatypes derived from Interval datatypes genome graphs datasets have a header row with appropriate columnnames The first column is always the marker - eg columnname = rs, first row= rs12345 if the rows are snps subsequent row values are all numeric ! Will fail if any non numeric (eg '+' or 'NA') values ross lazarus for rgenetics august 20 2007

class `galaxy.datatypes.genetics.Affybatch` (**kwd)
Bases: `galaxy.datatypes.genetics.RexpBase`
derived class for BioC data structures in Galaxy

file_ext = 'affybatch'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class `galaxy.datatypes.genetics.Eigenstratgeno` (**kwd)
Bases: `galaxy.datatypes.genetics.Rgenetics`

Eigenstrat format - may be able to get rid of this if we move to shellfish Rgenetics data collections

file_ext = 'eigenstratgeno'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class `galaxy.datatypes.genetics.Eigenstratpca` (**kwd)
Bases: `galaxy.datatypes.genetics.Rgenetics`

Eigenstrat PCA file for case control adjustment Rgenetics data collections

file_ext = 'eigenstratpca'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class `galaxy.datatypes.genetics.Eset` (**kwd)
Bases: `galaxy.datatypes.genetics.RexpBase`

derived class for BioC data structures in Galaxy

file_ext = 'eset'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

```
class galaxy.datatypes.genetics.Fped (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    FBAT pedigree format - single file, map is header row of rs numbers. Strange. Rgenetics data collections

    file_ext = 'fped'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.genetics.Fphe (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    fbat pedigree file - mad format with ! as first char on header row Rgenetics data collections

    file_ext = 'fphe'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.genetics.GenomeGraphs (**kwd)
    Bases: galaxy.datatypes.tabular.Tabular

    Tab delimited data containing a marker id and any number of numeric values

    as_ucsc_display_file (dataset, **kwd)
        Returns file

    file_ext = 'gg'

    get_mime ()
        Returns the mime type of the datatype

    make_html_table (dataset, skipchars=[])
        Create HTML table, used for displaying peek

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

    set_meta (dataset, **kwd)

    sniff (filename)
        Determines whether the file is in gg format

    ucsc_links (dataset, type, app, base_url)
        from the ever-helpful angie hinrichs angie@soe.ucsc.edu a genome graphs call looks like this

        http://genome.ucsc.edu/cgi-bin/hgGenome?clade=mammal&org=Human&db=hg18&hgGenome\_dataSetName=dname&hgGenome\_dataSetDescription=test&hgGenome\_formatType=best%20guess&hgGenome\_markerType=best%20guess&hgGenome\_columnLabels=best%20guess&hgGenome\_maxVal=&hgGenome\_labelVals=&hgGenome\_maxGapToFill=25000000&hgGenome\_uploadFile=http://galaxy.esphealth.org/datasets/333/display/index&hgGenome\_doSubmitUpload=submit

        Galaxy gives this for an interval file

        http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg18&position=chr1:1-1000&hgt.customText=http%3A%2F%2Fgalaxy.esphealth.org%2Fdisplay\_as%3Fid%3D339%26display\_app%3Ducsc

    validate (dataset)
        Validate a gg file - all numeric after header row

class galaxy.datatypes.genetics.Lped (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    linkage pedigree (ped,map) Rgenetics data collections

    file_ext = 'lped'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
```



```

class galaxy.datatypes.genetics.MAlist (**kwd)
    Bases: galaxy.datatypes.genetics.RexpBase
    derived class for BioC data structures in Galaxy
    file_ext = 'malist'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.genetics.Pbed (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics
    Plink Binary compressed 2bit/geno Rgenetics data collections
    file_ext = 'pbed'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.genetics.Phe (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics
    Phenotype file
    file_ext = 'phe'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.genetics.Pheno (**kwd)
    Bases: galaxy.datatypes.tabular.Tabular
    base class for pheno files
    file_ext = 'pheno'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.genetics.Pphe (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics
    Plink phenotype file - header must have FID IID... Rgenetics data collections
    file_ext = 'pphe'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.genetics.RexpBase (**kwd)
    Bases: galaxy.datatypes.images.Html
    base class for BioC data structures in Galaxy must be constructed with the pheno data in place since that goes
    into the metadata for each instance
    allow_datatype_change = False
    composite_type = 'auto_primary_file'
    display_peek (dataset)
        Returns formatted html of peek
    file_ext = 'rexpbase'
    generate_primary_file (dataset=None)
        This is called only at upload to write the html file cannot rename the datasets here - they come with the
        default unfortunately
    get_file_peek (filename)
        can't really peek at a filename - need the extra_files_path and such?

```

```
get_mime ()
    Returns the mime type of the datatype

get_peek (dataset)
    expects a .pheno file in the extra_files_dir - ugh

get_phecols (phenolist=[], maxConc=20)
    sept 2009: cannot use whitespace to split - make a more complex structure here and adjust the methods
    that rely on this structure return interesting phenotype column names for an reexpression eset or affybatch
    to use in array subsetting and so on. Returns a data structure for a dynamic Galaxy select parameter. A
    column with only 1 value doesn't change, so is not interesting for analysis. A column with a different
    value in every row is equivalent to a unique identifier so is also not interesting for anova or limma analysis
    - both these are removed after the concordance (count of unique terms) is constructed for each column.
    Then a complication - each remaining pair of columns is tested for redundancy - if two columns are always
    paired, then only one is needed :)

get_pheno (dataset)
    expects a .pheno file in the extra_files_dir - ugh note that R is wierd and adds the row.name in
    the header so the columns are all wrong - unless you tell it not to. A file can be written as
    write.table(file='foo.pheno',pData(foo),sep=' ',quote=F,row.names=F)

html_table = None

init_meta (dataset, copy_from=None)

is_binary = True

make_html_table (pp='nothing supplied from peek\n')
    Create HTML table, used for displaying peek

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

regenerate_primary_file (dataset)
    cannot do this until we are setting metadata

set_meta (dataset, **kwd)
    NOTE we apply the tabular machinery to the phenodata extracted from a BioC eSet or affybatch.

set_peek (dataset, **kwd)
    expects a .pheno file in the extra_files_dir - ugh note that R is weird and does not include the row.name in
    the header. why?

class galaxy.datatypes.genetics.Rgenetics (**kwd)
    Bases: galaxy.datatypes.images.Html

    base class to use for rgenetics datatypes derived from html - composite datatype elements stored in extra files
    path

    allow_datatype_change = False

    composite_type = 'auto_primary_file'

    file_ext = 'rgenetics'

    generate_primary_file (dataset=None)

    get_mime ()
        Returns the mime type of the datatype

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

    regenerate_primary_file (dataset)
        cannot do this until we are setting metadata
```

```

    set_meta (dataset, **kwd)
        for lped/pbed eg
class galaxy.datatypes.genetics.SNPMatrix (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    BioC SNPMatrix Rgenetics data collections

    file_ext = 'snpmatrix'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    set_peek (dataset, **kwd)

    sniff (filename)
        need to check the file header hex code
class galaxy.datatypes.genetics.Snpptest (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    BioC snptest Rgenetics data collections

    file_ext = 'snptest'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
class galaxy.datatypes.genetics.ldIndep (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    LD (a good measure of redundancy of information) depleted Plink Binary compressed 2bit/geno This is really a
    plink binary, but some tools work better with less redundancy so are constrained to these files

    file_ext = 'ldreduced'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
class galaxy.datatypes.genetics.rgFeatureList (**kwd)
    Bases: galaxy.datatypes.genetics.rgTabList

    for featureid lists of exclusions or inclusions in the clean tool output from QC eg low maf, high missingness,
    bad hwe in controls, excess mendel errors,... featureid subsets on statistical criteria -> specialized display such
    as gg same infrastructure for expression?

    file_ext = 'rgFList'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
class galaxy.datatypes.genetics.rgSampleList (**kwd)
    Bases: galaxy.datatypes.genetics.rgTabList

    for sampleid exclusions or inclusions in the clean tool output from QC eg excess het, gender error, ibd pair mem-
    ber,eigen outlier,excess mendel errors,... since they can be uploaded, should be flexible but they are persistent
    at least same infrastructure for expression?

    file_ext = 'rgSList'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    sniff (filename)
class galaxy.datatypes.genetics.rgTabList (**kwd)
    Bases: galaxy.datatypes.tabular.Tabular

    for sampleid and for featureid lists of exclusions or inclusions in the clean tool featureid subsets on statistical
    criteria -> specialized display such as gg

```

display_peek (*dataset*)

Returns formatted html of peek

file_ext = 'rgTList'

get_mime ()

Returns the mime type of the datatype

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

images Module Image classes

class galaxy.datatypes.images.**Bmp** (**kwd)

Bases: *galaxy.datatypes.images.Image*

file_ext = 'bmp'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

sniff (filename, image=None)

Determine if the file is in bmp format.

class galaxy.datatypes.images.**Eps** (**kwd)

Bases: *galaxy.datatypes.images.Image*

file_ext = 'eps'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

sniff (filename, image=None)

Determine if the file is in eps format.

class galaxy.datatypes.images.**Gif** (**kwd)

Bases: *galaxy.datatypes.images.Image*

file_ext = 'gif'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

sniff (filename, image=None)

Determine if the file is in gif format.

class galaxy.datatypes.images.**Gmaj** (**kwd)

Bases: *galaxy.datatypes.data.Data*

Class describing a GMAJ Applet

copy_safe_peek = False

display_peek (*dataset*)

file_ext = 'gmaj.zip'

get_mime ()

Returns the mime type of the datatype

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

set_peek (*dataset*, *is_multi_byte=False*)

sniff (filename)

NOTE: the sniff.convert_newlines() call in the upload utility will keep Gmaj data types from being correctly sniffed, but the files can be uploaded (they'll be sniffed as 'txt'). This sniff function is here to provide an example of a sniffer for a zip file.

```

class galaxy.datatypes.images.Html (**kwd)
    Bases: galaxy.datatypes.data.Text

    Class describing an html file

    file_ext = 'html'

    get_mime ()
        Returns the mime type of the datatype

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    data_lines (MetadataParameter): Number of lines to peek

    set_peek (dataset, is_multi_byte=False)

    sniff (filename)
        Determines whether the file is in html format

```

```

>>> fname = get_test_fname( 'complete.bed' )
>>> Html().sniff( fname )
False
>>> fname = get_test_fname( 'file.html' )
>>> Html().sniff( fname )
True

```

```

class galaxy.datatypes.images.Im (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'im'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in im format.

class galaxy.datatypes.images.Image (**kwd)
    Bases: galaxy.datatypes.data.Data

    Class describing an image

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    set_peek (dataset, is_multi_byte=False)

    sniff (filename)

class galaxy.datatypes.images.Jpg (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'jpeg'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in jpg format.

class galaxy.datatypes.images.Laj (**kwd)
    Bases: galaxy.datatypes.data.Text

    Class describing a LAJ Applet

    copy_safe_peek = False

    display_peek (dataset)

    file_ext = 'laj'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    data_lines (MetadataParameter): Number of lines to peek

```

```
    set_peek (dataset, is_multi_byte=False)

class galaxy.datatypes.images.Pbm (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pbm'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in PBM format

class galaxy.datatypes.images.Pcd (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pcd'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in pcd format.

class galaxy.datatypes.images.Pcx (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pcx'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in pcx format.

class galaxy.datatypes.images.Pdf (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pdf'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename)
        Determine if the file is in pdf format.

class galaxy.datatypes.images.Pgm (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pgm'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in PGM format

class galaxy.datatypes.images.Png (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'png'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in png format.

class galaxy.datatypes.images.Ppm (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'ppm'
```

```

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

sniff (filename, image=None)
    Determine if the file is in ppm format.

class galaxy.datatypes.images.Psd (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'psd'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in psd format.

class galaxy.datatypes.images.Rast (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'rast'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in rast format

class galaxy.datatypes.images.Rgb (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'rgb'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in RGB format.

class galaxy.datatypes.images.Tiff (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'tiff'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in tiff format.

class galaxy.datatypes.images.Xbm (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'xbm'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in XBM format.

class galaxy.datatypes.images.Xpm (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'xpm'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in XPM format.

galaxy.datatypes.images.create_applet_tag_peek (class_name, archive, params)

```

interval Module Interval datatypes

class `galaxy.datatypes.interval.Bed` (**kwd)
Bases: `galaxy.datatypes.interval.Interval`

Tab delimited data in BED format

as_ucsc_display_file (*dataset*, **kwd)
Returns file contents with only the bed data. If bed 6+, treat as interval.

data_sources = {'index': 'bigwig', 'data': 'tabix', 'feature_search': 'fli'}

file_ext = 'bed'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

set_meta (*dataset*, *overwrite=True*, **kwd)
Sets the metadata information for datasets previously determined to be in bed format.

sniff (*filename*)
Checks for 'bedness'

BED lines have three required fields and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an annotation track. The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are used. The data type of all 12 columns is: 1-str, 2-int, 3-int, 4-str, 5-int, 6-str, 7-int, 8-int, 9-int or list, 10-int, 11-list, 12-list

For complete details see <http://genome.ucsc.edu/FAQ/FAQformat#format1>

```
>>> fname = get_test_fname( 'test_tab.bed' )
>>> Bed().sniff( fname )
True
>>> fname = get_test_fname( 'interval1.bed' )
>>> Bed().sniff( fname )
True
>>> fname = get_test_fname( 'complete.bed' )
>>> Bed().sniff( fname )
True
```

track_type = 'FeatureTrack'
Add metadata elements

class `galaxy.datatypes.interval.Bed12` (**kwd)
Bases: `galaxy.datatypes.interval.BedStrict`

Tab delimited data in strict BED format - no non-standard columns allowed; column count forced to 12

file_ext = 'bed12'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class `galaxy.datatypes.interval.Bed6` (**kwd)
Bases: `galaxy.datatypes.interval.BedStrict`

Tab delimited data in strict BED format - no non-standard columns allowed; column count forced to 6

file_ext = 'bed6'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class `galaxy.datatypes.interval.BedGraph` (**kwd)
Bases: `galaxy.datatypes.interval.Interval`

Tab delimited chrom/start/end/datavalue dataset


```

as_ucsc_display_file (dataset, **kwd)
    Returns file contents as is with no modifications. TODO: this is a functional stub and will need to be
    enhanced moving forward to provide additional support for bedgraph.

data_sources = {'index': 'bigwig', 'data': 'bigwig'}

file_ext = 'bedgraph'

get_estimated_display_viewport (dataset, chrom_col=0, start_col=1, end_col=2)
    Set viewport based on dataset's first 100 lines.

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
track_type = 'LineTrack'

class galaxy.datatypes.interval.BedStrict (**kwd)
    Bases: galaxy.datatypes.interval.Bed

    Tab delimited data in strict BED format - no non-standard columns allowed

    allow_datatype_change = False

    file_ext = 'bedstrict'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    set_meta (dataset, overwrite=True, **kwd)
    sniff (filename)

class galaxy.datatypes.interval.ChromatinInteractions (**kwd)
    Bases: galaxy.datatypes.interval.Interval

    Chromatin interactions obtained from 3C/5C/Hi-C experiments.

    column_names = ['Chrom1', 'Start1', 'End1', 'Chrom2', 'Start2', 'End2', 'Value']
        Add metadata elements

    data_sources = {'index': 'bigwig', 'data': 'tabix'}

    file_ext = 'chrint'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    sniff (filename)
    track_type = 'DiagonalHeatmapTrack'

class galaxy.datatypes.interval.CustomTrack (**kwd)
    Bases: galaxy.datatypes.tabular.Tabular

    UCSC CustomTrack

    display_peek (dataset)
        Returns formatted html of peek

    file_ext = 'customtrack'

    get_estimated_display_viewport (dataset, chrom_col=None, start_col=None, end_col=None)
        Return a chrom, start, stop tuple for viewing a file.

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    set_meta (dataset, overwrite=True, **kwd)

```

sniff (*filename*)

Determines whether the file is in customtrack format.

CustomTrack files are built within Galaxy and are basically bed or interval files with the first line looking something like this.

track name="User Track" description="User Supplied Track (from Galaxy)" color=0,0,0 visibility=1

```
>>> fname = get_test_fname( 'complete.bed' )
>>> CustomTrack().sniff( fname )
False
>>> fname = get_test_fname( 'ucsc.customtrack' )
>>> CustomTrack().sniff( fname )
True
```

ucsc_links (*dataset, type, app, base_url*)

class galaxy.datatypes.interval.**ENCODEPeak** (***kwd*)

Bases: *galaxy.datatypes.interval.Interval*

Human ENCODE peak format. There are both broad and narrow peak formats. Formats are very similar; narrow peak has an additional column, though.

Broad peak (<http://genome.ucsc.edu/FAQ/FAQformat#format13>): This format is used to provide called regions of signal enrichment based on pooled, normalized (interpreted) data. It is a BED 6+3 format.

Narrow peak <http://genome.ucsc.edu/FAQ/FAQformat#format12> and : This format is used to provide called peaks of signal enrichment based on pooled, normalized (interpreted) data. It is a BED6+4 format.

column_names = ['Chrom', 'Start', 'End', 'Name', 'Score', 'Strand', 'SignalValue', 'pValue', 'qValue', 'Peak']

data_sources = {'index': 'bigwig', 'data': 'tabix'}

Add metadata elements

file_ext = 'encodepeak'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number

sniff (*filename*)

class galaxy.datatypes.interval.**Gff** (***kwd*)

Bases: *galaxy.datatypes.tabular.Tabular*, *galaxy.datatypes.interval._RemoteCallMixin*

Tab delimited data in Gff format

column_names = ['Seqname', 'Source', 'Feature', 'Start', 'End', 'Score', 'Strand', 'Frame', 'Group']

data_sources = {'index': 'bigwig', 'data': 'interval_index', 'feature_search': 'fli'}

dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7fc078141b18>, 'chunk64': <function

display_peek (*dataset*)

Returns formatted html of peek

file_ext = 'gff'

gbrowse_links (*dataset, type, app, base_url*)

genomic_region_dataprovider (**args, **kwargs*)

genomic_region_dict_dataprovider (**args, **kwargs*)

get_estimated_display_viewport (*dataset*)

Return a chrom, start, stop tuple for viewing a file. There are slight differences between gff 2 and gff 3 formats. This function should correctly handle both...

interval_dataprovider (**args, **kwargs*)

interval_dict_dataprovider (*args, **kwargs)

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of lines

set_attribute_metadata (dataset)

Sets metadata elements for dataset's attributes.

set_meta (dataset, overwrite=True, **kwd)

sniff (filename)

Determines whether the file is in gff format

GFF lines have nine required fields that must be tab-separated.

For complete details see <http://genome.ucsc.edu/FAQ/FAQformat#format3>

```
>>> fname = get_test_fname( 'gff_version_3.gff' )
>>> Gff().sniff( fname )
False
>>> fname = get_test_fname( 'test.gff' )
>>> Gff().sniff( fname )
True
```

track_type = 'FeatureTrack'

Add metadata elements

ucsc_links (dataset, type, app, base_url)

class galaxy.datatypes.interval.**Gff3** (**kwd)

Bases: *galaxy.datatypes.interval.Gff*

Tab delimited data in Gff3 format

column_names = ['Seqid', 'Source', 'Type', 'Start', 'End', 'Score', 'Strand', 'Phase', 'Attributes']

file_ext = 'gff3'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of lines

set_meta (dataset, overwrite=True, **kwd)

sniff (filename)

Determines whether the file is in gff version 3 format

GFF 3 format:

- 1.adds a mechanism for representing more than one level of hierarchical grouping of features and subfeatures.
- 2.separates the ideas of group membership and feature name/id
- 3.constrains the feature type field to be taken from a controlled vocabulary.
- 4.allows a single feature, such as an exon, to belong to more than one group at a time.
- 5.provides an explicit convention for pairwise alignments
- 6.provides an explicit convention for features that occupy disjunct regions

The format consists of 9 columns, separated by tabs (NOT spaces).

Undefined fields are replaced with the "." character, as described in the original GFF spec.

For complete details see <http://song.sourceforge.net/gff3.shtml>

```

>>> fname = get_test_fname( 'test.gff' )
>>> Gff3().sniff( fname )
False
>>> fname = get_test_fname('gff_version_3.gff')
>>> Gff3().sniff( fname )
True

```

track_type = 'FeatureTrack'

Add metadata elements

valid_gff3_phase = ['.', '0', '1', '2']

valid_gff3_strand = ['+', '-', '.', '?']

class galaxy.datatypes.interval.**Gtf** (**kwd)

Bases: *galaxy.datatypes.interval.Gff*

Tab delimited data in Gtf format

column_names = ['Seqname', 'Source', 'Feature', 'Start', 'End', 'Score', 'Strand', 'Frame', 'Attributes']

file_ext = 'gtf'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

sniff (filename)

Determines whether the file is in gtf format

GTF lines have nine required fields that must be tab-separated. The first eight GTF fields are the same as GFF. The group field has been expanded into a list of attributes. Each attribute consists of a type/value pair. Attributes must end in a semi-colon, and be separated from any following attribute by exactly one space. The attribute list must begin with the two mandatory attributes:

gene_id value - A globally unique identifier for the genomic source of the sequence. trans-

script_id value - A globally unique identifier for the predicted transcript.

For complete details see <http://genome.ucsc.edu/FAQ/FAQformat#format4>

```

>>> fname = get_test_fname( '1.bed' )
>>> Gtf().sniff( fname )
False
>>> fname = get_test_fname( 'test.gff' )
>>> Gtf().sniff( fname )
False
>>> fname = get_test_fname( 'test.gtf' )
>>> Gtf().sniff( fname )
True

```

track_type = 'FeatureTrack'

Add metadata elements

class galaxy.datatypes.interval.**Interval** (**kwd)

Bases: *galaxy.datatypes.tabular.Tabular*

Tab delimited data containing interval information

as_ucsc_display_file (dataset, **kwd)

Returns file contents with only the bed data

data_sources = {'index': 'bigwig', 'data': 'tabix'}

Add metadata elements

dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7fc078141b18>, 'chunk64': <function chunk64_dataprovider at 0x7fc078141b18>}

display_peek (*dataset*)

Returns formatted html of peek

displayable (*dataset*)

file_ext = 'interval'

genomic_region_dataprovider (**args, **kwargs*)

genomic_region_dict_dataprovider (**args, **kwargs*)

get_estimated_display_viewport (*dataset, chrom_col=None, start_col=None, end_col=None*)

Return a chrom, start, stop tuple for viewing a file.

get_track_resolution (*dataset, start, end*)

get_track_window (*dataset, data, start, end*)

Assumes the incoming track data is sorted already.

init_meta (*dataset, copy_from=None*)

interval_dataprovider (**args, **kwargs*)

interval_dict_dataprovider (**args, **kwargs*)

line_class = 'region'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number

repair_methods (*dataset*)

Return options for removing errors along with a description

set_meta (*dataset, overwrite=True, first_line_is_header=False, **kwd*)

Tries to guess from the line the location number of the column for the chromosome, region start-end and strand

sniff (*filename*)

Checks for 'intervalness'

This format is mostly used by galaxy itself. Valid interval files should include a valid header comment, but this seems to be loosely regulated.

```
>>> fname = get_test_fname( 'test_space.txt' )
>>> Interval().sniff( fname )
False
>>> fname = get_test_fname( 'interval.interval' )
>>> Interval().sniff( fname )
True
```

track_type = 'FeatureTrack'

ucsc_links (*dataset, type, app, base_url*)

Generate links to UCSC genome browser sites based on the dbkey and content of dataset.

validate (*dataset*)

Validate an interval file using the bx GenomicIntervalReader

class galaxy.datatypes.interval.Wiggle (***kwd*)

Bases: *galaxy.datatypes.tabular.Tabular*, *galaxy.datatypes.interval._RemoteCallMixin*

Tab delimited data in wiggle format

data_sources = {'index': 'bigwig', 'data': 'bigwig'}

dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7fc078141b18>, 'chunk64': <function

display_peek (*dataset*)

Returns formatted html of peek

file_ext = 'wig'

gbrowse_links (*dataset, type, app, base_url*)

get_estimated_display_viewport (*dataset*)

Return a chrom, start, stop tuple for viewing a file.

get_track_resolution (*dataset, start, end*)

get_track_window (*dataset, data, start, end*)

Assumes we have a numpy file.

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to display

set_meta (*dataset, overwrite=True, **kwd*)

sniff (*filename*)

Determines whether the file is in wiggle format

The .wig format is line-oriented. Wiggle data is preceded by a track definition line, which adds a number of options for controlling the default display of this track. Following the track definition line is the track data, which can be entered in several different formats.

The track definition line begins with the word 'track' followed by the track type. The track type with version is REQUIRED, and it currently must be wiggle_0. For example, track type=wiggle_0...

For complete details see <http://genome.ucsc.edu/goldenPath/help/wiggle.html>

```
>>> fname = get_test_fname( 'intervall.bed' )
>>> Wiggle().sniff( fname )
False
>>> fname = get_test_fname( 'wiggle.wig' )
>>> Wiggle().sniff( fname )
True
```

track_type = 'LineTrack'

ucsc_links (*dataset, type, app, base_url*)

wiggle_dataprovider (**args, **kwargs*)

wiggle_dict_dataprovider (**args, **kwargs*)

metadata Module Galaxy Metadata

class galaxy.datatypes.metadata.**ColumnParameter** (*spec*)

Bases: *galaxy.datatypes.metadata.RangeParameter*

get_html (*value, context=None, other_values=None, values=None, **kwd*)

get_html_field (*value=None, context=None, other_values=None, values=None, **kwd*)

class galaxy.datatypes.metadata.**ColumnTypesParameter** (*spec*)

Bases: *galaxy.datatypes.metadata.MetadataParameter*

to_string (*value*)

class galaxy.datatypes.metadata.**DBKeyParameter** (*spec*)

Bases: *galaxy.datatypes.metadata.SelectParameter*

get_html (*value=None, context=None, other_values=None, values=None, **kwd*)

```

    get_html_field (value=None, context=None, other_values=None, values=None, **kwd)

class galaxy.datatypes.metadata.DictParameter (spec)
    Bases: galaxy.datatypes.metadata.MetadataParameter

    to_safe_string (value)

    to_string (value)

class galaxy.datatypes.metadata.FileParameter (spec)
    Bases: galaxy.datatypes.metadata.MetadataParameter

    from_external_value (value, parent, path_rewriter=None)
        Turns a value read from a external dict into its value to be pushed directly into the metadata dict.

    get_html (value=None, context=None, other_values=None, **kwd)

    get_html_field (value=None, context=None, other_values=None, **kwd)

    make_copy (value, target_context, source_context)

    classmethod marshal (value)

    new_file (dataset=None, **kws)

    to_external_value (value)
        Turns a value read from a metadata into its value to be pushed directly into the external dict.

    to_safe_string (value)

    to_string (value)

    wrap (value, session)

class galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper (job)
    Bases: object

    cleanup_external_metadata (sa_session)

    external_metadata_set_successfully (dataset, sa_session)

    get_dataset_metadata_key (dataset)

    get_output_filenames_by_dataset (dataset, sa_session)

    set_job_runner_external_pid (pid, sa_session)

    setup_external_metadata (datasets, sa_session, exec_dir=None, tmp_dir=None,
                             dataset_files_path=None, output_fnames=None, config_root=None,
                             config_file=None, datatypes_config=None, job_metadata=None,
                             compute_tmp_dir=None, kws=None)

class galaxy.datatypes.metadata.ListParameter (spec)
    Bases: galaxy.datatypes.metadata.MetadataParameter

    to_string (value)

class galaxy.datatypes.metadata.MetadataCollection (parent)
    Bases: object

    MetadataCollection is not a collection at all, but rather a proxy to the real metadata which is stored as a Dictionary. This class handles processing the metadata elements when they are set and retrieved, returning default values in cases when metadata is not set.

    element_is_set (name)

    from_JSON_dict (filename=None, path_rewriter=None, json_dict=None)

```

```
get (key, default=None)
get_html_by_name (name, **kwd)
get_parent ()
items ()
make_dict_copy (to_copy)
    Makes a deep copy of input iterable to_copy according to self.spec
parent
set_parent (parent)
spec
to_JSON_dict (filename=None)

galaxy.datatypes.metadata.MetadataElement = <galaxy.datatypes.metadata.Statement object>
    MetadataParameter sub-classes.

class galaxy.datatypes.metadata.MetadataElementSpec (datatype,          name=None,
                                                    desc=None,          param=<class
                                                    'galaxy.datatypes.metadata.MetadataParameter'>,
                                                    default=None, no_value=None, vis-
                                                    ible=True,      set_in_upload=False,
                                                    **kwargs)

Bases: object

Defines a metadata element and adds it to the metadata_spec (which is a MetadataSpecCollection) of datatype.

get (name, default=None)

unwrap (value)
    Turns an incoming value into its storable form.

wrap (value, session)
    Turns a stored value into its usable form.

class galaxy.datatypes.metadata.MetadataParameter (spec)
    Bases: object

    from_external_value (value, parent)
        Turns a value read from an external dict into its value to be pushed directly into the metadata dict.

    get_html (value, context=None, other_values=None, **kwd)
        The “context” is simply the metadata collection/bunch holding this piece of metadata. This is passed in to
        allow for metadata to validate against each other (note: this could turn into a huge, recursive mess if not
        done with care). For example, a column assignment should validate against the number of columns in the
        dataset.

    get_html_field (value=None, context=None, other_values=None, **kwd)

    make_copy (value, target_context=None, source_context=None)

    classmethod marshal (value)
        This method should/can be overridden to convert the incoming value to whatever type it is supposed to
        be.

    to_external_value (value)
        Turns a value read from a metadata into its value to be pushed directly into the external dict.

    to_safe_string (value)

    to_string (value)
```


unwrap (*form_value*)
Turns a value into its storable form.

validate (*value*)
Throw an exception if the value is invalid.

wrap (*value, session*)
Turns a value into its usable form.

class `galaxy.datatypes.metadata.MetadataSpecCollection` (*dict=None*)
Bases: `galaxy.util.odict.odict`

A simple extension of dict which allows cleaner access to items and allows the values to be iterated over directly as if it were a list. `append()` is also implemented for simplicity and does not “append”.

append (*item*)

iter ()

class `galaxy.datatypes.metadata.MetadataTempFile` (***kws*)
Bases: `object`

classmethod `cleanup_from_JSON_dict_filename` (*filename*)

file_name

classmethod `from_JSON` (*json_dict*)

classmethod `is_JSONified_value` (*value*)

tmp_dir = ‘database/tmp’

to_JSON ()

class `galaxy.datatypes.metadata.PythonObjectParameter` (*spec*)
Bases: `galaxy.datatypes.metadata.MetadataParameter`

get_html (*value=None, context=None, other_values=None, **kwd*)

get_html_field (*value=None, context=None, other_values=None, **kwd*)

classmethod `marshal` (*value*)

to_string (*value*)

class `galaxy.datatypes.metadata.RangeParameter` (*spec*)
Bases: `galaxy.datatypes.metadata.SelectParameter`

get_html (*value, context=None, other_values=None, values=None, **kwd*)

get_html_field (*value=None, context=None, other_values=None, values=None, **kwd*)

classmethod `marshal` (*value*)

class `galaxy.datatypes.metadata.SelectParameter` (*spec*)
Bases: `galaxy.datatypes.metadata.MetadataParameter`

get_html (*value, context=None, other_values=None, values=None, **kwd*)

get_html_field (*value=None, context=None, other_values=None, values=None, **kwd*)

classmethod `marshal` (*value*)

to_string (*value*)

wrap (*value, session*)

```
class galaxy.datatypes.metadata.Statement(target)
    Bases: object
```

This class inserts its target into a list in the surrounding class. the data.Data class has a metaclass which executes these statements. This is how we shove the metadata element spec into the class.

```
classmethod process(element)
```

ngsindex Module NGS indexes

```
class galaxy.datatypes.ngsindex.BowtieBaseIndex(**kwd)
    Bases: galaxy.datatypes.ngsindex.BowtieIndex
```

Bowtie base space index

```
file_ext = 'bowtie_base_index'
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
```

```
class galaxy.datatypes.ngsindex.BowtieColorIndex(**kwd)
    Bases: galaxy.datatypes.ngsindex.BowtieIndex
```

Bowtie color space index

```
file_ext = 'bowtie_color_index'
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
```

```
class galaxy.datatypes.ngsindex.BowtieIndex(**kwd)
    Bases: galaxy.datatypes.images.Html
```

base class for BowtieIndex is subclassed by BowtieColorIndex and BowtieBaseIndex

```
allow_datatype_change = False
```

```
composite_type = 'auto_primary_file'
```

```
display_peek(dataset)
```

```
file_ext = 'bowtie_index'
```

```
generate_primary_file(dataset=None)
```

This is called only at upload to write the html file cannot rename the datasets here - they come with the default unfortunately

```
is_binary = True
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
```

```
regenerate_primary_file(dataset)
```

cannot do this until we are setting metadata

```
set_peek(dataset, is_multi_byte=False)
```

```
sniff(filename)
```

qualityscore Module Qualityscore class

```
class galaxy.datatypes.qualityscore.QualityScore(**kwd)
    Bases: galaxy.datatypes.data.Text
```

until we know more about quality score formats

```
file_ext = 'qual'
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
```

```

class galaxy.datatypes.qualityscore.QualityScore454 (**kwd)
    Bases: galaxy.datatypes.qualityscore.QualityScore

    until we know more about quality score formats

    file_ext = 'qual454'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    sniff (filename)

```

```

>>> fname = get_test_fname( 'sequence.fasta' )
>>> QualityScore454().sniff( fname )
False
>>> fname = get_test_fname( 'sequence.qual454' )
>>> QualityScore454().sniff( fname )
True

```

```

class galaxy.datatypes.qualityscore.QualityScoreIllumina (**kwd)
    Bases: galaxy.datatypes.qualityscore.QualityScore

    until we know more about quality score formats

    file_ext = 'qualillumina'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.qualityscore.QualityScoreSOLiD (**kwd)
    Bases: galaxy.datatypes.qualityscore.QualityScore

    until we know more about quality score formats

    file_ext = 'qualsolid'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    set_meta (dataset, **kwd)
    sniff (filename)

```

```

>>> fname = get_test_fname( 'sequence.fasta' )
>>> QualityScoreSOLiD().sniff( fname )
False
>>> fname = get_test_fname( 'sequence.qualsolid' )
>>> QualityScoreSOLiD().sniff( fname )
True

```

```

class galaxy.datatypes.qualityscore.QualityScoreSolexa (**kwd)
    Bases: galaxy.datatypes.qualityscore.QualityScore

    until we know more about quality score formats

    file_ext = 'qualsolexa'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

```

registry Module Provides mapping between extensions and datatypes, mime-types, etc.

```

exception galaxy.datatypes.registry.ConfigurationError
    Bases: exceptions.Exception

```

```

class galaxy.datatypes.registry.Registry
    Bases: object

```

change_datatype (*data*, *ext*)

find_conversion_destination_for_dataset_by_extensions (*dataset*, *accepted_formats*, *converter_safe=True*)

Returns (*target_ext*, existing converted dataset)

get_available_tracks ()

get_composite_extensions ()

get_converter_by_target_type (*source_ext*, *target_ext*)

Returns a converter based on source and target datatypes

get_converters_by_datatype (*ext*)

Returns available converters by source type

get_datatype_by_extension (*ext*)

Returns a datatype based on an extension

get_datatype_class_by_name (*name*)

Return the datatype class where the datatype's *type* attribute (as defined in the *datatype_conf.xml* file) contains *name*.

get_display_sites (*site_type*)

get_legacy_sites_by_build (*site_type*, *build*)

get_mimetype_by_extension (*ext*, *default='application/octet-stream'*)

Returns a mimetype based on an extension

get_upload_metadata_params (*context*, *group*, *tool*)

Returns dict of case value:inputs for metadata conditional for upload tool

integrated_datatypes_configs

load_build_sites (*root*)

load_datatype_converters (*toolbox*, *installed_repository_dict=None*, *deactivate=False*)

If *deactivate* is *False*, add datatype converters from *self.converters* or *self.proprietary_converters* to the calling app's toolbox. If *deactivate* is *True*, eliminates relevant converters from the calling app's toolbox.

load_datatype_sniffers (*root*, *deactivate=False*, *handling_proprietary_datatypes=False*, *override=False*)

Process the sniffers element from a parsed a datatypes XML file located at *root_dir/config* (if processing the Galaxy distributed config) or contained within an installed Tool Shed repository. If *deactivate* is *True*, an installed Tool Shed repository that includes custom sniffers is being deactivated or uninstalled, so appropriate loaded sniffers will be removed from the registry. The value of *override* will be *False* when a Tool Shed repository is being installed. Since installation is occurring after the datatypes registry has been initialized at server startup, its contents cannot be overridden by newly introduced conflicting sniffers.

load_datatypes (*root_dir=None*, *config=None*, *deactivate=False*, *override=True*)

Parse a datatypes XML file located at *root_dir/config* (if processing the Galaxy distributed config) or contained within an installed Tool Shed repository. If *deactivate* is *True*, an installed Tool Shed repository that includes custom datatypes is being deactivated or uninstalled, so appropriate loaded datatypes will be removed from the registry. The value of *override* will be *False* when a Tool Shed repository is being installed. Since installation is occurring after the datatypes registry has been initialized at server startup, its contents cannot be overridden by newly introduced conflicting data types.

load_display_applications (*installed_repository_dict=None*, *deactivate=False*)

If *deactivate* is *False*, add display applications from *self.display_app_containers* or *self.proprietary_display_app_containers* to appropriate datatypes. If *deactivate* is *True*, eliminates relevant display applications from appropriate datatypes.

load_external_metadata_tool (*toolbox*)

Adds a tool which is used to set external metadata

old_change_datatype (*data, ext*)

Creates and returns a new datatype based on an existing data and an extension

set_default_values ()

to_xml_file ()

sequence Module Sequence classes

class `galaxy.datatypes.sequence.Alignment` (***kwd*)

Bases: `galaxy.datatypes.data.Text`

Class describing an alignment

metadata_spec = **dbkey** (**DBKeyParameter**): Database/Build, defaults to '?', **data_lines** (**MetadataParameter**): Number of lines

split (*input_datasets, subdir_generator_function, split_params*)

Split a generic alignment file (not sensible or possible, see subclasses).

class `galaxy.datatypes.sequence.Axt` (***kwd*)

Bases: `galaxy.datatypes.data.Text`

Class describing an axt alignment

file_ext = 'axt'

metadata_spec = **dbkey** (**DBKeyParameter**): Database/Build, defaults to '?', **data_lines** (**MetadataParameter**): Number of lines

sniff (*filename*)

Determines whether the file is in axt format

axt alignment files are produced from Blastz, an alignment tool available from Webb Miller's lab at Penn State University.

Each alignment block in an axt file contains three lines: a summary line and 2 sequence lines. Blocks are separated from one another by blank lines.

The summary line contains chromosomal position and size information about the alignment. It consists of 9 required fields.

The sequence lines contain the sequence of the primary assembly (line 2) and aligning assembly (line 3) with inserts. Repeats are indicated by lower-case letters.

For complete details see <http://genome.ucsc.edu/goldenPath/help/axt.html>

```
>>> fname = get_test_fname( 'alignment.axt' )
>>> Axt().sniff( fname )
True
>>> fname = get_test_fname( 'alignment.lav' )
>>> Axt().sniff( fname )
False
```

class `galaxy.datatypes.sequence.Fasta` (***kwd*)

Bases: `galaxy.datatypes.sequence.Sequence`

Class representing a FASTA sequence

file_ext = 'fasta'

metadata_spec = **dbkey** (**DBKeyParameter**): Database/Build, defaults to '?', **data_lines** (**MetadataParameter**): Number of lines

sniff (*filename*)

Determines whether the file is in fasta format

A sequence in FASTA format consists of a single-line description, followed by lines of sequence data. The first character of the description line is a greater-than (“>”) symbol in the first column. All lines should be shorter than 80 characters

For complete details see <http://www.ncbi.nlm.nih.gov/blast/fasta.shtml>

Rules for sniffing as True:

We don’t care about line length (other than empty lines).

The first non-empty line must start with ‘>’ and the Very Next line.strip() must have sequence data and not be a header.

‘sequence data’ here is loosely defined as non-empty lines which do not start with ‘>’

This will cause Color Space FASTA (csfasta) to be detected as True (they are, after all, still FASTA files - they have a header line followed by sequence data)

Previously this method did some checking to determine if the sequence data had integers (presumably to differentiate between fasta and csfasta)

This should be done through sniff order, where csfasta (currently has a null sniff function) is detected for first (stricter definition) followed sometime after by fasta

We will only check that the first purported sequence is correctly formatted.

```
>>> fname = get_test_fname( 'sequence.maf' )
>>> Fasta().sniff( fname )
False
>>> fname = get_test_fname( 'sequence.fasta' )
>>> Fasta().sniff( fname )
True
```

classmethod split (*input_datasets, subdir_generator_function, split_params*)

Split a FASTA file sequence by sequence.

Note that even if split_mode=”number_of_parts”, the actual number of sub-files produced may not match that requested by split_size.

If split_mode=”to_size” then split_size is treated as the number of FASTA records to put in each sub-file (not size in bytes).

class galaxy.datatypes.sequence.**Fastq** (**kwd)

Bases: *galaxy.datatypes.sequence.Sequence*

Class representing a generic FASTQ sequence

file_ext = ‘fastq’

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to ‘?’, data_lines (MetadataParameter): Number

static process_split_file (*data*)

This is called in the context of an external process launched by a Task (possibly not on the Galaxy machine) to create the input files for the Task. The parameters: data - a dict containing the contents of the split file

set_meta (*dataset, **kwd*)

Set the number of sequences and the number of data lines in dataset. FIXME: This does not properly handle line wrapping

sniff (*filename*)

Determines whether the file is in generic fastq format For details, see <http://maq.sourceforge.net/fastq.shtml>

Note: There are three kinds of FASTQ files, known as “Sanger” (sometimes called “Standard”), Solexa, and Illumina. These differ in the representation of the quality scores

```
>>> fname = get_test_fname( '1.fastqsanger' )
>>> Fastq().sniff( fname )
True
>>> fname = get_test_fname( '2.fastqsanger' )
>>> Fastq().sniff( fname )
True
```

classmethod **split** (*input_datasets*, *subdir_generator_function*, *split_params*)

FASTQ files are split on cluster boundaries, in increments of 4 lines

class `galaxy.datatypes.sequence.FastqCSSanger` (***kwd*)

Bases: `galaxy.datatypes.sequence.Fastq`

Class representing a Color Space FASTQ sequence (e.g a SOLiD variant)

file_ext = 'fastqcssanger'

metadata_spec = **dbkey** (**DBKeyParameter**): Database/Build, defaults to '?', **data_lines** (**MetadataParameter**): Number of lines

class `galaxy.datatypes.sequence.FastqIllumina` (***kwd*)

Bases: `galaxy.datatypes.sequence.Fastq`

Class representing a FASTQ sequence (the Illumina 1.3+ variant)

file_ext = 'fastqillumina'

metadata_spec = **dbkey** (**DBKeyParameter**): Database/Build, defaults to '?', **data_lines** (**MetadataParameter**): Number of lines

class `galaxy.datatypes.sequence.FastqSanger` (***kwd*)

Bases: `galaxy.datatypes.sequence.Fastq`

Class representing a FASTQ sequence (the Sanger variant)

file_ext = 'fastqsanger'

metadata_spec = **dbkey** (**DBKeyParameter**): Database/Build, defaults to '?', **data_lines** (**MetadataParameter**): Number of lines

class `galaxy.datatypes.sequence.FastqSolexa` (***kwd*)

Bases: `galaxy.datatypes.sequence.Fastq`

Class representing a FASTQ sequence (the Solexa variant)

file_ext = 'fastqsolexa'

metadata_spec = **dbkey** (**DBKeyParameter**): Database/Build, defaults to '?', **data_lines** (**MetadataParameter**): Number of lines

class `galaxy.datatypes.sequence.Lav` (***kwd*)

Bases: `galaxy.datatypes.data.Text`

Class describing a LAV alignment

file_ext = 'lav'

metadata_spec = **dbkey** (**DBKeyParameter**): Database/Build, defaults to '?', **data_lines** (**MetadataParameter**): Number of lines

sniff (*filename*)

Determines whether the file is in lav format

LAV is an alignment format developed by Webb Miller’s group. It is the primary output format for BLASTZ. The first line of a .lav file begins with #:lav.

For complete details see http://www.bioperl.org/wiki/LAV_alignment_format

```
>>> fname = get_test_fname( 'alignment.lav' )
>>> Lav().sniff( fname )
True
>>> fname = get_test_fname( 'alignment.axt' )
>>> Lav().sniff( fname )
False
```

class galaxy.datatypes.sequence.**Maf** (**kwd)
Bases: *galaxy.datatypes.sequence.Alignment*

Class describing a Maf alignment

display_peek (dataset)
Returns formatted html of peek

file_ext = 'maf'

init_meta (dataset, copy_from=None)

make_html_table (dataset, skipchars=[])
Create HTML table, used for displaying peek

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to display

set_meta (dataset, overwrite=True, **kwd)
Parses and sets species, chromosomes, index from MAF file.

set_peek (dataset, is_multi_byte=False)

sniff (filename)
Determines whether the file is in maf format

The .maf format is line-oriented. Each multiple alignment ends with a blank line. Each sequence in an alignment is on a single line, which can get quite long, but there is no length limit. Words in a line are delimited by any white space. Lines starting with # are considered to be comments. Lines starting with ## can be ignored by most programs, but contain meta-data of one form or another.

The first line of a .maf file begins with ##maf. This word is followed by white-space-separated variable=value pairs. There should be no white space surrounding the “=”.

For complete details see <http://genome.ucsc.edu/FAQ/FAQformat#format5>

```
>>> fname = get_test_fname( 'sequence.maf' )
>>> Maf().sniff( fname )
True
>>> fname = get_test_fname( 'sequence.fasta' )
>>> Maf().sniff( fname )
False
```

class galaxy.datatypes.sequence.**MafCustomTrack** (**kwd)
Bases: *galaxy.datatypes.data.Text*

file_ext = 'mafcustomtrack'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to display

set_meta (dataset, overwrite=True, **kwd)
Parses and sets viewport metadata from MAF file.

class galaxy.datatypes.sequence.**Sequence** (**kwd)
Bases: *galaxy.datatypes.data.Text*

Class describing a sequence


```
classmethod do_fast_split (input_datasets, toc_file_datasets, subdir_generator_function,
                           split_params)
```

```
classmethod do_slow_split (input_datasets, subdir_generator_function, split_params)
```

```
static get_sequences_per_file (total_sequences, split_params)
```

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of data lines

```
set_meta (dataset, **kwd)
```

Set the number of sequences and the number of data lines in dataset.

```
set_peek (dataset, is_multi_byte=False)
```

```
split (input_datasets, subdir_generator_function, split_params)
```

Split a generic sequence file (not sensible or possible, see subclasses).

```
classmethod write_split_files (input_datasets, toc_file_datasets, subdir_generator_function,
                               sequences_per_file)
```

```
class galaxy.datatypes.sequence.SequenceSplitLocations (**kwd)
```

Bases: `galaxy.datatypes.data.Text`

Class storing information about a sequence file composed of multiple gzip files concatenated as one OR an uncompressed file. In the GZIP case, each sub-file's location is stored in start and end.

The format of the file is JSON:

```
{ "sections" : [
    { "start" : "x", "end" : "y", "sequences" : "z" },
    ...
  ] }
```

```
file_ext = 'fqtoc'
```

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of data lines

```
set_peek (dataset, is_multi_byte=False)
```

```
sniff (filename)
```

```
class galaxy.datatypes.sequence.csFasta (**kwd)
```

Bases: `galaxy.datatypes.sequence.Sequence`

Class representing the SOLID Color-Space sequence (csfasta)

```
file_ext = 'csfasta'
```

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of data lines

```
set_meta (dataset, **kwd)
```

```
sniff (filename)
```

Color-space sequence: >2_15_85_F3 T213021013012303002332212012112221222112212222

```
>>> fname = get_test_fname( 'sequence.fasta' )
>>> csFasta().sniff( fname )
False
>>> fname = get_test_fname( 'sequence.csfasta' )
>>> csFasta().sniff( fname )
True
```

sniff Module File format detector

exception galaxy.datatypes.sniff.InappropriateDatasetContentError

Bases: `exceptions.Exception`

```
galaxy.datatypes.sniff.check_newlines (fname, bytes_to_read=52428800)
```

Determines if there are any non-POSIX newlines in the first number of bytes (by default, 50MB) of the file.

```
galaxy.datatypes.sniff.convert_newlines(fname, in_place=True, tmp_dir=None,
                                         tmp_prefix=None)
```

Converts in place a file from universal line endings to Posix line endings.

```
>>> fname = get_test_fname('temp.txt')
>>> file(fname, 'wt').write("1 2\r3 4")
>>> convert_newlines(fname, tmp_prefix="gxtest", tmp_dir=tempfile.gettempdir())
(2, None)
>>> file(fname).read()
'1 2\n3 4\n'
```

[illegible]

Combines above methods: `convert_newlines()` and `sep2tabs()` so that files do not need to be read twice

```
>>> fname = get_test_fname('temp.txt')
>>> file(fname, 'wt').write("1 2\r3 4")
>>> convert_newlines_sep2tabs(fname, tmp_prefix="gxtest", tmp_dir=tempfile.gettempdir())
(2, None)
>>> file(fname).read()
'1\t2\n3\t4\n'
```

`galaxy.datatypes.sniff.get_headers` (*fname*, *sep*, *count*=60, *is_multi_byte*=False)

Returns a list with the first ‘count’ lines split by ‘sep’

```
>>> fname = get_test_fname('complete.bed')
>>> get_headers(fname, '\t')
[['chr7', '127475281', '127491632', 'NM_000230', '0', '+', '127486022', '127488767', '0', '3',
```

```
galaxy.datatypes.sniff.get test fname(fname)
```

Returns test data filename

```
galaxy.datatypes.sniff.guess_ext (fname, sniff_order=None, is_multi_byte=False)
```

Returns an extension that can be used in the datatype factory to generate a data for the ‘fname’ file

```
>>> fname = get_test_fname('megablast_xml_parser_test1.blastxml')
>>> guess_ext(fname)
'xml'
>>> fname = get_test_fname('interval.interval')
>>> guess_ext(fname)
'interval'
>>> fname = get_test_fname('interval1.bed')
>>> guess_ext(fname)
'bed'
>>> fname = get_test_fname('test_tab.bed')
>>> guess_ext(fname)
'bed'
>>> fname = get_test_fname('sequence.maf')
>>> guess_ext(fname)
'maf'
```

```

>>> fname = get_test_fname('sequence.fasta')
>>> guess_ext(fname)
'fasta'
>>> fname = get_test_fname('file.html')
>>> guess_ext(fname)
'html'
>>> fname = get_test_fname('test.gtf')
>>> guess_ext(fname)
'gtf'
>>> fname = get_test_fname('test.gff')
>>> guess_ext(fname)
'gff'
>>> fname = get_test_fname('gff_version_3.gff')
>>> guess_ext(fname)
'gff3'
>>> fname = get_test_fname('temp.txt')
>>> file(fname, 'wt').write("a\t2\nc\t1\nd\t0")
>>> guess_ext(fname)
'tabular'
>>> fname = get_test_fname('temp.txt')
>>> file(fname, 'wt').write("a 1 2 x\nb 3 4 y\nc 5 6 z")
>>> guess_ext(fname)
'txt'
>>> fname = get_test_fname('test_tab1.tabular')
>>> guess_ext(fname)
'tabular'
>>> fname = get_test_fname('alignment.lav')
>>> guess_ext(fname)
'lav'
>>> fname = get_test_fname('1.sff')
>>> guess_ext(fname)
'sff'
>>> fname = get_test_fname('1.bam')
>>> guess_ext(fname)
'bam'
>>> fname = get_test_fname('3unsorted.bam')
>>> guess_ext(fname)
'bam'

```

galaxy.datatypes.sniff.**handle_compressed_file**(filename, datatypes_registry, ext='auto')

galaxy.datatypes.sniff.**handle_uploaded_dataset_file**(filename, datatypes_registry,
ext='auto',
is_multi_byte=False)

galaxy.datatypes.sniff.**is_column_based**(fname, sep='\t', skip=0, is_multi_byte=False)

Checks whether the file is column based with respect to a separator (defaults to tab separator).

```

>>> fname = get_test_fname('test.gff')
>>> is_column_based(fname)
True
>>> fname = get_test_fname('test_tab.bed')
>>> is_column_based(fname)
True
>>> is_column_based(fname, sep=' ')
False
>>> fname = get_test_fname('test_space.txt')
>>> is_column_based(fname)

```

```
False
>>> is_column_based(fname, sep=' ')
True
>>> fname = get_test_fname('test_ensembl.tab')
>>> is_column_based(fname)
True
>>> fname = get_test_fname('test_tab1.tabular')
>>> is_column_based(fname, sep=' ', skip=0)
False
>>> fname = get_test_fname('test_tab1.tabular')
>>> is_column_based(fname)
True
```

`galaxy.datatypes.sniff.sep2tabs` (*fname*, *in_place=True*, *patt='\s+'*)
Transforms in place a ‘sep’ separated file to a tab separated one

```
>>> fname = get_test_fname('temp.txt')
>>> file(fname, 'wt').write("1 2\n3 4\n")
>>> sep2tabs(fname)
(2, None)
>>> file(fname).read()
'1\t2\n3\t4\n'
```

`galaxy.datatypes.sniff.stream_to_file` (*stream*, *suffix=''*, *prefix=''*, *dir=None*, *text=False*,
***kwd*)

Writes a stream to a temporary file, returns the temporary file’s name

`galaxy.datatypes.sniff.stream_to_open_named_file` (*stream*, *fd*, *filename*,
source_encoding=None,
source_error='strict', *tar-*
get_encoding=None, *tar-*
get_error='strict')

Writes a stream to the provided file descriptor, returns the file’s name and bool(*is_multi_byte*). Closes file descriptor

tabular Module Tabular datatype

class `galaxy.datatypes.tabular.Eland` (***kwd*)

Bases: `galaxy.datatypes.tabular.Tabular`

Support for the export.txt.gz file used by Illumina’s ELANDv2e aligner

file_ext = ‘_export.txt.gz’

make_html_table (*dataset*, *skipchars=None*)

Create HTML table, used for displaying peek

metadata_spec = **dbkey** (DBKeyParameter): Database/Build, defaults to ‘?’, **data_lines** (MetadataParameter): Number

set_meta (*dataset*, *overwrite=True*, *skip=None*, *max_data_lines=5*, ***kwd*)

sniff (*filename*)

Determines whether the file is in ELAND export format

A file in ELAND export format consists of lines of tab-separated data. There is no header.

Rules for sniffing as True:

- There must be 22 columns on each line
- LANE, TILEm X, Y, INDEX, READ_NO, SEQ, QUAL, POSITION, *STRAND, FILT must be correct
- We will only check that up to the first 5 alignments are correctly formatted.

class galaxy.datatypes.tabular.**ElandMulti** (**kwd)

Bases: *galaxy.datatypes.tabular.Tabular*

file_ext = 'elandmulti'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

sniff (filename)

class galaxy.datatypes.tabular.**FeatureLocationIndex** (**kwd)

Bases: *galaxy.datatypes.tabular.Tabular*

An index that stores feature locations in tabular format.

file_ext = 'fli'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.tabular.**Pileup** (**kwd)

Bases: *galaxy.datatypes.tabular.Tabular*

Tab delimited data in pileup (6- or 10-column) format

data_sources = {'data': 'tabix'}

Add metadata elements

dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7fc078141b18>, 'chunk64': <function chunk64_dataprovider at 0x7fc078141b18>}

display_peek (dataset)

Returns formatted html of peek

file_ext = 'pileup'

genomic_region_dataprovider (*args, **kwargs)

genomic_region_dict_dataprovider (*args, **kwargs)

init_meta (dataset, copy_from=None)

line_class = 'genomic coordinate'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

repair_methods (dataset)

Return options for removing errors along with a description

sniff (filename)

Checks for 'pileup-ness'

There are two main types of pileup: 6-column and 10-column. For both, the first three and last two columns are the same. We only check the first three to allow for some personalization of the format.

```
>>> fname = get_test_fname( 'interval.interval' )
>>> Pileup().sniff( fname )
False
>>> fname = get_test_fname( '6col.pileup' )
>>> Pileup().sniff( fname )
True
>>> fname = get_test_fname( '10col.pileup' )
>>> Pileup().sniff( fname )
True
```

```

class galaxy.datatypes.tabular.Sam (**kwd)
    Bases: galaxy.datatypes.tabular.Tabular

    column_dataprovider (*args, **kwargs)

    data_sources = {'index': 'bigwig', 'data': 'bam'}

    dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7fc0782d5758>, 'chunk64': <function chunk64_dataprovider at 0x7fc0782d5758>}

    dataset_column_dataprovider (*args, **kwargs)

    dataset_dict_dataprovider (*args, **kwargs)

    dict_dataprovider (*args, **kwargs)

    display_peek (dataset)
        Returns formatted html of peek

    file_ext = 'sam'

    genomic_region_dataprovider (*args, **kwargs)

    genomic_region_dict_dataprovider (*args, **kwargs)

    header_dataprovider (*args, **kwargs)

    id_seq_qual_dataprovider (*args, **kwargs)

    line_dataprovider (*args, **kwargs)

    static merge (split_files, output_file)
        Multiple SAM files may each have headers. Since the headers should all be the same, remove the headers
        from files 1-n, keeping them in the first file only

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read
    regex_line_dataprovider (*args, **kwargs)

    set_meta (dataset, overwrite=True, skip=None, max_data_lines=5, **kwd)

    sniff (filename)
        Determines whether the file is in SAM format

        A file in SAM format consists of lines of tab-separated data. The following header line may be the first
        line:

```

@QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	MRNM	MPOS	ISIZE	SEQ	QUAL	
or											
@QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	MRNM	MPOS	ISIZE	SEQ	QUAL	OPT

Data in the OPT column is optional and can consist of tab-separated data

For complete details see <http://samtools.sourceforge.net/SAM1.pdf>

Rules for sniffing as True:

There must be 11 or more columns of data on each line
Columns 2 (FLAG), 4 (POS), 5 (MAPQ), 8 (MPOS), and 9 (ISIZE) must be numbers (9 can be negative)
We will only check that up to the first 5 alignments are correctly formatted.

```

>>> fname = get_test_fname( 'sequence.maf' )
>>> Sam().sniff( fname )
False
>>> fname = get_test_fname( '1.sam' )
>>> Sam().sniff( fname )
True

```

```

track_type = 'ReadTrack'

class galaxy.datatypes.tabular.Tabular (**kwd)
    Bases: galaxy.datatypes.data.Text

    Tab delimited data

    CHUNKABLE = True
        Add metadata elements

    as_gbrowse_display_file (dataset, **kwd)

    as_ucsc_display_file (dataset, **kwd)

    column_dataprovider (*args, **kwargs)
        Uses column settings that are passed in

    dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7fc078141b18>, 'chunk64': <function dataset_chunk64_dataprovider at 0x7fc078141b18>}

    dataset_column_dataprovider (*args, **kwargs)
        Attempts to get column settings from dataset.metadata

    dataset_dict_dataprovider (*args, **kwargs)
        Attempts to get column settings from dataset.metadata

    dict_dataprovider (*args, **kwargs)
        Uses column settings that are passed in

    display_data (trans, dataset, preview=False, filename=None, to_ext=None, chunk=None, **kwd)

    display_peek (dataset)
        Returns formatted html of peek

    displayable (dataset)

    get_chunk (trans, dataset, chunk)

    make_html_peek_header (dataset, skipchars=None, column_names=None, column_number_format='%s', column_parameter_alias=None, **kwargs)

    make_html_peek_rows (dataset, skipchars=None, **kwargs)

    make_html_table (dataset, **kwargs)
        Create HTML table, used for displaying peek

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of data lines to process

    set_meta (dataset, overwrite=True, skip=None, max_data_lines=100000, max_guess_type_data_lines=None, **kwd)
        Tries to determine the number of columns as well as those columns that contain numerical values in the dataset. A skip parameter is used because various tabular data types reuse this function, and their data type classes are responsible to determine how many invalid comment lines should be skipped. Using None for skip will cause skip to be zero, but the first line will be processed as a header. A max_data_lines parameter is used because various tabular data types reuse this function, and their data type classes are responsible to determine how many data lines should be processed to ensure that the non-optional metadata parameters are properly set; if used, optional metadata parameters will be set to None, unless the entire file has already been read. Using None for max_data_lines will process all data lines.

```

Items of interest:

1. We treat 'overwrite' as always True (we always want to set tabular metadata when called).
2. If a tabular file has no data, it will have one column of type 'str'.

3. We used to check only the first 100 lines when setting metadata and this class's `set_peek()` method read the entire file to determine the number of lines in the file. Since metadata can now be processed on cluster nodes, we've merged the line count portion of the `set_peek()` processing here, and we now check the entire contents of the file.

```
set_peek (dataset, line_count=None, is_multi_byte=False)
```

```
class galaxy.datatypes.tabular.Taxonomy (**kwd)  
    Bases: galaxy.datatypes.tabular.Tabular
```

```
display_peek (dataset)  
    Returns formatted html of peek
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read
```

```
class galaxy.datatypes.tabular.Vcf (**kwd)  
    Bases: galaxy.datatypes.tabular.Tabular
```

Variant Call Format for describing SNPs and other simple genome variations.

```
column_names = ['Chrom', 'Pos', 'ID', 'Ref', 'Alt', 'Qual', 'Filter', 'Info', 'Format', 'data']
```

```
data_sources = {'index': 'bigwig', 'data': 'tabix'}
```

```
dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7fc078141b18>, 'chunk64': <function chunk64_dataprovider at 0x7fc078141b18>}
```

```
display_peek (dataset)  
    Returns formatted html of peek
```

```
file_ext = 'vcf'
```

```
genomic_region_dataprovider (*args, **kwargs)
```

```
genomic_region_dict_dataprovider (*args, **kwargs)
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read
```

```
set_meta (dataset, **kwd)
```

```
sniff (filename)
```

```
track_type = 'VariantTrack'
```

tracks Module Datatype classes for tracks/track views within galaxy.

```
class galaxy.datatypes.tracks.GeneTrack (**kwargs)  
    Bases: galaxy.datatypes.binary.Binary
```

```
file_ext = 'genetrack'
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
```

xml Module XML format classes

```
class galaxy.datatypes.xml.CisML (**kwd)  
    Bases: galaxy.datatypes.xml.GenericXml
```

CisML XML data

```
file_ext = 'cismml'
```

```
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read
```

```
set_peek (dataset, is_multi_byte=False)  
    Set the peek and blurb text
```


sniff (*filename*)

class galaxy.datatypes.xml.**GenericXml** (***kwd*)
 Bases: [galaxy.datatypes.data.Text](#)

Base format class for any XML file.

dataproviders = {'xml': <function xml_dataprovider at 0x7fc077d77b90>, 'chunk64': <function chunk64_dataprovider at 0x7fc077d77b90>}

file_ext = 'xml'

static merge (*split_files*, *output_file*)

Merging multiple XML files is non-trivial and must be done in subclasses.

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read

set_peek (*dataset*, *is_multi_byte=False*)

Set the peek and blurb text

sniff (*filename*)

Determines whether the file is XML or not

```
>>> fname = get_test_fname( 'megablast_xml_parser_test1.blastxml' )
>>> GenericXml().sniff( fname )
True
>>> fname = get_test_fname( 'interval.interval' )
>>> GenericXml().sniff( fname )
False
```

xml_dataprovider (**args*, ***kwargs*)

class galaxy.datatypes.xml.**MEMEXml** (***kwd*)
 Bases: [galaxy.datatypes.xml.GenericXml](#)

MEME XML Output data

file_ext = 'memexml'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read

set_peek (*dataset*, *is_multi_byte=False*)

Set the peek and blurb text

sniff (*filename*)

class galaxy.datatypes.xml.**Owl** (***kwd*)
 Bases: [galaxy.datatypes.xml.GenericXml](#)

Web Ontology Language OWL format description <http://www.w3.org/TR/owl-ref/>

file_ext = 'owl'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read

set_peek (*dataset*, *is_multi_byte=False*)

sniff (*filename*)

Checking for keyword - 'owl' in the first 200 lines.

class galaxy.datatypes.xml.**Phyloxml** (***kwd*)
 Bases: [galaxy.datatypes.xml.GenericXml](#)

Format for defining phyloxml data <http://www.phyloxml.org/>

file_ext = 'phyloxml'

get_visualizations (*dataset*)

Returns a list of visualizations for datatype.

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data_lines** (MetadataParameter): Number of lines

set_peek (*dataset, is_multi_byte=False*)

Set the peek and blurb text

sniff (*filename*)

“Checking for keyword - ‘phyloxml’ always in lowercase in the first few lines

Subpackages

converters Package

bed_to_genetrack_converter Module

bed_to_gff_converter Module

bedgraph_to_array_tree_converter Module

bgzip Module Uses pysam to bgzip a file

usage: %prog in_file out_file

galaxy.datatypes.converters.bgzip.**main**()

fasta_to_len Module Input: fasta, int Output: tabular Return titles with lengths of corresponding seq

```
galaxy.datatypes.converters.fasta_to_len.compute_fasta_length (fasta_file,  
                                                                out_file,  
                                                                keep_first_char,  
                                                                keep_first_word=False)
```

fasta_to_tabular_converter Module Input: fasta, minimal length, maximal length Output: fasta Return sequences whose lengths are within the range.

fastq_to_fqtoc Module

galaxy.datatypes.converters.fastq_to_fqtoc.**main**()

The format of the file is JSON:

```
{ "sections" : [  
    { "start" : "x", "end" : "y", "sequences" : "z" },  
    ...  
]}
```

This works only for UNCOMPRESSED fastq files. The Python GzipFile does not provide seekable offsets via tell(), so clients just have to split the slow way

fastqsolexa_to_fasta_converter Module convert fastqsolexa file to separated sequence and quality files.

assume each sequence and quality score are contained in one line the order should be: 1st line: @title_of_seq 2nd line: nucleotides 3rd line: +title_of_qualityscore (might be skipped) 4th line: quality scores (in three forms: a. digits, b. ASCII codes, the first char as the coding base, c. ASCII codes without the first char.)

Usage: `%python fastqsolexa_to_fasta_converter.py <your_fastqsolexa_filename> <output_seq_filename> <output_score_filename>`

`galaxy.datatypes.converters.fastqsolexa_to_fasta_converter.stop_err(msg)`

fastqsolexa_to_qual_converter Module convert fastqsolexa file to separated sequence and quality files.

assume each sequence and quality score are contained in one line the order should be: 1st line: @title_of_seq 2nd line: nucleotides 3rd line: +title_of_qualityscore (might be skipped) 4th line: quality scores (in three forms: a. digits, b. ASCII codes, the first char as the coding base, c. ASCII codes without the first char.)

Usage: `%python fastqsolexa_to_qual_converter.py <your_fastqsolexa_filename> <output_seq_filename> <output_score_filename>`

`galaxy.datatypes.converters.fastqsolexa_to_qual_converter.stop_err(msg)`

gff_to_bed_converter Module

gff_to_interval_index_converter Module Convert from GFF file to interval index file.

usage: `python gff_to_interval_index_converter.py [input] [output]`

`galaxy.datatypes.converters.gff_to_interval_index_converter.main()`

interval_to_bed_converter Module

`galaxy.datatypes.converters.interval_to_bed_converter.stop_err(msg)`

interval_to_bedstrict_converter Module

`galaxy.datatypes.converters.interval_to_bedstrict_converter.force_bed_field_count(fields, re-gion_count, force_num_c`

`galaxy.datatypes.converters.interval_to_bedstrict_converter.stop_err(msg)`

interval_to_coverage Module Converter to generate 3 (or 4) column base-pair coverage from an interval file.

usage: `%prog bed_file out_file -1, -cols1=N,N,N,N: Columns for chrom, start, end, strand in interval file -2, -cols2=N,N,N,N: Columns for chrom, start, end, strand in coverage file`

`class galaxy.datatypes.converters.interval_to_coverage.CoverageWriter(out_stream=None, chrom-Col=0, position-Col=1, forward-Col=2, reverse-Col=3)`

Bases: object

close()

write(kwargs)**

galaxy.datatypes.converters.interval_to_coverage.**main**(*interval, coverage*)

Uses a sliding window of partitions to count coverages. Every interval record adds its start and end to the partitions. The result is a list of partitions, or every position that has a (maybe) different number of basepairs covered. We don't worry about merging because we pop as the sorted intervals are read in. As the input start positions exceed the partition positions in partitions, coverages are kicked out in bulk.

interval_to_fli Module Creates a feature location index (FLI) for a given BED/GFF file. FLI index has the form:

```
[line_length]
<symbol1_in_lowercase><tab><symbol1><tab><location>
<symbol2_in_lowercase><tab><symbol2><tab><location>
...
```

where location is formatted as:

contig:start-end

and symbols are sorted in lexicographical order.

galaxy.datatypes.converters.interval_to_fli.**main**()

interval_to_interval_index_converter Module Convert from interval file to interval index file.

usage: %prog <options> in_file out_file -c, --chr-col: chromosome column, default=1 -s, --start-col: start column, default=2 -e, --end-col: end column, default=3

galaxy.datatypes.converters.interval_to_interval_index_converter.**main**()

interval_to_summary_tree_converter Module

interval_to_tabix_converter Module Uses pysam to index a bgzipped interval file with tabix Supported presets: bed, gff, vcf

usage: %prog in_file out_file

galaxy.datatypes.converters.interval_to_tabix_converter.**main**()

lped_to_fped_converter Module

galaxy.datatypes.converters.lped_to_fped_converter.**main**()

call fbat need to work with rgenetics composite datatypes so in and out are html files with data in extrafiles path
<command interpreter="python">rg_convert_lped_fped.py '\$input1/\$input1.metadata.base_name' '\$output1' '\$output1.extra_files_path' </command>

galaxy.datatypes.converters.lped_to_fped_converter.**rgConv**(*inpedfilepath, outhtml-name, outfilepath*)

convert linkage ped/map to fbat

galaxy.datatypes.converters.lped_to_fped_converter.**timenow**()

return current time as a string

lped_to_pbed_converter Module

```
galaxy.datatypes.converters.lped_to_pbed_converter.getMissval(inped='')
    read some lines...ugly hack - try to guess missing value should be N or 0 but might be . or -
galaxy.datatypes.converters.lped_to_pbed_converter.main()
    need to work with rgenetics composite datatypes so in and out are html files with data in extrafiles path
    <command interpreter="python">lped_to_pbed_converter.py '$input1/$input1.metadata.base_name' '$out-
    put1' '$output1.extra_files_path' '${GALAXY_DATA_INDEX_DIR}/rg/bin/plink' </command>

galaxy.datatypes.converters.lped_to_pbed_converter.rgConv(inpedfilepath,      out-
                                                         htmlname, outfilepath,
                                                         plink)

galaxy.datatypes.converters.lped_to_pbed_converter.timenow()
    return current time as a string
```

maf_to_fasta_converter Module**maf_to_interval_converter Module****pbed_ldreduced_converter Module**

```
galaxy.datatypes.converters.pbed_ldreduced_converter.main()
    need to work with rgenetics composite datatypes so in and out are html files with data in extrafiles path
galaxy.datatypes.converters.pbed_ldreduced_converter.makeLDreduced(basename,
                                                                    inf-
                                                                    path=None,
                                                                    outf-
                                                                    path=None,
                                                                    plinke='plink',
                                                                    forcere-
                                                                    build=False,
                                                                    returnF-
                                                                    name=False,
                                                                    win-
                                                                    size='60',
                                                                    win-
                                                                    move='40',
                                                                    r2thresh='0.1')

    not there so make and leave in output dir for post job hook to copy back into input extra files path for next time
galaxy.datatypes.converters.pbed_ldreduced_converter.pruneLD(plinktasks=[],
                                                             cd='./',
                                                             vclbase=[])

galaxy.datatypes.converters.pbed_ldreduced_converter.timenow()
    return current time as a string
```

pbed_to_lped_converter Module

```
galaxy.datatypes.converters.pbed_to_lped_converter.main()
    need to work with rgenetics composite datatypes so in and out are html files with data in extrafiles path
    <command interpreter="python">pbed_to_lped_converter.py '$input1/$input1.metadata.base_name' '$out-
    put1' '$output1.extra_files_path' '${GALAXY_DATA_INDEX_DIR}/rg/bin/plink' </command>
galaxy.datatypes.converters.pbed_to_lped_converter.rgConv(inpedfilepath,      out-
                                                         htmlname, outfilepath,
                                                         plink)
```

```
galaxy.datatypes.converters.pbed_to_lped_converter.timenow()  
    return current time as a string
```

picard_interval_list_to_bed6_converter Module

sam_or_bam_to_summary_tree_converter Module

sam_to_bam Module A wrapper script for converting SAM to BAM, with sorting. %prog input_filename.sam
output_filename.bam

```
galaxy.datatypes.converters.sam_to_bam.cleanup_before_exit (tmp_dir)
```

vcf_to_interval_index_converter Module Convert from VCF file to interval index file.

```
galaxy.datatypes.converters.vcf_to_interval_index_converter.main()
```

vcf_to_summary_tree_converter Module

vcf_to_vcf_bgzip Module Uses pysam to bgzip a vcf file as-is. Headers, which are important, are kept. Original ordering, which may be specifically needed by tools or external display applications, is also maintained.

usage: %prog in_file out_file

```
galaxy.datatypes.converters.vcf_to_vcf_bgzip.main()
```

wiggle_to_array_tree_converter Module

```
galaxy.datatypes.converters.wiggle_to_array_tree_converter.main()
```

wiggle_to_simple_converter Module Read a wiggle track and print out a series of lines containing “chrom position score”. Ignores track lines, handles bed, variableStep and fixedStep wiggle lines.

```
galaxy.datatypes.converters.wiggle_to_simple_converter.main()
```

```
galaxy.datatypes.converters.wiggle_to_simple_converter.stop_err (msg)
```

display_applications Package

application Module

```
class galaxy.datatypes.display_applications.application.DisplayApplication (display_id,  
                                                                           name,  
                                                                           datatypes_registry,  
                                                                           ver-  
                                                                           sion=None)
```

Bases: object

```
filter_by_dataset (data, trans)
```

```
classmethod from_elem (elem, datatypes_registry)
```

```
classmethod from_file (filename, datatypes_registry)
```

```
get_link (link_name, data, dataset_hash, user_hash, trans, app_kwds)
```

```

class galaxy.datatypes.display_applications.application.DisplayApplicationLink (display_application)
    Bases: object

    build_parameter_dict (data, dataset_hash, user_hash, trans, app_kwds)

    filter_by_dataset (data, trans)

    classmethod from_elem (elem, display_application, other_values=None)

    get_display_url (data, trans)

    get_initail_values (data, trans)

class galaxy.datatypes.display_applications.application.DynamicDisplayApplicationBuilder (elem,
                                                                                               dis-
                                                                                               play_
                                                                                               build,

    Bases: object

class galaxy.datatypes.display_applications.application.PopulatedDisplayApplicationLink (display
                                                                                               data,
                                                                                               dataset
                                                                                               user_h
                                                                                               trans,
                                                                                               app_kv

    Bases: object

    display_ready ()

    display_url ()

    get_param_name_by_url (url)

    get_param_value (name)

    prepare_display ()

    preparing_display ()

```

parameters Module

```

class galaxy.datatypes.display_applications.parameters.DisplayApplicationDataParameter (elem,
                                                                                               link)
    Bases: galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter

    Parameter that returns a file_name containing the requested content

    formats

    get_value (other_values, dataset_hash, user_hash, trans)

    is_preparing (other_values)

    prepare (other_values, dataset_hash, user_hash, trans)

    ready (other_values)

    type = 'data'

class galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter (elem,
                                                                                               link)
    Bases: object

    Abstract Class for Display Application Parameters

    build_url (other_values)

    classmethod from_elem (elem, link)

```

```
get_value (other_values, dataset_hash, user_hash, trans)
```

```
is_preparing (other_values)
```

```
prepare (other_values, dataset_hash, user_hash, trans)
```

```
ready (other_values)
```

```
type = None
```

```
class galaxy.datatypes.display_applications.parameters.DisplayApplicationTemplateParameter (element, line)
```

Bases: *galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter*

Parameter that returns a string containing the requested content

```
get_value (other_values, dataset_hash, user_hash, trans)
```

```
type = 'template'
```

```
class galaxy.datatypes.display_applications.parameters.DisplayDataValueWrapper (value,
```

```
parameter,
other_values,
dataset_hash,
user_hash,
trans)
```

Bases: *galaxy.datatypes.display_applications.parameters.DisplayParameterValueWrapper*

```
ACTION_NAME = 'data'
```

```
action_name
```

```
mime_type ()
```

```
qp
```

```
class galaxy.datatypes.display_applications.parameters.DisplayParameterValueWrapper (value,
```

```
parameter,
other_values,
dataset_hash,
user_hash,
trans)
```

Bases: *object*

```
ACTION_NAME = 'param'
```

```
action_name
```

```
mime_type ()
```

```
qp
```

```
url
```

util Module


```
galaxy.datatypes.display_applications.util.decode_dataset_user(trans,
                                                              dataset_hash,
                                                              user_hash)
galaxy.datatypes.display_applications.util.encode_dataset_user(trans, dataset,
                                                                user)
```

util Package

util Package Utilities for Galaxy datatypes.

gff_util Module Provides utilities for working with GFF files.

```
class galaxy.datatypes.util.gff_util.GFFFeature(reader, chrom_col=0, feature_col=2,
                                                start_col=3, end_col=4, strand_col=6,
                                                score_col=5, default_strand='.',
                                                fix_strand=False, intervals=[],
                                                raw_size=0)
```

Bases: `galaxy.datatypes.util.gff_util.GFFInterval`

A GFF feature, which can include multiple intervals.

copy()

lines()

name()

Returns feature's name.

```
class galaxy.datatypes.util.gff_util.GFFInterval(reader, fields, chrom_col=0, feature_col=2, start_col=3, end_col=4,
                                                strand_col=6, score_col=5, default_strand='.', fix_strand=False)
```

Bases: `bx.intervals.io.GenomicInterval`

A GFF interval, including attributes. If file is strictly a GFF file, only attribute is 'group.'

copy()

```
class galaxy.datatypes.util.gff_util.GFFIntervalToBEDReaderWrapper(reader,
                                                                    **kwargs)
```

Bases: `bx.intervals.io.NiceReaderWrapper`

Reader wrapper that reads GFF intervals/lines and automatically converts them to BED format.

parse_row(line)

```
class galaxy.datatypes.util.gff_util.GFFReaderWrapper(reader, chrom_col=0, feature_col=2, start_col=3,
                                                         end_col=4, strand_col=6,
                                                         score_col=5, fix_strand=False,
                                                         convert_to_bed_coord=False,
                                                         **kwargs)
```

Bases: `bx.intervals.io.NiceReaderWrapper`

Reader wrapper for GFF files.

Wrapper has two major functions:

1. group entries for GFF file (via group column), GFF3 (via id attribute), or GTF (via gene_id/transcript id);

2.convert coordinates from GFF format—starting and ending coordinates are 1-based, closed—to the ‘traditional’/BED interval format—0 based, half-open. This is useful when using GFF files as inputs to tools that expect traditional interval format.

next ()

Returns next GFFFeature.

parse_row (line)

`galaxy.datatypes.util.gff_util.convert_bed_coords_to_gff(interval)`

Converts an interval object’s coordinates from BED format to GFF format. Accepted object types include GenomicInterval and list (where the first element in the list is the interval’s start, and the second element is the interval’s end).

`galaxy.datatypes.util.gff_util.convert_gff_coords_to_bed(interval)`

Converts an interval object’s coordinates from GFF format to BED format. Accepted object types include GFFFeature, GenomicInterval, and list (where the first element in the list is the interval’s start, and the second element is the interval’s end).

`galaxy.datatypes.util.gff_util.gff_attributes_to_str(attrs, gff_format)`

Convert GFF attributes to string. Supported formats are GFF3, GTF.

`galaxy.datatypes.util.gff_util.parse_gff_attributes(attr_str)`

Parses a GFF/GTF attribute string and returns a dictionary of name-value pairs. The general format for a GFF3 attributes string is

name1=value1;name2=value2

The general format for a GTF attribute string is

name1 “value1” ; name2 “value2”

The general format for a GFF attribute string is a single string that denotes the interval’s group; in this case, method returns a dictionary with a single key-value pair, and key name is ‘group’

`galaxy.datatypes.util.gff_util.read_unordered_gtf(iterator, strict=False)`

Returns GTF features found in an iterator. GTF lines need not be ordered or clustered for reader to work. Reader returns GFFFeature objects sorted by transcript_id, chrom, and start position.

image_util Module Provides utilities for working with image files.

`galaxy.datatypes.util.image_util.check_image_type(filename, types, image=None)`

`galaxy.datatypes.util.image_util.get_image_ext(file_path, image)`

`galaxy.datatypes.util.image_util.image_type(filename, image=None)`

eggs Package

eggs Package Manage Galaxy eggs

class `galaxy.eggs.CaseSensitiveConfigParser` (*defaults=None, dict_type=<class ‘collections.OrderedDict’>, allow_no_value=False*)

Bases: `ConfigParser.SafeConfigParser`

optionxform (*optionstr*)

class `galaxy.eggs.Crate` (*galaxy_config_file, platform=None*)

Bases: `object`

Reads the eggs.ini file for use with checking and fetching.

all_eggs

Return a list of all eggs in the crate.

```

all_missing
    Return true if any eggs in the eggs config file are missing.

all_names
    Return a list of names of all eggs in the crate.

config_eggs
    Return a list of all eggs in the crate that are needed based on the options set in the Galaxy config file.

config_file = '/var/build/user_builds/jmchilton-galaxy/checkouts/stable/eggs.ini'

config_missing
    Return true if any eggs are missing, conditional on options set in the Galaxy config file.

config_names
    Return a list of names of all eggs in the crate that are needed based on the options set in the Galaxy config
    file.

parse()

parse_egg_section(eggs, tags, full_platform=False, egg_class=<class 'galaxy.eggs.Egg'>)

resolve(all=False)
    Try to resolve (e.g. fetch) all eggs in the crate.

class galaxy.eggs.Egg(name=None, version=None, tag=None, url=None, platform=None, crate=None)
    Bases: object

    Contains information about locating and downloading eggs.

    fetch() serves as the install method to pkg_resources.working_set.resolve()

    path
        Return the path of the egg, if it exists, or None

    remove_doppelgangers()

    require()

    resolve()

    set_dir()

    set_distribution()
        Stores a pkg_resources Distribution object for reference later

    unpack_if_needed()

    version_conflict(conflict_dist, conflict_req)

exception galaxy.eggs.EggNotFetchable(eggs)
    Bases: exceptions.Exception

class galaxy.eggs.GalaxyConfig(config_file)
    Bases: object

    always_conditional = ('pysam', 'ctypes', 'python_daemon')

    check_conditional(egg_name)

class galaxy.eggs.URLRetriever(*args, **kwargs)
    Bases: urllib.FancyURLopener

    http_error_default(*args)

galaxy.eggs.get_env()

```

```
galaxy.eggs.remove_file_or_path(f)
galaxy.eggs.require(req_str)
galaxy.eggs.unpack_zipfile(filename, extract_dir, ignores=[])
```

dist Module Manage Galaxy eggs

```
class galaxy.eggs.dist.DistScrambleCrate(galaxy_config_file, build_on='all')
    Bases: galaxy.eggs.scramble.ScrambleCrate
    Holds eggs with info on how to build them for distribution.

    dist_config_file = '/var/build/user_builds/jmchilton-galaxy/checkouts/stable/dist-eggs.ini'
    get_platforms(wanted)
    parse()
    parse_egg_section(eggs, tags, full_platform=False)
class galaxy.eggs.dist.DistScrambleEgg(*args, **kwargs)
    Bases: galaxy.eggs.scramble.ScrambleEgg
    path
    run_scramble_script()
    set_dir()
    unpack_if_needed()
```

scramble Module Manage Galaxy eggs

```
class galaxy.eggs.scramble.ScrambleCrate(galaxy_config_file, platform=None)
    Bases: galaxy.eggs.__init__.Crate
    Reads the eggs.ini file for use with scrambling eggs.

    parse()
    parse_egg_section(*args, **kwargs)
    scramble(all=False)
class galaxy.eggs.scramble.ScrambleEgg(*args, **kwargs)
    Bases: galaxy.eggs.__init__.Egg
    Contains information about scrambling eggs.

    archive_dir = '/var/build/user_builds/jmchilton-galaxy/checkouts/stable/scripts/scramble/archives'
    build_dir = '/var/build/user_builds/jmchilton-galaxy/checkouts/stable/scripts/scramble/build'
    copy_build_script()
    ez_setup = '/var/build/user_builds/jmchilton-galaxy/checkouts/stable/scripts/scramble/lib/ez_setup.py'
    ez_setup_url = 'http://peak.telecommunity.com/dist/ez_setup.py'
    fetch_one(urls)
        Fetches the first available archive out of a list.
    fetch_source()
        Get egg (and dependent) source
    get_tld(names)
```

```

run_scramble_script()

scramble()

scramble_dir = '/var/build/user_builds/jmchilton-galaxy/checkouts/stable/scripts/scramble'
script_dir = '/var/build/user_builds/jmchilton-galaxy/checkouts/stable/scripts/scramble/scripts'

unpack_source()

unpack_tar()

unpack_zip()

exception galaxy.eggs.scramble.ScrambleFailure(eggs, msg=None)
    Bases: exceptions.Exception

```

exceptions Package

exceptions Package Custom exceptions for Galaxy

```

exception galaxy.exceptions.ActionInputError(err_msg, type='error')
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 400

exception galaxy.exceptions.AdminRequiredException(err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 403

exception galaxy.exceptions.AuthenticationFailed(err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 401

exception galaxy.exceptions.AuthenticationRequired(err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 403

exception galaxy.exceptions.ConfigDoesNotAllowException(err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 403

exception galaxy.exceptions.Conflict(err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 409

```

```
exception galaxy.exceptions.DeprecatedMethod (err_msg=None,      type='info',      **ex-
                                             tra_error_info)
Bases: galaxy.exceptions.MessageException
Method (or a particular form/arg signature) has been removed and won't be available later
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
status_code = 404

exception galaxy.exceptions.DuplicatedSlugException (err_msg=None, type='info', **ex-
                                                  tra_error_info)
Bases: galaxy.exceptions.MessageException
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
status_code = 400

exception galaxy.exceptions.InconsistentDatabase (err_msg=None,   type='info',   **ex-
                                                  tra_error_info)
Bases: galaxy.exceptions.MessageException
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
status_code = 500

exception galaxy.exceptions.InsufficientPermissionsException (err_msg=None,
                                                             type='info',
                                                             **ex-
                                                             tra_error_info)
Bases: galaxy.exceptions.MessageException
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
status_code = 403

exception galaxy.exceptions.InternalServerError (err_msg=None,   type='info',   **ex-
                                                  tra_error_info)
Bases: galaxy.exceptions.MessageException
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
status_code = 500

exception galaxy.exceptions.ItemAccessibilityException (err_msg=None, type='info',
                                                         **extra_error_info)
Bases: galaxy.exceptions.MessageException
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
status_code = 403

exception galaxy.exceptions.ItemDeletionException (err_msg=None,   type='info',   **ex-
                                                  tra_error_info)
Bases: galaxy.exceptions.MessageException

exception galaxy.exceptions.ItemOwnershipException (err_msg=None, type='info', **ex-
                                                  tra_error_info)
Bases: galaxy.exceptions.MessageException
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
status_code = 403

exception galaxy.exceptions.MalformedId (err_msg=None, type='info', **extra_error_info)
Bases: galaxy.exceptions.MessageException
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
```

```

    status_code = 400

exception galaxy.exceptions.MessageException (err_msg=None,          type='info',          **ex-
                                             tra_error_info)

Bases: exceptions.Exception

Exception to make throwing errors from deep in controllers easier.

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

exception galaxy.exceptions.NotImplemented (err_msg=None, type='info', **extra_error_info)
Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 501

exception galaxy.exceptions.ObjectAttributeInvalidException (err_msg=None,
                                                            type='info',          **ex-
                                                            tra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

exception galaxy.exceptions.ObjectAttributeMissingException (err_msg=None,
                                                            type='info',          **ex-
                                                            tra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

exception galaxy.exceptions.ObjectInvalid
Bases: exceptions.Exception

Accessed object store ID is invalid

exception galaxy.exceptions.ObjectNotFound (err_msg=None, type='info', **extra_error_info)
Bases: galaxy.exceptions.MessageException

Accessed object was not found

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 404

exception galaxy.exceptions.RequestParameterInvalidException (err_msg=None,
                                                              type='info',          **ex-
                                                              tra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

exception galaxy.exceptions.RequestParameterMissingException (err_msg=None,
                                                              type='info',          **ex-
                                                              tra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

```

```
status_code = 400
```

```
exception galaxy.exceptions.ToolMetaParameterException (err_msg=None, type='info',
                                                         **extra_error_info)
```

```
Bases: galaxy.exceptions.MessageException
```

```
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
```

```
status_code = 400
```

```
exception galaxy.exceptions.UnknownContentsType (err_msg=None, type='info', **ex-
                                                  tra_error_info)
```

```
Bases: galaxy.exceptions.MessageException
```

```
err_code = <galaxy.exceptions.error_codes.ErrorCode object>
```

```
status_code = 400
```

external_services Package

actions Module

```
class galaxy.external_services.actions.ExternalServiceAction (elem, parent)
```

```
Bases: object
```

```
Abstract Class for External Service Actions
```

```
classmethod from_elem (elem, parent)
```

```
get_action_access_link (trans, param_dict)
```

```
handle_action (completed_action, param_dict, trans)
```

```
perform_action (param_dict)
```

```
populate_action (param_dict)
```

```
type = None
```

```
class galaxy.external_services.actions.ExternalServiceResult (name, param_dict)
```

```
Bases: object
```

```
content
```

```
class galaxy.external_services.actions.ExternalServiceTemplateAction (elem, par-
                                                                      ent)
```

```
Bases: galaxy.external_services.actions.ExternalServiceAction
```

```
Action that redirects to an external URL
```

```
perform_action (param_dict)
```

```
type = 'template'
```

```
class galaxy.external_services.actions.ExternalServiceValueResult (name,
                                                                    param_dict,
                                                                    value)
```

```
Bases: galaxy.external_services.actions.ExternalServiceResult
```

```
content
```

```
class galaxy.external_services.actions.ExternalServiceWebAPIAction (elem, parent)
```

```
Bases: galaxy.external_services.actions.ExternalServiceAction
```

```
Action that accesses an external Web API and provides handlers for the requested content
```



```

class ExternalServiceWebAPIActionRequest (elem, parent)
    Bases: object

    get_web_api_action (param_dict)

    ExternalServiceWebAPIAction.perform_action (param_dict)

    ExternalServiceWebAPIAction.type = 'web_api'
class galaxy.external_services.actions.ExternalServiceWebAPIActionResult (name,
                                                                              param_dict,
                                                                              url,
                                                                              method,
                                                                              tar-
                                                                              get)

    Bases: galaxy.external_services.actions.ExternalServiceResult

    content

class galaxy.external_services.actions.ExternalServiceWebAction (elem, parent)
    Bases: galaxy.external_services.actions.ExternalServiceAction

    Action that accesses an external web application

    get_action_access_link (trans, param_dict)

    type = 'web'
class galaxy.external_services.actions.PopulatedExternalServiceAction (action,
                                                                           param_dict)

    Bases: object

    get_action_access_link (trans)

    handle_results (trans)

    perform_action ()
class galaxy.external_services.actions.Template (elem, parent)
    Bases: object

    build_template (param_dict)

parameters Module
class galaxy.external_services.parameters.ExternalServiceParameter (elem, parent)
    Bases: object

    Abstract Class for External Service Parameters

    classmethod from_elem (elem, parent)

    get_value (param_dict)

    requires_user_input = False

    type = None
class galaxy.external_services.parameters.ExternalServiceTemplateParameter (elem,
                                                                               par-
                                                                               ent)

    Bases: galaxy.external_services.parameters.ExternalServiceParameter

    Parameter that returns a string containing the requested content

    get_value (param_dict)

```

```
type = 'template'
```

service Module

```
class galaxy.external_services.service.ActionSection (name, label)
    Bases: list

    has_action ()

class galaxy.external_services.service.BooleanExternalServiceActionsGroupWhen (parent,
                                                                                   name,
                                                                                   value,
                                                                                   la-
                                                                                   bel=None)

    Bases: galaxy.external_services.service.ExternalServiceActionsGroupWhen

    classmethod from_elem (parent, elem)
        Returns an instance of this when

    is_case (param_dict)

    type = 'boolean'

class galaxy.external_services.service.ExternalServiceActionsConditional (elem,
                                                                           par-
                                                                           ent)

    Bases: object

    get_current_cases (param_dict)

    type = 'conditional'

class galaxy.external_services.service.ExternalServiceActionsGroup (parent, name,
                                                                      label=None)

    Bases: object

    add_item (item)

    classmethod from_elem (elem, parent=None)
        Return ExternalServiceActionsGroup created from an xml element.

    load_sub_elems (elem)

    populate (service_instance, item=None, param_dict=None)

    prepare_actions (param_dict, parent_dict, parent_section)

class galaxy.external_services.service.ExternalServiceActionsGroupWhen (parent,
                                                                           name,
                                                                           la-
                                                                           bel=None)

    Bases: galaxy.external_services.service.ExternalServiceActionsGroup

    classmethod from_elem (parent, elem)
        Loads the proper when by attributes of elem

    get_ref (param_dict)

    is_case (param_dict)

    type = 'when'
```

```
class galaxy.external_services.service.ItemIsInstanceExternalServiceActionsGroupWhen (parent,
                                                                                       name,
                                                                                       value,
                                                                                       la-
                                                                                       bel=None)
```

Bases: *galaxy.external_services.service.ExternalServiceActionsGroupWhen*

classmethod from_elem (parent, elem)
Returns an instance of this when

is_case (param_dict)

type = 'item_type'

```
class galaxy.external_services.service.PopulatedExternalService (service_group,
                                                                 service_instance,
                                                                 item,
                                                                 param_dict=None)
```

Bases: object

get_action_by_name (actions_list)

get_action_links ()

perform_action_by_name (actions_list)

populate ()

```
class galaxy.external_services.service.ValueExternalServiceActionsGroupWhen (parent,
                                                                                   name,
                                                                                   value,
                                                                                   la-
                                                                                   bel=None)
```

Bases: *galaxy.external_services.service.ExternalServiceActionsGroupWhen*

classmethod from_elem (parent, elem)
Returns an instance of this when

is_case (param_dict)

type = 'value'

```
galaxy.external_services.service.class_type
alias of ItemIsInstanceExternalServiceActionsGroupWhen
```

Subpackages

result_handlers Package

basic Module

```
class galaxy.external_services.result_handlers.basic.ExternalServiceActionJQueryGridResultHan
```

Bases: *galaxy.external_services.result_handlers.basic.ExternalServiceActionResultHandler*

Class for External Service Actions JQuery Result Handler

handle_result (result, param_dict, trans)

type = 'jquery_grid'

```
class galaxy.external_services.result_handlers.basic.ExternalServiceActionJSONResultHandler (elem,
```

Bases: *galaxy.external_services.result_handlers.basic.ExternalServiceActionResultHandler*

Class for External Service Actions JQuery Result Handler

```
    handle_result (result, param_dict, trans)
```

```
    type = 'json_display'
```

```
class galaxy.external_services.result_handlers.basic.ExternalServiceActionResultHandler (elem,
                                                                                          par-
                                                                                          ent)
```

Bases: object

Basic Class for External Service Actions Result Handlers

```
    classmethod from_elem (elem, parent)
```

```
    handle_result (result, param_dict, trans)
```

```
    type = 'display'
```

```
class galaxy.external_services.result_handlers.basic.ExternalServiceActionURLRedirectResultHa
```

Bases: *galaxy.external_services.result_handlers.basic.ExternalServiceActionResultHandler*

Basic Class for External Service Actions Result Handlers

```
    classmethod from_elem (elem, parent)
```

```
    handle_result (result, param_dict, trans)
```

```
    type = 'web_redirect'
```

```
galaxy.external_services.result_handlers.basic.handler_class
    alias of ExternalServiceActionJSONResultHandler
```

forms Package

forms Module FormDefinition and field factories

```
class galaxy.forms.forms.FormDefinitionAddressFieldFactory
```

Bases: *galaxy.forms.forms.FormDefinitionFieldFactory*

```
    from_elem (elem, layout=None)
```

Return FormDefinition field created from an xml element.

```
    new (name=None, label=None, required=False, helptext=None, default=None, visible=True, lay-
        out=None)
```

Return new FormDefinition field.

```
    type = 'address'
```

```
class galaxy.forms.forms.FormDefinitionFactory (form_types, field_type_factories)
```

Bases: object

```
    from_elem (elem, form_definition_current=None)
```

Return FormDefinition created from an xml element.

```
    new (form_type, name, description=None, fields=None, layout=None, form_definition_current=None)
```

Return new FormDefinition.

```

class galaxy.forms.forms.FormDefinitionFieldFactory
    Bases: object

    from_elem (elem, layout=None)
        Return FormDefinition created from an xml element.

    new (name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None)
        Return new FormDefinition field.

    type = None

class galaxy.forms.forms.FormDefinitionHistoryFieldFactory
    Bases: galaxy.forms.forms.FormDefinitionFieldFactory

    from_elem (elem, layout=None)
        Return FormDefinition field created from an xml element.

    new (label=None, required=False, helptext=None, default=None, visible=True, layout=None)
        Return new FormDefinition field.

    type = 'history'

class galaxy.forms.forms.FormDefinitionPasswordFieldFactory
    Bases: galaxy.forms.forms.FormDefinitionFieldFactory

    from_elem (elem, layout=None)
        Return FormDefinition field created from an xml element.

    new (name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None, area=False)
        Return new FormDefinition field.

    type = 'password'

class galaxy.forms.forms.FormDefinitionSelectFieldFactory
    Bases: galaxy.forms.forms.FormDefinitionFieldFactory

    from_elem (elem, layout=None)
        Return FormDefinition field created from an xml element.

    new (name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None, options=[], checkboxes=False)
        Return new FormDefinition field.

    type = 'select'

class galaxy.forms.forms.FormDefinitionTextFieldFactory
    Bases: galaxy.forms.forms.FormDefinitionFieldFactory

    from_elem (elem, layout=None)
        Return FormDefinition field created from an xml element.

    new (name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None, area=False)
        Return new FormDefinition field.

    type = 'text'

class galaxy.forms.forms.FormDefinitionWorkflowFieldFactory
    Bases: galaxy.forms.forms.FormDefinitionFieldFactory

    from_elem (elem, layout=None)
        Return FormDefinition field created from an xml element.

```

new (*name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None*)
Return new FormDefinition field.

type = 'workflow'

class galaxy.forms.forms.**FormDefinitionWorkflowMappingFieldFactory**

Bases: *galaxy.forms.forms.FormDefinitionFieldFactory*

from_elem (*elem, layout=None*)

Return FormDefinition field created from an xml element.

new (*name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None*)
Return new FormDefinition field.

type = 'workflowmapping'

galaxy.forms.forms.**field**

alias of *FormDefinitionHistoryFieldFactory*

jobs Package

jobs Package Support for running a tool in Galaxy via an internal job management system

class galaxy.jobs.**ComputeEnvironment**

Bases: object

Definition of the job as it will be run on the (potentially) remote compute server.

config_directory ()

Directory containing config files (potentially remote)

input_paths ()

Input DatasetPaths defined by job.

new_file_path ()

Location to dump new files for this job on remote server.

output_paths ()

Output DatasetPaths defined by job.

sep ()

os.path.sep for the platform this job will execute in.

unstructured_path_rewriter ()

Return a function that takes in a value, determines if it is path to be rewritten (will be passed non-path values as well - onus is on this function to determine both if its input is a path and if it should be rewritten.)

version_path ()

Location of the version file for the underlying tool.

working_directory ()

Job working directory (potentially remote)

class galaxy.jobs.**JobConfiguration** (*app*)

Bases: object

A parser and interface to advanced job management features.

These features are configured in the job configuration, by default, `job_conf.xml`

DEFAULT_NWORKERS = 4

convert_legacy_destinations (*job_runners*)

Converts legacy (from a URL) destinations to contain the appropriate runner params defined in the URL.

Parameters **job_runners** (*list of job runner plugins*) – All loaded job runner plugins.

default_job_tool_configuration

The default JobToolConfiguration, used if a tool does not have an explicit definition in the configuration. It consists of a reference to the default handler and default destination.

Returns JobToolConfiguration – a representation of a <tool> element that uses the default handler and destination

get_destination (*id_or_tag*)

Given a destination ID or tag, return the JobDestination matching the provided ID or tag

Parameters **id_or_tag** (*str*) – A destination ID or tag.

Returns JobDestination – A valid destination

Destinations are deepcopied as they are expected to be passed in to job runners, which will modify them for persisting params set at runtime.

get_destinations (*id_or_tag*)

Given a destination ID or tag, return all JobDestinations matching the provided ID or tag

Parameters **id_or_tag** (*str*) – A destination ID or tag.

Returns list or tuple of JobDestinations

Destinations are not deepcopied, so they should not be passed to anything which might modify them.

get_handler (*id_or_tag*)

Given a handler ID or tag, return the provided ID or an ID matching the provided tag

Parameters **id_or_tag** (*str*) – A handler ID or tag.

Returns str – A valid job handler ID.

get_job_runner_plugins (*handler_id*)

Load all configured job runner plugins

Returns list of job runner plugins

get_job_tool_configurations (*ids*)

Get all configured JobToolConfigurations for a tool ID, or, if given a list of IDs, the JobToolConfigurations for the first id in *ids* matching a tool definition.

Note: You should not mix tool shed tool IDs, versionless tool shed IDs, and tool config tool IDs that refer to the same tool.

Parameters **ids** (*list or str*) – Tool ID or IDs to fetch the JobToolConfiguration of.

Returns list – JobToolConfiguration Bunches representing <tool> elements matching the specified ID(s).

Example tool ID strings include:

- Full tool shed id: `toolshed.example.org/repos/nate/filter_tool_repo/filter_tool/1.0.0`
- Tool shed id less version: `toolshed.example.org/repos/nate/filter_tool_repo/filter_tool`
- Tool config tool id: `filter_tool`

get_tool_resource_parameters (*tool_id*)

Given a tool id, return XML elements describing parameters to insert into job resources.

Tool id A tool ID (a string)

Returns List of parameter elements.

is_handler (*server_name*)

Given a server name, indicate whether the server is a job handler

Parameters **server_name** (*str*) – The name to check

Returns bool

is_id (*collection*)

Given a collection of handlers or destinations, indicate whether the collection represents a tag or a real ID

Parameters **collection** (*tuple or list*) – A representation of a destination or handler

Returns bool

is_tag (*collection*)

Given a collection of handlers or destinations, indicate whether the collection represents a tag or a real ID

Parameters **collection** (*tuple or list*) – A representation of a destination or handler

Returns bool

class `galaxy.jobs.JobDestination` (***kws*)

Bases: `galaxy.util.bunch.Bunch`

Provides details about where a job runs

class `galaxy.jobs.JobToolConfiguration` (***kws*)

Bases: `galaxy.util.bunch.Bunch`

Provides details on what handler and destination a tool should use

A JobToolConfiguration will have the required attribute ‘id’ and optional attributes ‘handler’, ‘destination’, and ‘params’

get_resource_group ()

class `galaxy.jobs.JobWrapper` (*job, queue, use_persisted_destination=False*)

Bases: `object`

Wraps a ‘model.Job’ with convenience methods for running processes and state management.

can_split ()

change_ownership_for_run ()

change_state (*state, info=False*)

check_limits (*runtime=None*)

check_tool_output (*stdout, stderr, tool_exit_code, job*)

cleanup (*delete_files=True*)

compute_outputs ()

default_compute_environment (*job=None*)

fail (*message, exception=False, stdout='', stderr='', exit_code=None*)

Indicate job failure by setting state and message on all output datasets.

finish (*stdout, stderr, tool_exit_code=None, remote_working_directory=None*)

Called to indicate that the associated command has been run. Updates the output datasets based on stderr and stdout from the command, and the contents of the output files.

galaxy_system_pwent

get_command_line ()

get_dataset_finish_context (*job_context, dataset*)

get_env_setup_clause ()

get_id_tag ()

get_input_dataset_fnames (*ds*)

get_input_fnames ()

get_input_paths (*job=None*)

get_job ()

get_job_runner ()

get_job_runner_url ()

get_mutable_output_fnames ()

get_output_destination (*output_path*)

Destination for outputs marked as from_work_dir. This is the normal case, just copy these files directly to the ultimate destination.

get_output_file_id (*file*)

get_output_fnames ()

get_output_hdas_and_fnames ()

get_output_sizes ()

get_parallelism ()

get_param_dict ()

Restore the dictionary of parameters from the database.

get_session_id ()

get_state ()

get_tool_provided_job_metadata ()

get_version_string_path ()

has_limits ()

job_destination

Return the JobDestination that this job will use to run. This will either be a configured destination, a randomly selected destination if the configured destination was a tag, or a dynamically generated destination from the dynamic runner.

Calling this method for the first time causes the dynamic runner to do its calculation, if any.

Returns JobDestination

mark_as_resubmitted ()

pause (*job=None, message=None*)

prepare (*compute_environment=None*)

Prepare the job to run by creating the working directory and the config files.

reclaim_ownership ()

requires_setting_metadata

set_job_destination (*job_destination, external_id=None*)

Persist job destination params in the database for recovery.

self.job_destination is not used because a runner may choose to rewrite parts of the destination (e.g. the params).

set_runner (*runner_url, external_id*)

setup_external_metadata (*exec_dir=None, tmp_dir=None, dataset_files_path=None, config_root=None, config_file=None, datatypes_config=None, set_extension=True, **kws*)

user

user_system_pwent

class `galaxy.jobs.NoopQueue`

Bases: `object`

Implements the JobQueue / JobStopQueue interface but does nothing

put (**args, **kwargs*)

put_stop (**args*)

shutdown ()

class `galaxy.jobs.ParallelismInfo` (*tag*)

Bases: `object`

Stores the information (if any) for running multiple instances of the tool in parallel on the same set of inputs.

class `galaxy.jobs.SharedComputeEnvironment` (*job_wrapper, job*)

Bases: `galaxy.jobs.SimpleComputeEnvironment`

Default ComputeEnvironment for job and task wrapper to pass to ToolEvaluator - valid when Galaxy and compute share all the relevant file systems.

input_paths ()

new_file_path ()

output_paths ()

version_path ()

working_directory ()

class `galaxy.jobs.SimpleComputeEnvironment`

Bases: `object`

config_directory ()

sep ()

unstructured_path_rewriter ()

class `galaxy.jobs.TaskWrapper` (*task, queue*)

Bases: `galaxy.jobs.JobWrapper`

Extension of JobWrapper intended for running tasks. Should be refactored into a generalized executable unit wrapper parent, then jobs and tasks.

can_split ()

change_state (*state*, *info=False*)

cleanup ()

fail (*message*, *exception=False*)

finish (*stdout*, *stderr*, *tool_exit_code=None*)

Called to indicate that the associated command has been run. Updates the output datasets based on stderr and stdout from the command, and the contents of the output files.

get_command_line ()

get_dataset_finish_context (*job_context*, *dataset*)

get_exit_code ()

get_id_tag ()

get_job ()

get_output_destination (*output_path*)

Destination for outputs marked as from_work_dir. These must be copied with the same basenme as the path for the ultimate output destination. This is required in the task case so they can be merged.

get_output_file_id (*file*)

get_param_dict ()

Restore the dictionary of parameters from the database.

get_session_id ()

get_state ()

get_task ()

get_tool_provided_job_metadata ()

prepare (*compute_environment=None*)

Prepare the job to run by creating the working directory and the config files.

set_runner (*runner_url*, *external_id*)

setup_external_metadata (*exec_dir=None*, *tmp_dir=None*, *dataset_files_path=None*, *config_root=None*, *config_file=None*, *datatypes_config=None*, *set_extension=True*, ***kws*)

galaxy.jobs.config_exception (*e*, *file*)

handler Module Galaxy job handler, prepares, runs, tracks, and finishes Galaxy jobs

class `galaxy.jobs.handler.DefaultJobDispatcher` (*app*)

Bases: `object`

put (*job_wrapper*)

recover (*job*, *job_wrapper*)

shutdown ()

stop (*job*)

Stop the given job. The input variable job may be either a Job or a Task.

url_to_destination (*url*)

This is used by the runner mapper (a.k.a. dynamic runner) and recovery methods to have runners convert URLs to destinations.

New-style runner plugin IDs must match the URL's scheme for this to work.

class `galaxy.jobs.handler.JobHandler` (*app*)

Bases: `object`

Handle the preparation, running, tracking, and finishing of jobs

shutdown ()

start ()

class `galaxy.jobs.handler.JobHandlerQueue` (*app, dispatcher*)

Bases: `object`

Job Handler's Internal Queue, this is what actually implements waiting for jobs to be runnable and dispatching to a JobRunner.

STOP_SIGNAL = <object object>

get_total_job_count_per_destination ()

get_user_job_count (*user_id*)

get_user_job_count_per_destination (*user_id*)

increase_running_job_count (*user_id, destination_id*)

job_pair_for_id (*id*)

job_wrapper (*job, use_persisted_destination=False*)

put (*job_id, tool_id*)

Add a job to the queue (by job identifier)

shutdown ()

Attempts to gracefully shut down the worker thread

start ()

Starts the JobHandler's thread after checking for any unhandled jobs.

class `galaxy.jobs.handler.JobHandlerStopQueue` (*app, dispatcher*)

Bases: `object`

A queue for jobs which need to be terminated prematurely.

STOP_SIGNAL = <object object>

monitor ()

Continually iterate the waiting jobs, stop any that are found.

monitor_step ()

Called repeatedly by *monitor* to stop jobs.

put (*job_id, error_msg=None*)

shutdown ()

Attempts to gracefully shut down the worker thread

manager Module Top-level Galaxy job manager, moves jobs to handler(s)

class `galaxy.jobs.manager.JobManager` (*app*)

Bases: `object`

Highest level interface to job management.

TODO: Currently the app accesses “job_queue” and “job_stop_queue” directly. This should be decoupled.

shutdown ()

start ()

class `galaxy.jobs.manager.NoopHandler` (**args, **kwargs*)

Bases: `object`

shutdown (**args*)

start ()

mapper Module

exception `galaxy.jobs.mapper.JobMappingException` (*failure_message*)

Bases: `exceptions.Exception`

exception `galaxy.jobs.mapper.JobNotReadyException` (*job_state=None, message=None*)

Bases: `exceptions.Exception`

class `galaxy.jobs.mapper.JobRunnerMapper` (*job_wrapper, url_to_destination, job_config*)

Bases: `object`

This class is responsible to managing the mapping of jobs (in the form of *job_wrappers*) to job runner url strings.

cache_job_destination (*raw_job_destination*)

get_job_destination (*params*)

Cache the *job_destination* to avoid recalculation.

transfer_manager Module Manage transfers from arbitrary URLs to temporary files. Socket interface for IPC with multiple process configurations.

class `galaxy.jobs.transfer_manager.Sleeper`

Bases: `object`

Provides a ‘sleep’ method that sleeps for a number of seconds *unless* the notify method is called (from a different thread).

sleep (*seconds*)

wake ()

class `galaxy.jobs.transfer_manager.TransferManager` (*app*)

Bases: `object`

Manage simple data transfers from URLs to temporary locations.

get_state (*transfer_jobs, via_socket=False*)

new (*path=None, **kwd*)

run (*transfer_jobs*)

This method blocks, so if invoking the transfer manager ever starts taking too long, we should move it to a thread. However, the *transfer_manager* will either daemonize or return after submitting to a running daemon, so it should be fairly quick to return.

```
shutdown()
```

Subpackages

actions Package

actions Package This package contains job action classes.

post Module Actions to be run at job completion. Currently only used in workflows.

```
class galaxy.jobs.actions.post.ActionBox
```

Bases: `object`

```
actions = {'ChangeDatatypeAction': <class 'galaxy.jobs.actions.post.ChangeDatatypeAction'>, 'RenameDatasetAction'
```

```
classmethod execute (app, sa_session, pja, job, replacement_dict=None)
```

```
classmethod get_add_list ()
```

```
classmethod get_forms (trans)
```

```
classmethod get_short_str (action)
```

```
classmethod handle_incoming (incoming)
```

```
immediate_actions = ['ChangeDatatypeAction', 'RenameDatasetAction']
```

```
public_actions = ['RenameDatasetAction', 'ChangeDatatypeAction', 'ColumnSetAction', 'EmailAction', 'DeleteInte
```

```
class galaxy.jobs.actions.post.ChangeDatatypeAction
```

Bases: `galaxy.jobs.actions.post.DefaultJobAction`

```
classmethod execute (app, sa_session, action, job, replacement_dict)
```

```
classmethod get_config_form (trans)
```

```
classmethod get_short_str (pja)
```

```
name = 'ChangeDatatypeAction'
```

```
verbose_name = 'Change Datatype'
```

```
class galaxy.jobs.actions.post.ColumnSetAction
```

Bases: `galaxy.jobs.actions.post.DefaultJobAction`

```
classmethod execute (app, sa_session, action, job, replacement_dict)
```

```
classmethod get_config_form (trans)
```

```
classmethod get_short_str (pja)
```

```
name = 'ColumnSetAction'
```

```
verbose_name = 'Assign Columns'
```

```
class galaxy.jobs.actions.post.DefaultJobAction
```

Bases: `object`

Base job action.

```
classmethod execute (app, sa_session, action, job, replacement_dict=None)
```

```
classmethod get_config_form (trans)
```

```

    classmethod get_short_str (pja)
    name = 'DefaultJobAction'
    verbose_name = 'Default Job'
class galaxy.jobs.actions.post.DeleteDatasetAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'DeleteDatasetAction'
    verbose_name = 'Delete Dataset'
class galaxy.jobs.actions.post.DeleteIntermediatesAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'DeleteIntermediatesAction'
    verbose_name = 'Delete Non-Output Completed Intermediate Steps'
class galaxy.jobs.actions.post.EmailAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    This action sends an email to the galaxy user responsible for a job.
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'EmailAction'
    verbose_name = 'Email Notification'
class galaxy.jobs.actions.post.HideDatasetAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'HideDatasetAction'
    verbose_name = 'Hide Dataset'
class galaxy.jobs.actions.post.RenameDatasetAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'RenameDatasetAction'

```

```
    verbose_name = 'Rename Dataset'

class galaxy.jobs.actions.post.SetMetadataAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction

    classmethod execute (app, sa_session, action, job, replacement_dict)

    classmethod get_config_form (trans)

    name = 'SetMetadataAction'

galaxy.jobs.actions.post.get_form_template (action_type, title, content, help,
                                             on_output=True)
```

deferred Package

deferred Package Queue for running deferred code via plugins.

```
class galaxy.jobs.deferred.DeferredJobQueue (app)
    Bases: object

    job_states = <galaxy.util.bunch.Bunch object>

    shutdown ()

class galaxy.jobs.deferred.FakeTrans (app, history=None, user=None)
    Bases: object

    A fake trans for calling the external set metadata tool

    db_dataset_for (dbkey)

    get_current_user_roles ()

    get_galaxy_session ()

    log_event (message, tool_id=None)

class galaxy.jobs.deferred.Sleeper
    Bases: object

    Provides a 'sleep' method that sleeps for a number of seconds unless the notify method is called (from a different thread).

    sleep (seconds)

    wake ()
```

data_transfer Module Module for managing data transfer jobs.

```
class galaxy.jobs.deferred.data_transfer.DataTransfer (app)
    Bases: object

    check_interval = 15

    check_job (job)

    create_job (trans, **kwd)

    dataset_datatype_re = <_sre.SRE_Pattern object>

    dataset_name_re = <_sre.SRE_Pattern object>

    run_job (job)
```


genome_index Module**genome_transfer Module****liftover_transfer Module**

manual_data_transfer Module Generic module for managing manual data transfer jobs using Galaxy’s built-in file browser. This module can be used by various external services that are configured to transfer data manually.

```
class galaxy.jobs.deferred.manual_data_transfer.ManualDataTransferPlugin (app)
    Bases: galaxy.jobs.deferred.data_transfer.DataTransfer

    check_job (job)

    create_job (trans, **kwd)
```

pacific_biosciences_smrt_portal Module Module for managing jobs in Pacific Bioscience’s SMRT Portal and automatically transferring files produced by SMRT Portal.

```
class galaxy.jobs.deferred.pacific_biosciences_smrt_portal.SMRTPortalPlugin (app)
    Bases: galaxy.jobs.deferred.data_transfer.DataTransfer

    api_path = '/smrtportal/api'

    check_job (job)

    create_job (trans, **kwd)
```

runners Package

runners Package Base classes for job runner plugins.

```
class galaxy.jobs.runners.AsynchronousJobRunner (app, nworkers, **kwargs)
    Bases: galaxy.jobs.runners.BaseJobRunner
```

Parent class for any job runner that runs jobs asynchronously (e.g. via a distributed resource manager). Provides general methods for having a thread to monitor the state of asynchronous jobs and submitting those jobs to the correct methods (queue, finish, cleanup) at appropriate times..

```
check_watched_item (job_state)
```

```
check_watched_items ()
```

This method is responsible for iterating over self.watched and handling state changes and updating self.watched with a new list of watched job states. Subclasses can opt to override this directly (as older job runners will initially) or just override check_watched_item and allow the list processing to reuse the logic here.

```
fail_job (job_state)
```

```
finish_job (job_state)
```

Get the output/error for a finished job, pass to *job_wrapper.finish* and cleanup all the job’s temporary files.

```
handle_stop ()
```

```
mark_as_failed (job_state)
```

```
mark_as_finished (job_state)
```

monitor()

Watches jobs currently in the monitor queue and deals with state changes (queued to running) and job completion.

monitor_job(job_state)

shutdown()

Attempts to gracefully shut down the monitor thread

```
class galaxy.jobs.runners.AsynchronousJobState(files_dir=None,      job_wrapper=None,
                                              job_id=None,   job_file=None,   out-
                                              put_file=None,      error_file=None,
                                              exit_code_file=None,  job_name=None,
                                              job_destination=None)
```

Bases: *galaxy.jobs.runners.JobState*

Encapsulate the state of an asynchronous job, this should be subclassed as needed for various job runners to capture additional information needed to communicate with distributed resource manager.

check_limits(runtime=None)

cleanup()

register_cleanup_file_attribute(attribute)

running

```
class galaxy.jobs.runners.BaseJobRunner(app, nworkers, **kwargs)
```

Bases: *object*

DEFAULT_SPECS = {'recheck_missing_job_retries': {'default': 0, 'map': <type 'int'>, 'valid': <function <lambda> at 0x...

build_command_line(job_wrapper, include_metadata=False, include_work_dir_outputs=True)

get_job_file(job_wrapper, **kws)

get_work_dir_outputs(job_wrapper, job_working_directory=None)

Returns list of pairs (source_file, destination) describing path to work_dir output file and ultimate destination.

mark_as_queued(job_wrapper)

mark_as_resubmitted(job_state)

parse_destination_params(params)

Parse the JobDestination params dict and return the runner's native representation of those params.

prepare_job(job_wrapper, include_metadata=False, include_work_dir_outputs=True)

Some sanity checks that all runners' queue_job() methods are likely to want to do

put(job_wrapper)

Add a job to the queue (by job identifier), indicate that the job is ready to run.

queue_job(job_wrapper)

recover(job, job_wrapper)

run_next()

Run the next item in the work queue (a job waiting to run)

shutdown()

Attempts to gracefully shut down the worker threads

stop_job(job)

url_to_destination (*url*)
Convert a legacy URL to a JobDestination.

Job runner URLs are deprecated, JobDestinations should be used instead. This base class method converts from a URL to a very basic JobDestination without destination params.

class `galaxy.jobs.runners.JobState`

Bases: `object`

Encapsulate state of jobs.

static default_exit_code_file (*files_dir*, *id_tag*)

static default_job_file (*files_dir*, *id_tag*)

runner_states = `<galaxy.util.bunch.Bunch object>`

set_defaults (*files_dir*)

class `galaxy.jobs.runners.RunnerParams` (*specs=None*, *params=None*)

Bases: `galaxy.util.ParamsWithSpecs`

cli Module Job control via a command line interface (e.g. qsub/qstat), possibly over a remote connection (e.g. ssh).

class `galaxy.jobs.runners.cli.ShellJobRunner` (*app*, *nworkers*)

Bases: `galaxy.jobs.runners.AsynchronousJobRunner`

Job runner backed by a finite pool of worker threads. FIFO scheduling

check_watched_items ()

Called by the monitor thread to look at each watched job and deal with state changes.

finish_job (*job_state*)

For recovery of jobs started prior to standardizing the naming of files in the AsynchronousJobState object

get_cli_plugins (*shell_params*, *job_params*)

parse_destination_params (*params*)

queue_job (*job_wrapper*)

Create job script and submit it to the DRM

recover (*job*, *job_wrapper*)

Recovers jobs stuck in the queued/running state when Galaxy started

runner_name = `'ShellRunner'`

stop_job (*job*)

Attempts to delete a dispatched job

url_to_destination (*url*)

condor Module Job control via the Condor DRM.

class `galaxy.jobs.runners.condor.CondorJobRunner` (*app*, *nworkers*)

Bases: `galaxy.jobs.runners.AsynchronousJobRunner`

Job runner backed by a finite pool of worker threads. FIFO scheduling

check_watched_items ()

Called by the monitor thread to look at each watched job and deal with state changes.

queue_job (*job_wrapper*)

Create job script and submit it to the DRM

recover (*job, job_wrapper*)

Recovers jobs stuck in the queued/running state when Galaxy started

runner_name = 'CondorRunner'

stop_job (*job*)

Attempts to delete a job from the DRM queue

drmaa Module Job control via the DRMAA API.

class `galaxy.jobs.runners.drmaa.DRMAAJobRunner` (*app, nworkers, **kwargs*)

Bases: `galaxy.jobs.runners.AsynchronousJobRunner`

Job runner backed by a finite pool of worker threads. FIFO scheduling

check_watched_items ()

Called by the monitor thread to look at each watched job and deal with state changes.

external_runjob (*jobtemplate_filename, username*)

runs an external script the will QSUB a new job. The external script will be run with sudo, and will setuid() to the specified user. Effectively, will QSUB as a different user (then the one used by Galaxy).

get_native_spec (*url*)

Get any native DRM arguments specified by the site configuration

queue_job (*job_wrapper*)

Create job script and submit it to the DRM

recover (*job, job_wrapper*)

Recovers jobs stuck in the queued/running state when Galaxy started

runner_name = 'DRMAARunner'

stop_job (*job*)

Attempts to delete a job from the DRM queue

store_jobtemplate (*job_wrapper, jt*)

Stores the content of a DRMAA JobTemplate object in a file as a JSON string. Path is hard-coded, but it's no worse than other path in this module. Uses Galaxy's JobID, so file is expected to be unique.

url_to_destination (*url*)

Convert a legacy URL to a job destination

local Module Job runner plugin for executing jobs on the local system via the command line.

class `galaxy.jobs.runners.local.LocalJobRunner` (*app, nworkers*)

Bases: `galaxy.jobs.runners.BaseJobRunner`

Job runner backed by a finite pool of worker threads. FIFO scheduling

queue_job (*job_wrapper*)

recover (*job, job_wrapper*)

runner_name = 'LocalRunner'

stop_job (*job*)

lwr Module

class `galaxy.jobs.runners.lwr.LwrJobRunner` (*app*, *nworkers*, ***kws*)

Bases: `galaxy.jobs.runners.AsynchronousJobRunner`

LWR Job Runner

check_pid (*pid*)

check_watched_item (*job_state*)

fail_job (*job_state*)

Seperated out so we can use the worker threads for it.

finish_job (*job_state*)

get_client (*job_destination_params*, *job_id*, *env=[]*)

get_client_from_state (*job_state*)

get_client_from_wrapper (*job_wrapper*)

get_input_files (*job_wrapper*)

get_output_files (*job_wrapper*)

queue_job (*job_wrapper*)

recover (*job*, *job_wrapper*)

Recovers jobs stuck in the queued/running state when Galaxy started

runner_name = 'LWRRunner'

shutdown ()

stop_job (*job*)

url_to_destination (*url*)

Convert a legacy URL to a job destination

pbs Module**tasks Module**

class `galaxy.jobs.runners.tasks.TaskedJobRunner` (*app*, *nworkers*)

Bases: `galaxy.jobs.runners.BaseJobRunner`

Job runner backed by a finite pool of worker threads. FIFO scheduling

queue_job (*job_wrapper*)

recover (*job*, *job_wrapper*)

runner_name = 'TaskRunner'

stop_job (*job*)

Subpackages**cli_job Package****cli_job Package**

torque Module

cli_shell Package

cli_shell Package

rsh Module

splitters Package

basic Module

galaxy.jobs.splitters.basic.**do_merge** (*job_wrapper, task_wrappers*)

galaxy.jobs.splitters.basic.**do_split** (*job_wrapper*)

galaxy.jobs.splitters.basic.**set_basic_defaults** (*job_wrapper*)

multi Module

galaxy.jobs.splitters.multi.**do_merge** (*job_wrapper, task_wrappers*)

galaxy.jobs.splitters.multi.**do_split** (*job_wrapper*)

model Package

model Package Galaxy data model classes

Naming: try to use class names that have a distinct plural form so that the relationship cardinalities are obvious (e.g. prefer Dataset to Data)

class galaxy.model.**APIKeys** (*id=None, user_id=None, key=None*)
Bases: object

class galaxy.model.**BaseJobMetric** (*plugin, metric_name, metric_value*)
Bases: object

exception galaxy.model.**ConverterDependencyException** (*value*)
Bases: exceptions.Exception

class galaxy.model.**DataManagerHistoryAssociation** (*id=None, history=None, user=None*)
Bases: object

class galaxy.model.**DataManagerJobAssociation** (*id=None, job=None, data_manager_id=None*)
Bases: object

class galaxy.model.**Dataset** (*id=None, state=None, external_filename=None, extra_files_path=None, file_size=None, purgable=True, uuid=None*)
Bases: object

conversion_messages = <galaxy.util.bunch.Bunch object>

engine = None

extra_files_path

file_name

file_path = '/tmp/'

```

full_delete()
    Remove the file and extra files, marks deleted and purged

get_access_roles(trans)

get_extra_files_path()

get_file_name()

get_manage_permissions_roles(trans)

get_size(nice_size=False)
    Returns the size of the data on disk

get_total_size()

has_data()
    Detects whether there is any data

has_manage_permissions_roles(trans)

is_multi_byte()

mark_deleted(include_children=True)

object_store = None

permitted_actions = <galaxy.util.bunch.Bunch object>

set_extra_files_path(extra_files_path)

set_file_name(filename)

set_size()
    Returns the size of the data on disk

set_total_size()

states = <galaxy.util.bunch.Bunch object>

user_can_purge

class galaxy.model.DatasetCollection(id=None, collection_type=None)
    Bases: object, galaxy.model.item\_attrs.Dictifiable, galaxy.model.item\_attrs.UsesAnnotations

    copy()

    dataset_instances

    dict_collection_visible_keys = ('id', 'collection_type')

    dict_element_visible_keys = ('id', 'collection_type')

    set_from_dict(new_data)

    state

    validate()

class galaxy.model.DatasetCollectionElement(id=None, collection=None, element=None, element_index=None, element_identifier=None)
    Bases: object, galaxy.model.item\_attrs.Dictifiable

    Associates a DatasetInstance (hda or ldda) with a DatasetCollection.

    copy_to_collection(collection)

    dataset

```

```
dataset_instance
dict_collection_visible_keys = ('id', 'element_type', 'element_index', 'element_identifier')
dict_element_visible_keys = ('id', 'element_type', 'element_index', 'element_identifier')
element_object
element_type
first_dataset_instance()
is_collection

class galaxy.model.DatasetCollectionInstance (collection=None, deleted=False)
    Bases: object, galaxy.model.HasName

    display_name()

    set_from_dict(new_data)
        Set object attributes to the values in dictionary new_data limiting to only those keys in
        dict_element_visible_keys.

        Returns a dictionary of the keys, values that have been changed.

    state

class galaxy.model.DatasetInstance (id=None, hid=None, name=None, info=None, blurb=None,
                                     peek=None, tool_version=None, extension=None,
                                     dbkey=None, metadata=None, history=None, dataset=None,
                                     deleted=False, designation=None, parent_id=None, vali-
                                     dation_errors=None, visible=True, create_dataset=False,
                                     sa_session=None, extended_metadata=None)
    Bases: object

    A base class for all 'dataset instances', HDAs, LDAs, etc

    add_validation_error(validation_error)

    as_display_type(type, **kwd)

    can_convert_to(format)

    change_datatype(new_ext)

    clear_associated_files(metadata_safe=False, purge=False)

    conversion_messages = <galaxy.util.bunch.Bunch object>

    convert_dataset(trans, target_type)
        Converts a dataset to the target_type and returns a message indicating status of the conversion. None is
        returned to indicate that dataset was converted successfully.

    creating_job

    datatype

    dbkey

    display_info()

    display_name()

    display_peek()

    ext

    extend_validation_errors(validation_errors)
```


extra_files_path

file_name

find_conversion_destination (*accepted_formats*, ***kwd*)
Returns (*target_ext*, existing converted dataset)

get_child_by_designation (*designation*)

get_converted_dataset (*trans*, *target_ext*)
Return converted dataset(s) if they exist, along with a dict of dependencies. If not converted yet, do so and return None (the first time). If unconvertible, raise exception.

get_converted_dataset_deps (*trans*, *target_ext*)
Returns dict of { "dependency" => HDA }

get_converted_files_by_type (*file_type*)

get_converter_types ()

get_dataset_state ()

get_datasources (*trans*)
Returns datasources for dataset; if datasources are not available due to indexing, indexing is started. Return value is a dictionary with entries of type (<datasource_type> : {<datasource_name>, <indexing_message>}).

get_dbkey ()

get_display_applications (*trans*)

get_file_name ()

get_metadata ()

get_metadata_dataset (*dataset_ext*)
Returns an HDA that points to a metadata file which contains a converted data with the requested extension.

get_mime ()
Returns the mime type of the data

get_raw_data ()
Returns the full data. To stream it open the *file_name* and read/write as needed

get_size (*nice_size=False*)
Returns the size of the data on disk

get_total_size ()

get_visualizations ()

has_data ()
Detects whether there is any data

init_meta (*copy_from=None*)

is_multi_byte ()
Data consists of multi-byte characters

is_pending
Return true if the dataset is neither ready nor in error

mark_deleted (*include_children=True*)

mark_undeleted (*include_children=True*)

```
mark_unhidden (include_children=True)

metadata

missing_meta (**kwd)

permitted_actions = <galaxy.util.bunch.Bunch object>

set_dataset_state (state)

set_dbkey (value)

set_file_name (filename)

set_meta (**kwd)

set_metadata (bunch)

set_peek (is_multi_byte=False)

set_raw_data (data)
    Saves the data on the disc

set_size ()
    Returns the size of the data on disk

set_total_size ()

source_dataset_chain

source_library_dataset

state

states = <galaxy.util.bunch.Bunch object>

undeletable ()

write_from_stream (stream)
    Writes data from a stream

class galaxy.model.DatasetPermissions (action, dataset, role)
    Bases: object

class galaxy.model.DatasetTagAssociation (id=None, user=None, item_id=None, tag_id=None,
                                          user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.DatasetToValidationErrorAssociation (dataset, validation_error)
    Bases: object

class galaxy.model.DefaultHistoryPermissions (history, action, role)
    Bases: object

class galaxy.model.DefaultQuotaAssociation (type, quota)
    Bases: galaxy.model.Quota, galaxy.model.item_attrs.Dictifiable

    dict_element_visible_keys = ('type',)

    types = <galaxy.util.bunch.Bunch object>

class galaxy.model.DefaultUserPermissions (user, action, role)
    Bases: object

class galaxy.model.DeferredJob (state=None, plugin=None, params=None)
    Bases: object

    check_interval
```

```

    get_check_interval()
    get_last_check()
    is_check_time
    last_check
    set_check_interval(seconds)
    set_last_check(seconds)
    states = <galaxy.util.bunch.Bunch object>

class galaxy.model.Event(message=None, history=None, user=None, galaxy_session=None)
    Bases: object

class galaxy.model.ExtendedMetadata(data)
    Bases: object

class galaxy.model.ExtendedMetadataIndex(extended_metadata, path, value)
    Bases: object

class galaxy.model.ExternalService(name=None, description=None, external_service_type_id=None, version=None, form_definition_id=None, form_values_id=None, deleted=None)
    Bases: object

    data_transfer_protocol = <galaxy.util.bunch.Bunch object>
    get_external_service_type(trans)
    load_data_transfer_settings(trans)
    populate_actions(trans, item, param_dict=None)

class galaxy.model.FormDefinition(name=None, desc=None, fields=[], form_definition_current=None, form_type=None, layout=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable

    dict_collection_visible_keys = ('id', 'name')
    dict_element_visible_keys = ('id', 'name', 'desc', 'form_definition_current_id', 'fields', 'layout')
    field_as_html(field)
        Generates disabled html for a field
    get_widgets(user, contents={}, **kwd)
        Return the list of widgets that comprise a form definition, including field contents if any.
    grid_fields(grid_index)
    supported_field_types = [<class 'galaxy.web.form_builder.AddressField'>, <class 'galaxy.web.form_builder.CheckField'>]
    types = <galaxy.util.bunch.Bunch object>

class galaxy.model.FormDefinitionCurrent(form_definition=None)
    Bases: object

class galaxy.model.FormValues(form_def=None, content=None)
    Bases: object

```

```
class galaxy.model.GalaxySession (id=None, user=None, remote_host=None, remote_addr=None,
                                referer=None, current_history=None, session_key=None,
                                is_valid=False, prev_session_id=None)
```

Bases: object

add_history (*history*, *association=None*)

get_disk_usage ()

set_disk_usage (*bytes*)

total_disk_usage

```
class galaxy.model.GalaxySessionToHistoryAssociation (galaxy_session, history)
```

Bases: object

```
class galaxy.model.GenomeIndexToolData (job=None, params=None, dataset=None,
                                         deferred_job=None, transfer_job=None,
                                         fasta_path=None, created_time=None, modified_time=None,
                                         dbkey=None, user=None, indexer=None)
```

Bases: object

```
class galaxy.model.Group (name=None)
```

Bases: object, *galaxy.model.item_attrs.Dictifiable*

dict_collection_visible_keys = ('id', 'name')

dict_element_visible_keys = ('id', 'name')

```
class galaxy.model.GroupQuotaAssociation (group, quota)
```

Bases: object, *galaxy.model.item_attrs.Dictifiable*

dict_element_visible_keys = ('group',)

```
class galaxy.model.GroupRoleAssociation (group, role)
```

Bases: object

```
class galaxy.model.HasJobMetrics
```

add_metric (*plugin*, *metric_name*, *metric_value*)

metrics

```
class galaxy.model.HasName
```

get_display_name ()

These objects have a name attribute can be either a string or a unicode object. If string, convert to unicode object assuming 'utf-8' format.

```
class galaxy.model.History (id=None, name=None, user=None)
```

Bases: object, *galaxy.model.item_attrs.Dictifiable*, *galaxy.model.item_attrs.UsesAnnotations*, *galaxy.model.HasName*

activatable_datasets

active_contents

Return all active contents ordered by hid.

active_datasets_children_and_roles

add_dataset (*dataset*, *parent_id=None*, *genome_build=None*, *set_hid=True*, *quota=True*)

add_dataset_collection (*history_dataset_collection*, *set_hid=True*)

```

add_galaxy_session (galaxy_session, association=None)

contents_iter (**kws)
    Fetch filtered list of contents of history.

copy (name=None, target_user=None, activatable=False, all_datasets=False)
    Return a copy of this history using the given name and target_user. If activatable, copy only non-deleted
    datasets. If all_datasets, copy non-deleted, deleted, and purged datasets.

copy_tags_from (target_user, source_history)

default_name = 'Unnamed history'

dict_collection_visible_keys = ('id', 'name', 'published', 'deleted')

dict_element_visible_keys = ('id', 'name', 'genome_build', 'deleted', 'purged', 'update_time', 'published', 'imp

empty

get_disk_size (nice_size=False)

get_disk_size_bytes

latest_export

resume_paused_jobs ()

set_from_dict (new_data)
    Set object attributes to the values in dictionary new_data limiting to only those keys in
    dict_element_visible_keys.

    Returns a dictionary of the keys, values that have been changed.

to_dict (view='collection', value_mapper=None)

unhide_datasets ()

class galaxy.model.HistoryAnnotationAssociation
    Bases: object

class galaxy.model.HistoryDatasetAssociation (hid=None, history=None,
                                              copied_from_history_dataset_association=None,
                                              copied_from_library_dataset_association=None,
                                              sa_session=None, **kwd)
    Bases: galaxy.model.DatasetInstance, galaxy.model.item_attrs.Dictifiable,
           galaxy.model.item_attrs.UsesAnnotations, galaxy.model.HasName

    Resource class that creates a relation between a dataset and a user history.

clear_associated_files (metadata_safe=False, purge=False)

copy (copy_children=False, parent_id=None)
    Create a copy of this HDA.

get_access_roles (trans)
    Return The access roles associated with this HDA's dataset.

history_content_type

quota_amount (user)
    Return the disk space used for this HDA relevant to user quotas.

    If the user has multiple instances of this dataset, it will not affect their disk usage statistic.

set_from_dict (new_data)
    Set object attributes to the values in dictionary new_data limiting to only the following keys: name,
    deleted, visible, genome_build, info, and blurb.

```

Returns a dictionary of the keys, values that have been changed.

to_dict (*view='collection', expose_dataset_path=False*)

Return attributes of this HDA that are exposed using the API.

to_library_dataset_dataset_association (*trans, target_folder, replace_dataset=None, parent_id=None, user=None, roles=None, ldda_message=''*)

Copy this HDA to a library optionally replacing an existing LDDA.

class `galaxy.model.HistoryDatasetAssociationAnnotationAssociation`

Bases: `object`

class `galaxy.model.HistoryDatasetAssociationDisplayAtAuthorization` (*hda=None, user=None, site=None*)

Bases: `object`

class `galaxy.model.HistoryDatasetAssociationRatingAssociation` (*id=None, user=None, item=None, rating=0*)

Bases: `galaxy.model.ItemRatingAssociation`

set_item (*history_dataset_association*)

class `galaxy.model.HistoryDatasetAssociationSubset` (*hda, subset, location*)

Bases: `object`

class `galaxy.model.HistoryDatasetAssociationTagAssociation` (*id=None, user=None, item_id=None, tag_id=None, user_tname=None, value=None*)

Bases: `galaxy.model.ItemTagAssociation`

class `galaxy.model.HistoryDatasetCollectionAnnotationAssociation`

Bases: `object`

class `galaxy.model.HistoryDatasetCollectionAssociation` (*id=None, hid=None, collection=None, history=None, name=None, deleted=False, visible=True, copied_from_history_dataset_collection_association=None, implicit_output_name=None, implicit_input_collections=[]*)

Bases: `galaxy.model.DatasetCollectionInstance, galaxy.model.item_attrs.Dictifiable`

Associates a DatasetCollection with a History.

add_implicit_input_collection (*name, history_dataset_collection*)

copy ()

Create a copy of this history dataset collection association. Copy underlying collection.

editable_keys = ('name', 'deleted', 'visible')

find_implicit_input_collection (*name*)

history_content_type

to_dict (*view='collection'*)

```

class galaxy.model.HistoryDatasetCollectionRatingAssociation (id=None, user=None,
                                                             item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation
    set_item (dataset_collection)

class galaxy.model.HistoryDatasetCollectionTagAssociation (id=None, user=None,
                                                           item_id=None,
                                                           tag_id=None,
                                                           user_tname=None,
                                                           value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.HistoryRatingAssociation (id=None, user=None, item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation
    set_item (history)

class galaxy.model.HistoryTagAssociation (id=None, user=None, item_id=None, tag_id=None,
                                          user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.HistoryUserShareAssociation
    Bases: object

class galaxy.model.ImplicitlyConvertedDatasetAssociation (id=None, parent=None, dataset=None,
                                                           file_type=None,
                                                           deleted=False,
                                                           purged=False, meta-
                                                           data_safe=True)
    Bases: object
    clear (purge=False, delete_dataset=True)

class galaxy.model.ImplicitlyCreatedDatasetCollectionInput (name, input_dataset_collection)
    Bases: object

class galaxy.model.ItemRatingAssociation (id=None, user=None, item=None, rating=0)
    Bases: object
    set_item (item)
        Set association's item.

class galaxy.model.ItemTagAssociation (id=None, user=None, item_id=None, tag_id=None,
                                       user_tname=None, value=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    copy ()
    dict_collection_visible_keys = ('id', 'user_tname', 'user_value')
    dict_element_visible_keys = ('id', 'user_tname', 'user_value')

class galaxy.model.Job
    Bases: object, galaxy.model.HasJobMetrics, galaxy.model.item_attrs.Dictifiable
    add_input_dataset (name, dataset)
    add_input_dataset_collection (name, dataset)
    add_input_library_dataset (name, dataset)
    add_output_dataset (name, dataset)

```

add_output_dataset_collection (*name, dataset*)

add_output_library_dataset (*name, dataset*)

add_parameter (*name, value*)

add_post_job_action (*pja*)

check_if_output_datasets_deleted ()

Return true if all of the output datasets associated with this job are in the deleted state

dict_collection_visible_keys = ['id', 'state', 'exit_code', 'update_time', 'create_time']

dict_element_visible_keys = ['id', 'state', 'exit_code', 'update_time', 'create_time']

A job represents a request to run a tool given input datasets, tool parameters, and output datasets.

finished

get_command_line ()

get_external_output_metadata ()

The external_output_metadata is currently a reference from Job to JobExternalOutputMetadata. It exists for a job but not a task.

get_handler ()

get_id ()

get_id_tag ()

Return a tag that can be useful in identifying a Job. This returns the Job's get_id

get_imported ()

get_info ()

get_input_datasets ()

get_input_library_datasets ()

get_job ()

get_job_runner_external_id ()

get_job_runner_name ()

get_output_datasets ()

get_output_library_datasets ()

get_param_filename ()

get_param_values (*app, ignore_errors=False*)

Read encoded parameter values from the database and turn back into a dict of tool parameter values.

get_parameters ()

get_params ()

get_post_job_actions ()

get_session_id ()

get_state ()

get_tasks ()

get_tool_id ()

get_tool_version ()


```

get_user()
get_user_id()
mark_deleted(track_jobs_in_database=False)
    Mark this job as deleted, and mark any output datasets as discarded.
raw_param_dict()
set_command_line(command_line)
set_final_state(final_state)
set_handler(handler)
set_imported(imported)
set_info(info)
set_input_datasets(input_datasets)
set_input_library_datasets(input_library_datasets)
set_output_datasets(output_datasets)
set_output_library_datasets(output_library_datasets)
set_param_filename(param_filename)
set_parameters(parameters)
set_params(params)
set_post_job_actions(post_job_actions)
set_runner_external_id(job_runner_external_id)
set_runner_name(job_runner_name)
set_session_id(session_id)
set_state(state)
    Save state history
set_tool_id(tool_id)
set_tool_version(tool_version)
set_user_id(user_id)
states = <galaxy.util.bunch.Bunch object>
to_dict(view='collection')

```

class galaxy.model.JobExportHistoryArchive(*job=None, history=None, dataset=None, compressed=False, history_attrs_filename=None, datasets_attrs_filename=None, jobs_attrs_filename=None*)

Bases: object

```

export_name
preparing
ready
up_to_date
    Return False, if a new export should be generated for corresponding history.

```

```
class galaxy.model.JobExternalOutputMetadata (job=None, dataset=None)
    Bases: object

    dataset

class galaxy.model.JobImportHistoryArchive (job=None, history=None, archive_dir=None)
    Bases: object

class galaxy.model.JobMetricNumeric (plugin, metric_name, metric_value)
    Bases: galaxy.model.BaseJobMetric

class galaxy.model.JobMetricText (plugin, metric_name, metric_value)
    Bases: galaxy.model.BaseJobMetric

class galaxy.model.JobParameter (name, value)
    Bases: object

class galaxy.model.JobStateHistory (job)
    Bases: object

class galaxy.model.JobToInputDatasetAssociation (name, dataset)
    Bases: object

class galaxy.model.JobToInputDatasetCollectionAssociation (name, dataset)
    Bases: object

class galaxy.model.JobToInputLibraryDatasetAssociation (name, dataset)
    Bases: object

class galaxy.model.JobToOutputDatasetAssociation (name, dataset)
    Bases: object

class galaxy.model.JobToOutputDatasetCollectionAssociation (name,
                                                             dataset_collection)
    Bases: object

class galaxy.model.JobToOutputLibraryDatasetAssociation (name, dataset)
    Bases: object

class galaxy.model.Library (name=None, description=None, synopsis=None, root_folder=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable, galaxy.model.HasName

    dict_collection_visible_keys = ('id', 'name')

    dict_element_visible_keys = ('id', 'deleted', 'name', 'description', 'synopsis', 'root_folder_id')

    get_access_roles (trans)

    get_active_folders (folder, folders=None)

    get_info_association (restrict=False, inherited=False)

    get_template_widgets (trans, get_contents=True)

    permitted_actions = <galaxy.util.bunch.Bunch object>

    to_dict (view='collection', value_mapper=None)
        We prepend an F to folders.

class galaxy.model.LibraryDataset (folder=None, order_id=None, name=None, info=None, li-
                                   brary_dataset_dataset_association=None, **kwd)
    Bases: object

    display_name ()

    get_info ()
```

```

    get_name ()
    info
    name
    set_info (info)
    set_library_dataset_dataset_association (ldda)
    set_name (name)
    to_dict (view='collection')
    upload_options = [('upload_file', 'Upload files'), ('upload_directory', 'Upload directory of files'), ('upload_paths', 'U

class galaxy.model.LibraryDatasetCollectionAnnotationAssociation
    Bases: object

class galaxy.model.LibraryDatasetCollectionAssociation (id=None,      collection=None,
                                                         name=None,      deleted=False,
                                                         folder=None)
    Bases: galaxy.model.DatasetCollectionInstance, galaxy.model.item_attrs.Dictifiable
    Associates a DatasetCollection with a library folder.
    editable_keys = ('name', 'deleted')
    to_dict (view='collection')

class galaxy.model.LibraryDatasetCollectionRatingAssociation (id=None, user=None,
                                                              item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation
    set_item (dataset_collection)

class galaxy.model.LibraryDatasetCollectionTagAssociation (id=None,      user=None,
                                                           item_id=None,
                                                           tag_id=None,
                                                           user_tname=None,
                                                           value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.LibraryDatasetDatasetAssociation (copied_from_history_dataset_association=None,
                                                       copied_from_library_dataset_dataset_association=None,
                                                       library_dataset=None, user=None,
                                                       sa_session=None, **kwd)
    Bases: galaxy.model.DatasetInstance, galaxy.model.HasName
    clear_associated_files (metadata_safe=False, purge=False)
    copy (copy_children=False, parent_id=None, target_folder=None)
    get_access_roles (trans)
    get_info_association (restrict=False, inherited=False)
    get_manage_permissions_roles (trans)
    get_template_widgets (trans, get_contents=True)
    has_manage_permissions_roles (trans)
    templates_dict (use_name=False)
        Returns a dict of template info
    templates_json (use_name=False)

```

```
    to_dict (view='collection')
    to_history_dataset_association (target_history, parent_id=None, add_to_history=False)
class galaxy.model.LibraryDatasetDatasetAssociationPermissions (action, library_item, role)
    Bases: object
class galaxy.model.LibraryDatasetDatasetInfoAssociation (library_dataset_dataset_association, form_definition, info)
    Bases: object
    inheritable
class galaxy.model.LibraryDatasetPermissions (action, library_item, role)
    Bases: object
class galaxy.model.LibraryFolder (name=None, description=None, item_count=0, order_id=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable, galaxy.model.HasName
    activatable_library_datasets
    add_folder (folder)
    add_library_dataset (library_dataset, genome_build=None)
    dict_element_visible_keys = ('id', 'parent_id', 'name', 'description', 'item_count', 'genome_build', 'update_time')
    get_info_association (restrict=False, inherited=False)
    get_template_widgets (trans, get_contents=True)
    library_path
    parent_library
    to_dict (view='collection', value_mapper=None)
class galaxy.model.LibraryFolderInfoAssociation (folder, form_definition, info, inheritable=False)
    Bases: object
class galaxy.model.LibraryFolderPermissions (action, library_item, role)
    Bases: object
class galaxy.model.LibraryInfoAssociation (library, form_definition, info, inheritable=False)
    Bases: object
class galaxy.model.LibraryPermissions (action, library_item, role)
    Bases: object
class galaxy.model.MetadataFile (dataset=None, name=None)
    Bases: object
    file_name
exception galaxy.model.NoConverterException (value)
    Bases: exceptions.Exception
class galaxy.model.Page
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_element_visible_keys = ['id', 'title', 'latest_revision_id', 'slug', 'published', 'importable', 'deleted']
    to_dict (view='element')
```

```

class galaxy.model.PageAnnotationAssociation
    Bases: object

class galaxy.model.PageRatingAssociation (id=None, user=None, item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation

    set_item (page)

class galaxy.model.PageRevision
    Bases: object, galaxy.model.item_attrs.Dictifiable

    dict_element_visible_keys = ['id', 'page_id', 'title', 'content']

    to_dict (view='element')

class galaxy.model.PageTagAssociation (id=None, user=None, item_id=None, tag_id=None,
                                         user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.PageUserShareAssociation
    Bases: object

class galaxy.model.PostJobAction (action_type, workflow_step, output_name=None, ac-
                                tion_arguments=None)
    Bases: object

class galaxy.model.PostJobActionAssociation (pja, job)
    Bases: object

class galaxy.model.Quota (name='', description='', amount=0, operation='')
    Bases: object, galaxy.model.item_attrs.Dictifiable

    amount

    dict_collection_visible_keys = ('id', 'name')

    dict_element_visible_keys = ('id', 'name', 'description', 'bytes', 'operation', 'display_amount', 'default', 'users')

    display_amount

    get_amount ()

    set_amount (amount)

    valid_operations = ('+', '-', '=')

class galaxy.model.Request (name=None, desc=None, request_type=None, user=None,
                             form_values=None, notification=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable

    dict_collection_visible_keys = ('id', 'name', 'state')

    get_sample (sample_name)

    is_complete

    is_new

    is_rejected

    is_submitted

    is_unsubmitted

    last_comment

    latest_event

```

```
samples_have_common_state
    Returns the state of this request's samples when they are all in one common state. Otherwise returns False.

samples_with_bar_code

samples_without_library_destinations

send_email_notification (trans, common_state, final_state=False)

state

states = <galaxy.util.bunch.Bunch object>

class galaxy.model.RequestEvent (request=None, request_state=None, comment='')
    Bases: object

class galaxy.model.RequestType (name=None, desc=None, request_form=None, sample_form=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable

    add_external_service_association (trans, external_service)

    delete_external_service_associations (trans)
        Deletes all external service associations.

    dict_collection_visible_keys = ('id', 'name', 'desc')

    dict_element_visible_keys = ('id', 'name', 'desc', 'request_form_id', 'sample_form_id')

    external_services

    final_sample_state

    get_external_service (external_service_type_id)

    get_external_services_for_manual_data_transfer (trans)
        Returns all external services that use manual data transfer

    get_template_widgets (trans, get_contents=True)

    permitted_actions = <galaxy.util.bunch.Bunch object>

    rename_dataset_options = <galaxy.util.bunch.Bunch object>

    run_details

class galaxy.model.RequestTypeExternalServiceAssociation (request_type, external_service)
    Bases: object

class galaxy.model.RequestTypePermissions (action, request_type, role)
    Bases: object

class galaxy.model.RequestTypeRunAssociation (request_type, run)
    Bases: object

class galaxy.model.Role (name='', description='', type='system', deleted=False)
    Bases: object, galaxy.model.item_attrs.Dictifiable

    dict_collection_visible_keys = ('id', 'name')

    dict_element_visible_keys = ('id', 'name', 'description', 'type')

    private_id = None

    types = <galaxy.util.bunch.Bunch object>
```

```

class galaxy.model.Run(form_definition, form_values, subindex=None)
    Bases: object

class galaxy.model.Sample(name=None, desc=None, request=None, form_values=None,
                           bar_code=None, library=None, folder=None, workflow=None, history=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    adding_to_library_dataset_files
    bulk_operations = <galaxy.util.bunch.Bunch object>
    dict_collection_visible_keys = ('id', 'name')
    get_template_widgets(trans, get_contents=True)
    get_untransferred_dataset_size(filepath, scp_configs)
    inprogress_dataset_files
    latest_event
    populate_external_services(param_dict=None, trans=None)
    queued_dataset_files
    run_details
    state
    supported_field_types = [<class 'galaxy.web.form_builder.CheckboxField'>, <class 'galaxy.web.form_builder.Select'>]
    transfer_error_dataset_files
    transferred_dataset_files
    transferring_dataset_files
    untransferred_dataset_files

class galaxy.model.SampleDataset(sample=None, name=None, file_path=None, status=None, error_msg=None, size=None, external_service=None)
    Bases: object
    transfer_status = <galaxy.util.bunch.Bunch object>

class galaxy.model.SampleEvent(sample=None, sample_state=None, comment='')
    Bases: object

class galaxy.model.SampleRunAssociation(sample, run)
    Bases: object

class galaxy.model.SampleState(name=None, desc=None, request_type=None)
    Bases: object

class galaxy.model.StoredWorkflow
    Bases: object, galaxy.model.item_attrs.Dictifiable
    copy_tags_from(target_user, source_workflow)
    dict_collection_visible_keys = ('id', 'name', 'published', 'deleted')
    dict_element_visible_keys = ('id', 'name', 'published', 'deleted')
    to_dict(view='collection', value_mapper=None)

class galaxy.model.StoredWorkflowAnnotationAssociation
    Bases: object

```

```
class galaxy.model.StoredWorkflowMenuEntry
```

```
    Bases: object
```

```
class galaxy.model.StoredWorkflowRatingAssociation (id=None, user=None, item=None,
                                                    rating=0)
```

```
    Bases: galaxy.model.ItemRatingAssociation
```

```
    set_item (stored_workflow)
```

```
class galaxy.model.StoredWorkflowTagAssociation (id=None, user=None, item_id=None,
                                                  tag_id=None, user_tname=None,
                                                  value=None)
```

```
    Bases: galaxy.model.ItemTagAssociation
```

```
class galaxy.model.StoredWorkflowUserShareAssociation
```

```
    Bases: object
```

```
class galaxy.model.Tag (id=None, type=None, parent_id=None, name=None)
```

```
    Bases: object
```

```
class galaxy.model.Task (job, working_directory, prepare_files_cmd)
```

```
    Bases: object, galaxy.model.HasJobMetrics
```

A task represents a single component of a job.

```
    get_command_line ()
```

```
    get_external_output_metadata ()
```

The external_output_metadata is currently a backref to JobExternalOutputMetadata. It exists for a job but not a task, and when a task is cancelled its corresponding parent Job will be cancelled. So None is returned now, but that could be changed to self.get_job().get_external_output_metadata().

```
    get_id ()
```

```
    get_id_tag ()
```

Return an id tag suitable for identifying the task. This combines the task's job id and the task's own id.

```
    get_info ()
```

```
    get_job ()
```

```
    get_job_runner_external_id ()
```

Runners will use the same methods to get information about the Task class as they will about the Job class, so this method just returns the task's external id.

```
    get_job_runner_name ()
```

Since runners currently access Tasks the same way they access Jobs, this method just refers to *this* instance's runner.

```
    get_param_values (app)
```

Read encoded parameter values from the database and turn back into a dict of tool parameter values.

```
    get_parameters ()
```

```
    get_prepare_input_files_cmd ()
```

```
    get_session_id ()
```

```
    get_state ()
```

```
    get_stderr ()
```

```
    get_stdout ()
```

```
    get_task_runner_external_id ()
```



```

    get_task_runner_name ()
    get_working_directory ()
    set_command_line (command_line)
    set_id (id)
    set_info (info)
    set_job (job)
    set_job_runner_external_id (task_runner_external_id)
    set_parameters (parameters)
    set_prepare_input_files_cmd (prepare_input_files_cmd)
    set_state (state)
    set_stderr (stderr)
    set_stdout (stdout)
    set_task_runner_external_id (task_runner_external_id)
    set_task_runner_name (task_runner_name)
    set_working_directory (working_directory)
    states = <galaxy.util.bunch.Bunch object>
class galaxy.model.TaskMetricNumeric (plugin, metric_name, metric_value)
    Bases: galaxy.model.BaseJobMetric
class galaxy.model.TaskMetricText (plugin, metric_name, metric_value)
    Bases: galaxy.model.BaseJobMetric
class galaxy.model.ToolTagAssociation (id=None, user=None, tool_id=None, tag_id=None,
                                       user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation
class galaxy.model.TransferJob (state=None, path=None, info=None, pid=None, socket=None,
                               params=None)
    Bases: object
    states = <galaxy.util.bunch.Bunch object>
    terminal_states = ['error', 'done']
class galaxy.model.UCI
    Bases: object
class galaxy.model.User (email=None, password=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    all_roles ()
        Return a unique list of Roles associated with this user or any of their groups.
    calculate_disk_usage ()
        Return byte count total of disk space used by all non-purged, non-library HDAs in non-purged histories.
    check_password (cleartext)
        Check if cleartext matches user password when hashed.
    dict_collection_visible_keys = ('id', 'email')
    dict_element_visible_keys = ('id', 'email', 'username', 'total_disk_usage', 'nice_total_disk_usage')

```

static expand_user_properties (*user*, *in_string*)

get_disk_usage (*nice_size=False*)

Return byte count of disk space used by user or a human-readable string if *nice_size* is *True*.

nice_total_disk_usage

Return byte count of disk space used in a human-readable string.

set_disk_usage (*bytes*)

Manually set the disk space used by a user to *bytes*.

set_password_cleartext (*cleartext*)

Set user password to the digest of *cleartext*.

total_disk_usage

Return byte count of disk space used by user or a human-readable string if *nice_size* is *True*.

use_pbkdf2 = True

Data for a Galaxy user or admin and relations to their histories, credentials, and roles.

static user_template_environment (*user*)

```
>>> env = User.user_template_environment(None)
>>> env['__user_email__']
'Anonymous'
>>> env['__user_id__']
'Anonymous'
>>> user = User('foo@example.com')
>>> user.id = 6
>>> user.username = 'foo2'
>>> env = User.user_template_environment(user)
>>> env['__user_id__']
'6'
>>> env['__user_name__']
'foo2'
```

class galaxy.model.**UserAction** (*id=None, create_time=None, user_id=None, session_id=None, action=None, params=None, context=None*)

Bases: object

class galaxy.model.**UserAddress** (*user=None, desc=None, name=None, institution=None, address=None, city=None, state=None, postal_code=None, country=None, phone=None*)

Bases: object

get_html ()

class galaxy.model.**UserGroupAssociation** (*user, group*)

Bases: object

class galaxy.model.**UserOpenID** (*user=None, session=None, openid=None*)

Bases: object

class galaxy.model.**UserPreference** (*name=None, value=None*)

Bases: object

class galaxy.model.**UserQuotaAssociation** (*user, quota*)

Bases: object, *galaxy.model.item_attrs.Dictifiable*

dict_element_visible_keys = ('user',)

class galaxy.model.**UserRoleAssociation** (*user, role*)

Bases: object

```

class galaxy.model.ValidationError (message=None, err_type=None, attributes=None)
    Bases: object

class galaxy.model.Visualization (id=None, user=None, type=None, title=None, dbkey=None,
                                   slug=None, latest_revision=None)
    Bases: object
    copy (user=None, title=None)
        Provide copy of visualization with only its latest revision.

class galaxy.model.VisualizationAnnotationAssociation
    Bases: object

class galaxy.model.VisualizationRatingAssociation (id=None, user=None, item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation
    set_item (visualization)

class galaxy.model.VisualizationRevision (visualization=None, title=None, dbkey=None, config=None)
    Bases: object
    copy (visualization=None)
        Returns a copy of this object.

class galaxy.model.VisualizationTagAssociation (id=None, user=None, item_id=None, tag_id=None, user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.VisualizationUserShareAssociation
    Bases: object

class galaxy.model.WorkRequestTagAssociation (id=None, user=None, workflow_request_id=None, tag_id=None, user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.Workflow (uuid=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_collection_visible_keys = ('name', 'has_cycles', 'has_errors')
    dict_element_visible_keys = ('name', 'has_cycles', 'has_errors')
    has_outputs_defined ()
        Returns true or false indicating whether or not a workflow has outputs defined.
    to_dict (view='collection', value_mapper=None)

class galaxy.model.WorkflowInvocation
    Bases: object, galaxy.model.item_attrs.Dictifiable
    active
        Indicates the workflow invocation is somehow active - and in particular valid actions may be performed on its "WorkflowInvocationStep"s.
    add_input (content, step_id)
    cancel ()
    dict_collection_visible_keys = ('id', 'update_time', 'workflow_id', 'history_id', 'uuid', 'state')
    dict_element_visible_keys = ('id', 'update_time', 'workflow_id', 'history_id', 'uuid', 'state')

```

```
fail()
has_input_for_step(step_id)
static poll_active_workflow_ids(sa_session, scheduler=None, handler=None)
states = <galaxy.util.bunch.Bunch object>
step_invocations_by_step_id()
step_states_by_step_id()
to_dict(view='collection', value_mapper=None)
update()

class galaxy.model.WorkflowInvocationStep
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_collection_visible_keys = ('id', 'update_time', 'job_id', 'workflow_step_id', 'action')
    dict_element_visible_keys = ('id', 'update_time', 'job_id', 'workflow_step_id', 'action')
    to_dict(view='collection', value_mapper=None)
    update()

class galaxy.model.WorkflowOutput(workflow_step, output_name)
    Bases: object

class galaxy.model.WorkflowRequest
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_collection_visible_keys = ['id', 'name', 'type', 'state', 'history_id', 'workflow_id']
    dict_element_visible_keys = ['id', 'name', 'type', 'state', 'history_id', 'workflow_id']
    to_dict(view='collection', value_mapper=None)

class galaxy.model.WorkflowRequestInputParameter(name=None, value=None, type=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Workflow-related parameters not tied to steps or inputs.
    dict_collection_visible_keys = ['id', 'name', 'value', 'type']
    types = <galaxy.util.bunch.Bunch object>

class galaxy.model.WorkflowRequestStepState(workflow_step=None, name=None, value=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Workflow step value parameters.
    dict_collection_visible_keys = ['id', 'name', 'value', 'workflow_step_id']

class galaxy.model.WorkflowRequestToInputDatasetAssociation
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Workflow step input dataset parameters.
    dict_collection_visible_keys = ['id', 'workflow_invocation_id', 'workflow_step_id', 'dataset_id', 'name']

class galaxy.model.WorkflowRequestToInputDatasetCollectionAssociation
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Workflow step input dataset collection parameters.
    dict_collection_visible_keys = ['id', 'workflow_invocation_id', 'workflow_step_id', 'dataset_collection_id', 'na
```

```

class galaxy.model.WorkflowStep
    Bases: object

class galaxy.model.WorkflowStepAnnotationAssociation
    Bases: object

class galaxy.model.WorkflowStepConnection
    Bases: object

class galaxy.model.WorkflowStepTagAssociation (id=None, user=None, item_id=None,
                                              tag_id=None, user_tname=None,
                                              value=None)
    Bases: galaxy.model.ItemTagAssociation

galaxy.model.set_datatypes_registry (d_registry)
    Set up datatypes_registry

custom_types Module
class galaxy.model.custom_types.JSONType (*args, **kwargs)
    Bases: sqlalchemy.types.TypeDecorator

    Defines a JSONType for SQLAlchemy. Takes a primitive as input and JSONifies it. This should replace Pickle-
    Type throughout Galaxy.

    compare_values (x, y)

    copy_value (value)

    impl
        alias of LargeBinary

    is_mutable ()

    load_dialect_impl (dialect)

    process_bind_param (value, dialect)

    process_result_value (value, dialect)
class galaxy.model.custom_types.MetadataType (*args, **kwargs)
    Bases: galaxy.model.custom_types.JSONType

    Backward compatible metadata type. Can read pickles or JSON, but always writes in JSON.

    process_result_value (value, dialect)

class galaxy.model.custom_types.TrimmedString (*args, **kwargs)
    Bases: sqlalchemy.types.TypeDecorator

    impl
        alias of String

    process_bind_param (value, dialect)
        Automatically truncate string values

class galaxy.model.custom_types.UUIDType (*args, **kwargs)
    Bases: sqlalchemy.types.TypeDecorator

    Platform-independent UUID type.

    Based on http://docs.sqlalchemy.org/en/rel\_0\_8/core/types.html#backend-agnostic-guid-type Changed to re-
    move sqlalchemy 0.8 specific code

    CHAR(32), storing as stringified hex values.

```

```
impl
    alias of CHAR

load_dialect_impl (dialect)

process_bind_param (value, dialect)

process_result_value (value, dialect)
```

item_attrs Module

class galaxy.model.item_attrs.Dictifiable
Mixin that enables objects to be converted to dictionaries. This is useful when for sharing objects across boundaries, such as the API, tool scripts, and JavaScript code.

to_dict (view='collection', value_mapper=None)
Return item dictionary.

exception galaxy.model.item_attrs.RuntimeException
Bases: exceptions.Exception

class galaxy.model.item_attrs.UsesAnnotations
Mixin for getting and setting item annotations.

add_item_annotation (db_session, user, item, annotation)
Add or update an item's annotation; a user can only have a single annotation for an item.

copy_item_annotation (db_session, source_user, source_item, target_user, target_item)
Copy an annotation from a user/item source to a user/item target.

delete_item_annotation (db_session, user, item)

get_item_annotation_obj (db_session, user, item)
Returns a user's annotation object for an item.

get_item_annotation_str (db_session, user, item)
Returns a user's annotation string for an item.

class galaxy.model.item_attrs.UsesItemRatings
Mixin for getting and setting item ratings.

Class makes two assumptions: (1) item-rating association table is named <item_class>RatingAssociation (2) item-rating association table has a column with a foreign key referencing item table that contains the item's id.

get_ave_item_rating_data (db_session, item, webapp_model=None)
Returns the average rating for an item.

get_user_item_rating (db_session, user, item, webapp_model=None)
Returns user's rating for an item. Return type is <item_class>RatingAssociation.

rate_item (db_session, user, item, rating, webapp_model=None)
Rate an item. Return type is <item_class>RatingAssociation.

mapping Module Details of how the data model objects are mapped onto the relational database are encapsulated here.

galaxy.model.mapping.annotation_mapping (annotation_class, **kwds)

galaxy.model.mapping.db_next_hid (self)

Override __next_hid to generate from the database in a concurrency safe way. Loads the next history ID from the DB and returns it. It also saves the future next_id into the DB.

Return type int

Returns the next history id

```
galaxy.model.mapping.init (file_path, url, engine_options={}, create_tables=False,
                           map_install_models=False, database_query_profiling_proxy=False,
                           object_store=None, trace_logger=None, use_pbkdf2=True)
```

Connect mappings to the database

```
galaxy.model.mapping.now ()
```

Return a new datetime representing UTC day and time.

```
galaxy.model.mapping.rating_mapping (rating_class, **kws)
```

```
galaxy.model.mapping.simple_mapping (model, **kws)
```

```
galaxy.model.mapping.tag_mapping (tag_association_class, backref_name)
```

mapping_tests Module

Subpackages

migrate Package

check Module

```
galaxy.model.migrate.check.create_or_verify_database (url, galaxy_config_file, engine_options={}, app=None)
```

Check that the database is use-able, possibly creating it if empty (this is the only time we automatically create tables, otherwise we force the user to do it using the management script so they can create backups).

1. Empty database -> initialize with latest version and return
2. Database older than migration support -> fail and require manual update
3. Database at state where migrate support introduced -> add version control information but make no changes (might still require manual update)
4. Database versioned but out of date -> fail with informative message, user must run “sh manage_db.sh upgrade”

```
galaxy.model.migrate.check.migrate_to_current_version (engine, schema)
```

orm Package

orm Package

```
galaxy.model.orm.load_egg_for_url (url)
```

logging_connection_proxy Module

```
class galaxy.model.orm.logging_connection_proxy.LoggingProxy
```

Bases: sqlalchemy.interfaces.ConnectionProxy

Logs SQL statements using standard logging module

```
cursor_execute (execute, cursor, statement, parameters, context, executemany)
```

```
class galaxy.model.orm.logging_connection_proxy.TraceLoggerProxy (trace_logger)
```

Bases: sqlalchemy.interfaces.ConnectionProxy

Logs SQL statements using a metlog client

```
cursor_execute (execute, cursor, statement, parameters, context, executemany)
```

```
galaxy.model.orm.logging_connection_proxy.pretty_stack ()
```

```
galaxy.model.orm.logging_connection_proxy.stripwd (s)
```

Subpackages

ext Package

ext Package

assignmapper Module

managers Package

managers Package Classes that manage resources (models, tools, etc.) by using the current Transaction.

Encapsulates the intersection of trans (or trans.sa_session), models, and Controllers.

Responsibilities: model operations that involve the trans/sa_session (CRUD) security:

ownership, accessibility

common aspect-oriented operations via new mixins: sharable, annotatable, tagable, ratable

Not responsible for: encoding/decoding ids any http gobblygook formatting of returned data (always python structures) formatting of raised errors

The goal is to have Controllers only handle: query-string/payload parsing and encoding/decoding ids http return formatting

and: control, improve namespacing in Controllers DRY for Controller ops (define here - use in both UI/API Controllers)

In other words, 'Business logic' independent of web transactions/user context (trans) should be pushed into models - but logic that requires the context trans should be placed under this module.

api_keys Module

```
class galaxy.managers.api_keys.ApiKeyManager (app)
    Bases: object

    create_api_key (user)

    get_or_create_api_key (user)
```

base Module

```
class galaxy.managers.base.ModelManager
    Bases: object
class galaxy.managers.base.ModelSerializer
    Bases: object
```

```
galaxy.managers.base.get_class (class_name)
    Returns the class object that a string denotes. Without this method, we'd have to do eval(<class_name>).
```

```
galaxy.managers.base.get_object (trans, id, class_name, check_ownership=False,
                                   check_accessible=False, deleted=None)
```

Convenience method to get a model object with the specified checks. This is a generic method for dealing with objects uniformly from the older controller mixin code - however whenever possible the managers for a particular model should be used to load objects.


```
galaxy.managers.base.security_check (trans, item, check_ownership=False,
                                       check_accessible=False)
```

Security checks for an item: checks if (a) user owns item or (b) item is accessible to user. This is a generic method for dealing with objects uniformly from the older controller mixin code - however whenever possible the managers for a particular model should be used to perform security checks.

citations Module

```
class galaxy.managers.citations.BaseCitation
```

Bases: object

```
equals (other_citation)
```

```
has_doi ()
```

```
to_dict (citation_format)
```

```
class galaxy.managers.citations.BibtexCitation (elem, directory, citation_manager)
```

Bases: *galaxy.managers.citations.BaseCitation*

```
to_bibtex ()
```

```
class galaxy.managers.citations.CitationCollection
```

Bases: object

```
add (new_citation)
```

```
class galaxy.managers.citations.CitationsManager (app)
```

Bases: object

```
citations_for_tool (tool)
```

```
citations_for_tool_ids (tool_ids)
```

```
parse_citation (citation_elem, tool_directory)
```

```
class galaxy.managers.citations.DoiCache (config)
```

Bases: object

```
get_bibtex (doi)
```

```
class galaxy.managers.citations.DoiCitation (elem, directory, citation_manager)
```

Bases: *galaxy.managers.citations.BaseCitation*

```
BIBTEX_UNSET = <object object>
```

```
doi ()
```

```
has_doi ()
```

```
to_bibtex ()
```

```
galaxy.managers.citations.parse_citation (elem, directory, citation_manager)
```

Parse an abstract citation entry from the specified XML element. The directory parameter should be used to find external files for this citation.

collections Module

```
class galaxy.managers.collections.DatasetCollectionManager (app)
```

Bases: object

Abstraction for interfacing with dataset collections instance - ideally abstracts out model and plugin details.

```
copy (trans, parent, source, encoded_source_id)
```

```
create (trans, parent, name, collection_type, element_identifiers=None, elements=None, implicit_collection_info=None)
```

delete (*trans, instance_type, id*)

get_dataset_collection (*trans, encoded_id*)

get_dataset_collection_instance (*trans, instance_type, id, **kwds*)

history_dataset_collections (*history, query*)

match_collections (*collections_to_match*)

May seem odd to place it here, but planning to grow sophistication and get plugin types involved so it will likely make sense in the future.

update (*trans, instance_type, id, payload*)

collections_util Module

galaxy.managers.collections_util.**api_payload_to_create_params** (*payload*)

Cleanup API payload to pass into dataset_collections.

galaxy.managers.collections_util.**dictify_dataset_collection_instance** (*dataset_colleciton_instance, parent, security, view='element'*)

galaxy.managers.collections_util.**dictify_element** (*element*)

galaxy.managers.collections_util.**validate_input_element_identifiers** (*element_identifiers*)

Scan through the list of element identifiers supplied by the API consumer and verify the structure is valid.

context Module Mixins for transaction-like objects.

class galaxy.managers.context.**ProvidesAppContext**

Bases: object

For transaction-like objects to provide Galaxy convience layer for database and event handling.

Mixed in class must provide *app* property.

expunge_all ()

get_toolbox ()

Returns the application toolbox

install_model

log_action (*user=None, action=None, context=None, params=None*)

Application-level logging of user actions.

log_event (*message, tool_id=None, **kwargs*)

Application level logging. Still needs fleshing out (log levels and such) Logging events is a config setting - if False, do not log.

model

request_types ()

sa_session

Returns a SQLAlchemy session – currently just gets the current session from the threadlocal session context, but this is provided to allow migration toward a more SQLAlchemy 0.4 style of use.

class galaxy.managers.context.**ProvidesHistoryContext**

Bases: object

For transaction-like objects to provide Galaxy convience layer for reasoning about histories.

Mixed in class must provide *user*, *history*, and *app* properties.

db_builds

Returns the builds defined by galaxy and the builds defined by the user (chromInfo in history).

db_dataset_for (*dbkey*)

Returns the db_file dataset associated/needed by *dataset*, or *None*.

class galaxy.managers.context.**ProvidesUserContext**

Bases: object

For transaction-like objects to provide Galaxy convenience layer for reasoning about users.

Mixed in class must provide *user*, *api_inherit_admin*, and *app* properties.

anonymous

get_current_user_roles ()

user_can_do_run_as ()

user_ftp_dir

user_is_admin ()

folders Module Manager and Serializer for Library Folders.

class galaxy.managers.folders.**FolderManager**

Bases: object

Interface/service object for interacting with folders.

can_add_item (*trans*, *folder*)

Return true if the user has permissions to add item to the given folder.

check_accessible (*trans*, *folder*)

Check whether the folder is accessible to current user. By default every folder is accessible (contents have their own permissions).

check_manageable (*trans*, *folder*)

Check whether the user can manage the folder.

Returns the original folder

Return type *LibraryFolder*

Raises AuthenticationRequired, InsufficientPermissionsException

create (*trans*, *parent_folder_id*, *new_folder_name*, *new_folder_description*='')

Create a new folder under the given folder.

Parameters

- **parent_folder_id** (*int*) – decoded id
- **new_folder_name** (*str*) – name of the new folder
- **new_folder_description** (*str*) – description of the folder (optional, defaults to empty string)

Returns the new folder

Return type *LibraryFolder*

Raises InsufficientPermissionsException

cut_and_decode (*trans, encoded_folder_id*)

Cuts the folder prefix (the prepended 'F') and returns the decoded id.

cut_the_prefix (*encoded_folder_id*)

Remove the prefix from the encoded folder id.

decode_folder_id (*trans, encoded_folder_id*)

Decode the folder id given that it has already lost the prefixed 'F'.

delete (*trans, folder, undelete=False*)

Mark given folder deleted/undeleted based on the flag.

Raises ItemAccessibilityException

get (*trans, decoded_folder_id, check_manageable=False, check_accessible=True*)

Get the folder from the DB.

Parameters

- **decoded_folder_id** (*int*) – decoded folder id
- **check_manageable** (*bool*) – flag whether the check that user can manage item
- **check_accessible** (*bool*) – flag whether to check that user can access item

Returns the requested folder

Return type [LibraryFolder](#)

get_current_roles (*trans, folder*)

Find all roles currently connected to relevant permissions on the folder.

Parameters **folder** ([LibraryFolder](#)) – the model object

Returns dict of current roles for all available permission types

Return type dictionary

get_folder_dict (*trans, folder*)

Return folder data in the form of a dictionary.

Parameters **folder** ([LibraryFolder](#)) – folder item

Returns dict with data about the folder

Return type dictionary

secure (*trans, folder, check_manageable=True, check_accessible=True*)

Check if (a) user can manage folder or (b) folder is accessible to user.

Parameters

- **folder** ([LibraryFolder](#)) – folder item
- **check_manageable** (*bool*) – flag whether to check that user can manage item
- **check_accessible** (*bool*) – flag whether to check that user can access item

Returns the original folder

Return type [LibraryFolder](#)

hdas Module Manager and Serializer for HDAs.

HistoryDatasetAssociations (HDAs) are datasets contained or created in a history.

```

class galaxy.managers.hdas.HDAManager
    Bases: galaxy.managers.base.ModelManager

    Interface/service object for interacting with HDAs.

    can_access_dataset (trans, hda)
        Use security agent to see if current user has access to dataset.

    check_accessible (trans, hda)
        Raise error if HDA is not accessible.

    check_ownership (trans, hda)
        Use history to see if current user owns HDA.

    err_if_uploading (trans, hda)
        Raise error if HDA is still uploading.

    get (trans, unencoded_id, check_ownership=True, check_accessible=True)
        Get an HDA by its unencoded db id, checking ownership (via its history) or accessibility (via dataset
        shares/permissions).

    get_display_apps (trans, hda)
        Return dictionary containing new-style display app urls.

    get_hda_dict (trans, hda)
        Return full details of this HDA in dictionary form.

    get_hda_dict_with_error (trans, hda=None, history_id=None, id=None, error_msg='Error')
        Return truncated serialization of HDA when error raised getting details.

    get_inaccessible_hda_dict (trans, hda)
        Return truncated serialization of HDA when inaccessible to user.

    get_old_display_applications (trans, hda)
        Return dictionary containing old-style display app urls.

    secure (trans, hda, check_ownership=True, check_accessible=True)
        check ownership (via its history) or accessibility (via dataset shares/permissions).

class galaxy.managers.hdas.HistorySerializer
    Bases: galaxy.managers.base.ModelSerializer

    Interface/service object for serializing HDAs into dictionaries.

```

histories Module Manager and Serializer for histories.

Histories are containers for datasets or dataset collections created (or copied) by users over the course of an analysis.

```

class galaxy.managers.histories.HistoryManager (*args, **kwargs)
    Bases: galaxy.managers.base.ModelManager

    Interface/service object for interacting with HDAs.

    by_user (trans, user=None, include_deleted=False, only_deleted=False)
        Get all the histories for a given user (defaulting to trans.user) ordered by update time and filtered on
        whether they've been deleted.

    check_accessible (trans, history)
        Raises error if the current user can't access the history.

    check_ownership (trans, history)
        Raises error if the current user is not the owner of the history.

```

get (*trans, unencoded_id, check_ownership=True, check_accessible=True, deleted=None*)
Get a History from the database by id, verifying ownership.

get_history_dict (*trans, history, contents_dictionaries=None*)
Returns history data in the form of a dictionary.

is_accessible (*trans, history*)
True if the user can access (read) the current history.

is_current (*trans, history*)
True if the given history is the user's current history.
Returns False if the session has no current history.

is_owner (*trans, history*)
True if the current user is the owner of the given history.

most_recent (*trans, user=None, deleted=False*)

secure (*trans, history, check_ownership=True, check_accessible=True*)
Checks if (a) user owns item or (b) item is accessible to user.

class `galaxy.managers.histories.HistorySerializer`
Bases: `galaxy.managers.base.ModelSerializer`
Interface/service object for serializing histories into dictionaries.

lddas Module

class `galaxy.managers.lddas.LDDAManager`
Bases: `galaxy.managers.base.ModelManager`
A fairly sparse manager for LDDAs.
get (*trans, id, check_accessible=True*)

libraries Module Manager and Serializer for libraries.

class `galaxy.managers.libraries.LibraryManager` (**args, **kwargs*)
Bases: `object`
Interface/service object for interacting with libraries.
check_accessible (*trans, library*)
Check whether the library is accessible to current user.
create (*trans, name, description='', synopsis=''*)
Create a new library.
delete (*trans, library, undelete=False*)
Mark given library deleted/undeleted based on the flag.
get (*trans, decoded_library_id, check_accessible=True*)
Get the library from the DB.

Parameters

- **decoded_library_id** (*int*) – decoded library id
- **check_accessible** (*bool*) – flag whether to check that user can access item

Returns the requested library

Return type `Library`

get_access_roles (*trans, library*)

Load access roles for all library permissions

get_add_roles (*trans, library*)

Load add roles for all library permissions

get_current_roles (*trans, library*)

Load all permissions currently related to the given library.

Parameters **library** ([Library](#)) – the model object

Return type dictionary

Returns dict of current roles for all available permission types

get_library_dict (*trans, library*)

Return library data in the form of a dictionary.

Parameters **library** ([Library](#)) – library

Returns dict with data about the library

Return type dictionary

get_manage_roles (*trans, library*)

Load manage roles for all library permissions

get_modify_roles (*trans, library*)

Load modify roles for all library permissions

is_public (*trans, library*)

Return true if lib is public.

list (*trans, deleted=False*)

Return a list of libraries from the DB.

Parameters **deleted** (*boolean (optional)*) – if True, show only deleted libraries, if False show only non-deleted

Returns query that will emit all accessible libraries

Return type sqlalchemy query

make_public (*trans, library*)

Makes the given library public (removes all access roles)

secure (*trans, library, check_accessible=True*)

Check if library is accessible to user.

Parameters

- **folder** ([Library](#)) – library
- **check_accessible** (*bool*) – flag whether to check that user can access library

Returns the original folder

Return type [LibraryFolder](#)

set_permission_roles (*trans, library, access_roles, modify_roles, manage_roles, add_roles*)

Set permissions on the given library.

update (*trans, library, name=None, description=None, synopsis=None*)

Update the given library

roles Module Manager and Serializer for Roles.

class `galaxy.managers.roles.RoleManager`

Bases: `object`

Interface/service object for interacting with folders.

get (*trans*, *decoded_role_id*)

Method loads the role from the DB based on the given role id.

Parameters `decoded_role_id` (*int*) – id of the role to load from the DB

Returns the loaded Role object

Return type `Role`

Raises `InconsistentDatabase`, `RequestParamterInvalidException`, `InternalServerError`

tags Module

class `galaxy.managers.tags.TagsManager` (*app*)

Bases: `object`

Manages CRUD operations related to tagging objects.

set_tags_from_list (*trans*, *item*, *new_tags_list*, *user=None*)

workflows Module

class `galaxy.managers.workflows.WorkflowsManager` (*app*)

Bases: `object`

Handle CRUD type operaitons related to workflows. More interesting stuff regarding workflow execution, step sorting, etc... can be found in the `galaxy.workflow` module.

build_invocations_query (*trans*, *decoded_stored_workflow_id*)

cancel_invocation (*trans*, *decoded_invocation_id*)

check_security (*trans*, *has_workflow*, *check_ownership=True*, *check_accessible=True*)

check accessibility or ownership of workflows, storedworkflows, and workflowinvocations. Throw an exception or returns True if user has needed level of access.

get_invocation (*trans*, *decoded_invocation_id*)

get_invocation_step (*trans*, *decoded_workflow_invocation_step_id*)

update_invocation_step (*trans*, *decoded_workflow_invocation_step_id*, *action*)

objectstore Package

objectstore Package objectstore package, abstraction for storing blobs of data for use in Galaxy, all providers ensure that data can be accessed on the filesystem for running tools

class `galaxy.objectstore.CachingObjectStore` (*path*, *backend*)

Bases: `galaxy.objectstore.ObjectStore`

Object store that uses a directory for caching files, but defers and writes back to another object store.

class `galaxy.objectstore.DiskObjectStore` (*config*, *config_xml=None*, *file_path=None*, *extra_dirs=None*)

Bases: `galaxy.objectstore.ObjectStore`

Standard Galaxy object store, stores objects in files under a specific directory on disk.


```

>>> from galaxy.util.bunch import Bunch
>>> import tempfile
>>> file_path=tempfile.mkdtemp()
>>> obj = Bunch(id=1)
>>> s = DiskObjectStore(Bunch(umask=077, job_working_directory=file_path, new_file_path=file_path))
>>> s.create(obj)
>>> s.exists(obj)
True
>>> assert s.get_filename(obj) == file_path + '/000/dataset_1.dat'

```

create (*obj*, ***kwargs*)

delete (*obj*, *entire_dir=False*, ***kwargs*)

empty (*obj*, ***kwargs*)

exists (*obj*, ***kwargs*)

get_data (*obj*, *start=0*, *count=-1*, ***kwargs*)

get_filename (*obj*, ***kwargs*)

get_object_url (*obj*, ***kwargs*)

get_store_usage_percent ()

size (*obj*, ***kwargs*)

update_from_file (*obj*, *file_name=None*, *create=False*, ***kwargs*)

create parameter is not used in this implementation

class `galaxy.objectstore.DistributedObjectStore` (*config*, *config_xml=None*, *fsmon=False*)

Bases: `galaxy.objectstore.NestedObjectStore`

ObjectStore that defers to a list of backends, for getting objects the first store where the object exists is used, objects are created in a store selected randomly, but with weighting.

create (*obj*, ***kwargs*)

create() is the only method in which *obj.object_store_id* may be None

shutdown ()

class `galaxy.objectstore.HierarchicalObjectStore` (*config*, *config_xml=None*, *fsmon=False*)

Bases: `galaxy.objectstore.NestedObjectStore`

ObjectStore that defers to a list of backends, for getting objects the first store where the object exists is used, objects are always created in the first store.

create (*obj*, ***kwargs*)

Create will always be called by the primary *object_store*

exists (*obj*, ***kwargs*)

Exists must check all child object stores

class `galaxy.objectstore.NestedObjectStore` (*config*, *config_xml=None*)

Bases: `galaxy.objectstore.ObjectStore`

Base for ObjectStores that use other ObjectStores (`DistributedObjectStore`, `HierarchicalObjectStore`)

create (*obj*, ***kwargs*)

delete (*obj*, ***kwargs*)

empty (*obj*, ***kwargs*)

```
exists (obj, **kwargs)  
file_ready (obj, **kwargs)  
get_data (obj, **kwargs)  
get_filename (obj, **kwargs)  
get_object_url (obj, **kwargs)  
shutdown ()  
size (obj, **kwargs)  
update_from_file (obj, **kwargs)
```

```
class galaxy.objectstore.ObjectStore (config, config_xml=None, **kwargs)
```

Bases: `object`

ObjectStore abstract interface

```
create (obj, base_dir=None, dir_only=False, extra_dir=None, extra_dir_at_root=False,  
        alt_name=None)
```

Mark the object identified by *obj* as existing in the store, but with no content. This method will create a proper directory structure for the file if the directory does not already exist. See *exists* method for the description of other fields.

```
delete (obj, entire_dir=False, base_dir=None, extra_dir=None, extra_dir_at_root=False,  
        alt_name=None)
```

Deletes the object identified by *obj*. See *exists* method for the description of other fields.

Parameters *entire_dir* (*bool*) – If True, delete the entire directory pointed to by *extra_dir*. For safety reasons, this option applies only for and in conjunction with the *extra_dir* option.

```
empty (obj, base_dir=None, extra_dir=None, extra_dir_at_root=False, alt_name=None)
```

Test if the object identified by *obj* has content. If the object does not exist raises *ObjectNotFound*. See *exists* method for the description of the fields.

```
exists (obj, base_dir=None, dir_only=False, extra_dir=None, extra_dir_at_root=False,  
        alt_name=None)
```

Returns True if the object identified by *obj* exists in this file store, False otherwise.

FIELD DESCRIPTIONS (these apply to all the methods in this class):

Parameters

- **obj** (*object*) – A Galaxy object with an assigned database ID accessible via the *.id* attribute.
- **base_dir** (*string*) – A key in *self.extra_dirs* corresponding to the base directory in which this object should be created, or None to specify the default directory.
- **dir_only** (*bool*) – If True, check only the path where the file identified by *obj* should be located, not the dataset itself. This option applies to *extra_dir* argument as well.
- **extra_dir** (*string*) – Append *extra_dir* to the directory structure where the dataset identified by *obj* should be located. (e.g., 000/*extra_dir*/*obj.id*)
- **extra_dir_at_root** (*bool*) – Applicable only if *extra_dir* is set. If True, the *extra_dir* argument is placed at root of the created directory structure rather than at the end (e.g., *extra_dir*/000/*obj.id* vs. 000/*extra_dir*/*obj.id*)
- **alt_name** (*string*) – Use this name as the alternative name for the created dataset rather than the default.

file_ready (*obj*, *base_dir=None*, *dir_only=False*, *extra_dir=None*, *extra_dir_at_root=False*, *alt_name=None*)

A helper method that checks if a file corresponding to a dataset is ready and available to be used. Return True if so, False otherwise.

get_data (*obj*, *start=0*, *count=-1*, *base_dir=None*, *extra_dir=None*, *extra_dir_at_root=False*, *alt_name=None*)

Fetch *count* bytes of data starting at offset *start* from the object identified uniquely by *obj*. If the object does not exist raises *ObjectNotFound*. See *exists* method for the description of other fields.

Parameters

- **start** (*int*) – Set the position to start reading the dataset file
- **count** (*int*) – Read at most *count* bytes from the dataset

get_filename (*obj*, *base_dir=None*, *dir_only=False*, *extra_dir=None*, *extra_dir_at_root=False*, *alt_name=None*)

Get the expected filename (including the absolute path) which can be used to access the contents of the object uniquely identified by *obj*. See *exists* method for the description of the fields.

get_object_url (*obj*, *extra_dir=None*, *extra_dir_at_root=False*, *alt_name=None*)

If the store supports direct URL access, return a URL. Otherwise return None. Note: need to be careful to to bypass dataset security with this. See *exists* method for the description of the fields.

get_store_usage_percent ()

Return the percentage indicating how full the store is

shutdown ()

size (*obj*, *extra_dir=None*, *extra_dir_at_root=False*, *alt_name=None*)

Return size of the object identified by *obj*. If the object does not exist, return 0. See *exists* method for the description of the fields.

update_from_file (*obj*, *base_dir=None*, *extra_dir=None*, *extra_dir_at_root=False*, *alt_name=None*, *file_name=None*, *create=False*)

Inform the store that the file associated with the object has been updated. If *file_name* is provided, update from that file instead of the default. If the object does not exist raises *ObjectNotFound*. See *exists* method for the description of other fields.

Parameters

- **file_name** (*string*) – Use file pointed to by *file_name* as the source for updating the dataset identified by *obj*
- **create** (*bool*) – If True and the default dataset does not exist, create it first.

galaxy.objectstore.build_object_store_from_config (*config*, *fsmon=False*, *config_xml=None*)

Depending on the configuration setting, invoke the appropriate object store

galaxy.objectstore.convert_bytes (*bytes*)

A helper function used for pretty printing disk usage

galaxy.objectstore.create_object_in_session (*obj*)

galaxy.objectstore.local_extra_dirs (*func*)

A decorator for non-local plugins to utilize local directories for their *extra_dirs* (*job_working_directory* and *temp*).

s3_multipart_upload Module Split large file into multiple pieces for upload to S3. This parallelizes the task over available cores using multiprocessing. Code mostly taken from CloudBioLinux.

```
galaxy.objectstore.s3_multipart_upload.map_wrap(f)
```

```
galaxy.objectstore.s3_multipart_upload.mp_from_ids(mp_id, mp_keyname,
                                                    mp_bucketname,
                                                    aws_access_key_id,
                                                    aws_secret_access_key)
```

Get the multipart upload from the bucket and multipart IDs.

This allows us to reconstitute a connection to the upload from within multiprocessing functions.

```
galaxy.objectstore.s3_multipart_upload.multimap(*args, **kws)
```

Provide multiprocessing imap like function.

The context manager handles setting up the pool, worked around interrupt issues and terminating the pool on completion.

```
galaxy.objectstore.s3_multipart_upload.multipart_upload(bucket, s3_key_name,
                                                         tarball, mb_size,
                                                         aws_access_key_id,
                                                         aws_secret_access_key,
                                                         use_rr=True)
```

Upload large files using Amazon's multipart upload functionality.

```
galaxy.objectstore.s3_multipart_upload.transfer_part(*args, **kwargs)
```

Transfer a part of a multipart upload. Designed to be run in parallel.

openid Package

openid Package OpenID functionality

providers Module Contains OpenID provider functionality

```
class galaxy.openid.providers.OpenIDProvider(id, name, op_endpoint_url,
                                             sreg_required=None, sreg_optional=None,
                                             use_for=None, store_user_preference=None,
                                             never_associate_with_user=None)
```

Bases: object

An OpenID Provider object.

```
classmethod from_elem(xml_root)
```

```
classmethod from_file(filename)
```

```
has_post_authentication_actions()
```

```
post_authentication(trans, openid_manager, info)
```

```
class galaxy.openid.providers.OpenIDProviders(providers=None)
```

Bases: object

Collection of OpenID Providers

```
NO_PROVIDER_ID = 'None'
```

```
classmethod from_elem(xml_root)
```

```
classmethod from_file(filename)
```

```
get(name, default=None)
```

```
new_provider_from_identifier(identifier)
```

quota Package

quota Package Galaxy Quotas

class `galaxy.quota.NoQuotaAgent(model)`

Bases: `object`

Base quota agent, always returns no quota

default_quota

get_percent (*trans=None, user=False, history=False, usage=False, quota=False*)

get_quota (*user, nice_size=False*)

get_usage (*trans=None, user=False, history=False*)

get_user_quotas (*user*)

class `galaxy.quota.QuotaAgent(model)`

Bases: `galaxy.quota.NoQuotaAgent`

Class that handles galaxy quotas

default_registered_quota

default_unregistered_quota

get_percent (*trans=None, user=False, history=False, usage=False, quota=False*)

Return the percentage of any storage quota applicable to the user/transaction.

get_quota (*user, nice_size=False*)

Calculated like so:

1. Anonymous users get the default quota.
2. Logged in users start with the highest of their associated '=' quotas or the default quota, if there are no associated '=' quotas. If an '=' unlimited (-1 in the database) quota is found during this process, the user has no quota (aka unlimited).
3. Quota is increased or decreased by any corresponding '+' or '-' quotas.

get_user_quotas (*user*)

set_default_quota (*default_type, quota*)

set_entity_quota_associations (*quotas=[], users=[], groups=[], delete_existing_assocs=True*)

sample_tracking Package

data_transfer Module

class `galaxy.sample_tracking.data_transfer.DataTransferFactory`

Bases: `object`

parse ()

type = `None`

class `galaxy.sample_tracking.data_transfer.FtpDataTransferFactory`

Bases: `galaxy.sample_tracking.data_transfer.DataTransferFactory`

parse (*elem*)

type = `'ftp'`

```
class galaxy.sample_tracking.data_transfer.HttpDataTransferFactory
    Bases: galaxy.sample_tracking.data_transfer.DataTransferFactory
```

```
    parse (config_file, elem)
```

```
    type = 'http'
```

```
class galaxy.sample_tracking.data_transfer.ScpDataTransferFactory
    Bases: galaxy.sample_tracking.data_transfer.DataTransferFactory
```

```
    parse (config_file, elem)
```

```
    type = 'scp'
```

```
galaxy.sample_tracking.data_transfer.data_transfer
    alias of FtpDataTransferFactory
```

external_service_types Module

```
class galaxy.sample_tracking.external_service_types.ExternalServiceType (external_service_type_xml_conf,
                                                                           root,
                                                                           visible=True)
```

```
    Bases: object
```

```
    parse (root)
```

```
    parse_data_transfer_settings (root)
```

```
    parse_run_details (root)
```

```
    parse_run_details_results (root)
```

```
exception galaxy.sample_tracking.external_service_types.ExternalServiceTypeNotFoundException
    Bases: exceptions.Exception
```

```
class galaxy.sample_tracking.external_service_types.ExternalServiceTypesCollection (config_filename,
                                                                                       root_dir,
                                                                                       app)
```

```
    Bases: object
```

```
    load_all (config_filename)
```

```
    load_external_service_type (config_file, visible=True)
```

```
    reload (external_service_type_id)
```

```
        Attempt to reload the external_service_type identified by 'external_service_type_id', if successful replace
        the old external_service_type.
```

request_types Module RequestType

```
class galaxy.sample_tracking.request_types.RequestTypeFactory (sample_state_factory,
                                                                re-
                                                                name_dataset_options)
```

```
    Bases: object
```

```
    from_elem (elem, request_form, sample_form, external_service)
        Return RequestType created from an xml string.
```

```
    new (name, request_form, sample_form, external_service, description=None, sample_states=None)
        Return new RequestType.
```

sample Module Sample

```
class galaxy.sample_tracking.sample.SampleStateFactory
```

```
    Bases: object
```

```
    from_elem (request_type, elem)
```

```
        Return SampleState created from an xml string.
```

```
    new (request_type, name, description=None)
```

```
        Return new SampleState.
```

security Package**security Package** Galaxy Security

```
class galaxy.security.Action (action, description, model)
```

```
    Bases: object
```

```
class galaxy.security.GalaxyRBACAgent (model, permitted_actions=None)
```

```
    Bases: galaxy.security.RBACAgent
```

```
    allow_action (roles, action, item)
```

```
        Method for checking a permission for the current user ( based on roles ) to perform a specific action on an
        item, which must be one of: Dataset, Library, LibraryFolder, LibraryDataset, LibraryDatasetDatasetAs-
        sociation
```

```
    allow_action_on_libitems (trans, user_roles, action, items)
```

```
        This should be the equivalent of allow_action defined on multiple items. It is meant to specifically replace
        allow_action for multiple LibraryDatasets, but it could be reproduced or modified for allow_action's per-
        mitted classes - Dataset, Library, LibraryFolder, and LDDAs.
```

```
    associate_action_dataset_role (action, dataset, role)
```

```
    associate_components (**kwd)
```

```
    associate_group_role (group, role)
```

```
    associate_user_group (user, group)
```

```
    associate_user_role (user, role)
```

```
    can_access_dataset (user_roles, dataset)
```

```
    can_access_library (roles, library)
```

```
    can_access_library_item (roles, item, user)
```

```
    can_access_request_type (roles, request_type)
```

```
    can_add_library_item (roles, item)
```

```
    can_manage_dataset (roles, dataset)
```

```
    can_manage_library_item (roles, item)
```

```
    can_modify_library_item (roles, item)
```

```
    check_folder_contents (user, roles, folder, hidden_folder_ids='')
```

```
        This method must always be sent an instance of LibraryFolder(). Recursive execution produces a comma-
        separated string of folder ids whose folders do NOT meet the criteria for showing. Along with the string,
        True is returned if the current user has permission to access folder. Otherwise, cycle through all sub-
        folders in folder until one is found that meets this criteria, if it exists. This method does not necessarily
        scan the entire library as it returns when it finds the first folder that is accessible to user.
```

copy_dataset_permissions (*src, dst*)

copy_library_permissions (*trans, source_library_item, target_library_item, user=None*)

create_private_user_role (*user*)

dataset_access_mapping (*trans, user_roles, datasets*)

For the given list of datasets, return a mapping of the datasets' ids to whether they can be accessed by the user or not. The datasets input is expected to be a simple list of Dataset objects.

dataset_is_private_to_user (*trans, dataset*)

If the LibraryDataset object has exactly one access role and that is the current user's private role then we consider the dataset private.

dataset_is_public (*dataset*)

A dataset is considered public if there are no "access" actions associated with it. Any other actions ('manage permissions', 'edit metadata') are irrelevant. Accessing dataset.actions will cause a query to be emitted.

dataset_is_unrestricted (*trans, dataset*)

Different implementation of the method above with signature: def dataset_is_public(self, dataset)

dataset_permission_map_for_access (*trans, user_roles, libitems*)

For a given list of library items (e.g., Datasets), return a map of the datasets' ids to whether they can have permission to use that action (e.g., "access" or "modify") on the dataset. The libitems input is expected to be a simple list of library items, such as Datasets or LibraryDatasets. NB: This is currently only usable for Datasets; it was intended to be used for any library item.

datasets_are_public (*trans, datasets*)

Given a transaction object and a list of Datasets, return a mapping from Dataset ids to whether the Dataset is public or not. All Dataset ids should be returned in the mapping's keys.

derive_roles_from_access (*trans, item_id, cntrller, library=False, **kwd*)

folder_is_public (*folder*)

folder_is_unrestricted (*folder*)

get_accessible_libraries (*trans, user*)

Return all data libraries that the received user can access

get_accessible_request_types (*trans, user*)

Return all RequestTypes that the received user has permission to access.

get_actions_for_items (*trans, action, permission_items*)

get_all_roles (*trans, cntrller*)

get_component_associations (***kwd*)

get_item_actions (*action, item*)

get_legitimate_roles (*trans, item, cntrller*)

Return a sorted list of legitimate roles that can be associated with a permission on item where item is a Library or a Dataset. The cntrller param is the controller from which the request is sent. We cannot use trans.user_is_admin() because the controller is what is important since admin users do not necessarily have permission to do things on items outside of the admin view.

If cntrller is from the admin side (e.g., library_admin):

- if item is public, all roles, including private roles, are legitimate.

- if item is restricted, legitimate roles are derived from the users and groups associated with each role that is associated with the access permission (i.e., DATASET_MANAGE_PERMISSIONS or LIBRARY_MANAGE) on item. Legitimate roles will include private roles.

If cntrller is not from the admin side (e.g., root, library):

- if item is public, all non-private roles, except for the current user's private role, are legitimate.
- if item is restricted, legitimate roles are derived from the users and groups associated with each role that is associated with the access permission on item. Private roles, except for the current user's private role, will be excluded.

get_permissions (*item*)

Return a dictionary containing the actions and associated roles on item where item is one of Library, LibraryFolder, LibraryDatasetDatasetAssociation, LibraryDataset, Dataset. The dictionary looks like: { Action : [Role, Role] }.

get_permitted_libraries (*trans, user, actions*)

This method is historical (it is not currently used), but may be useful again at some point. It returns a dictionary whose keys are library objects and whose values are a comma-separated string of folder ids. This method works with the show_library_item() method below, and it returns libraries for which the received user has permission to perform the received actions. Here is an example call to this method to return all libraries for which the received user has LIBRARY_ADD permission:

```
libraries = trans.app.security_agent.get_permitted_libraries( trans, user,
    [ trans.app.security_agent.permitted_actions.LIBRARY_ADD ] )
```

get_private_user_role (*user, auto_create=False*)

get_roles_for_action (*item, action*)

Return a list containing the roles associated with given action on given item where item is one of Library, LibraryFolder, LibraryDatasetDatasetAssociation, LibraryDataset, Dataset.

get_sharing_roles (*user*)

get_showable_folders (*user, roles, library_item, actions_to_check, hidden_folder_ids=[], showable_folders=[]*)

This method must be sent an instance of Library(), all the folders of which are scanned to determine if user is allowed to perform any action in actions_to_check. The param hidden_folder_ids, if passed, should contain a list of folder IDs which was generated when the library was previously scanned using the same actions_to_check. A list of showable folders is generated. This method scans the entire library.

get_valid_roles (*trans, item, query=None, page=None, page_limit=None, is_library_access=False*)

This method retrieves the list of possible roles that user can select in the item permissions form. Admins can select any role so the results are paginated in order to save the bandwidth and to speed things up. Standard users can select their own private role, any of their sharing roles and any public role (not private and not sharing).

guess_derived_permissions_for_datasets (*datasets=[]*)

Returns a dict of { action : [role, role, ...] } for the output dataset based upon provided datasets

has_accessible_folders (*trans, folder, user, roles, search_downward=True*)

has_accessible_library_datasets (*trans, folder, user, roles, search_downward=True*)

history_get_default_permissions (*history*)

history_set_default_permissions (*history, permissions={}, dataset=False, bypass_manage_permission=False*)

item_permission_map_for_add (*trans, user_roles, libitems*)

item_permission_map_for_manage (*trans, user_roles, libitems*)

item_permission_map_for_modify (*trans, user_roles, libitems*)

library_is_public (*library, contents=False*)

library_is_unrestricted (*library*)

make_dataset_public (*dataset*)

make_folder_public (*folder*)

make_library_public (*library, contents=False*)

ok_to_display (*user, role*)
Method for checking if: - a role is private and is the current user's private role - a role is a sharing role and belongs to the current user

privately_share_dataset (*dataset, users=[]*)

sa_session
Returns a SQLAlchemy session

set_all_dataset_permissions (*dataset, permissions={}*)
Set new full permissions on a dataset, eliminating all current permissions. Permission looks like: { Action : [Role, Role] }

set_all_library_permissions (*trans, library_item, permissions={}*)

set_dataset_permission (*dataset, permission={}*)
Set a specific permission on a dataset, leaving all other current permissions on the dataset alone. Permission looks like: { Action.action : [Role, Role] }

set_entity_group_associations (*groups=[], users=[], roles=[], delete_existing_assocs=True*)

set_entity_role_associations (*roles=[], users=[], groups=[], delete_existing_assocs=True*)

set_entity_user_associations (*users=[], roles=[], groups=[], delete_existing_assocs=True*)

set_library_item_permission (*library_item, permission={}*)
Set a specific permission on a library item, leaving all other current permissions on the item alone. Permission looks like: { Action.action : [Role, Role] }

set_request_type_permissions (*request_type, permissions={}*)

show_library_item (*user, roles, library_item, actions_to_check, hidden_folder_ids=''*)
This method must be sent an instance of Library() or LibraryFolder(). Recursive execution produces a comma-separated string of folder ids whose folders do NOT meet the criteria for showing. Along with the string, True is returned if the current user has permission to perform any 1 of actions_to_check on library_item. Otherwise, cycle through all sub-folders in library_item until one is found that meets this criteria, if it exists. This method does not necessarily scan the entire library as it returns when it finds the first library_item that allows user to perform any one action in actions_to_check.

sort_by_attr (*seq, attr*)
Sort the sequence of objects by object's attribute Arguments: seq - the list or any sequence (including immutable one) of objects to sort. attr - the name of attribute to sort by

user_get_default_permissions (*user*)

user_set_default_permissions (*user, permissions={}, history=False, dataset=False, bypass_manage_permission=False, default_access_private=False*)

```

class galaxy.security.HostAgent(model, permitted_actions=None)
    Bases: galaxy.security.RBACAgent

    A simple security agent which allows access to datasets based on host. This exists so that external sites such as
    UCSC can gain access to datasets which have permissions which would normally prevent such access.

    allow_action(addr, action, **kwd)

    sa_session
        Returns a SQLAlchemy session

    set_dataset_permissions(hda, user, site)

    sites = <galaxy.util.bunch.Bunch object>

class galaxy.security.RBACAgent
    Class that handles galaxy security

    associate_components(**kwd)

    can_access_dataset(roles, dataset)

    can_access_library(roles, library)

    can_add_library_item(roles, item)

    can_manage_dataset(roles, dataset)

    can_manage_library_item(roles, item)

    can_modify_library_item(roles, item)

    components_are_associated(**kwd)

    convert_permitted_action_strings(permitted_action_strings)
        When getting permitted actions from an untrusted source like a form, ensure that they match our actual
        permitted actions.

    create_private_user_role(user)

    dataset_is_public(dataset)

    derive_roles_from_access(trans, item_id, cntrller, library=False, **kwd)

    folder_is_public(library)

    get_accessible_libraries(trans, user)

    get_accessible_request_types(trans, user)

    get_action(name, default=None)
        Get a permitted action by its dict key or action name

    get_actions()
        Get all permitted actions as a list of Action objects

    get_all_roles(trans, cntrller)

    get_component_associations(**kwd)

    get_item_actions(action, item)

    get_legitimate_roles(trans, item, cntrller)

    get_permissions(library_dataset)

    get_permitted_libraries(trans, user, actions)

    get_private_user_role(user)

```

```
guess_derived_permissions_for_datasets (datasets=[])  
history_set_default_permissions (history, permissions=None, dataset=False, by-  
pass_manage_permission=False)  
library_is_public (library)  
make_dataset_public (dataset)  
make_folder_public (folder, count=0)  
make_library_public (library)  
permitted_actions = <galaxy.util.bunch.Bunch object>  
set_all_dataset_permissions (dataset, permissions)  
set_all_library_permissions (trans, dataset, permissions)  
set_dataset_permission (dataset, permission)  
set_library_item_permission (library_item, permission)  
user_set_default_permissions (user, permissions={}, history=False, dataset=False)
```

`galaxy.security.get_permitted_actions` (*filter=None*)
Utility method to return a subset of RBACAgent's permitted actions

validate_user_input Module Utilities for validating inputs related to user objects.

The `validate_*` methods in this file return simple messages that do not contain user inputs - so these methods do not need to be escaped.

```
galaxy.security.validate_user_input.transform_publicname (trans, publicname,  
user=None)  
galaxy.security.validate_user_input.validate_email (trans, email, user=None,  
check_dup=True)  
Validates the email format, also checks whether the domain is blacklisted in the disposable domains configura-  
tion.  
galaxy.security.validate_user_input.validate_password (trans, password, confirm)  
galaxy.security.validate_user_input.validate_publicname (trans, publicname,  
user=None)
```

tags Package

tags Package Galaxy tagging classes and methods.

tag_handler Module

```
class galaxy.tags.tag_handler.CommunityTagHandler  
    Bases: galaxy.tags.tag_handler.TagHandler  
class galaxy.tags.tag_handler.GalaxyTagHandler  
    Bases: galaxy.tags.tag_handler.TagHandler  
class galaxy.tags.tag_handler.ItemTagAssocInfo (item_class, tag_assoc_class, item_id_col)  
    Bases: object  
class galaxy.tags.tag_handler.TagHandler  
    Bases: object
```

apply_item_tag (*trans, user, item, name, value=None*)

apply_item_tags (*trans, user, item, tags_str*)

Apply tags to an item.

delete_item_tags (*trans, user, item*)

Delete tags from an item.

get_community_tags (*trans, item=None, limit=None*)

Returns community tags for an item.

get_id_col_in_item_tag_assoc_table (*item_class*)

Returns item id column in class' item-tag association table.

get_tag_assoc_class (*item_class*)

Returns tag association class for item class.

get_tag_by_id (*trans, tag_id*)

Get a Tag object from a tag id.

get_tag_by_name (*trans, tag_name*)

Get a Tag object from a tag name (string).

get_tags_str (*tags*)

Build a string from an item's tags.

get_tool_tags (*trans*)

item_has_tag (*trans, user, item, tag*)

Returns true if item is has a given tag.

parse_tags (*tag_str*)

Returns a list of raw (tag-name, value) pairs derived from a string; method scrubs tag names and values as well. Return value is a dictionary where tag-names are keys.

remove_item_tag (*trans, user, item, tag_name*)

Remove a tag from an item.

tool_shed Package

tool_shed Package

common_util Module

encoding_util Module

install_manager Module

tool_shed_registry Module

update_manager Module

Subpackages

migrate Package

check Module

common Module

tool_dependencies Package

common_util Module

fabric_util Module

install_util Module

tools Package

tools Package Classes encapsulating galaxy tools and tool configuration.

```
class galaxy.tools.AsyncDataSourceTool (config_file, root, app, guid=None, repository_id=None)
    Bases: galaxy.tools.DataSourceTool
```

```
    tool_type = 'data_source_async'
```

```
class galaxy.tools.BadValue (value)
    Bases: object
```

```
class galaxy.tools.DataDestinationTool (config_file, root, app, guid=None, repository_id=None)
    Bases: galaxy.tools.Tool
```

```
    tool_type = 'data_destination'
```

```
class galaxy.tools.DataManagerTool (config_file, root, app, guid=None, data_manager_id=None,
                                     **kws)
    Bases: galaxy.tools.OutputParameterJSONTool
```

```
    allow_user_access (user)
```

Returns bool – Whether the user is allowed to access the tool.

 Data Manager tools are only accessible to admins.

```
    default_tool_action
        alias of DataManagerToolAction
```

```
    exec_after_process (app, inp_data, out_data, param_dict, job=None, **kws)
```

```
    get_default_history_by_trans (trans, create=False)
```

```
    tool_type = 'manage_data'
```

```
class galaxy.tools.DataSourceTool (config_file, root, app, guid=None, repository_id=None)
    Bases: galaxy.tools.OutputParameterJSONTool
```

 Alternate implementation of Tool for data_source tools – those that allow the user to query and extract data from another web site.

```
    default_tool_action
        alias of DataSourceToolAction
```

```
    exec_before_job (app, inp_data, out_data, param_dict=None)
```

```

    parse_inputs (root)

    tool_type = 'data_source'

class galaxy.tools.DefaultToolState
    Bases: object

    Keeps track of the state of a users interaction with a tool between requests. The default tool state keeps track of
    the current page (for multipage “wizard” tools) and the values of all

    copy ()
        WARNING! Makes a shallow copy, SHOULD rework to have it make a deep copy.

    decode (value, tool, app, secure=True)
        Restore the state from a string

    encode (tool, app, secure=True)
        Convert the data to a string

class galaxy.tools.ExportHistoryTool (config_file, root, app, guid=None, repository_id=None)
    Bases: galaxy.tools.Tool

    tool_type = 'export_history'

class galaxy.tools.GenomeIndexTool (config_file, root, app, guid=None, repository_id=None)
    Bases: galaxy.tools.Tool

    tool_type = 'index_genome'

class galaxy.tools.ImportHistoryTool (config_file, root, app, guid=None, repository_id=None)
    Bases: galaxy.tools.Tool

    tool_type = 'import_history'

exception galaxy.tools.InterruptedUpload
    Bases: exceptions.Exception

class galaxy.tools.OutputParameterJSONTool (config_file, root, app, guid=None, repository_id=None)
    Bases: galaxy.tools.Tool

    Alternate implementation of Tool that provides parameters and other values JSONified within the contents of an
    output dataset

    exec_before_job (app, inp_data, out_data, param_dict=None)

    tool_type = 'output_parameter_json'

class galaxy.tools.SetMetadataTool (config_file, root, app, guid=None, repository_id=None)
    Bases: galaxy.tools.Tool

    Tool implementation for special tool that sets metadata on an existing dataset.

    exec_after_process (app, inp_data, out_data, param_dict, job=None)

    job_failed (job_wrapper, message, exception=False)

    requires_setting_metadata = False

    tool_type = 'set_metadata'

class galaxy.tools.SetParamAction (name, output_name)
    Set parameter action.

    static parse (elt)
        Parse action from element.

```

```
class galaxy.tools.Tool (config_file, root, app, guid=None, repository_id=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable

    Represents a computational tool that can be executed through Galaxy.

    allow_user_access (user)
        Returns bool – Whether the user is allowed to access the tool.

    build_dependency_shell_commands ()
        Return a list of commands to be run to populate the current environment to include this tools requirements.

    build_redirect_url_params (param_dict)
        Substitute parameter values into self.redirect_url_params

    call_hook (hook_name, *args, **kwargs)
        Call the custom code hook function identified by 'hook_name' if any, and return the results

    check_and_update_param_values (values, trans, update_values=True, allow_workflow_parameters=False)
        Check that all parameters have values, and fill in with default values where necessary. This could be called
        after loading values from a database in case new parameters have been added.

    check_and_update_param_values_helper (inputs, values, trans, messages, context=None, prefix='', update_values=True,
                                          allow_workflow_parameters=False)
        Recursive helper for check_and_update_param_values_helper

    check_workflow_compatible (root)
        Determine if a tool can be used in workflows. External tools and the upload tool are currently not supported
        by workflows.

    collect_associated_files (output, job_working_directory)
        Find extra files in the job working directory and move them into the appropriate dataset's files directory

    collect_child_datasets (output, job_working_directory)
        Look for child dataset files, create HDA and attach to parent.

    collect_primary_datasets (output, job_working_directory, input_ext)
        Find any additional datasets generated by a tool and attach (for cases where number of outputs is not
        known in advance).

    default_template = 'tool_form.mako'

    default_tool_action
        alias of DefaultToolAction

    dict_collection_visible_keys = ('id', 'name', 'version', 'description')

    exec_after_process (app, inp_data, out_data, param_dict, job=None)

    exec_before_job (app, inp_data, out_data, param_dict={})

    execute (trans, incoming={}, set_output_hid=True, history=None, **kwargs)
        Execute the tool using parameter values in incoming. This just dispatches to the ToolAction instance
        specified by self.tool_action. In general this will create a Job that when run will build the tool's outputs,
        e.g. DefaultToolAction.

    fill_in_new_state (trans, inputs, state, context=None, history=None)
        Fill in a tool state dictionary with default values for all parameters in the dictionary inputs. Grouping
        elements are filled in recursively.

    find_fieldstorage (x)

    get_default_history_by_trans (trans, create=False)
```


classmethod `get_externally_referenced_paths` (*path*)

Return relative paths to externally referenced files by the tool described by file at *path*. External components should not assume things about the structure of tool xml files (this is the tool's responsibility).

get_hook (*name*)

Returns an object from the code file referenced by *code_namespace* (this will normally be a callable object)

get_job_destination (*job_params=None*)

Returns `galaxy.jobs.JobDestination` – The destination definition and runner parameters.

get_job_handler (*job_params=None*)

Get a suitable job handler for this *Tool* given the provided *job_params*. If multiple handlers are valid for combination of *Tool* and *job_params* (e.g. the defined handler is a handler tag), one will be selected at random.

Parameters *job_params* (*dict or None*) – Any params specific to this job (e.g. the job source)

Returns *str* – The id of a job handler for a job run of this *Tool*

get_panel_section ()

get_param (*key*)

Returns the parameter named *key* or *None* if there is no such parameter.

get_param_html_map (*trans, page=0, other_values={}*)

Return a dictionary containing the HTML representation of each parameter. This is used for rendering display elements. It is currently not compatible with grouping constructs.

NOTE: This should be considered deprecated, it is only used for tools with *display* elements. These should be eliminated.

get_static_param_values (*trans*)

Returns a map of parameter names and values if the tool does not require any user input. Will raise an exception if any parameter does require input.

handle_input (*trans, incoming, history=None, old_errors=None, process_state='update', source='html'*)

Process incoming parameters for this tool from the dict *incoming*, update the tool state (or create if none existed), and either return to the form or execute the tool (only if 'execute' was clicked and there were no errors).

process_state can be either 'update' (to incrementally build up the state over several calls - one repeat per handle for instance) or 'populate' force a complete build of the state and submission all at once (like from API). May want an incremental version of the API also at some point, that is why this is not just called *for_api*.

handle_interrupted (*trans, inputs*)

Upon handling inputs, if it appears that we have received an incomplete form, do some cleanup or anything else deemed necessary. Currently this is only likely during file uploads, but this method could be generalized and a method standardized for handling other tools.

handle_job_failure_exception (*e*)

Called by *job.fail* when an exception is generated to allow generation of a better error message (returning *None* yields the default behavior)

handle_single_execution (*trans, rerun_remap_job_id, params, history*)

Return a pair with whether execution is successful as well as either resulting output data or an error message indicating the problem.

handle_unvalidated_param_values (*input_values, app*)
Find any instances of *UnvalidatedValue* within *input_values* and validate them (by calling *ToolParameter.from_html* and *ToolParameter.validate*).

handle_unvalidated_param_values_helper (*inputs, input_values, app, context=None, pre-fix=''*)
Recursive helper for *handle_unvalidated_param_values*

installed_tool_dependencies

job_failed (*job_wrapper, message, exception=False*)
Called when a job has failed

new_state (*trans, all_pages=False, history=None*)
Create a new *DefaultToolState* for this tool. It will be initialized with default values for inputs.

Only inputs on the first page will be initialized unless *all_pages* is True, in which case all inputs regardless of page are initialized.

params_from_strings (*params, app, ignore_errors=False*)

params_to_strings (*params, app*)

params_with_missing_data_table_entry
Return all parameters that are dynamically generated select lists whose options require an entry not currently in the *tool_data_table_conf.xml* file.

params_with_missing_index_file
Return all parameters that are dynamically generated select lists whose options refer to a missing .loc file.

parse (*root, guid=None*)
Read tool configuration from the element *root* and fill in *self*.

parse_error_level (*err_level*)
Parses error level and returns error level enumeration. If unparsable, returns 'fatal'

parse_help (*root*)
Parse the help text for the tool. Formatted in reStructuredText, but stored as Mako to allow for dynamic image paths. This implementation supports multiple pages.

parse_input_elem (*parent_elem, encetypes, context=None*)
Parse a parent element whose children are inputs – these could be groups (repeat, conditional) or param elements. Groups will be parsed recursively.

parse_input_page (*input_elem, encetypes*)
Parse a page of inputs. This basically just calls 'parse_input_elem', but it also deals with possible 'display' elements which are supported only at the top/page level (not in groups).

parse_inputs (*root*)
Parse the "<inputs>" element and create appropriate *ToolParameter*'s. This implementation supports multiple pages and grouping constructs.

parse_outputs (*root*)
Parse <outputs> elements and fill in *self.outputs* (keyed by name)

parse_param_elem (*input_elem, encetypes, context*)
Parse a single "<param>" element and return a *ToolParameter* instance. Also, if the parameter has a 'required_encype' add it to the set *encetypes*.

parse_redirect_url (*data, param_dict*)
Parse the REDIRECT_URL tool param. Tools that send data to an external application via a redirect must include the following 3 tool params:

1. REDIRECT_URL - the url to which the data is being sent

2.`DATA_URL` - the url to which the receiving application will send an http post to retrieve the Galaxy data

3.`GALAXY_URL` - the url to which the external application may post data as a response

parse_stdio (*root*)

Parse `<stdio>` element(s) and fill in `self.return_codes`, `self.stderr_rules`, and `self.stdout_rules`. Return codes have a range and an error type (fault or warning). Stderr and stdout rules have a regular expression and an error level (fault or warning).

parse_stdio_exit_codes (*stdio_elem*)

Parse the tool's `<stdio>` element's `<exit_code>` subelements. This will add all of those elements, if any, to `self.stderr_exit_codes`.

parse_stdio_regexes (*stdio_elem*)

Look in the tool's `<stdio>` elem for all `<regex>` subelements that define how to look for warnings and fatal errors in stdout and stderr. This will add all such regex elements to the Tols's `stdio_regexes` list.

populate_state (*trans*, *inputs*, *state*, *incoming*, *history=None*, *source='html'*, *prefix=''*, *context=None*)

populate_tool_shed_info ()

requires_setting_metadata = True

sa_session

Returns a SQLAlchemy session

tests

to_dict (*trans*, *link_details=False*, *io_details=False*)

Returns dict of tool.

tool_shed_repository

tool_type = 'default'

tool_version

Return a ToolVersion if one exists for our id

tool_version_ids

tool_versions

update_state (*trans*, *inputs*, *state*, *incoming*, *source='html'*, *prefix=''*, *context=None*, *update_only=False*, *old_errors={}*, *item_callback=None*)

Update the tool state in *state* using the user input in *incoming*. This is designed to be called recursively: *inputs* contains the set of inputs being processed, and *prefix* specifies a prefix to add to the name of each input to extract its value from *incoming*.

If *update_only* is True, values that are not in *incoming* will not be modified. In this case *old_errors* can be provided, and any errors for parameters which were *not* updated will be preserved.

visit_inputs (*value*, *callback*)

Call the function *callback* on each parameter of this tool. Visits grouping parameters recursively and constructs unique prefixes for each nested set of The callback method is then called as:

callback(level_prefix, parameter, parameter_value)

class `galaxy.tools.ToolBox` (*config_filenames*, *tool_root_dir*, *app*)

Bases: `object`, `galaxy.model.item_attrs.Dictifiable`

Container for a collection of tools

build_tool_version_select_field (*tools, tool_id, set_selected*)
Build a SelectField whose options are the ids for the received list of tools.

fix_integrated_tool_panel_dict ()

get_loaded_tools_by_lineage (*tool_id*)
Get all loaded tools associated by lineage to the tool whose id is *tool_id*.

get_shed_config_dict_by_filename (*filename, default=None*)

get_tool (*tool_id, tool_version=None, get_all_versions=False*)
Attempt to locate a tool in the tool box.

get_tool_components (*tool_id, tool_version=None, get_loaded_tools_by_lineage=False, set_selected=False*)
Retrieve all loaded versions of a tool from the toolbox and return a select list enabling selection of a different version, the list of the tool's loaded versions, and the specified tool.

init_dependency_manager ()

init_tools (*config_filename*)
Read the configuration file and load each tool. The following tags are currently supported:

load_integrated_tool_panel_keys ()
Load the integrated tool panel keys, setting values for tools and workflows to None. The values will be reset when the various tool panel config files are parsed, at which time the tools and workflows are loaded.

load_label_tag_set (*elem, panel_dict, integrated_panel_dict, load_panel_dict, index=None*)

load_section_tag_set (*elem, tool_path, load_panel_dict, index=None*)

load_tool (*config_file, guid=None, repository_id=None, **kwds*)
Load a single tool from the file named by *config_file* and return an instance of *Tool*.

load_tool_panel ()

load_tool_tag_set (*elem, panel_dict, integrated_panel_dict, tool_path, load_panel_dict, guid=None, index=None*)

load_tool_dir_tag_set (*sub_elem, elems, tool_path, integrated_elems, load_panel_dict*)

load_workflow (*workflow_id*)
Return an instance of 'Workflow' identified by *id*, which is encoded in the tool panel.

load_workflow_tag_set (*elem, panel_dict, integrated_panel_dict, load_panel_dict, index=None*)

package_tool (*trans, tool_id*)
Create a tarball with the tool's xml, help images, and test data. :param *trans*: the web transaction
:param *tool_id*: the tool ID from app.toolbox :returns: tuple of tarball filename, success True/False, message/None

reload_tool_by_id (*tool_id*)
Attempt to reload the tool identified by 'tool_id', if successful replace the old tool.

remove_tool_by_id (*tool_id*)
Attempt to remove the tool identified by 'tool_id'.

sa_session
Returns a SQLAlchemy session

to_dict (*trans, in_panel=True, **kwds*)
to_dict toolbox.

write_integrated_tool_panel_config_file()

Write the current in-memory version of the integrated_tool_panel.xml file to disk. Since Galaxy administrators use this file to manage the tool panel, we'll not use xml_to_string() since it doesn't write XML quite right.

exception galaxy.tools.ToolNotFoundException

Bases: exceptions.Exception

class galaxy.tools.ToolOutput (*name, format=None, format_source=None, metadata_source=None, parent=None, label=None, filters=None, actions=None, hidden=False*)

Bases: object, *galaxy.model.item_attrs.Dictifiable*

Represents an output datasets produced by a tool. For backward compatibility this behaves as if it were the tuple:

(format, metadata_source, parent)

dict_collection_visible_keys = ('name', 'format', 'label', 'hidden')

class galaxy.tools.ToolSection (*elem=None*)

Bases: object, *galaxy.model.item_attrs.Dictifiable*

A group of tools with similar type/purpose that will be displayed as a group in the user interface.

copy()

dict_collection_visible_keys = ('id', 'name', 'version')

to_dict (*trans, link_details=False*)

Return a dict that includes section's attributes.

class galaxy.tools.ToolSectionLabel (*elem*)

Bases: object, *galaxy.model.item_attrs.Dictifiable*

A label for a set of tools that can be displayed above groups of tools and sections in the user interface

dict_collection_visible_keys = ('id', 'text', 'version')

class galaxy.tools.ToolStdioExitCode

Bases: object

This is a container for the <stdio> element's <exit_code> subelement. The exit_code element has a range of exit codes and the error level.

class galaxy.tools.ToolStdioRegex

Bases: object

This is a container for the <stdio> element's regex subelement. The regex subelement has a "match" attribute, a "sources" attribute that contains "output" and/or "error", and a "level" attribute that contains "warning" or "fatal".

class galaxy.tools.TracksterConfig (*actions*)

Trackster configuration encapsulation.

static parse (*root*)

galaxy.tools.check_param_from_incoming (*trans, state, input, incoming, key, context, source*)

Unlike "update" state, this preserves default if no incoming value found. This lets API user specify just a subset of params and allow defaults to be used when available.

galaxy.tools.get_incoming_value (*incoming, key, default*)

Fetch value from incoming dict directly or check special nginx upload created variants of this key.

`galaxy.tools.json_fix(val)`

`galaxy.tools.to_dict_helper(obj, kwargs)`

Helper function that provides the appropriate kwargs to `to_dict` an object.

`galaxy.tools.tool_class`

alias of `DataDestinationTool`

exception_handling Module Exceptions and handlers for tools.

FIXME: These are used by tool scripts, not the framework, and should not live in this package.

exception `galaxy.tools.exception_handling.UCSCLimitException`

Bases: `exceptions.Exception`

class `galaxy.tools.exception_handling.UCSCOutWrapper(other)`

Bases: `object`

File-like object that throws an exception if it encounters the UCSC limit error lines

next()

readline()

test Module

class `galaxy.tools.test.ParamContext(name, index=None, parent_context=None)`

Bases: `object`

extract_value(raw_inputs)

for_state()

param_names()

class `galaxy.tools.test.RootParamContext`

Bases: `object`

for_state()

get_index()

param_names()

class `galaxy.tools.test.TestCollectionDef(elem)`

Bases: `object`

collect_inputs()

class `galaxy.tools.test.ToolTestBuilder(tool, test_elem, i, default_interactor)`

Bases: `object`

Encapsulates information about a tool test, and allows creation of a dynamic `TestCase` class (the unittest framework is very class oriented, doing dynamic tests in this way allows better integration)

test_data()

Iterator over metadata representing the required files for upload.

`galaxy.tools.test.expand_input_elems(root_elem, prefix='')`

`galaxy.tools.test.nottest(x)`

`galaxy.tools.test.parse_output_elems(test_elem)`

`galaxy.tools.test.parse_param_elem(param_elem, i=0)`

`galaxy.tools.test.parse_tests_elem(tool, tests_elem)`

Build ToolTestBuilder objects for each “<test>” elements and return default interactor (if any).

`galaxy.tools.test.require_file(name, value, extra, required_files)`

`galaxy.tools.test.test_data_iter(required_files)`

Subpackages

actions Package

actions Package

class `galaxy.tools.actions.DefaultToolAction`

Bases: `object`

Default tool action is to run an external command

collect_input_dataset_collections (*tool, param_values*)

collect_input_datasets (*tool, param_values, trans*)

Collect any dataset inputs from incoming. Returns a mapping from parameter name to Dataset instance for each tool parameter that is of the DataToolParameter type.

execute (*tool, trans, incoming={}, return_job=False, set_output_hid=True, set_output_history=True, history=None, job_params=None, rerun_remap_job_id=None*)

Executes a tool, creating job and tool outputs, associating them, and submitting the job to the job queue. If history is not specified, use trans.history as destination for tool’s output datasets.

get_output_name (*output, dataset, tool, on_text, trans, incoming, history, params, job_params*)

class `galaxy.tools.actions.ObjectStorePopulator` (*app*)

Bases: `object`

Small helper for interacting with the object store and making sure all datasets from a job end up with the same object_store_id.

set_object_store_id (*data*)

class `galaxy.tools.actions.ToolAction`

Bases: `object`

The actions to be taken when a tool is run (after parameters have been converted and validated).

execute (*tool, trans, incoming={}, set_output_hid=True*)

`galaxy.tools.actions.determine_output_format` (*output, parameter_context, input_datasets, random_input_ext*)

Determines the output format for a dataset based on an abstract description of the output (`galaxy.tools.ToolOutput`), the parameter wrappers, a map of the input datasets (name => HDA), and the last input extensions in the tool form.

TODO: Don’t deal with XML here - move this logic into ToolOutput. TODO: Make the input extension used deterministic instead of random.

`galaxy.tools.actions.filter_output` (*output, incoming*)

`galaxy.tools.actions.on_text_for_names` (*input_names*)

history_imp_exp Module**class** galaxy.tools.actions.history_imp_exp.**ExportHistoryToolAction**

Bases: galaxy.tools.actions.__init__.ToolAction

Tool action used for exporting a history to an archive.

execute (tool, trans, incoming={}, set_output_hid=False, overwrite=True, history=None, **kwargs)**class** galaxy.tools.actions.history_imp_exp.**ImportHistoryToolAction**

Bases: galaxy.tools.actions.__init__.ToolAction

Tool action used for importing a history to an archive.

execute (tool, trans, incoming={}, set_output_hid=False, overwrite=True, history=None, **kwargs)**index_genome Module****metadata Module****class** galaxy.tools.actions.metadata.**SetMetadataToolAction**

Bases: galaxy.tools.actions.__init__.ToolAction

Tool action used for setting external metadata on an existing dataset

execute (tool, trans, incoming={}, set_output_hid=False, overwrite=True, history=None, job_params=None, **kwargs)

Execute using a web transaction.

execute_via_app (tool, app, session_id, history_id, user=None, incoming={}, set_output_hid=False, overwrite=True, history=None, job_params=None)

Execute using application.

upload Module**class** galaxy.tools.actions.upload.**UploadToolAction**

Bases: galaxy.tools.actions.__init__.ToolAction

execute (tool, trans, incoming={}, set_output_hid=True, history=None, **kwargs)**upload_common Module**galaxy.tools.actions.upload_common.**active_folders** (trans, folder)galaxy.tools.actions.upload_common.**cleanup_unused_precreated_datasets** (precreated_datasets)galaxy.tools.actions.upload_common.**create_job** (trans, params, tool, json_file_path, data_list, folder=None, history=None)

Create the upload job.

galaxy.tools.actions.upload_common.**create_paramfile** (trans, uploaded_datasets)

Create the upload tool's JSON "param" file.

galaxy.tools.actions.upload_common.**get_precreated_dataset** (precreated_datasets, name)

Return a dataset matching a name from the list of precreated (via async upload) datasets. If there's more than one upload with the exact same name, we need to pop one (the first) so it isn't chosen next time.

galaxy.tools.actions.upload_common.**get_precreated_datasets** (trans, params, data_obj, controller='root')

Get any precreated datasets (when using asynchronous uploads).


```
galaxy.tools.actions.upload_common.get_uploaded_datasets(trans, cntrlr, params,
                                                         precreated_datasets,
                                                         dataset_upload_inputs,
                                                         library_bunch=None,
                                                         history=None)

galaxy.tools.actions.upload_common.handle_library_params(trans, params, folder_id,
                                                         replace_dataset=None)

galaxy.tools.actions.upload_common.new_upload(trans, cntrlr, uploaded_dataset, li-
                                              brary_bunch=None,      history=None,
                                              state=None)

galaxy.tools.actions.upload_common.persist_uploads(params)
    Turn any uploads in the submitted form to persisted files.
```

data Package

data Package Manage tool data tables, which store (at the application level) data that is used by tools, for example in the generation of dynamic options. Tables are loaded and stored by names which tools use to refer to them. This allows users to configure data tables for a local Galaxy instance without needing to modify the tool configurations.

```
class galaxy.tools.data.TabularToolDataTable(config_element,          tool_data_path,
                                              from_shed_config=False, filename=None)
    Bases: galaxy.tools.data.ToolDataTable, galaxy.model.item_attrs.Dictifiable
```

Data stored in a tabular / separated value format on disk, allows multiple files to be merged but all must have the same column definitions:

```
<table type="tabular" name="test">
  <column name='...' index = '...' />
  <file path="..." />
  <file path="..." />
</table>
```

```
configure_and_load(config_element, tool_data_path, from_shed_config=False)
    Configure and load table from an XML element.
```

```
dict_collection_visible_keys = ['name']
```

```
extend_data_with(filename, errors=None)
```

```
get_column_name_list()
```

```
get_entry(query_attr, query_val, return_attr, default=None)
    Returns table entry associated with a col/val pair.
```

```
get_fields()
```

```
get_named_fields_list()
```

```
get_version_fields()
```

```
handle_found_index_file(filename)
```

```
merge_tool_data_table(other_table,      allow_duplicates=True,      persist=False,      per-
                      sist_on_error=False, entry_source=None, **kwd)
```

```
parse_column_spec(config_element)
```

Parse column definitions, which can either be a set of 'column' elements with a name and index (as in

dynamic options config), or a shorthand comma separated list of names in order as the text of a 'column_names' element.

A column named 'value' is required.

parse_file_fields (*reader, errors=None, here='__HERE__'*)

Parse separated lines from file and return a list of tuples.

TODO: Allow named access to fields using the column names.

to_dict (*view='collection'*)

type_key = 'tabular'

xml_string

```
class galaxy.tools.data.ToolDataTable(config_element, tool_data_path,
                                     from_shed_config=False, filename=None)
```

Bases: object

add_entries (*entries, allow_duplicates=True, persist=False, persist_on_error=False, entry_source=None, **kwd*)

add_entry (*entry, allow_duplicates=True, persist=False, persist_on_error=False, entry_source=None, **kwd*)

classmethod from_elem (*table_elem, tool_data_path, from_shed_config, filename*)

get_empty_field_by_name (*name*)

is_current_version (*other_version*)

merge_tool_data_table (*other_table, allow_duplicates=True, persist=False, persist_on_error=False, entry_source=None, **kwd*)

reload_from_files ()

```
class galaxy.tools.data.ToolDataTableManager(tool_data_path, config_filename=None)
```

Bases: object

Manages a collection of tool data tables

add_new_entries_from_config_file (*config_filename, tool_data_path, shed_tool_data_table_config, persist=False*)

This method is called when a tool shed repository that includes a tool_data_table_conf.xml.sample file is being installed into a local galaxy instance. We have 2 cases to handle, files whose root tag is <tables>, for example:

```
<tables>
  <!-- Location of Tmap files -->
  <table name="tmap_indexes" comment_char="#">
    <columns>value, dbkey, name, path</columns>
    <file path="tool-data/tmap_index.loc" />
  </table>
</tables>
```

and files whose root tag is <table>, for example:

```
<!-- Location of Tmap files -->
<table name="tmap_indexes" comment_char="#">
  <columns>value, dbkey, name, path</columns>
  <file path="tool-data/tmap_index.loc" />
</table>
```

get (*name, default=None*)

get_tables()

load_from_config_file(*config_filename, tool_data_path, from_shed_config=False*)

This method is called under 3 conditions:

1. When the ToolDataTableManager is initialized (see `__init__` above).
2. Just after the ToolDataTableManager is initialized and the additional entries defined by `shed_tool_data_table_conf.xml` are being loaded into the ToolDataTableManager.data_tables.
3. When a tool shed repository that includes a `tool_data_table_conf.xml.sample` file is being installed into a local Galaxy instance. In this case, we have 2 entry types to handle, files whose root tag is `<tables>`, for example:

reload_tables(*table_names=None*)

set(*name, value*)

to_xml_file(*shed_tool_data_table_config, new_elems=None, remove_elems=None*)

Write the current in-memory version of the `shed_tool_data_table_conf.xml` file to disk. `remove_elems` are removed before `new_elems` are added.

`galaxy.tools.data.cls`

alias of *TabularToolDataTable*

`galaxy.tools.data.expand_here_template`(*content, here=None*)

deps Package

deps Package Dependency management for tools.

class `galaxy.tools.deps.DependencyManager`(*default_base_path, conf_file=None*)

Bases: `object`

A DependencyManager attempts to resolve named and versioned dependencies by searching for them under a list of directories. Directories should be of the form:

`$BASE/name/version/...`

and should each contain a file `'env.sh'` which can be sourced to make the dependency available in the current shell environment.

dependency_shell_commands(*requirements, **kwds*)

find_dep(*name, version=None, type='package', **kwds*)

uses_tool_shed_dependencies()

class `galaxy.tools.deps.NullDependencyManager`

Bases: `object`

dependency_shell_commands(*requirements, **kwds*)

find_dep(*name, version=None, type='package', **kwds*)

uses_tool_shed_dependencies()

`galaxy.tools.deps.build_dependency_manager`(*config*)

tests Module

genome_index Package

genome_index Package**index_genome Module****imp_exp Package****imp_exp Package**

```
class galaxy.tools.imp_exp.JobExportHistoryArchiveWrapper (job_id)
    Bases:
        object,
        galaxy.web.base.controller.UsesHistoryMixin,
        galaxy.model.item_attrs.UsesAnnotations
```

Class provides support for performing jobs that export a history to an archive.

```
cleanup_after_job (db_session)
```

Remove temporary directory and attribute files generated during setup for this job.

```
setup_job (trans, jeha, include_hidden=False, include_deleted=False)
```

Perform setup for job to export a history into an archive. Method generates attribute files for export, sets the corresponding attributes in the jeha object, and returns a command line for running the job. The command line includes the command, inputs, and options; it does not include the output file because it must be set at runtime.

```
class galaxy.tools.imp_exp.JobImportHistoryArchiveWrapper (app, job_id)
```

```
    Bases:
        object,
        galaxy.web.base.controller.UsesHistoryMixin,
        galaxy.model.item_attrs.UsesAnnotations
```

Class provides support for performing jobs that import a history from an archive.

```
cleanup_after_job ()
```

Set history, datasets, and jobs' attributes and clean up archive directory.

```
galaxy.tools.imp_exp.load_history_imp_exp_tools (toolbox)
```

Adds tools for importing/exporting histories to archives.

export_history Module Export a history to an archive file using attribute files.

usage: *%prog* history_attrs dataset_attrs job_attrs out_file -G, -gzip: gzip archive file

```
galaxy.tools.imp_exp.export_history.create_archive (history_attrs_file,
                                                    datasets_attrs_file, jobs_attrs_file,
                                                    out_file, gzip=False)
```

Create archive from the given attribute/metadata files and save it to out_file.

```
galaxy.tools.imp_exp.export_history.get_dataset_filename (name, ext)
```

Builds a filename for a dataset using its name and extension.

unpack_tar_gz_archive Module Unpack a tar or tar.gz archive into a directory.

usage: *%prog* archive_source dest_dir *[-urlfile]* source type, either a URL or a file.

```
galaxy.tools.imp_exp.unpack_tar_gz_archive.unpack_archive (archive_file, dest_dir)
    Unpack a tar and/or gzipped archive into a destination directory.
```

```
galaxy.tools.imp_exp.unpack_tar_gz_archive.url_to_file (url, dest_file)
```

Transfer a file from a remote URL to a temporary file.

parameters Package

parameters Package Classes encapsulating Galaxy tool parameters.

```
galaxy.tools.parameters.check_param(trans, param, incoming_value, param_values,
                                     source='html')
```

Check the value of a single parameter *param*. The value in *incoming_value* is converted from its HTML encoding and validated. The *param_values* argument contains the processed values of previous parameters (this may actually be an ExpressionContext when dealing with grouping scenarios).

```
galaxy.tools.parameters.params_from_strings(params, param_values, app, ignore_errors=False)
```

Convert a dictionary of strings as produced by *params_to_strings* back into parameter values (decode the json representation and then allow each parameter to convert the basic types into the parameters preferred form).

```
galaxy.tools.parameters.params_to_incoming(incoming, inputs, input_values, app,
                                             name_prefix='')
```

Given a tool's parameter definition (*inputs*) and a specific set of parameter *input_values* objects, populate *incoming* with the html values.

Useful for e.g. the rerun function.

```
galaxy.tools.parameters.params_to_strings(params, param_values, app)
```

Convert a dictionary of parameter values to a dictionary of strings suitable for persisting. The *value_to_basic* method of each parameter is called to convert its value to basic types, the result of which is then json encoded (this allowing complex nested parameters and such).

```
galaxy.tools.parameters.visit_input_values(inputs, input_values, callback,
                                             name_prefix='', label_prefix='')
```

Given a tools parameter definition (*inputs*) and a specific set of parameter *values*, call *callback* for each non-grouping parameter, passing the parameter object, value, a constructed unique name, and a display label.

If the callback returns a value, it will be replace the old value.

FIXME: There is redundancy between this and the visit_inputs methods of Repeat and Group. This tracks labels and those do not. It would be nice to unify all the places that recursively visit inputs.

basic Module Basic tool parameters.

```
class galaxy.tools.parameters.basic.BaseDataToolParameter(tool, elem, trans)
```

Bases: *galaxy.tools.parameters.basic.ToolParameter*

```
class galaxy.tools.parameters.basic.BaseURLToolParameter(tool, elem)
```

Bases: *galaxy.tools.parameters.basic.ToolParameter*

Returns a parameter the contains its value prepended by the current server base url. Used in all redirects.

```
get_html_field(trans=None, value=None, other_values={})
```

```
get_initial_value(trans, context, history=None)
```

```
get_label()
```

```
get_value(trans)
```

```
to_dict(trans, view='collection', value_mapper=None)
```

```
class galaxy.tools.parameters.basic.BooleanToolParameter(tool, elem)
```

Bases: *galaxy.tools.parameters.basic.ToolParameter*

Parameter that takes one of two values.

```
>>> p = BooleanToolParameter( None, XML( '<param name="blah" type="boolean" checked="yes" trueva
>>> print p.name
blah
>>> print p.get_html()
```

```

<input type="checkbox" id="blah" name="blah" value="true" checked="checked"><input type="hidden"
>>> print p.from_html( ["true","true"] )
True
>>> print p.to_param_dict_string( True )
bulletproof vests
>>> print p.from_html( ["true"] )
False
>>> print p.to_param_dict_string( False )
cellophane chests

```

```

from_html (value, trans=None, other_values={})
from_json (value, trans=None, other_values={})
get_html_field (trans=None, value=None, other_values={})
get_initial_value (trans, context, history=None)
legal_values
to_dict (trans, view='collection', value_mapper=None)
to_html_value (value, app)
to_param_dict_string (value, other_values={})
to_python (value, app)

```

class galaxy.tools.parameters.basic.**ColumnListParameter** (tool, elem)

Bases: *galaxy.tools.parameters.basic.SelectToolParameter*

Select list that consists of either the total number of columns or only those columns that contain numerical values in the associated DataToolParameter.

TODO: we need better testing here, but not sure how to associate a DataToolParameter with a ColumnListParameter # from a twill perspective...

```

>>> # Mock up a history (not connected to database)
>>> from galaxy.model import History, HistoryDatasetAssociation
>>> from galaxy.util.bunch import Bunch
>>> from galaxy.model.mapping import init
>>> sa_session = init( "/tmp", "sqlite:///memory:", create_tables=True ).session
>>> hist = History()
>>> sa_session.add( hist )
>>> sa_session.flush()
>>> hda = hist.add_dataset( HistoryDatasetAssociation( id=1, extension='interval', create_dataset=True ) )
>>> dtp = DataToolParameter( None, XML( '<param name="blah" type="data" format="interval"/>' ) )
>>> print dtp.name
blah
>>> clp = ColumnListParameter ( None, XML( '<param name="numerical_column" type="data_column" data_format="interval"/>' ) )
>>> print clp.name
numerical_column

```

```

from_html (value, trans=None, context={})
    Label convention prepends column number with a 'c', but tool uses the integer. This removes the 'c' when
    entered into a workflow.
get_column_list (trans, other_values)
    Generate a select list containing the columns of the associated dataset (if found).
get_dependencies ()

```

```

    get_initial_value (trans, context, history=None)
    get_legal_values (trans, other_values)
    get_options (trans, other_values)
        show column labels rather than cl.cn if use_header_names=True
    need_late_validation (trans, context)
    to_dict (trans, view='collection', value_mapper=None, other_values={})
galaxy.tools.parameters.basic.DEFAULT_VALUE_MAP (x)
class galaxy.tools.parameters.basic.DataCollectionToolParameter (tool, elem,
    trans=None)
    Bases: galaxy.tools.parameters.basic.BaseDataToolParameter
    collection_type
    from_html (value, trans, other_values={})
    get_html_field (trans=None, value=None, other_values={})
    match_collections (trans, history, dataset_matcher)
    match_multirun_collections (trans, history, dataset_matcher)
    to_dict (trans, view='collection', value_mapper=None, other_values=None)
    to_python (value, app)
    to_string (value, app)
    validate (value, history=None)
    value_to_display_text (value, app)
class galaxy.tools.parameters.basic.DataToolParameter (tool, elem, trans=None)
    Bases: galaxy.tools.parameters.basic.BaseDataToolParameter
    Parameter that takes on one (or many) or a specific set of values.
    TODO: There should be an alternate display that allows single selects to be displayed as radio buttons and
        multiple selects as a set of checkboxes
    TODO: The following must be fixed to test correctly for the new security_check tag in the DataToolParameter (
        the last test below is broken ) Nate's next pass at the dataset security stuff will dramatically alter this anyway.
    converter_safe (other_values, trans)
    from_html (value, trans, other_values={})
    get_dependencies ()
        Get the names of the other params this param depends on.
    get_html_field (trans=None, value=None, other_values={})
    get_initial_value (trans, context, history=None)
    get_initial_value_from_history_prevent_repeats (trans, context, already_used, his-
        tory=None)
        NOTE: This is wasteful since dynamic options and dataset collection happens twice (here and when
            generating HTML).
    match_collections (history, dataset_matcher, reduction=True)
    match_datasets (history, dataset_matcher)
    need_late_validation (trans, context)

```

`to_dict` (*trans*, *view*='collection', *value_mapper*=None, *other_values*=None)

`to_param_dict_string` (*value*, *other_values*={})

`to_python` (*value*, *app*)

`to_string` (*value*, *app*)

`validate` (*value*, *history*=None)

`value_to_display_text` (*value*, *app*)

class `galaxy.tools.parameters.basic.DrillDownSelectToolParameter` (*tool*, *elem*, *context*=None)

Bases: `galaxy.tools.parameters.basic.SelectToolParameter`

Parameter that takes on one (or many) of a specific set of values. Creating a hierarchical select menu, which allows users to ‘drill down’ a tree-like set of options.

```
>>> p = DrillDownSelectToolParameter( None, XML(
...     '''
...     <param name="some_name" type="drill_down" display="checkbox" hierarchy="recurse" multiple="t
...     <options>
...     <option name="Heading 1" value="heading1">
...         <option name="Option 1" value="option1"/>
...         <option name="Option 2" value="option2"/>
...         <option name="Heading 1" value="heading1">
...             <option name="Option 3" value="option3"/>
...             <option name="Option 4" value="option4"/>
...         </option>
...     </option>
...     <option name="Option 5" value="option5"/>
... </options>
... </param>
...     ''' ) )
>>> print p.get_html()
<div class="form-row drilldown-container" id="drilldown--736f6d655f6e616d65">
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--736f6d655f6e616d65-68656"
<input type="checkbox" name="some_name" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--736f6d655f6e616d65-68656164696e6731-container" style=
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option1" >Option 1
</div>
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option2" >Option 2
</div>
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--736f6d655f6e616d65-68656"
<input type="checkbox" name="some_name" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--736f6d655f6e616d65-68656164696e6731-68656164696e6731-
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option3" >Option 3
</div>
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option4" >Option 4
</div>
</div>
</div>
</div>
</div>
```



```

<div class="form-row-input">
<input type="checkbox" name="some_name" value="option5" >Option 5
</div>
</div>
>>> p = DrillDownSelectToolParameter( None, XML(
... '''
... <param name="some_name" type="drill_down" display="radio" hierarchy="recurse" multiple="false"
... <options>
... <option name="Heading 1" value="heading1">
... <option name="Option 1" value="option1"/>
... <option name="Option 2" value="option2"/>
... <option name="Heading 1" value="heading1">
... <option name="Option 3" value="option3"/>
... <option name="Option 4" value="option4"/>
... </option>
... </option>
... <option name="Option 5" value="option5"/>
... </options>
... </param>
... ''' ) )
>>> print p.get_html()
<div class="form-row drilldown-container" id="drilldown--736f6d655f6e616d65">
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--736f6d655f6e616d65-68656"
<input type="radio" name="some_name" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--736f6d655f6e616d65-68656164696e6731-container" style=
<div class="form-row-input">
<input type="radio" name="some_name" value="option1" >Option 1
</div>
<div class="form-row-input">
<input type="radio" name="some_name" value="option2" >Option 2
</div>
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--736f6d655f6e616d65-68656"
<input type="radio" name="some_name" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--736f6d655f6e616d65-68656164696e6731-68656164696e6731-
<div class="form-row-input">
<input type="radio" name="some_name" value="option3" >Option 3
</div>
<div class="form-row-input">
<input type="radio" name="some_name" value="option4" >Option 4
</div>
</div>
</div>
</div>
</div>
<div class="form-row-input">
<input type="radio" name="some_name" value="option5" >Option 5
</div>
</div>
>>> print p.options
[{'selected': False, 'name': 'Heading 1', 'value': 'heading1', 'options': [{'selected': False, '

```

from_html (value, trans=None, other_values={})

get_dependencies ()

Get the *names* of the other params this param depends on.

get_html (trans=None, value=None, other_values={})

Returns the html widget corresponding to the paramter. Optionally attempt to retain the current value specific by 'value'

```

get_html_field (trans=None, value=None, other_values={})
get_initial_value (trans, context, history=None)
get_legal_values (trans, other_values)
get_options (trans=None, value=None, other_values={})
to_dict (trans, view='collection', value_mapper=None, other_values={})
to_param_dict_string (value, other_values={}, value_map=<function <lambda>>)
value_to_display_text (value, app)

```

```

class galaxy.tools.parameters.basic.DummyDataset
    Bases: object

```

```

class galaxy.tools.parameters.basic.FTPFileToolParameter (tool, elem)
    Bases: galaxy.tools.parameters.basic.ToolParameter

```

Parameter that takes a file uploaded via FTP as a value.

```

from_html (value, trans=None, other_values={})
get_html_field (trans=None, value=None, other_values={})
get_initial_value (trans, context, history=None)
to_python (value, app)
to_string (value, app)
visible

```

```

class galaxy.tools.parameters.basic.FileToolParameter (tool, elem)
    Bases: galaxy.tools.parameters.basic.ToolParameter

```

Parameter that takes an uploaded file as a value.

```

>>> p = FileToolParameter( None, XML( '<param name="blah" type="file"/>' ) )
>>> print p.name
blah
>>> print p.get_html()
<input type="file" name="blah">
>>> p = FileToolParameter( None, XML( '<param name="blah" type="file" ajax-upload="true"/>' ) )
>>> print p.get_html()
<input type="file" name="blah" galaxy-ajax-upload="true">

```

```

from_html (value, trans=None, other_values={})
get_html_field (trans=None, value=None, other_values={})
get_initial_value (trans, context, history=None)
get_required enctype ()
    File upload elements require the multipart/form-data encoding
to_python (value, app)
to_string (value, app)

```

class galaxy.tools.parameters.basic.**FloatToolParameter**(*tool, elem*)

Bases: *galaxy.tools.parameters.basic.TextToolParameter*

Parameter that takes a real number value.

```
>>> p = FloatToolParameter( None, XML( '<param name="blah" type="float" size="4" value="3.141592'
>>> print p.name
blah
>>> print p.get_html()
<input type="text" name="blah" size="4" value="3.141592">
>>> type( p.from_html( "36.1" ) )
<type 'float'>
>>> type( p.from_html( "bleh" ) )
Traceback (most recent call last):
...
ValueError: A real number is required
```

dict_collection_visible_keys = ('name', 'type', 'label', 'help', 'min', 'max')

from_html (*value, trans=None, other_values={}*)

get_html_field (*trans=None, value=None, other_values={}*)

get_initial_value (*trans, context, history=None*)

to_python (*value, app*)

to_string (*value, app*)

Convert a value to a string representation suitable for persisting

class galaxy.tools.parameters.basic.**GenomeBuildParameter**(**args, **kwargs*)

Bases: *galaxy.tools.parameters.basic.SelectToolParameter*

Select list that sets the last used genome build for the current history as “selected”.

```
>>> # Create a mock transaction with 'hg17' as the current build
>>> from galaxy.util.bunch import Bunch
>>> trans = Bunch( history=Bunch( genome_build='hg17' ), db_builds=util.read_dbnames( None ) )
```

```
>>> p = GenomeBuildParameter( None, XML(
... '''
... <param name="blah" type="genomebuild" />
... ''' ) )
>>> print p.name
blah
```

```
>>> # hg17 should be selected by default
>>> print p.get_html( trans )
<select name="blah" last_selected_value="hg17">
<option value="?">unspecified (?)</option>
...
<option value="hg18">Human Mar. 2006 (NCBI36/hg18) (hg18)</option>
<option value="hg17" selected>Human May 2004 (NCBI35/hg17) (hg17)</option>
...
</select>
```

```
>>> # If the user selected something else already, that should be used
>>> # instead
>>> print p.get_html( trans, value='hg18' )
<select name="blah" last_selected_value="hg18">
<option value="?">unspecified (?)</option>
...
<option value="hg18" selected>Human Mar. 2006 (NCBI36/hg18) (hg18)</option>
<option value="hg17">Human May 2004 (NCBI35/hg17) (hg17)</option>
...
</select>
```

```
>>> print p.filter_value( "hg17" )
hg17
```

get_legal_values (trans, other_values)

get_options (trans, other_values)

to_dict (trans, view='collection', value_mapper=None)

class galaxy.tools.parameters.basic.**HiddenDataToolParameter** (tool, elem)

Bases: *galaxy.tools.parameters.basic.HiddenToolParameter*,
galaxy.tools.parameters.basic.DataToolParameter

Hidden parameter that behaves as a DataToolParameter. As with all hidden parameters, this is a HACK.

get_html_field (trans=None, value=None, other_values={})

get_initial_value (trans, context, history=None)

class galaxy.tools.parameters.basic.**HiddenToolParameter** (tool, elem)

Bases: *galaxy.tools.parameters.basic.ToolParameter*

Parameter that takes one of two values.

FIXME: This seems hacky, parameters should only describe things the user might change. It is used for 'initializing' the UCSC proxy tool

```
>>> p = HiddenToolParameter( None, XML( '<param name="blah" type="hidden" value="wax so rockin"/>' ) )
>>> print p.name
blah
>>> print p.get_html()
<input type="hidden" name="blah" value="wax so rockin">
```

get_html_field (trans=None, value=None, other_values={})

get_initial_value (trans, context, history=None)

get_label ()

class galaxy.tools.parameters.basic.**IntegerToolParameter** (tool, elem)

Bases: *galaxy.tools.parameters.basic.TextToolParameter*

Parameter that takes an integer value.

```
>>> p = IntegerToolParameter( None, XML( '<param name="blah" type="integer" size="4" value="10"/>' ) )
>>> print p.name
blah
>>> print p.get_html()
<input type="text" name="blah" size="4" value="10">
>>> type( p.from_html( "10" ) )
```

```
<type 'int'>
>>> type( p.from_html( "bleh" ) )
Traceback (most recent call last):
...
ValueError: An integer is required
```

dict_collection_visible_keys = ('name', 'type', 'label', 'help', 'min', 'max')

from_html (value, trans=None, other_values={})

get_html_field (trans=None, value=None, other_values={})

get_initial_value (trans, context, history=None)

to_python (value, app)

to_string (value, app)

Convert a value to a string representation suitable for persisting

class galaxy.tools.parameters.basic.**LibraryDatasetToolParameter** (tool, elem)
Bases: *galaxy.tools.parameters.basic.ToolParameter*

Parameter that lets users select a LDDA from a modal window, then use it within the wrapper.

from_html (value, trans, other_values={})

get_html_field (trans=None, value=None, other_values={})

get_initial_value (trans, context, history=None)

to_python (value, app)

to_string (value, app)

class galaxy.tools.parameters.basic.**RuntimeValue**
Bases: object

Wrapper to note a value that is not yet set, but will be required at runtime.

class galaxy.tools.parameters.basic.**SelectToolParameter** (tool, elem, context=None)
Bases: *galaxy.tools.parameters.basic.ToolParameter*

Parameter that takes on one (or many) or a specific set of values.

```
>>> p = SelectToolParameter( None, XML(
...     '''
...     <param name="blah" type="select">
...         <option value="x">I am X</option>
...         <option value="y" selected="true">I am Y</option>
...         <option value="z">I am Z</option>
...     </param>
...     ''' ) )
>>> print p.name
blah
>>> print p.get_html()
<select name="blah" last_selected_value="y">
<option value="x">I am X</option>
<option value="y" selected>I am Y</option>
<option value="z">I am Z</option>
</select>
>>> print p.get_html( value="z" )
<select name="blah" last_selected_value="z">
<option value="x">I am X</option>
```

```

<option value="y">I am Y</option>
<option value="z" selected>I am Z</option>
</select>
>>> print p.filter_value( "y" )
y

```

```

>>> p = SelectToolParameter( None, XML(
...     '''
...     <param name="blah" type="select" multiple="true">
...         <option value="x">I am X</option>
...         <option value="y" selected="true">I am Y</option>
...         <option value="z" selected="true">I am Z</option>
...     </param>
...     ''' ) )
>>> print p.name
blah
>>> print p.get_html()
<select name="blah" multiple last_selected_value="z">
<option value="x">I am X</option>
<option value="y" selected>I am Y</option>
<option value="z" selected>I am Z</option>
</select>
>>> print p.get_html( value=["x","y"])
<select name="blah" multiple last_selected_value="y">
<option value="x" selected>I am X</option>
<option value="y" selected>I am Y</option>
<option value="z">I am Z</option>
</select>
>>> print p.to_param_dict_string( ["y", "z"] )
y,z

```

```

>>> p = SelectToolParameter( None, XML(
...     '''
...     <param name="blah" type="select" multiple="true" display="checkboxes">
...         <option value="x">I am X</option>
...         <option value="y" selected="true">I am Y</option>
...         <option value="z" selected="true">I am Z</option>
...     </param>
...     ''' ) )
>>> print p.name
blah
>>> print p.get_html()
<div class="checkUncheckAllPlaceholder" checkbox_name="blah"></div>
<div><input type="checkbox" name="blah" value="x" id="blah|x"><label class="inline" for="blah|x">
<div class="odd_row"><input type="checkbox" name="blah" value="y" id="blah|y" checked='checked'>
<div><input type="checkbox" name="blah" value="z" id="blah|z" checked='checked'><label class="inline" for="blah|z">
>>> print p.get_html( value=["x","y"])
<div class="checkUncheckAllPlaceholder" checkbox_name="blah"></div>
<div><input type="checkbox" name="blah" value="x" id="blah|x" checked='checked'><label class="inline" for="blah|x">
<div class="odd_row"><input type="checkbox" name="blah" value="y" id="blah|y" checked='checked'>
<div><input type="checkbox" name="blah" value="z" id="blah|z"><label class="inline" for="blah|z">
>>> print p.to_param_dict_string( ["y", "z"] )
y,z

```

from_html (value, trans=None, context={})

get_dependencies ()

Get the *names* of the other params this param depends on.

get_html_field (*trans=None, value=None, context={}*)

get_initial_value (*trans, context, history=None*)

get_legal_values (*trans, other_values*)

get_options (*trans, other_values*)

need_late_validation (*trans, context*)

Determine whether we need to wait to validate this parameters value given the current state. For parameters with static options this is always false (can always validate immediately). For parameters with dynamic options, we need to check whether the other parameters which determine what options are valid have been set. For the old style dynamic options which do not specify dependencies, this is always true (must validate at runtime).

to_dict (*trans, view='collection', value_mapper=None, other_values={}*)

to_html_value (*value, app*)

to_param_dict_string (*value, other_values={}, value_map=<function <lambda>>*)

value_from_basic (*value, app, ignore_errors=False*)

value_to_basic (*value, app*)

value_to_display_text (*value, app*)

class galaxy.tools.parameters.basic.**TextToolParameter** (*tool, elem*)

Bases: *galaxy.tools.parameters.basic.ToolParameter*

Parameter that can take on any text value.

```
>>> p = TextToolParameter( None, XML( '<param name="blah" type="text" size="4" value="default" />' ))
>>> print p.name
blah
>>> print p.get_html()
<input type="text" name="blah" size="4" value="default">
>>> print p.get_html( value="meh" )
<input type="text" name="blah" size="4" value="meh">
```

get_html_field (*trans=None, value=None, other_values={}*)

get_initial_value (*trans, context, history=None*)

to_dict (*trans, view='collection', value_mapper=None*)

class galaxy.tools.parameters.basic.**ToolParameter** (*tool, param, context=None*)

Bases: object, *galaxy.model.item_attrs.Dictifiable*

Describes a parameter accepted by a tool. This is just a simple stub at the moment but in the future should encapsulate more complex parameters (lists of valid choices, validation logic, ...)

classmethod **build** (*tool, param*)

Factory method to create parameter of correct type

dict_collection_visible_keys = ('name', 'type', 'label', 'help')

filter_value (*value, trans=None, other_values={}*)

Parse the value returned by the view into a form usable by the tool OR raise a ValueError.

from_html (*value, trans=None, other_values={}*)

Convert a value from an HTML POST into the parameters preferred value format.

from_json (*value*, *trans=None*, *other_values={}*)

get_dependencies ()

Return the names of any other parameters this parameter depends on

get_html (*trans=None*, *value=None*, *other_values={}*)

Returns the html widget corresponding to the parameter. Optionally attempt to retain the current value specific by 'value'

get_html_field (*trans=None*, *value=None*, *other_values={}*)

get_initial_value (*trans*, *context*, *history=None*)

Return the starting value of the parameter

get_initial_value_from_history_prevent_repeats (*trans*, *context*, *already_used*, *history=None*)

Get the starting value for the parameter, but if fetching from the history, try to find a value that has not yet been used. *already_used* is a list of objects that tools must manipulate (by adding to it) to store a memento that they can use to detect if a value has already been chosen from the history. This is to support the capability to choose each dataset once

get_label ()

Return user friendly name for the parameter

get_required_ctype ()

If this parameter needs the form to have a specific encoding return it, otherwise return None (indicating compatibility with any encoding)

need_late_validation (*trans*, *context*)

to_dict (*trans*, *view='collection'*, *value_mapper=None*)

to_dict tool parameter. This can be overridden by subclasses.

to_html_value (*value*, *app*)

Convert an object value to the value expected from an html post

to_param_dict_string (*value*, *other_values={}*)

Called via `__str__` when used in the Cheetah template

to_python (*value*, *app*)

Convert a value created with `to_string` back to an object representation

to_string (*value*, *app*)

Convert a value to a string representation suitable for persisting

validate (*value*, *history=None*)

value_from_basic (*value*, *app*, *ignore_errors=False*)

value_to_basic (*value*, *app*)

value_to_display_text (*value*, *app*)

Convert a value to a text representation suitable for displaying to the user

visible

Return true if the parameter should be rendered on the form

class galaxy.tools.parameters.basic.UnvalidatedValue (*value*)

Bases: object

Wrapper to mark a value that has not been validated

dynamic_options Module Support for generating the options for a SelectToolParameter dynamically (based on the values of other parameters or other aspects of the current state)

class `galaxy.tools.parameters.dynamic_options.AdditionalValueFilter` (*d_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Adds a single static value to an options list.

Type: `add_value`

Required Attributes: `value`: value to appear in select list

Optional Attributes: `name`: Display name to appear in select list (value) `index`: Index of option list to add value (APPEND)

filter_options (*options, trans, other_values*)

class `galaxy.tools.parameters.dynamic_options.AttributeValueSplitterFilter` (*d_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Filters a list of attribute-value pairs to be unique attribute names.

Type: `attribute_value_splitter`

Required Attributes: `column`: column in options to compare with

Optional Attributes: `pair_separator`: Split column by this (,) `name_val_separator`: Split name-value pair by this (whitespace)

filter_options (*options, trans, other_values*)

class `galaxy.tools.parameters.dynamic_options.DataMetaFilter` (*d_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Filters a list of options on a column by a dataset metadata value.

Type: `data_meta`

When no 'from' source has been specified in the <options> tag, this will populate the options list with (meta_value, meta_value, False). Otherwise, options which do not match the metadata value in the column are discarded.

Required Attributes:

- `ref`: Name of input dataset
- `key`: Metadata key to use for comparison
- `column`: column in options to compare with (not required when not associated with input options)

Optional Attributes:

- `multiple`: Option values are multiple, split column by separator (True)
- `separator`: When multiple split by this (,)

filter_options (*options, trans, other_values*)

get_dependency_name ()

class `galaxy.tools.parameters.dynamic_options.DynamicOptions` (*elem, tool_param*)

Bases: `object`

Handles dynamically generated SelectToolParameter options

column_spec_to_index (*column_spec*)

Convert a column specification (as read from the config file), to an index. A column specification can just be a number, a column name, or a column alias.

get_dependency_names ()

Return the names of parameters these options depend on – both data and other param types.

get_field_by_name_for_value (*field_name, value, trans, other_values*)

Get contents of field by name for specified value.

get_fields (*trans, other_values*)

get_fields_by_value (*value, trans, other_values*)

Return a list of fields with column 'value' matching provided value.

get_options (*trans, other_values*)

parse_column_definitions (*elem*)

parse_file_fields (*reader*)

class `galaxy.tools.parameters.dynamic_options.Filter` (*d_option, elem*)

Bases: `object`

A filter takes the current options list and modifies it.

filter_options (*options, trans, other_values*)

Returns a list of options after the filter is applied

classmethod from_element (*d_option, elem*)

Loads the proper filter by the type attribute of elem

get_dependency_name ()

Returns the name of any dependencies, otherwise None

class `galaxy.tools.parameters.dynamic_options.MultipleSplitterFilter` (*d_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Turns a single line of options into multiple lines, by splitting a column and creating a line for each item.

Type: `multiple_splitter`

Required Attributes: column: column in options to compare with

Optional Attributes: separator: Split column by this (,)

filter_options (*options, trans, other_values*)

class `galaxy.tools.parameters.dynamic_options.ParamValueFilter` (*d_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Filters a list of options on a column by the value of another input.

Type: `param_value`

Required Attributes:

•**ref:** Name of input value

•**column:** column in options to compare with

Optional Attributes:

•**keep: Keep columns matching value (True)** Discard columns matching value (False)

•**ref_attribute:** Period (.) separated attribute chain of input (ref) to use as value for filter

filter_options (*options, trans, other_values*)

get_dependency_name ()

class `galaxy.tools.parameters.dynamic_options.RemoveValueFilter` (*d_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Removes a value from an options list.

Type: `remove_value`

Required Attributes:

value: value to remove from select list

or

ref: param to refer to

```

    or
    meta_ref: dataset to refer to
    key: metadata key to compare to

```

filter_options (*options, trans, other_values*)

class galaxy.tools.parameters.dynamic_options.**SortByColumnFilter** (*d_option, elem*)
 Bases: *galaxy.tools.parameters.dynamic_options.Filter*

Sorts an options list by a column

Type: sort_by

Required Attributes: column: column to sort by

filter_options (*options, trans, other_values*)

class galaxy.tools.parameters.dynamic_options.**StaticValueFilter** (*d_option, elem*)
 Bases: *galaxy.tools.parameters.dynamic_options.Filter*

Filters a list of options on a column by a static value.

Type: static_value

Required Attributes: value: static value to compare to column: column in options to compare with

Optional Attributes:

keep: Keep columns matching value (True) Discard columns matching value (False)

filter_options (*options, trans, other_values*)

class galaxy.tools.parameters.dynamic_options.**UniqueValueFilter** (*d_option, elem*)
 Bases: *galaxy.tools.parameters.dynamic_options.Filter*

Filters a list of options to be unique by a column value.

Type: unique_value

Required Attributes: column: column in options to compare with

filter_options (*options, trans, other_values*)

get_dependency_name ()

grouping Module Constructs for grouping tool parameters

class galaxy.tools.parameters.grouping.**Conditional**
 Bases: *galaxy.tools.parameters.grouping.Group*

get_current_case (*value, trans*)

get_initial_value (*trans, context, history=None*)

is_job_resource_conditional

label

to_dict (*trans, view='collection', value_mapper=None*)

type = 'conditional'

value_from_basic (*value, app, ignore_errors=False*)

value_to_basic (*value, app*)

visit_inputs (*prefix, value, callback*)

class galaxy.tools.parameters.grouping.**ConditionalWhen**
 Bases: object, *galaxy.model.item_attrs.Dictifiable*

dict_collection_visible_keys = ('value',)

```
    to_dict (trans, view='collection', value_mapper=None)

class galaxy.tools.parameters.grouping.Group
    Bases: object, galaxy.model.item_attrs.Dictifiable

    dict_collection_visible_keys = ('name', 'type')

    get_initial_value (trans, context, history=None)
        Return the initial state/value for this group

    to_dict (trans, view='collection', value_mapper=None)

    value_from_basic (value, app, ignore_errors=False)
        Convert a basic representation as produced by value_to_basic back into the preferred value form.

    value_to_basic (value, app)
        Convert value to a (possibly nested) representation using only basic types (dict, list, tuple, str, unicode,
        int, long, float, bool, None)

    visible

class galaxy.tools.parameters.grouping.Repeat
    Bases: galaxy.tools.parameters.grouping.Group

    dict_collection_visible_keys = ('name', 'type', 'title', 'help', 'default', 'min', 'max')

    get_initial_value (trans, context, history=None)

    label ()

    title_plural

    to_dict (trans, view='collection', value_mapper=None)

    type = 'repeat'

    value_from_basic (value, app, ignore_errors=False)

    value_to_basic (value, app)

    visit_inputs (prefix, value, callback)

class galaxy.tools.parameters.grouping.UploadDataset
    Bases: galaxy.tools.parameters.grouping.Group

    get_composite_dataset_name (context)

    get_datatype (trans, context)

    get_datatype_ext (trans, context)

    get_file_base_name (context)

    get_file_type (context)

    get_initial_value (trans, context, history=None)

    get_uploaded_datasets (trans, context, override_name=None, override_info=None)

    group_title (context)

    title_by_index (trans, index, context)

    title_plural

    type = 'upload_dataset'

    value_from_basic (value, app, ignore_errors=False)
```

```

value_to_basic (value, app)

visit_inputs (prefix, value, callback)

```

input_translation Module Tool Input Translation.

class galaxy.tools.parameters.input_translation.**ToolInputTranslator**

Bases: object

Handles Tool input translation. This is used for data source tools

```

>>> from galaxy.util import Params
>>> from elementtree.ElementTree import XML
>>> translator = ToolInputTranslator.from_element( XML(
...     '''
...     <request_param_translation>
...     <request_param galaxy_name="URL_method" remote_name="URL_method" missing="post" />
...     <request_param galaxy_name="URL" remote_name="URL" missing="" >
...         <append_param separator="&" first_separator="?" join="=">
...             <value name="_export" missing="1" />
...             <value name="GALAXY_URL" missing="0" />
...         </append_param>
...     </request_param>
...     <request_param galaxy_name="dbkey" remote_name="db" missing="?" />
...     <request_param galaxy_name="organism" remote_name="org" missing="unknown species" />
...     <request_param galaxy_name="table" remote_name="hgta_table" missing="unknown table" />
...     <request_param galaxy_name="description" remote_name="hgta_regionType" missing="no description" />
...     <request_param galaxy_name="data_type" remote_name="hgta_outputType" missing="tabular" >
...         <value_translation>
...             <value galaxy_value="tabular" remote_value="primaryTable" />
...             <value galaxy_value="tabular" remote_value="selectedFields" />
...             <value galaxy_value="wig" remote_value="wigData" />
...             <value galaxy_value="interval" remote_value="tab" />
...             <value galaxy_value="html" remote_value="hyperlinks" />
...             <value galaxy_value="fasta" remote_value="sequence" />
...         </value_translation>
...     </request_param>
... </request_param_translation>
...     ''' ) )
>>> params = Params( { 'db':'hg17', 'URL':'URL_value', 'org':'Human', 'hgta_outputType':'primaryTable' } )
>>> translator.translate( params )
>>> print params
{'hgta_outputType': 'primaryTable', 'data_type': 'tabular', 'table': 'unknown table', 'URL': 'URL_value', 'db': 'hg17', 'org': 'Human'}

```

classmethod **from_element** (elem)

Loads the proper filter by the type attribute of elem

translate (params)

update params in-place

output Module Support for dynamically modifying output attributes.

class galaxy.tools.parameters.output.**BooleanFilter** (parent, elem)

Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

filter_options (options, other_values)

tag = 'boolean'

```
class galaxy.tools.parameters.output.ColumnReplaceFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter
    filter_options (options, other_values)
    tag = 'column_replace'

class galaxy.tools.parameters.output.ColumnStripFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter
    filter_options (options, other_values)
    tag = 'column_strip'

class galaxy.tools.parameters.output.DatatypeIsInstanceToolOutputActionConditionalWhen (parent,
                                                                                             con-
                                                                                             fig_elem
                                                                                             value)
    Bases: galaxy.tools.parameters.output.ToolOutputActionConditionalWhen
    is_case (output_dataset, other_values)
    tag = 'when datatype_isinstance'

class galaxy.tools.parameters.output.FormatToolOutputAction (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputAction
    apply_action (output_dataset, other_values)
    tag = 'format'

class galaxy.tools.parameters.output.FromDataTableOutputActionOption (parent,
                                                                                             elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOption
    get_value (other_values)
    tag = 'from_data_table'

class galaxy.tools.parameters.output.FromFileToolOutputActionOption (parent,
                                                                                             elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOption
    get_value (other_values)
    tag = 'from_file'

class galaxy.tools.parameters.output.FromParamToolOutputActionOption (parent,
                                                                                             elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOption
    get_value (other_values)
    tag = 'from_param'

class galaxy.tools.parameters.output.InsertColumnToolOutputActionOptionFilter (parent,
                                                                                             elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter
    filter_options (options, other_values)
    tag = 'insert_column'

class galaxy.tools.parameters.output.MetadataToolOutputAction (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputAction
    apply_action (output_dataset, other_values)
```

```

    tag = 'metadata'

class galaxy.tools.parameters.output.MetadataValueFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

    filter_options (options, other_values)

    tag = 'metadata_value'

class galaxy.tools.parameters.output.MultipleSplitterFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

    filter_options (options, other_values)

    tag = 'multiple_splitter'

class galaxy.tools.parameters.output.NullToolOutputActionOption (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOption

    get_value (other_values)

    tag = 'null_option'

class galaxy.tools.parameters.output.ParamValueToolOutputActionOptionFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

    filter_options (options, other_values)

    tag = 'param_value'

class galaxy.tools.parameters.output.StringFunctionFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

    filter_options (options, other_values)

    tag = 'string_function'

class galaxy.tools.parameters.output.ToolOutputAction (parent, elem)
    Bases: object

    apply_action (output_dataset, other_values)

    classmethod from_elem (parent, elem)
        Loads the proper action by the type attribute of elem

    tag = 'action'

    tool

class galaxy.tools.parameters.output.ToolOutputActionConditional (parent, con-
                                                                    fig_elem)
    Bases: object

    apply_action (output_dataset, other_values)

    tag = 'conditional'

    tool

class galaxy.tools.parameters.output.ToolOutputActionConditionalWhen (parent,
                                                                    con-
                                                                    fig_elem,
                                                                    value)
    Bases: galaxy.tools.parameters.output.ToolOutputActionGroup

    apply_action (output_dataset, other_values)

```

```
classmethod from_elem (parent, when_elem)
    Loads the proper when by attributes of elem

get_ref (output_dataset, other_values)

is_case (output_dataset, other_values)

tag = 'when'

class galaxy.tools.parameters.output.ToolOutputActionGroup (parent, config_elem)
    Bases: object

    Manages a set of tool output dataset actions directives

    apply_action (output_dataset, other_values)

    tag = 'group'

    tool

class galaxy.tools.parameters.output.ToolOutputActionOption (parent, elem)
    Bases: object

    classmethod from_elem (parent, elem)
        Loads the proper action by the type attribute of elem

    get_value (other_values)

    tag = 'object'

    tool

class galaxy.tools.parameters.output.ToolOutputActionOptionFilter (parent, elem)
    Bases: object

    filter_options (options, other_values)

    classmethod from_elem (parent, elem)
        Loads the proper action by the type attribute of elem

    tag = 'filter'

    tool

class galaxy.tools.parameters.output.ValueToolOutputActionConditionalWhen (parent,
                                                                              con-
                                                                              fig_elem,
                                                                              value)
    Bases: galaxy.tools.parameters.output.ToolOutputActionConditionalWhen

    is_case (output_dataset, other_values)

    tag = 'when value'

galaxy.tools.parameters.output.action_type
    alias of FormatToolOutputAction

galaxy.tools.parameters.output.compare_endswith (value1, value2)

galaxy.tools.parameters.output.compare_eq (value1, value2)

galaxy.tools.parameters.output.compare_gt (value1, value2)

galaxy.tools.parameters.output.compare_gte (value1, value2)

galaxy.tools.parameters.output.compare_in (value1, value2)

galaxy.tools.parameters.output.compare_lt (value1, value2)
```



```
galaxy.tools.parameters.output.compare_lte(value1, value2)
galaxy.tools.parameters.output.compare_neq(value1, value2)
galaxy.tools.parameters.output.compare_re_search(value1, value2)
galaxy.tools.parameters.output.compare_startswith(value1, value2)
galaxy.tools.parameters.output.filter_type
    alias of ColumnReplaceFilter
galaxy.tools.parameters.output.option_type
    alias of FromDataTableOutputActionOption
galaxy.tools.parameters.output.parse_cast_attribute(cast)
galaxy.tools.parameters.output.parse_compare_type(compare)
```

sanitize Module	Tool Parameter specific sanitizing.
------------------------	-------------------------------------

```
class galaxy.tools.parameters.sanitize.ToolParameterSanitizer
    Bases: object
```

Handles tool parameter specific sanitizing.

```
>>> from elementtree.ElementTree import XML
>>> sanitizer = ToolParameterSanitizer.from_element( XML(
...     '''
...     <sanitizer invalid_char="">
...       <valid initial="string.letters"/>
...     </sanitizer>
...     ''' ) )
>>> sanitizer.sanitize_param( ''.join( sorted( [ c for c in string.printable ] ) ) ) == ''.join(
True
>>> slash = chr( 92 )
>>> sanitizer = ToolParameterSanitizer.from_element( XML(
...     '''
...     <sanitizer>
...       <valid initial="none">
...         <add preset="string.printable"/>
...         <remove value="&quot;"/>
...         <remove value="%s"/>
...       </valid>
...       <mapping initial="none">
...         <add source="&quot;" target="%s&quot;"/>
...         <add source="%s" target="%s%s"/>
...       </mapping>
...     </sanitizer>
...     ''' % ( slash, slash, slash, slash ) ) )
>>> text = '%s"%$rm&#!' % slash
>>> [ c for c in sanitizer.sanitize_param( text ) ] == [ slash, slash, slash, '""', '$', 'r', 'm'
True
```

```
DEFAULT_INVALID_CHAR = 'X'
```

```
MAPPING_PRESET = {'default': {'@': '__at__', '\t': '__tc__', '\n': '__cn__', '\r': '__cr__', '[': '__ob__', ']': '__cb__', '#'
```

```
VALID_PRESET = {'default': 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 -_.(
```

classmethod from_element (*elem*)

Loads the proper filter by the type attribute of elem

```

classmethod get_mapping_by_name (name)
classmethod get_valid_by_name (name)
restore_param (value)
restore_text (text)
    Restores sanitized text
sanitize_param (value)
    Clean incoming parameters (strings or lists)
sanitize_text (text)
    Restricts the characters that are allowed in a text

```

validation Module Classes related to parameter validation.

```

class galaxy.tools.parameters.validation.DatasetOkValidator (message=None)
    Bases: galaxy.tools.parameters.validation.Validator

    Validator that checks if a dataset is in an 'ok' state

    classmethod from_element (param, elem)
    validate (value, history=None)

class galaxy.tools.parameters.validation.EmptyTextfieldValidator (message=None)
    Bases: galaxy.tools.parameters.validation.Validator

    Validator that checks for empty text field

    classmethod from_element (param, elem)
    validate (value, history=None)

class galaxy.tools.parameters.validation.ExpressionValidator (message,           ex-
                                                                pression,           substi-
                                                                tute_value_in_message)

    Bases: galaxy.tools.parameters.validation.Validator

    Validator that evaluates a python expression using the value

```

```

>>> from galaxy.tools.parameters import ToolParameter
>>> p = ToolParameter.build( None, XML( '''
... <param name="blah" type="text" size="10" value="10">
...     <validator type="expression" message="Not gonna happen">value.lower() == "foo"</validator>
... </param>
... ''' ) )
>>> t = p.validate( "Foo" )
>>> t = p.validate( "foo" )
>>> t = p.validate( "Fop" )
Traceback (most recent call last):
...
ValueError: Not gonna happen

```

```

    classmethod from_element (param, elem)
    validate (value, history=None)

class galaxy.tools.parameters.validation.InRangeValidator (message,           range_min,
                                                                range_max)

    Bases: galaxy.tools.parameters.validation.Validator

```

Validator that ensures a number is in a specific range

```
>>> from galaxy.tools.parameters import ToolParameter
>>> p = ToolParameter.build( None, XML( '''
... <param name="blah" type="integer" size="10" value="10">
...     <validator type="in_range" message="Not gonna happen" min="10" max="20"/>
... </param>
... ''' ) )
>>> t = p.validate( 10 )
>>> t = p.validate( 15 )
>>> t = p.validate( 20 )
>>> t = p.validate( 21 )
Traceback (most recent call last):
...
ValueError: Not gonna happen
```

classmethod `from_element` (*param, elem*)

validate (*value, history=None*)

exception `galaxy.tools.parameters.validation.LateValidationError` (*message*)

Bases: `exceptions.Exception`

class `galaxy.tools.parameters.validation.LengthValidator` (*message, length_min, length_max*)

Bases: `galaxy.tools.parameters.validation.Validator`

Validator that ensures the length of the provided string (*value*) is in a specific range

```
>>> from galaxy.tools.parameters import ToolParameter
>>> p = ToolParameter.build( None, XML( '''
... <param name="blah" type="text" size="10" value="foobar">
...     <validator type="length" min="2" max="8"/>
... </param>
... ''' ) )
>>> t = p.validate( "foo" )
>>> t = p.validate( "bar" )
>>> t = p.validate( "f" )
Traceback (most recent call last):
...
ValueError: Must have length of at least 2
>>> t = p.validate( "foobarbaz" )
Traceback (most recent call last):
...
ValueError: Must have length no more than 8
```

classmethod `from_element` (*param, elem*)

validate (*value, history=None*)

```
class galaxy.tools.parameters.validation.MetadataInDataTableColumnValidator (tool_data_table,
                                                                              meta-
                                                                              data_name,
                                                                              meta-
                                                                              data_column,
                                                                              mes-
                                                                              sage='Value
                                                                              for
                                                                              meta-
                                                                              data
                                                                              not
                                                                              found.',
                                                                              line_startswith=None)
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that checks if the value for a dataset's metadata item exists in a file.

classmethod **from_element** (*param, elem*)

validate (*value, history=None*)

```
class galaxy.tools.parameters.validation.MetadataInFileColumnValidator (filename,
                                                                          meta-
                                                                          data_name,
                                                                          meta-
                                                                          data_column,
                                                                          mes-
                                                                          sage='Value
                                                                          for
                                                                          meta-
                                                                          data
                                                                          not
                                                                          found.',
                                                                          line_startswith=None)
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that checks if the value for a dataset's metadata item exists in a file.

classmethod **from_element** (*param, elem*)

validate (*value, history=None*)

```
class galaxy.tools.parameters.validation.MetadataValidator (message=None,
                                                            check='', skip='')
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that checks for missing metadata

classmethod **from_element** (*param, elem*)

validate (*value, history=None*)

```
class galaxy.tools.parameters.validation.NoOptionsValidator (message=None)
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that checks for empty select list

classmethod **from_element** (*param, elem*)

validate (*value, history=None*)

```
class galaxy.tools.parameters.validation.RegexValidator (message, expression)
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that evaluates a regular expression

```
>>> from galaxy.tools.parameters import ToolParameter
>>> p = ToolParameter.build( None, XML( '''
... <param name="blah" type="text" size="10" value="10">
...     <validator type="regex" message="Not gonna happen">[Ff]oo</validator>
... </param>
... ''' ) )
>>> t = p.validate( "Foo" )
>>> t = p.validate( "foo" )
>>> t = p.validate( "Fop" )
Traceback (most recent call last):
...
ValueError: Not gonna happen
```

classmethod `from_element` (*param, elem*)

validate (*value, history=None*)

class `galaxy.tools.parameters.validation.UnspecifiedBuildValidator` (*message=None*)
Bases: `galaxy.tools.parameters.validation.Validator`

Validator that checks for dbkey not equal to ‘?’

classmethod `from_element` (*param, elem*)

validate (*value, history=None*)

class `galaxy.tools.parameters.validation.Validator`
Bases: `object`

A validator checks that a value meets some conditions OR raises `ValueError`

classmethod `from_element` (*param, elem*)

validate (*value, history=None*)

`galaxy.tools.parameters.validation.get_suite` ()
Get unittest suite for this module

search Package

search Package

class `galaxy.tools.search.ToolBoxSearch` (*toolbox*)
Bases: `object`

Support searching tools in a toolbox. This implementation uses the “whoosh” search library.

build_index ()

search (*query, return_attribute='id'*)

util Package

util Package Utilities used by various Galaxy tools

FIXME: These are used by tool scripts, not the framework, and should not live in this package.

maf_utilities Module Provides wrappers and utilities for working with MAF files and alignments.

```
class galaxy.tools.util.maf_utilities.GenomicRegionAlignment (start, end, species=[],
                                                             temp_file_handler=None)
    Bases: galaxy.tools.util.maf_utilities.RegionAlignment
```

```
class galaxy.tools.util.maf_utilities.RegionAlignment (size,                                     species=[],
                                                         temp_file_handler=None)
    Bases: object
```

```
DNA_COMPLEMENT = '\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\x00'
```

```
MAX_SEQUENCE_SIZE = 9223372036854775807
```

```
add_species (species)
```

```
flush (species=None)
```

```
get_sequence (species)
```

```
get_sequence_reverse_complement (species)
```

```
get_species_names (skip=[])
```

```
set_position (index, species, base)
```

```
set_range (index, species, bases)
```

```
class galaxy.tools.util.maf_utilities.SplicedAlignment (exon_starts,
                                                         exon_ends,                                     species=[],
                                                         temp_file_handler=None)
    Bases: object
```

```
DNA_COMPLEMENT = '\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\x00'
```

```
end
```

```
get_sequence (species)
```

```
get_sequence_reverse_complement (species)
```

```
get_species_names (skip=[])
```

```
start
```

```
class galaxy.tools.util.maf_utilities.TempFileHandler (max_open_files=None, **kwargs)
    Bases: object
```

Handles creating, opening, closing, and deleting of Temp files, with a maximum number of files open at one time.

```
DEFAULT_MAX_OPEN_FILES = 512
```

```
close (index, delete=False)
```

```
flush (index)
```

```
get_open_tempfile (index=None, **kwargs)
```

```
galaxy.tools.util.maf_utilities.build_maf_index (maf_file, species=None)
```

```
galaxy.tools.util.maf_utilities.build_maf_index_species_chromosomes (filename,
                                                                       in-
                                                                       dex_species=None)
```

```
galaxy.tools.util.maf_utilities.chop_block_by_region (block,          src,          region,
                                                         species=None, mincols=0)
```

```
galaxy.tools.util.maf_utilities.component_overlaps_region (c, region)
```

```

galaxy.tools.util.maf_utilities.fill_region_alignment (alignment, index, primary_species, chrom, start, end, strand='+', species=None, mincols=0, overwrite_with_gaps=True)

galaxy.tools.util.maf_utilities.get_attributes_from_fasta_header (header)

galaxy.tools.util.maf_utilities.get_chopped_blocks_for_region (index, src, region, species=None, mincols=0)

galaxy.tools.util.maf_utilities.get_chopped_blocks_with_index_offset_for_region (index, src, re- gion, species=None, min- cols=0)

galaxy.tools.util.maf_utilities.get_components_by_src (block, src)

galaxy.tools.util.maf_utilities.get_components_by_src_start (block, src)

galaxy.tools.util.maf_utilities.get_fasta_header (component, attributes={}, suf- fix=None)

galaxy.tools.util.maf_utilities.get_oriented_chopped_blocks_for_region (index, src, re- gion, species=None, min- cols=0, force_strand=None)

galaxy.tools.util.maf_utilities.get_oriented_chopped_blocks_with_index_offset_for_region (in- sr- re- gi- sp- m- co- fo-)

galaxy.tools.util.maf_utilities.get_region_alignment (index, primary_species, chrom, start, end, strand='+', species=None, mincols=0, overwrite_with_gaps=True, temp_file_handler=None)

galaxy.tools.util.maf_utilities.get_species_in_block (block)

galaxy.tools.util.maf_utilities.get_species_in_maf (maf_filename)

```

```
galaxy.tools.util.maf_utilities.get_spliced_region_alignment(index, primary_species,
                                                             chrom, starts,
                                                             ends, strand='+',
                                                             species=None,
                                                             mincols=0, over-
                                                             write_with_gaps=True,
                                                             temp_file_handler=None)

galaxy.tools.util.maf_utilities.get_starts_ends_fields_from_gene_bed(line)

galaxy.tools.util.maf_utilities.iter_blocks_split_by_species(block,
                                                             species=None)

galaxy.tools.util.maf_utilities.iter_blocks_split_by_src(block, src)

galaxy.tools.util.maf_utilities.iter_components_by_src(block, src)

galaxy.tools.util.maf_utilities.iter_components_by_src_start(block, src)

galaxy.tools.util.maf_utilities.iter_fasta_alignment(filename)

galaxy.tools.util.maf_utilities.line_enumerator(lines, comment_start='#')

galaxy.tools.util.maf_utilities.maf_index_by_uid(maf_uid, index_location_file)

galaxy.tools.util.maf_utilities.open_or_build_maf_index(maf_file, index_filename,
                                                         species=None)

galaxy.tools.util.maf_utilities.orient_block_by_region(block, src, region,
                                                         force_strand=None)

galaxy.tools.util.maf_utilities.parse_species_option(species)

galaxy.tools.util.maf_utilities.reduce_block_by_primary_genome(block, species,
                                                                chromosome,
                                                                region_start)

galaxy.tools.util.maf_utilities.remove_temp_index_file(index_filename)

galaxy.tools.util.maf_utilities.sort_block_components_by_block(block1, block2)

galaxy.tools.util.maf_utilities.src_merge(spec, chrom, contig=None)

galaxy.tools.util.maf_utilities.src_split(src)

galaxy.tools.util.maf_utilities.tool_fail(msg='Unknown Error')
```

Subpackages

galaxyops Package

galaxyops Package Utility functions for galaxyops

```
galaxy.tools.util.galaxyops.default_printer(stream, exc, obj)

galaxy.tools.util.galaxyops.fail(msg)

galaxy.tools.util.galaxyops.parse_cols_arg(cols)
    Parse a columns command line argument into a four-tuple

galaxy.tools.util.galaxyops.skipped(reader, filedesc='')

galaxy.tools.util.galaxyops.warn(msg)
```


util Package

util Package Utility functions used systemwide.

class `galaxy.util.Params` (*params*, *sanitize=True*)

Bases: `object`

Stores and ‘sanitizes’ parameters. Alphanumeric characters and the non-alphanumeric ones that are deemed safe are let to pass through (see `L{valid_chars}`). Some non-safe characters are escaped to safe forms for example `C{>}` becomes `C{__lt__}` (see `L{mapped_chars}`). All other characters are replaced with `C{X}`.

Operates on string or list values only (HTTP parameters).

```

>>> values = { 'status':'on', 'symbols':[ 'alpha', '<>', '$rm&#!' ] }
>>> par = Params(values)
>>> par.status
'on'
>>> par.value == None      # missing attributes return None
True
>>> par.get('price', 0)
0
>>> par.symbols            # replaces unknown symbols with X
['alpha', '__lt____gt__', 'XrmX__pd__!']
>>> par.flatten()         # flattening to a list
[('status', 'on'), ('symbols', 'alpha'), ('symbols', '__lt____gt__'), ('symbols', 'XrmX__pd__!')]

```

NEVER_SANITITIZE = `['file_data', 'url_paste', 'URL', 'filesystem_paths']`

flatten()

Creates a tuple list from a dict with a tuple/value pair for every value that is a list

get (*key*, *default*)

update (*values*)

class `galaxy.util.ParamsWithSpecs` (*specs=None*, *params=None*)

Bases: `collections.defaultdict`

`galaxy.util.asbool` (*obj*)

`galaxy.util.commaify` (*amount*)

`galaxy.util.compare_urls` (*url1*, *url2*, *compare_scheme=True*, *compare_hostname=True*, *compare_path=True*)

`galaxy.util.docstring_trim` (*docstring*)

Trimming python doc strings. Taken from: <http://www.python.org/dev/peps/pep-0257/>

`galaxy.util.file_iter` (*fname*, *sep=None*)

This generator iterates over a file and yields its lines splitted via the `C{sep}` parameter. Skips empty lines and lines starting with the `C{#}` character.

```

>>> lines = [ line for line in file_iter(__file__) ]
>>> len(lines) != 0
True

```

`galaxy.util.file_reader` (*fp*, *chunk_size=65536*)

This generator yields the open fileobject in chunks (default 64k). Closes the file at the end

`galaxy.util.force_symlink` (*source*, *link_name*)

`galaxy.util.galaxy_directory()`

`galaxy.util.get_charset_from_http_headers(headers, default=None)`

`galaxy.util.get_file_size(value, default=None)`

`galaxy.util.in_directory(file, directory, local_path_module=<module 'posixpath' from
'/home/docs/checkouts/readthedocs.org/user_builds/jmchilton-
galaxy/envs/stable/lib/python2.7/posixpath.pyc'>)`

Return true, if the common prefix of both is equal to directory e.g. /a/b/c/d.rst and directory is /a/b, the common prefix is /a/b

`galaxy.util.is_binary(value, binary_chars=None)`

File is binary if it contains a null-byte by default (e.g. behavior of grep, etc.). This may fail for utf-16 files, but so would ASCII encoding. >>> is_binary(string.printable) False >>> is_binary('xcex94') False >>> is_binary('000') True

`galaxy.util.is_multi_byte(chars)`

`galaxy.util.is_uuid(value)`

This method returns True if value is a UUID, otherwise False. >>> is_uuid("123e4567-e89b-12d3-a456-426655440000") True >>> is_uuid("0x3242340298902834") False

`galaxy.util.listify(item, do_strip=False)`

Make a single item a single item list, or return a list if passed a list. Passing a None returns an empty list.

`galaxy.util.merge_sorted_iterables(operator, *iterables)`

```
>>> operator = lambda x: x
>>> list( merge_sorted_iterables( operator, [1,2,3], [4,5] ) )
[1, 2, 3, 4, 5]
>>> list( merge_sorted_iterables( operator, [4, 5], [1,2,3] ) )
[1, 2, 3, 4, 5]
>>> list( merge_sorted_iterables( operator, [1, 4, 5], [2], [3] ) )
[1, 2, 3, 4, 5]
```

`galaxy.util.mkstemp_ln(src, prefix='mkstemp_ln_')`

From tempfile.mkstemp_inner, generate a hard link in the same dir with a random name. Created so we can persist the underlying file of a NamedTemporaryFile upon its closure.

`galaxy.util.move_merge(source, target)`

`galaxy.util.nice_size(size)`

Returns a readably formatted string with the size

```
>>> nice_size(100)
'100 bytes'
>>> nice_size(10000)
'9.8 KB'
>>> nice_size(1000000)
'976.6 KB'
>>> nice_size(100000000)
'95.4 MB'
```

`galaxy.util.object_to_string(obj)`

`galaxy.util.parse_xml(fname)`

Returns a parsed xml tree

`galaxy.util.parse_xml_string(xml_string)`

```
galaxy.util.pretty_print_json(json_data, is_json_string=False)
```

```
galaxy.util.pretty_print_xml(elem, level=0)
```

```
galaxy.util.read_build_sites(filename, check_builds=True)
```

read db names to ucsc mappings from file, this file should probably be merged with the one above

```
galaxy.util.read_dbnames(filename)
```

Read build names from file

```
galaxy.util.ready_name_for_url(raw_name)
```

General method to convert a string (i.e. object name) to a URL-ready slug.

```
>>> ready_name_for_url( "My Cool Object" )
'My-Cool-Object'
>>> ready_name_for_url( "!My Cool Object!" )
'My-Cool-Object'
>>> ready_name_for_url( "HelloW" )
'Hello'
```

```
galaxy.util.recursively_stringify_dictionary_keys(d)
```

```
galaxy.util.relativize_symlinks(path, start=None, followlinks=False)
```

```
galaxy.util.restore_text(text, character_map={'@': '__at__', '\t': '__tc__', '\n': '__cn__', '\r':
        '__cr__', '[': '__ob__', ']': '__cb__', '#': '__pd__', '"': '__dq__', "'":
        '__sq__', '{': '__oc__', '}': '__cc__', '<': '__lt__', '>': '__gt__'})
```

Restores sanitized text

```
galaxy.util.roundify(amount, sfs=2)
```

Take a number in string form and truncate to 'sfs' significant figures.

```
galaxy.util.rst_to_html(s)
```

Convert a blob of reStructuredText to HTML

```
galaxy.util.safe_str_cmp(a, b)
```

```
galaxy.util.sanitize_for_filename(text, default=None)
```

Restricts the characters that are allowed in a filename portion; Returns default value or a unique id string if result is not a valid name. Method is overly aggressive to minimize possible complications, but a maximum length is not considered.

```
galaxy.util.sanitize_lists_to_string(values, valid_characters=set(['!', ' ', ')', '(', '+', '*',
        '-', ',', '/', ':', 'I', 'O', '3', '2', '5', '4', '7', '6', '9', '8',
        ':', '=', '?', 'A', 'C', 'B', 'E', 'D', 'G', 'F', 'I', 'H', 'K',
        'J', 'M', 'L', 'O', 'N', 'Q', 'P', 'S', 'R', 'U', 'T', 'W',
        'V', 'Y', 'X', 'Z', '_', '^', 'a', 'c', 'b', 'e', 'd', 'g', 'f', 'i',
        'h', 'k', 'j', 'm', 'l', 'o', 'n', 'q', 'p', 's', 'r', 'u', 't', 'w',
        'v', 'y', 'x', 'z']), character_map={'@': '__at__', '\t':
        '__tc__', '\n': '__cn__', '\r': '__cr__', '[': '__ob__',
        ']': '__cb__', '#': '__pd__', '"': '__dq__', "'":
        '__sq__', '{': '__oc__', '}': '__cc__', '<': '__lt__',
        '>': '__gt__'}, invalid_character='X')
```

```
galaxy.util.sanitize_param(value, valid_characters=set(['!', ' ', ')', '(', '+', '*', '-', ' ', '/', ':',
'I', 'O', '3', '2', '5', '4', '7', '6', '9', '8', ':', '=', '?', 'A', 'C', 'B',
'E', 'D', 'G', 'F', 'T', 'H', 'K', 'J', 'M', 'L', 'O', 'N', 'Q', 'P', 'S',
'R', 'U', 'T', 'W', 'V', 'Y', 'X', 'Z', '_', '^', 'a', 'c', 'b', 'e', 'd', 'g',
'f', 'i', 'h', 'k', 'j', 'm', 'l', 'o', 'n', 'q', 'p', 's', 'r', 'u', 't', 'w',
'v', 'y', 'x', 'z']), character_map={'@': '__at__', '\': '__tc__', '\n':
'__cn__', '\r': '__cr__', '[': '__ob__', ']': '__cb__', '#': '__pd__',
'"': '__dq__', "'": '__sq__', '{': '__oc__', '}': '__cc__', '<':
'__lt__', '>': '__gt__'}, invalid_character='X')
```

Clean incoming parameters (strings or lists)

```
galaxy.util.sanitize_text(text, valid_characters=set(['!', ' ', ')', '(', '+', '*', '-', ' ', '/', ':',
'I', 'O', '3', '2', '5', '4', '7', '6', '9', '8', ':', '=', '?', 'A', 'C', 'B',
'E', 'D', 'G', 'F', 'T', 'H', 'K', 'J', 'M', 'L', 'O', 'N', 'Q', 'P', 'S',
'R', 'U', 'T', 'W', 'V', 'Y', 'X', 'Z', '_', '^', 'a', 'c', 'b', 'e', 'd', 'g',
'f', 'i', 'h', 'k', 'j', 'm', 'l', 'o', 'n', 'q', 'p', 's', 'r', 'u', 't', 'w',
'v', 'y', 'x', 'z']), character_map={'@': '__at__', '\': '__tc__', '\n':
'__cn__', '\r': '__cr__', '[': '__ob__', ']': '__cb__', '#': '__pd__',
'"': '__dq__', "'": '__sq__', '{': '__oc__', '}': '__cc__', '<': '__lt__',
'>': '__gt__'}, invalid_character='X')
```

Restricts the characters that are allowed in text; accepts both strings and lists of strings; non-string entities will be cast to strings.

```
galaxy.util.send_mail(frm, to, subject, body, config)
```

Sends an email.

```
galaxy.util.shrink_stream_by_size(value, size, join_by='.', left_larger=True, begin-
ning_on_size_error=False, end_on_size_error=False)
```

```
galaxy.util.shrink_string_by_size(value, size, join_by='.', left_larger=True, begin-
ning_on_size_error=False, end_on_size_error=False)
```

```
galaxy.util.size_to_bytes(size)
```

Returns a number of bytes if given a reasonably formatted string with the size

```
galaxy.util.smart_str(s, encoding='utf-8', strings_only=False, errors='strict')
```

Returns a bytestring version of 's', encoded as specified in 'encoding'.

If strings_only is True, don't convert (some) non-string-like objects.

Adapted from an older, simpler version of django.utils.encoding.smart_str.

```
galaxy.util.string_as_bool(string)
```

```
galaxy.util.string_as_bool_or_none(string)
```

Returns True, None or False based on the argument: True if passed True, 'True', 'Yes', or 'On' None if passed None or 'None' False otherwise

Note: string comparison is case-insensitive so lowercase versions of those function equivalently.

```
galaxy.util.string_to_object(s)
```

```
galaxy.util.stringify_dictionary_keys(in_dict)
```

```
galaxy.util.synchronized(func)
```

This wrapper will serialize access to 'func' to a single thread. Use it as a decorator.

```
galaxy.util.umask_fix_perms(path, umask, unmasked_perms, gid=None)
```

umask-friendly permissions fixing

```
galaxy.util.unicodify(value, encoding='utf-8', error='replace', default=None)
```

Returns a unicode string or None

`galaxy.util.unique_id(KEY_SIZE=128)`
Generates an unique id

```
>>> ids = [ unique_id() for i in range(1000) ]
>>> len(set(ids))
1000
```

`galaxy.util.xml_element_compare(elem1, elem2)`

`galaxy.util.xml_element_list_compare(elem_list1, elem_list2)`

`galaxy.util.xml_element_to_dict(elem)`

`galaxy.util.xml_text(root, name=None)`
Returns the text inside an element

`galaxy.util.xml_to_string(elem, pretty=False)`
Returns a string from an xml tree

aliaspickler Module

`class galaxy.util.aliaspickler.AliasPickleModule(alises)`
Bases: object

`dump(obj, fileobj, protocol=0)`

`dumps(obj, protocol=0)`

`load(fileobj)`

`loads(string)`

`class galaxy.util.aliaspickler.AliasUnpickler(alises, *args, **kw)`
Bases: `pickle.Unpickler`

`find_class(module, name)`

bunch Module

`class galaxy.util.bunch.Bunch(**kws)`
Bases: object

<http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/52308>

Often we want to just collect a bunch of stuff together, naming each item of the bunch; a dictionary's OK for that, but a small do-nothing class is even handier, and prettier to use.

`get(key, default=None)`

`items()`

`keys()`

`values()`

debugging Module

`class galaxy.util.debugging.SimpleProfiler(log=None)`
Bases: object

Simple profiler that captures the duration between calls to *report* and stores the results in a list.

`REPORT_FORMAT = '%20f: %s'`

`get_reports()`

report (*msg*)

start (*msg=None*)

`galaxy.util.debugging.stack_trace_string` (*max_depth=None*,
line_format='{index}:{file}:{function}:{line}')

Returns a string representation of the current stack.

Parameters **depth** – positive integer to control how many levels of the stack are returned. `max_depth=None` returns the entire stack (default).

expressions Module Expression evaluation support.

For the moment this depends on python’s eval. In the future it should be replaced with a “safe” parser.

class `galaxy.util.expressions.ExpressionContext` (*dict*, *parent=None*)

Bases: `object`, `UserDict.DictMixin`

hash_util Module Utility functions for bi-directional Python version compatibility. Python 2.5 introduced hash-lib which replaced sha in Python 2.4 and previous versions.

`galaxy.util.hash_util.hmac_new` (*key*, *value*)

`galaxy.util.hash_util.is_hashable` (*value*)

`galaxy.util.hash_util.new_secure_hash` (*text_type=None*)

Returns either a sha1 hash object (if called with no arguments), or a hexdigest of the sha1 hash of the argument *text_type*.

heartbeat Module

class `galaxy.util.heartbeat.Heartbeat` (*name='Heartbeat Thread'*, *period=20*,
fname='heartbeat.log')

Bases: `threading.Thread`

Thread that periodically dumps the state of all threads to a file

get_interesting_stack_frame (*stack_frames*)

Scans a given backtrace stack frames, returns a single quadruple of [filename, line, function-name, text] of the single, deepest, most interesting frame.

Interesting being:

inside the galaxy source code (" <code>/lib/galaxy</code> "), preferably not an egg.

print_nonsleeping (*threads_object_dict*)

run ()

shutdown ()

thread_is_sleeping (*last_stack_frame*)

Returns True if the given stack-frame represents a known sleeper function (at least in python 2.5)

`galaxy.util.heartbeat.get_current_thread_object_dict` ()

Get a dictionary of all ‘Thread’ objects created via the threading module keyed by `thread_id`. Note that not all interpreter threads have a thread objects, only the main thread and any created via the ‘threading’ module. Threads created via the low level ‘thread’ module will not be in the returned dictionary.

HACK: This mucks with the internals of the threading module since that module does not expose any way to match ‘Thread’ objects with interpreter thread identifiers (though it should).

inflection Module**class** `galaxy.util.inflection.Base`

Locale inflectors must inherit from this base class in order to provide the basic Inflector functionality

camelize (*word*)

Returns given word as CamelCased Converts a word like “send_email” to “SendEmail”. It will remove non alphanumeric character from the word, so “who’s online” will be converted to “WhoSOnline”

classify (*table_name*)

Converts a table name to its class name according to rails naming conventions. Example: Converts “people” to “Person”

cond_plural (*number_of_records, word*)

Returns the plural form of a word if first parameter is greater than 1

demodulize (*module_name*)**foreignKey** (*class_name, separate_class_name_and_id_with_underscore=1*)

Returns class_name in underscored form, with “_id” tacked on at the end. This is for use in dealing with the database.

humanize (*word, uppercase=''*)

Returns a human-readable string from word Returns a human-readable string from word, by replacing underscores with a space, and by upper-casing the initial character by default. If you need to uppercase all the words you just have to pass ‘all’ as a second parameter.

modulize (*module_description*)**ordinalize** (*number*)

Converts number to its ordinal English form. This method converts 13 to 13th, 2 to 2nd ...

string_replace (*word, find, replace*)

This function returns a copy of word, translating all occurrences of each character in find to the corresponding character in replace

tableize (*class_name*)

Converts a class name to its table name according to rails naming conventions. Example. Converts “Person” to “people”

titleize (*word, uppercase=''*)

Converts an underscored or CamelCase word into a English sentence. The titleize function converts text like “WelcomePage”, “welcome_page” or “welcome page” to this “Welcome Page”. If second parameter is set to ‘first’ it will only capitalize the first character of the title.

unaccent (*text*)

Transforms a string to its unaccented version. This might be useful for generating “friendly” URLs

underscore (*word*)

Converts a word “into_it_s_underscored_version” Convert any “CamelCased” or “ordinary Word” into an “underscored_word”. This can be really useful for creating friendly URLs.

urlize (*text*)

Transform a string its unaccented and underscored version ready to be inserted in friendly URLs

variablize (*word*)

Same as camelize but first char is lowercased Converts a word like “send_email” to “sendEmail”. It will remove non alphanumeric character from the word, so “who’s online” will be converted to “whoSOnline”

class `galaxy.util.inflection.English`Bases: `galaxy.util.inflection.Base`

Inflector for pluralize and singularize English nouns.

This is the default Inflector for the Inflector obj

pluralize (*word*)

Pluralizes English nouns.

singularize (*word*)

Singularizes English nouns.

class `galaxy.util.inflection.Inflector` (*Inflector=<class galaxy.util.inflection.English>*)

Inflector for pluralizing and singularizing nouns.

It provides methods for helping on creating programs based on naming conventions like on Ruby on Rails.

camelize (*word*)

Returns given word as CamelCased Converts a word like “send_email” to “SendEmail”. It will remove non alphanumeric character from the word, so “who’s online” will be converted to “WhoSOnline”

classify (*table_name*)

Converts a table name to its class name according to rails naming conventions. Example: Converts “people” to “Person”

cond_plural (*number_of_records, word*)

Returns the plural form of a word if first parameter is greater than 1

demodulize (*module_name*)

foreignKey (*class_name, separate_class_name_and_id_with_underscore=1*)

Returns class_name in underscored form, with “_id” tacked on at the end. This is for use in dealing with the database.

humanize (*word, uppercase=''*)

Returns a human-readable string from word Returns a human-readable string from word, by replacing underscores with a space, and by upper-casing the initial character by default. If you need to uppercase all the words you just have to pass ‘all’ as a second parameter.

modulize (*module_description*)

ordinalize (*number*)

Converts number to its ordinal form. This method converts 13 to 13th, 2 to 2nd ...

pluralize (*word*)

Pluralizes nouns.

singularize (*word*)

Singularizes nouns.

tableize (*class_name*)

Converts a class name to its table name according to rails naming conventions. Example. Converts “Person” to “people”

titleize (*word, uppercase=''*)

Converts an underscored or CamelCase word into a sentence. The titleize function converts text like “WelcomePage”, “welcome_page” or “welcome page” to this “Welcome Page”. If the “uppercase” parameter is set to ‘first’ it will only capitalize the first character of the title.

unaccent (*text*)

Transforms a string to its unaccented version. This might be useful for generating “friendly” URLs

underscore (*word*)

Converts a word “into_it_s_underscored_version” Convert any “CamelCased” or “ordinary Word” into an “underscored_word”. This can be really useful for creating friendly URLs.

urlize (*text*)

Transform a string to its unaccented and underscored version ready to be inserted in friendly URLs

variablize (*word*)

Same as camelize but first char is lowercased Converts a word like “send_email” to “sendEmail”. It will remove non alphanumeric character from the word, so “who’s online” will be converted to “whoSONline”

json Module

`galaxy.util.json.dumps` (*obj*, *skipkeys=False*, *ensure_ascii=True*, *check_circular=True*, *allow_nan=True*, *cls=None*, *indent=None*, *separators=None*, *encoding='utf-8'*, *default=None*, *sort_keys=False*, ***kw*)

Serialize *obj* to a JSON formatted *str*.

If *skipkeys* is false then dict keys that are not basic types (*str*, *unicode*, *int*, *long*, *float*, *bool*, *None*) will be skipped instead of raising a *TypeError*.

If *ensure_ascii* is false, all non-ASCII characters are not escaped, and the return value may be a *unicode* instance. See *dump* for details.

If *check_circular* is false, then the circular reference check for container types will be skipped and a circular reference will result in an *OverflowError* (or worse).

If *allow_nan* is false, then it will be a *ValueError* to serialize out of range float values (*nan*, *inf*, *-inf*) in strict compliance of the JSON specification, instead of using the JavaScript equivalents (*NaN*, *Infinity*, *-Infinity*).

If *indent* is a non-negative integer, then JSON array elements and object members will be pretty-printed with that indent level. An indent level of 0 will only insert newlines. *None* is the most compact representation. Since the default item separator is *' , '*, the output might include trailing whitespace when *indent* is specified. You can use *separators=(', ', ': ')* to avoid this.

If *separators* is an (*item_separator*, *dict_separator*) tuple then it will be used instead of the default *(' , ', ': ')* separators. *(' , ', ': ')* is the most compact JSON representation.

encoding is the character encoding for *str* instances, default is UTF-8.

default (*obj*) is a function that should return a serializable version of *obj* or raise *TypeError*. The default simply raises *TypeError*.

If *sort_keys* is *True* (default: *False*), then the output of dictionaries will be sorted by key.

To use a custom *JSONEncoder* subclass (e.g. one that overrides the *.default()* method to serialize additional types), specify it with the *cls* kwarg; otherwise *JSONEncoder* is used.

`galaxy.util.json.loads` (*s*, *encoding=None*, *cls=None*, *object_hook=None*, *parse_float=None*, *parse_int=None*, *parse_constant=None*, *object_pairs_hook=None*, ***kw*)

Deserialize *s* (a *str* or *unicode* instance containing a JSON document) to a Python object.

If *s* is a *str* instance and is encoded with an ASCII based encoding other than utf-8 (e.g. latin-1) then an appropriate encoding name must be specified. Encodings that are not ASCII based (such as UCS-2) are not allowed and should be decoded to *unicode* first.

object_hook is an optional function that will be called with the result of any object literal decode (a *dict*). The return value of *object_hook* will be used instead of the *dict*. This feature can be used to implement custom decoders (e.g. JSON-RPC class hinting).

object_pairs_hook is an optional function that will be called with the result of any object literal decoded with an ordered list of pairs. The return value of *object_pairs_hook* will be used instead of the *dict*. This feature can be used to implement custom decoders that rely on the order that the key and value pairs are decoded (for example, *collections.OrderedDict* will remember the order of insertion). If *object_hook* is also defined, the *object_pairs_hook* takes priority.

`parse_float`, if specified, will be called with the string of every JSON float to be decoded. By default this is equivalent to `float(num_str)`. This can be used to use another datatype or parser for JSON floats (e.g. `decimal.Decimal`).

`parse_int`, if specified, will be called with the string of every JSON int to be decoded. By default this is equivalent to `int(num_str)`. This can be used to use another datatype or parser for JSON integers (e.g. `float`).

`parse_constant`, if specified, will be called with one of the following strings: `-Infinity`, `Infinity`, `NaN`, `null`, `true`, `false`. This can be used to raise an exception if invalid JSON numbers are encountered.

To use a custom `JSONDecoder` subclass, specify it with the `cls` kwarg; otherwise `JSONDecoder` is used.

`galaxy.util.json.safe_dumps(*args, **kwargs)`

This is a wrapper around `dumps` that encodes `Infinity` and `NaN` values. It's a fairly rare case (which will be low in request volume). Basically, we tell `json.dumps` to blow up if it encounters `Infinity/NaN`, and we 'fix' it before re-encoding.

`galaxy.util.json.json_fix(val)`

`galaxy.util.json.validate_jsonrpc_request(request, regular_methods, notification_methods)`

`galaxy.util.json.validate_jsonrpc_response(response, id=None)`

`galaxy.util.json.jsonrpc_request(method, params=None, id=None, jsonrpc='2.0')`

`galaxy.util.json.jsonrpc_response(request=None, id=None, result=None, error=None, jsonrpc='2.0')`

lru_cache Module Kanwei Li, 03/2010

Simple LRU cache that uses a dictionary to store a specified number of objects at a time.

class `galaxy.util.lru_cache.LRUCache(num_elements)`

clear()

Clears/initiates storage variables

memdump Module

class `galaxy.util.memdump.Memdump(signum=10, fname='memdump.log')`

Bases: `object`

dump (*signum, stack*)

get (*update=False*)

setref ()

none_like Module Objects with No values

class `galaxy.util.none_like.NoneDataset(datatypes_registry=None, ext='data', dbkey='')`

Bases: `galaxy.util.none_like.RecursiveNone`

missing_meta ()

class `galaxy.util.none_like.RecursiveNone`

odict Module Ordered dictionary implementation.

```
class galaxy.util.odict.odict (dict=None)
    Bases: UserDict.UserDict
```

<http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/107747>

This dictionary class extends UserDict to record the order in which items are added. Calling keys(), values(), items(), etc. will return results in this order.

```
clear ()
copy ()
insert (index, key, item)
items ()
iteritems ()
iterkeys ()
itervalues ()
keys ()
popitem ()
reverse ()
setdefault (key, failobj=None)
update (dict)
values ()
```

sanitize_html Module HTML Sanitizer (ripped from feedparser)

```
galaxy.util.sanitize_html.sanitize_html (htmlSource, encoding='utf-8', type='text/html')
```

shed_util Module

shed_util_common Module

streamball Module A simple wrapper for writing tarballs as a stream.

```
class galaxy.util.streamball.StreamBall (mode, members=None)
    Bases: object
```

```
    add (file, relpath, check_file=False)
```

```
    stream (environ, start_response)
```

```
class galaxy.util.streamball.ZipBall (tmpf, tmpd)
    Bases: object
```

```
    stream (environ, start_response)
```

template Module

```
galaxy.util.template.fill_template (template_text, context=None, **kwargs)
```

topsort Module Topological sort.

From Tim Peters, see: <http://mail.python.org/pipermail/python-list/1999-July/006660.html>

topsort takes a list of pairs, where each pair (x, y) is taken to mean that $x \leq y$ wrt some abstract partial ordering. The return value is a list, representing a total ordering that respects all the input constraints. E.g.,

```
topsort( [(1,2), (3,3)] )
```

Valid topological sorts would be any of (but nothing other than)

```
[3, 1, 2] [1, 3, 2] [1, 2, 3]
```

... however this variant ensures that ‘key’ order (first element of tuple) is preserved so the following will be result returned:

```
[1, 3, 2]
```

because those are the permutations of the input elements that respect the “1 precedes 2” and “3 precedes 3” input constraints. Note that a constraint of the form (x, x) is really just a trick to make sure x appears *somewhere* in the output list.

If there’s a cycle in the constraints, say

```
topsort( [(1,2), (2,1)] )
```

then `CycleError` is raised, and the exception object supports many methods to help analyze and break the cycles. This requires a good deal more code than topsort itself!

exception `galaxy.util.topsort.CycleError` (*sofar, numpreds, succs*)

Bases: `exceptions.Exception`

`get_elements()`

`get_pairlist()`

`get_partial()`

`get_pred_counts()`

`get_preds()`

`get_succs()`

`pick_a_cycle()`

`galaxy.util.topsort.topsort(pairlist)`

`galaxy.util.topsort.topsort_levels(pairlist)`

Subpackages

backports Package

backports Package Modules for providing backward compatibility with future versions of Python

Subpackages

importlib Package

importlib Package Backport of importlib.import_module from 3.x.

`galaxy.util.backports.importlib.import_module(name, package=None)`
 Import a module.

The 'package' argument is required when performing a relative import. It specifies the package to use as the anchor point from which to resolve the relative import to an absolute import.

visualization Package

visualization Package Package for Galaxy visualization plugins.

genomes Module

class `galaxy.visualization.genomes.Genome(key, description, len_file=None, twobit_file=None)`
 Bases: object

Encapsulates information about a known genome/dbkey.

to_dict (*num=None, chrom=None, low=None*)
 Returns representation of self as a dictionary.

class `galaxy.visualization.genomes.GenomeRegion(chrom=None, start=0, end=0, sequence=None)`
 Bases: object

A genomic region on an individual chromosome.

static from_dict (*obj_dict*)

static from_str (*obj_str*)

class `galaxy.visualization.genomes.Genomes(app)`
 Bases: object

Provides information about available genome data and methods for manipulating that data.

chroms (*trans, dbkey=None, num=None, chrom=None, low=None*)
 Returns a naturally sorted list of chroms/contigs for a given dbkey. Use either chrom or low to specify the starting chrom in the return list.

get_build (*dbkey*)
 Returns build for the given key.

get_dbkeys (*trans, chrom_info=False, **kwd*)
 Returns all known dbkeys. If chrom_info is True, only dbkeys with chromosome lengths are returned.

has_reference_data (*dbkey, dbkey_owner=None*)
 Returns true if there is reference data for the specified dbkey. If dbkey is custom, dbkey_owner is needed to determine if there is reference data.

reference (*trans, dbkey, chrom, low, high*)
 Return reference data for a build.

`galaxy.visualization.genomes.decode_dbkey(dbkey)`
 Decodes dbkey and returns tuple (username, dbkey)

Subpackages

data_providers Package

data_providers Package Galaxy visualization/visual analysis data providers.

basic Module

```
class galaxy.visualization.data_providers.basic.BaseDataProvider (converted_dataset=None,
                                                                    origi-
                                                                    nal_dataset=None,
                                                                    dependen-
                                                                    cies=None,  er-
                                                                    ror_max_vals='Only
                                                                    the first %i
                                                                    values are
                                                                    returned.')
```

Bases: object

Base class for data providers. Data providers (a) read and package data from datasets; and (b) write subsets of data to new datasets.

get_data (*chrom, start, end, start_val=0, max_vals=9223372036854775807, **kwargs*)
Returns data as specified by kwargs. start_val is the first element to return and max_vals indicates the number of values to return.

Return value must be a dictionary with the following attributes: dataset_type, data

get_iterator (***kwargs*)
Returns an iterator that provides data in the region chrom:start-end

has_data (***kwargs*)
Returns true if dataset has data in the specified genome window, false otherwise.

process_data (*iterator, start_val=0, max_vals=None, **kwargs*)
Process data from an iterator to a format that can be provided to client.

write_data_to_file (*filename, **kwargs*)
Write data in region defined by chrom, start, and end to a file.

```
class galaxy.visualization.data_providers.basic.ColumnDataProvider (original_dataset,
                                                                    max_lines_returned=30000)
Bases: galaxy.visualization.data_providers.basic.BaseDataProvider
```

Data provider for columnar data

MAX_LINES_RETURNED = 30000

get_data (*columns=None, start_val=0, max_vals=None, skip_comments=True, **kwargs*)
Returns data from specified columns in dataset. Format is list of lists where each list is a line of data.

genome Module

registry Module

Subpackages

phyloviz Package

phyloviz Package Data providers code for PhyloViz

```
class galaxy.visualization.data_providers.phyloviz.PhylovizDataProvider (original_dataset=None)
    Bases: galaxy.visualization.data_providers.basic.BaseDataProvider
```

dataset_type = 'phylo'

get_data (*tree_index=0*)

Returns trees. Trees are actually an array of JsonDicts. It's usually one tree, except in the case of Nexus

baseparser Module

```
class galaxy.visualization.data_providers.phyloviz.baseparser.Base_Parser
    Bases: object
```

Base parsers contain all the methods to handle phylogeny tree creation and converting the data to json that all parsers should have

parseFile (*filePath*)

Base method that all phylogeny file parser should have

toJson (*jsonDict*)

Convenience method to get a json string from a python json dict

```
class galaxy.visualization.data_providers.phyloviz.baseparser.Node (nodeName,
                                                                **kwargs)
```

Bases: object

Node class of PhyloTree, which represents a CLAUDE in a phylogenetic tree

addChildNode (*child*)

Adds a child node to the current node

addChildrenToJson (*jsonDict*)

Needs a special method to addChildren, such that the key does not appear in the Jsondict when the children is empty this requirement is due to the layout algorithm used by d3 layout for hiding subtree

addMiscToJson (*jsonDict*)

Adds other misc attributes to json if they are present

toJson ()

Converts the data in the node to a dict representation of json

```
class galaxy.visualization.data_providers.phyloviz.baseparser.PhyloTree
    Bases: object
```

Standardized python based class to represent the phylogenetic tree parsed from different phylogenetic file formats.

addAttributesToRoot (*attrDict*)

Adds attributes to root, but first we put it in a temp store and bind it with root when .toJson is called

addRoot (*root*)

Creates a root for phyloTree

generateJsonableDict ()

Changes itself into a dictionary by recurssively calling the toJson on all its nodes. Think of it as a dict in an array of dict in an array of dict and so on...

makeNode (*nodeName, **kwargs*)

Called to make a node within PhyloTree, arbitrary kwargs can be passed to annotate nodes Tracks the number of nodes via internally incremented id

newickparser Module

class `galaxy.visualization.data_providers.phyloviz.newickparser.Newick_Parser`
Bases: `galaxy.visualization.data_providers.phyloviz.baseparser.Base_Parser`

For parsing trees stored in the newick format (.nhx) It is necessarily more complex because this parser is later extended by Nexus for parsing newick as well..

cleanNewickString (*rawNewick*)
removing semi colon, and illegal json characters (',') and white spaces

parseData (*newickString*)
To be called on a newickString directly to parse it. Returns: jsonableDict

parseFile (*filePath*)
Parses a newick file to obtain the string inside. Returns: jsonableDict

parseNode (*string, depth*)
Recursive method for parsing newick string, works by stripping down the string into substring of newick contained with brackets, which is used to call itself.

Eg ... (A, B, (D, E)C, F, G) ...

We will make the preceeding nodes first A, B, then the internal node C, its children D, E, and finally the succeeding nodes F, G

nexusparser Module

class `galaxy.visualization.data_providers.phyloviz.nexusparser.Nexus_Parser`
Bases: `galaxy.visualization.data_providers.phyloviz.newickparser.Newick_Parser`

checkComments (*line*)
Check to see if the line/lines is a comment.

parseFile (*filePath*)
passes a file and extracts its Nexus content.

parseNexus (*filename*)
Nexus data is stored in blocks between a line starting with begin and another line starting with end; Comments inside square brackets are to be ignored, For more information: http://wiki.christophchamp.com/index.php/NEXUS_file_format Nexus can store multiple trees

splitLinebyWhitespaces (*line*)
replace tabs and write spaces to a single write space, so we can properly split it.

phyloxmlparser Module

class `galaxy.visualization.data_providers.phyloviz.phyloxmlparser.Phyloxml_Parser`
Bases: `galaxy.visualization.data_providers.phyloviz.baseparser.Base_Parser`

Parses a phyloxml file into a json file that will be passed to PhyloViz for display

cleanTag (*tagString*)

parseFile (*filePath*)
passes a file and extracts its Phylogeny Tree content.

parseNode (*node, depth*)
Parses any node within a phyloxml tree and looks out for claude, which signals the creation of nodes - internal OR leaf

genome Package

genome Package Code for Galaxy genome visualizations.

visual_analytics Module

tracks Package

ttracks Package Summary.py required to be in this module due to pickling.

summary Module

web Package

web Package The Galaxy web application framework

buildapp Module For backward compatibility only, pulls app_factor from galaxy.webapps.main

form_builder Module Classes for generating HTML forms

class galaxy.web.form_builder.**AddressField**(name, user=None, value=None, params=None)

Bases: *galaxy.web.form_builder.BaseField*

static fields()

get_html(disabled=False)

class galaxy.web.form_builder.**BaseField**

Bases: object

get_disabled_str(disabled=False)

get_html(prefix='')

Returns the html widget corresponding to the parameter

class galaxy.web.form_builder.**CheckboxField**(name, checked=None, fresh_on_change=False, fresh_on_change_values=None)

Bases: *galaxy.web.form_builder.BaseField*

A checkbox (boolean input)

```
>>> print CheckboxField( "foo" ).get_html()
<input type="checkbox" id="foo" name="foo" value="true"><input type="hidden" name="foo" value="t
>>> print CheckboxField( "bar", checked="yes" ).get_html()
<input type="checkbox" id="bar" name="bar" value="true" checked="checked"><input type="hidden" n
```

get_html(prefix='', disabled=False)

static is_checked(value)

set_checked(value)

```
class galaxy.web.form_builder.DrillDownField(name, multiple=None, display=None, re-
                                           fresh_on_change=False, options=[], value=[],
                                           refresh_on_change_values=[])
```

Bases: `galaxy.web.form_builder.BaseField`

A hierarchical select field, which allows users to ‘drill down’ a tree-like set of options.

```
>>> t = DrillDownField( "foo", multiple=True, display="checkbox", options=[{'name': 'Heading 1',
>>> print t.get_html()
<div class="form-row drilldown-container" id="drilldown--666f6f">
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--666f6f-68656164696e6731-
<input type="checkbox" name="foo" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--666f6f-68656164696e6731-container" style="float: left
<div class="form-row-input">
<input type="checkbox" name="foo" value="option1" >Option 1
</div>
<div class="form-row-input">
<input type="checkbox" name="foo" value="option2" >Option 2
</div>
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--666f6f-68656164696e6731-
<input type="checkbox" name="foo" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--666f6f-68656164696e6731-68656164696e6731-container" s
<div class="form-row-input">
<input type="checkbox" name="foo" value="option3" >Option 3
</div>
<div class="form-row-input">
<input type="checkbox" name="foo" value="option4" >Option 4
</div>
</div>
</div>
</div>
</div>
<div class="form-row-input">
<input type="checkbox" name="foo" value="option5" >Option 5
</div>
</div>
>>> t = DrillDownField( "foo", multiple=False, display="radio", options=[{'name': 'Heading 1',
>>> print t.get_html()
<div class="form-row drilldown-container" id="drilldown--666f6f">
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--666f6f-68656164696e6731-
<input type="radio" name="foo" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--666f6f-68656164696e6731-container" style="float: left
<div class="form-row-input">
<input type="radio" name="foo" value="option1" >Option 1
</div>
<div class="form-row-input">
<input type="radio" name="foo" value="option2" >Option 2
</div>
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--666f6f-68656164696e6731-
<input type="radio" name="foo" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--666f6f-68656164696e6731-68656164696e6731-container" s
<div class="form-row-input">
<input type="radio" name="foo" value="option3" >Option 3
</div>
<div class="form-row-input">
```

A file upload input.

```
get_html (prefix='', disabled=False)
```

class galaxy.web.form_builder.PasswordField(*name*, *size=None*, *value=None*)

Bases: *galaxy.web.form_builder.BaseField*

A password input box. text appears as “***”

```
>>> print PasswordField( "foo" ).get_html()
<input type="password" name="foo" size="10" value="">
>>> print PasswordField( "bins", size=4, value="default" ).get_html()
<input type="password" name="bins" size="4" value="default">
```

get_html (*prefix=''*, *disabled=False*)

set_size (*size*)

class galaxy.web.form_builder.SelectField(*name*, *multiple=None*, *display=None*,
refresh_on_change=False, *re-*
fresh_on_change_values=None, *size=None*)

Bases: *galaxy.web.form_builder.BaseField*

A select field.

```
>>> t = SelectField( "foo", multiple=True )
>>> t.add_option( "tuti", 1 )
>>> t.add_option( "fruity", "x" )
>>> print t.get_html()
<select name="foo" multiple>
<option value="1">tuti</option>
<option value="x">fruity</option>
</select>
```

```
>>> t = SelectField( "bar" )
>>> t.add_option( "automatic", 3 )
>>> t.add_option( "bazooty", 4, selected=True )
>>> print t.get_html()
<select name="bar" last_selected_value="4">
<option value="3">automatic</option>
<option value="4" selected>bazooty</option>
</select>
```

```
>>> t = SelectField( "foo", display="radio" )
>>> t.add_option( "tuti", 1 )
>>> t.add_option( "fruity", "x" )
>>> print t.get_html()
<div><input type="radio" name="foo" value="1" id="foo|1"><label class="inline" for="foo|1">tuti<
<div><input type="radio" name="foo" value="x" id="foo|x"><label class="inline" for="foo|x">fruit
```

```
>>> t = SelectField( "bar", multiple=True, display="checkboxes" )
>>> t.add_option( "automatic", 3 )
>>> t.add_option( "bazooty", 4, selected=True )
>>> print t.get_html()
<div class="checkUncheckAllPlaceholder" checkbox_name="bar"></div>
<div><input type="checkbox" name="bar" value="3" id="bar|3"><label class="inline" for="bar|3">au
<div><input type="checkbox" name="bar" value="4" id="bar|4" checked='checked'><label class="inli
```

add_option (*text*, *value*, *selected=False*)

```

get_html (prefix='', disabled=False)
get_html_checkboxes (prefix='', disabled=False)
get_html_default (prefix='', disabled=False)
get_html_radio (prefix='', disabled=False)
get_selected (return_label=False, return_value=False, multi=False)
    Return the currently selected option's label, value or both as a tuple. For multi-select lists, a list is returned.
to_dict ()

```

```

class galaxy.web.form_builder.SwitchingSelectField (delegate_fields, default_field=None)
    Bases: galaxy.web.form_builder.BaseField

```

```

get_html (prefix='', disabled=False)
primary_field

```

```

class galaxy.web.form_builder.TextArea (name, size=None, value=None)
    Bases: galaxy.web.form_builder.BaseField

    A standard text area box.

```

```

>>> print TextArea( "foo" ).get_html()
<textarea name="foo" rows="5" cols="25"></textarea>
>>> print TextArea( "bins", size="4x5", value="default" ).get_html()
<textarea name="bins" rows="4" cols="5">default</textarea>

```

```

get_html (prefix='', disabled=False)
set_size (rows, cols)

```

```

class galaxy.web.form_builder.TextField (name, size=None, value=None)
    Bases: galaxy.web.form_builder.BaseField

    A standard text input box.

```

```

>>> print TextField( "foo" ).get_html()
<input type="text" name="foo" size="10" value="">
>>> print TextField( "bins", size=4, value="default" ).get_html()
<input type="text" name="bins" size="4" value="default">

```

```

get_html (prefix='', disabled=False)
set_size (size)

```

```

class galaxy.web.form_builder.WorkflowField (name, user=None, value=None, params=None)
    Bases: galaxy.web.form_builder.BaseField

```

```

get_html (disabled=False)

```

```

class galaxy.web.form_builder.WorkflowMappingField (name, user=None, value=None,
                                                    params=None, **kwd)
    Bases: galaxy.web.form_builder.BaseField

```

```

get_display_text ()
get_html (disabled=False)

```

```
galaxy.web.form_builder.build_select_field(trans, objs, label_attr, select_field_name, initial_value='none', selected_value='none', refresh_on_change=False, multiple=False, display=None, size=None)
```

Build a SelectField given a set of objects. The received params are:

- **objs**: the set of objects used to populate the option list
- **label_attr**: the attribute of each obj (e.g., name, email, etc) whose value is used to populate each option label.
 - If the string 'self' is passed as label_attr, each obj in objs is assumed to be a string, so the obj itself is used
- **select_field_name**: the name of the SelectField
- **initial_value**: the value of the first option in the SelectField - allows for an option telling the user to select something
- **selected_value**: the value of the currently selected option
- **refresh_on_change**: True if the SelectField should perform a refresh_on_change

```
galaxy.web.form_builder.get_suite()
```

Get unittest suite for this module

params Module Mixins for parsing web form and API parameters

```
class galaxy.web.params.BaseParamParser
    Bases: object

    get_params (kwargs)

class galaxy.web.params.QuotaParamParser
    Bases: galaxy.web.params.BaseParamParser

    get_quota_params (kwargs)
```

Subpackages

base Package

controller Module Contains functionality needed in every web interface

```
class galaxy.web.base.controller.BaseAPIController (app)
    Bases: galaxy.web.base.controller.BaseController

    get_object (trans, id, class_name, check_ownership=False, check_accessible=False, deleted=None)

    not_implemented (trans, **kwd)

    validate_in_users_and_groups (trans, payload)
        For convenience, in_users and in_groups can be encoded IDs or emails/group names in the API.

class galaxy.web.base.controller.BaseController (app)
    Bases: object

    Base class for Galaxy web application controllers.

    encode_all_ids (trans, rval, recursive=False)
        Encodes all integer values in the dict rval whose keys are 'id' or end with '_id'

        It might be useful to turn this in to a decorator
```

```

get_class (class_name)
    Returns the class object that a string denotes. Without this method, we'd have to do eval(<class_name>).

get_group (trans, id, check_ownership=False, check_accessible=False, deleted=None)

get_object (trans, id, class_name, check_ownership=False, check_accessible=False,
              deleted=None)
    Convenience method to get a model object with the specified checks.

get_role (trans, id, check_ownership=False, check_accessible=False, deleted=None)

get_toolbox ()
    Returns the application toolbox

get_user (trans, id, check_ownership=False, check_accessible=False, deleted=None)

validate_and_sanitize_basestring (key, val)

validate_and_sanitize_basestring_list (key, val)

validate_boolean (key, val)

class galaxy.web.base.controller.BaseUIController (app)
    Bases: galaxy.web.base.controller.BaseController

    get_object (trans, id, class_name, check_ownership=False, check_accessible=False,
              deleted=None)

exception galaxy.web.base.controller.ControllerUnavailable
    Bases: exceptions.Exception

    Deprecated: BaseController used to be available under the name Root

class galaxy.web.base.controller.CreatesApiKeysMixin
    Mixing centralizing logic for creating API keys for user objects.

    Deprecated - please use api_keys.ApiKeyManager for new development.

    create_api_key (trans, user)

class galaxy.web.base.controller.CreatesUsersMixin
    Mixin centralizing logic for user creation between web and API controller.

    Web controller handles additional features such e-mail subscription, activation, user forms, etc.... API created
    users are much more vanilla for the time being.

    create_user (trans, email, username, password)

class galaxy.web.base.controller.Datatype (extension, dtype, type_extension, mimetype, display_in_upload)
    Bases: object

    Used for storing in-memory list of datatypes currently in the datatypes registry.

class galaxy.web.base.controller.ExportsHistoryMixin

    queue_history_export (trans, history, gzip=True, include_hidden=False, include_deleted=False)

    serve_ready_history_export (trans, jeha)

class galaxy.web.base.controller.ImportsHistoryMixin

    queue_history_import (trans, archive_type, archive_source)

```

`galaxy.web.base.controller.Root`
alias of `BaseController`

class `galaxy.web.base.controller.SharableItemSecurityMixin`

Mixin for handling security for sharable items.

security_check (*trans, item, check_ownership=False, check_accessible=False*)

Security checks for an item: checks if (a) user owns item or (b) item is accessible to user.

class `galaxy.web.base.controller.SharableMixin`

Mixin for a controller that manages an item that can be shared.

create_item_slug (*sa_session, item*)

Create/set item slug. Slug is unique among user's importable items for item's class. Returns true if item's slug was set/changed; false otherwise.

display_by_username_and_slug (*trans, username, slug*)

Display item by username and slug.

get_item (*trans, id*)

Return item based on id.

get_item_content_async (*trans, *args, **kwargs*)

Returns item content in HTML format.

get_name_and_link_async (*trans, *args, **kwargs*)

Returns item's name and link.

set_public_username (*trans, *args, **kwargs*)

Set user's public username and delegate to sharing()

set_slug_async (*trans, *args, **kwargs*)

share (*trans, *args, **kwargs*)

Handle sharing an item with a particular user.

sharing (*trans, *args, **kwargs*)

Handle item sharing.

class `galaxy.web.base.controller.UsesExtendedMetadataMixin`

Bases: `galaxy.web.base.controller.SharableItemSecurityMixin`

Mixin for getting and setting item extended metadata.

create_extended_metadata (*trans, extmeta*)

Create/index an extended metadata object. The returned object is not associated with any items

delete_extended_metadata (*trans, item*)

get_item_extended_metadata_obj (*trans, item*)

Given an item object (such as a LibraryDatasetDatasetAssociation), find the object of the associated extended metadata

set_item_extended_metadata_obj (*trans, item, extmeta_obj, check_writable=False*)

unset_item_extended_metadata_obj (*trans, item, check_writable=False*)

class `galaxy.web.base.controller.UsesFormDefinitionsMixin`

Mixin for controllers that use Galaxy form objects.

add_template (*trans, cntrller, item_type, form_type, **kwd*)

build_form_id_select_field (*trans, forms, selected_value='none'*)

clean_field_contents (*widgets, **kwd*)


```

delete_template (trans, cntrller, item_type, form_type, **kwd)
edit_template (trans, cntrller, item_type, form_type, **kwd)
edit_template_info (trans, cntrller, item_type, form_type, **kwd)
field_param_values_ok (widget_name, widget_type, **kwd)
get_all_forms (trans, all_versions=False, filter=None, form_type='All')
    Return all the latest forms from the form_definition_current table if all_versions is set to True. Otherwise
    return all the versions of all the forms from the form_definition table.
get_all_forms_by_type (trans, cntrller, form_type)
get_form_values (trans, user, form_definition, **kwd)
    Returns the name:value dictionary containing all the form values
get_item_and_stuff (trans, item_type, **kwd)
populate_widgets_from_kwd (trans, widgets, **kwd)
save_widget_field (trans, field_obj, widget_name, **kwd)
widget_fields_have_contents (widgets)

```

class `galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin`
 Mixin for controllers that use HistoryDatasetAssociation objects.

```

check_dataset_state (trans, dataset)
    Returns a message if dataset is not ready to be used in visualization.
get_data (dataset, preview=True)
    Gets a dataset's data.
get_dataset (trans, dataset_id, check_ownership=True, check_accessible=False,
               check_state=True)
    Get an HDA object by id performing security checks using the current transaction.
get_display_apps (trans, hda)
get_hda_dict (trans, hda)
    Return full details of this HDA in dictionary form.
get_hda_dict_with_error (trans, hda=None, history_id=None, id=None, error_msg='Error')
get_hda_job (hda)
get_hda_list (trans, hda_ids, check_ownership=True, check_accessible=False, check_state=True)
    Returns one or more datasets in a list.

    If a dataset is not found or is inaccessible to trans.user, add None in its place in the list.
get_history_dataset_association (trans, history, dataset_id, check_ownership=True,
                                   check_accessible=False, check_state=False)
    Get a HistoryDatasetAssociation from the database by id, verifying ownership.
get_history_dataset_association_from_ids (trans, id, history_id)
get_inaccessible_hda_dict (trans, hda)
get_old_display_applications (trans, hda)
set_hda_from_dict (trans, hda, new_data)
    Changes HDA data using the given dictionary new_data.
stop_hda_creating_job (hda)
    Stops an HDA's creating job if all the job's other outputs are deleted.

```

class `galaxy.web.base.controller.UsesHistoryMixin`

Bases: `galaxy.web.base.controller.SharableItemSecurityMixin`

Mixin for controllers that use History objects.

get_hda_state_counts (*trans, history, include_deleted=False, include_hidden=False*)

Returns a dictionary with state counts for history's HDAs. Key is a dataset state, value is the number of states in that count.

get_hda_summary_dicts (*trans, history*)

Returns a list of dictionaries containing summary information for each HDA in the given history.

get_history (*trans, id, check_ownership=True, check_accessible=False, deleted=None*)

Get a History from the database by id, verifying ownership.

get_history_datasets (*trans, history, show_deleted=False, show_hidden=False, show_purged=False*)

Returns history's datasets.

get_history_dict (*trans, history, hda_dictionaries=None*)

Returns history data in the form of a dictionary.

get_user_histories (*trans, user=None, include_deleted=False, only_deleted=False*)

Get all the histories for a given user (defaulting to *trans.user*) ordered by update time and filtered on whether they've been deleted.

set_history_from_dict (*trans, history, new_data*)

Changes history data using the given dictionary *new_data*.

class `galaxy.web.base.controller.UsesLibraryMixin`

get_library (*trans, id, check_ownership=False, check_accessible=True*)

class `galaxy.web.base.controller.UsesLibraryMixinItems`

Bases: `galaxy.web.base.controller.SharableItemSecurityMixin`

can_current_user_add_to_library_item (*trans, item*)

check_user_can_add_to_library_item (*trans, item, check_accessible=True*)

Raise exception if user cannot add to the specified library item (i.e. Folder). Can set *check_accessible* to False if folder was loaded with this check.

copy_hda_to_library_folder (*trans, hda, library_folder, roles=None, ldda_message=''*)

get_library_dataset (*trans, id, check_ownership=False, check_accessible=True*)

get_library_dataset_dataset_association (*trans, id, check_ownership=False, check_accessible=True*)

get_library_folder (*trans, id, check_ownership=False, check_accessible=True*)

class `galaxy.web.base.controller.UsesQuotaMixin`

Bases: `object`

get_quota (*trans, id, check_ownership=False, check_accessible=False, deleted=None*)

class `galaxy.web.base.controller.UsesStoredWorkflowMixin`

Bases: `galaxy.web.base.controller.SharableItemSecurityMixin`,
`galaxy.model.item_attrs.UsesAnnotations`

Mixin for controllers that use StoredWorkflow objects.

get_stored_workflow (*trans, id, check_ownership=True, check_accessible=False*)

Get a StoredWorkflow from the database by id, verifying ownership.

```

get_stored_workflow_steps (trans, stored_workflow)
    Restores states for a stored workflow's steps.

class galaxy.web.base.controller.UsesTagsMixin
    Bases: galaxy.web.base.controller.SharableItemSecurityMixin

    get_tag_handler (trans)

    get_user_tags_used (trans, user=None)
        Return a list of distinct 'user_tname:user_value' strings that the given user has used.

        user defaults to trans.user. Returns an empty list if no user is given and trans.user is anonymous.

    set_tags_from_list (trans, item, new_tags_list, user=None)

class galaxy.web.base.controller.UsesVisualizationMixin
    Bases: galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin,
           galaxy.web.base.controller.UsesLibraryMixinItems

    Mixin for controllers that use Visualization objects.

    add_visualization_revision (trans, visualization, config, title, dbkey)
        Adds a new VisualizationRevision to the given visualization with the given parameters and set its parent
        visualization's latest_revision to the new revision.

    create_visualization (trans, type, title='Untitled Visualization', slug=None, dbkey=None, an-
                          notation=None, config={}, save=True)
        Create visualiation and first revision.

    get_hda_or_ldda (trans, hda_ldda, dataset_id)
        Returns either HDA or LDDA for hda/ldda and id combination.

    get_new_track_config (trans, dataset)
        Returns track configuration dict for a dataset.

    get_published_visualizations (trans, exclude_user=None, order_by=None,
                                  query_only=False)
        Return query or query results for published visualizations optionally excluding the user in exclude_user.

        Set order_by to a column or list of columns to change the order returned. Defaults to DE-
FAULT_ORDER_BY. Set query_only to return just the query for further filtering or processing.

    get_tool_def (trans, hda)
        Returns definition of an interactive tool for an HDA.

    get_visualization (trans, id, check_ownership=True, check_accessible=False)
        Get a Visualization from the database by id, verifying ownership.

    get_visualization_config (trans, visualization)
        Returns a visualization's configuration. Only works for trackster visualizations right now.

    get_visualization_dict (visualization)
        Return a set of detailed attributes for a visualization in dictionary form. The visualization's latest_revision
        is returned in its own sub-dictionary. NOTE: that encoding ids isn't done here should happen at the caller
        level.

    get_visualization_revision_dict (revision)
        Return a set of detailed attributes for a visualization in dictionary form. NOTE: that encoding ids isn't
        done here should happen at the caller level.

    get_visualization_summary_dict (visualization)
        Return a set of summary attributes for a visualization in dictionary form. NOTE: that encoding ids isn't
        done here should happen at the caller level.

```

get_visualizations_by_user (*trans, user, order_by=None, query_only=False*)

Return query or query results of visualizations filtered by a user.

Set *order_by* to a column or list of columns to change the order returned. Defaults to *DEFAULT_ORDER_BY*. Set *query_only* to return just the query for further filtering or processing.

get_visualizations_shared_with_user (*trans, user, order_by=None, query_only=False*)

Return query or query results for visualizations shared with the given user.

Set *order_by* to a column or list of columns to change the order returned. Defaults to *DEFAULT_ORDER_BY*. Set *query_only* to return just the query for further filtering or processing.

import_visualization (*trans, id, user=None*)

Copy the visualization with the given id and associate the copy with the given user (defaults to *trans.user*).

Raises *ItemAccessibilityException* if *user* is not passed and the current user is anonymous, and if the visualization is not *importable*. Raises *ItemDeletionException* if the visualization has been deleted.

save_visualization (*trans, config, type, id=None, title=None, dbkey=None, slug=None, annotation=None*)

viz_types = ['trackster']

`galaxy.web.base.controller.sort_by_attr` (*seq, attr*)

Sort the sequence of objects by object's attribute Arguments: *seq* - the list or any sequence (including immutable one) of objects to sort. *attr* - the name of attribute to sort by

Subpackages

controllers Package

admin Module

class `galaxy.web.base.controllers.admin.Admin`

Bases: `object`

center (*trans, *args, **kwargs*)

create_group (*trans, *args, **kwargs*)

create_new_user (*trans, *args, **kwargs*)

create_role (*trans, *args, **kwargs*)

delete_operation = `None`

group_list_grid = `None`

groups (*trans, *args, **kwargs*)

index (*trans, *args, **kwargs*)

job_info (*trans, *args, **kwargs*)

jobs (*trans, *args, **kwargs*)

manage_roles_and_groups_for_user (*trans, *args, **kwargs*)

manage_users_and_groups_for_role (*trans, *args, **kwargs*)

manage_users_and_roles_for_group (*trans, *args, **kwargs*)

mark_group_deleted (*trans, *args, **kwargs*)

mark_role_deleted (*trans, *args, **kwargs*)

```

mark_user_deleted (trans, *args, **kwargs)
memdump (trans, *args, **kwargs)
name_autocomplete_data (trans, *args, **kwargs)
    Return autocomplete data for user emails
package_tool (trans, *args, **kwargs)
purge_group (trans, *args, **kwargs)
purge_operation = None
purge_role (trans, *args, **kwargs)
purge_user (trans, *args, **kwargs)
quota_list_grid = None
reload_tool (trans, *args, **kwargs)
rename_group (trans, *args, **kwargs)
rename_role (trans, *args, **kwargs)
repository_list_grid = None
reset_user_password (trans, *args, **kwargs)
role_list_grid = None
roles (trans, *args, **kwargs)
tool_version_list_grid = None
tool_versions (trans, *args, **kwargs)
undetele_group (trans, *args, **kwargs)
undetele_operation = None
undetele_role (trans, *args, **kwargs)
undetele_user (trans, *args, **kwargs)
user_list_grid = None
users (trans, *args, **kwargs)
galaxy.web.base.controllers.admin.get_group (trans, id)
    Get a Group from the database by id.
galaxy.web.base.controllers.admin.get_quota (trans, id)
    Get a Quota from the database by id.
galaxy.web.base.controllers.admin.get_role (trans, id)
    Get a Role from the database by id.
galaxy.web.base.controllers.admin.get_user (trans, user_id)
    Get a User from the database by id.
galaxy.web.base.controllers.admin.get_user_by_username (trans, username)
    Get a user from the database by username

```

framework Package

framework Package Galaxy web application framework

base Module A simple WSGI application/framework.

class `galaxy.web.framework.base.DefaultWebTransaction` (*environ*)

Bases: `object`

Wraps the state of a single web transaction (request/response cycle).

TODO: Provide hooks to allow application specific state to be included in here.

session

Property that replaces itself with a calculated value the first time it is used.

```
class galaxy.web.framework.base.FieldStorage (fp=None, headers=None,
outerboundary='', environ={ 'CELERY_LOG_REDIRECT_LEVEL':
'WARNING', '_MP_FORK_LOGFILE_':
'/home/docs/log/celery_proc.log',
'NEW_RELIC_CONFIG_FILE':
'/home/docs/newrelic.ini', 'CEL-
ERY_LOG_REDIRECT': '1', 'LOG-
NAME': 'docs', 'USER': 'docs', 'PATH':
'/home/docs/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
HOME': '/home/docs', 'PS1':
'(docs)', 'TERM': 'linux', 'SHELL':
'/bin/bash', 'TZ': 'America/Chicago',
'_MP_FORK_LOGFORMAT_': '[(asc-
time)s: %(levelname)s/(processName)s]
%(message)s', 'SHLVL': '1', 'SUPER-
VISOR_ENABLED': '1', 'EDITOR':
'vim', 'DJANGO_PROJECT_DIR':
'/home/docs/checkouts/readthedocs.org/readthedocs',
'SUDO_USER': 'root',
'CELERY_LOG_FILE':
'/home/docs/log/celery_proc.log',
'USERNAME': 'docs', 'READTHE-
DOCS': 'True', 'SUDO_UID': '0',
'VIRTUAL_ENV': '/home/docs', 'SU-
PERVISOR_PROCESS_NAME': 'cel-
ery', 'SUPERVISOR_SERVER_URL':
'unix:///home/docs/run/supervisord-
docs.sock', '_': '/home/docs/bin/supervisord',
'SUDO_COMMAND': '/bin/bash -c
/home/docs/bin/supervisord -nodaemon',
'SUDO_GID': '0', 'CELERY_LOADER':
'djcelery.loaders.DjangoLoader',
'OLDPWD': '/home/docs', 'PWD':
'/home/docs/checkouts/readthedocs.org/readthedocs',
'_MP_FORK_LOGLEVEL_': '20', 'MAIL':
'/var/mail/docs', 'CELERY_LOG_LEVEL':
'20', 'SUPERVISOR_GROUP_NAME':
'celery'}, keep_blank_values=0,
strict_parsing=0)
```

Bases: `cgi.FieldStorage`

make_file (*binary=None*)

read_lines ()

class `galaxy.web.framework.base.LazyProperty` (*func*)

Bases: object

Property that replaces itself with a calculated value the first time it is used.

class `galaxy.web.framework.base.Request` (*environ*)

Bases: `webob.Request`

Encapsulates an HTTP request.

base

Property that replaces itself with a calculated value the first time it is used.

browser_url

Property that replaces itself with a calculated value the first time it is used.

cookies

Property that replaces itself with a calculated value the first time it is used.

path

Property that replaces itself with a calculated value the first time it is used.

protocol

Descriptor that delegates a property to a key in the `environ` member of the associated object (provides property style access to keys in the WSGI environment)

remote_host

Property that replaces itself with a calculated value the first time it is used.

remote_hostname

Property that replaces itself with a calculated value the first time it is used.

remote_port

Descriptor that delegates a property to a key in the `environ` member of the associated object (provides property style access to keys in the WSGI environment)

class `galaxy.web.framework.base.Response`

Bases: object

Describes an HTTP response. Currently very simple since the actual body of the request is handled separately.

get_content_type ()

send_redirect (*url*)

Send an HTTP redirect response to (target *url*)

set_content_type (*type*)

Sets the Content-Type header

wsgi_headeritems ()

Return headers in format appropriate for WSGI *start_response*

wsgi_status ()

Return status line in format appropriate for WSGI *start_response*

class `galaxy.web.framework.base.WSGIEnvironmentProperty` (*key*, *default*='')

Bases: object

Descriptor that delegates a property to a key in the `environ` member of the associated object (provides property style access to keys in the WSGI environment)

class `galaxy.web.framework.base.WebApplication`

Bases: object

A simple web application which maps requests to objects using routes, and to methods on those objects in the CherryPy style. Thus simple argument mapping in the CherryPy style occurs automatically, but more complicated encoding of arguments in the PATH_INFO can be performed with routes.

add_api_controller (*controller_name, controller*)

add_route (*route, **kwargs*)

Add a route to match a URL with a method. Accepts all keyword arguments of *routes.Mapper.connect*. Every route should result in at least a controller value which corresponds to one of the objects added with *add_controller*. It optionally may yield an *action* argument which will be used to locate the method to call on the controller. Additional arguments will be passed to the method as keyword args.

add_ui_controller (*controller_name, controller*)

Add a controller class to this application. A controller class has methods which handle web requests. To connect a URL to a controller's method use *add_route*.

finalize_config ()

Call when application is completely configured and ready to serve requests

handle_controller_exception (*e, trans, **kwargs*)

Allow handling of exceptions raised in controller methods.

handle_request (*environ, start_response*)

make_body_iterable (*trans, body*)

set_transaction_factory (*transaction_factory*)

Use the callable *transaction_factory* to create the transaction which will be passed to requests.

trace (***fields*)

`galaxy.web.framework.base.flatten` (*seq*)

Flatten a possible nested set of iterables

`galaxy.web.framework.base.iterate_file` (*file*)

Progressively return chunks from *file*.

`galaxy.web.framework.base.lazy_property`

alias of *LazyProperty*

`galaxy.web.framework.base.send_file` (*start_response, trans, body*)

openid_manager Module Manage the OpenID consumer and related data stores.

class `galaxy.web.framework.openid_manager.OpenIDManager` (*cache_path*)

Bases: object

CANCEL = 'cancel'

FAILURE = 'failure'

SETUP_NEEDED = 'setup_needed'

SUCCESS = 'success'

add_sreg (*trans, request, required=None, optional=None*)

get_consumer (*trans*)

get_session (*trans*)

get_sreg (*info*)

persist_session (*trans, oidconsumer*)

`galaxy.web.framework.openid_manager.oidlog(message, level=0)`

Subpackages

helpers Package

helpers Package Galaxy web framework helpers

`galaxy.web.framework.helpers.css(*args)`

Take a list of stylesheet names (no extension) and return appropriate string of link tags.

Cache-bust with time that server started running on

`galaxy.web.framework.helpers.iff(a, b, c)`

Ternary shortcut

`galaxy.web.framework.helpers.is_true(val)`

Returns true if input is a boolean and true or is a string and looks like a true value.

`galaxy.web.framework.helpers.js(*args)`

Take a prefix and list of javascript names and return appropriate string of script tags.

`galaxy.web.framework.helpers.js_helper(prefix, *args)`

Take a prefix and list of javascript names and return appropriate string of script tags.

Cache-bust with time that server started running on

`galaxy.web.framework.helpers.md5(s)`

Return hex encoded md5 hash of string s

`galaxy.web.framework.helpers.templates(*args)`

Take a list of template names (no extension) and return appropriate string of script tags.

`galaxy.web.framework.helpers.time_ago(x)`

Convert a datetime to a string.

`galaxy.web.framework.helpers.to_unicode(a_string)`

Convert a string to unicode in utf-8 format; if string is already unicode, does nothing because string's encoding cannot be determined by introspection.

`galaxy.web.framework.helpers.truncate(content, length=100, suffix='...')`

Smart string truncation

grids Module

```
class galaxy.web.framework.helpers.grids.BooleanColumn(label,
                                                         key=None,
                                                         model_class=None,
                                                         method=None,
                                                         format=None,
                                                         link=None,
                                                         attach_popup=False,
                                                         visible=True,
                                                         nowrap=False,
                                                         filterable=None,
                                                         sortable=True,
                                                         label_id_prefix=None,
                                                         in-bound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

`get_single_filter(user, a_filter)`

`sort(trans, query, ascending, column_name=None)`

Sort query using this column.

```
class galaxy.web.framework.helpers.grids.CommunityRatingColumn(label, key=None,
                                                                model_class=None,
                                                                method=None,
                                                                format=None,
                                                                link=None, attach_popup=False,
                                                                visible=True,
                                                                nowrap=False,
                                                                filterable=None,
                                                                sortable=True, label_id_prefix=None,
                                                                inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`,
`galaxy.model.item_attrs.UsesItemRatings`

Column that displays community ratings for an item.

get_value (*trans, grid, item*)

sort (*trans, query, ascending, column_name=None*)

```
class galaxy.web.framework.helpers.grids.CommunityTagsColumn(col_name, key,
                                                                model_class=None,
                                                                model_tag_association_class=None,
                                                                filterable=None,
                                                                grid_name=None)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

Column that supports community tags.

filter (*trans, user, query, column_filter*)

Modify query to filter model_class by tag. Multiple filters are ANDed.

get_filter (*trans, user, column_filter*)

get_value (*trans, grid, item*)

```
class galaxy.web.framework.helpers.grids.DateTimeColumn(label, key=None,
                                                           model_class=None,
                                                           method=None,
                                                           format=None, link=None,
                                                           attach_popup=False,
                                                           visible=True, nowrap=False,
                                                           filterable=None, sortable=True,
                                                           label_id_prefix=None,
                                                           inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

sort (*trans, query, ascending, column_name=None*)

Sort query using this column.

```
class galaxy.web.framework.helpers.grids.DeletedColumn(label, key=None,
                                                         model_class=None,
                                                         method=None,
                                                         format=None, link=None,
                                                         attach_popup=False,
                                                         visible=True, nowrap=False,
                                                         filterable=None, sortable=True,
                                                         label_id_prefix=None,
                                                         inbound=False)
```

Bases: *galaxy.web.framework.helpers.grid.GridColumn*

Column that tracks and filters for items with deleted attribute.

filter (*trans, user, query, column_filter*)
 Modify query to filter self.model_class by state.

get_accepted_filters ()
 Returns a list of accepted filters for this column.

class *galaxy.web.framework.helpers.grid.DisplayByUsernameAndSlugGridOperation* (*label,*
key=None,
con-
di-
tion=None,
al-
low_multiple=True,
al-
low_popup=True,
tar-
get=None,
url_args=None,
async_compatible=1,
con-
firm=None,
global_operation=None,
in-
bound=False)

Bases: *galaxy.web.framework.helpers.grid.GridOperation*

Operation to display an item by username and slug.

get_url_args (*item*)

class *galaxy.web.framework.helpers.grid.Grid*

Bases: *object*

Specifies the content and format of a grid (data table).

apply_query_filter (*trans, query, **kwargs*)

async_template = 'grid_base_async.mako'

build_initial_query (*trans, **kwargs*)

columns = []

cur_filter_pref_name = '.filter'

cur_sort_key_pref_name = '.sort_key'

default_filter = {}

default_sort_key = None

exposed = True

get_current_item (*trans, **kwargs*)

get_ids (***kwargs*)

global_actions = []

handle_operation (*trans, operation, ids, **kwargs*)

```
info_text = None
legend = None
model_class = None
num_page_links = 10
num_rows_per_page = 25
operations = []
pass_through_operations = {}
preserve_state = False
show_item_checkboxes = False
standard_filters = []
template = 'grid_base.mako'
title = ''
use_async = False
use_hide_message = True
use_paging = False
```

```
class galaxy.web.framework.helpers.grids.GridAction(label=None, url_args=None, in-
                                                    bound=False)
```

Bases: object

```
class galaxy.web.framework.helpers.grids.GridColumn(label,
                                                    key=None,
                                                    model_class=None, method=None,
                                                    format=None, link=None,
                                                    attach_popup=False, visible=True,
                                                    nowrap=False, filterable=None,
                                                    sortable=True, label_id_prefix=None,
                                                    in-bound=False)
```

Bases: object

```
filter (trans, user, query, column_filter)
    Modify query to reflect the column filter.
```

```
get_accepted_filters ()
    Returns a list of accepted filters for this column.
```

```
get_link (trans, grid, item)
```

```
get_value (trans, grid, item)
```

```
sort (trans, query, ascending, column_name=None)
    Sort query using this column.
```

```
class galaxy.web.framework.helpers.grids.GridColumnFilter(label, args=None)
```

Bases: object

```
get_url_args ()
```

```
class galaxy.web.framework.helpers.grid.GridOperation (label,      key=None,      con-
                                                    dition=None,      al-
                                                    low_multiple=True,      al-
                                                    low_popup=True,      tar-
                                                    get=None,      url_args=None,
                                                    async_compatible=False,
                                                    confirm=None,
                                                    global_operation=None,
                                                    inbound=False)
```

Bases: object

allowed (item)

get_url_args (item)

```
class galaxy.web.framework.helpers.grid.IndividualTagsColumn (col_name,      key,
                                                                model_class=None,
                                                                model_tag_association_class=None,
                                                                filterable=None,
                                                                grid_name=None)
```

Bases: *galaxy.web.framework.helpers.grid.CommunityTagsColumn*

Column that supports individual tags.

get_filter (trans, user, column_filter)

get_value (trans, grid, item)

```
class galaxy.web.framework.helpers.grid.IntegerColumn (label,      key=None,
                                                         model_class=None,
                                                         method=None,      for-
                                                         mat=None,      link=None,
                                                         attach_popup=False,      visi-
                                                         ble=True,      nowrap=False,      fil-
                                                         terable=None,      sortable=True,
                                                         label_id_prefix=None,      in-
                                                         bound=False)
```

Bases: *galaxy.web.framework.helpers.grid.TextColumn*

Integer column that employs freetext, but checks that the text is an integer, so support filtering on integer values.

IMPORTANT NOTE: grids that use this column type should not include the column in the cols_to_filter list of MulticolFilterColumn (i.e., searching on this column type should not be performed in the grid's standard search - it won't throw exceptions, but it also will not find what you're looking for). Grids that search on this column should use 'filterable="advanced"' so that searching is only performed in the advanced search component, restricting the search to the specific column.

This is useful for searching on object ids or other integer columns. See the JobIdColumn column in the SpecifiedDateListGrid class in the jobs controller of the reports webapp for an example.

get_single_filter (user, a_filter)

sort (trans, query, ascending, column_name=None)

Sort query using this column.

```
class galaxy.web.framework.helpers.grid.MulticolFilterColumn (col_name,
                                                                cols_to_filter,
                                                                key,      visible,      fil-
                                                                terable='default')
```

Bases: *galaxy.web.framework.helpers.grid.TextColumn*

Column that performs multicolumn filtering.

filter (*trans, user, query, column_filter*)

Modify query to filter model_class by tag. Multiple filters are ANDed.

```
class galaxy.web.framework.helpers.grid.OwnerAnnotationColumn (col_name,      key,
                                                                model_class=None,
                                                                model_annotation_association_class=None,
                                                                filterable=None)
```

Bases: *galaxy.web.framework.helpers.grid.TextColumn*,
galaxy.model.item_attrs.UsesAnnotations

Column that displays and filters item owner's annotations.

get_single_filter (*user, a_filter*)

Filter by annotation and annotation owner.

get_value (*trans, grid, item*)

Returns first 150 characters of annotation.

```
class galaxy.web.framework.helpers.grid.OwnerColumn (label,                  key=None,
                                                       model_class=None,
                                                       method=None,      format=None,
                                                       link=None,   attach_popup=False,
                                                       visible=True,  nowrap=False,  fil-
                                                       terable=None,  sortable=True,
                                                       label_id_prefix=None,      in-
                                                       bound=False)
```

Bases: *galaxy.web.framework.helpers.grid.TextColumn*

Column that lists item's owner.

get_value (*trans, grid, item*)

sort (*trans, query, ascending, column_name=None*)

Sort column using case-insensitive alphabetical sorting on item's username.

```
class galaxy.web.framework.helpers.grid.PublicURLColumn (label,              key=None,
                                                           model_class=None,
                                                           method=None,      for-
                                                           mat=None,      link=None,
                                                           attach_popup=False,  visi-
                                                           ble=True,   nowrap=False,
                                                           filterable=None,
                                                           sortable=True,      la-
                                                           bel_id_prefix=None,  in-
                                                           bound=False)
```

Bases: *galaxy.web.framework.helpers.grid.TextColumn*

Column displays item's public URL based on username and slug.

get_link (*trans, grid, item*)

```
class galaxy.web.framework.helpers.grids.ReverseSortColumn(label,
                                                             key=None,
                                                             model_class=None,
                                                             method=None,
                                                             format=None,
                                                             link=None,
                                                             attach_popup=False,
                                                             visible=True,
                                                             nowrap=False,
                                                             filterable=None,
                                                             sortable=True,
                                                             label_id_prefix=None,
                                                             inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

Column that reverses sorting; this is useful when the natural sort is descending.

sort (*trans, query, ascending, column_name=None*)

```
class galaxy.web.framework.helpers.grids.SharingStatusColumn(label,
                                                                key=None,
                                                                model_class=None,
                                                                method=None,
                                                                format=None,
                                                                link=None,
                                                                attach_popup=False,
                                                                visible=True,
                                                                nowrap=False,
                                                                filterable=None,
                                                                sortable=True,
                                                                label_id_prefix=None,
                                                                inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

Grid column to indicate sharing status.

filter (*trans, user, query, column_filter*)
 Modify query to filter histories by sharing status.

get_accepted_filters ()
 Returns a list of accepted filters for this column.

get_link (*trans, grid, item*)

get_value (*trans, grid, item*)

```
class galaxy.web.framework.helpers.grids.StateColumn(label,
                                                         key=None,
                                                         model_class=None,
                                                         method=None,
                                                         format=None,
                                                         link=None,
                                                         attach_popup=False,
                                                         visible=True,
                                                         nowrap=False,
                                                         filterable=None,
                                                         sortable=True,
                                                         label_id_prefix=None,
                                                         inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

Column that tracks and filters for items with state attribute.

IMPORTANT NOTE: self.model_class must have a states Bunch or dict if this column type is used in the grid.

filter (*trans, user, query, column_filter*)
 Modify query to filter self.model_class by state.

get_accepted_filters ()

Returns a list of accepted filters for this column.

get_value (*trans, grid, item*)

```
class galaxy.web.framework.helpers.grids.TextColumn (label,                                key=None,
                                                    model_class=None, method=None,
                                                    format=None,                link=None,
                                                    attach_popup=False,            visi-
                                                    ble=True,        nowrap=False,    fil-
                                                    terable=None,        sortable=True,
                                                    label_id_prefix=None,        in-
                                                    bound=False)
```

Bases: *galaxy.web.framework.helpers.grids.GridColumn*

Generic column that employs freetext and, hence, supports freetext, case-independent filtering.

filter (*trans, user, query, column_filter*)

Modify query to filter using free text, case independence.

get_filter (*trans, user, column_filter*)

Returns a SQLAlchemy criterion derived from *column_filter*.

get_single_filter (*user, a_filter*)

Returns a SQLAlchemy criterion derived for a single filter. Single filter is the most basic filter—usually a string—and cannot be a list.

sort (*trans, query, ascending, column_name=None*)

Sort column using case-insensitive alphabetical sorting.

middleware Package

middleware Package WSGI Middleware.

profile Module Middleware that profiles the request with cProfile and displays profiling information at the bottom of each page.

```
class galaxy.web.framework.middleware.profile.ProfileMiddleware (app,
                                                                    global_conf=None,
                                                                    limit=40)
```

Bases: *object*

Middleware that profiles all requests.

All HTML pages will have profiling information appended to them. The data is isolated to that single request, and does not include data from previous requests.

galaxy.web.framework.middleware.profile.func_std_string (*func_name*)

Match what old profile produced

galaxy.web.framework.middleware.profile.get_func_list (*stats, sel_list*)

Use 'sel_list' to select a list of functions to display.

galaxy.web.framework.middleware.profile.pstats_as_html (*stats, *sel_list*)

Return an HTML representation of a *pstats.Stats* object.

remoteuser Module Middleware for handling \$REMOTE_USER if use_remote_user is enabled.

```
class galaxy.web.framework.middleware.remoteuser.RemoteUser (app, maildomain=None,
                                                             display_servers=None,
                                                             admin_users=None, re-
                                                             mote_user_header=None)
```

Bases: object

```
error (start_response, title='Access denied', message='Please contact your local Galaxy administra-
tor:')
```

static Module

```
class galaxy.web.framework.middleware.static.CacheableStaticURLParser (directory,
                                                                       cache_seconds=None)

Bases: paste.urlparser.StaticURLParser
galaxy.web.framework.middleware.static.make_static (global_conf,      document_root,
                                                    cache_seconds=None)
```

translogger Module Middleware for logging requests, using Apache combined log format

```
class galaxy.web.framework.middleware.translogger.TransLogger (application,      log-
                                                                ger=None,      for-
                                                                mat=None,      log-
                                                                ging_level=20, log-
                                                                ger_name='wsgi',
                                                                setup_console_handler=True,
                                                                set_logger_level=10)
```

Bases: object

This logging middleware will log all requests as they go through. They are, by default, sent to a logger named 'wsgi' at the INFO level.

If setup_console_handler is true, then messages for the named logger will be sent to the console.

```
format = '%(REMOTE_ADDR)s - %(REMOTE_USER)s [%(time)s] “%(REQUEST_METHOD)s %(REQUEST_URI)s”
```

```
write_log (environ, method, req_uri, start, status, bytes)
```

```
galaxy.web.framework.middleware.translogger.make_filter (app, global_conf, log-
                                                         ger_name='wsgi',
                                                         format=None,      log-
                                                         ging_level=20,
                                                         setup_console_handler=True,
                                                         set_logger_level=10)
```

This logging middleware will log all requests as they go through. They are, by default, sent to a logger named 'wsgi' at the INFO level.

If setup_console_handler is true, then messages for the named logger will be sent to the console.

xforwardedhost Module

```
class galaxy.web.framework.middleware.xforwardedhost.XForwardedHostMiddleware (app,
                                                                                global_conf=None)
```

Bases: object

A WSGI middleware that changes the HTTP host header in the WSGI environ based on the X-Forwarded-Host header IF found

security Package

security Package

```
class galaxy.web.security.SecurityHelper (**config)
    Bases: object

    decode_guid (session_key)

    decode_id (obj_id, kind=None)

    encode_all_ids (rval, recursive=False)
        Encodes all integer values in the dict rval whose keys are 'id' or end with '_id' excluding tool_id which
        are consumed and produced as is via the API.

    encode_dict_ids (a_dict, kind=None)
        Encode all ids in dictionary. Ids are identified by (a) an 'id' key or (b) a key that ends with '_id'

    encode_guid (session_key)

    encode_id (obj_id, kind=None)

    get_new_guid ()

galaxy.web.security.get_random_bytes (nbytes)
```

webapps Package

webapps Package Galaxy webapps root package – this is a namespace package.

Subpackages

community Package

community Package

app Module

buildapp Module

config Module

Subpackages

controllers Package

controllers Package

admin Module

common Module

hg Module

repository Module

repository_review Module

upload Module

user Module

framework Package

framework Package

Subpackages

middleware Package

middleware Package

hg Module

remoteuser Module

model Package

model Package

mapping Module

Subpackages

migrate Package

check Module

security Package

security Package

util Package

container_util Module

hgweb_config Module

shed_statistics Module

workflow_util Module

demo_sequencer Package

demo_sequencer Package The Galaxy Demo Sequencer Emulator application.

app Module

```
class galaxy.webapps.demo_sequencer.app.UniverseApplication(**kwargs)
    Bases: object

    Encapsulates the state of a Universe application

    shutdown()
```

buildapp Module Provides factory methods to assemble the Galaxy web application

```
galaxy.webapps.demo_sequencer.buildapp.add_ui_controllers(webapp, app)
    Search for controllers in the 'galaxy.webapps.demo_sequencer.controllers' directory and add them to the webapp.

galaxy.webapps.demo_sequencer.buildapp.app_factory(global_conf, **kwargs)
    Return a wsgi application serving the root object

galaxy.webapps.demo_sequencer.buildapp.build_template_error_formatters()
    Build a list of template error formatters for WebError. When an error occurs, WebError pass the exception to each function in this list until one returns a value, which will be displayed on the error page.

galaxy.webapps.demo_sequencer.buildapp.wrap_in_middleware(app, global_conf, **local_conf)
    Based on the configuration wrap app in a set of common and useful middleware.

galaxy.webapps.demo_sequencer.buildapp.wrap_in_static(app, global_conf, **local_conf)
```

config Module Universe configuration builder.

```
class galaxy.webapps.demo_sequencer.config.Configuration(**kwargs)
    Bases: object

    check()

    get(key, default)

    get_bool(key, default)

    is_admin_user(user)
        Determine if the provided user is listed in admin_users.

exception galaxy.webapps.demo_sequencer.config.ConfigurationError
    Bases: exceptions.Exception

galaxy.webapps.demo_sequencer.config.configure_logging(config)
    Allow some basic logging configuration to be read from the cherrpy config.
```

`galaxy.webapps.demo_sequencer.config.resolve_path(path, root)`
 If 'path' is relative make absolute by prepending 'root'

registry Module

class `galaxy.webapps.demo_sequencer.registry.Registry` (*root_dir=None, config=None*)
 Bases: `object`
`parse_request_elem(request_elem)`

Subpackages

controllers Package

controllers Package Galaxy demo sequencer emulator controllers.

common Module

class `galaxy.webapps.demo_sequencer.controllers.common.CommonController` (*app*)
 Bases: `galaxy.web.base.controller.BaseUIController`
`empty_page(trans, **kwd)`
`handle_failure(trans, url, message)`
`handle_request(trans, url, http_method=None, **kwd)`
`index(trans, **kwd)`
`login(trans, **kwd)`
`parse_request_tup(request_tup, **kwd)`
`put(url, **kwd)`

framework Package

framework Package Demo sequencer web application framework

class `galaxy.webapps.demo_sequencer.framework.DemoWebAPITransaction` (*environ, app, webapp*)
 Bases: `galaxy.webapps.demo_sequencer.framework.DemoWebTransaction`
class `galaxy.webapps.demo_sequencer.framework.DemoWebTransaction` (*environ, app, webapp*)
 Bases: `galaxy.web.framework.base.DefaultWebTransaction`
 Encapsulates web transaction specific state for the Demo application (specifically the user's "cookie")
`fill_template(filename, **kwargs)`
 Fill in a template, putting any keyword arguments on the context.
`fill_template_mako(filename, **kwargs)`
`fill_template_string(template_string, context=None, **kwargs)`
 Fill in a template, putting any keyword arguments on the context.

get_cookie (*name='demosequencersession'*)
Convenience method for getting a session cookie

get_galaxy_session ()
Return the current galaxy session

get_message ()
Convenience method for getting the 'message' element of the template context.

set_cookie (*value, name='demosequencersession', path='/', age=90, version='1'*)
Convenience method for setting a session cookie

set_message (*message, type=None*)
Convenience method for setting the 'message' and 'message_type' element of the template context.

show_error_message (*message, refresh_frames=[], use_panels=False, active_view=''*)
Convenience method for displaying an error message. See *show_message*.

show_form (*form, header=None, template='form.mako', use_panels=False, active_view=''*)
Convenience method for displaying a simple page with a single HTML form.

show_message (*message, type='info', refresh_frames=[], cont=None, use_panels=False, active_view=''*)
Convenience method for displaying a simple page with a single message.

type: one of "error", "warning", "info", or "done"; determines the type of dialog box and icon displayed with the message

refresh_frames: names of frames in the interface that should be refreshed when the message is displayed

show_ok_message (*message, refresh_frames=[], use_panels=False, active_view=''*)
Convenience method for displaying an ok message. See *show_message*.

show_warn_message (*message, refresh_frames=[], use_panels=False, active_view=''*)
Convenience method for displaying an warn message. See *show_message*.

stream_template_mako (*filename, **kwargs*)

template_context
Property that replaces itself with a calculated value the first time it is used.

class galaxy.webapps.demo_sequencer.framework.**DemoWebUITransaction** (*environ, app, webapp, session_cookie*)
Bases: *galaxy.webapps.demo_sequencer.framework.DemoWebTransaction*

class galaxy.webapps.demo_sequencer.framework.**WebApplication** (*demo_app, session_cookie='demosequencersession'*)
Bases: *galaxy.web.framework.base.WebApplication*

handle_controller_exception (*e, trans, **kwargs*)

make_body_iterable (*trans, body*)

transaction_chooser (*environ, demo_app, session_cookie*)

galaxy Package

buildapp Module Provides factory methods to assemble the Galaxy web application

```

class galaxy.webapps.galaxy.buildapp.GalaxyWebApplication (galaxy_app,          ses-
                                                         sion_cookie='galaxysession',
                                                         name=None)

    Bases: galaxy.web.framework.webapp.WebApplication

galaxy.webapps.galaxy.buildapp.app_factory (global_conf, **kwargs)
    Return a wsgi application serving the root object

galaxy.webapps.galaxy.buildapp.build_template_error_formatters ()
    Build a list of template error formatters for WebError. When an error occurs, WebError pass the exception to
    each function in this list until one returns a value, which will be displayed on the error page.

galaxy.webapps.galaxy.buildapp.pack_scripts ()

galaxy.webapps.galaxy.buildapp.populate_api_routes (webapp, app)

galaxy.webapps.galaxy.buildapp.wrap_in_middleware (app, global_conf, **local_conf)
    Based on the configuration wrap app in a set of common and useful middleware.

galaxy.webapps.galaxy.buildapp.wrap_in_static (app,          global_conf,          plu-
                                                         gin_frameworks=None, **local_conf)

```

Subpackages

Galaxy API Documentation

Background In addition to being accessible through a web interface, Galaxy can also be accessed programmatically, through shell scripts and other programs. The web interface is appropriate for things like exploratory analysis, visualization, construction of workflows, and rerunning workflows on new datasets.

The web interface is less suitable for things like

- Connecting a Galaxy instance directly to your sequencer and running workflows whenever data is ready.
- Running a workflow against multiple datasets (which can be done with the web interface, but is tedious).
- When the analysis involves complex control, such as looping and branching.

The Galaxy API addresses these and other situations by exposing Galaxy internals through an additional interface, known as an Application Programming Interface, or API.

Quickstart Log in as your user, navigate to the API Keys page in the User menu, and generate a new API key. Make a note of the API key, and then pull up a terminal. Now we'll use the `display.py` script in your `galaxy/scripts/api` directory for a short example:

```

% ./display.py my_key http://localhost:4096/api/histories
Collection Members
-----
#1: /api/histories/8c49be448cfe29bc
    name: Unnamed history
    id: 8c49be448cfe29bc
#2: /api/histories/33b43b4e7093c91f
    name: output test
    id: 33b43b4e7093c91f

```

The result is a Collection of the histories of the user specified by the API key (you). To look at the details of a particular history, say #1 above, do the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc
Member Information
-----
state_details: {'ok': 1, 'failed_metadata': 0, 'upload': 0, 'discarded': 0, 'running': 0, 'setting_m
state: ok
contents_url: /api/histories/8c49be448cfe29bc/contents
id: 8c49be448cfe29bc
name: Unnamed history
```

This gives detailed information about the specific member in question, in this case the History. To view history contents, do the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc/contents
Collection Members
-----
#1: /api/histories/8c49be448cfe29bc/contents/6f91353f3eb0fa4a
  name: Pasted Entry
  type: file
  id: 6f91353f3eb0fa4a
```

What we have here is another Collection of items containing all of the datasets in this particular history. Finally, to view details of a particular dataset in this collection, execute the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc/contents/6f91353f3eb0fa4a
Member Information
-----
misc_blurb: 1 line
name: Pasted Entry
data_type: txt
deleted: False
file_name: /Users/yoplait/work/galaxy-stock/database/files/000/dataset_82.dat
state: ok
download_url: /datasets/6f91353f3eb0fa4a/display?to_ext=txt
visible: True
genome_build: ?
model_class: HistoryDatasetAssociation
file_size: 17
metadata_data_lines: 1
id: 6f91353f3eb0fa4a
misc_info: uploaded txt file
metadata_dbkey: ?
```

And now you've successfully used the API to request and select a history, browse the contents of that history, and then look at detailed information about a particular dataset.

For a more comprehensive Data Library example, set the following option in your galaxy.ini as well, and restart galaxy again:

```
admin_users = you@example.org
library_import_dir = /path/to/some/directory
```

In the directory you specified for 'library_import_dir', create some subdirectories, and put (or symlink) files to import into Galaxy into those subdirectories.

In Galaxy, create an account that matches the address you put in 'admin_users', then browse to that user's preferences and generate a new API Key. Copy the key to your clipboard and then use these scripts:


```
% ./display.py my_key http://localhost:4096/api/libraries
Collection Members
-----

0 elements in collection

% ./library_create_library.py my_key http://localhost:4096/api/libraries api_test 'API Test Library'
Response
-----
/api/libraries/f3f73e481f432006
  name: api_test
  id: f3f73e481f432006

% ./display.py my_key http://localhost:4096/api/libraries
Collection Members
-----
/api/libraries/f3f73e481f432006
  name: api_test
  id: f3f73e481f432006

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006
Member Information
-----
synopsis: None
contents_url: /api/libraries/f3f73e481f432006/contents
description: API Test Library
name: api_test

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents
Collection Members
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591f61ddda595d2c3670
  name: /
  type: folder
  id: 28202595c0d2591f61ddda595d2c3670

% ./library_create_folder.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents 28202595c0d2591f61ddda595d2c3670
Response
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
  name: api_test_folder1
  id: 28202595c0d2591fa4f9089d2303fd89

% ./library_upload_from_import_dir.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents 28202595c0d2591fa4f9089d2303fd89
Response
-----
/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
  name: 2.bed
  id: e9ef7fdb2db87d7b
/api/libraries/f3f73e481f432006/contents/3b7f6a31f80a5018
  name: 3.bed
  id: 3b7f6a31f80a5018

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents
Collection Members
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591f61ddda595d2c3670
  name: /
```

```

    type: folder
    id: 28202595c0d2591f61ddda595d2c3670
/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
    name: /api_test_folder1
    type: folder
    id: 28202595c0d2591fa4f9089d2303fd89
/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
    name: /api_test_folder1/2.bed
    type: file
    id: e9ef7fdb2db87d7b
/api/libraries/f3f73e481f432006/contents/3b7f6a31f80a5018
    name: /api_test_folder1/3.bed
    type: file
    id: 3b7f6a31f80a5018

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
Member Information
-----
misc_blurb: 68 regions
metadata_endCol: 3
data_type: bed
metadata_columns: 6
metadata_nameCol: 4
uploaded_by: nate@...
metadata_strandCol: 6
name: 2.bed
genome_build: hg19
metadata_comment_lines: None
metadata_startCol: 2
metadata_chromCol: 1
file_size: 4272
metadata_data_lines: 68
message:
metadata_dbkey: hg19
misc_info: uploaded bed file
date_uploaded: 2010-06-22T17:01:51.266119
metadata_column_types: str, int, int, str, int, str

```

Other parameters are valid when uploading, they are the same parameters as are used in the web form, like 'link_data_only' and etc.

The request and response format should be considered alpha and are subject to change.

API Design Guidelines The following section outlines guidelines related to extending and/or modifying the Galaxy API. The Galaxy API has grown in an ad-hoc fashion over time by many contributors and so clients SHOULD NOT expect the API will conform to these guidelines - but developers contributing to the Galaxy API SHOULD follow these guidelines.

- API functionality should include docstring documentation for consumption by readthedocs.org.
- Developers should familiarize themselves with the HTTP status code definitions <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>. The API responses should properly set the status code according to the result - in particular 2XX responses should be used for successful requests, 4XX for various kinds of client errors, and 5XX for the errors on the server side.
- If there is an error processing some part of request (one item in a list for instance), the status code should be set to reflect the error and the partial result may or may not be returned depending on the controller - this behavior should be documented.

- API methods should throw a finite number of exceptions (defined in [exceptions Package](#)) and these should subclass *MessageException* and not paste/wsgi HTTP exceptions. When possible, the framework itself should be responsible catching these exceptions, setting the status code, and building an error response.
- Error responses should not consist of plain text strings - they should be dictionaries describing the error and containing the following:

```
{
  "status_code": 400,
  "err_code": 400007,
  "err_msg": "Request contained invalid parameter, action could not be completed.",
  "type": "error",
  "extra_error_info": "Extra information."
}
```

Various error conditions (once a format has been chosen and framework to enforce it in place) should be spelled out in this document.

- Backward compatibility is important and should be maintained when possible. If changing behavior in a non-backward compatible way please ensure one of the following holds - there is a strong reason to believe no consumers depend on a behavior, the behavior is effectively broken, or the API method being modified has not been part of a tagged dist release.

The following bullet points represent good practices more than guidelines, please consider them when modifying the API.

- Functionality should not be copied and pasted between controllers - consider refactoring functionality into associated classes or short of that into Mixins (http://en.wikipedia.org/wiki/Composition_over_inheritance) or into Managers ([managers Package](#)).
- API additions are more permanent changes to Galaxy than many other potential changes and so a second opinion on API changes should be sought. (Consider a pull request!)
- New API functionality should include functional tests. These functional tests should be implemented in Python and placed in *test/functional/api*. (Once such a framework is in place - it is not right now).
- Changes to reflect modifications to the API should be pushed upstream to the BioBlend project if possible.

Longer term goals/notes.

- It would be advantageous to have a clearer separation of anonymous and admin handling functionality.
- If at some point in the future, functionality needs to be added that breaks backward compatibility in a significant way to a component used by the community - a “dev” variant of the API will be established and the community should be alerted and given a timeframe for when the old behavior will be replaced with the new behavior.
- Consistent standards for range-based requests, batch requests, filtered requests, etc... should be established and documented here.

API Controllers Galaxy offers the following API controllers:

annotations Module API operations on annotations.

class `galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesHistory`, `galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin`, `galaxy.web.base.controller.UsesStoredWorkflowMixin`, `galaxy.model.item_attrs.UsesAnnotationsMixin`

```
create (trans, *args, **kwargs)  
delete (trans, *args, **kwargs)  
index (trans, *args, **kwargs)  
undelele (trans, *args, **kwargs)  
  
class galaxy.webapps.galaxy.api.annotations.HistoryAnnotationsController (app)  
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController  
  
    controller_name = 'history_annotations'  
    tagged_item_id = 'history_id'  
  
class galaxy.webapps.galaxy.api.annotations.HistoryContentAnnotationsController (app)  
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController  
  
    controller_name = 'history_content_annotations'  
    tagged_item_id = 'history_content_id'  
  
class galaxy.webapps.galaxy.api.annotations.WorkflowAnnotationsController (app)  
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController  
  
    controller_name = 'workflow_annotations'  
    tagged_item_id = 'workflow_id'
```

authenticate Module API key retrieval through BaseAuth Sample usage:

```
curl -user zipzap@foo.com:password http://localhost:8080/api/authenticate/baseauth
```

Returns:

```
{ "api_key": "baa4d6e3a156d3033f05736255f195f9"  
}
```

```
class galaxy.webapps.galaxy.api.authenticate.AuthenticationController (app)  
    Bases: galaxy.web.base.controller.BaseAPIController  
  
    get_api_key (trans, *args, **kwargs)  
        def get_api_key( self, trans, **kwd ) * GET /api/authenticate/baseauth  
            returns an API key for authenticated user based on BaseAuth headers  
  
            Returns api_key in json format  
            Return type dict  
            Raises ObjectNotFound, HTTPBadRequest
```

configuration Module API operations allowing clients to determine Galaxy instance's capabilities and configuration settings.

```
class galaxy.webapps.galaxy.api.configuration.ConfigurationController (app)  
    Bases: galaxy.web.base.controller.BaseAPIController  
  
    EXPOSED_ADMIN_OPTIONS = ['library_import_dir', 'user_library_import_dir', 'allow_library_path_paste', 'allow_use  
    EXPOSED_USER_OPTIONS = ['enable_unique_workflow_defaults', 'ftp_upload_site', 'ftp_upload_dir', 'wiki_url', 'supp  
    get_config_dict (config, is_admin=False)
```

index (*trans*, *args, **kwargs)

GET /api/configuration Return an object containing exposable configuration settings.

Note: a more complete list is returned if the user is an admin.

dataset_collections Module

class galaxy.webapps.galaxy.api.dataset_collections.**DatasetCollectionsController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*, *galaxy.web.base.controller.UsesHistoryMixin*, *galaxy.web.base.controller.UsesLibraryMixinItems*

create (*trans*, *args, **kwargs)

•**POST /api/dataset_collections:** create a new dataset collection instance.

Parameters **payload** (*dict*) – (optional) dictionary structure containing: * *collection_type*: dataset colltion type to create. * *instance_type*: Instance type - ‘history’ or ‘library’. * *name*: the new dataset collections’s name * *datasets*: object describing datasets for collection

Return type dict

Returns element view of new dataset collection

index (*trans*, *args, **kwargs)

show (*trans*, *args, **kwargs)

datasets Module

datatypes Module API operations allowing clients to determine datatype supported by Galaxy.

class galaxy.webapps.galaxy.api.datatypes.**DatatypesController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

index (*trans*, *args, **kwargs)

GET /api/datatypes Return an object containing upload datatypes.

mapping (*trans*, *args, **kwargs)

GET /api/datatypes/mapping Return a dictionary of class to class mappings.

sniffers (*trans*, *args, **kwargs)

GET /api/datatypes/sniffers Return a list of sniffers.

extended_metadata Module API operations on annotations.

class galaxy.webapps.galaxy.api.extended_metadata.**BaseExtendedMetadataController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*, *galaxy.web.base.controller.UsesExtendedMetadataMixin*, *galaxy.web.base.controller.UsesHistoryMixin*, *galaxy.web.base.controller.UsesLibraryMixinItems*, *galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin*, *galaxy.web.base.controller.UsesStoredWorkflowMixin*

create (*trans*, *args, **kwargs)

delete (*trans*, *args, **kwargs)

index (*trans*, *args, **kwargs)

undelete (*trans*, *args, **kwargs)

```
class galaxy.webapps.galaxy.api.extended_metadata.HistoryDatasetExtendMetadataController (app)
    Bases: galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController
```

```
    controller_name = 'history_dataset_extended_metadata'
```

```
    exmeta_item_id = 'history_content_id'
```

```
class galaxy.webapps.galaxy.api.extended_metadata.LibraryDatasetExtendMetadataController (app)
    Bases: galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController
```

```
    controller_name = 'library_dataset_extended_metadata'
```

```
    exmeta_item_id = 'library_content_id'
```

folder_contents Module API operations on the contents of a library folder.

```
class galaxy.webapps.galaxy.api.folder_contents.FolderContentsController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibrary,
    galaxy.web.base.controller.UsesLibraryMixinItems, galaxy.web.base.controller.UsesHistory
```

Class controls retrieval, creation and updating of folder contents.

build_path (*trans*, *folder*)

Search the path upwards recursively and load the whole route of names and ids for breadcrumb building purposes.

Parameters

- **folder** – current folder for navigating up
- **type** – Galaxy LibraryFolder

Returns list consisting of full path to the library

Type list

create (*self*, *trans*, *library_id*, *payload*, ***kwd*)

• **POST /api/folders/{encoded_id}/contents** create a new library file from an HDA

Parameters **payload** – dictionary structure containing:

Returns a dictionary containing the id, name, and ‘show’ url of the new item

Return type dict

Raises ObjectAttributeInvalidException, InsufficientPermissionsException, ItemAccessibilityException, InternalServerError

index (*trans*, **args*, ***kwargs*)

GET /api/folders/{encoded_folder_id}/contents

Displays a collection (list) of a folder’s contents (files and folders). Encoded folder ID is prepended with ‘F’ if it is a folder as opposed to a data set which does not have it. Full path is provided in response as a separate object providing data for breadcrumb path building.

Parameters

- **folder_id** (*encoded string*) – encoded ID of the folder which contents should be library_dataset_dict
- **kwd** (*dict*) – keyword dictionary with other params

Returns dictionary containing all items and metadata

Type dict

Raises MalformedId, InconsistentDatabase, ObjectNotFound, InternalServerError

show (*trans*, **args*, ***kwargs*)

GET /api/folders/{encoded_folder_id}/

update (*trans*, **args*, ***kwargs*)

PUT /api/folders/{encoded_folder_id}/contents

folders Module API operations on library folders.

class galaxy.webapps.galaxy.api.folders.**FoldersController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*, *galaxy.web.base.controller.UsesLibraryMixinItems*

create (*self*, *trans*, *encoded_parent_folder_id*, ***kwd*)

*POST /api/folders/{encoded_parent_folder_id}

Create a new folder object underneath the one specified in the parameters.

Parameters

- **encoded_parent_folder_id** (*an encoded id string (should be prefixed by 'F')*) – the parent folder’s id (required)
- **name** (*str*) – the name of the new folder (required)
- **description** (*str*) – the description of the new folder

Returns information about newly created folder, notably including ID

Return type dictionary

Raises RequestParameterMissingException

delete (*self*, *trans*, *id*, ***kwd*)

- **DELETE /api/folders/{id}** marks the folder with the given *id* as *deleted* (or removes the *deleted* mark if the *undelete* param is true)

Note: Currently, only admin users can un/delete folders.

Parameters

- **id** (*an encoded id string*) – the encoded id of the folder to un/delete
- **undelete** (*bool*) – (optional) flag specifying whether the item should be deleted or undeleted, defaults to false:

Returns detailed folder information

Return type dictionary

Raises ItemAccessibilityException, MalformedId, ObjectNotFound

get_permissions (*trans*, **args*, ***kwargs*)

- GET /api/folders/{id}/permissions

Load all permissions for the given folder id and return it.

Parameters

- **encoded_folder_id** (*an encoded id string*) – the encoded id of the folder
- **scope** (*string*) – either ‘current’ or ‘available’

Returns dictionary with all applicable permissions’ values

Return type dictionary

Raises ObjectNotFound, InsufficientPermissionsException

index (*trans, *args, **kwargs*)

*GET /api/folders/ This would normally display a list of folders. However, that would be across multiple libraries, so it’s not implemented.

set_permissions (*trans, *args, **kwargs*)

def set_permissions(self, trans, encoded_folder_id, **kwd): *POST
/api/folders/{encoded_folder_id}/permissions

Parameters

- **encoded_folder_id** (*an encoded id string*) – the encoded id of the folder to set the permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: set_permissions
- **add_ids** [] (*string or list*) – list of Role.id defining roles that should have add item permission on the folder
- **manage_ids** [] (*string or list*) – list of Role.id defining roles that should have manage permission on the folder
- **modify_ids** [] (*string or list*) – list of Role.id defining roles that should have modify permission on the folder

Return type dictionary

Returns dict of current roles for all available permission types.

Raises RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException

show (*self, trans, id, **kwd*)

*GET /api/folders/{encoded_folder_id}

Displays information about a folder.

Parameters **id** (*an encoded id string (has to be prefixed by ‘F’)*) – the folder’s encoded id (required)

Returns dictionary including details of the folder

Return type dict

update (*trans, *args, **kwargs*)

PUT /api/folders/{encoded_folder_id}

forms Module API operations on FormDefinition objects.

class galaxy.webapps.galaxy.api.forms.**FormDefinitionAPIController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

create (*trans*, **args*, ***kwargs*)
 POST /api/forms Creates a new form.

index (*trans*, **args*, ***kwargs*)
 GET /api/forms Displays a collection (list) of forms.

show (*trans*, **args*, ***kwargs*)
 GET /api/forms/{encoded_form_id} Displays information about a form.

ftp_files Module API operations on remote files.

class galaxy.webapps.galaxy.api.ftp_files.**FTPFilesAPIController** (*app*)
 Bases: *galaxy.web.base.controller.BaseAPIController*

index (*trans*, **args*, ***kwargs*)
 GET /api/ftp_files/
 Displays remote files.

Parameters

- **target** (*str*) – target to load available datasets from, defaults to ftp possible values: ftp, userdir
- **format** – requested format of data, defaults to flat possible values: flat, jstree, ajax

Returns list of available files

Return type *list*

genomes Module

class galaxy.webapps.galaxy.api.genomes.**GenomesController** (*app*)
 Bases: *galaxy.web.base.controller.BaseAPIController*

RESTful controller for interactions with genome data.

index (*trans*, **args*, ***kwargs*)
 GET /api/genomes: returns a list of installed genomes

show (*trans*, **args*, ***kwargs*)
 GET /api/genomes/{id}

Returns information about build <id>

galaxy.webapps.galaxy.api.genomes.**get_id** (*base*, *format*)

group_roles Module API operations on Group objects.

class galaxy.webapps.galaxy.api.group_roles.**GroupRolesAPIController** (*app*)
 Bases: *galaxy.web.base.controller.BaseAPIController*

delete (*trans*, **args*, ***kwargs*)
 DELETE /api/groups/{encoded_group_id}/roles/{encoded_role_id} Removes a role from a group

index (*trans*, **args*, ***kwargs*)
 GET /api/groups/{encoded_group_id}/roles Displays a collection (list) of groups.

show (*trans*, **args*, ***kwargs*)
 GET /api/groups/{encoded_group_id}/roles/{encoded_role_id} Displays information about a group role.

update (*trans*, **args*, ***kwargs*)
PUT /api/groups/{encoded_group_id}/roles/{encoded_role_id} Adds a role to a group

group_users Module API operations on Group objects.

class `galaxy.webapps.galaxy.api.group_users.GroupUsersAPIController` (*app*)
Bases: `galaxy.web.base.controller.BaseAPIController`

delete (*trans*, **args*, ***kwargs*)
DELETE /api/groups/{encoded_group_id}/users/{encoded_user_id} Removes a user from a group

index (*trans*, **args*, ***kwargs*)
GET /api/groups/{encoded_group_id}/users Displays a collection (list) of groups.

show (*trans*, **args*, ***kwargs*)
GET /api/groups/{encoded_group_id}/users/{encoded_user_id} Displays information about a group user.

update (*trans*, **args*, ***kwargs*)
PUT /api/groups/{encoded_group_id}/users/{encoded_user_id} Adds a user to a group

groups Module API operations on Group objects.

class `galaxy.webapps.galaxy.api.groups.GroupAPIController` (*app*)
Bases: `galaxy.web.base.controller.BaseAPIController`

create (*trans*, **args*, ***kwargs*)
POST /api/groups Creates a new group.

index (*trans*, **args*, ***kwargs*)
GET /api/groups Displays a collection (list) of groups.

show (*trans*, **args*, ***kwargs*)
GET /api/groups/{encoded_group_id} Displays information about a group.

update (*trans*, **args*, ***kwargs*)
PUT /api/groups/{encoded_group_id} Modifies a group.

histories Module API operations on a history.

See also:

`galaxy.model.History`

class `galaxy.webapps.galaxy.api.histories.HistoriesController` (*app*)
Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesHistoryMixin`, `galaxy.web.base.controller.UsesTagsMixin`, `galaxy.web.base.controller.ExportsHistoryMixin`, `galaxy.web.base.controller.ImportsHistoryMixin`

archive_download (*trans*, **args*, ***kwargs*)
export_download(self, trans, id, jeha_id) * GET /api/histories/{id}/exports/{jeha_id}:

If ready and available, return raw contents of exported history. Use/poll “PUT /api/histories/{id}/exports” to initiate the creation of such an export - when ready that route will return 200 status code (instead of 202) with a JSON dictionary containing a *download_url*.

archive_export (*trans*, **args*, ***kwargs*)
export_archive(self, trans, id, payload) * PUT /api/histories/{id}/exports:

start job (if needed) to create history export for corresponding history.

Parameters `id` (*str*) – the encoded id of the history to undelete

Return type dict

Returns object containing url to fetch export from.

citations (*trans*, **args*, ***kwargs*)

create (*trans*, *payload*)

•**POST /api/histories:** create a new history

Parameters **payload** (*dict*) – (optional) dictionary structure containing: * `name`: the new history's name * `history_id`: the id of the history to copy * `archive_source`: the url that will generate the archive to import * `archive_type`: 'url' (default)

Return type dict

Returns element view of new history

delete (*self*, *trans*, *id*, ***kwd*)

•**DELETE /api/histories/{id}** delete the history with the given `id`

Note: Stops all active jobs in the history if `purge` is set.

Parameters

- `id` (*str*) – the encoded id of the history to delete
- `kwd` (*dict*) – (optional) dictionary structure containing:
 - **payload: a dictionary itself containing:**
 - * `purge`: if True, purge the history and all of its HDAs

Return type dict

Returns an error object if an error occurred or a dictionary containing: * `id`: the encoded id of the history, * `deleted`: if the history was marked as deleted, * `purged`: if the history was purged

index (*trans*, *deleted*=*'False'*)

- GET /api/histories:** return undeleted histories for the current user
- GET /api/histories/deleted:** return deleted histories for the current user

Note: Anonymous users are allowed to get their current history

Parameters **deleted** (*boolean*) – if True, show only deleted histories, if False, non-deleted

Return type *list*

Returns list of dictionaries containing summary history information

show (*trans*, *id*, *deleted*=*'False'*)

- GET /api/histories/{id}:** return the history with `id`
- GET /api/histories/deleted/{id}:** return the deleted history with `id`
- GET /api/histories/most_recently_used:** return the most recently used history

Parameters

- **id** (*an encoded id string*) – the encoded id of the history to query or the string ‘most_recently_used’
- **deleted** (*boolean*) – if True, allow information on a deleted history to be shown.

Return type dictionary**Returns** detailed history information from `galaxy.web.base.controller.UsesHistoryMixin.get_h`**undelelete** (*self, trans, id, **kwd*)

- **POST /api/histories/deleted/{id}/undelelete:** undelelete history (that hasn’t been purged) with the given id

Parameters **id** (*str*) – the encoded id of the history to undelelete**Return type** str**Returns** ‘OK’ if the history was undeleleted**update** (*self, trans, id, payload, **kwd*)

- **PUT /api/histories/{id}** updates the values for the history with the given id

Parameters

- **id** (*str*) – the encoded id of the history to undelelete
- **payload** (*dict*) – a dictionary containing any or all the fields in `galaxy.model.History.to_dict()` and/or the following:
 - **annotation**: an annotation for the history

Return type dict**Returns** an error object if an error occurred or a dictionary containing any values that were different from the original and, therefore, updated**history_contents Module** API operations on the contents of a history.

```
class galaxy.webapps.galaxy.api.history_contents.HistoryContentsController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesHistoryMixin, galaxy.web.base.controller.UsesLibraryMixin, galaxy.web.base.controller.UsesLibraryMixinItems, galaxy.web.base.controller.UsesTagsMixin
```

create (*self, trans, history_id, payload, **kwd*)

- **POST /api/histories/{history_id}/contents/{type}** create a new HDA by copying an accessible LibraryDataset

Parameters

- **history_id** (*str*) – encoded id string of the new HDA’s History
- **type** (*str*) – Type of history content - ‘dataset’ (default) or ‘dataset_collection’.
- **payload** (*dict*) – dictionary structure containing:: copy from library (for type ‘dataset’): ‘source’ = ‘library’ ‘content’ = [the encoded id from the library dataset]

copy from history dataset (for type 'dataset'): 'source' = 'hda' 'content' = [the encoded id from the HDA]

copy from history dataset collection (for type 'dataset_collection') 'source' = 'hdca' 'content' = [the encoded id from the HDCA]

create new history dataset collection (for type 'dataset_collection') 'source' = 'new_collection' (default 'source' if type is

'dataset_collection' - no need to specify this)

'collection_type' = For example, "list", "paired", "list:paired". 'name' = Name of new dataset collection. 'element_identifiers' = Recursive list structure defining collection.

Each element must have 'src' which can be 'hda', 'ldda', 'hdca', or 'new_collection', as well as a 'name' which is the name of element (e.g. "forward" or "reverse" for paired datasets, or arbitrary sample names for instance for lists). For all src's except 'new_collection' - a encoded 'id' attribute must be included with element as well. 'new_collection' sources must defined a 'collection_type' and their own list of (potentially) nested 'element_identifiers'.

..note: Currently, a user can only copy an HDA from a history that the user owns.

Return type dict

Returns dictionary containing detailed information for the new HDA

delete (*self*, *trans*, *history_id*, *id*, ***kwd*)

• **DELETE /api/histories/{history_id}/contents/{id}** delete the HDA with the given *id*

Note: Currently does not stop any active jobs for which this dataset is an output.

Parameters

- **id** (*str*) – the encoded id of the history to delete
- **purge** (*bool*) – if True, purge the HDA
- **kwd** (*dict*) – (optional) dictionary structure containing:
 - **payload: a dictionary itself containing:**
 - * **purge:** if True, purge the HDA

Note: that payload optionally can be placed in the query string of the request. This allows clients that strip the request body to still purge the dataset.

Return type dict

Returns an error object if an error occurred or a dictionary containing: * *id*: the encoded id of the history, * *deleted*: if the history was marked as deleted, * *purged*: if the history was purged

index (*self*, *trans*, *history_id*, *ids=None*, ***kwd*)

- **GET /api/histories/{history_id}/contents** return a list of HDA data for the history with the given `id`

Note: Anonymous users are allowed to get their current history contents

If `ids` is not given, index returns a list of *summary* objects for every HDA associated with the given *history_id*.

If `ids` is given, index returns a *more complete* json object for each HDA in the `ids` list.

Parameters

- **history_id** (*str*) – encoded id string of the HDA's History
- **ids** (*str*) – (optional) a comma separated list of encoded *HDA* ids
- **types** (*str*) – (optional) kinds of contents to index (currently just dataset, but `dataset_collection` will be added shortly).

Return type *list*

Returns dictionaries containing summary or detailed HDA information

See also:

`_summary_hda_dict()` and `galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin`

show (*self*, *trans*, *id*, *history_id*, ***kwd*)

- **GET /api/histories/{history_id}/contents/{id}** return detailed information about an HDA within a history

Note: Anonymous users are allowed to get their current history contents

Parameters

- **ids** – the encoded id of the HDA to return
- **history_id** (*str*) – encoded id string of the HDA's History

Return type *dict*

Returns dictionary containing detailed HDA information

See also:

`galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin.get_hda_dict()`

update (*self*, *trans*, *history_id*, *id*, *payload*, ***kwd*)

- **PUT /api/histories/{history_id}/contents/{id}** updates the values for the HDA with the given `id`

Parameters

- **history_id** (*str*) – encoded id string of the HDA's History
- **id** (*str*) – the encoded id of the history to undelete
- **payload** (*dict*) – a dictionary containing any or all the fields in `galaxy.model.HistoryDatasetAssociation.to_dict()` and/or the following:
 - annotation: an annotation for the HDA

Return type *dict*

Returns an error object if an error occurred or a dictionary containing any values that were different from the original and, therefore, updated

item_tags Module API operations related to tagging items.

class `galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController(app)`

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesTagsM`

create (*trans*, *args, **kwargs)

delete (*trans*, *args, **kwargs)

index (*trans*, *args, **kwargs)

show (*trans*, *args, **kwargs)

update (*trans*, *args, **kwargs)

class `galaxy.webapps.galaxy.api.item_tags.HistoryContentTagsController(app)`

Bases: `galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController`

controller_name = 'history_content_tags'

tagged_item_class = 'HistoryDatasetAssociation'

tagged_item_id = 'history_content_id'

class `galaxy.webapps.galaxy.api.item_tags.HistoryTagsController(app)`

Bases: `galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController`

controller_name = 'history_tags'

tagged_item_class = 'History'

tagged_item_id = 'history_id'

class `galaxy.webapps.galaxy.api.item_tags.WorkflowTagsController(app)`

Bases: `galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController`

controller_name = 'workflow_tags'

tagged_item_class = 'StoredWorkflow'

tagged_item_id = 'workflow_id'

job_files Module API for asynchronous job running mechanisms can use to fetch or put files related to running and queued jobs.

class `galaxy.webapps.galaxy.api.job_files.JobFilesAPIController(app)`

Bases: `galaxy.web.base.controller.BaseAPIController`

This job files controller allows remote job running mechanisms to read and modify the current state of files for queued and running jobs. It is certainly not meant to represent part of Galaxy's stable, user facing API.

Furthermore, even if a user key corresponds to the user running the job, it should not be accepted for authorization - this API allows access to low-level unfiltered files and such authorization would break Galaxy's security model for tool execution.

create (*self*, *trans*, *job_id*, *payload*, **kwargs)

- **POST /api/jobs/{job_id}/files** Populate an output file (formal dataset, task split part, working directory file (such as those related to metadata)). This should be a multipart post with a 'file' parameter containing the contents of the actual file to create.

Parameters

- **job_id** (*str*) – encoded id string of the job
- **payload** (*dict*) – dictionary structure containing:: ‘job_key’ = Key authenticating ‘path’ = Path to file to create.

..note: This API method is intended only for consumption by job runners, not end users.

Return type dict

Returns an okay message

index (*self, trans, job_id, **kwargs*)

- **GET /api/jobs/{job_id}/files** Get a file required to staging a job (proper datasets, extra inputs, task-split inputs, working directory files).

Parameters

- **job_id** (*str*) – encoded id string of the job
- **path** (*str*) – Path to file.
- **job_key** (*str*) – A key used to authenticate this request as acting on behalf of a job runner for the specified job.

..note: This API method is intended only for consumption by job runners, not end users.

Return type *binary*

Returns contents of file

jobs Module API operations on a jobs.

See also:

`galaxy.model.Jobs`

class `galaxy.webapps.galaxy.api.jobs.JobController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesHistory`, `galaxy.web.base.controller.UsesLibraryMixinItems`

create (*trans, *args, **kwargs*)

index (*trans, state=None, tool_id=None, history_id=None*)

- **GET /api/jobs:** return jobs for current user

Parameters **state** (*string or list*) – limit listing of jobs to those that match one of the included states. If none, all are returned.

Valid Galaxy job states include: ‘new’, ‘upload’, ‘waiting’, ‘queued’, ‘running’, ‘ok’, ‘error’, ‘paused’, ‘deleted’, ‘deleted_new’

Parameters

- **tool_id** (*string or list*) – limit listing of jobs to those that match one of the included tool_ids. If none, all are returned.

- **history_id** (*string*) – limit listing of jobs to those that match the history_id. If none, all are returned.

Return type *list*

Returns list of dictionaries containing summary job information

inputs (*trans, *args, **kwargs*)

show(trans, id) * GET /api/jobs/{job_id}/inputs

returns input datasets created by job

Parameters **id** (*string*) – Encoded job id

Return type dictionary

Returns dictionary containing input dataset associations

outputs (*trans, *args, **kwargs*)

show(trans, id) * GET /api/jobs/{job_id}/outputs

returns output datasets created by job

Parameters **id** (*string*) – Encoded job id

Return type dictionary

Returns dictionary containing output dataset associations

search (*trans, payload*)

• **POST /api/jobs/search:** return jobs for current user

Parameters **payload** (*dict*) – Dictionary containing description of requested job. This is in the same format as a request to POST /api/tools would take to initiate a job

Return type *list*

Returns list of dictionaries containing summary job information of the jobs that match the requested job run

This method is designed to scan the list of previously run jobs and find records of jobs that had the exact same input parameters and datasets. This can be used to minimize the amount of repeated work, and simply recycle the old results.

show (*trans, id*)

• **GET /api/jobs/{job_id}:** return jobs for current user

Parameters

- **id** (*string*) – Specific job id
- **full** (*boolean*) – whether to return extra information

Return type dictionary

Returns dictionary containing full description of job data

lda_datasets Module API operations on the library datasets.

class `galaxy.webapps.galaxy.api.lda_datasets.LibraryDatasetsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesVisual`

delete (*trans*, **args*, ***kwargs*)

`delete(self, trans, encoded_dataset_id, **kwd): * DELETE /api/libraries/datasets/{encoded_dataset_id}`

Marks the dataset deleted or undeleted based on the value of the undelete flag. If the flag is not present it is considered False and the item is marked deleted.

Parameters `encoded_dataset_id` (*an encoded id string*) – the encoded id of the dataset to change

Returns dict containing information about the dataset

Return type dictionary

download (*self*, *trans*, *format*, ***kwd*)

• **GET** `/api/libraries/datasets/download/{format}`

• **POST** `/api/libraries/datasets/download/{format}` Downloads requested datasets (identified by encoded IDs) in requested format.

example: `GET localhost:8080/api/libraries/datasets/download/tbz?ld_ids%255B%255D=a0d`

Note: supported format values are: ‘zip’, ‘tgz’, ‘tbz’, ‘uncompressed’

Parameters

- **format** (*string*) – string representing requested archive format
- **ld_ids** [] (*an array*) – an array of encoded ids

Return type file

Returns either archive with the requested datasets packed inside or a single uncompressed dataset

Raises `MessageException`, `ItemDeletionException`, `ItemAccessibilityException`, `HTTP-BadRequest`, `OSError`, `IOError`, `ObjectNotFound`

load (*trans*, **args*, ***kwargs*)

`load(self, trans, **kwd): * POST /api/libraries/datasets` Load dataset from the given source into the library. Source can be:

user directory - root folder specified in `galaxy.ini` as “`$user_library_import_dir`”

example path: `path/to/galaxy/$user_library_import_dir/user@example.com/{user can browse everything here}` the folder with the user login has to be created beforehand

(admin)import directory - root folder specified in `galaxy ini` as “`$library_import_dir`”

example path: `path/to/galaxy/$library_import_dir/{admin can browse everything here}`

(admin)any absolute or relative path - option allowed with “`allow_library_path_paste`” in `galaxy.ini`

Parameters

- **encoded_folder_id** (*an encoded id string*) – the encoded id of the folder to import dataset(s) to

- **source** (*str*) – source the datasets should be loaded from
- **link_data** (*bool*) – flag whether to link the dataset to data or copy it to Galaxy, defaults to copy while linking is set to True all symlinks will be resolved `_once_`
- **preserve_dirs** (*bool*) – flag whether to preserve the directory structure when importing dir if False only datasets will be imported
- **file_type** (*str*) – file type of the loaded datasets, defaults to ‘auto’ (autodetect)
- **dbkey** (*str*) – dbkey of the loaded genome, defaults to ‘?’ (unknown)

Returns dict containing information about the created upload job

Return type dictionary

show (*self, trans, id, **kwd*)

- **GET /api/libraries/datasets/{encoded_dataset_id}**: Displays information about the dataset identified by the encoded ID.

Parameters **id** (*an encoded id string*) – the encoded id of the dataset to query

Returns detailed dataset information from base controller

Return type dictionary

See also:

`galaxy.web.base.controller.UsesLibraryMixinItems.get_library_dataset`

show_roles (*trans, *args, **kwargs*)

`show_roles(self, trans, id, **kwd)`: * GET /api/libraries/datasets/{encoded_dataset_id}/permissions

Displays information about current or available roles for a given dataset permission.

Parameters

- **encoded_dataset_id** (*an encoded id string*) – the encoded id of the dataset to query
- **scope** (*string*) – either ‘current’ or ‘available’

Return type dictionary

Returns either dict of current roles for all permission types or dict of available roles to choose from (is the same for any permission type)

show_version (*trans, *args, **kwargs*)

`show_version(self, trans, encoded_dataset_id, encoded_ldda_id, **kwd)`: * GET /api/libraries/datasets/:encoded_dataset_id/versions/:encoded_ldda_id

Displays information about specific version of the library_dataset (i.e. ldda).

Parameters

- **encoded_dataset_id** (*an encoded id string*) – the encoded id of the dataset to query
- **encoded_ldda_id** (*an encoded id string*) – the encoded id of the ldda to query

Return type dictionary

Returns dict of ldda’s details

update_permissions (*trans*, **args*, ***kwargs*)

def update(*self*, *trans*, *encoded_dataset_id*, ***kwd*): *POST /api/libraries/datasets/{encoded_dataset_id}/permissions

Parameters

- **encoded_dataset_id** (*an encoded id string*) – the encoded id of the dataset to update permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: `make_private`, `remove_restrictions`, `set_permissions`
- **access_ids** [] (*string or list*) – list of `Role.name` defining roles that should have access permission on the dataset
- **manage_ids** [] (*string or list*) – list of `Role.name` defining roles that should have manage permission on the dataset
- **modify_ids** [] (*string or list*) – list of `Role.name` defining roles that should have modify permission on the library dataset item

Return type dictionary

Returns dict of current roles for all available permission types

Raises `RequestParameterInvalidException`, `ObjectNotFound`, `InsufficientPermissionsException`, `InternalServerError` `RequestParameterMissingException`

libraries Module API operations on a data library.

class `galaxy.webapps.galaxy.api.libraries.LibrariesController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`

create (*self*, *trans*, *payload*, ***kwd*)

- **POST /api/libraries:** Creates a new library. Only `name` parameter is required.

Note: Currently, only admin users can create libraries.

Parameters **payload** (*dict*) – dictionary structure containing:: `'name'`: the new library's name (required) `'description'`: the new library's description (optional) `'synopsis'`: the new library's synopsis (optional)

Returns detailed library information

Return type dict

Raises `ItemAccessibilityException`, `RequestParameterMissingException`

delete (*self*, *trans*, *id*, ***kwd*)

- **DELETE /api/libraries/{id}** marks the library with the given `id` as *deleted* (or removes the *deleted* mark if the *undelete* param is true)

Note: Currently, only admin users can un/delete libraries.

Parameters

- **id** (*an encoded id string*) – the encoded id of the library to un/delete

- **undelele** (*bool*) – (optional) flag specifying whether the item should be deleted or undeleted, defaults to false:

Returns detailed library information

Return type dictionary

See also:

`galaxy.model.Library.dict_element_visible_keys`

Raises ItemAccessibilityException, MalformedId, ObjectNotFound

get_permissions (*trans, *args, **kwargs*)

- GET /api/libraries/{id}/permissions

Load all permissions for the given library id and return it.

Parameters

- **encoded_library_id** (*an encoded id string*) – the encoded id of the library
- **scope** (*string*) – either ‘current’ or ‘available’
- **is_library_access** (*bool*) – indicates whether the roles available for the library access are requested

Returns dictionary with all applicable permissions’ values

Return type dictionary

Raises ObjectNotFound, InsufficientPermissionsException

index (*self, trans, **kwd*)

- GET /api/libraries: Returns a list of summary data for all libraries.

Parameters **deleted** (*boolean (optional)*) – if True, show only deleted libraries, if False show only non-deleted

Returns list of dictionaries containing library information

Return type *list*

See also:

`galaxy.model.Library.dict_collection_visible_keys`

set_permissions (*trans, *args, **kwargs*)

def set_permissions(self, trans, encoded_dataset_id, **kwd): *POST
/api/libraries/{encoded_library_id}/permissions

Parameters

- **encoded_library_id** (*an encoded id string*) – the encoded id of the library to set the permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: remove_restrictions, set_permissions
- **access_ids** [*string or list*] – list of Role.id defining roles that should have access permission on the library

- **add_ids**[] (*string or list*) – list of Role.id defining roles that should have add item permission on the library
- **manage_ids**[] (*string or list*) – list of Role.id defining roles that should have manage permission on the library
- **modify_ids**[] (*string or list*) – list of Role.id defining roles that should have modify permission on the library

Return type dictionary

Returns dict of current roles for all available permission types

Raises RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException

set_permissions_old (*trans, library, payload, **kwd*)
* old implementation for backward compatibility *

POST /api/libraries/{encoded_library_id}/permissions Updates the library permissions.

show (*self, trans, id, deleted='False', **kwd*)

- **GET /api/libraries/{encoded_id}**: returns detailed information about a library
- **GET /api/libraries/deleted/{encoded_id}**: returns detailed information about a `deleted` library

Parameters

- **id** (*an encoded id string*) – the encoded id of the library
- **deleted** (*boolean*) – if True, allow information on a `deleted` library

Returns detailed library information

Return type dictionary

See also:

`galaxy.model.Library.dict_element_visible_keys`

Raises MalformedId, ObjectNotFound

update (*trans, *args, **kwargs*)

- **PATCH /api/libraries/{encoded_id}** Updates the library defined by an `encoded_id` with the data in the payload.

Note: Currently, only admin users can update libraries. Also the library must not be *deleted*.

param id the encoded id of the library

type id an encoded id string

param payload (required) dictionary structure containing:: 'name': new library's name, cannot be empty 'description': new library's description 'synopsis': new library's synopsis

type payload dict

returns detailed library information

rtype dict

raises ItemAccessibilityException, MalformedId, ObjectNotFound, RequestParameterInvalidException, RequestParameterMissingException

library_contents Module API operations on the contents of a data library.

class `galaxy.webapps.galaxy.api.library_contents.LibraryContentsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesLibraryMixin`, `galaxy.web.base.controller.UsesHistoryMixin`

create (*self, trans, library_id, payload, **kwd*)

• **POST /api/libraries/{library_id}/contents:** create a new library file or folder

To copy an HDA into a library send `create_type` of 'file' and the HDA's encoded id in `from_hda_id` (and optionally `ldda_message`).

Parameters

- **library_id** (*str*) – encoded id string of the library where to create the new item
- **payload** (*dict*) – dictionary structure containing:
 - `folder_id`: the parent folder of the new item
 - `create_type`: the type of item to create ('file', 'folder' or 'collection')
 - **from_hda_id: (optional) the id of an accessible HDA to copy into the library**
 - `ldda_message`: (optional) the new message attribute of the LDDA created
 - **extended_metadata: (optional) dict-dictionary containing any extended metadata to associate with the item**
 - `link_data_only`: (optional) either 'copy_files' (default) or 'link_to_files'
 - `upload_option`: (optional) one of 'upload_file' (default), 'upload_directory' or 'upload_paths'
 - `server_dir`: (optional) only if `upload_option` is 'upload_directory'
 - `filesystem_paths`: (optional) only if `upload_option` is 'upload_paths' and the user is an admin

Return type dict

Returns a dictionary containing the id, name, and 'show' url of the new item

delete (*self, trans, library_id, id, **kwd*)

• **DELETE /api/libraries/{library_id}/contents/{id}** delete the LibraryDataset with the given `id`

Parameters

- **id** (*str*) – the encoded id of the library dataset to delete
- **kwd** (*dict*) – (optional) dictionary structure containing:
 - **payload: a dictionary itself containing:**
 - * `purge`: if True, purge the LD

Return type dict

Returns an error object if an error occurred or a dictionary containing: * `id`: the encoded id of the library dataset, * `deleted`: if the library dataset was marked as deleted, * `purged`: if the library dataset was purged

index (*self*, *trans*, *library_id*, ***kwd*)

• **GET /api/libraries/{library_id}/contents**: Returns a list of library files and folders.

Note: May be slow! Returns all content traversing recursively through all folders.

See also:

`galaxy.webapps.galaxy.api.FolderContentsController.index` for a non-recursive solution

Parameters **library_id** (*string*) – encoded id string of the library

Returns

list of dictionaries of the form: * `id`: the encoded id of the library item * `name`: the ‘library path’

or relationship of the library item to the root

- `type`: ‘file’ or ‘folder’
- `url`: the url to get detailed information on the library item

Return type *list*

Raises `MalformedId`, `InconsistentDatabase`, `RequestParameterInvalidException`, `InternalServerError`

show (*self*, *trans*, *id*, *library_id*, ***kwd*)

• **GET /api/libraries/{library_id}/contents/{id}** Returns information about library file or folder.

Parameters

- **id** (*str*) – the encoded id of the library item to return
- **library_id** (*str*) – encoded id string of the library that contains this item

Returns detailed library item information

Return type `dict`

See also:

`galaxy.model.LibraryDataset.to_dict()` and `galaxy.model.LibraryFolder.dict_element_v`

update (*self*, *trans*, *id*, *library_id*, *payload*, ***kwd*)

• **PUT /api/libraries/{library_id}/contents/{id}** create a `ImplicitlyConvertedDatasetAssociation`

See also:

`galaxy.model.ImplicitlyConvertedDatasetAssociation`

Parameters

- **id** (*str*) – the encoded id of the library item to return

- **library_id** (*str*) – encoded id string of the library that contains this item
- **payload** (*dict*) – dictionary structure containing:: ‘converted_dataset_id’:

Return type None

Returns None

metrics Module API operations for for querying and recording user metrics from some client (typically a user’s browser).

class `galaxy.webapps.galaxy.api.metrics.MetricsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`

create (*trans, payload*)

- **POST /api/metrics:** record any metrics sent and return some status object

Note: Anonymous users can post metrics

Parameters **payload** (*dict*) – (optional) dictionary structure containing: * metrics: a list containing dictionaries of the form:

 ** namespace: label indicating the source of the metric ** time: isoformat
 datetime when the metric was recorded ** level: an integer representing the
 metric’s log level ** args: a json string containing an array of extra data

Return type dict

Returns status object

debugging = None

set to true to send additional debugging info to the log

page_revisions Module API for updating Galaxy Pages

class `galaxy.webapps.galaxy.api.page_revisions.PageRevisionsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.SharableIt`
`galaxy.model.item_attrs.UsesAnnotations`, `galaxy.web.base.controller.SharableMixin`

create (*self, trans, page_id, payload **kwd*)

- **POST /api/pages/{page_id}/revisions** Create a new revision for a page

Parameters

- **page_id** – Add revision to Page with ID=page_id
- **payload** – A dictionary containing:: ‘title’ = New title of the page ‘content’ =
 New content of the page

Return type dictionary

Returns Dictionary with ‘success’ or ‘error’ element to indicate the result of the request

index (*self, trans, page_id, **kwd*)

- **GET /api/pages/{page_id}/revisions** return a list of Page revisions

Parameters `page_id` – Display the revisions of Page with ID=`page_id`

Return type *list*

Returns dictionaries containing different revisions of the page

pages Module API for updating Galaxy Pages

class `galaxy.webapps.galaxy.api.pages.PagesController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.SharableIt`
`galaxy.model.item_attrs.UsesAnnotations`, `galaxy.web.base.controller.SharableMixin`

create (*self*, *trans*, *payload*, ***kwd*)

•**POST /api/pages** Create a page and return dictionary containing Page summary

Parameters `payload` – dictionary structure containing:: ‘slug’ = The title slug for the page
URL, must be unique ‘title’ = Title of the page ‘content’ = HTML contents of the page
‘annotation’ = Annotation that will be attached to the page

Return type dict

Returns Dictionary return of the Page.to_dict call

delete (*self*, *trans*, *id*, ***kwd*)

•**DELETE /api/pages/{id}** Create a page and return dictionary containing Page summary

Parameters `id` – ID of page to be deleted

Return type dict

Returns Dictionary with ‘success’ or ‘error’ element to indicate the result of the request

index (*self*, *trans*, *deleted=False*, ***kwd*)

•**GET /api/pages** return a list of Pages viewable by the user

Parameters `deleted` – Display deleted pages

Return type *list*

Returns dictionaries containing summary or detailed Page information

show (*self*, *trans*, *id*, ***kwd*)

•**GET /api/pages/{id}** View a page summary and the content of the latest revision

Parameters `id` – ID of page to be displayed

Return type dict

Returns Dictionary return of the Page.to_dict call with the ‘content’ field populated by the
most recent revision

provenance Module API operations provenance

```

class galaxy.webapps.galaxy.api.provenance.BaseProvenanceController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesHistory

    create(trans, *args, **kwargs)

    delete(trans, *args, **kwargs)

    index(trans, *args, **kwargs)

    show(trans, *args, **kwargs)

class galaxy.webapps.galaxy.api.provenance.HDAProvenanceController(app)
    Bases: galaxy.webapps.galaxy.api.provenance.BaseProvenanceController

    controller_name = 'history_content_provenance'

    provenance_item_class = 'HistoryDatasetAssociation'

    provenance_item_id = 'history_content_id'

class galaxy.webapps.galaxy.api.provenance.LDDAProvenanceController(app)
    Bases: galaxy.webapps.galaxy.api.provenance.BaseProvenanceController

    controller_name = 'ldda_provenance'

    provenance_item_class = 'LibraryDatasetDatasetAssociation'

    provenance_item_id = 'library_content_id'

```

quotas Module API operations on Quota objects.

```

class galaxy.webapps.galaxy.api.quotas.QuotaAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controllers.admin.AdminAPIController, galaxy.actions.admin.AdminActions, galaxy.web.base.controller.UsesQuotaMixin, galaxy.web.params.QuotaParamParser

    create(trans, *args, **kwargs)
        POST /api/quotas Creates a new quota.

    delete(trans, *args, **kwargs)
        DELETE /api/quotas/{encoded_quota_id} Deletes a quota

    index(trans, *args, **kwargs)
        GET /api/quotas GET /api/quotas/deleted Displays a collection (list) of quotas.

    show(trans, *args, **kwargs)
        GET /api/quotas/{encoded_quota_id} GET /api/quotas/deleted/{encoded_quota_id} Displays information about a quota.

    undelete(trans, *args, **kwargs)
        POST /api/quotas/deleted/{encoded_quota_id}/undelete Undeletes a quota

    update(trans, *args, **kwargs)
        PUT /api/quotas/{encoded_quota_id} Modifies a quota.

```

request_types Module API operations on RequestType objects.

```

class galaxy.webapps.galaxy.api.request_types.RequestTypeAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController

```

```
create (trans, *args, **kwargs)  
    POST /api/request_types Creates a new request type (external_service configuration).  
index (trans, *args, **kwargs)  
    GET /api/request_types Displays a collection (list) of request_types.  
show (trans, *args, **kwargs)  
    GET /api/request_types/{encoded_request_type_id} Displays information about a request_type.
```

requests Module API operations on a sample tracking system.

```
class galaxy.webapps.galaxy.api.requests.RequestsAPIController (app)  
    Bases: galaxy.web.base.controller.BaseAPIController  
  
    index (trans, *args, **kwargs)  
        GET /api/requests Displays a collection (list) of sequencing requests.  
  
    show (trans, *args, **kwargs)  
        GET /api/requests/{encoded_request_id} Displays details of a sequencing request.  
  
    update (trans, *args, **kwargs)  
        PUT /api/requests/{encoded_request_id} Updates a request state, sample state or sample dataset transfer  
        status depending on the update_type  
  
    v = ('REQUEST', 'request_state')
```

roles Module API operations on Role objects.

```
class galaxy.webapps.galaxy.api.roles.RoleAPIController (app)  
    Bases: galaxy.web.base.controller.BaseAPIController  
  
    create (trans, *args, **kwargs)  
        POST /api/roles Creates a new role.  
  
    index (trans, *args, **kwargs)  
        GET /api/roles Displays a collection (list) of roles.  
  
    show (trans, *args, **kwargs)  
        GET /api/roles/{encoded_role_id} Displays information about a role.
```

samples Module API operations for samples in the Galaxy sample tracking system.

```
class galaxy.webapps.galaxy.api.samples.SamplesAPIController (app)  
    Bases: galaxy.web.base.controller.BaseAPIController  
  
    index (trans, *args, **kwargs)  
        GET /api/requests/{encoded_request_id}/samples Displays a collection (list) of sample of a sequencing  
        request.  
  
    k = 'SAMPLE_DATASET'  
  
    update (trans, *args, **kwargs)  
        PUT /api/samples/{encoded_sample_id} Updates a sample or objects related ( mapped ) to a sample.  
  
    update_type_values = ['sample_state', 'run_details', 'sample_dataset_transfer_status']  
  
    update_types = <galaxy.util.bunch.Bunch object>  
  
    v = ['sample_dataset_transfer_status']
```

search Module API for searching Galaxy Datasets

class `galaxy.webapps.galaxy.api.search.SearchController` (*app*)
 Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.SharableIt`

create (*trans*, **args*, ***kwargs*)
 POST /api/search Do a search of the various elements of Galaxy.

tool_data Module

class `galaxy.webapps.galaxy.api.tool_data.ToolData` (*app*)
 Bases: `galaxy.web.base.controller.BaseAPIController`

RESTful controller for interactions with tool data

index (*trans*, **args*, ***kwargs*)
 GET /api/tool_data: returns a list tool_data tables:

show (*trans*, **args*, ***kwargs*)

tool_shed_repositories Module

class `galaxy.webapps.galaxy.api.tool_shed_repositories.ToolShedRepositoriesController` (*app*)
 Bases: `galaxy.web.base.controller.BaseAPIController`

RESTful controller for interactions with tool shed repositories.

exported_workflows (*trans*, **args*, ***kwargs*)
 GET /api/tool_shed_repositories/{encoded_tool_shed_repository_id}/exported_workflows
 Display a list of dictionaries containing information about this tool shed repository's exported workflows.

Parameters **id** – the encoded id of the ToolShedRepository object

get_latest_installable_revision (*trans*, **args*, ***kwargs*)
 POST /api/tool_shed_repositories/get_latest_installable_revision Get the latest installable revision of a specified repository from a specified Tool Shed.

Parameters **key** – the current Galaxy admin user's API key

The following parameters are included in the payload. :param tool_shed_url (required): the base URL of the Tool Shed from which to retrieve the Repository revision. :param name (required): the name of the Repository :param owner (required): the owner of the Repository

import_workflow (*trans*, **args*, ***kwargs*)
 POST /api/tool_shed_repositories/import_workflow
 Import the specified exported workflow contained in the specified installed tool shed repository into Galaxy.

Parameters

- **key** – the API key of the Galaxy user with which the imported workflow will be associated.
- **id** – the encoded id of the ToolShedRepository object

The following parameters are included in the payload. :param index: the index location of the workflow tuple in the list of exported workflows stored in the metadata for the specified repository

import_workflows (*trans*, **args*, ***kwargs*)
 POST /api/tool_shed_repositories/import_workflows

Import all of the exported workflows contained in the specified installed tool shed repository into Galaxy.

Parameters

- **key** – the API key of the Galaxy user with which the imported workflows will be associated.
- **id** – the encoded id of the ToolShedRepository object

index (*trans, *args, **kwargs*)

GET /api/tool_shed_repositories Display a list of dictionaries containing information about installed tool shed repositories.

install_repository_revision (*trans, *args, **kwargs*)

POST /api/tool_shed_repositories/install_repository_revision Install a specified repository revision from a specified tool shed into Galaxy.

Parameters **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool_shed_url (required): the base URL of the Tool Shed from which to install the Repository :param name (required): the name of the Repository :param owner (required): the owner of the Repository :param changeset_revision (required): the changeset_revision of the RepositoryMetadata object associated with the Repository :param new_tool_panel_section_label (optional): label of a new section to be added to the Galaxy tool panel in which to load

tools contained in the Repository. Either this parameter must be an empty string or the tool_panel_section_id parameter must be an empty string or both must be an empty string (both cannot be used simultaneously).

Parameters

- **(optional)** (*shed_tool_conf*) – id of the Galaxy tool panel section in which to load tools contained in the Repository. If this parameter is an empty string and the above new_tool_panel_section_label parameter is an empty string, tools will be loaded outside of any sections in the tool panel. Either this parameter must be an empty string or the tool_panel_section_id parameter must be an empty string or both must be an empty string (both cannot be used simultaneously).
- **(optional)** – Set to True if you want to install repository dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – Set to True if you want to install tool dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – The shed-related tool panel configuration file configured in the “tool_config_file” setting in the Galaxy config file (e.g., galaxy.ini). At least one shed-related tool panel config file is required to be configured. Setting this parameter to a specific file enables you to choose where the specified repository will be installed because the tool_path attribute of the <toolbox> from the specified file is used as the installation location (e.g., <toolbox tool_path=”../shed_tools”>). If this parameter is not set, a shed-related tool panel configuration file will be selected automatically.

install_repository_revisions (*trans, *args, **kwargs*)

POST /api/tool_shed_repositories/install_repository_revisions Install one or more specified repository revisions from one or more specified tool sheds into Galaxy. The received parameters must be ordered lists so that positional values in tool_shed_urls, names, owners and changeset_revisions are associated.

It’s questionable whether this method is needed as the above method for installing a single repository can probably cover all desired scenarios. We’ll keep this one around just in case...

Parameters **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool_shed_urls: the base URLs of the Tool Sheds from which to install a specified Repository :param names: the names of the Repositories to be installed :param owners: the owners of the Repositories to be installed :param changeset_revisions: the changeset_revisions of each RepositoryMetadata object associated with each Repository to be installed :param new_tool_panel_section_label: optional label of a new section to be added to the Galaxy tool panel in which to load

tools contained in the Repository. Either this parameter must be an empty string or the tool_panel_section_id parameter must be an empty string, as both cannot be used.

Parameters

- **tool_panel_section_id** – optional id of the Galaxy tool panel section in which to load tools contained in the Repository. If not set, tools will be loaded outside of any sections in the tool panel. Either this parameter must be an empty string or the tool_panel_section_id parameter must be an empty string, as both cannot be used.
- **(optional) (shed_tool_conf)** – Set to True if you want to install repository dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – Set to True if you want to install tool dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – The shed-related tool panel configuration file configured in the “tool_config_file” setting in the Galaxy config file (e.g., galaxy.ini). At least one shed-related tool panel config file is required to be configured. Setting this parameter to a specific file enables you to choose where the specified repository will be installed because the tool_path attribute of the <toolbox> from the specified file is used as the installation location (e.g., <toolbox tool_path=”../shed_tools”>). If this parameter is not set, a shed-related tool panel configuration file will be selected automatically.

repair_repository_revision (*trans*, *args, **kwargs)

POST /api/tool_shed_repositories/repair_repository_revision Repair a specified repository revision previously installed into Galaxy.

Parameters **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool_shed_url (required): the base URL of the Tool Shed from which the Repository was installed :param name (required): the name of the Repository :param owner (required): the owner of the Repository :param changeset_revision (required): the changeset_revision of the RepositoryMetadata object associated with the Repository

reset_metadata_on_installed_repositories (*trans*, *args, **kwargs)

PUT /api/tool_shed_repositories/reset_metadata_on_installed_repositories

Resets all metadata on all repositories installed into Galaxy in an “orderly fashion”.

Parameters **key** – the API key of the Galaxy admin user.

show (*trans*, *args, **kwargs)

GET /api/tool_shed_repositories/{encoded_tool_shed_repository_id} Display a dictionary containing information about a specified tool_shed_repository.

Parameters **id** – the encoded id of the ToolShedRepository object

galaxy.webapps.galaxy.api.tool_shed_repositories.get_message_for_no_shed_tool_config()

tools Module**users Module** API operations on User objects.**class** `galaxy.webapps.galaxy.api.users.UserAPIController` (*app*)*Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesTagsMixin, galaxy.web.base.controller.CreatesUsersMixin, galaxy.web.base.controller.CreatesApiKeysMixin***anon_user_api_value** (*trans*)

Returns data for an anonymous user, truncated to only usage and quota_percent

api_key (*trans, *args, **kwargs*)

POST /api/users/{encoded_user_id}/api_key Creates a new API key for specified user.

create (*trans, *args, **kwargs*)

POST /api/users Creates a new Galaxy user.

delete (*trans, *args, **kwargs*)**index** (*trans, *args, **kwargs*)

GET /api/users GET /api/users/deleted Displays a collection (list) of users.

show (*trans, *args, **kwargs*)GET /api/users/{encoded_user_id} GET /api/users/deleted/{encoded_user_id} GET /api/users/current
Displays information about a user.**undelete** (*trans, *args, **kwargs*)**update** (*trans, *args, **kwargs*)**visualizations Module** Visualizations resource control over the API.

NOTE!: this is a work in progress and functionality and data structures may change often.

class `galaxy.webapps.galaxy.api.visualizations.VisualizationsController` (*app*)*Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesVisualizationMixin, galaxy.web.base.controller.SharableMixin, galaxy.model.item_attrs.UsesAnnotations*

RESTful controller for interactions with visualizations.

create (*trans, *args, **kwargs*)

POST /api/visualizations creates a new visualization using the given payload

POST /api/visualizations?import_id={encoded_visualization_id} imports a copy of an existing visualization into the user's workspace

index (*trans, *args, **kwargs*)

GET /api/visualizations:

show (*trans, *args, **kwargs*)

GET /api/visualizations/{viz_id}

update (*trans, *args, **kwargs*)

PUT /api/visualizations/{encoded_id}

workflows Module API operations for Workflows**class** `galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController` (*app*)*Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesStoreMixin*

galaxy.web.base.controller.UsesHistoryMixin, galaxy.model.item_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin

cancel_workflow_invocation (*trans, *args, **kwargs*)

DELETE /api/workflows/{workflow_id}/usage/{usage_id} Cancel the specified workflow invocation.

Parameters

- **workflow_id** (*str*) – the workflow id (required)
- **usage_id** (*str*) – the usage id (required)

Raises exceptions.MessageException, exceptions.ObjectNotFound

create (*trans, *args, **kwargs*)

POST /api/workflows

Run or create workflows from the api.

If installed_repository_file or from_history_id is specified a new workflow will be created for this user. Otherwise, workflow_id must be specified and this API method will cause a workflow to execute.

:param installed_repository_file The path of a workflow to import. Either workflow_id, installed_repository_file or from_history_id must be specified :type installed_repository_file str

Parameters

- **workflow_id** (*str*) – An existing workflow id. Either workflow_id, installed_repository_file or from_history_id must be specified
- **parameters** (*dict*) – If workflow_id is set - see _update_step_parameters()
- **ds_map** (*dict*) – If workflow_id is set - a dictionary mapping each input step id to a dictionary with 2 keys: 'src' (which can be 'ldda', 'ld' or 'hda') and 'id' (which should be the id of a LibraryDatasetDatasetAssociation, LibraryDataset or HistoryDatasetAssociation respectively)
- **no_add_to_history** (*str*) – If workflow_id is set - if present in the payload with any value, the input datasets will not be added to the selected history
- **history** (*str*) – If workflow_id is set - optional history where to run the workflow, either the name of a new history or "hist_id=HIST_ID" where HIST_ID is the id of an existing history. If not specified, the workflow will be run a new unnamed history
- **replacement_params** (*dict*) – If workflow_id is set - an optional dictionary used when renaming datasets
- **from_history_id** (*str*) – Id of history to extract a workflow from. Either workflow_id, installed_repository_file or from_history_id must be specified
- **job_ids** (*str*) – If from_history_id is set - optional list of jobs to include when extracting a workflow from history
- **dataset_ids** (*str*) – If from_history_id is set - optional list of HDA 'hid's corresponding to workflow inputs when extracting a workflow from history
- **dataset_collection_ids** (*str*) – If from_history_id is set - optional list of HDCA 'hid's corresponding to workflow inputs when extracting a workflow from history
- **workflow_name** (*str*) – If from_history_id is set - name of the workflow to create when extracting a workflow from history

delete (*trans*, **args*, ***kwargs*)

DELETE /api/workflows/{encoded_workflow_id} Deletes a specified workflow Author: rpark

copied from galaxy.web.controllers.workflows.py (delete)

import_new_workflow_deprecated (*trans*, **args*, ***kwargs*)

POST /api/workflows/upload Importing dynamic workflows from the api. Return newly generated workflow id. Author: rpark

currently assumes payload['workflow'] is a json representation of a workflow to be inserted into the database

Deprecated in favor to POST /api/workflows with encoded 'workflow' in payload the same way.

import_shared_workflow_deprecated (*trans*, **args*, ***kwargs*)

POST /api/workflows/import Import a workflow shared by other users.

Parameters **workflow_id** (*str*) – the workflow id (required)

Raises exceptions.MessageException, exceptions.ObjectNotFound

index (*trans*, **args*, ***kwargs*)

GET /api/workflows

Displays a collection of workflows.

Parameters **show_published** (*boolean*) – if True, show also published workflows

show (*trans*, **args*, ***kwargs*)

GET /api/workflows/{encoded_workflow_id}

Displays information needed to run a workflow from the command line.

workflow_dict (*trans*, **args*, ***kwargs*)

GET /api/workflows/{encoded_workflow_id}/download Returns a selected workflow as a json dictionary.

workflow_invocation_step (*trans*, **args*, ***kwargs*)

GET /api/workflows/{ workflow_id }/usage/{ usage_id }/steps/{ step_id }

Parameters

- **workflow_id** (*str*) – the workflow id (required)
- **usage_id** (*str*) – the usage id (required)
- **step_id** (*str*) – encoded id of the WorkflowInvocationStep (required)
- **payload** – payload containing update action information for running workflow.

Raises exceptions.MessageException, exceptions.ObjectNotFound

workflow_invocation_step_update (*trans*, **args*, ***kwargs*)

PUT /api/workflows/{ workflow_id }/usage/{ usage_id }/steps/{ step_id } Update state of running workflow step invocation - still very nebulous but this would be for stuff like confirming paused steps can proceed etc....

Parameters

- **workflow_id** (*str*) – the workflow id (required)
- **usage_id** (*str*) – the usage id (required)
- **step_id** (*str*) – encoded id of the WorkflowInvocationStep (required)

Raises exceptions.MessageException, exceptions.ObjectNotFound

```

workflow_request (trans, *args, **kwargs)
    POST /api/workflows/{encoded_workflow_id}/usage

    Schedule the workflow specified by workflow_id to run.

workflow_usage (trans, *args, **kwargs)
    GET /api/workflows/{ workflow_id}/usage Get the list of the workflow usage

    Parameters workflow_id (str) – the workflow id (required)

    Raises exceptions.MessageException, exceptions.ObjectNotFound

workflow_usage_contents (trans, *args, **kwargs)
    GET /api/workflows/{ workflow_id}/usage/{ usage_id} Get detailed description of workflow usage

    Parameters

    • workflow_id (str) – the workflow id (required)

    • usage_id (str) – the usage id (required)

    Raises exceptions.MessageException, exceptions.ObjectNotFound

```

controllers Package

controllers Package Galaxy web controllers.

admin Module

```

class galaxy.webapps.galaxy.controllers.admin.AdminGalaxy (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controllers.admin.AdminGalaxy,
    galaxy.actions.admin.AdminActions, galaxy.web.base.controller.UsesQuotaMixin,
    galaxy.web.params.QuotaParamParser

    check_for_tool_dependencies (trans, migration_stage)

    create_quota (trans, *args, **kwargs)

    delete_operation = <galaxy.web.framework.helpers.grids.GridOperation object>

    edit_quota (trans, *args, **kwargs)

    group_list_grid = <galaxy.webapps.galaxy.controllers.admin.GroupListGrid object>

    impersonate (trans, *args, **kwargs)

    manage_users_and_groups_for_quota (trans, *args, **kwargs)

    mark_quota_deleted (trans, *args, **kwargs)

    purge_operation = <galaxy.web.framework.helpers.grids.GridOperation object>

    purge_quota (trans, *args, **kwargs)

    quota_list_grid = <galaxy.webapps.galaxy.controllers.admin.QuotaListGrid object>

    quotas (trans, *args, **kwargs)

    rename_quota (trans, *args, **kwargs)

    review_tool_migration_stages (trans, *args, **kwargs)

    role_list_grid = <galaxy.webapps.galaxy.controllers.admin.RoleListGrid object>

    set_quota_default (trans, *args, **kwargs)

```

```

tool_version_list_grid = <galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid object>
undelete_operation = <galaxy.web.framework.helpers.grids.GridOperation object>
undelete_quota (trans, *args, **kwargs)
unset_quota_default (trans, *args, **kwargs)
user_list_grid = <galaxy.webapps.galaxy.controllers.admin.UserListGrid object>
view_datatypes_registry (trans, *args, **kwargs)
view_tool_data_tables (trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.admin.GroupListGrid
Bases: galaxy.web.framework.helpers.grids.Grid

class NameColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.TextColumn

get_value (trans, grid, group)

class GroupListGrid.RolesColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

get_value (trans, grid, group)

class GroupListGrid.StatusColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

get_value (trans, grid, group)

class GroupListGrid.UsersColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

get_value (trans, grid, group)

GroupListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.NameColumn object at 0x7fc070eb0ed0>, <galaxy.webapps.galaxy.controllers.admin.RolesColumn object at 0x7fc070eb0ed0>, <galaxy.webapps.galaxy.controllers.admin.StatusColumn object at 0x7fc070eb0ed0>, <galaxy.webapps.galaxy.controllers.admin.UsersColumn object at 0x7fc070eb0ed0>]
GroupListGrid.default_sort_key = 'name'
GroupListGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7fc0725ae510>]
GroupListGrid.model_class
alias of Group
GroupListGrid.num_rows_per_page = 50
GroupListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc0725ae3d0>, <galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc0725ae3d0>]
GroupListGrid.preserve_state = False
GroupListGrid.standard_filters = [<galaxy.web.framework.helpers.grids.GridColumnFilter object at 0x7fc0725ae3d0>]
GroupListGrid.template = '/admin/dataset_security/group/grid.mako'
GroupListGrid.title = 'Groups'

```

```

    GroupListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.admin.QuotaListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class AmountColumn (label, key=None, model_class=None, method=None, format=None, link=None,
                        attach_popup=False, visible=True, nowrap=False, filterable=None,
                        sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, quota)

    class QuotaListGrid.DescriptionColumn (label, key=None, model_class=None,
                                           method=None, format=None, link=None,
                                           attach_popup=False, visible=True, nowrap=False,
                                           filterable=None, sortable=True, label_id_prefix=None,
                                           inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, quota)

    class QuotaListGrid.GroupsColumn (label, key=None, model_class=None, method=None,
                                       format=None, link=None, attach_popup=False,
                                       visible=True, nowrap=False, filterable=None, sortable=True,
                                       label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, quota)

    class QuotaListGrid.NameColumn (label, key=None, model_class=None, method=None,
                                    format=None, link=None, attach_popup=False,
                                    visible=True, nowrap=False, filterable=None, sortable=True,
                                    label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, quota)

    class QuotaListGrid.StatusColumn (label, key=None, model_class=None, method=None,
                                       format=None, link=None, attach_popup=False,
                                       visible=True, nowrap=False, filterable=None, sortable=True,
                                       label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, quota)

    class QuotaListGrid.UsersColumn (label, key=None, model_class=None, method=None,
                                      format=None, link=None, attach_popup=False,
                                      visible=True, nowrap=False, filterable=None, sortable=True,
                                      label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, quota)

QuotaListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.NameColumn object at 0x7fc06c58f7d0>, <ga
QuotaListGrid.default_sort_key = 'name'
QuotaListGrid.global_actions = [<galaxy.web.framework.helpers.grid.GridAction object at 0x7fc06b06ba10>]
QuotaListGrid.model_class
    alias of Quota
QuotaListGrid.num_rows_per_page = 50
QuotaListGrid.operations = [<galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc06b06bb90>, <g

```

```
QuotaListGrid.preserve_state = False
QuotaListGrid.standard_filters = [<galaxy.web.framework.helpers.grid.GridColumnFilter object at 0x7fc069
QuotaListGrid.template = '/admin/quota/grid.mako'
QuotaListGrid.title = 'Quotas'
QuotaListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.admin.RoleListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class DescriptionColumn(label, key=None, model_class=None, method=None, format=None,
                             link=None, attach_popup=False, visible=True, nowrap=False, filter-
                             able=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value(trans, grid, role)

    class RoleListGrid.GroupsColumn(label, key=None, model_class=None, method=None,
                                      format=None, link=None, attach_popup=False, visi-
                                      ble=True, nowrap=False, filterable=None, sortable=True,
                                      label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value(trans, grid, role)

    class RoleListGrid.NameColumn(label, key=None, model_class=None, method=None,
                                    format=None, link=None, attach_popup=False, visi-
                                    ble=True, nowrap=False, filterable=None, sortable=True,
                                    label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value(trans, grid, role)

    class RoleListGrid.StatusColumn(label, key=None, model_class=None, method=None,
                                      format=None, link=None, attach_popup=False, visi-
                                      ble=True, nowrap=False, filterable=None, sortable=True,
                                      label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value(trans, grid, role)

    class RoleListGrid.TypeColumn(label, key=None, model_class=None, method=None,
                                    format=None, link=None, attach_popup=False, visi-
                                    ble=True, nowrap=False, filterable=None, sortable=True,
                                    label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value(trans, grid, role)

    class RoleListGrid.UsersColumn(label, key=None, model_class=None, method=None,
                                    format=None, link=None, attach_popup=False, visi-
                                    ble=True, nowrap=False, filterable=None, sortable=True,
                                    label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value(trans, grid, role)

    RoleListGrid.apply_query_filter(trans, query, **kwargs)

    RoleListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.NameColumn object at 0x7fc06c5adc90>, <gala
    RoleListGrid.default_sort_key = 'name'
```

```

RoleListGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7fc070171250>]
RoleListGrid.model_class
    alias of Role
RoleListGrid.num_rows_per_page = 50
RoleListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc070171390>, <galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc070171390>]
RoleListGrid.preserve_state = False
RoleListGrid.standard_filters = [<galaxy.web.framework.helpers.grids.GridColumnFilter object at 0x7fc070171390>]
RoleListGrid.template = '/admin/dataset_security/role/grid.mako'
RoleListGrid.title = 'Roles'
RoleListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class ToolIdColumn (label, key=None, model_class=None, method=None, format=None, link=None,
                        attach_popup=False, visible=True, nowrap=False, filterable=None,
                        sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn
        get_value (trans, grid, tool_version)

    class ToolVersionListGrid.ToolVersionsColumn (label, key=None, model_class=None,
                                                    method=None, format=None, link=None,
                                                    attach_popup=False, visible=True, nowrap=False, filterable=None,
                                                    sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn
        get_value (trans, grid, tool_version)

    ToolVersionListGrid.build_initial_query (trans, **kwd)

    ToolVersionListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.ToolIdColumn object at 0x7fc069538000>]

    ToolVersionListGrid.default_filter = {}

    ToolVersionListGrid.default_sort_key = 'tool_id'

    ToolVersionListGrid.global_actions = []

    ToolVersionListGrid.model_class
        alias of ToolVersion
    ToolVersionListGrid.num_rows_per_page = 50
    ToolVersionListGrid.operations = []
    ToolVersionListGrid.preserve_state = False
    ToolVersionListGrid.standard_filters = []
    ToolVersionListGrid.template = '/admin/tool_version/grid.mako'
    ToolVersionListGrid.title = 'Tool versions'
    ToolVersionListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.admin.UserListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

```



```
class ActivatedColumn(label, key=None, model_class=None, method=None, format=None,
                      link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.EmailColumn(label, key=None, model_class=None, method=None,
                               format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                               label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.TextColumn

    get_value(trans, grid, user)

class UserListGrid.ExternalColumn(label, key=None, model_class=None, method=None,
                                  format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                  sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.GroupsColumn(label, key=None, model_class=None, method=None,
                                format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                                label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.LastLoginColumn(label, key=None, model_class=None, method=None,
                                   format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                   sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.RolesColumn(label, key=None, model_class=None, method=None,
                               format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                               label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.StatusColumn(label, key=None, model_class=None, method=None,
                                format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                                label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.TimeCreatedColumn(label, key=None, model_class=None,
                                     method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False,
                                     filterable=None, sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)
```



```

class UserListGrid.UserNameColumn(label, key=None, model_class=None, method=None,
                                   format=None, link=None, attach_popup=False,
                                   visible=True, nowrap=False, filterable=None,
                                   sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grid.TextColumn
    get_value(trans, grid, user)

UserListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.EmailColumn object at 0x7fc0750b79d0>, <galaxy.webapps.galaxy.controllers.admin.UserNameColumn object at 0x7fc0750b79d0>]
UserListGrid.default_sort_key = 'email'
UserListGrid.get_current_item(trans, **kwargs)
UserListGrid.global_actions = [<galaxy.web.framework.helpers.grid.GridAction object at 0x7fc0729dbb90>]
UserListGrid.model_class = User
    alias of User
UserListGrid.num_rows_per_page = 50
UserListGrid.operations = [<galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc0729db910>, <galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc0729db910>]
UserListGrid.preserve_state = False
UserListGrid.standard_filters = [<galaxy.web.framework.helpers.grid.GridColumnFilter object at 0x7fc0729db910>]
UserListGrid.template = '/admin/user/grid.mako'
UserListGrid.title = 'Users'
UserListGrid.use_paging = True

```

admin_toolshed Module

```

class galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed(app)
    Bases: galaxy.webapps.galaxy.controllers.admin.AdminGalaxy
    activate_repository(trans, *args, **kwargs)
        Activate a repository that was deactivated but not uninstalled.
    browse_repositories(trans, *args, **kwargs)
    browse_repository(trans, *args, **kwargs)
    browse_tool_dependency(trans, *args, **kwargs)
    browse_tool_shed(trans, *args, **kwargs)
    browse_tool_sheds(trans, *args, **kwargs)
    check_for_updates(trans, *args, **kwargs)
        Send a request to the relevant tool shed to see if there are any updates.
    deactivate_or_uninstall_repository(trans, *args, **kwargs)
        Handle all changes when a tool shed repository is being deactivated or uninstalled. Notice that if the repository contents include a file named tool_data_table_conf.xml.sample, its entries are not removed from the defined config.shed_tool_data_table_config. This is because it becomes a bit complex to determine if other installed repositories include tools that require the same entry. For now we'll never delete entries from config.shed_tool_data_table_config, but we may choose to do so in the future if it becomes necessary.
    display_image_in_repository(trans, **kwd)
        Open an image file that is contained in an installed tool shed repository or that is referenced by a URL for display. The image can be defined in either a README.rst file contained in the repository or the

```

help section of a Galaxy tool config that is contained in the repository. The following image definitions are all supported. The former `$PATH_TO_IMAGES` is no longer required, and is now ignored. .. image:: https://raw.githubusercontent.com/galaxy/some_image.png .. image:: `$PATH_TO_IMAGES/some_image.png` .. image:: `/static/images/some_image.gif` .. image:: `some_image.jpg` .. image:: `/deep/some_image.png`

find_tools_in_tool_shed (*trans*, *args, **kwargs)

find_workflows_in_tool_shed (*trans*, *args, **kwargs)

generate_workflow_image (*trans*, *args, **kwargs)

Return an svg image representation of a workflow dictionary created when the workflow was exported.

get_file_contents (*trans*, *args, **kwargs)

get_tool_dependencies (*trans*, *args, **kwargs)

Send a request to the appropriate tool shed to retrieve the dictionary of tool dependencies defined for the received repository name, owner and changeset revision. The received repository_id is the encoded id of the installed tool shed repository in Galaxy. We need it so that we can derive the tool shed from which it was installed.

get_updated_repository_information (*trans*, *args, **kwargs)

Send a request to the appropriate tool shed to retrieve the dictionary of information required to reinstall an updated revision of an uninstalled tool shed repository.

import_workflow (*trans*, *args, **kwargs)

Import a workflow contained in an installed tool shed repository into Galaxy.

initiate_tool_dependency_installation (*trans*, *args, **kwargs)

Install specified dependencies for repository tools. The received list of tool_dependencies are the database records for those dependencies defined in the tool_dependencies.xml file (contained in the repository) that should be installed. This allows for filtering out dependencies that have not been checked for installation on the 'Manage tool dependencies' page for an installed tool shed repository.

install_latest_repository_revision (*trans*, *args, **kwargs)

Install the latest installable revision of a repository that has been previously installed.

install_tool_dependencies_with_update (*trans*, *args, **kwargs)

Updating an installed tool shed repository where new tool dependencies but no new repository dependencies are included in the updated revision.

install_tool_shed_repositories (*trans*, *args, **kwargs)

Install specified tool shed repositories.

installed_repository_grid = <tool_shed.galaxy_install.grids.admin_toolshed_grids.InstalledRepositoryGrid object>

manage_repositories (*trans*, *args, **kwargs)

manage_repository (*trans*, *args, **kwargs)

manage_repository_tool_dependencies (*trans*, *args, **kwargs)

manage_tool_dependencies (*trans*, *args, **kwargs)

monitor_repository_installation (*trans*, *args, **kwargs)

open_folder (*trans*, *args, **kwargs)

prepare_for_install (*trans*, *args, **kwargs)

purge_repository (*trans*, *args, **kwargs)

Purge a "white ghost" repository from the database.

reinstall_repository (*trans*, **args*, ***kwargs*)

Reinstall a tool shed repository that has been previously uninstalled, making sure to handle all repository and tool dependencies of the repository.

repair_repository (*trans*, **args*, ***kwargs*)

Inspect the repository dependency hierarchy for a specified repository and attempt to make sure they are all properly installed as well as each repository's tool dependencies.

repair_tool_shed_repositories (*trans*, **args*, ***kwargs*)

Repair specified tool shed repositories.

repository_installation_grid = <tool_shed.galaxy_install.grids.admin_toolshed_grids.RepositoryInstallationGrid>

repository_installation_status_updates (*trans*, **args*, ***kwargs*)

reselect_tool_panel_section (*trans*, **args*, ***kwargs*)

Select or change the tool panel section to contain the tools included in the tool shed repository being reinstalled. If there are updates available for the repository in the tool shed, the tool_dependencies and repository_dependencies associated with the updated changeset revision will have been retrieved from the tool shed and passed in the received kwd. In this case, the stored tool shed repository metadata from the Galaxy database will not be used since it is outdated.

reset_metadata_on_selected_installed_repositories (*trans*, **args*, ***kwargs*)

reset_repository_metadata (*trans*, **args*, ***kwargs*)

Reset all metadata on a single installed tool shed repository.

reset_to_install (*trans*, **args*, ***kwargs*)

An error occurred while cloning the repository, so reset everything necessary to enable another attempt.

set_tool_versions (*trans*, **args*, ***kwargs*)

Get the tool_versions from the tool shed for each tool in the installed revision of a selected tool shed repository and update the metadata for the repository's revision in the Galaxy database.

tool_dependency_grid = <tool_shed.galaxy_install.grids.admin_toolshed_grids.ToolDependencyGrid object>

tool_dependency_status_updates (*trans*, **args*, ***kwargs*)

uninstall_tool_dependencies (*trans*, **args*, ***kwargs*)

update_to_changeset_revision (*trans*, **args*, ***kwargs*)

Update a cloned repository to the latest revision possible.

update_tool_shed_status_for_installed_repository (*trans*, **args*, ***kwargs*)

view_tool_metadata (*trans*, **args*, ***kwargs*)

view_workflow (*trans*, **args*, ***kwargs*)

Retrieve necessary information about a workflow from the database so that it can be displayed in an svg image.

async Module Upload class

class galaxy.webapps.galaxy.controllers.async.**ASync** (*app*)

Bases: *galaxy.web.base.controller.BaseUIController*

default (*trans*, *tool_id=None*, *data_id=None*, *data_secret=None*, ***kwd*)

Catches the tool id and redirects as needed

index (*trans*, *tool_id=None*, *data_secret=None*, ***kwd*)

Manages asynchronous connections

cloudlaunch Module**data_admin Module****dataset Module**

class galaxy.webapps.galaxy.controllers.dataset.**DatasetInterface** (*app*)

Bases: *galaxy.web.base.controller.BaseUIController, galaxy.model.item_attrs.UsesAnnotation, galaxy.web.base.controller.UsesHistoryMixin, galaxy.web.base.controller.UsesHistoryData, galaxy.model.item_attrs.UsesItemRatings, galaxy.web.base.controller.UsesExtendedMetadata*

annotate_async (*trans, id, new_annotation=None, **kwargs*)

copy_datasets (*trans, source_history=None, source_content_ids='', target_history_id=None, target_history_ids='', new_history_name='', do_copy=False, **kwd*)

default (*trans, dataset_id=None, **kwd*)

delete (*trans, dataset_id, filename, show_deleted_on_refresh=False*)

delete_async (*trans, dataset_id, filename*)

display (*trans, dataset_id=None, preview=False, filename=None, to_ext=None, chunk=None, **kwd*)

display_application (*trans, dataset_id=None, user_id=None, app_name=None, link_name=None, app_action=None, action_param=None, **kwds*)
Access to external display applications

display_at (*trans, dataset_id, filename=None, **kwd*)
Sets up a dataset permissions so it is viewable at an external site

display_by_username_and_slug (*trans, username, slug, filename=None, preview=True*)
Display dataset by username and slug; because datasets do not yet have slugs, the slug is the dataset's id.

edit (*trans, dataset_id=None, filename=None, hid=None, **kwd*)
Allows user to modify parameters of an HDA.

errors (*trans, id*)

exit_code (*trans, dataset_id=None, **kwargs*)

get_annotation_async (*trans, id*)

get_embed_html_async (*trans, id*)
Returns HTML for embedding a dataset in a page.

get_item_content_async (*trans, id*)
Returns item content in HTML format.

get_metadata_file (*trans, hda_id, metadata_name*)
Allows the downloading of metadata files associated with datasets (eg. bai index for bam files)

get_name_and_link_async (*trans, *args, **kwargs*)
Returns dataset's name and link.

imp (*trans, dataset_id=None, **kwd*)
Import another user's dataset via a shared URL; dataset is added to user's current history.

list (*trans, *args, **kwargs*)
List all available datasets

purge (*trans, dataset_id, filename, show_deleted_on_refresh=False*)

```

purge_async (trans, dataset_id, filename)

rate_async (trans, *args, **kwargs)
    Rate a dataset asynchronously and return updated community data.

report_error (trans, id, email='', message='', **kwd)

set_accessible_async (trans, *args, **kwargs)
    Does nothing because datasets do not have an importable/accessible attribute. This method could potentially set another attribute.

show_params (trans, dataset_id=None, from_noframe=None, **kwd)
    Show the parameters used for the job associated with an HDA

stderr (trans, dataset_id=None, **kwargs)

stdout (trans, dataset_id=None, **kwargs)

stored_list_grid = <galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid object>

transfer_status (trans, *args, **kwargs)
    Primarily used for the S3ObjectStore - get the status of data transfer if the file is not in cache

undelete (trans, dataset_id, filename)

undelete_async (trans, dataset_id, filename)

unhide (trans, dataset_id, filename)
class galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class HistoryColumn (label, key=None, model_class=None, method=None, format=None,
                        link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, hda)

    class HistoryDatasetAssociationListGrid.StatusColumn (label, key=None,
                                                         model_class=None,
                                                         method=None, format=None, link=None,
                                                         attach_popup=False, visible=True, nowrap=False,
                                                         filterable=None,
                                                         sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_accepted_filters ()
            Returns a list of accepted filters for this column.

        get_value (trans, grid, hda)

HistoryDatasetAssociationListGrid.build_initial_query (trans, **kwargs)

HistoryDatasetAssociationListGrid.columns = [<galaxy.web.framework.helpers.grid.TextColumn object at
HistoryDatasetAssociationListGrid.default_filter = {'deleted': 'False', 'name': 'All', 'tags': 'All'}
HistoryDatasetAssociationListGrid.default_sort_key = '-update_time'
HistoryDatasetAssociationListGrid.model_class
    alias of HistoryDatasetAssociation

```

```
HistoryDatasetAssociationListGrid.num_rows_per_page = 50
HistoryDatasetAssociationListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation ob
HistoryDatasetAssociationListGrid.preserve_state = False
HistoryDatasetAssociationListGrid.standard_filters = []
HistoryDatasetAssociationListGrid.template = '/dataset/grid.mako'
HistoryDatasetAssociationListGrid.title = 'Saved Datasets'
HistoryDatasetAssociationListGrid.use_async = True
HistoryDatasetAssociationListGrid.use_paging = True
```

error Module

```
class galaxy.webapps.galaxy.controllers.error.Error(app)
    Bases: galaxy.web.base.controller.BaseUIController

    index(trans)
```

external_service Module

```
class galaxy.webapps.galaxy.controllers.external_service.ExternalService(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDe

    browse_external_services(trans, *args, **kwargs)
    create_external_service(trans, *args, **kwargs)
    delete_external_service(trans, *args, **kwargs)
    edit_external_service(trans, *args, **kwargs)
    edit_external_service_form_definition(trans, *args, **kwargs)
    external_service_grid = <galaxy.webapps.galaxy.controllers.external_service.ExternalServiceGrid object>
    get_external_service_type(trans, external_service_type_id, ac-
        tion='browse_external_services')
    reload_external_service_types(trans, *args, **kwargs)
    undelete_external_service(trans, *args, **kwargs)
    update_external_service_form_definition(trans, *args, **kwargs)
    view_external_service(trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.external_service.ExternalServiceGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class ExternalServiceTypeColumn(label, key=None, model_class=None, method=None,
        format=None, link=None, attach_popup=False, visi-
        ble=True, nowrap=False, filterable=None, sortable=True,
        label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value(trans, grid, external_service)
```

```
class ExternalServiceGrid.NameColumn (label,          key=None,          model_class=None,
                                         method=None,  format=None,  link=None,  at-
                                         tach_popup=False,  visible=True,  nowrap=False,
                                         filterable=None,      sortable=True,      la-
                                         bel_id_prefix=None, inbound=False)
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*

```
get_value (trans, grid, external_service)
```

```
ExternalServiceGrid.columns = [<galaxy.webapps.galaxy.controllers.external_service.NameColumn object at 0x7fc065640000>]
```

```
ExternalServiceGrid.default_filter = {'deleted': 'False'}
```

```
ExternalServiceGrid.default_sort_key = '-create_time'
```

```
ExternalServiceGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7fc065640000>]
```

```
ExternalServiceGrid.model_class
```

alias of *ExternalService*

```
ExternalServiceGrid.num_rows_per_page = 50
```

```
ExternalServiceGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc065640000>]
```

```
ExternalServiceGrid.preserve_state = True
```

```
ExternalServiceGrid.template = 'admin/external_service/grid.mako'
```

```
ExternalServiceGrid.title = 'External Services'
```

```
ExternalServiceGrid.use_paging = True
```

external_services Module

```
class galaxy.webapps.galaxy.controllers.external_services.ExternalServiceController (app)
```

Bases: *galaxy.web.base.controller.BaseUIController*

```
access_action (trans, *args, **kwargs)
```

forms Module

```
class galaxy.webapps.galaxy.controllers.forms.Forms (app)
```

Bases: *galaxy.web.base.controller.BaseUIController*

```
browse_form_definitions (trans, *args, **kwargs)
```

```
build_form_definition_field_widgets (trans, layout_grids, field_index, field, form_type)
```

This method returns a list of widgets which describes a form definition field. This includes the field label, helptext, type, selectfield options, required/optional & layout

```
create_form_definition (trans, *args, **kwargs)
```

```
delete_form_definition (trans, *args, **kwargs)
```

```
edit_form_definition (trans, *args, **kwargs)
```

This callback method is for handling form editing. The value of response_redirect should be an URL that is defined by the caller. This allows for redirecting as desired when the form changes have been saved. For an example of how this works, see the edit_template() method in the base controller.

```
empty_field = {'visible': True, 'helptext': '', 'name': '', 'default': '', 'layout': 'none', 'selectlist': [], 'required': False, 'type': ''}
```

```
forms_grid = <galaxy.webapps.galaxy.controllers.forms.FormsGrid object>
```



```

get_current_form (trans, **kwd)
    This method gets all the unsaved user-entered form details and returns a dictionary containing the name,
    desc, type, layout & fields of the form

get_saved_form (form_definition)
    This retrieves the saved form and returns a dictionary containing the name, desc, type, layout & fields of
    the form

save_form_definition (trans, form_definition_current_id=None, **kwd)
    This method saves the current form

show_editable_form_definition (trans, form_definition, current_form, message='', status='done', response_redirect=None, **kwd)
    Displays the form and any of the changes made to it in edit mode. In this method all the widgets are build
    for all name, description and all the fields of a form definition.

undelele_form_definition (trans, *args, **kwargs)

view_latest_form_definition (trans, *args, **kwargs)
    Displays the layout of the latest version of the form definition
class galaxy.webapps.galaxy.controllers.forms.FormsGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class DescriptionColumn (label, key=None, model_class=None, method=None, format=None,
        link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, form)

    class FormsGrid.NameColumn (label, key=None, model_class=None, method=None, format=None,
        link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, form)

    class FormsGrid.TypeColumn (label, key=None, model_class=None, method=None, format=None,
        link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, form)

    FormsGrid.build_initial_query (trans, **kwargs)

    FormsGrid.columns = [<galaxy.webapps.galaxy.controllers.forms.NameColumn object at 0x7fc0694a2d90>, <galaxy.w

    FormsGrid.default_filter = {'deleted': 'False'}

    FormsGrid.default_sort_key = '-create_time'

    FormsGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7fc0694a2250>]

    FormsGrid.model_class
        alias of FormDefinitionCurrent

    FormsGrid.num_rows_per_page = 50

    FormsGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc0694a23d0>, <galaxy.w

    FormsGrid.preserve_state = True

```



```

FormsGrid.template = 'admin/forms/grid.mako'

FormsGrid.title = 'Forms'

FormsGrid.use_paging = True

```

history Module

```
class galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid
```

```
    Bases: galaxy.web.framework.helpers.grid.Grid
```

```
class NameURLColumn(label, key=None, model_class=None, method=None, format=None,
                    link=None, attach_popup=False, visible=True, nowrap=False, filter-
                    able=None, sortable=True, label_id_prefix=None, inbound=False)
```

```
    Bases: galaxy.web.framework.helpers.grid.PublicURLColumn,
           galaxy.webapps.galaxy.controllers.history.NameColumn
```

```
HistoryAllPublishedGrid.apply_query_filter(trans, query, **kwargs)
```

```
HistoryAllPublishedGrid.build_initial_query(trans, **kwargs)
```

```
HistoryAllPublishedGrid.columns = [<galaxy.webapps.galaxy.controllers.history.NameURLColumn object at 0x
```

```
HistoryAllPublishedGrid.default_filter = {'username': 'All', 'public_url': 'All', 'tags': 'All'}
```

```
HistoryAllPublishedGrid.default_sort_key = 'update_time'
```

```
HistoryAllPublishedGrid.model_class
    alias of History
```

```
HistoryAllPublishedGrid.num_rows_per_page = 50
```

```
HistoryAllPublishedGrid.operations = []
```

```
HistoryAllPublishedGrid.title = 'Published Histories'
```

```
HistoryAllPublishedGrid.use_async = True
```

```
HistoryAllPublishedGrid.use_paging = True
```

```
class galaxy.webapps.galaxy.controllers.history.HistoryController(app)
```

```
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.SharableMixin,
           galaxy.model.item_attrs.UsesAnnotations, galaxy.model.item_attrs.UsesItemRatings,
           galaxy.web.base.controller.UsesHistoryMixin, galaxy.web.base.controller.UsesHistoryData,
           galaxy.web.base.controller.ExportsHistoryMixin, galaxy.web.base.controller.ImportsHistoryMixin
```

```
    citations(trans)
```

```
    copy(trans, *args, **kwargs)
        Copy one or more histories
```

```
    create_new_current(trans, *args, **kwargs)
```

```
    current_history_json(trans, *args, **kwargs)
```

```
    delete_current(trans, purge=False)
        Delete just the active history – this does not require a logged in user.
```

```
    delete_hidden_datasets(trans)
        This method deletes all hidden datasets in the current history.
```

```
    display_by_username_and_slug(trans, username, slug)
        Display history based on a username and slug.
```

display_structured (*trans*, *id=None*)
Display a history as a nested structure showing the jobs and workflow invocations that created each dataset (if any).

export_archive (*trans*, *id=None*, *gzip=True*, *include_hidden=False*, *include_deleted=False*, *pre-view=False*)
Export a history to an archive.

get_item (*trans*, *id*)

get_item_content_async (*trans*, *id*)
Returns item content in HTML format.

get_name_and_link_async (*trans*, **args*, ***kwargs*)
Returns history's name and link.

history_data (*trans*, *history*)

imp (*trans*, *id=None*, *confirm=False*, ***kwd*)
Import another user's history via a shared URL

import_archive (*trans*, **args*, ***kwargs*)
Import a history from a file archive.

index (*trans*)

list (*trans*, **args*, ***kwargs*)
List all available histories

list_as_xml (*trans*)
XML history list for functional tests

list_published (*trans*, ***kwargs*)

list_shared (*trans*, **args*, ***kwargs*)
List histories shared with current user by others

name_autocomplete_data (*trans*, *q=None*, *limit=None*, *timestamp=None*)
Return autocomplete data for history names

published_list_grid = <galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid object>

purge_deleted_datasets (*trans*)

rate_async (*trans*, **args*, ***kwargs*)
Rate a history asynchronously and return updated community data.

rename (*trans*, **args*, ***kwargs*)

resume_paused_jobs (*trans*, *current=False*, *ids=None*)
Resume paused jobs the active history – this does not require a logged in user.

set_accessible_async (*trans*, **args*, ***kwargs*)
Set history's importable attribute and slug.

set_as_current (*trans*, **args*, ***kwargs*)

share (*trans*, **args*, ***kwargs*)

share_restricted (*trans*, **args*, ***kwargs*)

shared_list_grid = <galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid object>

sharing (*trans*, **args*, ***kwargs*)
Handle history sharing.

stored_list_grid = <galaxy.webapps.galaxy.controllers.history.HistoryListGrid object>

```

structure (trans, id=None, **kwargs)

switch_to_history (trans, *args, **kwargs)

unhide_datasets (trans, current=False, ids=None)
    Unhide the datasets in the active history – this does not require a logged in user.

view (trans, id=None, show_deleted=False, show_hidden=False, use_panels=True)
    View a history. If a history is importable, then it is viewable by any user.

view_multiple (trans, include_deleted_histories=False, order='update')

class galaxy.webapps.galaxy.controllers.history.HistoryListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class DatasetsByStateColumn (label, key=None, model_class=None, method=None,
                                format=None, link=None, attach_popup=False, visible=True,
                                nowrap=False, filterable=None, sortable=True,
                                label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn, galaxy.web.base.controller.UsesHistoryMixin

        get_value (trans, grid, history)

    class HistoryListGrid.DeletedColumn (label, key=None, model_class=None, method=None,
                                          format=None, link=None, attach_popup=False,
                                          visible=True, nowrap=False, filterable=None,
                                          sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.DeletedColumn

        get_value (trans, grid, history)

        sort (trans, query, ascending, column_name=None)

    class HistoryListGrid.HistoryListNameColumn (label, key=None, model_class=None,
                                                  method=None, format=None, link=None,
                                                  attach_popup=False, visible=True,
                                                  nowrap=False, filterable=None,
                                                  sortable=True, label_id_prefix=None,
                                                  inbound=False)
        Bases: galaxy.webapps.galaxy.controllers.history.NameColumn

        get_link (trans, grid, history)

    HistoryListGrid.apply_query_filter (trans, query, **kwargs)

    HistoryListGrid.columns = [<galaxy.webapps.galaxy.controllers.history.HistoryListNameColumn object at 0x7fc0...>]

    HistoryListGrid.default_filter = {'deleted': 'False', 'sharing': 'All', 'name': 'All', 'tags': 'All'}

    HistoryListGrid.default_sort_key = '-update_time'

    HistoryListGrid.get_current_item (trans, **kwargs)

    HistoryListGrid.info_text = 'Histories that have been deleted for more than a time period specified by the Galaxy'

    HistoryListGrid.model_class
        alias of History

    HistoryListGrid.num_rows_per_page = 50

    HistoryListGrid.operations = [<galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc0663a8ed0>]

    HistoryListGrid.preserve_state = False

```

```
HistoryListGrid.standard_filters = [<galaxy.web.framework.helpers.grids.GridColumnFilter object at 0x7fc0
HistoryListGrid.template = '/history/grid.mako'
HistoryListGrid.title = 'Saved Histories'
HistoryListGrid.use_async = True
HistoryListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.history.NameColumn (label,          key=None,
                                                            model_class=None,
                                                            method=None,          for-
                                                            mat=None,   link=None,
                                                            attach_popup=False,
                                                            visible=True,
                                                            nowrap=False,
                                                            filterable=None,
                                                            sortable=True,          la-
                                                            bel_id_prefix=None,
                                                            inbound=False)

Bases: galaxy.web.framework.helpers.grids.TextColumn

get_value (trans, grid, history)

class galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid
Bases: galaxy.web.framework.helpers.grids.Grid

class DatasetsByStateColumn (label,   key=None,   model_class=None,   method=None,
                             format=None,   link=None,   attach_popup=False,   visi-
                             ble=True,   nowrap=False,   filterable=None,   sortable=True,
                             label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

get_value (trans, grid, history)

class SharedHistoryListGrid.SharedByColumn (label,   key=None,   model_class=None,
                                             method=None, format=None, link=None,
                                             attach_popup=False,   visible=True,
                                             nowrap=False,   filterable=None,
                                             sortable=True,   label_id_prefix=None,
                                             inbound=False)
Bases: galaxy.web.framework.helpers.grids.GridColumn

get_value (trans, grid, history)

SharedHistoryListGrid.apply_query_filter (trans, query, **kwargs)
SharedHistoryListGrid.build_initial_query (trans, **kwargs)
SharedHistoryListGrid.columns = [<galaxy.web.framework.helpers.grids.GridColumn object at 0x7fc0753a8b10>
SharedHistoryListGrid.default_filter = {}
SharedHistoryListGrid.default_sort_key = '-update_time'
SharedHistoryListGrid.model_class
    alias of History
SharedHistoryListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc0662
SharedHistoryListGrid.standard_filters = []
SharedHistoryListGrid.title = 'Histories shared with you by others'
```

library Module

```

class galaxy.webapps.galaxy.controllers.library.Library (app)
    Bases: galaxy.web.base.controller.BaseUIController

    browse_libraries (trans, **kwd)

    index (trans, **kwd)

    library_list_grid = <galaxy.webapps.galaxy.controllers.library.LibraryListGrid object>

    list (trans, **kwd)
class galaxy.webapps.galaxy.controllers.library.LibraryListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class DescriptionColumn (label, key=None, model_class=None, method=None, format=None,
                             link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, library)

    class LibraryListGrid.NameColumn (label, key=None, model_class=None, method=None,
                                       format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                                       label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, library)

    LibraryListGrid.apply_query_filter (trans, query, **kwd)

    LibraryListGrid.build_initial_query (trans, **kwargs)

    LibraryListGrid.columns = [<galaxy.webapps.galaxy.controllers.library.NameColumn object at 0x7fc066da8ad0>,
    LibraryListGrid.default_filter = {'deleted': 'False', 'description': 'All', 'purged': 'False', 'name': 'All'}

    LibraryListGrid.default_sort_key = 'name'

    LibraryListGrid.model_class
        alias of Library

    LibraryListGrid.num_rows_per_page = 50

    LibraryListGrid.preserve_state = False

    LibraryListGrid.standard_filters = []

    LibraryListGrid.template = '/library/grid.mako'

    LibraryListGrid.title = 'Data Libraries'

    LibraryListGrid.use_paging = True

```

library_admin Module

```

class galaxy.webapps.galaxy.controllers.library_admin.LibraryAdmin (app)
    Bases: galaxy.web.base.controller.BaseUIController

    browse_libraries (trans, *args, **kwargs)

    create_library (trans, *args, **kwargs)

    delete_library (trans, *args, **kwargs)

    library_list_grid = <galaxy.webapps.galaxy.controllers.library_admin.LibraryListGrid object>

```

```
purge_library (trans, *args, **kwargs)
undelele_library (trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.library_admin.LibraryListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class DescriptionColumn (label, key=None, model_class=None, method=None, format=None,
                              link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, library)

    class LibraryListGrid.NameColumn (label, key=None, model_class=None, method=None,
                                        format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                                        label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, library)

    class LibraryListGrid.StatusColumn (label, key=None, model_class=None, method=None,
                                        format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, library)

    LibraryListGrid.columns = [<galaxy.webapps.galaxy.controllers.library_admin.NameColumn object at 0x7fc07470
    LibraryListGrid.default_filter = {'deleted': 'False', 'description': 'All', 'purged': 'False', 'name': 'All'}
    LibraryListGrid.default_sort_key = 'name'
    LibraryListGrid.global_actions = [<galaxy.web.framework.helpers.grid.GridAction object at 0x7fc073f80a90>
    LibraryListGrid.model_class
        alias of Library

    LibraryListGrid.num_rows_per_page = 50
    LibraryListGrid.preserve_state = False
    LibraryListGrid.standard_filters = [<galaxy.web.framework.helpers.grid.GridColumnFilter object at 0x7fc0
    LibraryListGrid.template = '/admin/library/grid.mako'
    LibraryListGrid.title = 'Data Libraries'
    LibraryListGrid.use_paging = True
```

library_common Module

```
class galaxy.webapps.galaxy.controllers.library_common.LibraryCommon (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDe
    galaxy.web.base.controller.UsesExtendedMetadataMixin,
    galaxy.web.base.controller.UsesLibraryMixinItems

    act_on_multiple_datasets (trans, cntrller, library_id=None, ldda_ids='', **kwd)
    add_history_datasets_to_library (trans, cntrller, library_id, folder_id, hda_ids='', **kwd)
    browse_library (trans, cntrller='library', **kwd)
    create_folder (trans, cntrller, parent_id, library_id, **kwd)
```

```

delete_library_item (trans, cntrller, library_id, item_id, item_type, **kwd)

download_dataset_from_folder (trans, cntrller, id, library_id=None, **kwd)
    Catches the dataset id and displays file contents as directed

folder_info (trans, cntrller, id, library_id, **kwd)

folder_permissions (trans, cntrller, id, library_id, **kwd)

get_path_paste_uploaded_datasets (trans, cntrller, params, library_bunch, response_code,
    message)

get_server_dir_uploaded_datasets (trans, cntrller, params, full_dir, import_dir_desc, li-
    brary_bunch, response_code, message)

import_datasets_to_histories (trans, cntrller, library_id='', folder_id='',
    ldda_ids='', target_history_id='', target_history_ids='',
    new_history_name='', **kwd)

ldda_edit_info (trans, cntrller, library_id, folder_id, id, **kwd)

ldda_info (trans, cntrller, library_id, folder_id, id, **kwd)

ldda_permissions (trans, cntrller, library_id, folder_id, id, **kwd)

library_dataset_info (trans, cntrller, id, library_id, **kwd)

library_dataset_permissions (trans, cntrller, id, library_id, **kwd)

library_info (trans, cntrller, **kwd)

library_item_updates (trans, *args, **kwargs)

library_permissions (trans, cntrller, **kwd)

make_library_item_public (trans, cntrller, library_id, item_type, id, **kwd)

make_library_uploaded_dataset (trans, cntrller, params, name, path, type, library_bunch,
    in_folder=None)

manage_template_inheritance (trans, cntrller, item_type, library_id, folder_id=None,
    ldda_id=None, **kwd)

move_library_item (trans, cntrller, item_type, item_id, source_library_id='',
    make_target_current=True, **kwd)

undelete_library_item (trans, cntrller, library_id, item_id, item_type, **kwd)

upload_dataset (trans, cntrller, library_id, folder_id, replace_dataset=None, **kwd)

upload_library_dataset (trans, cntrller, library_id, folder_id, **kwd)
galaxy.webapps.galaxy.controllers.library_common.activatable_folders (trans,
    folder)

galaxy.webapps.galaxy.controllers.library_common.activatable_folders_and_library_datasets (trans,
    folder)

galaxy.webapps.galaxy.controllers.library_common.active_folders (trans, folder)

galaxy.webapps.galaxy.controllers.library_common.active_folders_and_library_datasets (trans,
    folder)

galaxy.webapps.galaxy.controllers.library_common.branch_deleted (folder)

galaxy.webapps.galaxy.controllers.library_common.datasets_for_lddas (trans, ldda-
    das)
    Given a list of LDDAs, return a list of Datasets for them.

galaxy.webapps.galaxy.controllers.library_common.get_comptypes (trans)

```

```
galaxy.webapps.galaxy.controllers.library_common.get_containing_library_from_library_dataset
```

Given a library_dataset, get the containing library

```
galaxy.webapps.galaxy.controllers.library_common.get_sorted_accessible_library_items (trans,
cn-
tr-
ller,
items,
sort_att
```

```
galaxy.webapps.galaxy.controllers.library_common.lucene_search (trans, cntrlr,
search_term,
search_url,
**kwd)
```

Return display of results from a full-text lucene search of data libraries.

```
galaxy.webapps.galaxy.controllers.library_common.map_library_datasets_to_lddas (trans,
lib_datasets)
```

Given a list of LibraryDatasets, return a map from the LibraryDatasets to their LDDAs. If an LDDA does not exist for a LibraryDataset, then there will be no entry in the return hash.

```
galaxy.webapps.galaxy.controllers.library_common.sort_by_attr (seq, attr)
Sort the sequence of objects by object's attribute Arguments: seq - the list or any sequence (including immutable
one) of objects to sort. attr - the name of attribute to sort by
```

```
galaxy.webapps.galaxy.controllers.library_common.whoosh_search (trans, cntrlr,
search_term,
**kwd)
```

Return display of results from a full-text whoosh search of data libraries.

mobile Module

```
class galaxy.webapps.galaxy.controllers.mobile.Mobile (app)
```

Bases: *galaxy.web.base.controller.BaseUIController*

```
dataset_detail (trans, id)
```

```
dataset_peek (trans, id)
```

```
history_detail (trans, id)
```

```
history_list (trans)
```

```
index (trans, **kwargs)
```

```
settings (trans, email=None, password=None)
```

page Module

```
class galaxy.webapps.galaxy.controllers.page.HistoryDatasetAssociationSelectionGrid
```

Bases: *galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid*

Grid for selecting HDAs.

```
apply_query_filter (trans, query, **kwargs)
```

```
columns = [<galaxy.webapps.galaxy.controllers.page.NameColumn object at 0x7fc06434df10>, <galaxy.web.framework.l
```

```
model_class
```

alias of HistoryDatasetAssociation

```
title = 'Saved Datasets'
```



```

class galaxy.webapps.galaxy.controllers.page.HistorySelectionGrid
    Bases: galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid

    Grid for selecting histories.

    apply_query_filter (trans, query, **kwargs)

    columns = [<galaxy.webapps.galaxy.controllers.page.NameColumn object at 0x7fc06434dd50>, <galaxy.web.framework.

    model_class
        alias of History

    title = 'Saved Histories'

class galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    Base class for pages' item selection grids.

    class NameColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, item)

    ItemSelectionGrid.apply_query_filter (trans, query, **kwargs)
    ItemSelectionGrid.default_filter = {'deleted': 'False', 'sharing': 'All'}
    ItemSelectionGrid.default_sort_key = '-update_time'
    ItemSelectionGrid.num_rows_per_page = 10
    ItemSelectionGrid.show_item_checkboxes = True
    ItemSelectionGrid.template = '/page/select_items_grid.mako'
    ItemSelectionGrid.use_async = True
    ItemSelectionGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    apply_query_filter (trans, query, **kwargs)

    build_initial_query (trans, **kwargs)

    columns = [<galaxy.web.framework.helpers.grid.PublicURLColumn object at 0x7fc06434d950>, <galaxy.web.framework.

    default_filter = {'username': 'All', 'title': 'All'}

    default_sort_key = 'update_time'

    model_class
        alias of Page

    title = 'Published Pages'

    use_async = True

    use_panels = True

class galaxy.webapps.galaxy.controllers.page.PageController (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.SharableMixin, galaxy.web.base.controller.UsesHistoryMixin, galaxy.web.base.controller.UsesStoredWorkf

```

galaxy.web.base.controller.UsesVisualizationMixin, galaxy.model.item_attrs.UsesItemRati

create (*trans*, **args*, ***kwargs*)

Create a new page

display (*trans*, **args*, ***kwargs*)

display_by_username_and_slug (*trans*, *username*, *slug*)

Display page based on a username and slug.

edit (*trans*, **args*, ***kwargs*)

Edit a page's attributes.

edit_content (*trans*, **args*, ***kwargs*)

Render the main page editor interface.

get_editor_iframe (*trans*)

Returns the document for the page editor's iframe.

get_embed_html_async (*trans*, *id*)

Returns HTML for embedding a workflow in a page.

get_item (*trans*, *id*)

get_name_and_link_async (*trans*, **args*, ***kwargs*)

Returns page's name and link.

get_page (*trans*, *id*, *check_ownership=True*, *check_accessible=False*)

Get a page from the database by id.

list (*trans*, **args*, ***kwargs*)

List user's pages.

list_datasets_for_selection (*trans*, **args*, ***kwargs*)

Returns HTML that enables a user to select one or more datasets.

list_histories_for_selection (*trans*, **args*, ***kwargs*)

Returns HTML that enables a user to select one or more histories.

list_pages_for_selection (*trans*, **args*, ***kwargs*)

Returns HTML that enables a user to select one or more pages.

list_published (*trans*, **args*, ***kwargs*)

list_visualizations_for_selection (*trans*, **args*, ***kwargs*)

Returns HTML that enables a user to select one or more visualizations.

list_workflows_for_selection (*trans*, **args*, ***kwargs*)

Returns HTML that enables a user to select one or more workflows.

rate_async (*trans*, **args*, ***kwargs*)

Rate a page asynchronously and return updated community data.

save (*trans*, **args*, ***kwargs*)

set_accessible_async (*trans*, **args*, ***kwargs*)

Set page's importable attribute and slug.

share (*trans*, **args*, ***kwargs*)

Handle sharing with an individual user.

sharing (*trans*, **args*, ***kwargs*)

Handle page sharing.

```

class galaxy.webapps.galaxy.controllers.page.PageListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class URLColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.PublicURLColumn

        get_value (trans, grid, item)

    PageListGrid.apply_query_filter (trans, query, **kwargs)

    PageListGrid.columns = [<galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc06434d250>, <galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc06434d250>]

    PageListGrid.default_filter = {'title': 'All', 'sharing': 'All', 'tags': 'All', 'published': 'All'}

    PageListGrid.default_sort_key = '-update_time'

    PageListGrid.global_actions = [<galaxy.web.framework.helpers.grid.GridAction object at 0x7fc06434d4d0>]

    PageListGrid.model_class
        alias of Page

    PageListGrid.operations = [<galaxy.web.framework.helpers.grid.DisplayByUsernameAndSlugGridOperation object at 0x7fc06434d4d0>]

    PageListGrid.title = 'Pages'

    PageListGrid.use_panels = True

class galaxy.webapps.galaxy.controllers.page.PageSelectionGrid
    Bases: galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid

    Grid for selecting pages.

    columns = [<galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc06432e290>, <galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc06432e290>]

    model_class
        alias of Page

    title = 'Saved Pages'

class galaxy.webapps.galaxy.controllers.page.VisualizationSelectionGrid
    Bases: galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid

    Grid for selecting visualizations.

    columns = [<galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc06432e450>, <galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc06432e450>]

    model_class
        alias of Visualization

    title = 'Saved Visualizations'

class galaxy.webapps.galaxy.controllers.page.WorkflowSelectionGrid
    Bases: galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid

    Grid for selecting workflows.

    columns = [<galaxy.webapps.galaxy.controllers.page.NameColumn object at 0x7fc065095690>, <galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc065095690>]

    model_class
        alias of StoredWorkflow

    title = 'Saved Workflows'

galaxy.webapps.galaxy.controllers.page.format_bool (b)

```

request_type Module**class** galaxy.webapps.galaxy.controllers.request_type.**RequestType** (app)Bases: *galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDe***browse_request_types** (trans, *args, **kwargs)**create_request_type** (trans, *args, **kwargs)**delete_request_type** (trans, *args, **kwargs)**edit_request_type** (trans, *args, **kwargs)**request_type_grid** = <galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid object>**request_type_permissions** (trans, *args, **kwargs)**undelele_request_type** (trans, *args, **kwargs)**view_editable_request_type** (trans, *args, **kwargs)**view_form_definition** (trans, *args, **kwargs)**view_request_type** (trans, *args, **kwargs)**class** galaxy.webapps.galaxy.controllers.request_type.**RequestTypeGrid**Bases: *galaxy.web.framework.helpers.grid.Grid***class** **DescriptionColumn** (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)Bases: *galaxy.web.framework.helpers.grid.TextColumn***get_value** (trans, grid, request_type)**class** RequestTypeGrid.**ExternalServiceColumn** (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)Bases: *galaxy.web.framework.helpers.grid.IntegerColumn***get_value** (trans, grid, request_type)**class** RequestTypeGrid.**NameColumn** (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)Bases: *galaxy.web.framework.helpers.grid.TextColumn***get_value** (trans, grid, request_type)**class** RequestTypeGrid.**RequestFormColumn** (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)Bases: *galaxy.web.framework.helpers.grid.TextColumn***get_value** (trans, grid, request_type)

```
class RequestTypeGrid.SampleFormColumn (label,          key=None,          model_class=None,
                                         method=None, format=None, link=None, at-
                                         tach_popup=False, visible=True, nowrap=False,
                                         filterable=None,          sortable=True,          la-
                                         bel_id_prefix=None, inbound=False)
```

```
Bases: galaxy.web.framework.helpers.grids.TextColumn
```

```
get_value (trans, grid, request_type)
```

```
RequestTypeGrid.columns = [<galaxy.webapps.galaxy.controllers.request_type.NameColumn object at 0x7fc06434d
```

```
RequestTypeGrid.default_filter = {'deleted': 'False'}
```

```
RequestTypeGrid.default_sort_key = '-create_time'
```

```
RequestTypeGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7fc063c70a90>
```

```
RequestTypeGrid.model_class
```

```
alias of RequestType
```

```
RequestTypeGrid.num_rows_per_page = 50
```

```
RequestTypeGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc063c70950>
```

```
RequestTypeGrid.preserve_state = True
```

```
RequestTypeGrid.template = 'admin/request_type/grid.mako'
```

```
RequestTypeGrid.title = 'Request Types'
```

```
RequestTypeGrid.use_paging = True
```

requests Module

```
class galaxy.webapps.galaxy.controllers.requests.Requests (app)
```

```
Bases: galaxy.web.base.controller.BaseUIController
```

```
browse_requests (trans, **kwd)
```

```
find_samples_index (trans, *args, **kwargs)
```

```
index (trans, *args, **kwargs)
```

```
request_grid = <galaxy.webapps.galaxy.controllers.requests.UserRequestsGrid object>
```

```
class galaxy.webapps.galaxy.controllers.requests.UserRequestsGrid
```

```
Bases: galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid
```

```
apply_query_filter (trans, query, **kwd)
```

```
operation = <galaxy.web.framework.helpers.grids.GridOperation object>
```

```
operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc065695590>, <galaxy.web.framework
```

requests_admin Module

```
class galaxy.webapps.galaxy.controllers.requests_admin.AdminRequestsGrid
```

```
Bases: galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid
```

```
class UserColumn (label, key=None, model_class=None, method=None, format=None, link=None, at-
                  tach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                  label_id_prefix=None, inbound=False)
```

```
Bases: galaxy.web.framework.helpers.grids.TextColumn
```

```
get_value (trans, grid, request)
```

```
AdminRequestsGrid.col = <galaxy.web.framework.helpers.grids.MulticolFilterColumn object>
```

```

AdminRequestsGrid.columns = [<galaxy.webapps.galaxy.controllers.requests_common.NameColumn object at 0x7fc063849d50>]
AdminRequestsGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7fc063849d50>]
AdminRequestsGrid.operation = <galaxy.web.framework.helpers.grids.GridOperation object>
AdminRequestsGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc065695590>]
class galaxy.webapps.galaxy.controllers.requests_admin.DataTransferGrid
Bases: galaxy.web.framework.helpers.grids.Grid

class ExternalServiceColumn(label, key=None, model_class=None, method=None,
                             format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                             label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.TextColumn

    get_value(trans, grid, sample_dataset)

class DataTransferGrid.NameColumn(label, key=None, model_class=None, method=None,
                                   format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                   sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.TextColumn

    get_value(trans, grid, sample_dataset)

class DataTransferGrid.SizeColumn(label, key=None, model_class=None, method=None,
                                   format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                   sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.TextColumn

    get_value(trans, grid, sample_dataset)

class DataTransferGrid.StatusColumn(label, key=None, model_class=None, method=None,
                                     format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                     sortable=True, label_id_prefix=None, inbound=False)
Bases: galaxy.web.framework.helpers.grids.TextColumn

    get_value(trans, grid, sample_dataset)

DataTransferGrid.apply_query_filter(trans, query, **kwd)
DataTransferGrid.columns = [<galaxy.webapps.galaxy.controllers.requests_admin.NameColumn object at 0x7fc063849d50>]
DataTransferGrid.default_sort_key = 'create_time'
DataTransferGrid.model_class
    alias of SampleDataset
DataTransferGrid.num_rows_per_page = 50
DataTransferGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc063849e50>]
DataTransferGrid.preserve_state = True
DataTransferGrid.template = 'admin/requests/sample_datasets_grid.mako'
DataTransferGrid.title = 'Sample Datasets'
DataTransferGrid.use_paging = False

```

```

class galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDe

    browse_requests (trans, *args, **kwargs)

    datatx_grid = <galaxy.webapps.galaxy.controllers.requests_admin.DataTransferGrid object>

    get_file_details (trans, *args, **kwargs)

    index (trans, *args, **kwargs)

    initiate_data_transfer (trans, *args, **kwargs)

    manage_datasets (trans, *args, **kwargs)

    open_folder (trans, *args, **kwargs)

    reject_request (trans, *args, **kwargs)

    rename_datasets (trans, *args, **kwargs)

    request_grid = <galaxy.webapps.galaxy.controllers.requests_admin.AdminRequestsGrid object>

    select_datasets_to_transfer (trans, *args, **kwargs)

    update_sample_dataset_status (trans, cntrller, sample_dataset_ids, new_status, error_msg=None)

galaxy.webapps.galaxy.controllers.requests_admin.build_rename_datasets_for_sample_select_f

```

requests_common Module

```

class galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDe

    add_sample (trans, *args, **kwargs)

    add_samples (trans, *args, **kwargs)

    create_request (trans, *args, **kwargs)

    dataset_transfer_status_updates (trans, *args, **kwargs)

    delete_request (trans, *args, **kwargs)

    delete_sample (trans, *args, **kwargs)

    edit_basic_request_info (trans, *args, **kwargs)

    edit_email_settings (trans, *args, **kwargs)
        Allow for changing the email notification settings where email is sent to a list of users whenever the
        request state changes to one selected for notification.

    edit_samples (trans, *args, **kwargs)

    find_samples (trans, *args, **kwargs)

    sample_datasets_updates (trans, *args, **kwargs)

    sample_state_updates (trans, *args, **kwargs)

    submit_request (trans, *args, **kwargs)

```



```

undeleter_request (trans, *args, **kwargs)
update_request_state (trans, *args, **kwargs)
update_sample_state (trans, cntrller, sample_ids, new_state, comment=None)
view_request (trans, *args, **kwargs)
view_request_history (trans, *args, **kwargs)
view_sample (trans, *args, **kwargs)
view_sample_datasets (trans, *args, **kwargs)
view_sample_history (trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class DescriptionColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, request)

    class RequestsGrid.NameColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, request)

    class RequestsGrid.SamplesColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, request)

    class RequestsGrid.StateColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.StateColumn

        filter (trans, user, query, column_filter)
            Modify query to filter request by state.

        get_value (trans, grid, request)

    class RequestsGrid.TypeColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, request)

    RequestsGrid.columns = [<galaxy.webapps.galaxy.controllers.requests_common.NameColumn object at 0x7fc06562
    RequestsGrid.default_filter = {'deleted': 'False', 'state': 'All'}
    RequestsGrid.default_sort_key = '-update_time'

```



```

RequestsGrid.model_class
    alias of Request

RequestsGrid.num_rows_per_page = 50

RequestsGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc065695590>]

RequestsGrid.template = 'requests/grid.mako'

RequestsGrid.title = 'Sequencing Requests'

RequestsGrid.use_paging = True

galaxy.webapps.galaxy.controllers.requests_common.invalid_id_redirect (trans,
                                                                    cntr-
                                                                    ller,
                                                                    obj_id,
                                                                    item='sequencing
                                                                    re-
                                                                    quest',
                                                                    ac-
                                                                    tion='browse_requests')

```

root Module Contains the main interface in the Universe class

```

class galaxy.webapps.galaxy.controllers.root.RootController (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesHistory,
           galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin,
           galaxy.model.item_attrs.UsesAnnotations

    Controller class that maps to the url root of Galaxy (i.e. '/').

    bucket_proxy (trans, bucket=None, **kwd)

    clear_history (trans)
        Clears the history for a user.

    dataset_make_primary (trans, id=None)
        Copies a dataset and makes primary.

    default (trans, target1=None, target2=None, **kwd)
        Called on any url that does not match a controller method.

    display (trans, id=None, hid=None, tofile=None, toext='.txt', encoded_id=None, **kwd)
        Returns data directly into the browser.

        Sets the mime-type according to the extension.

        Used by the twill tool test driver - used anywhere else? Would like to drop hid argument and path if
        unneeded now. Likewise, would like to drop encoded_id=XXX and use assume id is encoded (likely id
        wouldn't be coming in encoded if this is used anywhere else though.)

    display_as (trans, id=None, display_app=None, **kwd)
        Returns a file in a format that can successfully be displayed in display_app.

    display_child (trans, parent_id=None, designation=None, tofile=None, toext='.txt')
        Returns child data directly into the browser, based upon parent_id and designation.

    echo (trans, **kwd)
        Echos parameters (debugging).

```

echo_json (*trans*, *args, **kwargs)

Echos parameters as JSON (debugging).

Attempts to parse values passed as boolean, float, then int. Defaults to string. Non-recursive (will not parse lists).

generate_error (*trans*, code=500)

Raises an exception (debugging).

generate_json_error (*trans*, *args, **kwargs)

Raises an exception (debugging).

history (*trans*, as_xml=False, show_deleted=None, show_hidden=None, **kwd)

Display the current history in its own page or as xml.

history_add_to (*trans*, history_id=None, file_data=None, name='Data Added to History',
info=None, ext='txt', dbkey='?', copy_access_from=None, **kwd)

Adds a POSTed file to a History.

history_as_xml (*trans*, show_deleted=None, show_hidden=None)

history_delete (*trans*, id)

Backward compatibility with check_galaxy script.

history_import (*trans*, id=None, confirm=False, **kwd)

history_new (*trans*, name=None)

Create a new history with the given name and refresh the history panel.

history_options (*trans*)

Displays a list of history related actions.

history_set_default_permissions (*trans*, id=None, **kwd)

Sets the permissions on a history.

index (*trans*, id=None, tool_id=None, mode=None, workflow_id=None, m_c=None, m_a=None,
**kwd)

Called on the root url to display the main Galaxy page.

peek (*trans*, id=None)

Returns a 'peek' at the data.

tool_help (*trans*, id)

Return help page for tool identified by 'id' if available

tool_search (*trans*, *args, **kwargs)

Searches the tool database and returns data for any tool whose text matches the query.

Data are returned in JSON format.

welcome (*trans*)

tag Module Tags Controller: handles tagging/untagging of entities and provides autocomplete support.

class galaxy.webapps.galaxy.controllers.tag.**TagsController** (*app*)

Bases: *galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesTagsMix*

add_tag_async (*trans*, *args, **kwargs)

Add tag to an item.

get_tagging_elt_async (*trans*, *args, **kwargs)

Returns HTML for editing an item's tags.

remove_tag_async (*trans*, **args*, ***kwargs*)

Remove tag from an item.

retag_async (*trans*, **args*, ***kwargs*)

Apply a new set of tags to an item; previous tags are deleted.

tag_autocomplete_data (*trans*, **args*, ***kwargs*)

Get autocomplete data for an item's tags.

tool_runner Module Upload class

class galaxy.webapps.galaxy.controllers.tool_runner.**AddFrameData**

class galaxy.webapps.galaxy.controllers.tool_runner.**ToolRunner** (*app*)

Bases: *galaxy.web.base.controller.BaseUIController*

biomart (*trans*, *tool_id*=*'biomart'*, ***kwd*)

Catches the tool id and redirects as needed

data_source_redirect (*trans*, *tool_id*=*None*)

Redirects a user accessing a Data Source tool to its target action link. This method will subvert mix-mode content blocking in several browsers when accessing non-https data_source tools from an https galaxy server.

Tested as working on Safari 7.0 and FireFox 26 Subverting did not work on Chrome 31

default (*trans*, *tool_id*=*None*, ***kwd*)

Catches the tool id and redirects as needed

hapmapmart (*trans*, *tool_id*=*'hapmapmart'*, ***kwd*)

Catches the tool id and redirects as needed

index (*trans*, *tool_id*=*None*, *from_noframe*=*None*, ***kwd*)

redirect (*trans*, *redirect_url*=*None*, ***kwd*)

rerun (*trans*, *id*=*None*, *from_noframe*=*None*, *job_id*=*None*, ***kwd*)

Given a HistoryDatasetAssociation id, find the job and that created the dataset, extract the parameters, and display the appropriate tool form with parameters already filled in.

upload_async_create (*trans*, **args*, ***kwargs*)

Precreate datasets for asynchronous uploading.

upload_async_message (*trans*, ***kwd*)

ucsc_proxy Module Contains the UCSC proxy

class galaxy.webapps.galaxy.controllers.ucsc_proxy.**UCSCProxy** (*app*)

Bases: *galaxy.web.base.controller.BaseUIController*

create_display (*store*)

Creates a more meaningful display name

index (*trans*, *init*=*False*, ***kwd*)

user Module Contains the user interface in the Universe class

class galaxy.webapps.galaxy.controllers.user.**User** (*app*)

Bases: *galaxy.web.base.controller.BaseUIController*, *galaxy.web.base.controller.UsesFormDe*
galaxy.web.base.controller.CreatesUsersMixin, *galaxy.web.base.controller.CreatesApiKeys*

activate (*trans*, ***kwd*)
Check whether token fits the user and then activate the user's account.

api_keys (*trans*, **args*, ***kwargs*)

create (*trans*, *cntrller='user'*, *redirect_url=''*, *refresh_frames=[]*, ***kwd*)

dbkeys (*trans*, **args*, ***kwargs*)
Handle custom builds.

delete_address (*trans*, **args*, ***kwargs*)

edit_address (*trans*, **args*, ***kwargs*)

edit_info (*trans*, *cntrller*, ***kwd*)
Edit user information = username, email or password.

edit_toolbox_filters (*trans*, **args*, ***kwargs*)

edit_username (*trans*, **args*, ***kwargs*)

get_activation_token (*trans*, *email*)
Check for the activation token. Create new activation token and store it in the database if no token found.

get_most_recently_used_tool_async (*trans*, **args*, ***kwargs*)
Returns information about the most recently used tool.

index (*trans*, *cntrller*, ***kwd*)

installed_len_files = None

is_outside_grace_period (*trans*, *create_time*)
Function checks whether the user is outside the config-defined grace period for inactive accounts.

log_user_action_async (*trans*, *action*, *context*, *params*)
Log a user action asynchronously. If user is not logged in, do nothing.

login (*trans*, *refresh_frames=[]*, ***kwd*)
Handle Galaxy Log in

logout (*trans*, *logout_all=False*)

manage_addresses (*trans*, ***kwd*)

manage_user_info (*trans*, *cntrller*, ***kwd*)
Manage a user's login, password, public username, type, addresses, etc.

new_address (*trans*, **args*, ***kwargs*)

openid_associate (*trans*, *cntrller='user'*, ***kwd*)
Associates a user with an OpenID log in

openid_auth (*trans*, ***kwd*)
Handles user request to access an OpenID provider

openid_disassociate (*trans*, **args*, ***kwargs*)
Disassociates a user with an OpenID

openid_manage (*trans*, **args*, ***kwargs*)
Manage OpenIDs for user

openid_process (*trans*, ***kwd*)
Handle's response from OpenID Providers

prepare_activation_link (*trans*, *email*)
Prepare the account activation link for the user.

proceed_login (*trans, user, redirect*)

Function processes user login. It is called in case all the login requirements are valid.

resend_verification (*trans*)

Exposed function for use outside of the class. E.g. when user click on the resend link in the masthead.

resend_verification_email (*trans, email, username*)

Function resends the verification email in case user wants to log in with an inactive account or he clicks the resend link.

reset_password (*trans, email=None, **kwd*)

Reset the user's password. Send an email with the new password.

send_verification_email (*trans, email, username*)

Send the verification email containing the activation link to the user's email.

set_default_permissions (*trans, cntrller, **kwd*)

Set the user's default permissions for the new histories

set_user_pref_async (*trans, pref_name, pref_value*)

Set a user preference asynchronously. If user is not logged in, do nothing.

toolbox_filters (*trans, *args, **kwargs*)

Sets the user's default filters for the toolbox. Toolbox filters are specified in galaxy.ini. The user can activate them and the choice is stored in user_preferences.

undeleete_address (*trans, *args, **kwargs*)

user_openid_grid = <galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid object>

class galaxy.webapps.galaxy.controllers.user.**UserOpenIDGrid**

Bases: *galaxy.web.framework.helpers.grid.Grid*

build_initial_query (*trans, **kwd*)

columns = [<galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc073f03650>, <galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc073f03650>]

default_filter = {'openid': 'All'}

default_sort_key = '-create_time'

model_class

alias of UserOpenID

operations = [<galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc073f03090>]

template = '/user/openid_manage.mako'

title = 'OpenIDs linked to your account'

use_panels = False

visualization Module

```
class galaxy.webapps.galaxy.controllers.visualization.DbKeyColumn (label,
                                                                    key=None,
                                                                    model_class=None,
                                                                    method=None,
                                                                    format=None,
                                                                    link=None, attach_popup=False,
                                                                    visible=True,
                                                                    nowrap=False,
                                                                    filterable=None,
                                                                    sortable=True,
                                                                    label_id_prefix=None,
                                                                    include_bound=False)
```

Bases: *galaxy.web.framework.helpers.grid.GridColumn*

Column for filtering by and displaying dataset dbkey.

filter (*trans, user, query, dbkey*)
Filter by dbkey.

```
class galaxy.webapps.galaxy.controllers.visualization.DbKeyPlaceholderColumn (label,
                                                                                key=None,
                                                                                model_class=None,
                                                                                method=None,
                                                                                format=None,
                                                                                link=None,
                                                                                attach_popup=False,
                                                                                visible=True,
                                                                                nowrap=False,
                                                                                filterable=None,
                                                                                sortable=True,
                                                                                label_id_prefix=None,
                                                                                include_bound=False)
```

Bases: *galaxy.web.framework.helpers.grid.GridColumn*

Placeholder to keep track of dbkey.

filter (*trans, user, query, dbkey*)

```
class galaxy.webapps.galaxy.controllers.visualization.HistoryColumn (label,
                                                                    key=None,
                                                                    model_class=None,
                                                                    method=None,
                                                                    for-
                                                                    mat=None,
                                                                    link=None,
                                                                    at-
                                                                    tach_popup=False,
                                                                    visi-
                                                                    ble=True,
                                                                    nowrap=False,
                                                                    filter-
                                                                    able=None,
                                                                    sortable=True,
                                                                    la-
                                                                    bel_id_prefix=None,
                                                                    in-
                                                                    bound=False)
```

Bases: `galaxy.web.framework.helpers.grid.GridColumn`

Column for filtering by history id.

filter (*trans, user, query, history_id*)

```
class galaxy.webapps.galaxy.controllers.visualization.HistoryDatasetsSelectionGrid
Bases: galaxy.web.framework.helpers.grid.Grid
```

apply_query_filter (*trans, query, **kwargs*)

available_tracks = None

build_initial_query (*trans, **kwargs*)

columns = [`<galaxy.web.framework.helpers.grid.GridColumn object at 0x7fc062f9e290>`, `<galaxy.web.framework.help`

default_filter = {'deleted': 'False', 'shared': 'All'}

default_sort_key = '-hid'

get_current_item (*trans, **kwargs*)

Current item for grid is the history being queried. This is a bit of hack since `current_item` typically means the current item in the grid.

model_class

alias of `HistoryDatasetAssociation`

template = 'tracks/history_datasets_select_grid.mako'

title = 'Add Datasets'

use_async = True

use_paging = False

```
class galaxy.webapps.galaxy.controllers.visualization.HistorySelectionGrid
Bases: galaxy.web.framework.helpers.grid.Grid
```

Grid enables user to select a history, which is then used to display datasets from the history.

apply_query_filter (*trans, query, **kwargs*)

columns = [`<galaxy.webapps.galaxy.controllers.visualization.NameColumn object at 0x7fc062ea2ed0>`, `<galaxy.web.fran`

```
datasets_action = 'list_history_datasets'
datasets_param = 'f-history'
default_sort_key = '-update_time'
model_class
    alias of History
num_rows_per_page = 10
template = '/tracks/history_select_grid.mako'
title = 'Add Track: Select History'
use_async = True
use_paging = True
```

```
class galaxy.webapps.galaxy.controllers.visualization.LibrarySelectionGrid
    Bases: galaxy.webapps.galaxy.controllers.library.LibraryListGrid
```

Grid enables user to select a Library, which is then used to display datasets from the history.

```
columns = [<galaxy.webapps.galaxy.controllers.visualization.NameColumn object at 0x7fc06472f190>]
datasets_action = 'list_library_datasets'
datasets_param = 'f-library'
model_class
    alias of Library
num_rows_per_page = 10
template = '/tracks/history_select_grid.mako'
title = 'Add Track: Select Library'
use_async = True
use_paging = True
```

```
class galaxy.webapps.galaxy.controllers.visualization.NameColumn (label, key=None,
                                                                model_class=None,
                                                                method=None,
                                                                format=None,
                                                                link=None, attach_popup=False,
                                                                visible=True,
                                                                nowrap=False,
                                                                filterable=None,
                                                                sortable=True,
                                                                label_id_prefix=None,
                                                                inbound=False)
```

Bases: *galaxy.web.framework.helpers.grid.TextColumn*

```
get_link (trans, grid, history)
get_value (trans, grid, history)
```

```
class galaxy.webapps.galaxy.controllers.visualization.OpenObject
    Bases: dict
```

A dict that allows assignment and attribute retrieval using the dot operator.

If an attribute isn't contained in the dict *None* is returned (no *KeyError*). JSON-serializable.

```
class galaxy.webapps.galaxy.controllers.visualization.TracksterSelectionGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    apply_query_filter (trans, query, **kwargs)

    build_initial_query (trans, **kwargs)

    columns = [<galaxy.web.framework.helpers.grid.TextColumn object at 0x7fc062ec8bd0>, <galaxy.web.framework.helpers.grid.PublicURLColumn object at 0x7fc062e62110>]

    default_sort_key = '-update_time'

    model_class
        alias of Visualization

    show_item_checkboxes = True

    template = '/tracks/add_to_viz.mako'

    title = 'Insert into visualization'

    use_async = True

    use_paging = False

class galaxy.webapps.galaxy.controllers.visualization.VisualizationAllPublishedGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    apply_query_filter (trans, query, **kwargs)

    build_initial_query (trans, **kwargs)

    columns = [<galaxy.web.framework.helpers.grid.PublicURLColumn object at 0x7fc062e62110>, <galaxy.web.framework.helpers.grid.PublicURLColumn object at 0x7fc062e62110>]

    default_filter = {'username': 'All', 'title': 'All'}

    default_sort_key = 'update_time'

    model_class
        alias of Visualization

    title = 'Published Visualizations'

    use_async = True

    use_panels = True

class galaxy.webapps.galaxy.controllers.visualization.VisualizationController (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.SharableMixin, galaxy.model.item_attrs.UsesAnnotations, galaxy.web.base.controller.UsesVisualizationMixin, galaxy.model.item_attrs.UsesItemRatings

    bookmarks_from_dataset (trans, *args, **kwargs)

    circster (trans, id=None, hda_ldda=None, dataset_id=None, dbkey=None)
        Display a circster visualization.

    copy (trans, *args, **kwargs)

    create (trans, *args, **kwargs)
        Creates a new visualization or returns a form for creating visualization.

    display_by_username_and_slug (trans, username, slug)
        Display visualization based on a username and slug.

    edit (trans, *args, **kwargs)
        Edit a visualization's attributes.
```

get_item (*trans*, *id*)

get_item_content_async (*trans*, *id*)
Returns item content in HTML format.

get_name_and_link_async (*trans*, **args*, ***kwargs*)
Returns visualization's name and link.

imp (*trans*, **args*, ***kwargs*)
Import a visualization into user's workspace.

index (*trans*, **args*, ***kwargs*)
Lists user's saved visualizations.

list (*trans*, **args*, ***kwargs*)

list_current_history_datasets (*trans*, **args*, ***kwargs*)
List a history's datasets that can be added to a visualization.

list_datasets (*trans*, **args*, ***kwargs*)
List all datasets that can be added as tracks

list_histories (*trans*, **args*, ***kwargs*)
List all histories that can be used for selecting datasets.

list_history_datasets (*trans*, **args*, ***kwargs*)
List a history's datasets that can be added to a visualization.

list_libraries (*trans*, **args*, ***kwargs*)
List all libraries that can be used for selecting datasets.

list_library_datasets (*trans*, **args*, ***kwargs*)
List a library's datasets that can be added to a visualization.

list_published (*trans*, **args*, ***kwargs*)

list_tracks (*trans*, ***kwargs*)

phyloviz (*trans*, *id=None*, *dataset_id=None*, *tree_index=0*, ***kwargs*)

rate_async (*trans*, **args*, ***kwargs*)
Rate a visualization asynchronously and return updated community data.

render (*trans*, **args*, ***kwargs*)
Render the appropriate visualization template, parsing the *kwargs* into appropriate variables and resources (such as ORM models) based on this visualizations *param* data in *visualizations_conf.xml*.

URL: /visualization/show/{visualization_name}

save (*trans*, **args*, ***kwargs*)
Save a visualization; if visualization does not have an ID, a new visualization is created. Returns JSON of visualization.

saved (*trans*, **args*, ***kwargs*)

set_accessible_async (*trans*, **args*, ***kwargs*)
Set visualization's importable attribute and slug.

share (*trans*, **args*, ***kwargs*)
Handle sharing a visualization with a particular user.

sharing (*trans*, **args*, ***kwargs*)
Handle visualization sharing.

sweepster (*trans*, *id=None*, *hda_ldda=None*, *dataset_id=None*, *regions=None*)
 Displays a sweepster visualization using the incoming parameters. If *id* is available, get the visualization with the given *id*; otherwise, create a new visualization using a given *dataset* and *regions*.

trackster (*trans*, **args*, ***kwargs*)
 Display browser for the visualization denoted by *id* and add the datasets listed in *dataset_ids*.

class `galaxy.webapps.galaxy.controllers.visualization.VisualizationListGrid`
 Bases: `galaxy.web.framework.helpers.grid.Grid`

apply_query_filter (*trans*, *query*, ***kwargs*)

columns = [`<galaxy.web.framework.helpers.grid.GridColumn object at 0x7fc062ec8cd0>`, `<galaxy.web.framework.helpers.grid.GridColumn object at 0x7fc062ec8cd0>`]

default_filter = {'deleted': 'False', 'sharing': 'All', 'tags': 'All', 'title': 'All'}

default_sort_key = 'update_time'

get_display_name (*trans*, *item*)

get_url_args (*item*)

Returns dictionary used to create item link.

global_actions = [`<galaxy.web.framework.helpers.grid.GridAction object at 0x7fc062ec8f10>`]

model_class

alias of `Visualization`

operations = [`<galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc062ec8f50>`, `<galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc062ec8f50>`]

title = 'Saved Visualizations'

workflow Module

class `galaxy.webapps.galaxy.controllers.workflow.SingleTagContentsParser` (*target_tag*)
 Bases: `sgmllib.SGMLParser`

handle_data (*text*)

Called for each block of plain text.

unknown_starttag (*tag*, *attrs*)

Called for each start tag.

class `galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAllPublishedGrid`
 Bases: `galaxy.web.framework.helpers.grid.Grid`

apply_query_filter (*trans*, *query*, ***kwargs*)

build_initial_query (*trans*, ***kwargs*)

columns = [`<galaxy.web.framework.helpers.grid.PublicURLColumn object at 0x7fc067847cd0>`, `<galaxy.web.framework.helpers.grid.PublicURLColumn object at 0x7fc067847cd0>`]

default_filter = {'username': 'All', 'public_url': 'All', 'tags': 'All'}

default_sort_key = 'update_time'

model_class

alias of `StoredWorkflow`

operations = [`<galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc067847ed0>`, `<galaxy.web.framework.helpers.grid.GridOperation object at 0x7fc067847ed0>`]

title = 'Published Workflows'

use_async = True

class `galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid`
 Bases: `galaxy.web.framework.helpers.grid.Grid`

```
class StepsColumn(label, key=None, model_class=None, method=None, format=None,
                  link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                  sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, workflow)

StoredWorkflowListGrid.apply_query_filter(trans, query, **kwargs)

StoredWorkflowListGrid.columns = [<galaxy.web.framework.helpers.grids.TextColumn object at 0x7fc067847990>]

StoredWorkflowListGrid.default_filter = {'name': 'All', 'tags': 'All'}

StoredWorkflowListGrid.default_sort_key = 'update_time'

StoredWorkflowListGrid.model_class
    alias of StoredWorkflow

StoredWorkflowListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7fc067847990>]

StoredWorkflowListGrid.title = 'Saved Workflows'

StoredWorkflowListGrid.use_panels = True

class galaxy.webapps.galaxy.controllers.workflow.WorkflowController(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.SharableMixin, galaxy.web.base.controller.UsesStoredWorkflowMixin, galaxy.model.item_attrs.UsesItemRatings

    annotate_async(trans, *args, **kwargs)

    build_from_current_history(trans, job_ids=None, dataset_ids=None,
                              dataset_collection_ids=None, workflow_name=None)

    configure_menu(trans, workflow_ids=None)

    copy(trans, *args, **kwargs)

    create(trans, *args, **kwargs)
        Create a new stored workflow with name workflow_name.

    delete(trans, id=None)
        Mark a workflow as deleted

    display_by_id(trans, id)
        Display workflow based on id.

    display_by_username_and_slug(trans, username, slug, format='html')
        Display workflow based on a username and slug. Format can be html, json, or json-download.

    edit_attributes(trans, *args, **kwargs)

    editor(trans, *args, **kwargs)
        Render the main workflow editor interface. The canvas is embedded as an iframe (necessary for scrolling
        to work properly), which is rendered by editor_canvas.

    editor_form_post(trans, *args, **kwargs)
        Accepts a tool state and incoming values, and generates a new tool form and some additional information,
        packed into a json dictionary. This is used for the form shown in the right pane when a node is selected.

    export(trans, *args, **kwargs)
        Handles download/export workflow command.

    export_to_file(trans, *args, **kwargs)
        Get the latest Workflow for the StoredWorkflow identified by id and encode it as a json string that can be
        imported back into Galaxy
```

This has slightly different information than the above. In particular, it does not attempt to decode forms and build UIs, it just stores the raw state.

export_to_myexp (*trans*, **args*, ***kwargs*)

Exports a workflow to myExperiment website.

for_direct_import (*trans*, **args*, ***kwargs*)

Get the latest Workflow for the StoredWorkflow identified by *id* and encode it as a json string that can be imported back into Galaxy

This has slightly different information than the above. In particular, it does not attempt to decode forms and build UIs, it just stores the raw state.

gen_image (*trans*, **args*, ***kwargs*)

get_embed_html_async (*trans*, *id*)

Returns HTML for embedding a workflow in a page.

get_item (*trans*, *id*)

get_item_content_async (*trans*, *id*)

Returns item content in HTML format.

get_name_and_link_async (*trans*, **args*, ***kwargs*)

Returns workflow's name and link.

get_new_module_info (*trans*, **args*, ***kwargs*)

Get the info for a new instance of a module initialized with default parameters (any keyword arguments will be passed along to the module). Result includes data inputs and outputs, html representation of the initial form, and the initial tool state (with default values). This is called asynchronously whenever a new node is added.

imp (*trans*, **args*, ***kwargs*)

Imports a workflow shared by other users.

import_workflow (*trans*, *cntrlr*='workflow', ***kwd*)

Import a workflow by reading an url, uploading a file, opening and reading the contents of a local file, or receiving the textual representation of a workflow via http.

index (*trans*)

list (*trans*, **args*, ***kwargs*)

Render workflow main page (management of existing workflows)

list_for_run (*trans*, **args*, ***kwargs*)

Render workflow list for analysis view (just allows running workflow or switching to management view)

list_grid (*trans*, **args*, ***kwargs*)

List user's stored workflows.

list_published (*trans*, ***kwargs*)

load_workflow (*trans*, **args*, ***kwargs*)

Get the latest Workflow for the StoredWorkflow identified by *id* and encode it as a json string that can be read by the workflow editor web interface.

published_list_grid = <galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAllPublishedGrid object>

rate_async (*trans*, **args*, ***kwargs*)

Rate a workflow asynchronously and return updated community data.

rename (*trans*, **args*, ***kwargs*)

rename_async (*trans*, **args*, ***kwargs*)

```
run (trans, id, history_id=None, hide_fixed_params=False, **kwargs)  
save_workflow (trans, *args, **kwargs)  
    Save the workflow described by workflow_data with id id.  
set_accessible_async (trans, *args, **kwargs)  
    Set workflow's importable attribute and slug.  
share (trans, *args, **kwargs)  
sharing (trans, *args, **kwargs)  
    Handle workflow sharing.  
stored_list_grid = <galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid object>  
tag_outputs (trans, id, **kwargs)
```

reports Package

reports Package The Galaxy Reports application.

app Module

```
class galaxy.webapps.reports.app.UniverseApplication (**kwargs)  
    Bases: object  
  
    Encapsulates the state of a Universe application  
  
    shutdown ()
```

buildapp Module Provides factory methods to assemble the Galaxy web application

```
class galaxy.webapps.reports.buildapp.ReportsWebApplication (galaxy_app, session_cookie='galaxy_session',  
                                                             name=None)  
  
    Bases: galaxy.web.framework.webapp.WebApplication  
  
galaxy.webapps.reports.buildapp.add_ui_controllers (webapp, app)  
    Search for controllers in the 'galaxy.webapps.controllers' module and add them to the webapp.  
  
galaxy.webapps.reports.buildapp.app_factory (global_conf, **kwargs)  
    Return a wsgi application serving the root object  
  
galaxy.webapps.reports.buildapp.build_template_error_formatters ()  
    Build a list of template error formatters for WebError. When an error occurs, WebError pass the exception to  
    each function in this list until one returns a value, which will be displayed on the error page.  
  
galaxy.webapps.reports.buildapp.wrap_in_middleware (app, global_conf, **local_conf)  
    Based on the configuration wrap app in a set of common and useful middleware.  
  
galaxy.webapps.reports.buildapp.wrap_in_static (app, global_conf, **local_conf)
```

config Module Universe configuration builder.

```
class galaxy.webapps.reports.config.Configuration (**kwargs)  
    Bases: object  
  
    check ()  
  
    get (key, default)
```

exception `galaxy.webapps.reports.config.ConfigurationError`

Bases: `exceptions.Exception`

`galaxy.webapps.reports.config.configure_logging` (*config*)

Allow some basic logging configuration to be read from the cherrpy config.

`galaxy.webapps.reports.config.get_database_engine_options` (*kwargs*)

Allow options for the SQLAlchemy database engine to be passed by using the prefix “database_engine_option”.

`galaxy.webapps.reports.config.resolve_path` (*path*, *root*)

If ‘path’ is relative make absolute by prepending ‘root’

Subpackages

controllers Package

controllers Package Galaxy reports controllers.

jobs Module

class `galaxy.webapps.reports.controllers.jobs.Jobs` (*app*)

Bases: `galaxy.web.base.controller.BaseUIController`

Class contains functions for querying data requested by user via the webapp. It exposes the functions and responds to requests with the filled .mako templates.

`job_info` (*trans*, ***kwd*)

`per_month_all` (*trans*, ***kwd*)

Queries the DB for all jobs. Avoids monitor jobs.

`per_month_in_error` (*trans*, ***kwd*)

Queries the DB for user jobs in error. Filters out monitor jobs.

`per_tool` (*trans*, ***kwd*)

`per_user` (*trans*, ***kwd*)

`specified_date_handler` (*trans*, ***kwd*)

`specified_date_list_grid` = <galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid object>

`specified_month_all` (*trans*, ***kwd*)

Queries the DB for all jobs in given month, defaults to current month.

`specified_month_in_error` (*trans*, ***kwd*)

Queries the DB for the user jobs in error.

`tool_per_month` (*trans*, ***kwd*)

`user_per_month` (*trans*, ***kwd*)

class `galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid`

Bases: `galaxy.web.framework.helpers.grid.Grid`

class `CreateTimeColumn` (*label*, *key=None*, *model_class=None*, *method=None*, *format=None*,
link=None, *attach_popup=False*, *visible=True*, *nowrap=False*, *filter-able=None*,
sortable=True, *label_id_prefix=None*, *inbound=False*)

Bases: `galaxy.web.framework.helpers.grid.DateTimeColumn`

`get_value` (*trans*, *grid*, *job*)

```
class SpecifiedDateListGrid.EmailColumn(label, key=None, model_class=None,
                                         method=None, format=None, link=None,
                                         attach_popup=False, visible=True,
                                         nowrap=False, filterable=None, sortable=True,
                                         label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

filter (*trans, user, query, column_filter*)

```
class SpecifiedDateListGrid.JobIdColumn(label, key=None, model_class=None,
                                         method=None, format=None, link=None,
                                         attach_popup=False, visible=True,
                                         nowrap=False, filterable=None, sortable=True,
                                         label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.IntegerColumn`

get_value (*trans, grid, job*)

```
class SpecifiedDateListGrid.SpecifiedDateColumn(label, key=None,
                                                  model_class=None, method=None,
                                                  format=None, link=None,
                                                  attach_popup=False, visible=True,
                                                  nowrap=False, filterable=None,
                                                  sortable=True,
                                                  label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

filter (*trans, user, query, column_filter*)

```
class SpecifiedDateListGrid.StateColumn(label, key=None, model_class=None,
                                         method=None, format=None, link=None,
                                         attach_popup=False, visible=True,
                                         nowrap=False, filterable=None, sortable=True,
                                         label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

filter (*trans, user, query, column_filter*)

get_value (*trans, grid, job*)

```
class SpecifiedDateListGrid.ToolColumn(label, key=None, model_class=None,
                                         method=None, format=None, link=None,
                                         attach_popup=False, visible=True, nowrap=False,
                                         filterable=None, sortable=True, label_id_prefix=None,
                                         inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

get_value (*trans, grid, job*)

```
class SpecifiedDateListGrid.UserColumn(label, key=None, model_class=None,
                                         method=None, format=None, link=None,
                                         attach_popup=False, visible=True, nowrap=False,
                                         filterable=None, sortable=True, label_id_prefix=None,
                                         inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

get_value (*trans, grid, job*)

`SpecifiedDateListGrid.build_initial_query` (*trans, **kwd*)

`SpecifiedDateListGrid.columns` = [`<galaxy.webapps.reports.controllers.jobs.JobIdColumn object at 0x7fc062261`


```

SpecifiedDateListGrid.default_filter = {'specified_date': 'All'}
SpecifiedDateListGrid.default_sort_key = 'id'
SpecifiedDateListGrid.model_class
    alias of Job
SpecifiedDateListGrid.num_rows_per_page = 50
SpecifiedDateListGrid.preserve_state = False
SpecifiedDateListGrid.standard_filters = []
SpecifiedDateListGrid.template = '/webapps/reports/grid.mako'
SpecifiedDateListGrid.title = 'Jobs'
SpecifiedDateListGrid.use_async = False
SpecifiedDateListGrid.use_paging = True

galaxy.webapps.reports.controllers.jobs.get_job(trans, id)
galaxy.webapps.reports.controllers.jobs.get_monitor_id(trans, monitor_email)
    A convenience method to obtain the monitor job id.

```

root Module

```

class galaxy.webapps.reports.controllers.root.Report(app)
    Bases: galaxy.web.base.controller.BaseUIController

    index(trans, **kwd)

```

sample_tracking Module

```

class galaxy.webapps.reports.controllers.sample_tracking.SampleTracking(app)
    Bases: galaxy.web.base.controller.BaseUIController

    per_month_all(trans, **kwd)
    per_user(trans, **kwd)
    specified_date_handler(trans, **kwd)
    specified_date_list_grid = <galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid object>
    user_per_month(trans, **kwd)

class galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class CreateTimeColumn(label, key=None, model_class=None, method=None, format=None,
                           link=None, attach_popup=False, visible=True, nowrap=False, filter-
                           able=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.DateTimeColumn

        get_value(trans, grid, request)

    class SpecifiedDateListGrid.EmailColumn(label, key=None, model_class=None,
                                             method=None, format=None, link=None,
                                             attach_popup=False, visible=True,
                                             nowrap=False, filterable=None, sortable=True,
                                             label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        filter(trans, user, query, column_filter)

```

```
class SpecifiedDateListGrid.RequestNameColumn (label, key=None, model_class=None,
                                                method=None, format=None,
                                                link=None, attach_popup=False,
                                                visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*

get_value (*trans, grid, request*)

```
class SpecifiedDateListGrid.SpecifiedDateColumn (label, key=None,
                                                  model_class=None, method=None,
                                                  format=None, link=None,
                                                  attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: *galaxy.web.framework.helpers.grids.GridColumn*

filter (*trans, user, query, column_filter*)

```
class SpecifiedDateListGrid.UserColumn (label, key=None, model_class=None,
                                         method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*

get_value (*trans, grid, request*)

SpecifiedDateListGrid.build_initial_query (*trans, **kwd*)

SpecifiedDateListGrid.columns = [*galaxy.webapps.reports.controllers.sample_tracking.RequestNameColumn*]

SpecifiedDateListGrid.default_filter = {'specified_date': 'All'}

SpecifiedDateListGrid.default_sort_key = 'name'

SpecifiedDateListGrid.model_class
alias of Request

SpecifiedDateListGrid.num_rows_per_page = 50

SpecifiedDateListGrid.preserve_state = False

SpecifiedDateListGrid.standard_filters = []

SpecifiedDateListGrid.template = '/webapps/reports/grid.mako'

SpecifiedDateListGrid.title = 'Sequencing Requests'

SpecifiedDateListGrid.use_async = False

SpecifiedDateListGrid.use_paging = True

galaxy.webapps.reports.controllers.sample_tracking.get_request (*trans, id*)

system Module

```
class galaxy.webapps.reports.controllers.system.System (app)
```

Bases: *galaxy.web.base.controller.BaseUIController*

dataset_info (*trans, **kwd*)

deleted_datasets (*trans*, ***kwd*)

The number of datasets that were deleted more than the specified number of days ago, but have not yet been purged.

deleted_histories (*trans*, ***kwd*)

The number of histories that were deleted more than the specified number of days ago, but have not yet been purged. Also included is the number of datasets associated with the histories.

disk_usage (*trans*, ***kwd*)

get_disk_usage (*file_path*)

index (*trans*, ***kwd*)

userless_histories (*trans*, ***kwd*)

The number of userless histories and associated datasets that have not been updated for the specified number of days.

`galaxy.webapps.reports.controllers.system.nice_size` (*size*, *include_bytes=False*)

Returns a readably formatted string with the size

users Module

class `galaxy.webapps.reports.controllers.users.Users` (*app*)

Bases: `galaxy.web.base.controller.BaseUIController`

last_access_date (*trans*, ***kwd*)

registered_users (*trans*, ***kwd*)

registered_users_per_month (*trans*, ***kwd*)

specified_date (*trans*, ***kwd*)

specified_month (*trans*, ***kwd*)

user_disk_usage (*trans*, ***kwd*)

workflows Module

class `galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid`

Bases: `galaxy.web.framework.helpers.grid.Grid`

class `CreateTimeColumn` (*label*, *key=None*, *model_class=None*, *method=None*, *format=None*,
link=None, *attach_popup=False*, *visible=True*, *nowrap=False*, *filterable=None*, *sortable=True*, *label_id_prefix=None*, *inbound=False*)

Bases: `galaxy.web.framework.helpers.grid.DateTimeColumn`

get_value (*trans*, *grid*, *stored_workflow*)

class `SpecifiedDateListGrid.EmailColumn` (*label*, *key=None*, *model_class=None*,
method=None, *format=None*, *link=None*,
attach_popup=False, *visible=True*,
nowrap=False, *filterable=None*, *sortable=True*,
label_id_prefix=None, *inbound=False*)

Bases: `galaxy.web.framework.helpers.grid.GridColumn`

filter (*trans*, *user*, *query*, *column_filter*)

```

class SpecifiedDateListGrid.SpecifiedDateColumn (label, key=None,
                                                    model_class=None, method=None,
                                                    format=None, link=None,
                                                    attach_popup=False, visible=True,
                                                    nowrap=False, filterable=None,
                                                    sortable=True, label_id_prefix=None,
                                                    inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn
    filter (trans, user, query, column_filter)

class SpecifiedDateListGrid.UserColumn (label, key=None, model_class=None,
                                           method=None, format=None, link=None,
                                           attach_popup=False, visible=True, nowrap=False,
                                           filterable=None, sortable=True, label_id_prefix=None,
                                           inbound=False)
    Bases: galaxy.web.framework.helpers.grids.TextColumn
    get_value (trans, grid, stored_workflow)

class SpecifiedDateListGrid.WorkflowNameColumn (label, key=None, model_class=None,
                                                  method=None, format=None,
                                                  link=None, attach_popup=False,
                                                  visible=True, nowrap=False,
                                                  filterable=None, sortable=True,
                                                  label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.TextColumn
    get_value (trans, grid, stored_workflow)

SpecifiedDateListGrid.build_initial_query (trans, **kwd)

SpecifiedDateListGrid.columns = [<galaxy.webapps.reports.controllers.workflows.WorkflowNameColumn object>]

SpecifiedDateListGrid.default_filter = {'specified_date': 'All'}

SpecifiedDateListGrid.default_sort_key = 'name'

SpecifiedDateListGrid.model_class
    alias of StoredWorkflow

SpecifiedDateListGrid.num_rows_per_page = 50

SpecifiedDateListGrid.preserve_state = False

SpecifiedDateListGrid.standard_filters = []

SpecifiedDateListGrid.template = '/webapps/reports/grid.mako'

SpecifiedDateListGrid.title = 'Workflows'

SpecifiedDateListGrid.use_async = False

SpecifiedDateListGrid.use_paging = True

class galaxy.webapps.reports.controllers.workflows.Workflows (app)
    Bases: galaxy.web.base.controller.BaseUIController
    per_month_all (trans, **kwd)
    per_user (trans, **kwd)
    specified_date_handler (trans, **kwd)

```

```

    specified_date_list_grid = <galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid object>
    user_per_month (trans, **kwd)
galaxy.webapps.reports.controllers.workflows.get_workflow(trans, id)

```

workflow Package

modules Module Modules used in building workflows

```

exception galaxy.workflow.modules.CancelWorkflowEvaluation
    Bases: exceptions.Exception

```

```

exception galaxy.workflow.modules.DelayedWorkflowEvaluation
    Bases: exceptions.Exception

```

```

class galaxy.workflow.modules.InputDataCollectionModule(trans)
    Bases: galaxy.workflow.modules.InputModule
    collection_type = 'list'
    default_collection_type = 'list'
    default_name = 'Input Dataset Collection'
    classmethod default_state(Class)
    get_data_outputs()
    get_runtime_inputs(filter_set=['data'])
    name = 'Input dataset collection'
    state_fields = ['name', 'collection_type']
    type = 'data_collection_input'

```

```

class galaxy.workflow.modules.InputDataModule(trans)
    Bases: galaxy.workflow.modules.InputModule
    default_name = 'Input Dataset'
    classmethod default_state(Class)
    get_data_outputs()
    get_runtime_inputs(filter_set=['data'])
    name = 'Input dataset'
    state_fields = ['name']
    type = 'data_input'

```

```

class galaxy.workflow.modules.InputModule(trans)
    Bases: galaxy.workflow.modules.SimpleWorkflowModule
    execute(trans, progress, invocation, step)
    get_data_inputs()
    get_runtime_input_dicts(step_annotation)
    get_runtime_state()
    recover_mapping(step, step_invocations, progress)

```

exception `galaxy.workflow.modules.MissingToolException`

Bases: `exceptions.Exception`

WorkflowModuleInjector will raise this if the tool corresponding to the module is missing.

class `galaxy.workflow.modules.PauseModule` (*trans*)

Bases: `galaxy.workflow.modules.SimpleWorkflowModule`

Initially this module will unconditionally pause a workflow - will aim to allow conditional pausing later on.

default_name = 'Pause for Dataset Review'

classmethod **default_state** (*Class*)

do_invocation_step_action (*step, action*)

Update or set the workflow invocation state action - generic extension point meant to allow users to interact with interactive workflow modules. The action object returned from this method will be attached to the WorkflowInvocationStep and be available the next time the workflow scheduler visits the workflow.

execute (*trans, progress, invocation, step*)

get_data_inputs ()

get_data_outputs ()

get_runtime_input_dicts (*step_annotation*)

get_runtime_inputs (***kwargs*)

get_runtime_state ()

name = 'Pause for dataset review'

recover_mapping (*step, step_invocations, progress*)

state_fields = ['name']

type = 'pause'

class `galaxy.workflow.modules.SimpleWorkflowModule` (*trans*)

Bases: `galaxy.workflow.modules.WorkflowModule`

compute_runtime_state (*trans, step_updates=None, source='html'*)

decode_runtime_state (*trans, string*)

classmethod **default_state** (*Class*)

This method should return a dictionary describing each configuration property and its default value.

encode_runtime_state (*trans, state*)

classmethod **from_dict** (*Class, trans, d, secure=True*)

classmethod **from_workflow_step** (*Class, trans, step*)

get_config_form ()

get_state (*secure=True*)

classmethod **new** (*Class, trans, tool_id=None*)

normalize_runtime_state (*runtime_state*)

recover_runtime_state (*runtime_state*)

Take secure runtime state from persisted invocation and convert it into a DefaultToolState object for use during workflow invocation.

```

recover_state (state, **kwds)
    Recover state dict from simple dictionary describing configuration state (potentially from persisted step
    state).

    Sub-classes should supply default_state method and state_fields attribute which are used to build up the
    state dict.

save_to_step (step)

update_runtime_state (trans, state, values)

update_state (incoming)

class galaxy.workflow.modules.ToolModule (trans, tool_id)
    Bases: galaxy.workflow.modules.WorkflowModule

    add_dummy_datasets (connections=None)

    check_and_update_state ()

    compute_runtime_state (trans, step_updates=None, source='html')

    encode_runtime_state (trans, state)

    execute (trans, progress, invocation, step)

    classmethod from_dict (Class, trans, d, secure=True)

    classmethod from_workflow_step (Class, trans, step)

    get_config_form ()

    get_data_inputs ()

    get_data_outputs ()

    get_errors ()

    get_name ()

    get_post_job_actions ()

    get_runtime_input_dicts (step_annotation)

    get_state (secure=True)

    get_tool_id ()

    get_tool_version ()

    get_tooltip (static_path='')

    classmethod new (Class, trans, tool_id=None)

    normalize_runtime_state (runtime_state)

    recover_mapping (step, step_invocations, progress)

    recover_runtime_state (runtime_state)
        Take secure runtime state from persisted invocation and convert it into a DefaultToolState object for use
        during workflow invocation.

    recover_state (state, **kwds)
        Recover module configuration state property (a DefaultToolState object) using the tool's
        params_from_strings method.

    save_to_step (step)

    type = 'tool'

```

```
update_state (incoming)
```

class `galaxy.workflow.modules.WorkflowModule` (*trans*)
Bases: `object`

```
add_dummy_datasets (connections=None)
```

```
check_and_update_state ()
```

If the state is not in sync with the current implementation of the module, try to update. Returns a list of messages to be displayed

```
compute_runtime_state (trans, step_updates=None, source='html')
```

Determine the runtime state (potentially different from `self.state` which describes configuration state). This (again unlike `self.state`) is currently always a *DefaultToolState* object.

If *step_updates* is *None*, this is likely for rendering the run form for instance and no runtime properties are available and state must be solely determined by the default runtime state described by the step.

If *step_updates* are available they describe the runtime properties supplied by the workflow runner (potentially including a *tool_state* parameter which is the serialized default encoding state created with `encode_runtime_state` above).

```
do_invocation_step_action (step, action)
```

Update or set the workflow invocation state action - generic extension point meant to allow users to interact with interactive workflow modules. The action object returned from this method will be attached to the `WorkflowInvocationStep` and be available the next time the workflow scheduler visits the workflow.

```
encode_runtime_state (trans, state)
```

Encode the default runtime state at return as a simple *str* for use in a hidden parameter on the workflow run submission form.

This default runtime state will be combined with user supplied parameters in *compute_runtime_state* below at workflow invocation time to actually describe how each step will be executed.

```
execute (trans, progress, invocation, step)
```

Execute the given workflow step in the given workflow invocation. Use the supplied workflow progress object to track outputs, find inputs, etc...

```
classmethod from_dict (Class, trans, d)
```

Create a new instance of the module initialized from values in the dictionary *d*.

```
classmethod from_workflow_step (Class, trans, step)
```

```
get_config_form ()
```

Render form that is embedded in workflow editor for modifying the step state of a node.

```
get_data_inputs ()
```

Get configure time data input descriptions.

```
get_data_outputs ()
```

```
get_errors ()
```

It seems like this is effectively just used as boolean - some places in the tool shed `self.errors` is set to boolean, other places 'unavailable', likewise in Galaxy it stores a list containing a string with an unrecognized tool id error message.

```
get_name ()
```

```
get_runtime_input_dicts (step_annotation)
```

Get runtime inputs (inputs and parameters) as simple dictionary.

```
get_runtime_inputs ()
```

Used internally by modules and when displaying inputs in workflow editor and run workflow templates.

Note: The ToolModule doesn't implement this and these templates contain specialized logic for dealing with the tool and state directly in the case of ToolModules.

get_state()

Return a serializable representation of the persistable state of the step - for tools it DefaultToolState.encode returns a string and for simpler module types a json description is dumped out.

get_tool_id()

get_tooltip(*static_path*='')

get_type()

classmethod new(*Class, trans, tool_id=None*)

Create a new instance of the module with default state

recover_mapping(*step, step_invocations, progress*)

Re-populate progress object with information about connections from previously executed steps recorded via step_invocations.

save_to_step(*step*)

update_state(*incoming*)

Update the current state of the module against the user supplied parameters in the dict-like object *incoming*.

class galaxy.workflow.modules.**WorkflowModuleFactory**(*module_types*)

Bases: object

from_dict(*trans, d, **kwargs*)

Return module initialized from the data in dictionary *d*.

from_workflow_step(*trans, step*)

Return module initialized from the WorkflowStep object *step*.

new(*trans, type, tool_id=None*)

Return module for type and (optional) tool_id initialized with new / default state.

class galaxy.workflow.modules.**WorkflowModuleInjector**(*trans*)

Bases: object

Injects workflow step objects from the ORM with appropriate module and module generated/influenced state.

inject(*step, step_args=None, source='html'*)

Pre-condition: *step* is an ORM object coming from the database, if supplied *step_args* is the representation of the inputs for that step supplied via web form.

Post-condition: The supplied *step* has new non-persistent attributes useful during workflow invocation. These include 'upgrade_messages', 'state', 'input_connections_by_name', and 'module'.

If *step_args* is provided from a web form this is applied to generate 'state' else it is just obtained from the database.

galaxy.workflow.modules.**is_tool_module_type**(*module_type*)

galaxy.workflow.modules.**load_module_sections**(*trans*)

Get abstract description of the workflow modules this Galaxy instance is configured with.

galaxy.workflow.modules.**populate_module_and_state**(*trans, workflow, param_map*)

Used by API but not web controller, walks through a workflow's steps and populates transient module and state attributes on each.

1.2.3 galaxy_utils Package

Subpackages

sequence Package

fasta Module

```
class galaxy_utils.sequence.fasta.fastaNamedReader (fh)
    Bases: object
```

```
    close()
```

```
    get (sequence_id)
```

```
    has_data()
```

```
class galaxy_utils.sequence.fasta.fastaReader (fh)
    Bases: object
```

```
    close()
```

```
    next()
```

```
class galaxy_utils.sequence.fasta.fastaSequence
    Bases: object
```

```
class galaxy_utils.sequence.fasta.fastaWriter (fh)
    Bases: object
```

```
    close()
```

```
    write (fastq_read)
```

fastq Module

```
class galaxy_utils.sequence.fastq.ReadlineCountFile (f)
    Bases: object
```

```
    readline (*args, **kws)
```

```
class galaxy_utils.sequence.fastq.fastqAggregator
    Bases: object
```

```
    VALID_FORMATS = ['solexa', 'sanger', 'cssanger', 'illumina']
```

```
    consume_read (fastq_read)
```

```
    get_ascii_range()
```

```
    get_base_counts_for_column (column)
```

```
    get_decimal_range()
```

```
    get_length_counts()
```

```
    get_max_read_length()
```

```
    get_read_count()
```

```
    get_read_count_for_column (column)
```

```
    get_score_at_position_for_column (column, position)
```

```
    get_score_list_for_column (column)
```

```
    get_score_max_for_column (column)
```

```

    get_score_min_for_column(column)
    get_score_sum_for_column(column)
    get_summary_statistics_for_column(i)
    get_valid_formats(check_list=None)
class galaxy_utils.sequence.fastq.fastqCSSangerRead
    Bases: galaxy_utils.sequence.fastq.fastqSequencingRead
    apply_galaxy_conventions()
    ascii_max = 126
    ascii_min = 33
    assert_sequence_quality_lengths()
    change_adapter(new_adapter, clone=True)
    complement(clone=True)
    format = 'cssanger'
    get_sequence()
    has_adapter_base()
    insufficient_quality_length()
    quality_max = 93
    quality_min = 0
    reverse(clone=True)
    score_system = 'phred'
    sequence_space = 'color'
    valid_sequence_list = ['0', '1', '2', '3', '4', '5', '6', '.']
class galaxy_utils.sequence.fastq.fastqCombiner(format)
    Bases: object
    combine(fasta_seq, quality_seq)
class galaxy_utils.sequence.fastq.fastqFakeFastaScoreReader(format='sanger', quality_encoding=None)
    Bases: object
    close()
    get(sequence)
    has_data()
class galaxy_utils.sequence.fastq.fastqIlluminaRead
    Bases: galaxy_utils.sequence.fastq.fastqSequencingRead
    ascii_max = 126
    ascii_min = 64
    format = 'illumina'
    quality_max = 62
    quality_min = 0

```

```
    score_system = 'phred'
    sequence_space = 'base'
class galaxy_utils.sequence.fastq.fastqJoiner (format, force_quality_encoding=None)
    Bases: object
    get_paired_identifier (fastq_read)
    is_first_mate (sequence_id)
    join (read1, read2)
class galaxy_utils.sequence.fastq.fastqNamedReader (fh, format='sanger', apply_galaxy_conventions=False)
    Bases: object
    close ()
    get (sequence_identifier)
    has_data ()
class galaxy_utils.sequence.fastq.fastqReader (fh, format='sanger', apply_galaxy_conventions=False)
    Bases: object
    close ()
    next ()
class galaxy_utils.sequence.fastq.fastqSangerRead
    Bases: galaxy_utils.sequence.fastq.fastqSequencingRead
    ascii_max = 126
    ascii_min = 33
    format = 'sanger'
    quality_max = 93
    quality_min = 0
    score_system = 'phred'
    sequence_space = 'base'
class galaxy_utils.sequence.fastq.fastqSequencingRead
    Bases: galaxy_utils.sequence.sequence.SequencingRead
    apply_galaxy_conventions ()
    ascii_max = 126
    ascii_min = 33
    assert_sequence_quality_lengths ()
    classmethod convert_base_to_color_space (sequence)
    classmethod convert_color_to_base_space (sequence)
    convert_read_to_format (format, force_quality_encoding=None)
    classmethod convert_score_phred_to_solexa (decimal_score_list)
    classmethod convert_score_solexa_to_phred (decimal_score_list)
    format = 'sanger'
```

```

    get_ascii_quality_scores()
    get_ascii_quality_scores_len()
        Compute ascii quality score length, without generating relatively expensive quality score array.
    classmethod get_class_by_format(format)
    get_decimal_quality_scores()
    get_sequence()
    insufficient_quality_length()
    is_ascii_encoded()
    is_valid_format()
    is_valid_sequence()
    quality_max = 93
    quality_min = 0
    classmethod restrict_scores_to_valid_range(decimal_score_list)
    reverse(clone=True)
    score_system = 'phred'
    sequence_space = 'base'
    slice(left_column_offset, right_column_offset)
    classmethod transform_scores_to_valid_range(decimal_score_list)
    classmethod transform_scores_to_valid_range_ascii(decimal_score_list)
class galaxy_utils.sequence.fastq.fastqSolexaRead
    Bases: galaxy_utils.sequence.fastq.fastqSequencingRead
    ascii_max = 126
    ascii_min = 59
    format = 'solexa'
    quality_max = 62
    quality_min = -5
    score_system = 'solexa'
    sequence_space = 'base'
class galaxy_utils.sequence.fastq.fastqSplitter
    Bases: object
    split(fastq_read)
class galaxy_utils.sequence.fastq.fastqVerboseErrorReader(fh, **kws)
    Bases: galaxy_utils.sequence.fastq.fastqReader
    MAX_PRINT_ERROR_BYTES = 1024
    next()
class galaxy_utils.sequence.fastq.fastqWriter(fh,
                                              force_quality_encoding=None,
                                              format=None,
    Bases: object

```

```
close()
```

```
write(fastq_read)
```

```
galaxy_utils.sequence.fastq.format  
alias of fastqCSSangerRead
```

sequence Module

```
class galaxy_utils.sequence.sequence.SequencingRead
```

```
Bases: object
```

```
append_quality(quality)
```

```
append_sequence(sequence)
```

```
clone()
```

```
color_space_converter = <galaxy_utils.sequence.transform.ColorSpaceConverter object>
```

```
complement(clone=True)
```

```
is_DNA()
```

```
reverse(clone=True)
```

```
reverse_complement(clone=True)
```

```
sequence_as_DNA(clone=True)
```

```
sequence_as_RNA(clone=True)
```

```
valid_sequence_list = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
```

transform Module

```
class galaxy_utils.sequence.transform.ColorSpaceConverter(fake_adapter_base='G')
```

```
Bases: object
```

```
base = 'N'
```

```
base_to_color_dict = {'A': {'A': '0', 'C': '1', 'T': '3', 'G': '2', 'N': '4'}, 'C': {'A': '1', 'C': '0', 'T': '2', 'G': '3', 'N':
```

```
color_dict = {'1': 'N', '0': 'N', '3': 'N', '2': 'N', '5': 'N', '4': 'N', '6': 'N', ':': 'N'}
```

```
color_to_base_dict = {'A': {'1': 'C', '0': 'A', '3': 'T', '2': 'G', '5': 'N', '4': 'N', '6': 'N', ':': 'N'}, 'C': {'1': 'A', '0':
```

```
key = '.'
```

```
to_base_space(sequence)
```

```
to_color_space(sequence, adapter_base=None)
```

```
unknown_base = 'N'
```

```
unknown_color = '.'
```

```
value = 'N'
```

```
galaxy_utils.sequence.transform.DNA_complement(sequence)
```

```
galaxy_utils.sequence.transform.DNA_reverse_complement(sequence)
```

```
galaxy_utils.sequence.transform.RNA_complement(sequence)
```

```
galaxy_utils.sequence.transform.RNA_reverse_complement(sequence)
```

```
galaxy_utils.sequence.transform.reverse(sequence)
```

```
galaxy_utils.sequence.transform.to_DNA(sequence)
```

```
galaxy_utils.sequence.transform.to_RNA(sequence)
```

vcf Module

```
class galaxy_utils.sequence.vcf.Reader(fh)
    Bases: object

    next()

class galaxy_utils.sequence.vcf.VariantCall(vcf_line, metadata, sample_names)
    Bases: object

    classmethod get_class_by_format(format)

    header_startswith = None

    required_header_fields = None

    required_header_length = None

    version = None

class galaxy_utils.sequence.vcf.VariantCall33(vcf_line, metadata, sample_names)
    Bases: galaxy_utils.sequence.vcf.VariantCall

    header_startswith = '#CHROM\tPOS\tID\tREF\tALT\tQUAL\tFILTER\tINFO'

    required_header_fields = ['#CHROM', 'POS', 'ID', 'REF', 'ALT', 'QUAL', 'FILTER', 'INFO']

    required_header_length = 8

    version = 'VCFv3.3'

class galaxy_utils.sequence.vcf.VariantCall40(vcf_line, metadata, sample_names)
    Bases: galaxy_utils.sequence.vcf.VariantCall33

    version = 'VCFv4.0'

class galaxy_utils.sequence.vcf.VariantCall41(vcf_line, metadata, sample_names)
    Bases: galaxy_utils.sequence.vcf.VariantCall40

    version = 'VCFv4.1'

galaxy_utils.sequence.vcf.format
    alias of VariantCall41
```

1.2.4 log_tempfile Module

```
class log_tempfile.TempFile
    Bases: object

    NamedTemporaryFile(*args, **kwargs)

    mkstemp(*args, **kwargs)
```

1.2.5 mimeparse Module

MIME-Type Parser

This module provides basic functions for handling mime-types. It can handle matching mime-types against a list of media-ranges. See section 14.1 of the HTTP specification [RFC 2616] for a complete explanation.

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.1>

Contents:

- `parse_mime_type()`: Parses a mime-type into its component parts.
- `parse_media_range()`: Media-ranges are mime-types with wild-cards and a 'q' quality parameter.
- `quality()`: Determines the quality ('q') of a mime-type when compared against a list of media-ranges.
- `quality_parsed()`: Just like `quality()` except the second parameter must be pre-parsed.
- `best_match()`: Choose the mime-type with the highest quality ('q') from a list of candidates.

`mimeparse.best_match(supported, header)`

Takes a list of supported mime-types and finds the best match for all the media-ranges listed in header. The value of header must be a string that conforms to the format of the HTTP Accept: header. The value of 'supported' is a list of mime-types.

```
>>> best_match(['application/xbel+xml', 'text/xml'], 'text/*;q=0.5,*/*; q=0.1')
'text/xml'
```

`mimeparse.fitness_and_quality_parsed(mime_type, parsed_ranges)`

Find the best match for a given mime-type against a list of media_ranges that have already been parsed by `parse_media_range()`. Returns a tuple of the fitness value and the value of the 'q' quality parameter of the best match, or (-1, 0) if no match was found. Just as for `quality_parsed()`, 'parsed_ranges' must be a list of parsed media ranges.

`mimeparse.parse_media_range(range)`

Carves up a media range and returns a tuple of the (type, subtype, params) where 'params' is a dictionary of all the parameters for the media range. For example, the media range 'application/*;q=0.5' would get parsed into:

In addition this function also guarantees that there is a value for 'q' in the params dictionary, filling it in with a proper default if necessary.

`mimeparse.parse_mime_type(mime_type)`

Carves up a mime-type and returns a tuple of the (type, subtype, params) where 'params' is a dictionary of all the parameters for the media range. For example, the media range 'application/xhtml;q=0.5' would get parsed into:

('application', 'xhtml', {'q', '0.5'})

`mimeparse.quality(mime_type, ranges)`

Returns the quality 'q' of a mime-type when compared against the media-ranges in ranges. For example:

```
>>> quality('text/html', 'text/*;q=0.3, text/html;q=0.7, text/html;level=1, text/html;level=2;q=0.4')
0.7
```

`mimeparse.quality_parsed(mime_type, parsed_ranges)`

Find the best match for a given mime-type against a list of media_ranges that have already been parsed by `parse_media_range()`. Returns the 'q' quality parameter of the best match, 0 if no match was found. This function behaves the same as `quality()` except that 'parsed_ranges' must be a list of parsed media ranges.

1.2.6 pkg_resources Module

Package resource API

A resource is a logical file contained within a package, or a logical subdirectory thereof. The package resource API expects resource names to have their path parts separated with /, *not* whatever the local path separator is. Do not use `os.path` operations to manipulate resource names being passed into the API.

The package resource API is designed to work with normal filesystem packages, .egg files, and unpacked .egg files. It can also work in a limited way with .zip files and with custom PEP 302 loaders that support the `get_data()` method.

`pkg_resources.require(req_str)`

`pkg_resources.get_provider(moduleOrReq)`

Return an IResourceProvider for the named module or requirement

`pkg_resources.get_distribution(dist)`

Return a current distribution object for a Requirement or string

`pkg_resources.load_entry_point(dist, group, name)`

Return *name* entry point of *group* for *dist* or raise ImportError

`pkg_resources.get_entry_map(dist, group=None)`

Return the entry point map for *group*, or the full entry map

`pkg_resources.get_entry_info(dist, group, name)`

Return the EntryPoint object for *group* + *name*, or None

`pkg_resources.declare_namespace(packageName)`

Declare that package *'packageName'* is a namespace package

`pkg_resources.find_distributions(path_item, only=False)`

Yield distributions accessible via *path_item*

`pkg_resources.get_default_cache()`

Determine the default cache location

This returns the `PYTHON_EGG_CACHE` environment variable, if set. Otherwise, on Windows, it returns a "Python-Eggs" subdirectory of the "Application Data" directory. On all other systems, it's "`~/.python-eggs`".

class `pkg_resources.Environment` (*search_path=None, platform='linux-x86_64', python='2.7'*)

Bases: object

Searchable snapshot of distributions on a search path

add (*dist*)

Add *dist* if we `can_add()` it and it isn't already added

best_match (*req, working_set, installer=None*)

Find distribution best matching *req* and usable on *working_set*

This calls the `find(req)` method of the *working_set* to see if a suitable distribution is already active. (This may raise `VersionConflict` if an unsuitable version of the project is already active in the specified *working_set*.) If a suitable distribution isn't active, this method returns the newest distribution in the environment that meets the Requirement in *req*. If no suitable distribution is found, and *installer* is supplied, then the result of calling the environment's `obtain(req, installer)` method will be returned.

can_add (*dist*)

Is distribution *dist* acceptable for this environment?

The distribution must match the platform and python version requirements specified when this environment was created, or False is returned.

obtain (*requirement, installer=None*)

Obtain a distribution matching *requirement* (e.g. via download)

Obtain a distro that matches requirement (e.g. via download). In the base `Environment` class, this routine just returns `installer(requirement)`, unless *installer* is None, in which case None is

returned instead. This method is a hook that allows subclasses to attempt other ways of obtaining a distribution before falling back to the *installer* argument.

remove (*dist*)

Remove *dist* from the environment

scan (*search_path=None*)

Scan *search_path* for distributions usable in this environment

Any distributions found are added to the environment. *search_path* should be a sequence of `sys.path` items. If not supplied, `sys.path` is used. Only distributions conforming to the platform/python version defined at initialization are added.

class `pkg_resources.WorkingSet` (*entries=None*)

Bases: `object`

A collection of active distributions on `sys.path` (or a similar list)

add (*dist, entry=None, insert=True*)

Add *dist* to working set, associated with *entry*

If *entry* is unspecified, it defaults to the `.location` of *dist*. On exit from this routine, *entry* is added to the end of the working set's `.entries` (if it wasn't already present).

dist is only added to the working set if it's for a project that doesn't already have a distribution in the set. If it's added, any callbacks registered with the `subscribe()` method will be called.

add_entry (*entry*)

Add a path item to `.entries`, finding any distributions on it

`find_distributions(entry, True)` is used to find distributions corresponding to the path entry, and they are added. *entry* is always appended to `.entries`, even if it is already present. (This is because `sys.path` can contain the same value more than once, and the `.entries` of the `sys.path` `WorkingSet` should always equal `sys.path`.)

find (*req*)

Find a distribution matching requirement *req*

If there is an active distribution for the requested project, this returns it as long as it meets the version requirement specified by *req*. But, if there is an active distribution for the project and it does *not* meet the *req* requirement, `VersionConflict` is raised. If there is no active distribution for the requested project, `None` is returned.

find_plugins (*plugin_env, full_env=None, installer=None, fallback=True*)

Find all activatable distributions in *plugin_env*

Example usage:

```
distributions, errors = working_set.find_plugins(
    Environment(plugin_dirlist)
)
map(working_set.add, distributions) # add plugins+libs to sys.path
print 'Could not load', errors     # display errors
```

The *plugin_env* should be an `Environment` instance that contains only distributions that are in the project's "plugin directory" or directories. The *full_env*, if supplied, should be an `Environment` contains all currently-available distributions. If *full_env* is not supplied, one is created automatically from the `WorkingSet` this method is called on, which will typically mean that every directory on `sys.path` will be scanned for distributions.

installer is a standard installer callback as used by the `resolve()` method. The *fallback* flag indicates whether we should attempt to resolve older versions of a plugin if the newest version cannot be resolved.

This method returns a 2-tuple: (*distributions*, *error_info*), where *distributions* is a list of the distributions found in *plugin_env* that were loadable, along with any other distributions that are needed to resolve their dependencies. *error_info* is a dictionary mapping unloadable plugin distributions to an exception instance describing the error that occurred. Usually this will be a `DistributionNotFound` or `VersionConflict` instance.

iter_entry_points (*group*, *name=None*)

Yield entry point objects from *group* matching *name*

If *name* is `None`, yields all entry points in *group* from all distributions in the working set, otherwise only ones matching both *group* and *name* are yielded (in distribution order).

require (**requirements*)

Ensure that distributions matching *requirements* are activated

requirements must be a string or a (possibly-nested) sequence thereof, specifying the distributions and versions required. The return value is a sequence of the distributions that needed to be activated to fulfill the requirements; all relevant distributions are included, even if they were already activated in this working set.

resolve (*requirements*, *env=None*, *installer=None*)

List all distributions needed to (recursively) meet *requirements*

requirements must be a sequence of `Requirement` objects. *env*, if supplied, should be an `Environment` instance. If not supplied, it defaults to all distributions available within any entry or distribution in the working set. *installer*, if supplied, will be invoked with each requirement that cannot be met by an already-installed distribution; it should return a `Distribution` or `None`.

run_script (*requires*, *script_name*)

Locate distribution for *requires* and run *script_name* script

subscribe (*callback*)

Invoke *callback* for all distributions (including existing ones)

class `pkg_resources.ResourceManager`

Manage resource extraction and packages

cleanup_resources (*force=False*)

Delete all extracted resource files and directories, returning a list of the file and directory names that could not be successfully removed. This function does not have any concurrency protection, so it should generally only be called when the extraction path is a temporary directory exclusive to a single process. This method is not automatically called; you must call it explicitly or register it as an `atexit` function if you wish to ensure cleanup of a temporary directory used for extractions.

extraction_error ()

Give an error message for problems extracting file(s)

extraction_path = `None`

get_cache_path (*archive_name*, *names=()*)

Return absolute location in cache for *archive_name* and *names*

The parent directory of the resulting path will be created if it does not already exist. *archive_name* should be the base filename of the enclosing egg (which may not be the name of the enclosing zipfile!), including its `".egg"` extension. *names*, if provided, should be a sequence of path name parts “under” the egg’s extraction location.

This method should only be called by resource providers that need to obtain an extraction location, and only for names they intend to extract, as it tracks the generated names for possible cleanup later.

postprocess (*tempname*, *filename*)

Perform any platform-specific postprocessing of *tempname*

This is where Mac header rewrites should be done; other platforms don't have anything special they should do.

Resource providers should call this method **ONLY** after successfully extracting a compressed resource. They must **NOT** call it on resources that are already in the filesystem.

tempname is the current (temporary) name of the file, and *filename* is the name it will be renamed to by the caller after this routine returns.

resource_exists (*package_or_requirement*, *resource_name*)

Does the named resource exist?

resource_filename (*package_or_requirement*, *resource_name*)

Return a true filesystem path for specified resource

resource_isdir (*package_or_requirement*, *resource_name*)

Is the named resource an existing directory?

resource_listdir (*package_or_requirement*, *resource_name*)

List the contents of the named resource directory

resource_stream (*package_or_requirement*, *resource_name*)

Return a readable file-like object for specified resource

resource_string (*package_or_requirement*, *resource_name*)

Return specified resource as a string

set_extraction_path (*path*)

Set the base path where resources will be extracted to, if needed.

If you do not call this routine before any extractions take place, the path defaults to the return value of `get_default_cache()`. (Which is based on the `PYTHON_EGG_CACHE` environment variable, with various platform-specific fallbacks. See that routine's documentation for more details.)

Resources are extracted to subdirectories of this path based upon information given by the `IResourceProvider`. You may set this to a temporary directory, but then you must call `cleanup_resources()` to delete the extracted files when done. There is no guarantee that `cleanup_resources()` will be able to remove all extracted files.

(Note: you may not change the extraction path for a given resource manager once resources have been extracted, unless you first call `cleanup_resources()`.)

class `pkg_resources.Distribution` (*location=None*, *metadata=None*, *project_name=None*, *version=None*, *py_version='2.7'*, *platform=None*, *precedence=3*)

Bases: `object`

Wrap an actual or potential `sys.path` entry w/metadata

PKG_INFO = 'PKG-INFO'

activate (*path=None*)

Ensure distribution is importable on *path* (default=`sys.path`)

as_requirement ()

Return a `Requirement` that matches this distribution exactly

check_version_conflict ()

clone (***kw*)

Copy this distribution, substituting in any changed keyword args

egg_name ()

Return what this distribution's standard `.egg` filename should be

extras

```

classmethod from_filename (filename, metadata=None, **kw)
classmethod from_location (location, basename, metadata=None, **kw)
get_entry_info (group, name)
    Return the EntryPoint object for group + name, or None
get_entry_map (group=None)
    Return the entry point map for group, or the full entry map
has_version ()
hashcmp
insert_on (path, loc=None)
    Insert self.location in path before its nearest parent directory
key
load_entry_point (group, name)
    Return the name entry point of group or raise ImportError
parsed_version
requires (extras=())
    List of Requirements needed for this distro if extras are used
version
class pkg_resources.Requirement (project_name, specs, extras)

    static parse (s)
class pkg_resources.EntryPoint (name, module_name, attrs=(), extras=(), dist=None)
    Bases: object

    Object representing an advertised importable object
load (require=True, env=None, installer=None)
classmethod parse (src, dist=None)
    Parse a single entry point from string src

    Entry point syntax follows the form:

```

<pre>name = some.module:some.attr [extral,extra2]</pre>

```

    The entry name and module name are required, but the :attrs and [extras] parts are optional
classmethod parse_group (group, lines, dist=None)
    Parse an entry point group
classmethod parse_map (data, dist=None)
    Parse a map of entry point groups
require (env=None, installer=None)
exception pkg_resources.ResolutionError
    Bases: exceptions.Exception

    Abstract base for dependency resolution errors
exception pkg_resources.VersionConflict
    Bases: pkg_resources.ResolutionError

    An already-installed version conflicts with the requested version

```

exception `pkg_resources.DistributionNotFound`

Bases: `pkg_resources.ResolutionError`

A requested distribution was not found

exception `pkg_resources.UnknownExtra`

Bases: `pkg_resources.ResolutionError`

Distribution doesn't have an "extra feature" of the given name

exception `pkg_resources.ExtractionError`

Bases: `exceptions.RuntimeError`

An error occurred extracting a resource

The following attributes are available from instances of this exception:

manager The resource manager that raised this exception

cache_path The base directory for resource extraction

original_error The exception instance that caused extraction to fail

`pkg_resources.parse_requirements(strs)`

Yield Requirement objects for each specification in *strs*

strs must be an instance of `basestring`, or a (possibly-nested) iterable thereof.

`pkg_resources.parse_version(s)`

Convert a version string to a chronologically-sortable key

This is a rough cross between `distutils.StrictVersion` and `LooseVersion`; if you give it versions that would work with `StrictVersion`, then it behaves the same; otherwise it acts like a slightly-smarter `LooseVersion`. It is *possible* to create pathological version coding schemes that will fool this parser, but they should be very rare in practice.

The returned value will be a tuple of strings. Numeric portions of the version are padded to 8 digits so they will compare numerically, but without relying on how numbers compare relative to strings. Dots are dropped, but dashes are retained. Trailing zeros between alpha segments or dashes are suppressed, so that e.g. "2.4.0" is considered the same as "2.4". Alphanumeric parts are lower-cased.

The algorithm assumes that strings like "-" and any alpha string that alphabetically follows "final" represents a "patch level". So, "2.4-1" is assumed to be a branch or patch of "2.4", and therefore "2.4.1" is considered newer than "2.4-1", which in turn is newer than "2.4".

Strings like "a", "b", "c", "alpha", "beta", "candidate" and so on (that come before "final" alphabetically) are assumed to be pre-release versions, so that the version "2.4" is considered newer than "2.4a1".

Finally, to handle miscellaneous cases, the strings "pre", "preview", and "rc" are treated as if they were "c", i.e. as though they were release candidates, and therefore are not as new as a version string that does not contain them, and "dev" is replaced with an '@' so that it sorts lower than any other pre-release tag.

`pkg_resources.safe_name(name)`

Convert an arbitrary string to a standard distribution name

Any runs of non-alphanumeric/. characters are replaced with a single '-'.

`pkg_resources.safe_version(version)`

Convert an arbitrary string to a standard version string

Spaces become dots, and all other non-alphanumeric characters become dashes, with runs of multiple dashes condensed to a single dash.

`pkg_resources.get_platform()`

`pkg_resources.compatible_platforms(provided, required)`

Can code for the *provided* platform run on the *required* platform?

Returns true if either platform is `None`, or the platforms are equal.

XXX Needs compatibility checks for Linux and other unixy OSes.

`pkg_resources.yield_lines(strs)`

Yield non-empty/non-comment lines of a basestring or sequence

`pkg_resources.split_sections(s)`

Split a string or iterable thereof into (section,content) pairs

Each `section` is a stripped version of the section header (“[section]”) and each `content` is a list of stripped lines excluding blank lines and comment-only lines. If there are any such lines before the first section header, they’re returned in a first `section` of `None`.

`pkg_resources.safe_extra(extra)`

Convert an arbitrary string to a standard ‘extra’ name

Any runs of non-alphanumeric characters are replaced with a single ‘_’, and the result is always lowercased.

`pkg_resources.to_filename(name)`

Convert a project or version name to its filename-escaped form

Any ‘-’ characters are currently replaced with ‘_’.

`pkg_resources.invalid_marker(text)`

Validate text as a PEP 426 environment marker; return exception or False

`pkg_resources.evaluate_marker(text, extra=None, _ops={})`

Evaluate a PEP 426 environment marker on CPython 2.4+. Return a boolean indicating the marker result in this environment. Raise `SyntaxError` if marker is invalid.

This implementation uses the ‘parser’ module, which is not implemented on Jython and has been superseded by the ‘ast’ module in Python 2.6 and later.

`pkg_resources.ensure_directory(path)`

Ensure that the parent directory of *path* exists

`pkg_resources.normalize_path(filename)`

Normalize a file/dir name for comparison purposes

class `pkg_resources.IMetadataProvider`

get_metadata (*name*)

The named metadata resource as a string

get_metadata_lines (*name*)

Yield named metadata resource as list of non-blank non-comment lines

Leading and trailing whitespace is stripped from each line, and lines with # as the first non-blank character are omitted.

has_metadata (*name*)

Does the package’s distribution contain the named metadata?

metadata_isdir (*name*)

Is the named metadata a directory? (like `os.path.isdir()`)

metadata_listdir (*name*)

List of metadata names in the directory (like `os.listdir()`)

run_script (*script_name*, *namespace*)

Execute the named script in the supplied namespace dictionary

```
class pkg_resources.IResourceProvider
    Bases: pkg_resources.IMetadataProvider

    An object that provides access to package resources

    get_resource_filename (manager, resource_name)
        Return a true filesystem path for resource_name

        manager must be an IResourceManager

    get_resource_stream (manager, resource_name)
        Return a readable file-like object for resource_name

        manager must be an IResourceManager

    get_resource_string (manager, resource_name)
        Return a string containing the contents of resource_name

        manager must be an IResourceManager

    has_resource (resource_name)
        Does the package contain the named resource?

    resource_isdir (resource_name)
        Is the named resource a directory? (like os.path.isdir())

    resource_listdir (resource_name)
        List of resource names in the directory (like os.listdir())
```

```
class pkg_resources.FileMetadata (path)
    Bases: pkg_resources.EmptyProvider

    Metadata handler for standalone PKG-INFO files

    Usage:
```

```
metadata = FileMetadata("/path/to/PKG-INFO")
```

This provider rejects all data and metadata requests except for PKG-INFO, which is treated as existing, and will be the contents of the file at the provided location.

```
get_metadata (name)
get_metadata_lines (name)
has_metadata (name)
```

```
class pkg_resources.PathMetadata (path, egg_info)
    Bases: pkg_resources.DefaultProvider

    Metadata provider for egg directories

    Usage:
```

```
# Development eggs:

egg_info = "/path/to/PackageName.egg-info"
base_dir = os.path.dirname(egg_info)
metadata = PathMetadata(base_dir, egg_info)
dist_name = os.path.splitext(os.path.basename(egg_info))[0]
dist = Distribution(base_dir, project_name=dist_name, metadata=metadata)

# Unpacked egg directories:
```



```
egg_path = "/path/to/PackageName-ver-pyver-etc.egg"
metadata = PathMetadata(egg_path, os.path.join(egg_path, 'EGG-INFO'))
dist = Distribution.from_filename(egg_path, metadata=metadata)
```

```
class pkg_resources.EggMetadata (importer)
    Bases: pkg_resources.ZipProvider

    Metadata provider for .egg files

class pkg_resources.EmptyProvider
    Bases: pkg_resources.NullProvider

    Provider that returns nothing for all requests

    module_path = None

class pkg_resources.NullProvider (module)
    Try to implement resources and metadata for arbitrary PEP 302 loaders

    egg_info = None
    egg_name = None
    get_metadata (name)
    get_metadata_lines (name)
    get_resource_filename (manager, resource_name)
    get_resource_stream (manager, resource_name)
    get_resource_string (manager, resource_name)
    has_metadata (name)
    has_resource (resource_name)
    loader = None
    metadata_isdir (name)
    metadata_listdir (name)
    resource_isdir (resource_name)
    resource_listdir (resource_name)
    run_script (script_name, namespace)

class pkg_resources.EggProvider (module)
    Bases: pkg_resources.NullProvider

    Provider based on a virtual filesystem

class pkg_resources.DefaultProvider (module)
    Bases: pkg_resources.EggProvider

    Provides access to package resources in the filesystem

    get_resource_stream (manager, resource_name)

class pkg_resources.ZipProvider (module)
    Bases: pkg_resources.EggProvider

    Resource support for zips and eggs

    eagers = None
```

get_resource_filename (*manager, resource_name*)

`pkg_resources.register_finder` (*importer_type, distribution_finder*)

Register *distribution_finder* to find distributions in `sys.path` items

importer_type is the type or class of a PEP 302 “Importer” (`sys.path` item handler), and *distribution_finder* is a callable that, passed a path item and the importer instance, yields `Distribution` instances found on that path item. See `pkg_resources.find_on_path` for an example.

`pkg_resources.register_namespace_handler` (*importer_type, namespace_handler*)

Register *namespace_handler* to declare namespace packages

importer_type is the type or class of a PEP 302 “Importer” (`sys.path` item handler), and *namespace_handler* is a callable like this:

```
def namespace_handler(importer, path_entry, moduleName, module):
    # return a path_entry to use for child packages
```

Namespace handlers are only called if the importer object has already agreed that it can handle the relevant path item, and they should only return a subpath if the module `__path__` does not already contain an equivalent subpath. For an example namespace handler, see `pkg_resources.file_ns_handler`.

`pkg_resources.register_loader_type` (*loader_type, provider_factory*)

Register *provider_factory* to make providers for *loader_type*

loader_type is the type or class of a PEP 302 module `__loader__`, and *provider_factory* is a function that, passed a *module* object, returns an `IResourceProvider` for that module.

`pkg_resources.fixup_namespace_packages` (*path_item, parent=None*)

Ensure that previously-declared namespace packages include *path_item*

`pkg_resources.get_importer` (*path_item*)

Retrieve a PEP 302 importer for the given path item

The returned importer is cached in `sys.path_importer_cache` if it was newly created by a path hook.

If there is no importer, a wrapper around the basic import machinery is returned. This wrapper is never inserted into the importer cache (None is inserted instead).

The cache (or part of it) can be cleared manually if a rescan of `sys.path_hooks` is necessary.

`pkg_resources.AvailableDistributions`

alias of *Environment*

1.2.7 psyco_full Module

Attempt to call `psyco.full`, but ignore any errors.

Indices and tables

- `genindex`
- `modindex`
- `search`

g

[galaxy](#), 40
[galaxy.actions.admin](#), 41
[galaxy.config](#), 40
[galaxy.datatypes.assembly](#), 41
[galaxy.datatypes.binary](#), 42
[galaxy.datatypes.checkers](#), 46
[galaxy.datatypes.chrominfo](#), 46
[galaxy.datatypes.converters.bed_to_gff_converter](#), 86
[galaxy.datatypes.converters.bgzip](#), 86
[galaxy.datatypes.converters.fasta_to_len](#), 86
[galaxy.datatypes.converters.fasta_to_tabular_converter](#), 86
[galaxy.datatypes.converters.fastq_to_fqtoc](#), 86
[galaxy.datatypes.converters.fastqsolexa_to_fasta_converter](#), 87
[galaxy.datatypes.converters.fastqsolexa_to_qual_converter](#), 87
[galaxy.datatypes.converters.gff_to_bed_converter](#), 87
[galaxy.datatypes.converters.gff_to_interval_index_converter](#), 87
[galaxy.datatypes.converters.interval_to_bed_converter](#), 87
[galaxy.datatypes.converters.interval_to_bedstrict_converter](#), 87
[galaxy.datatypes.converters.interval_to_coverage](#), 87
[galaxy.datatypes.converters.interval_to_gff](#), 88
[galaxy.datatypes.converters.interval_to_interval_index_converter](#), 88
[galaxy.datatypes.converters.interval_to_tabix_converter](#), 88
[galaxy.datatypes.converters.lped_to_fped_converter](#), 88
[galaxy.datatypes.converters.lped_to_pbed_converter](#), 89
[galaxy.datatypes.converters.maf_to_fasta_converter](#), 89
[galaxy.datatypes.converters.maf_to_interval_converter](#), 89
[galaxy.datatypes.converters.pbed_ldreduced_converter](#), 89
[galaxy.datatypes.converters.pbed_to_lped_converter](#), 89
[galaxy.datatypes.converters.picard_interval_list_to](#), 90
[galaxy.datatypes.converters.sam_to_bam](#), 90
[galaxy.datatypes.converters.vcf_to_interval_index_converter](#), 90
[galaxy.datatypes.converters.vcf_to_vcf_bgzip](#), 90
[galaxy.datatypes.converters.vcf_to_fasta_converter](#), 90
[galaxy.datatypes.converters.wiggle_to_array_tree_converter](#), 90
[galaxy.datatypes.converters.wiggle_to_simple_converter](#), 90
[galaxy.datatypes.coverage](#), 46
[galaxy.datatypes.data](#), 46
[galaxy.datatypes.display_applications.application](#), 90
[galaxy.datatypes.display_applications.parameters](#), 91
[galaxy.datatypes.display_applications.util](#), 92
[galaxy.datatypes.genetics](#), 51
[galaxy.datatypes.images](#), 56
[galaxy.datatypes.interval](#), 60
[galaxy.datatypes.metadata](#), 66
[galaxy.datatypes.ngsindex](#), 70
[galaxy.datatypes.qualityscore](#), 70
[galaxy.datatypes.registry](#), 71
[galaxy.datatypes.sequence](#), 73
[galaxy.datatypes.sniff](#), 78
[galaxy.datatypes.tabular](#), 80
[galaxy.datatypes.tracks](#), 84

galaxy.datatypes.util, 93
galaxy.datatypes.util.gff_util, 93
galaxy.datatypes.util.image_util, 94
galaxy.datatypes.xml, 84
galaxy.eggs, 94
galaxy.eggs.dist, 96
galaxy.eggs.scramble, 96
galaxy.exceptions, 97
galaxy.external_services.actions, 100
galaxy.external_services.parameters, 101
galaxy.external_services.result_handlers, 103
galaxy.external_services.service, 102
galaxy.forms.forms, 104
galaxy.jobs, 106
galaxy.jobs.actions, 114
galaxy.jobs.actions.post, 114
galaxy.jobs.deferred, 116
galaxy.jobs.deferred.data_transfer, 116
galaxy.jobs.deferred.manual_data_transfer, 117
galaxy.jobs.deferred.pacific_biosciences, 117
galaxy.jobs.handler, 111
galaxy.jobs.manager, 113
galaxy.jobs.mapper, 113
galaxy.jobs.runners, 117
galaxy.jobs.runners.cli, 119
galaxy.jobs.runners.condor, 119
galaxy.jobs.runners.drmaa, 120
galaxy.jobs.runners.local, 120
galaxy.jobs.runners.lwr, 121
galaxy.jobs.runners.tasks, 121
galaxy.jobs.splitters.basic, 122
galaxy.jobs.splitters.multi, 122
galaxy.jobs.transfer_manager, 113
galaxy.managers, 148
galaxy.managers.api_keys, 148
galaxy.managers.base, 148
galaxy.managers.citations, 149
galaxy.managers.collections, 149
galaxy.managers.collections_util, 150
galaxy.managers.context, 150
galaxy.managers.folders, 151
galaxy.managers.hdas, 152
galaxy.managers.histories, 153
galaxy.managers.lddas, 154
galaxy.managers.libraries, 154
galaxy.managers.roles, 156
galaxy.managers.tags, 156
galaxy.managers.workflows, 156
galaxy.model, 122
galaxy.model.custom_types, 145
galaxy.model.item_attrs, 146
galaxy.model.mapping, 146
galaxy.model.migrate.check, 147
galaxy.model.orm, 147
galaxy.model.orm.logging_connection_proxy, 147
galaxy.objectstore, 156
galaxy.objectstore.s3_multipart_upload, 159
galaxy.openid, 160
galaxy.openid.providers, 160
galaxy.openid.quota, 161
galaxy.sample_tracking.data_transfer, 161
galaxy.sample_tracking.external_service_types, 162
galaxy.sample_tracking.request_types, 162
galaxy.sample_tracking.sample, 163
galaxy.security, 163
galaxy.security.validate_user_input, 168
galaxy.tags, 168
galaxy.tag_handler, 168
galaxy.tools, 170
galaxy.tools.actions, 179
galaxy.tools.actions.history_imp_exp, 180
galaxy.tools.actions.metadata, 180
galaxy.tools.actions.upload, 180
galaxy.tools.actions.upload_common, 180
galaxy.tools.data, 181
galaxy.tools.deps, 183
galaxy.tools.exception_handling, 178
galaxy.tools.imp_exp, 184
galaxy.tools.imp_exp.export_history, 184
galaxy.tools.imp_exp.unpack_tar_gz_archive, 184
galaxy.tools.parameters, 185
galaxy.tools.parameters.basic, 185
galaxy.tools.parameters.dynamic_options, 197
galaxy.tools.parameters.grouping, 199
galaxy.tools.parameters.input_translation, 201
galaxy.tools.parameters.output, 201
galaxy.tools.parameters.sanitize, 205
galaxy.tools.parameters.validation, 206
galaxy.tools.search, 209
galaxy.tools.test, 178
galaxy.tools.util, 209
galaxy.tools.util.galaxyops, 212
galaxy.tools.util.maf_utilities, 210
galaxy.util, 213
galaxy.util.aliaspickler, 217
galaxy.util.backports, 224

[galaxy.util.backports.importlib](#), 225
[galaxy.util.bunch](#), 217
[galaxy.util.debugging](#), 217
[galaxy.util.expressions](#), 218
[galaxy.util.hash_util](#), 218
[galaxy.util.heartbeat](#), 218
[galaxy.util.inflection](#), 219
[galaxy.util.json](#), 221
[galaxy.util.lrucache](#), 222
[galaxy.util.memdump](#), 222
[galaxy.util.none_like](#), 222
[galaxy.util.odict](#), 223
[galaxy.util.sanitize_html](#), 223
[galaxy.util.streamball](#), 223
[galaxy.util.template](#), 223
[galaxy.util.topsort](#), 224
[galaxy.visualization](#), 225
[galaxy.visualization.data_providers](#), 226
[galaxy.visualization.data_providers.basic](#), 226
[galaxy.visualization.data_providers.phylloviz](#), 227
[galaxy.visualization.data_providers.phylloviz.basic_parser](#), 227
[galaxy.visualization.data_providers.phylloviz.newick_parser](#), 228
[galaxy.visualization.data_providers.phylloviz.phylx_parser](#), 228
[galaxy.visualization.data_providers.phylloviz.phylx_parser_base](#), 228
[galaxy.visualization.genome](#), 229
[galaxy.visualization.genomes](#), 225
[galaxy.visualization.tracks](#), 229
[galaxy.web](#), 229
[galaxy.web.base.controller](#), 234
[galaxy.web.base.controllers.admin](#), 240
[galaxy.web.buildapp](#), 229
[galaxy.web.form_builder](#), 229
[galaxy.web.framework](#), 241
[galaxy.web.framework.base](#), 242
[galaxy.web.framework.helpers](#), 245
[galaxy.web.framework.helpers.grid](#), 245
[galaxy.web.framework.middleware](#), 252
[galaxy.web.framework.middleware.profile](#), 252
[galaxy.web.framework.middleware.remoteuser](#), 253
[galaxy.web.framework.middleware.static](#), 253
[galaxy.web.framework.middleware.translogger](#), 253
[galaxy.web.framework.middleware.xforwardedfor](#), 253
[galaxy.web.framework.openid_manager](#), 244
[galaxy.web.params](#), 234
[galaxy.web.security](#), 254
[galaxy.webapps](#), 254
[galaxy.webapps.demo_sequencer](#), 256
[galaxy.webapps.demo_sequencer.app](#), 256
[galaxy.webapps.demo_sequencer.buildapp](#), 256
[galaxy.webapps.demo_sequencer.config](#), 256
[galaxy.webapps.demo_sequencer.controllers](#), 257
[galaxy.webapps.demo_sequencer.controllers.common](#), 257
[galaxy.webapps.demo_sequencer.framework](#), 257
[galaxy.webapps.demo_sequencer.registry](#), 257
[galaxy.webapps.galaxy.api.annotations](#), 263
[galaxy.webapps.galaxy.api.authenticate](#), 264
[galaxy.webapps.galaxy.api.configuration](#), 264
[galaxy.webapps.galaxy.api.dataset_collections](#), 265
[galaxy.webapps.galaxy.api.datatypes](#), 265
[galaxy.webapps.galaxy.api.extended_metadata](#), 265
[galaxy.webapps.galaxy.api.folder_contents](#), 266
[galaxy.webapps.galaxy.api.folders](#), 267
[galaxy.webapps.galaxy.api.forms](#), 268
[galaxy.webapps.galaxy.api.ftp_files](#), 269
[galaxy.webapps.galaxy.api.genomes](#), 269
[galaxy.webapps.galaxy.api.group_roles](#), 269
[galaxy.webapps.galaxy.api.group_users](#), 270
[galaxy.webapps.galaxy.api.groups](#), 270
[galaxy.webapps.galaxy.api.histories](#), 270
[galaxy.webapps.galaxy.api.history_contents](#), 272
[galaxy.webapps.galaxy.api.item_tags](#), 275
[galaxy.webapps.galaxy.api.job_files](#), 275
[galaxy.webapps.galaxy.api.jobs](#), 276
[galaxy.webapps.galaxy.api.lda_datasets](#), 278
[galaxy.webapps.galaxy.api.libraries](#), 280
[galaxy.webapps.galaxy.api.library_contents](#), 283
[galaxy.webapps.galaxy.api.metrics](#), 285
[galaxy.webapps.galaxy.api.page_revisions](#), 285
[galaxy.webapps.galaxy.api.pages](#), 286

galaxy.webapps.galaxy.api.provenance, 287
 galaxy.webapps.galaxy.api.quotas, 287
 galaxy.webapps.galaxy.api.request_types, 287
 galaxy.webapps.galaxy.api.requests, 288
 galaxy.webapps.galaxy.api.roles, 288
 galaxy.webapps.galaxy.api.samples, 288
 galaxy.webapps.galaxy.api.search, 289
 galaxy.webapps.galaxy.api.tool_data, 289
 galaxy.webapps.galaxy.api.tool_shed_repository, 289
 galaxy.webapps.galaxy.api.users, 292
 galaxy.webapps.galaxy.api.visualizations, 292
 galaxy.webapps.galaxy.api.workflows, 292
 galaxy.webapps.galaxy.buildapp, 258
 galaxy.webapps.galaxy.controllers, 295
 galaxy.webapps.galaxy.controllers.admin, 295
 galaxy.webapps.galaxy.controllers.admin_toolshed, 301
 galaxy.webapps.galaxy.controllers.async, 303
 galaxy.webapps.galaxy.controllers.dataset, 304
 galaxy.webapps.galaxy.controllers.error, 306
 galaxy.webapps.galaxy.controllers.external_service, 306
 galaxy.webapps.galaxy.controllers.external_services, 307
 galaxy.webapps.galaxy.controllers.forms, 307
 galaxy.webapps.galaxy.controllers.history, 309
 galaxy.webapps.galaxy.controllers.library, 313
 galaxy.webapps.galaxy.controllers.library_admin, 313
 galaxy.webapps.galaxy.controllers.library_common, 314
 galaxy.webapps.galaxy.controllers.mobile, 316
 galaxy.webapps.galaxy.controllers.page, 316
 galaxy.webapps.galaxy.controllers.request_type, 320
 galaxy.webapps.galaxy.controllers.requests, 321
 galaxy.webapps.galaxy.controllers.requests_admin, 321
 galaxy.webapps.galaxy.controllers.requests_common, 323
 galaxy.webapps.galaxy.controllers.root, 325
 galaxy.webapps.galaxy.controllers.tag, 326
 galaxy.webapps.galaxy.controllers.tool_runner, 327
 galaxy.webapps.galaxy.controllers.ucsc_proxy, 327
 galaxy.webapps.galaxy.controllers.user, 327
 galaxy.webapps.galaxy.controllers.visualization, 329
 galaxy.webapps.galaxy.controllers.workflow, 335
 galaxy.webapps.reports, 338
 galaxy.webapps.reports.app, 338
 galaxy.webapps.reports.buildapp, 338
 galaxy.webapps.reports.config, 338
 galaxy.webapps.reports.controllers, 339
 galaxy.webapps.reports.controllers.jobs, 339
 galaxy.webapps.reports.controllers.root, 341
 galaxy.webapps.reports.controllers.sample_tracking, 341
 galaxy.webapps.reports.controllers.system, 342
 galaxy.webapps.reports.controllers.users, 343
 galaxy.webapps.reports.controllers.workflows, 343
 galaxy.workflow.modules, 345
 galaxy_utils.sequence.fasta, 350
 galaxy_utils.sequence.fastq, 350
 galaxy_utils.sequence.sequence, 354
 galaxy_utils.sequence.transform, 354
 galaxy_utils.sequence.vcf, 355

m

p

pkg_resources, 356
 psych_full, 366

A

- Ab1 (class in `galaxy.datatypes.binary`), 42
- `access_action()` (`galaxy.webapps.galaxy.controllers.external_services.ExternalServiceController` module method), 307
- `act_on_multiple_datasets()` (`galaxy.webapps.galaxy.controllers.library_common.LibraryCommon` module method), 314
- Action (class in `galaxy.security`), 163
- `ACTION_NAME` (`galaxy.datatypes.display_applications.parameters.DisplayDataValueWrapper` attribute), 92
- `action_name` (`galaxy.datatypes.display_applications.parameters.DisplayDataValueWrapper` attribute), 92
- `ACTION_NAME` (`galaxy.datatypes.display_applications.parameters.DisplayParameterValueWrapper` attribute), 92
- `action_name` (`galaxy.datatypes.display_applications.parameters.DisplayParameterValueWrapper` attribute), 92
- `action_type` (in module `galaxy.tools.parameters.output`), 204
- ActionBox (class in `galaxy.jobs.actions.post`), 114
- ActionInputError, 97
- `actions` (`galaxy.jobs.actions.post.ActionBox` attribute), 114
- ActionSection (class in `galaxy.external_services.service`), 102
- `activatable_datasets` (`galaxy.model.History` attribute), 128
- `activatable_folders()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 315
- `activatable_folders_and_library_datasets()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 315
- `activatable_library_datasets` (`galaxy.model.LibraryFolder` attribute), 136
- `activate()` (`galaxy.webapps.galaxy.controllers.user.User` method), 327
- `activate()` (`pkg_resources.Distribution` method), 360
- `activate_repository()` (`galaxy.webapps.galaxy.controllers.admin_toolsshed.AdminToolsshed` method), 301
- `active` (`galaxy.model.WorkflowInvocation` attribute), 143
- `active_contents` (`galaxy.model.History` attribute), 128
- `active_datasets_children_and_roles` (`galaxy.model.History` attribute), 128
- `active_folders()` (in module `galaxy.tools.actions.upload_common`), 180
- `active_folders()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 315
- `active_folders_and_library_datasets()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 315
- `add()` (`galaxy.managers.citations.CitationCollection` method), 149
- `add()` (`galaxy.util.streamball.StreamBall` method), 223
- `add()` (`pkg_resources.Environment` method), 357
- `add()` (`pkg_resources.WorkingSet` method), 358
- `add_api_controller()` (`galaxy.web.framework.base.WebApplication` method), 244
- `add_composite_file()` (`galaxy.datatypes.data.Data` method), 47
- `add_dataset()` (`galaxy.model.History` method), 128
- `add_dataset_collection()` (`galaxy.model.History` method), 128
- `add_display_app()` (`galaxy.datatypes.data.Data` method), 47
- `add_display_application()` (`galaxy.datatypes.data.Data` method), 47
- `add_dummy_datasets()` (`galaxy.workflow.modules.ToolModule` method), 347
- `add_dummy_datasets()` (`galaxy.workflow.modules.WorkflowModule` method), 348
- `add_entries()` (`galaxy.tools.data.ToolDataTable` method), 182
- `add_entry()` (`galaxy.tools.data.ToolDataTable` method), 182
- `add_entry()` (`pkg_resources.WorkingSet` method), 358
- `add_external_service_association()` (`galaxy.model.RequestType` method), 138
- `add_folder()` (`galaxy.model.LibraryFolder` method), 136
- `add_galaxy_session()` (`galaxy.model.History` method), 128
- `add_history()` (`galaxy.model.GalaxySession` method), 128

[add_history_datasets_to_library\(\)](#) (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 314
[add_implicit_input_collection\(\)](#) (galaxy.model.HistoryDatasetCollectionAssociation method), 130
[add_input\(\)](#) (galaxy.model.WorkflowInvocation method), 143
[add_input_dataset\(\)](#) (galaxy.model.Job method), 131
[add_input_dataset_collection\(\)](#) (galaxy.model.Job method), 131
[add_input_library_dataset\(\)](#) (galaxy.model.Job method), 131
[add_item\(\)](#) (galaxy.external_services.service.ExternalServiceActionsGroup method), 102
[add_item_annotation\(\)](#) (galaxy.model.item_attrs.UsesAnnotations method), 146
[add_library_dataset\(\)](#) (galaxy.model.LibraryFolder method), 136
[add_metric\(\)](#) (galaxy.model.HasJobMetrics method), 128
[add_new_entries_from_config_file\(\)](#) (galaxy.tools.data.ToolDataTableManager method), 182
[add_option\(\)](#) (galaxy.web.form_builder.SelectField method), 232
[add_output_dataset\(\)](#) (galaxy.model.Job method), 131
[add_output_dataset_collection\(\)](#) (galaxy.model.Job method), 131
[add_output_library_dataset\(\)](#) (galaxy.model.Job method), 132
[add_parameter\(\)](#) (galaxy.model.Job method), 132
[add_post_job_action\(\)](#) (galaxy.model.Job method), 132
[add_route\(\)](#) (galaxy.web.framework.base.WebApplication method), 244
[add_sample\(\)](#) (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 323
[add_samples\(\)](#) (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 323
[add_species\(\)](#) (galaxy.tools.util.maf_utilities.RegionAlignment method), 210
[add_sreg\(\)](#) (galaxy.web.framework.openid_manager.OpenIDManager method), 244
[add_tag_async\(\)](#) (galaxy.webapps.galaxy.controllers.tag.TagsController method), 326
[add_template\(\)](#) (galaxy.web.base.controller.UsesFormDefinitionMixin method), 236
[add_ui_controller\(\)](#) (galaxy.web.framework.base.WebApplication method), 244
[add_ui_controllers\(\)](#) (in module galaxy.webapps.demo_sequencer.buildapp), 256
[add_ui_controllers\(\)](#) (in module galaxy.webapps.reports.buildapp), 338
[add_validation_error\(\)](#) (galaxy.model.DatasetInstance method), 124
[add_visualization_revision\(\)](#) (galaxy.web.base.controller.UsesVisualizationMixin method), 239
[addAttributesToRoot\(\)](#) (galaxy.visualization.data_providers.phyloviz.baseparser.N method), 227
[addChildNode\(\)](#) (galaxy.visualization.data_providers.phyloviz.baseparser.N method), 227
[addChildrenToJson\(\)](#) (galaxy.visualization.data_providers.phyloviz.baseparser.N method), 227
[AddFrameData](#) (class in galaxy.webapps.galaxy.controllers.tool_runner), 271
[adding_to_library_dataset_files](#) (galaxy.model.Sample attribute), 139
[AdditionalValueFilter](#) (class in galaxy.tools.parameters.dynamic_options), 197
[addMiscToJson\(\)](#) (galaxy.visualization.data_providers.phyloviz.baseparser.N method), 227
[AddressField](#) (class in galaxy.web.form_builder), 229
[addRoot\(\)](#) (galaxy.visualization.data_providers.phyloviz.baseparser.PhyloTr method), 227
[Admin](#) (class in galaxy.web.base.controllers.admin), 240
[AdminActions](#) (class in galaxy.actions.admin), 41
[AdminGalaxy](#) (class in galaxy.webapps.galaxy.controllers.admin), 295
[AdminRequestsGrid](#) (class in galaxy.webapps.galaxy.controllers.requests_admin), 321
[AdminRequestsGrid.UserColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_admin), 321
[AdminRequiredException](#), 97
[AdminToolshed](#) (class in galaxy.webapps.galaxy.controllers.admin_toolshed), 301
[Affybatch](#) (class in galaxy.datatypes.genetics), 51
[Adding_metadata\(\)](#) (galaxy.datatypes.data.Data method), 47
[AliasPicklerModule](#) (class in galaxy.util.aliaspickler), 217
[AliasUnpickler](#) (class in galaxy.util.aliaspickler), 217
[AlignMixin](#) (class in galaxy.datatypes.sequence), 73
[all_eggs](#) (galaxy.eggs.Crate attribute), 94
[all_missing](#) (galaxy.eggs.Crate attribute), 94
[all_names](#) (galaxy.eggs.Crate attribute), 95
[all_roles\(\)](#) (galaxy.model.User method), 141
[allow_action\(\)](#) (galaxy.security.GalaxyRBACAgent method), 163
[allow_action\(\)](#) (galaxy.security.HostAgent method), 167
[allow_action_on_libitems\(\)](#) (galaxy.security.GalaxyRBACAgent method),

- 163
- `allow_datatype_change` (galaxy.datatypes.assembly.Velvet attribute), 42
- `allow_datatype_change` (galaxy.datatypes.data.Data attribute), 47
- `allow_datatype_change` (galaxy.datatypes.genetics.RexpBase attribute), 53
- `allow_datatype_change` (galaxy.datatypes.genetics.Rgenetics attribute), 54
- `allow_datatype_change` (galaxy.datatypes.interval.BedStrict attribute), 61
- `allow_datatype_change` (galaxy.datatypes.ngsindex.BowtieIndex attribute), 70
- `allow_user_access()` (galaxy.tools.DataManagerTool method), 170
- `allow_user_access()` (galaxy.tools.Tool method), 172
- `allowed()` (galaxy.web.framework.helpers.grids.GridOperation method), 249
- `always_conditional` (galaxy.eggs.GalaxyConfig attribute), 95
- Amos (class in galaxy.datatypes.assembly), 41
- `amount` (galaxy.model.Quota attribute), 137
- `annotate_async()` (galaxy.webapps.galaxy.controllers.datasets.DataSetController method), 304
- `annotate_async()` (galaxy.webapps.galaxy.controllers.workflows.WorkflowController method), 336
- `annotation_mapping()` (in module galaxy.model.mapping), 146
- `anon_user_api_value()` (galaxy.webapps.galaxy.api.users.UserAPIController method), 36, 292
- `anonymous` (galaxy.managers.context.ProvidesUserContext attribute), 151
- `api_key()` (galaxy.webapps.galaxy.api.users.UserAPIController method), 37, 292
- `api_keys()` (galaxy.webapps.galaxy.controllers.user.UserAPIController method), 328
- `api_path` (galaxy.jobs.deferred.pacific_biosciences_smrt_portals.SMRTPortal attribute), 117
- `api_payload_to_create_params()` (in module galaxy.managers.collections_util), 150
- ApiKeyManager (class in galaxy.managers.api_keys), 148
- APIKeys (class in galaxy.model), 122
- `app_factory()` (in module galaxy.webapps.demo_sequencer.buildapp), 256
- `app_factory()` (in module galaxy.webapps.galaxy.buildapp), 259
- `app_factory()` (in module galaxy.webapps.reports.buildapp), 338
- `append()` (galaxy.datatypes.metadata.MetadataSpecCollection method), 69
- `append_quality()` (galaxy_utils.sequence.sequence.SequencingRead method), 354
- `append_sequence()` (galaxy_utils.sequence.sequence.SequencingRead method), 354
- `apply_action()` (galaxy.tools.parameters.output.FormatToolOutputAction method), 202
- `apply_action()` (galaxy.tools.parameters.output.MetadataToolOutputAction method), 202
- `apply_action()` (galaxy.tools.parameters.output.ToolOutputAction method), 203
- `apply_action()` (galaxy.tools.parameters.output.ToolOutputActionCondition method), 203
- `apply_action()` (galaxy.tools.parameters.output.ToolOutputActionCondition method), 203
- `apply_action()` (galaxy.tools.parameters.output.ToolOutputActionGroup method), 204
- `apply_galaxy_conventions()` (galaxy_utils.sequence.fastq.fastqCSSangerRead method), 351
- `apply_galaxy_conventions()` (galaxy_utils.sequence.fastq.fastqSequencingRead method), 352
- `apply_item_tag()` (galaxy.tags.tag_handler.TagHandler method), 168
- `apply_item_tag()` (galaxy.tags.tag_handler.TagHandler method), 169
- `apply_query_filter()` (galaxy.web.framework.helpers.grids.GridOperation method), 247
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.admin.RoleListGrid method), 298
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.history.HistoryAllPublic method), 309
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.history.HistoryList method), 311
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.history.SharedHistory method), 312
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.library.LibraryList method), 313
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.page.HistoryDatabase method), 316
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.page.HistorySelect method), 317
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.page.ItemSelection method), 317
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.page.PageAllPublic method), 317
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.page.PageListGrid method), 319
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.requests.UserRequest method), 321
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.requests_admin.Dashboard method), 322
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.visualization.History method), 331
- `apply_query_filter()` (galaxy.webapps.galaxy.controllers.visualization.History method), 331

[apply_query_filter\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.TracksSelectionGrid (class in galaxy.webapps.galaxy.controllers.visualization.TracksSelectionGrid), 333)
[apply_query_filter\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.TracksSelectionGrid (class in galaxy.webapps.galaxy.controllers.visualization.TracksSelectionGrid), 333)
[apply_query_filter\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.TracksSelectionGrid (class in galaxy.webapps.galaxy.controllers.visualization.TracksSelectionGrid), 335)
[apply_query_filter\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAppPublicationGrid (class in galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAppPublicationGrid), 335)
[apply_query_filter\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAppPublicationGrid (class in galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAppPublicationGrid), 336)
[archive_dir](#) (galaxy.eggs.scramble.ScrambleEgg attribute), 96
[archive_download\(\)](#) (galaxy.webapps.galaxy.api.histories.HistoriesController (class in galaxy.webapps.galaxy.api.histories.HistoriesController), 15, 270)
[archive_export\(\)](#) (galaxy.webapps.galaxy.api.histories.HistoriesController (class in galaxy.webapps.galaxy.api.histories.HistoriesController), 15, 270)
[as_display_type\(\)](#) (galaxy.datatypes.data.Data method), 47
[as_display_type\(\)](#) (galaxy.model.DatasetInstance method), 124
[as_gbrowse_display_file\(\)](#) (galaxy.datatypes.tabular.Tabular method), 83
[as_requirement\(\)](#) (pkg_resources.Distribution method), 360
[as_ucsc_display_file\(\)](#) (galaxy.datatypes.genetics.GenomeGraph (class in galaxy.datatypes.genetics.GenomeGraph), 52)
[as_ucsc_display_file\(\)](#) (galaxy.datatypes.interval.Bed method), 60
[as_ucsc_display_file\(\)](#) (galaxy.datatypes.interval.BedGraph method), 60
[as_ucsc_display_file\(\)](#) (galaxy.datatypes.interval.Interval method), 64
[as_ucsc_display_file\(\)](#) (galaxy.datatypes.tabular.Tabular method), 83
[asbool\(\)](#) (in module galaxy.util), 213
[ascii_max](#) (galaxy_utils.sequence.fastq.fastqCSSangerRead attribute), 351
[ascii_max](#) (galaxy_utils.sequence.fastq.fastqIlluminaRead attribute), 351
[ascii_max](#) (galaxy_utils.sequence.fastq.fastqSangerRead attribute), 352
[ascii_max](#) (galaxy_utils.sequence.fastq.fastqSequencingRead attribute), 352
[ascii_max](#) (galaxy_utils.sequence.fastq.fastqSolexaRead attribute), 353
[ascii_min](#) (galaxy_utils.sequence.fastq.fastqCSSangerRead attribute), 351
[ascii_min](#) (galaxy_utils.sequence.fastq.fastqIlluminaRead attribute), 351
[ascii_min](#) (galaxy_utils.sequence.fastq.fastqSangerRead attribute), 352
[ascii_min](#) (galaxy_utils.sequence.fastq.fastqSequencingRead attribute), 352
[assert_sequence_quality_lengths\(\)](#) (galaxy_utils.sequence.fastq.fastqCSSangerRead attribute), 351
[associate_components\(\)](#) (galaxy.security.GalaxyRBACAgent method), 163
[associate_components\(\)](#) (galaxy.security.RBACAgent method), 167
[associate_group_role\(\)](#) (galaxy.security.GalaxyRBACAgent method), 163
[associate_user_group\(\)](#) (galaxy.security.GalaxyRBACAgent method), 163
[associate_user_role\(\)](#) (galaxy.security.GalaxyRBACAgent method), 163
[ASync](#) (class in galaxy.webapps.galaxy.controllers.async), 303
[async_template](#) (galaxy.web.framework.helpers.grid.Grid attribute), 247
[AsyncDataSourceTool](#) (class in galaxy.tools), 170
[AsynchronousJobRunner](#) (class in galaxy.jobs.runners), 117
[AsynchronousJobState](#) (class in galaxy.jobs.runners), 118
[AttributeValueSplitterFilter](#) (class in galaxy.tools.parameters.dynamic_options), 197
[AuthenticationController](#) (class in galaxy.webapps.galaxy.api.authenticate), 8, 264
[AuthenticationFailed](#), 97
[AuthenticationRequired](#), 97
[available_tracks](#) (galaxy.webapps.galaxy.controllers.visualization.HistoryData attribute), 331
[AvailableDistributions](#) (in module pkg_resources), 366
[Axt](#) (class in galaxy.datatypes.sequence), 73

B

[BadValue](#) (class in galaxy.tools), 170
[Bam](#) (class in galaxy.datatypes.binary), 42
[Base](#) (class in galaxy.util.inflection), 219
[base](#) (galaxy.web.framework.base.Request attribute), 243
[base](#) (galaxy_utils.sequence.transform.ColorSpaceConverter attribute), 354
[base_dataprovider\(\)](#) (galaxy.datatypes.data.Data method), 47
[Base_Parser](#) (class in galaxy.visualization.data_providers.phyloviz.basepars), 227

base_to_color_dict (galaxy_utils.sequence.transform.ColorSpaceConverter (class in galaxy.web.framework.helpers.grids), 245 attribute), 354

BaseAnnotationsController (class in galaxy.webapps.galaxy.api.annotations), 7, 263

BaseApiController (class in galaxy.web.base.controller), 234

BaseCitation (class in galaxy.managers.citations), 149

BaseController (class in galaxy.web.base.controller), 234

BaseDataProvider (class in galaxy.visualization.data_providers.basic), 226

BaseDataToolParameter (class in galaxy.tools.parameters.basic), 185

BaseExtendedMetadataController (class in galaxy.webapps.galaxy.api.extended_metadata), 9, 265

BaseField (class in galaxy.web.form_builder), 229

BaseItemTagsController (class in galaxy.webapps.galaxy.api.item_tags), 19, 275

BaseJobMetric (class in galaxy.model), 122

BaseJobRunner (class in galaxy.jobs.runners), 118

BaseParamParser (class in galaxy.web.params), 234

BaseProvenanceController (class in galaxy.webapps.galaxy.api.provenance), 31, 287

BaseUIController (class in galaxy.web.base.controller), 235

BaseURLToolParameter (class in galaxy.tools.parameters.basic), 185

Bcf (class in galaxy.datatypes.binary), 43

Bed (class in galaxy.datatypes.interval), 60

Bed12 (class in galaxy.datatypes.interval), 60

Bed6 (class in galaxy.datatypes.interval), 60

BedGraph (class in galaxy.datatypes.interval), 60

BedStrict (class in galaxy.datatypes.interval), 61

before_setting_metadata() (galaxy.datatypes.data.Data method), 47

best_match() (in module mimeparse), 356

best_match() (pkg_resources.Environment method), 357

BIBTEX_UNSET (galaxy.managers.citations.DoiCitation attribute), 149

BibtexCitation (class in galaxy.managers.citations), 149

BigBed (class in galaxy.datatypes.binary), 43

BigWig (class in galaxy.datatypes.binary), 43

Binary (class in galaxy.datatypes.binary), 44

biomart() (galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner method), 327

Bmp (class in galaxy.datatypes.images), 56

bookmarks_from_dataset() (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 333

BooleanColumn (class in galaxy.web.framework.helpers.grids), 245

BooleanExternalServiceActionsGroupWhen (class in galaxy.external_services.service), 102

BooleanFilter (class in galaxy.tools.parameters.output), 201

BooleanToolParameter (class in galaxy.tools.parameters.basic), 185

BowtieBaseIndex (class in galaxy.datatypes.ngsindex), 70

BowtieColorIndex (class in galaxy.datatypes.ngsindex), 70

BowtieIndex (class in galaxy.datatypes.ngsindex), 70

branch_deleted() (in module galaxy.webapps.galaxy.controllers.library_common), 315

browse_external_services() (galaxy.webapps.galaxy.controllers.external_service.ExternalService method), 306

browse_form_definitions() (galaxy.webapps.galaxy.controllers.forms.Forms method), 307

browse_libraries() (galaxy.webapps.galaxy.controllers.library.Library method), 313

browse_libraries() (galaxy.webapps.galaxy.controllers.library_admin.LibraryAdmin method), 313

browse_library() (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 314

browse_repositories() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 301

browse_repository() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 301

browse_request_types() (galaxy.webapps.galaxy.controllers.request_type.RequestType method), 320

browse_requests() (galaxy.webapps.galaxy.controllers.requests.Requests method), 321

browse_requests() (galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin method), 323

browse_tool_dependency() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 301

browse_tool_shed() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 301

browse_tool_sheds() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 301

browser_url (galaxy.web.framework.base.Request attribute), 243

bucket_proxy() (galaxy.webapps.galaxy.controllers.root.RootController method), 325

build() (galaxy.tools.parameters.basic.ToolParameter class method), 195

build_command_line() (galaxy.jobs.runners.BaseJobRunner method), 118

build_dependency_manager() (in module galaxy.tools.deps), 183

build_dependency_shell_commands() (galaxy.tools.Tool method), 172
 build_dir (galaxy.eggs.scramble.ScrambleEgg attribute), 96
 build_form_definition_field_widgets() (galaxy.webapps.galaxy.controllers.forms.Forms method), 307
 build_form_id_select_field() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 236
 build_from_current_history() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
 build_index() (galaxy.tools.search.ToolBoxSearch method), 209
 build_initial_query() (galaxy.web.framework.helpers.grids.Grid method), 247
 build_initial_query() (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid method), 299
 build_initial_query() (galaxy.webapps.galaxy.controllers.dataset.HistogramDataVisualizationGrid method), 305
 build_initial_query() (galaxy.webapps.galaxy.controllers.forms.FormsGalaxy method), 308
 build_initial_query() (galaxy.webapps.galaxy.controllers.history.HistoryPublishedGrid method), 309
 build_initial_query() (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid method), 312
 build_initial_query() (galaxy.webapps.galaxy.controllers.library.BunchClassListGrid method), 313
 build_initial_query() (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid method), 317
 build_initial_query() (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid method), 329
 build_initial_query() (galaxy.webapps.galaxy.controllers.visualization.HistoryDatasetsSelectionGrid method), 331
 build_initial_query() (galaxy.webapps.galaxy.controllers.visualization.TracksterSelectionGrid method), 333
 build_initial_query() (galaxy.webapps.galaxy.controllers.visualization.VisualizationAllPublishedGrid method), 333
 build_initial_query() (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAllPublishedGrid method), 335
 build_initial_query() (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid method), 340
 build_initial_query() (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid method), 342
 build_initial_query() (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid method), 344
 build_invocations_query() (galaxy.managers.workflows.WorkflowsManager method), 156
 build_maf_index() (in module galaxy.tools.util.maf_utilities), 210
 build_maf_index_species_chromosomes() (in module galaxy.tools.util.maf_utilities), 210
 build_object_store_from_config() (in module galaxy.objectstore), 159
 build_parameter_dict() (galaxy.datatypes.display_applications.application.DisplayApp method), 91
 build_path() (galaxy.webapps.galaxy.api.folder_contents.FolderContentsController method), 10, 266
 build_redirect_url_params() (galaxy.tools.Tool method), 172
 build_rename_datasets_for_sample_select_field() (in module galaxy.webapps.galaxy.controllers.requests_admin), 323
 build_select_field() (in module galaxy.web.form_builder), 233
 build_template() (galaxy.external_services.actions.Template method), 101
 build_template_error_formatters() (in module galaxy.webapps.demo_sequencer.buildapp), 256
 build_template_error_formatters() (in module galaxy.webapps.sigbio.buildapp), 259
 build_template_error_formatters() (in module galaxy.webapps.reports.buildapp), 338
 build_tool_version_select_field() (galaxy.tools.ToolBox method), 101
 build_url() (galaxy.datatypes.display_applications.parameters.DisplayApp method), 101
 bulk_operations (galaxy.model.Sample attribute), 139
 by_user() (galaxy.managers.histories.HistoryManager method), 217
 cache_job_destination() (galaxy.jobs.mapper.JobRunnerMapper method), 115
 CacheableStaticURLParser (class in galaxy.web.framework.middleware.static), 253
 caching_object_store (class in galaxy.objectstore), 156
 calculate_disk_usage() (galaxy.model.User method), 141
 call_tool() (galaxy.tools.Tool method), 72
 camelize() (galaxy.util.inflection.Base method), 219
 camelize() (galaxy.util.inflection.Inflector method), 220
 can_access_dataset() (galaxy.managers.hdas.HDAManager method), 153
 can_access_dataset() (galaxy.security.GalaxyRBACAgent method), 163
 can_access_dataset() (galaxy.security.RBACAgent method), 167
 can_access_library() (galaxy.security.GalaxyRBACAgent method), 163
 can_access_library() (galaxy.security.RBACAgent method), 167
 can_access_library_item() (galaxy.security.GalaxyRBACAgent method), 167

163
`can_access_request_type()` (galaxy.security.GalaxyRBACAgent method), 163
`can_add()` (pkg_resources.Environment method), 357
`can_add_item()` (galaxy.managers.folders.FolderManager method), 151
`can_add_library_item()` (galaxy.security.GalaxyRBACAgent method), 163
`can_add_library_item()` (galaxy.security.RBACAgent method), 167
`can_convert_to()` (galaxy.model.DatasetInstance method), 124
`can_current_user_add_to_library_item()` (galaxy.web.base.controller.UsesLibraryMixinItems method), 238
`can_manage_dataset()` (galaxy.security.GalaxyRBACAgent method), 163
`can_manage_dataset()` (galaxy.security.RBACAgent method), 167
`can_manage_library_item()` (galaxy.security.GalaxyRBACAgent method), 163
`can_manage_library_item()` (galaxy.security.RBACAgent method), 167
`can_modify_library_item()` (galaxy.security.GalaxyRBACAgent method), 163
`can_modify_library_item()` (galaxy.security.RBACAgent method), 167
`can_split()` (galaxy.jobs.JobWrapper method), 108
`can_split()` (galaxy.jobs.TaskWrapper method), 111
`CANCEL` (galaxy.web.framework.openid_manager.OpenIDManager attribute), 244
`cancel()` (galaxy.model.WorkflowInvocation method), 143
`cancel_invocation()` (galaxy.managers.workflows.WorkflowsManager method), 156
`cancel_workflow_invocation()` (galaxy.webapps.galaxy.api.workflows.WorkflowsManager method), 37, 293
`CancelWorkflowEvaluation`, 345
`CaseSensitiveConfigParser` (class in galaxy.eggs), 94
`center()` (galaxy.web.base.controllers.admin.Admin method), 240
`change_adapter()` (galaxy_utils.sequence.fastq.fastqCSSangerRead method), 351
`change_datatype()` (galaxy.datatypes.registry.Registry method), 71
`change_datatype()` (galaxy.model.DatasetInstance method), 124
`change_ownership_for_run()` (galaxy.jobs.JobWrapper method), 108
`change_state()` (galaxy.jobs.JobWrapper method), 108
`change_state()` (galaxy.jobs.TaskWrapper method), 111
`ChangeDatatypeAction` (class in galaxy.jobs.actions.post), 114
`check()` (galaxy.config.Configuration method), 40
`check()` (galaxy.webapps.demo_sequencer.config.Configuration method), 256
`check()` (galaxy.webapps.reports.config.Configuration method), 338
`check_accessible()` (galaxy.managers.folders.FolderManager method), 151
`check_accessible()` (galaxy.managers.hdas.HDAManager method), 153
`check_accessible()` (galaxy.managers.histories.HistoryManager method), 153
`check_accessible()` (galaxy.managers.libraries.LibraryManager method), 154
`check_and_update_param_values()` (galaxy.tools.Tool method), 172
`check_and_update_param_values_helper()` (galaxy.tools.Tool method), 172
`check_and_update_state()` (galaxy.workflow.modules.ToolModule method), 347
`check_and_update_state()` (galaxy.workflow.modules.WorkflowModule method), 348
`check_binary()` (in module galaxy.datatypes.checkers), 46
`check_bz2()` (in module galaxy.datatypes.checkers), 46
`check_conditional()` (galaxy.eggs.GalaxyConfig method), 95
`check_dataset_state()` (galaxy.web.base.controller.UsesHistoryDatasetAssociation method), 237
`check_folder_contents()` (galaxy.security.GalaxyRBACAgent method), 163
`check_for_tool_dependencies()` (galaxy.webapps.galaxy.controllers.admin.AdminGalaxyManager method), 295
`check_for_updates()` (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshedController method), 301
`check_gzip()` (in module galaxy.datatypes.checkers), 46
`check_html()` (in module galaxy.datatypes.checkers), 46
`check_if_output_datasets_deleted()` (galaxy.model.Job method), 132
`check_image()` (in module galaxy.datatypes.checkers), 46
`check_image_type()` (in galaxy.datatypes.util.image_util), 94
`check_interval` (galaxy.jobs.deferred.data_transfer.DataTransfer attribute), 116
`check_interval` (galaxy.model.DeferredJob attribute), 126
`check_job()` (galaxy.jobs.deferred.data_transfer.DataTransfer method), 116
`check_job()` (galaxy.jobs.deferred.manual_data_transfer.ManualDataTransfer method), 117
`check_job()` (galaxy.jobs.deferred.pacific_biosciences_smrt_portal.SMRTPortal method), 117

- check_limits() (galaxy.jobs.JobWrapper method), 108
- check_limits() (galaxy.jobs.runners.AsynchronousJobState method), 118
- check_manageable() (galaxy.managers.folders.FolderManager method), 151
- check_newlines() (in module galaxy.datatypes.sniff), 78
- check_ownership() (galaxy.managers.hdas.HDAManager method), 153
- check_ownership() (galaxy.managers.histories.HistoryManager method), 153
- check_param() (in module galaxy.tools.parameters), 185
- check_param_from_incoming() (in module galaxy.tools), 177
- check_password() (galaxy.model.User method), 141
- check_pid() (galaxy.jobs.runners.lwr.LwrJobRunner method), 121
- check_security() (galaxy.managers.workflows.WorkflowsManager method), 156
- check_tool_output() (galaxy.jobs.JobWrapper method), 108
- check_user_can_add_to_library_item() (galaxy.web.base.controller.UsesLibraryMixinItem method), 238
- check_version_conflict() (pkg_resources.Distribution method), 360
- check_watched_item() (galaxy.jobs.runners.AsynchronousJobRunner method), 117
- check_watched_item() (galaxy.jobs.runners.lwr.LwrJobRunner method), 121
- check_watched_items() (galaxy.jobs.runners.AsynchronousJobRunner method), 117
- check_watched_items() (galaxy.jobs.runners.cli.ShellJobRunner method), 119
- check_watched_items() (galaxy.jobs.runners.condor.CondorJobRunner method), 119
- check_watched_items() (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 120
- check_workflow_compatible() (galaxy.tools.Tool method), 172
- check_zip() (in module galaxy.datatypes.checkers), 46
- CheckboxField (class in galaxy.web.form_builder), 229
- checkComments() (galaxy.visualization.data_providers.phyloviz.nexus_parser.NexusParser method), 228
- chop_block_by_region() (in module galaxy.tools.util.maf_utilities), 210
- ChromatinInteractions (class in galaxy.datatypes.interval), 61
- ChromInfo (class in galaxy.datatypes.chrominfo), 46
- chroms() (galaxy.visualization.genomes.Genomes method), 225
- chunk64_dataprovider() (galaxy.datatypes.data.Data method), 47
- chunk_dataprovider() (galaxy.datatypes.data.Data method), 47
- CHUNKABLE (galaxy.datatypes.data.Data attribute), 47
- CHUNKABLE (galaxy.datatypes.tabular.Tabular attribute), 83
- circos() (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 333
- CisML (class in galaxy.datatypes.xml), 84
- CitationCollection (class in galaxy.managers.citations), 149
- citations() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 15, 271
- citations() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 309
- citations_for_tool() (galaxy.managers.citations.CitationsManager method), 149
- citations_for_tool_ids() (galaxy.managers.citations.CitationsManager method), 149
- CitationsManager (class in galaxy.managers.citations), 149
- class_type (in module galaxy.external_services.service), 103
- classify() (galaxy.util.inflection.Base method), 219
- classify() (galaxy.util.inflection.Inflector method), 220
- clean_field_contents() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 236
- cleanNewickString() (galaxy.visualization.data_providers.phyloviz.newickparser.NewickParser method), 236
- cleanTag() (galaxy.visualization.data_providers.phyloviz.phyloxmlparser.PhyloxmlParser method), 228
- cleanup() (galaxy.jobs.JobWrapper method), 108
- cleanup() (galaxy.jobs.runners.AsynchronousJobState method), 118
- cleanup() (galaxy.jobs.TaskWrapper method), 111
- cleanup_after_job() (galaxy.tools.imp_exp.JobExportHistoryArchiveWrapper method), 184
- cleanup_after_job() (galaxy.tools.imp_exp.JobImportHistoryArchiveWrapper method), 184
- cleanup_before_exit() (in module galaxy.datatypes.converters.sam_to_bam), 90
- cleanup_external_metadata() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 67
- cleanup_from_JSON_dict_filename() (galaxy.datatypes.metadata.MetadataTempFile class method), 69
- cleanup_resources() (pkg_resources.ResourceManager method), 359
- cleanup_unused_precreated_datasets() (in module galaxy.tools.actions.upload_common), 180
- clear() (galaxy.model.ImplicitlyConvertedDatasetAssociation method), 131
- clear() (galaxy.util.lrucache.LRUCache method), 222
- clear() (galaxy.util.odict.odict method), 223

clear_associated_files() (galaxy.model.DatasetInstance method), 124
 clear_associated_files() (galaxy.model.HistoryDatasetAssociation method), 129
 clear_associated_files() (galaxy.model.LibraryDatasetDatasetAssociation method), 135
 clear_display_apps() (galaxy.datatypes.data.Data method), 47
 clear_history() (galaxy.webapps.galaxy.controllers.root.RootController method), 325
 clone() (galaxy_utils.sequence.sequence.SequencingRead method), 354
 clone() (pkg_resources.Distribution method), 360
 close() (galaxy.datatypes.converters.interval_to_coverage.CoverageWriter method), 88
 close() (galaxy.tools.util.maf_utilities.TempFileHandler method), 210
 close() (galaxy_utils.sequence.fasta.fastaNamedReader method), 350
 close() (galaxy_utils.sequence.fasta.fastaReader method), 350
 close() (galaxy_utils.sequence.fasta.fastaWriter method), 350
 close() (galaxy_utils.sequence.fastq.fastqFakeFastaScoreReader method), 351
 close() (galaxy_utils.sequence.fastq.fastqNamedReader method), 352
 close() (galaxy_utils.sequence.fastq.fastqReader method), 352
 close() (galaxy_utils.sequence.fastq.fastqWriter method), 353
 cls (in module galaxy.tools.data), 183
 col (galaxy.webapps.galaxy.controllers.requests_admin.AdminRequestsGrid attribute), 321
 collect_associated_files() (galaxy.tools.Tool method), 172
 collect_child_datasets() (galaxy.tools.Tool method), 172
 collect_input_dataset_collections() (galaxy.tools.actions.DefaultToolAction method), 179
 collect_input_datasets() (galaxy.tools.actions.DefaultToolAction method), 179
 collect_inputs() (galaxy.tools.test.TestCollectionDefinition method), 178
 collect_primary_datasets() (galaxy.tools.Tool method), 172
 collection_type (galaxy.tools.parameters.basic.DataCollectionDefinition attribute), 187
 collection_type (galaxy.workflow.modules.InputDataCollectionDefinition attribute), 345
 color_dict (galaxy_utils.sequence.transform.ColorSpaceConverter attribute), 354
 color_space_converter (galaxy_utils.sequence.sequence.SequencingRead attribute), 354
 color_to_base_dict (galaxy_utils.sequence.transform.ColorSpaceConverter attribute), 354
 ColorSpaceConverter (class in galaxy_utils.sequence.transform), 354
 ColumnDataProvider (galaxy.datatypes.binary.Bam method), 42
 column_dataprovider() (galaxy.datatypes.tabular.Sam method), 82
 ColumnDataProvider() (galaxy.datatypes.tabular.Tabular method), 83
 column_names (galaxy.datatypes.interval.ChromatinInteractions attribute), 61
 column_names (galaxy.datatypes.interval.ENCODEPeak attribute), 62
 column_names (galaxy.datatypes.interval.Gff attribute), 62
 column_names (galaxy.datatypes.interval.Gff3 attribute), 63
 column_names (galaxy.datatypes.interval.Gtf attribute), 64
 column_names (galaxy.datatypes.tabular.Vcf attribute), 84
 column_spec_to_index() (galaxy.tools.parameters.dynamic_options.Dynamic method), 197
 ColumnDataProvider (class in galaxy.visualization.data_providers.basic), 226
 ColumnListParameter (class in galaxy.tools.parameters.basic), 186
 ColumnParameter (class in galaxy.datatypes.metadata), 66
 ColumnReplaceFilter (class in galaxy.tools.parameters.output), 201
 columns (galaxy.web.framework.helpers.grids.Grid attribute), 247
 columns (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 296
 columns (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 297
 columns (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 298
 columns (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 299
 columns (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 301
 ColumnParameter (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociation attribute), 305
 ColumnModule (galaxy.webapps.galaxy.controllers.external_service.ExternalService attribute), 307
 columns (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 308
 ColumnModule (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid attribute), 309

columns (galaxy.webapps.galaxy.controllers.history.HistoryListGridSetAction (class in galaxy.jobs.actions.post), 114
attribute), 311 ColumnStripFilter (class in
columns (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid galaxy.tools.parameters.output), 202
attribute), 312 ColumnTypesParameter (class in
columns (galaxy.webapps.galaxy.controllers.library.LibraryListGrid galaxy.datatypes.metadata), 66
attribute), 313 combine() (galaxy_utils.sequence.fastq.fastqCombiner
columns (galaxy.webapps.galaxy.controllers.library_admin.LibraryListGrid), 351
attribute), 314 commaify() (in module galaxy.util), 213
columns (galaxy.webapps.galaxy.controllers.page.HistoryDataAccessControlSelectionGrid (class in
attribute), 316 galaxy.webapps.demo_sequencer.controllers.common),
columns (galaxy.webapps.galaxy.controllers.page.HistorySelectionGrid), 257
attribute), 317 CommunityRatingColumn (class in
columns (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid galaxy.web.framework.helpers.grids), 245
attribute), 317 CommunityTagHandler (class in
columns (galaxy.webapps.galaxy.controllers.page.PageListGrid galaxy.tags.tag_handler), 168
attribute), 319 CommunityTagsColumn (class in
columns (galaxy.webapps.galaxy.controllers.page.PageSelectionGrid galaxy.web.framework.helpers.grids), 246
attribute), 319 compare_endswith() (in module
columns (galaxy.webapps.galaxy.controllers.page.VisualizationSelectionGrid galaxy.tools.parameters.output), 204
attribute), 319 compare_eq() (in module
columns (galaxy.webapps.galaxy.controllers.page.WorkflowSelectionGrid galaxy.tools.parameters.output), 204
attribute), 319 compare_gt() (in module galaxy.tools.parameters.output),
columns (galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid
attribute), 321 compare_gte() (in module
columns (galaxy.webapps.galaxy.controllers.requests_admin.AdminRequestsGrid galaxy.tools.parameters.output), 204
attribute), 321 compare_in() (in module galaxy.tools.parameters.output),
columns (galaxy.webapps.galaxy.controllers.requests_admin.DataTransferGrid
attribute), 322 compare_lt() (in module galaxy.tools.parameters.output),
columns (galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid
attribute), 324 compare_lte() (in module
columns (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid galaxy.tools.parameters.output), 204
attribute), 329 compare_neq() (in module
columns (galaxy.webapps.galaxy.controllers.visualization.HistoryDataAccessControlGrid galaxy.tools.parameters.output), 205
attribute), 331 compare_re_search() (in module
columns (galaxy.webapps.galaxy.controllers.visualization.HistorySelectionGrid galaxy.tools.parameters.output), 205
attribute), 331 compare_startswith() (in module
columns (galaxy.webapps.galaxy.controllers.visualization.LibrarySelectionGrid galaxy.tools.parameters.output), 205
attribute), 332 compare_urls() (in module galaxy.util), 213
columns (galaxy.webapps.galaxy.controllers.visualization.TrackerSelectionGrid galaxy.model.custom_types.JSONType
attribute), 333 method), 145
columns (galaxy.webapps.galaxy.controllers.visualization.VisualizationAllPublishedGrid (in module pkg_resources), 362
attribute), 333 complement() (galaxy_utils.sequence.fastq.fastqCSSangerRead
columns (galaxy.webapps.galaxy.controllers.visualization.VisualizationMethodGrid), 351
attribute), 335 complement() (galaxy_utils.sequence.sequence.SequencingRead
columns (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAllPublishedGrid
attribute), 335 component_overlaps_region() (in module
columns (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowGrid galaxy.util.maf_utilities), 210
attribute), 336 components_are_associated()
columns (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid galaxy.security.RBACAgent method), 167
attribute), 340 composite_files (galaxy.datatypes.data.Data attribute), 47
columns (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid galaxy.datatypes.assembly.Velvet
attribute), 342 attribute), 42
columns (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid galaxy.datatypes.data.Data attribute), 47
attribute), 344

- [composite_type \(galaxy.datatypes.genetics.RexpBase attribute\), 53](#)
[composite_type \(galaxy.datatypes.genetics.Rgenetics attribute\), 54](#)
[composite_type \(galaxy.datatypes.ngsindex.BowtieIndex attribute\), 70](#)
[compute_fasta_length\(\) \(in module galaxy.datatypes.converters.fasta_to_len\), 86](#)
[compute_outputs\(\) \(galaxy.jobs.JobWrapper method\), 108](#)
[compute_runtime_state\(\) \(galaxy.workflow.modules.SimpleWorkflowModule method\), 346](#)
[compute_runtime_state\(\) \(galaxy.workflow.modules.ToolModule method\), 347](#)
[compute_runtime_state\(\) \(galaxy.workflow.modules.WorkflowModule method\), 348](#)
[ComputeEnvironment \(class in galaxy.jobs\), 106](#)
[cond_plural\(\) \(galaxy.util.inflection.Base method\), 219](#)
[cond_plural\(\) \(galaxy.util.inflection.Inflector method\), 220](#)
[Conditional \(class in galaxy.tools.parameters.grouping\), 199](#)
[ConditionalWhen \(class in galaxy.tools.parameters.grouping\), 199](#)
[CondorJobRunner \(class in galaxy.jobs.runners.condor\), 119](#)
[config_directory\(\) \(galaxy.jobs.ComputeEnvironment method\), 106](#)
[config_directory\(\) \(galaxy.jobs.SimpleComputeEnvironmentcontroller_name method\), 110](#)
[config_eggs \(galaxy.eggs.Crate attribute\), 95](#)
[config_exception\(\) \(in module galaxy.jobs\), 111](#)
[config_file \(galaxy.eggs.Crate attribute\), 95](#)
[config_missing \(galaxy.eggs.Crate attribute\), 95](#)
[config_names \(galaxy.eggs.Crate attribute\), 95](#)
[ConfigDoesNotAllowException, 97](#)
[Configuration \(class in galaxy.config\), 40](#)
[Configuration \(class in galaxy.webapps.demo_sequencer.config\), 256](#)
[Configuration \(class in galaxy.webapps.reports.config\), 338](#)
[ConfigurationController \(class in galaxy.webapps.galaxy.api.configuration\), 8, 264](#)
[ConfigurationError, 40, 71, 256, 338](#)
[configure_and_load\(\) \(galaxy.tools.data.TabularToolDataTable method\), 181](#)
[configure_logging\(\) \(in module galaxy.config\), 40](#)
[configure_logging\(\) \(in module galaxy.webapps.demo_sequencer.config\), 256](#)
[configure_logging\(\) \(in module galaxy.webapps.reports.config\), 339](#)
[configure_menu\(\) \(galaxy.webapps.galaxy.controllers.workflow.WorkflowC method\), 336](#)
[ConfiguresGalaxyMixin \(class in galaxy.config\), 40](#)
[Conflict, 97](#)
[consume_read\(\) \(galaxy_utils.sequence.fastq.fastqAggregator method\), 350](#)
[content \(galaxy.external_services.actions.ExternalServiceResult attribute\), 100](#)
[content \(galaxy.external_services.actions.ExternalServiceValueResult attribute\), 100](#)
[content \(galaxy.external_services.actions.ExternalServiceWebAPIActionRe attribute\), 101](#)
[contents_iter\(\) \(galaxy.model.History method\), 129](#)
[controller_name \(galaxy.webapps.galaxy.api.annotations.HistoryAnnotation attribute\), 8, 264](#)
[controller_name \(galaxy.webapps.galaxy.api.annotations.HistoryContentAn attribute\), 8, 264](#)
[controller_name \(galaxy.webapps.galaxy.api.annotations.WorkflowAnnotati attribute\), 8, 264](#)
[controller_name \(galaxy.webapps.galaxy.api.extended_metadata.HistoryDat attribute\), 10, 266](#)
[controller_name \(galaxy.webapps.galaxy.api.extended_metadata.LibraryDat attribute\), 10, 266](#)
[controller_name \(galaxy.webapps.galaxy.api.item_tags.HistoryContentTags attribute\), 19, 275](#)
[controller_name \(galaxy.webapps.galaxy.api.item_tags.HistoryTagsControll attribute\), 19, 275](#)
[controller_name \(galaxy.webapps.galaxy.api.item_tags.WorkflowTagsContr attribute\), 20, 275](#)
[controller_name \(galaxy.webapps.galaxy.api.provenance.HDAProvenanceC attribute\), 31, 287](#)
[controller_name \(galaxy.webapps.galaxy.api.provenance.LDDAProvenance attribute\), 31, 287](#)
[ControllerUnavailable, 235](#)
[conversion_messages \(galaxy.model.Dataset attribute\), 122](#)
[conversion_messages \(galaxy.model.DatasetInstance attribute\), 124](#)
[convert_base_to_color_space\(\) \(galaxy_utils.sequence.fastq.fastqSequencingRead class method\), 352](#)
[convert_bed_coords_to_gff\(\) \(in module galaxy.datatypes.util.gff_util\), 94](#)
[convert_bytes\(\) \(in module galaxy.objectstore\), 159](#)
[convert_color_to_base_space\(\) \(galaxy_utils.sequence.fastq.fastqSequencingRead class method\), 352](#)
[convert_dataset\(\) \(galaxy.datatypes.data.Data method\), 47](#)

`convert_dataset()` (galaxy.model.DatasetInstance method), 124
`convert_gff_coords_to_bed()` (in module galaxy.datatypes.util.gff_util), 94
`convert_legacy_destinations()` (galaxy.jobs.JobConfiguration method), 106
`convert_newlines()` (in module galaxy.datatypes.sniff), 78
`convert_newlines_sep2tabs()` (in module galaxy.datatypes.sniff), 78
`convert_permitted_action_strings()` (galaxy.security.RBACAgent method), 167
`convert_read_to_format()` (galaxy_utils.sequence.fastq.fastqSequencingRead method), 352
`convert_score_phred_to_solexa()` (galaxy_utils.sequence.fastq.fastqSequencingRead class method), 352
`convert_score_solexa_to_phred()` (galaxy_utils.sequence.fastq.fastqSequencingRead class method), 352
`converter_safe()` (galaxy.tools.parameters.basic.DataToolParameter method), 187
`ConverterDependencyException`, 122
`cookies` (galaxy.web.framework.base.Request attribute), 243
`copy()` (galaxy.datatypes.util.gff_util.GFFFeature method), 93
`copy()` (galaxy.datatypes.util.gff_util.GFFInterval method), 93
`copy()` (galaxy.managers.collections.DatasetCollectionManager method), 149
`copy()` (galaxy.model.DatasetCollection method), 123
`copy()` (galaxy.model.History method), 129
`copy()` (galaxy.model.HistoryDatasetAssociation method), 129
`copy()` (galaxy.model.HistoryDatasetCollectionAssociation method), 130
`copy()` (galaxy.model.ItemTagAssociation method), 131
`copy()` (galaxy.model.LibraryDatasetDatasetAssociation method), 135
`copy()` (galaxy.model.Visualization method), 143
`copy()` (galaxy.model.VisualizationRevision method), 143
`copy()` (galaxy.tools.DefaultToolState method), 171
`copy()` (galaxy.tools.ToolSection method), 177
`copy()` (galaxy.util.odict.odict method), 223
`copy()` (galaxy.webapps.galaxy.controllers.history.HistoryController method), 309
`copy()` (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 333
`copy()` (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
`copy_build_script()` (galaxy.eggs.scramble.ScrambleEgg method), 96
`copy_dataset_permissions()` (galaxy.security.GalaxyRBACAgent method), 163
`copy_datasets()` (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
`copy_hda_to_library_folder()` (galaxy.web.base.controller.UsesLibraryMixinItems method), 238
`copy_item_annotation()` (galaxy.model.item_attrs.UsesAnnotations method), 146
`copy_library_permissions()` (galaxy.security.GalaxyRBACAgent method), 164
`copy_safe_peek` (galaxy.datatypes.data.Data attribute), 47
`copy_safe_peek` (galaxy.datatypes.images.Gmaj attribute), 56
`copy_safe_peek` (galaxy.datatypes.images.Laj attribute), 57
`copy_tags_from()` (galaxy.model.History method), 129
`copy_tags_from()` (galaxy.model.StoredWorkflow method), 139
`copy_to_collection()` (galaxy.model.DatasetCollectionElement method), 123
`copy_value()` (galaxy.model.custom_types.JSONType method), 145
`count_data_lines()` (galaxy.datatypes.data.Text method), 50
`CoverageWriter` (class in galaxy.datatypes.converters.interval_to_coverage), 87
`Crate` (class in galaxy.eggs), 94
`create()` (galaxy.managers.collections.DatasetCollectionManager method), 149
`create()` (galaxy.managers.folders.FolderManager method), 151
`create()` (galaxy.managers.libraries.LibraryManager method), 154
`create()` (galaxy.objectstore.DiskObjectStore method), 157
`create()` (galaxy.objectstore.DistributedObjectStore method), 157
`create()` (galaxy.objectstore.HierarchicalObjectStore method), 157
`create()` (galaxy.objectstore.NestedObjectStore method), 157
`create()` (galaxy.objectstore.ObjectStore method), 158
`create()` (galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController method), 7, 263
`create()` (galaxy.webapps.galaxy.api.dataset_collections.DatasetCollectionsController method), 9, 265
`create()` (galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController method), 9, 265
`create()` (galaxy.webapps.galaxy.api.folder_contents.FolderContentsController method), 10, 266

[create\(\)](#) (galaxy.webapps.galaxy.api.folders.FoldersController method), 11, 267
[create_applet_tag_peek\(\)](#) (in module galaxy.datatypes.images), 59
[create\(\)](#) (galaxy.webapps.galaxy.api.forms.FormDefinitionAPIController method), 13, 268
[create_drive\(\)](#) (in module galaxy.tools.imp_exp.export_history), 184
[create\(\)](#) (galaxy.webapps.galaxy.api.groups.GroupAPIController method), 14, 270
[create_display\(\)](#) (galaxy.webapps.galaxy.controllers.ucsc_proxy.UCSCProxy method), 327
[create\(\)](#) (galaxy.webapps.galaxy.api.histories.HistoriesController method), 15, 271
[create_extended_metadata\(\)](#) (galaxy.web.base.controller.UsesExtendedMetadataMixin method), 236
[create\(\)](#) (galaxy.webapps.galaxy.api.history_contents.HistoryContentsController method), 17, 272
[create_external_service\(\)](#) (galaxy.webapps.galaxy.controllers.external_service.ExternalService method), 306
[create\(\)](#) (galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController method), 19, 275
[create_folder\(\)](#) (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 314
[create\(\)](#) (galaxy.webapps.galaxy.api.job_files.JobFilesAPIController method), 20, 275
[create_form_definition\(\)](#) (galaxy.webapps.galaxy.controllers.forms.Forms method), 307
[create\(\)](#) (galaxy.webapps.galaxy.api.jobs.JobController method), 21, 276
[create_group\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 240
[create\(\)](#) (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 25, 280
[create_contents_controller\(\)](#) (galaxy.web.base.controller.SharableMixin method), 236
[create\(\)](#) (galaxy.webapps.galaxy.api.library_contents.LibraryContentsController method), 27, 283
[create_job\(\)](#) (galaxy.jobs.deferred.data_transfer.DataTransfer method), 116
[create\(\)](#) (galaxy.webapps.galaxy.api.metrics.MetricsController method), 29, 285
[create_manual_data_transfer_job\(\)](#) (galaxy.jobs.deferred.manual_data_transfer.ManualDataTransfer method), 117
[create\(\)](#) (galaxy.webapps.galaxy.api.page_revisions.PageRevisionsController method), 30, 285
[create_job\(\)](#) (galaxy.jobs.deferred.pacific_biosciences_smrt_portal.SMRTPortal method), 117
[create\(\)](#) (galaxy.webapps.galaxy.api.pages.PagesController method), 30, 286
[create_controller\(\)](#) (in module galaxy.tools.actions.upload_common), 180
[create\(\)](#) (galaxy.webapps.galaxy.api.provenance.BaseProvenanceController method), 31, 287
[create_library\(\)](#) (galaxy.webapps.galaxy.controllers.library_admin.LibraryAdmin method), 313
[create\(\)](#) (galaxy.webapps.galaxy.api.quotas.QuotaAPIController method), 32, 287
[create_controller\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 309
[create\(\)](#) (galaxy.webapps.galaxy.api.request_types.RequestTypesAPIController method), 32, 287
[create_new_user\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 240
[create\(\)](#) (galaxy.webapps.galaxy.api.roles.RoleAPIController method), 33, 288
[create_object_in_session\(\)](#) (in module galaxy.objectstore), 159
[create\(\)](#) (galaxy.webapps.galaxy.api.search.SearchController method), 33, 289
[create_or_verify_database\(\)](#) (in module galaxy.model.migrate.check), 147
[create\(\)](#) (galaxy.webapps.galaxy.api.users.UserAPIController method), 37, 292
[create_controller\(\)](#) (in module galaxy.tools.actions.upload_common), 180
[create\(\)](#) (galaxy.webapps.galaxy.api.visualizations.VisualizationsController method), 37, 292
[create_user_role\(\)](#) (galaxy.security.GalaxyRBACAgent method), 164
[create\(\)](#) (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 38, 293
[create_private_user_role\(\)](#) (galaxy.security.RBACAgent method), 167
[create\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[create_quota\(\)](#) (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 295
[create\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), 328
[create_request\(\)](#) (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 323
[create\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 333
[create_request_type\(\)](#) (galaxy.webapps.galaxy.controllers.request_type.RequestType method), 320
[create_api_key\(\)](#) (galaxy.managers.api_keys.ApiKeyManager method), 148
[create_role\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 240
[create_api_key\(\)](#) (galaxy.web.base.controller.CreatesApiKeysMixin method), 235

create_user() (galaxy.web.base.controller.CreatesUsersMixin method), 235
 create_visualization() (galaxy.web.base.controller.UsesVisualizationMixin method), 239
 CreatesApiKeysMixin (class in galaxy.web.base.controller), 235
 CreatesUsersMixin (class in galaxy.web.base.controller), 235
 creating_job (galaxy.model.DatasetInstance attribute), 124
 csFasta (class in galaxy.datatypes.sequence), 77
 css() (in module galaxy.web.framework.helpers), 245
 cur_filter_pref_name (galaxy.web.framework.helpers.grids.Grid attribute), 247
 cur_sort_key_pref_name (galaxy.web.framework.helpers.grids.Grid attribute), 247
 current_history_json() (galaxy.webapps.galaxy.controllers.history_history_controller method), 309
 cursor_execute() (galaxy.model.orm.logging_connection_provider method), 147
 cursor_execute() (galaxy.model.orm.logging_connection_provider method), 147
 CustomTrack (class in galaxy.datatypes.interval), 61
 cut_and_decode() (galaxy.managers.folders.FolderManager method), 151
 cut_the_prefix() (galaxy.managers.folders.FolderManager method), 152
 CycleError, 224
D
 Data (class in galaxy.datatypes.data), 46
 data_source_redirect() (galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner method), 327
 data_sources (galaxy.datatypes.binary.Bam attribute), 42
 data_sources (galaxy.datatypes.binary.BigBed attribute), 43
 data_sources (galaxy.datatypes.binary.BigWig attribute), 43
 data_sources (galaxy.datatypes.data.Data attribute), 47
 data_sources (galaxy.datatypes.interval.Bed attribute), 60
 data_sources (galaxy.datatypes.interval.BedGraph attribute), 61
 data_sources (galaxy.datatypes.interval.ChromatinInteractions attribute), 61
 data_sources (galaxy.datatypes.interval.ENCODEPeak attribute), 62
 data_sources (galaxy.datatypes.interval.Gff attribute), 62
 data_sources (galaxy.datatypes.interval.Interval attribute), 64
 data_sources (galaxy.datatypes.interval.Wiggle attribute), 65
 data_sources (galaxy.datatypes.tabular.Pileup attribute), 81
 data_sources (galaxy.datatypes.tabular.Sam attribute), 82
 data_sources (galaxy.datatypes.tabular.Vcf attribute), 84
 data_sources (in module galaxy.sample_tracking.data_transfer), 162
 data_transfer_protocol (galaxy.model.ExternalService attribute), 127
 DataCollectionToolParameter (class in galaxy.tools.parameters.basic), 187
 DataDestinationTool (class in galaxy.tools), 170
 DataManagerHistoryAssociation (class in galaxy.model), 122
 DataManagerJobAssociation (class in galaxy.model), 122
 DataManagerTool (class in galaxy.tools), 170
 DataMeta (class in galaxy.datatypes.data), 49
 DataMetaFilter (class in galaxy.tools.parameters.dynamic_options), 197
 data_provider (galaxy.datatypes.data.Data method), 47
 dataproviders (galaxy.datatypes.binary.Bam attribute), 42
 dataproviders (galaxy.datatypes.binary.SQLite attribute), 45
 dataproviders (galaxy.datatypes.data.Data attribute), 47
 dataproviders (galaxy.datatypes.data.Text attribute), 50
 dataproviders (galaxy.datatypes.interval.Gff attribute), 62
 dataproviders (galaxy.datatypes.interval.Interval attribute), 64
 dataproviders (galaxy.datatypes.interval.Wiggle attribute), 65
 dataproviders (galaxy.datatypes.tabular.Pileup attribute), 81
 dataproviders (galaxy.datatypes.tabular.Sam attribute), 82
 dataproviders (galaxy.datatypes.tabular.Tabular attribute), 82
 dataproviders (galaxy.datatypes.tabular.Vcf attribute), 84
 dataproviders (galaxy.datatypes.xml.GenericXml attribute), 85
 Dataset (class in galaxy.model), 122
 dataset (galaxy.model.DatasetCollectionElement attribute), 123
 dataset (galaxy.model.JobExternalOutputMetadata attribute), 134
 dataset_access_mapping() (galaxy.security.GalaxyRBACAgent method), 164
 dataset_column_dataprovider() (galaxy.datatypes.tabular.Sam method), 82
 dataset_column_dataprovider() (galaxy.datatypes.tabular.Tabular method), 83
 dataset_content_needs_grooming() (galaxy.datatypes.binary.Bam method), 43
 dataset_content_needs_grooming() (galaxy.datatypes.data.Data method), 47

[dataset_datatype_re](#) (galaxy.jobs.deferred.data_transfer.DataTransfer attribute), 116
[dataset_detail\(\)](#) (galaxy.webapps.galaxy.controllers.mobile.Mobile attribute), 316
[dataset_dict_dataprovider\(\)](#) (galaxy.datatypes.tabular.Sam method), 82
[dataset_dict_dataprovider\(\)](#) (galaxy.datatypes.tabular.Tabular method), 83
[dataset_info\(\)](#) (galaxy.webapps.reports.controllers.system.System attribute), 342
[dataset_instance](#) (galaxy.model.DatasetCollectionElement attribute), 123
[dataset_instances](#) (galaxy.model.DatasetCollection attribute), 123
[dataset_is_private_to_user\(\)](#) (galaxy.security.GalaxyRBACAgent method), 164
[dataset_is_public\(\)](#) (galaxy.security.GalaxyRBACAgent method), 164
[dataset_is_public\(\)](#) (galaxy.security.RBACAgent method), 167
[dataset_is_unrestricted\(\)](#) (galaxy.security.GalaxyRBACAgent method), 164
[dataset_make_primary\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController method), 325
[dataset_name_re](#) (galaxy.jobs.deferred.data_transfer.DataTransfer attribute), 116
[dataset_peek\(\)](#) (galaxy.webapps.galaxy.controllers.mobile.Mobile method), 316
[dataset_permission_map_for_access\(\)](#) (galaxy.security.GalaxyRBACAgent method), 164
[dataset_transfer_status_updates\(\)](#) (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 323
[dataset_type](#) (galaxy.visualization.data_providers.phyloviz.PhylovizDataProvider attribute), 227
[DatasetCollection](#) (class in galaxy.model), 123
[DatasetCollectionElement](#) (class in galaxy.model), 123
[DatasetCollectionInstance](#) (class in galaxy.model), 124
[DatasetCollectionManager](#) (class in galaxy.managers.collections), 149
[DatasetCollectionsController](#) (class in galaxy.webapps.galaxy.api.dataset_collections), 9, 265
[DatasetInstance](#) (class in galaxy.model), 124
[DatasetInterface](#) (class in galaxy.webapps.galaxy.controllers.dataset), 304
[DatasetOkValidator](#) (class in galaxy.tools.parameters.validation), 206
[DatasetPermissions](#) (class in galaxy.model), 126
[DataTransfer](#) (class in galaxy.webapps.galaxy.controllers.visualization.HistorySelection attribute), 331
[DataTransfer](#) (class in galaxy.webapps.galaxy.controllers.visualization.LibrarySelection attribute), 332
[datasets_are_public\(\)](#) (galaxy.security.GalaxyRBACAgent method), 164
[datasets_for_lldas\(\)](#) (in galaxy.webapps.galaxy.controllers.library_common module), 315
[datasets_param](#) (galaxy.webapps.galaxy.controllers.visualization.HistorySelection attribute), 332
[datasets_param](#) (galaxy.webapps.galaxy.controllers.visualization.LibrarySelection attribute), 332
[DatasetTagAssociation](#) (class in galaxy.model), 126
[DatasetToValidationErrorAssociation](#) (class in galaxy.model), 126
[DataSourceTool](#) (class in galaxy.tools), 170
[DataToolParameter](#) (class in galaxy.tools.parameters.basic), 187
[DataTransfer](#) (class in galaxy.jobs.deferred.data_transfer), 116
[DataTransferFactory](#) (class in galaxy.sample_tracking.data_transfer), 161
[DataTransferGrid](#) (class in galaxy.webapps.galaxy.controllers.requests_admin), 322
[DataTransferGrid.ExternalServiceColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_admin), 322
[DataTransferGrid.NameColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_admin), 322
[DataTransferGrid.SizeColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_admin), 322
[DataTransferGrid.StatusColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_admin), 322
[datatx_grid](#) (galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin attribute), 323
[Datatype](#) (class in galaxy.web.base.controller), 235
[datatype](#) (galaxy.model.DatasetInstance attribute), 124
[DatatypeIsInstanceToolOutputActionConditionalWhen](#) (class in galaxy.tools.parameters.output), 202
[DatatypesController](#) (class in galaxy.webapps.galaxy.api.datatypes), 9, 265
[DateTimeColumn](#) (class in galaxy.web.framework.helpers.grids), 246
[db_builds](#) (galaxy.managers.context.ProvidesHistoryContext attribute), 151
[db_dataset_for\(\)](#) (galaxy.jobs.deferred.FakeTransfer method), 116

db_dataset_for() (galaxy.managers.context.ProvidesHistoryContext method), 151
 db_next_hid() (in module galaxy.model.mapping), 146
 dbkey (galaxy.model.DatasetInstance attribute), 124
 DbKeyColumn (class in galaxy.webapps.galaxy.controllers.visualization), 329
 DBKeyParameter (class in galaxy.datatypes.metadata), 66
 DbKeyPlaceholderColumn (class in galaxy.webapps.galaxy.controllers.visualization), 330
 dbkeys() (galaxy.webapps.galaxy.controllers.user.User method), 328
 deactivate_or_uninstall_repository() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 301
 debugging (galaxy.webapps.galaxy.api.metrics.MetricsController attribute), 30, 285
 declare_namespace() (in module pkg_resources), 357
 decode() (galaxy.tools.DefaultToolState method), 171
 decode_dataset_user() (in module galaxy.datatypes.display_applications.util), 92
 decode_dbkey() (in module galaxy.visualization.genomes), 225
 decode_folder_id() (galaxy.managers.folders.FolderManager method), 152
 decode_guid() (galaxy.web.security.SecurityHelper method), 254
 decode_id() (galaxy.web.security.SecurityHelper method), 254
 decode_runtime_state() (galaxy.workflow.modules.SimpleWorkflowModule method), 346
 default() (galaxy.webapps.galaxy.controllers.async.ASync method), 303
 default() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
 default() (galaxy.webapps.galaxy.controllers.root.RootController method), 325
 default() (galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner method), 327
 default_collection_type (galaxy.workflow.modules.InputDataCollectionModule attribute), 345
 default_compute_environment() (galaxy.jobs.JobWrapper method), 108
 default_exit_code_file() (galaxy.jobs.runners.JobState static method), 119
 default_filter (galaxy.web.framework.helpers.grids.Grid attribute), 247
 default_filter (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 299
 default_filter (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid attribute), 305
 default_filter (galaxy.webapps.galaxy.controllers.external_service.ExternalService attribute), 307
 default_filter (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 308
 default_filter (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid attribute), 309
 default_filter (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 311
 default_filter (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid attribute), 312
 default_filter (galaxy.webapps.galaxy.controllers.library.LibraryListGrid attribute), 313
 default_filter (galaxy.webapps.galaxy.controllers.library_admin.LibraryListGrid attribute), 314
 default_filter (galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid attribute), 317
 default_filter (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid attribute), 317
 default_filter (galaxy.webapps.galaxy.controllers.page.PageListGrid attribute), 319
 default_filter (galaxy.webapps.galaxy.controllers.request_type.RequestType attribute), 321
 default_filter (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon attribute), 324
 default_filter (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid attribute), 329
 default_filter (galaxy.webapps.galaxy.controllers.visualization.HistoryDatasetGrid attribute), 331
 default_filter (galaxy.webapps.galaxy.controllers.visualization.VisualizationGrid attribute), 333
 default_filter (galaxy.webapps.galaxy.controllers.visualization.VisualizationListGrid attribute), 335
 default_filter (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowGrid attribute), 335
 default_filter (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid attribute), 336
 default_filter (galaxy.webapps.galaxy.controllers.workflow.WorkflowGrid attribute), 340
 default_filter (galaxy.webapps.galaxy.controllers.workflow.WorkflowListGrid attribute), 342
 default_filter (galaxy.webapps.galaxy.controllers.workflow.WorkflowListGrid attribute), 344
 DEFAULT_INVALID_CHAR (galaxy.tools.parameters.sanitize.ToolParameterSanitizer attribute), 205
 default_job_file() (galaxy.jobs.runners.JobState static method), 119
 default_job_tool_configuration (galaxy.jobs.JobConfiguration attribute), 119
 DEFAULT_MAX_OPEN_FILES (galaxy.utilities.TempFileHandler attribute), 210
 default_name (galaxy.model.History attribute), 129

DelayedWorkflowEvaluation, 345

delete() (galaxy.managers.collections.DatasetCollectionManager method), 150

delete() (galaxy.managers.folders.FolderManager method), 152

delete() (galaxy.managers.libraries.LibraryManager method), 154

delete() (galaxy.objectstore.DiskObjectStore method), 157

delete() (galaxy.objectstore.NestedObjectStore method), 157

delete() (galaxy.objectstore.ObjectStore method), 158

delete() (galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController method), 8, 264

delete() (galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController method), 9, 265

delete() (galaxy.webapps.galaxy.api.folders.FoldersController method), 11, 267

delete() (galaxy.webapps.galaxy.api.group_roles.GroupRolesApiController method), 14, 269

delete() (galaxy.webapps.galaxy.api.group_users.GroupUsersApiController method), 14, 270

delete() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 15, 271

delete() (galaxy.webapps.galaxy.api.history_contents.HistoryContentsController method), 17, 273

delete() (galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController method), 19, 275

delete() (galaxy.webapps.galaxy.api.lda_datasets.LibraryDatasetsController method), 22, 278

delete() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 25, 280

delete() (galaxy.webapps.galaxy.api.library_contents.LibraryContentsController method), 28, 283

delete() (galaxy.webapps.galaxy.api.pages.PagesController method), 31, 286

delete() (galaxy.webapps.galaxy.api.provenance.BaseProvenanceController method), 31, 287

delete() (galaxy.webapps.galaxy.api.quotas.QuotaApiController method), 32, 287

delete() (galaxy.webapps.galaxy.api.users.UserApiController method), 37, 292

delete() (galaxy.webapps.galaxy.api.workflows.WorkflowsApiController method), 38, 293

delete() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304

delete() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336

delete_address() (galaxy.webapps.galaxy.controllers.user.User method), 328

delete_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304

delete_current() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 309

delete_extended_metadata() (galaxy.web.base.controller.UsesExtendedMetadataMixin method), 236

delete_external_service() (galaxy.webapps.galaxy.controllers.external_service.ExternalService method), 306

delete_external_service_associations() (galaxy.model.RequestType method), 138

delete_form_definition() (galaxy.webapps.galaxy.controllers.forms.Forms method), 307

delete_hidden_datasets() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 309

delete_item_annotation() (galaxy.model.item_attrs.UsesAnnotations method), 295

delete_item_tags() (galaxy.tags.tag_handler.TagHandler method), 169

delete_library() (galaxy.webapps.galaxy.controllers.library_admin.LibraryAdminController method), 313

delete_library_item() (galaxy.webapps.galaxy.controllers.library_common.LibraryCommonController method), 314

delete_operation (galaxy.web.base.controllers.admin.Admin attribute), 240

delete_operation (galaxy.webapps.galaxy.controllers.admin.AdminGalaxyController method), 295

delete_request() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommonController method), 323

delete_request_type() (galaxy.webapps.galaxy.controllers.request_type.RequestTypeController method), 320

delete_sample() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommonController method), 323

delete_template() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 236

deleted_datasets() (galaxy.webapps.reports.controllers.system.SystemController method), 342

deleted_histories() (galaxy.webapps.reports.controllers.system.SystemController method), 343

DeleteDatasetAction (class in galaxy.jobs.actions.post), 115

DeletedColumn (class in galaxy.web.framework.helpers.grids), 246

DeleteIntermediatesAction (class in galaxy.jobs.actions.post), 115

demodulize() (galaxy.util.inflection.Base method), 219

demodulize() (galaxy.util.inflection.Inflector method), 220

DemowebAPITransaction (class in galaxy.webapps.demo_sequencer.framework), 257

DemoWebTransaction (class in galaxy.webapps.demo_sequencer.framework), 257

DistrowebAPITransaction (class in galaxy.webapps.demo_sequencer.framework), 257

- 258
- `dependency_shell_commands()`
(`galaxy.tools.deps.DependencyManager` method), 183
- `dependency_shell_commands()`
(`galaxy.tools.deps.NullDependencyManager` method), 183
- `DependencyManager` (class in `galaxy.tools.deps`), 183
- `deprecated_options` (`galaxy.config.Configuration` attribute), 40
- `DeprecatedMethod`, 97
- `derive_roles_from_access()`
(`galaxy.security.GalaxyRBACAgent` method), 164
- `derive_roles_from_access()` (`galaxy.security.RBACAgent` method), 167
- `determine_output_format()` (in module `galaxy.tools.actions`), 179
- `dict_collection_visible_keys`
(`galaxy.model.DatasetCollection` attribute), 123
- `dict_collection_visible_keys`
(`galaxy.model.DatasetCollectionElement` attribute), 124
- `dict_collection_visible_keys`
(`galaxy.model.FormDefinition` attribute), 127
- `dict_collection_visible_keys` (`galaxy.model.Group` attribute), 128
- `dict_collection_visible_keys` (`galaxy.model.History` attribute), 129
- `dict_collection_visible_keys`
(`galaxy.model.ItemTagAssociation` attribute), 131
- `dict_collection_visible_keys` (`galaxy.model.Job` attribute), 132
- `dict_collection_visible_keys` (`galaxy.model.Library` attribute), 134
- `dict_collection_visible_keys` (`galaxy.model.Quota` attribute), 137
- `dict_collection_visible_keys` (`galaxy.model.Request` attribute), 137
- `dict_collection_visible_keys` (`galaxy.model.RequestType` attribute), 138
- `dict_collection_visible_keys` (`galaxy.model.Role` attribute), 138
- `dict_collection_visible_keys` (`galaxy.model.Sample` attribute), 139
- `dict_collection_visible_keys`
(`galaxy.model.StoredWorkflow` attribute), 139
- `dict_collection_visible_keys` (`galaxy.model.User` attribute), 141
- `dict_collection_visible_keys` (`galaxy.model.Workflow` attribute), 143
- `dict_collection_visible_keys`
(`galaxy.model.WorkflowInvocation` attribute), 143
- `dict_collection_visible_keys`
(`galaxy.model.WorkflowInvocationStep` attribute), 144
- `dict_collection_visible_keys`
(`galaxy.model.WorkflowRequest` attribute), 144
- `dict_collection_visible_keys`
(`galaxy.model.WorkflowRequestInputParameter` attribute), 144
- `dict_collection_visible_keys`
(`galaxy.model.WorkflowRequestStepState` attribute), 144
- `dict_collection_visible_keys`
(`galaxy.model.WorkflowRequestToInputDatasetAssociation` attribute), 144
- `dict_collection_visible_keys`
(`galaxy.model.WorkflowRequestToInputDatasetCollectionAssociation` attribute), 144
- `dict_collection_visible_keys`
(`galaxy.tools.data.TabularToolDataTable` attribute), 181
- `dict_collection_visible_keys`
(`galaxy.tools.parameters.basic.FloatToolParameter` attribute), 191
- `dict_collection_visible_keys`
(`galaxy.tools.parameters.basic.IntegerToolParameter` attribute), 193
- `dict_collection_visible_keys`
(`galaxy.tools.parameters.basic.ToolParameter` attribute), 195
- `dict_collection_visible_keys`
(`galaxy.tools.parameters.grouping.ConditionalWhen` attribute), 199
- `dict_collection_visible_keys`
(`galaxy.tools.parameters.grouping.Group` attribute), 200
- `dict_collection_visible_keys`
(`galaxy.tools.parameters.grouping.Repeat` attribute), 200
- `dict_collection_visible_keys` (`galaxy.tools.Tool` attribute), 172
- `dict_collection_visible_keys` (`galaxy.tools.ToolOutput` attribute), 177
- `dict_collection_visible_keys` (`galaxy.tools.ToolSection` attribute), 177
- `dict_collection_visible_keys`
(`galaxy.tools.ToolSectionLabel` attribute), 177

<code>dict_dataprovider()</code> (galaxy.datatypes.binary.Bam method), 43	<code>dict_element_visible_keys</code> (galaxy.model.WorkflowInvocation attribute), 143
<code>dict_dataprovider()</code> (galaxy.datatypes.tabular.Sam method), 82	<code>dict_element_visible_keys</code> (galaxy.model.WorkflowInvocationStep attribute), 144
<code>dict_dataprovider()</code> (galaxy.datatypes.tabular.Tabular method), 83	<code>dict_element_visible_keys</code> (galaxy.model.WorkflowRequest attribute), 144
<code>dict_element_visible_keys</code> (galaxy.model.DatasetCollection attribute), 123	<code>Dictifiable</code> (class in galaxy.model.item_attrs), 146
<code>dict_element_visible_keys</code> (galaxy.model.DatasetCollectionElement attribute), 124	<code>dictify_dataset_collection_instance()</code> (in module galaxy.managers.collections_util), 150
<code>dict_element_visible_keys</code> (galaxy.model.DefaultQuotaAssociation attribute), 126	<code>dictify_element()</code> (in module galaxy.managers.collections_util), 150
<code>dict_element_visible_keys</code> (galaxy.model.FormDefinition attribute), 127	<code>DictParameter</code> (class in galaxy.datatypes.metadata), 67
<code>dict_element_visible_keys</code> (galaxy.model.Group attribute), 128	<code>disk_usage()</code> (galaxy.webapps.reports.controllers.system.System method), 343
<code>dict_element_visible_keys</code> (galaxy.model.GroupQuotaAssociation attribute), 128	<code>DiskObjectStore</code> (class in galaxy.objectstore), 156
<code>dict_element_visible_keys</code> (galaxy.model.History attribute), 129	<code>display()</code> (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
<code>dict_element_visible_keys</code> (galaxy.model.ItemTagAssociation attribute), 131	<code>display()</code> (galaxy.webapps.galaxy.controllers.page.PageController method), 318
<code>dict_element_visible_keys</code> (galaxy.model.Job attribute), 132	<code>display()</code> (galaxy.webapps.galaxy.controllers.root.RootController method), 325
<code>dict_element_visible_keys</code> (galaxy.model.Library attribute), 134	<code>display_amount</code> (galaxy.model.Quota attribute), 137
<code>dict_element_visible_keys</code> (galaxy.model.LibraryFolder attribute), 136	<code>display_application()</code> (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
<code>dict_element_visible_keys</code> (galaxy.model.Page attribute), 136	<code>display_as()</code> (galaxy.webapps.galaxy.controllers.root.RootController method), 325
<code>dict_element_visible_keys</code> (galaxy.model.PageRevision attribute), 137	<code>display_at()</code> (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
<code>dict_element_visible_keys</code> (galaxy.model.Quota attribute), 137	<code>display_by_id()</code> (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
<code>dict_element_visible_keys</code> (galaxy.model.RequestType attribute), 138	<code>display_by_username_and_slug()</code> (galaxy.web.base.controller.SharableMixin method), 236
<code>dict_element_visible_keys</code> (galaxy.model.Role attribute), 138	<code>display_by_username_and_slug()</code> (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
<code>dict_element_visible_keys</code> (galaxy.model.StoredWorkflow attribute), 139	<code>display_by_username_and_slug()</code> (galaxy.webapps.galaxy.controllers.history.HistoryController method), 309
<code>dict_element_visible_keys</code> (galaxy.model.User attribute), 141	<code>display_by_username_and_slug()</code> (galaxy.webapps.galaxy.controllers.page.PageController method), 318
<code>dict_element_visible_keys</code> (galaxy.model.UserQuotaAssociation attribute), 142	<code>display_by_username_and_slug()</code> (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 333
<code>dict_element_visible_keys</code> (galaxy.model.Workflow attribute), 143	<code>display_by_username_and_slug()</code> (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
	<code>display_child()</code> (galaxy.webapps.galaxy.controllers.root.RootController method), 325

- display_data() (galaxy.datatypes.binary.Binary method), 44
- display_data() (galaxy.datatypes.data.Data method), 47
- display_data() (galaxy.datatypes.tabular.Tabular method), 83
- display_image_in_repository()
(galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 301
- display_info() (galaxy.datatypes.data.Data method), 48
- display_info() (galaxy.model.DatasetInstance method), 124
- display_name() (galaxy.datatypes.data.Data method), 48
- display_name() (galaxy.model.DatasetCollectionInstance method), 124
- display_name() (galaxy.model.DatasetInstance method), 124
- display_name() (galaxy.model.LibraryDataset method), 134
- display_peek() (galaxy.datatypes.binary.Ab1 method), 42
- display_peek() (galaxy.datatypes.binary.Bam method), 43
- display_peek() (galaxy.datatypes.binary.BigWig method), 44
- display_peek() (galaxy.datatypes.binary.GeminiSQLite method), 44
- display_peek() (galaxy.datatypes.binary.H5 method), 44
- display_peek() (galaxy.datatypes.binary.Scf method), 45
- display_peek() (galaxy.datatypes.binary.Sff method), 45
- display_peek() (galaxy.datatypes.binary.SQLite method), 45
- display_peek() (galaxy.datatypes.binary.TwoBit method), 45
- display_peek() (galaxy.datatypes.data.Data method), 48
- display_peek() (galaxy.datatypes.genetics.RexpBase method), 53
- display_peek() (galaxy.datatypes.genetics.rgTabList method), 55
- display_peek() (galaxy.datatypes.images.Gmaj method), 56
- display_peek() (galaxy.datatypes.images.Laj method), 57
- display_peek() (galaxy.datatypes.interval.CustomTrack method), 61
- display_peek() (galaxy.datatypes.interval.Gff method), 62
- display_peek() (galaxy.datatypes.interval.Interval method), 64
- display_peek() (galaxy.datatypes.interval.Wiggle method), 65
- display_peek() (galaxy.datatypes.ngsindex.BowtieIndex method), 70
- display_peek() (galaxy.datatypes.sequence.Maf method), 76
- display_peek() (galaxy.datatypes.tabular.Pileup method), 81
- display_peek() (galaxy.datatypes.tabular.Sam method), 82
- display_peek() (galaxy.datatypes.tabular.Tabular method), 83
- display_peek() (galaxy.datatypes.tabular.Taxonomy method), 84
- display_peek() (galaxy.datatypes.tabular.Vcf method), 84
- display_peek() (galaxy.model.DatasetInstance method), 84
- display_ready() (galaxy.datatypes.display_applications.application.Populated method), 91
- display_structured() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 309
- display_url() (galaxy.datatypes.display_applications.application.Populated method), 91
- displayable() (galaxy.datatypes.interval.Interval method), 65
- displayable() (galaxy.datatypes.tabular.Tabular method), 83
- DisplayApplication (class in galaxy.datatypes.display_applications.application), 90
- DisplayApplicationDataParameter (class in galaxy.datatypes.display_applications.parameters), 91
- DisplayApplicationLink (class in galaxy.datatypes.display_applications.application), 91
- DisplayApplicationParameter (class in galaxy.datatypes.display_applications.parameters), 91
- DisplayApplicationTemplateParameter (class in galaxy.datatypes.display_applications.parameters), 92
- DisplayByUsernameAndSlugGridOperation (class in galaxy.web.framework.helpers.grids), 247
- DisplayDataValueWrapper (class in galaxy.datatypes.display_applications.parameters), 92
- DisplayParameterValueWrapper (class in galaxy.datatypes.display_applications.parameters), 92
- dist_config_file (galaxy.eggs.dist.DistScrambleCrate attribute), 96
- DistributedObjectStore (class in galaxy.objectstore), 157
- Distribution (class in pkg_resources), 360
- DistributionNotFound, 361
- DistScrambleCrate (class in galaxy.eggs.dist), 96
- DistScrambleEgg (class in galaxy.eggs.dist), 96
- DNA_COMPLEMENT (galaxy.tools.util.maf_utilities.RegionAlignment attribute), 210
- DNA_COMPLEMENT (galaxy.tools.util.maf_utilities.SplicedAlignment attribute), 210
- DNA_complement() (in module galaxy_utils.sequence.transform), 354

DNA_reverse_complement() (in module galaxy_utils.sequence.transform), 354
 do_fast_split() (galaxy.datatypes.sequence.Sequence class method), 76
 do_invocation_step_action() (galaxy.workflow.modules.PauseModule method), 346
 do_invocation_step_action() (galaxy.workflow.modules.WorkflowModule method), 348
 do_merge() (in module galaxy.jobs.splitters.basic), 122
 do_merge() (in module galaxy.jobs.splitters.multi), 122
 do_slow_split() (galaxy.datatypes.sequence.Sequence class method), 77
 do_split() (in module galaxy.jobs.splitters.basic), 122
 do_split() (in module galaxy.jobs.splitters.multi), 122
 docstring_trim() (in module galaxy.util), 213
 doi() (galaxy.managers.citations.DoiCitation method), 149
 DoiCache (class in galaxy.managers.citations), 149
 DoiCitation (class in galaxy.managers.citations), 149
 download() (galaxy.webapps.galaxy.api.lda_datasets.LibraryDatasetsController method), 22, 278
 download_dataset_from_folder() (galaxy.webapps.galaxy.controllers.library_common_library_controller method), 315
 DrillDownField (class in galaxy.web.form_builder), 229
 DrillDownSelectToolParameter (class in galaxy.tools.parameters.basic), 188
 DRMAAJobRunner (class in galaxy.jobs.runners.drmaa), 120
 DummyDataset (class in galaxy.tools.parameters.basic), 190
 dump() (galaxy.util.aliaspickler.AliasPickleModule method), 217
 dump() (galaxy.util.memdump.Memdump method), 222
 dumps() (galaxy.util.aliaspickler.AliasPickleModule method), 217
 dumps() (in module galaxy.util.json), 221
 DuplicatedSlugException, 98
 DynamicDisplayApplicationBuilder (class in galaxy.datatypes.display_applications.application), 91
 DynamicOptions (class in galaxy.tools.parameters.dynamic_options), 197

E

eagers (pkg_resources.ZipProvider attribute), 365
 echo() (galaxy.webapps.galaxy.controllers.root.RootController method), 325
 echo_json() (galaxy.webapps.galaxy.controllers.root.RootController method), 325
 edit() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
 edit() (galaxy.webapps.galaxy.controllers.page.PageController method), 318
 edit() (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 333
 edit_address() (galaxy.webapps.galaxy.controllers.user.User method), 328
 edit_attributes() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
 edit_basic_request_info() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommonController method), 323
 edit_content() (galaxy.webapps.galaxy.controllers.page.PageController method), 318
 edit_email_settings() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommonController method), 323
 edit_external_service() (galaxy.webapps.galaxy.controllers.external_service.ExternalServiceController method), 306
 edit_external_service_form_definition() (galaxy.webapps.galaxy.controllers.external_service.ExternalServiceController method), 306
 edit_form_definition() (galaxy.webapps.galaxy.controllers.forms.FormsController method), 307
 edit_info() (galaxy.webapps.galaxy.controllers.user.User method), 328
 edit_quota() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxyController method), 295
 edit_request_type() (galaxy.webapps.galaxy.controllers.request_type.RequestTypeController method), 320
 edit_samples() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommonController method), 323
 edit_template() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 237
 edit_template_info() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 237
 edit_toolbox_filters() (galaxy.webapps.galaxy.controllers.user.User method), 328
 edit_username() (galaxy.webapps.galaxy.controllers.user.User method), 328
 editable_keys (galaxy.model.HistoryDatasetCollectionAssociation attribute), 130
 editable_keys (galaxy.model.LibraryDatasetCollectionAssociation attribute), 135
 editor() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
 editor_form_post() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
 Egg (class in galaxy.eggs), 95
 egg_info (pkg_resources.NullProvider attribute), 365
 egg_name (pkg_resources.NullProvider attribute), 365
 egg_name() (pkg_resources.Distribution method), 360
 EggMetadata (class in pkg_resources), 365
 EggNotFetchable, 95

- EggProvider (class in pkg_resources), 365
- Eigenstratgeno (class in galaxy.datatypes.genetics), 51
- Eigenstratpca (class in galaxy.datatypes.genetics), 51
- Eland (class in galaxy.datatypes.tabular), 80
- ElandMulti (class in galaxy.datatypes.tabular), 81
- element_is_set() (galaxy.datatypes.metadata.MetadataCollection method), 67
- element_object (galaxy.model.DatasetCollectionElement attribute), 124
- element_type (galaxy.model.DatasetCollectionElement attribute), 124
- EmailAction (class in galaxy.jobs.actions.post), 115
- empty (galaxy.model.History attribute), 129
- empty() (galaxy.objectstore.DiskObjectStore method), 157
- empty() (galaxy.objectstore.NestedObjectStore method), 157
- empty() (galaxy.objectstore.ObjectStore method), 158
- empty_field (galaxy.webapps.galaxy.controllers.forms.Forms attribute), 307
- empty_page() (galaxy.webapps.demo_sequencer.controllers.common.CommonController method), 257
- EmptyProvider (class in pkg_resources), 365
- EmptyTextfieldValidator (class in galaxy.tools.parameters.validation), 206
- encode() (galaxy.tools.DefaultToolState method), 171
- encode_all_ids() (galaxy.web.base.controller.BaseController method), 234
- encode_all_ids() (galaxy.web.security.SecurityHelper method), 254
- encode_dataset_user() (in module galaxy.datatypes.display_applications.util), 93
- encode_dict_ids() (galaxy.web.security.SecurityHelper method), 254
- encode_guid() (galaxy.web.security.SecurityHelper method), 254
- encode_id() (galaxy.web.security.SecurityHelper method), 254
- encode_runtime_state() (galaxy.workflow.modules.SimpleWorkflowModule method), 346
- encode_runtime_state() (galaxy.workflow.modules.ToolModule method), 347
- encode_runtime_state() (galaxy.workflow.modules.WorkflowModule method), 348
- ENCODEPeak (class in galaxy.datatypes.interval), 62
- end (galaxy.tools.util.maf_utilities.SplicedAlignment attribute), 210
- engine (galaxy.model.Dataset attribute), 122
- English (class in galaxy.util.inflection), 219
- ensure_directory() (in module pkg_resources), 363
- ensure_tempdir() (galaxy.config.Configuration method), 40
- EntryPoint (class in pkg_resources), 361
- Environment (class in pkg_resources), 357
- Eps (class in galaxy.datatypes.images), 56
- equals() (galaxy.managers.citations.BaseCitation method), 149
- err_code (galaxy.exceptions.ActionInputError attribute), 97
- err_code (galaxy.exceptions.AdminRequiredException attribute), 97
- err_code (galaxy.exceptions.AuthenticationFailed attribute), 97
- err_code (galaxy.exceptions.AuthenticationRequired attribute), 97
- err_code (galaxy.exceptions.ConfigDoesNotAllowException attribute), 97
- err_code (galaxy.exceptions.Conflict attribute), 97
- err_code (galaxy.exceptions.DeprecatedMethod attribute), 98
- err_code (galaxy.exceptions.DuplicatedSlugException attribute), 98
- err_code (galaxy.exceptions.InconsistentDatabase attribute), 98
- err_code (galaxy.exceptions.InsufficientPermissionsException attribute), 98
- err_code (galaxy.exceptions.InternalServerError attribute), 98
- err_code (galaxy.exceptions.ItemAccessibilityException attribute), 98
- err_code (galaxy.exceptions.ItemOwnershipException attribute), 98
- err_code (galaxy.exceptions.MalformedId attribute), 98
- err_code (galaxy.exceptions.MessageException attribute), 99
- err_code (galaxy.exceptions.NotImplemented attribute), 99
- err_code (galaxy.exceptions.ObjectAttributeInvalidException attribute), 99
- err_code (galaxy.exceptions.ObjectAttributeMissingException attribute), 99
- err_code (galaxy.exceptions.ObjectNotFound attribute), 99
- err_code (galaxy.exceptions.RequestParameterInvalidException attribute), 99
- err_code (galaxy.exceptions.RequestParameterMissingException attribute), 99
- err_code (galaxy.exceptions.ToolMetaParameterException attribute), 100
- err_code (galaxy.exceptions.UnknownContentsType attribute), 100
- err_if_uploading() (galaxy.managers.hdas.HDAManager method), 153
- Error (class in galaxy.webapps.galaxy.controllers.error), 306
- error() (galaxy.web.framework.middleware.remoteuser.RemoteUser method), 253

- errors() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
- Eset (class in galaxy.datatypes.genetics), 51
- estimate_file_lines() (galaxy.datatypes.data.Text method), 50
- evaluate_marker() (in module pkg_resources), 363
- Event (class in galaxy.model), 127
- exec_after_process() (galaxy.tools.DataManagerTool method), 170
- exec_after_process() (galaxy.tools.SetMetadataTool method), 171
- exec_after_process() (galaxy.tools.Tool method), 172
- exec_before_job() (galaxy.tools.DataSourceTool method), 170
- exec_before_job() (galaxy.tools.OutputParameterJSONTool method), 171
- exec_before_job() (galaxy.tools.Tool method), 172
- execute() (galaxy.jobs.actions.post.ActionBox class method), 114
- execute() (galaxy.jobs.actions.post.ChangeDatatypeAction class method), 114
- execute() (galaxy.jobs.actions.post.ColumnSetAction class method), 114
- execute() (galaxy.jobs.actions.post.DefaultJobAction class method), 114
- execute() (galaxy.jobs.actions.post.DeleteDatasetAction class method), 115
- execute() (galaxy.jobs.actions.post.DeleteIntermediatesAction class method), 115
- execute() (galaxy.jobs.actions.post.EmailAction class method), 115
- execute() (galaxy.jobs.actions.post.HideDatasetAction class method), 115
- execute() (galaxy.jobs.actions.post.RenameDatasetAction class method), 115
- execute() (galaxy.jobs.actions.post.SetMetadataAction class method), 116
- execute() (galaxy.tools.actions.DefaultToolAction method), 179
- execute() (galaxy.tools.actions.history_imp_exp.ExportHistoryToolAction method), 180
- execute() (galaxy.tools.actions.history_imp_exp.ImportHistoryToolAction method), 180
- execute() (galaxy.tools.actions.metadata.SetMetadataToolAction method), 180
- execute() (galaxy.tools.actions.ToolAction method), 179
- execute() (galaxy.tools.actions.upload.UploadToolAction method), 180
- execute() (galaxy.tools.Tool method), 172
- execute() (galaxy.workflow.modules.InputModule method), 345
- execute() (galaxy.workflow.modules.PauseModule method), 346
- execute() (galaxy.workflow.modules.ToolModule method), 347
- execute() (galaxy.workflow.modules.WorkflowModule method), 348
- execute_via_app() (galaxy.tools.actions.metadata.SetMetadataToolAction method), 180
- exists() (galaxy.objectstore.DiskObjectStore method), 157
- exists() (galaxy.objectstore.HierarchicalObjectStore method), 157
- exists() (galaxy.objectstore.NestedObjectStore method), 157
- exists() (galaxy.objectstore.ObjectStore method), 158
- exit_code() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
- exmeta_item_id (galaxy.webapps.galaxy.api.extended_metadata.HistoryData attribute), 10, 266
- exmeta_item_id (galaxy.webapps.galaxy.api.extended_metadata.LibraryData attribute), 10, 266
- expand_here_template() (in module galaxy.tools.data), 183
- expand_input_elems() (in module galaxy.tools.test), 178
- expand_user_properties() (galaxy.model.User static method), 141
- export() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
- export_archive() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
- export_name (galaxy.model.JobExportHistoryArchive attribute), 133
- export_to_file() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 336
- export_to_myexp() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
- exported_workflows() (galaxy.webapps.galaxy.api.tool_shed_repositories.ToolShedRepositories method), 34, 289
- ExportHistoryTool (class in galaxy.tools), 171
- ExportHistoryToolAction (class in galaxy.tools.actions.history_imp_exp), 180
- ExportHistoryToolMixin (class in galaxy.web.base.controller), 235
- ExportToolAction (class in galaxy.web.framework.helpers.grids.Grid attribute), 247
- EXPOSED_ADMIN_OPTIONS (galaxy.webapps.galaxy.api.configuration.ConfigurationController attribute), 8, 264
- EXPOSED_USER_OPTIONS (galaxy.webapps.galaxy.api.configuration.ConfigurationController attribute), 8, 264
- ExpressionContext (class in galaxy.util.expressions), 218
- ExpressionValidator (class in galaxy.tools.parameters.validation), 206
- expunge_all() (galaxy.managers.context.ProvidesAppContext method), 150

ext (galaxy.model.DatasetInstance attribute), 124
 extend_data_with() (galaxy.tools.data.TabularToolDataTable method), 181
 extend_validation_errors() (galaxy.model.DatasetInstance method), 124
 ExtendedMetadata (class in galaxy.model), 127
 ExtendedMetadataIndex (class in galaxy.model), 127
 external_metadata_set_successfully() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 67
 external_runjob() (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 120
 external_service_grid (galaxy.webapps.galaxy.controllers.external_service_grid.ExternalServiceGrid attribute), 306
 external_services (galaxy.model.RequestType attribute), 138
 ExternalService (class in galaxy.model), 127
 ExternalService (class in galaxy.webapps.galaxy.controllers.external_service_grid.ExternalServiceGrid), 306
 ExternalServiceAction (class in galaxy.external_services.actions), 100
 ExternalServiceActionJQueryGridResultHandler (class in galaxy.external_services.result_handlers.basic), 103
 ExternalServiceActionJSONResultHandler (class in galaxy.external_services.result_handlers.basic), 104
 ExternalServiceActionResultHandler (class in galaxy.external_services.result_handlers.basic), 104
 ExternalServiceActionsConditional (class in galaxy.external_services.service), 102
 ExternalServiceActionsGroup (class in galaxy.external_services.service), 102
 ExternalServiceActionsGroupWhen (class in galaxy.external_services.service), 102
 ExternalServiceActionURLRedirectResultHandler (class in galaxy.external_services.result_handlers.basic), 104
 ExternalServiceController (class in galaxy.webapps.galaxy.controllers.external_service_grid.ExternalServiceGrid), 307
 ExternalServiceGrid (class in galaxy.webapps.galaxy.controllers.external_service_grid.ExternalServiceGrid), 306
 ExternalServiceGrid.ExternalServiceTypeColumn (class in galaxy.webapps.galaxy.controllers.external_service_grid.ExternalServiceGrid), 306
 ExternalServiceGrid.NameColumn (class in galaxy.webapps.galaxy.controllers.external_service_grid.ExternalServiceGrid), 306
 ExternalServiceParameter (class in galaxy.external_services.parameters), 101
 ExternalServiceResult (class in galaxy.external_services.actions), 100
 ExternalServiceTemplateAction (class in galaxy.external_services.actions), 100
 ExternalServiceTemplateParameter (class in galaxy.external_services.parameters), 101
 ExternalServiceType (class in galaxy.sample_tracking.external_service_types), 162
 ExternalServiceTypeNotFoundException, 162
 ExternalServiceTypesCollection (class in galaxy.sample_tracking.external_service_types), 162
 ExternalServiceValueResult (class in galaxy.external_services.actions), 100
 ExternalServiceWebAction (class in galaxy.external_services.actions), 101
 ExternalServiceWebAPIAction (class in galaxy.external_services.actions), 100
 ExternalServiceWebAPIAction.ExternalServiceWebAPIActionRequest (class in galaxy.external_services.actions), 100
 ExternalServiceWebAPIActionResult (class in galaxy.external_services.actions), 101
 extra_files_path (galaxy.model.Dataset attribute), 122
 extra_files_path (galaxy.model.DatasetInstance attribute), 124
 extract_value() (galaxy.tools.test.ParamContext method), 178
 extraction_error() (pkg_resources.ResourceManager method), 359
 extraction_path (pkg_resources.ResourceManager attribute), 359
 ExtractionError, 362
 extras (pkg_resources.Distribution attribute), 360
 ez_setup (galaxy.eggs.scramble.ScrambleEgg attribute), 96
 ez_setup_url (galaxy.eggs.scramble.ScrambleEgg attribute), 96

F

fail() (galaxy.jobs.JobWrapper method), 108
 fail() (galaxy.jobs.TaskWrapper method), 111
 fail() (galaxy.model.WorkflowInvocation method), 143
 fail() (in module galaxy.tools.util.galaxyops), 212
 fail_job() (galaxy.jobs.runners.AsynchronousJobRunner method), 117
 fail_job() (galaxy.jobs.runners.lwr.LwrJobRunner method), 121
 FAILURE (galaxy.web.framework.openid_manager.OpenIDManager attribute), 244
 FakeTrans (class in galaxy.jobs.deferred), 116
 Fasta (class in galaxy.datatypes.sequence), 73

fastaNamedReader (class in galaxy_utils.sequence.fasta), 350
 fastaReader (class in galaxy_utils.sequence.fasta), 350
 fastaSequence (class in galaxy_utils.sequence.fasta), 350
 fastaWriter (class in galaxy_utils.sequence.fasta), 350
 Fastq (class in galaxy.datatypes.sequence), 74
 fastqAggregator (class in galaxy_utils.sequence.fastq), 350
 fastqCombiner (class in galaxy_utils.sequence.fastq), 351
 FastqCSSanger (class in galaxy.datatypes.sequence), 75
 fastqCSSangerRead (class in galaxy_utils.sequence.fastq), 351
 fastqFakeFastaScoreReader (class in galaxy_utils.sequence.fastq), 351
 FastqIllumina (class in galaxy.datatypes.sequence), 75
 fastqIlluminaRead (class in galaxy_utils.sequence.fastq), 351
 fastqJoiner (class in galaxy_utils.sequence.fastq), 352
 fastqNamedReader (class in galaxy_utils.sequence.fastq), 352
 fastqReader (class in galaxy_utils.sequence.fastq), 352
 FastqSanger (class in galaxy.datatypes.sequence), 75
 fastqSangerRead (class in galaxy_utils.sequence.fastq), 352
 fastqSequencingRead (class in galaxy_utils.sequence.fastq), 352
 FastqSolexa (class in galaxy.datatypes.sequence), 75
 fastqSolexaRead (class in galaxy_utils.sequence.fastq), 353
 fastqSplitter (class in galaxy_utils.sequence.fastq), 353
 fastqVerboseErrorReader (class in galaxy_utils.sequence.fastq), 353
 fastqWriter (class in galaxy_utils.sequence.fastq), 353
 FeatureLocationIndex (class in galaxy.datatypes.tabular), 81
 fetch() (galaxy.eggs.Egg method), 95
 fetch_one() (galaxy.eggs.scramble.ScrambleEgg method), 96
 fetch_source() (galaxy.eggs.scramble.ScrambleEgg method), 96
 field (in module galaxy.forms.forms), 106
 field_as_html() (galaxy.model.FormDefinition method), 127
 field_param_values_ok() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 237
 fields() (galaxy.web.form_builder.AddressField static method), 229
 FieldStorage (class in galaxy.web.framework.base), 242
 file_ext (galaxy.datatypes.assembly.Amos attribute), 41
 file_ext (galaxy.datatypes.assembly.Velvet attribute), 42
 file_ext (galaxy.datatypes.binary.Abl attribute), 42
 file_ext (galaxy.datatypes.binary.Bam attribute), 43
 file_ext (galaxy.datatypes.binary.Bcf attribute), 43
 file_ext (galaxy.datatypes.binary.GeminiSQLite attribute), 44
 file_ext (galaxy.datatypes.binary.GenericAsn1Binary attribute), 44
 file_ext (galaxy.datatypes.binary.H5 attribute), 45
 file_ext (galaxy.datatypes.binary.Scf attribute), 45
 file_ext (galaxy.datatypes.binary.Sff attribute), 45
 file_ext (galaxy.datatypes.binary.SQLite attribute), 45
 file_ext (galaxy.datatypes.binary.TwoBit attribute), 45
 file_ext (galaxy.datatypes.binary.Xlsx attribute), 46
 file_ext (galaxy.datatypes.chrominfo.ChromInfo attribute), 46
 file_ext (galaxy.datatypes.coverage.LastzCoverage attribute), 46
 file_ext (galaxy.datatypes.data.GenericAsn1 attribute), 49
 file_ext (galaxy.datatypes.data.Newick attribute), 50
 file_ext (galaxy.datatypes.data.Nexus attribute), 50
 file_ext (galaxy.datatypes.data.Text attribute), 50
 file_ext (galaxy.datatypes.genetics.Affybatch attribute), 51
 file_ext (galaxy.datatypes.genetics.Eigenstratgeno attribute), 51
 file_ext (galaxy.datatypes.genetics.Eigenstratpca attribute), 51
 file_ext (galaxy.datatypes.genetics.Eset attribute), 51
 file_ext (galaxy.datatypes.genetics.Fped attribute), 52
 file_ext (galaxy.datatypes.genetics.Fphe attribute), 52
 file_ext (galaxy.datatypes.genetics.GenomeGraphs attribute), 52
 file_ext (galaxy.datatypes.genetics.IdIndep attribute), 55
 file_ext (galaxy.datatypes.genetics.Lped attribute), 52
 file_ext (galaxy.datatypes.genetics.MAlst attribute), 53
 file_ext (galaxy.datatypes.genetics.Pbed attribute), 53
 file_ext (galaxy.datatypes.genetics.Phe attribute), 53
 file_ext (galaxy.datatypes.genetics.Phenotype attribute), 53
 file_ext (galaxy.datatypes.genetics.Pphe attribute), 53
 file_ext (galaxy.datatypes.genetics.RexpBase attribute), 53
 file_ext (galaxy.datatypes.genetics.Rgenetics attribute), 54
 file_ext (galaxy.datatypes.genetics.rgFeatureList attribute), 55
 file_ext (galaxy.datatypes.genetics.rgSampleList attribute), 55
 file_ext (galaxy.datatypes.genetics.rgTabList attribute), 56
 file_ext (galaxy.datatypes.genetics.SNPMatrix attribute), 55
 file_ext (galaxy.datatypes.genetics.Snptest attribute), 55
 file_ext (galaxy.datatypes.images.Bmp attribute), 56
 file_ext (galaxy.datatypes.images.Eps attribute), 56
 file_ext (galaxy.datatypes.images.Gif attribute), 56
 file_ext (galaxy.datatypes.images.Gmaj attribute), 56
 file_ext (galaxy.datatypes.images.Html attribute), 57

- `file_ext` (galaxy.datatypes.images.Im attribute), 57
- `file_ext` (galaxy.datatypes.images.Jpg attribute), 57
- `file_ext` (galaxy.datatypes.images.Laj attribute), 57
- `file_ext` (galaxy.datatypes.images.Pbm attribute), 58
- `file_ext` (galaxy.datatypes.images.Pcd attribute), 58
- `file_ext` (galaxy.datatypes.images.Pcx attribute), 58
- `file_ext` (galaxy.datatypes.images.Pdf attribute), 58
- `file_ext` (galaxy.datatypes.images.Pgm attribute), 58
- `file_ext` (galaxy.datatypes.images.Png attribute), 58
- `file_ext` (galaxy.datatypes.images.Ppm attribute), 58
- `file_ext` (galaxy.datatypes.images.Psd attribute), 59
- `file_ext` (galaxy.datatypes.images.Rast attribute), 59
- `file_ext` (galaxy.datatypes.images.Rgb attribute), 59
- `file_ext` (galaxy.datatypes.images.Tiff attribute), 59
- `file_ext` (galaxy.datatypes.images.Xbm attribute), 59
- `file_ext` (galaxy.datatypes.images.Xpm attribute), 59
- `file_ext` (galaxy.datatypes.interval.Bed attribute), 60
- `file_ext` (galaxy.datatypes.interval.Bed12 attribute), 60
- `file_ext` (galaxy.datatypes.interval.Bed6 attribute), 60
- `file_ext` (galaxy.datatypes.interval.BedGraph attribute), 61
- `file_ext` (galaxy.datatypes.interval.BedStrict attribute), 61
- `file_ext` (galaxy.datatypes.interval.ChromatinInteractions attribute), 61
- `file_ext` (galaxy.datatypes.interval.CustomTrack attribute), 61
- `file_ext` (galaxy.datatypes.interval.ENCODEPeak attribute), 62
- `file_ext` (galaxy.datatypes.interval.Gff attribute), 62
- `file_ext` (galaxy.datatypes.interval.Gff3 attribute), 63
- `file_ext` (galaxy.datatypes.interval.Gtf attribute), 64
- `file_ext` (galaxy.datatypes.interval.Interval attribute), 65
- `file_ext` (galaxy.datatypes.interval.Wiggle attribute), 66
- `file_ext` (galaxy.datatypes.ngsindex.BowtieBaseIndex attribute), 70
- `file_ext` (galaxy.datatypes.ngsindex.BowtieColorIndex attribute), 70
- `file_ext` (galaxy.datatypes.ngsindex.BowtieIndex attribute), 70
- `file_ext` (galaxy.datatypes.qualityscore.QualityScore attribute), 70
- `file_ext` (galaxy.datatypes.qualityscore.QualityScore454 attribute), 71
- `file_ext` (galaxy.datatypes.qualityscore.QualityScoreIllumina attribute), 71
- `file_ext` (galaxy.datatypes.qualityscore.QualityScoreSolexa attribute), 71
- `file_ext` (galaxy.datatypes.qualityscore.QualityScoreSOLiD attribute), 71
- `file_ext` (galaxy.datatypes.sequence.Axt attribute), 73
- `file_ext` (galaxy.datatypes.sequence.csFasta attribute), 77
- `file_ext` (galaxy.datatypes.sequence.Fasta attribute), 73
- `file_ext` (galaxy.datatypes.sequence.Fastq attribute), 74
- `file_ext` (galaxy.datatypes.sequence.FastqCSSanger attribute), 75
- `file_ext` (galaxy.datatypes.sequence.FastqIllumina attribute), 75
- `file_ext` (galaxy.datatypes.sequence.FastqSanger attribute), 75
- `file_ext` (galaxy.datatypes.sequence.FastqSolexa attribute), 75
- `file_ext` (galaxy.datatypes.sequence.Lav attribute), 75
- `file_ext` (galaxy.datatypes.sequence.Maf attribute), 76
- `file_ext` (galaxy.datatypes.sequence.MafCustomTrack attribute), 76
- `file_ext` (galaxy.datatypes.sequence.SequenceSplitLocations attribute), 77
- `file_ext` (galaxy.datatypes.tabular.Eland attribute), 80
- `file_ext` (galaxy.datatypes.tabular.ElandMulti attribute), 81
- `file_ext` (galaxy.datatypes.tabular.FeatureLocationIndex attribute), 81
- `file_ext` (galaxy.datatypes.tabular.Pileup attribute), 81
- `file_ext` (galaxy.datatypes.tabular.Sam attribute), 82
- `file_ext` (galaxy.datatypes.tabular.Vcf attribute), 84
- `file_ext` (galaxy.datatypes.tracks.GeneTrack attribute), 84
- `file_ext` (galaxy.datatypes.xml.CisML attribute), 84
- `file_ext` (galaxy.datatypes.xml.GenericXml attribute), 85
- `file_ext` (galaxy.datatypes.xml.MEMEXml attribute), 85
- `file_ext` (galaxy.datatypes.xml.Owl attribute), 85
- `file_ext` (galaxy.datatypes.xml.Phyloxml attribute), 85
- `file_iter()` (in module galaxy.util), 213
- `file_name` (galaxy.datatypes.metadata.MetadataTempFile attribute), 69
- `file_name` (galaxy.model.Dataset attribute), 122
- `file_name` (galaxy.model.DatasetInstance attribute), 125
- `file_name` (galaxy.model.MetadataFile attribute), 136
- `file_path` (galaxy.model.Dataset attribute), 122
- `file_reader()` (in module galaxy.util), 213
- `file_ready()` (galaxy.objectstore.NestedObjectStore method), 158
- `file_ready()` (galaxy.objectstore.ObjectStore method), 158
- `FileField` (class in galaxy.web.form_builder), 231
- `FileMetadata` (class in pkg_resources), 364
- `FileParameter` (class in galaxy.datatypes.metadata), 67
- `FileToolParameter` (class in galaxy.tools.parameters.basic), 190
- `fill_in_new_state()` (galaxy.tools.Tool method), 172
- `fill_region_alignment()` (in module galaxy.tools.util.maf_utilities), 210
- `fill_template()` (galaxy.webapps.demo_sequencer.framework.DemoWebTran method), 257
- `fill_template()` (in module galaxy.util.template), 223
- `fill_template_mako()` (galaxy.webapps.demo_sequencer.framework.DemoW method), 257
- `fill_template_string()` (galaxy.webapps.demo_sequencer.framework.DemoW method), 257

[find_plugins\(\) \(pkg_resources.WorkingSet method\), 358](#)
[find_samples\(\) \(galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method\), 323](#)
[find_samples_index\(\) \(galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method\), 321](#)
[find_tools_in_tool_shed\(\) \(galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method\), 302](#)
[find_workflows_in_tool_shed\(\) \(galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method\), 302](#)
[finish\(\) \(galaxy.jobs.JobWrapper method\), 108](#)
[finish\(\) \(galaxy.jobs.TaskWrapper method\), 111](#)
[finish_job\(\) \(galaxy.jobs.runners.AsynchronousJobRunner method\), 117](#)
[finish_job\(\) \(galaxy.jobs.runners.cli.ShellJobRunner method\), 119](#)
[finish_job\(\) \(galaxy.jobs.runners.lwr.LwrJobRunner method\), 121](#)
[finished \(galaxy.model.Job attribute\), 132](#)
[first_dataset_instance\(\) \(galaxy.model.DatasetCollectionElement method\), 124](#)
[fitness_and_quality_parsed\(\) \(in module mimeparse\), 356](#)
[fix_integrated_tool_panel_dict\(\) \(galaxy.tools.ToolBox method\), 176](#)
[fixup_namespace_packages\(\) \(in module pkg_resources\), 366](#)
[flatten\(\) \(galaxy.util.Params method\), 213](#)
[flatten\(\) \(in module galaxy.web.framework.base\), 244](#)
[FloatToolParameter \(class in galaxy.tools.parameters.basic\), 190](#)
[flush\(\) \(galaxy.tools.util.maf_utilities.RegionAlignment method\), 210](#)
[flush\(\) \(galaxy.tools.util.maf_utilities.TempFileHandler method\), 210](#)
[folder_info\(\) \(galaxy.webapps.galaxy.controllers.library_controller.LibraryController method\), 315](#)
[folder_is_public\(\) \(galaxy.security.GalaxyRBACAgent method\), 164](#)
[folder_is_public\(\) \(galaxy.security.RBACAgent method\), 167](#)
[folder_is_unrestricted\(\) \(galaxy.security.GalaxyRBACAgent method\), 164](#)
[folder_permissions\(\) \(galaxy.webapps.galaxy.controllers.library_controller.LibraryController method\), 315](#)
[FolderContentsController \(class in galaxy.webapps.galaxy.api.folder_contents\), 10, 266](#)
[FolderManager \(class in galaxy.managers.folders\), 151](#)
[FoldersController \(class in galaxy.webapps.galaxy.api.folders\), 11, 267](#)
[for_direct_import\(\) \(galaxy.webapps.galaxy.controllers.workflows_controller.WorkflowsController method\), 337](#)
[for_state\(\) \(galaxy.tools.test.ParamContext method\), 178](#)
[for_state\(\) \(galaxy.tools.test.RootParamContext method\), 178](#)
[force_bed_field_count\(\) \(in module galaxy.datatypes.converters.interval_to_bedstrict_converter\), 87](#)
[force_symlink\(\) \(in module galaxy.util\), 213](#)
[foreignKey\(\) \(in module galaxy.util.inflection.Base method\), 219](#)
[foreignKey\(\) \(galaxy.util.inflection.Inflector method\), 220](#)
[format \(galaxy.datatypes.framework.middleware.translogger.TransLogger attribute\), 253](#)
[format \(galaxy_utils.sequence.fastq.fastqCSSangerRead attribute\), 351](#)
[format \(galaxy_utils.sequence.fastq.fastqIlluminaRead attribute\), 351](#)
[format \(galaxy_utils.sequence.fastq.fastqSangerRead attribute\), 352](#)
[format \(galaxy_utils.sequence.fastq.fastqSequencingRead attribute\), 352](#)
[format \(galaxy_utils.sequence.fastq.fastqSolexaRead attribute\), 353](#)
[format \(in module galaxy_utils.sequence.fastq\), 354](#)
[format \(in module galaxy_utils.sequence.vcf\), 355](#)
[format_bool\(\) \(in module galaxy.webapps.galaxy.controllers.page\), 319](#)
[formats \(galaxy.datatypes.display_applications.parameters.DisplayApplication attribute\), 91](#)
[FormatToolOutputAction \(class in galaxy.tools.parameters.output\), 202](#)
[FormDefinition \(class in galaxy.model\), 127](#)
[FormDefinitionAddressFieldFactory \(class in galaxy.forms.forms\), 104](#)
[FormDefinitionAPIController \(class in galaxy.webapps.galaxy.api.forms\), 13, 268](#)
[FormDefinitionCommon \(class in galaxy.model\), 127](#)
[FormDefinitionFactory \(class in galaxy.forms.forms\), 104](#)
[FormDefinitionFieldFactory \(class in galaxy.forms.forms\), 104](#)
[FormDefinitionHistoryFieldFactory \(class in galaxy.forms.forms\), 105](#)
[FormDefinitionPasswordFieldFactory \(class in galaxy.forms.forms\), 105](#)
[FormDefinitionSelectFieldFactory \(class in galaxy.forms.forms\), 105](#)
[FormDefinitionTextFieldFactory \(class in galaxy.forms.forms\), 105](#)
[FormDefinitionWorkflowFieldFactory \(class in galaxy.forms.forms\), 105](#)
[FormDefinitionWorkflowMappingFieldFactory \(class in galaxy.forms.forms\), 106](#)
[FormWorkflowController \(class in galaxy.webapps.galaxy.controllers.forms\), 307](#)

[forms_grid \(galaxy.webapps.galaxy.controllers.forms.Formsfrom_elem\(\) \(galaxy.forms.forms.FormDefinitionFieldFactory attribute\), 307](#)
[FormsGrid \(class in galaxy.webapps.galaxy.controllers.formsfrom_elem\(\) \(galaxy.forms.forms.FormDefinitionHistoryFieldFactory method\), 105](#)
[FormsGrid.DescriptionColumn \(class in from_elem\(\) \(galaxy.forms.forms.FormDefinitionPasswordFieldFactory method\), 105](#)
[FormsGrid.NameColumn \(class in from_elem\(\) \(galaxy.forms.forms.FormDefinitionSelectFieldFactory method\), 105](#)
[FormsGrid.TypeColumn \(class in from_elem\(\) \(galaxy.forms.forms.FormDefinitionTextFieldFactory method\), 105](#)
[FormsGrid.TypeColumn \(class in from_elem\(\) \(galaxy.forms.forms.FormDefinitionWorkflowFieldFactory method\), 105](#)
[FormsGrid.TypeColumn \(class in from_elem\(\) \(galaxy.forms.forms.FormDefinitionWorkflowMappingFieldFactory method\), 106](#)
[FormValues \(class in galaxy.model\), 127](#)
[Fped \(class in galaxy.datatypes.genetics\), 51](#)
[Fphe \(class in galaxy.datatypes.genetics\), 52](#)
[from_dict\(\) \(galaxy.visualization.genomes.GenomeRegion static method\), 225](#)
[from_dict\(\) \(galaxy.workflow.modules.SimpleWorkflowModule class method\), 346](#)
[from_dict\(\) \(galaxy.workflow.modules.ToolModule class method\), 347](#)
[from_dict\(\) \(galaxy.workflow.modules.WorkflowModule class method\), 348](#)
[from_dict\(\) \(galaxy.workflow.modules.WorkflowModuleFactory class method\), 349](#)
[from_elem\(\) \(galaxy.datatypes.display_applications.application_from_display_application tools.parameters.output.ToolOutputActionConditional class method\), 90](#)
[from_elem\(\) \(galaxy.datatypes.display_applications.application_from_display_application tools.parameters.output.ToolOutputActionOption class method\), 91](#)
[from_elem\(\) \(galaxy.datatypes.display_applications.parameter_from_display_application parameters.output.ToolOutputActionOptionFilter class method\), 91](#)
[from_elem\(\) \(galaxy.external_services.actions.ExternalServiceAction\(\) \(galaxy.tools.parameters.dynamic_options.Filter class method\), 100](#)
[from_elem\(\) \(galaxy.external_services.parameters.ExternalServiceParameter\(\) \(galaxy.tools.parameters.input_translation.ToolInputTranslator class method\), 101](#)
[from_elem\(\) \(galaxy.external_services.result_handlers.basic_from_elem\(\) \(galaxy.tools.parameters.input_translation.ToolInputTranslator class method\), 104](#)
[from_elem\(\) \(galaxy.external_services.result_handlers.basic_from_elem\(\) \(galaxy.tools.parameters.input_translation.ToolInputTranslator class method\), 104](#)
[from_elem\(\) \(galaxy.external_services.service.BooleanExternalService\(\) \(galaxy.tools.parameters.input_translation.ToolInputTranslator class method\), 102](#)
[from_elem\(\) \(galaxy.external_services.service.ExternalService\(\) \(galaxy.tools.parameters.input_translation.ToolInputTranslator class method\), 102](#)
[from_elem\(\) \(galaxy.external_services.service.ExternalService\(\) \(galaxy.tools.parameters.input_translation.ToolInputTranslator class method\), 102](#)
[from_elem\(\) \(galaxy.external_services.service.ItemIsInstanceOf\(\) \(galaxy.tools.parameters.input_translation.ToolInputTranslator class method\), 103](#)
[from_elem\(\) \(galaxy.external_services.service.ValueExternalService\(\) \(galaxy.tools.parameters.input_translation.ToolInputTranslator class method\), 103](#)
[from_elem\(\) \(galaxy.forms.forms.FormDefinitionAddressFieldFactory\(\) \(galaxy.tools.parameters.validation.MetadataInDataTableColumnValidator class method\), 104](#)
[from_elem\(\) \(galaxy.forms.forms.FormDefinitionFactory from_element\(\) \(galaxy.tools.parameters.validation.MetadataValidator class method\), 104](#)

from_element() (galaxy.tools.parameters.validation.NoOptionsValidator class method), 208
 from_element() (galaxy.tools.parameters.validation.RegexValidator class method), 209
 from_element() (galaxy.tools.parameters.validation.UnspecifiedValidator class method), 209
 from_element() (galaxy.tools.parameters.validation.Validator class method), 209
 from_external_value() (galaxy.datatypes.metadata.FileParameter class method), 67
 from_external_value() (galaxy.datatypes.metadata.MetadataParameter class method), 68
 from_file() (galaxy.datatypes.display_applications.application_from_file class method), 90
 from_file() (galaxy.openid.providers.OpenIDProvider class method), 160
 from_file() (galaxy.openid.providers.OpenIDProviders class method), 160
 from_filename() (pkg_resources.Distribution class method), 360
 from_html() (galaxy.tools.parameters.basic.BooleanToolParameter class method), 186
 from_html() (galaxy.tools.parameters.basic.ColumnListParameter class method), 186
 from_html() (galaxy.tools.parameters.basic.DataCollectionToolParameter class method), 187
 from_html() (galaxy.tools.parameters.basic.DataToolParameter class method), 187
 from_html() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter class method), 189
 from_html() (galaxy.tools.parameters.basic.FileToolParameter class method), 190
 from_html() (galaxy.tools.parameters.basic.FloatToolParameter class method), 191
 from_html() (galaxy.tools.parameters.basic.FTPFileToolParameter class method), 190
 from_html() (galaxy.tools.parameters.basic.IntegerToolParameter class method), 193
 from_html() (galaxy.tools.parameters.basic.LibraryDatasetToolParameter class method), 193
 from_html() (galaxy.tools.parameters.basic.SelectToolParameter class method), 194
 from_html() (galaxy.tools.parameters.basic.ToolParameter class method), 195
 from_JSON() (galaxy.datatypes.metadata.MetadataTempFile class method), 69
 from_json() (galaxy.tools.parameters.basic.BooleanToolParameter class method), 186
 from_json() (galaxy.tools.parameters.basic.ToolParameter class method), 196
 from_JSON_dict() (galaxy.datatypes.metadata.MetadataCollection class method), 67
 from_location() (pkg_resources.Distribution class method), 361
 from_validate() (galaxy.visualization.genomes.GenomeRegion class static method), 225
 from_workflow_step() (galaxy.workflow.modules.SimpleWorkflowModule class method), 346
 from_workflow_step() (galaxy.workflow.modules.ToolModule class method), 347
 from_workflow_step() (galaxy.workflow.modules.WorkflowModule class method), 348
 from_workflow_step() (galaxy.workflow.modules.WorkflowModuleFactory class method), 349
 FromDataTableOutputActionOption (class in galaxy.tools.parameters.output), 202
 FromDisplayToolOutputActionOption (class in galaxy.tools.parameters.output), 202
 FromParamToolOutputActionOption (class in galaxy.tools.parameters.output), 202
 FtpDataTransferFactory (class in galaxy.sample_tracking.data_transfer), 161
 FTPFileField (class in galaxy.web.form_builder), 231
 FTPFilesAPIController (class in galaxy.webapps.galaxy.api.ftp_files), 13, 269
 FTPFileToolParameter (class in galaxy.tools.parameters.basic), 190
 ToolParameter (class in galaxy.model.Dataset method), 122
 func_std_string() (in module galaxy.web.framework.middleware.profile), 252

G

galaxy (module), 40
 galaxy.actions.admin (module), 41
 galaxy.config (module), 40
 galaxy.datatypes.assembly (module), 41
 galaxy.datatypes.binary (module), 42
 galaxy.datatypes.checkers (module), 46
 galaxy.datatypes.chrominfo (module), 46
 galaxy.datatypes.converters.bed_to_gff_converter (module), 86
 galaxy.datatypes.converters.bgzip (module), 86
 galaxy.datatypes.converters.fasta_to_len (module), 86
 galaxy.datatypes.converters.fasta_to_tabular_converter (module), 86
 galaxy.datatypes.converters.fastq_to_fqtoc (module), 86
 galaxy.datatypes.converters.fastqsolexa_to_fasta_converter (module), 87
 galaxy.datatypes.converters.fastqsolexa_to_qual_converter (module), 87
 galaxy.datatypes.converters.gff_to_bed_converter (module), 87
 galaxy.datatypes.converters.gff_to_interval_index_converter (module), 87
 galaxy.datatypes.converters.interval_to_bed_converter (module), 87

galaxy.datatypes.converters.interval_to_bedstrict_converter (module), 87
 galaxy.datatypes.converters.interval_to_coverage (module), 87
 galaxy.datatypes.converters.interval_to_fli (module), 88
 galaxy.datatypes.converters.interval_to_interval_index_converter (module), 88
 galaxy.datatypes.converters.interval_to_tabix_converter (module), 88
 galaxy.datatypes.converters.lped_to_fped_converter (module), 88
 galaxy.datatypes.converters.lped_to_pbed_converter (module), 89
 galaxy.datatypes.converters.maf_to_fasta_converter (module), 89
 galaxy.datatypes.converters.maf_to_interval_converter (module), 89
 galaxy.datatypes.converters.pbed_ldreduced_converter (module), 89
 galaxy.datatypes.converters.pbed_to_lped_converter (module), 89
 galaxy.datatypes.converters.picard_interval_list_to_bed6_converter (module), 90
 galaxy.datatypes.converters.sam_to_bam (module), 90
 galaxy.datatypes.converters.vcf_to_interval_index_converter (module), 90
 galaxy.datatypes.converters.vcf_to_vcf_bgzip (module), 90
 galaxy.datatypes.converters.wiggle_to_array_tree_converter (module), 90
 galaxy.datatypes.converters.wiggle_to_simple_converter (module), 90
 galaxy.datatypes.coverage (module), 46
 galaxy.datatypes.data (module), 46
 galaxy.datatypes.display_applications.application (module), 90
 galaxy.datatypes.display_applications.parameters (module), 91
 galaxy.datatypes.display_applications.util (module), 92
 galaxy.datatypes.genetics (module), 51
 galaxy.datatypes.images (module), 56
 galaxy.datatypes.interval (module), 60
 galaxy.datatypes.metadata (module), 66
 galaxy.datatypes.ngsindex (module), 70
 galaxy.datatypes.qualityscore (module), 70
 galaxy.datatypes.registry (module), 71
 galaxy.datatypes.sequence (module), 73
 galaxy.datatypes.sniff (module), 78
 galaxy.datatypes.tabular (module), 80
 galaxy.datatypes.tracks (module), 84
 galaxy.datatypes.util (module), 93
 galaxy.datatypes.util.gff_util (module), 93
 galaxy.datatypes.util.image_util (module), 94
 galaxy.datatypes.xml (module), 84
 galaxy.eggs (module), 94
 galaxy.eggs.dist (module), 96
 galaxy.eggs.scramble (module), 96
 galaxy.exceptions (module), 97
 galaxy.external_services.actions (module), 100
 galaxy.external_services.parameters (module), 101
 galaxy.external_services.result_handlers.basic (module), 103
 galaxy.external_services.service (module), 102
 galaxy.forms.forms (module), 104
 galaxy.jobs (module), 106
 galaxy.jobs.actions (module), 114
 galaxy.jobs.actions.post (module), 114
 galaxy.jobs.deferred (module), 116
 galaxy.jobs.deferred.data_transfer (module), 116
 galaxy.jobs.deferred.manual_data_transfer (module), 117
 galaxy.jobs.deferred.pacific_biosciences_smrt_portal (module), 117
 galaxy.jobs.handler (module), 111
 galaxy.jobs.manager (module), 113
 galaxy.jobs.mapper (module), 113
 galaxy.jobs.runners (module), 117
 galaxy.jobs.runners.cli (module), 119
 galaxy.jobs.runners.condor (module), 119
 galaxy.jobs.runners.drmaa (module), 120
 galaxy.jobs.runners.local (module), 120
 galaxy.jobs.runners.lwr (module), 121
 galaxy.jobs.runners.tasks (module), 121
 galaxy.jobs.splitters.basic (module), 122
 galaxy.jobs.splitters.multi (module), 122
 galaxy.jobs.transfer_manager (module), 113
 galaxy.managers (module), 148
 galaxy.managers.api_keys (module), 148
 galaxy.managers.base (module), 148
 galaxy.managers.citations (module), 149
 galaxy.managers.collections (module), 149
 galaxy.managers.collections_util (module), 150
 galaxy.managers.context (module), 150
 galaxy.managers.folders (module), 151
 galaxy.managers.hdas (module), 152
 galaxy.managers.histories (module), 153
 galaxy.managers.lddas (module), 154
 galaxy.managers.libraries (module), 154
 galaxy.managers.roles (module), 156
 galaxy.managers.tags (module), 156
 galaxy.managers.workflows (module), 156
 galaxy.model (module), 122
 galaxy.model.custom_types (module), 145
 galaxy.model.item_attrs (module), 146
 galaxy.model.mapping (module), 146
 galaxy.model.migrate.check (module), 147
 galaxy.model.orm (module), 147
 galaxy.model.orm.logging_connection_proxy (module), 147

galaxy.objectstore (module), 156
 galaxy.objectstore.s3_multipart_upload (module), 159
 galaxy.openid (module), 160
 galaxy.openid.providers (module), 160
 galaxy.quota (module), 161
 galaxy.sample_tracking.data_transfer (module), 161
 galaxy.sample_tracking.external_service_types (module), 162
 galaxy.sample_tracking.request_types (module), 162
 galaxy.sample_tracking.sample (module), 163
 galaxy.security (module), 163
 galaxy.security.validate_user_input (module), 168
 galaxy.tags (module), 168
 galaxy.tags.tag_handler (module), 168
 galaxy.tools (module), 170
 galaxy.tools.actions (module), 179
 galaxy.tools.actions.history_imp_exp (module), 180
 galaxy.tools.actions.metadata (module), 180
 galaxy.tools.actions.upload (module), 180
 galaxy.tools.actions.upload_common (module), 180
 galaxy.tools.data (module), 181
 galaxy.tools.deps (module), 183
 galaxy.tools.exception_handling (module), 178
 galaxy.tools.imp_exp (module), 184
 galaxy.tools.imp_exp.export_history (module), 184
 galaxy.tools.imp_exp.unpack_tar_gz_archive (module), 184
 galaxy.tools.parameters (module), 185
 galaxy.tools.parameters.basic (module), 185
 galaxy.tools.parameters.dynamic_options (module), 197
 galaxy.tools.parameters.grouping (module), 199
 galaxy.tools.parameters.input_translation (module), 201
 galaxy.tools.parameters.output (module), 201
 galaxy.tools.parameters.sanitize (module), 205
 galaxy.tools.parameters.validation (module), 206
 galaxy.tools.search (module), 209
 galaxy.tools.test (module), 178
 galaxy.tools.util (module), 209
 galaxy.tools.util.galaxyops (module), 212
 galaxy.tools.util.maf_utilities (module), 210
 galaxy.util (module), 213
 galaxy.util.aliaspickler (module), 217
 galaxy.util.backports (module), 224
 galaxy.util.backports.importlib (module), 225
 galaxy.util.bunch (module), 217
 galaxy.util.debugging (module), 217
 galaxy.util.expressions (module), 218
 galaxy.util.hash_util (module), 218
 galaxy.util.heartbeat (module), 218
 galaxy.util.inflection (module), 219
 galaxy.util.json (module), 221
 galaxy.util.lrucache (module), 222
 galaxy.util.memdump (module), 222
 galaxy.util.none_like (module), 222
 galaxy.util.odict (module), 223
 galaxy.util.sanitize_html (module), 223
 galaxy.util.streamball (module), 223
 galaxy.util.template (module), 223
 galaxy.util.topsort (module), 224
 galaxy.visualization (module), 225
 galaxy.visualization.data_providers (module), 226
 galaxy.visualization.data_providers.basic (module), 226
 galaxy.visualization.data_providers.phyloviz (module), 227
 galaxy.visualization.data_providers.phyloviz.baseparser (module), 227
 galaxy.visualization.data_providers.phyloviz.newickparser (module), 228
 galaxy.visualization.data_providers.phyloviz.nexusparser (module), 228
 galaxy.visualization.data_providers.phyloviz.phyloxmlparser (module), 228
 galaxy.visualization.genome (module), 229
 galaxy.visualization.genomes (module), 225
 galaxy.visualization.tracks (module), 229
 galaxy.web (module), 229
 galaxy.web.base.controller (module), 234
 galaxy.web.base.controllers.admin (module), 240
 galaxy.web.buildapp (module), 229
 galaxy.web.form_builder (module), 229
 galaxy.web.framework (module), 241
 galaxy.web.framework.base (module), 242
 galaxy.web.framework.helpers (module), 245
 galaxy.web.framework.helpers.grids (module), 245
 galaxy.web.framework.middleware (module), 252
 galaxy.web.framework.middleware.profile (module), 252
 galaxy.web.framework.middleware.remoteuser (module), 253
 galaxy.web.framework.middleware.static (module), 253
 galaxy.web.framework.middleware.translogger (module), 253
 galaxy.web.framework.middleware.xforwardedhost (module), 253
 galaxy.web.framework.openid_manager (module), 244
 galaxy.web.params (module), 234
 galaxy.web.security (module), 254
 galaxy.webapps (module), 254
 galaxy.webapps.demo_sequencer (module), 256
 galaxy.webapps.demo_sequencer.app (module), 256
 galaxy.webapps.demo_sequencer.buildapp (module), 256
 galaxy.webapps.demo_sequencer.config (module), 256
 galaxy.webapps.demo_sequencer.controllers (module), 257
 galaxy.webapps.demo_sequencer.controllers.common (module), 257
 galaxy.webapps.demo_sequencer.framework (module), 257
 galaxy.webapps.demo_sequencer.registry (module), 257

galaxy.webapps.galaxy.api.annotations (module), 7, 263
galaxy.webapps.galaxy.api.authenticate (module), 8, 264
galaxy.webapps.galaxy.api.configuration (module), 8, 264
galaxy.webapps.galaxy.api.dataset_collections (module), 9, 265
galaxy.webapps.galaxy.api.datatypes (module), 9, 265
galaxy.webapps.galaxy.api.extended_metadata (module), 9, 265
galaxy.webapps.galaxy.api.folder_contents (module), 10, 266
galaxy.webapps.galaxy.api.folders (module), 11, 267
galaxy.webapps.galaxy.api.forms (module), 13, 268
galaxy.webapps.galaxy.api.ftp_files (module), 13, 269
galaxy.webapps.galaxy.api.genomes (module), 13, 269
galaxy.webapps.galaxy.api.group_roles (module), 14, 269
galaxy.webapps.galaxy.api.group_users (module), 14, 270
galaxy.webapps.galaxy.api.groups (module), 14, 270
galaxy.webapps.galaxy.api.histories (module), 15, 270
galaxy.webapps.galaxy.api.history_contents (module), 17, 272
galaxy.webapps.galaxy.api.item_tags (module), 19, 275
galaxy.webapps.galaxy.api.job_files (module), 20, 275
galaxy.webapps.galaxy.api.jobs (module), 21, 276
galaxy.webapps.galaxy.api.lda_datasets (module), 22, 278
galaxy.webapps.galaxy.api.libraries (module), 25, 280
galaxy.webapps.galaxy.api.library_contents (module), 27, 283
galaxy.webapps.galaxy.api.metrics (module), 29, 285
galaxy.webapps.galaxy.api.page_revisions (module), 30, 285
galaxy.webapps.galaxy.api.pages (module), 30, 286
galaxy.webapps.galaxy.api.provenance (module), 31, 287
galaxy.webapps.galaxy.api.quotas (module), 32, 287
galaxy.webapps.galaxy.api.request_types (module), 32, 287
galaxy.webapps.galaxy.api.requests (module), 32, 288
galaxy.webapps.galaxy.api.roles (module), 33, 288
galaxy.webapps.galaxy.api.samples (module), 33, 288
galaxy.webapps.galaxy.api.search (module), 33, 289
galaxy.webapps.galaxy.api.tool_data (module), 33, 289
galaxy.webapps.galaxy.api.tool_shed_repositories (module), 34, 289
galaxy.webapps.galaxy.api.users (module), 36, 292
galaxy.webapps.galaxy.api.visualizations (module), 37, 292
galaxy.webapps.galaxy.api.workflows (module), 37, 292
galaxy.webapps.galaxy.buildapp (module), 258
galaxy.webapps.galaxy.controllers (module), 295
galaxy.webapps.galaxy.controllers.admin (module), 295
galaxy.webapps.galaxy.controllers.admin_toolshed (module), 301
galaxy.webapps.galaxy.controllers.async (module), 303
galaxy.webapps.galaxy.controllers.dataset (module), 304
galaxy.webapps.galaxy.controllers.error (module), 306
galaxy.webapps.galaxy.controllers.external_service (module), 306
galaxy.webapps.galaxy.controllers.external_services (module), 307
galaxy.webapps.galaxy.controllers.forms (module), 307
galaxy.webapps.galaxy.controllers.history (module), 309
galaxy.webapps.galaxy.controllers.library (module), 313
galaxy.webapps.galaxy.controllers.library_admin (module), 313
galaxy.webapps.galaxy.controllers.library_common (module), 314
galaxy.webapps.galaxy.controllers.mobile (module), 316
galaxy.webapps.galaxy.controllers.page (module), 316
galaxy.webapps.galaxy.controllers.request_type (module), 320
galaxy.webapps.galaxy.controllers.requests (module), 321
galaxy.webapps.galaxy.controllers.requests_admin (module), 321
galaxy.webapps.galaxy.controllers.requests_common (module), 323
galaxy.webapps.galaxy.controllers.root (module), 325
galaxy.webapps.galaxy.controllers.tag (module), 326
galaxy.webapps.galaxy.controllers.tool_runner (module), 327
galaxy.webapps.galaxy.controllers.ucsc_proxy (module), 327
galaxy.webapps.galaxy.controllers.user (module), 327
galaxy.webapps.galaxy.controllers.visualization (module), 329
galaxy.webapps.galaxy.controllers.workflow (module), 335
galaxy.webapps.reports (module), 338
galaxy.webapps.reports.app (module), 338
galaxy.webapps.reports.buildapp (module), 338
galaxy.webapps.reports.config (module), 338
galaxy.webapps.reports.controllers (module), 339
galaxy.webapps.reports.controllers.jobs (module), 339
galaxy.webapps.reports.controllers.root (module), 341
galaxy.webapps.reports.controllers.sample_tracking (module), 341
galaxy.webapps.reports.controllers.system (module), 342
galaxy.webapps.reports.controllers.users (module), 343
galaxy.webapps.reports.controllers.workflows (module), 343
galaxy.workflow.modules (module), 345
galaxy_directory() (in module galaxy.util), 213
galaxy_system_pwent (galaxy.jobs.JobWrapper attribute), 109
galaxy_utils.sequence.fasta (module), 350
galaxy_utils.sequence.fastq (module), 350

[galaxy_utils.sequence.sequence \(module\)](#), 354
[galaxy_utils.sequence.transform \(module\)](#), 354
[galaxy_utils.sequence.vcf \(module\)](#), 355
[GalaxyConfig \(class in galaxy.eggs\)](#), 95
[GalaxyRBACAgent \(class in galaxy.security\)](#), 163
[GalaxySession \(class in galaxy.model\)](#), 127
[GalaxySessionToHistoryAssociation \(class in galaxy.model\)](#), 128
[GalaxyTagHandler \(class in galaxy.tags.tag_handler\)](#), 168
[GalaxyWebApplication \(class in galaxy.webapps.galaxy.buildapp\)](#), 258
[gbrowse_links\(\) \(galaxy.datatypes.interval.Gff method\)](#), 62
[gbrowse_links\(\) \(galaxy.datatypes.interval.Wiggle method\)](#), 66
[GeminiSQLite \(class in galaxy.datatypes.binary\)](#), 44
[gen_image\(\) \(galaxy.webapps.galaxy.controllers.workflow.WorkflowController method\)](#), 337
[generate_auto_primary_file\(\) \(galaxy.datatypes.data.Data method\)](#), 48
[generate_error\(\) \(galaxy.webapps.galaxy.controllers.root.RootController method\)](#), 326
[generate_json_error\(\) \(galaxy.webapps.galaxy.controllers.root.RootController method\)](#), 326
[generate_primary_file\(\) \(galaxy.datatypes.assembly.Velvet method\)](#), 42
[generate_primary_file\(\) \(galaxy.datatypes.genetics.RexpBam method\)](#), 53
[generate_primary_file\(\) \(galaxy.datatypes.genetics.RgeneticGenomicRegionAlignment \(class in galaxy.tools.util.maf_utilities\)](#), 210
[generate_primary_file\(\) \(galaxy.datatypes.ngsindex.BowtieIndex method\)](#), 70
[generate_workflow_image\(\) \(galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method\)](#), 302
[generateJsonableDict\(\) \(galaxy.visualization.data_providers.phyllo\(galaxy.visualization.phyllo\) method\)](#), 227
[GenericAsn1 \(class in galaxy.datatypes.data\)](#), 49
[GenericAsn1Binary \(class in galaxy.datatypes.binary\)](#), 44
[GenericXml \(class in galaxy.datatypes.xml\)](#), 85
[GeneTrack \(class in galaxy.datatypes.tracks\)](#), 84
[Genome \(class in galaxy.visualization.genomes\)](#), 225
[GenomeBuildParameter \(class in galaxy.tools.parameters.basic\)](#), 191
[GenomeGraphs \(class in galaxy.datatypes.genetics\)](#), 52
[GenomeIndexTool \(class in galaxy.tools\)](#), 171
[GenomeIndexToolData \(class in galaxy.model\)](#), 128
[GenomeRegion \(class in galaxy.visualization.genomes\)](#), 225
[Genomes \(class in galaxy.visualization.genomes\)](#), 225
[GenomesController \(class in galaxy.webapps.galaxy.api.genomes\)](#), 13, 269
[genomic_region_dataprovider\(\) \(galaxy.datatypes.binary.Bam method\)](#), 43
[genomic_region_dataprovider\(\) \(galaxy.datatypes.interval.Gff method\)](#), 62
[genomic_region_dataprovider\(\) \(galaxy.datatypes.interval.Interval method\)](#), 65
[genomic_region_dataprovider\(\) \(galaxy.datatypes.tabular.Pileup method\)](#), 81
[genomic_region_dataprovider\(\) \(galaxy.datatypes.tabular.Sam method\)](#), 82
[genomic_region_dataprovider\(\) \(galaxy.datatypes.tabular.Vcf method\)](#), 84
[genomic_region_dict_dataprovider\(\) \(galaxy.datatypes.binary.Bam method\)](#), 43
[genomic_region_dict_dataprovider\(\) \(galaxy.datatypes.interval.Gff method\)](#), 62
[genomic_region_dict_dataprovider\(\) \(galaxy.datatypes.interval.Interval method\)](#), 65
[genomic_region_dict_dataprovider\(\) \(galaxy.datatypes.tabular.Pileup method\)](#), 81
[genomic_region_dict_dataprovider\(\) \(galaxy.datatypes.tabular.Sam method\)](#), 82
[genomic_region_dict_dataprovider\(\) \(galaxy.datatypes.tabular.Vcf method\)](#), 84
[GenomicRegionAlignment \(class in galaxy.tools.util.maf_utilities\)](#), 210
[get\(\) \(galaxy.config.Configuration method\)](#), 40
[get\(\) \(galaxy.datatypes.metadata.MetadataCollection method\)](#), 67
[get\(\) \(galaxy.datatypes.metadata.MetadataElementSpec method\)](#), 68
[get\(\) \(galaxy.managers.hdas.HDAManager method\)](#), 153
[get\(\) \(galaxy.managers.histories.HistoryManager method\)](#), 153
[get\(\) \(galaxy.managers.lddas.LDDAManager method\)](#), 154
[get\(\) \(galaxy.managers.libraries.LibraryManager method\)](#), 154
[get\(\) \(galaxy.managers.roles.RoleManager method\)](#), 156
[get\(\) \(galaxy.openid.providers.OpenIDProviders method\)](#), 160
[get\(\) \(galaxy.tools.data.ToolDataTableManager method\)](#), 182
[get\(\) \(galaxy.util.bunch.Bunch method\)](#), 217
[get\(\) \(galaxy.util.memdump.Memdump method\)](#), 222
[get\(\) \(galaxy.util.Params method\)](#), 213
[get\(\) \(galaxy.webapps.demo_sequencer.config.Configuration method\)](#), 256

get() (galaxy.webapps.reports.config.Configuration method), 338	get_add_list() (galaxy.jobs.actions.post.ActionBox class method), 114
get() (galaxy_utils.sequence.fasta.fastaNamedReader method), 350	get_add_roles() (galaxy.managers.libraries.LibraryManager method), 155
get() (galaxy_utils.sequence.fastq.fastqFakeFastaScoreReader method), 351	get_all_forms() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 237
get() (galaxy_utils.sequence.fastq.fastqNamedReader method), 352	get_all_forms_by_type() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 237
get_accepted_filters() (galaxy.web.framework.helpers.grids.GridCollection method), 247	get_auth_token() (galaxy.security.GalaxyRBACAgent method), 164
get_accepted_filters() (galaxy.web.framework.helpers.grids.GridCollection method), 248	get_auth_token() (galaxy.security.RBACAgent method), 167
get_accepted_filters() (galaxy.web.framework.helpers.grids.GridCollection method), 251	get_auth_token() (galaxy.model.Quota method), 137
get_accepted_filters() (galaxy.web.framework.helpers.grids.StateColumn method), 251	get_annotation_async() (galaxy.webapps.galaxy.controllers.dataset.Dataset method), 304
get_accepted_filters() (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociation method), 305	get_api_key() (galaxy.webapps.galaxy.api.authenticate.AuthenticationController method), 364
get_access_roles() (galaxy.managers.libraries.LibraryManager method), 154	get_ascii_quality_scores() (galaxy_utils.sequence.fastq.fastqSequencingRead method), 352
get_access_roles() (galaxy.model.Dataset method), 123	get_ascii_quality_scores_len() (galaxy_utils.sequence.fastq.fastqSequencingRead method), 353
get_access_roles() (galaxy.model.HistoryDatasetAssociation method), 129	get_ascii_range() (galaxy_utils.sequence.fastq.fastqAggregator method), 350
get_access_roles() (galaxy.model.Library method), 134	get_attributes_from_fasta_header() (in module galaxy.tools.util.maf_utilities), 211
get_access_roles() (galaxy.model.LibraryDatasetDatasetAssociation method), 135	get_available_tracks() (galaxy.datatypes.registry.Registry method), 72
get_accessible_libraries() (galaxy.security.GalaxyRBACAgent method), 164	get_ave_item_rating_data() (galaxy.model.item_attrs.UsesItemRatings method), 146
get_accessible_libraries() (galaxy.security.RBACAgent method), 167	get_base_counts_for_column() (galaxy_utils.sequence.fastq.fastqAggregator method), 350
get_accessible_request_types() (galaxy.security.GalaxyRBACAgent method), 164	get_bibtex() (galaxy.managers.citations.DoiCache method), 149
get_accessible_request_types() (galaxy.security.RBACAgent method), 167	get_bibtex() (galaxy.webapps.config.Configuration method), 40
get_action() (galaxy.security.RBACAgent method), 167	get_bool() (galaxy.webapps.demo_sequencer.config.Configuration method), 100
get_action_access_link() (galaxy.external_services.actions.ExternalServiceAction method), 100	get_build() (galaxy.visualization.genomes.Genomes method), 225
get_action_access_link() (galaxy.external_services.actions.ExternalServiceAction method), 101	get_cache_path() (pkg_resources.ResourceManager method), 159
get_action_access_link() (galaxy.external_services.actions.PopulatedExternalService method), 101	get_charset_from_http_headers() (in module galaxy.util), 159
get_action_by_name() (galaxy.external_services.service.PopulatedExternalService method), 103	get_check_interval() (galaxy.model.DeferredJob method), 126
get_action_links() (galaxy.external_services.service.PopulatedExternalService method), 103	get_child_by_designation() (galaxy.model.DatasetInstance method), 125
get_actions() (galaxy.security.RBACAgent method), 167	get_chopped_blocks_for_region() (in module galaxy.tools.util.maf_utilities), 211
get_actions_for_items() (galaxy.security.GalaxyRBACAgent method), 164	
get_activation_token() (galaxy.webapps.galaxy.controllers.user.User method), 328	
get_active_folders() (galaxy.model.Library method), 134	

- get_chopped_blocks_with_index_offset_for_region() (in module galaxy.tools.util.maf_utilities), 211
- get_chunk() (galaxy.datatypes.tabular.Tabular method), 83
- get_class() (galaxy.web.base.controller.BaseController method), 234
- get_class() (in module galaxy.managers.base), 148
- get_class_by_format() (galaxy_utils.sequence.fastq.fastqSequencingReads method), 353
- get_class_by_format() (galaxy_utils.sequence.vcf.VariantCall method), 355
- get_cli_plugins() (galaxy.jobs.runners.cli.ShellJobRunner method), 119
- get_client() (galaxy.jobs.runners.lwr.LwrJobRunner method), 121
- get_client_from_state() (galaxy.jobs.runners.lwr.LwrJobRunner method), 121
- get_client_from_wrapper() (galaxy.jobs.runners.lwr.LwrJobRunner method), 121
- get_column_list() (galaxy.tools.parameters.basic.ColumnListParameter method), 186
- get_column_name_list() (galaxy.tools.data.TabularToolDataTable method), 181
- get_command_line() (galaxy.jobs.JobWrapper method), 109
- get_command_line() (galaxy.jobs.TaskWrapper method), 111
- get_command_line() (galaxy.model.Job method), 132
- get_command_line() (galaxy.model.Task method), 140
- get_community_tags() (galaxy.tags.tag_handler.TagHandler method), 169
- get_component_associations() (galaxy.security.GalaxyRBACAgent method), 164
- get_component_associations() (galaxy.security.RBACAgent method), 167
- get_components_by_src() (in module galaxy.tools.util.maf_utilities), 211
- get_components_by_src_start() (in module galaxy.tools.util.maf_utilities), 211
- get_composite_dataset_name() (galaxy.tools.parameters.grouping.UploadDataset method), 200
- get_composite_extensions() (galaxy.datatypes.registry.Registry method), 72
- get_composite_files() (galaxy.datatypes.data.Data method), 48
- get_comptypes() (in module galaxy.webapps.galaxy.controllers.library_common), 315
- get_config_dict() (galaxy.webapps.galaxy.api.configuration.Configuration method), 8, 264
- get_config_form() (galaxy.jobs.actions.post.ChangeDatatypeAction class method), 114
- get_config_form() (galaxy.jobs.actions.post.ColumnSetAction class method), 114
- get_config_form() (galaxy.jobs.actions.post.DefaultJobAction class method), 114
- get_config_form() (galaxy.jobs.actions.post.DeleteDatasetAction class method), 115
- get_config_form() (galaxy.jobs.actions.post.DeleteIntermediatesAction class method), 115
- get_config_form() (galaxy.jobs.actions.post.EmailAction class method), 115
- get_config_form() (galaxy.jobs.actions.post.HideDatasetAction class method), 115
- get_config_form() (galaxy.jobs.actions.post.RenameDatasetAction class method), 115
- get_config_form() (galaxy.jobs.actions.post.SetMetadataAction class method), 116
- get_config_form() (galaxy.workflow.modules.SimpleWorkflowModule method), 346
- get_config_form() (galaxy.workflow.modules.ToolModule method), 347
- get_config_form() (galaxy.workflow.modules.WorkflowModule method), 348
- get_consumer() (galaxy.web.framework.openid_manager.OpenIDManager method), 244
- get_containing_library_from_library_dataset() (in module galaxy.webapps.galaxy.controllers.library_common), 315
- get_content_type() (galaxy.web.framework.base.Response method), 243
- get_converted_dataset() (galaxy.model.DatasetInstance method), 125
- get_converted_dataset_deps() (galaxy.model.DatasetInstance method), 125
- get_converted_files_by_type() (galaxy.model.DatasetInstance method), 125
- get_converter_by_target_type() (galaxy.datatypes.registry.Registry method), 72
- get_converter_types() (galaxy.datatypes.data.Data method), 48
- get_converter_types() (galaxy.model.DatasetInstance method), 125
- get_converters_by_datatype() (galaxy.datatypes.registry.Registry method), 72
- get_cookie() (galaxy.webapps.demo_sequencer.framework.DemoWebTrans method), 257
- get_current_case() (galaxy.tools.parameters.grouping.Conditional method), 199
- get_current_case() (galaxy.external_services.service.ExternalServiceAction method), 102

[get_current_form\(\)](#) (galaxy.webapps.galaxy.controllers.forms.FolderForm method), 307
[get_current_item\(\)](#) (galaxy.web.framework.helpers.grids.Grid method), 247
[get_current_item\(\)](#) (galaxy.webapps.galaxy.controllers.admin.dataset_collection method), 301
[get_current_item\(\)](#) (galaxy.webapps.galaxy.controllers.history.dataset_collection_instance method), 311
[get_current_item\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.history_dataset_selection_grid method), 331
[get_current_roles\(\)](#) (galaxy.managers.folders.FolderManager method), 152
[get_current_roles\(\)](#) (galaxy.managers.libraries.LibraryManager method), 155
[get_current_thread_object_dict\(\)](#) (in module galaxy.util.heartbeat), 218
[get_current_user_roles\(\)](#) (galaxy.jobs.deferred.FakeTrans method), 116
[get_current_user_roles\(\)](#) (galaxy.managers.context.ProvidesExternalContext method), 151
[get_data\(\)](#) (galaxy.objectstore.DiskObjectStore method), 157
[get_data\(\)](#) (galaxy.objectstore.NestedObjectStore method), 158
[get_data\(\)](#) (galaxy.objectstore.ObjectStore method), 159
[get_data\(\)](#) (galaxy.visualization.data_providers.basic.BaseDataProvider method), 226
[get_data\(\)](#) (galaxy.visualization.data_providers.basic.ColumnDataProvider method), 226
[get_data\(\)](#) (galaxy.visualization.data_providers.phyloviz.PhylovizDataProvider method), 227
[get_data\(\)](#) (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
[get_data_inputs\(\)](#) (galaxy.workflow.modules.InputModule method), 345
[get_data_inputs\(\)](#) (galaxy.workflow.modules.PauseModule method), 346
[get_data_inputs\(\)](#) (galaxy.workflow.modules.ToolModule method), 347
[get_data_inputs\(\)](#) (galaxy.workflow.modules.WorkflowModule method), 348
[get_data_outputs\(\)](#) (galaxy.workflow.modules.InputDataCollectionModule method), 345
[get_data_outputs\(\)](#) (galaxy.workflow.modules.InputDataModule method), 345
[get_data_outputs\(\)](#) (galaxy.workflow.modules.PauseModule method), 346
[get_data_outputs\(\)](#) (galaxy.workflow.modules.ToolModule method), 347
[get_data_outputs\(\)](#) (galaxy.workflow.modules.WorkflowModule method), 348
[get_database_engine_options\(\)](#) (in module galaxy.config), 41
[get_database_engine_options\(\)](#) (in module galaxy.webapps.reports.config), 339
[get_dataset\(\)](#) (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
[get_dataset_collection\(\)](#) (galaxy.managers.collections.DatasetCollectionManager method), 150
[get_dataset_collection_instance\(\)](#) (galaxy.managers.collections.DatasetCollectionManager method), 150
[get_dataset_filename\(\)](#) (in module galaxy.tools.imp_exp.export_history), 184
[get_dataset_finish_context\(\)](#) (galaxy.jobs.JobWrapper method), 109
[get_dataset_finish_context\(\)](#) (galaxy.jobs.TaskWrapper method), 111
[get_dataset_metadata_key\(\)](#) (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 67
[get_dataset_state\(\)](#) (galaxy.model.DatasetInstance method), 125
[get_datasources\(\)](#) (galaxy.model.DatasetInstance method), 125
[get_datatype\(\)](#) (galaxy.tools.parameters.grouping.UploadDataset method), 200
[get_datatype_by_extension\(\)](#) (galaxy.datatypes.registry.Registry method), 72
[get_datatype_class_by_name\(\)](#) (galaxy.datatypes.registry.Registry method), 72
[get_datatype_ext\(\)](#) (galaxy.tools.parameters.grouping.UploadDataset method), 200
[get_dbkey\(\)](#) (galaxy.model.DatasetInstance method), 125
[get_diffusion\(\)](#) (galaxy.visualization.genomes.Genomes method), 225
[get_decimal_quality_scores\(\)](#) (galaxy_utils.sequence.fastq.fastqSequencingRead method), 353
[get_decimal_range\(\)](#) (galaxy_utils.sequence.fastq.fastqAggregator method), 350
[get_default_cache\(\)](#) (in module pkg_resources), 357
[get_default_history_by_trans\(\)](#) (galaxy.tools.DataManagerTool method), 172
[get_dependencies\(\)](#) (galaxy.tools.parameters.basic.ColumnListParameter method), 186
[get_dependencies\(\)](#) (galaxy.tools.parameters.basic.DataToolParameter method), 187
[get_dependencies\(\)](#) (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 189
[get_dependencies\(\)](#) (galaxy.tools.parameters.basic.SelectToolParameter method), 194
[get_dependencies\(\)](#) (galaxy.tools.parameters.basic.ToolParameter method), 196

[get_dependency_name\(\) \(galaxy.tools.parameters.dynamic_options.DynamicOptions method\), 197](#)
[get_dependency_name\(\) \(galaxy.tools.parameters.dynamic_options.DynamicOptions method\), 198](#)
[get_dependency_name\(\) \(galaxy.tools.parameters.dynamic_options.DynamicOptions method\), 198](#)
[get_dependency_name\(\) \(galaxy.tools.parameters.dynamic_options.DynamicOptions method\), 199](#)
[get_dependency_names\(\) \(galaxy.tools.parameters.dynamic_options.DynamicOptions method\), 197](#)
[get_destination\(\) \(galaxy.jobs.JobConfiguration method\), 107](#)
[get_destinations\(\) \(galaxy.jobs.JobConfiguration method\), 107](#)
[get_disabled_str\(\) \(galaxy.web.form_builder.BaseField method\), 229](#)
[get_disk_size\(\) \(galaxy.model.History method\), 129](#)
[get_disk_size_bytes \(galaxy.model.History attribute\), 129](#)
[get_disk_usage\(\) \(galaxy.model.GalaxySession method\), 128](#)
[get_disk_usage\(\) \(galaxy.model.User method\), 142](#)
[get_disk_usage\(\) \(galaxy.webapps.reports.controllers.system_status.SystemStatus method\), 343](#)
[get_display_application\(\) \(galaxy.datatypes.data.Data method\), 48](#)
[get_display_applications\(\) \(galaxy.model.DatasetInstance method\), 125](#)
[get_display_applications_by_dataset\(\) \(galaxy.datatypes.data.Data method\), 48](#)
[get_display_apps\(\) \(galaxy.managers.hdas.HDAManager method\), 153](#)
[get_display_apps\(\) \(galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method\), 237](#)
[get_display_label\(\) \(galaxy.datatypes.data.Data method\), 48](#)
[get_display_links\(\) \(galaxy.datatypes.data.Data method\), 48](#)
[get_display_name\(\) \(galaxy.model.HasName method\), 128](#)
[get_display_name\(\) \(galaxy.webapps.galaxy.controllers.visualization.VisualizationMixin method\), 335](#)
[get_display_sites\(\) \(galaxy.datatypes.registry.Registry method\), 72](#)
[get_display_text\(\) \(galaxy.web.form_builder.HistoryField method\), 231](#)
[get_display_text\(\) \(galaxy.web.form_builder.LibraryField method\), 231](#)
[get_display_text\(\) \(galaxy.web.form_builder.WorkflowMappingField method\), 233](#)
[get_display_types\(\) \(galaxy.datatypes.data.Data method\), 48](#)
[get_display_name\(\) \(galaxy.datatypes.display_applications.application.DisplayApplication method\), 91](#)
[get_distribution\(\) \(in module pkg_resources\), 357](#)
[get_editor_iframe\(\) \(galaxy.webapps.galaxy.controllers.page.PageController method\), 318](#)
[get_elements\(\) \(galaxy.util.topsort.CycleError method\), 318](#)
[get_embed_html_async\(\) \(galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method\), 304](#)
[get_embed_html_async\(\) \(galaxy.webapps.galaxy.controllers.page.PageController method\), 318](#)
[get_embed_html_async\(\) \(galaxy.webapps.galaxy.controllers.workflow.WorkflowController method\), 337](#)
[get_empty_field_by_name\(\) \(galaxy.tools.data.ToolDataTable method\), 182](#)
[get_entry\(\) \(galaxy.tools.data.TabularToolDataTable method\), 181](#)
[get_entry_info\(\) \(in module pkg_resources\), 357](#)
[get_entry_info\(\) \(pkg_resources.Distribution method\), 361](#)
[get_entry_map\(\) \(in module pkg_resources\), 357](#)
[get_entry_map\(\) \(pkg_resources.Distribution method\), 361](#)
[get_env\(\) \(in module galaxy.eggs\), 95](#)
[get_env_setup_clause\(\) \(galaxy.jobs.JobWrapper method\), 109](#)
[get_errors\(\) \(galaxy.workflow.modules.ToolModule method\), 347](#)
[get_errors\(\) \(galaxy.workflow.modules.WorkflowModule method\), 347](#)
[get_estimated_display_viewport\(\) \(galaxy.datatypes.interval.BedGraph method\), 61](#)
[get_estimated_display_viewport\(\) \(galaxy.datatypes.interval.CustomTrack method\), 61](#)
[get_estimated_display_viewport\(\) \(galaxy.datatypes.interval.Gff method\), 62](#)
[get_estimated_display_viewport\(\) \(galaxy.datatypes.interval.Interval method\), 65](#)
[get_estimated_display_viewport\(\) \(galaxy.datatypes.interval.Wiggle method\), 66](#)
[get_exit_code\(\) \(galaxy.jobs.TaskWrapper method\), 111](#)
[get_external_output_metadata\(\) \(galaxy.model.Job method\), 132](#)
[get_external_output_metadata\(\) \(galaxy.model.Task method\), 140](#)

[get_external_service\(\)](#) (galaxy.model.RequestType method), 138
[get_external_service_type\(\)](#) (galaxy.model.ExternalService method), 127
[get_external_service_type\(\)](#) (galaxy.webapps.galaxy.controllers.external_service_from_dataset method), 306
[get_external_services_for_manual_data_transfer\(\)](#) (galaxy.model.RequestType method), 138
[get_externally_referenced_paths\(\)](#) (galaxy.tools.Tool class method), 172
[get_extra_files_path\(\)](#) (galaxy.model.Dataset method), 123
[get_fasta_header\(\)](#) (in module galaxy.tools.util.maf_utilities), 211
[get_field_by_name_for_value\(\)](#) (galaxy.tools.parameters.dynamic_options.DynamicOptions method), 197
[get_fields\(\)](#) (galaxy.tools.data.TabularToolDataTable method), 181
[get_fields\(\)](#) (galaxy.tools.parameters.dynamic_options.DynamicOptions method), 197
[get_fields_by_value\(\)](#) (galaxy.tools.parameters.dynamic_options.DynamicOptions method), 198
[get_file_base_name\(\)](#) (galaxy.tools.parameters.grouping.UploadDataset method), 200
[get_file_contents\(\)](#) (galaxy.webapps.galaxy.controllers.admin_tools_admin_tools method), 302
[get_file_details\(\)](#) (galaxy.webapps.galaxy.controllers.requests_admin_requests_admin method), 323
[get_file_name\(\)](#) (galaxy.model.Dataset method), 123
[get_file_name\(\)](#) (galaxy.model.DatasetInstance method), 125
[get_file_peek\(\)](#) (galaxy.datatypes.genetics.RexpBase method), 53
[get_file_peek\(\)](#) (in module galaxy.datatypes.data), 51
[get_file_size\(\)](#) (in module galaxy.util), 214
[get_file_type\(\)](#) (galaxy.tools.parameters.grouping.UploadDataset method), 200
[get_filename\(\)](#) (galaxy.objectstore.DiskObjectStore method), 157
[get_filename\(\)](#) (galaxy.objectstore.NestedObjectStore method), 158
[get_filename\(\)](#) (galaxy.objectstore.ObjectStore method), 159
[get_filter\(\)](#) (galaxy.web.framework.helpers.grids.CommunityTagsColumn method), 246
[get_filter\(\)](#) (galaxy.web.framework.helpers.grids.IndividualTagsColumn method), 249
[get_filter\(\)](#) (galaxy.web.framework.helpers.grids.TextColumn method), 252
[get_folder_dict\(\)](#) (galaxy.managers.folders.FolderManager method), 152
[get_form_template\(\)](#) (in module galaxy.jobs.actions.post), 116
[get_form_values\(\)](#) (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 237
[get_forms\(\)](#) (galaxy.jobs.actions.post.ActionBox class method), 114
[get_from_dataset_service\(\)](#) (in module galaxy.web.framework.middleware.profile), 252
[get_galaxy_session\(\)](#) (galaxy.jobs.deferred.FakeTrans method), 116
[get_galaxy_session\(\)](#) (galaxy.webapps.demo_sequencer.framework.DemoWorkflow method), 258
[get_group\(\)](#) (galaxy.web.base.controller.BaseController method), 235
[get_group\(\)](#) (in module galaxy.web.base.controllers.admin), 241
[get_handler\(\)](#) (galaxy.jobs.JobConfiguration method), 107
[get_handler\(\)](#) (galaxy.model.Job method), 132
[get_hda_dict\(\)](#) (galaxy.managers.hdas.HDAManager method), 153
[get_hda_dict\(\)](#) (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
[get_hda_dict_with_error\(\)](#) (galaxy.managers.hdas.HDAManager method), 153
[get_hda_dict_with_error\(\)](#) (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
[get_hda_job\(\)](#) (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
[get_hda_list\(\)](#) (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
[get_hda_or_ldda\(\)](#) (galaxy.web.base.controller.UsesVisualizationMixin method), 239
[get_hda_state_counts\(\)](#) (galaxy.web.base.controller.UsesHistoryMixin method), 238
[get_hda_summary_dicts\(\)](#) (galaxy.web.base.controller.UsesHistoryMixin method), 238
[get_headers\(\)](#) (in module galaxy.datatypes.sniff), 78
[get_history\(\)](#) (galaxy.web.base.controller.UsesHistoryMixin method), 238
[get_history_dataset_association\(\)](#) (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
[get_history_dataset_association_from_ids\(\)](#) (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
[get_history_datasets\(\)](#) (galaxy.web.base.controller.UsesHistoryMixin method), 238
[get_history_dict\(\)](#) (galaxy.managers.histories.HistoryManager method), 154

`get_history_dict()` (galaxy.web.base.controller.UsesHistoryMixin method), 238
`get_hook()` (galaxy.tools.Tool method), 173
`get_html()` (galaxy.datatypes.metadata.ColumnParameter method), 66
`get_html()` (galaxy.datatypes.metadata.DBKeyParameter method), 66
`get_html()` (galaxy.datatypes.metadata.FileParameter method), 67
`get_html()` (galaxy.datatypes.metadata.MetadataParameter method), 68
`get_html()` (galaxy.datatypes.metadata.PythonObjectParameter method), 69
`get_html()` (galaxy.datatypes.metadata.RangeParameter method), 69
`get_html()` (galaxy.datatypes.metadata.SelectParameter method), 69
`get_html()` (galaxy.model.UserAddress method), 142
`get_html()` (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 189
`get_html()` (galaxy.tools.parameters.basic.ToolParameter method), 196
`get_html()` (galaxy.web.form_builder.AddressField method), 229
`get_html()` (galaxy.web.form_builder.BaseField method), 229
`get_html()` (galaxy.web.form_builder.CheckboxField method), 229
`get_html()` (galaxy.web.form_builder.DrillDownField method), 231
`get_html()` (galaxy.web.form_builder.FileField method), 231
`get_html()` (galaxy.web.form_builder.FTPFileField method), 231
`get_html()` (galaxy.web.form_builder.HiddenField method), 231
`get_html()` (galaxy.web.form_builder.HistoryField method), 231
`get_html()` (galaxy.web.form_builder.LibraryField method), 231
`get_html()` (galaxy.web.form_builder.PasswordField method), 232
`get_html()` (galaxy.web.form_builder.SelectField method), 232
`get_html()` (galaxy.web.form_builder.SwitchingSelectField method), 233
`get_html()` (galaxy.web.form_builder.TextArea method), 233
`get_html()` (galaxy.web.form_builder.TextField method), 233
`get_html()` (galaxy.web.form_builder.WorkflowField method), 233
`get_html()` (galaxy.web.form_builder.WorkflowMappingField method), 233
`get_html_by_name()` (galaxy.datatypes.metadata.MetadataCollection method), 68
`get_html_checkboxes()` (galaxy.web.form_builder.SelectField method), 233
`get_html_default()` (galaxy.web.form_builder.SelectField method), 233
`get_html_field()` (galaxy.datatypes.metadata.ColumnParameter method), 66
`get_html_field()` (galaxy.datatypes.metadata.DBKeyParameter method), 66
`get_html_field()` (galaxy.datatypes.metadata.FileParameter method), 67
`get_html_field()` (galaxy.datatypes.metadata.MetadataParameter method), 68
`get_html_field()` (galaxy.datatypes.metadata.PythonObjectParameter method), 69
`get_html_field()` (galaxy.datatypes.metadata.RangeParameter method), 69
`get_html_field()` (galaxy.datatypes.metadata.SelectParameter method), 69
`get_html_field()` (galaxy.tools.parameters.basic.BaseURLToolParameter method), 185
`get_html_field()` (galaxy.tools.parameters.basic.BooleanToolParameter method), 186
`get_html_field()` (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 187
`get_html_field()` (galaxy.tools.parameters.basic.DataToolParameter method), 187
`get_html_field()` (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 190
`get_html_field()` (galaxy.tools.parameters.basic.FileToolParameter method), 190
`get_html_field()` (galaxy.tools.parameters.basic.FloatToolParameter method), 191
`get_html_field()` (galaxy.tools.parameters.basic.FTPFileToolParameter method), 190
`get_html_field()` (galaxy.tools.parameters.basic.HiddenDataToolParameter method), 192
`get_html_field()` (galaxy.tools.parameters.basic.HiddenToolParameter method), 192
`get_html_field()` (galaxy.tools.parameters.basic.IntegerToolParameter method), 193
`get_html_field()` (galaxy.tools.parameters.basic.LibraryDatasetToolParameter method), 193
`get_html_field()` (galaxy.tools.parameters.basic.SelectToolParameter method), 195
`get_html_field()` (galaxy.tools.parameters.basic.TextToolParameter method), 195
`get_html_field()` (galaxy.tools.parameters.basic.ToolParameter method), 196
`get_html_radio()` (galaxy.web.form_builder.SelectField method), 233
`get_id()` (galaxy.model.Job method), 132
`get_id()` (galaxy.model.Task method), 140

[get_id\(\)](#) (in module `galaxy.webapps.galaxy.api.genomes`), 13, 269
[get_id_col_in_item_tag_assoc_table\(\)](#) (`galaxy.tags.tag_handler.TagHandler` method), 169
[get_id_tag\(\)](#) (`galaxy.jobs.JobWrapper` method), 109
[get_id_tag\(\)](#) (`galaxy.jobs.TaskWrapper` method), 111
[get_id_tag\(\)](#) (`galaxy.model.Job` method), 132
[get_id_tag\(\)](#) (`galaxy.model.Task` method), 140
[get_ids\(\)](#) (`galaxy.web.framework.helpers.grids.Grid` method), 247
[get_image_ext\(\)](#) (in module `galaxy.datatypes.util.image_util`), 94
[get_imported\(\)](#) (`galaxy.model.Job` method), 132
[get_importer\(\)](#) (in module `pkg_resources`), 366
[get_inaccessible_hda_dict\(\)](#) (`galaxy.managers.hdas.HDAManager` method), 153
[get_inaccessible_hda_dict\(\)](#) (`galaxy.web.base.controller.UsesHistoryDatasetAssociationMethod` method), 237
[get_incoming_value\(\)](#) (in module `galaxy.tools`), 177
[get_index\(\)](#) (`galaxy.tools.test.RootParamContext` method), 178
[get_info\(\)](#) (`galaxy.model.Job` method), 132
[get_info\(\)](#) (`galaxy.model.LibraryDataset` method), 134
[get_info\(\)](#) (`galaxy.model.Task` method), 140
[get_info_association\(\)](#) (`galaxy.model.Library` method), 134
[get_info_association\(\)](#) (`galaxy.model.LibraryDatasetDatasetAssociation` method), 135
[get_info_association\(\)](#) (`galaxy.model.LibraryFolder` method), 136
[get_inital_values\(\)](#) (`galaxy.datatypes.display_applications.application.DisplayApplicationLink` method), 91
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.BaseURLToolParameter` method), 185
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.BooleanToolParameter` method), 186
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.ColumnListParameter` method), 186
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.DataToolParameter` method), 187
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.DrillDownSelectToolParameter` method), 190
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.FileToolParameter` method), 190
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.FloatToolParameter` method), 191
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.FTPFileToolParameter` method), 190
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.HiddenDatasetToolParameter` method), 192
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.HiddenToolParameter` method), 192
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.IntegerToolParameter` method), 193
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.LibraryDatasetToolParameter` method), 193
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.SelectToolParameter` method), 195
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.TextToolParameter` method), 195
[get_initial_value\(\)](#) (`galaxy.tools.parameters.basic.ToolParameter` method), 196
[get_initial_value\(\)](#) (`galaxy.tools.parameters.grouping.Conditional` method), 199
[get_initial_value\(\)](#) (`galaxy.tools.parameters.grouping.Group` method), 200
[get_initial_value\(\)](#) (`galaxy.tools.parameters.grouping.Repeat` method), 200
[get_initial_value\(\)](#) (`galaxy.tools.parameters.grouping.UploadDataset` method), 200
[get_initial_value_from_history_prevent_repeats\(\)](#) (`galaxy.tools.parameters.basic.DataToolParameter` method), 187
[get_initial_value_from_history_prevent_repeats\(\)](#) (`galaxy.tools.parameters.basic.ToolParameter` method), 196
[get_input_dataset_fnames\(\)](#) (`galaxy.jobs.JobWrapper` method), 109
[get_input_datasets\(\)](#) (`galaxy.model.Job` method), 132
[get_input_files\(\)](#) (`galaxy.jobs.runners.lwr.LwrJobRunner` method), 121
[get_input_fnames\(\)](#) (`galaxy.jobs.JobWrapper` method), 109
[get_input_tool_params\(\)](#) (`galaxy.jobs.JobWrapper` method), 109
[get_interesting_stack_frame\(\)](#) (`galaxy.util.heartbeat.Heartbeat` method), 218
[get_invocation\(\)](#) (`galaxy.managers.workflows.WorkflowsManager` method), 156
[get_invocation_step\(\)](#) (`galaxy.managers.workflows.WorkflowsManager` method), 156
[get_ip_address\(\)](#) (`galaxy.web.base.controller.SharableMixin` method), 236
[get_item\(\)](#) (`galaxy.webapps.galaxy.controllers.history.HistoryController` method), 310
[get_item\(\)](#) (`galaxy.webapps.galaxy.controllers.page.PageController` method), 318
[get_item\(\)](#) (`galaxy.webapps.galaxy.controllers.visualization.VisualizationController` method), 333
[get_item\(\)](#) (`galaxy.webapps.galaxy.controllers.workflow.WorkflowController` method), 337

get_item_actions() (galaxy.security.GalaxyRBACAgent method), 164
 get_item_actions() (galaxy.security.RBACAgent method), 167
 get_item_and_stuff() (galaxy.web.base.controller.UsesFormDefinitionMixin method), 237
 get_item_annotation_obj() (galaxy.model.item_attrs.UsesAnnotations method), 146
 get_item_annotation_str() (galaxy.model.item_attrs.UsesAnnotations method), 146
 get_item_content_async() (galaxy.web.base.controller.SharableMixin method), 236
 get_item_content_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
 get_item_content_async() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
 get_item_content_async() (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
 get_item_content_async() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
 get_item_extended_metadata_obj() (galaxy.web.base.controller.UsesExtendedMetadataMixin method), 236
 get_iterator() (galaxy.visualization.data_providers.basic.BaseDataProvider method), 226
 get_job() (galaxy.jobs.JobWrapper method), 109
 get_job() (galaxy.jobs.TaskWrapper method), 111
 get_job() (galaxy.model.Job method), 132
 get_job() (galaxy.model.Task method), 140
 get_job() (in module galaxy.webapps.reports.controllers.jobs), 341
 get_job_destination() (galaxy.jobs.mapper.JobRunnerMapper method), 113
 get_job_destination() (galaxy.tools.Tool method), 173
 get_job_file() (galaxy.jobs.runners.BaseJobRunner method), 118
 get_job_handler() (galaxy.tools.Tool method), 173
 get_job_runner() (galaxy.jobs.JobWrapper method), 109
 get_job_runner_external_id() (galaxy.model.Job method), 132
 get_job_runner_external_id() (galaxy.model.Task method), 140
 get_job_runner_name() (galaxy.model.Job method), 132
 get_job_runner_name() (galaxy.model.Task method), 140
 get_job_runner_plugins() (galaxy.jobs.JobConfiguration method), 107
 get_job_runner_url() (galaxy.jobs.JobWrapper method), 109
 get_job_tool_configurations() (galaxy.jobs.JobConfiguration method), 107
 get_tool_definition() (galaxy.tools.parameters.basic.BaseURLToolParameter method), 185
 get_label() (galaxy.tools.parameters.basic.HiddenToolParameter method), 192
 get_label() (galaxy.tools.parameters.basic.ToolParameter method), 196
 get_last_check() (galaxy.model.DeferredJob method), 127
 get_latest_installable_revision() (galaxy.webapps.galaxy.api.tool_shed_repositories.ToolShedRepository method), 34, 289
 get_legacy_sites_by_build() (galaxy.datatypes.registry.Registry method), 72
 get_legal_values() (galaxy.tools.parameters.basic.ColumnListParameter method), 187
 get_legal_values() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 190
 get_legal_values() (galaxy.tools.parameters.basic.GenomeBuildParameter method), 192
 get_legal_values() (galaxy.tools.parameters.basic.SelectToolParameter method), 195
 get_legal_roles() (galaxy.security.GalaxyRBACAgent method), 164
 get_legitimate_roles() (galaxy.security.RBACAgent method), 167
 get_length_counts() (galaxy_utils.sequence.fastq.fastqAggregator method), 350
 get_library() (galaxy.web.base.controller.UsesLibraryMixin method), 238
 get_library_dataset() (galaxy.web.base.controller.UsesLibraryMixinItems method), 238
 get_library_dataset_dataset_association() (galaxy.web.base.controller.UsesLibraryMixinItems method), 238
 get_library_dict() (galaxy.managers.libraries.LibraryManager method), 155
 get_library_folder() (galaxy.web.base.controller.UsesLibraryMixinItems method), 238
 get_link() (galaxy.datatypes.display_applications.application.DisplayApplication method), 90
 get_link() (galaxy.web.framework.helpers.grids.GridColumn method), 248
 get_link() (galaxy.web.framework.helpers.grids.PublicURLColumn method), 250
 get_link() (galaxy.web.framework.helpers.grids.SharingStatusColumn method), 251
 get_link() (galaxy.webapps.galaxy.controllers.history.HistoryListGrid.HistoryListGrid method), 311
 get_link() (galaxy.webapps.galaxy.controllers.visualization.NameColumn method), 332

[get_loaded_tools_by_lineage\(\)](#) (galaxy.tools.ToolBox method), 176
[get_manage_permissions_roles\(\)](#) (galaxy.model.Dataset method), 123
[get_manage_permissions_roles\(\)](#) (galaxy.model.LibraryDatasetDatasetAssociation method), 135
[get_manage_roles\(\)](#) (galaxy.managers.libraries.LibraryManager method), 155
[get_mapping_by_name\(\)](#) (galaxy.tools.parameters.sanitize.ToolParameterSanitizer class method), 205
[get_max_optional_metadata_filesize\(\)](#) (galaxy.datatypes.data.Data method), 48
[get_max_read_length\(\)](#) (galaxy_utils.sequence.fastq.fastqAggregator method), 350
[get_message\(\)](#) (galaxy.webapps.demo_sequencer.framework.DemoWebTransaction method), 258
[get_message_for_no_shed_tool_config\(\)](#) (in module galaxy.webapps.galaxy.api.tool_shed_repositories), 36, 291
[get_metadata\(\)](#) (galaxy.model.DatasetInstance method), 125
[get_metadata\(\)](#) (pkg_resources.FileMetadata method), 364
[get_metadata\(\)](#) (pkg_resources.IMetadataProvider method), 363
[get_metadata\(\)](#) (pkg_resources.NullProvider method), 365
[get_metadata_dataset\(\)](#) (galaxy.model.DatasetInstance method), 125
[get_metadata_file\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
[get_metadata_lines\(\)](#) (pkg_resources.FileMetadata method), 364
[get_metadata_lines\(\)](#) (pkg_resources.IMetadataProvider method), 363
[get_metadata_lines\(\)](#) (pkg_resources.NullProvider method), 365
[get_mime\(\)](#) (galaxy.datatypes.binary.Binary method), 44
[get_mime\(\)](#) (galaxy.datatypes.data.Data method), 48
[get_mime\(\)](#) (galaxy.datatypes.data.Text method), 50
[get_mime\(\)](#) (galaxy.datatypes.genetics.GenomeGraphs method), 52
[get_mime\(\)](#) (galaxy.datatypes.genetics.RexpBase method), 53
[get_mime\(\)](#) (galaxy.datatypes.genetics.Rgenetics method), 54
[get_mime\(\)](#) (galaxy.datatypes.genetics.rgTabList method), 56
[get_mime\(\)](#) (galaxy.datatypes.images.Gmaj method), 56
[get_mime\(\)](#) (galaxy.datatypes.images.Html method), 57
[get_mime\(\)](#) (galaxy.model.DatasetInstance method), 125
[get_mimetype_by_extension\(\)](#) (galaxy.datatypes.registry.Registry method), 72
[get_modify_roles\(\)](#) (galaxy.managers.libraries.LibraryManager method), 155
[get_monitor_id\(\)](#) (in module galaxy.webapps.reports.controllers.jobs), 341
[get_most_recently_used_tool_async\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), 328
[get_output_filenames\(\)](#) (galaxy.jobs.JobWrapper method), 109
[get_name\(\)](#) (galaxy.model.LibraryDataset method), 134
[get_name\(\)](#) (galaxy.workflow.modules.ToolModuleAggregator method), 347
[get_name\(\)](#) (galaxy.workflow.modules.WorkflowModuleAggregator method), 348
[get_name_and_link_async\(\)](#) (galaxy.web.base.controller.SharableMixin method), 236
[get_name_and_link_async\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
[get_name_and_link_async\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[get_name_and_link_async\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[get_name_and_link_async\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[get_name_and_link_async\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
[get_named_fields_list\(\)](#) (galaxy.tools.data.TabularToolDataTable method), 181
[get_native_spec\(\)](#) (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 120
[get_new_guid\(\)](#) (galaxy.web.security.SecurityHelper method), 254
[get_new_module_info\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
[get_new_track_config\(\)](#) (galaxy.web.base.controller.UsesVisualizationMixin method), 239
[get_object\(\)](#) (galaxy.web.base.controller.BaseAPIController method), 234
[get_object\(\)](#) (galaxy.web.base.controller.BaseController method), 235
[get_object\(\)](#) (galaxy.web.base.controller.BaseUIController method), 235
[get_object\(\)](#) (in module galaxy.managers.base), 148
[get_object_url\(\)](#) (galaxy.objectstore.DiskObjectStore method), 157

get_object_url() (galaxy.objectstore.NestedObjectStore method), 158
 get_object_url() (galaxy.objectstore.ObjectStore method), 159
 get_old_display_applications() (galaxy.managers.hdas.HDAManager method), 153
 get_old_display_applications() (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
 get_open_tempfile() (galaxy.tools.util.maf_utilities.TempFileHandler filename() method), 210
 get_options() (galaxy.tools.parameters.basic.ColumnListParameter method), 187
 get_options() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 190
 get_options() (galaxy.tools.parameters.basic.GenomeBuildParameter method), 192
 get_options() (galaxy.tools.parameters.basic.SelectToolParameter method), 195
 get_options() (galaxy.tools.parameters.dynamic_options.DynamicOptions method), 198
 get_or_create_api_key() (galaxy.managers.api_keys.ApiKeyManager method), 148
 get_oriented_chopped_blocks_for_region() (in module galaxy.tools.util.maf_utilities), 211
 get_oriented_chopped_blocks_with_index_offset_for_region() (in module galaxy.tools.util.maf_utilities), 211
 get_output_datasets() (galaxy.model.Job method), 132
 get_output_destination() (galaxy.jobs.JobWrapper method), 109
 get_output_destination() (galaxy.jobs.TaskWrapper method), 111
 get_output_file_id() (galaxy.jobs.JobWrapper method), 109
 get_output_file_id() (galaxy.jobs.TaskWrapper method), 111
 get_output_filenames_by_dataset() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 67
 get_output_files() (galaxy.jobs.runners.lwr.LwrJobRunner method), 121
 get_output_fnames() (galaxy.jobs.JobWrapper method), 109
 get_output_hdas_and_fnames() (galaxy.jobs.JobWrapper method), 109
 get_output_library_datasets() (galaxy.model.Job method), 132
 get_output_name() (galaxy.tools.actions.DefaultToolAction method), 179
 get_output_sizes() (galaxy.jobs.JobWrapper method), 109
 get_page() (galaxy.webapps.galaxy.controllers.page.PageController method), 318
 get_paired_identifier() (galaxy_utils.sequence.fastq.fastqJoiner method), 352
 get_pairlist() (galaxy.util.topsort.CycleError method), 224
 get_panel_section() (galaxy.tools.Tool method), 173
 get_parallelism() (galaxy.jobs.JobWrapper method), 109
 get_param() (galaxy.tools.Tool method), 173
 get_param_dict() (galaxy.jobs.JobWrapper method), 109
 get_param_mixin() (galaxy.jobs.TaskWrapper method), 111
 get_param_filename() (galaxy.model.Job method), 132
 get_param_html_map() (galaxy.tools.Tool method), 173
 get_param_name_by_url() (galaxy.datatypes.display_applications.application.PopulatedDisplayApplication method), 91
 get_param_value() (galaxy.datatypes.display_applications.application.PopulatedDisplayApplication method), 91
 get_param_values() (galaxy.model.Job method), 132
 get_param_values() (galaxy.model.Task method), 140
 get_parameters() (galaxy.model.Job method), 132
 get_parameters() (galaxy.model.Task method), 140
 get_params() (galaxy.model.Job method), 132
 get_params() (galaxy.web.params.BaseParamParser method), 234
 get_parent() (galaxy.datatypes.metadata.MetadataCollection method), 68
 get_partial() (galaxy.util.topsort.CycleError method), 224
 get_path_paste_uploaded_datasets() (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315
 get_peek() (galaxy.datatypes.genetics.RexpBase method), 54
 get_percent() (galaxy.quota.NoQuotaAgent method), 161
 get_percent() (galaxy.quota.QuotaAgent method), 161
 get_permissions() (galaxy.security.GalaxyRBACAgent method), 165
 get_permissions() (galaxy.security.RBACAgent method), 167
 get_permissions() (galaxy.webapps.galaxy.api.folders.FoldersController method), 12, 267
 get_permissions() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 25, 281
 get_permitted_actions() (in module galaxy.security), 168
 get_permitted_libraries() (galaxy.security.GalaxyRBACAgent method), 165
 get_permitted_libraries() (galaxy.security.RBACAgent method), 167
 get_phecols() (galaxy.datatypes.genetics.RexpBase method), 54
 get_pheno() (galaxy.datatypes.genetics.RexpBase method), 54
 get_platform() (in module pkg_resources), 362

[get_platforms\(\)](#) (galaxy.eggs.dist.DistScrambleCrate method), 96
[get_post_job_actions\(\)](#) (galaxy.model.Job method), 132
[get_post_job_actions\(\)](#) (galaxy.workflow.modules.ToolModule method), 347
[get_precreated_dataset\(\)](#) (in module galaxy.tools.actions.upload_common), 180
[get_precreated_datasets\(\)](#) (in module galaxy.tools.actions.upload_common), 180
[get_pred_counts\(\)](#) (galaxy.util.topsort.CycleError method), 224
[get_preds\(\)](#) (galaxy.util.topsort.CycleError method), 224
[get_prepare_input_files_cmd\(\)](#) (galaxy.model.Task method), 140
[get_private_user_role\(\)](#) (galaxy.security.GalaxyRBACAgent method), 165
[get_private_user_role\(\)](#) (galaxy.security.RBACAgent method), 167
[get_provider\(\)](#) (in module pkg_resources), 357
[get_published_visualizations\(\)](#) (galaxy.web.base.controller.UsesVisualizationMixin method), 239
[get_quota\(\)](#) (galaxy.quota.NoQuotaAgent method), 161
[get_quota\(\)](#) (galaxy.quota.QuotaAgent method), 161
[get_quota\(\)](#) (galaxy.web.base.controller.UsesQuotaMixin method), 238
[get_quota\(\)](#) (in module galaxy.web.base.controllers.admin), 241
[get_quota_params\(\)](#) (galaxy.web.params.QuotaParamParser method), 234
[get_random_bytes\(\)](#) (in module galaxy.web.security), 254
[get_raw_data\(\)](#) (galaxy.datatypes.data.Data method), 48
[get_raw_data\(\)](#) (galaxy.model.DatasetInstance method), 125
[get_read_count\(\)](#) (galaxy_utils.sequence.fastq.fastqAggregator method), 350
[get_read_count_for_column\(\)](#) (galaxy_utils.sequence.fastq.fastqAggregator method), 350
[get_ref\(\)](#) (galaxy.external_services.service.ExternalServiceActionsGroup method), 102
[get_ref\(\)](#) (galaxy.tools.parameters.output.ToolOutputActionConditionalWhen method), 204
[get_region_alignment\(\)](#) (in module galaxy.tools.util.maf_utilities), 211
[get_reports\(\)](#) (galaxy.util.debugging.SimpleProfiler method), 217
[get_request\(\)](#) (in module galaxy.webapps.reports.controllers.sample_tracking), 342
[get_required_enctype\(\)](#) (galaxy.tools.parameters.basic.FileToolParameter method), 190
[get_required_enctype\(\)](#) (galaxy.tools.parameters.basic.ToolParameter method), 196
[get_resource_filename\(\)](#) (pkg_resources.IResourceProvider method), 364
[get_resource_filename\(\)](#) (pkg_resources.NullProvider method), 365
[get_resource_filename\(\)](#) (pkg_resources.ZipProvider method), 365
[get_resource_group\(\)](#) (galaxy.jobs.JobToolConfiguration method), 108
[get_resource_stream\(\)](#) (pkg_resources.DefaultProvider method), 365
[get_resource_stream\(\)](#) (pkg_resources.IResourceProvider method), 364
[get_resource_stream\(\)](#) (pkg_resources.NullProvider method), 365
[get_resource_string\(\)](#) (pkg_resources.IResourceProvider method), 364
[get_resource_string\(\)](#) (pkg_resources.NullProvider method), 365
[get_role\(\)](#) (galaxy.web.base.controller.BaseController method), 235
[get_role\(\)](#) (in module galaxy.web.base.controllers.admin), 241
[get_roles_for_action\(\)](#) (galaxy.security.GalaxyRBACAgent method), 165
[get_runtime_input_dicts\(\)](#) (galaxy.workflow.modules.InputModule method), 345
[get_runtime_input_dicts\(\)](#) (galaxy.workflow.modules.PauseModule method), 346
[get_runtime_input_dicts\(\)](#) (galaxy.workflow.modules.ToolModule method), 347
[get_runtime_input_dicts\(\)](#) (galaxy.workflow.modules.WorkflowModule method), 348
[get_runtime_inputs\(\)](#) (galaxy.workflow.modules.InputDataCollectionModule method), 345
[get_runtime_inputs\(\)](#) (galaxy.workflow.modules.InputDataModule method), 345
[get_runtime_inputs\(\)](#) (galaxy.workflow.modules.PauseModule method), 346
[get_runtime_inputs\(\)](#) (galaxy.workflow.modules.WorkflowModule method), 348
[get_runtime_state\(\)](#) (galaxy.workflow.modules.InputModule method), 345
[get_runtime_state\(\)](#) (galaxy.workflow.modules.PauseModule method), 346
[get_sample\(\)](#) (galaxy.model.Request method), 137
[get_saved_form\(\)](#) (galaxy.webapps.galaxy.controllers.forms.Forms method), 308
[get_score_at_position_for_column\(\)](#) (galaxy_utils.sequence.fastq.fastqAggregator method), 350

get_score_list_for_column() (galaxy_utils.sequence.fastq.fastqAggregator method), 350
 get_score_max_for_column() (galaxy_utils.sequence.fastq.fastqAggregator method), 350
 get_score_min_for_column() (galaxy_utils.sequence.fastq.fastqAggregator method), 350
 get_score_sum_for_column() (galaxy_utils.sequence.fastq.fastqAggregator method), 351
 get_selected() (galaxy.web.form_builder.SelectField method), 233
 get_sequence() (galaxy.tools.util.maf_utilities.RegionAlignment method), 210
 get_sequence() (galaxy.tools.util.maf_utilities.SplicedAlignment method), 210
 get_sequence() (galaxy_utils.sequence.fastq.fastqCSSangerReads method), 351
 get_sequence() (galaxy_utils.sequence.fastq.fastqSequencingRead method), 353
 get_sequence_reverse_complement() (galaxy.tools.util.maf_utilities.RegionAlignment method), 210
 get_sequence_reverse_complement() (galaxy.tools.util.maf_utilities.SplicedAlignment method), 210
 get_sequences_per_file() (galaxy.datatypes.sequence.Sequence static method), 77
 get_server_dir_uploaded_datasets() (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315
 get_session() (galaxy.web.framework.openid_manager.OpenIDManager method), 244
 get_session_id() (galaxy.jobs.JobWrapper method), 109
 get_session_id() (galaxy.jobs.TaskWrapper method), 111
 get_session_id() (galaxy.model.Job method), 132
 get_session_id() (galaxy.model.Task method), 140
 get_sharing_roles() (galaxy.security.GalaxyRBACAgent method), 165
 get_shed_config_dict_by_filename() (galaxy.tools.ToolBox method), 176
 get_short_str() (galaxy.jobs.actions.post.ActionBox class method), 114
 get_short_str() (galaxy.jobs.actions.post.ChangeDatatypeAction class method), 114
 get_short_str() (galaxy.jobs.actions.post.ColumnSetAction class method), 114
 get_short_str() (galaxy.jobs.actions.post.DefaultJobAction class method), 114
 get_short_str() (galaxy.jobs.actions.post.DeleteDatasetAction class method), 115
 get_short_str() (galaxy.jobs.actions.post.DeleteIntermediatesAction class method), 115
 get_short_str() (galaxy.jobs.actions.post.EmailAction class method), 115
 get_short_str() (galaxy.jobs.actions.post.HideDatasetAction class method), 115
 get_short_str() (galaxy.jobs.actions.post.RenameDatasetAction class method), 115
 get_showable_folders() (galaxy.security.GalaxyRBACAgent method), 165
 get_single_filter() (galaxy.web.framework.helpers.grids.BooleanColumn method), 245
 get_single_filter() (galaxy.web.framework.helpers.grids.IntegerColumn method), 249
 get_single_filter() (galaxy.web.framework.helpers.grids.OwnerAnnotation method), 250
 get_single_filter() (galaxy.web.framework.helpers.grids.TextColumn method), 252
 get_size() (galaxy.model.Dataset method), 123
 get_size() (galaxy.model.DatasetInstance method), 125
 get_reported_accessible_library_items() (in module galaxy.webapps.galaxy.controllers.library_common), 316
 get_species_in_block() (in module galaxy.tools.util.maf_utilities), 211
 get_species_in_maf() (in module galaxy.tools.util.maf_utilities), 211
 get_species_names() (galaxy.tools.util.maf_utilities.RegionAlignment method), 210
 get_species_names() (galaxy.tools.util.maf_utilities.SplicedAlignment method), 210
 get_spliced_region_alignment() (in module galaxy.tools.util.maf_utilities), 211
 get_sreg() (galaxy.web.framework.openid_manager.OpenIDManager method), 244
 get_starts_ends_fields_from_gene_bed() (in module galaxy.tools.util.maf_utilities), 212
 get_state() (galaxy.jobs.JobWrapper method), 109
 get_state() (galaxy.jobs.TaskWrapper method), 111
 get_state() (galaxy.jobs.transfer_manager.TransferManager method), 113
 get_state() (galaxy.model.Job method), 132
 get_state() (galaxy.model.Task method), 140
 get_state() (galaxy.workflow.modules.SimpleWorkflowModule method), 346
 get_state() (galaxy.workflow.modules.ToolModule method), 347
 get_state() (galaxy.workflow.modules.WorkflowModule method), 349
 get_static_param_values() (galaxy.tools.Tool method), 173
 get_stderr() (galaxy.model.Task method), 140
 get_stdout() (galaxy.model.Task method), 140

[get_store_usage_percent\(\)](#) (galaxy.objectstore.DiskObjectStore method), [157](#)
[get_store_usage_percent\(\)](#) (galaxy.objectstore.ObjectStore method), [159](#)
[get_stored_workflow\(\)](#) (galaxy.web.base.controller.UsesStoredWorkflowMixin method), [238](#)
[get_stored_workflow_steps\(\)](#) (galaxy.web.base.controller.UsesStoredWorkflowMixin method), [238](#)
[get_succs\(\)](#) (galaxy.util.topsort.CycleError method), [224](#)
[get_suite\(\)](#) (in module galaxy.tools.parameters.validation), [209](#)
[get_suite\(\)](#) (in module galaxy.web.form_builder), [234](#)
[get_summary_statistics_for_column\(\)](#) (galaxy_utils.sequence.fastq.fastqAggregator method), [351](#)
[get_tables\(\)](#) (galaxy.tools.data.ToolDataTableManager method), [182](#)
[get_tag_assoc_class\(\)](#) (galaxy.tags.tag_handler.TagHandler method), [169](#)
[get_tag_by_id\(\)](#) (galaxy.tags.tag_handler.TagHandler method), [169](#)
[get_tag_by_name\(\)](#) (galaxy.tags.tag_handler.TagHandler method), [169](#)
[get_tag_handler\(\)](#) (galaxy.web.base.controller.UsesTagsMixin method), [239](#)
[get_tagging_elt_async\(\)](#) (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), [326](#)
[get_tags_str\(\)](#) (galaxy.tags.tag_handler.TagHandler method), [169](#)
[get_task\(\)](#) (galaxy.jobs.TaskWrapper method), [111](#)
[get_task_runner_external_id\(\)](#) (galaxy.model.Task method), [140](#)
[get_task_runner_name\(\)](#) (galaxy.model.Task method), [140](#)
[get_tasks\(\)](#) (galaxy.model.Job method), [132](#)
[get_template_widgets\(\)](#) (galaxy.model.Library method), [134](#)
[get_template_widgets\(\)](#) (galaxy.model.LibraryDatasetDatasetAssociation method), [135](#)
[get_template_widgets\(\)](#) (galaxy.model.LibraryFolder method), [136](#)
[get_template_widgets\(\)](#) (galaxy.model.RequestType method), [138](#)
[get_template_widgets\(\)](#) (galaxy.model.Sample method), [139](#)
[get_test_fname\(\)](#) (in module galaxy.datatypes.data), [51](#)
[get_test_fname\(\)](#) (in module galaxy.datatypes.sniff), [78](#)
[get_tld\(\)](#) (galaxy.eggs.scramble.ScrambleEgg method), [96](#)
[get_tool\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[get_tool_components\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[get_tool_def\(\)](#) (galaxy.web.base.controller.UsesVisualizationMixin method), [239](#)
[get_tool_dependencies\(\)](#) (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), [302](#)
[get_tool_id\(\)](#) (galaxy.model.Job method), [132](#)
[get_tool_id\(\)](#) (galaxy.workflow.modules.ToolModule method), [347](#)
[get_tool_id\(\)](#) (galaxy.workflow.modules.WorkflowModule method), [349](#)
[get_tool_provided_job_metadata\(\)](#) (galaxy.jobs.JobWrapper method), [109](#)
[get_tool_provided_job_metadata\(\)](#) (galaxy.jobs.TaskWrapper method), [111](#)
[get_tool_resource_parameters\(\)](#) (galaxy.jobs.JobConfiguration method), [107](#)
[get_tool_tags\(\)](#) (galaxy.tags.tag_handler.TagHandler method), [169](#)
[get_tool_version\(\)](#) (galaxy.model.Job method), [132](#)
[get_tool_version\(\)](#) (galaxy.workflow.modules.ToolModule method), [347](#)
[get_toolbox\(\)](#) (galaxy.managers.context.ProvidesAppContext method), [150](#)
[get_toolbox\(\)](#) (galaxy.web.base.controller.BaseController method), [235](#)
[get_tooltip\(\)](#) (galaxy.workflow.modules.ToolModule method), [347](#)
[get_tooltip\(\)](#) (galaxy.workflow.modules.WorkflowModule method), [349](#)
[get_total_job_count_per_destination\(\)](#) (galaxy.jobs.handler.JobHandlerQueue method), [112](#)
[get_total_size\(\)](#) (galaxy.model.Dataset method), [123](#)
[get_total_size\(\)](#) (galaxy.model.DatasetInstance method), [125](#)
[get_track_resolution\(\)](#) (galaxy.datatypes.coverage.LastzCoverage method), [46](#)
[get_track_resolution\(\)](#) (galaxy.datatypes.interval.Interval method), [65](#)
[get_track_resolution\(\)](#) (galaxy.datatypes.interval.Wiggle method), [66](#)
[get_track_window\(\)](#) (galaxy.datatypes.coverage.LastzCoverage method), [46](#)
[get_track_window\(\)](#) (galaxy.datatypes.interval.Interval method), [65](#)
[get_track_window\(\)](#) (galaxy.datatypes.interval.Wiggle method), [66](#)
[get_type\(\)](#) (galaxy.workflow.modules.WorkflowModule method), [349](#)
[get_untransferred_dataset_size\(\)](#) (galaxy.model.Sample method), [139](#)
[get_updated_repository_information\(\)](#) (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), [302](#)

method), 302

get_upload_metadata_params()
(galaxy.datatypes.registry.Registry method), 72

get_uploaded_datasets() (galaxy.tools.parameters.grouping.UploadData method), 200

get_uploaded_datasets() (in module
galaxy.tools.actions.upload_common), 180

get_url_args() (galaxy.web.framework.helpers.grids.DisplayByUsernameAndSlugGridOperation method), 247

get_url_args() (galaxy.web.framework.helpers.grids.GridColumnFilter method), 248

get_url_args() (galaxy.web.framework.helpers.grids.GridOperation method), 249

get_url_args() (galaxy.webapps.galaxy.controllers.visualization.VisualizationListGrid method), 335

get_usage() (galaxy.quota.NoQuotaAgent method), 161

get_user() (galaxy.model.Job method), 132

get_user() (galaxy.web.base.controller.BaseController method), 235

get_user() (in module
galaxy.web.base.controllers.admin), 241

get_user_by_username() (in module
galaxy.web.base.controllers.admin), 241

get_user_histories() (galaxy.web.base.controller.UsesHistoryMixin method), 238

get_user_id() (galaxy.model.Job method), 133

get_user_item_rating() (galaxy.model.item_attrs.UsesItemRatingMixin method), 146

get_user_job_count() (galaxy.jobs.handler.JobHandlerQueue method), 112

get_user_job_count_per_destination()
(galaxy.jobs.handler.JobHandlerQueue method), 112

get_user_quotas() (galaxy.quota.NoQuotaAgent method), 161

get_user_quotas() (galaxy.quota.QuotaAgent method), 161

get_user_tags_used() (galaxy.web.base.controller.UsesTagsMixin method), 239

get_valid_by_name() (galaxy.tools.parameters.sanitize.ToolParameterSanitizer class method), 206

get_valid_formats() (galaxy_utils.sequence.fastq.fastqAggregator method), 351

get_valid_roles() (galaxy.security.GalaxyRBACAgent method), 165

get_value() (galaxy.datatypes.display_applications.parameters.DisplayApplicationDataParameter method), 91

get_value() (galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter method), 91

get_value() (galaxy.datatypes.display_applications.parameters.DisplayApplicationTemplateParameter method), 92

get_value() (galaxy.external_services.parameters.ExternalServiceParameter method), 101

get_value() (galaxy.external_services.parameters.ExternalServiceTemplateParameter method), 101

get_value() (galaxy.tools.parameters.basic.BaseURLToolParameter method), 185

get_value() (galaxy.tools.parameters.output.FromDataTableOutputActionOption method), 202

get_value() (galaxy.tools.parameters.output.FromFileToolOutputActionOption method), 202

get_value() (galaxy.tools.parameters.output.FromParamToolOutputActionOption method), 202

get_value() (galaxy.tools.parameters.output.NullToolOutputActionOption method), 203

get_value() (galaxy.tools.parameters.output.ToolOutputActionOption method), 203

get_value() (galaxy.web.framework.helpers.grids.CommunityRatingColumn method), 246

get_value() (galaxy.web.framework.helpers.grids.CommunityTagsColumn method), 246

get_value() (galaxy.web.framework.helpers.grids.GridColumn method), 248

get_value() (galaxy.web.framework.helpers.grids.IndividualTagsColumn method), 249

get_value() (galaxy.web.framework.helpers.grids.OwnerAnnotationColumn method), 250

get_value() (galaxy.web.framework.helpers.grids.OwnerColumn method), 250

get_value() (galaxy.web.framework.helpers.grids.SharingStatusColumn method), 251

get_value() (galaxy.web.framework.helpers.grids.StateColumn method), 252

get_value() (galaxy.webapps.galaxy.controllers.admin.GroupListGrid.Name method), 296

get_value() (galaxy.webapps.galaxy.controllers.admin.GroupListGrid.Roles method), 296

get_value() (galaxy.webapps.galaxy.controllers.admin.GroupListGrid.Status method), 296

get_value() (galaxy.webapps.galaxy.controllers.admin.GroupListGrid.Users method), 296

get_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Amount method), 297

get_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Description method), 297

get_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Group method), 297

get_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Name method), 297

get_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Status method), 297

get_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Users method), 297

get_value() (galaxy.webapps.galaxy.controllers.admin.RoleListGrid.Description method), 298

get_value() (galaxy.webapps.galaxy.controllers.admin.RoleListGrid.Groups method), 298

[get_value\(\) \(galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGridToolSol \(galaxy.jobs.runners.BaseJobRunner method\), 340](#)
[get_value\(\) \(galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGridTool \(galaxy.jobs.runners.BaseJobRunner method\), 340](#)
[get_value\(\) \(galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid.CreateTimeColumn \(galaxy.webapps.reports.controllers.workflows\), 341](#)
[get_value\(\) \(galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid.RequestNameColumn \(galaxy.webapps.reports.controllers.workflows\), 342](#)
[get_value\(\) \(galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid.UserColumn \(galaxy.webapps.reports.controllers.workflows\), 342](#)
[get_value\(\) \(galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid.CreateTimeColumn \(galaxy.webapps.reports.controllers.workflows\), 343](#)
[get_value\(\) \(galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid.UserColumn \(galaxy.webapps.reports.controllers.workflows\), 344](#)
[get_value\(\) \(galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid.WorkflowDataTypesColumn \(galaxy.webapps.reports.controllers.workflows\), 344](#)
[get_version_fields\(\) \(galaxy.tools.data.TabularToolDataTableGFFIntervalToBEDReaderWrapper \(class in galaxy.datatypes.util.gff_util\), 181](#)
[get_version_string_path\(\) \(galaxy.jobs.JobWrapper GFFReaderWrapper \(class in galaxy.datatypes.util.gff_util\), 109](#)
[get_visualization\(\) \(galaxy.web.base.controller.UsesVisualizationMixin \(class in galaxy.datatypes.images\), 239](#)
[get_visualization_config\(\) \(galaxy.web.base.controller.UsesVisualizationMixin \(class in galaxy.datatypes.images\), 239](#)
[get_visualization_dict\(\) \(galaxy.web.base.controller.UsesVisualizationMixin \(class in galaxy.datatypes.images\), 239](#)
[get_visualization_revision_dict\(\) \(galaxy.web.base.controller.UsesVisualizationMixin \(class in galaxy.datatypes.images\), 239](#)
[get_visualization_summary_dict\(\) \(galaxy.web.base.controller.UsesVisualizationMixin \(class in galaxy.datatypes.images\), 239](#)
[get_visualizations\(\) \(galaxy.datatypes.data.Data method\), 48](#)
[get_visualizations\(\) \(galaxy.datatypes.data.Newick method\), 50](#)
[get_visualizations\(\) \(galaxy.datatypes.data.Nexus method\), 50](#)
[get_visualizations\(\) \(galaxy.datatypes.xml.Phyloxml method\), 85](#)
[get_visualizations\(\) \(galaxy.model.DatasetInstance method\), 125](#)
[get_visualizations_by_user\(\) \(galaxy.web.base.controller.UsesVisualizationMixin \(class in galaxy.datatypes.images\), 239](#)
[get_visualizations_shared_with_user\(\) \(galaxy.web.base.controller.UsesVisualizationMixin \(class in galaxy.datatypes.images\), 240](#)
[get_web_api_action\(\) \(galaxy.external_services.actions.ExternalServiceWebAPIAction \(class in galaxy.web.framework.helpers.grids\), 101](#)
[get_widgets\(\) \(galaxy.model.FormDefinition method\), 127](#)

GridColumnFilter	(class	in	H
galaxy.web.framework.helpers.grids), 248			
GridOperation	(class	in	H5 (class in galaxy.datatypes.binary), 44
galaxy.web.framework.helpers.grids), 248			handle_action() (galaxy.external_services.actions.ExternalServiceAction method), 100
groom_dataset_content()	(galaxy.datatypes.binary.Bam method), 43		handle_compressed_file() (in module galaxy.datatypes.sniff), 79
groom_dataset_content()	(galaxy.datatypes.data.Data method), 48		handle_controller_exception() (galaxy.web.framework.base.WebApplication method), 244
Group (class in galaxy.model), 128			handle_controller_exception() (galaxy.webapps.demo_sequencer.framework.WebApplication method), 258
Group (class in galaxy.tools.parameters.grouping), 200			handle_data() (galaxy.webapps.galaxy.controllers.workflow.SingleTagConte method), 335
group_list_grid (galaxy.web.base.controllers.admin.Admin attribute), 240			handle_failure() (galaxy.webapps.demo_sequencer.controllers.common.Cor method), 257
group_list_grid (galaxy.webapps.galaxy.controllers.admin.Admin attribute), 295			handle_found_index_file() (galaxy.tools.data.TabularToolDataTable method), 181
group_title() (galaxy.tools.parameters.grouping.UploadDataset method), 200			handle_incoming() (galaxy.jobs.actions.post.ActionBox class method), 114
GroupApiController	(class	in	handle_input() (galaxy.tools.Tool method), 173
galaxy.webapps.galaxy.api.groups), 14, 270			handle_interrupted() (galaxy.tools.Tool method), 173
GroupListGrid	(class	in	handle_job_failure_exception() (galaxy.tools.Tool method), 173
galaxy.webapps.galaxy.controllers.admin), 296			handle_library_params() (in module galaxy.tools.actions.upload_common), 181
GroupListGrid.NameColumn	(class	in	handle_operation() (galaxy.web.framework.helpers.grids.Grid method), 247
galaxy.webapps.galaxy.controllers.admin), 296			handle_request() (galaxy.web.framework.base.WebApplication method), 244
GroupListGrid.RolesColumn	(class	in	handle_request() (galaxy.webapps.demo_sequencer.controllers.common.Co method), 257
galaxy.webapps.galaxy.controllers.admin), 296			handle_result() (galaxy.external_services.result_handlers.basic.ExternalServ method), 103
GroupListGrid.StatusColumn	(class	in	handle_result() (galaxy.external_services.result_handlers.basic.ExternalServ method), 104
galaxy.webapps.galaxy.controllers.admin), 296			handle_result() (galaxy.external_services.result_handlers.basic.ExternalServ method), 104
GroupListGrid.UsersColumn	(class	in	handle_result() (galaxy.external_services.result_handlers.basic.ExternalServ method), 104
galaxy.webapps.galaxy.controllers.admin), 296			handle_results() (galaxy.external_services.actions.PopulatedExternalService method), 101
GroupQuotaAssociation (class in galaxy.model), 128			handle_single_execution() (galaxy.tools.Tool method), 173
GroupRoleAssociation (class in galaxy.model), 128			handle_stop() (galaxy.jobs.runners.AsynchronousJobRunner method), 117
GroupRolesApiController	(class	in	handle_unvalidated_param_values() (galaxy.tools.Tool method), 173
galaxy.webapps.galaxy.api.group_roles), 14, 269			handle_unvalidated_param_values_helper() (galaxy.tools.Tool method), 174
groups()	(galaxy.web.base.controllers.admin.Admin method), 240		handle_uploaded_dataset_file() (in module galaxy.datatypes.sniff), 79
GroupUsersApiController	(class	in	
galaxy.webapps.galaxy.api.group_users), 14, 270			
Gtf (class in galaxy.datatypes.interval), 64			
guess_derived_permissions_for_datasets()	(galaxy.security.GalaxyRBACAgent method), 165		
guess_derived_permissions_for_datasets()	(galaxy.security.RBACAgent method), 167		
guess_ext()	(in module galaxy.datatypes.sniff), 78		
guess_galaxy_port()	(galaxy.config.Configuration method), 40		

[handler_class](#) (in [module](#) [has_resource\(\)](#) ([pkg_resources.NullProvider](#) [method](#)), [365](#)
[galaxy.external_services.result_handlers.basic](#)), [104](#)
[hapmapmart\(\)](#) ([galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner](#) [method](#)), [327](#)
[has_accessible_folders\(\)](#) ([galaxy.security.GalaxyRBACAgent](#) [method](#)), [165](#)
[has_accessible_library_datasets\(\)](#) ([galaxy.security.GalaxyRBACAgent](#) [method](#)), [165](#)
[has_action\(\)](#) ([galaxy.external_services.service.ActionSection](#) [method](#)), [102](#)
[has_adapter_base\(\)](#) ([galaxy_utils.sequence.fastq.fastqCSSangerRead](#) [method](#)), [351](#)
[has_data\(\)](#) ([galaxy.model.Dataset](#) [method](#)), [123](#)
[has_data\(\)](#) ([galaxy.model.DatasetInstance](#) [method](#)), [125](#)
[has_data\(\)](#) ([galaxy.visualization.data_providers.basic.BaseDataProvider](#) [method](#)), [226](#)
[has_data\(\)](#) ([galaxy_utils.sequence.fasta.fastaNamedReader](#) [method](#)), [350](#)
[has_data\(\)](#) ([galaxy_utils.sequence.fastq.fastqFakeFastaScoreReader](#) [method](#)), [351](#)
[has_data\(\)](#) ([galaxy_utils.sequence.fastq.fastqNamedReader](#) [method](#)), [352](#)
[has_dataprovider\(\)](#) ([galaxy.datatypes.data.Data](#) [method](#)), [48](#)
[has_doi\(\)](#) ([galaxy.managers.citations.BaseCitation](#) [method](#)), [149](#)
[has_doi\(\)](#) ([galaxy.managers.citations.DoiCitation](#) [method](#)), [149](#)
[has_input_for_step\(\)](#) ([galaxy.model.WorkflowInvocation](#) [method](#)), [144](#)
[has_limits\(\)](#) ([galaxy.jobs.JobWrapper](#) [method](#)), [109](#)
[has_manage_permissions_roles\(\)](#) ([galaxy.model.Dataset](#) [method](#)), [123](#)
[has_manage_permissions_roles\(\)](#) ([galaxy.model.LibraryDatasetDatasetAssociation](#) [method](#)), [135](#)
[has_metadata\(\)](#) ([pkg_resources.FileMetadata](#) [method](#)), [364](#)
[has_metadata\(\)](#) ([pkg_resources.IMetadataProvider](#) [method](#)), [363](#)
[has_metadata\(\)](#) ([pkg_resources.NullProvider](#) [method](#)), [365](#)
[has_outputs_defined\(\)](#) ([galaxy.model.Workflow](#) [method](#)), [143](#)
[has_post_authentication_actions\(\)](#) ([galaxy.openid.providers.OpenIDProvider](#) [method](#)), [160](#)
[has_reference_data\(\)](#) ([galaxy.visualization.genomes.Genomes](#) [method](#)), [225](#)
[has_resolution](#) ([galaxy.datatypes.data.Data](#) [attribute](#)), [48](#)
[has_resource\(\)](#) ([pkg_resources.IResourceProvider](#) [method](#)), [364](#)
[has_resource\(\)](#) ([pkg_resources.NullProvider](#) [method](#)), [365](#)
[has_version\(\)](#) ([pkg_resources.Distribution](#) [method](#)), [361](#)
[has_tool_public_key](#) ([pkg_resources.Distribution](#) [attribute](#)), [361](#)
[HasJobMetrics](#) ([class](#) in [galaxy.model](#)), [128](#)
[HasName](#) ([class](#) in [galaxy.model](#)), [128](#)
[HDAManager](#) ([class](#) in [galaxy.managers.hdas](#)), [152](#)
[HDAProvenanceController](#) ([class](#) in [galaxy.webapps.galaxy.api.provenance](#)), [31](#), [287](#)
[header_dataprovider\(\)](#) ([galaxy.datatypes.binary.Bam](#) [method](#)), [43](#)
[header_read_dataprovider\(\)](#) ([galaxy.datatypes.tabular.Sam](#) [method](#)), [82](#)
[header_startswith](#) ([galaxy_utils.sequence.vcf.VariantCall](#) [attribute](#)), [355](#)
[header_endswith](#) ([galaxy_utils.sequence.vcf.VariantCall](#) [attribute](#)), [355](#)
[Heartbeat](#) ([class](#) in [galaxy.util.heartbeat](#)), [218](#)
[HiddenDataToolParameter](#) ([class](#) in [galaxy.tools.parameters.basic](#)), [192](#)
[HiddenField](#) ([class](#) in [galaxy.web.form_builder](#)), [231](#)
[HiddenToolParameter](#) ([class](#) in [galaxy.tools.parameters.basic](#)), [192](#)
[HideDatasetAction](#) ([class](#) in [galaxy.jobs.actions.post](#)), [115](#)
[HierarchicalObjectStore](#) ([class](#) in [galaxy.objectstore](#)), [157](#)
[HistoriesController](#) ([class](#) in [galaxy.webapps.galaxy.api.histories](#)), [15](#), [270](#)
[History](#) ([class](#) in [galaxy.model](#)), [128](#)
[history\(\)](#) ([galaxy.webapps.galaxy.controllers.root.RootController](#) [method](#)), [326](#)
[history_add_to\(\)](#) ([galaxy.webapps.galaxy.controllers.root.RootController](#) [method](#)), [326](#)
[history_as_xml\(\)](#) ([galaxy.webapps.galaxy.controllers.root.RootController](#) [method](#)), [326](#)
[history_content_type](#) ([galaxy.model.HistoryDatasetAssociation](#) [attribute](#)), [129](#)
[history_content_type](#) ([galaxy.model.HistoryDatasetCollectionAssociation](#) [attribute](#)), [130](#)
[history_data\(\)](#) ([galaxy.webapps.galaxy.controllers.history.HistoryController](#) [method](#)), [310](#)
[history_dataset_collections\(\)](#) ([galaxy.managers.collections.DatasetCollectionManager](#) [method](#)), [150](#)
[history_delete\(\)](#) ([galaxy.webapps.galaxy.controllers.root.RootController](#) [method](#)), [326](#)
[history_detail\(\)](#) ([galaxy.webapps.galaxy.controllers.mobile.Mobile](#) [method](#)), [316](#)
[history_get_default_permissions\(\)](#) ([galaxy.security.GalaxyRBACAgent](#) [method](#)), [165](#)

[history_import\(\) \(galaxy.webapps.galaxy.controllers.root.RootController method\), 326](#)
[history_list\(\) \(galaxy.webapps.galaxy.controllers.mobile.Mobile method\), 316](#)
[history_new\(\) \(galaxy.webapps.galaxy.controllers.root.RootController method\), 326](#)
[history_options\(\) \(galaxy.webapps.galaxy.controllers.root.RootController method\), 326](#)
[history_set_default_permissions\(\) \(galaxy.security.GalaxyRBACAgent method\), 165](#)
[history_set_default_permissions\(\) \(galaxy.security.RBACAgent method\), 168](#)
[history_set_default_permissions\(\) \(galaxy.webapps.galaxy.controllers.root.RootController method\), 326](#)
[HistoryAllPublishedGrid \(class in galaxy.webapps.galaxy.controllers.history\), 309](#)
[HistoryAllPublishedGrid.NameURLColumn \(class in galaxy.webapps.galaxy.controllers.history\), 309](#)
[HistoryAnnotationAssociation \(class in galaxy.model\), 129](#)
[HistoryAnnotationsController \(class in galaxy.webapps.galaxy.api.annotations\), 264](#)
[HistoryColumn \(class in galaxy.webapps.galaxy.controllers.visualization\), 330](#)
[HistoryContentAnnotationsController \(class in galaxy.webapps.galaxy.api.annotations\), 264](#)
[HistoryContentsController \(class in galaxy.webapps.galaxy.api.history_contents\), 17, 272](#)
[HistoryContentTagsController \(class in galaxy.webapps.galaxy.api.item_tags\), 19, 275](#)
[HistoryController \(class in galaxy.webapps.galaxy.controllers.history\), 309](#)
[HistoryDatasetAssociation \(class in galaxy.model\), 129](#)
[HistoryDatasetAssociationAnnotationAssociation \(class in galaxy.model\), 130](#)
[HistoryDatasetAssociationDisplayAtAuthorization \(class in galaxy.model\), 130](#)
[HistoryDatasetAssociationListGrid \(class in galaxy.webapps.galaxy.controllers.dataset\), 305](#)
[HistoryDatasetAssociationListGrid.HistoryColumn \(class in galaxy.webapps.galaxy.controllers.dataset\), 305](#)
[HistoryDatasetAssociationListGrid.StatusColumn \(class in galaxy.webapps.galaxy.controllers.dataset\), 305](#)
[HistoryDatasetAssociationRatingAssociation \(class in galaxy.model\), 130](#)
[HistoryDatasetAssociationSelectionGrid \(class in galaxy.webapps.galaxy.controllers.page\), 316](#)
[HistoryDatasetAssociationSubset \(class in galaxy.model\), 130](#)
[HistoryDatasetAssociationTagAssociation \(class in galaxy.model\), 130](#)
[HistoryDatasetCollectionAnnotationAssociation \(class in galaxy.model\), 130](#)
[HistoryDatasetCollectionAssociation \(class in galaxy.model\), 130](#)
[HistoryDatasetCollectionRatingAssociation \(class in galaxy.model\), 130](#)
[HistoryDatasetCollectionTagAssociation \(class in galaxy.model\), 131](#)
[HistoryDatasetExtendMetadataController \(class in galaxy.webapps.galaxy.api.extended_metadata\), 10, 265](#)
[HistoryDatasetsSelectionGrid \(class in galaxy.webapps.galaxy.controllers.visualization\), 331](#)
[HistoryField \(class in galaxy.web.form_builder\), 231](#)
[HistoryListGrid \(class in galaxy.webapps.galaxy.controllers.history\), 311](#)
[HistoryListGrid.DatasetsByStateColumn \(class in galaxy.webapps.galaxy.controllers.history\), 311](#)
[HistoryListGrid.DeletedColumn \(class in galaxy.webapps.galaxy.controllers.history\), 311](#)
[HistoryListGrid.HistoryListNameColumn \(class in galaxy.webapps.galaxy.controllers.history\), 311](#)
[HistoryManager \(class in galaxy.managers.histories\), 153](#)
[HistoryRatingAssociation \(class in galaxy.model\), 131](#)
[HistorySelectionGrid \(class in galaxy.webapps.galaxy.controllers.page\), 317](#)
[HistorySelectionGrid \(class in galaxy.webapps.galaxy.controllers.visualization\), 331](#)
[HistorySerializer \(class in galaxy.managers.hdas\), 153](#)
[HistorySerializer \(class in galaxy.managers.histories\), 154](#)
[HistoryTagAssociation \(class in galaxy.model\), 131](#)
[HistoryTagsController \(class in galaxy.webapps.galaxy.api.item_tags\), 19, 275](#)

HistoryUserShareAssociation (class in galaxy.model), 131

hmac_new() (in module galaxy.util.hash_util), 218

HostAgent (class in galaxy.security), 166

Html (class in galaxy.datatypes.images), 56

html_table (galaxy.datatypes.genetics.RexpBase attribute), 54

http_error_default() (galaxy.eggs.URLRetriever method), 95

HttpDataTransferFactory (class in galaxy.sample_tracking.data_transfer), 161

humanize() (galaxy.util.inflection.Base method), 219

humanize() (galaxy.util.inflection.Inflector method), 220

I

id_seq_qual_dataprovider() (galaxy.datatypes.binary.Bam method), 43

id_seq_qual_dataprovider() (galaxy.datatypes.tabular.Sam method), 82

iff() (in module galaxy.web.framework.helpers), 245

Im (class in galaxy.datatypes.images), 57

Image (class in galaxy.datatypes.images), 57

image_type() (in module galaxy.datatypes.util.image_util), 94

IMetadataProvider (class in pkg_resources), 363

immediate_actions (galaxy.jobs.actions.post.ActionBox attribute), 114

imp() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304

imp() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310

imp() (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334

imp() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337

impersonate() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 295

impl (galaxy.model.custom_types.JSONType attribute), 145

impl (galaxy.model.custom_types.TrimmedString attribute), 145

impl (galaxy.model.custom_types.UUIDType attribute), 145

ImplicitlyConvertedDatasetAssociation (class in galaxy.model), 131

ImplicitlyCreatedDatasetCollectionInput (class in galaxy.model), 131

import_archive() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310

import_datasets_to_histories() (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315

import_module() (in module galaxy.util.backports.importlib), 225

import_new_workflow_deprecated() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 38, 294

import_shared_workflow_deprecated() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 39, 294

import_visualization() (galaxy.web.base.controller.UsesVisualizationMixin method), 240

import_workflow() (galaxy.webapps.galaxy.api.tool_shed_repositories.Tool method), 34, 289

import_workflow() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 302

import_workflow() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337

import_workflows() (galaxy.webapps.galaxy.api.tool_shed_repositories.Tool method), 34, 289

ImportHistoryTool (class in galaxy.tools), 171

ImportHistoryToolAction (class in galaxy.tools.actions.history_imp_exp), 180

ImportsHistoryMixin (class in galaxy.web.base.controller), 235

in_directory() (in module galaxy.util), 214

InappropriateDatasetContentError, 78

InconsistentDatabase, 98

increase_running_job_count() (galaxy.jobs.handler.JobHandlerQueue method), 112

index() (galaxy.web.base.controllers.admin.Admin method), 240

index() (galaxy.webapps.demo_sequencer.controllers.common.CommonController method), 257

index() (galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController method), 8, 264

index() (galaxy.webapps.galaxy.api.configuration.ConfigurationController method), 9, 264

index() (galaxy.webapps.galaxy.api.dataset_collections.DatasetCollectionsController method), 9, 265

index() (galaxy.webapps.galaxy.api.datatypes.DatatypesController method), 9, 265

index() (galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController method), 10, 265

index() (galaxy.webapps.galaxy.api.folder_contents.FolderContentsController method), 10, 266

index() (galaxy.webapps.galaxy.api.folders.FoldersController method), 12, 268

index() (galaxy.webapps.galaxy.api.forms.FormDefinitionAPIController method), 13, 269

index() (galaxy.webapps.galaxy.api.ftp_files.FTPFilesAPIController method), 13, 269

index() (galaxy.webapps.galaxy.api.genomes.GenomesController method), 13, 269

index() (galaxy.webapps.galaxy.api.group_roles.GroupRolesAPIController method), 14, 269

[index\(\) \(galaxy.webapps.galaxy.api.group_users.GroupUsersAPIController method\), 14, 270](#)
[index\(\) \(galaxy.webapps.galaxy.api.groups.GroupAPIController method\), 14, 270](#)
[index\(\) \(galaxy.webapps.galaxy.api.histories.HistoriesController method\), 16, 271](#)
[index\(\) \(galaxy.webapps.galaxy.api.history_contents.HistoryContentsController method\), 18, 273](#)
[index\(\) \(galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController method\), 19, 275](#)
[index\(\) \(galaxy.webapps.galaxy.api.job_files.JobFilesAPIController method\), 20, 276](#)
[index\(\) \(galaxy.webapps.galaxy.api.jobs.JobController method\), 21, 276](#)
[index\(\) \(galaxy.webapps.galaxy.api.libraries.LibrariesController method\), 26, 281](#)
[index\(\) \(galaxy.webapps.galaxy.api.library_contents.LibraryContentsController method\), 28, 284](#)
[index\(\) \(galaxy.webapps.galaxy.api.page_revisions.PageRevisionsController method\), 30, 285](#)
[index\(\) \(galaxy.webapps.galaxy.api.pages.PagesController method\), 31, 286](#)
[index\(\) \(galaxy.webapps.galaxy.api.provenance.BaseProvenanceController method\), 31, 287](#)
[index\(\) \(galaxy.webapps.galaxy.api.quotas.QuotaAPIController method\), 32, 287](#)
[index\(\) \(galaxy.webapps.galaxy.api.request_types.RequestTypesAPIController method\), 32, 288](#)
[index\(\) \(galaxy.webapps.galaxy.api.requests.RequestsAPIController method\), 32, 288](#)
[index\(\) \(galaxy.webapps.galaxy.api.roles.RoleAPIController method\), 33, 288](#)
[index\(\) \(galaxy.webapps.galaxy.api.samples.SamplesAPIController method\), 33, 288](#)
[index\(\) \(galaxy.webapps.galaxy.api.tool_data.ToolDataController method\), 33, 289](#)
[index\(\) \(galaxy.webapps.galaxy.api.tool_shed_repositories.ToolShedRepositoriesController method\), 34, 290](#)
[index\(\) \(galaxy.webapps.galaxy.api.users.UserAPIController method\), 37, 292](#)
[index\(\) \(galaxy.webapps.galaxy.api.visualizations.VisualizationController method\), 37, 292](#)
[index\(\) \(galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method\), 39, 294](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.async.ASync method\), 303](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.error.Error method\), 306](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.history.HistoryController method\), 310](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.library.Library method\), 313](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.mobile.Mobile method\), 316](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.requests.RequestsAPIController method\), 321](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdminController method\), 323](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.root.RootController method\), 326](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner method\), 327](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.ucsc_proxy.UCSCProxy method\), 327](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.user.User method\), 328](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.visualization.VisualizationController method\), 334](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.workflow.WorkflowController method\), 337](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.galaxy_lowwebapps.reports.controllers.root.ReportController method\), 341](#)
[index\(\) \(galaxy.webapps.galaxy.controllers.galaxy_lowwebapps.reports.controllers.system.SystemController method\), 343](#)
[IndividualTagsColumn \(class in galaxy.web.framework.helpers.grids\), 249](#)
[info \(class in galaxy.util.inflection\), 220](#)
[info \(galaxy.model.LibraryDataset attribute\), 135](#)
[info_text \(galaxy.web.framework.helpers.grids.Grid attribute\), 247](#)
[info_text \(galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute\), 311](#)
[info_text \(galaxy.model.LibraryDatasetDatasetInfoAssociation attribute\), 136](#)
[init\(\) \(in module galaxy.model.mapping\), 147](#)
[init_dependency_manager\(\) \(galaxy.tools.ToolBox method\), 176](#)
[init_meta\(\) \(galaxy.datatypes.binary.Bam method\), 43](#)
[init_meta\(\) \(galaxy.datatypes.binary.SQLite method\), 45](#)
[init_meta\(\) \(galaxy.datatypes.data.Data method\), 48](#)
[init_meta\(\) \(galaxy.datatypes.data.Newick method\), 50](#)
[init_meta\(\) \(galaxy.datatypes.data.Nexus method\), 50](#)
[init_meta\(\) \(galaxy.datatypes.genetics.RexpBase method\), 54](#)
[init_meta\(\) \(galaxy.datatypes.interval.Interval method\), 65](#)
[init_meta\(\) \(galaxy.datatypes.sequence.Maf method\), 76](#)
[init_meta\(\) \(galaxy.datatypes.tabular.Pileup method\), 81](#)
[init_meta\(\) \(galaxy.model.DatasetInstance method\), 125](#)
[init_tools\(\) \(galaxy.tools.ToolBox method\), 176](#)
[initiate_data_transfer\(\) \(galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdminController method\), 323](#)
[install_tool_dependency_installation\(\) \(galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshedController method\), 302](#)
[inject\(\) \(galaxy.workflow.modules.WorkflowModuleInjector method\), 349](#)

Inprogress_dataset_files (galaxy.model.Sample attribute),
 139
 input_paths() (galaxy.jobs.ComputeEnvironment
 method), 106
 input_paths() (galaxy.jobs.SharedComputeEnvironment
 method), 110
 InputDataCollectionModule (class in
 galaxy.workflow.modules), 345
 InputDataModule (class in galaxy.workflow.modules),
 345
 InputModule (class in galaxy.workflow.modules), 345
 inputs() (galaxy.webapps.galaxy.api.jobs.JobController
 method), 21, 277
 InRangeValidator (class in
 galaxy.tools.parameters.validation), 206
 insert() (galaxy.util.odict.odict method), 223
 insert_on() (pkg_resources.Distribution method), 361
 InsertColumnToolOutputActionOptionFilter (class in
 galaxy.tools.parameters.output), 202
 install_latest_repository_revision()
 (galaxy.webapps.galaxy.controllers.admin_toolshed.
 AdminToolshedController method), 302
 install_model (galaxy.managers.context.ProvidesAppContext
 attribute), 150
 install_repository_revision()
 (galaxy.webapps.galaxy.api.tool_shed_repositories.
 ToolShedRepositoriesController
 method), 34, 290
 install_repository_revisions()
 (galaxy.webapps.galaxy.api.tool_shed_repositories.
 ToolShedRepositoriesController
 method), 35, 290
 install_tool_dependencies_with_update()
 (galaxy.webapps.galaxy.controllers.admin_toolshed.
 AdminToolshedController
 method), 302
 install_tool_shed_repositories()
 (galaxy.webapps.galaxy.controllers.admin_toolshed.
 AdminToolshedController
 method), 302
 installed_len_files (galaxy.webapps.galaxy.controllers.user.User
 attribute), 328
 installed_repository_grid
 (galaxy.webapps.galaxy.controllers.admin_toolshed.
 AdminToolshedController
 attribute), 302
 installed_tool_dependencies (galaxy.tools.Tool attribute),
 174
 insufficient_quality_length()
 (galaxy_utils.sequence.fastq.fastqCSSangerRead
 method), 351
 insufficient_quality_length()
 (galaxy_utils.sequence.fastq.fastqSequencingReads_case()
 method), 353
 InsufficientPermissionsException, 98
 IntegerColumn (class in
 galaxy.web.framework.helpers.grids), 249
 IntegerToolParameter (class in
 galaxy.tools.parameters.basic), 192
 integrated_datatypes_configs
 (galaxy.datatypes.registry.Registry
 attribute), 72
 InternalServerError, 98
 InterruptedUpload, 171
 Interval (class in galaxy.datatypes.interval), 64
 interval_dataprovider() (galaxy.datatypes.interval.Gff
 method), 62
 interval_dataprovider() (galaxy.datatypes.interval.Interval
 method), 65
 interval_dict_dataprovider()
 (galaxy.datatypes.interval.Gff method), 63
 interval_dict_dataprovider()
 (galaxy.datatypes.interval.Interval
 method), 65
 invalid_id_redirect() (in
 galaxy.webapps.galaxy.controllers.requests_common),
 325
 invalid_marker() (in module pkg_resources), 363
 IResourceProvider (class in pkg_resources), 363
 is_admin_toolshed (galaxy.managers.histories.HistoryManager
 method), 154
 is_admin_user() (galaxy.config.Configuration
 attribute), 40
 is_admin_user() (galaxy.webapps.demo_sequencer.config.Configuration
 attribute), 256
 is_ascii_encoded() (galaxy_utils.sequence.fastq.fastqSequencingRead
 method), 353
 is_binary (galaxy.datatypes.genetics.RexpBase attribute),
 54
 is_binary (galaxy.datatypes.ngsindex.BowtieIndex
 attribute), 70
 is_binary() (in module galaxy.util), 214
 is_case() (in module galaxy.datatypes.checkers), 46
 is_case() (galaxy.external_services.service.BooleanExternalServiceActionsGroupW
 method), 102
 is_case() (galaxy.external_services.service.ExternalServiceActionsGroupW
 method), 102
 is_case() (galaxy.external_services.service.ItemIsInstanceExternalServiceA
 method), 103
 is_case() (galaxy.external_services.service.ValueExternalServiceActionsGro
 method), 103
 is_case() (galaxy.tools.parameters.output.DatatypeIsInstanceToolOutputAct
 method), 202
 is_case() (galaxy.tools.parameters.output.ToolOutputActionConditionalWh
 method), 204
 is_case() (galaxy.tools.parameters.output.ValueToolOutputActionCondition
 method), 204
 is_check_time (galaxy.model.DeferredJob attribute), 127
 is_checked() (galaxy.web.form_builder.CheckboxField
 static method), 229
 is_collection (galaxy.model.DatasetCollectionElement at-
 tribute), 124

- [is_column_based\(\)](#) (in module `galaxy.datatypes.sniff`), 79
[is_complete](#) (`galaxy.model.Request` attribute), 137
[is_current\(\)](#) (`galaxy.managers.histories.HistoryManager` method), 154
[is_current_version\(\)](#) (`galaxy.tools.data.ToolDataTable` method), 182
[is_DNA\(\)](#) (`galaxy_utils.sequence.sequence.SequencingRead` method), 354
[is_ext_unsniffable\(\)](#) (`galaxy.datatypes.binary.Binary` static method), 44
[is_first_mate\(\)](#) (`galaxy_utils.sequence.fastq.fastqJoiner` method), 352
[is_gzip\(\)](#) (in module `galaxy.datatypes.checkers`), 46
[is_handler\(\)](#) (`galaxy.jobs.JobConfiguration` method), 108
[is_hashable\(\)](#) (in module `galaxy.util.hash_util`), 218
[is_id\(\)](#) (`galaxy.jobs.JobConfiguration` method), 108
[is_job_resource_conditional](#) (`galaxy.tools.parameters.grouping.Conditional` attribute), 199
[is_JSONified_value\(\)](#) (`galaxy.datatypes.metadata.MetadataTemplate` class method), 69
[is_multi_byte\(\)](#) (`galaxy.model.Dataset` method), 123
[is_multi_byte\(\)](#) (`galaxy.model.DatasetInstance` method), 125
[is_multi_byte\(\)](#) (in module `galaxy.util`), 214
[is_mutable\(\)](#) (`galaxy.model.custom_types.JSONType` method), 145
[is_new](#) (`galaxy.model.Request` attribute), 137
[is_outside_grace_period\(\)](#) (`galaxy.webapps.galaxy.controllers.user.User` method), 328
[is_owner\(\)](#) (`galaxy.managers.histories.HistoryManager` method), 154
[is_pending](#) (`galaxy.model.DatasetInstance` attribute), 125
[is_preparing\(\)](#) (`galaxy.datatypes.display_applications.parameters.DisplayApplicationParameters` method), 91
[is_preparing\(\)](#) (`galaxy.datatypes.display_applications.parameters.DisplayApplicationParameters` method), 92
[is_public\(\)](#) (`galaxy.managers.libraries.LibraryManager` method), 155
[is_rejected](#) (`galaxy.model.Request` attribute), 137
[is_sniffable_binary\(\)](#) (`galaxy.datatypes.binary.Binary` static method), 44
[is_submitted](#) (`galaxy.model.Request` attribute), 137
[is_tag\(\)](#) (`galaxy.jobs.JobConfiguration` method), 108
[is_tool_module_type\(\)](#) (in module `galaxy.workflow.modules`), 349
[is_true\(\)](#) (in module `galaxy.web.framework.helpers`), 245
[is_unsubmitted](#) (`galaxy.model.Request` attribute), 137
[is_uuid\(\)](#) (in module `galaxy.util`), 214
[is_valid_format\(\)](#) (`galaxy_utils.sequence.fastq.fastqSequencingRead` method), 353
[is_valid_sequence\(\)](#) (`galaxy_utils.sequence.fastq.fastqSequencingRead` method), 353
[item_has_tag\(\)](#) (`galaxy.tags.tag_handler.TagHandler` method), 169
[item_permission_map_for_add\(\)](#) (`galaxy.security.GalaxyRBACAgent` method), 165
[item_permission_map_for_manage\(\)](#) (`galaxy.security.GalaxyRBACAgent` method), 165
[item_permission_map_for_modify\(\)](#) (`galaxy.security.GalaxyRBACAgent` method), 166
[ItemAccessibilityException](#), 98
[ItemDeletionException](#), 98
[ItemIsInstanceExternalServiceActionsGroupWhen](#) (class in `galaxy.external_services.service`), 102
[ItemOwnershipException](#), 98
[ItemRatingAssociation](#) (class in `galaxy.model`), 131
[items\(\)](#) (`galaxy.datatypes.metadata.MetadataCollection` method), 68
[items\(\)](#) (`galaxy.util.bunch.Bunch` method), 217
[items\(\)](#) (`galaxy.util.odict.odict` method), 223
[ItemSelectionGrid](#) (class in `galaxy.webapps.galaxy.controllers.page`), 317
[ItemSelectionGrid.NameColumn](#) (class in `galaxy.webapps.galaxy.controllers.page`), 317
[ItemTagAssociation](#) (class in `galaxy.model`), 131
[ItemTagAssocInfo](#) (class in `galaxy.tags.tag_handler`), 168
[iter\(\)](#) (`galaxy.datatypes.metadata.MetadataSpecCollection` method), 69
[iter_blocks_split_by_species\(\)](#) (in module `galaxy.tools.util.maf_utilities`), 212
[iter_blocks_split_by_src\(\)](#) (in module `galaxy.tools.util.maf_utilities`), 212
[iter_components_by_src\(\)](#) (in module `galaxy.tools.util.maf_utilities`), 212
[iter_components_by_src_start\(\)](#) (in module `galaxy.tools.util.maf_utilities`), 212
[iter_entry_points\(\)](#) (`pkg_resources.WorkingSet` method), 359
[iter_fasta_alignment\(\)](#) (in module `galaxy.tools.util.maf_utilities`), 212
[iterate_file\(\)](#) (in module `galaxy.web.framework.base`), 244
[iteritems\(\)](#) (`galaxy.util.odict.odict` method), 223
[iterkeys\(\)](#) (`galaxy.util.odict.odict` method), 223
[itervalues\(\)](#) (`galaxy.util.odict.odict` method), 223
- ## J
- [job_read](#) (class in `galaxy.model`), 131
[job_destination](#) (`galaxy.jobs.JobWrapper` attribute), 109
[job_finished\(\)](#) (`galaxy.tools.SetMetadataTool` method), 171
[job_failed\(\)](#) (`galaxy.tools.Tool` method), 174

- [job_info\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 240
[job_info\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 339
[job_pair_for_id\(\)](#) (galaxy.jobs.handler.JobHandlerQueue method), 112
[job_states](#) (galaxy.jobs.deferred.DeferredJobQueue attribute), 116
[job_wrapper\(\)](#) (galaxy.jobs.handler.JobHandlerQueue method), 112
[JobConfiguration](#) (class in galaxy.jobs), 106
[JobController](#) (class in galaxy.webapps.galaxy.api.jobs), 21, 276
[JobDestination](#) (class in galaxy.jobs), 108
[JobExportHistoryArchive](#) (class in galaxy.model), 133
[JobExportHistoryArchiveWrapper](#) (class in galaxy.tools.imp_exp), 184
[JobExternalOutputMetadata](#) (class in galaxy.model), 133
[JobExternalOutputMetadataWrapper](#) (class in galaxy.datatypes.metadata), 67
[JobFilesAPIController](#) (class in galaxy.webapps.galaxy.api.job_files), 20, 275
[JobHandler](#) (class in galaxy.jobs.handler), 112
[JobHandlerQueue](#) (class in galaxy.jobs.handler), 112
[JobHandlerStopQueue](#) (class in galaxy.jobs.handler), 112
[JobImportHistoryArchive](#) (class in galaxy.model), 134
[JobImportHistoryArchiveWrapper](#) (class in galaxy.tools.imp_exp), 184
[JobManager](#) (class in galaxy.jobs.manager), 113
[JobMappingException](#), 113
[JobMetricNumeric](#) (class in galaxy.model), 134
[JobMetricText](#) (class in galaxy.model), 134
[JobNotReadyException](#), 113
[JobParameter](#) (class in galaxy.model), 134
[JobRunnerMapper](#) (class in galaxy.jobs.mapper), 113
[Jobs](#) (class in galaxy.webapps.reports.controllers.jobs), 339
[jobs\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 240
[JobState](#) (class in galaxy.jobs.runners), 119
[JobStateHistory](#) (class in galaxy.model), 134
[JobToInputDatasetAssociation](#) (class in galaxy.model), 134
[JobToInputDatasetCollectionAssociation](#) (class in galaxy.model), 134
[JobToInputLibraryDatasetAssociation](#) (class in galaxy.model), 134
[JobToolConfiguration](#) (class in galaxy.jobs), 108
[JobToOutputDatasetAssociation](#) (class in galaxy.model), 134
[JobToOutputDatasetCollectionAssociation](#) (class in galaxy.model), 134
[JobToOutputLibraryDatasetAssociation](#) (class in galaxy.model), 134
[JobWrapper](#) (class in galaxy.jobs), 108
[join\(\)](#) (galaxy_utils.sequence.fastq.fastqJoiner method), 352
[Jpg](#) (class in galaxy.datatypes.images), 57
[js\(\)](#) (in module galaxy.web.framework.helpers), 245
[js_helper\(\)](#) (in module galaxy.web.framework.helpers), 245
[json_fix\(\)](#) (in module galaxy.tools), 177
[json_fix\(\)](#) (in module galaxy.util.json), 222
[jsonrpc_request\(\)](#) (in module galaxy.util.json), 222
[jsonrpc_response\(\)](#) (in module galaxy.util.json), 222
[JSONType](#) (class in galaxy.model.custom_types), 145
- ## K
- [k](#) (galaxy.webapps.galaxy.api.samples.SamplesAPIController attribute), 33, 288
[key](#) (galaxy_utils.sequence.transform.ColorSpaceConverter attribute), 354
[key](#) (pkg_resources.Distribution attribute), 361
[keys\(\)](#) (galaxy.util.bunch.Bunch method), 217
[keys\(\)](#) (galaxy.util.odict.odict method), 223
- ## L
- [label](#) (galaxy.tools.parameters.grouping.Conditional attribute), 199
[label\(\)](#) (galaxy.tools.parameters.grouping.Repeat method), 200
[Laj](#) (class in galaxy.datatypes.images), 57
[last_access_date\(\)](#) (galaxy.webapps.reports.controllers.users.Users method), 343
[last_check](#) (galaxy.model.DeferredJob attribute), 127
[last_comment](#) (galaxy.model.Request attribute), 137
[LastzCoverage](#) (class in galaxy.datatypes.coverage), 46
[latest_event](#) (galaxy.model.Request attribute), 137
[latest_event](#) (galaxy.model.Sample attribute), 139
[latest_export](#) (galaxy.model.History attribute), 129
[LateValidationError](#), 207
[Lav](#) (class in galaxy.datatypes.sequence), 75
[lazy_property](#) (in module galaxy.web.framework.base), 244
[LazyProperty](#) (class in galaxy.web.framework.base), 242
[ldda_edit_info\(\)](#) (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315
[ldda_info\(\)](#) (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315
[ldda_permissions\(\)](#) (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315
[LDDAManager](#) (class in galaxy.managers.lddas), 154
[LDDAProvenanceController](#) (class in galaxy.webapps.galaxy.api.provenance), 31, 287
[IdIndep](#) (class in galaxy.datatypes.genetics), 55

[legal_values \(galaxy.tools.parameters.basic.BooleanToolParameter attribute\), 186](#)
[legend \(galaxy.web.framework.helpers.grids.Grid attribute\), 248](#)
[LengthValidator \(class in galaxy.tools.parameters.validation\), 207](#)
[LibrariesController \(class in galaxy.webapps.galaxy.api.libraries\), 25, 280](#)
[Library \(class in galaxy.model\), 134](#)
[Library \(class in galaxy.webapps.galaxy.controllers.library\), 313](#)
[library_dataset_info\(\) \(galaxy.webapps.galaxy.controllers.library_common_library_dataset_info method\), 315](#)
[library_dataset_permissions\(\) \(galaxy.webapps.galaxy.controllers.library_common_library_dataset_permissions method\), 315](#)
[library_info\(\) \(galaxy.webapps.galaxy.controllers.library_common_library_info method\), 315](#)
[library_is_public\(\) \(galaxy.security.GalaxyRBACAgent method\), 166](#)
[library_is_public\(\) \(galaxy.security.RBACAgent method\), 168](#)
[library_is_unrestricted\(\) \(galaxy.security.GalaxyRBACAgent method\), 166](#)
[library_item_updates\(\) \(galaxy.webapps.galaxy.controllers.library_common_library_item_updates method\), 315](#)
[library_list_grid \(galaxy.webapps.galaxy.controllers.library.Library attribute\), 313](#)
[library_list_grid \(galaxy.webapps.galaxy.controllers.library_admin.Library attribute\), 313](#)
[library_path \(galaxy.model.LibraryFolder attribute\), 136](#)
[library_permissions\(\) \(galaxy.webapps.galaxy.controllers.library_common_library_permissions method\), 315](#)
[LibraryAdmin \(class in galaxy.webapps.galaxy.controllers.library_admin\), 313](#)
[LibraryCommon \(class in galaxy.webapps.galaxy.controllers.library_common\), 314](#)
[LibraryContentsController \(class in galaxy.webapps.galaxy.api.library_contents\), 27, 283](#)
[LibraryDataset \(class in galaxy.model\), 134](#)
[LibraryDatasetCollectionAnnotationAssociation \(class in galaxy.model\), 135](#)
[LibraryDatasetCollectionAssociation \(class in galaxy.model\), 135](#)
[LibraryDatasetCollectionRatingAssociation \(class in galaxy.model\), 135](#)
[LibraryDatasetCollectionTagAssociation \(class in galaxy.model\), 135](#)
[LibraryDatasetDatasetAssociation \(class in galaxy.model\), 135](#)
[LibraryDatasetDatasetAssociationPermissions \(class in galaxy.model\), 136](#)
[LibraryDatasetDatasetInfoAssociation \(class in galaxy.model\), 136](#)
[LibraryDatasetExtendMetadataController \(class in galaxy.webapps.galaxy.api.extended_metadata\), 10, 266](#)
[LibraryDatasetPermissions \(class in galaxy.model\), 136](#)
[LibraryDatasetsController \(class in galaxy.webapps.galaxy.api.lda_datasets\), 22, 278](#)
[LibraryDatasetToolParameter \(class in galaxy.tools.parameters.basic\), 193](#)
[LibraryField \(class in galaxy.web.form_builder\), 231](#)
[LibraryFolder \(class in galaxy.model\), 136](#)
[LibraryFolderCollectionAssociation \(class in galaxy.model\), 136](#)
[LibraryFieldPermissions \(class in galaxy.model\), 136](#)
[LibraryInfoAssociation \(class in galaxy.model\), 136](#)
[LibraryListGrid \(class in galaxy.webapps.galaxy.controllers.library\), 313](#)
[LibraryListGrid \(class in galaxy.webapps.galaxy.controllers.library_admin\), 314](#)
[LibraryListGrid.DescriptionColumn \(class in galaxy.webapps.galaxy.controllers.library_admin\), 314](#)
[LibraryListGrid.NameColumn \(class in galaxy.webapps.galaxy.controllers.library_admin\), 313](#)
[LibraryListGrid.StatusColumn \(class in galaxy.webapps.galaxy.controllers.library_admin\), 314](#)
[LibraryManager \(class in galaxy.managers.libraries\), 154](#)
[LibraryPermissions \(class in galaxy.model\), 136](#)
[LibrarySelectionGrid \(class in galaxy.webapps.galaxy.controllers.visualization\), 332](#)
[line_class \(galaxy.datatypes.data.Text attribute\), 50](#)
[line_class \(galaxy.datatypes.interval.Interval attribute\), 65](#)
[line_class \(galaxy.datatypes.tabular.Pileup attribute\), 81](#)
[line_dataprovider\(\) \(galaxy.datatypes.binary.Bam method\), 43](#)
[line_dataprovider\(\) \(galaxy.datatypes.data.Text method\), 50](#)

[line_dataprovider\(\)](#) (galaxy.datatypes.tabular.Sam method), 82
[line_enumerator\(\)](#) (in module galaxy.tools.util.maf_utilities), 212
[LineCount](#) (class in galaxy.datatypes.data), 49
[lines\(\)](#) (galaxy.datatypes.util.gff_util.GFFFeature method), 93
[list\(\)](#) (galaxy.managers.libraries.LibraryManager method), 155
[list\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
[list\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[list\(\)](#) (galaxy.webapps.galaxy.controllers.library.Library method), 313
[list\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[list\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
[list_as_xml\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[list_current_history_datasets\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list_datasets\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list_datasets_for_selection\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[list_for_run\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
[list_grid\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
[list_histories\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list_histories_for_selection\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[list_history_datasets\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list_libraries\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list_library_datasets\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list_pages_for_selection\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[list_published\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[list_published\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[list_published\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list_published\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
[list_shared\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[list_tracks\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[list_visualizations_for_selection\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[list_workflows_for_selection\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[listify\(\)](#) (in module galaxy.util), 214
[ListParameter](#) (class in galaxy.datatypes.metadata), 67
[load\(\)](#) (galaxy.util.aliaspickler.AliasPickleModule method), 217
[load\(\)](#) (galaxy.webapps.galaxy.api.lda_datasets.LibraryDatasetsController method), 23, 278
[load\(\)](#) (pkg_resources.EntryPoint method), 361
[load_cache\(\)](#) (galaxy.sample_tracking.external_service_types.ExternalServiceTypes method), 162
[load_build_sites\(\)](#) (galaxy.datatypes.registry.Registry method), 72
[load_data_transfer_settings\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 127
[load_datatype_converters\(\)](#) (galaxy.datatypes.registry.Registry method), 72
[load_datatype_sniffers\(\)](#) (galaxy.datatypes.registry.Registry method), 72
[load_datatypes\(\)](#) (galaxy.datatypes.registry.Registry method), 72
[load_dialect_impl\(\)](#) (galaxy.model.custom_types.JSONType method), 146
[load_dialect_impl\(\)](#) (galaxy.model.custom_types.UUIDType method), 146
[load_display_applications\(\)](#) (galaxy.datatypes.registry.Registry method), 72
[load_entry_point\(\)](#) (in module pkg_resources), 357
[load_external_service_type\(\)](#) (galaxy.sample_tracking.external_service_types.ExternalServiceTypes method), 162
[load_file\(\)](#) (galaxy.tools.data.ToolDataTableManager method), 183
[load_imp_exp_tools\(\)](#) (in module galaxy.tools.imp_exp), 184

[load_integrated_tool_panel_keys\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[load_label_tag_set\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[load_module_sections\(\)](#) (in module galaxy.workflow.modules), [349](#)
[load_section_tag_set\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[load_sub_elems\(\)](#) (galaxy.external_services.service.ExternalServiceActionsGroup method), [102](#)
[load_tool\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[load_tool_panel\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[load_tool_tag_set\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[load_tooldir_tag_set\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[load_workflow\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[load_workflow\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), [337](#)
[load_workflow_tag_set\(\)](#) (galaxy.tools.ToolBox method), [176](#)
[loader](#) (pkg_resources.NullProvider attribute), [365](#)
[loads\(\)](#) (galaxy.util.aliaspickler.AliasPickleModule method), [217](#)
[loads\(\)](#) (in module galaxy.util.json), [221](#)
[local_extra_dirs\(\)](#) (in module galaxy.objectstore), [159](#)
[LocalJobRunner](#) (class in galaxy.jobs.runners.local), [120](#)
[log_action\(\)](#) (galaxy.managers.context.ProvidesAppContext method), [150](#)
[log_event\(\)](#) (galaxy.jobs.deferred.FakeTrans method), [116](#)
[log_event\(\)](#) (galaxy.managers.context.ProvidesAppContext method), [150](#)
[log_tempfile](#) (module), [355](#)
[log_user_action_async\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), [328](#)
[LoggingProxy](#) (class in galaxy.model.orm.logging_connection_proxy), [147](#)
[login\(\)](#) (galaxy.webapps.demo_sequencer.controllers.common.CommonController method), [257](#)
[login\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), [328](#)
[logout\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), [328](#)
[Lped](#) (class in galaxy.datatypes.genetics), [52](#)
[LRUCache](#) (class in galaxy.util.lrucache), [222](#)
[lucene_search\(\)](#) (in module galaxy.webapps.galaxy.controllers.library_common), [316](#)
[LwrJobRunner](#) (class in galaxy.jobs.runners.lwr), [121](#)

M

[Maf](#) (class in galaxy.datatypes.sequence), [76](#)
[maf_index_by_uid\(\)](#) (in module galaxy.tools.util.maf_utilities), [212](#)
[MafCustomTrack](#) (class in galaxy.datatypes.sequence), [76](#)
[main\(\)](#) (in module galaxy.datatypes.converters.bgzip), [86](#)
[main\(\)](#) (in module galaxy.datatypes.converters.fastq_to_fqtc), [86](#)
[main\(\)](#) (in module galaxy.datatypes.converters.gff_to_interval_index_converter), [87](#)
[main\(\)](#) (in module galaxy.datatypes.converters.interval_to_coverage), [88](#)
[main\(\)](#) (in module galaxy.datatypes.converters.interval_to_fli), [88](#)
[main\(\)](#) (in module galaxy.datatypes.converters.interval_to_interval_index_converter), [88](#)
[main\(\)](#) (in module galaxy.datatypes.converters.interval_to_tabix_converter), [88](#)
[main\(\)](#) (in module galaxy.datatypes.converters.lped_to_fped_converter), [88](#)
[main\(\)](#) (in module galaxy.datatypes.converters.lped_to_pbed_converter), [89](#)
[main\(\)](#) (in module galaxy.datatypes.converters.pbed_ldreduced_converter), [89](#)
[main\(\)](#) (in module galaxy.datatypes.converters.pbed_to_lped_converter), [89](#)
[main\(\)](#) (in module galaxy.datatypes.converters.vcf_to_interval_index_converter), [90](#)
[main\(\)](#) (in module galaxy.datatypes.converters.vcf_to_vcf_bgzip), [90](#)
[main\(\)](#) (in module galaxy.datatypes.converters.wiggle_to_array_tree_converter), [90](#)
[main\(\)](#) (in module galaxy.datatypes.converters.wiggle_to_simple_converter), [90](#)
[make_body_iterable\(\)](#) (galaxy.web.framework.base.WebApplication method), [244](#)
[make_body_iterable\(\)](#) (galaxy.webapps.demo_sequencer.framework.WebApplication method), [258](#)
[make_copy\(\)](#) (galaxy.datatypes.metadata.FileParameter method), [67](#)
[make_copy\(\)](#) (galaxy.datatypes.metadata.MetadataParameter method), [68](#)
[make_dataset_public\(\)](#) (galaxy.security.GalaxyRBACAgent method), [166](#)
[make_dataset_public\(\)](#) (galaxy.security.RBACAgent method), [168](#)
[make_dict_copy\(\)](#) (galaxy.datatypes.metadata.MetadataCollection method), [68](#)
[make_file\(\)](#) (galaxy.web.framework.base.FieldStorage method), [242](#)
[make_filter\(\)](#) (in module galaxy.web.framework.middleware.translogger), [253](#)
[make_folder_public\(\)](#) (galaxy.security.GalaxyRBACAgent method), [166](#)
[make_folder_public\(\)](#) (galaxy.security.RBACAgent method), [168](#)

`make_html_peek_header()` (method), 315
`(galaxy.datatypes.tabular.Tabular method), 83`
`make_html_peek_rows()` (method), 302
`(galaxy.datatypes.tabular.Tabular method), 83`
`make_html_table()` (galaxy.datatypes.genetics.GenomeGraph method), 52
`make_html_table()` (galaxy.datatypes.genetics.RexpBase method), 54
`make_html_table()` (galaxy.datatypes.sequence.Maf method), 76
`make_html_table()` (galaxy.datatypes.tabular.Eland method), 80
`make_html_table()` (galaxy.datatypes.tabular.Tabular method), 83
`make_library_item_public()` (class in galaxy.jobs.deferred.manual_data_transfer),
`(galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315`
`make_library_public()` (galaxy.security.GalaxyRBACAgent method), 166
`make_library_public()` (galaxy.security.RBACAgent method), 168
`make_library_uploaded_dataset()` (in module galaxy.web.framework.middleware.static), 253
`(galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315`
`make_public()` (galaxy.managers.libraries.LibraryManager method), 155
`make_static()` (in module galaxy.web.framework.middleware.static), 253
`makeLDreduced()` (in module galaxy.datatypes.converters.pbед_ldreduced_converter), 89
`makeNode()` (galaxy.visualization.data_providers.phyloviz.base_provider.PhyloViz method), 227
`MalformedId`, 98
`MAList` (class in galaxy.datatypes.genetics), 52
`manage_addresses()` (galaxy.webapps.galaxy.controllers.user.User method), 328
`manage_datasets()` (galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin method), 323
`manage_repositories()` (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 302
`manage_repository()` (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 302
`manage_repository_tool_dependencies()` (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 302
`manage_roles_and_groups_for_user()` (galaxy.web.base.controllers.admin.Admin method), 240
`manage_template_inheritance()` (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 241
`manage_tool_dependencies()` (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 302
`manage_user_info()` (galaxy.webapps.galaxy.controllers.user.User method), 328
`manage_users_and_groups_for_quota()` (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 295
`manage_users_and_groups_for_role()` (galaxy.web.base.controllers.admin.Admin method), 240
`manage_users_and_roles_for_group()` (galaxy.web.base.controllers.admin.Admin method), 240
`ManualDataTransferPlugin` (class in galaxy.jobs.deferred.manual_data_transfer), 159
`map_library_datasets_to_lddas()` (in module galaxy.webapps.galaxy.controllers.library_common), 316
`map_wrap()` (in module galaxy.objectstore.s3_multipart_upload), 159
`mapping_controller()` (galaxy.webapps.galaxy.api.datatypes.DatatypesController method), 9, 265
`MAPPING_PRESET` (galaxy.tools.parameters.sanitize.ToolParameterSanitizer attribute), 205
`mark_as_failed()` (galaxy.jobs.runners.AsynchronousJobRunner method), 117
`mark_as_finished()` (galaxy.jobs.runners.AsynchronousJobRunner method), 117
`mark_as_queued()` (galaxy.jobs.runners.BaseJobRunner method), 118
`mark_as_resubmitted()` (galaxy.jobs.runners.BaseJobRunner method), 118
`mark_as_resubmitted()` (galaxy.jobs.runners.BaseJobRunner method), 118
`mark_deleted()` (galaxy.model.Dataset method), 123
`mark_deleted()` (galaxy.model.DatasetInstance method), 123
`mark_deleted()` (galaxy.model.Job method), 133
`mark_tool_deleted()` (galaxy.web.base.controllers.admin.Admin method), 240
`mark_tool_deleted()` (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 295
`mark_role_deleted()` (galaxy.web.base.controllers.admin.Admin method), 240
`mark_undeleted()` (galaxy.model.DatasetInstance method), 125
`mark_unhidden()` (galaxy.model.DatasetInstance method), 125
`mark_user_deleted()` (galaxy.web.base.controllers.admin.Admin method), 241

[marshal\(\)](#) (galaxy.datatypes.metadata.FileParameter class method), 67
[marshal\(\)](#) (galaxy.datatypes.metadata.MetadataParameter class method), 68
[marshal\(\)](#) (galaxy.datatypes.metadata.PythonObjectParameter class method), 69
[marshal\(\)](#) (galaxy.datatypes.metadata.RangeParameter class method), 69
[marshal\(\)](#) (galaxy.datatypes.metadata.SelectParameter class method), 69
[match_collections\(\)](#) (galaxy.managers.collections.DatasetCollectionManager method), 150
[match_collections\(\)](#) (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 187
[match_collections\(\)](#) (galaxy.tools.parameters.basic.DataToolParameter method), 187
[match_datasets\(\)](#) (galaxy.tools.parameters.basic.DataToolParameter method), 187
[match_multirun_collections\(\)](#) (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 187
[matches_any\(\)](#) (galaxy.datatypes.data.Data method), 48
[MAX_LINES_RETURNED](#) (galaxy.visualization.data_providers.basic.ColumnDataProvider attribute), 226
[max_optional_metadata_filesize](#) (galaxy.datatypes.data.Data attribute), 49
[MAX_PRINT_ERROR_BYTES](#) (galaxy_utils.sequence.fastq.fastqVerboseErrorReader attribute), 353
[MAX_SEQUENCE_SIZE](#) (galaxy.tools.util.maf_utilities.RegionAlignment attribute), 210
[md5\(\)](#) (in module galaxy.web.framework.helpers), 245
[Memdump](#) (class in galaxy.util.memdump), 222
[memdump\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 241
[MEMEXml](#) (class in galaxy.datatypes.xml), 85
[merge\(\)](#) (galaxy.datatypes.data.Data static method), 49
[merge\(\)](#) (galaxy.datatypes.tabular.Sam static method), 82
[merge\(\)](#) (galaxy.datatypes.xml.GenericXml static method), 85
[merge_sorted_iterables\(\)](#) (in module galaxy.util), 214
[merge_tool_data_table\(\)](#) (galaxy.tools.data.TabularToolDataTable method), 181
[merge_tool_data_table\(\)](#) (galaxy.tools.data.ToolDataTable method), 182
[MessageException](#), 99
[metadata](#) (galaxy.model.DatasetInstance attribute), 126
[metadata_isdir\(\)](#) (pkg_resources.IMetadataProvider method), 363
[metadata_isdir\(\)](#) (pkg_resources.NullProvider method), 365
[metadata_listdir\(\)](#) (pkg_resources.IMetadataProvider method), 363
[metadata_listdir\(\)](#) (pkg_resources.NullProvider method), 365
[metadata_spec](#) (galaxy.datatypes.assembly.Amos attribute), 41
[metadata_spec](#) (galaxy.datatypes.assembly.Roadmaps attribute), 42
[metadata_spec](#) (galaxy.datatypes.assembly.Sequences attribute), 42
[metadata_spec](#) (galaxy.datatypes.assembly.Velvet attribute), 42
[metadata_spec](#) (galaxy.datatypes.binary.Abl attribute), 42
[metadata_spec](#) (galaxy.datatypes.binary.Bam attribute), 43
[metadata_spec](#) (galaxy.datatypes.binary.Bcf attribute), 43
[metadata_spec](#) (galaxy.datatypes.binary.BigBed attribute), 43
[metadata_spec](#) (galaxy.datatypes.binary.BigWig attribute), 44
[metadata_spec](#) (galaxy.datatypes.binary.Binary attribute), 44
[metadata_spec](#) (galaxy.datatypes.binary.GeminiSQLite attribute), 44
[metadata_spec](#) (galaxy.datatypes.binary.GenericAsn1Binary attribute), 44
[metadata_spec](#) (galaxy.datatypes.binary.H5 attribute), 45
[metadata_spec](#) (galaxy.datatypes.binary.Scf attribute), 45
[metadata_spec](#) (galaxy.datatypes.binary.Sff attribute), 45
[metadata_spec](#) (galaxy.datatypes.binary.SQLite attribute), 45
[metadata_spec](#) (galaxy.datatypes.binary.TwoBit attribute), 45
[metadata_spec](#) (galaxy.datatypes.binary.Xlsx attribute), 46
[metadata_spec](#) (galaxy.datatypes.chrominfo.ChromInfo attribute), 46
[metadata_spec](#) (galaxy.datatypes.coverage.LastzCoverage attribute), 46
[metadata_spec](#) (galaxy.datatypes.data.Data attribute), 49
[metadata_spec](#) (galaxy.datatypes.data.GenericAsn1 attribute), 49
[metadata_spec](#) (galaxy.datatypes.data.LineCount attribute), 49
[metadata_spec](#) (galaxy.datatypes.data.Newick attribute), 50
[metadata_spec](#) (galaxy.datatypes.data.Nexus attribute), 50
[metadata_spec](#) (galaxy.datatypes.data.Text attribute), 50
[metadata_spec](#) (galaxy.datatypes.genetics.Affybatch attribute), 51
[metadata_spec](#) (galaxy.datatypes.genetics.Eigenstratgeno attribute), 51

metadata_spec (galaxy.datatypes.genetics.Eigenstratpca attribute), 51	metadata_spec (galaxy.datatypes.images.Pbm attribute), 58
metadata_spec (galaxy.datatypes.genetics.Eset attribute), 51	metadata_spec (galaxy.datatypes.images.Pcd attribute), 58
metadata_spec (galaxy.datatypes.genetics.Fped attribute), 52	metadata_spec (galaxy.datatypes.images.Pcx attribute), 58
metadata_spec (galaxy.datatypes.genetics.Fphe attribute), 52	metadata_spec (galaxy.datatypes.images.Pdf attribute), 58
metadata_spec (galaxy.datatypes.genetics.GenomeGraphs attribute), 52	metadata_spec (galaxy.datatypes.images.Pgm attribute), 58
metadata_spec (galaxy.datatypes.genetics.IdIndep attribute), 55	metadata_spec (galaxy.datatypes.images.Png attribute), 58
metadata_spec (galaxy.datatypes.genetics.Lped attribute), 52	metadata_spec (galaxy.datatypes.images.Ppm attribute), 58
metadata_spec (galaxy.datatypes.genetics.MAlist attribute), 53	metadata_spec (galaxy.datatypes.images.Psd attribute), 59
metadata_spec (galaxy.datatypes.genetics.Pbed attribute), 53	metadata_spec (galaxy.datatypes.images.Rast attribute), 59
metadata_spec (galaxy.datatypes.genetics.Phe attribute), 53	metadata_spec (galaxy.datatypes.images.Rgb attribute), 59
metadata_spec (galaxy.datatypes.genetics.Phenotype attribute), 53	metadata_spec (galaxy.datatypes.images.Tiff attribute), 59
metadata_spec (galaxy.datatypes.genetics.Pphe attribute), 53	metadata_spec (galaxy.datatypes.images.Xbm attribute), 59
metadata_spec (galaxy.datatypes.genetics.RexpBase attribute), 54	metadata_spec (galaxy.datatypes.images.Xpm attribute), 59
metadata_spec (galaxy.datatypes.genetics.Rgenetics attribute), 54	metadata_spec (galaxy.datatypes.interval.Bed attribute), 60
metadata_spec (galaxy.datatypes.genetics.rgFeatureList attribute), 55	metadata_spec (galaxy.datatypes.interval.Bed12 attribute), 60
metadata_spec (galaxy.datatypes.genetics.rgSampleList attribute), 55	metadata_spec (galaxy.datatypes.interval.Bed6 attribute), 60
metadata_spec (galaxy.datatypes.genetics.rgTabList attribute), 56	metadata_spec (galaxy.datatypes.interval.BedGraph attribute), 61
metadata_spec (galaxy.datatypes.genetics.SNPMatrix attribute), 55	metadata_spec (galaxy.datatypes.interval.BedStrict attribute), 61
metadata_spec (galaxy.datatypes.genetics.Snptest attribute), 55	metadata_spec (galaxy.datatypes.interval.ChromatinInteractions attribute), 61
metadata_spec (galaxy.datatypes.images.Bmp attribute), 56	metadata_spec (galaxy.datatypes.interval.CustomTrack attribute), 61
metadata_spec (galaxy.datatypes.images.Eps attribute), 56	metadata_spec (galaxy.datatypes.interval.ENCODEPeak attribute), 62
metadata_spec (galaxy.datatypes.images.Gif attribute), 56	metadata_spec (galaxy.datatypes.interval.Gff attribute), 63
metadata_spec (galaxy.datatypes.images.Gmaj attribute), 56	metadata_spec (galaxy.datatypes.interval.Gff3 attribute), 63
metadata_spec (galaxy.datatypes.images.Html attribute), 57	metadata_spec (galaxy.datatypes.interval.Gtf attribute), 64
metadata_spec (galaxy.datatypes.images.Im attribute), 57	metadata_spec (galaxy.datatypes.interval.Interval attribute), 65
metadata_spec (galaxy.datatypes.images.Image attribute), 57	metadata_spec (galaxy.datatypes.interval.Wiggle attribute), 66
metadata_spec (galaxy.datatypes.images.Jpg attribute), 57	metadata_spec (galaxy.datatypes.ngsindex.BowtieBaseIndex attribute), 70
metadata_spec (galaxy.datatypes.images.Laj attribute), 57	

[metadata_spec \(galaxy.datatypes.ngsindex.BowtieColorIndex attribute\), 70](#)
[metadata_spec \(galaxy.datatypes.ngsindex.BowtieIndex attribute\), 70](#)
[metadata_spec \(galaxy.datatypes.qualityscore.QualityScore attribute\), 70](#)
[metadata_spec \(galaxy.datatypes.qualityscore.QualityScore attribute\), 71](#)
[metadata_spec \(galaxy.datatypes.qualityscore.QualityScore attribute\), 71](#)
[metadata_spec \(galaxy.datatypes.qualityscore.QualityScore attribute\), 71](#)
[metadata_spec \(galaxy.datatypes.qualityscore.QualityScore attribute\), 71](#)
[metadata_spec \(galaxy.datatypes.sequence.Alignment attribute\), 73](#)
[metadata_spec \(galaxy.datatypes.sequence.Axt attribute\), 73](#)
[metadata_spec \(galaxy.datatypes.sequence.csFasta attribute\), 77](#)
[metadata_spec \(galaxy.datatypes.sequence.Fasta attribute\), 73](#)
[metadata_spec \(galaxy.datatypes.sequence.Fastq attribute\), 74](#)
[metadata_spec \(galaxy.datatypes.sequence.FastqCSSanger attribute\), 75](#)
[metadata_spec \(galaxy.datatypes.sequence.FastqIllumina attribute\), 75](#)
[metadata_spec \(galaxy.datatypes.sequence.FastqSanger attribute\), 75](#)
[metadata_spec \(galaxy.datatypes.sequence.FastqSolexa attribute\), 75](#)
[metadata_spec \(galaxy.datatypes.sequence.Lav attribute\), 75](#)
[metadata_spec \(galaxy.datatypes.sequence.Maf attribute\), 76](#)
[metadata_spec \(galaxy.datatypes.sequence.MafCustomTrack attribute\), 76](#)
[metadata_spec \(galaxy.datatypes.sequence.Sequence attribute\), 77](#)
[metadata_spec \(galaxy.datatypes.sequence.SequenceSplitLocations attribute\), 77](#)
[metadata_spec \(galaxy.datatypes.tabular.Eland attribute\), 80](#)
[metadata_spec \(galaxy.datatypes.tabular.ElandMulti attribute\), 81](#)
[metadata_spec \(galaxy.datatypes.tabular.FeatureLocationIndex attribute\), 81](#)
[metadata_spec \(galaxy.datatypes.tabular.Pileup attribute\), 81](#)
[metadata_spec \(galaxy.datatypes.tabular.Sam attribute\), 82](#)
[metadata_spec \(galaxy.datatypes.tabular.Tabular attribute\), 83](#)
[metadata_spec \(galaxy.datatypes.tabular.Taxonomy attribute\), 84](#)
[metadata_spec \(galaxy.datatypes.tabular.Vcf attribute\), 84](#)
[metadata_spec \(galaxy.datatypes.tracks.GeneTrack attribute\), 84](#)
[metadata_spec \(galaxy.datatypes.xml.CisML attribute\), 84](#)
[metadata_spec \(galaxy.datatypes.xml.GenericXml attribute\), 85](#)
[metadata_spec \(galaxy.datatypes.xml.MEMEXml attribute\), 85](#)
[metadata_spec \(galaxy.datatypes.xml.Owl attribute\), 85](#)
[metadata_spec \(galaxy.datatypes.xml.Phyloxml attribute\), 86](#)
[MetadataCollection \(class in galaxy.datatypes.metadata\), 67](#)
[MetadataElement \(in module galaxy.datatypes.metadata\), 68](#)
[MetadataElementSpec \(class in galaxy.datatypes.metadata\), 68](#)
[MetadataFile \(class in galaxy.model\), 136](#)
[MetadataInDataTableColumnValidator \(class in galaxy.tools.parameters.validation\), 207](#)
[MetadataInFileColumnValidator \(class in galaxy.tools.parameters.validation\), 208](#)
[MetadataParameter \(class in galaxy.datatypes.metadata\), 68](#)
[MetadataSpecCollection \(class in galaxy.datatypes.metadata\), 69](#)
[MetadataTempFile \(class in galaxy.datatypes.metadata\), 69](#)
[MetadataToolOutputAction \(class in galaxy.tools.parameters.output\), 202](#)
[MetadataType \(class in galaxy.model.custom_types\), 145](#)
[MetadataValidator \(class in galaxy.tools.parameters.validation\), 208](#)
[MetadataValueFilter \(class in galaxy.tools.parameters.output\), 203](#)
[metrics \(galaxy.model.HasJobMetrics attribute\), 128](#)
[MetricsController \(class in galaxy.webapps.galaxy.api.metrics\), 29, 285](#)
[migrate_to_current_version\(\) \(in module galaxy.model.migrate.check\), 147](#)
[mime_type\(\) \(galaxy.datatypes.display_applications.parameters.DisplayParameters method\), 92](#)
[mime_type\(\) \(galaxy.datatypes.display_applications.parameters.DisplayParameters method\), 92](#)
[mimeparse \(module\), 355](#)
[missing_meta\(\) \(galaxy.datatypes.data.Data method\), 49](#)
[missing_meta\(\) \(galaxy.model.DatasetInstance method\), 126](#)

missing_meta() (galaxy.util.none_like.NoneDataset method), 222
 MissingToolException, 345
 mkstemp() (log_tempfile.TempFile method), 355
 mkstemp_in() (in module galaxy.util), 214
 Mobile (class in galaxy.webapps.galaxy.controllers.mobile), 316
 model (galaxy.managers.context.ProvidesAppContext attribute), 150
 model_class (galaxy.web.framework.helpers.grids.Grid attribute), 248
 model_class (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 332
 model_class (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 333
 model_class (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 333
 model_class (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 335
 model_class (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 335
 model_class (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAttributeListGrid attribute), 305
 model_class (galaxy.webapps.galaxy.controllers.external_service.ExternalServiceGrid attribute), 307
 model_class (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 308
 model_class (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid attribute), 309
 model_class (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 311
 model_class (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid attribute), 312
 model_class (galaxy.webapps.galaxy.controllers.library.LibraryListGrid attribute), 313
 model_class (galaxy.webapps.galaxy.controllers.library_admin.LibraryAdminGrid attribute), 314
 model_class (galaxy.webapps.galaxy.controllers.page.HistoryDatasetAssociationSelectionGrid attribute), 316
 model_class (galaxy.webapps.galaxy.controllers.page.HistorySelectionGrid attribute), 317
 model_class (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid attribute), 317
 model_class (galaxy.webapps.galaxy.controllers.page.PageListGrid attribute), 319
 model_class (galaxy.webapps.galaxy.controllers.page.PageSelectionGrid attribute), 319
 model_class (galaxy.webapps.galaxy.controllers.page.VisualizationSelectionGrid attribute), 319
 model_class (galaxy.webapps.galaxy.controllers.page.WorkflowSelectionGrid attribute), 319
 model_class (galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid attribute), 321
 model_class (galaxy.webapps.galaxy.controllers.requests_admin.DataTransferGrid attribute), 322
 model_class (galaxy.webapps.galaxy.controllers.requests_common.RequestCommonGrid attribute), 324
 model_class (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid attribute), 329
 model_class (galaxy.webapps.galaxy.controllers.visualization.HistoryDatasetAttributeListGrid attribute), 331
 model_class (galaxy.webapps.galaxy.controllers.visualization.HistorySelectionGrid attribute), 332
 model_class (galaxy.webapps.galaxy.controllers.visualization.LibrarySelectionGrid attribute), 332
 model_class (galaxy.webapps.galaxy.controllers.visualization.TracksterSelectionGrid attribute), 333
 model_class (galaxy.webapps.galaxy.controllers.visualization.VisualizationGrid attribute), 333
 model_class (galaxy.webapps.galaxy.controllers.visualization.VisualizationListGrid attribute), 335
 model_class (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAttributeListGrid attribute), 335
 model_class (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid attribute), 342
 model_class (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute), 342
 model_class (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid attribute), 344
 model_class (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid attribute), 344
 ModelManager (class in galaxy.managers.base), 148
 ModelSerializer (class in galaxy.managers.base), 148
 module_path (pkg_resources.EmptyProvider attribute), 148
 modularize() (galaxy.util.inflection.Base method), 219
 modularize() (galaxy.util.inflection.Inflector method), 220
 monitor() (galaxy.jobs.handler.JobHandlerStopQueue method), 112
 monitor() (galaxy.jobs.runners.AsynchronousJobRunner method), 112
 monitor_job() (galaxy.jobs.runners.AsynchronousJobRunner method), 118
 monitor_repository_installation() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 302
 monitor_step() (galaxy.jobs.handler.JobHandlerStopQueue method), 112
 monitor_toolshed() (galaxy.managers.histories.HistoryManager method), 154
 monitor_toolshed() (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315
 mp_from_ids() (in module galaxy.util), 214
 mp_from_ids() (in module galaxy.util), 214
 multipart_upload() (in module galaxy.util), 160

MulticolFilterColumn (class in galaxy.web.framework.helpers.grids), 249
 multimap() (in galaxy.objectstore.s3_multipart_upload), 160
 multipart_upload() (in galaxy.objectstore.s3_multipart_upload), 160
 MultipleSplitterFilter (class in galaxy.tools.parameters.dynamic_options), 198
 MultipleSplitterFilter (class in galaxy.tools.parameters.output), 203
N
 name (galaxy.jobs.actions.post.ChangeDatatypeAction attribute), 114
 name (galaxy.jobs.actions.post.ColumnSetAction attribute), 114
 name (galaxy.jobs.actions.post.DefaultJobAction attribute), 115
 name (galaxy.jobs.actions.post.DeleteDatasetAction attribute), 115
 name (galaxy.jobs.actions.post.DeleteIntermediatesAction attribute), 115
 name (galaxy.jobs.actions.post.EmailAction attribute), 115
 name (galaxy.jobs.actions.post.HideDatasetAction attribute), 115
 name (galaxy.jobs.actions.post.RenameDatasetAction attribute), 115
 name (galaxy.jobs.actions.post.SetMetadataAction attribute), 116
 name (galaxy.model.LibraryDataset attribute), 135
 name (galaxy.workflow.modules.InputDataCollectionModule attribute), 345
 name (galaxy.workflow.modules.InputDataModule attribute), 345
 name (galaxy.workflow.modules.PauseModule attribute), 346
 name() (galaxy.datatypes.util.gff_util.GFFFeature method), 93
 name_autocomplete_data() (galaxy.web.base.controllers.admin.Admin method), 241
 name_autocomplete_data() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
 NameColumn (class in galaxy.webapps.galaxy.controllers.history), 312
 NameColumn (class in galaxy.webapps.galaxy.controllers.visualization), 332
 NamedTemporaryFile() (log_tempfile.TempFile method), 355
 need_late_validation() (galaxy.tools.parameters.basic.ColumnListParameter method), 187
 need_late_validation() (galaxy.tools.parameters.basic.DataToolParameter method), 187
 need_late_validation() (galaxy.tools.parameters.basic.SelectToolParameter method), 195
 need_late_validation() (galaxy.tools.parameters.basic.ToolParameter method), 196
 NestedObjectStore (class in galaxy.objectstore), 157
 NEVER_SANITIZE (galaxy.util.Params attribute), 213
 new() (galaxy.forms.forms.FormDefinitionAddressFieldFactory method), 104
 new() (galaxy.forms.forms.FormDefinitionFactory method), 104
 new() (galaxy.forms.forms.FormDefinitionFieldFactory method), 105
 new() (galaxy.forms.forms.FormDefinitionHistoryFieldFactory method), 105
 new() (galaxy.forms.forms.FormDefinitionPasswordFieldFactory method), 105
 new() (galaxy.forms.forms.FormDefinitionSelectFieldFactory method), 105
 new() (galaxy.forms.forms.FormDefinitionTextFieldFactory method), 105
 new() (galaxy.forms.forms.FormDefinitionWorkflowFieldFactory method), 105
 new() (galaxy.forms.forms.FormDefinitionWorkflowMappingFieldFactory method), 106
 new() (galaxy.jobs.transfer_manager.TransferManager method), 113
 new() (galaxy.sample_tracking.request_types.RequestTypeFactory method), 162
 new() (galaxy.sample_tracking.sample.SampleStateFactory method), 163
 new() (galaxy.workflow.modules.SimpleWorkflowModule class method), 346
 new() (galaxy.workflow.modules.ToolModule class method), 347
 new() (galaxy.workflow.modules.WorkflowModule class method), 349
 new() (galaxy.workflow.modules.WorkflowModuleFactory method), 349
 new_address() (galaxy.webapps.galaxy.controllers.user.User method), 328
 new_file() (galaxy.datatypes.metadata.FileParameter method), 67
 new_file_path() (galaxy.jobs.ComputeEnvironment method), 106
 new_file_path() (galaxy.jobs.SharedComputeEnvironment method), 110
 new_provider_from_identifier() (galaxy.openid.providers.OpenIDProviders

method), 160

new_secure_hash() (in module galaxy.util.hash_util), 218

new_state() (galaxy.tools.Tool method), 174

new_upload() (in module galaxy.tools.actions.upload_common), 181

Newick (class in galaxy.datatypes.data), 49

Newick_Parser (class in galaxy.visualization.data_providers.phyloviz.newickparser), 228

next() (galaxy.datatypes.util.gff_util.GFFReaderWrapper method), 94

next() (galaxy.tools.exception_handling.UCSCOutWrapper method), 178

next() (galaxy_utils.sequence.fasta.fastaReader method), 350

next() (galaxy_utils.sequence.fastq.fastqReader method), 352

next() (galaxy_utils.sequence.fastq.fastqVerboseErrorReader method), 353

next() (galaxy_utils.sequence.vcf.Reader method), 355

Nexus (class in galaxy.datatypes.data), 50

Nexus_Parser (class in galaxy.visualization.data_providers.phyloviz.nexusparser), 228

nice_size() (in module galaxy.util), 214

nice_size() (in module galaxy.webapps.reports.controllers.system), 343

nice_total_disk_usage (galaxy.model.User attribute), 142

NO_PROVIDER_ID (galaxy.openid.providers.OpenIDProviders attribute), 160

NoConverterException, 136

Node (class in galaxy.visualization.data_providers.phyloviz.baseparser), 227

NoneDataset (class in galaxy.util.none_like), 222

NoopHandler (class in galaxy.jobs.manager), 113

NoopQueue (class in galaxy.jobs), 110

NoOptionsValidator (class in galaxy.tools.parameters.validation), 208

NoQuotaAgent (class in galaxy.quota), 161

normalize_path() (in module pkg_resources), 363

normalize_runtime_state() (galaxy.workflow.modules.SimpleWorkflowModule method), 346

normalize_runtime_state() (galaxy.workflow.modules.ToolModule method), 347

not_implemented() (galaxy.web.base.controller.BaseAPIController method), 234

NotImplemented, 99

nottest() (in module galaxy.tools.test), 178

now() (in module galaxy.model.mapping), 147

NullDependencyManager (class in galaxy.tools.deps), 183

NullProvider (class in pkg_resources), 365

NullToolOutputActionOption (class in galaxy.tools.parameters.output), 203

num_page_links (galaxy.web.framework.helpers.grids.Grid attribute), 248

num_rows_per_page (galaxy.web.framework.helpers.grids.Grid attribute), 248

num_rows_per_page (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 296

num_rows_per_page (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 297

num_rows_per_page (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 299

num_rows_per_page (galaxy.webapps.galaxy.controllers.admin.ToolVersion attribute), 299

num_rows_per_page (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 301

num_rows_per_page (galaxy.webapps.galaxy.controllers.dataset.HistoryData attribute), 305

num_rows_per_page (galaxy.webapps.galaxy.controllers.external_service.E attribute), 307

num_rows_per_page (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 308

num_rows_per_page (galaxy.webapps.galaxy.controllers.history.HistoryAll attribute), 309

num_rows_per_page (galaxy.webapps.galaxy.controllers.history.HistoryList attribute), 311

num_rows_per_page (galaxy.webapps.galaxy.controllers.library.LibraryList attribute), 313

num_rows_per_page (galaxy.webapps.galaxy.controllers.library_admin.Lib attribute), 314

num_rows_per_page (galaxy.webapps.galaxy.controllers.page.ItemSelection attribute), 317

num_rows_per_page (galaxy.webapps.galaxy.controllers.request_type.Requ attribute), 321

num_rows_per_page (galaxy.webapps.galaxy.controllers.requests_admin.D attribute), 322

num_rows_per_page (galaxy.webapps.galaxy.controllers.requests_common attribute), 325

num_rows_per_page (galaxy.webapps.galaxy.controllers.visualization.Histo attribute), 332

num_rows_per_page (galaxy.webapps.galaxy.controllers.visualization.Libra attribute), 332

num_rows_per_page (galaxy.webapps.reports.controllers.jobs.SpecifiedData attribute), 341

num_rows_per_page (galaxy.webapps.reports.controllers.sample_tracking.S attribute), 342

num_rows_per_page (galaxy.webapps.reports.controllers.workflows.Specifi attribute), 344

O

object_store (galaxy.model.Dataset attribute), 123

object_to_string() (in module galaxy.util), 214

ObjectAttributeInvalidException, 99

- ObjectAttributeMissingException, 99
- ObjectInvalid, 99
- ObjectNotFound, 99
- ObjectStore (class in galaxy.objectstore), 158
- ObjectStorePopulator (class in galaxy.tools.actions), 179
- obtain() (pkg_resources.Environment method), 357
- odict (class in galaxy.util.odict), 223
- oidlog() (in module galaxy.web.framework.openid_manager), 244
- ok_to_display() (galaxy.security.GalaxyRBACAgent method), 166
- old_change_datatype() (galaxy.datatypes.registry.Registry method), 73
- on_text_for_names() (in module galaxy.tools.actions), 179
- open_folder() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed attribute), 302
- open_folder() (galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin attribute), 323
- open_or_build_maf_index() (in module galaxy.tools.util.maf_utilities), 212
- openid_associate() (galaxy.webapps.galaxy.controllers.user.User attribute), 328
- openid_auth() (galaxy.webapps.galaxy.controllers.user.User attribute), 328
- openid_disassociate() (galaxy.webapps.galaxy.controllers.user.User attribute), 328
- openid_manage() (galaxy.webapps.galaxy.controllers.user.User attribute), 328
- openid_process() (galaxy.webapps.galaxy.controllers.user.User attribute), 328
- OpenIDManager (class in galaxy.web.framework.openid_manager), 244
- OpenIDProvider (class in galaxy.openid.providers), 160
- OpenIDProviders (class in galaxy.openid.providers), 160
- OpenObject (class in galaxy.webapps.galaxy.controllers.visualization), 332
- operation (galaxy.webapps.galaxy.controllers.requests.UserRequestsGrid attribute), 321
- operation (galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin attribute), 322
- operations (galaxy.web.framework.helpers.grids.Grid attribute), 248
- operations (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 296
- operations (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 297
- operations (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 299
- operations (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 299
- operations (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 301
- operations (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociation attribute), 306
- operations (galaxy.webapps.galaxy.controllers.external_service.ExternalService attribute), 307
- operations (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 308
- operations (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedCollection attribute), 309
- operations (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 311
- operations (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid attribute), 312
- operations (galaxy.webapps.galaxy.controllers.page.PageListGrid attribute), 319
- operations (galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid attribute), 321
- operations (galaxy.webapps.galaxy.controllers.requests.UserRequestsGrid attribute), 321
- operations (galaxy.webapps.galaxy.controllers.requests_admin.AdminRequests attribute), 322
- operations (galaxy.webapps.galaxy.controllers.requests_admin.DataTransfer attribute), 322
- operations (galaxy.webapps.galaxy.controllers.requests_common.RequestsCollection attribute), 325
- operations (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid attribute), 329
- operations (galaxy.webapps.galaxy.controllers.visualization.VisualizationList attribute), 335
- operations (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAll attribute), 335
- operations (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowList attribute), 336
- option_type (in module galaxy.tools.parameters.output), 205
- optionxform() (galaxy.eggs.CaseSensitiveConfigParser method), 94
- ordinalize() (galaxy.util.inflection.Base method), 219
- ordinalize() (galaxy.util.inflection.Inflector method), 220
- orient_by_region() (in module galaxy.tools.util.maf_utilities), 212
- output_paths() (galaxy.jobs.SharedComputeEnvironment attribute), 106
- output_paths() (galaxy.jobs.SharedComputeEnvironment attribute), 110
- OutputParameterJSONTool (class in galaxy.tools), 171
- outputs() (galaxy.webapps.galaxy.api.jobs.JobController method), 21, 277
- Owl (class in galaxy.datatypes.xml), 85
- OwlGridAnnotationColumn (class in galaxy.web.framework.helpers.grids), 250
- OwlGridAnnotationColumn (class in galaxy.web.framework.helpers.grids), 250

P

- pack_scripts() (in module galaxy.webapps.galaxy.buildapp), 259
- package_tool() (galaxy.tools.ToolBox method), 176
- package_tool() (galaxy.web.base.controllers.admin.Admin method), 241
- Page (class in galaxy.model), 136
- PageAllPublishedGrid (class in galaxy.webapps.galaxy.controllers.page), 317
- PageAnnotationAssociation (class in galaxy.model), 136
- PageController (class in galaxy.webapps.galaxy.controllers.page), 317
- PageListGrid (class in galaxy.webapps.galaxy.controllers.page), 318
- PageListGrid.URLColumn (class in galaxy.webapps.galaxy.controllers.page), 319
- PageRatingAssociation (class in galaxy.model), 137
- PageRevision (class in galaxy.model), 137
- PageRevisionsController (class in galaxy.webapps.galaxy.api.page_revisions), 30, 285
- PagesController (class in galaxy.webapps.galaxy.api.pages), 30, 286
- PageSelectionGrid (class in galaxy.webapps.galaxy.controllers.page), 319
- PageTagAssociation (class in galaxy.model), 137
- PageUserShareAssociation (class in galaxy.model), 137
- ParallelismInfo (class in galaxy.jobs), 110
- param_names() (galaxy.tools.test.ParamContext method), 178
- param_names() (galaxy.tools.test.RootParamContext method), 178
- ParamContext (class in galaxy.tools.test), 178
- Params (class in galaxy.util), 213
- params_from_strings() (galaxy.tools.Tool method), 174
- params_from_strings() (in module galaxy.tools.parameters), 185
- params_to_incoming() (in module galaxy.tools.parameters), 185
- params_to_strings() (galaxy.tools.Tool method), 174
- params_to_strings() (in module galaxy.tools.parameters), 185
- params_with_missing_data_table_entry (galaxy.tools.Tool attribute), 174
- params_with_missing_index_file (galaxy.tools.Tool attribute), 174
- ParamsWithSpecs (class in galaxy.util), 213
- ParamValueFilter (class in galaxy.tools.parameters.dynamic_options), 198
- ParamValueToolOutputActionOptionFilter (class in galaxy.tools.parameters.output), 203
- parent (galaxy.datatypes.metadata.MetadataCollection attribute), 68
- parent_library (galaxy.model.LibraryFolder attribute), 136
- parse() (galaxy.eggs.Crate method), 95
- parse() (galaxy.eggs.dist.DistScrambleCrate method), 96
- parse() (galaxy.eggs.scramble.ScrambleCrate method), 96
- parse() (galaxy.sample_tracking.data_transfer.DataTransferFactory method), 161
- parse() (galaxy.sample_tracking.data_transfer.FtpDataTransferFactory method), 161
- parse() (galaxy.sample_tracking.data_transfer.HttpDataTransferFactory method), 162
- parse() (galaxy.sample_tracking.data_transfer.ScipDataTransferFactory method), 162
- parse() (galaxy.sample_tracking.external_service_types.ExternalServiceType method), 162
- parse() (galaxy.tools.SetParamAction static method), 171
- parse() (galaxy.tools.Tool method), 174
- parse() (galaxy.tools.TracksterConfig static method), 177
- parse() (pkg_resources.EntryPoint class method), 361
- parse() (pkg_resources.Requirement static method), 361
- parse_cast_attribute() (in module galaxy.tools.parameters.output), 205
- parse_citation() (galaxy.managers.citations.CitationsManager method), 149
- parse_citation() (in module galaxy.managers.citations), 149
- parse_cols_arg() (in module galaxy.tools.util.galaxyops), 212
- parse_column_definitions() (galaxy.tools.parameters.dynamic_options.DynamicOptions method), 198
- parse_column_spec() (galaxy.tools.data.TabularToolDataTable method), 181
- parse_compare_type() (in module galaxy.tools.parameters.output), 205
- parse_data_transfer_settings() (galaxy.sample_tracking.external_service_types.ExternalServiceType method), 162
- parse_destination_params() (galaxy.jobs.runners.BaseJobRunner method), 118
- parse_destination_params() (galaxy.jobs.runners.cli.ShellJobRunner method), 119
- parse_egg_section() (galaxy.eggs.Crate method), 95
- parse_egg_section() (galaxy.eggs.dist.DistScrambleCrate method), 96

[parse_egg_section\(\)](#) (galaxy.eggs.scramble.ScrambleCrate method), 96
[parse_error_level\(\)](#) (galaxy.tools.Tool method), 174
[parse_file_fields\(\)](#) (galaxy.tools.data.TabularToolDataTable method), 182
[parse_file_fields\(\)](#) (galaxy.tools.parameters.dynamic_options.DynamicOptions method), 198
[parse_gff_attributes\(\)](#) (in module galaxy.datatypes.util.gff_util), 94
[parse_group\(\)](#) (pkg_resources.EntryPoint class method), 361
[parse_help\(\)](#) (galaxy.tools.Tool method), 174
[parse_input_elem\(\)](#) (galaxy.tools.Tool method), 174
[parse_input_page\(\)](#) (galaxy.tools.Tool method), 174
[parse_inputs\(\)](#) (galaxy.tools.DataSourceTool method), 170
[parse_inputs\(\)](#) (galaxy.tools.Tool method), 174
[parse_map\(\)](#) (pkg_resources.EntryPoint class method), 361
[parse_media_range\(\)](#) (in module mimeparse), 356
[parse_mime_type\(\)](#) (in module mimeparse), 356
[parse_output_elems\(\)](#) (in module galaxy.tools.test), 178
[parse_outputs\(\)](#) (galaxy.tools.Tool method), 174
[parse_param_elem\(\)](#) (galaxy.tools.Tool method), 174
[parse_param_elem\(\)](#) (in module galaxy.tools.test), 178
[parse_redirect_url\(\)](#) (galaxy.tools.Tool method), 174
[parse_request_elem\(\)](#) (galaxy.webapps.demo_sequencer.registry.Registry method), 257
[parse_request_tup\(\)](#) (galaxy.webapps.demo_sequencer.controller.Controller method), 257
[parse_requirements\(\)](#) (in module pkg_resources), 362
[parse_row\(\)](#) (galaxy.datatypes.util.gff_util.GFFIntervalToBEDReader method), 93
[parse_row\(\)](#) (galaxy.datatypes.util.gff_util.GFFReaderWrapper method), 94
[parse_run_details\(\)](#) (galaxy.sample_tracking.external_service_types.ExternalServiceType method), 162
[parse_run_details_results\(\)](#) (galaxy.sample_tracking.external_service_types.ExternalServiceType method), 162
[parse_species_option\(\)](#) (in module galaxy.tools.util.maf_utilities), 212
[parse_stdio\(\)](#) (galaxy.tools.Tool method), 175
[parse_stdio_exit_codes\(\)](#) (galaxy.tools.Tool method), 175
[parse_stdio_regexes\(\)](#) (galaxy.tools.Tool method), 175
[parse_tags\(\)](#) (galaxy.tags.tag_handler.TagHandler method), 169
[parse_tests_elem\(\)](#) (in module galaxy.tools.test), 178
[parse_version\(\)](#) (in module pkg_resources), 362
[parse_xml\(\)](#) (in module galaxy.util), 214
[parse_xml_string\(\)](#) (in module galaxy.util), 214
[parsed_version](#) (pkg_resources.Distribution attribute), 361
[parseData\(\)](#) (galaxy.visualization.data_providers.phyloviz.newickparser.NewickParser method), 228
[parseFile\(\)](#) (galaxy.visualization.data_providers.phyloviz.baseparser.BaseParser method), 227
[parseFile\(\)](#) (galaxy.visualization.data_providers.phyloviz.newickparser.NewickParser method), 228
[parseFile\(\)](#) (galaxy.visualization.data_providers.phyloviz.nexusparser.NexusParser method), 228
[parseFile\(\)](#) (galaxy.visualization.data_providers.phyloviz.phyloxmlparser.PhyloxmlParser method), 228
[parseNexus\(\)](#) (galaxy.visualization.data_providers.phyloviz.nexusparser.NexusParser method), 228
[parseNode\(\)](#) (galaxy.visualization.data_providers.phyloviz.newickparser.NewickParser method), 228
[parseNode\(\)](#) (galaxy.visualization.data_providers.phyloviz.phyloxmlparser.PhyloxmlParser method), 228
[pass_through_operations](#) (galaxy.web.framework.helpers.grids.Grid attribute), 248
[PasswordField](#) (class in galaxy.web.form_builder), 231
[path](#) (galaxy.eggs.dist.DistScrambleEgg attribute), 96
[path](#) (galaxy.eggs.Egg attribute), 95
[path](#) (galaxy.web.framework.base.Request attribute), 243
[PathMetadata](#) (class in pkg_resources), 364
[pause\(\)](#) (galaxy.jobs.JobWrapper method), 109
[PauseModule](#) (class in galaxy.workflow.modules), 346
[Pbe](#) (class in galaxy.datatypes.genetics), 53
[Pbm](#) (class in galaxy.datatypes.images), 58
[Pbm](#) (class in galaxy.datatypes.images), 58
[Pcx](#) (class in galaxy.datatypes.images), 58
[Pdf](#) (class in galaxy.datatypes.images), 58
[PerMonthAll](#) (galaxy.webapps.galaxy.controllers.root.RootController method), 326
[per_month_all\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 339
[per_month_all\(\)](#) (galaxy.webapps.reports.controllers.sample_tracking.SampleTracking method), 341
[per_month_all\(\)](#) (galaxy.webapps.reports.controllers.workflows.Workflows method), 344
[per_month_in_error\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 339
[per_tool\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 339
[per_user\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 339
[per_user\(\)](#) (galaxy.webapps.reports.controllers.sample_tracking.SampleTracking method), 341
[per_user\(\)](#) (galaxy.webapps.reports.controllers.workflows.Workflows method), 344
[perform_action\(\)](#) (galaxy.external_services.actions.ExternalServiceAction method), 100
[perform_action\(\)](#) (galaxy.external_services.actions.ExternalServiceTemplate method), 100

perform_action() (galaxy.external_services.actions.ExternalServiceAction method), 101
 perform_action() (galaxy.external_services.actions.PopulatedExternalServiceAction method), 101
 perform_action_by_name() (galaxy.external_services.service.PopulatedExternalService method), 103
 permitted_actions (galaxy.model.Dataset attribute), 123
 permitted_actions (galaxy.model.DatasetInstance attribute), 126
 permitted_actions (galaxy.model.Library attribute), 134
 permitted_actions (galaxy.model.RequestType attribute), 138
 permitted_actions (galaxy.security.RBACAgent attribute), 168
 persist_session() (galaxy.web.framework.openid_manager.OpenIDManager method), 244
 persist_uploads() (in module galaxy.tools.actions.upload_common), 181
 Pgm (class in galaxy.datatypes.images), 58
 Phe (class in galaxy.datatypes.genetics), 53
 Pheno (class in galaxy.datatypes.genetics), 53
 PhyloTree (class in galaxy.visualization.data_providers.phyloviz), 227
 phyloviz() (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
 PhylovizDataProvider (class in galaxy.visualization.data_providers.phyloviz), 227
 Phyloxml (class in galaxy.datatypes.xml), 85
 Phyloxml_Parser (class in galaxy.visualization.data_providers.phyloviz), 228
 pick_a_cycle() (galaxy.util.topsort.CycleError method), 224
 Pileup (class in galaxy.datatypes.tabular), 81
 PKG_INFO (pkg_resources.Distribution attribute), 360
 pkg_resources (module), 356
 pluralize() (galaxy.util.inflection.English method), 220
 pluralize() (galaxy.util.inflection.Inflector method), 220
 Png (class in galaxy.datatypes.images), 58
 poll_active_workflow_ids() (galaxy.model.WorkflowInvocation static method), 144
 popitem() (galaxy.util.odict.odict method), 223
 populate() (galaxy.external_services.service.ExternalServiceActionsGroup method), 102
 populate() (galaxy.external_services.service.PopulatedExternalService method), 103
 populate_action() (galaxy.external_services.actions.ExternalServiceAction attribute), 248
 populate_actions() (galaxy.model.ExternalService method), 127
 populate_webapp_plugins() (in module galaxy.webapps.galaxy.buildapp), 259
 populate_external_services() (galaxy.model.Sample method), 139
 populate_module_and_state() (in module galaxy.workflow.modules), 349
 populate_state() (galaxy.tools.Tool method), 175
 populate_tool_shed_info() (galaxy.tools.Tool method), 175
 populate_widgets_from_kwd() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 237
 PopulatedDisplayApplicationLink (class in galaxy.datatypes.display_applications.application), 91
 PopulatedExternalService (class in galaxy.external_services.service), 103
 PopulatedExternalServiceAction (class in galaxy.external_services.actions), 101
 post_authentication() (galaxy.openid.providers.OpenIDProvider method), 160
 PostJobAction (class in galaxy.model), 137
 PostJobAssociation (class in galaxy.model), 137
 postprocess() (pkg_resources.ResourceManager method), 356
 Pphe (class in galaxy.datatypes.genetics), 53
 Ppm (class in galaxy.datatypes.images), 58
 prepare() (galaxy.datatypes.display_applications.parameters.DisplayApplicationParameters method), 91
 prepare() (galaxy.datatypes.display_applications.parameters.DisplayApplicationParameters method), 92
 prepare() (galaxy.jobs.JobWrapper method), 109
 prepare() (galaxy.jobs.TaskWrapper method), 111
 prepare_actions() (galaxy.external_services.service.ExternalServiceActionsGroup method), 102
 prepare_activation_link() (galaxy.webapps.galaxy.controllers.user.User method), 328
 prepare_display() (galaxy.datatypes.display_applications.application.PopulatedDisplayApplicationParameters method), 91
 prepare_for_install() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 302
 prepare_job() (galaxy.jobs.runners.BaseJobRunner method), 118
 preparing (galaxy.model.JobExportHistoryArchive attribute), 133
 preparing_display() (galaxy.datatypes.display_applications.application.PopulatedDisplayApplicationParameters method), 91
 preserve_state (galaxy.web.framework.helpers.grids.Grid attribute), 248
 preserve_state (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 296
 preserve_state (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 298

preserve_state (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 299
 preserve_state (galaxy.webapps.galaxy.controllers.admin.TopVersionListGrid attribute), 299
 preserve_state (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 301
 preserve_state (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAsynchronousListGrid attribute), 306
 preserve_state (galaxy.webapps.galaxy.controllers.external_process_split_service_grid attribute), 307
 preserve_state (galaxy.webapps.galaxy.controllers.forms.FormGrid attribute), 308
 preserve_state (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 311
 preserve_state (galaxy.webapps.galaxy.controllers.library.LibraryListGrid attribute), 313
 preserve_state (galaxy.webapps.galaxy.controllers.library_admin.LibraryAdminListGrid attribute), 314
 preserve_state (galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid attribute), 321
 preserve_state (galaxy.webapps.galaxy.controllers.requests_admin.DataAttributesGrid attribute), 322
 preserve_state (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute), 341
 preserve_state (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid attribute), 342
 preserve_state (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid attribute), 344
 pretty_print_json() (in module galaxy.util), 214
 pretty_print_xml() (in module galaxy.util), 215
 pretty_stack() (in module galaxy.model.orm.logging_connection_proxy), 147
 primary_field (galaxy.web.form_builder.SwitchingSelectField attribute), 233
 primary_file_name (galaxy.datatypes.data.Data attribute), 49
 print_nonsleeping() (galaxy.util.heartbeat.Heartbeat method), 218
 private_id (galaxy.model.Role attribute), 138
 privately_share_dataset() (galaxy.security.GalaxyRBACAgent method), 166
 proceed_login() (galaxy.webapps.galaxy.controllers.user.User method), 328
 process() (galaxy.datatypes.metadata.Statement class method), 70
 process_bind_param() (galaxy.model.custom_types.JSONType method), 145
 process_bind_param() (galaxy.model.custom_types.TrimmedString method), 145
 process_bind_param() (galaxy.model.custom_types.UUIDType method), 146
 provide_data() (galaxy.visualization.data_providers.basic.BaseDataProvider method), 226
 provide_list_grid_value() (galaxy.model.custom_types.JSONType method), 145
 provide_result_value() (galaxy.model.custom_types.MetadataType method), 145
 provide_split_service_grid() (galaxy.datatypes.sequence.Fastq static method), 74
 ProfileMiddleware (class in galaxy.web.framework.middleware.profile), 67
 protocol (galaxy.web.framework.base.Request attribute), 44
 provenance_item_class (galaxy.webapps.galaxy.api.provenance.HDAProvenanceItem class), 287
 provenance_item_class (galaxy.webapps.galaxy.api.provenance.LDDAProvenanceItem class), 287
 provenance_item_id (galaxy.webapps.galaxy.api.provenance.HDAProvenanceItem attribute), 287
 provenance_item_id (galaxy.webapps.galaxy.api.provenance.LDDAProvenanceItem attribute), 287
 ProvidesAppContext (class in galaxy.managers.context), 151
 ProvidesHistoryContext (class in galaxy.managers.context), 150
 ProvidesUserContext (class in galaxy.managers.context), 151
 pruneLD() (in module galaxy.datatypes.converters.pbед_ldreduced_converter), 89
 Psd (class in galaxy.datatypes.images), 59
 stats_as_html() (in module galaxy.web.framework.middleware.profile), 252
 psyc_full (module), 366
 public_actions (galaxy.jobs.actions.post.ActionBox attribute), 114
 PublicURLColumn (class in galaxy.web.framework.helpers.grids), 250
 published_list_grid (galaxy.webapps.galaxy.controllers.history.HistoryController attribute), 310
 published_list_grid (galaxy.webapps.galaxy.controllers.workflow.WorkflowController attribute), 337
 purge() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
 purge_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 304
 purge_deleted_datasets() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
 purge_group() (galaxy.web.base.controllers.admin.Admin method), 241

[purge_library\(\)](#) (galaxy.webapps.galaxy.controllers.library_admin.LibraryAdminController method), 313
[purge_operation](#) (galaxy.web.base.controllers.admin.AdminQualityScore454 (class in galaxy.datatypes.qualityscore), attribute), 241
[purge_operation](#) (galaxy.webapps.galaxy.controllers.admin.AdminQualityScoreIllumina (class in galaxy.datatypes.qualityscore), attribute), 295
[purge_quota\(\)](#) (galaxy.webapps.galaxy.controllers.admin.AdminQualityScoreSolexa (class in galaxy.datatypes.qualityscore), method), 295
[purge_repository\(\)](#) (galaxy.webapps.galaxy.controllers.admin.AdminQualityScoreSolexa (class in galaxy.datatypes.qualityscore), method), 302
[purge_role\(\)](#) (galaxy.web.base.controllers.admin.AdminQualityScoreSolexa (class in galaxy.datatypes.qualityscore), method), 241
[purge_user\(\)](#) (galaxy.web.base.controllers.admin.AdminQualityScoreSolexa (class in galaxy.datatypes.qualityscore), method), 241
[put\(\)](#) (galaxy.jobs.handler.DefaultJobDispatcher method), 111
[put\(\)](#) (galaxy.jobs.handler.JobHandlerQueue method), 112
[put\(\)](#) (galaxy.jobs.handler.JobHandlerStopQueue method), 112
[put\(\)](#) (galaxy.jobs.NoopQueue method), 110
[put\(\)](#) (galaxy.jobs.runners.BaseJobRunner method), 118
[put\(\)](#) (galaxy.webapps.demo_sequencer.controllers.common.CommonJobController method), 257
[put_stop\(\)](#) (galaxy.jobs.NoopQueue method), 110
[PythonObjectParameter](#) (class in galaxy.datatypes.metadata), 69

Q

[qp](#) (galaxy.datatypes.display_applications.parameters.DisplayDataValueWrapper attribute), 92
[qp](#) (galaxy.datatypes.display_applications.parameters.DisplayParameterValuesWrapper attribute), 92
[quality\(\)](#) (in module mimeparse), 356
[quality_max](#) (galaxy_utils.sequence.fastq.fastqCSSangerRead attribute), 351
[quality_max](#) (galaxy_utils.sequence.fastq.fastqIlluminaRead attribute), 351
[quality_max](#) (galaxy_utils.sequence.fastq.fastqSangerRead attribute), 352
[quality_max](#) (galaxy_utils.sequence.fastq.fastqSequencingRead attribute), 353
[quality_max](#) (galaxy_utils.sequence.fastq.fastqSolexaRead attribute), 353
[quality_min](#) (galaxy_utils.sequence.fastq.fastqCSSangerRead attribute), 351
[quality_min](#) (galaxy_utils.sequence.fastq.fastqIlluminaRead attribute), 351
[quality_min](#) (galaxy_utils.sequence.fastq.fastqSangerRead attribute), 352
[quality_min](#) (galaxy_utils.sequence.fastq.fastqSequencingRead attribute), 353
[quality_min](#) (galaxy_utils.sequence.fastq.fastqSolexaRead attribute), 353

[QualityScore454](#) (class in galaxy.datatypes.qualityscore), 71
[QualityScoreIllumina](#) (class in galaxy.datatypes.qualityscore), 71
[QualityScoreSolexa](#) (class in galaxy.datatypes.qualityscore), 71
[QualityScoreSolexaDoolshed](#) (class in galaxy.datatypes.qualityscore), 71
[queue_history_export\(\)](#) (galaxy.web.base.controller.ExportsHistoryMixin method), 235
[queue_history_import\(\)](#) (galaxy.web.base.controller.ImportsHistoryMixin method), 235
[queue_job\(\)](#) (galaxy.jobs.runners.BaseJobRunner method), 118
[queue_job\(\)](#) (galaxy.jobs.runners.cli.ShellJobRunner method), 119
[queue_job\(\)](#) (galaxy.jobs.runners.condor.CondorJobRunner method), 119
[queue_job\(\)](#) (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 120
[queue_job\(\)](#) (galaxy.jobs.runners.local.LocalJobRunner method), 120
[queue_job\(\)](#) (galaxy.jobs.runners.lwr.LwrJobRunner method), 121
[queue_job\(\)](#) (galaxy.jobs.runners.tasks.TaskedJobRunner method), 121
[queued_dataset_files](#) (galaxy.model.Sample attribute), 139
[Quota](#) (class in galaxy.model), 137
[quota_amount\(\)](#) (galaxy.model.HistoryDatasetAssociation method), 129
[quota_list_grid](#) (galaxy.web.base.controllers.admin.Admin attribute), 241
[quota_list_grid](#) (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy attribute), 295
[QuotaAgent](#) (class in galaxy.quota), 161
[QuotaApiController](#) (class in galaxy.webapps.galaxy.api.quotas), 32, 287
[QuotaListGrid](#) (class in galaxy.webapps.galaxy.controllers.admin), 297
[QuotaListGrid.AmountColumn](#) (class in galaxy.webapps.galaxy.controllers.admin), 297
[QuotaListGrid.DescriptionColumn](#) (class in galaxy.webapps.galaxy.controllers.admin), 297
[QuotaListGrid.GroupsColumn](#) (class in galaxy.webapps.galaxy.controllers.admin), 297
[QuotaListGrid.NameColumn](#) (class in galaxy.webapps.galaxy.controllers.admin), 297

[297](#)
 QuotaListGroup.StatusColumn (class in [galaxy.webapps.galaxy.controllers.admin](#)), [297](#)
 QuotaListGroup.UsersColumn (class in [galaxy.webapps.galaxy.controllers.admin](#)), [297](#)
 QuotaParamParser (class in [galaxy.web.params](#)), [234](#)
 quotas() ([galaxy.webapps.galaxy.controllers.admin.AdminController](#) method), [295](#)
R
 RangeParameter (class in [galaxy.datatypes.metadata](#)), [69](#)
 Rast (class in [galaxy.datatypes.images](#)), [59](#)
 rate_async() ([galaxy.webapps.galaxy.controllers.dataset.DatasetController](#) method), [305](#)
 rate_async() ([galaxy.webapps.galaxy.controllers.history.HistoryController](#) method), [310](#)
 rate_async() ([galaxy.webapps.galaxy.controllers.page.PageController](#) method), [318](#)
 rate_async() ([galaxy.webapps.galaxy.controllers.visualization.VisualizationController](#) method), [334](#)
 rate_async() ([galaxy.webapps.galaxy.controllers.workflow.WorkflowController](#) method), [337](#)
 rate_item() ([galaxy.model.item_attrs.UsesItemRatings](#) method), [146](#)
 rating_mapping() (in module [galaxy.model.mapping](#)), [147](#)
 raw_param_dict() ([galaxy.model.Job](#) method), [133](#)
 RBACAgent (class in [galaxy.security](#)), [167](#)
 read_build_sites() (in module [galaxy.util](#)), [215](#)
 read_dbnames() (in module [galaxy.util](#)), [215](#)
 read_lines() ([galaxy.web.framework.base.FieldStorage](#) method), [242](#)
 read_unordered_gtf() (in module [galaxy.datatypes.util.gff_util](#)), [94](#)
 Reader (class in [galaxy_utils.sequence.vcf](#)), [355](#)
 readline() ([galaxy.tools.exception_handling.UCSCOutWrapper](#) method), [178](#)
 readline() ([galaxy_utils.sequence.fastq.ReadlineCountFile](#) method), [350](#)
 ReadlineCountFile (class in [galaxy_utils.sequence.fastq](#)), [350](#)
 ready ([galaxy.model.JobExportHistoryArchive](#) attribute), [133](#)
 ready() ([galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter](#) method), [91](#)
 ready() ([galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter](#) method), [92](#)
 ready_name_for_url() (in module [galaxy.util](#)), [215](#)
 reclaim_ownership() ([galaxy.jobs.JobWrapper](#) method), [110](#)
 recover() ([galaxy.jobs.handler.DefaultJobDispatcher](#) method), [111](#)
 recover() ([galaxy.jobs.runners.BaseJobRunner](#) method), [118](#)
 recover() ([galaxy.jobs.runners.cli.ShellJobRunner](#) method), [119](#)
 recover() ([galaxy.jobs.runners.condor.CondorJobRunner](#) method), [120](#)
 recover() ([galaxy.jobs.runners.drmaa.DRMAAJobRunner](#) method), [120](#)
 recover() ([galaxy.jobs.runners.local.LocalJobRunner](#) method), [120](#)
 recover() ([galaxy.jobs.runners.lwr.LwrJobRunner](#) method), [121](#)
 recover() ([galaxy.jobs.runners.tasks.TaskedJobRunner](#) method), [121](#)
 recover_mapping() ([galaxy.workflow.modules.InputModule](#) method), [345](#)
 recover_mapping() ([galaxy.workflow.modules.PauseModule](#) method), [346](#)
 recover_mapping() ([galaxy.workflow.modules.ToolModule](#) method), [347](#)
 recover_mapping() ([galaxy.workflow.modules.WorkflowModule](#) method), [349](#)
 recover_runtime_state() ([galaxy.workflow.modules.SimpleWorkflowModule](#) method), [346](#)
 recover_runtime_state() ([galaxy.workflow.modules.ToolModule](#) method), [347](#)
 recover_state() ([galaxy.workflow.modules.SimpleWorkflowModule](#) method), [346](#)
 recover_state() ([galaxy.workflow.modules.ToolModule](#) method), [347](#)
 recursively_stringify_dictionary_keys() (in module [galaxy.util](#)), [215](#)
 RecursiveNone (class in [galaxy.util.none_like](#)), [222](#)
 redirect() ([galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner](#) method), [327](#)
 reduce_block_by_primary_genome() (in module [galaxy.tools.util.maf_utilities](#)), [212](#)
 reference() ([galaxy.visualization.genomes.Genomes](#) method), [225](#)
 regenerate_primary_file() ([galaxy.datatypes.assembly.Velvet](#) method), [42](#)
 regenerate_primary_file() ([galaxy.datatypes.genetics.RexpBase](#) method), [54](#)
 regenerate_primary_file() ([galaxy.datatypes.genetics.Rgenetics](#) method), [54](#)
 regenerate_primary_file() ([galaxy.datatypes.ngsindex.BowtieIndex](#) method), [70](#)
 regex_line_dataprovider() ([galaxy.datatypes.binary.Bam](#) method), [43](#)
 regex_line_dataprovider() ([galaxy.datatypes.data.Text](#) method), [50](#)

[regex_line_dataprovider\(\)](#) (galaxy.datatypes.tabular.Sam method), 82
[RegexValidator](#) (class in galaxy.tools.parameters.validation), 208
[RegionAlignment](#) (class in galaxy.tools.util.maf_utilities), 210
[register_cleanup_file_attribute\(\)](#) (galaxy.jobs.runners.AsynchronousJobState method), 118
[register_finder\(\)](#) (in module pkg_resources), 366
[register_loader_type\(\)](#) (in module pkg_resources), 366
[register_namespace_handler\(\)](#) (in module pkg_resources), 366
[register_sniffable_binary_format\(\)](#) (galaxy.datatypes.binary.Binary static method), 44
[register_unsniffable_binary_ext\(\)](#) (galaxy.datatypes.binary.Binary static method), 44
[registered_users\(\)](#) (galaxy.webapps.reports.controllers.users.Users method), 343
[registered_users_per_month\(\)](#) (galaxy.webapps.reports.controllers.users.Users method), 343
[Registry](#) (class in galaxy.datatypes.registry), 71
[Registry](#) (class in galaxy.webapps.demo_sequencer.registry), 257
[reinstall_repository\(\)](#) (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 302
[reject_request\(\)](#) (galaxy.webapps.galaxy.controllers.requests_admin.RequestAdmin method), 323
[relativize_symlinks\(\)](#) (in module galaxy.util), 215
[reload\(\)](#) (galaxy.sample_tracking.external_service_types.ExternalService method), 162
[reload_external_service_types\(\)](#) (galaxy.webapps.galaxy.controllers.external_service.ExternalService method), 306
[reload_from_files\(\)](#) (galaxy.tools.data.ToolDataTable method), 182
[reload_tables\(\)](#) (galaxy.tools.data.ToolDataTableManager method), 183
[reload_tool\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 241
[reload_tool_by_id\(\)](#) (galaxy.tools.ToolBox method), 176
[remote_host](#) (galaxy.web.framework.base.Request attribute), 243
[remote_hostname](#) (galaxy.web.framework.base.Request attribute), 243
[remote_port](#) (galaxy.web.framework.base.Request attribute), 243
[RemoteUser](#) (class in galaxy.web.framework.middleware.remoteuser), 341
[remove\(\)](#) (pkg_resources.Environment method), 358
[remove_display_app\(\)](#) (galaxy.datatypes.data.Data method), 49
[remove_doppelgangers\(\)](#) (galaxy.eggs.Egg method), 95
[remove_file_or_path\(\)](#) (in module galaxy.eggs), 95
[remove_item_tag\(\)](#) (galaxy.tags.tag_handler.TagHandler method), 169
[remove_tag_async\(\)](#) (galaxy.webapps.galaxy.controllers.tag.TagsController method), 326
[remove_temp_index_file\(\)](#) (in module galaxy.tools.util.maf_utilities), 212
[remove_tool_by_id\(\)](#) (galaxy.tools.ToolBox method), 176
[RemoveValueFilter](#) (class in galaxy.tools.parameters.dynamic_options), 198
[rename\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[rename\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
[rename_async\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
[rename_dataset_options](#) (galaxy.model.RequestType attribute), 138
[rename_datasets\(\)](#) (galaxy.webapps.galaxy.controllers.requests_admin.RequestAdmin method), 323
[rename_group\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 241
[rename_quota\(\)](#) (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 241
[rename_role\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 241
[RenameDatasetAction](#) (class in galaxy.jobs.actions.post), 115
[render\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 241
[render\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[repair_methods\(\)](#) (galaxy.datatypes.data.Data method), 49
[repair_methods\(\)](#) (galaxy.datatypes.interval.Interval method), 65
[repair_methods\(\)](#) (galaxy.datatypes.tabular.Pileup method), 81
[repair_repository\(\)](#) (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303
[repair_repository_revision\(\)](#) (galaxy.webapps.galaxy.api.tool_shed_repositories.ToolShedRepositories method), 36, 291
[repair_tool_shed_repositories\(\)](#) (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303
[Repeat](#) (class in galaxy.tools.parameters.grouping), 200
[Report](#) (class in galaxy.webapps.reports.controllers.root), 253
[report\(\)](#) (galaxy.util.debugging.SimpleProfiler method), 217

[report_error\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetGrid.StateColumn (class in galaxy.webapps.galaxy.controllers.requests_common), method), 305
[REPORT_FORMAT](#) (galaxy.util.debugging.SimpleProfiler attribute), 217
[ReportsWebApplication](#) (class in galaxy.webapps.galaxy.controllers.requests_common), galaxy.webapps.reports.buildapp), 338
[repository_installation_grid](#) (galaxy.webapps.galaxy.controllers.admin_toolshed.RequestTypePolished (class in galaxy.webapps.galaxy.controllers.request_type), attribute), 303
[repository_installation_status_updates\(\)](#) (galaxy.webapps.galaxy.controllers.admin_toolshed.RequestTypePolished (class in galaxy.webapps.galaxy.controllers.request_type), method), 303
[repository_list_grid](#) (galaxy.web.base.controllers.admin.Admin attribute), 241
[Request](#) (class in galaxy.model), 137
[Request](#) (class in galaxy.web.framework.base), 243
[request_grid](#) (galaxy.webapps.galaxy.controllers.requests.Requests galaxy.sample_tracking.request_types), 162
[request_grid](#) (galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin galaxy.webapps.galaxy.controllers.request_type), attribute), 323
[request_type_grid](#) (galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin galaxy.webapps.galaxy.controllers.request_type), attribute), 320
[request_type_permissions\(\)](#) (galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid.ExternalServiceColumn (class in galaxy.webapps.galaxy.controllers.request_type), method), 320
[request_types\(\)](#) (galaxy.managers.context.ProvidesAppContext method), 150
[RequestEvent](#) (class in galaxy.model), 138
[RequestParameterInvalidException](#), 99
[RequestParameterMissingException](#), 99
[Requests](#) (class in galaxy.webapps.galaxy.controllers.requests), galaxy.webapps.galaxy.controllers.request_type), 321
[RequestsAdmin](#) (class in galaxy.webapps.galaxy.controllers.requests_admin), galaxy.webapps.galaxy.controllers.request_type), 322
[RequestsApiController](#) (class in galaxy.webapps.galaxy.api.requests), 32, 288
[RequestsCommon](#) (class in galaxy.webapps.galaxy.controllers.requests_common), galaxy.webapps.galaxy.controllers.requests_common), 323
[RequestsGrid](#) (class in galaxy.webapps.galaxy.controllers.requests_common), galaxy.webapps.galaxy.controllers.requests_common), 324
[RequestsGrid.DescriptionColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_common), galaxy.webapps.galaxy.controllers.requests_common), 324
[RequestsGrid.NameColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_common), galaxy.webapps.galaxy.controllers.requests_common), 324
[RequestsGrid.SamplesColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_common), galaxy.webapps.galaxy.controllers.requests_common), 324
[RequestsGrid.StateColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_common), galaxy.webapps.galaxy.controllers.requests_common), 324
[RequestsGrid.TypeColumn](#) (class in galaxy.webapps.galaxy.controllers.requests_common), galaxy.webapps.galaxy.controllers.requests_common), 324
[RequestType](#) (class in galaxy.model), 138
[RequestTypePolished](#) (class in galaxy.webapps.galaxy.controllers.request_type), 320
[RequestTypeApiController](#) (class in galaxy.webapps.galaxy.api.request_types), 32, 287
[RequestTypeExternalServiceAssociation](#) (class in galaxy.model), 138
[RequestTypeFactory](#) (class in galaxy.sample_tracking.request_types), 162
[RequestTypeGrid](#) (class in galaxy.webapps.galaxy.controllers.request_type), 320
[RequestTypeGrid.DescriptionColumn](#) (class in galaxy.webapps.galaxy.controllers.request_type), 320
[RequestTypeGrid.ExternalServiceColumn](#) (class in galaxy.webapps.galaxy.controllers.request_type), 320
[RequestTypeGrid.NameColumn](#) (class in galaxy.webapps.galaxy.controllers.request_type), 320
[RequestTypeGrid.RequestFormColumn](#) (class in galaxy.webapps.galaxy.controllers.request_type), 320
[RequestTypeGrid.SampleFormColumn](#) (class in galaxy.webapps.galaxy.controllers.request_type), 320
[RequestTypePermissions](#) (class in galaxy.model), 138
[RequestTypeRunAssociation](#) (class in galaxy.model), 138
[require\(\)](#) (galaxy.eggs.Egg method), 95
[require\(\)](#) (in module galaxy.eggs), 96
[require\(\)](#) (in module pkg_resources), 357
[require\(\)](#) (pkg_resources.EntryPoint method), 361
[require\(\)](#) (pkg_resources.WorkingSet method), 359
[require_file\(\)](#) (in module galaxy.tools.test), 179
[required_header_fields](#) (galaxy_utils.sequence.vcf.VariantCall attribute), 355
[required_header_fields](#) (galaxy_utils.sequence.vcf.VariantCall33 attribute), 355
[required_header_length](#) (galaxy_utils.sequence.vcf.VariantCall attribute), 355
[required_header_length](#) (galaxy_utils.sequence.vcf.VariantCall33 attribute), 355
[Requirement](#) (class in pkg_resources), 361
[requires\(\)](#) (pkg_resources.Distribution method), 361

- requires_setting_metadata (galaxy.jobs.JobWrapper attribute), 110
- requires_setting_metadata (galaxy.tools.SetMetadataTool attribute), 171
- requires_setting_metadata (galaxy.tools.Tool attribute), 175
- requires_user_input (galaxy.external_services.parameters.ExternalServiceParameter attribute), 101
- rerun() (galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner method), 327
- reselect_tool_panel_section() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303
- resend_verification() (galaxy.webapps.galaxy.controllers.user.User method), 329
- resend_verification_email() (galaxy.webapps.galaxy.controllers.user.User method), 329
- reset_metadata_on_installed_repositories() (galaxy.webapps.galaxy.api.tool_shed_repositories.ToolShedRepositoriesController method), 36, 291
- reset_metadata_on_selected_installed_repositories() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303
- reset_password() (galaxy.webapps.galaxy.controllers.user.User method), 329
- reset_repository_metadata() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303
- reset_to_install() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303
- reset_user_password() (galaxy.web.base.controllers.admin.Admin method), 241
- ResolutionError, 361
- resolve() (galaxy.eggs.Crate method), 95
- resolve() (galaxy.eggs.Egg method), 95
- resolve() (pkg_resources.WorkingSet method), 359
- resolve_path() (galaxy.config.Configuration method), 40
- resolve_path() (in module galaxy.config), 41
- resolve_path() (in module galaxy.webapps.demo_sequencer.config), 257
- resolve_path() (in module galaxy.webapps.reports.config), 339
- resource_exists() (pkg_resources.ResourceManager method), 360
- resource_filename() (pkg_resources.ResourceManager method), 360
- resource_isdir() (pkg_resources.IResourceProvider method), 364
- resource_isdir() (pkg_resources.NullProvider method), 365
- resource_isdir() (pkg_resources.ResourceManager method), 360
- resource_listdir() (pkg_resources.IResourceProvider method), 364
- resource_listdir() (pkg_resources.NullProvider method), 365
- resource_listdir() (pkg_resources.ResourceManager method), 360
- Resource_string() (pkg_resources.ResourceManager method), 360
- ResourceManager (class in pkg_resources), 359
- ResourceToolshed (class in galaxy.web.framework.base), 243
- restore_param() (galaxy.tools.parameters.sanitize.ToolParameterSanitizer method), 206
- restore_text() (galaxy.tools.parameters.sanitize.ToolParameterSanitizer method), 206
- restore_text() (in module galaxy.util), 215
- restrict_scores_to_valid_range() (galaxy_utils.sequence.fastq.fastqSequencingReadController method), 361
- resume_paused_jobs() (galaxy.model.History method), 129
- resume_paused_jobs() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
- retag_async() (galaxy.webapps.galaxy.controllers.tag.TagsController method), 327
- reverse() (galaxy.util.odict.odict method), 223
- reverse() (galaxy_utils.sequence.fastq.fastqCSSangerRead method), 351
- reverse() (galaxy_utils.sequence.fastq.fastqSequencingRead method), 353
- reverse() (galaxy_utils.sequence.sequence.SequencingRead method), 354
- reverse() (in module galaxy_utils.sequence.transform), 354
- reverse_complement() (galaxy_utils.sequence.sequence.SequencingRead method), 354
- ReverseSortColumn (class in galaxy.web.framework.helpers.grids), 250
- review_tool_migration_stages() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 295
- RexpBase (class in galaxy.datatypes.genetics), 53
- Rgb (class in galaxy.datatypes.images), 59
- rgConv() (in module galaxy.datatypes.converters.lped_to_fped_converter), 88
- rgConv() (in module galaxy.datatypes.converters.lped_to_pbed_converter), 89
- rgConv() (in module galaxy.datatypes.converters.pbed_to_lped_converter), 89
- Rgenetics (class in galaxy.datatypes.genetics), 54
- rgFeatureList (class in galaxy.datatypes.genetics), 55
- rgSampleList (class in galaxy.datatypes.genetics), 55
- rgTabList (class in galaxy.datatypes.genetics), 55

RNA_complement() (in module galaxy_utils.sequence.transform), 354
 RNA_reverse_complement() (in module galaxy_utils.sequence.transform), 354
 Roadmaps (class in galaxy.datatypes.assembly), 41
 Role (class in galaxy.model), 138
 role_list_grid (galaxy.web.base.controllers.admin.Admin attribute), 241
 role_list_grid (galaxy.webapps.galaxy.controllers.admin.Admin attribute), 295
 RoleApiController (class in galaxy.webapps.galaxy.api.roles), 33, 288
 RoleListGrid (class in galaxy.webapps.galaxy.controllers.admin), 298
 RoleListGrid.DescriptionColumn (class in galaxy.webapps.galaxy.controllers.admin), 298
 RoleListGrid.GroupsColumn (class in galaxy.webapps.galaxy.controllers.admin), 298
 RoleListGrid.NameColumn (class in galaxy.webapps.galaxy.controllers.admin), 298
 RoleListGrid.StatusColumn (class in galaxy.webapps.galaxy.controllers.admin), 298
 RoleListGrid.TypeColumn (class in galaxy.webapps.galaxy.controllers.admin), 298
 RoleListGrid.UsersColumn (class in galaxy.webapps.galaxy.controllers.admin), 298
 RoleManager (class in galaxy.managers.roles), 156
 roles() (galaxy.web.base.controllers.admin.Admin method), 241
 Root (in module galaxy.web.base.controller), 235
 RootController (class in galaxy.webapps.galaxy.controllers.root), 325
 RootParamContext (class in galaxy.tools.test), 178
 roundify() (in module galaxy.util), 215
 rst_to_html() (in module galaxy.util), 215
 Run (class in galaxy.model), 138
 run() (galaxy.jobs.transfer_manager.TransferManager method), 113
 run() (galaxy.util.heartbeat.Heartbeat method), 218
 run() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 337
 run_details (galaxy.model.RequestType attribute), 138
 run_details (galaxy.model.Sample attribute), 139
 run_job() (galaxy.jobs.deferred.data_transfer.DataTransfer method), 116
 run_next() (galaxy.jobs.runners.BaseJobRunner method), 118
 run_scramble_script() (galaxy.eggs.dist.DistScrambleEgg method), 96
 run_scramble_script() (galaxy.eggs.scramble.ScrambleEgg method), 97
 run_script() (pkg_resources.IMetadataProvider method), 363
 run_script() (pkg_resources.NullProvider method), 365
 run_script() (pkg_resources.WorkingSet method), 359
 runner_name (galaxy.jobs.runners.cli.ShellJobRunner attribute), 119
 runner_name (galaxy.jobs.runners.condor.CondorJobRunner attribute), 120
 runner_name (galaxy.jobs.runners.drmaa.DRMAAJobRunner attribute), 120
 runner_name (galaxy.jobs.runners.local.LocalJobRunner attribute), 120
 runner_name (galaxy.jobs.runners.lwr.LwrJobRunner attribute), 121
 runner_name (galaxy.jobs.runners.tasks.TaskedJobRunner attribute), 121
 runner_states (galaxy.jobs.runners.JobState attribute), 119
 RunnerParams (class in galaxy.jobs.runners), 119
 running (galaxy.jobs.runners.AsynchronousJobState attribute), 118
 RuntimeException, 146
 RuntimeValue (class in galaxy.tools.parameters.basic), 193

S

sa_session (galaxy.managers.context.ProvidesAppContext attribute), 150
 sa_session (galaxy.security.GalaxyRBACAgent attribute), 166
 sa_session (galaxy.security.HostAgent attribute), 167
 sa_session (galaxy.tools.Tool attribute), 175
 sa_session (galaxy.tools.ToolBox attribute), 176
 safe_dumps() (in module galaxy.util.json), 222
 safe_extra() (in module pkg_resources), 363
 safe_name() (in module pkg_resources), 362
 safe_str_cmp() (in module galaxy.util), 215
 safe_version() (in module pkg_resources), 362
 Sam (class in galaxy.datatypes.tabular), 81
 Sample (class in galaxy.model), 139
 sample_datasets_updates() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 323
 sample_state_updates() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 323
 SampleDataset (class in galaxy.model), 139
 SampleEvent (class in galaxy.model), 139
 SampleRunAssociation (class in galaxy.model), 139

[samples_have_common_state](#) ([galaxy.model.Request](#) attribute), 137
[samples_with_bar_code](#) ([galaxy.model.Request](#) attribute), 138
[samples_without_library_destinations](#) ([galaxy.model.Request](#) attribute), 138
[SamplesApiController](#) (class in [galaxy.webapps.galaxy.api.samples](#)), 33, 288
[SampleState](#) (class in [galaxy.model](#)), 139
[SampleStateFactory](#) (class in [galaxy.sample_tracking.sample](#)), 163
[SampleTracking](#) (class in [galaxy.webapps.reports.controllers.sample_tracking](#)), 341
[samtools_dataprovider\(\)](#) ([galaxy.datatypes.binary.Bam](#) method), 43
[sanitize_for_filename\(\)](#) (in module [galaxy.util](#)), 215
[sanitize_html\(\)](#) (in module [galaxy.util.sanitize_html](#)), 223
[sanitize_lists_to_string\(\)](#) (in module [galaxy.util](#)), 215
[sanitize_param\(\)](#) ([galaxy.tools.parameters.sanitize.ToolParameterSanitizer](#) method), 206
[sanitize_param\(\)](#) (in module [galaxy.util](#)), 215
[sanitize_text\(\)](#) ([galaxy.tools.parameters.sanitize.ToolParameterSanitizer](#) method), 206
[sanitize_text\(\)](#) (in module [galaxy.util](#)), 216
[save\(\)](#) ([galaxy.webapps.galaxy.controllers.page.PageController](#) method), 318
[save\(\)](#) ([galaxy.webapps.galaxy.controllers.visualization.VisualizationController](#) method), 334
[save_form_definition\(\)](#) ([galaxy.webapps.galaxy.controllers.forms.FormsController](#) method), 308
[save_to_step\(\)](#) ([galaxy.workflow.modules.SimpleWorkflowModule](#) method), 347
[save_to_step\(\)](#) ([galaxy.workflow.modules.ToolModule](#) method), 347
[save_to_step\(\)](#) ([galaxy.workflow.modules.WorkflowModule](#) method), 349
[save_visualization\(\)](#) ([galaxy.web.base.controller.UsesVisualizationsController](#) method), 240
[save_widget_field\(\)](#) ([galaxy.web.base.controller.UsesFormDefinitionController](#) method), 237
[save_workflow\(\)](#) ([galaxy.webapps.galaxy.controllers.workflow.WorkflowController](#) method), 338
[saved\(\)](#) ([galaxy.webapps.galaxy.controllers.visualization.VisualizationController](#) method), 334
[scan\(\)](#) ([pkg_resources.Environment](#) method), 358
[Scf](#) (class in [galaxy.datatypes.binary](#)), 45
[score_system](#) ([galaxy_utils.sequence.fastq.fastqCSSangerRead](#) attribute), 351
[score_system](#) ([galaxy_utils.sequence.fastq.fastqIlluminaRead](#) attribute), 351
[score_system](#) ([galaxy_utils.sequence.fastq.fastqSangerRead](#) attribute), 352
[score_system](#) ([galaxy_utils.sequence.fastq.fastqSequencingRead](#) attribute), 353
[score_system](#) ([galaxy_utils.sequence.fastq.fastqSolexaRead](#) attribute), 353
[ScpDataTransferFactory](#) (class in [galaxy.sample_tracking.data_transfer](#)), 162
[scramble\(\)](#) ([galaxy.eggs.scramble.ScrambleCrate](#) method), 96
[scramble\(\)](#) ([galaxy.eggs.scramble.ScrambleEgg](#) method), 97
[scramble_dir](#) ([galaxy.eggs.scramble.ScrambleEgg](#) attribute), 97
[ScrambleCrate](#) (class in [galaxy.eggs.scramble](#)), 96
[ScrambleEgg](#) (class in [galaxy.eggs.scramble](#)), 96
[ScrambleFailure](#), 97
[script_dir](#) ([galaxy.eggs.scramble.ScrambleEgg](#) attribute), 97
[search\(\)](#) ([galaxy.tools.search.ToolBoxSearch](#) method), 209
[search\(\)](#) ([galaxy.webapps.galaxy.api.jobs.JobController](#) method), 21, 277
[SearchController](#) (class in [galaxy.webapps.galaxy.api.search](#)), 33, 289
[searcher](#) ([galaxy.managers.folders.FolderManager](#) method), 152
[secure\(\)](#) ([galaxy.managers.hdas.HDAManager](#) method), 153
[secure\(\)](#) ([galaxy.managers.histories.HistoryManager](#) method), 154
[secure\(\)](#) ([galaxy.managers.libraries.LibraryManager](#) method), 155
[security_check\(\)](#) ([galaxy.web.base.controller.SharableItemSecurityMixin](#) method), 236
[security_check\(\)](#) (in module [galaxy.managers.base](#)), 148
[SecurityHelper](#) (class in [galaxy.web.security](#)), 254
[select_datasets_to_transfer\(\)](#) ([galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdminController](#) method), 323
[SelectField](#) (class in [galaxy.web.form_builder](#)), 232
[SelectParameter](#) (class in [galaxy.datatypes.metadata](#)), 69
[SelectToolParameter](#) (class in [galaxy.tools.parameters.basic](#)), 193
[SendWorkflowConfirmation\(\)](#) ([galaxy.model.Request](#) method), 138
[serialize\(\)](#) (in module [galaxy.web.framework.base](#)), 244
[send_mail\(\)](#) (in module [galaxy.util](#)), 216
[send_redirect\(\)](#) ([galaxy.web.framework.base.Response](#) method), 243
[send_verification_email\(\)](#) ([galaxy.webapps.galaxy.controllers.user.User](#) method), 329
[sentry_dsn_public](#) ([galaxy.config.Configuration](#) attribute), 40
[sep\(\)](#) ([galaxy.jobs.ComputeEnvironment](#) method), 106

sep() (galaxy.jobs.SimpleComputeEnvironment method), 110
 sep2tabs() (in module galaxy.datatypes.sniff), 80
 Sequence (class in galaxy.datatypes.sequence), 76
 sequence_as_DNA() (galaxy_utils.sequence.sequence.Sequence method), 354
 sequence_as_RNA() (galaxy_utils.sequence.sequence.Sequence method), 354
 sequence_space (galaxy_utils.sequence.fastq.fastqCSSangerRead attribute), 351
 sequence_space (galaxy_utils.sequence.fastq.fastqIlluminaRead attribute), 352
 sequence_space (galaxy_utils.sequence.fastq.fastqSangerRead attribute), 352
 sequence_space (galaxy_utils.sequence.fastq.fastqSequencingRead attribute), 353
 sequence_space (galaxy_utils.sequence.fastq.fastqSolexaRead attribute), 353
 Sequences (class in galaxy.datatypes.assembly), 42
 SequenceSplitLocations (class in galaxy.datatypes.sequence), 77
 SequencingRead (class in galaxy_utils.sequence.sequence), 354
 serve_ready_history_export() (galaxy.web.base.controller.ExportsHistoryMixin method), 235
 session (galaxy.web.framework.base.DefaultWebTransaction attribute), 242
 set() (galaxy.tools.data.ToolDataTableManager method), 183
 set_accessible_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 305
 set_accessible_async() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
 set_accessible_async() (galaxy.webapps.galaxy.controllers.page.PageController method), 318
 set_accessible_async() (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
 set_accessible_async() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 338
 set_all_dataset_permissions() (galaxy.security.GalaxyRBACAgent method), 166
 set_all_dataset_permissions() (galaxy.security.RBACAgent method), 168
 set_all_library_permissions() (galaxy.security.GalaxyRBACAgent method), 166
 set_all_library_permissions() (galaxy.security.RBACAgent method), 168
 set_amount() (galaxy.model.Quota method), 137
 set_as_current() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
 set_attribute_metadata() (galaxy.datatypes.interval.Gff method), 63
 set_basic_defaults() (in module galaxy.jobs.splitters.basic), 122
 set_inhibitor_interval() (galaxy.model.DeferredJob method), 127
 set_inhibitor() (galaxy.web.form_builder.CheckboxField method), 229
 set_command_line() (galaxy.model.Job method), 133
 set_command_line() (galaxy.model.Task method), 141
 set_content_type() (galaxy.web.framework.base.Response method), 243
 set_cookie() (galaxy.webapps.demo_sequencer.framework.DemoWebTransaction method), 258
 set_dataset_permission() (galaxy.security.GalaxyRBACAgent method), 166
 set_dataset_permission() (galaxy.security.RBACAgent method), 168
 set_dataset_permissions() (galaxy.security.HostAgent method), 167
 set_dataset_state() (galaxy.model.DatasetInstance method), 126
 set_datatypes_registry() (in module galaxy.model), 145
 set_dbkey() (galaxy.model.DatasetInstance method), 126
 set_default_permissions() (galaxy.webapps.galaxy.controllers.user.User method), 329
 set_default_quota() (galaxy.quota.QuotaAgent method), 161
 set_default_values() (galaxy.datatypes.registry.Registry method), 73
 set_defaults() (galaxy.jobs.runners.JobState method), 119
 set_dir() (galaxy.eggs.Egg method), 95
 set_dir() (galaxy.eggs.Egg method), 95
 set_dir() (galaxy.model.GalaxySession method), 128
 set_distribution() (galaxy.eggs.Egg method), 95
 set_distribution() (galaxy.eggs.Egg method), 95
 set_entity_quota_associations() (galaxy.quota.QuotaAgent method), 161
 set_entity_role_associations() (galaxy.security.GalaxyRBACAgent method), 166
 set_entity_user_associations() (galaxy.security.GalaxyRBACAgent method), 166
 set_extra_files_path() (galaxy.model.Dataset method), 123
 set_file_name() (galaxy.model.Dataset method), 123

set_file_name() (galaxy.model.DatasetInstance method), 126
 set_final_state() (galaxy.model.Job method), 133
 set_from_dict() (galaxy.model.DatasetCollection method), 123
 set_from_dict() (galaxy.model.DatasetCollectionInstance method), 124
 set_from_dict() (galaxy.model.History method), 129
 set_from_dict() (galaxy.model.HistoryDatasetAssociation method), 129
 set_handler() (galaxy.model.Job method), 133
 set_hda_from_dict() (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 237
 set_history_from_dict() (galaxy.web.base.controller.UsesHistoryDatasetAssociationMixin method), 238
 set_id() (galaxy.model.Task method), 141
 set_imported() (galaxy.model.Job method), 133
 set_info() (galaxy.model.Job method), 133
 set_info() (galaxy.model.LibraryDataset method), 135
 set_info() (galaxy.model.Task method), 141
 set_input_datasets() (galaxy.model.Job method), 133
 set_input_library_datasets() (galaxy.model.Job method), 133
 set_item() (galaxy.model.HistoryDatasetAssociationRatingAssociation method), 130
 set_item() (galaxy.model.HistoryDatasetCollectionRatingAssociation method), 131
 set_item() (galaxy.model.HistoryRatingAssociation method), 131
 set_item() (galaxy.model.ItemRatingAssociation method), 131
 set_item() (galaxy.model.LibraryDatasetCollectionRatingAssociation method), 135
 set_item() (galaxy.model.PageRatingAssociation method), 137
 set_item() (galaxy.model.StoredWorkflowRatingAssociation method), 140
 set_item() (galaxy.model.VisualizationRatingAssociation method), 143
 set_item_extended_metadata_obj() (galaxy.web.base.controller.UsesExtendedMetadataMixin method), 236
 set_job() (galaxy.model.Task method), 141
 set_job_destination() (galaxy.jobs.JobWrapper method), 110
 set_job_runner_external_id() (galaxy.model.Task method), 141
 set_job_runner_external_pid() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 67
 set_last_check() (galaxy.model.DeferredJob method), 127
 set_library_dataset_dataset_association() (galaxy.model.LibraryDataset method), 135
 set_library_item_permission() (galaxy.security.GalaxyRBACAgent method), 166
 set_library_item_permission() (galaxy.security.RBACAgent method), 168
 set_max_optional_metadata_filesize() (galaxy.datatypes.data.Data method), 49
 set_message() (galaxy.webapps.demo_sequencer.framework.DemoWebTracker method), 258
 set_meta() (galaxy.datatypes.assembly.Velvet method), 42
 set_meta() (galaxy.datatypes.binary.Bam method), 43
 set_meta() (galaxy.datatypes.binary.GeminiSQLite method), 44
 set_meta() (galaxy.datatypes.binary.SQLite method), 45
 set_meta() (galaxy.datatypes.data.Data method), 49
 set_meta() (galaxy.datatypes.data.Text method), 50
 set_meta() (galaxy.datatypes.genetics.GenomeGraphs method), 52
 set_meta() (galaxy.datatypes.genetics.RexpBase method), 54
 set_meta() (galaxy.datatypes.genetics.Rgenetics method), 54
 set_meta() (galaxy.datatypes.interval.Bed method), 60
 set_meta() (galaxy.datatypes.interval.BedStrict method), 61
 set_meta() (galaxy.datatypes.interval.CustomTrack method), 61
 set_meta() (galaxy.datatypes.interval.Gff method), 63
 set_meta() (galaxy.datatypes.interval.Gff3 method), 63
 set_meta() (galaxy.datatypes.interval.Interval method), 65
 set_meta() (galaxy.datatypes.interval.Wiggle method), 66
 set_meta() (galaxy.datatypes.qualityscore.QualityScoreSOLiD method), 71
 set_meta() (galaxy.datatypes.sequence.csFasta method), 77
 set_meta() (galaxy.datatypes.sequence.Fastq method), 74
 set_meta() (galaxy.datatypes.sequence.Maf method), 76
 set_meta() (galaxy.datatypes.sequence.MafCustomTrack method), 76
 set_meta() (galaxy.datatypes.sequence.Sequence method), 77
 set_meta() (galaxy.datatypes.tabular.Eland method), 80
 set_meta() (galaxy.datatypes.tabular.Sam method), 82
 set_meta() (galaxy.datatypes.tabular.Tabular method), 83
 set_meta() (galaxy.datatypes.tabular.Vcf method), 84
 set_meta() (galaxy.model.DatasetInstance method), 126
 set_metadata() (galaxy.model.DatasetInstance method), 126
 set_name() (galaxy.model.LibraryDataset method), 135
 set_object_store_id() (galaxy.tools.actions.ObjectStorePopulator method), 179
 set_output_datasets() (galaxy.model.Job method), 133

set_output_library_datasets() (galaxy.model.Job method), 133
 set_param_filename() (galaxy.model.Job method), 133
 set_parameters() (galaxy.model.Job method), 133
 set_parameters() (galaxy.model.Task method), 141
 set_params() (galaxy.model.Job method), 133
 set_parent() (galaxy.datatypes.metadata.MetadataCollection method), 68
 set_password_cleartext() (galaxy.model.User method), 142
 set_peek() (galaxy.datatypes.binary.Ab1 method), 42
 set_peek() (galaxy.datatypes.binary.Bam method), 43
 set_peek() (galaxy.datatypes.binary.BigWig method), 44
 set_peek() (galaxy.datatypes.binary.Binary method), 44
 set_peek() (galaxy.datatypes.binary.GeminiSQLite method), 44
 set_peek() (galaxy.datatypes.binary.H5 method), 45
 set_peek() (galaxy.datatypes.binary.Scf method), 45
 set_peek() (galaxy.datatypes.binary.Sff method), 45
 set_peek() (galaxy.datatypes.binary.SQLite method), 45
 set_peek() (galaxy.datatypes.binary.TwoBit method), 45
 set_peek() (galaxy.datatypes.data.Data method), 49
 set_peek() (galaxy.datatypes.data.Text method), 50
 set_peek() (galaxy.datatypes.genetics.RexpBase method), 54
 set_peek() (galaxy.datatypes.genetics.SNPMatrix method), 55
 set_peek() (galaxy.datatypes.images.Gmaj method), 56
 set_peek() (galaxy.datatypes.images.Html method), 57
 set_peek() (galaxy.datatypes.images.Image method), 57
 set_peek() (galaxy.datatypes.images.Laj method), 57
 set_peek() (galaxy.datatypes.ngsindex.BowtieIndex method), 70
 set_peek() (galaxy.datatypes.sequence.Maf method), 76
 set_peek() (galaxy.datatypes.sequence.Sequence method), 77
 set_peek() (galaxy.datatypes.sequence.SequenceSplitLocations method), 77
 set_peek() (galaxy.datatypes.tabular.Tabular method), 84
 set_peek() (galaxy.datatypes.xml.CisML method), 84
 set_peek() (galaxy.datatypes.xml.GenericXml method), 85
 set_peek() (galaxy.datatypes.xml.MEMEXml method), 85
 set_peek() (galaxy.datatypes.xml.Owl method), 85
 set_peek() (galaxy.datatypes.xml.Phyloxml method), 86
 set_peek() (galaxy.model.DatasetInstance method), 126
 set_permission_roles() (galaxy.managers.libraries.LibraryManager method), 155
 set_permissions() (galaxy.webapps.galaxy.api.folders.FoldersController method), 12, 268
 set_permissions() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 26, 281
 set_permissions_old() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 26, 282
 set_position() (galaxy.tools.util.maf_utilities.RegionAlignment method), 210
 set_post_job_actions() (galaxy.model.Job method), 133
 set_prepare_input_files_cmd() (galaxy.model.Task method), 141
 set_public_username() (galaxy.web.base.controller.SharableMixin method), 236
 set_quota_default() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 295
 set_range() (galaxy.tools.util.maf_utilities.RegionAlignment method), 210
 set_raw_data() (galaxy.datatypes.data.Data method), 49
 set_raw_data() (galaxy.datatypes.data.Text method), 51
 set_raw_data() (galaxy.model.DatasetInstance method), 126
 set_request_type_permissions() (galaxy.security.GalaxyRBACAgent method), 166
 set_runner() (galaxy.jobs.JobWrapper method), 110
 set_runner() (galaxy.jobs.TaskWrapper method), 111
 set_runner_external_id() (galaxy.model.Job method), 133
 set_runner_name() (galaxy.model.Job method), 133
 set_session_id() (galaxy.model.Job method), 133
 set_size() (galaxy.model.Dataset method), 123
 set_size() (galaxy.model.DatasetInstance method), 126
 set_size() (galaxy.web.form_builder.PasswordField method), 232
 set_size() (galaxy.web.form_builder.TextArea method), 233
 set_size() (galaxy.web.form_builder.TextField method), 233
 set_slug_async() (galaxy.web.base.controller.SharableMixin method), 236
 set_state() (galaxy.model.Job method), 133
 set_state() (galaxy.model.Task method), 141
 set_stderr() (galaxy.model.Task method), 141
 set_stdout() (galaxy.model.Task method), 141
 set_tags_from_list() (galaxy.managers.tags.TagsManager method), 156
 set_tags_from_list() (galaxy.web.base.controller.UsesTagsMixin method), 239
 set_task_runner_external_id() (galaxy.model.Task method), 141
 set_task_runner_name() (galaxy.model.Task method), 141
 set_tool_id() (galaxy.model.Job method), 133
 set_tool_version() (galaxy.model.Job method), 133
 set_tool_versions() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshedController method), 303
 set_total_size() (galaxy.model.Dataset method), 123
 set_total_size() (galaxy.model.DatasetInstance method), 126

[set_transaction_factory\(\)](#) (galaxy.web.framework.base.WebApplication method), 244
[set_user_id\(\)](#) (galaxy.model.Job method), 133
[set_user_pref_async\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), 329
[set_working_directory\(\)](#) (galaxy.model.Task method), 141
[setdefault\(\)](#) (galaxy.util.odict.odict method), 223
[SetMetadataAction](#) (class in galaxy.jobs.actions.post), 116
[SetMetadataTool](#) (class in galaxy.tools), 171
[SetMetadataToolAction](#) (class in galaxy.tools.actions.metadata), 180
[SetParamAction](#) (class in galaxy.tools), 171
[setref\(\)](#) (galaxy.util.memdump.Memdump method), 222
[settings\(\)](#) (galaxy.webapps.galaxy.controllers.mobile.Mobileshow method), 316
[setup_external_metadata\(\)](#) (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 67
[setup_external_metadata\(\)](#) (galaxy.jobs.JobWrapper method), 110
[setup_external_metadata\(\)](#) (galaxy.jobs.TaskWrapper method), 111
[setup_job\(\)](#) (galaxy.tools.imp_exp.JobExportHistoryArchiveWrapper method), 184
[SETUP_NEEDED](#) (galaxy.web.framework.openid_manager.OpenIDManager attribute), 244
[Sff](#) (class in galaxy.datatypes.binary), 45
[SharableItemSecurityMixin](#) (class in galaxy.web.base.controller), 236
[SharableMixin](#) (class in galaxy.web.base.controller), 236
[share\(\)](#) (galaxy.web.base.controller.SharableMixin method), 236
[share\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[share\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[share\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[share\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 338
[share_restricted\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[shared_list_grid](#) (galaxy.webapps.galaxy.controllers.history.HistoryController attribute), 310
[SharedComputeEnvironment](#) (class in galaxy.jobs), 110
[SharedHistoryListGrid](#) (class in galaxy.webapps.galaxy.controllers.history), 312
[SharedHistoryListGrid.DatasetsByStateColumn](#) (class in galaxy.webapps.galaxy.controllers.history), 312
[SharedHistoryListGrid.SharedByColumn](#) (class in galaxy.webapps.galaxy.controllers.history), 312
[sharing\(\)](#) (galaxy.web.base.controller.SharableMixin method), 236
[sharing\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 310
[sharing\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 318
[sharing\(\)](#) (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 334
[sharing\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 338
[SharingStatusColumn](#) (class in galaxy.web.framework.helpers.grids), 251
[ShellJobRunner](#) (class in galaxy.jobs.runners.cli), 119
[show\(\)](#) (galaxy.webapps.galaxy.api.dataset_collections.DatasetCollectionsController method), 9, 265
[show\(\)](#) (galaxy.webapps.galaxy.api.folder_contents.FolderContentsController method), 11, 267
[show\(\)](#) (galaxy.webapps.galaxy.api.folders.FoldersController method), 12, 268
[show\(\)](#) (galaxy.webapps.galaxy.api.forms.FormDefinitionApiController method), 13, 269
[show\(\)](#) (galaxy.webapps.galaxy.api.genomes.GenomesController method), 13, 269
[show\(\)](#) (galaxy.webapps.galaxy.api.group_roles.GroupRolesApiController method), 14, 269
[show\(\)](#) (galaxy.webapps.galaxy.api.group_users.GroupUsersApiController method), 14, 270
[show\(\)](#) (galaxy.webapps.galaxy.api.groups.GroupApiController method), 14, 270
[show\(\)](#) (galaxy.webapps.galaxy.api.histories.HistoriesController method), 16, 271
[show\(\)](#) (galaxy.webapps.galaxy.api.history_contents.HistoryContentsController method), 18, 274
[show\(\)](#) (galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController method), 19, 275
[show\(\)](#) (galaxy.webapps.galaxy.api.jobs.JobController method), 22, 277
[show\(\)](#) (galaxy.webapps.galaxy.api.lda_datasets.LibraryDatasetsController method), 23, 279
[show\(\)](#) (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 26, 282
[show\(\)](#) (galaxy.webapps.galaxy.api.library_contents.LibraryContentsController method), 29, 284
[show\(\)](#) (galaxy.webapps.galaxy.api.pages.PagesController method), 31, 286
[show\(\)](#) (galaxy.webapps.galaxy.api.provenance.BaseProvenanceController method), 31, 287
[show\(\)](#) (galaxy.webapps.galaxy.api.quotas.QuotaApiController method), 32, 287
[show\(\)](#) (galaxy.webapps.galaxy.api.request_types.RequestTypeApiController method), 32, 288

- sniff() (galaxy.datatypes.assembly.Sequences method), 42
- sniff() (galaxy.datatypes.binary.Bam method), 43
- sniff() (galaxy.datatypes.binary.Bcf method), 43
- sniff() (galaxy.datatypes.binary.BigWig method), 44
- sniff() (galaxy.datatypes.binary.GeminiSQLite method), 44
- sniff() (galaxy.datatypes.binary.Sff method), 45
- sniff() (galaxy.datatypes.binary.SQLite method), 45
- sniff() (galaxy.datatypes.binary.TwoBit method), 46
- sniff() (galaxy.datatypes.binary.Xlsx method), 46
- sniff() (galaxy.datatypes.data.Newick method), 50
- sniff() (galaxy.datatypes.data.Nexus method), 50
- sniff() (galaxy.datatypes.genetics.GenomeGraphs method), 52
- sniff() (galaxy.datatypes.genetics.rgSampleList method), 55
- sniff() (galaxy.datatypes.genetics.SNPMatrix method), 55
- sniff() (galaxy.datatypes.images.Bmp method), 56
- sniff() (galaxy.datatypes.images.Eps method), 56
- sniff() (galaxy.datatypes.images.Gif method), 56
- sniff() (galaxy.datatypes.images.Gmaj method), 56
- sniff() (galaxy.datatypes.images.Html method), 57
- sniff() (galaxy.datatypes.images.Im method), 57
- sniff() (galaxy.datatypes.images.Image method), 57
- sniff() (galaxy.datatypes.images.Jpg method), 57
- sniff() (galaxy.datatypes.images.Pbm method), 58
- sniff() (galaxy.datatypes.images.Pcd method), 58
- sniff() (galaxy.datatypes.images.Pcx method), 58
- sniff() (galaxy.datatypes.images.Pdf method), 58
- sniff() (galaxy.datatypes.images.Pgm method), 58
- sniff() (galaxy.datatypes.images.Png method), 58
- sniff() (galaxy.datatypes.images.Ppm method), 59
- sniff() (galaxy.datatypes.images.Psd method), 59
- sniff() (galaxy.datatypes.images.Rast method), 59
- sniff() (galaxy.datatypes.images.Rgb method), 59
- sniff() (galaxy.datatypes.images.Tiff method), 59
- sniff() (galaxy.datatypes.images.Xbm method), 59
- sniff() (galaxy.datatypes.images.Xpm method), 59
- sniff() (galaxy.datatypes.interval.Bed method), 60
- sniff() (galaxy.datatypes.interval.BedStrict method), 61
- sniff() (galaxy.datatypes.interval.ChromatinInteractions method), 61
- sniff() (galaxy.datatypes.interval.CustomTrack method), 61
- sniff() (galaxy.datatypes.interval.ENCODERPeak method), 62
- sniff() (galaxy.datatypes.interval.Gff method), 63
- sniff() (galaxy.datatypes.interval.Gff3 method), 63
- sniff() (galaxy.datatypes.interval.Gtf method), 64
- sniff() (galaxy.datatypes.interval.Interval method), 65
- sniff() (galaxy.datatypes.interval.Wiggle method), 66
- sniff() (galaxy.datatypes.ngsindex.BowtieIndex method), 70
- sniff() (galaxy.datatypes.qualityscore.QualityScore454 method), 71
- sniff() (galaxy.datatypes.qualityscore.QualityScoreSOLiD method), 71
- sniff() (galaxy.datatypes.sequence.Axt method), 73
- sniff() (galaxy.datatypes.sequence.csFasta method), 77
- sniff() (galaxy.datatypes.sequence.Fasta method), 73
- sniff() (galaxy.datatypes.sequence.Fastq method), 74
- sniff() (galaxy.datatypes.sequence.Lav method), 75
- sniff() (galaxy.datatypes.sequence.Maf method), 76
- sniff() (galaxy.datatypes.sequence.SequenceSplitLocations method), 77
- sniff() (galaxy.datatypes.tabular.Eland method), 80
- sniff() (galaxy.datatypes.tabular.ElandMulti method), 81
- sniff() (galaxy.datatypes.tabular.Pileup method), 81
- sniff() (galaxy.datatypes.tabular.Sam method), 82
- sniff() (galaxy.datatypes.tabular.Vcf method), 84
- sniff() (galaxy.datatypes.xml.CisML method), 84
- sniff() (galaxy.datatypes.xml.GenericXml method), 85
- sniff() (galaxy.datatypes.xml.MEMEXml method), 85
- sniff() (galaxy.datatypes.xml.Owl method), 85
- sniff() (galaxy.datatypes.xml.Phyloxml method), 86
- sniffable_binary_formats (galaxy.datatypes.binary.Binary attribute), 44
- sniffers() (galaxy.webapps.galaxy.api.datatypes.DatatypesController method), 9, 265
- SNPMatrix (class in galaxy.datatypes.genetics), 55
- Snptest (class in galaxy.datatypes.genetics), 55
- sort() (galaxy.web.framework.helpers.grids.BooleanColumn method), 245
- sort() (galaxy.web.framework.helpers.grids.CommunityRatingColumn method), 246
- sort() (galaxy.web.framework.helpers.grids.DateTimeColumn method), 246
- sort() (galaxy.web.framework.helpers.grids.GridColumn method), 248
- sort() (galaxy.web.framework.helpers.grids.IntegerColumn method), 249
- sort() (galaxy.web.framework.helpers.grids.OwnerColumn method), 250
- sort() (galaxy.web.framework.helpers.grids.ReverseSortColumn method), 251
- sort() (galaxy.web.framework.helpers.grids.TextColumn method), 252
- sort() (galaxy.webapps.galaxy.controllers.history.HistoryListGrid.DeletedC method), 311
- sort_block_components_by_block() (in module galaxy.tools.util.maf_utilities), 212
- sort_by_attr() (galaxy.security.GalaxyRBACAgent method), 166
- sort_by_attr() (in module galaxy.web.base.controller), 240
- sort_by_attr() (in module galaxy.webapps.galaxy.controllers.library_common),

SortByColumnFilter	(class in galaxy.tools.parameters.dynamic_options), 199	SpecifiedDateListGrid.EmailColumn	(class in galaxy.webapps.reports.controllers.workflows), 343
source_dataset_chain	(galaxy.model.DatasetInstance attribute), 126	SpecifiedDateListGrid.JobIdColumn	(class in galaxy.webapps.reports.controllers.jobs), 340
source_library_dataset	(galaxy.model.DatasetInstance attribute), 126	SpecifiedDateListGrid.RequestNameColumn	(class in galaxy.webapps.reports.controllers.sample_tracking), 341
spec	(galaxy.datatypes.metadata.MetadataCollection attribute), 68	SpecifiedDateListGrid.SpecifiedDateColumn	(class in galaxy.webapps.reports.controllers.jobs), 340
specified_date()	(galaxy.webapps.reports.controllers.users.Users method), 343	SpecifiedDateListGrid.SpecifiedDateColumn	(class in galaxy.webapps.reports.controllers.sample_tracking), 342
specified_date_handler()	(galaxy.webapps.reports.controllers.jobs.Jobs method), 339	SpecifiedDateListGrid.StateColumn	(class in galaxy.webapps.reports.controllers.workflows), 344
specified_date_handler()	(galaxy.webapps.reports.controllers.workflows.Workflows method), 344	SpecifiedDateListGrid.StateColumn	(class in galaxy.webapps.reports.controllers.jobs), 340
specified_date_list_grid	(galaxy.webapps.reports.controllers.jobs.Jobs attribute), 339	SpecifiedDateListGrid.StateColumn	(class in galaxy.webapps.reports.controllers.jobs), 340
specified_date_list_grid	(galaxy.webapps.reports.controllers.workflows.Workflows attribute), 344	SpecifiedDateListGrid.UserColumn	(class in galaxy.webapps.reports.controllers.sample_tracking), 342
specified_month()	(galaxy.webapps.reports.controllers.users.Users method), 343	SpecifiedDateListGrid.UserColumn	(class in galaxy.webapps.reports.controllers.workflows), 344
specified_month_all()	(galaxy.webapps.reports.controllers.jobs.Jobs method), 339	SpecifiedDateListGrid.UserColumn	(class in galaxy.webapps.reports.controllers.sample_tracking), 342
specified_month_in_error()	(galaxy.webapps.reports.controllers.jobs.Jobs method), 339	SpecifiedDateListGrid.UserColumn	(class in galaxy.webapps.reports.controllers.workflows), 344
SpecifiedDateListGrid	(class in galaxy.webapps.reports.controllers.jobs), 339	SpecifiedDateListGrid.WorkflowNameColumn	(class in galaxy.webapps.reports.controllers.workflows), 344
SpecifiedDateListGrid	(class in galaxy.webapps.reports.controllers.sample_tracking), 341	SplicedAlignment	(class in galaxy.tools.util.maf_utilities), 210
SpecifiedDateListGrid	(class in galaxy.webapps.reports.controllers.workflows), 343	split()	(galaxy.datatypes.data.Text class method), 51
SpecifiedDateListGrid.CreateTimeColumn	(class in galaxy.webapps.reports.controllers.jobs), 339	split()	(galaxy.datatypes.sequence.Alignment method), 73
SpecifiedDateListGrid.CreateTimeColumn	(class in galaxy.webapps.reports.controllers.sample_tracking), 341	split()	(galaxy.datatypes.sequence.Fasta class method), 74
SpecifiedDateListGrid.CreateTimeColumn	(class in galaxy.webapps.reports.controllers.workflows), 343	split()	(galaxy.datatypes.sequence.Fastq class method), 75
SpecifiedDateListGrid.EmailColumn	(class in galaxy.webapps.reports.controllers.jobs), 339	split()	(galaxy.datatypes.sequence.Sequence method), 77
SpecifiedDateListGrid.EmailColumn	(class in galaxy.webapps.reports.controllers.sample_tracking), 341	split()	(galaxy_utils.sequence.fastq.fastqSplitter method), 353
SpecifiedDateListGrid.CreateTimeColumn	(class in galaxy.webapps.reports.controllers.workflows), 343	split_sections()	(in module pkg_resources), 363
SpecifiedDateListGrid.EmailColumn	(class in galaxy.webapps.reports.controllers.jobs), 339	splitLinebyWhitespaces()	(galaxy.visualization.data_providers.phylviz.nexusparser.NexusParser method), 228
SpecifiedDateListGrid.EmailColumn	(class in galaxy.webapps.reports.controllers.sample_tracking), 341	SQLite	(class in galaxy.datatypes.binary), 45
		sqlite_dataprovider()	(galaxy.datatypes.binary.SQLite method), 45
		src_merge()	(in module galaxy.tools.util.maf_utilities), 212
		src_split()	(in module galaxy.tools.util.maf_utilities), 212

- stack_trace_string() (in module galaxy.util.debugging), 218
- standard_filters (galaxy.web.framework.helpers.grids.Grid attribute), 248
- standard_filters (galaxy.webapps.galaxy.controllers.admin.Grid attribute), 296
- standard_filters (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 298
- standard_filters (galaxy.webapps.galaxy.controllers.admin.RSListGrid attribute), 299
- standard_filters (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 299
- standard_filters (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 301
- standard_filters (galaxy.webapps.galaxy.controllers.dataset.HistoryDataCollectionListGrid attribute), 306
- standard_filters (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 311
- standard_filters (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid attribute), 312
- standard_filters (galaxy.webapps.galaxy.controllers.library.LibraryListGrid attribute), 313
- standard_filters (galaxy.webapps.galaxy.controllers.library_admin.LibraryListGrid attribute), 314
- standard_filters (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute), 341
- standard_filters (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid attribute), 342
- standard_filters (galaxy.webapps.reports.controllers.workflow_tracking.SpecifiedDateListGrid attribute), 344
- start (galaxy.tools.util.maf_utilities.SplicedAlignment attribute), 210
- start() (galaxy.jobs.handler.JobHandler method), 112
- start() (galaxy.jobs.handler.JobHandlerQueue method), 112
- start() (galaxy.jobs.manager.JobManager method), 113
- start() (galaxy.jobs.manager.NoopHandler method), 113
- start() (galaxy.util.debugging.SimpleProfiler method), 218
- state (galaxy.model.DatasetCollection attribute), 123
- state (galaxy.model.DatasetCollectionInstance attribute), 124
- state (galaxy.model.DatasetInstance attribute), 126
- state (galaxy.model.Request attribute), 138
- state (galaxy.model.Sample attribute), 139
- state_fields (galaxy.workflow.modules.InputDataCollectionModule attribute), 345
- state_fields (galaxy.workflow.modules.InputDataModule attribute), 345
- state_fields (galaxy.workflow.modules.PauseModule attribute), 346
- StateColumn (class in galaxy.web.framework.helpers.grids), 251
- Statement (class in galaxy.datatypes.metadata), 69
- states (galaxy.model.Dataset attribute), 123
- states (galaxy.model.DatasetInstance attribute), 126
- states (galaxy.model.DeferredJob attribute), 127
- states (galaxy.model.Job attribute), 133
- states (galaxy.model.Request attribute), 138
- states (galaxy.model.Task attribute), 141
- states (galaxy.model.TransferJob attribute), 141
- states (galaxy.model.WorkflowInvocation attribute), 144
- StaticGridFilter (class in galaxy.tools.parameters.dynamic_options), 193
- status_code (galaxy.exceptions.ActionInputError attribute), 97
- status_code (galaxy.exceptions.AdminRequiredException attribute), 97
- status_code (galaxy.exceptions.AuthenticationFailed attribute), 97
- status_code (galaxy.exceptions.AuthenticationRequired attribute), 97
- status_code (galaxy.exceptions.ConfigDoesNotAllowException attribute), 97
- status_code (galaxy.exceptions.Conflict attribute), 97
- status_code (galaxy.exceptions.DeprecatedMethod attribute), 98
- status_code (galaxy.exceptions.DuplicatedSlugException attribute), 98
- status_code (galaxy.exceptions.InconsistentDatabase attribute), 98
- status_code (galaxy.exceptions.InsufficientPermissionsException attribute), 98
- status_code (galaxy.exceptions.InternalServerError attribute), 98
- status_code (galaxy.exceptions.ItemAccessibilityException attribute), 98
- status_code (galaxy.exceptions.ItemOwnershipException attribute), 98
- status_code (galaxy.exceptions.MalformedId attribute), 98
- status_code (galaxy.exceptions.MessageException attribute), 99
- status_code (galaxy.exceptions.NotImplemented attribute), 99
- status_code (galaxy.exceptions.ObjectAttributeInvalidException attribute), 99
- status_code (galaxy.exceptions.ObjectAttributeMissingException attribute), 99
- status_code (galaxy.exceptions.ObjectNotFound attribute), 99
- status_code (galaxy.exceptions.RequestParameterInvalidException attribute), 99
- status_code (galaxy.exceptions.RequestParameterMissingException attribute), 99
- status_code (galaxy.exceptions.ToolMetaParameterException attribute), 100

System (class in galaxy.webapps.reports.controllers.system), tag_autocomplete_data() (galaxy.webapps.galaxy.controllers.tag.TagsController method), 342

T

tableize() (galaxy.util.inflection.Base method), 219
 tableize() (galaxy.util.inflection.Inflector method), 220
 Tabular (class in galaxy.datatypes.tabular), 83
 TabularToolDataTable (class in galaxy.tools.data), 181
 Tag (class in galaxy.model), 140
 tag (galaxy.tools.parameters.output.BooleanFilter attribute), 201
 tag (galaxy.tools.parameters.output.ColumnReplaceFilter attribute), 202
 tag (galaxy.tools.parameters.output.ColumnStripFilter attribute), 202
 tag (galaxy.tools.parameters.output.DatatypeIsInstanceToolOutputActionOption attribute), 202
 tag (galaxy.tools.parameters.output.FormatToolOutputActionOption attribute), 202
 tag (galaxy.tools.parameters.output.FromDataTableOutputActionOption attribute), 202
 tag (galaxy.tools.parameters.output.FromFileToolOutputActionOption attribute), 202
 tag (galaxy.tools.parameters.output.FromParamToolOutputActionOption attribute), 202
 tag (galaxy.tools.parameters.output.InsertColumnToolOutputActionOption attribute), 202
 tag (galaxy.tools.parameters.output.MetadataToolOutputActionOption attribute), 202
 tag (galaxy.tools.parameters.output.MetadataValueFilter attribute), 203
 tag (galaxy.tools.parameters.output.MultipleSplitterFilter attribute), 203
 tag (galaxy.tools.parameters.output.NullToolOutputActionOption attribute), 203
 tag (galaxy.tools.parameters.output.ParamValueToolOutputActionOption attribute), 203
 tag (galaxy.tools.parameters.output.StringFunctionFilter attribute), 203
 tag (galaxy.tools.parameters.output.ToolOutputAction attribute), 203
 tag (galaxy.tools.parameters.output.ToolOutputActionConditional attribute), 203
 tag (galaxy.tools.parameters.output.ToolOutputActionConditionalWhen attribute), 204
 tag (galaxy.tools.parameters.output.ToolOutputActionGroup attribute), 204
 tag (galaxy.tools.parameters.output.ToolOutputActionOption attribute), 204
 tag (galaxy.tools.parameters.output.ToolOutputActionOptionFilter attribute), 204
 tag (galaxy.tools.parameters.output.ValueToolOutputActionConditionalWhen attribute), 204
 tag_mapping() (in module galaxy.model.mapping), 147
 tag_outputs() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 338
 tagged_item_class (galaxy.webapps.galaxy.api.item_tags.HistoryContentTagsController attribute), 19, 275
 tagged_item_class (galaxy.webapps.galaxy.api.item_tags.HistoryTagsController attribute), 19, 275
 tagged_item_class (galaxy.webapps.galaxy.api.item_tags.WorkflowTagsController attribute), 20, 275
 tagged_item_id (galaxy.webapps.galaxy.api.annotations.HistoryAnnotationsController attribute), 8, 264
 tagged_item_id (galaxy.webapps.galaxy.api.annotations.HistoryContentAnnotationsController attribute), 8, 264
 tagged_item_id (galaxy.webapps.galaxy.api.annotations.WorkflowAnnotationsController attribute), 8, 264
 tagged_item_id (galaxy.webapps.galaxy.api.item_tags.HistoryContentTagsController attribute), 19, 275
 tagged_item_id (galaxy.webapps.galaxy.api.item_tags.HistoryTagsController attribute), 20, 275
 tagged_item_id (galaxy.webapps.galaxy.api.item_tags.WorkflowTagsController attribute), 20, 275
 TagHandler (class in galaxy.tags.tag_handler), 168
 TagsController (class in galaxy.webapps.galaxy.controllers.tag), 326
 TagsManager (class in galaxy.managers.tags), 156
 Task (class in galaxy.model), 140
 TaskedJobRunner (class in galaxy.jobs.runners.tasks), 121
 TaskMetricNumeric (class in galaxy.model), 141
 TaskMetricText (class in galaxy.model), 141
 TaskWrapper (class in galaxy.jobs), 110
 Taxonomy (class in galaxy.datatypes.tabular), 84
 TempFile (class in log_tempfile), 355
 TempFileHandler (class in galaxy.tools.util.maf_utilities), 210
 Template (class in galaxy.external_services.actions), 101
 template (galaxy.web.framework.helpers.grids.Grid attribute), 248
 template (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 296
 template (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 298
 template (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 299
 template (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 299
 template (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 301
 template (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationController attribute), 306
 template (galaxy.webapps.galaxy.controllers.external_service.ExternalServiceController attribute), 307

template (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 308
 template (galaxy.webapps.galaxy.controllers.history.HistoryListGrid (galaxy.web.form_builder.FTPFileField attribute), attribute), 312
 template (galaxy.webapps.galaxy.controllers.library.LibraryListGrid.sleeping() (galaxy.util.heartbeat.Heartbeat attribute), 313 method), 218
 template (galaxy.webapps.galaxy.controllers.library_admin.LibraryAdminGrid (galaxy.datatypes.images), 59 attribute), 314
 template (galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid attribute), 317
 template (galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid (in module galaxy.datatypes.converters.lped_to_fped_converter), attribute), 321
 template (galaxy.webapps.galaxy.controllers.requests_admin.RequestAdminGrid (in module galaxy.datatypes.converters.lped_to_pbped_converter), attribute), 322
 template (galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid (in module galaxy.datatypes.converters.pbped_ldreduced_converter), attribute), 325
 template (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid (galaxy.datatypes.converters.pbped_ldreduced_converter), attribute), 329
 template (galaxy.webapps.galaxy.controllers.visualization.HistoryDatasetSelectionGrid (in module galaxy.datatypes.converters.pbped_to_lped_converter), attribute), 331
 template (galaxy.webapps.galaxy.controllers.visualization.HistorySelectionGrid attribute), 332
 template (galaxy.webapps.galaxy.controllers.visualization.LibrarySelectionGrid attribute), 332
 template (galaxy.webapps.galaxy.controllers.visualization.TracksterSelectionGrid attribute), 333
 template (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute), 341
 template (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid attribute), 342
 template (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid attribute), 344
 template_context (galaxy.webapps.demo_sequencer.framework.DemoWorkflowAction attribute), 258
 templates() (in module galaxy.web.framework.helpers), 245
 templates_dict() (galaxy.model.LibraryDatasetDatasetAssociation attribute), 307
 templates_json() (galaxy.model.LibraryDatasetDatasetAssociation attribute), 309
 terminal_states (galaxy.model.TransferJob attribute), 141
 test_data() (galaxy.tools.test.ToolTestBuilder method), 178
 test_data_iter() (in module galaxy.tools.test), 179
 TestCollectionDef (class in galaxy.tools.test), 178
 tests (galaxy.tools.Tool attribute), 175
 Text (class in galaxy.datatypes.data), 50
 TextArea (class in galaxy.web.form_builder), 233
 TextColumn (class in galaxy.web.framework.helpers.grids), 252
 TextField (class in galaxy.web.form_builder), 233
 TextToolParameter (class in galaxy.tools.parameters.basic), 195

title (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid attribute),	317
title (galaxy.webapps.galaxy.controllers.page.PageListGrid attribute),	319
title (galaxy.webapps.galaxy.controllers.page.PageSelectionGrid attribute),	319
title (galaxy.webapps.galaxy.controllers.page.VisualizationSelectionGrid attribute),	319
title (galaxy.webapps.galaxy.controllers.page.WorkflowSelectionGrid attribute),	319
title (galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid attribute),	321
title (galaxy.webapps.galaxy.controllers.requests_admin.DataTransferGrid attribute),	322
title (galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid attribute),	325
title (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid attribute),	329
title (galaxy.webapps.galaxy.controllers.visualization.HistoryDatasetSelectionGrid attribute),	331
title (galaxy.webapps.galaxy.controllers.visualization.HistorySelectOnClick attribute),	332
title (galaxy.webapps.galaxy.controllers.visualization.LibrarySelectOnClick attribute),	332
title (galaxy.webapps.galaxy.controllers.visualization.TracksToSelectionGrid attribute),	333
title (galaxy.webapps.galaxy.controllers.visualization.VisualizationAllPublishedGrid attribute),	333
title (galaxy.webapps.galaxy.controllers.visualization.VisualizationListGrid attribute),	335
title (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAllPublishedGrid attribute),	335
title (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowJobsGrid attribute),	336
title (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute),	341
title (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDataGrid attribute),	342
title (galaxy.webapps.reports.controllers.workflows.SpecifiedDataListGrid attribute),	344
title_by_index() (galaxy.tools.parameters.grouping.UploadDataset method),	200
title_plural (galaxy.tools.parameters.grouping.Repeat attribute),	200
title_plural (galaxy.tools.parameters.grouping.UploadDataset attribute),	200
titleize() (galaxy.util.inflexion.Base method),	219
titleize() (galaxy.util.inflexion.Inflector method),	220
tmp_dir (galaxy.datatypes.metadata.MetadataTempFile attribute),	69
to_base_space() (galaxy_utils.sequence.transform.ColorSpaceConverter method),	354
to_bibtex() (galaxy.managers.citations.BibtexCitation method),	149
to_color_space() (galaxy_utils.sequence.transform.ColorSpaceConverter method),	354
to_dict() (galaxy.managers.citations.BaseCitation method),	149
to_dict() (galaxy.model.History method),	129
to_dict() (galaxy.model.HistoryDatasetAssociation method),	130
to_dict() (galaxy.model.HistoryDatasetCollectionAssociation method),	130
to_dict() (galaxy.model.item_attrs.Dictifiable method),	130
to_dict() (galaxy.model.Job method),	133
to_dict() (galaxy.model.Library method),	134
to_dict() (galaxy.model.LibraryDataset method),	135
to_dict() (galaxy.model.LibraryDatasetCollectionAssociation method),	135
to_dict() (galaxy.model.LibraryDatasetDatasetAssociation method),	135
to_dict() (galaxy.model.LibraryFolder method),	136
to_dict() (galaxy.model.Page method),	136
to_dict() (galaxy.model.PageRevision method),	137
to_dict() (galaxy.model.StoredWorkflow method),	139
to_dict() (galaxy.model.Workflow method),	143
to_dict() (galaxy.model.WorkflowInvocation method),	143
to_dict() (galaxy.model.WorkflowInvocationStep method),	144
to_dict() (galaxy.model.WorkflowRequest method),	144
tools.all_published_get_grid (galaxy.tools.data.TabularToolDataTable method),	182
tools.parameters.basic.BaseURLToolParameter (galaxy.tools.parameters.basic.BaseURLToolParameter method),	185
tools.parameters.basic.BooleanToolParameter (galaxy.tools.parameters.basic.BooleanToolParameter method),	186
tools.parameters.basic.ColumnListParameter (galaxy.tools.parameters.basic.ColumnListParameter method),	187
tools.parameters.basic.DataCollectionToolParameter (galaxy.tools.parameters.basic.DataCollectionToolParameter method),	187
tools.parameters.basic.DataToolParameter (galaxy.tools.parameters.basic.DataToolParameter method),	187
tools.parameters.basic.DrillDownSelectToolParameter (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method),	190
tools.parameters.basic.GenomeBuildParameter (galaxy.tools.parameters.basic.GenomeBuildParameter method),	192
tools.parameters.basic.SelectToolParameter (galaxy.tools.parameters.basic.SelectToolParameter method),	195
tools.parameters.basic.TextToolParameter (galaxy.tools.parameters.basic.TextToolParameter method),	195
tools.parameters.basic.ToolParameter (galaxy.tools.parameters.basic.ToolParameter method),	196
tools.parameters.grouping.Conditional (galaxy.tools.parameters.grouping.Conditional method),	199

[to_dict\(\) \(galaxy.tools.parameters.grouping.ConditionalWhen method\), 200](#)
[to_dict\(\) \(galaxy.tools.parameters.grouping.Group method\), 200](#)
[to_dict\(\) \(galaxy.tools.parameters.grouping.Repeat method\), 200](#)
[to_dict\(\) \(galaxy.tools.Tool method\), 175](#)
[to_dict\(\) \(galaxy.tools.ToolBox method\), 176](#)
[to_dict\(\) \(galaxy.tools.ToolSection method\), 177](#)
[to_dict\(\) \(galaxy.visualization.genomes.Genome method\), 225](#)
[to_dict\(\) \(galaxy.web.form_builder.SelectField method\), 233](#)
[to_dict_helper\(\) \(in module galaxy.tools\), 178](#)
[to_DNA\(\) \(in module galaxy_utils.sequence.transform\), 354](#)
[to_external_value\(\) \(galaxy.datatypes.metadata.FileParameter method\), 67](#)
[to_external_value\(\) \(galaxy.datatypes.metadata.MetadataParameter method\), 68](#)
[to_filename\(\) \(in module pkg_resources\), 363](#)
[to_history_dataset_association\(\) \(galaxy.model.LibraryDatasetDatasetAssociation method\), 136](#)
[to_html_value\(\) \(galaxy.tools.parameters.basic.BooleanToolParameter method\), 186](#)
[to_html_value\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 195](#)
[to_html_value\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 196](#)
[to_JSON\(\) \(galaxy.datatypes.metadata.MetadataTempFile method\), 69](#)
[to_JSON_dict\(\) \(galaxy.datatypes.metadata.MetadataCollection method\), 68](#)
[to_library_dataset_dataset_association\(\) \(galaxy.model.HistoryDatasetAssociation method\), 130](#)
[to_param_dict_string\(\) \(galaxy.tools.parameters.basic.BooleanToolParameter method\), 186](#)
[to_param_dict_string\(\) \(galaxy.tools.parameters.basic.DataToolParameter method\), 188](#)
[to_param_dict_string\(\) \(galaxy.tools.parameters.basic.DrillDownSelectToolParameter method\), 190](#)
[to_param_dict_string\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 195](#)
[to_param_dict_string\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 196](#)
[to_python\(\) \(galaxy.tools.parameters.basic.BooleanToolParameter method\), 186](#)
[to_python\(\) \(galaxy.tools.parameters.basic.DataCollectionToolParameter method\), 187](#)
[to_python\(\) \(galaxy.tools.parameters.basic.DataToolParameter method\), 188](#)
[to_python\(\) \(galaxy.tools.parameters.basic.FileToolParameter method\), 190](#)
[to_python\(\) \(galaxy.tools.parameters.basic.FloatToolParameter method\), 191](#)
[to_python\(\) \(galaxy.tools.parameters.basic.FTPFileToolParameter method\), 190](#)
[to_python\(\) \(galaxy.tools.parameters.basic.IntegerToolParameter method\), 193](#)
[to_python\(\) \(galaxy.tools.parameters.basic.LibraryDatasetToolParameter method\), 193](#)
[to_python\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 196](#)
[to_RNA\(\) \(in module galaxy_utils.sequence.transform\), 355](#)
[to_safe_string\(\) \(galaxy.datatypes.metadata.DictParameter method\), 67](#)
[to_safe_string\(\) \(galaxy.datatypes.metadata.FileParameter method\), 67](#)
[to_safe_string\(\) \(galaxy.datatypes.metadata.MetadataParameter method\), 68](#)
[to_string\(\) \(galaxy.datatypes.metadata.ColumnTypesParameter method\), 66](#)
[to_string\(\) \(galaxy.datatypes.metadata.DictParameter method\), 67](#)
[to_string\(\) \(galaxy.datatypes.metadata.FileParameter method\), 67](#)
[to_string\(\) \(galaxy.datatypes.metadata.ListParameter method\), 67](#)
[to_string\(\) \(galaxy.datatypes.metadata.MetadataParameter method\), 68](#)
[to_string\(\) \(galaxy.datatypes.metadata.PythonObjectParameter method\), 69](#)
[to_string\(\) \(galaxy.datatypes.metadata.SelectParameter method\), 69](#)
[to_string\(\) \(galaxy.tools.parameters.basic.DataCollectionToolParameter method\), 187](#)
[to_string\(\) \(galaxy.tools.parameters.basic.DataToolParameter method\), 188](#)
[to_string\(\) \(galaxy.tools.parameters.basic.FileToolParameter method\), 190](#)
[to_string\(\) \(galaxy.tools.parameters.basic.FloatToolParameter method\), 190](#)
[to_string\(\) \(galaxy.tools.parameters.basic.FTPFileToolParameter method\), 190](#)
[to_string\(\) \(galaxy.tools.parameters.basic.IntegerToolParameter method\), 193](#)
[to_string\(\) \(galaxy.tools.parameters.basic.LibraryDatasetToolParameter method\), 193](#)
[to_string\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 196](#)
[to_unicode\(\) \(in module galaxy.web.framework.helpers\), 245](#)
[to_xml_file\(\) \(galaxy.datatypes.registry.Registry method\), 73](#)

[to_xml_file\(\)](#) (galaxy.tools.data.ToolDataTableManager method), 183
[toJson\(\)](#) (galaxy.visualization.data_providers.phyloviz.baseparser.BaseParser method), 227
[toJson\(\)](#) (galaxy.visualization.data_providers.phyloviz.baseparser.BaseParser method), 227
[Tool](#) (class in galaxy.tools), 171
[tool](#) (galaxy.tools.parameters.output.ToolOutputAction attribute), 203
[tool](#) (galaxy.tools.parameters.output.ToolOutputActionConditional attribute), 203
[tool](#) (galaxy.tools.parameters.output.ToolOutputActionGroup attribute), 204
[tool](#) (galaxy.tools.parameters.output.ToolOutputActionOption attribute), 204
[tool](#) (galaxy.tools.parameters.output.ToolOutputActionOption attribute), 204
[tool_class](#) (in module galaxy.tools), 178
[tool_dependency_grid](#) (galaxy.webapps.galaxy.controllers.admin_tools_shed.ToolShedTools attribute), 303
[tool_dependency_status_updates\(\)](#) (galaxy.webapps.galaxy.controllers.admin_tools_shed.ToolShedTools method), 303
[tool_fail\(\)](#) (in module galaxy.tools.util.maf_utilities), 212
[tool_help\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController method), 326
[tool_per_month\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 339
[tool_search\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController method), 326
[tool_shed_repository](#) (galaxy.tools.Tool attribute), 175
[tool_type](#) (galaxy.tools.AsyncDataSourceTool attribute), 170
[tool_type](#) (galaxy.tools.DataDestinationTool attribute), 170
[tool_type](#) (galaxy.tools.DataManagerTool attribute), 170
[tool_type](#) (galaxy.tools.DataSourceTool attribute), 171
[tool_type](#) (galaxy.tools.ExportHistoryTool attribute), 171
[tool_type](#) (galaxy.tools.GenomeIndexTool attribute), 171
[tool_type](#) (galaxy.tools.ImportHistoryTool attribute), 171
[tool_type](#) (galaxy.tools.OutputParameterJSONTool attribute), 171
[tool_type](#) (galaxy.tools.SetMetadataTool attribute), 171
[tool_type](#) (galaxy.tools.Tool attribute), 175
[tool_version](#) (galaxy.tools.Tool attribute), 175
[tool_version_ids](#) (galaxy.tools.Tool attribute), 175
[tool_version_list_grid](#) (galaxy.web.base.controllers.admin.Admin attribute), 241
[tool_version_list_grid](#) (galaxy.webapps.galaxy.controllers.admin.Admin attribute), 295
[tool_versions](#) (galaxy.tools.Tool attribute), 175
[tool_versions\(\)](#) (galaxy.web.base.controllers.admin.Admin method), 241
[ToolAction](#) (class in galaxy.tools.actions), 179
[ToolBox](#) (class in galaxy.tools), 175
[toolbox_filters\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), 329
[ToolBoxSearch](#) (class in galaxy.tools.search), 209
[ToolDataNode](#) (class in galaxy.webapps.galaxy.api.tool_data), 33, 289
[ToolDataTable](#) (class in galaxy.tools.data), 182
[ToolDataTableManager](#) (class in galaxy.tools.data), 182
[ToolInputTranslator](#) (class in galaxy.tools.parameters.input_translation), 201
[ToolMetaParameterException](#), 100
[ToolModule](#) (class in galaxy.workflow.modules), 347
[ToolNotFoundException](#), 177
[ToolOutput](#) (class in galaxy.tools), 177
[ToolOutputAction](#) (class in galaxy.tools.parameters.output), 203
[ToolOutputActionConditional](#) (class in galaxy.tools.parameters.output), 203
[ToolOutputActionConditionalWhen](#) (class in galaxy.tools.parameters.output), 203
[ToolOutputActionGroup](#) (class in galaxy.tools.parameters.output), 204
[ToolOutputActionOption](#) (class in galaxy.tools.parameters.output), 204
[ToolOutputActionOptionFilter](#) (class in galaxy.tools.parameters.output), 204
[ToolParameter](#) (class in galaxy.tools.parameters.basic), 195
[ToolParameterSanitizer](#) (class in galaxy.tools.parameters.sanitize), 205
[ToolRunner](#) (class in galaxy.webapps.galaxy.controllers.tool_runner), 327
[ToolSection](#) (class in galaxy.tools), 177
[ToolSectionLabel](#) (class in galaxy.tools), 177
[ToolShedRepositoriesController](#) (class in galaxy.webapps.galaxy.api.tool_shed_repositories), 34, 289
[ToolStdioExitCode](#) (class in galaxy.tools), 177
[ToolStdioRegex](#) (class in galaxy.tools), 177
[ToolTagAssociation](#) (class in galaxy.model), 141
[ToolTestBuilder](#) (class in galaxy.tools.test), 178
[ToolVersionListGrid](#) (class in galaxy.webapps.galaxy.controllers.admin), 299
[ToolVersionListGrid.ToolIdColumn](#) (class in galaxy.webapps.galaxy.controllers.admin), 299
[ToolVersionListGrid.ToolVersionsColumn](#) (class in galaxy.webapps.galaxy.controllers.admin), 299
[topsort\(\)](#) (in module galaxy.util.topsort), 224
[topsort_levels\(\)](#) (in module galaxy.util.topsort), 224
[total_disk_usage](#) (galaxy.model.GalaxySession attribute), 128

- total_disk_usage (galaxy.model.User attribute), 142
- trace() (galaxy.web.framework.base.WebApplication method), 244
- TraceLoggerProxy (class in galaxy.model.orm.logging_connection_proxy), 147
- track_type (galaxy.datatypes.binary.Bam attribute), 43
- track_type (galaxy.datatypes.binary.BigWig attribute), 44
- track_type (galaxy.datatypes.data.Data attribute), 49
- track_type (galaxy.datatypes.interval.Bed attribute), 60
- track_type (galaxy.datatypes.interval.BedGraph attribute), 61
- track_type (galaxy.datatypes.interval.ChromatinInteractions attribute), 61
- track_type (galaxy.datatypes.interval.Gff attribute), 63
- track_type (galaxy.datatypes.interval.Gff3 attribute), 64
- track_type (galaxy.datatypes.interval.Gtf attribute), 64
- track_type (galaxy.datatypes.interval.Interval attribute), 65
- track_type (galaxy.datatypes.interval.Wiggle attribute), 66
- track_type (galaxy.datatypes.tabular.Sam attribute), 82
- track_type (galaxy.datatypes.tabular.Vcf attribute), 84
- trackster() (galaxy.webapps.galaxy.controllers.visualization.VisualizationController method), 335
- TracksterConfig (class in galaxy.tools), 177
- TracksterSelectionGrid (class in galaxy.webapps.galaxy.controllers.visualization), 333
- transaction_chooser() (galaxy.webapps.demo_sequencer.framework.web.application method), 258
- transfer_error_dataset_files (galaxy.model.Sample attribute), 139
- transfer_part() (in module galaxy.objectstore.s3_multipart_upload), 160
- transfer_status (galaxy.model.SampleDataset attribute), 139
- transfer_status() (galaxy.webapps.galaxy.controllers.dataset.DatasetInteractor method), 305
- TransferJob (class in galaxy.model), 141
- TransferManager (class in galaxy.jobs.transfer_manager), 113
- transferred_dataset_files (galaxy.model.Sample attribute), 139
- transferring_dataset_files (galaxy.model.Sample attribute), 139
- transform_publicname() (in module galaxy.security.validate_user_input), 168
- transform_scores_to_valid_range() (galaxy_utils.sequence.fastq.fastqSequencingRead class method), 353
- transform_scores_to_valid_range_ascii() (galaxy_utils.sequence.fastq.fastqSequencingRead class method), 353
- translate() (galaxy.tools.parameters.input_translation.ToolInputTranslator method), 201
- TransLogger (class in galaxy.web.framework.middleware.translogger), 253
- TrimmedString (class in galaxy.model.custom_types), 145
- throw (galaxy.web.form_builder.FTPFileField attribute), 231
- truncate() (in module galaxy.web.framework.helpers), 245
- TwoBit (class in galaxy.datatypes.binary), 45
- type (galaxy.datatypes.display_applications.parameters.DisplayApplication attribute), 91
- type (galaxy.datatypes.display_applications.parameters.DisplayApplication attribute), 92
- type (galaxy.datatypes.display_applications.parameters.DisplayApplication attribute), 92
- type (galaxy.external_services.actions.ExternalServiceAction attribute), 100
- type (galaxy.external_services.actions.ExternalServiceTemplateAction attribute), 100
- type (galaxy.external_services.actions.ExternalServiceWebAction attribute), 101
- type (galaxy.external_services.actions.ExternalServiceWebAPIAction attribute), 101
- type (galaxy.external_services.parameters.ExternalServiceParameter attribute), 101
- type (galaxy.external_services.parameters.ExternalServiceTemplateParameter attribute), 101
- type (galaxy.external_services.result_handlers.basic.ExternalServiceAction attribute), 103
- type (galaxy.external_services.result_handlers.basic.ExternalServiceAction attribute), 104
- type (galaxy.external_services.result_handlers.basic.ExternalServiceAction attribute), 104
- type (galaxy.external_services.result_handlers.basic.ExternalServiceAction attribute), 104
- type (galaxy.external_services.service.BooleanExternalServiceActionsGroup attribute), 102
- type (galaxy.external_services.service.ExternalServiceActionsConditional attribute), 102
- type (galaxy.external_services.service.ExternalServiceActionsGroupWhen attribute), 102
- type (galaxy.external_services.service.ItemIsInstanceExternalServiceAction attribute), 103
- type (galaxy.external_services.service.ValueExternalServiceActionsGroupWhen attribute), 103
- type (galaxy.forms.forms.FormDefinitionAddressFieldFactory attribute), 104
- type (galaxy.forms.forms.FormDefinitionFieldFactory attribute), 105

type (galaxy.forms.forms.FormDefinitionHistoryFieldFactory attribute), 105
 type (galaxy.forms.forms.FormDefinitionPasswordFieldFactory attribute), 105
 type (galaxy.forms.forms.FormDefinitionSelectFieldFactory attribute), 105
 type (galaxy.forms.forms.FormDefinitionTextFieldFactory attribute), 105
 type (galaxy.forms.forms.FormDefinitionWorkflowFieldFactory attribute), 106
 type (galaxy.forms.forms.FormDefinitionWorkflowMappingFieldFactory attribute), 106
 type (galaxy.sample_tracking.data_transfer.DataTransferFactory attribute), 161
 type (galaxy.sample_tracking.data_transfer.FtpDataTransferFactory attribute), 161
 type (galaxy.sample_tracking.data_transfer.HttpDataTransferFactory attribute), 162
 type (galaxy.sample_tracking.data_transfer.ScpDataTransferFactory attribute), 162
 type (galaxy.tools.parameters.grouping.Conditional attribute), 199
 type (galaxy.tools.parameters.grouping.Repeat attribute), 200
 type (galaxy.tools.parameters.grouping.UploadDataset attribute), 200
 type (galaxy.workflow.modules.InputDataCollectionModule attribute), 345
 type (galaxy.workflow.modules.InputDataModule attribute), 345
 type (galaxy.workflow.modules.PauseModule attribute), 346
 type (galaxy.workflow.modules.ToolModule attribute), 347
 type_key (galaxy.tools.data.TabularToolDataTable attribute), 182
 types (galaxy.model.DefaultQuotaAssociation attribute), 126
 types (galaxy.model.FormDefinition attribute), 127
 types (galaxy.model.Role attribute), 138
 types (galaxy.model.WorkflowRequestInputParameter attribute), 144

U

UCI (class in galaxy.model), 141
 ucsc_links() (galaxy.datatypes.genetics.GenomeGraphs method), 52
 ucsc_links() (galaxy.datatypes.interval.CustomTrack method), 62
 ucsc_links() (galaxy.datatypes.interval.Gff method), 63
 ucsc_links() (galaxy.datatypes.interval.Interval method), 65
 ucsc_links() (galaxy.datatypes.interval.Wiggle method), 66
 UCSCLimitException, 178
 UCSCOutWrapper (class in galaxy.tools.exception_handling), 178
 UCSCProxy (class in galaxy.webapps.galaxy.controllers.ucsc_proxy), 327
 umask_fix_perms() (in module galaxy.util), 216
 unaccent() (galaxy.util.inflection.Base method), 219
 unaccent() (galaxy.util.inflection.Inflector method), 220
 undeletable() (galaxy.model.DatasetInstance method), 126
 undelete() (galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController method), 8, 264
 undelete() (galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController method), 10, 265
 undelete() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 16, 272
 undelete() (galaxy.webapps.galaxy.api.quotas.QuotaApiController method), 32, 287
 undelete() (galaxy.webapps.galaxy.api.users.UserApiController method), 37, 292
 undelete() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 305
 undelete_address() (galaxy.webapps.galaxy.controllers.user.User method), 329
 undelete_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 305
 undelete_external_service() (galaxy.webapps.galaxy.controllers.external_service.ExternalService method), 306
 undelete_form_definition() (galaxy.webapps.galaxy.controllers.forms.Forms method), 308
 undelete_group() (galaxy.web.base.controllers.admin.Admin method), 241
 undelete_library() (galaxy.webapps.galaxy.controllers.library_admin.LibraryAdmin method), 314
 undelete_library_item() (galaxy.webapps.galaxy.controllers.library_common.LibraryCommon method), 315
 undelete_operation (galaxy.web.base.controllers.admin.Admin attribute), 241
 undelete_operation (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy attribute), 296
 undelete_quota() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 296
 undelete_request() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 324
 undelete_request_type() (galaxy.webapps.galaxy.controllers.request_type.RequestType method), 320
 undelete_role() (galaxy.web.base.controllers.admin.Admin method), 241
 undelete_user() (galaxy.web.base.controllers.admin.Admin method), 241
 underscore() (galaxy.util.inflection.Base method), 219
 underscore() (galaxy.util.inflection.Inflector method), 220

`unhide()` (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface.transfer_dataset_files (galaxy.model.Sample attribute), 305)

`unhide_datasets()` (galaxy.model.History method), 129

`unhide_datasets()` (galaxy.webapps.galaxy.controllers.history.HistoryController.parameters.basic), 196

`unicodify()` (in module galaxy.util), 216

`uninstall_tool_dependencies()` (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303

`unique_id()` (in module galaxy.util), 216

`UniqueValueFilter` (class in galaxy.tools.parameters.dynamic_options), 199

`UniverseApplication` (class in galaxy.webapps.demo_sequencer.app), 256

`UniverseApplication` (class in galaxy.webapps.reports.app), 338

`unknown_base` (galaxy_utils.sequence.transform.ColorSpaceConverter attribute), 354

`unknown_color` (galaxy_utils.sequence.transform.ColorSpaceConverter attribute), 354

`unknown_starttag()` (galaxy.webapps.galaxy.controllers.workflow.SingleTagContentsParser method), 335

`UnknownContentsType`, 100

`UnknownExtra`, 362

`unpack_archive()` (in module galaxy.tools.imp_exp.unpack_tar_gz_archive), 184

`unpack_if_needed()` (galaxy.eggs.dist.DistScrambleEgg method), 96

`unpack_if_needed()` (galaxy.eggs.Egg method), 95

`unpack_source()` (galaxy.eggs.scramble.ScrambleEgg method), 97

`unpack_tar()` (galaxy.eggs.scramble.ScrambleEgg method), 97

`unpack_zip()` (galaxy.eggs.scramble.ScrambleEgg method), 97

`unpack_zipfile()` (in module galaxy.eggs), 96

`unset_item_extended_metadata_obj()` (galaxy.web.base.controller.UsesExtendedMetadataMixin method), 236

`unset_quota_default()` (galaxy.webapps.galaxy.controllers.admin.AdminIndex method), 296

`unsniffable_binary_formats` (galaxy.datatypes.binary.Binary attribute), 44

`UnspecifiedBuildValidator` (class in galaxy.tools.parameters.validation), 209

`unstructured_path_rewriter()` (galaxy.jobs.ComputeEnvironment method), 106

`unstructured_path_rewriter()` (galaxy.jobs.SimpleComputeEnvironment method), 110

`UnvalidatedValue` (class in galaxy.datatypes.metadata.MetadataElementSpec method), 68

`unwrap()` (galaxy.datatypes.metadata.MetadataParameter method), 69

`up_to_date` (galaxy.model.JobExportHistoryArchive attribute), 133

`update()` (galaxy.managers.collections.DatasetCollectionManager method), 150

`update()` (galaxy.managers.libraries.LibraryManager method), 155

`update()` (galaxy.model.WorkflowInvocation method), 144

`update()` (galaxy.model.WorkflowInvocationStep method), 144

`update()` (galaxy.util.odict.odict method), 223

`update()` (galaxy.util.Params method), 213

`update()` (galaxy.webapps.galaxy.api.folder_contents.FolderContentsController method), 13, 268

`update()` (galaxy.webapps.galaxy.api.group_roles.GroupRolesAPIController method), 14, 269

`update()` (galaxy.webapps.galaxy.api.group_users.GroupUsersAPIController method), 14, 270

`update()` (galaxy.webapps.galaxy.api.groups.GroupAPIController method), 14, 270

`update()` (galaxy.webapps.galaxy.api.histories.HistoriesController method), 16, 272

`update()` (galaxy.webapps.galaxy.api.history_contents.HistoryContentsController method), 19, 274

`update()` (galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController method), 19, 275

`update()` (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 27, 282

`update()` (galaxy.webapps.galaxy.api.library_contents.LibraryContentsController method), 29, 284

`update()` (galaxy.webapps.galaxy.api.quotas.QuotaAPIController method), 32, 287

`update()` (galaxy.webapps.galaxy.api.requests.RequestsAPIController method), 32, 288

`update()` (galaxy.webapps.galaxy.api.samples.SamplesAPIController method), 33, 288

`update()` (galaxy.webapps.galaxy.api.users.UserAPIController method), 37, 292

`update()` (galaxy.webapps.galaxy.api.visualizations.VisualizationsController method), 37, 292

`update_external_service_form_definition()` (galaxy.webapps.galaxy.controllers.external_service.ExternalServiceForm method), 306

<code>update_from_file()</code> (galaxy.objectstore.DiskObjectStore method), 157	<code>url_to_destination()</code> (galaxy.jobs.handler.DefaultJobDispatcher method), 111
<code>update_from_file()</code> (galaxy.objectstore.NestedObjectStore method), 158	<code>url_to_destination()</code> (galaxy.jobs.runners.BaseJobRunner method), 118
<code>update_from_file()</code> (galaxy.objectstore.ObjectStore method), 159	<code>url_to_destination()</code> (galaxy.jobs.runners.cli.ShellJobRunner method), 119
<code>update_invocation_step()</code> (galaxy.managers.workflows.WorkflowsManager method), 156	<code>url_to_destination()</code> (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 120
<code>update_permissions()</code> (galaxy.webapps.galaxy.api.lda_datasets.LibraryDatasetController method), 24, 279	<code>url_to_destination()</code> (galaxy.jobs.runners.lwr.LwrJobRunner method), 120
<code>update_request_state()</code> (galaxy.webapps.galaxy.controllers.requests_controller.RequestsController method), 324	<code>url_to_file()</code> (in module <code>galaxy.util.infection.BaseMethod</code>), 219
<code>update_runtime_state()</code> (galaxy.workflow.modules.SimpleWorkflowModule method), 347	<code>urlize()</code> (galaxy.util.infection.Infecter method), 220
<code>update_sample_dataset_status()</code> (galaxy.webapps.galaxy.controllers.requests_admin_toolshed.AdminToolshed method), 323	<code>URLRetriever</code> (class in galaxy.eggs), 95
<code>update_sample_state()</code> (galaxy.webapps.galaxy.controllers.requests_admin_toolshed.AdminToolshed method), 324	<code>use_async</code> (galaxy.web.framework.helpers.grids.Grid attribute), 248
<code>update_state()</code> (galaxy.tools.Tool method), 175	<code>use_async</code> (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid attribute), 309
<code>update_state()</code> (galaxy.workflow.modules.SimpleWorkflowModule method), 347	<code>use_async</code> (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 312
<code>update_state()</code> (galaxy.workflow.modules.ToolModule method), 347	<code>use_async</code> (galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid attribute), 317
<code>update_state()</code> (galaxy.workflow.modules.WorkflowModule method), 349	<code>use_async</code> (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid attribute), 317
<code>update_to_changeset_revision()</code> (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303	<code>use_async</code> (galaxy.webapps.galaxy.controllers.visualization.HistoryDatasets attribute), 331
<code>update_tool_shed_status_for_installed_repository()</code> (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303	<code>use_async</code> (galaxy.webapps.galaxy.controllers.visualization.HistorySelection attribute), 332
<code>update_type_values</code> (galaxy.webapps.galaxy.api.samples.SamplesAPIController attribute), 33, 288	<code>use_async</code> (galaxy.webapps.galaxy.controllers.visualization.LibrarySelection attribute), 332
<code>update_types</code> (galaxy.webapps.galaxy.api.samples.SamplesAPIController attribute), 33, 288	<code>use_async</code> (galaxy.webapps.galaxy.controllers.visualization.TracksterSelect attribute), 333
<code>upload_async_create()</code> (galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner method), 327	<code>use_async</code> (galaxy.webapps.galaxy.controllers.visualization.VisualizationAll attribute), 333
<code>upload_async_message()</code> (galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner method), 327	<code>use_async</code> (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAll attribute), 335
<code>upload_dataset()</code> (galaxy.webapps.galaxy.controllers.library_upload(LibraryUpload method), 315	<code>use_async</code> (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute), 341
<code>upload_library_dataset()</code> (galaxy.webapps.galaxy.controllers.library_upload(LibraryUpload method), 315	<code>use_async</code> (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid attribute), 342
<code>upload_options</code> (galaxy.model.LibraryDataset attribute), 135	<code>use_async</code> (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid attribute), 344
<code>UploadDataset</code> (class in galaxy.tools.parameters.grouping), 200	<code>use_hide_message</code> (galaxy.web.framework.helpers.grids.Grid attribute), 248
<code>UploadToolAction</code> (class in galaxy.tools.actions.upload), 180	<code>use_paging</code> (galaxy.web.framework.helpers.grids.Grid attribute), 248
<code>url</code> (galaxy.datatypes.display_applications.parameters.DisplayParametersGalaxyWebapps attribute), 92	<code>use_paging</code> (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 296
	<code>use_paging</code> (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 298

- 300
- UserListGroup.Column (class in galaxy.webapps.galaxy.controllers.admin), 300
- UserListGroup.LastLoginColumn (class in galaxy.webapps.galaxy.controllers.admin), 300
- UserListGroup.RolesColumn (class in galaxy.webapps.galaxy.controllers.admin), 300
- UserListGroup.StatusColumn (class in galaxy.webapps.galaxy.controllers.admin), 300
- UserListGroup.TimeCreatedColumn (class in galaxy.webapps.galaxy.controllers.admin), 300
- UserListGroup.UserNameColumn (class in galaxy.webapps.galaxy.controllers.admin), 300
- UserOpenID (class in galaxy.model), 142
- UserOpenIDGrid (class in galaxy.webapps.galaxy.controllers.user), 329
- UserPreference (class in galaxy.model), 142
- UserQuotaAssociation (class in galaxy.model), 142
- UserRequestsGrid (class in galaxy.webapps.galaxy.controllers.requests), 321
- UserRoleAssociation (class in galaxy.model), 142
- Users (class in galaxy.webapps.reports.controllers.users), 343
- users() (galaxy.web.base.controllers.admin.Admin method), 241
- uses_tool_shed_dependencies() (galaxy.tools.deps.DependencyManager method), 183
- uses_tool_shed_dependencies() (galaxy.tools.deps.NullDependencyManager method), 183
- UsesAnnotations (class in galaxy.model.item_attrs), 146
- UsesExtendedMetadataMixin (class in galaxy.web.base.controller), 236
- UsesFormDefinitionsMixin (class in galaxy.web.base.controller), 236
- UsesHistoryDatasetAssociationMixin (class in galaxy.web.base.controller), 237
- UsesHistoryMixin (class in galaxy.web.base.controller), 237
- UsesItemRatings (class in galaxy.model.item_attrs), 146
- UsesLibraryMixin (class in galaxy.web.base.controller), 238
- UsesLibraryMixinItems (class in galaxy.web.base.controller), 238
- UsesQuotaMixin (class in galaxy.web.base.controller), 238
- UsesStoredWorkflowMixin (class in galaxy.web.base.controller), 238
- UsesTagsMixin (class in galaxy.web.base.controller), 239
- UsesVisualizationMixin (class in galaxy.web.base.controller), 239
- UUIDType (class in galaxy.model.custom_types), 145
- ## V
- v (galaxy.webapps.galaxy.api.requests.RequestsApiController attribute), 32, 288
- v (galaxy.webapps.galaxy.api.samples.SamplesApiController attribute), 33, 288
- VALID_FORMATS (galaxy_utils.sequence.fastq.fastqAggregator attribute), 350
- valid_gff3_phase (galaxy.datatypes.interval.Gff3 attribute), 64
- valid_gff3_strand (galaxy.datatypes.interval.Gff3 attribute), 64
- valid_operations (galaxy.model.Quota attribute), 137
- VALID_PRESET (galaxy.tools.parameters.sanitize.ToolParameterSanitizer attribute), 205
- valid_sequence_list (galaxy_utils.sequence.fastq.fastqCSSangerRead attribute), 351
- valid_sequence_list (galaxy_utils.sequence.sequence.SequencingRead attribute), 354
- validate() (galaxy.datatypes.data.Data method), 49
- validate() (galaxy.datatypes.genetics.GenomeGraphs method), 52
- validate() (galaxy.datatypes.interval.Interval method), 65
- validate() (galaxy.datatypes.metadata.MetadataParameter method), 69
- validate() (galaxy.model.DatasetCollection method), 123
- validate() (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 187
- validate() (galaxy.tools.parameters.basic.DataToolParameter method), 188
- validate() (galaxy.tools.parameters.basic.ToolParameter method), 196
- validate() (galaxy.tools.parameters.validation.DatasetOkValidator method), 206
- validate() (galaxy.tools.parameters.validation.EmptyTextfieldValidator method), 206
- validate() (galaxy.tools.parameters.validation.ExpressionValidator method), 206
- validate() (galaxy.tools.parameters.validation.InRangeValidator method), 207
- validate() (galaxy.tools.parameters.validation.LengthValidator method), 207
- validate() (galaxy.tools.parameters.validation.MetadataInDataTableColumn method), 208
- validate() (galaxy.tools.parameters.validation.MetadataInFileColumnValidator method), 208

[validate\(\) \(galaxy.tools.parameters.validation.MetadataValidator method\), 208](#)
[validate\(\) \(galaxy.tools.parameters.validation.NoOptionsValidator method\), 208](#)
[validate\(\) \(galaxy.tools.parameters.validation.RegexValidator method\), 209](#)
[validate\(\) \(galaxy.tools.parameters.validation.UnspecifiedBuilderValidator method\), 209](#)
[validate\(\) \(galaxy.tools.parameters.validation.Validator method\), 209](#)
[validate_and_sanitize_basestring\(\) \(galaxy.web.base.controller.BaseController method\), 235](#)
[validate_and_sanitize_basestring_list\(\) \(galaxy.web.base.controller.BaseController method\), 235](#)
[validate_boolean\(\) \(galaxy.web.base.controller.BaseController method\), 235](#)
[validate_email\(\) \(in module galaxy.security.validate_user_input\), 168](#)
[validate_in_users_and_groups\(\) \(galaxy.web.base.controller.BaseApiController method\), 234](#)
[validate_input_element_identifiers\(\) \(in module galaxy.managers.collections_util\), 150](#)
[validate_jsonrpc_request\(\) \(in module galaxy.util.json\), 222](#)
[validate_jsonrpc_response\(\) \(in module galaxy.util.json\), 222](#)
[validate_password\(\) \(in module galaxy.security.validate_user_input\), 168](#)
[validate_publicname\(\) \(in module galaxy.security.validate_user_input\), 168](#)
[ValidationError \(class in galaxy.model\), 143](#)
[Validator \(class in galaxy.tools.parameters.validation\), 209](#)
[value \(galaxy_utils.sequence.transform.ColorSpaceConverter attribute\), 354](#)
[value_from_basic\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 195](#)
[value_from_basic\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 196](#)
[value_from_basic\(\) \(galaxy.tools.parameters.grouping.ConditionalGroup method\), 199](#)
[value_from_basic\(\) \(galaxy.tools.parameters.grouping.Group method\), 200](#)
[value_from_basic\(\) \(galaxy.tools.parameters.grouping.Repeat method\), 200](#)
[value_from_basic\(\) \(galaxy.tools.parameters.grouping.UploadDataset method\), 200](#)
[value_to_basic\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 195](#)
[value_to_basic\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 196](#)
[value_to_basic\(\) \(galaxy.tools.parameters.grouping.ConditionalGroup method\), 199](#)
[value_to_basic\(\) \(galaxy.tools.parameters.grouping.Group method\), 200](#)
[value_to_basic\(\) \(galaxy.tools.parameters.grouping.Repeat method\), 200](#)
[value_to_basic\(\) \(galaxy.tools.parameters.grouping.UploadDataset method\), 200](#)
[value_to_display_text\(\) \(galaxy.tools.parameters.basic.DataCollectionToolParameter method\), 187](#)
[value_to_display_text\(\) \(galaxy.tools.parameters.basic.DataToolParameter method\), 188](#)
[value_to_display_text\(\) \(galaxy.tools.parameters.basic.DrillDownSelectToolParameter method\), 190](#)
[value_to_display_text\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 195](#)
[value_to_display_text\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 196](#)
[ValueExternalServiceActionsGroupWhen \(class in galaxy.external_services.service\), 103](#)
[values\(\) \(galaxy.util.bunch.Bunch method\), 217](#)
[values\(\) \(galaxy.util.odict.odict method\), 223](#)
[ValueToolOutputActionConditionalWhen \(class in galaxy.tools.parameters.output\), 204](#)
[variablize\(\) \(galaxy.util.inflection.Base method\), 219](#)
[variablize\(\) \(galaxy.util.inflection.Inflector method\), 221](#)
[VariantCall \(class in galaxy_utils.sequence.vcf\), 355](#)
[VariantCall33 \(class in galaxy_utils.sequence.vcf\), 355](#)
[VariantCall40 \(class in galaxy_utils.sequence.vcf\), 355](#)
[VariantCall41 \(class in galaxy_utils.sequence.vcf\), 355](#)
[Vcf \(class in galaxy.datatypes.tabular\), 84](#)
[Velvet \(class in galaxy.datatypes.assembly\), 42](#)
[verbose_name \(galaxy.jobs.actions.post.ChangeDatatypeAction attribute\), 114](#)
[verbose_name \(galaxy.jobs.actions.post.ColumnSetAction attribute\), 114](#)
[verbose_name \(galaxy.jobs.actions.post.DefaultJobAction attribute\), 115](#)
[verbose_name \(galaxy.jobs.actions.post.DeleteDatasetAction attribute\), 115](#)
[verbose_name \(galaxy.jobs.actions.post.DeleteIntermediatesAction attribute\), 115](#)
[verbose_name \(galaxy.jobs.actions.post.EmailAction attribute\), 115](#)
[verbose_name \(galaxy.jobs.actions.post.HideDatasetAction attribute\), 115](#)
[verbose_name \(galaxy.jobs.actions.post.RenameDatasetAction attribute\), 115](#)
[Variant \(class in galaxy_utils.sequence.vcf.VariantCall attribute\), 355](#)
[Variant33 \(class in galaxy_utils.sequence.vcf.VariantCall33 attribute\), 355](#)
[Variant40 \(class in galaxy_utils.sequence.vcf.VariantCall40 attribute\), 355](#)

version (galaxy_utils.sequence.vcf.VariantCall41 attribute), 355
 version (pkg_resources.Distribution attribute), 361
 version_conflict() (galaxy.eggs.Egg method), 95
 version_path() (galaxy.jobs.ComputeEnvironment method), 106
 version_path() (galaxy.jobs.SharedComputeEnvironment method), 110
 VersionConflict, 361
 view() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 311
 view_datatypes_registry() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 296
 view_editable_request_type() (galaxy.webapps.galaxy.controllers.request_type.RequestType method), 320
 view_external_service() (galaxy.webapps.galaxy.controllers.request_type.RequestType method), 306
 view_form_definition() (galaxy.webapps.galaxy.controllers.request_type.RequestType method), 320
 view_latest_form_definition() (galaxy.webapps.galaxy.controllers.forms.Forms method), 308
 view_multiple() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 311
 view_request() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 324
 view_request_history() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 324
 view_request_type() (galaxy.webapps.galaxy.controllers.request_type.RequestType method), 320
 view_sample() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 324
 view_sample_datasets() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 324
 view_sample_history() (galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon method), 324
 view_tool_data_tables() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 296
 view_tool_metadata() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303
 view_workflow() (galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed method), 303
 visible (galaxy.tools.parameters.basic.FTPFileToolParameter attribute), 190
 visible (galaxy.tools.parameters.basic.ToolParameter attribute), 196
 visible (galaxy.tools.parameters.grouping.Group attribute), 200
 visit_input_values() (in module galaxy.tools.parameters), 185
 visit_inputs() (galaxy.tools.parameters.grouping.Conditional method), 199
 visit_inputs() (galaxy.tools.parameters.grouping.Repeat method), 200
 visit_inputs() (galaxy.tools.parameters.grouping.UploadDataset method), 201
 visit_inputs() (galaxy.tools.Tool method), 175
 Visualization (class in galaxy.model), 143
 VisualizationAllPublishedGrid (class in galaxy.webapps.galaxy.controllers.visualization), 333
 VisualizationAnnotationAssociation (class in galaxy.model), 143
 VisualizationController (class in galaxy.webapps.galaxy.controllers.visualization), 333
 VisualizationListGrid (class in galaxy.webapps.galaxy.controllers.visualization), 335
 VisualizationRatingAssociation (class in galaxy.model), 143
 VisualizationRequestType (class in galaxy.model), 143
 VisualizationsController (class in galaxy.webapps.galaxy.api.visualizations), 37, 292
 VisualizationSelectionGrid (class in galaxy.webapps.galaxy.controllers.page), 319
 VisualizationTagsAssociation (class in galaxy.model), 143
 VisualizationUserShareAssociation (class in galaxy.model), 143
 viz_types (galaxy.web.base.controller.UsesVisualizationMixin attribute), 340
W
 wake() (galaxy.jobs.deferred.Sleeper method), 116
 wake() (galaxy.jobs.queue_transfer_manager.Sleeper method), 113
 WebApplication (class in galaxy.web.framework.base), 212
 WebApplication (class in galaxy.webapps.django_sequencer.framework), 258
 welcome() (galaxy.webapps.galaxy.controllers.root.RootController method), 326
 whoosh_search() (in module galaxy.webapps.galaxy.controllers.library_common), 316
 widget_fields_have_contents() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 237
 Wiggle (class in galaxy.datatypes.interval), 65
 wiggle_dataprovider() (galaxy.datatypes.interval.Wiggle method), 66

wiggle_dict_dataprovider() (galaxy.datatypes.interval.Wiggle method), 66	WorkflowsManager (class in galaxy.managers.workflows), 156
Workflow (class in galaxy.model), 143	WorkflowStep (class in galaxy.model), 144
workflow_dict() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 39, 294	WorkflowStepAnnotationAssociation (class in galaxy.model), 145
workflow_invocation_step() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 39, 294	WorkflowStepConnection (class in galaxy.model), 145
workflow_invocation_step_update() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 39, 294	WorkflowStepTagAssociation (class in galaxy.model), 145
workflow_request() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 39, 294	WorkflowTagsController (class in galaxy.webapps.galaxy.api.item_tags), 20,
workflow_usage() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 39, 295	working_directory() (galaxy.jobs.ComputeEnvironment APIController method), 39, 294
workflow_usage_contents() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 39, 295	working_directory() (galaxy.jobs.SharedComputeEnvironment APIController method), 39, 295
WorkflowAnnotationsController (class in galaxy.webapps.galaxy.api.annotations), 264	WorkingSet (class in pkg_resources), 358
WorkflowController (class in galaxy.webapps.galaxy.controllers.workflow), 336	WorkRequestTagAssociation (class in galaxy.model), 143
WorkflowField (class in galaxy.web.form_builder), 233	wrap() (galaxy.datatypes.metadata.MetadataElementSpec method), 68
WorkflowInvocation (class in galaxy.model), 143	wrap() (galaxy.datatypes.metadata.MetadataParameter method), 69
WorkflowInvocationStep (class in galaxy.model), 144	wrap() (galaxy.datatypes.metadata.SelectParameter method), 69
WorkflowMappingField (class in galaxy.web.form_builder), 233	wrap_in_middleware() (in module galaxy.webapps.demo_sequencer.buildapp), 256
WorkflowModule (class in galaxy.workflow.modules), 348	wrap_in_middleware() (in module galaxy.webapps.galaxy.buildapp), 259
WorkflowModuleFactory (class in galaxy.workflow.modules), 349	wrap_in_middleware() (in module galaxy.webapps.reports.buildapp), 338
WorkflowModuleInjector (class in galaxy.workflow.modules), 349	wrap_in_static() (in module galaxy.webapps.demo_sequencer.buildapp), 256
WorkflowOutput (class in galaxy.model), 144	wrap_in_static() (in module galaxy.webapps.galaxy.buildapp), 259
WorkflowRequest (class in galaxy.model), 144	wrap_in_static() (in module galaxy.webapps.reports.buildapp), 338
WorkflowRequestInputParameter (class in galaxy.model), 144	writable_files (galaxy.datatypes.data.Data attribute), 49
WorkflowRequestStepState (class in galaxy.model), 144	write() (galaxy.datatypes.converters.interval_to_coverage.CoverageWriter method), 88
WorkflowRequestToInputDatasetAssociation (class in galaxy.model), 144	write() (galaxy_utils.sequence.fasta.fastaWriter method), 350
WorkflowRequestToInputDatasetCollectionAssociation (class in galaxy.model), 144	write() (galaxy_utils.sequence.fastq.fastqWriter method), 354
Workflows (class in galaxy.webapps.reports.controllers.workflows), 344	write_data_to_file() (galaxy.visualization.data_providers.basic.BaseDataProvider method), 226
WorkflowsAPIController (class in galaxy.webapps.galaxy.api.workflows), 292	write_from_stream() (galaxy.datatypes.data.Data method), 49
WorkflowSelectionGrid (class in galaxy.webapps.galaxy.controllers.page), 319	write_from_stream() (galaxy.datatypes.data.Text method), 51
	write_from_stream() (galaxy.model.DatasetInstance method), 126

[write_integrated_tool_panel_config_file\(\)](#)
 (galaxy.tools.ToolBox method), [176](#)
[write_log\(\)](#) (galaxy.web.framework.middleware.translogger.TransLogger
 method), [253](#)
[write_split_files\(\)](#) (galaxy.datatypes.sequence.Sequence
 class method), [77](#)
[wsgi_headeritems\(\)](#) (galaxy.web.framework.base.Response
 method), [243](#)
[wsgi_status\(\)](#) (galaxy.web.framework.base.Response
 method), [243](#)
[WSGIEnvironmentProperty](#) (class in
 galaxy.web.framework.base), [243](#)

X

[Xbm](#) (class in galaxy.datatypes.images), [59](#)
[XForwardedHostMiddleware](#) (class in
 galaxy.web.framework.middleware.xforwardedhost),
[253](#)
[Xlsx](#) (class in galaxy.datatypes.binary), [46](#)
[xml_dataprovider\(\)](#) (galaxy.datatypes.xml.GenericXml
 method), [85](#)
[xml_element_compare\(\)](#) (in module galaxy.util), [217](#)
[xml_element_list_compare\(\)](#) (in module galaxy.util), [217](#)
[xml_element_to_dict\(\)](#) (in module galaxy.util), [217](#)
[xml_string](#) (galaxy.tools.data.TabularToolDataTable at-
 tribute), [182](#)
[xml_text\(\)](#) (in module galaxy.util), [217](#)
[xml_to_string\(\)](#) (in module galaxy.util), [217](#)
[Xpm](#) (class in galaxy.datatypes.images), [59](#)

Y

[yield_lines\(\)](#) (in module pkg_resources), [363](#)

Z

[ZipBall](#) (class in galaxy.util.streamball), [223](#)
[ZipProvider](#) (class in pkg_resources), [365](#)