MyTardis Documentation

Release 3.8

Steve Androulakis

Jun 28, 2018

Contents

1 Overview **3**

2 Key features for users **5**

3 Key features for instrument facilities **7**

4 Developing for MyTardis **9**

5 Find out more **11**

6 Known deployments **13**

7 Related projects and repositories **15**

8 Releases **17**

9 Reporting Bugs **19**

10 Contributing **21**

11 Documentation **23**
- 11.1 User Guide . . . 23
- 11.2 Configuration and Administration . . . 37
- 11.3 Development . . . 55
- 11.4 Documentation for included Apps . . . 177
- 11.5 Releases . . . 186

12 Indices and tables **191**

Python Module Index **193**

CHAPTER 1

Overview

MyTardis began at Monash University to solve the problem of users needing to store large datasets and share them with collaborators online. Its particular focus is on integration with scientific instruments, instrument facilities and research storage and computing infrastructure; to address the challenges of data storage, data access, collaboration and data publication.

Read more...

CHAPTER 2

Key features for users

The MyTardis data management platform is a software solution that manages research data and the associated metadata. MyTardis handles the underlying storage to ensure that data is securely archived and provides access to the data through a web portal. Data hosted in MyTardis can also be accessed via SFTP.

Read more...

CHAPTER 3

Key features for instrument facilities

MyTardis takes care of distributing data to your users. Its instrument integration technology takes care of securely and automatically shifting data from instrument to repository and makes it accessible by the right users.

Read more. . .

CHAPTER 4

Developing for MyTardis

MyTardis is mostly written in the Python programming language and is built on top of the Django web framework. A complete installation of the service also includes an Elasticsearch index, a RabbitMQ-based task queue, an Nginx server, and a PostgreSQL database.

To set up and manage these services we employ the SaltStack orchestration software and cloud technologies.

Read more...

CHAPTER 5

Find out more

Project homepage http://mytardis.org

The source code is hosted at https://github.com/mytardis/mytardis

Documentation at http://mytardis.readthedocs.org includes

- User documentation
- Administrator documentation
- Developer documentation

The wiki at https://github.com/mytardis/mytardis/wiki includes

- Links to MyTardis add-ons, including apps and post-save filters
- Information about MyTardis community events

CHAPTER 6

Known deployments

- **Store.Synchrotron**: https://store.synchrotron.org.au/
- **Store.Monash**: https://store.erc.monash.edu
- **NIF ImageTrove**: https://imagetrove.cai.uq.edu.au/
- **MHTP Medical Genomics**: https://mhtp-seq.erc.monash.edu/
- **ANSTO**: https://tardis.nbi.ansto.gov.au/
- **Monash MyTardis Demo**: https://mytardisdemo.erc.monash.edu/

CHAPTER 7

Related projects and repositories

- **MyData**: https://github.com/mytardis/mydata
  - A desktop application for managing uploads to MyTardis
- **NIF ImageTrove**: https://github.com/NIF-au/imagetrove
  - A tool for ingesting and archiving NIF datasets, including
    - Web front end: MyTardis
    - A DICOM server: DICOM ToolKit
    - A dataset uploader: imagetrove-uploader
    - Federated authentication with the Australian Access Federation's Rapid Connect service
- **NIF ImageTrove Docker deployment**: https://github.com/UWA-FoS/docker-mytardis
  - For ease of deployment, all of ImageTrove's components are packaged into a Docker container.
- **NIF Trusted Data Repositories**: https://github.com/NIF-au/TDR
  - Delivering durable, reliable, high-quality image data

CHAPTER 8

Releases

The default branch on GitHub is `develop`. This is the cutting edge development version. Please DO NOT use this in production, as it may have bugs that eat your data.

The `master` branch is the current stable release with all the latest bugfixes included. It will move to newer versions automatically. Follow this branch if you want to stay up to date in a production environment.

Each version has its own branch named by version number. At the time of writing this is `3.7`. Follow this branch for your production installation and perform version upgradres manually.

Each bugfix or set of fixes bumps the minor version and each new release is tagged, eg. `3.7.2`. Use tagged releases if you are paranoid about changes to the code you have not tested yourself.

To follow development, please see the contributing section below.

CHAPTER 9

Reporting Bugs

Bug reports and feature requests can be made via our public issue tracker.

CHAPTER 10

Contributing

New contributors are always welcome, however all developers should review the pull-request checklist before making pull requests.

For any wishes, comments, praise etc. either open a GitHub issue or contact us.

Active developers are also welcome to join our Slack team.

Contact details can be found on mytardis.org.

CHAPTER 11

Documentation

11.1 User Guide

11.1.1 Basics

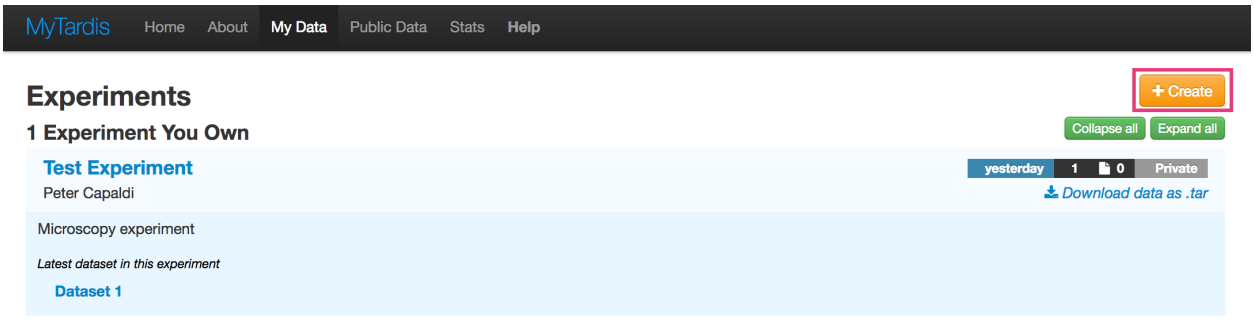
Organising Data

Data Structure Overview

Data in MyTardis is organised into a uniform hierarchy, which includes Experiments, Datasets and Data Files. Experiments are at the top of the hierarchy and can contain multiple Datasets. Similarly, Datasets can contain multiple Data Files. The hierarchy can be thought of like a directory structure on a typical PC (Fig. 1) in which a the Experiment directory contains multiple Dataset directories, which in turn contain multiple Data Files.

Creating Experiments

1. Navigate to the **My Data** page.
2. Click on the **Create** button to the right of the page.



3. Fill in the 'Create Experiment' form. Requires at least a Title and one author.

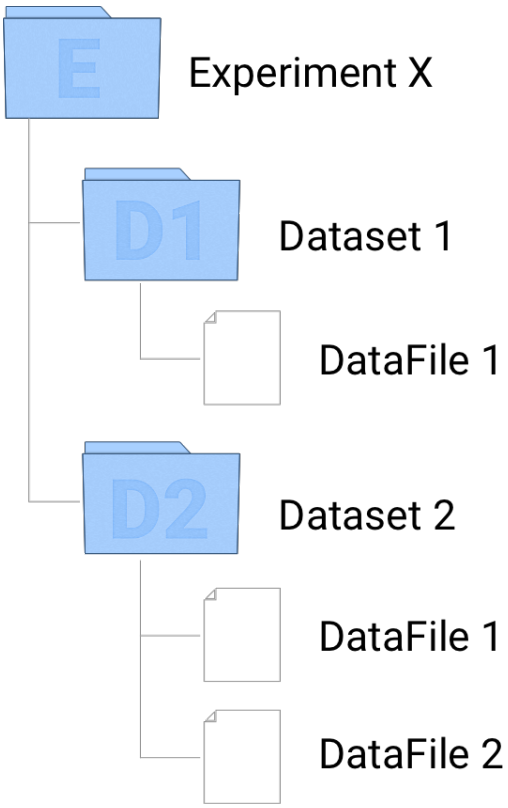
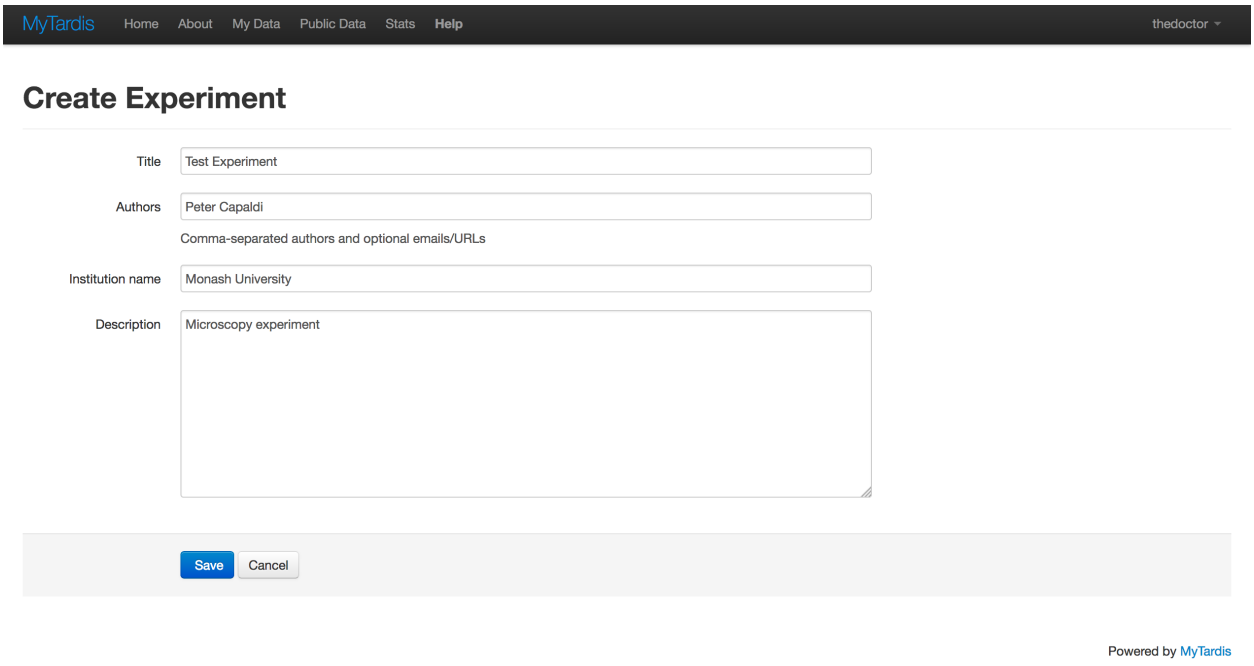


Fig. 1: Fig. 1. Data hierarchy in MyTardis

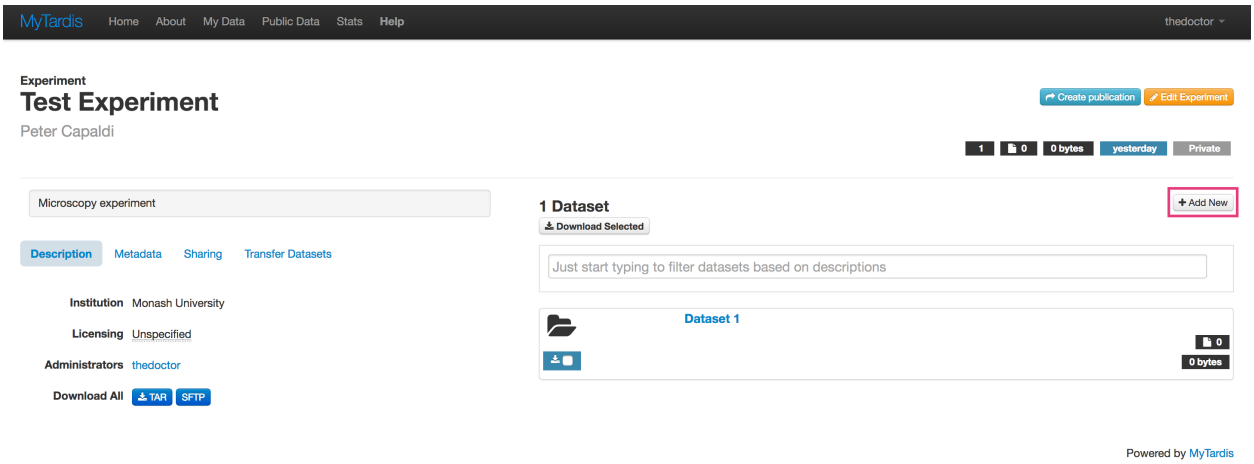


4. Click **Save**.

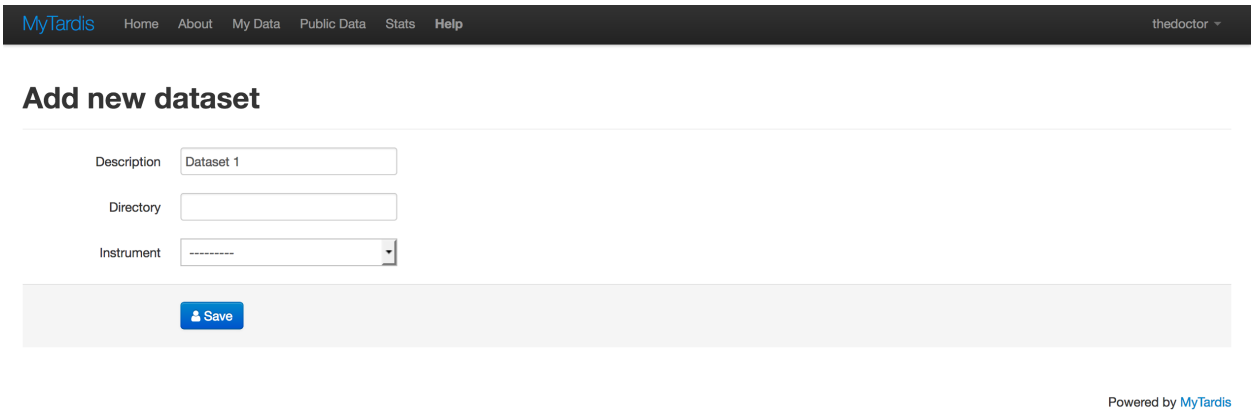
Adding Datasets

Datasets always have at least one parent Experiment. To add a dataset to an Experiment:

1. Navigate to the Experiment page into which you would like to add a dataset.
2. Click the **+ Add New** button to the right of the page.



3. Fill in the 'Add new dataset' form. Note: The 'Description' field is actually the name of the dataset.

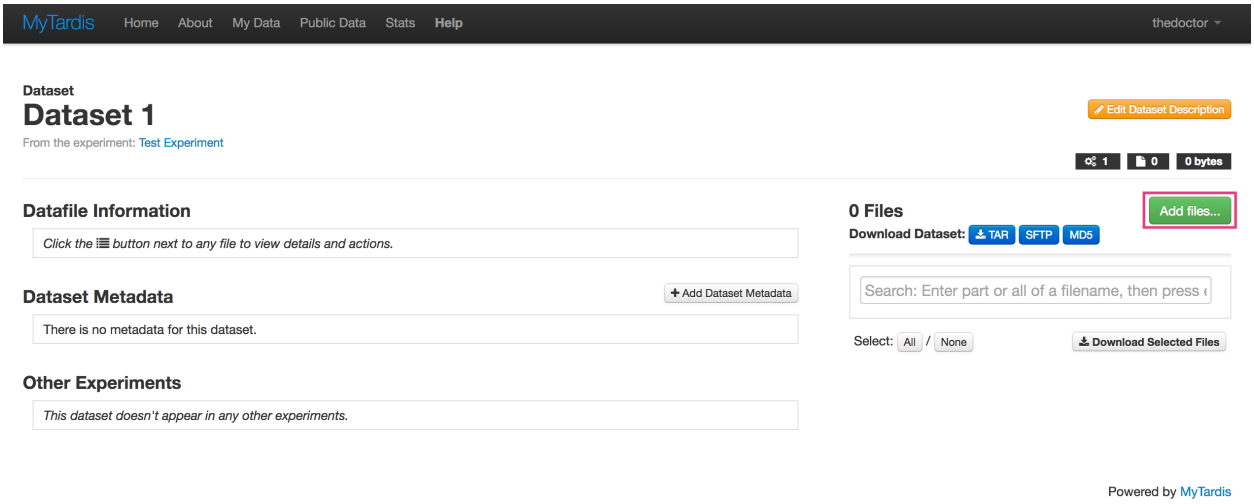


4. Click **Save**.

Adding Data Files

Data Files can be added to a particular Dataset from the Dataset page.

1. Navigate to the Dataset into which you want to add new files.
2. Click the green **Add files...** button to the right of the page.



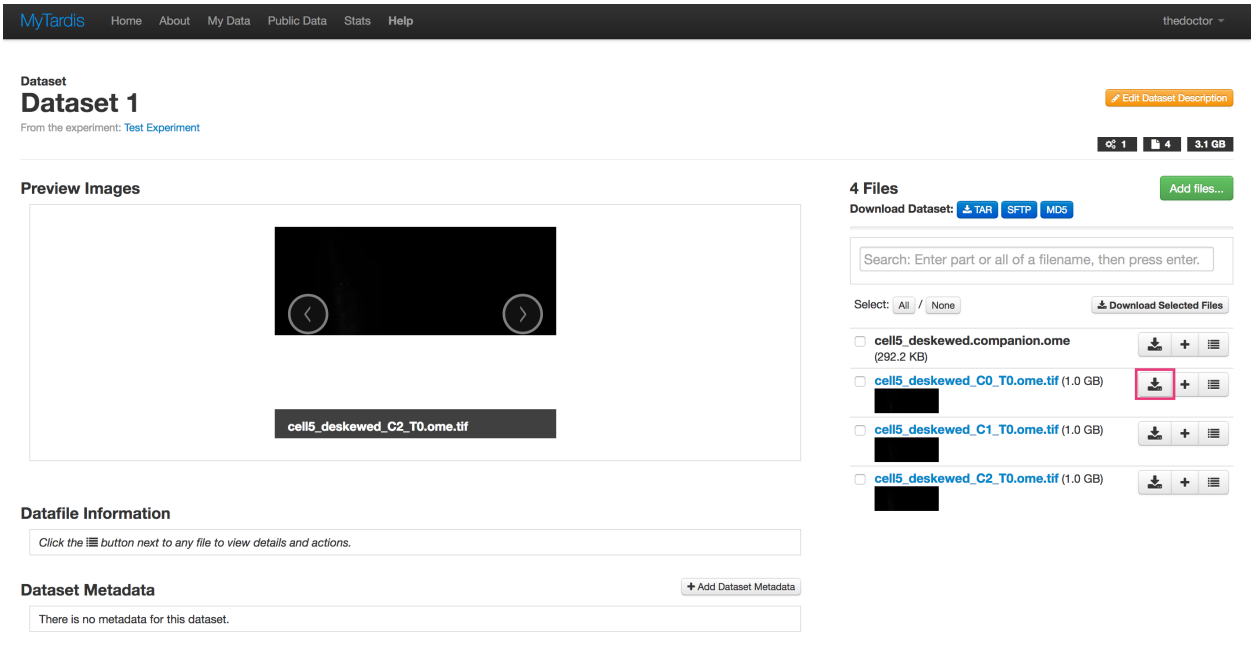
3. Select the files you want to add from the file selector dialog and Click **Open**. You can add multiple files by holding the "Shift" or "Ctrl" keys to select multiple files in the dialog.

Accessing Data

Download a Data File

Individual Data Files can be downloaded as follows:

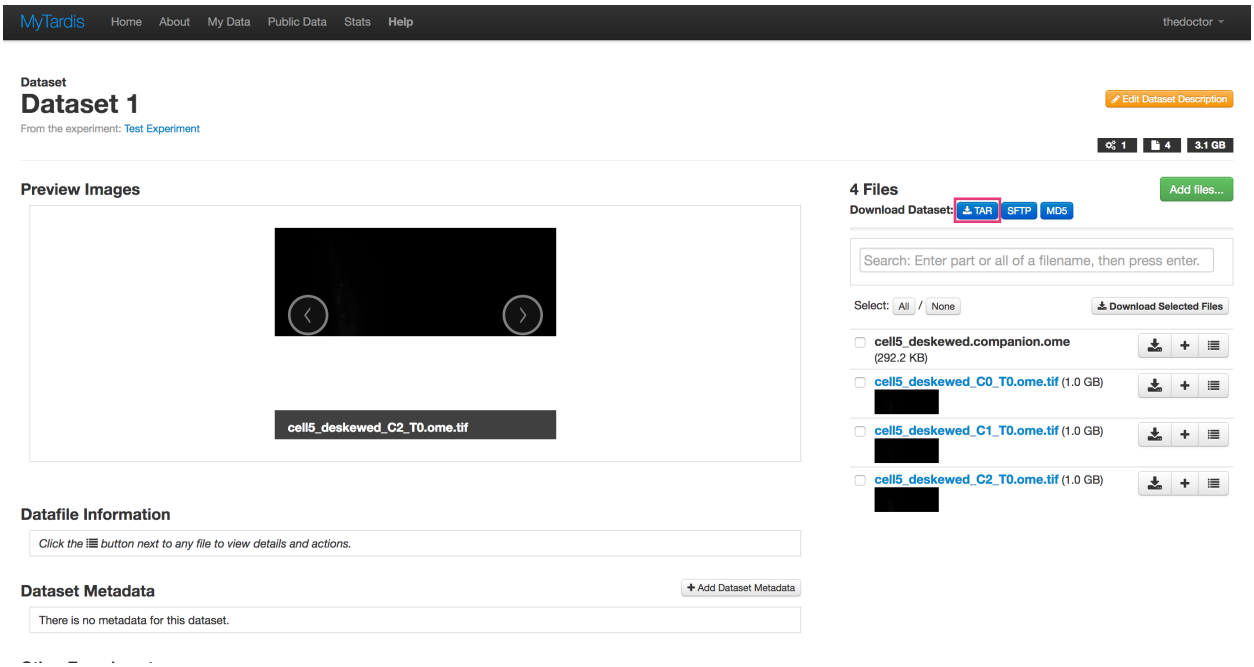
1. Navigate to the Dataset containing the Data File you want to download.
2. Click the **Download** button on the Data File you want to download.



Download a Dataset

Entire Datasets can be downloaded as a tar archive.

1. Navigate to the Dataset you want to download.
2. Click the **tar** button to the right of the page.

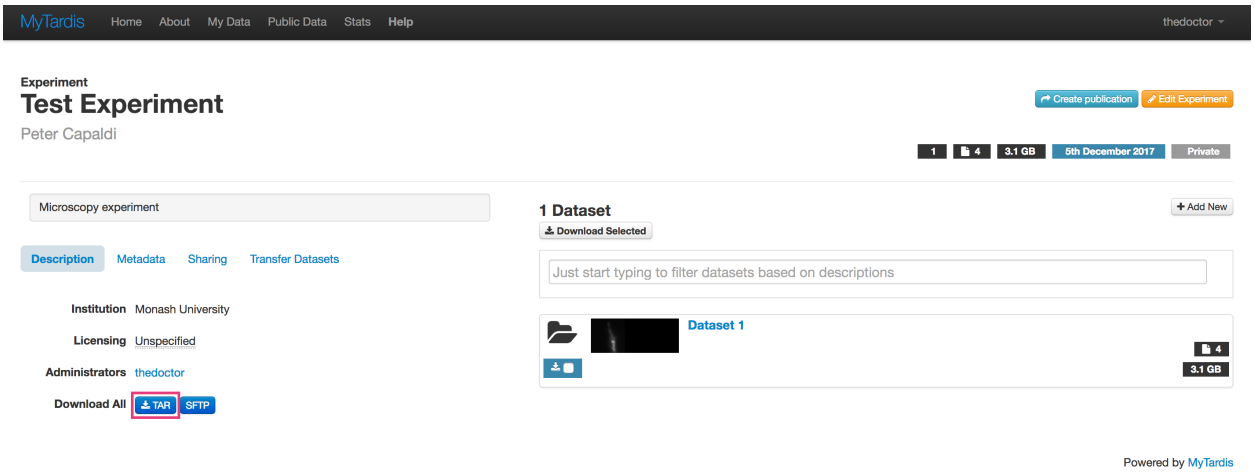


Note: *tar* is an archive format like *zip* that may not be familiar to Windows users and a default Windows installation may not have the appropriate software to expand *tar* archives. We suggest that users install 7-zip to expand *tar* archives from MyTardis.

Download an Experiment

Entire Experiments can be downloaded as a tar archive.

1. Navigate to the Experiment you want to download.
2. Click the **tar** button under the 'Description' section of the page.



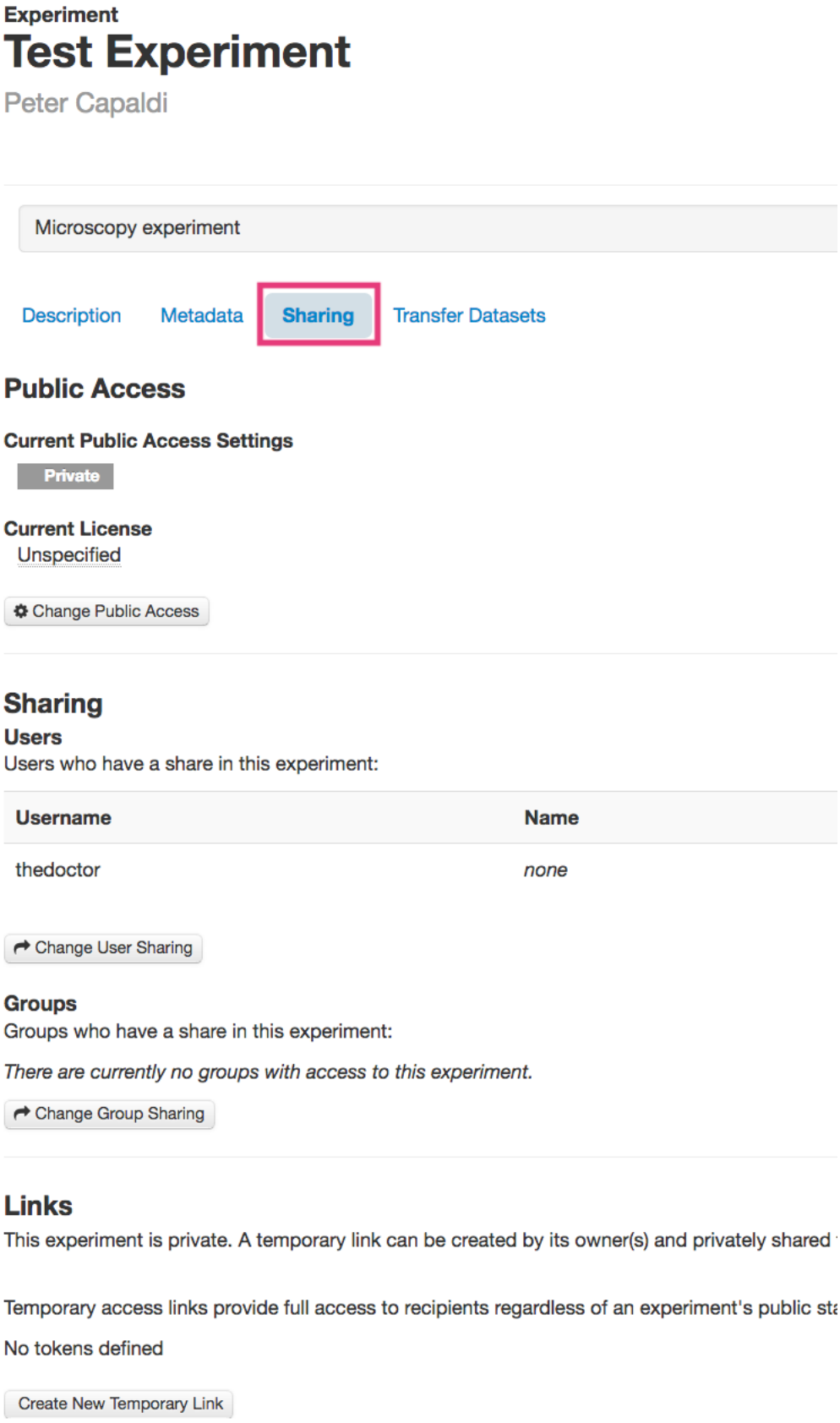
Note: *tar* is an archive format like *zip* that may not be familiar to Windows users and a default Windows installation may not have the appropriate software to expand *tar* archives. We suggest that users install 7-zip to expand *tar* archives from MyTardis.

Sharing and Publishing Data

MyTardis provides 3 primary mechanisms for sharing your data:

- Sharing with another MyTardis user.
- Sharing via a temporary (obfuscated) link.
- Making data public.

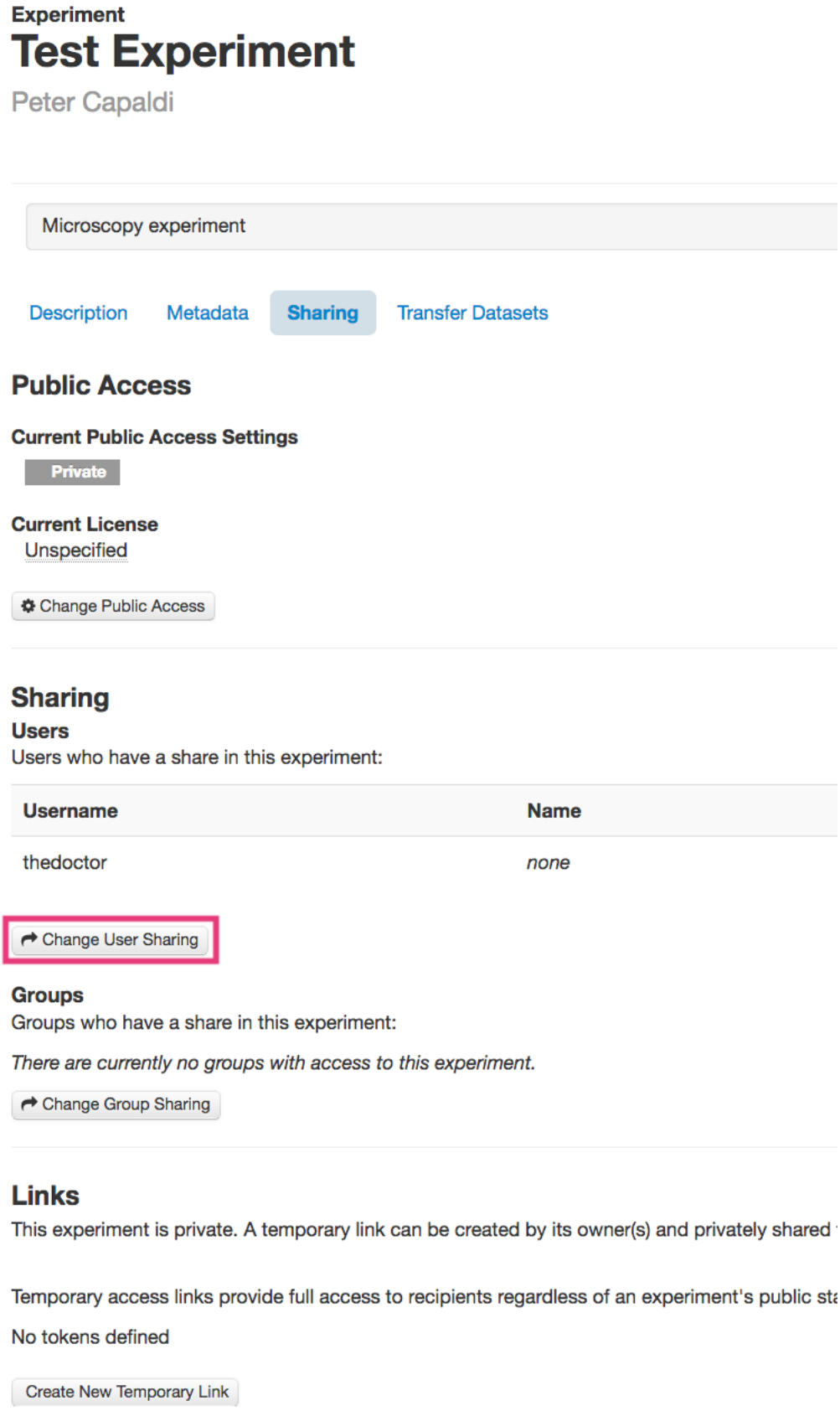
Data sharing is done at the Experiment level and you can find the options to share an Experiment under the **Sharing** tab on the left side of an Experiment page.


Experiment
Test Experiment
Peter Capaldi
Microscopy experiment
Description
Metadata
Sharing
Transfer Datasets
Public Access
Current Public Access Settings
Private
Current License
Unspecified
Change Public Access
Sharing
Users
Users who have a share in this experiment:
Username
Name
thedoctor
none
Change User Sharing
Groups
Groups who have a share in this experiment:
There are currently no groups with access to this experiment.
Change Group Sharing
Links
This experiment is private. A temporary link can be created by its owner(s) and privately shared
Temporary access links provide full access to recipients regardless of an experiment's public st
No tokens defined
Create New Temporary Link

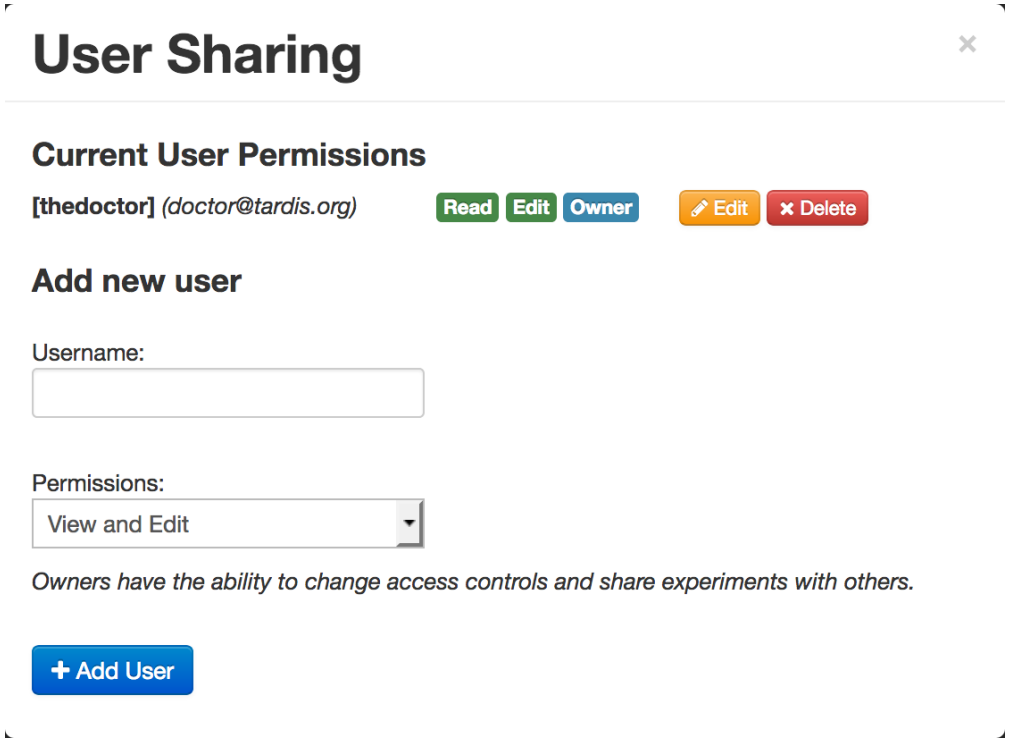
Sharing Data with Another MyTardis User

To share an Experiment with another MyTardis user:

1. Navigate to the Experiment you want to share.
2. Click on the **Sharing** tab on the left side of the page.
3. Click the **Change User Sharing** button.

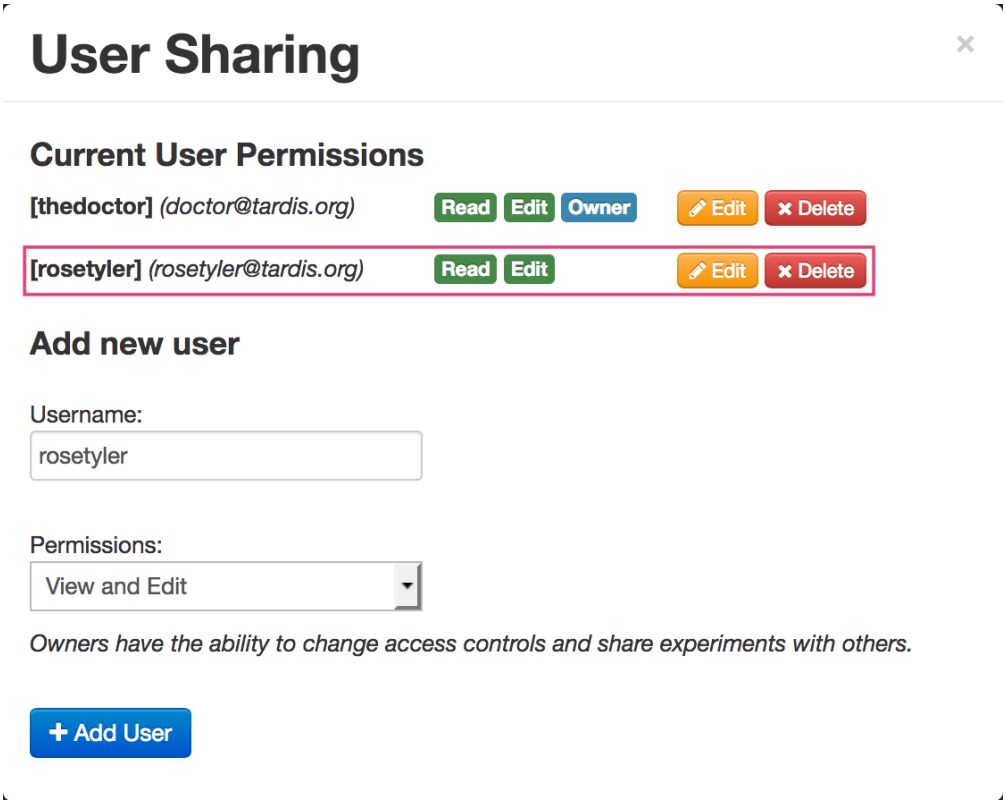

Experiment
Test Experiment
Peter Capaldi
Microscopy experiment
Description
Metadata
Sharing
Transfer Datasets
Public Access
Current Public Access Settings
Private
Current License
Unspecified
Change Public Access
Sharing
Users
Users who have a share in this experiment:
Username
Name
thedoctor
none
Change User Sharing
Groups
Groups who have a share in this experiment:
There are currently no groups with access to this experiment.
Change Group Sharing
Links
This experiment is private. A temporary link can be created by its owner(s) and privately shared
Temporary access links provide full access to recipients regardless of an experiment's public st
No tokens defined
Create New Temporary Link

4. Enter the **Username** of the user with which you want to share the data.



Note: Entering the users name or email address with activate autocomplete, which helps to find the username of the user.

5. Select the permissions you'd like to give the user from the **Permissions** dropdown.
6. Click the **+Add User** button.
7. You should now see a new entry under the **Current User Permissions** section showing the user you've shared the Experiment with and the permissions you've given them.

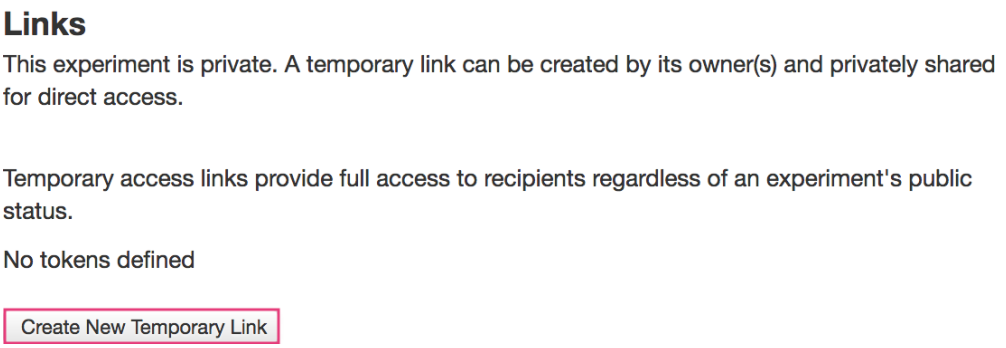


Sharing Data via a Temporary Link

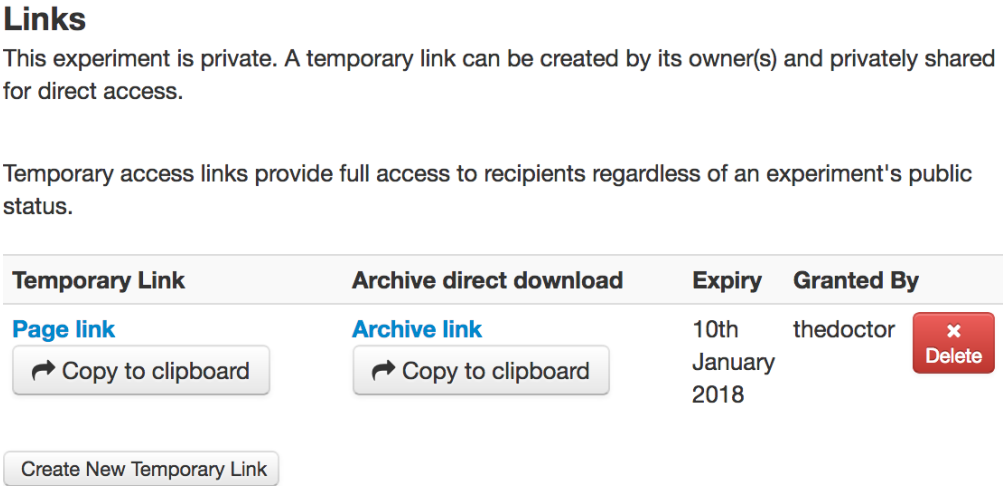
You can also share Data in an Experiment via a temporary (obfuscated) link. Temporary links expire after 1 month or can be deleted manually. Note: temporary links are not protected (i.e., anybody with the link can access the data); however, they are obfuscated and are thus difficult to find without prior knowledge of the link.

To share an Experiment via a temporary link:

1. Navigate to the Experiment you want to share.
2. Click on the **Sharing** tab on the left of the page.
3. Click the **Create New Temporary Link** button in the links section.



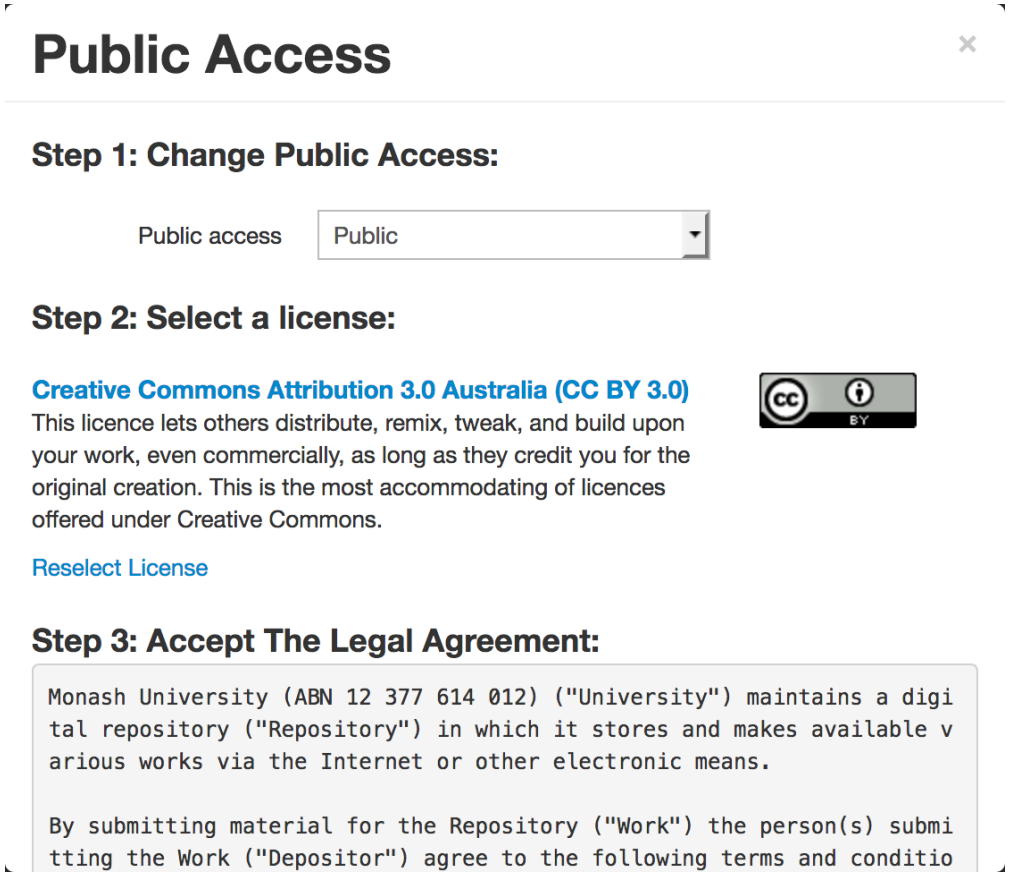
4. You should now see a new entry in the **Links** section. The new entry will provide links to both the Experiment page and a direct download link. These can be copied to the clipboard using the relevant buttons. These can then be shared with collaborators using email etc.



Publishing Data

Access to an Experiment in MyTardis is set to "private" by default and no license is attached to the data; however, MyTardis allows data from an Experiment to be made publicly available with an appropriate license. To make an Experiment publicly available:

1. Navigate to the Experiment you wish to publish and activate the **Sharing** tab to the left of the page.
2. Click the **Change Public Access** button. This will activate the Public Access dialog and display the current status of the data.
3. Select **Public** in the "Public access" dropdown menu.
4. Select an appropriate license from the list of licenses presented. In the following screenshot, a "Creative Commons Attribution" license is selected; however, your choices may be different as licenses are configured by your service provider.



5. Agree to the "Terms of Publishing" of your service provider.
6. Click **Confirm**

11.1.2 SFTP Access

SFTP is a secure and convenient way to access, browse and download large Experiments, Datasets and Files from MyTardis.

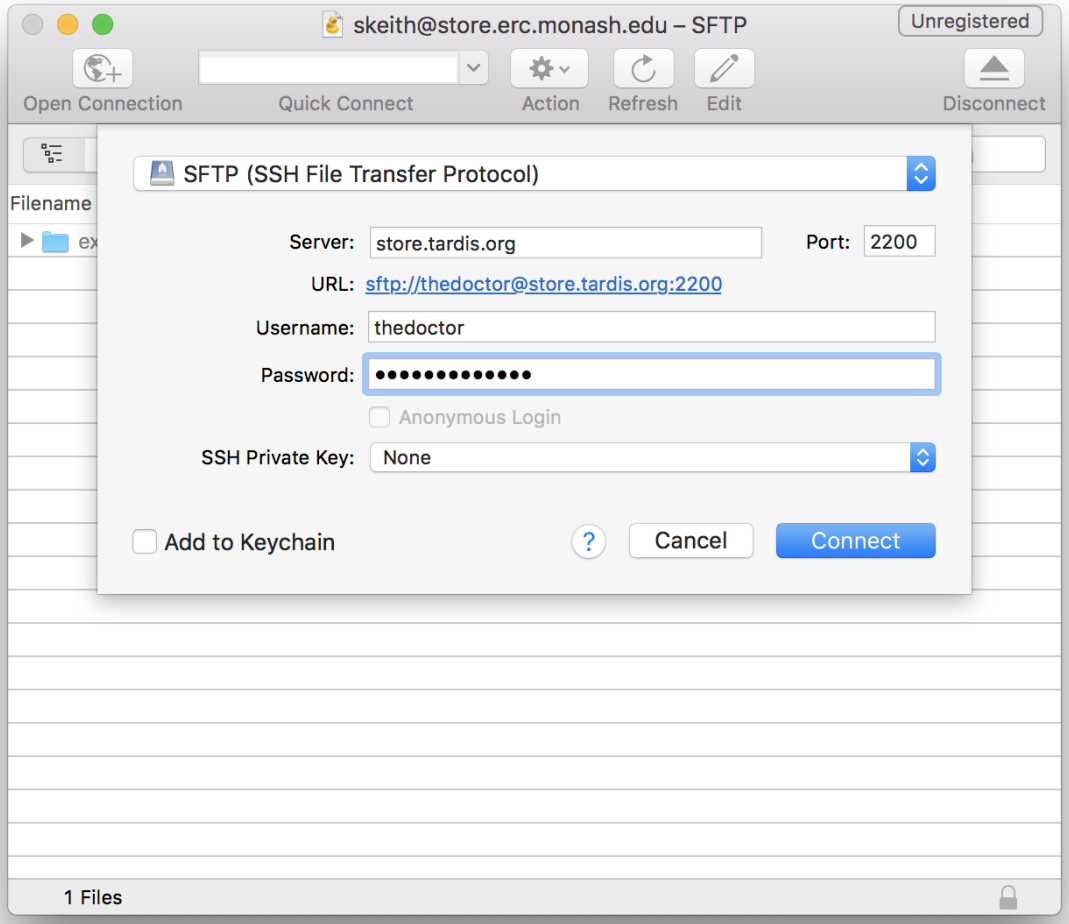
In order to access data on MyTardis via SFTP, you will first need to install an SFTP client. There are many free and commercial SFTP clients for you to choose from; however, we recommend Cyberduck (Win & Mac), FileZilla (All platforms) or WinSCP (Win only) for the majority of users. The instructions here will focus on Cyberduck (Win & Mac).

Browse and Download your MyTardis data using SFTP

1. Open your SFTP client and create a new connection with the following configuration:

Parameter	Value
Server:	*URL for your MyTardis deployment* e.g. https://store.erc.monash.edu
Port:	2200
Username:	*Your MyTardis username*
Password:	*Your MyTardis password*

Note: you should substitute your MyTardis details for *italicised* values.



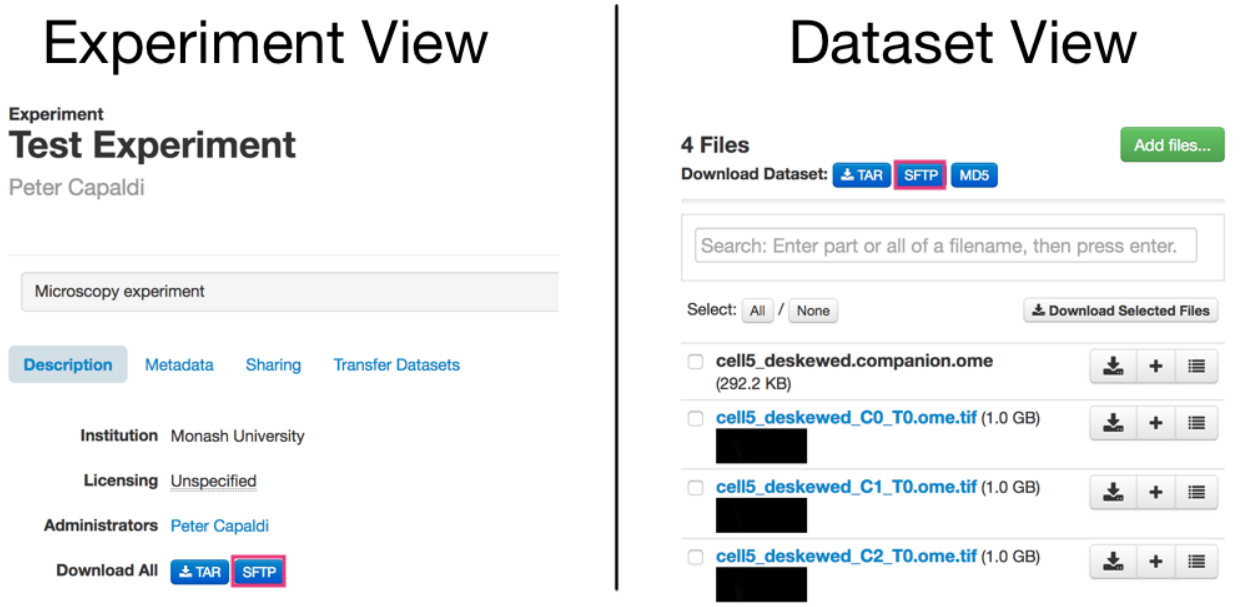
2. Click **Connect**
3. Upon successful connection you will be presented with a file browser showing all your data on MyTardis.

Data is organised according to the Experiment/Dataset/Data File hierarchy/structure described in the *Organising Data* section.

Browse and/or Download a Specific Experiment or Dataset

MyTardis also provides a convenient way to access/browse a particular Experiment or Dataset via SFTP.

1. Navigate to the Experiment or Dataset page that you want to access via SFTP using your web browser.
2. There is an **SFTP** button in the *Download* section on both the Experiment and Dataset views.



3. Clicking the **SFTP** button at either of these two locations will redirect you to a page with instructions and links for starting an SFTP session for a specific experiment or dataset.

11.2 Configuration and Administration

11.2.1 Installation

The sections through to Extended Configuration below provide a Quick Start guide for getting a basic MyTardis installation up and running. The following section provides additional information on advanced configuration and add-on capabilities of MyTardis.

Prerequisites

Ubuntu 14.04

Run the script:

```
sudo bash install-ubuntu-requirements.sh
```

It will install required packages with this command:

```
apt-get update
apt-get install git ipython libldap2-dev libsasl2-dev \
  libssl-dev libxml2-dev libxslt1-dev nodejs npm python-anyjson \
  python-billiard python-bs4 python-crypto python-dateutil \
  python-dev python-flexmock \
  python-pip python-pystache \
  python-virtualenv python-wand python-yaml virtualenvwrapper \
  zlib1g-dev libfreetype6-dev libjpeg-dev
```

Redhat/CentOS 7

Redhat based distributions are not actively supported. Here is a starting point, but you may require additional packages:

```
sudo yum install epel-release
sudo yum install cyrus-sasl-ldap cyrus-sasl-devel openldap-devel \
libxslt libxslt-devel libxslt-python git gcc graphviz-devel \
python-virtualenv python-virtualenvwrapper python-pip php-devel php-pear \
ImageMagick ImageMagick-devel libevent-devel compat-libevent14-devel
```

Note that at least one of `libevent-devel` and `compat-libevent14-devel` needs to be successfully installed as they are alternatives of the same package for different distributions.

Download

To get the most recent stable release:

```
git clone -b master https://github.com/mytardis/mytardis.git
cd mytardis
```

This clones the repository as read-only.

Or, to get the current development branch:

```
git clone -b develop git://github.com/mytardis/mytardis.git
cd mytardis
```

Quick configuration

It is recommended that you use a virtualenv. The list of packages above includes the `virtualenvwrapper` toolkit. Set up your environment with these commands:

Ubuntu 14.04:

```
source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
```

Redhat/CentOS 7:

```
source /usr/bin/virtualenvwrapper.sh
```

Then create the `mytardis` virtual environment

```
mkvirtualenv --system-site-packages mytardis
pip install -U pip
```

Note: the next time you want to work with this virtualenv, run the appropriate `source` command and then use the command: `workon mytardis`

MyTardis dependencies are then installed with pip:

```
pip install -U -r requirements.txt
```

For Redhat/CentOS 7, also run:

```
pip install -U -r requirements-centos.txt
```

Javascript dependencies are then installed with npm:

```
npm install
```

Configuring MyTardis is done through a standard Django *settings.py* file. MyTardis comes with a sample configuration file at `tardis/default_settings.py`. You can import this as the basis of your own config file - options defined here will override the relevant options in `default_settings.py`.

Create a new file `tardis/settings.py` containing the following:

```
from default_settings import *

# Add site specific changes here.

# Turn on django debug mode.
DEBUG = True

# Use the built-in SQLite database for testing.
# The database needs to be named something other than "tardis" to avoid
# a conflict with a directory of the same name.
DATABASES['default']['ENGINE'] = 'django.db.backends.sqlite3'
DATABASES['default']['NAME'] = 'tardis_db'
```

In addition you will need to create a new `SECRET_KEY` for your installation. This is important for security reasons.

A convenient method is to run the following command in your mytardis installation location:

```
python -c "import os; from random import choice; key_line = '%sSECRET_KEY=\"%s\"  #␣
↪generated from build.sh\n' % ('from tardis.default_settings import * \n\n' if not␣
↪os.path.isfile('tardis/settings.py') else '', ''.join([choice(
↪'abcdefghijklmnopqrstuvwxyz0123456789\\!@#$%^&*(-_=+)') for i in range(50)]));␣
↪f=open('tardis/settings.py', 'a+'); f.write(key_line); f.close()"
```

This is the minimum set of changes required to successfully run the server in development mode. You can make any other site-specific changes as necessary.

Initialisation

Create and configure the database:

```
python mytardis.py migrate
```

This avoids creating a superuser before the MyTardis specific `UserProfile` table has been created. More information about the `migrate` commands can be found at admin.

Next, create a superuser:

```
python mytardis.py createmysuperuser
```

MyTardis can now be executed in its simplest form using:

```
python mytardis.py runserver
```

This will start the Django web server at http://localhost:8000/.

Extended configuration

See below for some extra configuration options that are specific to MyTardis.

An automatically generated documentation of the settings can be found in pydoc/tardis.

Essential Production Settings

These settings are essential if you want to run MyTardis in production mode (`DEBUG = False`).

SECRET_KEY
: This key needs to be unique per installation and, as the name implies, be kept secret.

  A new one can be conveniently generated with the command:

```
echo "SECRET_KEY='`python mytardis.py generate_secret_key`'" >> tardis/settings.py
```

However, the more complex command shown above needs to be used at installation time.

ALLOWED_HOSTS
: `ALLOWED_HOSTS` is a list of hostnames and/or IP addresses under which the server is accessible. If this is not set you will get a 500 Error for any request.

Database

`tardis.default_settings.`**DATABASE_ENGINE**
: The database server engine that will be used to store the MyTardis metadata, possible values are *postgresql_psycopg2*, *postgresql*, *mysql*, *sqlite3* or *oracle*.

`tardis.default_settings.`**DATABASE_NAME**
: The name of the database to used to store the data, this is the path to the database if you are using the SQLite storage engine.

`tardis.default_settings.`**DATABASE_USER**
: The user name used to authenticate to the database. If you are using SQLite this field is not used.

`tardis.default_settings.`**DATABASE_PASSWORD**
: The password used to authenticate to the database. If you are using SQLite this field is not used.

`tardis.default_settings.`**DATABASE_HOST**
: The host name of the machine hosting the database service. If this is empty then localhost will be used. If you are using SQLite then this field is ignored.

`tardis.default_settings.`**DATABASE_PORT**
: The port the database is running on. If this is empty then the default port for the database engine will be used. If you are using SQLite then this field is ignored.

LDAP

For further information see *LDAP authentication*

Repository

`tardis.default_settings.`**`DEFAULT_STORAGE_BASE_DIR`**
The path to the default MyTardis storage location. This is where files will be stored to if you do not provide any other location explicitly through ‘‘StorageBox‘‘es.

`tardis.default_settings.`**`STAGING_PATH`**
The path to the staging path. This is where new files to be included in datasets will be sourced. This functionality, however, is not available in current MyTardis versions.

`tardis.default_settings.`**`REQUIRE_DATAFILE_CHECKSUMS`**
If True, a Datafile requires an MD5 or SHA-512 checksum from the time it is first recorded in the MyTardis database. This enables a model-level constraint check each time a Datafile record is saved. Defaults to True. Datafile record is saved.

`tardis.default_settings.`**`REQUIRE_DATAFILE_SIZES`**
If True, a Datafile require a size from the time it is first recorded in the MyTardis database. This enables a model-level constraint check each time a Datafile record is saved. Defaults to True.

`tardis.default_settings.`**`REQUIRE_VALIDATION_ON_INGESTION`**
If True, ingestion of a Datafile is only permitted if the Datafile matches its supplied size and/or checksums. Defaults to True.

Access Rights & Licensing

Licences

By default, no licences are loaded. A user can make metadata public without assigning a licence, but they cannot allow public access to their data.

Creative Commons licences (for Australia) are available in `tardis/tardis_portal/fixtures/cc_licenses.json`. You can load them with

```
python mytardis.py loaddata.
```

You can use the admin interface to add other licences. Please ensure `allows_distribution` is set to the correct value to ensure the licence appears in conjunction with suitable public access types.

Legal Notice

When changing the public access rights or licence for an experiment, a legal notice is displayed. You can override it by either:

1. creating a new app (probably your site theme) and putting your legal text in `tardis/apps/<app_name>/static/publishing_legal.txt`, or
2. directly making changes to `tardis/tardis_portal/static/publishing_legal.txt`.

Filters

`tardis.default_settings.`**`POST_SAVE_FILTERS`**
This contains a list of post save filters that are execute when a new data file is created.

The **POST_SAVE_FILTERS** variable is specified like:

```
POST_SAVE_FILTERS = [
    ("tardis.tardis_portal.filters.exif.EXIFFilter", ["EXIF", "http://exif.schema
↪"]),
    ]
```

For further details please see the *Filter Setup* section.

Archive Organizations

tardis.default_settings.DEFAULT_ARCHIVE_FORMATS.
This is a prioritized list of download archive formats to be used in contexts where only one choice is offered to the user; e.g. the "download selected" buttons. (The list allows for using different archive formats depending on the user's platform.)

tardis.default_settings.DEFAULT_PATH_MAPPER.
This gives the default archive "organization" to be used. Organizations are defined via the next attribute.

tardis.default_settings.DOWNLOAD_PATH_MAPPERS.
This is a hash that maps archive organization names to Datafile filename mapper functions. These functions are reponsible for generating the archive pathnames used for files written to "tar" and "zip" archives by the downloads module.

The **DOWNLOAD_PATH_MAPPERS** variable is specified like:

```
DOWNLOAD_PATH_MAPPERS = {
    'test': ('tardis.apps.example.ExampleMapper',),
    'test2': ('tardis.apps.example.ExampleMapper', {'foo': 1})
}
```

The key for each entry is the logical name for the organization, and the value is a tuple consisting of the function's pathname and a set of optional keyword arguments to be passed to the function. At runtime, the function is called with each Datafile as a positional argument, and an additional 'rootdir' keyword argument. The function should compute and return a (unique) pathname based on the Datafile and associated objects. If the function returns **None**, this tells the archive builder to leave out the file.

By default, the archive builder uses the built-in "deep-storage" mapper which gives pathnames that try to use directory information to rebuild a file tree.

Storage Locations (`StorageBox` es)

A MyTardis instance can be configured to support multiple locations (`StorageBox` es) for storing data files. Each location holds copies (`DataFileObject` s) of `DataFile` s that are recorded in the MyTardis database.

The `StorageBox` architecture is compatible with the Django File Storage API. This makes it relatively easy to support a number of different storage backends such as cloud storage or SFTP servers. Please refer to the *StorageBox documentation* for more detailed information.

Single Search

Instructions on installing and configuring Elasticsearch for advanced search are available from *Setting Up Search*.

Additional Tabs

Additional and custom tabs may be configured in MyTardis on a per-installation basis. The tabs are implemented as separate Django applications with a single view (index), listed in the TARDIS_APPS configuration item and either linked to, or installed in the TARDIS_APP_ROOT directory, by default `tardis/apps`.

Documentation on the additional tabs is available from tabs.

Additional Views

Custom views may be configured in MyTardis on a per-installation basis. The tabs are implemented as separate Django applications with a single view function listed in the `*_VIEWS` configuration item and added to the `INSTALLED_APPS` list.

Refer to the views documentation for further information.

Site Customisations

Some settings that allow customised messages and styles.

```
PUBLICATION_INTRODUCTION = """
<p><strong>... introduction and publication agreement ...</strong></p>
"""
SITE_STYLES = ''  # should be CSS

# if either GA setting is empty, GA is disabled
GOOGLE_ANALYTICS_ID = ''  # whatever Google provides
GOOGLE_ANALYTICS_HOST = ''  # the host registered with Google

# these refer to any template finder findable location, e.g. APPDIR/templates/...
CUSTOM_ABOUT_SECTION_TEMPLATE = 'tardis_portal/about_include.html'
CUSTOM_USER_GUIDE = 'user_guide/index.html'
```

Deployment

Collecting Static Files

For performance reasons you should avoid static files being served via the application, and instead serve them directly through the webserver.

To collect all the static files to a single directory:

```
python mytardis.py collectstatic
```

`tardis.default_settings.`**`STATIC_ROOT`**
: This contains the location to deposit static content for serving.

`tardis.default_settings.`**`STATIC_URL`**
: The path static content will be served from. (eg. `/static` or `http://mytardis-resources.example.com/`)

See also:

collectstatic, STATIC_ROOT, STATIC_URL

Serving with Nginx + Gunicorn

In this configuration, Nginx serves static files and proxies application requests to a Gunicorn server:

```
   HTTP  +-----------+       +------------------+
+------->|   Nginx   +------>| Gunicorn Server |
         +-----------+       +------------------+
           0.0.0.0:80          127.0.0.1:8000
```

Nginx should then be configured to send requests to the server. Here is an example configuration (SSL part from Mozilla SSL configurator). Please amend for your own needs and understand the settings before deploying it.:

```
upstream mytardis {
    server unix:/var/run/gunicorn/mytardis/socket;
    server 127.0.0.1:8000 backup;
}
server {
    listen 80 default_server;
    server_name demo.mytardis.org;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 default_server ssl;
    server_name demo.mytardis.org;

    # certs sent to the client in SERVER HELLO are concatenated in ssl_certificate
    ssl_certificate /path/to/signed_cert_plus_intermediates;
    ssl_certificate_key /path/to/private_key;
    ssl_session_timeout 5m;
    ssl_session_cache shared:SSL:50m;

    # Diffie-Hellman parameter for DHE ciphersuites, recommended 2048 bits
    ssl_dhparam /path/to/dhparam.pem;

    # intermediate configuration. tweak to your needs.
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-
↪AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-
↪AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
↪SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-
↪ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-
↪SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-
↪AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-
↪SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!
↪EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-
↪DES-CBC3-SHA';
    ssl_prefer_server_ciphers on;

    # HSTS (ngx_http_headers_module is required) (15768000 seconds = 6 months)
    add_header Strict-Transport-Security max-age=15768000;

    # OCSP Stapling ---
    # fetch OCSP records from URL in ssl_certificate and cache them
    ssl_stapling on;
    ssl_stapling_verify on;
```

(continues on next page)

(continued from previous page)

```
    ## verify chain of trust of OCSP response using Root CA and Intermediate certs
    ssl_trusted_certificate /path/to/root_CA_cert_plus_intermediates;

    resolver <IP DNS resolver>;

    client_max_body_size 4G;
    keepalive_timeout 5;

    gzip off;  # security reasons
    gzip_proxied any;
    # MyTardis generates uncompressed archives, so compress them in transit
    gzip_types application/x-javascript text/css;
    gzip_min_length 1024;
    gzip_vary on;

    location / {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Protocol $scheme;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://mytardis;
        # this is to solve centos 6 error:
        # upstream prematurely closed
        client_max_body_size 4G;
        client_body_buffer_size 8192k;
        proxy_connect_timeout 2000;
        proxy_send_timeout 2000;
        proxy_read_timeout 2000;
    }

    location /static/ {
        expires 7d;
        alias /srv/static_files/;
    }
}
```

Don't forget to create the static files directory and give it appropriate permissions. The location is set in the `settings.py` file.

```
# Collect static files to ``settings.STATIC_ROOT``
python mytardis.py collectstatic
# Allow Nginx read permissions
setfacl -R -m user:nginx:rx static_dir
```

See also:

Django with Gunicorn

Serving with Apache HTTPD + mod_wsgi

We do not support the use of Apache. If you need this and want to support this use case, we welcome your contribution of any relevant documentation.

11.2.2 Authentication Methods

Users

MyTardis supports several sources of authentication and identity, referred to as user providers.

In the *settings.py* user providers are activated by specifying them within the **USER_PROVIDERS** variable:

```
USER_PROVIDERS = ('tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider',)
```

Groups

MyTardis also supports several sources for group membership information, referred to as group providers.

In the *settings.py* group providers are activated by specifying them within the **GROUP_PROVIDERS** variable:

```
GROUP_PROVIDERS = ('tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider',
                   'tardis.tardis_portal.auth.vbl_auth.VblGroupProvider',)
```

Included Auth Plugins

HTTP Basic Endpoint Authentication

tardis.tardis_portal.auth.httpbasicendpoint_auth

HTTP Basic Endpoint authentication uses access to a HTTP resource (or endpoint) to determine if authentication should be allowed.

To use HTTP Basic Endpoint authentication, you'll need a HTTP resource protected by HTTP Basic Auth which is accessible from the MyTardis server.

In the *settings.py* add the following to *AUTH_PROVIDERS* with an appropriate name. eg.

```
AUTH_PROVIDERS = (
    ('acls', 'acls', 'tardis.tardis_portal.auth.httpbasicendpoint_auth.
↪HttpBasicEndpointAuth'),
)
```

On each request, MyTardis will attempt to use basic authentication with the provided credentials to access the HTTP resource. If it fails, access is denied.

Endpoint to use in HTTP Basic Endpoint Auth. eg.

```
HTTP_BASIC_AUTH_ENDPOINT = 'https://test.example/endpoint'
```

LDAP Authentication

tardis.tardis_portal.auth.ldap_auth

To enable LDAP you'll need to specify which components of the LDAP authentication backend are enabled. In the *settings.py* add the following to either Authentication, User Provider, Group Provider slugs.:

```
'tardis.tardis_portal.auth.ldap_auth.ldap_auth'
```

This is a wrapper function that allows initialisation of the LDAP provider using *settings.py* values.

`ldap_auth()` Function

The following are configuration settings that are used when initialising the LDAP backend.

`tardis.default_settings.`**`LDAP_TLS`**
Enable TLS connections.

`tardis.default_settings.`**`LDAP_URL`**
Set the URL of the LDAP server, e.g. *ldap://localhost:389/*

`tardis.default_settings.`**`LDAP_USER_LOGIN_ATTR`**
Set the login attribute of the users, usually this will be either *cn* or *uid*

`tardis.default_settings.`**`LDAP_USER_ATTR_MAP`**
The LDAP user attribute map is used to map internal identifiers like *first_name*, *last_name* and *email* to their LDAP equivalents e.g. *{"givenName": "first_name", "sn": "last_name", "mail": "email"}*

`tardis.default_settings.`**`LDAP_GROUP_ID_ATTR`**
This is where you specify the group identifier from LDAP, usually it will be *cn*.

`tardis.default_settings.`**`LDAP_GROUP_ATTR_MAP`**
This map is used to map internal identifiers like *display* e.g. *{"description": "display"}*

`tardis.default_settings.`**`LDAP_BASE`**
Sets the search base of the LDAP queries *dc=example, dc=com*

`tardis.default_settings.`**`LDAP_USER_BASE`**
Sets the suffix to append to the user RDN (e.g. uid=jsmith) to construct the DN e.g. *"ou=People, "* + *LDAP_BASE*

`tardis.default_settings.`**`LDAP_GROUP_BASE`**
Sets the suffix to append to the group RDN (e.g. cn=group1) to construct the DN e.g. *"ou=Group, "* + *LDAP_BASE*

Temporary Token Authentication

`tardis.tardis_portal.auth.token_auth`

To use token authentication, you'll need to specify the following *settings.py*

'tardis.tardis_portal.auth.token_auth.TokenGroupProvider',

TOKEN_EXPIRY_DAYS = 30

TOKEN_LENGTH = 30

TOKEN_USERNAME = 'tokenuser'

and create a user with

bin/django createtokenuser

Cleaning up

```
bin/django cleanuptokens
```

It is recommended that you schedule regular purging of expired tokens. Set a cronjob to run bin/django cleanuptokens

Expiry

Token auth works by hijacking the group provider system.

MyTardis groups are calculated and cached when a user logs in.

This means that if a session is active, and a token becomes in valid (either through deletion or expiry) that access will still be granted. To mitigate this, when a token user logs in, an explicit expiry is set on their session - the earlier of 4am the next day, or the session expiry date (the end of the day)

This forces the user to attempt to log in again, and be denied access.

- `tardis.tardis_portal.auth.ip_auth`

The *tardis.tardis_portal.auth package* module contains the authentication specific code.

11.2.3 Filter Setup

filters

Filters are called once an object has been saved to the database. They build on the Django Signals Infrastructure.

In the *settings.py* file filters are activated by specifying them within the `POST_SAVE_FILTERS` variable, for example

```
POST_SAVE_FILTERS = [
    ("tardis.tardis_portal.filters.exif.EXIFFilter",
     ["EXIF", "http://exif.schema"]),
]
```

The format they are specified in is

```
(<filter class path>, [args], {kwargs})
```

Where `args` and `kwargs` are both optional.

11.2.4 Database

Initialising

When creating a new database the `syncdb` command will need to be called to initialise the schema and insert the initial data fixtures.

Usage

```
python mytardis.py syncdb --noinput --all
```

Migrating

Some of the upgrades to MyTardis will require that the database schema be upgraded to match the internal data model. This tool migrates data from old database schemas to the current one. It detects which version of the database you are currently running and will automatically migrate to the current version. If you decided not to use South for migrating the data models and have removed it from the list of INSTALLED_APPS in the settings.py file, you could skip this step and go on initiating an administrator account with the command `createmysuperuser` , which is described in more detail below.

In certain cases it is also necessary to update the permissions table.

Usage

Intial migration after `syncdb --all`:

```
python mytardis.py migrate --fake
```

Then:

```
python mytardis.py migrate
```

If the model changes require it, run:

```
python mytardis.py update_permissions
```

creating superuser

After success of database initialization or migration, please use a command line utility called `createmysuperuser` to create an administrator account using the admin site which is hooked to the URL /admin/.

Usage

```
python mytardis.py createmysuperuser
```

Backup

The backupdb command allows to backup and to restore of the MyTardis database. The command uses the corresponding database tool to facilitate this task. Currently implemented are PostgreSQL and MySQL. In case of backup, a directory called `backups` is created (if it does not exists) in the working directory. In case of restore, the database for storing the tablespace must already exist before loading the backup file into the database.

Usage

```
python mytardis.py backupdb
-r FILE, --restore=FILE
-v VERBOSITY, --verbosity=VERBOSITY
--settings=SETTINGS
--pythonpath=PYTHONPATH
--traceback
--version
-h, --help
```

11.2.5 Schema and Parameter Sets

MyTardis stores metadata as *Parameters*, which are grouped in to *Parameter Sets*, which are defined by a *Schema*.

Managing Schema

Schema are managed through the Django administrative interface. The administrative interface is normally accesible from a link similar to:

```
http://domain.com:8000/admin/
```

Selecting "Schemas" in the adminstrative interface will display a list of the installed schemas. Clicking on a schema displays the editor for that schema.

Schema definitions are the combination of two tables, *Schema* and *ParameterName*.

The *Schema* fields are:

Namespace The namespace uniquely identifies the schema. When exporting an experiment as a METS file the namespace is used as an XML Namespace, and thus must follow the XML standard, i.e. in the form of a URL.

The MyTardis naming convention is:

```
http://domain.com/localidentifiers/schemaname/version
```

Name The display name of the schema.

Type Experiment, Dataset or Datafile.

Subtype Used to group and identify schema for forms based searching.

The *ParameterName* fields are:

Schema The namespace of the schema which this parameter belongs to.

Name The identifier used to ingest parameters.

Full Name The display name of the parameter.

Units The display name of the units for numerical values.

Data Type One of:

- Numeric
- String
- Longstring
- URL
- Filename
- Datetime
- Link

Strings use the input field widget for editing, while longstrings use a textarea widget.

Immutable If true, no user editing of the parameter is allowed, regardless of access.

Comparison Type The type of search to be performed with Forms Based Search. Not used by Advanced Search.

Is Searchable Flag whether this parameter is searchable.

Choices If defined, a drop down list of values is displayed.

Order The display order of the parameters within the schema. Sorting is by Order, then alphabetically.

11.2.6 Setting Up Search

Tardis comes with a single search option which provides users with a search field that returns a list of matching Experiments, Datasets and Datafiles.

The single search box uses Elasticsearch, with a django-haystack frontend, and accordingly requires some setup. The single search box is disabled by default.

Setting up Elasticsearch

Elasticsearch doesn't work out of the box with MyTardis. It is not currently installed with pip and requires a number of manual steps to get working.

Elasticsearch can be found here: https://www.elastic.co/products/elasticsearch

The following are a very simple list of steps that will get everything up and running. It is advisable to follow up with the person responsible for overseeing security policy at your home institution to see if any extra setup is necessary.

Django Configuration

Enabling Single Search

A list of default settings for Single Search are already in default_settings.py in the MyTardis repository. Single search is enabled by setting the SINGLE_SEARCH_ENABLED option to True.

Other settings are shown below:

HAYSTACK_CONNECTIONS

The default value is

```
HAYSTACK_CONNECTIONS = {
     'default': {
         'ENGINE': 'haystack.backends.elasticsearch_backend.'
                   'ElasticsearchSearchEngine',
         'URL': 'http://127.0.0.1:9200/',
         'INDEX_NAME': 'haystack',
     },
 }
```

HAYSTACK_SIGNAL_PROCESSOR

This setting determines when the information in Elasticsearch is updated. The default value updates it in real time with db changes

```
HAYSTACK_SIGNAL_PROCESSOR = 'haystack.signals.RealtimeSignalProcessor'
```

A more performant implementation would trigger an update task on each db change.

Please see django-haystack documentation for further information: http://haystacksearch.org

Updating Indexes

Once Elasticsearch is set up, and Single Search is enabled (i.e. the `SINGLE_SEARCH_ENABLED` option in settings is set to True) Haystack will automatically register the addition of and changes to models and reflect these in the search index. That is, as soon as a new instance of a model is added to the database, or changes are made to an existing isntance, these changes will be searchable.

If you're adding search to an existing deployment of Django then you'll need to manually trigger a rebuild of the indexes (automatic indexing only happens through signals when models are added or changed).

Haystack registers a number of management commands with the Django framework, the important one here being the *rebuild_index* command. To rebuild, navigate to your checkout and call the following command

```
python mytardis.py rebuild_index
```

Haystack will then ask you to confirm your decision (Note: Rebuilding will destroy your existing indexes, and will take a while for large datasets, so be sure), and then start rebuilding.

11.2.7 Storage Options and Architecture in MyTardis

Database layout for storage

The storage for each DataFile is configured individually. A "way to store data" is called StorageBox. Each file has one or many related DataFileObjects, which link a DataFile with a StorageBox. A DataFile can have several copies stored in different StorageBoxes via several DataFileObjects.

StorageBoxes

StorageBoxes contain all the information needed to store a file except an id unique to the file and storagebox.

Each StorageBox points to a class that implements the Django storage API via a python module path as string.

Optional instatiation parameters for each `StorageBox` can be stored in StorageBoxOptions. These are used as parameters to the storage class set in the `django_storage_class` attribute of a `StorageBox`

These parameters are string types by default. However, by setting the optional parameter `value_type` to `'pickle'`, any picklable object can be stored here and hence used for instantiation of the storage class.

Optional classification and other metadata can be stored in StorageBoxAttributes.

A special case is where someone registers a file and wants to put it into location themselves but needs to be given the place to put it (via the API). Such situation can only be resolved with StorageBoxes that implement the "build_save_location" function. Such StorageBoxes need to have a StorageBoxAttribute with key "staging" and value "True".

DataFiles

DataFiles are the logical representation of the file. They contain information about the name, size, checksum, dates etc.

DataFileObjects

DataFileObjects point to the DataFile they belong to and the StorageBox they reside in. They also have an identifier that the StorageBox uses to find the actual file. DataFileObjects also have a date, and a state-flag.

Available backends

Django storage API compatible backends are available for example at https://github.com/jschneier/django-storages

We have tested the following backends with MyTardis:

- File on disk or any other system mounted storage
- SFTP
- SWIFT Object Storage

Documented backends

S3 compatible storage

S3 and S3 compatible storage backends are supported by the django-storages package.

We added some modifications that allow database driven configuration. Until these changes are accepted by the maintainer, use this fork and branch: https://github.com/monash-merc/django-storages/tree/mytardis-branch

Install it with:

```
pip install -e git+https://github.com/monash-merc/django-storages.git@mytardis-branch
↪#egg=django-storages
```

Configuration

The following options can be set as StorageBoxOption on your S3 storage box or system wide in your Django settings (a default of * labels mandatory settings):

StorageBoxOption	Django Setting	Default
access_key	AWS_S3_ACCESS_KEY_ID AWS_ACCESS_KEY_ID	*
secret_key	AWS_S3_SECRET_ACCESS_KEY AWS_SECRET_ACCESS_KEY	*
file_overwrite	AWS_S3_FILE_OVERWRITE	True
headers	AWS_HEADERS	`{}`
bucket_name	AWS_STORAGE_BUCKET_NAME	*
auto_create_bucket	AWS_AUTO_CREATE_BUCKET	False
default_acl	AWS_DEFAULT_ACL	`public-read`
bucket_acl	AWS_BUCKET_ACL	`default_acl`
querystring_auth	AWS_QUERYSTRING_AUTH	True
querystring_expire	AWS_QUERYSTRING_EXPIRE	3600
reduced_redundancy	AWS_REDUCED_REDUNDANCY	False
location	AWS_LOCATION	`''`
encryption	AWS_S3_ENCRYPTION	False
custom_domain	AWS_S3_CUSTOM_DOMAIN	None
calling_format	AWS_S3_CALLING_FORMAT	`SubdomainCallingFormat()`
secure_urls	AWS_S3_SECURE_URLS	True
file_name_charset	AWS_S3_FILE_NAME_CHARSET	`utf-8`
gzip	AWS_IS_GZIPPED	False
preload_metadata	AWS_PRELOAD_METADATA	False
gzip_content_types	GZIP_CONTENT_TYPES	`('text/css', 'text/javascript', 'application/javascript', 'application/x-javascript', 'image/svg+xml')`
url_protocol	AWS_S3_URL_PROTOCOL	`http:`
host	AWS_S3_HOST	`S3Connection.DefaultHost`
use_ssl	AWS_S3_USE_SSL	True
port	AWS_S3_PORT	None
proxy	AWS_S3_PROXY_HOST	None
proxy_port	AWS_S3_PROXY_PORT	None
The max amount of memory a returned file can take up before being rolled over into a temporary file on disk. Default is 0: Do not roll over.		
max_memory_size	AWS_S3_MAX_MEMORY_SIZE	0

Appendix: Conversion of 'Replicas'

Replicas used to be the method of file storage abstraction in MyTardis versions 3.x. The StorageBoxes replace this. For pain-free upgrading, a conversion has been included with the database migrations that works as follows:

All 'Locations' that are local are converted to default (folder on disk) StorageBoxes. All 'Locations' that are not local are converted to dummy 'link only' StorageBoxes with the corresponding name. These can be upgraded manually to a StorageBox with an appropriate backend after the migration has taken place.

Max size is set to size of disk, hence for multiple locations on the same disk this number provides no protection. This also should be set to reasonable values manually after the migration.

Each 'Replica' of a file will be converted to a DataFileObject pointing to the relevant StorageBox.

All files are manually reverified, so that unverified entries can be checked for errors.

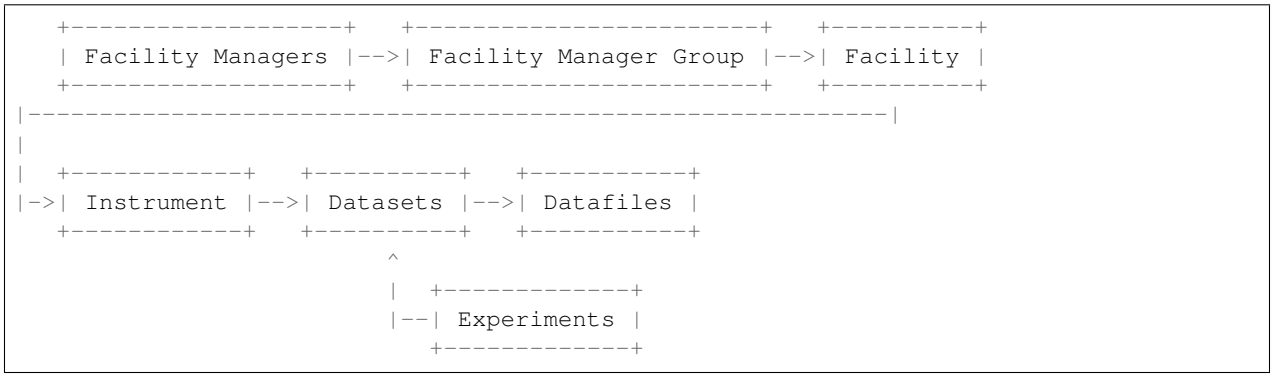
11.2.8 Facility Overview

Introduction

The Facility Overview included with MyTardis allows facility administrators to monitor the output of connected instruments, highlighting unverified files, dataset size, file counts and file listings.

Setup

In order for datasets to appear in the Facility Overview, each dataset must be associated with an Instrument, which is itself associated with a Facility. The Facility object will reference a facility administrator's group, members of which may view the Facility Overview.



The facility managers, facility manager groups, facilities and instruments may be configured via the django admin interface. MyData, the desktop uploader client for the MyTardis server, can be configured to assign the appropriate instrument to uploaded datasets at the point of ingestion.

It should be noted that the dataset visibility within the facility overview is limited to dataset and datafile listings only. Access to the experiment and dataset views, as well as raw data, is still controlled via the *ACL framework*.

Usage

Members of the facility manager groups for one or more facilities will see the "Facility Overview" menu item in the MyTardis web portal. After opening the facility overview, a list of recently ingested datasets will be displayed from the facility being managed. If a user manages multiple facilities, a blue drop-down selector will also appear on the right-hand side of the page. As the facility overview is designed to give a snapshot of recently uploaded datasets, older data is not immediately accessible; MyTardis' *search feature* is better suited to this.

In addition to simply listing the most recent datasets, the datasets can be grouped by instrument or by owner, and filtered by username, experiment name and instrument name. Note that while filters are active, it may appear as though no new pages are loaded by clicking "Load more", since the additional datasets fetched from the server might not match the active filters.

11.3 Development

11.3.1 Architecture

This page describes the architecture of MyTardis.

MyTardis is built on the Django web framework, which itself is built on Python, thus MyTardis follows the architectural model of Django.

Component Architecture

Web Server MyTardis is typically deployed with the standard Nginx + Gunicorn + Django + Python stack.

RDBMS Ideally, use PostgreSQL. MySQL or MariaDB may be usable but it is not as well tested.

Data Storage Local disk, network mounted disks as well as S3 and SFTP are supported.

Ingestion Ingestion is normally site specific. A desktop application is available (MyData).

Functional Architecture

MyTardis follows the standard Django architecture.

The major functional modules in MyTardis are:

API A RESTful API provides anonymous and authenticated access to most of the stored data.

Auth Authentication and Authorisation.

Download Data download functionality.

Filters Processing of metadata as data is uploaded, e.g. extract EXIF metadata from JPEG files.

Management Additional commands for the Django CLI. The backup utility is implemented as a Django command.

Migrations Database migration code.

Publish RIF-CS metadata publication.

Search Provides a searchable and authorisation enabled index for quickly finding data.

SFTP server Provides read access to data in MyTardis via the SFTP protocol

Information on the individual modules is available from the modindex.

Scalability Model

At the component / module level, performance and Scalability within MyTardis is achieved by:

- Allowing long lived or compute intensive operations to be hosted on separate machines from the main web server.
  - E.g. by running multiple web servers and sending long lived operations, such as ingestion, to a server reserved for such operations.
- Performance and Scalability of the database is achieved by a combination of 1) optimising the SQL requests issued by MyTardis, and 2) database specific scalability, please refer to the appropriate database documentation.
- Performance and Scalability of the web server is provided through the normal mechanisms for nginx / wsgi hosted python applications, e,g,:
  - Increasing individual Server capability
    - Individual Server performance / utilization may be managed by controlling the number of python (django) processes active at any one time.
  - Deploying multiple web servers

    * Multiple web servers may be deployed using standard mechanisms, e.g. load balancers. State (session) information is distributed using Django's standard session model.

- The Data Archive is a normal file system, e.g. NFS, SAMBA, etc., with performance and scalability dependent on the implementation and deployment.
- Extraction and formatting of metadata for ingestion is up to the client and may be distributed across any number of machines.

SQL Scalability Notes

The datafileparameter table is expected to be the single largest table by an order of magnitude, in the hundreds of millions of records (at the Australian Synchrotron).

For Postgres, this will probably be addressed by table partioning. In this case using a date field to partition is often recommended. Dates may be introduced as part of the support for anotations, raw data, versioned data and derived data.

Persistance and Data Architecture

Django provides an Object-Relational-Model that maps Django Models to the underlying relational database.

The Django Models are defined in `tardis.tardis_portal.models`.

Security Model

Authentication

Access to data catalogued by MyTardis may be either public, i.e. anonymous access allowed, or private, requiring authentication. The Authentication Framework is documented in ref-authframework.

Authorisation

MyTardis includes an extensible authorisation engine, documented in ref-authframework and authorisation.

Class Diagram

Unless the application has a very rich business model (which MyTardis does not), a class diagram doesn't convey much information in Django applications (it ends up being a flat list of classes). To understand the context in which the MyTardis classes are created, please refer to the Django documentation, http://docs.djangoproject.com/ on the responsibilities of models, views, managers and templates.

Source Code

This section describes the top level layout of the source code in the GitHub repository.

- `docs`
  - MyTardis User and Administrative documentation
- tardis
  - `apps`

    * **Contains all the optional functionality and installation specific** functionality. Examples include the optional ANDS Register (ands_register) and Related Info (related_info) tabs, and the ANSTO (mecat-ansto) and Australian Synchrotron (mecat-as) installation specific modules.
    * **Some applications are part of the main code base, others can be added** at installation time.

- `search`
    * Search related code
- `tardis_portal`
    * This is the main django application that contains most of the core MyTardis functionality

11.3.2 Authorisation Framework

Django Authorisation

Django has a built-in authorisation/permission mechanism that is in use by default. It is enabled in MyTardis in `default_settings.py` together with the custom object level permission framework described below.

```
AUTHENTICATION_BACKENDS = (
    'django.contrib.auth.backends.ModelBackend',
    'tardis.tardis_portal.auth.authorisation.ACLAwareBackend',
)
```

The Django default permissions are automatically available for each Model. The verbs are `add`, `change`, `delete`, and they can be queried on the user object as follows:

```
user.has_perm('tardis_portal.add_experiment')
user.has_perm('tardis_portal.add_dataset')
user.has_perm('tardis_portal.change_experiment')
user.has_perm('tardis_portal.delete_datasetparameterset')
```

There is a function in `tardis.tardis_portal.auth.authservice` called `_set_user_from_dict` that adds the following permissions for each new user created using custom methods:

```
'add_experiment'
'change_experiment'
'change_group'
'change_userauthentication'
'change_experimentacl'
```

These permissions apply in general and are augmented by ACLs

Object Level Permissions and Access Control Lists

The main purpose of the ACL system is to manage per experiment permissions. The architecture allows for future expansion to more find grained permission management. However, at this stage only the Experiment level is supported by the user interface.

Permissions are applied with a few predefined roles:

read read permission allows individuals and groups access to view an experiment.

write write permissions cover addition of new datasets and datafiles and also deletion of datafile.

delete delete permission allows deletion of datasets and experiments.

Roles are applied through the web using the *Control Panel* and can be applied to either users or groups.

To make an experiment public requires an explicit publish action.

The ACL permissions can be queried on the user object just like standard permissions, however, with the addition of the object in question:

```
user.has_perm('tardis_acls.change_experiment', experiment)
```

Verbs currently available are `change`, `view`, `delete`, `owns`, `share`.

The translation of ACLs to `has_perm` verbs is defined in a function in `tardis.tardis_portal.auth.authorisation`.

To allow for querying on any object related to experiments, extra logic was added to some of the models. To support the logic, in addition to ACLs, `has_perm` calls model functions named `_has_VERB_perm`, which allows model-specific permission logic.

The current policy is that if those functions return True or False then that result is returned without further checking. If they return an object, permissions will be checked for this object thereby allowing delegation.

11.3.3 RESTful API for MyTardis

The data and metadata stored in MyTardis instances is made accessible through a RESTful API.

Not all functionality has been exposed via the API at this time. This documentation reflects what is available and tested.

The API version `v1` is built on the Tastypie framework.

The RESTful API can also be explored via the automatically generated Swagger documentation at http://mytardis-example.com/api/v1/swagger/.

See swagger.io for details on the Swagger standard.

API accessible models

- Experiment
- ExperimentParameterSet
- ExperimentParameter
-
- Dataset
- DatasetParameterSet
- DatasetParameter
-
- DataFile
- DatafileParameterSet
- DatafileParameter
-
- StorageBox
- StorageBoxOption

- StorageBoxAttribute
-
- Schema
- ParameterName
-
- User
- Group
-
- Facility
- Instrument
-
- ObjectACL

Authentication

Currently implemented are Basic Auth, to be used via HTTPS only, and SessionAuth which queries Django sessions.

Due to our desire to provide information to users without any login, eg. for public data, the Basic Auth mechanism is slightly non-standard.

The standard sends an anonymous request, awaits a WWW-Authenticate header, then sends authentication credentials. Instead, this API sends public data for anonymous requests.

Using `curl` or the `requests` library this poses no problem. However, using `urllib2` or web browser without a Django session is not going to work out of the box.

Here is a snippet (found here: http://stackoverflow.com/questions/4628610/does-urllib2-support-preemptive-authentication-authentication) that makes urllib2 work, should you want to use this library:

```
class PreemptiveBasicAuthHandler(urllib2.BaseHandler):

        def __init__(self, password_mgr=None):
                if password_mgr is None:
                    password_mgr = urllib2.HTTPPasswordMgrWithDefaultRealm()
                self.passwd = password_mgr
                self.add_password = self.passwd.add_password

        def http_request(self, req):
                uri = req.get_full_url()
                user, pw = self.passwd.find_user_password(None, uri)
                if pw is None:
                    return req

                raw = "%s:%s" % (user, pw)
                auth = 'Basic %s' % base64.b64encode(raw).strip()
                req.add_unredirected_header('Authorization', auth)
                return req

auth_handler = PreemptiveBasicAuthHandler()
auth_handler.add_password(realm=None,
```

(continues on next page)

(continued from previous page)

```
                          uri=url,
                          user='mytardis',
                          passwd='mytardis')
opener = urllib2.build_opener(auth_handler)
# ...and install it globally so it can be used with urlopen.
urllib2.install_opener(opener)
```

Querying the database (GET)

All endpoints support querying lists and individual records via GET requests. Some support more complex queries via GET parameters as well.

Creating objects, adding files (POST)

The creation of Experiments, Datasets and Dataset_Files via POSTs with the option to include metadata/parametersets has been implemented and tested.

The following examples demonstrate how to go about it.

In all except the file attachment case the POST data should be a JSON string, the `Content-Type` header needs to be set to `application/json` and the `Accept` header as well. Other response formats may be made available in the future.

In all cases the URI of the created object is returned in the `Location` header of the response.

Experiments

Example JSON input

```
{
  "title": "API-created Experiment #1",
  "description": "Wow, all automatic!",
  "institution_name": "Monash University",
  "parameter_sets": [
    {
      "schema": "http://institution.com/my/schema",
      "parameters": [
          {
            "name": "important_parameter_1",
            "value": "Test16"
          },
          {
            "name": "important_parameter_3",
            "value": "57.136"
          }
      ]
    },
    {
      "schema": "http://company.com/some/other/schema",
      "parameters": [
          {
            "name": "meaningful_name",
            "value": "Test17"
```

(continues on next page)

(continued from previous page)

```
          },
          {
            "name": "meaningless_name",
            "value": "1234"
          }
      ]
    }
  ]
}
```

This creates an experiment with two parametersets with two parameters each.

Alternative to Schema namespaces and Parameter names, you can also specify the URIs to each. Until the querying of Schemas and Parameters is documented this is discouraged.

Datasets

Example JSON input:

```
{
  "description": "API-created Dataset",
  "experiments": [
    "/api/v1/experiment/1/",
    "/api/v1/experiment/2/"
  ],
  "immutable": false,
  "parameter_sets": [
    {
      "parameters": [
        {
          "name": "obscure-instrument-setting-52",
          "value": "awesome dataset api POST"
        },
        {
          "name": "temperature",
          "value": "301"
        }
      ],
      "schema": "http://datasets.com/need/schemas/too"
    },
    {
      "parameters": [
        {
          "name": "someotherparameter",
          "value": "some other value"
        }
      ],
      "schema": "http://better-datasets.com/offers/better/schemas"
    }
  ]
}
```

DataFiles

There are three ways to add a file to MyTardis via the API.

Via multipart form POST

This works for single files at the moment.

The key is to send a multipart-form instead of 'application/json'. This can be accomplished with the requests library as shown in the following example.

To use requests you need to install it first, eg. `pip install requests`.

Also, for this to work, the POST data needs to be sent with the JSON string called `'json_data'` and the file called `'attached_file'`.

Example JSON input:

```
{
    "dataset": "/api/v1/dataset/1/",
    "filename": "mytestfile.txt",
    "md5sum": "c858d6319609d6db3c091b09783c479c",
    "size": "12",
    "mimetype": "text/plain",
    "parameter_sets": [{
        "schema": "http://datafileshop.com/fileinfo/v1",
        "parameters": [{
            "name": "fileparameter1",
            "value": "123"
        },
        {
            "name": "fileparameter2",
            "value": "1234"
        }]
    }]
}
```

Example requests script:

```
import requests
from requests.auth import HTTPBasicAuth

url = "http://localhost:8000/api/v1/dataset_file/"
headers = {'Accept': 'application/json'}
response = requests.post(url, data={"json_data": data}, headers=headers,
                         files={'attached_file': open(filename, 'rb')},
                         auth=HTTPBasicAuth(username, password)
                         )
```

Via staging location

Another way to add a file is to create the database entry first without providing a storage location. This will return back a location on the server that you are assumed to have access to. Once the file appears there, for example when you copy it there, it will be moved to its permanent storage location managed by MyTardis.

The full file path that you should copy/move the file to is returned as the content of the response.

Example JSON input:

```
{
    "dataset": "/api/v1/dataset/1/",
    "filename": "mytestfile.txt",
    "md5sum": "c858d6319609d6db3c091b09783c479c",
    "size": "12",
    "mimetype": "text/plain",
    "parameter_sets": [{
        "schema": "http://datafileshop.com/fileinfo/v1",
        "parameters": [{
            "name": "fileparameter1",
            "value": "123"
        },
        {
            "name": "fileparameter2",
            "value": "1234"
        }]
    }]
}
```

Via shared permanent storage location

This method assumes that there exists a storage that is shared between MyTardis and you. The registered file will remain in this location.

For this to work you need to get a `Location` (internal MyTardis settings) name to submit with your metadata.

Examples JSON:

```
{
   "dataset": "/api/v1/dataset/1/",
   "filename": "mytestfile.txt",
   "md5sum": "c858d6319609d6db3c091b09783c479c",
   "size": "12",
   "mimetype": "text/plain",
   "replicas": [{
       "url": "mytestfile.txt",
       "location": "local",
       "protocol": "file"
   }],
   "parameter_sets": [{
       "schema": "http://datafileshop.com/fileinfo/2",
       "parameters": [{
           "name": "fileparameter1",
           "value": "123"
       },
       {
           "name": "fileparameter2",
           "value": "123"
       }]
   }]
}
```

urllib2 POST example script

Replace `MODEL` with one of the available model names in lower case. `data` is the JSON as a string.

```
import urllib2
url = "http://localhost:8000/api/v1/MODEL/"
headers = {'Accept': 'application/json',
           'Content-Type': 'application/json'}
auth_handler = urllib2.HTTPBasicAuthHandler()
auth_handler.add_password(realm="django-tastypie",
                          uri=url,
                          user=username,
                          passwd=password)
opener = urllib2.build_opener(auth_handler)
urllib2.install_opener(opener)
myrequest = urllib2.Request(url=url, data=data,
                            headers=headers)
myrequest.get_method = lambda: 'POST'
output = "error"
output = urllib2.urlopen(myrequest)
print output.headers["Location"]
```

11.3.4 Tests

Running the Test Suite

Run this command to run the unit tests:

```
./test.py
```

If you want to speciy any options or specific tests to run, the test argument is required first:

    ./test.py test –some-argument

You can choose to run the test suite with different options (e.g. with coverage, with different verbosity, etc.). To see the full list of options, run the same command with the –help flag.

Running Individual Unit Tests

The unit tests reside in the `tardis/tardis_portal/tests directory`. To run the unit tests individually, you can use this command:

```
./test.py test <test_module_name_here>
```

Note that the test module name argument should be the relative file path with "." as folder separator. For example, if you want to run the test "test_authentication.py", then your command to execute this test would be:

```
./test.py test tardis.tardis_portal.tests.test_authentication
```

Other options

You can choose to include different options when running the unit tests (e.g. run in verbose mode, print out a traceback, etc.). Run the test or django test command with –help flag to see the the full list of options:

```
./test.py test --help
```

Running BDD tests

To run BDD tests with the default settings and a headless browser, call:

```
./test.py behave
```

To use another browser, e.g. Chrome, follow the steps to install "chromedriver" in http://splinter.readthedocs.io/en/latest/drivers/chrome.html and then call:

```
./test.py behave -D browser=chrome
```

11.3.5 tardis

tardis package

Subpackages

tardis.analytics package

Submodules

tardis.analytics.apps module

class tardis.analytics.apps.**AnalyticsConfig**(*app_name*, *app_module*)
Bases: django.apps.config.AppConfig

name = 'tardis.analytics'

verbose_name = 'Analytics framework'

tardis.analytics.ga module

Google analyitics tracking

tardis.analytics.ga.**track_download**(*label*, *session_id*, *ip*, *user*, *total_size=None*, *num_files=None*, *ua=None*)

tardis.analytics.ga.**track_login**(*label*, *session_id*, *ip*, *user*)

tardis.analytics.ga.**track_logout**(*label*, *session_id*, *ip*, *user*)

tardis.analytics.tracker module

Generic tracking and analytics interface Supports Google Analytics through ga.py, others may follow

class tardis.analytics.tracker.**IteratorTracker**(*iterator*, *tracker_data=None*)
Bases: object

wraps file iterator to track successful and incomplete downloads

close()

next()

tardis.analytics.tracker.**service = <module 'tardis.analytics.ga' from '/home/docs/checkouts**
This can become a setting to other service in the future

Module contents

tardis.apps package

Subpackages

tardis.apps.anzsrc_codes package

Subpackages

tardis.apps.anzsrc_codes.tests package

Submodules

tardis.apps.anzsrc_codes.tests.test_oaipmh module

class tardis.apps.anzsrc_codes.tests.test_oaipmh.**RifCSTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testExistsInOaipmh()

tardis.apps.anzsrc_codes.tests.tests module

class tardis.apps.anzsrc_codes.tests.tests.**TabTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testAccessWithReadPerms()

testAccessWithoutReadPerms()

Module contents

Submodules

tardis.apps.anzsrc_codes.forms module

class tardis.apps.anzsrc_codes.forms.**FoRCodeForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)

Bases: django.forms.forms.Form

base_fields = {'code': <django.forms.fields.CharField object at 0x7f11af4de110>, 'nam

declared_fields = {'code': <django.forms.fields.CharField object at 0x7f11af4de110>,

media

tardis.apps.anzsrc_codes.models module

tardis.apps.anzsrc_codes.settings module

tardis.apps.anzsrc_codes.urls module

tardis.apps.anzsrc_codes.views module

Module contents

tardis.apps.deep_storage_download_mapper package

Submodules

tardis.apps.deep_storage_download_mapper.mapper module

File mapper that works for files stored in deep directory structures. It recreates the structure as stored in the datafile directory

tardis.apps.deep_storage_download_mapper.mapper.**deep_storage_mapper**(*obj*, *root-dir=None*)

If rootdir is None, just return a filesystem-safe representation of the object, e.g. "DatasetDescription_123" or "strange %2F filename.txt"

For now, only DataFiles are supported when rootdir is not None.

Parameters

- **obj** (DataFile, Dataset *or* Experiment) – The model instance (DataFile, Dataset or Experiment) to generate a path for.
- **rootdir** (*basestring*) – The top-level directory name, or None

Returns Filesystem-safe path for the object in the archive or SFTP view.

Return type basestring

Raises

- **Exception** –
- **NotImplementedError** –

tardis.apps.deep_storage_download_mapper.urls module

Module contents

tardis.apps.dl_mapper_df_dir_only package

Submodules

tardis.apps.dl_mapper_df_dir_only.mapper module

File mapper that works for files stored in deep directory structures. It recreates the structure as stored in the datafile directory

tardis.apps.dl_mapper_df_dir_only.mapper.**df_dir_only**(*datafile*, *rootdir*)

tardis.apps.dl_mapper_df_dir_only.urls module

Module contents

tardis.apps.equipment package

Subpackages

tardis.apps.equipment.migrations package

Submodules

tardis.apps.equipment.migrations.0001_initial module

class tardis.apps.equipment.migrations.0001_initial.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0002_auto_20150528_1128')]

operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'ke

Module contents

tardis.apps.equipment.south_migrations package

Submodules

tardis.apps.equipment.south_migrations.0001_initial module

Module contents

Submodules

tardis.apps.equipment.admin module

tardis.apps.equipment.forms module

class tardis.apps.equipment.forms.**EquipmentSearchForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)

Bases: django.forms.forms.Form

base_fields = {'description': <django.forms.fields.CharField object at 0x7f11aeaf2410

declared_fields = {'description': <django.forms.fields.CharField object at 0x7f11aeaf

media

tardis.apps.equipment.models module

class tardis.apps.equipment.models.**Equipment**(*id*, *key*, *description*, *make*, *model*, *type*, *serial*, *comm*, *decomm*, *url*)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

dataset

objects = <django.db.models.manager.Manager object>

tardis.apps.equipment.tests module

tests.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.apps.equipment.tests.**EquipmentTestCase**(*methodName='runTest'*)

Bases: django.test.testcases.TestCase

fixtures = ['initial_data.json']

setUp()
Hook method for setting up the test fixture before exercising it.

testEquipmentDetail()

testEquipmentList()

testSearchEquipmentForm()

testSearchEquipmentResult()

tardis.apps.equipment.tests.**suite**()

tardis.apps.equipment.urls module

tardis.apps.equipment.views module

tardis.apps.equipment.views.**index**(*request*)

tardis.apps.equipment.views.**search**(*request*)

tardis.apps.equipment.views.**view_id**(*request*, *object_id*)

tardis.apps.equipment.views.**view_key**(*request*, *object_key*)

Module contents

tardis.apps.filepicker package

Submodules

tardis.apps.filepicker.filepicker_settings module

tardis.apps.filepicker.urls module

tardis.apps.filepicker.utils module

taken from https://raw.github.com/Filepicker/django-filepicker/master/django_filepicker/utils.py on 11 Apr 2013

class tardis.apps.filepicker.utils.**FilepickerFile**(*url*)
Bases: object

cleanup()
Removes any downloaded objects and closes open files.

filepicker_url_regex = <_sre.SRE_Pattern object>

get_file()
Downloads the file from filepicker.io and returns a Django File wrapper object

tardis.apps.filepicker.views module

Filepicker.io button view and upload handler

Module contents

tardis.apps.mx_views package

Submodules

tardis.apps.mx_views.urls module

tardis.apps.mx_views.views module

Module contents

tardis.apps.oaipmh package

Subpackages

tardis.apps.oaipmh.provider package

Submodules

tardis.apps.oaipmh.provider.base module

class tardis.apps.oaipmh.provider.base.**BaseProvider**(*site*)
Bases: oaipmh.interfaces.IOAI, object

A base provider which roughly implements the PyOAI interface for OAI-PMH servers.

Extend this if you're writing your own provider for a new type or a different metadata format.

getRecord(*metadataPrefix*, *identifier*)
Get a record for a metadataPrefix and identifier.

Parameters

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **identifier** (*string*) –
  - repository-unique identifier of record

Raises

- **oaipmh.error.CannotDisseminateFormatError** – if metadataPrefix is unknown or not supported by identifier.
- **oaipmh.error.IdDoesNotExistError** – if identifier is unknown or illegal.

Returns a header, metadata, about tuple describing the record.

identify()
Retrieve information about the repository.

Returns an Identify object describing the repository.

listIdentifiers(*metadataPrefix*, *set=None*, *from_=None*, *until=None*)
Get a list of header information on records.

Parameters

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **set** (*string*) – set identifier; only return headers in set
- **from** (*datetime*) – only retrieve headers from from_ date forward (in naive UTC)
- **until** (*datetime*) – only retrieve headers with dates up to and including until date (in naive UTC)

Raises

- **error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **error.NoSetHierarchyError** – if the repository does not support sets.

Returns an iterable of headers.

listMetadataFormats(*identifier=None*)

List metadata formats supported by repository or record.

Parameters **identifier** (*string*) – identify record for which we want to know all supported metadata formats. If absent, list all metadata formats supported by repository.

Raises

- **error.IdDoesNotExistError** – if record with identifier does not exist.
- **error.NoMetadataFormatsError** – if no formats are available for the indicated record.

Returns an iterable of `metadataPrefix`, `schema`, `metadataNamespace` tuples (each entry in the tuple is a string).

listRecords(*metadataPrefix*, *set=None*, *from_=None*, *until=None*)

Get a list of header, metadata and about information on records.

Parameters

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **set** (*string*) – set identifier; only return records in set
- **from** (*datetime*) – only retrieve records from `from_` date forward (in naive UTC)
- **until** (*datetime*) – only retrieve records with dates up to and including until date (in naive UTC)

Raises

- **oaipmh.error.CannotDisseminateFormatError** – if `metadataPrefix` is not supported by the repository.
- **oaipmh.error.NoSetHierarchyError** – if the repository does not support sets.

Returns an iterable of `header`, `metadata`, `about` tuples.

listSets()

Get a list of sets in the repository.

Raises **error.NoSetHierarchyError** – if the repository does not support sets.

Returns an iterable of setSpec, setName tuples (strings).

writeMetadata(*element*, *metadata*)

Create XML elements under the given element, using the provided metadata.

Should avoid doing any model-lookups, as they should be done when creating the metadata.

Parameters

- **element** (*lxml.etree.Element*) – element to put all content under (as SubElements)
- **metadata** (*oaipmh.common.Metadata*) – metadata to turn into XML

tardis.apps.oaipmh.provider.experiment module

Module contents

tardis.apps.oaipmh.tests package

Subpackages

tardis.apps.oaipmh.tests.provider package

Submodules

tardis.apps.oaipmh.tests.provider.test_base module

class tardis.apps.oaipmh.tests.provider.test_base.**BaseProviderTestCase**(*methodName='runTest'*)

Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testGetRecord()
Default behaviour should be to not handle the identifier.

testIdentify()
There can be only one provider that responds. By default, don't.

testListIdentifiers()
By default a provider cannot handle the given metadata prefix.

testListMetadataFormats()
By default a provider handles no metadata formats.

testListRecords()
By default a provider cannot handle the given metadata prefix.

testListSets()
By default a provider does not implement sets.

tardis.apps.oaipmh.tests.provider.test_experiment module

Module contents

Submodules

tardis.apps.oaipmh.tests.test_oai module

class tardis.apps.oaipmh.tests.test_oai.**EndpointTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testGetRecord()

testIdentify()

testListIdentifiers()

testListMetadataFormats()

testListRecords()

Module contents

Submodules

tardis.apps.oaipmh.models module

tardis.apps.oaipmh.server module

class tardis.apps.oaipmh.server.**ProxyingMetadataRegistry**(*providers*)
Bases: oaipmh.metadata.MetadataRegistry

A registry that only writes, and does so by proxying to Providers.

hasReader(*metadata_prefix*)

hasWriter(*metadata_prefix*)

readMetadata(*metadata_prefix*, *element*)
Turn XML into metadata object.

element - element to read in

returns - metadata object

registerReader(*metadata_prefix*, *reader*)

registerWriter(*metadata_prefix*, *writer*)

writeMetadata(*metadata_prefix*, *element*, *metadata*)
Write metadata as XML.

element - ElementTree element to write under metadata - metadata object to write

class tardis.apps.oaipmh.server.**ProxyingServer**(*providers*)
Bases: oaipmh.interfaces.IOAI

getRecord(*metadataPrefix*, *identifier*)
Get a record for a metadataPrefix and identifier.

Raises

- **oaipmh.error.CannotDisseminateFormatError** – if no provider returns a result, but at least one provider responds with oaipmh.error.CannotDisseminateFormatError (meaning the identifier exists)
- **oaipmh.error.IdDoesNotExistError** – if all providers fail with oaipmh.error.IdDoesNotExistError

Returns first successful provider response

Return type response

identify()
Retrieve information about the repository.

Returns an oaipmh.common.Identify object describing the repository.

Return type oaipmh.common.Identify

listIdentifiers(*metadataPrefix*, ***kwargs*)
Lists identifiers from all providers as a single set.

Raises

- **error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **error.NoSetHierarchyError** – if a set is provided, as the repository does not support sets.

Returns a set.Set of headers.

Return type set

listMetadataFormats(***kwargs*)
List metadata formats from all providers in a single set.

Raises

- **error.IdDoesNotExistError** – if record with identifier does not exist.
- **error.NoMetadataFormatsError** – if no formats are available for the indicated record, but it does exist.

Returns a *frozenset* of metadataPrefix, schema, metadataNamespace tuples (each entry in the tuple is a string).

Return type frozenset

listRecords(*metadataPrefix*, ***kwargs*)
Lists records from all providers as a single set.

Raises

- **error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **error.NoSetHierarchyError** – if a set is provided, as the repository does not support sets.

Returns a `set.Set` of `header`, `metadata`, `about` tuples.

Return type set

listSets()
List sets.

Raises **oaipmh.error.NoSetHierarchyError** – because set hierarchies are currrently not implemented

tardis.apps.oaipmh.server.**get_server**(*current_site*)

tardis.apps.oaipmh.urls module

tardis.apps.oaipmh.views module

tardis.apps.oaipmh.views.**endpoint**(*request*)

Module contents

tardis.apps.publication_forms package

Subpackages

tardis.apps.publication_forms.tests package

Submodules

tardis.apps.publication_forms.tests.test_email_templates module

class tardis.apps.publication_forms.tests.test_email_templates.**EmailTemplates**(*methodName='ru*
Bases: `django.test.testcases.TestCase`

test_pub_approved_template()

test_pub_awaiting_approval_template()

test_pub_rejected_template()

test_pub_released_template()

test_pub_requires_authorisation_template()

test_pub_reverted_to_draft_template()

Module contents

Submodules

tardis.apps.publication_forms.default_settings module

default settings for publication app override in main settings

tardis.apps.publication_forms.doi module

class tardis.apps.publication_forms.doi.**DOI**(*doi=None*)

activate(*doi=None*)

deactivate(*doi=None*)

mint(*experiment_id*, *uri*, *publisher='Monash University'*)

tardis.apps.publication_forms.email_text module

tardis.apps.publication_forms.email_text.**email_pub_approved**(*pub_title*, *pub_url*, *doi=None*, *message=None*)

tardis.apps.publication_forms.email_text.**email_pub_awaiting_approval**(*pub_title*)

tardis.apps.publication_forms.email_text.**email_pub_rejected**(*pub_title*, *message=None*)

tardis.apps.publication_forms.email_text.**email_pub_released**(*pub_title*, *doi=None*)

tardis.apps.publication_forms.email_text.**email_pub_requires_authorisation**(*user_name*, *pub_url*, *approvals_url*)

tardis.apps.publication_forms.email_text.**email_pub_reverted_to_draft**(*pub_title*, *message=None*)

tardis.apps.publication_forms.email_text.**interpolate_template**(*template_name*, ***kwargs*)

tardis.apps.publication_forms.pub_form_config module

Set up publication form schemas

class tardis.apps.publication_forms.pub_form_config.**PubFormConfig**

do_setup(*force=False*)

tardis.apps.publication_forms.tasks module

tardis.apps.publication_forms.tasks.**get_release_date**(*publication*)

tardis.apps.publication_forms.tasks.**has_pdb_embargo**(*publication*)

tardis.apps.publication_forms.tasks.**populate_pdb_pub_records**(**args*, ***kwargs*)

tardis.apps.publication_forms.tasks.**process_embargos**(**args*, ***kwargs*)

tardis.apps.publication_forms.urls module

tardis.apps.publication_forms.utils module

class tardis.apps.publication_forms.utils.**CifHelper**(*cif_url*)
Bases: object

as_list(*obj*)

get_cif_file_object()

class tardis.apps.publication_forms.utils.**PDBCifHelper**(*pdb_id*)
Bases: *tardis.apps.publication_forms.utils.CifHelper*

get_citations()

get_free_r_value()

get_obs_r_value()

get_pdb_id()

get_pdb_title()

get_pdb_url()

get_resolution()

get_sequence_info()

get_spacegroup()

get_unit_cell()

tardis.apps.publication_forms.utils.**check_pdb_status**(*pdb_id*)

tardis.apps.publication_forms.utils.**get_pub_admin_email_addresses**()

tardis.apps.publication_forms.utils.**get_site_admin_email**()

tardis.apps.publication_forms.utils.**get_unreleased_pdb_info**(*pdb_id*)

tardis.apps.publication_forms.utils.**send_mail_to_authors**(*publication*, *subject*, *message*, *fail_silently=False*)

tardis.apps.publication_forms.views module

tardis.apps.publication_forms.views.**approval_ajax**(*request*)

tardis.apps.publication_forms.views.**approval_view**(*request*, **args*, ***kwargs*)

tardis.apps.publication_forms.views.**approve_publication**(**args*, ***kwargs*)

tardis.apps.publication_forms.views.**clear_publication_metadata**(*publication*)

tardis.apps.publication_forms.views.**create_draft_publication**(*user*, *publication_title*, *publication_description*)

tardis.apps.publication_forms.views.**fetch_experiments_and_datasets**(*request*, **args*, ***kwargs*)

tardis.apps.publication_forms.views.**get_draft_publication**(*user*, *publication_id*)

tardis.apps.publication_forms.views.**get_licenses**()

tardis.apps.publication_forms.views.**get_publications_awaiting_approval**()

tardis.apps.publication_forms.views.**index**(*request*, **args*, ***kwargs*)

tardis.apps.publication_forms.views.**map_form_to_schemas**(*extraInfo*, *publication*)

tardis.apps.publication_forms.views.**pdb_helper**(*request*, *pdb_id*)

tardis.apps.publication_forms.views.**process_form**(*request*, **args*, ***kwargs*)

tardis.apps.publication_forms.views.**reject_publication**(*publication*, *message=None*)

tardis.apps.publication_forms.views.**remove_draft_status**(*publication*)

tardis.apps.publication_forms.views.**require_publication_admin**(*f*)

tardis.apps.publication_forms.views.**revert_publication_to_draft**(*publication*, *message=None*)

tardis.apps.publication_forms.views.**select_forms**(*datasets*)

tardis.apps.publication_forms.views.**set_embargo_release_date**(*publication*, *release_date*)

tardis.apps.publication_forms.views.**set_publication_authors**(*author_list*, *publication*)

tardis.apps.publication_forms.views.**synchrotron_search_epn**(*publication*)

Module contents

tardis.apps.push_to package

Subpackages

tardis.apps.push_to.migrations package

Submodules

tardis.apps.push_to.migrations.0001_initial module

class tardis.apps.push_to.migrations.0001_initial.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'auth', u'__first__'), (u'auth', u'0006_require_contenttypes_0002')]

operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'ke

tardis.apps.push_to.migrations.0002_auto_20160518_1953 module

class tardis.apps.push_to.migrations.0002_auto_20160518_1953.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'push_to', u'0001_initial')]

```
operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'key_type',
```

Module contents

tardis.apps.push_to.tests package

Submodules

tardis.apps.push_to.tests.test_models module

class tardis.apps.push_to.tests.test_models.**ModelsTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

test_credential_generation()

test_get_allowed_signing_services()

test_key_type_detection()

test_keys_from_private_key_only()

test_pkey_to_credential()

Module contents

Submodules

tardis.apps.push_to.apps module

class tardis.apps.push_to.apps.**PushToConfig**(*app_name*, *app_module*)
Bases: *tardis.app_config.AbstractTardisAppConfig*

name = 'tardis.apps.push_to'

verbose_name = 'Push To'

tardis.apps.push_to.exceptions module

exception tardis.apps.push_to.exceptions.**NoSuitableCredential**
Bases: exceptions.Exception

This exception is thrown when attempting to find a suitable credential for a remote host but none is found

tardis.apps.push_to.models module

class tardis.apps.push_to.models.**Credential**(**args*, ***kwargs*)
Bases: *tardis.apps.push_to.models.KeyPair*

A credential that may contain a password and/or key. The auth method chosen depends on the credentials available, allowed auth methods, and priorities defined by the SSH client.

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

static generate_keypair_credential(*tardis_user*, *remote_user*, *remote_hosts*, *bit_length=2048*)
Generates and saves an RSA key pair credential. Credentials returned by this method are intended to be registered on remote systems before being used. @type tardis_user: User @type remote_user: str @type bit_length: int @type remote_hosts: list[RemoteHost] :return: the generated credential :rtype: object

get_client_for_host(*remote_host*)
Attempts to establish a connection with the remote_host using this credential object. The remote_host may be any host, but only those in the remote_hosts field are expected to work. @type remote_host: .RemoteHost :return: a connected SSH client :rtype: SSHClient

static get_suitable_credential(*tardis_user*, *remote_host*, *remote_user=None*)

objects = <django.db.models.manager.Manager object>

remote_hosts

user

verify_remote_access(*remote_host=None*)
@type remote_host: RemoteHost

class tardis.apps.push_to.models.**CredentialAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

form
alias of *CredentialForm*

media

class tardis.apps.push_to.models.**CredentialForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)
Bases: django.forms.models.ModelForm

class Meta

fields = '__all__'

model
alias of *Credential*

widgets = {'password': <django.forms.widgets.PasswordInput object at 0x7f11b210cf9

base_fields = {'key_type': <django.forms.fields.CharField object at 0x7f11b33adb50>,

declared_fields = {}

media

class tardis.apps.push_to.models.**DBHostKeyPolicy**
Bases: paramiko.client.MissingHostKeyPolicy

Host key verification policy based on the host key stored in the database.

missing_host_key(*client*, *hostname*, *key*)
@type key: PKey

class tardis.apps.push_to.models.**KeyPair**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

A key pair

class Meta

abstract = False

key

Returns a subclass of PKey of the appropriate key type

Return type PKey

Raises **ValidationError** –

save(**args*, ***kwargs*)

class tardis.apps.push_to.models.**OAuthSSHCertSigningService**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

Connection parameters for an OAuth2 SSH certificate signing service. Supports certificate signing server available here: https://github.com/monash-merc/ssh-authz

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

allowed_groups

allowed_remote_hosts

allowed_users

static get_available_signing_services(*user*)
Gets all SSH cert signing services available for a given user :param User user: User :return: allowed signing services :rtype: User

static get_oauth_service(*user*, *service_id*)
@type user: User @type service_id: int

objects = <django.db.models.manager.Manager object>

class tardis.apps.push_to.models.**RemoteHost**(**args*, ***kwargs*)
Bases: *tardis.apps.push_to.models.KeyPair*

A remote host that may be connected to via SSH

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

administrator

credential_set

oauthsshcertsigningservice_set

objects = <django.db.models.manager.Manager object>

class tardis.apps.push_to.models.**RemoteHostAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

Hides the private key field, which is not necessary for host keys

fields = ['nickname', 'administrator', 'host_name', 'port', 'key_type', 'public_key',

media

tardis.apps.push_to.oauth_tokens module

tardis.apps.push_to.oauth_tokens.**get_token**(*request*, *oauth_service*)
Returns the OAuth2 token from the current session :param Request request: django session object :param OAuthSSHCertSigningService oauth_service: an

OAuthSSHCertSigningService object

Returns

Return type string

tardis.apps.push_to.oauth_tokens.**get_token_data**(*oauth_service*, *token*)
Gets the OAuth2 user attributes using the supplied token :param OAuthSSHCertSigningService oauth_service: an

OAuthSSHCertSigningService object

Parameters **token** (*basestring*) – an OAuth2 token

Returns a json object of user attributes

Return type dict

tardis.apps.push_to.oauth_tokens.**set_token**(*request*, *oauth_service*, *token*)
Stores the OAuth2 token in the current session :param Request request: django request object :param OAuthSSHCertSigningService oauth_service: an

OAuthSSHCertSigningService object

Parameters **token** (*basestring*) – the OAuth2 token

tardis.apps.push_to.ssh_authz module

tardis.apps.push_to.ssh_authz.**sign_certificate**(*credential*, *token*, *url*)
An interface to the OAuth2 SSH certificate signing service @type credential: models.Credential

tardis.apps.push_to.tasks module

tardis.apps.push_to.urls module

tardis.apps.push_to.utils module

tardis.apps.push_to.utils.**bytes_available**(*ssh_client*, *path*)

tardis.apps.push_to.utils.**can_copy**(*ssh_client*, *object_type*, *object_id*, *path*)

tardis.apps.push_to.utils.**get_default_push_location**(*sftp_client*)

tardis.apps.push_to.utils.**get_object_size**(*type*, *id*)

tardis.apps.push_to.utils.**is_directory**(*sftp_client*, *path*)

tardis.apps.push_to.utils.**list_subdirectories**(*sftp_client*, *path*, *show_hidden=False*)

tardis.apps.push_to.utils.**shell_escape**(*s*)

tardis.apps.push_to.views module

Module contents

tardis.apps.related_info package

Subpackages

tardis.apps.related_info.tests package

Submodules

tardis.apps.related_info.tests.test_oaipmh module

class tardis.apps.related_info.tests.test_oaipmh.**RifCSTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TransactionTestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testExistsInOaipmh()

tardis.apps.related_info.tests.tests module

class tardis.apps.related_info.tests.tests.**CreateTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TransactionTestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testCanCreate()

testDetectsBadInput()

testMustHaveWrite()

class tardis.apps.related_info.tests.tests.**DeleteTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TransactionTestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testCanDelete()

testMustHaveWrite()

class tardis.apps.related_info.tests.tests.**GetTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TransactionTestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testHandlesFound()

testHandlesNotFound()

class tardis.apps.related_info.tests.tests.**ListTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TransactionTestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testHandlesEmptySet()

testHandlesMultipleEntries()

testHandlesSingleEntry()

class tardis.apps.related_info.tests.tests.**TabTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testAccessWithReadPerms()

testAccessWithoutReadPerms()

class tardis.apps.related_info.tests.tests.**UpdateTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TransactionTestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testDetectsBadInput()

testMustHaveWrite()

Module contents

Submodules

tardis.apps.related_info.forms module

class tardis.apps.related_info.forms.**RelatedInfoForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)
Bases: django.forms.forms.Form

base_fields = {'identifier': <django.forms.fields.URLField object at 0x7f11ae78bdd0>,

declared_fields = {'identifier': <django.forms.fields.URLField object at 0x7f11ae78bd

media

tardis.apps.related_info.models module

tardis.apps.related_info.settings module

tardis.apps.related_info.urls module

tardis.apps.related_info.views module

Module contents

tardis.apps.slideshow_view package

Submodules

tardis.apps.slideshow_view.urls module

tardis.apps.slideshow_view.views module

Module contents

Module contents

tardis.default_settings package

Submodules

tardis.default_settings.admins module

tardis.default_settings.analytics module

tardis.default_settings.apps module

tardis.default_settings.auth module

tardis.default_settings.auth.**AUTOGENERATE_API_KEY = False**
Generate a tastypie API key with user post_save (tardis/tardis_portal/models/hooks.py)

tardis.default_settings.auth.**REGISTRATION_OPEN = True**
Enable/disable the self-registration link and form in the UI. Note - this does not actually disable the URL endpoints for registration. You must also remove the *registration* app from *INSTALLED_APPS* to disable registration.

tardis.default_settings.caches module

tardis.default_settings.caches.**CACHES = {'celery-locks': {'LOCATION': 'celery_lock_cache'**
change the CACHES setting to memcached if you prefer. Requires additional dependencies.

tardis.default_settings.celery module

tardis.default_settings.celery.**BROKER_URL = 'django://'**
use django:, add kombu.transport.django to INSTALLED_APPS or use redis: install redis separately and add the following to a custom buildout.cfg:

django-celery-with-redis redis hiredis

tardis.default_settings.custom_views module

tardis.default_settings.custom_views.**DATASET_VIEWS = []**
Dataset view overrides ('contextual views') are specified as tuples mapping a Schema namespace to a class-based view (or view function). See: https://mytardis.readthedocs.org/en/develop/contextual_views.html#dataset-and-experiment-views

eg:

```
DATASET_VIEWS = [
    ('http://example.org/schemas/dataset/my_awesome_schema',
     'tardis.apps.my_awesome_app.views.CustomDatasetViewSubclass'),
]
```

tardis.default_settings.custom_views.**EXPERIMENT_VIEWS = []**
Experiment view overrides ('contextual views') are specified as tuples mapping a Schema namespace to a class-based view (or view function). See: https://mytardis.readthedocs.org/en/develop/contextual_views.html#dataset-and-experiment-views

eg:

```
EXPERIMENT_VIEWS = [
    ('http://example.org/schemas/expt/my_awesome_schema',
     'tardis.apps.my_awesome_app.views.CustomExptViewSubclass'),
]
```

tardis.default_settings.custom_views.**INDEX_VIEWS = {}**
A custom index page override is defined in as dictionary mapping a class-based view (or view function) to a Django `Site`, specified by SITE_ID (an integer) or the domain name of the incoming request. See: https://mytardis.readthedocs.org/en/develop/contextual_views.html#custom-index-view

eg:

```
INDEX_VIEWS = {
    1: 'tardis.apps.my_custom_app.views.MyCustomIndexSubclass',
    'store.example.com': 'tardis.apps.myapp.AnotherCustomIndexSubclass'
}
```

tardis.default_settings.database module

tardis.default_settings.debug module

tardis.default_settings.debug.**ALLOWED_HOSTS = ['*']**
For security reasons this needs to be set to your hostname and/or IP address in production.

tardis.default_settings.debug.**DEBUG = True**
Set to false for production use

tardis.default_settings.downloads module

tardis.default_settings.downloads.**DEFAULT_ARCHIVE_FORMATS = ['tar']**
Site's preferred archive types, with the most preferred first other available option: 'tgz'. Add to list if desired

tardis.default_settings.email module

tardis.default_settings.filters module

tardis.default_settings.frontend module

tardis.default_settings.frontend.**BLEACH_ALLOWED_ATTRIBUTES = {'a': ['href', 'title'], 'ab**
These are the default bleach values and shown here as an example.

tardis.default_settings.frontend.**BLEACH_ALLOWED_TAGS = ['a', 'abbr', 'acronym', 'b', 'bloc**
These are the default bleach values and shown here as an example.

tardis.default_settings.i18n module

tardis.default_settings.localisation module

tardis.default_settings.logging module

tardis.default_settings.middlewares module

tardis.default_settings.publication module

tardis.default_settings.search module

tardis.default_settings.sftp module

tardis.default_settings.sftp.**SFTP_USERNAME_ATTRIBUTE = 'email'**
The attribute from the User model ('email' or 'username') used to generate the SFTP login example on the sftp_access help page.

tardis.default_settings.sharing module

tardis.default_settings.site_customisations module

tardis.default_settings.site_customisations.**SITE_TITLE = 'MyTardis'**
customise the title of your site

tardis.default_settings.site_customisations.**SPONSORED_TEXT = None**
add text to the footer to acknowledge someone

tardis.default_settings.staging module

tardis.default_settings.static_files module

tardis.default_settings.static_files.**get_admin_media_path**()

tardis.default_settings.storage module

tardis.default_settings.storage.**METADATA_STORE_PATH = '/home/docs/checkouts/readthedocs.org**
storage path for image paths stored in parameters. Better to set to another location if possible

tardis.default_settings.templates module

tardis.default_settings.uploads module

tardis.default_settings.uploads.**UPLOAD_METHOD = False**
Old version: UPLOAD_METHOD = "uploadify". This can be changed to an app that provides an upload_button function, eg. "tardis.apps.filepicker.views.upload_button" to use a fancy commercial uploader. To use filepicker, please also get an API key at http://filepicker.io

tardis.default_settings.urls module

Module contents

tardis.default_settings.**get_git_version**()

tardis.search package

Submodules

tardis.search.forms module

class tardis.search.forms.**GroupedSearchForm**(**args*, ***kwargs*)
Bases: haystack.forms.ModelSearchForm

base_fields = {'q': <django.forms.fields.CharField object at 0x7f11b05c3610>}

declared_fields = {'q': <django.forms.fields.CharField object at 0x7f11b05c3610>}

media

search(**args*, ***kwargs*)

tardis.search.search_indexes module

search indexes for single search

class tardis.search.search_indexes.**DataFileIndex**
Bases: haystack.indexes.SearchIndex, haystack.constants.Indexable

datafile_filename = <haystack.fields.CharField object>

dataset_id_stored = <haystack.fields.IntegerField object>

experiment_id_stored = <haystack.fields.MultiValueField object>

fields = {'datafile_filename': <haystack.fields.CharField object at 0x7f11ae710610>,

get_model()

objects = <haystack.manager.SearchIndexManager object>

prepare_experiment_id_stored(*obj*)

prepare_text(*obj*)

text = <haystack.fields.CharField object>

class tardis.search.search_indexes.**DatasetIndex**

Bases: haystack.indexes.SearchIndex, haystack.constants.Indexable

dataset_description = <haystack.fields.CharField object>

dataset_id_stored = <haystack.fields.IntegerField object>

experiment_id_stored = <haystack.fields.MultiValueField object>

fields = {'dataset_description': <haystack.fields.CharField object at 0x7f11ae710550>

get_model()

objects = <haystack.manager.SearchIndexManager object>

prepare_experiment_id_stored(*obj*)

prepare_text(*obj*)

text = <haystack.fields.CharField object>

class tardis.search.search_indexes.**ExperimentIndex**

Bases: haystack.indexes.SearchIndex, haystack.constants.Indexable

experiment_author = <haystack.fields.MultiValueField object>

experiment_created_time = <haystack.fields.DateTimeField object>

experiment_creator = <haystack.fields.CharField object>

experiment_description = <haystack.fields.CharField object>

experiment_end_time = <haystack.fields.DateTimeField object>

experiment_id_stored = <haystack.fields.IntegerField object>

experiment_institution_name = <haystack.fields.CharField object>

experiment_start_time = <haystack.fields.DateTimeField object>

experiment_title = <haystack.fields.CharField object>

experiment_update_time = <haystack.fields.DateTimeField object>

fields = {'experiment_author': <haystack.fields.MultiValueField object at 0x7f11ae710

get_model()

index_queryset(*using=None*)

Used when the entire index for model is updated.

objects = <haystack.manager.SearchIndexManager object>

prepare_experimentauthor(*obj*)

prepare_text(*obj*)

text = <haystack.fields.CharField object>

tardis.search.urls module

tardis.search.utils module

class tardis.search.utils.**SearchQueryString**(*query_string*)
Bases: object

Class to manage switching between space separated search queries and '+' separated search queries (for addition to urls

TODO This would probably be better handled with filters

query_string()

url_safe_query()

tardis.search.views module

Module contents

tardis.tardis_portal package

Subpackages

tardis.tardis_portal.auth package

Submodules

tardis.tardis_portal.auth.authentication module

A module containing helper methods for the manage_auth_methods function in views.py.

tardis.tardis_portal.auth.authentication.**add_auth_method**(*request*)
Add a new authentication method to request.user's existing list of authentication methods. This method will ask for a confirmation if the user wants to merge two accounts if the authentication method he provided already exists as a method for another user.

Parameters **request** (*Request*) – the HTTP request object

Returns The HttpResponse which contains request.user's new list of authentication methods

Return type HttpResponse

tardis.tardis_portal.auth.authentication.**edit_auth_method**(*request*)
Change the local DB (Django) password for request.user.

tardis.tardis_portal.auth.authentication.**list_auth_methods**(*request*)
Generate a list of authentication methods that request.user uses to authenticate to the system and send it back in a HttpResponse.

Parameters **request** (*Request*) – the HTTP request object

Returns The HttpResponse which contains request.user's list of authentication methods

Return type HttpResponse

tardis.tardis_portal.auth.authentication.**merge_auth_method**(*request*)
Merge the account that the user is logged in as and the account that he provided in the Authentication Form. Merging accounts involve relinking the UserAuthentication table entries, transferring ObjectACL entries to the merged account, changing the Group memberships and deleting the unneeded account.

Parameters **request** (*Request*) – the HTTP request object

Returns The HttpResponse which contains request.user's new list of authentication methods

Return type HttpResponse

tardis.tardis_portal.auth.authentication.**remove_auth_method**(*request*)
Removes the non-local DB auth method from the UserAuthentication model.

Parameters **request** (*Request*) – the HTTP request object

Returns The HttpResponse which contains request.user's new list of authentication methods

Return type HttpResponse

tardis.tardis_portal.auth.authorisation module

Object-level authorisation backend

class tardis.tardis_portal.auth.authorisation.**ACLAwareBackend**
Bases: object

app_label = 'tardis_acls'

authenticate(*username*, *password*)
do not use this backend for authentication

get_perm_bool(*verb*)
relates ACLs to permissions

has_perm(*user_obj*, *perm*, *obj=None*)
main method, calls other methods based on permission type queried

supports_anonymous_user = True

supports_object_permissions = True

tardis.tardis_portal.auth.authservice module

models.py

class tardis.tardis_portal.auth.authservice.**AuthService**(*settings=<django.conf.LazySettings object>*)
The AuthService provides an interface for querying the auth(n|z) framework within MyTardis. The auth service works by reading the class path to plugins from the settings file.

Parameters **settings** (django.conf.settings) – the settings object that contains the list of user and group plugins.

authenticate(*authMethod*, ***credentials*)
Try and authenticate the user using the auth type he/she specified to use and if authentication didn't work using that

Parameters

- **authMethod** (*string*) – the shortname of the auth method.
- **credentials** (*kwargs*) – the credentials as expected by the auth plugin

Returns authenticated User or None

Return type User or None

getGroups(*user*)

Parameters **user** (*User*) – User

Returns a list of tuples containing pluginname and group id

Return type list

getGroupsForEntity(*entity*)
Return a list of the groups an entity belongs to

Parameters **entity** (*string*) – the entity to earch for, user or group.

Returns groups

Return type Group

The groups will be reurned as a list similar to:

```
[{'name': 'Group 456', 'id': '2'},
{'name': 'Group 123', 'id': '1'}]
```

getUser(*authMethod*, *user_id*, *force_user_create=False*)
Return a user model based on the given auth method and user id.

This function is responsible for creating the user within the Django DB and returning the resulting user model.

getUsernameByEmail(*authMethod*, *email*)
Return a username given the auth method and email address of a user.

get_or_create_user(*user_obj_or_dict*, *authMethod=None*)
refactored out for external use by AAF and possibly others

searchEntities(*filter*)
Return a list of users and/or groups

searchGroups(***kw*)
basestring id: the value of the id to search for basestring name: the value of the displayname to search for int max_results: the maximum number of elements to return basestring sort_by: the attribute the users should be sorted on basestring plugin: restrict the search to the specific group

provider

returns a list of users and/or groups rtype list

searchUsers(*filter*)
Return a list of users and/or groups

tardis.tardis_portal.auth.decorators module

tardis.tardis_portal.auth.decorators.**datafile_access_required**(*f*)

tardis.tardis_portal.auth.decorators.**dataset_access_required**(*f*)

tardis.tardis_portal.auth.decorators.**dataset_download_required**(*f*)

tardis.tardis_portal.auth.decorators.**dataset_write_permissions_required**(*f*)

tardis.tardis_portal.auth.decorators.**delete_permissions_required**(*f*)

tardis.tardis_portal.auth.decorators.**experiment_access_required**(*f*)

tardis.tardis_portal.auth.decorators.**experiment_download_required**(*f*)

tardis.tardis_portal.auth.decorators.**experiment_ownership_required**(*f*)

A decorator for Django views that validates if a user is an owner of an experiment or 'superuser' prior to further processing the request. Unauthenticated requests are redirected to the login page. If the user making the request satisfies none of these criteria, an error response is returned.

Parameters **f** (*types.FunctionType*) – A Django view function

Returns A Django view function

Return type types.FunctionType

tardis.tardis_portal.auth.decorators.**get_accessible_datafiles_for_user**(*request*)

tardis.tardis_portal.auth.decorators.**get_accessible_experiments**(*request*)

tardis.tardis_portal.auth.decorators.**get_accessible_experiments_for_dataset**(*request*, *dataset_id*)

tardis.tardis_portal.auth.decorators.**get_owned_experiments**(*request*)

tardis.tardis_portal.auth.decorators.**get_shared_experiments**(*request*)

tardis.tardis_portal.auth.decorators.**group_ownership_required**(*f*)

A decorator for Django views that validates if a user is a group admin or 'superuser' prior to further processing the request. Unauthenticated requests are redirected to the login page. If the user making the request satisfies none of these criteria, an error response is returned.

Parameters **f** (*types.FunctionType*) – A Django view function

Returns A Django view function

Return type types.FunctionType

tardis.tardis_portal.auth.decorators.**has_datafile_access**(*request*, *datafile_id*)

tardis.tardis_portal.auth.decorators.**has_datafile_download_access**(*request*, *datafile_id*)

tardis.tardis_portal.auth.decorators.**has_dataset_access**(*request*, *dataset_id*)

tardis.tardis_portal.auth.decorators.**has_dataset_download_access**(*request*, *dataset_id*)

tardis.tardis_portal.auth.decorators.**has_dataset_ownership**(*request*, *dataset_id*)

tardis.tardis_portal.auth.decorators.**has_dataset_write**(*request*, *dataset_id*)

tardis.tardis_portal.auth.decorators.**has_delete_permissions**(*request*, *experiment_id*)

tardis.tardis_portal.auth.decorators.**has_experiment_access**(*request*, *experiment_id*)

tardis.tardis_portal.auth.decorators.**has_experiment_download_access**(*request*, *experiment_id*)

tardis.tardis_portal.auth.decorators.**has_experiment_ownership**(*request*, *experiment_id*)

tardis.tardis_portal.auth.decorators.**has_experiment_write**(*request*, *experiment_id*)

tardis.tardis_portal.auth.decorators.**has_read_or_owner_ACL**(*request*, *experiment_id*)

Check whether the user has read access to the experiment - this means either they have been granted read access, or that they are the owner.

NOTE: This does not check whether the experiment is public or not, which means even when the experiment is public, this method does not automatically returns true.

As such, this method should NOT be used to check whether the user has general read permission.

tardis.tardis_portal.auth.decorators.**has_write_permissions**(*request*, *experiment_id*)

tardis.tardis_portal.auth.decorators.**is_group_admin**(*request*, **args*, ***kwargs*)

tardis.tardis_portal.auth.decorators.**upload_auth**(*f*)

tardis.tardis_portal.auth.decorators.**write_permissions_required**(*f*)

tardis.tardis_portal.auth.fix_circular module

tardis.tardis_portal.auth.fix_circular.**getGroups**(*user*)

tardis.tardis_portal.auth.httpbasicendpoint_auth module

Created on Dec 15, 2011

@author: uqtdettr

class tardis.tardis_portal.auth.httpbasicendpoint_auth.**HttpBasicEndpointAuth**(*openerDirector=<u instance>*, *endpoint=None*)

Bases: *tardis.tardis_portal.auth.interfaces.AuthProvider*

This class provides authentication against a HTTP resource protected by HTTP Basic authentication. Access is granted based on the user credentials being valid against that resource.

class SimplePasswordMgr

Bases: urllib2.HTTPPasswordMgr

Simple password manager which provides the same credentials, no matter the realm or the uri.

add_password(*realm*, *uri*, *username*, *password*)

clear()

find_user_password(*realm*, *authuri*)

authenticate(*request*)
Authenticate a user, expecting the user will be using form-based auth and the *username* and *password* will be passed in url-encoded form **POST** variables.

Parameters **request** (`django.http.HttpRequest`) – a HTTP Request instance

Returns authenticated user

Return type User

get_user(*user_id*)

tardis.tardis_portal.auth.interfaces module

class tardis.tardis_portal.auth.interfaces.**AuthProvider**

authenticate(*request*)
from a request authenticate try to authenticate the user. return a user dict if successful.

getUsernameByEmail(*email*)
returns the username (format string) from the auth domain

Implementing this function is optional- it is needed for resolving experiment owner email addresses to usernames during ingestion.

get_user(*user_id*)

class tardis.tardis_portal.auth.interfaces.**GroupProvider**

getGroupById(*id*)
return the group associated with the id

getGroups(*user*)
return an iteration of the available groups.

getGroupsForEntity(*id*)
return a list of groups associated with a particular entity id

searchGroups(***filter*)
return a list of groups that match the filter

class tardis.tardis_portal.auth.interfaces.**UserProvider**

getUserById(*id*)
return the user dictionary in the format of:

```
{"id": 123,
"first_name": "John",
"last_name": "Smith",
"email": "john@example.com"}
```

getUsernameByEmail(*email*)
returns the username (format string) from the auth domain needed for resolving experiment owners during ingestion

searchUsers(***filter*)
return a list of user descriptions from the auth domain.

each user is in the format of:

```
{"id": 123,
"first_name": "John",
"last_name": "Smith",
"email": "john@example.com"}
```

tardis.tardis_portal.auth.ldap_auth module

tardis.tardis_portal.auth.localdb_auth module

Local DB Authentication module.

class tardis.tardis_portal.auth.localdb_auth.**DjangoAuthBackend**
Bases: *tardis.tardis_portal.auth.interfaces.AuthProvider*

Authenticate against Django's Model Backend.

authenticate(*request*)
authenticate a user, this expect the user will be using form based auth and the *username* and *password* will be passed in as **POST** variables.

Parameters **request** (django.http.HttpRequest) – a HTTP Request instance

Returns authenticated User

Return type User

get_user(*user_id*)

class tardis.tardis_portal.auth.localdb_auth.**DjangoGroupProvider**
Bases: *tardis.tardis_portal.auth.interfaces.GroupProvider*

getGroupById(*id*)
return the group associated with the id:

```
{"id": 123,
```

"display": "Group Name",}

getGroups(*user*)
return an iteration of the available groups.

name = u'django_group'

searchGroups(***filter*)

class tardis.tardis_portal.auth.localdb_auth.**DjangoUserProvider**
Bases: *tardis.tardis_portal.auth.interfaces.UserProvider*

getUserById(*id*)
return the user dictionary in the format of:

```
{"id": 123,
"first_name": "John",
"last_name": "Smith",
"email": "john@example.com"}
```

name = u'django_user'

tardis.tardis_portal.auth.token_auth module

token authentication module

class tardis.tardis_portal.auth.token_auth.**TokenAuthMiddleware**
Bases: object

adds tokens to the user object and the session from a GET query

process_request(*request*)

class tardis.tardis_portal.auth.token_auth.**TokenGroupProvider**
Bases: *tardis.tardis_portal.auth.interfaces.GroupProvider*

Transforms tokens into auth groups

getGroups(*user*)

name = u'token_group'

searchGroups(***kwargs*)
return nothing because these are not groups in the standard sense

tardis.tardis_portal.auth.utils module

Created on 15/03/2011

@author: gerson

tardis.tardis_portal.auth.utils.**configure_user**(*user*)
Configure a user account that has just been created by adding the user to the default groups and creating a UserProfile.

Parameters **user** (*User*) – the User instance for the newly created account

Returns User profile for user

Return type *UserProfile*

tardis.tardis_portal.auth.utils.**create_user**(*auth_method*, *user_id*, *email=''*)

tardis.tardis_portal.auth.utils.**get_or_create_user**(*auth_method*, *user_id*, *email=''*)

Module contents

tardis.tardis_portal.filters package

Submodules

tardis.tardis_portal.filters.diffractionimage module

diffractionimage.py

class tardis.tardis_portal.filters.diffractionimage.**DiffractionImageFilter**(*name*, *schema*, *diffdump_path*, *diff2jpeg_path*, *tagsToFind=[]*, *tagsToExclude=[]*)

Bases: object

This filter runs the CCP4 diffdump binary on a diffraction image and collects its output into the trddatafile schema

If a white list is specified then it takes precidence and all other tags will be ignored.

param name: the short name of the schema. type name: string param schema: the name of the schema to load the EXIF data into. type schema: string param tagsToFind: a list of the tags to include. type tagsToFind: list of strings param tagsToExclude: a list of the tags to exclude. type tagsToExclude: list of strings

getDiffractionImageMetadata(*filename*)
Return a dictionary of the metadata.

getDiffractionPreviewImage(*filename*)
Return a base64 encoded preview image.

getParameters(*schema*, *metadata*)
Return a list of the paramaters that will be saved.

getSchema()
Return the schema object that the paramaterset will use.

output_beamcenter(*term*, *value*)

output_detectordistance(*term*, *value*)

output_exposuretime(*term*, *value*)

output_imagesize(*term*, *value*)

output_metadata(*term*, *value*)

output_oscillation(*term*, *value*)

output_pixelsize(*term*, *value*)

output_twotheta(*term*, *value*)

output_wavelength(*term*, *value*)

parse_output(*output*)

parse_term(*line*)

parse_value(*line*)

run_diff2jpeg(*filename*)

run_diffdump(*file_path*)

saveDiffractionImageMetadata(*instance*, *schema*, *metadata*)
Save all the metadata to a DataFiles paramamter set.

split_oscillation(*terms*, *value*)

split_output(*terms*, *value*, *strip*)

tardis.tardis_portal.filters.diffractionimage.**make_filter**(*name=''*, *schema=''*, *tagsToFind=[]*, *tagsToExclude=[]*)

This filter runs the CCP4 diffdump binary on a diffraction image and collects its output into the trddatafile schema

If a white list is specified then it takes precidence and all other tags will be ignored.

param name: the short name of the schema. type name: string param schema: the name of the schema to load the EXIF data into. type schema: string param tagsToFind: a list of the tags to include. type tagsToFind: list of strings param tagsToExclude: a list of the tags to exclude. type tagsToExclude: list of strings

tardis.tardis_portal.filters.jeolsem module

class tardis.tardis_portal.filters.jeolsem.**JEOLSEMFilter**

Bases: object

This filter collects metadata from JEOL SEM text files.

param name: the short name of the schema. type name: string param schema: the name of the schema to load the EXIF data into. type schema: string

ATTR_PREFIXES = ('$CM_', '$$SM_')

SCHEMA = 'http://www.jeol.com/#jeol-sem-schema'

get_file_contents(*datafile*)

get_metadata(*schema*, *filedata*)

is_already_processed(*datafile*)

is_jeol_sem_metadata(*filedata*)

is_text_file(*datafile*)

save_metadata(*datafile*, *schema*, *metadata*)

Module contents

__init__.py

class tardis.tardis_portal.filters.**FilterInitMiddleware**(*filters=None*)

Bases: object

tardis.tardis_portal.management package

Subpackages

tardis.tardis_portal.management.commands package

Submodules

tardis.tardis_portal.management.commands.backupdb module

Command for backup database

class tardis.tardis_portal.management.commands.backupdb.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

backup()

do_mysql_backup(*outfile*)

do_mysql_restore(*infile*)

do_postgresql_backup(*outfile*)

do_postgresql_restore(*infile*)

handle(**args*, ***options*)

help = 'Backup database. Only Mysql and Postgresql engines are implemented'

option_list = (<Option at 0x7f11aab28908: -r/--restore>, <Option at 0x7f11aaf95a28:

restore(*infile*)

tardis.tardis_portal.management.commands.checkhashes module

Management utility to create a token user

exception tardis.tardis_portal.management.commands.checkhashes.**AlgorithmNotSupported**

Bases: exceptions.Exception

class tardis.tardis_portal.management.commands.checkhashes.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

handle(**args*, ***options*)

help = 'Used to check file hashes against actual files.'

exception tardis.tardis_portal.management.commands.checkhashes.**NoHashAvailable**

Bases: exceptions.Exception

exception tardis.tardis_portal.management.commands.checkhashes.**ValidationFailed**

Bases: exceptions.Exception

tardis.tardis_portal.management.commands.checkhashes.**validate_digest**(*algorithm*, *datafile*)

tardis.tardis_portal.management.commands.chownexperiment module

Management command to change the owner of some experiments

class tardis.tardis_portal.management.commands.chownexperiment.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

args = '<user> <experiment id>...'

handle(**args*, ***options*)

help = 'Change the owner of the experiments.'

option_list = (<Option at 0x7f11aab21950: -n/--dryRun>, <Option at 0x7f11ab0f6c68: -

tardis.tardis_portal.management.commands.cleanuptokens module

Management utility to clean up tokens

class tardis.tardis_portal.management.commands.cleanuptokens.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

handle(**args*, ***options*)

help = 'Deletes unused tokens and optionally their ACLs'

option_list = (<Option at 0x7f11ab1e2098: --keep-acls>,)

tardis.tardis_portal.management.commands.createmysuperuser module

Management utility to create superusers.

class tardis.tardis_portal.management.commands.createmysuperuser.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

handle(**args*, ***options*)

help = 'Used to create a MyTardis superuser.'

option_list = (<Option at 0x7f11ab1df248: --username>, <Option at 0x7f11aab737a0: --

tardis.tardis_portal.management.commands.createmysuperuser.**is_valid_email**(*value*)

tardis.tardis_portal.management.commands.createtokenuser module

Management utility to create a token user

class tardis.tardis_portal.management.commands.createtokenuser.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

handle(**args*, ***options*)

help = 'Used to create a MyTardis tokenuser.'

option_list = (<Option at 0x7f11aabac758: --username>,)

tardis.tardis_portal.management.commands.createuser module

Management utility to create regular users.

class tardis.tardis_portal.management.commands.createuser.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

handle(**args*, ***options*)

help = 'Used to create a MyTardis user.'

option_list = (<Option at 0x7f11ab1673f8: --username>, <Option at 0x7f11aabb86c8: --

tardis.tardis_portal.management.commands.createuser.**is_valid_email**(*value*)

tardis.tardis_portal.management.commands.dumpschemas module

Command for dumping soft schema definitions

class tardis.tardis_portal.management.commands.dumpschemas.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

args = '[namespace...]'

handle(**args*, ***options*)

help = 'Dump soft schema definitions. No namespace = dump all schemas'

option_list = (<Option at 0x7f11ab1b8710: --format>, <Option at 0x7f11ab1b8d88: --da

tardis.tardis_portal.management.commands.loadschemas module

Command for loading soft schema definitions

class tardis.tardis_portal.management.commands.loadschemas.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

args = 'schema [schema ...]'

handle(**args*, ***options*)

help = 'Load soft schema definitions'

option_list = (<Option at 0x7f11ab116c68: --database>, <Option at 0x7f11ab1169e0: --

tardis.tardis_portal.management.commands.rmexperiment module

Management command to delete the specified experiment and its associated datasets, datafiles and parameters.

The operation is atomic, either the entire experiment is deleted, or nothing.

rmexperiment was introduced due to the Oracle DISTINCT workaround causing sql delete cascading to fail. The current implementation of rmexperiment still relies on some cascading.

class tardis.tardis_portal.management.commands.rmexperiment.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

args = '<MyTardis Exp ID>'

handle(**args*, ***options*)

help = 'Delete the supplied MyTardis Experiment ID'

option_list = (<Option at 0x7f11ab15efc8: --list>, <Option at 0x7f11ab01b7a0: --conf

tardis.tardis_portal.management.commands.runfilters module

Management command to (re-)run the ingestion filters by hand.

class tardis.tardis_portal.management.commands.runfilters.**Command**(*stdout=None*, *stderr=None*, *no_color=False*)

Bases: django.core.management.base.BaseCommand

args = '[<filter-no>] ...'

handle(**args*, ***options*)

help = "Run selected ingestion filters on all Datafiles.\nNote that a typical ingestio

instantiateFilters(*filterIds*)

listFilters()

option_list = (<Option at 0x7f11aabdbef0: -n/--dryRun>, <Option at 0x7f11aabdbf80: -

runFilters(*filters*, *dryRun=False*)

tardis.tardis_portal.management.commands.sftpd module

Module contents

Module contents

tardis.tardis_portal.migrations package

Submodules

tardis.tardis_portal.migrations.0001_initial module

class tardis.tardis_portal.migrations.0001_initial.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'contenttypes', u'0002_remove_content_type_name'), (u'auth', u'__firs

operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'fi

tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643 module

class tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643.**Migration**(*name*, *app_la*

Bases: django.db.migrations.migration.Migration

dependencies = [(u'contenttypes', u'0002_remove_content_type_name'), (u'auth', u'__firs

operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'fi

replaces = [('tardis_portal', u'0001_initial'), ('tardis_portal', u'0002_auto_20150528_

tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643.**cast_string_to_integ**

tardis.tardis_portal.migrations.0002_auto_20150528_1128 module

class tardis.tardis_portal.migrations.0002_auto_20150528_1128.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0001_initial')]

operations = [<AlterField field=<django.db.models.fields.related.OneToOneField>, name=

tardis.tardis_portal.migrations.0003_auto_20150907_1315 module

class tardis.tardis_portal.migrations.0003_auto_20150907_1315.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0002_auto_20150528_1128')]

operations = [<AlterField field=<django.db.models.fields.IntegerField>, name=u'data_ty

tardis.tardis_portal.migrations.0004_storageboxoption_value_type module

class tardis.tardis_portal.migrations.0004_storageboxoption_value_type.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0003_auto_20150907_1315')]

operations = [<AddField field=<django.db.models.fields.CharField>, name=u'value_type',

tardis.tardis_portal.migrations.0005_datafile_add_size_int_column module

class tardis.tardis_portal.migrations.0005_datafile_add_size_int_column.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0004_storageboxoption_value_type')]

operations = [<AddField field=<django.db.models.fields.BigIntegerField>, name=u'_size'

tardis.tardis_portal.migrations.0005_datafile_add_size_int_column.**cast_string_to_integer**(*ap*
sc

tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column module

class tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column.**Migration**(*nam*
app

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0005_datafile_add_size_int_column')]

operations = [<RemoveField name=u'size', model_name=u'datafile'>, <RenameField new_nam

tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index module

class tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index.**Migration**(*na*
ap

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0006_datafile_remove_size_string_column')]

operations = [<AlterField field=<django.db.models.fields.TextField>, name=u'string_val

tardis.tardis_portal.migrations.0008_string_value_partial_index_postgres module

class tardis.tardis_portal.migrations.0008_string_value_partial_index_postgres.**Migration**(*na*
ap

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0007_remove_parameter_string_value_index')]

operations = []

tardis.tardis_portal.migrations.0009_auto_20160128_1119 module

class tardis.tardis_portal.migrations.0009_auto_20160128_1119.**Migration**(*name*,
app_label)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0008_string_value_partial_index_postgres')]

operations = [<AlterUniqueTogether unique_together=set([(u'name', u'facility')]), name

tardis.tardis_portal.migrations.0010_auto_20160503_1443 module

class tardis.tardis_portal.migrations.0010_auto_20160503_1443.**Migration**(*name*,
app_label)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0009_auto_20160128_1119')]

operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'mimetype',

tardis.tardis_portal.migrations.0011_auto_20160505_1643 module

class tardis.tardis_portal.migrations.0011_auto_20160505_1643.**Migration**(*name*,
app_label)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'tardis_portal', u'0010_auto_20160503_1443')]

operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'directory'

Module contents

tardis.tardis_portal.models package

Submodules

tardis.tardis_portal.models.access_control module

class tardis.tardis_portal.models.access_control.**GroupAdmin**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

GroupAdmin links the Django User and Group tables for group administrators

Attribute user a forign key to the django.contrib.auth.models.User

Attribute group a forign key to the django.contrib.auth.models.Group

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

group

objects = <django.db.models.manager.Manager object>

user

class tardis.tardis_portal.models.access_control.**ObjectACL**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

The ObjectACL (formerly ExperimentACL) table is the core of the Tardis Authorisation framework

Attribute pluginId the the name of the auth plugin being used

Attribute entityId a foreign key to auth plugins

Attribute object_type a foreign key to ContentType

Attribute object_id the primary key/id of the object_type

Attribute canRead gives the user read access

Attribute canWrite gives the user write access

Attribute canDelete gives the user delete permission

Attribute isOwner the experiment owner flag.

Attribute effectiveDate the date when access takes into effect

Attribute expiryDate the date when access ceases

Attribute aclOwnershipType system-owned or user-owned.

System-owned ACLs will prevent users from removing or editing ACL entries to a particular experiment they own. User-owned ACLs will allow experiment owners to remove/add/edit ACL entries to the experiments they own.

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

OWNER_OWNED = 1

SYSTEM_OWNED = 2

content_object
Provides a generic relation to any object through content-type/object-id fields.

content_type

get_aclOwnershipType_display(***morekwargs*)

classmethod get_effective_query()

get_related_object()
If possible, resolve the pluginId/entityId combination to a user or group object.

get_related_object_group()
If possible, resolve the pluginId/entityId combination to a user or group object.

objects = <django.db.models.manager.Manager object>

class tardis.tardis_portal.models.access_control.**UserAuthentication**(*id*, *userProfile*, *username*, *authenticationMethod*)
Bases: django.db.models.base.Model

CHOICES = ()

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

getAuthMethodDescription()

objects = <django.db.models.manager.Manager object>

userProfile

class tardis.tardis_portal.models.access_control.**UserProfile**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

UserProfile class is an extension to the Django standard user model.

Attribute isDjangoAccount is the user a local DB user

Attribute user a foreign key to the django.contrib.auth.models.User

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

ext_groups

getUserAuthentications()

isValidPublicContact()
Checks if there's enough information on the user for it to be used as a public contact.

Note: Last name can't be required, because people don't necessarilly have a last (or family) name.

objects = <django.db.models.manager.Manager object>

user

userauthentication_set

tardis.tardis_portal.models.access_control.**create_user_api_key**(*sender*, ***kwargs*)

Auto-create ApiKey objects using Tastypie's create_api_key

tardis.tardis_portal.models.access_control.**create_user_profile**(*sender*, *instance*, *created*, ***kwargs*)

tardis.tardis_portal.models.datafile module

class tardis.tardis_portal.models.datafile.**DataFile**(**args*, ***kwargs*)

Bases: django.db.models.base.Model

Class to store meta-data about a file. The physical copies of a file are described by distinct DataFileObject instances.

Attribute dataset the foreign key to the tardis.tardis_portal.models.Dataset the file belongs to.

Attribute filename the name of the file, excluding the path.

Attribute size the size of the file.

Attribute created_time time the file was added to tardis

Attribute modification_time last modification time of the file

Attribute mimetype for example 'application/pdf'

Attribute md5sum digest of length 32, containing only hexadecimal digits

Attribute sha512sum digest of length 128, containing only hexadecimal digits

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

cache_file()

datafileparameterset_set

dataset

file_object

file_objects

getParameterSets(*schemaType=None*)

Return datafile parametersets associated with this datafile.

get_absolute_filepath()

get_as_temporary_file(***kwds*)

Returns a traditional file-system-based file object that is a copy of the original data. The file is deleted when the context is destroyed. :param basestring directory: the directory in which to create the temp

file

Returns the temporary file object

Return type NamedTemporaryFile

get_default_storage_box()
try to guess appropriate box from files, dataset or experiment

get_download_url()

get_file(*verified_only=True*)
Returns the file as a readable file-like object from the best avaiable storage box.

If verified_only=False, the return of files without a verified checksum is allowed, otherwise None is returned for unverified files.

Parameters **verified_only** (*bool*) – if False return files without verified checksums

Returns Python file object

Return type Python File object

get_image_data()

get_mimetype()

get_preferred_dfo(*verified_only=True*)

get_receiving_storage_box()

get_size()

get_view_url()

has_image()

is_image()
returns True if it's an image and not an x-icon and not an img the image/img mimetype is made up though and may need revisiting if there is an official img mimetype that does not refer to diffraction images

is_local()

is_online
return False if a file is on tape. At this stage it checks it returns true for no file objects, because those files are offline through other checks

is_public()

objects = <django.db.models.manager.Manager object>

save(**args*, ***kwargs*)

status
returns information about the status of the file. States are defined in StorageBox

classmethod sum_sizes(*datafiles*)
Takes a query set of datafiles and returns their total size.

update_mimetype(*mimetype=None*, *force=False*, *save=True*)

verified

verify(*reverify=False*)

class tardis.tardis_portal.models.datafile.**DataFileObject**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

holds one copy of the data for a datafile

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

cache_file()

copy_file(*dest_box=None*, *verify=True*)
copies verified file to new storage box checks for existing copy triggers async verification if not disabled :param StorageBox dest_box: StorageBox instance :param bool verify: :returns: DataFileObject of copy :rtype: DataFileObject

create_set_uri(*force=False*, *save=False*)
sets the uri as well as building it :param bool force: :param book save: :return: :rtype: basestring

datafile

delete_data()

file_object
A set of accessor functions that convert the file information to a standard Python file object for reading and copy the contents of an existing file_object into the storage backend.

Returns a file object

Return type Python File object

get_full_path()

get_next_by_created_time(***morekwargs*)

get_previous_by_created_time(***morekwargs*)

modified_time

move_file(*dest_box=None*)
moves a file copies first, then synchronously verifies deletes file if copy is true copy and has been verified

Parameters **dest_box** (StorageBox) – StorageBox instance

Returns moved file dfo

Return type *DataFileObject*

objects = <django.db.models.manager.Manager object>

save(**args*, ***kwargs*)

storage_box

storage_type

Returns storage_box type

Return type StorageBox type constant

verify(*add_checksums=True*, *add_size=True*)

tardis.tardis_portal.models.datafile.**compute_checksums**(*file_object*, *compute_md5=True*, *compute_sha512=True*, *close_file=True*)

Computes checksums for a python file object

Parameters

- **file_object** (*object*) – Python File object

- **compute_md5** (*bool*) – whether to compute md5 default=True
- **compute_sha512** (*bool*) – whether to compute sha512, default=True
- **close_file** (*bool*) – whether to close the file_object, default=True

Returns the checksums as {'md5sum': result, 'sha512sum': result}

Return type dict

tardis.tardis_portal.models.datafile.**delete_dfo**(*sender*, *instance*, ***kwargs*)

tardis.tardis_portal.models.dataset module

class tardis.tardis_portal.models.dataset.**Dataset**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

Class to link datasets to experiments

Attribute experiment a forign key to the tardis.tardis_portal.models.Experiment

Attribute facility the foreign key to the facility that generated this data

Attribute instrument the foreign key to the instrument that generated this data

Attribute description description of this dataset

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

datafile_set

datasetparameterset_set

equipment_set

experiments

getParameterSets(*schemaType=None*)
Return the dataset parametersets associated with this experiment.

get_absolute_url(***kwargs*)
Return the absolute url to the current Dataset

get_all_storage_boxes_used()

get_datafiles()

get_download_urls()

get_edit_url(***kwargs*)
Return the absolute url to the edit view of the current Dataset

get_first_experiment()

get_images()

get_path()

get_size()

get_thumbnail_url()

image

instrument

is_online

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

tardis.tardis_portal.models.experiment module

class tardis.tardis_portal.models.experiment.**Experiment**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

The Experiment model inherits from django.db.models.Model

Attribute url Undocumented

Attribute approved Undocumented

Attribute title the title of the experiment.

Attribute institution_name the name of the institution who created the dataset.

Attribute start_time Undocumented

Attribute end_time Undocumented

Attribute created_time Undocumented

Attribute handle Undocumented

Attribute public Undocumented

Attribute objects default model manager

Attribute safe ACL aware model manager

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

PUBLICATION_DETAILS_SCHEMA = 'http://www.tardis.edu.au/schemas/publication/details/'

PUBLICATION_DRAFT_SCHEMA = 'http://www.tardis.edu.au/schemas/publication/draft/'

PUBLICATION_SCHEMA_ROOT = 'http://www.tardis.edu.au/schemas/publication/'

PUBLIC_ACCESS_CHOICES = ((1, 'No public access (hidden)'), (25, 'Ready to be released

PUBLIC_ACCESS_EMBARGO = 25

PUBLIC_ACCESS_FULL = 100

PUBLIC_ACCESS_METADATA = 50

PUBLIC_ACCESS_NONE = 1

created_by

datasets

experimentauthor_set

experimentparameterset_set

getParameterSets(*schemaType=None*)
Return the experiment parametersets associated with this experiment.

get_absolute_url(***kwargs*)
Return the absolute url to the current `Experiment`

get_create_token_url(***kwargs*)
Return the absolute url to the create token view of the current `Experiment`

get_ct()

get_datafiles()

get_download_urls()

get_edit_url(***kwargs*)
Return the absolute url to the edit view of the current `Experiment`

get_groups()

get_images()

get_next_by_created_time(***morekwargs*)

get_next_by_update_time(***morekwargs*)

get_or_create_directory()

get_owners()

get_previous_by_created_time(***morekwargs*)

get_previous_by_update_time(***morekwargs*)

get_public_access_display(***morekwargs*)

get_size()

is_publication()

is_publication_draft()

license

objectacls
This class provides the functionality that makes the related-object managers available as attributes on a model class, for fields that have multiple "remote" values and have a GenericRelation defined in their model (rather than having another model pointed *at* them). In the example "article.publications", the publications attribute is a ReverseGenericRelatedObjectsDescriptor instance.

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

classmethod public_access_implies_distribution(*public_access_level*)
Determines if a level of public access implies that distribution should be allowed, or alternately if it should not be allowed. Used to prevent free-distribution licences for essentially private data, and overly-restrictive licences for public data.

public_download_allowed()
instance method version of 'public_access_implies_distribution'

safe = <tardis.tardis_portal.managers.ExperimentManager object>

save(**args*, ***kwargs*)

token_set

class tardis.tardis_portal.models.experiment.**ExperimentAuthor**(*id*, *experiment*, *author*, *institution*, *email*, *order*, *url*)

Bases: django.db.models.base.Model

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

experiment

objects = <django.db.models.manager.Manager object>

save(**args*, ***kwargs*)

tardis.tardis_portal.models.facility module

class tardis.tardis_portal.models.facility.**Facility**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

Represents a facility that produces data

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

instrument_set

manager_group

objects = <django.db.models.manager.Manager object>

tardis.tardis_portal.models.facility.**facilities_managed_by**(*user*)
Returns a list of facilities managed by a user

tardis.tardis_portal.models.facility.**is_facility_manager**(*user*)
Returns true if the user manages one or more facilities

tardis.tardis_portal.models.hooks module

tardis.tardis_portal.models.hooks.**ensure_doi_exists**(*sender*, ***kwargs*)

tardis.tardis_portal.models.hooks.**post_save_experiment**(*sender*, ***kwargs*)

tardis.tardis_portal.models.hooks.**post_save_experiment_parameter**(*sender*, ***kwargs*)

tardis.tardis_portal.models.hooks.**post_save_experimentauthor**(*sender*, ***kwargs*)

tardis.tardis_portal.models.hooks.**publish_public_expt_rifcs**(*experiment*)

tardis.tardis_portal.models.instrument module

class tardis.tardis_portal.models.instrument.**Instrument**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

Represents an instrument belonging to a facility that produces data

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

dataset_set

facility

getParameterSets(*schemaType=None*)
Return the instrument parametersets associated with this instrument.

instrumentparameterset_set

objects = <django.db.models.manager.Manager object>

tardis.tardis_portal.models.jti module

class tardis.tardis_portal.models.jti.**JTI**(*id*, *jti*, *created_time*)
Bases: django.db.models.base.Model

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

get_next_by_created_time(***morekwargs*)

get_previous_by_created_time(***morekwargs*)

objects = <django.db.models.manager.Manager object>

tardis.tardis_portal.models.license module

class tardis.tardis_portal.models.license.**License**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

Represents a licence for experiment content.

Instances should provide enough detail for both researchers to select the licence, and for the users of their data to divine correct usage of experiment content.

(Non-US developers: We're using US spelling in the code.)

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

experiment_set

classmethod get_none_option_license()

classmethod get_suitable_licenses(*public_access_method=None*)

objects = <django.db.models.manager.Manager object>

tardis.tardis_portal.models.parameters module

class tardis.tardis_portal.models.parameters.**DatafileParameter**(*id*, *name*, *string_value*, *numerical_value*, *datetime_value*, *link_id*, *link_ct*, *parameterset*)

Bases: *tardis.tardis_portal.models.parameters.Parameter*

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

link_ct

link_gfk
Provides a generic relation to any object through content-type/object-id fields.

name

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

parameter_type = 'Datafile'

parameterset

class tardis.tardis_portal.models.parameters.**DatafileParameterSet**(*id*, *schema*, *datafile*)

Bases: *tardis.tardis_portal.models.parameters.ParameterSet*

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

datafile

datafileparameter_set

objects = <django.db.models.manager.Manager object>

parameter_class
alias of *DatafileParameter*

schema

storage_box

class tardis.tardis_portal.models.parameters.**DatasetParameter**(*id*, *name*, *string_value*, *numerical_value*, *datetime_value*, *link_id*, *link_ct*, *parameterset*)

Bases: *tardis.tardis_portal.models.parameters.Parameter*

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

link_ct

link_gfk
Provides a generic relation to any object through content-type/object-id fields.

name

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

parameter_type = 'Dataset'

parameterset

class tardis.tardis_portal.models.parameters.**DatasetParameterSet**(*id*, *schema*, *dataset*)
Bases: *tardis.tardis_portal.models.parameters.ParameterSet*

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

dataset

datasetparameter_set

objects = <django.db.models.manager.Manager object>

parameter_class
alias of *DatasetParameter*

schema

storage_box

class tardis.tardis_portal.models.parameters.**ExperimentParameter**(*id*, *name*, *string_value*, *numerical_value*, *datetime_value*, *link_id*, *link_ct*, *parameterset*)
Bases: *tardis.tardis_portal.models.parameters.Parameter*

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

link_ct

link_gfk
Provides a generic relation to any object through content-type/object-id fields.

name

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

parameter_type = 'Experiment'

parameterset

save(**args*, ***kwargs*)

class tardis.tardis_portal.models.parameters.**ExperimentParameterSet**(*id*, *schema*, *experiment*)

Bases: *tardis.tardis_portal.models.parameters.ParameterSet*

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

experiment

experimentparameter_set

objects = <django.db.models.manager.Manager object>

parameter_class
alias of *ExperimentParameter*

schema

storage_box

class tardis.tardis_portal.models.parameters.**FreeTextSearchField**(*id*, *parameter_name*)

Bases: django.db.models.base.Model

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

objects = <django.db.models.manager.Manager object>

parameter_name

class tardis.tardis_portal.models.parameters.**InstrumentParameter**(*id*, *name*, *string_value*, *numerical_value*, *datetime_value*, *link_id*, *link_ct*, *parameterset*)

Bases: *tardis.tardis_portal.models.parameters.Parameter*

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

link_ct

link_gfk
Provides a generic relation to any object through content-type/object-id fields.

name

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

parameter_type = 'Instrument'

parameterset

class tardis.tardis_portal.models.parameters.**InstrumentParameterSet**(*id*, *schema*, *instrument*)

Bases: *tardis.tardis_portal.models.parameters.ParameterSet*

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

instrument

instrumentparameter_set

objects = <django.db.models.manager.Manager object>

parameter_class
alias of *InstrumentParameter*

schema

storage_box

class tardis.tardis_portal.models.parameters.**Parameter**(**args*, ***kwargs*)

Bases: django.db.models.base.Model

class Meta

abstract = False

app_label = 'tardis_portal'

ordering = ['name']

get()

link_ct

link_gfk
Provides a generic relation to any object through content-type/object-id fields.

link_url

name

objects

parameter_type = 'Abstract'

set_value(*value*)
Sets the parameter value, converting into the appropriate data type. Deals with date/time strings that are timezone naive or aware, based on the USE_TZ setting.

Parameters **value** (*basestring*) – a string (or string-like) repr of the value

Raises **SuspiciousOperation** –

class tardis.tardis_portal.models.parameters.**ParameterName**(*id*, *schema*, *name*, *full_name*, *units*, *data_type*, *immutable*, *comparison_type*, *is_searchable*, *choices*, *order*)

Bases: django.db.models.base.Model

CONTAINS_COMPARISON = 8

DATETIME = 6

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

EXACT_VALUE_COMPARISON = 1

FILENAME = 5

GREATER_THAN_COMPARISON = 4

GREATER_THAN_EQUAL_COMPARISON = 5

JSON = 8

LESS_THAN_COMPARISON = 6

LESS_THAN_EQUAL_COMPARISON = 7

LINK = 4

LONGSTRING = 7

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

NOT_EQUAL_COMPARISON = 2

NUMERIC = 1

RANGE_COMPARISON = 3

STRING = 2

URL = 3

datafileparameter_set

datasetparameter_set

experimentparameter_set

freetextsearchfield_set

getUniqueShortName()

get_comparison_type_display(**morekwargs*)

get_data_type_display(**morekwargs*)

instrumentparameter_set

isDateTime()

isFilename()

isLink()

isLongString()

isNumeric()

isString()

isURL()

is_json()

natural_key()

objects = <tardis.tardis_portal.managers.ParameterNameManager object>

schema

class tardis.tardis_portal.models.parameters.**ParameterSet**(**args*, ***kwargs*)

Bases: django.db.models.base.Model, *tardis.tardis_portal.models.parameters.ParameterSetManagerMixin*

class Meta

abstract = False

app_label = 'tardis_portal'

ordering = ['id']

parameter_class = None

save(**args*, ***kwargs*)

schema

storage_box

class tardis.tardis_portal.models.parameters.**ParameterSetManagerMixin**(*parameterset=None*, *parentObject=None*, *schema=None*)

Bases: *tardis.tardis_portal.ParameterSetManager.ParameterSetManager*

for clarity's sake and for future extension this class makes ParameterSetManager local to this file. At the moment its only function is increasing the line count

class tardis.tardis_portal.models.parameters.**Schema**(*id*, *namespace*, *name*, *type*, *subtype*, *immutable*, *hidden*)

Bases: django.db.models.base.Model

DATAFILE = 3

DATASET = 2

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

EXPERIMENT = 1

INSTRUMENT = 5

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

NONE = 4

exception UnsupportedType(*msg*)
Bases: exceptions.Exception

datafileparameterset_set

datasetparameterset_set

experimentparameterset_set

classmethod getNamespaces(*type_*, *subtype=None*)
Return the list of namespaces for equipment, sample, and experiment schemas.

classmethod getSubTypes()

classmethod get_internal_schema(*schema_type*)

classmethod get_schema_type_name(*schema_type*, *short=False*)

get_type_display(***morekwargs*)

instrumentparameterset_set

natural_key()

objects = <tardis.tardis_portal.managers.SchemaManager object>

parametername_set

tardis.tardis_portal.models.storage module

class tardis.tardis_portal.models.storage.**StorageBox**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

table that holds storage boxes of any type. to extend to new types, add fields if necessary

Attribute max_size max size in bytes

BUNDLE = 6

CACHE = 3

DISK = 1

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

TAPE = 2

TEMPORARY = 4

TYPES = {'bundle': 6, 'cache': 3, 'disk': 1, 'receiving': 4, 'tape': 2}

TYPE_UNKNOWN = 5

attributes

autocache
Whether to automatically copy data into faster storage

Returns True if data should be automatically cached

Return type bool

cache_box
Get cache box if set up

cache_files()
Copy all files to faster storage.

This can be used to copy data from a Vault cache (containing data which will soon be pushed to tape) to Object Storage, so that the data can always be accessed quickly from Object Storage, and the Vault can be used for disaster recovery if necessary.

child_boxes

copy_files(*dest_box=None*)

copy_to_master()

classmethod create_local_box(*location=None*)

datafileparametersets

datasetparametersets

experimentparametersets

file_objects

classmethod get_default_storage(*location=None*, *user=None*)
gets default storage box or get local storage box with given base location or create one if it doesn't exist.

policies: Have a StorageBoxAttribute: key='default', value=True find a storage box where location is DEFAULT_STORAGE_BASE_DIR create a default storage box at DEFAULT_STORAGE_BASE_DIR lowest id storage box is default no storage box defined, use hard coded default for now TODO: consider removing this

Would be nice: get largest free space one, test for authorisation

get_initialised_storage_instance()

get_options_as_dict()

instrumentparametersets

master_box

move_files(*dest_box=None*)

move_to_master()

objects = <django.db.models.manager.Manager object>

offline_types = [2]

online_types = [3, 1, 4, 6]

options

storage_type

type_order = [3, 6, 1, 2, 4, 5]

class tardis.tardis_portal.models.storage.**StorageBoxAttribute**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

can hold attributes/metadata about different storage locations.

built-ins: key values description type receiving holds files temporarily for ingestion only

permanent permanent location (assumed by default) cache holds files for fast access

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

objects = <django.db.models.manager.Manager object>

storage_box

class tardis.tardis_portal.models.storage.**StorageBoxOption**(**args*, ***kwargs*)
Bases: django.db.models.base.Model

holds the options passed to the storage class defined in StorageBox. key->value store with support for typed values through pickling when value_type is set to 'pickle'

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

PICKLE = 'pickle'

STRING = 'string'

TYPE_CHOICES = (('string', 'String value'), ('pickle', 'Pickled value'))

get_value_type_display(***morekwargs*)

objects = <django.db.models.manager.Manager object>

storage_box

unpickled_value

tardis.tardis_portal.models.token module

class tardis.tardis_portal.models.token.**Token**(*id*, *token*, *experiment*, *expiry_date*, *user*)
Bases: django.db.models.base.Model

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

experiment

get_absolute_url(***kwargs*)

get_next_by_expiry_date(***morekwargs*)

get_previous_by_expiry_date(***morekwargs*)

get_session_expiry()
A token login should expire at the earlier of a) tomorrow at 4am b) the (end of) the token's expiry date

It is the responsibility of token_auth to set the session expiry

is_expired()

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

save_with_random_token()

user

Module contents

models/__init__.py

tardis.tardis_portal.publish package

Subpackages

tardis.tardis_portal.publish.provider package

Submodules

tardis.tardis_portal.publish.provider.rifcsprovider module

class tardis.tardis_portal.publish.provider.rifcsprovider.**RifCsProvider**

Bases: object

can_publish(*experiment*)

get_rifcs_context(*experiment*)

get_template(*experiment*)

is_schema_valid(*experiment*)

tardis.tardis_portal.publish.provider.schemarifcsprovider module

Module contents

Submodules

tardis.tardis_portal.publish.publishservice module

class tardis.tardis_portal.publish.publishservice.**PublishService**(*providers*, *experiment*)

get_context()

get_template()

manage_rifcs(*oaipath*)

Module contents

tardis.tardis_portal.storage package

Submodules

tardis.tardis_portal.storage.file_system module

class tardis.tardis_portal.storage.file_system.**MyTardisLocalFileSystemStorage**(*location=None*, *base_url=None*)

Bases: django.core.files.storage.FileSystemStorage

Simply changes the FileSystemStorage default store location to the MyTardis file store location. Makes it easier to migrate 2.5 installations.

tardis.tardis_portal.storage.squashfs module

SquashFS Storage Box

setup requirements: user-owned mount root dir mytardis user must have access to files in squashfs file based on immutable permissions set in squashfile. In easy cases, one group owns everything and the mytardis user can be added to that group.

set up autofs to auto-mount squashfiles

class tardis.tardis_portal.storage.squashfs.**SquashFSStorage**(*sq_filename=None*, *datafile_id=None*, *sq_dir=None*)

Bases: django.core.files.storage.FileSystemStorage

Only works for autofs mounted squashfs files Please provide a setting that maps source dir and mount dir, e.g.: SQUASHFS_DIRS = {'tape': {'source': '/my/tape/mount',

'autofs': '/taped-squashfiles'},

'volume': {'source': '/my/volume/storage', 'autofs': '/volumed-squashfiles'}}

squashfs_dirs = None

walk(*top='.'*, *topdown=True*, *onerror=None*, *ignore_dotfiles=True*)

tardis.tardis_portal.storage.squashfs.**get_parse_status**(*squash_datafile*, *ns*)

Module contents

class tardis.tardis_portal.storage.**DummyStorage**

Bases: django.core.files.storage.Storage

Does nothing except serve as a place holder for Storage classes not implemented yet

exception tardis.tardis_portal.storage.**MisconfiguredStorageError**

Bases: exceptions.Exception

tardis.tardis_portal.templatetags package

Submodules

tardis.tardis_portal.templatetags.basiccomparisonfilters module

tardis.tardis_portal.templatetags.basiccomparisonfilters.**gt**(*value*, *arg*)
Returns a boolean of whether the value is greater than the argument.

tardis.tardis_portal.templatetags.basiccomparisonfilters.**gte**(*value*, *arg*)
Returns a boolean of whether the value is greater than or equal to the argument.

tardis.tardis_portal.templatetags.basiccomparisonfilters.**length_gt**(*value*, *arg*)
Returns a boolean of whether the value's length is greater than the argument.

tardis.tardis_portal.templatetags.basiccomparisonfilters.**length_gte**(*value*, *arg*)
Returns a boolean of whether the value's length is greater than or equal to the argument.

tardis.tardis_portal.templatetags.basiccomparisonfilters.**length_lt**(*value*, *arg*)
Returns a boolean of whether the value's length is less than the argument.

tardis.tardis_portal.templatetags.basiccomparisonfilters.**length_lte**(*value*, *arg*)
Returns a boolean of whether the value's length is less than or equal to the argument.

tardis.tardis_portal.templatetags.basiccomparisonfilters.**lt**(*value*, *arg*)
Returns a boolean of whether the value is less than the argument.

tardis.tardis_portal.templatetags.basiccomparisonfilters.**lte**(*value*, *arg*)
Returns a boolean of whether the value is less than or equal to the argument.

tardis.tardis_portal.templatetags.bleach_tag module

tardis.tardis_portal.templatetags.bleach_tag.**bleach_value**(*value*)

tardis.tardis_portal.templatetags.capture module

class tardis.tardis_portal.templatetags.capture.**CaptureNode**(*nodelist*, *varname*)
Bases: django.template.base.Node

render(*context*)

tardis.tardis_portal.templatetags.capture.**capture**(*parser*, *token*)
{% capture as [foo] %}

tardis.tardis_portal.templatetags.dataset_tags module

tardis.tardis_portal.templatetags.dynurl module

class tardis.tardis_portal.templatetags.dynurl.**DynUrlNode**(**args*)
Bases: django.template.base.Node

render(*context*)

tardis.tardis_portal.templatetags.dynurl.**dynurl**(*parser*, *token*)

tardis.tardis_portal.templatetags.experiment_tags module

tardis.tardis_portal.templatetags.experimentstats module

tardis.tardis_portal.templatetags.experimentstats.**experiment_file_count**(*value*)

tardis.tardis_portal.templatetags.facility_tags module

tardis.tardis_portal.templatetags.facility_tags.**check_if_facility_manager**(*request*)
Custom template filter to identify whether a user is a facility manager.

tardis.tardis_portal.templatetags.feed module

tardis.tardis_portal.templatetags.feed.**todatetime**(*value*)

tardis.tardis_portal.templatetags.formfieldfilters module

This module holds filters that can be used in postprocessing a form field.

@author: Gerson Galang

tardis.tardis_portal.templatetags.formfieldfilters.**parametername_form**(*value*)
Removes all values of arg from the given string

tardis.tardis_portal.templatetags.formfieldfilters.**sanitize_html**(*html, bad_tags=['body']*)
Removes identified malicious HTML content from the given string.

tardis.tardis_portal.templatetags.formfieldfilters.**size**(*value*, *actualSize*)
Add the size attribute to the text field.

tardis.tardis_portal.templatetags.lookupfilters module

tardis.tardis_portal.templatetags.lookupfilters.**get_item**(*dictionary*, *key*)
Returns a value from a dictionary.

tardis.tardis_portal.templatetags.uploadify_tags module

tardis.tardis_portal.templatetags.uploadify_tags.**multi_file_upload**(*context*, *upload_complete_url*, *dataset_id*)
Displays a Flash-based interface for uploading multiple files. When all files have been uploaded, the given URL is POSTed to. The returned page replaces (AJAX) the upload interface.

- filesUploaded - The total number of files uploaded
- errors - The total number of errors while uploading

- allBytesLoaded - The total number of bytes uploaded
- speed - The average speed of all uploaded files

tardis.tardis_portal.templatetags.xmldate module

tardis.tardis_portal.templatetags.xmldate.**toxmldatetime**(*value*)

Module contents

tardis.tardis_portal.tests package

Subpackages

tardis.tardis_portal.tests.auth package

Submodules

tardis.tardis_portal.tests.auth.test_httpbasicendpoint_auth module

Module contents

tardis.tardis_portal.tests.filters package

Submodules

tardis.tardis_portal.tests.filters.test_jeolsem module

class tardis.tardis_portal.tests.filters.test_jeolsem.**JEOLSEMFilterTestCase**(*methodName='runTe*
Bases: django.test.testcases.TransactionTestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testJEOLComplex()

testJEOLSimple()

tardis.tardis_portal.tests.filters.test_middleware module

class tardis.tardis_portal.tests.filters.test_middleware.**Filter1**

classmethod getTuples()

tuples = []

class tardis.tardis_portal.tests.filters.test_middleware.**Filter2**

classmethod getTuples()

tuples = []

class tardis.tardis_portal.tests.filters.test_middleware.**FilterInitTestCase**(*methodName='runTe*
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testFiltering()

Module contents

Submodules

tardis.tardis_portal.tests.ldap_ldif module

tardis.tardis_portal.tests.mock_vbl_auth module

Created on 24/01/2011

@author: gerson

class tardis.tardis_portal.tests.mock_vbl_auth.**MockBackend**
Authenticate against the VBL SOAP Webservice. It is assumed that the request object contains the username and password to be provided to the VBLgetExpIDs function.

a new local user is created if it doesn't already exist

if the authentication succeeds, the session will contain a VBL session key used for downloads as well as the user's EPN list

authenticate(*request*)

get_user(*user_id*)

tardis.tardis_portal.tests.mock_vbl_download module

tardis.tardis_portal.tests.mock_vbl_download.**download_datafile**(*request*, **args*, ***kwargs*)
Dummy view for vbl download tests

tardis.tardis_portal.tests.slapd module

Utilities for starting up a test slapd server and talking to it with ldapsearch/ldapadd.

class tardis.tardis_portal.tests.slapd.**Slapd**
Controller class for a slapd instance, OpenLDAP's server.

This class creates a temporary data store for slapd, runs it on a private port, and initialises it with a top-level dc and the root user.

When a reference to an instance of this class is lost, the slapd server is shut down.

PATH_LDAPADD = None

PATH_LDAPSEARCH = None

PATH_SCHEMA_DIR = '/home/docs/checkouts/readthedocs.org/user_builds/mytardis/checkouts

PATH_SLAPD = None

PATH_SLAPTEST = None

PATH_TMPDIR = '/tmp/tmppyYZ5y'

TEST_UTILS_DIR = '/home/docs/checkouts/readthedocs.org/user_builds/mytardis/checkouts/

classmethod check_paths()
Checks that the configured executable paths look valid. If they don't, then logs warning messages (not errors).

configure(*cfg*)
Appends slapd.conf configuration lines to cfg. Also re-initializes any backing storage. Feel free to subclass and override this method.

get_address()

get_dn_suffix()

get_root_dn()

get_root_password()

get_tmpdir()

get_url()

ldapadd(*ldif*, *extra_args=[]*)
Runs ldapadd on this slapd instance, passing it the ldif content

ldapsearch(*base=None*, *filter='(objectClass=*)'*, *attrs=[]*, *scope='sub'*, *extra_args=[]*)

restart()
Restarts the slapd server; ERASING previous content. Starts the server even it if isn't already running.

set_debug()

set_dn_suffix(*dn*)

set_port(*port*)

set_root_cn(*cn*)

set_root_password(*pw*)

set_slapd_debug_level(*level*)

set_tmpdir(*path*)

start()
Starts the slapd server process running, and waits for it to come up.

started()
This method is called when the LDAP server has started up and is empty. By default, this method adds the two initial objects, the domain object and the root user object.

stop()
Stops the slapd server, and waits for it to terminate

wait()
Waits for the slapd process to terminate by itself.

tardis.tardis_portal.tests.slapd.**delete_directory_content**(*path*)

tardis.tardis_portal.tests.slapd.**find_available_tcp_port**(*host='127.0.0.1'*)

tardis.tardis_portal.tests.slapd.**mkdirs**(*path*)
Creates the directory path unless it already exists

tardis.tardis_portal.tests.slapd.**quote**(*s*)
Quotes the " and characters in a string and surrounds with "..."

tardis.tardis_portal.tests.slapd.**which**(*executable*)

tardis.tardis_portal.tests.test_ands_doi module

test_ands_doi.py

class tardis.tardis_portal.tests.test_ands_doi.**ANDSDOITestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_doi_none()

test_init()

tardis.tardis_portal.tests.test_api module

Testing the tastypie-based mytardis api

class tardis.tardis_portal.tests.test_api.**ACLAuthorizationTest**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

class tardis.tardis_portal.tests.test_api.**DataFileResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

setUp()
Hook method for setting up the test fixture before exercising it.

test_post_single_file()

test_shared_fs_many_files()
tests sending many files with known permanent location (useful for Australian Synchrotron ingestions)

test_shared_fs_single_file()

class tardis.tardis_portal.tests.test_api.**DatafileParameterResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**DatafileParameterSetResourceTest**(*methodName='runTest*
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**DatasetParameterResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**DatasetParameterSetResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**DatasetResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

setUp()
Hook method for setting up the test fixture before exercising it.

test_get_dataset_filter_instrument()

test_get_dataset_no_instrument()

test_get_dataset_with_instrument()

test_post_dataset()

class tardis.tardis_portal.tests.test_api.**ExperimentParameterResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**ExperimentParameterSetResourceTest**(*methodName='runT*
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**ExperimentResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

setUp()
Hook method for setting up the test fixture before exercising it.

test_get_experiment()

test_post_experiment()

class tardis.tardis_portal.tests.test_api.**FacilityResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

test_get_facility_by_id()

test_get_facility_by_manager_group_id()
This type of query can be used to iterate through a user's groups, and use each group's id to determine which facilities a user manages, i.e. a way to obtain the functionality implemented by

tardis.tardis_portal.models.facility.facilities_managed_by

via the API

test_get_facility_by_name()

class tardis.tardis_portal.tests.test_api.**GroupResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

test_get_group_by_id()

test_get_group_by_name()

class tardis.tardis_portal.tests.test_api.**InstrumentResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

test_get_instrument_by_id()

test_get_instrument_by_name()

test_post_instrument()

test_rename_instrument()

class tardis.tardis_portal.tests.test_api.**LocationResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**MyTardisAuthenticationTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

test_apikey_authentication()

test_bad_credentials()

class tardis.tardis_portal.tests.test_api.**MyTardisResourceTestCase**(*methodName='runTest'*)
Bases: tastypie.test.ResourceTestCase

abstract class without tests to combine common settings in one place

get_apikey_credentials()

get_credentials()

setUp()
Hook method for setting up the test fixture before exercising it.

class tardis.tardis_portal.tests.test_api.**ReplicaResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**SchemaResourceTest**(*methodName='runTest'*)
Bases: *tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase*

class tardis.tardis_portal.tests.test_api.**SerializerTest**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

test_debug_serializer()

test_pretty_serializer()

tardis.tardis_portal.tests.test_authentication module

Created on 19/01/2011

class tardis.tardis_portal.tests.test_authentication.**AuthenticationTestCase**(*methodName='runTe*
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testCreateNewAccount()

testManageAuthMethods()

testSimpleAuthenticate()

test_djangoauth()

tardis.tardis_portal.tests.test_authorisation module

class tardis.tardis_portal.tests.test_authorisation.**ObjectACLTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testCantAddTokenuserToGroups()

testCantEditLockedExperiment()

testChangeUserPermissions()

testOwnedExperiments()

testReadAccess()

testWriteAccess()

urls = 'tardis.urls'

tardis.tardis_portal.tests.test_authservice module

class tardis.tardis_portal.tests.test_authservice.**AuthServiceTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testAuthenticate()

testGetGroupsForEntity()

testGroupProvider()

testGroupSearch()

testInitialisation()

urls = 'tardis.tardis_portal.tests.urls'

class tardis.tardis_portal.tests.test_authservice.**MockAuthProvider**

authenticate(*request*)

class tardis.tardis_portal.tests.test_authservice.**MockGroupProvider**
Bases: *tardis.tardis_portal.auth.interfaces.GroupProvider*

getGroupById(*id*)

getGroups(*user*)

getGroupsForEntity(*id*)

searchGroups(***filter*)

class tardis.tardis_portal.tests.test_authservice.**MockRequest**
Bases: django.http.request.HttpRequest

setPost(*field*, *value*)

class tardis.tardis_portal.tests.test_authservice.**MockSettings**
Bases: object

tardis.tardis_portal.tests.test_collectstatic module

class tardis.tardis_portal.tests.test_collectstatic.**CollectstaticTest**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_collectstatic()

tardis.tardis_portal.tests.test_download module

class tardis.tardis_portal.tests.test_download.**DownloadTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testDatasetFile()

testDownload()

testView()

tardis.tardis_portal.tests.test_download.**get_size_and_sha512sum**(*testfile*)

tardis.tardis_portal.tests.test_download_apikey module

class tardis.tardis_portal.tests.test_download_apikey.**ApiKeyDownloadTestCase**(*methodName='runT*
Bases: tastypie.test.ResourceTestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_download_apikey()

tardis.tardis_portal.tests.test_fetcher module

class tardis.tardis_portal.tests.test_fetcher.**PrivilegedOpenerTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

classmethod setUpClass()
Hook method for setting up class fixture before running tests in the class.

classmethod tearDownClass()
Hook method for deconstructing the class fixture after running all tests in the class.

testCredentials()
Test that the walker manages credentials.

class tardis.tardis_portal.tests.test_fetcher.**TestWebServer**
Utility class for running a test web server with a given handler.

class QuietSimpleHTTPRequestHandler(*request*, *client_address*, *server*)
Bases: SimpleHTTPServer.SimpleHTTPRequestHandler

Simple subclass that only prints output to STDOUT, not STDERR

do_GET()

log_message(*msg*, **args*)

class ThreadedTCPServer(*server_address*, *RequestHandlerClass*, *bind_and_activate=True*)
Bases: SocketServer.ThreadingMixIn, BaseHTTPServer.HTTPServer

classmethod getPort()

getUrl()

start()

stop()

tardis.tardis_portal.tests.test_forms module

test_models.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.tardis_portal.tests.test_forms.**RightsFormTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

test_ensures_suitable_license()

test_needs_confirmation()

tardis.tardis_portal.tests.test_iiif module

class tardis.tardis_portal.tests.test_iiif.**ExtraTestCases**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

As per: http://library.stanford.edu/iiif/image-api/compliance.html

setUp()
Hook method for setting up the test fixture before exercising it.

testImageCacheControl()

testImageHasEtags()

testInfoHasEtags()

class tardis.tardis_portal.tests.test_iiif.**Level0TestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

As per: http://library.stanford.edu/iiif/image-api/compliance.html

setUp()
Hook method for setting up the test fixture before exercising it.

testCanGetInfoAsJSON()

testCanGetInfoAsXML()

testCanGetOriginalImage()

class tardis.tardis_portal.tests.test_iiif.**Level1TestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

As per: http://library.stanford.edu/iiif/image-api/compliance.html

setUp()
Hook method for setting up the test fixture before exercising it.

testCanGetJpegFormat()

testHandleRegions()

testHandleRotation()

testHandleSizing()

class tardis.tardis_portal.tests.test_iiif.**Level2TestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

As per: http://library.stanford.edu/iiif/image-api/compliance.html

setUp()
Hook method for setting up the test fixture before exercising it.

testCanGetRequiredFormats()

testHandleSizing()

tardis.tardis_portal.tests.test_ldap module

class tardis.tardis_portal.tests.test_ldap.**LDAPErrorTest**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

test_search()

class tardis.tardis_portal.tests.test_ldap.**LDAPTest**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_authenticate()

test_getgroupbyid()

test_getgroups()

test_getgroupsforentity()

test_getuserbyid()

test_search()

test_searchgroups()

tardis.tardis_portal.tests.test_models module

test_models.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.tardis_portal.tests.test_models.**ModelTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

test_authors()

test_create_user_automatically_generate_api_key()

test_datafile()

test_dataset()

test_experiment()

test_parameter()

tardis.tardis_portal.tests.test_parametersetmanager module

test_views.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.tardis_portal.tests.test_parametersetmanager.**ParameterSetManagerTestCase**(*metho*
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_existing_parameterset()

test_link_parameter_type()
Test that Parameter.link_gfk (GenericForeignKey) is correctly assigned after using Parameter.set_value(some_url) for a LINK Parameter.

test_link_parameter_type_extra()

test_new_parameterset()

test_tz_aware_date_handling()
Ensure that dates are handling in a timezone-aware way.

test_tz_naive_date_handling()
Ensure that dates are handling in a timezone-aware way.

test_unresolvable_link_parameter()
Test that LINK Parameters that can't be resolved to a model (including non-URL values) still work.

tardis.tardis_portal.tests.test_publishservice module

class tardis.tardis_portal.tests.test_publishservice.**MockRifCsProvider**
Bases: *tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider*

get_beamline(*experiment*)

get_license_uri(*experiment*)

get_rifcs_context(*experiment*)

get_template(*experiment*)

is_schema_valid(*experiment*)

class tardis.tardis_portal.tests.test_publishservice.**PublishServiceTestCase**(*methodName='runTe*
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

testContext()

testInitialisation()

testInitialisationNoProvider()

testManageRifCsCheckContent()

testManageRifCsCreateAndRemove()

tardis.tardis_portal.tests.test_rmexperiment module

class tardis.tardis_portal.tests.test_rmexperiment.**RmExperimentTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

testList()

testRemove()

tardis.tardis_portal.tests.test_staging module

test_views.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.tardis_portal.tests.test_staging.**TestStagingFiles**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_stage_replica()

tardis.tardis_portal.tests.test_storage module

test_storage.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.tardis_portal.tests.test_storage.**ModelTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_storageboxoption()

tardis.tardis_portal.tests.test_tar_download module

class tardis.tardis_portal.tests.test_tar_download.**TarDownloadTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_tar_experiment_download()

tardis.tardis_portal.tests.test_tasks module

class tardis.tardis_portal.tests.test_tasks.**BackgroundTaskTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

As per: http://library.stanford.edu/iiif/image-api/compliance.html

setUp()
Hook method for setting up the test fixture before exercising it.

testLocalFile()

test_wrong_size_verification()

tardis.tardis_portal.tests.test_tokens module

test_tokens.py

class tardis.tardis_portal.tests.test_tokens.**FrozenTime**(**args*, ***kwargs*)

classmethod freeze_time(*time*)

classmethod now()

class tardis.tardis_portal.tests.test_tokens.**TokenTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_default_expiry()

test_get_session_expiry()

test_get_session_expiry_expired_token()

test_get_session_expiry_near_expiry()

test_is_expired()

test_save_with_random_token()

test_save_with_random_token_failures()

test_save_with_random_token_gives_up()

urls = 'tardis.tardis_portal.tests.urls'

tardis.tardis_portal.tests.test_views module

tardis.tardis_portal.tests.tests module

tests.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.tardis_portal.tests.tests.**SearchTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testSearchDatafileAuthentication(***kwargs*)

testSearchDatafileForm(***kwargs*)

testSearchExperimentAuthentication(***kwargs*)

testSearchExperimentForm(***kwargs*)

testSearchExperimentResults(***kwargs*)

class tardis.tardis_portal.tests.tests.**UserInterfaceTestCase**(*methodName='runTest'*)
Bases: django.test.testcases.TestCase

test_login()

test_root()

test_search_urls(***kwargs*)

test_urls()

test_urls_with_some_content()

tardis.tardis_portal.tests.tests.**suite**()

tardis.tardis_portal.tests.urls module

Module contents

tardis.tardis_portal.views package

Submodules

tardis.tardis_portal.views.ajax_actions module

tardis.tardis_portal.views.ajax_json module

tardis.tardis_portal.views.ajax_pages module

tardis.tardis_portal.views.authentication module

tardis.tardis_portal.views.authorisation module

tardis.tardis_portal.views.facilities module

tardis.tardis_portal.views.images module

tardis.tardis_portal.views.machine module

tardis.tardis_portal.views.pages module

tardis.tardis_portal.views.parameters module

tardis.tardis_portal.views.upload module

tardis.tardis_portal.views.utils module

Module contents

Submodules

tardis.tardis_portal.MultiPartForm module

class tardis.tardis_portal.MultiPartForm.**MultiPartForm**
Bases: object

Accumulate the data to be used when posting a form.

add_field(*name*, *value*)
Add a simple field to the form data.

add_file(*fieldname*, *filename*, *fileHandle*, *mimetype=None*)
Add a file to be uploaded.

get_content_type()

tardis.tardis_portal.ParameterSetManager module

class tardis.tardis_portal.ParameterSetManager.**ParameterSetManager**(*parameterset=None*, *parentObject=None*, *schema=None*)

Bases: object

blank_param = None

delete_all_params()

delete_params(*parname*)

get_param(*parname*, *value=False*)

get_params(*parname*, *value=False*)

get_schema()

new_param(*parname*, *value*, *fullparname=None*)

parameters = None

parameterset = None

set_param(*parname*, *value*, *fullparname=None*, *example_value=None*)

set_param_list(*parname*, *value_list*, *fullparname=None*)

set_params_from_dict(*dict*)

tardis.tardis_portal.admin module

class tardis.tardis_portal.admin.**DataFileObjectInline**(*parent_model*, *admin_site*)

Bases: django.contrib.admin.options.TabularInline

extra = 0

form

alias of *DataFileObjectInlineForm*

media

model

alias of *tardis.tardis_portal.models.datafile.DataFileObject*

class tardis.tardis_portal.admin.**DataFileObjectInlineForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)

Bases: django.forms.models.ModelForm

class Meta

fields = '__all__'

model
alias of *tardis.tardis_portal.models.datafile.DataFileObject*

widgets = {'uri': <django.forms.widgets.TextInput object at 0x7f11b204f410>}

base_fields = {'datafile': <django.forms.models.ModelChoiceField object at 0x7f11b204

declared_fields = {}

media

class tardis.tardis_portal.admin.**DatafileAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

form
alias of *DatafileAdminForm*

inlines = [<class 'tardis.tardis_portal.admin.DataFileObjectInline'>]

media

search_fields = ['filename', 'id']

class tardis.tardis_portal.admin.**DatafileAdminForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)
Bases: django.forms.models.ModelForm

class Meta

fields = '__all__'

model
alias of *tardis.tardis_portal.models.datafile.DataFile*

widgets = {'directory': <django.forms.widgets.TextInput object at 0x7f11b204fb50>}

base_fields = {'created_time': <django.forms.fields.DateTimeField object at 0x7f11b20

declared_fields = {}

media

class tardis.tardis_portal.admin.**DatasetAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

media

search_fields = ['description', 'id']

class tardis.tardis_portal.admin.**ExperimentAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

inlines = [<class 'tardis.tardis_portal.admin.ObjectACLInline'>]

media

search_fields = ['title', 'id']

class tardis.tardis_portal.admin.**ExperimentParameterInline**(*parent_model*, *admin_site*)
Bases: django.contrib.admin.options.TabularInline

extra = 0

formfield_overrides = {<class 'django.db.models.fields.TextField'>: {'widget': <clas

media

model
alias of *tardis.tardis_portal.models.parameters.ExperimentParameter*

class tardis.tardis_portal.admin.**ExperimentParameterSetAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

inlines = [<class 'tardis.tardis_portal.admin.ExperimentParameterInline'>]

media

class tardis.tardis_portal.admin.**FreeTextSearchFieldAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

media

class tardis.tardis_portal.admin.**InstrumentParameterInline**(*parent_model*, *admin_site*)
Bases: django.contrib.admin.options.TabularInline

extra = 0

media

model
alias of *tardis.tardis_portal.models.parameters.InstrumentParameter*

class tardis.tardis_portal.admin.**InstrumentParameterSetAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

inlines = [<class 'tardis.tardis_portal.admin.InstrumentParameterInline'>]

media

class tardis.tardis_portal.admin.**ObjectACLAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

list_display = ['__unicode__', 'pluginId', 'entityId', 'canRead', 'canWrite', 'canDele

media

search_fields = ['content_type', 'object_id']

class tardis.tardis_portal.admin.**ObjectACLInline**(*parent_model*, *admin_site*)
Bases: django.contrib.contenttypes.admin.GenericTabularInline

extra = 0

media

model
alias of *tardis.tardis_portal.models.access_control.ObjectACL*

class tardis.tardis_portal.admin.**ParameterNameAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

media

search_fields = ['name', 'schema__id']

class tardis.tardis_portal.admin.**ParameterNameInline**(*parent_model*, *admin_site*)
Bases: django.contrib.admin.options.TabularInline

extra = 0

media

model
alias of *tardis.tardis_portal.models.parameters.ParameterName*

class tardis.tardis_portal.admin.**SchemaAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

inlines = [<class 'tardis.tardis_portal.admin.ParameterNameInline'>]

media

search_fields = ['name', 'namespace']

class tardis.tardis_portal.admin.**StorageBoxAdmin**(*model*, *admin_site*)
Bases: django.contrib.admin.options.ModelAdmin

form
alias of *StorageBoxForm*

inlines = [<class 'tardis.tardis_portal.admin.StorageBoxOptionInline'>, <class 'tardis

media

class tardis.tardis_portal.admin.**StorageBoxAttributeInline**(*parent_model*, *admin_site*)
Bases: django.contrib.admin.options.TabularInline

extra = 0

form
alias of *StorageBoxAttributeInlineForm*

media

model
alias of *tardis.tardis_portal.models.storage.StorageBoxAttribute*

class tardis.tardis_portal.admin.**StorageBoxAttributeInlineForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)
Bases: django.forms.models.ModelForm

class Meta

fields = '__all__'

model
alias of *tardis.tardis_portal.models.storage.StorageBoxAttribute*

widgets = {'key': <django.forms.widgets.TextInput object at 0x7f11b2041890>, 'valu

base_fields = {'key': <django.forms.fields.CharField object at 0x7f11b2041d50>, 'stora

declared_fields = {}

media

class tardis.tardis_portal.admin.**StorageBoxForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)

Bases: django.forms.models.ModelForm

class Meta

fields = '__all__'

model
alias of *tardis.tardis_portal.models.storage.StorageBox*

widgets = {'description': <django.forms.widgets.TextInput object at 0x7f11b204f310

base_fields = {'description': <django.forms.fields.CharField object at 0x7f11b204f7d0

declared_fields = {}

media

class tardis.tardis_portal.admin.**StorageBoxOptionInline**(*parent_model*, *admin_site*)

Bases: django.contrib.admin.options.TabularInline

extra = 0

form
alias of *StorageBoxOptionInlineForm*

media

model
alias of *tardis.tardis_portal.models.storage.StorageBoxOption*

class tardis.tardis_portal.admin.**StorageBoxOptionInlineForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)

Bases: django.forms.models.ModelForm

class Meta

fields = '__all__'

model
alias of *tardis.tardis_portal.models.storage.StorageBoxOption*

widgets = {'key': <django.forms.widgets.TextInput object at 0x7f11b2041b50>, 'valu

base_fields = {'key': <django.forms.fields.CharField object at 0x7f11b204f190>, 'stor

declared_fields = {}

media

tardis.tardis_portal.ands_doi module

ands_doi.py

class tardis.tardis_portal.ands_doi.**DOIService**(*experiment*)
Bases: object

Mints DOIs using ANDS' Cite My Data service POSTs DataCite XML to a web services endpoint

get_doi()

Returns DOI or None

Return type string

get_or_mint_doi(*url*)

Parameters **url** (*string*) – the URL the DOI will resolve to

Returns the DOI string

Return type string

class tardis.tardis_portal.ands_doi.**DOIXMLProvider**(*experiment*)
Bases: object

provides datacite XML metadata for a given experiment

datacite_xml()

Returns datacite XML for self.experiment

Return type string

tardis.tardis_portal.api module

RESTful API for MyTardis models and data. Implemented with Tastypie.

class tardis.tardis_portal.api.**ACLAuthorization**
Bases: tastypie.authorization.Authorization

Authorisation class for Tastypie.

create_detail(*object_list*, *bundle*)

create_list(*object_list*, *bundle*)

delete_detail(*object_list*, *bundle*)

delete_list(*object_list*, *bundle*)

read_detail(*object_list*, *bundle*)

read_list(*object_list*, *bundle*)

update_detail(*object_list*, *bundle*)

update_list(*object_list*, *bundle*)

class tardis.tardis_portal.api.**DataFileResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

filtering = {'dataset': 2, 'directory': ('exact', 'startswith'), 'filename': ('e

object_class
alias of *tardis.tardis_portal.models.datafile.DataFile*

ordering = ['filename', 'modification_time']

queryset

resource_name = 'dataset_file'

base_fields = {'created_time': <tastypie.fields.DateTimeField object at 0x7f11af64525

declared_fields = {'datafile': <tastypie.fields.FileField object at 0x7f11af63ae50>,

deserialize(*request*, *data*, *format=None*)
from https://github.com/toastdriven/django-tastypie/issues/42 modified to deserialize json sent via POST.
Would fail if data is sent in a different format. uses a hack to get back pure json from request.POST

download_file(*request*, ***kwargs*)
curl needs the -J switch to get the filename right auth needs to be added manually here

hydrate(*bundle*)

obj_create(*bundle*, ***kwargs*)
Creates a new DataFile object from the provided bundle.data dict.

If a duplicate key error occurs, responds with HTTP Error 409: CONFLICT

post_list(*request*, ***kwargs*)

prepend_urls()

put_detail(*request*, ***kwargs*)
from https://github.com/toastdriven/django-tastypie/issues/42

temp_url = None

verify_file(*request*, ***kwargs*)
triggers verification of file, e.g. after non-POST upload complete

class tardis.tardis_portal.api.**DatafileParameterResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.ParameterResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.parameters.DatafileParameter*

queryset

base_fields = {'datetime_value': <tastypie.fields.DateTimeField object at 0x7f11af645

declared_fields = {'parameterset': <tastypie.fields.ForeignKey object at 0x7f11af6456

class tardis.tardis_portal.api.**DatafileParameterSetResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.ParameterSetResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.parameters.DatafileParameterSet*

queryset

base_fields = {'datafile': <tastypie.fields.ForeignKey object at 0x7f11af645090>, u'i

declared_fields = {'datafile': <tastypie.fields.ForeignKey object at 0x7f11af645090>,

class tardis.tardis_portal.api.**DatasetParameterResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.ParameterResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.parameters.DatasetParameter*

queryset

base_fields = {'datetime_value': <tastypie.fields.DateTimeField object at 0x7f11af630

declared_fields = {'parameterset': <tastypie.fields.ForeignKey object at 0x7f11af6303

class tardis.tardis_portal.api.**DatasetParameterSetResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.ParameterSetResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.parameters.DatasetParameterSet*

queryset

base_fields = {'dataset': <tastypie.fields.ForeignKey object at 0x7f11af6a8d90>, u'id

declared_fields = {'dataset': <tastypie.fields.ForeignKey object at 0x7f11af6a8d90>,

class tardis.tardis_portal.api.**DatasetResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

always_return_data = True

filtering = {'description': ('exact',), 'directory': ('exact',), 'experiments':

object_class
alias of *tardis.tardis_portal.models.dataset.Dataset*

ordering = ['description']

queryset

base_fields = {'description': <tastypie.fields.CharField object at 0x7f11af63ad50>, '

declared_fields = {'experiments': <tastypie.fields.ToManyField object at 0x7f11af63a8

get_datafiles(*request*, ***kwargs*)

prepend_urls()

class tardis.tardis_portal.api.**ExperimentParameterResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.ParameterResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.parameters.ExperimentParameter*

queryset

base_fields = {'datetime_value': <tastypie.fields.DateTimeField object at 0x7f11af6a8

declared_fields = {'parameterset': <tastypie.fields.ForeignKey object at 0x7f11af6a85

class tardis.tardis_portal.api.**ExperimentParameterSetResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.ParameterSetResource*

API for ExperimentParameterSets

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.parameters.ExperimentParameterSet*

queryset

base_fields = {'experiment': <tastypie.fields.ForeignKey object at 0x7f11af6a8210>, u

declared_fields = {'experiment': <tastypie.fields.ForeignKey object at 0x7f11af6a8210

class tardis.tardis_portal.api.**ExperimentResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

API for Experiments also creates a default ACL and allows ExperimentParameterSets to be read and written.

TODO: catch duplicate schema submissions for parameter sets

class Meta
Bases: tardis.tardis_portal.api.Meta

always_return_data = True

filtering = {'id': ('exact',), 'title': ('exact',)}

object_class
alias of *tardis.tardis_portal.models.experiment.Experiment*

ordering = ['title', 'created_time', 'update_time']

queryset

base_fields = {'approved': <tastypie.fields.BooleanField object at 0x7f11af6a8e50>, '

declared_fields = {'created_by': <tastypie.fields.ForeignKey object at 0x7f11af6a8850

dehydrate(*bundle*)

hydrate_m2m(*bundle*)
create ACL before any related objects are created in order to use ACL permissions for those objects.

obj_create(*bundle*, ***kwargs*)
experiments need at least one ACL to be available through the ExperimentManager (Experiment.safe) Currently not tested for failed db transactions as sqlite does not enforce limits.

obj_get_list(*bundle*, ***kwargs*)
responds to EPN query for Australian Synchrotron

class tardis.tardis_portal.api.**FacilityResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

always_return_data = True

filtering = {'id': ('exact',), 'manager_group': 2, 'name': ('exact',)}

object_class
alias of *tardis.tardis_portal.models.facility.Facility*

queryset

base_fields = {u'id': <tastypie.fields.IntegerField object at 0x7f11af63a610>, 'manag

declared_fields = {'manager_group': <tastypie.fields.ForeignKey object at 0x7f11af63a

class tardis.tardis_portal.api.**GroupResource**(*api_name=None*)
Bases: tastypie.resources.ModelResource

class Meta

authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>

authorization
Authorisation class for Tastypie.

filtering = {'id': ('exact',), 'name': ('exact',)}

object_class
alias of django.contrib.auth.models.Group

queryset

base_fields = {u'id': <tastypie.fields.IntegerField object at 0x7f11af6993d0>, 'name'

declared_fields = {}

class tardis.tardis_portal.api.**InstrumentResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

always_return_data = True

filtering = {'facility': 2, 'id': ('exact',), 'name': ('exact',)}

object_class
alias of *tardis.tardis_portal.models.instrument.Instrument*

queryset

base_fields = {'facility': <tastypie.fields.ForeignKey object at 0x7f11af63a590>, u'i

declared_fields = {'facility': <tastypie.fields.ForeignKey object at 0x7f11af63a590>}

class tardis.tardis_portal.api.**LocationResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.storage.StorageBox*

queryset

base_fields = {'description': <tastypie.fields.CharField object at 0x7f11af645f50>, '

declared_fields = {}

class tardis.tardis_portal.api.**MyTardisAuthentication**
Bases: object

custom tastypie authentication that works with both anonymous use and a number of available auth mechanisms.

get_identifier(*request*)

is_authenticated(*request*, ***kwargs*)
handles backends explicitly so that it can return False when credentials are given but wrong and return Anonymous User when credentials are not given or the session has expired (web use).

class tardis.tardis_portal.api.**MyTardisModelResource**(*api_name=None*)
Bases: tastypie.resources.ModelResource

class Meta

authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>

authorization
Authorisation class for Tastypie.

serializer = <tardis.tardis_portal.api.PrettyJSONSerializer object>

base_fields = {u'resource_uri': <tastypie.fields.CharField object at 0x7f11af6997d0>}

declared_fields = {}

lookup_kwargs_with_identifiers(*bundle*, *kwargs*)

class tardis.tardis_portal.api.**ObjectACLResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta

authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>

authorization
Authorisation class for Tastypie.

filtering = {'entityId': ('exact',), 'pluginId': ('exact',)}

object_class
alias of *tardis.tardis_portal.models.access_control.ObjectACL*

queryset

base_fields = {'aclOwnershipType': <tastypie.fields.IntegerField object at 0x7f11af64

declared_fields = {'content_object': <tastypie.contrib.contenttypes.fields.GenericFor

hydrate(*bundle*)

class tardis.tardis_portal.api.**ParameterNameResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

filtering = {'schema': 2}

object_class
alias of *tardis.tardis_portal.models.parameters.ParameterName*

queryset

base_fields = {'choices': <tastypie.fields.CharField object at 0x7f11af699f90>, 'compa

declared_fields = {'schema': <tastypie.fields.ForeignKey object at 0x7f11af699910>}

class tardis.tardis_portal.api.**ParameterResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

base_fields = {'name': <tastypie.fields.ForeignKey object at 0x7f11af699d10>, u'resou

declared_fields = {'name': <tastypie.fields.ForeignKey object at 0x7f11af699d10>, 'va

hydrate(*bundle*)
sets the parametername by uri or name if untyped value is given, set value via parameter method, otherwise use modelresource automatisms

class tardis.tardis_portal.api.**ParameterSetResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

base_fields = {u'resource_uri': <tastypie.fields.CharField object at 0x7f11af6a8290>,

declared_fields = {'schema': <tastypie.fields.ForeignKey object at 0x7f11af6a80d0>}

hydrate_schema(*bundle*)

class tardis.tardis_portal.api.**PrettyJSONSerializer**(*formats=None*, *content_types=None*, *datetime_formatting=None*)
Bases: tastypie.serializers.Serializer

json_indent = 2

to_json(*data*, *options=None*)

class tardis.tardis_portal.api.**ReplicaResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

filtering = {'url': ('exact', 'startswith'), 'verified': ('exact',)}

object_class
alias of *tardis.tardis_portal.models.datafile.DataFileObject*

queryset

base_fields = {'created_time': <tastypie.fields.DateTimeField object at 0x7f11af64c2d

declared_fields = {'datafile': <tastypie.fields.ForeignKey object at 0x7f11af645d90>}

dehydrate(*bundle*)

hydrate(*bundle*)

class tardis.tardis_portal.api.**SchemaResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

filtering = {'id': ('exact',), 'namespace': ('exact',)}

object_class
alias of *tardis.tardis_portal.models.parameters.Schema*

queryset

base_fields = {'hidden': <tastypie.fields.BooleanField object at 0x7f11af699b10>, u'i

declared_fields = {}

lookup_kwargs_with_identifiers(*bundle*, *kwargs*)

class tardis.tardis_portal.api.**StorageBoxAttributeResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.storage.StorageBoxAttribute*

queryset

base_fields = {u'id': <tastypie.fields.IntegerField object at 0x7f11af63a2d0>, 'key':

declared_fields = {'storage_box': <tastypie.fields.ForeignKey object at 0x7f11af630e9

class tardis.tardis_portal.api.**StorageBoxOptionResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.storage.StorageBoxOption*

queryset

accessible_keys = ['location']

base_fields = {u'id': <tastypie.fields.IntegerField object at 0x7f11af630f10>, 'key':

declared_fields = {'storage_box': <tastypie.fields.ForeignKey object at 0x7f11af630a9

class tardis.tardis_portal.api.**StorageBoxResource**(*api_name=None*)
Bases: *tardis.tardis_portal.api.MyTardisModelResource*

class Meta
Bases: tardis.tardis_portal.api.Meta

object_class
alias of *tardis.tardis_portal.models.storage.StorageBox*

queryset

base_fields = {'attributes': <tastypie.fields.ToManyField object at 0x7f11af630850>,

declared_fields = {'attributes': <tastypie.fields.ToManyField object at 0x7f11af63085

class tardis.tardis_portal.api.**UserResource**(*api_name=None*)
Bases: tastypie.resources.ModelResource

class Meta

allowed_methods = ['get']

authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>

authorization
Authorisation class for Tastypie.

fields = ['username', 'first_name', 'last_name', 'email']

filtering = {'email': ('iexact',), 'username': ('exact',)}

object_class
alias of django.contrib.auth.models.User

queryset

serializer = <tardis.tardis_portal.api.PrettyJSONSerializer object>

base_fields = {'email': <tastypie.fields.CharField object at 0x7f11af699710>, 'first_

declared_fields = {'groups': <tastypie.fields.ManyToManyField object at 0x7f11af69935

dehydrate(*bundle*)
use cases: public user:

anonymous: name, uri, email, id

authenticated:

other user: name, uri, email, id [, username if facility manager]

same user: name, uri, email, id, username

private user:

anonymous: none

authenticated:

other user: name, uri, id [, username, email if facility manager]

same user: name, uri, email, id, username

tardis.tardis_portal.api.**lookup_by_unique_id_only**(*resource*)
returns custom lookup function. initialise with resource type

tardis.tardis_portal.constants module

constants.py

@author: Gerson Galang

tardis.tardis_portal.context_processors module

tardis.tardis_portal.context_processors.**global_contexts**(*request*)

tardis.tardis_portal.context_processors.**google_analytics**(*request*)
adds context for portal_template.html

tardis.tardis_portal.context_processors.**manage_account_processor**(*request*)

tardis.tardis_portal.context_processors.**registration_processor**(*request*)

tardis.tardis_portal.context_processors.**single_search_processor**(*request*)

tardis.tardis_portal.context_processors.**tokenuser_processor**(*request*)

tardis.tardis_portal.context_processors.**user_details_processor**(*request*)

tardis.tardis_portal.creativecommonshandler module

Creative Commons Handler

A wrapper for creative commons interactions on a ParameterSet

class tardis.tardis_portal.creativecommonshandler.**CreativeCommonsHandler**(*experiment_id=None*, *create=True*)

experiment_id = None

get_or_create_cc_parameterset(*create=True*)
Gets the creative commons parameterset for the experiment :param create: If true, creates a new parameterset object to hold the cc license if one doesn't exist :type create: boolean :return: The parameterset manager for the cc parameterset :rtype: tardis.tardis_portal.ParameterSetManager.ParameterSetManager

has_cc_license()

Returns True if there's a cc license parameterset for the experiment

Return type boolean

psm = None

save_license(*request*)
Saves a license parameterset with the POST variables from the creative commons form :param request: a HTTP Request instance :type request: django.http.HttpRequest

schema = 'http://www.tardis.edu.au/schemas/creative_commons/2011/05/17'

tardis.tardis_portal.download module

tardis.tardis_portal.email module

tardis.tardis_portal.email.**build_template_html**(*template_filename*, *context*)

tardis.tardis_portal.email.**build_template_text**(*template_filename*, *context*)

tardis.tardis_portal.email.**email_user**(*subject*, *template_filename*, *context*, *user*)

tardis.tardis_portal.errors module

Created on 02/09/2010

exception tardis.tardis_portal.errors.**ParameterChoicesFormatError**(*msg*)
Bases: exceptions.Exception

exception tardis.tardis_portal.errors.**SearchQueryTypeUnprovidedError**(*msg*)
Bases: exceptions.Exception

exception tardis.tardis_portal.errors.**UnsupportedSearchQueryTypeError**(*msg*)
Bases: exceptions.Exception

tardis.tardis_portal.fetcher module

tardis.tardis_portal.fetcher.**get_credential_handler**()

tardis.tardis_portal.fetcher.**get_privileged_opener**()

tardis.tardis_portal.fields module

class tardis.tardis_portal.fields.**MultiValueCommaSeparatedField**(*fields=()*, **args*, ***kwargs*)
Bases: django.forms.fields.MultiValueField

compress(*data_list*)

tardis.tardis_portal.forms module

forms module

class tardis.tardis_portal.forms.**AddGroupPermissionsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)
Bases: django.forms.forms.Form

base_fields = {'addgroup': <django.forms.fields.CharField object at 0x7f11af6258d0>,

declared_fields = {'addgroup': <django.forms.fields.CharField object at 0x7f11af6258d

media

class tardis.tardis_portal.forms.**AddUserPermissionsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)
Bases: django.forms.forms.Form

base_fields = {'authMethod': <django.forms.fields.CharField object at 0x7f11af625610>

declared_fields = {'authMethod': <django.forms.fields.CharField object at 0x7f11af625

media

class tardis.tardis_portal.forms.**ChangeGroupPermissionsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)

Bases: django.forms.forms.Form

class SelectDateWidget(*attrs=None*, *years=None*, *months=None*, *empty_label=None*)

Bases: django.forms.widgets.Widget

A Widget that splits date input into three <select> boxes.

This also serves as an example of a Widget that has more than one HTML element and hence implements value_from_datadict.

create_select(*name*, *field*, *value*, *val*, *choices*, *none_value*)

day_field = u'%s_day'

id_for_label(*id_*)

media

month_field = u'%s_month'

none_value = (0, u'---')

render(*name*, *value*, *attrs=None*)

value_from_datadict(*data*, *files*, *name*)

year_field = u'%s_year'

base_fields = {'canDelete': <django.forms.fields.BooleanField object at 0x7f11af62529

declared_fields = {'canDelete': <django.forms.fields.BooleanField object at 0x7f11af6

media

class tardis.tardis_portal.forms.**ChangeUserPermissionsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)

Bases: django.forms.models.ModelForm

class Meta

class ObjectACL(**args*, ***kwargs*)

Bases: django.db.models.base.Model

The ObjectACL (formerly ExperimentACL) table is the core of the Tardis Authorisation framework

Attribute pluginId the the name of the auth plugin being used

Attribute entityId a foreign key to auth plugins

Attribute object_type a foreign key to ContentType
Attribute object_id the primary key/id of the object_type
Attribute canRead gives the user read access
Attribute canWrite gives the user write access
Attribute canDelete gives the user delete permission
Attribute isOwner the experiment owner flag.
Attribute effectiveDate the date when access takes into effect
Attribute expiryDate the date when access ceases
Attribute aclOwnershipType system-owned or user-owned.

System-owned ACLs will prevent users from removing or editing ACL entries to a particular experiment they own. User-owned ACLs will allow experiment owners to remove/add/edit ACL entries to the experiments they own.

exception DoesNotExist
Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned
Bases: `django.core.exceptions.MultipleObjectsReturned`

OWNER_OWNED = 1

SYSTEM_OWNED = 2

content_object
Provides a generic relation to any object through content-type/object-id fields.

content_type

get_aclOwnershipType_display(***morekwargs*)

classmethod get_effective_query()

get_related_object()
If possible, resolve the pluginId/entityId combination to a user or group object.

get_related_object_group()
If possible, resolve the pluginId/entityId combination to a user or group object.

objects = <django.db.models.manager.Manager object>

class SelectDateWidget(*attrs=None*, *years=None*, *months=None*, *empty_label=None*)
Bases: `django.forms.widgets.Widget`

A Widget that splits date input into three <select> boxes.

This also serves as an example of a Widget that has more than one HTML element and hence implements value_from_datadict.

create_select(*name*, *field*, *value*, *val*, *choices*, *none_value*)

day_field = u'%s_day'

id_for_label(*id_*)

media

month_field = u'%s_month'

none_value = (0, u'---')

render(*name*, *value*, *attrs=None*)

value_from_datadict(*data*, *files*, *name*)

year_field = u'%s_year'

fields = ['canDelete', 'canRead', 'canWrite', 'effectiveDate', 'expiryDate', 'isOwn

model
alias of *tardis.tardis_portal.models.access_control.ObjectACL*

widgets = {'effectiveDate': <django.forms.extras.widgets.SelectDateWidget object a

base_fields = {'canDelete': <django.forms.fields.BooleanField object at 0x7f11af62505

declared_fields = {}

media

class tardis.tardis_portal.forms.**CreateGroupPermissionsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)

Bases: django.forms.forms.Form

base_fields = {'addgroup': <django.forms.fields.CharField object at 0x7f11af625ad0>,

declared_fields = {'addgroup': <django.forms.fields.CharField object at 0x7f11af625ad

media

class tardis.tardis_portal.forms.**CreateUserPermissionsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)

Bases: *tardis.tardis_portal.forms.RegistrationForm*

base_fields = {'authMethod': <django.forms.fields.CharField object at 0x7f11af5ac0d0>

declared_fields = {'authMethod': <django.forms.fields.CharField object at 0x7f11af5ac

media

class tardis.tardis_portal.forms.**DatafileSearchForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)

Bases: django.forms.forms.Form

base_fields = {'filename': <django.forms.fields.CharField object at 0x7f11af5ac190>}

declared_fields = {'filename': <django.forms.fields.CharField object at 0x7f11af5ac19

media

class tardis.tardis_portal.forms.**DatasetForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)

Bases: django.forms.models.ModelForm

class Meta

fields = ['description', 'directory', 'instrument']

model

alias of *tardis.tardis_portal.models.dataset.Dataset*

base_fields = {'description': <django.forms.fields.CharField object at 0x7f11af5b7250

declared_fields = {'description': <django.forms.fields.CharField object at 0x7f11af5b

media

class tardis.tardis_portal.forms.**ExperimentAuthor**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)

Bases: django.forms.models.ModelForm

class Meta

fields = ['author', 'institution', 'email', 'order', 'url']

model

alias of *tardis.tardis_portal.models.experiment.ExperimentAuthor*

base_fields = {'author': <django.forms.fields.CharField object at 0x7f11af5b7450>, 'e

declared_fields = {}

media

class tardis.tardis_portal.forms.**ExperimentForm**(*data=None*, *files=None*, *auto_id='%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=':'*, *empty_permitted=False*, *instance=None*, *extra=0*)

Bases: django.forms.models.ModelForm

This handles the complex experiment forms.

All internal datasets forms are prefixed with *dataset_*, and all internal dataset file fields are prefixed with *file_*. These are parsed out of the post data and added to the form as internal lists.

class FullExperiment(***kwargs*)

Bases: UserDict.UserDict

This is a dict wrapper that store the values returned from the *tardis.tardis_portal.forms.ExperimentForm.save()* function. It provides a convience method for saving the model objects.

save_m2m()
{'experiment': experiment, 'experiment_authors': experiment_authors, 'authors': authors, 'datasets': datasets, 'datafiles': datafiles}

class Meta

fields = ('title', 'institution_name', 'description')

model
alias of *tardis.tardis_portal.models.experiment.Experiment*

base_fields = {'description': <django.forms.fields.CharField object at 0x7f11af5c6050

declared_fields = {'url': <django.forms.fields.CharField object at 0x7f11af5b7c50>}

is_valid()
Test the validity of the form, the form may be invalid even if the error attribute has no contents. This is because the returnd value is dependent on the validity of the nested forms.

This validity also takes into account forign keys that might be dependent on an unsaved model.

Returns validity

Return type bool

media

save(*commit=True*)

class tardis.tardis_portal.forms.**IRDatafileSearchForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)
Bases: *tardis.tardis_portal.forms.DatafileSearchForm*

base_fields = {'filename': <django.forms.fields.CharField object at 0x7f11af5ac190>}

declared_fields = {'filename': <django.forms.fields.CharField object at 0x7f11af5ac19

media

class tardis.tardis_portal.forms.**ImportParamsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)
Bases: django.forms.forms.Form

base_fields = {'params': <django.forms.fields.FileField object at 0x7f11af5ac990>, 'pa

declared_fields = {'params': <django.forms.fields.FileField object at 0x7f11af5ac990>

media

class tardis.tardis_portal.forms.**LoginForm**(**args*, ***kwargs*)
Bases: django.contrib.auth.forms.AuthenticationForm

base_fields = {'password': <django.forms.fields.CharField object at 0x7f11b20c3650>,

declared_fields = {'password': <django.forms.fields.CharField object at 0x7f11b20c365

media

class tardis.tardis_portal.forms.**MXDatafileSearchForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)

Bases: *tardis.tardis_portal.forms.DatafileSearchForm*

base_fields = {'crystalName': <django.forms.fields.CharField object at 0x7f11af5ac450

declared_fields = {'crystalName': <django.forms.fields.CharField object at 0x7f11af5a

media

class tardis.tardis_portal.forms.**ManageAccountForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)

Bases: `django.forms.models.ModelForm`

Form for changing account details.

class Meta

fields = ('first_name', 'last_name', 'email')

model

alias of `django.contrib.auth.models.User`

base_fields = {'email': <django.forms.fields.EmailField object at 0x7f11af5c6a50>, 'f

declared_fields = {}

media

class tardis.tardis_portal.forms.**ManageGroupPermissionsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)

Bases: `django.forms.forms.Form`

base_fields = {'adduser': <django.forms.fields.CharField object at 0x7f11af625e50>, '

declared_fields = {'adduser': <django.forms.fields.CharField object at 0x7f11af625e50

media

class tardis.tardis_portal.forms.**NoInput**(*attrs=None*)
Bases: django.forms.widgets.Widget

media

render(*name*, *value*, *attrs=None*)

class tardis.tardis_portal.forms.**RawSearchForm**(**args*, ***kwargs*)
Bases: haystack.forms.SearchForm

base_fields = {'q': <django.forms.fields.CharField object at 0x7f11b05c3610>}

declared_fields = {'q': <django.forms.fields.CharField object at 0x7f11b05c3610>}

media

search()

class tardis.tardis_portal.forms.**RegisterExperimentForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)
Bases: django.forms.forms.Form

base_fields = {'experiment_owner': <django.forms.fields.CharField object at 0x7f11af5

declared_fields = {'experiment_owner': <django.forms.fields.CharField object at 0x7f1

media

class tardis.tardis_portal.forms.**RegistrationForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*)
Bases: django.forms.forms.Form

Form for registering a new user account.

Validates that the requested username is not already in use, and requires the password to be entered twice to catch typos.

Subclasses should feel free to add any additional validation they need, but should avoid defining a save() method – the actual saving of collected user data is delegated to the active registration backend.

base_fields = {'email': <django.forms.fields.EmailField object at 0x7f11af622710>, 'p

clean()
Verifiy that the values entered into the two password fields match. Note that an error here will end up in non_field_errors() because it doesn't apply to a single field.

clean_username()
Validate that the username is alphanumeric and is not already in use.

declared_fields = {'email': <django.forms.fields.EmailField object at 0x7f11af622710>

media

save(***kwargs*)

class tardis.tardis_portal.forms.**RightsForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *instance=None*)

Bases: django.forms.models.ModelForm

Form for changing public access and licence.

class Meta

fields = ('public_access', 'license')

model

alias of *tardis.tardis_portal.models.experiment.Experiment*

widgets = {'license': <django.forms.widgets.HiddenInput object at 0x7f11af5c62d0>}

base_fields = {'license': <django.forms.models.ModelChoiceField object at 0x7f11af5c6

clean()

declared_fields = {}

media

class tardis.tardis_portal.forms.**StaticField**(*required=True*, *widget=None*, *label=None*, *initial=None*, *help_text=u''*, *error_messages=None*, *show_hidden_initial=False*, *validators=[]*, *localize=False*, *label_suffix=None*)

Bases: django.forms.fields.Field

clean(*value*)

widget

alias of *NoInput*

tardis.tardis_portal.forms.**createLinkedUserAuthenticationForm**(*authMethods*)

Create a LinkedUserAuthenticationForm and use the contents of authMethods to the list of options in the drop-down menu for authenticationMethod.

tardis.tardis_portal.forms.**createSearchDatafileForm**(*searchQueryType*)

tardis.tardis_portal.forms.**createSearchDatafileSelectionForm**(*initial=None*)

tardis.tardis_portal.forms.**createSearchExperimentForm**()

tardis.tardis_portal.forms.**create_datafile_add_form**(*schema*, *parentObject*, *request=None*)

tardis.tardis_portal.forms.**create_parameterset_edit_form**(*parameterset*, *request=None*)

tardis.tardis_portal.forms.**getAuthMethodChoices**()

tardis.tardis_portal.forms.**save_datafile_add_form**(*schema*, *parentObject*, *request*)

tardis.tardis_portal.forms.**save_datafile_edit_form**(*parameterset*, *request*)

tardis.tardis_portal.hacks module

hacks.py For hacks which should be removed at a later stage

tardis.tardis_portal.hacks.**oracle_dbops_hack**(*f*)
affects only oracle - safe for other database engines

tardis.tardis_portal.iiif module

tardis.tardis_portal.iiif.**compliance_header**(*f*)

tardis.tardis_portal.iiif.**compute_etag**(*request*, *datafile_id*, **args*, ***kwargs*)

tardis.tardis_portal.iiif.**download_image**(*request*, **args*, ***kwargs*)

tardis.tardis_portal.iiif.**download_info**(*request*, **args*, ***kwargs*)

tardis.tardis_portal.logging_middleware module

class tardis.tardis_portal.logging_middleware.**LoggingMiddleware**
Bases: object

process_exception(*request*, *exception*)

process_response(*request*, *response*)

tardis.tardis_portal.managers module

managers.py

class tardis.tardis_portal.managers.**ExperimentManager**
Bases: *tardis.tardis_portal.managers.OracleSafeManager*

Implements a custom manager for the Experiment model which checks the authorisation rules for the requesting user first

To make this work, the request must be passed to all class functions. The username and the group memberships are then resolved via the user.userprofile.ext_groups and user objects.

The tardis.tardis_portal.auth.AuthService is responsible for filling the request.groups object.

all(*user*)
Returns all experiments a user - either authenticated or anonymous - is allowed to see and search

Parameters **user** (*User*) – a User instance

Returns QuerySet of Experiments

Return type QuerySet

external_users(*experiment_id*)
returns a list of groups which have external ACL rules

Parameters **experiment_id** (*int*) – the ID of the experiment to be edited

Returns list of groups with external ACLs

Return type list

get(*user*, *experiment_id*)
Returns an experiment under the consideration of the ACL rules Raises PermissionDenied if the user does not have access.

Parameters

- **user** (*User*) – a User instance
- **experiment_id** (*int*) – the ID of the experiment to be edited

Returns Experiment

Return type *Experiment*

Raises **PermissionDenied** –

group_acls_system_owned(*experiment_id*)
Returns a list of ACL rules associated with this experiment.

Parameters **experiment_id** (*int*) – the ID of the experiment

Returns QuerySet of system-owned ACLs for experiment

Return type QuerySet

group_acls_user_owned(*experiment_id*)
Returns a list of ACL rules associated with this experiment.

Parameters **experiment_id** (*int*) – the ID of the experiment

Returns QuerySet of ACLs

Return type QuerySet

owned(*user*)
Return all experiments which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters **user** (*User*) – a User instance

Returns QuerySet of Experiments owned by user

Return type QuerySet

owned_and_shared(*user*)

owned_by_group(*group*)
Return all experiments that are owned by a particular group

owned_by_user(*user*)
Return all experiments which are owned by a particular user id

Parameters **user** (*User*) – a User Object

Returns QuerySet of Experiments owned by user

Return type QuerySet

owned_by_user_id(*userId*)
Return all experiments which are owned by a particular user id

Parameters **userId** (*int*) – a User ID

Returns QuerySet of Experiments owned by user id

Return type QuerySet

public()

shared(*user*)

system_owned_groups(*experiment_id*)
returns a list of sytem-owned groups which have ACL rules associated with this experiment

Parameters **experiment_id** (*string*) – the ID of the experiment to be edited

Returns system owned groups for experiment

Return type QuerySet

user_acls(*experiment_id*)
Returns a list of ACL rules associated with this experiment.

Parameters **experiment_id** (*string*) – the ID of the experiment

Returns QuerySet of ACLs

Return type QuerySet

user_owned_groups(*experiment_id*)
returns a list of user owned-groups which have ACL rules associated with this experiment

Parameters **experiment_id** (*int*) – the ID of the experiment to be edited

Returns QuerySet of non system Groups

Return type QuerySet

users(*experiment_id*)
Returns a list of users who have ACL rules associated with this experiment.

Parameters **experiment_id** (*int*) – the ID of the experiment

Returns QuerySet of Users with experiment access

Return type QuerySet

class tardis.tardis_portal.managers.**OracleSafeManager**
Bases: django.db.models.manager.Manager

Implements a custom manager which automatically defers the retreival of any TextField fields on calls to get_queryset. This is to avoid the known issue that 'distinct' calls on query_sets containing TextFields fail when Oracle is being used as the backend.

get_queryset()
Returns a new QuerySet object. Subclasses can override this method to easily customize the behavior of the Manager.

class tardis.tardis_portal.managers.**ParameterNameManager**
Bases: django.db.models.manager.Manager

get_by_natural_key(*namespace*, *name*)

class tardis.tardis_portal.managers.**SchemaManager**
Bases: django.db.models.manager.Manager

get_by_natural_key(*namespace*)

tardis.tardis_portal.rfc3339 module

The function *rfc3339* formats dates according to the **RFC 3339**. *rfc3339* tries to have as much as possible sensible defaults.

tardis.tardis_portal.rfc3339.**rfc3339**(*date*, *utc=False*, *use_system_timezone=True*)

Return a string formatted according to the **RFC 3339**. If called with *utc=True*, it normalizes *date* to the UTC date. If *date* does not have any timezone information, uses the local timezone:

```
>>> date = datetime.datetime(2008, 4, 2, 20)
>>> rfc3339(date, utc=True, use_system_timezone=False)
'2008-04-02T20:00:00Z'
>>> rfc3339(date)
'2008-04-02T20:00:00...'
```

If called with *user_system_time=False* don't use the local timezone and consider the offset to UTC to be zero:

```
>>> rfc3339(date, use_system_timezone=False)
'2008-04-02T20:00:00+00:00'
```

date must be a a *datetime.datetime*, *datetime.date* or a timestamp as returned by *time.time()*:

```
>>> rfc3339(0, utc=True, use_system_timezone=False)
'1970-01-01T00:00:00Z'
>>> rfc3339(datetime.date(2008, 9, 6), use_system_timezone=False)
'2008-09-06T00:00:00+00:00'
>>> rfc3339('foo bar')
Traceback (most recent call last):
...
TypeError: excepted datetime, got str instead
```

tardis.tardis_portal.search_backend module

class tardis.tardis_portal.search_backend.**HighlightSearchBackend**(*connection_alias*, ***connection_options*)

Bases: haystack.backends.elasticsearch_backend.ElasticsearchSearchBackend

search(*query_string*, *sort_by=None*, *start_offset=0*, *end_offset=None*, *fields=''*, *highlight=False*, *facets=None*, *date_facets=None*, *query_facets=None*, *narrow_queries=None*, *spelling_query=None*, *limit_to_registered_models=None*, *result_class=None*, ***kwargs*)

tardis.tardis_portal.search_query module

search_backend.py

class tardis.tardis_portal.search_query.**FacetFixedSearchQuery**(*using=u'default'*)

Bases: haystack.backends.elasticsearch_backend.ElasticsearchSearchQuery

get_facet_counts()

tardis.tardis_portal.sftp module

tardis.tardis_portal.shortcuts module

class tardis.tardis_portal.shortcuts.**RestfulExperimentParameterSet**(*schema_func*, *form_cls*)

Bases: object

Helper class which enables a Backbone.sync-compatible interface to be created for a ExperimentParameterSet just by specifying a function which provides the schema and a form.

(A function for the schema is required rather than the actual schema, as to run unit tests effectively the object needs to be able to create the schema after instantiation.)

For UI consistency, it's best to make sure the schema has hidden == true.

schema
Use schema function to get the schema.

view_functions

tardis.tardis_portal.shortcuts.**get_experiment_referer**(*request*, *dataset_id*)

tardis.tardis_portal.shortcuts.**render_error_message**(*request*, *message*, *status=400*)
Render a simple text error message in a generic error page. Any newlines are turned into <br>.

tardis.tardis_portal.shortcuts.**render_response_index**(*request*, **args*, ***kwargs*)

tardis.tardis_portal.shortcuts.**render_response_search**(*request*, *url*, *c*)

tardis.tardis_portal.shortcuts.**render_to_file**(*template*, *filename*, *context*)

tardis.tardis_portal.shortcuts.**return_response_error**(*request*)

tardis.tardis_portal.shortcuts.**return_response_error_message**(*request*, *redirect_path*, *context*)

tardis.tardis_portal.shortcuts.**return_response_not_found**(*request*)

tardis.tardis_portal.signals module

tardis.tardis_portal.staging module

staging.py

tardis.tardis_portal.staging.**get_dataset_path**(*dataset*)

tardis.tardis_portal.staging.**get_full_staging_path**(*username*)

tardis.tardis_portal.staging.**get_staging_path**()
return the path to the staging directory

tardis.tardis_portal.staging.**get_staging_url_and_size**(*username*, *filepath*)
Returns a file:// URL and the size of the file.

tardis.tardis_portal.staging.**get_sync_location**()

tardis.tardis_portal.staging.**get_sync_root**(*prefix=''*)

tardis.tardis_portal.staging.**get_sync_url_and_protocol**(*sync_path*, *filepath*)

tardis.tardis_portal.staging.**staging_list**(*pathname='/home/docs/checkouts/readthedocs.org/user_builds/mytardis*
dirname='/home/docs/checkouts/readthedocs.org/user_builds/mytardis/
root=False)

tardis.tardis_portal.staging.**write_uploaded_file_to_dataset**(*dataset*, *uploaded_file_post*, *subdir=None*)
Broken, now that the storagebox takes care of writing files. Writes file POST data to the dataset directory in the file store

tardis.tardis_portal.tasks module

tardis.tardis_portal.tasks.**init_filters**()
load filters and avoid circular import

tardis.tardis_portal.util module

tardis.tardis_portal.widgets module

class tardis.tardis_portal.widgets.**CommaSeparatedInput**(*attrs=None*)
Bases: django.forms.widgets.TextInput

media

render(*name*, *value*, *attrs=None*)

value_from_datadict(*data*, *files*, *name*)

class tardis.tardis_portal.widgets.**Label**(*attrs=None*)
Bases: django.forms.widgets.Widget

media

render(*name*, *value*, *attrs=None*)

tag = 'label'

class tardis.tardis_portal.widgets.**Span**(*attrs=None*)
Bases: *tardis.tardis_portal.widgets.Label*

media

tag = 'span'

tardis.tardis_portal.xmlwriter module

class tardis.tardis_portal.xmlwriter.**XMLWriter**

static write_template_to_dir(*dest_filename*, *template_path*, *context*)

Parameters

- **dest_dir** (*string*) – The directory to store the resulting document in
- **dest_filename** (*string*) – The name of the file to be output
- **template_path** – The relative path to the Django template to be

rendered :type template_path: string :param context: The Context object (dictionary of variables for template output) :type context: django.template.context.Context :returns: The full path to the created file :rtype: string

static write_template_to_file(*objectprefix*, *uniqueid*, *templatepath*, *context*)

Parameters **prefix_dir** – The subdirectory off of the OAI_DOCS_PATH to store

the resulting document in :type prefix_dir: string :param objectprefix: The name prefix of the resulting file. Files are output in the format prefix-uniqueid.xml :type objectprefix: string :param uniqueid: The unique ID of the file to be output :type uniqueid: string :param templatepath: The relative path to the Django template to be rendered :type templatepath: string :param context: The Context object (dictionary

of variables for template output) :type context: `django.template.context.Context` :returns: The full path to the created file :rtype: string

static write_xml_to_file(*objectprefix*, *uniqueid*, *xmlstring*)

Parameters **prefix_dir** – The subdirectory off of the OAI_DOCS_PATH to store

the resulting document in :type prefix_dir: string :param objectprefix: The name prefix of the resulting file. Files are output in the format prefix-uniqueid.xml :type objectprefix: string :param uniqueid: The unique ID of the file to be output :type uniqueid: string :param xmlstring: The relative path to the Django template to be rendered :type xmlstring: string :returns: The full path to the created file :rtype: string

tardis.tardis_portal.xmlwriter.**logger = <celery.utils.log.ProcessAwareLogger object>**

XML Writer

A set of static methods for writing xml files.

Module contents

models/__init__.py

Submodules

tardis.app_config module

class tardis.app_config.**AbstractTardisAppConfig**(*app_name*, *app_module*)

Bases: `django.apps.config.AppConfig`

All MyTardis app configuration classes should extend this abstract class to have their APIs and URLs automatically added to URL routing.

app_dependencies = []

tardis.app_config.**check_app_dependencies**(*app_configs*, ***kwargs*)

Checks currently installed apps for dependencies required by installed apps as defined by the app_dependencies attribute of the AppConfig object, if present.

Parameters

- **app_configs** (*AppConfig*) – a list of app_configs to check, or None for all apps to be checked
- **kwargs** (*list of args*) – unknown list of args

Returns a list of unsatisfied dependencies

Return type list of strings

tardis.app_config.**format_app_name_for_url**(*name*)

tardis.app_config.**get_tardis_apps**()

Gets a list of tuples where the first element is the app name, and the second is the module path

Returns a list of tardis apps

Return type list of apps

tardis.app_config.**is_tardis_app**(*app_config*)

Determines whether the installed app is a MyTardis app

Parameters **app_config** (*AppConfig*) – the AppConfig object of an installed app

Returns True if the app is a MyTardis app, False otherwise

Return type bool

tardis.manage module

tardis.search_sites module

tardis.settings_changeme module

tardis.test_on_mysql_settings module

tardis.test_on_postgresql_settings module

tardis.test_settings module

tardis.test_settings.**get_all_tardis_apps**()

tardis.urls module

tardis.views module

tardis.views.**error_handler**(*request*, ***kwargs*)

Module contents

models/__init__.py

11.4 Documentation for included Apps

11.4.1 Apps and Contextual Views

Introduction

In order to better represent specific data types and facilities, MyTardis allows apps to override the default views for Experiments, Datasets, DataFile metadata, and the main index page. The following sections detail settings and requirements of apps to make this happen.

Datafile Views

Rationale

By default there exists an option to show the metadata of individual `DataFile` s in the default `Dataset` view. Some kinds of files allow for rich and useful visualisation and/or processing. For this purpose there exist contextual views, views that are available depending on the type of file they refer to.

User Guide

A default installation has no contextual views. To enable them a few steps are needed:

- an app needs to be installed either in `tardis/apps/`, or the app's configuration must subclass `AbstractTardisAppConfig` thereby enabling autodetection. `AbstractTardisAppConfig` replaces `AppConfig` as described in these django docs.
- `DataFile` s need to be manually or automatically tagged with a schema that identifies them as viewable with a particular view. Filters are a convenient way to do this. See below for an example.
- settings need to be added to settings.py. A list called `DATAFILE_VIEWS` holds a tuple for each available view. The first entry of the tuple is a schema namespace and is matched against all schemas attached to the `DataFile`. If a match occurs, a link to the url given as second entry of the tuple is added to the Datafile Detail section of the default Dataset view and loaded via AJAX on demand. Example:

```
DATAFILE_VIEWS = [("http://example.org/schemas/datafile/my_awesome_schema",
                   "/apps/my-awesome-app/view"),]
```

Currently, the default view is always `DataFile` metadata. This can be changed, for example, by developing a custom `Dataset` view, which is explained in the following section.

Dataset and Experiment Views

Rationale

For some specific uses the data available can be presented and/or processed in useful ways. MyTardis allows views for Experiments and Datasets to be overriden by apps on a per-schema basis, allowing custom views for specifc data types. The example that this feature was built for are single-image and many-image datasets from the Australian Synchrotron. Single images can be displayed large and for a many-image dataset it is more useful to show a couple of example images taken at regular intervals not from the beginning of the set of files. These different datasets can be detected via their schema namespace and displayed differently.

User Guide

Akin to `DataFile` contextual views, `Dataset` and `Experiment` contextual views rely on matching a specific schema namespace in an attached ParameterSet.

Existing schemas can be used, or a special schema intended only for tagging an Experiment or Dataset for contextual view override can be attached (via an otherwise empty ParameterSet).

`Dataset` and `Experiment` contextual views are configured in settings by associating a schema namespace with a class-based view (or view function).

Unlike `DataFile` contextual views which inject content into the DOM via an AJAX call, these contextual views override the entire page.

Example:

```
DATASET_VIEWS = [
    ('http://example.org/schemas/dataset/my_awesome_schema',
     'tardis.apps.my_awesome_app.views.CustomDatasetViewSubclass'),
]

EXPERIMENT_VIEWS = [
```

(continues on next page)

(continued from previous page)

```
    ('http://example.org/schemas/expt/my_awesome_schema',
     'tardis.apps.my_awesome_app.views.CustomExptViewSubclass'),
]
```

Custom Index View

Rationale

Specific sites or facilities often want to display a custom index page that presents recently ingested experiments in a way which is more meaningful for their particular domain or application. MyTardis support overriding the index page (/) on a per-domain or per-`Site` basis.

User Guide

Example:

```
INDEX_VIEWS = {
    1: 'tardis.apps.my_custom_app.views.MyCustomIndexSubclass',
    'facility.example.org': 'tardis.apps.myapp.AnotherCustomIndexSubclass'
}
```

A custom view override is defined in settings as dictionary mapping a class-based view (or view function) to a Django Site. A `Site` is specified by SITE_ID (an integer) or the domain name of the incoming request.

Developers creating custom contextual index views are encouraged to subclass `tardis.tardis_portal.views.pages.IndexView`.

Good practice for app developers

In order to benefit from future bug and security fixes in core MyTardis, app developers are strongly encouraged to override `IndexView`, `DatasetView` and `ExperimentView` (from `tardis.tardis_portal.pages`) when creating custom contextual views.

The default and well-tested `index.html`, `view_dataset.html` and `view_experiment.html` templates can used as a basis for these custom contextual views.

New versions may change the default templates and view functions. If you copy and paste parts for your application, please check with each upgrade that you are still using up to date code.

11.4.2 OAI-PMH Producer

Using to provide RIF-CS

Minimal providers for Dublin Core and RIF-CS are included in the app.

To enable the app, include *`tardis.apps.oaipmh`* in `settings.INSTALLED_APPS`.

Your OAI-PMH query endpoint will be on: http://mytardis-example.com/apps/oaipmh/

Implementing your own providers

To allow multiple metadata formats (and types within them) the *tardis.apps.oaipmh.server.ProxyingServer* handles all requests and proxies them to the providers specified in `settings.OAIPMH_PROVIDERS`.

You should extend *tardis.apps.oaipmh.provider.base.BaseProvider* or one of the existing providers if you wish to extend the functionality in a site-specific way.

class `tardis.apps.oaipmh.provider.base.`**BaseProvider**(*site*)

A base provider which roughly implements the PyOAI interface for OAI-PMH servers.

Extend this if you're writing your own provider for a new type or a different metadata format.

getRecord(*metadataPrefix*, *identifier*)

Get a record for a metadataPrefix and identifier.

Parameters

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **identifier** (*string*) –
  - repository-unique identifier of record

Raises

- **oaipmh.error.CannotDisseminateFormatError** – if `metadataPrefix` is unknown or not supported by identifier.
- **oaipmh.error.IdDoesNotExistError** – if identifier is unknown or illegal.

Returns a `header`, `metadata`, `about` tuple describing the record.

identify()

Retrieve information about the repository.

Returns an Identify object describing the repository.

listIdentifiers(*metadataPrefix*, *set=None*, *from_=None*, *until=None*)

Get a list of header information on records.

Parameters

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **set** (*string*) – set identifier; only return headers in set
- **from** (*datetime*) – only retrieve headers from **from_** date forward (in naive UTC)
- **until** (*datetime*) – only retrieve headers with dates up to and including until date (in naive UTC)

Raises

- **error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **error.NoSetHierarchyError** – if the repository does not support sets.

Returns an iterable of headers.

listMetadataFormats(*identifier=None*)

List metadata formats supported by repository or record.

Parameters **identifier** (*string*) – identify record for which we want to know all supported metadata formats. If absent, list all metadata formats supported by repository.

Raises

- **error.IdDoesNotExistError** – if record with identifier does not exist.
- **error.NoMetadataFormatsError** – if no formats are available for the indicated record.

Returns an iterable of metadataPrefix, schema, metadataNamespace tuples (each entry in the tuple is a string).

listRecords(*metadataPrefix*, *set=None*, *from_=None*, *until=None*)

Get a list of header, metadata and about information on records.

Parameters

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **set** (*string*) – set identifier; only return records in set
- **from** (*datetime*) – only retrieve records from from_ date forward (in naive UTC)
- **until** (*datetime*) – only retrieve records with dates up to and including until date (in naive UTC)

Raises

- **oaipmh.error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **oaipmh.error.NoSetHierarchyError** – if the repository does not support sets.

Returns an iterable of header, metadata, about tuples.

listSets()

Get a list of sets in the repository.

Raises **error.NoSetHierarchyError** – if the repository does not support sets.

Returns an iterable of setSpec, setName tuples (strings).

writeMetadata(*element*, *metadata*)

Create XML elements under the given element, using the provided metadata.

Should avoid doing any model-lookups, as they should be done when creating the metadata.

Parameters

- **element** (*lxml.etree.Element*) – element to put all content under (as SubElements)
- **metadata** (*oaipmh.common.Metadata*) – metadata to turn into XML

class tardis.apps.oaipmh.server.**ProxyingServer**(*providers*)

getRecord(*metadataPrefix*, *identifier*)

Get a record for a metadataPrefix and identifier.

Raises

- **oaipmh.error.CannotDisseminateFormatError** – if no provider returns a result, but at least one provider responds with oaipmh.error.CannotDisseminateFormatError (meaning the identifier exists)
- **oaipmh.error.IdDoesNotExistError** – if all providers fail with oaipmh.error.IdDoesNotExistError

Returns first successful provider response

Return type response

identify()
Retrieve information about the repository.

Returns an `oaipmh.common.Identify` object describing the repository.

Return type oaipmh.common.Identify

listIdentifiers(*metadataPrefix*, ***kwargs*)
Lists identifiers from all providers as a single set.

Raises

- **error.CannotDisseminateFormatError** – if `metadataPrefix` is not supported by the repository.
- **error.NoSetHierarchyError** – if a set is provided, as the repository does not support sets.

Returns a `set.Set` of headers.

Return type set

listMetadataFormats(***kwargs*)
List metadata formats from all providers in a single set.

Raises

- **error.IdDoesNotExistError** – if record with identifier does not exist.
- **error.NoMetadataFormatsError** – if no formats are available for the indicated record, but it does exist.

Returns a *frozenset* of `metadataPrefix`, `schema`, `metadataNamespace` tuples (each entry in the tuple is a string).

Return type frozenset

listRecords(*metadataPrefix*, ***kwargs*)
Lists records from all providers as a single set.

Raises

- **error.CannotDisseminateFormatError** – if `metadataPrefix` is not supported by the repository.
- **error.NoSetHierarchyError** – if a set is provided, as the repository does not support sets.

Returns a `set.Set` of `header`, `metadata`, `about` tuples.

Return type set

listSets()
List sets.

Raises **oaipmh.error.NoSetHierarchyError** – because set hierarchies are currrently not implemented

11.4.3 Publication Forms

Introduction

MyTardis supports a simple, configurable, publication workflow allowing researchers to curate collections of data sets for public release. This workflow represents an alternative to simply setting an experiment to "public access" that encourages researchers to provide a higher standard of metadata and bibliographic information, including authorship and acknowledgements without altering the original experiment. Additional features include embargo support, admin publication approvals, and PDB metadata extraction from the Protein Data Bank. DOI minting is also supported but depends on a third-party DOI minting service. Set up of the minting service is beyond the scope of this document.

The publication workflow is a three-stage process:

1. The user selects data sets to include in the publication, along with a title and description
2. Based on the selected data sets, domain-specific fields are presented
3. Authorship, acknowledgements and embargo information is collected

After stage one is started, the publication remains in draft until it is submitted.

Under the hood, this is achieved by creating a new experiment object that is differentiated from regular experiments by attaching a special publication schema. This new experiment links to the selected data sets, which may be from one or many original experiments. Publication related tasks (e.g. embargo processing) filter based on the publication schema and act accordingly.

Basic setup

A minimum setup requires the *tardis.apps.publication_forms* app to be added to `INSTALLED_APPS` and the following settings to be defined:

- `PUBLICATION_DATA_ADMIN` set to the user name of the person responsible for research data management
- `PUBLICATION_NOTIFICATION_SENDER_EMAIL` set to the email from which publication related notifications should be sent
- `PUBLICATION_INTRODUCTION` set to an HTML formatted introductory text that is displayed to the user before begins the publication process
- the `apps.publication_forms.update_publication_records` task should be added to `CELERYBEAT_SCHEDULE` and set to a reasonable interval for embargo processing

To disable the approval workflow and have publications automatically released, `PUBLICATIONS_REQUIRE_APPROVAL` can be set to False (defaults to True).

DOI support

These additional settings are required for DOI minting support:

- `MODC_DOI_ENABLED` set to True to enable support
- `MODC_DOI_API_ID` set to the API id assigned to this app by the minting service
- `MODC_DOI_API_PASSWORD` set to the password corresponding to the API id
- `MODC_DOI_MINT_DEFINITION` set to the WSDL file for DOI minting
- `MODC_DOI_ACTIVATE_DEFINITION` set to the WSDL file for DOI activation
- `MODC_DOI_DEACTIVATE_DEFINITION` set to the WSDL file for DOI deactivation

- `MODC_DOI_ENDPOINT` set to the API SOAP endpoint
- `MODC_DOI_MINT_URL_ROOT` set to the URL root for registered

DOIs, e.g. *http://tardis.example.com/*

Email configuration

There are a number of different notifications that are sent during the publication process. Very basic templates are pre-configured, but may be overridden by defining the `PUBLICATION_EMAIL_MESSAGES` dictionary in the *settings.py* file. Review the *default_settings.py* file for this app to see defaults. The following keys in the dictionary must be defined if

overriding the defaults:

- **`PUBLICATION_DATA_ADMIN['requires_authorisation']`**
  - Sent to publication admins to authorise/approve a publication for release
  - *{user_name}*: the submitting user's user name
  - *{pub_url}*: the direct link to the publication for review
  - *{approvals_url}*: the link to the admin approvals page
- **`PUBLICATION_DATA_ADMIN['awaiting_approval']`**
  - Sent to all authors immediately after submission
  - *{pub_title}*: the title of the publication
- **`PUBLICATION_DATA_ADMIN['approved']`**
  - Sent to all authors after the publication has been approved but before being released (e.g. by embargo expiry)
  - *{pub_title}*: the title of the publication
  - *{pub_url}*: the direct link to the publication
- **`PUBLICATION_DATA_ADMIN['approved_with_doi']`**
  - As above, but contains a DOI
  - *{doi}*: the publication's DOI
- **`PUBLICATION_DATA_ADMIN['rejected']`**
  - Sent to all authors if the publication is rejected
  - *{pub_title}*: the title of the publication
- **`PUBLICATION_DATA_ADMIN['reverted_to_draft']`**
  - Sent to all authors when the publication requires amendments by the submitting user
  - *{pub_title}*: the title of the publication
- **`PUBLICATION_DATA_ADMIN['released']`**
  - Sent to all authors when the publication has been released following approval
  - *{pub_title}*: the title of the publication
- **`PUBLICATION_DATA_ADMIN['released_with_doi']`**
  - As above, but contains a DOI

- *{doi}*: the publication's DOI

Domain-specific metadata

Domain-specific metadata is collected in stage two of the publication process. The forms to display are determined by inspecting the attached data set schemas. Schema to form mappings are defined in the `PUBLICATION_FORM_MAPPINGS` setting, an example of which is shown below:

```
PUBLICATION_FORM_MAPPINGS = [
 {'dataset_schema': r'^http://synchrotron.org.au/mx/',
  'publication_schema': PDB_PUBLICATION_SCHEMA_ROOT,
  'form_template': '/static/publication-form/mx-pdb-template.html'},
 {'dataset_schema': r'^http://synchrotron.org.au/mx/',
  'publication_schema': MX_PUBLICATION_DATASET_SCHEMA,
  'form_template':
      '/static/publication-form/mx-dataset-description-template.html'}]
```

dataset_schema is a regular expression applied to each schema attached to each data set, and if matched, the corresponding *publication_schema* is added to the resulting publication, and its parameters are provided by the form template. Care could be taken in constructing the *dataset_schema* regex keeping in mind that the expression could match one or more data sets.

Constructing the form template HTML files are somewhat challenging; examples are provided in the *static/publication-form* directory. These forms require special syntax that is defined using *AngularJs*. All forms have access to a *formTemplate* variable, which includes the publication schema name (*formTemplate.name*) a list of affected data sets (*formTemplate.datasets*). Each form will populate *formData.extraInfo[x]* with the user supplied data, where *x* is a unique key.

The forms themselves must be enclosed in a <*tardis-form*> tag, which requires a *my-model* attribute (set to *formData.extraInfo[x]*, and a *schema* (set to *formTemplate.name*). Any form field tags must also include the *tardis-form-field* attribute, in addition to *parameter-name*, which is set to the publication schema's parameter to populate. The value given to *x* as the *formData.extraInfo[x]* key must be unique over all included forms. In our example below, the form and data set indices are concatenated to form this unique key. Each entry in *formData.extraInfo* is given its own parameter set in the final publication metadata.

Here is an example that collects some information for each dataset that matches the *dataset_schema* regex:

```
<div ng-repeat="ds in formTemplate.datasets"
     ng-init="f = $parent.$index; formData.extraInfo[f+'.'+$index]['dataset'] = ds.
↪description;">
     <h4>{{ ds.description }}</h4>
     <tardis-form my-model="formData.extraInfo[f+'.'+$index]" schema="formTemplate.
↪name">
               <textarea tardis-form-field parameter-name="additional-information"
                     rows="3"
                     style="width:100%;"
                     placeholder="Information about this dataset.">
          </textarea>
     </tardis-form>
</div>
```

The above form works as follows:

1. The code inside the outer <*div*> tag is repeated for each data set, which is provided by *formTemplate.datasets*. The outer <*div*> tag also initialises a form index, *f*, obtained from the parent scope's *$index* variable, which is used in conjunction with the inner scope's *$index* variable to form the unique key for *formData.extraInfo*; namely, *formData.extraInfo[f+'.'+$index]*.

2. *ng-init* in the outer <*div*> tag also saves a 'dataset' field, which is included in *formData.extraInfo*. This functions exactly as a hidden HTML input field would.
3. The corresponding dataset name is displayed to the user in the <h4> tag
4. A <*tardis-form*> is started, using *formData.extraInfo[f+'.'+$index]* as its model, linking it to the schema provided by *formTemplate.name*.
5. A <*textarea*> collects a data set description that will be added to the parameter named 'additional-information', as defined in the corresponding schema

Please note that it is extremely important to ensure that the *parameter-name* attribute for each form field matches exactly the schema parameter. Any fields that do not match are silently ignored!

11.5 Releases

11.5.1 3.8

- Refactored settings
- Added pagination to My Data view
- BDD tests using behave and phantomjs
- Added download MD5 checksum buttons to Dataset View
- Add *autocaching* task that allows data from a StorageBox to be cached to another StorageBox
- Re-wrote user documentation and switched to hosting docs on RTD
- Switched to using NPM to manage JS deps.
- Facility and instrument are now visible on Experiment and dataset views - thanks @avrljk
- Added setting that allows datasets ordered by id on the Experiment page.
- Added setting to make sha512 checksums optional.

11.5.2 3.7 - 17 March 2016

- DataFile size is now a BigInteger field
- New settings for customisations, contextual view overrides (eg INDEX_VIEWS).
- A new AbstractTardisAppConfig class that all new tardis apps should subclass
- Third-party tardis app dependency checking
- Removed database index from Parameter.string_value to allow longer strings in Postgres. Migrations add a Postgres partial index for string_values shorter than 256 characters.
- Changed constraints on the instrument model; facility and instrument name are now unique together
- changed method tasks to task functions, pre-empting the removal of methods tasks in new celery versions
- RESTful API now supports ordering, e.g. &order_by=-title, for Experiments, Datasets and DataFiles.
- Allowed groups to be 'owners' of an Experiment. Enforce rule in views for web UI requiring every Experiment to have at least one user owner.
- Registration support updated for latest django-registration-redux package

- Speed-ups for dataset view page loading for datasets with large numbers of images. The carousel is now limited to a maximum of 100 preview images.
- Reorganised and updated documentation

11.5.3 3.6 - 16 March 2015

- removed legacy operations files (foreman, apache, uwsgi, etc)
- moved CI from Travis CI to Semaphore app
- removed buildout build system and setup.py dependency management
- build instructions in build.sh, using requirements.txt for dependencies now
- gunicorn instead of uwsgi
- updated Django to version 1.6.10
- removed migrations app
- renamed `Dataset_File` to `DataFile`
- `DataFile` have a `deleted` and a `version` flag, for upcoming support of these features.
- verifying files does not have side-effects anymore
- renamed `Author_Experiment` to `ExperimentAuthor`
- an `ExperimentAuthor` can now have an email and or a URL
- recoded `Replica` and `Location` as `DataFileObject` with associated `StorageBox`, based on the Django File API
- API v1 got some additions, largely or fully backwards-compatible
- a publication workflow app, guided publication of data
- download data via SFTP using a built-in SFTP server
- removed most traces of METS
- AAF authentication support
- parameters that can store a generic foreign key (link to any database object)
- new models `Instrument` and `Facility`
- basic support for SquashFS archives as `StorageBox`. Probably requires installation-specific code such as what is used at the Australian Synchrotron.
- error pages are no normal-sized
- new view "Facility Overview", for facility administrators to have overview over data.
- "MyData" includes owned and shared data
- safely allowing HTML in descriptions now. Achieved by "bleaching" of tags
- stats page faster through DB-server-side aggregation
- layout improvements
- pep8 and pylint improvements
- bug fixes

11.5.4 3.5 - 26 August 2013

- REST API
- REST API keys
- Authorisation now supports object-level permissions
- Front page overview
- Contextual views for Datafiles, Datasets and Experiments
- Backwards incompatible database changes
- Replica multi file location support
- Migration of replicas
- Streaming downloads
- Django 1.5
- REDIS option for celery queue
- auto-verify files
- provisional directory support
- Pylint testing on Travis CI
- Some error pages are now functional
- optionally upload comfortably with Filepicker.io
- Experiment view page load speedup
- Removed ancient XML ingest format.

11.5.5 3.0 - unreleased

- Twitter Bootstrap
- javascript templates
- backbone.js rendering of datasets
- UI for transferring datasets
- bpython shell
- celery queue

11.5.6 2.0 - Unreleased

- Auth/Auth redesign [Gerson, Uli, Russel]
  - Authorisation. Support for several pluggable authorisation plugins (Django internal, LDAP, VBL). The added AuthService middleware provides a mechanism to query all available auth modules to determine what group memberships a users has.
  - Alternative authorisation. Rule based experiment access control engine was implemented with the following access attributes for indivdual users and groups: canRead, canWrite, canDelete, isOwner. Additionally, a time stamp can be specified for each access rule.

    Further information can be found at the wiki: Authorisation Engine design

- Metadata Editing [Steve, Grischa]
- New METS parser & METS exporter [Gerson]
- Dist/Buildout infrastructure [Russell]
- Through the web creation and editing of experiments [Steve, Russell]
- Through the web upload of files [Steve]
- Download protocol handler [Russel, Uli]
- Logging framework [Uli]
- Django 1.3

11.5.7 1.07 - 01/06/2010

- Publish to tardis.edu.au interface created, though not implemented, pending legal text

11.5.8 1.06 - 15/03/2010

- Parameter import interface for creation of new parameter/schema definitions
- iPhone Interface

11.5.9 1.05 - 01/03/2010

- Images as parameters supported
- Data / metadata transfer from synchrotron is now 'threaded' using asynchronous web service transfers.

11.5.10 1.0 - 01/02/2010

- MyTardis created from existin MyTardis python / django codebase
- Allows private data to be stored
- Open key/value parameter model, replacing current crystallography one
- Internal data store for data
- LDAP Login
- Pagination of files
- Creation of synchrotron-tardis from MyTardis codebase including specific code for the VBL login service and data transfer to MyTardis deployments.
- Web server changed to apache and mod_wsgi

11.5.11 0.5 - 2009

- Re-wrote federated index (python / django)
- Federated stores are now simple web server based with optional FTP access
- Runs on Jython / Tomcat

11.5.12 0.1 - 2007

- Federated index (php) running on Apache HTTP Server
- Crystallography data deposition and packaging tools for Fedora Commons (java swing desktop)
- Search Interface via web

CHAPTER 12

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

tardis, 177
tardis.analytics, 67
tardis.analytics.apps, 66
tardis.analytics.ga, 66
tardis.analytics.tracker, 66
tardis.app_config, 176
tardis.apps, 87
tardis.apps.anzsrc_codes, 68
tardis.apps.anzsrc_codes.forms, 68
tardis.apps.anzsrc_codes.models, 68
tardis.apps.anzsrc_codes.settings, 68
tardis.apps.anzsrc_codes.tests, 68
tardis.apps.anzsrc_codes.tests.test_oaipmh, 67
tardis.apps.anzsrc_codes.tests.tests, 67
tardis.apps.anzsrc_codes.urls, 68
tardis.apps.anzsrc_codes.views, 68
tardis.apps.deep_storage_download_mapper, 69
tardis.apps.deep_storage_download_mapper.mapper, 68
tardis.apps.deep_storage_download_mapper.urls, 69
tardis.apps.dl_mapper_df_dir_only, 69
tardis.apps.dl_mapper_df_dir_only.mapper, 69
tardis.apps.dl_mapper_df_dir_only.urls, 69
tardis.apps.equipment, 71
tardis.apps.equipment.admin, 70
tardis.apps.equipment.forms, 70
tardis.apps.equipment.migrations, 70
tardis.apps.equipment.migrations.0001_initial, 69
tardis.apps.equipment.models, 70
tardis.apps.equipment.south_migrations, 70
tardis.apps.equipment.tests, 70
tardis.apps.equipment.urls, 71
tardis.apps.equipment.views, 71
tardis.apps.filepicker, 72
tardis.apps.filepicker.filepicker_settings, 71
tardis.apps.filepicker.urls, 71
tardis.apps.filepicker.utils, 71
tardis.apps.filepicker.views, 71
tardis.apps.mx_views, 72
tardis.apps.mx_views.urls, 72
tardis.apps.mx_views.views, 72
tardis.apps.oaipmh, 77
tardis.apps.oaipmh.models, 75
tardis.apps.oaipmh.provider, 74
tardis.apps.oaipmh.provider.base, 72
tardis.apps.oaipmh.server, 75
tardis.apps.oaipmh.tests, 75
tardis.apps.oaipmh.tests.provider, 75
tardis.apps.oaipmh.tests.provider.test_base, 74
tardis.apps.oaipmh.tests.test_oai, 75
tardis.apps.oaipmh.urls, 77
tardis.apps.oaipmh.views, 77
tardis.apps.publication_forms, 80
tardis.apps.publication_forms.default_settings, 77
tardis.apps.publication_forms.doi, 78
tardis.apps.publication_forms.email_text, 78
tardis.apps.publication_forms.pub_form_config, 78
tardis.apps.publication_forms.tasks, 78
tardis.apps.publication_forms.tests, 77
tardis.apps.publication_forms.tests.test_email_temp 77
tardis.apps.publication_forms.utils, 79
tardis.apps.publication_forms.views, 79
tardis.apps.push_to, 85
tardis.apps.push_to.apps, 81

tardis.apps.push_to.exceptions, 81
tardis.apps.push_to.migrations, 81
tardis.apps.push_to.migrations.0001_initial, 80
tardis.apps.push_to.migrations.0002_auto_20160518_1958, 80
tardis.apps.push_to.models, 81
tardis.apps.push_to.oauth_tokens, 84
tardis.apps.push_to.ssh_authz, 84
tardis.apps.push_to.tests, 81
tardis.apps.push_to.tests.test_models, 81
tardis.apps.push_to.utils, 84
tardis.apps.related_info, 87
tardis.apps.related_info.forms, 86
tardis.apps.related_info.models, 87
tardis.apps.related_info.settings, 87
tardis.apps.related_info.tests, 86
tardis.apps.related_info.tests.test_oaipmh, 85
tardis.apps.related_info.tests.tests, 85
tardis.apps.related_info.urls, 87
tardis.apps.related_info.views, 87
tardis.apps.slideshow_view, 87
tardis.apps.slideshow_view.urls, 87
tardis.default_settings, 90
tardis.default_settings.admins, 87
tardis.default_settings.analytics, 87
tardis.default_settings.apps, 87
tardis.default_settings.auth, 87
tardis.default_settings.caches, 87
tardis.default_settings.celery, 88
tardis.default_settings.custom_views, 88
tardis.default_settings.database, 88
tardis.default_settings.debug, 88
tardis.default_settings.downloads, 89
tardis.default_settings.email, 89
tardis.default_settings.filters, 89
tardis.default_settings.frontend, 89
tardis.default_settings.i18n, 89
tardis.default_settings.localisation, 89
tardis.default_settings.logging, 89
tardis.default_settings.middlewares, 89
tardis.default_settings.publication, 89
tardis.default_settings.search, 89
tardis.default_settings.sftp, 89
tardis.default_settings.sharing, 89
tardis.default_settings.site_customisations, 89
tardis.default_settings.staging, 90
tardis.default_settings.static_files, 90
tardis.default_settings.storage, 90
tardis.default_settings.templates, 90
tardis.default_settings.uploads, 90
tardis.default_settings.urls, 90
tardis.manage, 177
tardis.search, 92
tardis.search.forms, 90
tardis.search.search_indexes, 90
tardis.search.utils, 92
tardis.settings_changeme, 177
tardis.tardis_portal, 176
tardis.tardis_portal.admin, 146
tardis.tardis_portal.ands_doi, 151
tardis.tardis_portal.api, 151
tardis.tardis_portal.auth, 99
tardis.tardis_portal.auth.authentication, 92
tardis.tardis_portal.auth.authorisation, 93
tardis.tardis_portal.auth.authservice, 93
tardis.tardis_portal.auth.decorators, 95
tardis.tardis_portal.auth.fix_circular, 96
tardis.tardis_portal.auth.httpbasicendpoint_auth, 96
tardis.tardis_portal.auth.interfaces, 97
tardis.tardis_portal.auth.localdb_auth, 98
tardis.tardis_portal.auth.token_auth, 99
tardis.tardis_portal.auth.utils, 99
tardis.tardis_portal.constants, 159
tardis.tardis_portal.context_processors, 159
tardis.tardis_portal.creativecommonshandler, 160
tardis.tardis_portal.email, 160
tardis.tardis_portal.errors, 160
tardis.tardis_portal.fetcher, 161
tardis.tardis_portal.fields, 161
tardis.tardis_portal.filters, 101
tardis.tardis_portal.filters.diffractionimage, 99
tardis.tardis_portal.filters.jeolsem, 101
tardis.tardis_portal.forms, 161
tardis.tardis_portal.hacks, 170
tardis.tardis_portal.iiif, 170

tardis.tardis_portal.logging_middleware, 170
tardis.tardis_portal.management, 105
tardis.tardis_portal.management.commands, 105
tardis.tardis_portal.management.commands.backupdb, 101
tardis.tardis_portal.management.commands.checksums, 102
tardis.tardis_portal.management.commands.cleanexperiments, 102
tardis.tardis_portal.management.commands.cleanuptokens, 103
tardis.tardis_portal.management.commands.createsuperuser, 103
tardis.tardis_portal.management.commands.createtokenuser, 103
tardis.tardis_portal.management.commands.createuser, 103
tardis.tardis_portal.management.commands.dumpschemas, 104
tardis.tardis_portal.management.commands.loadschemas, 104
tardis.tardis_portal.management.commands.rmexperiment, 104
tardis.tardis_portal.management.commands.runfilters, 105
tardis.tardis_portal.managers, 170
tardis.tardis_portal.migrations, 108
tardis.tardis_portal.migrations.0001_initial, 105
tardis.tardis_portal.migrations.0001_squashed_0011_auto_20150505_1648, 105
tardis.tardis_portal.migrations.0002_auto_20150528_1128, 106
tardis.tardis_portal.migrations.0003_auto_20150907_1315, 106
tardis.tardis_portal.migrations.0004_storageboxoption_value_type, 106
tardis.tardis_portal.migrations.0005_datafile_directory, 106
tardis.tardis_portal.migrations.0006_datafile_sha512sum, 106
tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index, 107
tardis.tardis_portal.migrations.0008_string_value_partial_index, 107
tardis.tardis_portal.migrations.0009_auto_20160128_1119, 107
tardis.tardis_portal.migrations.0010_auto_20160503_1443, 107
tardis.tardis_portal.migrations.0011_auto_20160505_1643, 107
tardis.tardis_portal.models, 127
tardis.tardis_portal.models.access_control, 108
tardis.tardis_portal.models.datafile, 110
tardis.tardis_portal.models.dataset, 113
tardis.tardis_portal.models.experiment, 114
tardis.tardis_portal.models.facility, 116
tardis.tardis_portal.models.hooks, 116
tardis.tardis_portal.models.instrument, 116
tardis.tardis_portal.models.jti, 117
tardis.tardis_portal.models.license, 117
tardis.tardis_portal.models.parameters, 118
tardis.tardis_portal.models.storage, 124
tardis.tardis_portal.models.token, 126
tardis.tardis_portal.MultiPartForm, 145
tardis.tardis_portal.ParameterSetManager, 146
tardis.tardis_portal.publish, 128
tardis.tardis_portal.publish.provider, 127
tardis.tardis_portal.publish.provider.rifcsprovider, 127
tardis.tardis_portal.publish.publishservice, 127
tardis.tardis_portal.rfc3339, 172
tardis.tardis_portal.search_backend, 173
tardis.tardis_portal.search_query, 173
tardis.tardis_portal.shortcuts, 173
tardis.tardis_portal.signals, 174
tardis.tardis_portal.staging, 174
tardis.tardis_portal.storage, 128
tardis.tardis_portal.storage.file_system, 128
tardis.tardis_portal.storage.squashfs, 128
tardis.tardis_portal.tasks, 175
tardis.tardis_portal.templatetags, 131
tardis.tardis_portal.templatetags.basiccomparisonfilters, 129
tardis.tardis_portal.templatetags.bleach_tag, 129
tardis.tardis_portal.templatetags.capture, 129
tardis.tardis_portal.templatetags.dynurl, 129
tardis.tardis_portal.templatetags.experimentstats, 130
tardis.tardis_portal.templatetags.facility_tags, 130
tardis.tardis_portal.templatetags.feed,

130
tardis.tardis_portal.templatetags.formfieldfilters, 130
tardis.tardis_portal.templatetags.lookupfilters, 130
tardis.tardis_portal.templatetags.uploadify_tags, 130
tardis.tardis_portal.templatetags.xmldate, 131
tardis.tardis_portal.tests, 145
tardis.tardis_portal.tests.auth, 131
tardis.tardis_portal.tests.filters, 132
tardis.tardis_portal.tests.filters.test_jerdisem, 131
tardis.tardis_portal.tests.filters.test_middleware, 131
tardis.tardis_portal.tests.ldap_ldif, 132
tardis.tardis_portal.tests.mock_vbl_auth, 132
tardis.tardis_portal.tests.mock_vbl_download, 132
tardis.tardis_portal.tests.slapd, 132
tardis.tardis_portal.tests.test_ands_doi, 134
tardis.tardis_portal.tests.test_api, 134
tardis.tardis_portal.tests.test_authentication, 136
tardis.tardis_portal.tests.test_authorisation, 136
tardis.tardis_portal.tests.test_authservice, 137
tardis.tardis_portal.tests.test_collectstatic, 137
tardis.tardis_portal.tests.test_download, 138
tardis.tardis_portal.tests.test_download_apikey, 138
tardis.tardis_portal.tests.test_fetcher, 138
tardis.tardis_portal.tests.test_forms, 139
tardis.tardis_portal.tests.test_iiif, 139
tardis.tardis_portal.tests.test_ldap, 140
tardis.tardis_portal.tests.test_models, 140
tardis.tardis_portal.tests.test_parametersetmanager, 141
tardis.tardis_portal.tests.test_publishservice, 141
tardis.tardis_portal.tests.test_rmexperiment, 142
tardis.tardis_portal.tests.test_staging, 142
tardis.tardis_portal.tests.test_storage, 142
tardis.tardis_portal.tests.test_tar_download, 142
tardis.tardis_portal.tests.test_tasks, 143
tardis.tardis_portal.tests.test_tokens, 143
tardis.tardis_portal.tests.tests, 143
tardis.tardis_portal.widgets, 175
tardis.tardis_portal.xmlwriter, 175
tardis.test_on_mysql_settings, 177
tardis.test_on_postgresql_settings, 177
tardis.test_settings, 177
tardis.views, 177

Index

Symbols

–pythonpath=PYTHONPATH
command line option, 49
–settings=SETTINGS
command line option, 49
–traceback
command line option, 49
–version
command line option, 49
-h, –help
command line option, 49
-r FILE, –restore=FILE
command line option, 49
-v VERBOSITY, –verbosity=VERBOSITY
command line option, 49

A

abstract (tardis.apps.push_to.models.KeyPair.Meta attribute), 83
abstract (tardis.tardis_portal.models.parameters.Parameter.Meta attribute), 121
abstract (tardis.tardis_portal.models.parameters.ParameterSet.Meta attribute), 123
AbstractTardisAppConfig (class in tardis.app_config), 176
accessible_keys (tardis.tardis_portal.api.StorageBoxOptionResource attribute), 158
ACLAuthorization (class in tardis.tardis_portal.api), 151
ACLAuthorizationTest (class in tardis.tardis_portal.tests.test_api), 134
ACLAwareBackend (class in tardis.tardis_portal.auth.authorisation), 93
activate() (tardis.apps.publication_forms.doi.DOI method), 78
add_auth_method() (in module tardis.tardis_portal.auth.authentication), 92
add_field() (tardis.tardis_portal.MultiPartForm.MultiPartForm method), 145
add_file() (tardis.tardis_portal.MultiPartForm.MultiPartForm method), 145
add_password() (tardis.tardis_portal.auth.httpbasicendpoint_auth.HttpBasic method), 96
AddGroupPermissionsForm (class in tardis.tardis_portal.forms), 161
AddUserPermissionsForm (class in tardis.tardis_portal.forms), 161
administrator (tardis.apps.push_to.models.RemoteHost attribute), 83
AlgorithmNotSupported, 102
all() (tardis.tardis_portal.managers.ExperimentManager method), 170
allowed_groups (tardis.apps.push_to.models.OAuthSSHCertSigningService attribute), 83
ALLOWED_HOSTS, 40
ALLOWED_HOSTS (in module tardis.default_settings.debug), 88
allowed_methods (tardis.tardis_portal.api.UserResource.Meta attribute), 159
allowed_remote_hosts (tardis.apps.push_to.models.OAuthSSHCertSigning attribute), 83
allowed_users (tardis.apps.push_to.models.OAuthSSHCertSigningService attribute), 83
always_return_data (tardis.tardis_portal.api.DatasetResource.Meta attribute), 153
always_return_data (tardis.tardis_portal.api.ExperimentResource.Meta attribute), 154
always_return_data (tardis.tardis_portal.api.FacilityResource.Meta attribute), 155
always_return_data (tardis.tardis_portal.api.InstrumentResource.Meta attribute), 155
AnalyticsConfig (class in tardis.analytics.apps), 66
ANDSDOITestCase (class in tardis.tardis_portal.tests.test_ands_doi), 134
ApiKeyDownloadTestCase (class in tardis.tardis_portal.tests.test_download_apikey), 138
app_dependencies (tardis.app_config.AbstractTardisAppConfig attribute), 176
app_label (tardis.tardis_portal.auth.authorisation.ACLAwareBackend

attribute), 93
app_label (tardis.tardis_portal.models.parameters.Parameter.Meta attribute), 121
app_label (tardis.tardis_portal.models.parameters.ParameterSet.Meta attribute), 123
approval_ajax() (in module tardis.apps.publication_forms.views), 79
approval_view() (in module tardis.apps.publication_forms.views), 79
approve_publication() (in module tardis.apps.publication_forms.views), 79
args (tardis.tardis_portal.management.commands.chownexperiment.Command attribute), 102
args (tardis.tardis_portal.management.commands.dumpschema.Command attribute), 104
args (tardis.tardis_portal.management.commands.loadschema.Command attribute), 104
args (tardis.tardis_portal.management.commands.rmexperiment.Command attribute), 104
args (tardis.tardis_portal.management.commands.runfilters.Command attribute), 105
as_list() (tardis.apps.publication_forms.utils.CifHelper method), 79
ATTR_PREFIXES (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter attribute), 101
attributes (tardis.tardis_portal.models.storage.StorageBox attribute), 124
authenticate() (tardis.tardis_portal.auth.authorisation.ACLAwareBackend method), 93
authenticate() (tardis.tardis_portal.auth.authservice.AuthService method), 93
authenticate() (tardis.tardis_portal.auth.httpbasicendpoint_auth.HttpBasicEndpointAuth method), 96
authenticate() (tardis.tardis_portal.auth.interfaces.AuthProvider method), 97
authenticate() (tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend method), 98
authenticate() (tardis.tardis_portal.tests.mock_vbl_auth.MockBackend method), 132
authenticate() (tardis.tardis_portal.tests.test_authservice.MockAuthProvider method), 137
authentication (tardis.tardis_portal.api.GroupResource.Meta attribute), 155
authentication (tardis.tardis_portal.api.MyTardisModelResource.Meta attribute), 156
authentication (tardis.tardis_portal.api.ObjectACLResource.Meta attribute), 156
authentication (tardis.tardis_portal.api.UserResource.Meta attribute), 159
AuthenticationTestCase (class in tardis.tardis_portal.tests.test_authentication), 136
authorization (tardis.tardis_portal.api.GroupResource.Meta attribute), 155
authorization (tardis.tardis_portal.api.MyTardisModelResource.Meta attribute), 156
authorization (tardis.tardis_portal.api.ObjectACLResource.Meta attribute), 156
authorization (tardis.tardis_portal.api.UserResource.Meta attribute), 159
AuthProvider (class in tardis.tardis_portal.auth.interfaces), 97
AuthService (class in tardis.tardis_portal.auth.authservice), 93
AuthServiceTestCase (class in tardis.tardis_portal.tests.test_authservice), 137
autocache (tardis.tardis_portal.models.storage.StorageBox attribute), 124
AUTOGENERATE_API_KEY (in module tardis.default_settings.auth), 87

B

BackgroundTaskTestCase (class in tardis.tardis_portal.tests.test_tasks), 143
backup() (tardis.tardis_portal.management.commands.backupdb.Command method), 102
base_fields (tardis.apps.anzsrc_codes.forms.FoRCodeForm attribute), 68
base_fields (tardis.apps.equipment.forms.EquipmentSearchForm attribute), 70
base_fields (tardis.apps.push_to.models.CredentialForm attribute), 82
base_fields (tardis.apps.related_info.forms.RelatedInfoForm attribute), 86
base_fields (tardis.search.forms.GroupedSearchForm attribute), 90
base_fields (tardis.tardis_portal.admin.DatafileAdminForm attribute), 147
base_fields (tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 147
base_fields (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 150
base_fields (tardis.tardis_portal.admin.StorageBoxForm attribute), 150
base_fields (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 151
base_fields (tardis.tardis_portal.api.DatafileParameterResource attribute), 152
base_fields (tardis.tardis_portal.api.DatafileParameterSetResource attribute), 153
base_fields (tardis.tardis_portal.api.DataFileResource attribute), 152
base_fields (tardis.tardis_portal.api.DatasetParameterResource attribute), 153
base_fields (tardis.tardis_portal.api.DatasetParameterSetResource attribute), 153

base_fields (tardis.tardis_portal.api.DatasetResource attribute), 153
base_fields (tardis.tardis_portal.api.ExperimentParameterResource attribute), 154
base_fields (tardis.tardis_portal.api.ExperimentParameterSetResource attribute), 154
base_fields (tardis.tardis_portal.api.ExperimentResource attribute), 154
base_fields (tardis.tardis_portal.api.FacilityResource attribute), 155
base_fields (tardis.tardis_portal.api.GroupResource attribute), 155
base_fields (tardis.tardis_portal.api.InstrumentResource attribute), 155
base_fields (tardis.tardis_portal.api.LocationResource attribute), 156
base_fields (tardis.tardis_portal.api.MyTardisModelResource attribute), 156
base_fields (tardis.tardis_portal.api.ObjectACLResource attribute), 156
base_fields (tardis.tardis_portal.api.ParameterNameResource attribute), 157
base_fields (tardis.tardis_portal.api.ParameterResource attribute), 157
base_fields (tardis.tardis_portal.api.ParameterSetResource attribute), 157
base_fields (tardis.tardis_portal.api.ReplicaResource attribute), 157
base_fields (tardis.tardis_portal.api.SchemaResource attribute), 158
base_fields (tardis.tardis_portal.api.StorageBoxAttributeResource attribute), 158
base_fields (tardis.tardis_portal.api.StorageBoxOptionResource attribute), 158
base_fields (tardis.tardis_portal.api.StorageBoxResource attribute), 158
base_fields (tardis.tardis_portal.api.UserResource attribute), 159
base_fields (tardis.tardis_portal.forms.AddGroupPermissionsForm attribute), 161
base_fields (tardis.tardis_portal.forms.AddUserPermissionsForm attribute), 161
base_fields (tardis.tardis_portal.forms.ChangeGroupPermissionsForm attribute), 162
base_fields (tardis.tardis_portal.forms.ChangeUserPermissionsForm attribute), 164
base_fields (tardis.tardis_portal.forms.CreateGroupPermissionsForm attribute), 164
base_fields (tardis.tardis_portal.forms.CreateUserPermissionsForm attribute), 164
base_fields (tardis.tardis_portal.forms.DatafileSearchForm attribute), 164
base_fields (tardis.tardis_portal.forms.DatasetForm attribute), 165
base_fields (tardis.tardis_portal.forms.ExperimentAuthor attribute), 165
base_fields (tardis.tardis_portal.forms.ExperimentForm attribute), 166
base_fields (tardis.tardis_portal.forms.ImportParamsForm attribute), 166
base_fields (tardis.tardis_portal.forms.IRDatafileSearchForm attribute), 166
base_fields (tardis.tardis_portal.forms.LoginForm attribute), 166
base_fields (tardis.tardis_portal.forms.ManageAccountForm attribute), 167
base_fields (tardis.tardis_portal.forms.ManageGroupPermissionsForm attribute), 167
base_fields (tardis.tardis_portal.forms.MXDatafileSearchForm attribute), 167
base_fields (tardis.tardis_portal.forms.RawSearchForm attribute), 168
base_fields (tardis.tardis_portal.forms.RegisterExperimentForm attribute), 168
base_fields (tardis.tardis_portal.forms.RegistrationForm attribute), 168
base_fields (tardis.tardis_portal.forms.RightsForm attribute), 169
BaseProvider (class in tardis.apps.oaipmh.provider.base), 72, 180
BaseProviderTestCase (class in tardis.apps.oaipmh.tests.provider.test_base), 74
blank_param (tardis.tardis_portal.ParameterSetManager.ParameterSetMana attribute), 146
BLEACH_ALLOWED_ATTRIBUTES (in module tardis.default_settings.frontend), 89
BLEACH_ALLOWED_TAGS (in module tardis.default_settings.frontend), 89
bleach_value() (in module tardis.tardis_portal.templatetags.bleach_tag), 129
BROKER_URL (in module tardis.default_settings.celery), 88
build_template_html() (in module tardis.tardis_portal.email), 160
build_template_text() (in module tardis.tardis_portal.email), 160
BUNDLE (tardis.tardis_portal.models.storage.StorageBox attribute), 124
bytes_available() (in module tardis.apps.push_to.utils), 84

C

CACHE (tardis.tardis_portal.models.storage.StorageBox attribute), 124
cache_box (tardis.tardis_portal.models.storage.StorageBox attribute), 124

cache_file() (tardis.tardis_portal.models.datafile.DataFile method), 110
cache_file() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
cache_files() (tardis.tardis_portal.models.storage.StorageBox method), 125
CACHES (in module tardis.default_settings.caches), 87
can_copy() (in module tardis.apps.push_to.utils), 84
can_publish() (tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider method), 127
capture() (in module tardis.tardis_portal.templatetags.capture), 129
CaptureNode (class in tardis.tardis_portal.templatetags.capture), 129
cast_string_to_integer() (in module tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643), 105
cast_string_to_integer() (in module tardis.tardis_portal.migrations.0005_datafile_add_size_int), 106
ChangeGroupPermissionsForm (class in tardis.tardis_portal.forms), 161
ChangeGroupPermissionsForm.SelectDateWidget (class in tardis.tardis_portal.forms), 162
ChangeUserPermissionsForm (class in tardis.tardis_portal.forms), 162
ChangeUserPermissionsForm.Meta (class in tardis.tardis_portal.forms), 162
ChangeUserPermissionsForm.Meta.ObjectACL (class in tardis.tardis_portal.forms), 162
ChangeUserPermissionsForm.Meta.ObjectACL.DoesNotExist, 163
ChangeUserPermissionsForm.Meta.ObjectACL.MultipleObjectsReturned, 163
ChangeUserPermissionsForm.Meta.SelectDateWidget (class in tardis.tardis_portal.forms), 163
check_app_dependencies() (in module tardis.app_config), 176
check_if_facility_manager() (in module tardis.tardis_portal.templatetags.facility_tags), 130
check_paths() (tardis.tardis_portal.tests.slapd.Slapd class method), 133
check_pdb_status() (in module tardis.apps.publication_forms.utils), 79
child_boxes (tardis.tardis_portal.models.storage.StorageBox attribute), 125
CHOICES (tardis.tardis_portal.models.access_control.UserAuthentication attribute), 109
CifHelper (class in tardis.apps.publication_forms.utils), 79
clean() (tardis.tardis_portal.forms.RegistrationForm method), 168
clean() (tardis.tardis_portal.forms.RightsForm method), 169
clean() (tardis.tardis_portal.forms.StaticField method), 169
clean_username() (tardis.tardis_portal.forms.RegistrationForm method), 168
cleanup() (tardis.apps.filepicker.utils.FilepickerFile method), 71
clear() (tardis.tardis_portal.auth.httpbasicendpoint_auth.HttpBasicEndpointAuth method), 96
clear_publication_metadata() (in module tardis.apps.publication_forms.views), 79
close() (tardis.analytics.tracker.IteratorTracker method), 66
CollectstaticTest (class in tardis.tardis_portal.tests.test_collectstatic), 187
Command (class in tardis.tardis_portal.management.commands.backupdb), 101
Command (class in tardis.tardis_portal.management.commands.checkhashes), 102
Command (class in tardis.tardis_portal.management.commands.chownexperiment), 102
Command (class in tardis.tardis_portal.management.commands.cleanuptokens), 103
Command (class in tardis.tardis_portal.management.commands.createmysuperuser), 103
Command (class in tardis.tardis_portal.management.commands.createtoken), 103
Command (class in tardis.tardis_portal.management.commands.createuser), 103
Command (class in tardis.tardis_portal.management.commands.dumpschema), 104
Command (class in tardis.tardis_portal.management.commands.loadschema), 104
Command (class in tardis.tardis_portal.management.commands.rmexperiment), 104
Command (class in tardis.tardis_portal.management.commands.runfilters), 105
command line option
  –pythonpath=PYTHONPATH, 49
  –settings=SETTINGS, 49
  –traceback, 49
  –version, 49
  -h, –help, 49
  -r FILE, –restore=FILE, 49
  -v VERBOSITY, –verbosity=VERBOSITY, 49
CommaSeparatedInput (class in tardis.tardis_portal.widgets), 175
compliance_header() (in module tardis.tardis_portal.iiif), 170
compress() (tardis.tardis_portal.fields.MultiValueCommaSeparatedField method), 161
compute_checksums() (in module

tardis.tardis_portal.models.datafile), 112
compute_etag() (in module tardis.tardis_portal.iiif), 170
configure() (tardis.tardis_portal.tests.slapd.Slapd method), 133
configure_user() (in module tardis.tardis_portal.auth.utils), 99
CONTAINS_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
content_object (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL attribute), 163
content_object (tardis.tardis_portal.models.access_control.ObjectACL attribute), 109
content_type (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL attribute), 163
content_type (tardis.tardis_portal.models.access_control.ObjectACL attribute), 109
copy_file() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
copy_files() (tardis.tardis_portal.models.storage.StorageBox method), 125
copy_to_master() (tardis.tardis_portal.models.storage.StorageBox method), 125
create_datafile_add_form() (in module tardis.tardis_portal.forms), 169
create_detail() (tardis.tardis_portal.api.ACLAuthorization method), 151
create_draft_publication() (in module tardis.apps.publication_forms.views), 79
create_list() (tardis.tardis_portal.api.ACLAuthorization method), 151
create_local_box() (tardis.tardis_portal.models.storage.StorageBox class method), 125
create_parameterset_edit_form() (in module tardis.tardis_portal.forms), 169
create_select() (tardis.tardis_portal.forms.ChangeGroupPermissionsForm.SelectDateWidget method), 162
create_select() (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.SelectDateWidget method), 163
create_set_uri() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
create_user() (in module tardis.tardis_portal.auth.utils), 99
create_user_api_key() (in module tardis.tardis_portal.models.access_control), 110
create_user_profile() (in module tardis.tardis_portal.models.access_control), 110
created_by (tardis.tardis_portal.models.experiment.Experiment attribute), 114
CreateGroupPermissionsForm (class in tardis.tardis_portal.forms), 164
createLinkedUserAuthenticationForm() (in module tardis.tardis_portal.forms), 169
createSearchDatafileForm() (in module tardis.tardis_portal.forms), 169
createSearchDatafileSelectionForm() (in module tardis.tardis_portal.forms), 169
createSearchExperimentForm() (in module tardis.tardis_portal.forms), 169
CreateTestCase (class in tardis.apps.related_info.tests.tests), 85
CreateUserPermissionsForm (class in tardis.tardis_portal.forms), 164
CreativeCommonsHandler (class in tardis.tardis_portal.creativecommonshandler), 170
Credential (class in tardis.apps.push_to.models), 81
Credential.DoesNotExist, 81
Credential.MultipleObjectsReturned, 82
credential_set (tardis.apps.push_to.models.RemoteHost attribute), 83
CredentialAdmin (class in tardis.apps.push_to.models), 82
CredentialForm (class in tardis.apps.push_to.models), 82
CredentialForm.Meta (class in tardis.apps.push_to.models), 82

D

DATABASE_ENGINE (tardis.default_settings attribute), 40
DATABASE_HOST (tardis.default_settings attribute), 40
DATABASE_NAME (tardis.default_settings attribute), 40
DATABASE_PASSWORD (tardis.default_settings attribute), 40
DATABASE_PORT (tardis.default_settings attribute), 40
DATABASE_USER (tardis.default_settings attribute), 40
datacite_xml() (tardis.tardis_portal.ands_doi.DOIXMLProvider method), 151
DataFile (class in tardis.tardis_portal.models.datafile), 110
datafile (tardis.tardis_portal.models.datafile.DataFileObject attribute), 112
datafile (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 118
DATAFILE (tardis.tardis_portal.models.parameters.Schema attribute), 123
DataFile.DoesNotExist, 110
DataFile.MultipleObjectsReturned, 110
datafile_access_required() (in module tardis.tardis_portal.auth.decorators), 95
datafile_filename (tardis.search.search_indexes.DataFileIndex attribute), 90
datafile_set (tardis.tardis_portal.models.dataset.Dataset attribute), 113
DatafileAdmin (class in tardis.tardis_portal.admin), 147

DatafileAdminForm (class in tardis.tardis_portal.admin), 147
DatafileAdminForm.Meta (class in tardis.tardis_portal.admin), 147
DataFileIndex (class in tardis.search.search_indexes), 90
DataFileObject (class in tardis.tardis_portal.models.datafile), 111
DataFileObject.DoesNotExist, 111
DataFileObject.MultipleObjectsReturned, 112
DataFileObjectInline (class in tardis.tardis_portal.admin), 146
DataFileObjectInlineForm (class in tardis.tardis_portal.admin), 146
DataFileObjectInlineForm.Meta (class in tardis.tardis_portal.admin), 146
DatafileParameter (class in tardis.tardis_portal.models.parameters), 118
DatafileParameter.DoesNotExist, 118
DatafileParameter.MultipleObjectsReturned, 118
datafileparameter_set (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 118
datafileparameter_set (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
DatafileParameterResource (class in tardis.tardis_portal.api), 152
DatafileParameterResource.Meta (class in tardis.tardis_portal.api), 152
DatafileParameterResourceTest (class in tardis.tardis_portal.tests.test_api), 134
DatafileParameterSet (class in tardis.tardis_portal.models.parameters), 118
DatafileParameterSet.DoesNotExist, 118
DatafileParameterSet.MultipleObjectsReturned, 118
datafileparameterset_set (tardis.tardis_portal.models.datafile.DataFile attribute), 110
datafileparameterset_set (tardis.tardis_portal.models.parameters.Schema attribute), 124
DatafileParameterSetResource (class in tardis.tardis_portal.api), 152
DatafileParameterSetResource.Meta (class in tardis.tardis_portal.api), 153
DatafileParameterSetResourceTest (class in tardis.tardis_portal.tests.test_api), 134
datafileparametersets (tardis.tardis_portal.models.storage.StorageBox attribute), 125
DataFileResource (class in tardis.tardis_portal.api), 152
DataFileResource.Meta (class in tardis.tardis_portal.api), 152
DataFileResourceTest (class in tardis.tardis_portal.tests.test_api), 134
DatafileSearchForm (class in tardis.tardis_portal.forms), 164
Dataset (class in tardis.tardis_portal.models.dataset), 113
dataset (tardis.apps.equipment.models.Equipment attribute), 70
dataset (tardis.tardis_portal.models.datafile.DataFile attribute), 110
dataset (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 119
DATASET (tardis.tardis_portal.models.parameters.Schema attribute), 123
Dataset.DoesNotExist, 113
Dataset.MultipleObjectsReturned, 113
dataset_access_required() (in module tardis.tardis_portal.auth.decorators), 95
dataset_description (tardis.search.search_indexes.DatasetIndex attribute), 91
dataset_download_required() (in module tardis.tardis_portal.auth.decorators), 95
dataset_id_stored (tardis.search.search_indexes.DataFileIndex attribute), 91
dataset_id_stored (tardis.search.search_indexes.DatasetIndex attribute), 91
dataset_instrument (tardis.tardis_portal.models.instrument.Instrument attribute), 117
DATASET_VIEWS (in module tardis.default_settings.custom_views), 88
dataset_write_permissions_required() (in module tardis.tardis_portal.auth.decorators), 95
DatasetAdmin (class in tardis.tardis_portal.admin), 147
DatasetForm (class in tardis.tardis_portal.forms), 164
DatasetForm.Meta (class in tardis.tardis_portal.forms), 165
DatasetIndex (class in tardis.search.search_indexes), 91
DatasetParameter (class in tardis.tardis_portal.models.parameters), 118
DatasetParameter.DoesNotExist, 118
DatasetParameter.MultipleObjectsReturned, 118
datasetparameter_set (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 119
datasetparameter_set (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
DatasetParameterResource (class in tardis.tardis_portal.api), 153
DatasetParameterResource.Meta (class in tardis.tardis_portal.api), 153
DatasetParameterResourceTest (class in tardis.tardis_portal.tests.test_api), 134
DatasetParameterSet (class in tardis.tardis_portal.models.parameters), 119
DatasetParameterSet.DoesNotExist, 119
DatasetParameterSet.MultipleObjectsReturned, 119
datasetparameterset_set (tardis.tardis_portal.models.dataset.Dataset attribute), 113
datasetparameterset_set (tardis.tardis_portal.models.parameters.Schema attribute), 124
DatasetParameterSetResource (class in tardis.tardis_portal.api), 153

DatasetParameterSetResource.Meta (class in tardis.tardis_portal.api), 153
DatasetParameterSetResourceTest (class in tardis.tardis_portal.tests.test_api), 134
datasetparametersets (tardis.tardis_portal.models.storage.StorageBox attribute), 125
DatasetResource (class in tardis.tardis_portal.api), 153
DatasetResource.Meta (class in tardis.tardis_portal.api), 153
DatasetResourceTest (class in tardis.tardis_portal.tests.test_api), 134
datasets (tardis.tardis_portal.models.experiment.Experiment attribute), 114
DATETIME (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
day_field (tardis.tardis_portal.forms.ChangeGroupPermissionsForm.SelectDateWidget attribute), 162
day_field (tardis.tardis_portal.forms.ChangeUserPermissionsForm.SelectDateWidget attribute), 163
DBHostKeyPolicy (class in tardis.apps.push_to.models), 82
deactivate() (tardis.apps.publication_forms.doi.DOI method), 78
DEBUG (in module tardis.default_settings.debug), 88
declared_fields (tardis.apps.anzsrc_codes.forms.FoRCodeForm attribute), 68
declared_fields (tardis.apps.equipment.forms.EquipmentSearchForm attribute), 70
declared_fields (tardis.apps.push_to.models.CredentialForm attribute), 82
declared_fields (tardis.apps.related_info.forms.RelatedInfoForm attribute), 86
declared_fields (tardis.search.forms.GroupedSearchForm attribute), 90
declared_fields (tardis.tardis_portal.admin.DatafileAdminForm attribute), 147
declared_fields (tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 147
declared_fields (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 150
declared_fields (tardis.tardis_portal.admin.StorageBoxForm attribute), 150
declared_fields (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 151
declared_fields (tardis.tardis_portal.api.DatafileParameterResource attribute), 152
declared_fields (tardis.tardis_portal.api.DatafileParameterSetResource attribute), 153
declared_fields (tardis.tardis_portal.api.DataFileResource attribute), 152
declared_fields (tardis.tardis_portal.api.DatasetParameterResource attribute), 153
declared_fields (tardis.tardis_portal.api.DatasetParameterSetResource attribute), 153
declared_fields (tardis.tardis_portal.api.DatasetResource attribute), 153
declared_fields (tardis.tardis_portal.api.ExperimentParameterResource attribute), 154
declared_fields (tardis.tardis_portal.api.ExperimentParameterSetResource attribute), 154
declared_fields (tardis.tardis_portal.api.ExperimentResource attribute), 154
declared_fields (tardis.tardis_portal.api.FacilityResource attribute), 155
declared_fields (tardis.tardis_portal.api.GroupResource attribute), 155
declared_fields (tardis.tardis_portal.api.InstrumentResource attribute), 155
declared_fields (tardis.tardis_portal.api.LocationResource attribute), 156
declared_fields (tardis.tardis_portal.api.MyTardisModelResource attribute), 156
declared_fields (tardis.tardis_portal.api.ObjectACLResource attribute), 156
declared_fields (tardis.tardis_portal.api.ParameterNameResource attribute), 157
declared_fields (tardis.tardis_portal.api.ParameterResource attribute), 157
declared_fields (tardis.tardis_portal.api.ParameterSetResource attribute), 157
declared_fields (tardis.tardis_portal.api.ReplicaResource attribute), 157
declared_fields (tardis.tardis_portal.api.SchemaResource attribute), 158
declared_fields (tardis.tardis_portal.api.StorageBoxAttributeResource attribute), 158
declared_fields (tardis.tardis_portal.api.StorageBoxOptionResource attribute), 158
declared_fields (tardis.tardis_portal.api.StorageBoxResource attribute), 158
declared_fields (tardis.tardis_portal.api.UserResource attribute), 159
declared_fields (tardis.tardis_portal.forms.AddGroupPermissionsForm attribute), 161
declared_fields (tardis.tardis_portal.forms.AddUserPermissionsForm attribute), 161
declared_fields (tardis.tardis_portal.forms.ChangeGroupPermissionsForm attribute), 162
declared_fields (tardis.tardis_portal.forms.ChangeUserPermissionsForm attribute), 164
declared_fields (tardis.tardis_portal.forms.CreateGroupPermissionsForm attribute), 164
declared_fields (tardis.tardis_portal.forms.CreateUserPermissionsForm attribute), 164
declared_fields (tardis.tardis_portal.forms.DatafileSearchForm attribute), 164
declared_fields (tardis.tardis_portal.forms.DatasetForm attribute), 165

declared_fields (tardis.tardis_portal.forms.ExperimentAuthor attribute), 165
declared_fields (tardis.tardis_portal.forms.ExperimentForm attribute), 166
declared_fields (tardis.tardis_portal.forms.ImportParamsForm attribute), 166
declared_fields (tardis.tardis_portal.forms.IRDatafileSearchForm attribute), 166
declared_fields (tardis.tardis_portal.forms.LoginForm attribute), 167
declared_fields (tardis.tardis_portal.forms.ManageAccountForm attribute), 167
declared_fields (tardis.tardis_portal.forms.ManageGroupPermissionsForm attribute), 167
declared_fields (tardis.tardis_portal.forms.MXDatafileSearchForm attribute), 167
declared_fields (tardis.tardis_portal.forms.RawSearchForm attribute), 168
declared_fields (tardis.tardis_portal.forms.RegisterExperimentForm attribute), 168
declared_fields (tardis.tardis_portal.forms.RegistrationForm attribute), 168
declared_fields (tardis.tardis_portal.forms.RightsForm attribute), 169
deep_storage_mapper() (in module tardis.apps.deep_storage_download_mapper.mapper), 68
DEFAULT_ARCHIVE_FORMATS (in module tardis.default_settings.downloads), 89
DEFAULT_STORAGE_BASE_DIR (tardis.default_settings attribute), 41
dehydrate() (tardis.tardis_portal.api.ExperimentResource method), 154
dehydrate() (tardis.tardis_portal.api.ReplicaResource method), 157
dehydrate() (tardis.tardis_portal.api.UserResource method), 159
delete_all_params() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
delete_data() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
delete_detail() (tardis.tardis_portal.api.ACLAuthorization method), 151
delete_dfo() (in module tardis.tardis_portal.models.datafile), 113
delete_directory_content() (in module tardis.tardis_portal.tests.slapd), 133
delete_list() (tardis.tardis_portal.api.ACLAuthorization method), 151
delete_params() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
delete_permissions_required() (in module tardis.tardis_portal.auth.decorators), 95
DeleteTestCase (class in tardis.apps.related_info.tests.tests), 85
dependencies (tardis.apps.equipment.migrations.0001_initial.Migration attribute), 69
dependencies (tardis.apps.push_to.migrations.0001_initial.Migration attribute), 80
dependencies (tardis.apps.push_to.migrations.0002_auto_20160518_1953.M attribute), 80
dependencies (tardis.tardis_portal.migrations.0001_initial.Migration attribute), 105
dependencies (tardis.tardis_portal.migrations.0001_squashed_0011_auto_2 attribute), 105
dependencies (tardis.tardis_portal.migrations.0002_auto_20150528_1128.M attribute), 106
dependencies (tardis.tardis_portal.migrations.0003_auto_20150907_1315.M attribute), 106
dependencies (tardis.tardis_portal.migrations.0004_storageboxoption_value attribute), 106
dependencies (tardis.tardis_portal.migrations.0005_datafile_add_size_int_c attribute), 106
dependencies (tardis.tardis_portal.migrations.0006_datafile_remove_size_st attribute), 106
dependencies (tardis.tardis_portal.migrations.0007_remove_parameter_strin attribute), 107
dependencies (tardis.tardis_portal.migrations.0008_string_value_partial_inc attribute), 107
dependencies (tardis.tardis_portal.migrations.0009_auto_20160128_1119.M attribute), 107
dependencies (tardis.tardis_portal.migrations.0010_auto_20160503_1443.M attribute), 107
dependencies (tardis.tardis_portal.migrations.0011_auto_20160505_1643.M attribute), 107
deserialize() (tardis.tardis_portal.api.DataFileResource method), 152
df_dir_only() (in module tardis.apps.dl_mapper_df_dir_only.mapper), 69
DiffractionImageFilter (class in tardis.tardis_portal.filters.diffractionimage), 99
DISK (tardis.tardis_portal.models.storage.StorageBox attribute), 124
DjangoAuthBackend (class in tardis.tardis_portal.auth.localdb_auth), 98
DjangoGroupProvider (class in tardis.tardis_portal.auth.localdb_auth), 98
DjangoUserProvider (class in tardis.tardis_portal.auth.localdb_auth), 98
do_GET() (tardis.tardis_portal.tests.test_fetcher.TestWebServer.QuietSimpl method), 138
do_mysql_backup() (tardis.tardis_portal.management.commands.backupdb method), 102
do_mysql_restore() (tardis.tardis_portal.management.commands.backupdb method), 102
do_postgresql_backup() (tardis.tardis_portal.management.commands.backu method), 102

do_postgresql_restore() (tardis.tardis_portal.management.commands.load_backup method), 102
do_setup() (tardis.apps.publication_forms.pub_form_config.PubFormConfig method), 78
DOI (class in tardis.apps.publication_forms.doi), 78
DOIService (class in tardis.tardis_portal.ands_doi), 151
DOIXMLProvider (class in tardis.tardis_portal.ands_doi), 151
download_datafile() (in module tardis.tardis_portal.tests.mock_vbl_download), 132
download_file() (tardis.tardis_portal.api.DataFileResource method), 152
download_image() (in module tardis.tardis_portal.iiif), 170
download_info() (in module tardis.tardis_portal.iiif), 170
DownloadTestCase (class in tardis.tardis_portal.tests.test_download), 138
DummyStorage (class in tardis.tardis_portal.storage), 128
dynurl() (in module tardis.tardis_portal.templatetags.dynurl), 130
DynUrlNode (class in tardis.tardis_portal.templatetags.dynurl), 129

E

edit_auth_method() (in module tardis.tardis_portal.auth.authentication), 92
email_pub_approved() (in module tardis.apps.publication_forms.email_text), 78
email_pub_awaiting_approval() (in module tardis.apps.publication_forms.email_text), 78
email_pub_rejected() (in module tardis.apps.publication_forms.email_text), 78
email_pub_released() (in module tardis.apps.publication_forms.email_text), 78
email_pub_requires_authorisation() (in module tardis.apps.publication_forms.email_text), 78
email_pub_reverted_to_draft() (in module tardis.apps.publication_forms.email_text), 78
email_user() (in module tardis.tardis_portal.email), 160
EmailTemplates (class in tardis.apps.publication_forms.tests.test_email_templates), 77
endpoint() (in module tardis.apps.oaipmh.views), 77
EndpointTestCase (class in tardis.apps.oaipmh.tests.test_oai), 75
ensure_doi_exists() (tardis.tardis_portal.models.hooks.Command method), 116
Equipment (class in tardis.apps.equipment.models), 70
Equipment.DoesNotExist, 70
Equipment.MultipleObjectsReturned, 70
equipment_set (tardis.tardis_portal.models.dataset.Dataset attribute), 113
EquipmentSearchForm (class in tardis.apps.equipment.forms), 70
EquipmentTestCase (class in tardis.apps.equipment.tests), 70
error_handler() (in module tardis.views), 177
EXACT_VALUE_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
Experiment (class in tardis.tardis_portal.models.experiment), 114
experiment (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 116
experiment (tardis.tardis_portal.models.parameters.ExperimentParameterSe attribute), 120
EXPERIMENT (tardis.tardis_portal.models.parameters.Schema attribute), 123
experiment (tardis.tardis_portal.models.token.Token attribute), 126
Experiment.DoesNotExist, 114
Experiment.MultipleObjectsReturned, 114
experiment_access_required() (in module tardis.tardis_portal.auth.decorators), 95
experiment_author (tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_created_time (tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_creator (tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_description (tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_download_required() (in module tardis.tardis_portal.auth.decorators), 95
experiment_end_time (tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_file_count() (in module tardis.tardis_portal.templatetags.experimentstats), 130
experiment_id (tardis.tardis_portal.creativecommonshandler.CreativeComm attribute), 160
experiment_id_stored (tardis.search.search_indexes.DataFileIndex attribute), 91
experiment_id_stored (tardis.search.search_indexes.DatasetIndex attribute), 91
experiment_id_stored (tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_institution_name

(tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_ownership_required() (in module tardis.tardis_portal.auth.decorators), 95
experiment_set (tardis.tardis_portal.models.license.License attribute), 117
experiment_start_time (tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_title (tardis.search.search_indexes.ExperimentIndex attribute), 91
experiment_update_time (tardis.search.search_indexes.ExperimentIndex attribute), 91
EXPERIMENT_VIEWS (in module tardis.default_settings.custom_views), 88
ExperimentAdmin (class in tardis.tardis_portal.admin), 147
ExperimentAuthor (class in tardis.tardis_portal.forms), 165
ExperimentAuthor (class in tardis.tardis_portal.models.experiment), 115
ExperimentAuthor.DoesNotExist, 116
ExperimentAuthor.Meta (class in tardis.tardis_portal.forms), 165
ExperimentAuthor.MultipleObjectsReturned, 116
experimentauthor_set (tardis.tardis_portal.models.experiment.Experiment attribute), 114
ExperimentForm (class in tardis.tardis_portal.forms), 165
ExperimentForm.FullExperiment (class in tardis.tardis_portal.forms), 165
ExperimentForm.Meta (class in tardis.tardis_portal.forms), 166
ExperimentIndex (class in tardis.search.search_indexes), 91
ExperimentManager (class in tardis.tardis_portal.managers), 170
ExperimentParameter (class in tardis.tardis_portal.models.parameters), 119
ExperimentParameter.DoesNotExist, 119
ExperimentParameter.MultipleObjectsReturned, 119
experimentparameter_set (tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute), 120
experimentparameter_set (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
ExperimentParameterInline (class in tardis.tardis_portal.admin), 147
ExperimentParameterResource (class in tardis.tardis_portal.api), 154
ExperimentParameterResource.Meta (class in tardis.tardis_portal.api), 154
ExperimentParameterResourceTest (class in tardis.tardis_portal.tests.test_api), 135
ExperimentParameterSet (class in tardis.tardis_portal.models.parameters), 120
ExperimentParameterSet.DoesNotExist, 120
ExperimentParameterSet.MultipleObjectsReturned, 120
experimentparameterset_set (tardis.tardis_portal.models.experiment.Experiment attribute), 114
experimentparameterset_set (tardis.tardis_portal.models.parameters.Schema attribute), 124
ExperimentParameterSetAdmin (class in tardis.tardis_portal.admin), 148
ExperimentParameterSetResource (class in tardis.tardis_portal.api), 154
ExperimentParameterSetResource.Meta (class in tardis.tardis_portal.api), 154
ExperimentParameterSetResourceTest (class in tardis.tardis_portal.tests.test_api), 135
experimentparametersets (tardis.tardis_portal.models.storage.StorageBox attribute), 125
ExperimentResource (class in tardis.tardis_portal.api), 154
ExperimentResource.Meta (class in tardis.tardis_portal.api), 154
ExperimentResourceTest (class in tardis.tardis_portal.tests.test_api), 135
experiments (tardis.tardis_portal.models.dataset.Dataset attribute), 113
ext_groups (tardis.tardis_portal.models.access_control.UserProfile attribute), 109
external_users() (tardis.tardis_portal.managers.ExperimentManager method), 170
extra (tardis.tardis_portal.admin.DataFileObjectInline attribute), 146
extra (tardis.tardis_portal.admin.ExperimentParameterInline attribute), 147
extra (tardis.tardis_portal.admin.InstrumentParameterInline attribute), 148
extra (tardis.tardis_portal.admin.ObjectACLInline attribute), 148
extra (tardis.tardis_portal.admin.ParameterNameInline attribute), 149
extra (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 149
extra (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 150
ExtraTestCases (class in tardis.tardis_portal.tests.test_iiif), 139

F

FacetFixedSearchQuery (class in tardis.tardis_portal.search_query), 173
facilities_managed_by() (in module tardis.tardis_portal.models.facility), 116

Facility (class in tardis.tardis_portal.models.facility), 116
facility (tardis.tardis_portal.models.instrument.Instrument attribute), 117
Facility.DoesNotExist, 116
Facility.MultipleObjectsReturned, 116
FacilityResource (class in tardis.tardis_portal.api), 155
FacilityResource.Meta (class in tardis.tardis_portal.api), 155
FacilityResourceTest (class in tardis.tardis_portal.tests.test_api), 135
fetch_experiments_and_datasets() (in module tardis.apps.publication_forms.views), 79
fields (tardis.apps.push_to.models.CredentialForm.Meta attribute), 82
fields (tardis.apps.push_to.models.RemoteHostAdmin attribute), 84
fields (tardis.search.search_indexes.DataFileIndex attribute), 91
fields (tardis.search.search_indexes.DatasetIndex attribute), 91
fields (tardis.search.search_indexes.ExperimentIndex attribute), 91
fields (tardis.tardis_portal.admin.DatafileAdminForm.Meta attribute), 147
fields (tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta attribute), 146
fields (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta attribute), 149
fields (tardis.tardis_portal.admin.StorageBoxForm.Meta attribute), 150
fields (tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute), 150
fields (tardis.tardis_portal.api.UserResource.Meta attribute), 159
fields (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta attribute), 163
fields (tardis.tardis_portal.forms.DatasetForm.Meta attribute), 165
fields (tardis.tardis_portal.forms.ExperimentAuthor.Meta attribute), 165
fields (tardis.tardis_portal.forms.ExperimentForm.Meta attribute), 166
fields (tardis.tardis_portal.forms.ManageAccountForm.Meta attribute), 167
fields (tardis.tardis_portal.forms.RightsForm.Meta attribute), 169
file_object (tardis.tardis_portal.models.datafile.DataFile attribute), 110
file_object (tardis.tardis_portal.models.datafile.DataFileObject attribute), 112
file_objects (tardis.tardis_portal.models.datafile.DataFile attribute), 110
file_objects (tardis.tardis_portal.models.storage.StorageBox attribute), 125
FILENAME (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
filepicker_url_regex (tardis.apps.filepicker.utils.FilepickerFile attribute), 71
FilepickerFile (class in tardis.apps.filepicker.utils), 71
Filter1 (class in tardis.tardis_portal.tests.filters.test_middleware), 131
Filter2 (class in tardis.tardis_portal.tests.filters.test_middleware), 131
filtering (tardis.tardis_portal.api.DataFileResource.Meta attribute), 152
filtering (tardis.tardis_portal.api.DatasetResource.Meta attribute), 153
filtering (tardis.tardis_portal.api.ExperimentResource.Meta attribute), 154
filtering (tardis.tardis_portal.api.FacilityResource.Meta attribute), 155
filtering (tardis.tardis_portal.api.GroupResource.Meta attribute), 155
filtering (tardis.tardis_portal.api.InstrumentResource.Meta attribute), 155
filtering (tardis.tardis_portal.api.ObjectACLResource.Meta attribute), 156
filtering (tardis.tardis_portal.api.ParameterNameResource.Meta attribute), 157
filtering (tardis.tardis_portal.api.ReplicaResource.Meta attribute), 157
filtering (tardis.tardis_portal.api.SchemaResource.Meta attribute), 158
filtering (tardis.tardis_portal.api.UserResource.Meta attribute), 159
FilterInitMiddleware (class in tardis.tardis_portal.filters), 101
FilterInitTestCase (class in tardis.tardis_portal.tests.filters.test_middleware), 132
find_available_tcp_port() (in module tardis.tardis_portal.tests.slapd), 133
find_user_password() (tardis.tardis_portal.auth.httpbasicendpoint_auth.Http method), 96
fixtures (tardis.apps.equipment.tests.EquipmentTestCase attribute), 70
FoRCodeForm (class in tardis.apps.anzsrc_codes.forms), 68
form (tardis.apps.push_to.models.CredentialAdmin attribute), 82
form (tardis.tardis_portal.admin.DatafileAdmin attribute), 147
form (tardis.tardis_portal.admin.DataFileObjectInline attribute), 146
form (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 149
form (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 149

form (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 150
format_app_name_for_url() (in module tardis.app_config), 176
formfield_overrides (tardis.tardis_portal.admin.ExperimentParameterInline attribute), 148
FreeTextSearchField (class in tardis.tardis_portal.models.parameters), 120
FreeTextSearchField.DoesNotExist, 120
FreeTextSearchField.MultipleObjectsReturned, 120
freetextsearchfield_set (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
FreeTextSearchFieldAdmin (class in tardis.tardis_portal.admin), 148
freeze_time() (tardis.tardis_portal.tests.test_tokens.FrozenTime class method), 143
FrozenTime (class in tardis.tardis_portal.tests.test_tokens), 143

G

generate_keypair_credential() (tardis.apps.push_to.models.Credential static method), 82
get() (tardis.tardis_portal.managers.ExperimentManager method), 170
get() (tardis.tardis_portal.models.parameters.Parameter method), 121
get_absolute_filepath() (tardis.tardis_portal.models.datafile.DataFile method), 110
get_absolute_url() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_absolute_url() (tardis.tardis_portal.models.experiment.Experiment method), 114
get_absolute_url() (tardis.tardis_portal.models.token.Token method), 126
get_accessible_datafiles_for_user() (in module tardis.tardis_portal.auth.decorators), 95
get_accessible_experiments() (in module tardis.tardis_portal.auth.decorators), 95
get_accessible_experiments_for_dataset() (in module tardis.tardis_portal.auth.decorators), 95
get_aclOwnershipType_display() (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL method), 163
get_aclOwnershipType_display() (tardis.tardis_portal.models.access_control.ObjectACL method), 109
get_address() (tardis.tardis_portal.tests.slapd.Slapd method), 133
get_admin_media_path() (in module tardis.default_settings.static_files), 90
get_all_storage_boxes_used() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_all_tardis_apps() (in module tardis.test_settings), 177
get_apikey_credentials() (tardis.tardis_portal.tests.test_api.MyTardisResour method), 136
get_as_temporary_file() (tardis.tardis_portal.models.datafile.DataFile method), 110
get_available_signing_services() (tardis.apps.push_to.models.OAuthSSHCertSigningService static method), 83
get_beamline() (tardis.tardis_portal.tests.test_publishservice.MockRifCsPro method), 141
get_by_natural_key() (tardis.tardis_portal.managers.ParameterNameManag method), 172
get_by_natural_key() (tardis.tardis_portal.managers.SchemaManager method), 172
get_cif_file_object() (tardis.apps.publication_forms.utils.CifHelper method), 79
get_citations() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_client_for_host() (tardis.apps.push_to.models.Credential method), 82
get_comparison_type_display() (tardis.tardis_portal.models.parameters.ParameterName method), 122
get_content_type() (tardis.tardis_portal.MultiPartForm.MultiPartForm method), 145
get_context() (tardis.tardis_portal.publish.publishservice.PublishService method), 127
get_create_token_url() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_credential_handler() (in module tardis.tardis_portal.fetcher), 161
get_credentials() (tardis.tardis_portal.tests.test_api.MyTardisResourceTestC method), 136
get_ct() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_data_type_display() (tardis.tardis_portal.models.parameters.ParameterN method), 122
get_datafiles() (tardis.tardis_portal.api.DatasetResource method), 153
get_datafiles() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_datafiles() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_dataset_path() (in module tardis.tardis_portal.staging), 174
get_default_push_location() (in module tardis.apps.push_to.utils), 85
get_default_storage() (tardis.tardis_portal.models.storage.StorageBox class method), 125
get_default_storage_box() (tardis.tardis_portal.models.datafile.DataFile method), 111
get_dn_suffix() (tardis.tardis_portal.tests.slapd.Slapd method), 133

get_doi() (tardis.tardis_portal.ands_doi.DOIService method), 151
get_download_url() (tardis.tardis_portal.models.datafile.DataFile method), 111
get_download_urls() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_download_urls() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_draft_publication() (in module tardis.apps.publication_forms.views), 79
get_edit_url() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_edit_url() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_effective_query() (tardis.tardis_portal.forms.ChangeUserPermissionsForm class method), 163
get_effective_query() (tardis.tardis_portal.models.access_control.ObjectACL class method), 109
get_experiment_referer() (in module tardis.tardis_portal.shortcuts), 174
get_facet_counts() (tardis.tardis_portal.search_query.FacetFixedSearchQuery method), 173
get_file() (tardis.apps.filepicker.utils.FilepickerFile method), 71
get_file() (tardis.tardis_portal.models.datafile.DataFile method), 111
get_file_contents() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter method), 101
get_first_experiment() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_free_r_value() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_full_path() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
get_full_staging_path() (in module tardis.tardis_portal.staging), 174
get_git_version() (in module tardis.default_settings), 90
get_groups() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_identifier() (tardis.tardis_portal.api.MyTardisAuthentication method), 156
get_image_data() (tardis.tardis_portal.models.datafile.DataFile method), 111
get_images() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_images() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_initialised_storage_instance() (tardis.tardis_portal.models.storage.StorageBox method), 125
get_internal_schema() (tardis.tardis_portal.models.parameters.Schema class method), 124
get_item() (in module tardis.tardis_portal.templatetags.lookupfilters), 130
get_license_uri() (tardis.tardis_portal.tests.test_publishservice.MockRifCsP method), 141
get_licenses() (in module tardis.apps.publication_forms.views), 80
get_metadata() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter method), 101
get_mimetype() (tardis.tardis_portal.models.datafile.DataFile method), 111
get_model() (tardis.search.search_indexes.DataFileIndex method), 91
get_model() (tardis.search.search_indexes.DatasetIndex method), 91
get_model() (tardis.search.search_indexes.ExperimentIndex method), 91
get_next_by_created_time() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
get_next_by_created_time() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_next_by_created_time() (tardis.tardis_portal.models.jti.JTI method), 117
get_next_by_expiry_date() (tardis.tardis_portal.models.token.Token method), 126
get_next_by_update_time() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_option_license() (tardis.tardis_portal.models.license.License class method), 117
get_oauth_service() (tardis.apps.push_to.models.OAuthSSHCertSigningSer static method), 83
get_object_size() (in module tardis.apps.push_to.utils), 85
get_obs_r_value() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_options_as_dict() (tardis.tardis_portal.models.storage.StorageBox method), 125
get_or_create_cc_parameterset() (tardis.tardis_portal.creativecommonshandler.CreativeCommonsH method), 160
get_or_create_directory() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_or_create_user() (in module tardis.tardis_portal.auth.utils), 99
get_or_create_user() (tardis.tardis_portal.auth.authservice.AuthService method), 94
get_or_mint_doi() (tardis.tardis_portal.ands_doi.DOIService method), 151
get_owned_experiments() (in module

tardis.tardis_portal.auth.decorators), 95
get_owners() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_param() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
get_params() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
get_parse_status() (in module tardis.tardis_portal.storage.squashfs), 128
get_path() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_pdb_id() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_pdb_title() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_pdb_url() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_perm_bool() (tardis.tardis_portal.auth.authorisation.ACLAwareBackend method), 93
get_preferred_dfo() (tardis.tardis_portal.models.datafile.DataFile method), 111
get_previous_by_created_time() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
get_previous_by_created_time() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_previous_by_created_time() (tardis.tardis_portal.models.jti.JTI method), 117
get_previous_by_expiry_date() (tardis.tardis_portal.models.token.Token method), 126
get_previous_by_update_time() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_privileged_opener() (in module tardis.tardis_portal.fetcher), 161
get_pub_admin_email_addresses() (in module tardis.apps.publication_forms.utils), 79
get_public_access_display() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_publications_awaiting_approval() (in module tardis.apps.publication_forms.views), 80
get_queryset() (tardis.tardis_portal.managers.OracleSafeManager method), 172
get_receiving_storage_box() (tardis.tardis_portal.models.datafile.DataFile method), 111
get_related_object() (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta method), 163
get_related_object() (tardis.tardis_portal.models.access_control.ObjectACL method), 109
get_related_object_group() (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta method), 163
get_related_object_group() (tardis.tardis_portal.models.access_control.ObjectACL method), 109
get_release_date() (in module tardis.apps.publication_forms.tasks), 78
get_resolution() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_rifcs_context() (tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider method), 127
get_rifcs_context() (tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider method), 141
get_root_dn() (tardis.tardis_portal.tests.slapd.Slapd method), 133
get_root_password() (tardis.tardis_portal.tests.slapd.Slapd method), 133
get_schema() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
get_schema_type_name() (tardis.tardis_portal.models.parameters.Schema class method), 124
get_sequence_info() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_server() (in module tardis.apps.oaipmh.server), 77
get_session_expiry() (tardis.tardis_portal.models.token.Token method), 126
get_shared_experiments() (in module tardis.tardis_portal.auth.decorators), 95
get_site_admin_email() (in module tardis.apps.publication_forms.utils), 79
get_size() (tardis.tardis_portal.models.datafile.DataFile method), 111
get_size() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_size() (tardis.tardis_portal.models.experiment.Experiment method), 115
get_size_and_sha512sum() (in module tardis.tardis_portal.tests.test_download), 138
get_spacegroup() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_staging_path() (in module tardis.tardis_portal.staging), 174
get_staging_url_and_size() (in module tardis.tardis_portal.staging), 174
get_suitable_credential() (tardis.apps.push_to.models.Credential static method), 82
get_suitable_licenses() (tardis.tardis_portal.models.license.License class method), 127
get_sync_location() (in module tardis.tardis_portal.staging), 174
get_sync_root() (in module tardis.tardis_portal.staging),

174
get_sync_url_and_protocol() (in module tardis.tardis_portal.staging), 174
get_tardis_apps() (in module tardis.app_config), 176
get_template() (tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider method), 127
get_template() (tardis.tardis_portal.publish.publishservice.PublishService method), 127
get_template() (tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider method), 141
get_thumbnail_url() (tardis.tardis_portal.models.dataset.Dataset method), 113
get_tmpdir() (tardis.tardis_portal.tests.slapd.Slapd method), 133
get_token() (in module tardis.apps.push_to.oauth_tokens), 84
get_token_data() (in module tardis.apps.push_to.oauth_tokens), 84
get_type_display() (tardis.tardis_portal.models.parameters.Schema method), 124
get_unit_cell() (tardis.apps.publication_forms.utils.PDBCifHelper method), 79
get_unreleased_pdb_info() (in module tardis.apps.publication_forms.utils), 79
get_url() (tardis.tardis_portal.tests.slapd.Slapd method), 133
get_user() (tardis.tardis_portal.auth.httpbasicendpoint_auth.HttpBasicEndpointAuth method), 97
get_user() (tardis.tardis_portal.auth.interfaces.AuthProvider method), 97
get_user() (tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend method), 98
get_user() (tardis.tardis_portal.tests.mock_vbl_auth.MockBackend method), 132
get_value_type_display() (tardis.tardis_portal.models.storage.StorageBoxOption method), 126
get_view_url() (tardis.tardis_portal.models.datafile.DataFile method), 111
getAuthMethodChoices() (in module tardis.tardis_portal.forms), 169
getAuthMethodDescription() (tardis.tardis_portal.models.access_control.UserAuthentication method), 109
getDiffractionImageMetadata() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
getDiffractionPreviewImage() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
getGroupById() (tardis.tardis_portal.auth.interfaces.GroupProvider method), 97
getGroupById() (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider method), 98
getGroupById() (tardis.tardis_portal.tests.test_authservice.MockGroupProvider method), 137
getGroups() (in module tardis.tardis_portal.auth.fix_circular), 96
getGroups() (tardis.tardis_portal.auth.authservice.AuthService method), 94
getGroups() (tardis.tardis_portal.auth.interfaces.GroupProvider method), 97
getGroups() (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider method), 98
getGroups() (tardis.tardis_portal.auth.token_auth.TokenGroupProvider method), 99
getGroups() (tardis.tardis_portal.tests.test_authservice.MockGroupProvider method), 137
getGroupsForEntity() (tardis.tardis_portal.auth.authservice.AuthService method), 94
getGroupsForEntity() (tardis.tardis_portal.auth.interfaces.GroupProvider method), 97
getGroupsForEntity() (tardis.tardis_portal.tests.test_authservice.MockGroupProvider method), 137
getNamespaces() (tardis.tardis_portal.models.parameters.Schema class method), 124
getParameters() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
getParameterSets() (tardis.tardis_portal.models.datafile.DataFile method), 110
getParameterSets() (tardis.tardis_portal.models.dataset.Dataset method), 113
getParameterSets() (tardis.tardis_portal.models.experiment.Experiment method), 114
getParameterSets() (tardis.tardis_portal.models.instrument.Instrument method), 117
getPort() (tardis.tardis_portal.tests.test_fetcher.TestWebServer class method), 139
getRecord() (tardis.apps.oaipmh.provider.base.BaseProvider method), 72, 180
getRecord() (tardis.apps.oaipmh.server.ProxyingServer method), 76, 181
getSchema() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
getSubTypes() (tardis.tardis_portal.models.parameters.Schema class method), 124
getTestCase() (class in tardis.apps.related_info.tests.tests), 85
getTuples() (tardis.tardis_portal.tests.filters.test_middleware.Filter1 method), 131
getTuples() (tardis.tardis_portal.tests.filters.test_middleware.Filter2 class method), 131
getUniqueShortName() (tardis.tardis_portal.models.parameters.ParameterName method), 122
getUrl() (tardis.tardis_portal.tests.test_fetcher.TestWebServer method), 139
getUserProvider() (tardis.tardis_portal.auth.authservice.AuthService method), 94

getUserAuthentications() (tardis.tardis_portal.models.access_control.UserProfile method), 109
getUserById() (tardis.tardis_portal.auth.interfaces.UserProvider method), 97
getUserById() (tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider method), 98
getUsernameByEmail() (tardis.tardis_portal.auth.authservice.AuthService method), 94
getUsernameByEmail() (tardis.tardis_portal.auth.interfaces.AuthProvider method), 97
getUsernameByEmail() (tardis.tardis_portal.auth.interfaces.UserProvider method), 97
global_contexts() (in module tardis.tardis_portal.context_processors), 159
google_analytics() (in module tardis.tardis_portal.context_processors), 159
GREATER_THAN_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
GREATER_THAN_EQUAL_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
group (tardis.tardis_portal.models.access_control.GroupAdmin attribute), 108
group_acls_system_owned() (tardis.tardis_portal.managers.ExperimentManager method), 171
group_acls_user_owned() (tardis.tardis_portal.managers.ExperimentManager method), 171
group_ownership_required() (in module tardis.tardis_portal.auth.decorators), 95
GroupAdmin (class in tardis.tardis_portal.models.access_control), 108
GroupAdmin.DoesNotExist, 108
GroupAdmin.MultipleObjectsReturned, 108
GroupedSearchForm (class in tardis.search.forms), 90
GroupProvider (class in tardis.tardis_portal.auth.interfaces), 97
GroupResource (class in tardis.tardis_portal.api), 155
GroupResource.Meta (class in tardis.tardis_portal.api), 155
GroupResourceTest (class in tardis.tardis_portal.tests.test_api), 135
gt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 129
gte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 129

H

handle() (tardis.tardis_portal.management.commands.backupdb.Command method), 102
handle() (tardis.tardis_portal.management.commands.checkhashes.Command method), 102
handle() (tardis.tardis_portal.management.commands.chownexperiment.Command method), 102
handle() (tardis.tardis_portal.management.commands.cleanuptokens.Command method), 103
handle() (tardis.tardis_portal.management.commands.createmysuperuser.Command method), 103
handle() (tardis.tardis_portal.management.commands.createtokenuser.Command method), 103
handle() (tardis.tardis_portal.management.commands.createuser.Command method), 103
handle() (tardis.tardis_portal.management.commands.dumpschemas.Command method), 104
handle() (tardis.tardis_portal.management.commands.loadschemas.Command method), 104
handle() (tardis.tardis_portal.management.commands.rmexperiment.Command method), 104
handle() (tardis.tardis_portal.management.commands.runfilters.Command method), 105
has_cc_license() (tardis.tardis_portal.creativecommonshandler.CreativeCommonsHandler method), 160
has_datafile_access() (in module tardis.tardis_portal.auth.decorators), 95
has_datafile_download_access() (in module tardis.tardis_portal.auth.decorators), 95
has_dataset_access() (in module tardis.tardis_portal.auth.decorators), 95
has_dataset_download_access() (in module tardis.tardis_portal.auth.decorators), 95
has_dataset_ownership() (in module tardis.tardis_portal.auth.decorators), 95
has_dataset_write() (in module tardis.tardis_portal.auth.decorators), 95
has_delete_permissions() (in module tardis.tardis_portal.auth.decorators), 95
has_experiment_access() (in module tardis.tardis_portal.auth.decorators), 95
has_experiment_download_access() (in module tardis.tardis_portal.auth.decorators), 95
has_experiment_ownership() (in module tardis.tardis_portal.auth.decorators), 96
has_experiment_write() (in module tardis.tardis_portal.auth.decorators), 96
has_image() (tardis.tardis_portal.models.datafile.DataFile method), 111
has_embargo() (in module tardis.apps.publication_forms.tasks), 78
has_perm() (tardis.tardis_portal.auth.authorisation.ACLAwareBackend method), 93
has_read_or_owner_ACL() (in module tardis.tardis_portal.auth.decorators), 96
has_write_permissions() (in module tardis.tardis_portal.auth.decorators), 96

hasReader() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry method), 75
hasWriter() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry method), 75
help (tardis.tardis_portal.management.commands.backupdb.Command attribute), 102
help (tardis.tardis_portal.management.commands.checkhashindex.Command attribute), 102
help (tardis.tardis_portal.management.commands.chownexperiment.Command attribute), 102
help (tardis.tardis_portal.management.commands.cleanuptokens.Command attribute), 103
help (tardis.tardis_portal.management.commands.createmysuperuser.Command attribute), 103
help (tardis.tardis_portal.management.commands.createtokenforuser.Command attribute), 103
help (tardis.tardis_portal.management.commands.createuserincluster.Command attribute), 104
help (tardis.tardis_portal.management.commands.dumpschemas.Command attribute), 104
help (tardis.tardis_portal.management.commands.loadschemas.Command attribute), 104
help (tardis.tardis_portal.management.commands.rmexperiment.Command attribute), 104
help (tardis.tardis_portal.management.commands.runfilters.Command attribute), 105
HighlightSearchBackend (class in tardis.tardis_portal.search_backend), 173
HttpBasicEndpointAuth (class in tardis.tardis_portal.auth.httpbasicendpoint_auth), 96
HttpBasicEndpointAuth.SimplePasswordMgr (class in tardis.tardis_portal.auth.httpbasicendpoint_auth), 96
hydrate() (tardis.tardis_portal.api.DataFileResource method), 152
hydrate() (tardis.tardis_portal.api.ObjectACLResource method), 156
hydrate() (tardis.tardis_portal.api.ParameterResource method), 157
hydrate() (tardis.tardis_portal.api.ReplicaResource method), 157
hydrate_m2m() (tardis.tardis_portal.api.ExperimentResource method), 154
hydrate_schema() (tardis.tardis_portal.api.ParameterSetResource method), 157

I

id_for_label() (tardis.tardis_portal.forms.ChangeGroupPermissionsForm method), 162
id_for_label() (tardis.tardis_portal.forms.ChangeUserPermissionsForm method), 163
identify() (tardis.apps.oaipmh.provider.base.BaseProvider method), 72, 180
identify() (tardis.apps.oaipmh.server.ProxyingServer method), 76, 182
image (tardis.tardis_portal.models.dataset.Dataset attribute), 113
ImportParamsForm (class in tardis.tardis_portal.forms), 166
index() (in module tardis.apps.equipment.views), 71
index() (in module tardis.apps.publication_forms.views), 80
index_queryset() (tardis.search.search_indexes.ExperimentIndex method), 91
INDEX_VIEWS (in module tardis.default_settings.custom_views), 88
init_filters() (in module tardis.tardis_portal.tasks), 175
inlines (tardis.tardis_portal.admin.DatafileAdmin attribute), 147
inlines (tardis.tardis_portal.admin.ExperimentAdmin attribute), 147
inlines (tardis.tardis_portal.admin.ExperimentParameterSetAdmin attribute), 148
inlines (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 148
inlines (tardis.tardis_portal.admin.SchemaAdmin attribute), 149
inlines (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 149
instantiateFilters() (tardis.tardis_portal.management.commands.runfilters.Command method), 105
Instrument (class in tardis.tardis_portal.models.instrument), 116
instrument (tardis.tardis_portal.models.dataset.Dataset attribute), 113
instrument (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 121
INSTRUMENT (tardis.tardis_portal.models.parameters.Schema attribute), 123
Instrument.DoesNotExist, 117
Instrument.MultipleObjectsReturned, 117
instrument_set (tardis.tardis_portal.models.facility.Facility attribute), 116
InstrumentParameter (class in tardis.tardis_portal.models.parameters), 120
InstrumentParameter.DoesNotExist, 120
InstrumentParameter.MultipleObjectsReturned, 120
instrumentparameter_set (tardis.tardis_portal.models.parameters.Instrument attribute), 121
instrumentparameter_set (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
InstrumentParameterInline (class in tardis.tardis_portal.admin), 148
InstrumentParameterSet (class in tardis.tardis_portal.models.parameters), 121
InstrumentParameterSet.DoesNotExist, 121
InstrumentParameterSet.MultipleObjectsReturned, 121

instrumentparameterset_set (tardis.tardis_portal.models.instrument.Instrument attribute), 117
instrumentparameterset_set (tardis.tardis_portal.models.parameters.Schema attribute), 124
InstrumentParameterSetAdmin (class in tardis.tardis_portal.admin), 148
instrumentparametersets (tardis.tardis_portal.models.storage.StorageBox attribute), 125
InstrumentResource (class in tardis.tardis_portal.api), 155
InstrumentResource.Meta (class in tardis.tardis_portal.api), 155
InstrumentResourceTest (class in tardis.tardis_portal.tests.test_api), 135
interpolate_template() (in module tardis.apps.publication_forms.email_text), 78
IRDatafileSearchForm (class in tardis.tardis_portal.forms), 166
is_already_processed() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter method), 101
is_authenticated() (tardis.tardis_portal.api.MyTardisAuthentication method), 156
is_directory() (in module tardis.apps.push_to.utils), 85
is_expired() (tardis.tardis_portal.models.token.Token method), 126
is_facility_manager() (in module tardis.tardis_portal.models.facility), 116
is_group_admin() (in module tardis.tardis_portal.auth.decorators), 96
is_image() (tardis.tardis_portal.models.datafile.DataFile method), 111
is_jeol_sem_metadata() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter method), 101
is_json() (tardis.tardis_portal.models.parameters.ParameterName method), 123
is_local() (tardis.tardis_portal.models.datafile.DataFile method), 111
is_online (tardis.tardis_portal.models.datafile.DataFile attribute), 111
is_online (tardis.tardis_portal.models.dataset.Dataset attribute), 114
is_public() (tardis.tardis_portal.models.datafile.DataFile method), 111
is_publication() (tardis.tardis_portal.models.experiment.Experiment method), 115
is_publication_draft() (tardis.tardis_portal.models.experiment.Experiment method), 115
is_schema_valid() (tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider method), 127
is_schema_valid() (tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider method), 141
is_tardis_app() (in module tardis.app_config), 176
is_text_file() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter method), 101
is_valid() (tardis.tardis_portal.forms.ExperimentForm method), 166
is_valid_email() (in module tardis.tardis_portal.management.commands.createmysuperuser), 103
is_valid_email() (in module tardis.tardis_portal.management.commands.createuser), 104
isDateTime() (tardis.tardis_portal.models.parameters.ParameterName method), 122
isFilename() (tardis.tardis_portal.models.parameters.ParameterName method), 122
isLink() (tardis.tardis_portal.models.parameters.ParameterName method), 122
isLongString() (tardis.tardis_portal.models.parameters.ParameterName method), 122
isNumeric() (tardis.tardis_portal.models.parameters.ParameterName method), 122
isString() (tardis.tardis_portal.models.parameters.ParameterName method), 123
isURL() (tardis.tardis_portal.models.parameters.ParameterName method), 123
isValidPublicContact() (tardis.tardis_portal.models.access_control.UserProf method), 109
IteratorTracker (class in tardis.analytics.tracker), 66

J

JEOLSEMFilter (class in tardis.tardis_portal.filters.jeolsem), 101
JEOLSEMFilterTestCase (class in tardis.tardis_portal.tests.filters.test_jeolsem), 131
JSON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
json_indent (tardis.tardis_portal.api.PrettyJSONSerializer attribute), 157
JTI (class in tardis.tardis_portal.models.jti), 117
JTI.DoesNotExist, 117
JTI.MultipleObjectsReturned, 117

K

key (tardis.apps.push_to.models.KeyPair attribute), 83
KeyPair (class in tardis.apps.push_to.models), 83
KeyPair.Meta (class in tardis.apps.push_to.models), 83

L

Label (class in tardis.tardis_portal.widgets), 175
LDAP_BASE (tardis.default_settings attribute), 47
LDAP_GROUP_ATTR_MAP (tardis.default_settings attribute), 47
LDAP_GROUP_BASE (tardis.default_settings attribute), 47

LDAP_GROUP_ID_ATTR (tardis.default_settings attribute), 47
LDAP_TLS (tardis.default_settings attribute), 47
LDAP_URL (tardis.default_settings attribute), 47
LDAP_USER_ATTR_MAP (tardis.default_settings attribute), 47
LDAP_USER_BASE (tardis.default_settings attribute), 47
LDAP_USER_LOGIN_ATTR (tardis.default_settings attribute), 47
ldapadd() (tardis.tardis_portal.tests.slapd.Slapd method), 133
LDAPErrorTest (class in tardis.tardis_portal.tests.test_ldap), 140
ldapsearch() (tardis.tardis_portal.tests.slapd.Slapd method), 133
LDAPTest (class in tardis.tardis_portal.tests.test_ldap), 140
length_gt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 129
length_gte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 129
length_lt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 129
length_lte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 129
LESS_THAN_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
LESS_THAN_EQUAL_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
Level0TestCase (class in tardis.tardis_portal.tests.test_iiif), 139
Level1TestCase (class in tardis.tardis_portal.tests.test_iiif), 139
Level2TestCase (class in tardis.tardis_portal.tests.test_iiif), 140
License (class in tardis.tardis_portal.models.license), 117
license (tardis.tardis_portal.models.experiment.Experiment attribute), 115
License.DoesNotExist, 117
License.MultipleObjectsReturned, 117
LINK (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
link_ct (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 118
link_ct (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 119
link_ct (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 119
link_ct (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 120
link_ct (tardis.tardis_portal.models.parameters.Parameter attribute), 121
link_gfk (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 118
link_gfk (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 119
link_gfk (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 119
link_gfk (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 120
link_gfk (tardis.tardis_portal.models.parameters.Parameter attribute), 121
link_url (tardis.tardis_portal.models.parameters.Parameter attribute), 121
list_auth_methods() (in module tardis.tardis_portal.auth.authentication), 92
list_display (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 148
list_subdirectories() (in module tardis.apps.push_to.utils), 85
listFilters() (tardis.tardis_portal.management.commands.runfilters.Comman method), 105
listIdentifiers() (tardis.apps.oaipmh.provider.base.BaseProvider method), 72, 180
listIdentifiers() (tardis.apps.oaipmh.server.ProxyingServer method), 76, 182
listMetadataFormats() (tardis.apps.oaipmh.provider.base.BaseProvider method), 73, 180
listMetadataFormats() (tardis.apps.oaipmh.server.ProxyingServer method), 76, 182
listRecords() (tardis.apps.oaipmh.provider.base.BaseProvider method), 73, 181
listRecords() (tardis.apps.oaipmh.server.ProxyingServer method), 76, 182
listSets() (tardis.apps.oaipmh.provider.base.BaseProvider method), 73, 181
listSets() (tardis.apps.oaipmh.server.ProxyingServer method), 77, 182
ListTestCase (class in tardis.apps.related_info.tests.tests), 86
LocationResource (class in tardis.tardis_portal.api), 155
LocationResource.Meta (class in tardis.tardis_portal.api), 155
LocationResourceTest (class in tardis.tardis_portal.tests.test_api), 135
log_message() (tardis.tardis_portal.tests.test_fetcher.TestWebServer.QuietSi method), 138
logger (in module tardis.tardis_portal.xmlwriter), 176
LoggingMiddleware (class in tardis.tardis_portal.logging_middleware), 170

LoginForm (class in tardis.tardis_portal.forms), 166
LONGSTRING (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
lookup_by_unique_id_only() (in module tardis.tardis_portal.api), 159
lookup_kwargs_with_identifiers() (tardis.tardis_portal.api.MyTardisModelResource method), 156
lookup_kwargs_with_identifiers() (tardis.tardis_portal.api.SchemaResource method), 158
lt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 129
lte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 129

M

make_filter() (in module tardis.tardis_portal.filters.diffractionimage), 100
manage_account_processor() (in module tardis.tardis_portal.context_processors), 160
manage_rifcs() (tardis.tardis_portal.publish.publishservice.PublishService method), 127
ManageAccountForm (class in tardis.tardis_portal.forms), 167
ManageAccountForm.Meta (class in tardis.tardis_portal.forms), 167
ManageGroupPermissionsForm (class in tardis.tardis_portal.forms), 167
manager_group (tardis.tardis_portal.models.facility.Facility attribute), 116
map_form_to_schemas() (in module tardis.apps.publication_forms.views), 80
master_box (tardis.tardis_portal.models.storage.StorageBox attribute), 125
media (tardis.apps.anzsrc_codes.forms.FoRCodeForm attribute), 68
media (tardis.apps.equipment.forms.EquipmentSearchForm attribute), 70
media (tardis.apps.push_to.models.CredentialAdmin attribute), 82
media (tardis.apps.push_to.models.CredentialForm attribute), 82
media (tardis.apps.push_to.models.RemoteHostAdmin attribute), 84
media (tardis.apps.related_info.forms.RelatedInfoForm attribute), 86
media (tardis.search.forms.GroupedSearchForm attribute), 90
media (tardis.tardis_portal.admin.DatafileAdmin attribute), 147
media (tardis.tardis_portal.admin.DatafileAdminForm attribute), 147
media (tardis.tardis_portal.admin.DataFileObjectInline attribute), 146
media (tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 147
media (tardis.tardis_portal.admin.DatasetAdmin attribute), 147
media (tardis.tardis_portal.admin.ExperimentAdmin attribute), 147
media (tardis.tardis_portal.admin.ExperimentParameterInline attribute), 148
media (tardis.tardis_portal.admin.ExperimentParameterSetAdmin attribute), 148
media (tardis.tardis_portal.admin.FreeTextSearchFieldAdmin attribute), 148
media (tardis.tardis_portal.admin.InstrumentParameterInline attribute), 148
media (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 148
media (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 148
media (tardis.tardis_portal.admin.ObjectACLInline attribute), 148
media (tardis.tardis_portal.admin.ParameterNameAdmin attribute), 148
media (tardis.tardis_portal.admin.ParameterNameInline attribute), 149
media (tardis.tardis_portal.admin.SchemaAdmin attribute), 149
media (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 149
media (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 149
media (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 150
media (tardis.tardis_portal.admin.StorageBoxForm attribute), 150
media (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 150
media (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 151
media (tardis.tardis_portal.forms.AddGroupPermissionsForm attribute), 161
media (tardis.tardis_portal.forms.AddUserPermissionsForm attribute), 161
media (tardis.tardis_portal.forms.ChangeGroupPermissionsForm attribute), 162
media (tardis.tardis_portal.forms.ChangeGroupPermissionsForm.SelectDate attribute), 162
media (tardis.tardis_portal.forms.ChangeUserPermissionsForm attribute), 164
media (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.Select attribute), 163
media (tardis.tardis_portal.forms.CreateGroupPermissionsForm attribute), 164

media (tardis.tardis_portal.forms.CreateUserPermissionsForm attribute), 164
media (tardis.tardis_portal.forms.DatafileSearchForm attribute), 164
media (tardis.tardis_portal.forms.DatasetForm attribute), 165
media (tardis.tardis_portal.forms.ExperimentAuthor attribute), 165
media (tardis.tardis_portal.forms.ExperimentForm attribute), 166
media (tardis.tardis_portal.forms.ImportParamsForm attribute), 166
media (tardis.tardis_portal.forms.IRDatafileSearchForm attribute), 166
media (tardis.tardis_portal.forms.LoginForm attribute), 167
media (tardis.tardis_portal.forms.ManageAccountForm attribute), 167
media (tardis.tardis_portal.forms.ManageGroupPermissionsForm attribute), 167
media (tardis.tardis_portal.forms.MXDatafileSearchForm attribute), 167
media (tardis.tardis_portal.forms.NoInput attribute), 168
media (tardis.tardis_portal.forms.RawSearchForm attribute), 168
media (tardis.tardis_portal.forms.RegisterExperimentForm attribute), 168
media (tardis.tardis_portal.forms.RegistrationForm attribute), 168
media (tardis.tardis_portal.forms.RightsForm attribute), 169
media (tardis.tardis_portal.widgets.CommaSeparatedInput attribute), 175
media (tardis.tardis_portal.widgets.Label attribute), 175
media (tardis.tardis_portal.widgets.Span attribute), 175
merge_auth_method() (in module tardis.tardis_portal.auth.authentication), 93
METADATA_STORE_PATH (in module tardis.default_settings.storage), 90
Migration (class in tardis.apps.equipment.migrations.0001_initial), 69
Migration (class in tardis.apps.push_to.migrations.0001_initial), 80
Migration (class in tardis.apps.push_to.migrations.0002_auto_20160518_1953), 80
Migration (class in tardis.tardis_portal.migrations.0001_initial), 105
Migration (class in tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643), 105
Migration (class in tardis.tardis_portal.migrations.0002_auto_20150528_1128), 106
Migration (class in tardis.tardis_portal.migrations.0003_auto_20150907_1315), 106
Migration (class in tardis.tardis_portal.migrations.0004_storageboxoption_string_value), 106
Migration (class in tardis.tardis_portal.migrations.0005_datafile_add_size_i
106
Migration (class in tardis.tardis_portal.migrations.0006_datafile_remove_siz
106
Migration (class in tardis.tardis_portal.migrations.0007_remove_parameter_
107
Migration (class in tardis.tardis_portal.migrations.0008_string_value_partial
107
Migration (class in tardis.tardis_portal.migrations.0009_auto_20160128_11
107
Migration (class in tardis.tardis_portal.migrations.0010_auto_20160503_14
107
Migration (class in tardis.tardis_portal.migrations.0011_auto_20160505_16
107
mint() (tardis.apps.publication_forms.doi.DOI method), 78
MisconfiguredStorageError, 128
missing_host_key() (tardis.apps.push_to.models.DBHostKeyPolicy method), 82
mkdirs() (in module tardis.tardis_portal.tests.slapd), 133
MockAuthProvider (class in tardis.tardis_portal.tests.test_authservice), 137
MockBackend (class in tardis.tardis_portal.tests.mock_vbl_auth), 132
MockGroupProvider (class in tardis.tardis_portal.tests.test_authservice), 137
MockRequest (class in tardis.tardis_portal.tests.test_authservice), 137
MockRifCsProvider (class in tardis.tardis_portal.tests.test_publishservice), 141
MockSettings (class in tardis.tardis_portal.tests.test_authservice), 137
model (tardis.apps.push_to.models.CredentialForm.Meta attribute), 82
model (tardis.tardis_portal.admin.DatafileAdminForm.Meta attribute), 147
model (tardis.tardis_portal.admin.DataFileObjectInline attribute), 146
model (tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta attribute), 146
model (tardis.tardis_portal.admin.ExperimentParameterInline attribute), 148
model (tardis.tardis_portal.admin.InstrumentParameterInline attribute), 148
model (tardis.tardis_portal.admin.ObjectACLInline attribute), 148
model (tardis.tardis_portal.admin.ParameterNameInline

attribute), 149
model (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 149
model (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta attribute), 149
model (tardis.tardis_portal.admin.StorageBoxForm.Meta attribute), 150
model (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 150
model (tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute), 151
model (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta attribute), 164
model (tardis.tardis_portal.forms.DatasetForm.Meta attribute), 165
model (tardis.tardis_portal.forms.ExperimentAuthor.Meta attribute), 165
model (tardis.tardis_portal.forms.ExperimentForm.Meta attribute), 166
model (tardis.tardis_portal.forms.ManageAccountForm.Meta attribute), 167
model (tardis.tardis_portal.forms.RightsForm.Meta attribute), 169
ModelsTestCase (class in tardis.apps.push_to.tests.test_models), 81
ModelTestCase (class in tardis.tardis_portal.tests.test_models), 140
ModelTestCase (class in tardis.tardis_portal.tests.test_storage), 142
modified_time (tardis.tardis_portal.models.datafile.DataFileObject attribute), 112
month_field (tardis.tardis_portal.forms.ChangeGroupPermissionsForm.SelectDateWidget attribute), 162
month_field (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.SelectDateWidget attribute), 163
move_file() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
move_files() (tardis.tardis_portal.models.storage.StorageBox method), 125
move_to_master() (tardis.tardis_portal.models.storage.StorageBox method), 125
multi_file_upload() (in module tardis.tardis_portal.templatetags.uploadify_tags), 130
MultiPartForm (class in tardis.tardis_portal.MultiPartForm), 145
MultiValueCommaSeparatedField (class in tardis.tardis_portal.fields), 161
MXDatafileSearchForm (class in tardis.tardis_portal.forms), 167
MyTardisAuthentication (class in tardis.tardis_portal.api), 156
MyTardisAuthenticationTest (class in tardis.tardis_portal.tests.test_api), 135
MyTardisLocalFileSystemStorage (class in tardis.tardis_portal.storage.file_system), 128
MyTardisModelResource (class in tardis.tardis_portal.api), 156
MyTardisModelResource.Meta (class in tardis.tardis_portal.api), 156
MyTardisResourceTestCase (class in tardis.tardis_portal.tests.test_api), 136

N

name (tardis.analytics.apps.AnalyticsConfig attribute), 66
name (tardis.apps.push_to.apps.PushToConfig attribute), 81
name (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider attribute), 98
name (tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider attribute), 98
name (tardis.tardis_portal.auth.token_auth.TokenGroupProvider attribute), 99
name (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 118
name (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 119
name (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 119
name (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 120
name (tardis.tardis_portal.models.parameters.Parameter attribute), 121
natural_key() (tardis.tardis_portal.models.parameters.ParameterName method), 123
natural_key() (tardis.tardis_portal.models.parameters.Schema method), 124
new_param() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
next() (tardis.analytics.tracker.IteratorTracker method), 67
NoHashAvailable, 102
NoInput (class in tardis.tardis_portal.forms), 167
NONE (tardis.tardis_portal.models.parameters.Schema attribute), 123
none_value (tardis.tardis_portal.forms.ChangeGroupPermissionsForm.Select attribute), 162
none_value (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.S attribute), 163
NoSuitableCredential, 81
NOT_EQUAL_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
now() (tardis.tardis_portal.tests.test_tokens.FrozenTime class method), 143
NUMERIC (tardis.tardis_portal.models.parameters.ParameterName attribute), 122

O

OAuthSSHCertSigningService (class in tardis.apps.push_to.models), 83
OAuthSSHCertSigningService.DoesNotExist, 83
OAuthSSHCertSigningService.MultipleObjectsReturned, 83
oauthsshcertsigningservice_set (tardis.apps.push_to.models.RemoteHost attribute), 83
obj_create() (tardis.tardis_portal.api.DataFileResource method), 152
obj_create() (tardis.tardis_portal.api.ExperimentResource method), 154
obj_get_list() (tardis.tardis_portal.api.ExperimentResource method), 154
object_class (tardis.tardis_portal.api.DatafileParameterResource.Meta attribute), 152
object_class (tardis.tardis_portal.api.DatafileParameterSetResource.Meta attribute), 153
object_class (tardis.tardis_portal.api.DataFileResource.Meta attribute), 152
object_class (tardis.tardis_portal.api.DatasetParameterResource.Meta attribute), 153
object_class (tardis.tardis_portal.api.DatasetParameterSetResource.Meta attribute), 153
object_class (tardis.tardis_portal.api.DatasetResource.Meta attribute), 153
object_class (tardis.tardis_portal.api.ExperimentParameterResource.Meta attribute), 154
object_class (tardis.tardis_portal.api.ExperimentParameterSetResource.Meta attribute), 154
object_class (tardis.tardis_portal.api.ExperimentResource.Meta attribute), 154
object_class (tardis.tardis_portal.api.FacilityResource.Meta attribute), 155
object_class (tardis.tardis_portal.api.GroupResource.Meta attribute), 155
object_class (tardis.tardis_portal.api.InstrumentResource.Meta attribute), 155
object_class (tardis.tardis_portal.api.LocationResource.Meta attribute), 156
object_class (tardis.tardis_portal.api.ObjectACLResource.Meta attribute), 156
object_class (tardis.tardis_portal.api.ParameterNameResource.Meta attribute), 157
object_class (tardis.tardis_portal.api.ReplicaResource.Meta attribute), 157
object_class (tardis.tardis_portal.api.SchemaResource.Meta attribute), 158
object_class (tardis.tardis_portal.api.StorageBoxAttributeResource.Meta attribute), 158
object_class (tardis.tardis_portal.api.StorageBoxOptionResource.Meta attribute), 158
object_class (tardis.tardis_portal.api.StorageBoxResource.Meta attribute), 158
object_class (tardis.tardis_portal.api.UserResource.Meta attribute), 159
ObjectACL (class in tardis.tardis_portal.models.access_control), 108
ObjectACL.DoesNotExist, 108
ObjectACL.MultipleObjectsReturned, 108
ObjectACLAdmin (class in tardis.tardis_portal.admin), 148
ObjectACLInline (class in tardis.tardis_portal.admin), 148
ObjectACLResource (class in tardis.tardis_portal.api), 156
ObjectACLResource.Meta (class in tardis.tardis_portal.api), 156
objectacls (tardis.tardis_portal.models.experiment.Experiment attribute), 115
ObjectACLTestCase (class in tardis.tardis_portal.tests.test_authorisation), 136
objects (tardis.apps.equipment.models.Equipment attribute), 70
objects (tardis.apps.push_to.models.Credential attribute), 82
objects (tardis.apps.push_to.models.OAuthSSHCertSigningService attribute), 83
objects (tardis.apps.push_to.models.RemoteHost attribute), 83
objects (tardis.search.search_indexes.DataFileIndex attribute), 91
objects (tardis.search.search_indexes.DatasetIndex attribute), 91
objects (tardis.search.search_indexes.ExperimentIndex attribute), 91
objects (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.Obje attribute), 163
objects (tardis.tardis_portal.models.access_control.GroupAdmin attribute), 108
objects (tardis.tardis_portal.models.access_control.ObjectACL attribute), 109
objects (tardis.tardis_portal.models.access_control.UserAuthentication attribute), 109
objects (tardis.tardis_portal.models.access_control.UserProfile attribute), 110
objects (tardis.tardis_portal.models.datafile.DataFile attribute), 111
objects (tardis.tardis_portal.models.datafile.DataFileObject attribute), 112
objects (tardis.tardis_portal.models.dataset.Dataset attribute), 114
objects (tardis.tardis_portal.models.experiment.Experiment attribute), 115
objects (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 116

objects (tardis.tardis_portal.models.facility.Facility attribute), 116
objects (tardis.tardis_portal.models.instrument.Instrument attribute), 117
objects (tardis.tardis_portal.models.jti.JTI attribute), 117
objects (tardis.tardis_portal.models.license.License attribute), 117
objects (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 118
objects (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 118
objects (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 119
objects (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 119
objects (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 119
objects (tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute), 120
objects (tardis.tardis_portal.models.parameters.FreeTextSearchField attribute), 120
objects (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 121
objects (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 121
objects (tardis.tardis_portal.models.parameters.Parameter attribute), 121
objects (tardis.tardis_portal.models.parameters.ParameterName attribute), 123
objects (tardis.tardis_portal.models.parameters.Schema attribute), 124
objects (tardis.tardis_portal.models.storage.StorageBox attribute), 125
objects (tardis.tardis_portal.models.storage.StorageBoxAttribute attribute), 126
objects (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 126
objects (tardis.tardis_portal.models.token.Token attribute), 126
offline_types (tardis.tardis_portal.models.storage.StorageBox attribute), 125
online_types (tardis.tardis_portal.models.storage.StorageBox attribute), 125
operations (tardis.apps.equipment.migrations.0001_initial.Migration attribute), 69
operations (tardis.apps.push_to.migrations.0001_initial.Migration attribute), 80
operations (tardis.apps.push_to.migrations.0002_auto_20160518_1953.Migration attribute), 80
operations (tardis.tardis_portal.migrations.0001_initial.Migration attribute), 105
operations (tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643.Migration attribute), 105
operations (tardis.tardis_portal.migrations.0002_auto_20150528_1128.Migration attribute), 106
operations (tardis.tardis_portal.migrations.0003_auto_20150907_1315.Migration attribute), 106
operations (tardis.tardis_portal.migrations.0004_storageboxoption_value_type attribute), 106
operations (tardis.tardis_portal.migrations.0005_datafile_add_size_int_column attribute), 106
operations (tardis.tardis_portal.migrations.0006_datafile_remove_size_string attribute), 106
operations (tardis.tardis_portal.migrations.0007_remove_parameter_string_value attribute), 107
operations (tardis.tardis_portal.migrations.0008_string_value_partial_index attribute), 107
operations (tardis.tardis_portal.migrations.0009_auto_20160128_1119.Migration attribute), 107
operations (tardis.tardis_portal.migrations.0010_auto_20160503_1443.Migration attribute), 107
operations (tardis.tardis_portal.migrations.0011_auto_20160505_1643.Migration attribute), 107
option_list (tardis.tardis_portal.management.commands.backupdb.Command attribute), 102
option_list (tardis.tardis_portal.management.commands.chownexperiment.Command attribute), 102
option_list (tardis.tardis_portal.management.commands.cleanuptokens.Command attribute), 103
option_list (tardis.tardis_portal.management.commands.createmysuperuser.Command attribute), 103
option_list (tardis.tardis_portal.management.commands.createtokenuser.Command attribute), 103
option_list (tardis.tardis_portal.management.commands.createuser.Command attribute), 104
option_list (tardis.tardis_portal.management.commands.dumpschemas.Command attribute), 104
option_list (tardis.tardis_portal.management.commands.loadschemas.Command attribute), 104
option_list (tardis.tardis_portal.management.commands.rmexperiment.Command attribute), 104
option_list (tardis.tardis_portal.management.commands.runfilters.Command attribute), 105
options (tardis.tardis_portal.models.storage.StorageBox attribute), 125
oracle_dbops_hack() (in module tardis.tardis_portal.hacks), 170
OracleSafeManager (class in tardis.tardis_portal.managers), 172
ordering (tardis.tardis_portal.api.DataFileResource.Meta attribute), 152
ordering (tardis.tardis_portal.api.DatasetResource.Meta attribute), 153
ordering (tardis.tardis_portal.api.ExperimentResource.Meta attribute), 154
ordering (tardis.tardis_portal.models.parameters.Parameter.Meta attribute), 121
ordering (tardis.tardis_portal.models.parameters.ParameterSet.Meta

attribute), 123
output_beamcenter() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
output_detectordistance() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
output_exposuretime() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
output_imagesize() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
output_metadata() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
output_oscillation() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
output_pixelsize() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
output_twotheta() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
output_wavelength() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
owned() (tardis.tardis_portal.managers.ExperimentManager method), 171
owned_and_shared() (tardis.tardis_portal.managers.ExperimentManager method), 171
owned_by_group() (tardis.tardis_portal.managers.ExperimentManager method), 171
owned_by_user() (tardis.tardis_portal.managers.ExperimentManager method), 171
owned_by_user_id() (tardis.tardis_portal.managers.ExperimentManager method), 171
OWNER_OWNED (tardis.tardis_portal.forms.ChangeUserPermissionsForm attribute), 163
OWNER_OWNED (tardis.tardis_portal.models.access_control.ObjectACL attribute), 109

P

Parameter (class in tardis.tardis_portal.models.parameters), 121
Parameter.Meta (class in tardis.tardis_portal.models.parameters), 121
parameter_class (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 118
parameter_class (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 119
parameter_class (tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute), 120
parameter_class (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 121
parameter_class (tardis.tardis_portal.models.parameters.ParameterSet attribute), 123
parameter_name (tardis.tardis_portal.models.parameters.FreeTextSearchField attribute), 120
parameter_type (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 118
parameter_type (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 119
parameter_type (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 119
parameter_type (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 121
parameter_type (tardis.tardis_portal.models.parameters.Parameter attribute), 121
ParameterDoesNotExist, 160
ParameterName (class in tardis.tardis_portal.models.parameters), 121
ParameterName.DoesNotExist, 122
ParameterName.MultipleObjectsReturned, 122
parametername_form() (in module tardis.tardis_portal.templatetags.formfieldfilters), 130
parameternames (tardis.tardis_portal.models.parameters.Schema attribute), 124
ParameterNameAdmin (class in tardis.tardis_portal.admin), 148
ParameterNameInline (class in tardis.tardis_portal.admin), 148
ParameterNameManager (class in tardis.tardis_portal.managers), 172
ParameterNameResource (class in tardis.tardis_portal.api), 156
ParameterNameResource.Meta (class in tardis.tardis_portal.api), 157
ParameterResource (class in tardis.tardis_portal.api), 157
parameters (tardis.tardis_portal.ParameterSetManager.ParameterSetManager attribute), 146
ParameterSet (class in tardis.tardis_portal.models.parameters), 123
parameterset (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 118
parameterset (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 119
parameterset (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 119
parameterset (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 121
parameterset (tardis.tardis_portal.ParameterSetManager.ParameterSetManager attribute), 146
ParameterSet.Meta (class in tardis.tardis_portal.models.parameters), 123
ParameterSetManager (class in tardis.tardis_portal.ParameterSetManager), 146
ParameterSetManagerMixin (class in tardis.tardis_portal.models.parameters), 123
ParameterSetManagerTestCase (class in tardis.tardis_portal.tests.test_parametersetmanager), 141
ParameterSetResource (class in tardis.tardis_portal.api),

157
parse_output() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
parse_term() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
parse_value() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilter method), 100
PATH_LDAPADD (tardis.tardis_portal.tests.slapd.Slapd attribute), 132
PATH_LDAPSEARCH (tardis.tardis_portal.tests.slapd.Slapd attribute), 132
PATH_SCHEMA_DIR (tardis.tardis_portal.tests.slapd.Slapd attribute), 132
PATH_SLAPD (tardis.tardis_portal.tests.slapd.Slapd attribute), 133
PATH_SLAPTEST (tardis.tardis_portal.tests.slapd.Slapd attribute), 133
PATH_TMPDIR (tardis.tardis_portal.tests.slapd.Slapd attribute), 133
pdb_helper() (in module tardis.apps.publication_forms.views), 80
PDBCifHelper (class in tardis.apps.publication_forms.utils), 79
PICKLE (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 126
populate_pdb_pub_records() (in module tardis.apps.publication_forms.tasks), 78
post_list() (tardis.tardis_portal.api.DataFileResource method), 152
post_save_experiment() (in module tardis.tardis_portal.models.hooks), 116
post_save_experiment_parameter() (in module tardis.tardis_portal.models.hooks), 116
post_save_experimentauthor() (in module tardis.tardis_portal.models.hooks), 116
POST_SAVE_FILTERS (tardis.default_settings attribute), 41
prepare_experiment_id_stored() (tardis.search.search_indexes.DataFileIndex method), 91
prepare_experiment_id_stored() (tardis.search.search_indexes.DatasetIndex method), 91
prepare_experimentauthor() (tardis.search.search_indexes.ExperimentIndex method), 91
prepare_text() (tardis.search.search_indexes.DataFileIndex method), 91
prepare_text() (tardis.search.search_indexes.DatasetIndex method), 91
prepare_text() (tardis.search.search_indexes.ExperimentIndex method), 92
prepend_urls() (tardis.tardis_portal.api.DataFileResource method), 152
prepend_urls() (tardis.tardis_portal.api.DatasetResource method), 153
PrettyJSONSerializer (class in tardis.tardis_portal.api), 157
PrivilegedOpenerTestCase (class in tardis.tardis_portal.tests.test_fetcher), 138
process_embargos() (in module tardis.apps.publication_forms.tasks), 78
process_exception() (tardis.tardis_portal.logging_middleware.LoggingMiddleware method), 170
process_form() (in module tardis.apps.publication_forms.views), 80
process_request() (tardis.tardis_portal.auth.token_auth.TokenAuthMiddleware method), 99
process_response() (tardis.tardis_portal.logging_middleware.LoggingMiddleware method), 170
ProxyingMetadataRegistry (class in tardis.apps.oaipmh.server), 75
ProxyingServer (class in tardis.apps.oaipmh.server), 75, 181
psm (tardis.tardis_portal.creativecommonshandler.CreativeCommonsHandler attribute), 160
PubFormConfig (class in tardis.apps.publication_forms.pub_form_config), 78
public() (tardis.tardis_portal.managers.ExperimentManager method), 171
PUBLIC_ACCESS_CHOICES (tardis.tardis_portal.models.experiment.Experiment attribute), 114
PUBLIC_ACCESS_EMBARGO (tardis.tardis_portal.models.experiment.Experiment attribute), 114
PUBLIC_ACCESS_FULL (tardis.tardis_portal.models.experiment.Experiment attribute), 114
public_access_implies_distribution() (tardis.tardis_portal.models.experiment.Experiment class method), 115
PUBLIC_ACCESS_METADATA (tardis.tardis_portal.models.experiment.Experiment attribute), 114
PUBLIC_ACCESS_NONE (tardis.tardis_portal.models.experiment.Experiment attribute), 114
public_download_allowed() (tardis.tardis_portal.models.experiment.Experiment method), 115
PUBLICATION_DETAILS_SCHEMA (tardis.tardis_portal.models.experiment.Experiment attribute), 114
PUBLICATION_DRAFT_SCHEMA (tardis.tardis_portal.models.experiment.Experiment attribute), 114

PUBLICATION_SCHEMA_ROOT (tardis.tardis_portal.models.experiment.Experiment attribute), 114
publish_public_expt_rifcs() (in module tardis.tardis_portal.models.hooks), 116
PublishService (class in tardis.tardis_portal.publish.publishservice), 127
PublishServiceTestCase (class in tardis.tardis_portal.tests.test_publishservice), 141
PushToConfig (class in tardis.apps.push_to.apps), 81
put_detail() (tardis.tardis_portal.api.DataFileResource method), 152

Q

query_string() (tardis.search.utils.SearchQueryString method), 92
queryset (tardis.tardis_portal.api.DatafileParameterResource.Meta attribute), 152
queryset (tardis.tardis_portal.api.DatafileParameterSetResource.Meta attribute), 153
queryset (tardis.tardis_portal.api.DataFileResource.Meta attribute), 152
queryset (tardis.tardis_portal.api.DatasetParameterResource.Meta attribute), 153
queryset (tardis.tardis_portal.api.DatasetParameterSetResource.Meta attribute), 153
queryset (tardis.tardis_portal.api.DatasetResource.Meta attribute), 153
queryset (tardis.tardis_portal.api.ExperimentParameterResource.Meta attribute), 154
queryset (tardis.tardis_portal.api.ExperimentParameterSetResource.Meta attribute), 154
queryset (tardis.tardis_portal.api.ExperimentResource.Meta attribute), 154
queryset (tardis.tardis_portal.api.FacilityResource.Meta attribute), 155
queryset (tardis.tardis_portal.api.GroupResource.Meta attribute), 155
queryset (tardis.tardis_portal.api.InstrumentResource.Meta attribute), 155
queryset (tardis.tardis_portal.api.LocationResource.Meta attribute), 156
queryset (tardis.tardis_portal.api.ObjectACLResource.Meta attribute), 156
queryset (tardis.tardis_portal.api.ParameterNameResource.Meta attribute), 157
queryset (tardis.tardis_portal.api.ReplicaResource.Meta attribute), 157
queryset (tardis.tardis_portal.api.SchemaResource.Meta attribute), 158
queryset (tardis.tardis_portal.api.StorageBoxAttributeResource.Meta attribute), 158
queryset (tardis.tardis_portal.api.StorageBoxOptionResource.Meta attribute), 158
queryset (tardis.tardis_portal.api.StorageBoxResource.Meta attribute), 158
queryset (tardis.tardis_portal.api.UserResource.Meta attribute), 159
quote() (in module tardis.tardis_portal.tests.slapd), 134

R

RANGE_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
RawSearchForm (class in tardis.tardis_portal.forms), 168
read_detail() (tardis.tardis_portal.api.ACLAuthorization method), 151
read_list() (tardis.tardis_portal.api.ACLAuthorization method), 151
readMetadata() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry method), 75
RegisterExperimentForm (class in tardis.tardis_portal.forms), 168
registerReader() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry method), 75
registerWriter() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry method), 75
REGISTRATION_OPEN (in module tardis.default_settings.auth), 87
registration_processor() (in module tardis.tardis_portal.context_processors), 160
RegistrationForm (class in tardis.tardis_portal.forms), 168
reject_publication() (in module tardis.apps.publication_forms.views), 80
RelatedInfoForm (class in tardis.apps.related_info.forms), 86
remote_hosts (tardis.apps.push_to.models.Credential attribute), 82
RemoteHost (class in tardis.apps.push_to.models), 83
RemoteHost.DoesNotExist, 83
RemoteHost.MultipleObjectsReturned, 83
RemoteHostAdmin (class in tardis.apps.push_to.models), 84
remove_auth_method() (in module tardis.tardis_portal.auth.authentication), 93
remove_draft_status() (in module tardis.apps.publication_forms.views), 80
render() (tardis.tardis_portal.forms.ChangeGroupPermissionsForm.SelectD
method), 162
render() (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.Sele
method), 163
render() (tardis.tardis_portal.forms.NoInput method), 168
render() (tardis.tardis_portal.templatetags.capture.CaptureNode method), 129

render() (tardis.tardis_portal.templatetags.dynurl.DynUrlNode method), 129
render() (tardis.tardis_portal.widgets.CommaSeparatedInput method), 175
render() (tardis.tardis_portal.widgets.Label method), 175
render_error_message() (in module tardis.tardis_portal.shortcuts), 174
render_response_index() (in module tardis.tardis_portal.shortcuts), 174
render_response_search() (in module tardis.tardis_portal.shortcuts), 174
render_to_file() (in module tardis.tardis_portal.shortcuts), 174
replaces (tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643_migration attribute), 105
ReplicaResource (class in tardis.tardis_portal.api), 157
ReplicaResource.Meta (class in tardis.tardis_portal.api), 157
ReplicaResourceTest (class in tardis.tardis_portal.tests.test_api), 136
REQUIRE_DATAFILE_CHECKSUMS (tardis.default_settings attribute), 41
REQUIRE_DATAFILE_SIZES (tardis.default_settings attribute), 41
require_publication_admin() (in module tardis.apps.publication_forms.views), 80
REQUIRE_VALIDATION_ON_INGESTION (tardis.default_settings attribute), 41
resource_name (tardis.tardis_portal.api.DataFileResource.Meta attribute), 152
restart() (tardis.tardis_portal.tests.slapd.Slapd method), 133
RestfulExperimentParameterSet (class in tardis.tardis_portal.shortcuts), 173
restore() (tardis.tardis_portal.management.commands.backupdb.Command method), 102
return_response_error() (in module tardis.tardis_portal.shortcuts), 174
return_response_error_message() (in module tardis.tardis_portal.shortcuts), 174
return_response_not_found() (in module tardis.tardis_portal.shortcuts), 174
revert_publication_to_draft() (in module tardis.apps.publication_forms.views), 80
RFC
    RFC 3339, 172, 173
rfc3339() (in module tardis.tardis_portal.rfc3339), 172
RifCsProvider (class in tardis.tardis_portal.publish.provider.rifcsprovider), 127
RifCSTestCase (class in tardis.apps.anzsrc_codes.tests.test_oaipmh), 67
RifCSTestCase (class in tardis.apps.related_info.tests.test_oaipmh), 85
RightsForm (class in tardis.tardis_portal.forms), 168
RightsForm.Meta (class in tardis.tardis_portal.forms), 169
RightsFormTestCase (class in tardis.tardis_portal.tests.test_forms), 139
RmExperimentTestCase (class in tardis.tardis_portal.tests.test_rmexperiment), 142
run_diff2jpeg() (tardis.tardis_portal.filters.diffractionimage.DiffractionImag method), 100
run_diffdump() (tardis.tardis_portal.filters.diffractionimage.DiffractionImag method), 100
run_filter() (tardis.tardis_portal.management.commands.runfilters.Comman method), 105

S

safe (tardis.tardis_portal.models.experiment.Experiment attribute), 115
sanitize_html() (in module tardis.tardis_portal.templatetags.formfieldfilters), 130
save() (tardis.apps.push_to.models.KeyPair method), 83
save() (tardis.tardis_portal.forms.ExperimentForm method), 166
save() (tardis.tardis_portal.forms.RegistrationForm method), 168
save() (tardis.tardis_portal.models.datafile.DataFile method), 111
save() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
save() (tardis.tardis_portal.models.experiment.Experiment method), 115
save() (tardis.tardis_portal.models.experiment.ExperimentAuthor method), 116
save() (tardis.tardis_portal.models.parameters.ExperimentParameter method), 120
save() (tardis.tardis_portal.models.parameters.ParameterSet method), 123
save_datafile_add_form() (in module tardis.tardis_portal.forms), 169
save_datafile_edit_form() (in module tardis.tardis_portal.forms), 169
save_license() (tardis.tardis_portal.creativecommonshandler.CreativeComm method), 160
save_m2m() (tardis.tardis_portal.forms.ExperimentForm.FullExperiment method), 166
save_metadata() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter method), 101
save_with_random_token() (tardis.tardis_portal.models.token.Token method), 126
saveDiffractionImageMetadata() (tardis.tardis_portal.filters.diffractionimage.DiffractionImageFilte

method), 100
Schema (class in tardis.tardis_portal.models.parameters), 123
schema (tardis.tardis_portal.creativecommonshandler.CreativeCommonsHandler attribute), 160
SCHEMA (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter attribute), 101
schema (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 118
schema (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 119
schema (tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute), 120
schema (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 121
schema (tardis.tardis_portal.models.parameters.ParameterName attribute), 123
schema (tardis.tardis_portal.models.parameters.ParameterSet attribute), 123
schema (tardis.tardis_portal.shortcuts.RestfulExperimentParameterSet attribute), 174
Schema.DoesNotExist, 123
Schema.MultipleObjectsReturned, 123
Schema.UnsupportedType, 123
SchemaAdmin (class in tardis.tardis_portal.admin), 149
SchemaManager (class in tardis.tardis_portal.managers), 172
SchemaResource (class in tardis.tardis_portal.api), 157
SchemaResource.Meta (class in tardis.tardis_portal.api), 158
SchemaResourceTest (class in tardis.tardis_portal.tests.test_api), 136
search() (in module tardis.apps.equipment.views), 71
search() (tardis.search.forms.GroupedSearchForm method), 90
search() (tardis.tardis_portal.forms.RawSearchForm method), 168
search() (tardis.tardis_portal.search_backend.HighlightSearchBackend method), 173
search_fields (tardis.tardis_portal.admin.DatafileAdmin attribute), 147
search_fields (tardis.tardis_portal.admin.DatasetAdmin attribute), 147
search_fields (tardis.tardis_portal.admin.ExperimentAdmin attribute), 147
search_fields (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 148
search_fields (tardis.tardis_portal.admin.ParameterNameAdmin attribute), 148
search_fields (tardis.tardis_portal.admin.SchemaAdmin attribute), 149
searchEntities() (tardis.tardis_portal.auth.authservice.AuthService method), 94
searchGroups() (tardis.tardis_portal.auth.authservice.AuthService method), 94
searchGroups() (tardis.tardis_portal.auth.interfaces.GroupProvider method), 97
searchGroups() (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider method), 98
searchGroups() (tardis.tardis_portal.auth.token_auth.TokenGroupProvider method), 99
searchGroups() (tardis.tardis_portal.tests.test_authservice.MockGroupProvider method), 137
SearchQueryString (class in tardis.search.utils), 92
SearchQueryTypeUnprovidedError, 161
SearchTestCase (class in tardis.tardis_portal.tests.tests), 143
searchUsers() (tardis.tardis_portal.auth.authservice.AuthService method), 94
searchUsers() (tardis.tardis_portal.auth.interfaces.UserProvider method), 97
SECRET_KEY, 40
select_forms() (in module tardis.apps.publication_forms.views), 80
send_mail_to_authors() (in module tardis.apps.publication_forms.utils), 79
serializer (tardis.tardis_portal.api.MyTardisModelResource.Meta attribute), 156
serializer (tardis.tardis_portal.api.UserResource.Meta attribute), 159
SerializerTest (class in tardis.tardis_portal.tests.test_api), 136
service (in module tardis.analytics.tracker), 67
set_debug() (tardis.tardis_portal.tests.slapd.Slapd method), 133
set_dn_suffix() (tardis.tardis_portal.tests.slapd.Slapd method), 133
set_embargo_release_date() (in module tardis.apps.publication_forms.views), 80
set_param() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
set_param_list() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
set_params_from_dict() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager method), 146
set_port() (tardis.tardis_portal.tests.slapd.Slapd method), 133
set_publication_authors() (in module tardis.apps.publication_forms.views), 80
set_root_cn() (tardis.tardis_portal.tests.slapd.Slapd method), 133
set_root_password() (tardis.tardis_portal.tests.slapd.Slapd method), 133
set_slapd_debug_level() (tardis.tardis_portal.tests.slapd.Slapd method), 133
set_tempdir() (tardis.tardis_portal.tests.slapd.Slapd method), 133
set_token() (in module

tardis.apps.push_to.oauth_tokens), 84
set_value() (tardis.tardis_portal.models.parameters.Parameter method), 121
setPost() (tardis.tardis_portal.tests.test_authservice.MockRequest method), 137
setUp() (tardis.apps.anzsrc_codes.tests.test_oaipmh.RifCSTestCase method), 67
setUp() (tardis.apps.anzsrc_codes.tests.tests.TabTestCase method), 67
setUp() (tardis.apps.equipment.tests.EquipmentTestCase method), 71
setUp() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase method), 74
setUp() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase method), 75
setUp() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 81
setUp() (tardis.apps.related_info.tests.test_oaipmh.RifCSTestCase method), 85
setUp() (tardis.apps.related_info.tests.tests.CreateTestCase method), 85
setUp() (tardis.apps.related_info.tests.tests.DeleteTestCase method), 85
setUp() (tardis.apps.related_info.tests.tests.GetTestCase method), 86
setUp() (tardis.apps.related_info.tests.tests.ListTestCase method), 86
setUp() (tardis.apps.related_info.tests.tests.TabTestCase method), 86
setUp() (tardis.apps.related_info.tests.tests.UpdateTestCase method), 86
setUp() (tardis.tardis_portal.tests.filters.test_jeolsem.JEOLSEMFilterTestCase method), 131
setUp() (tardis.tardis_portal.tests.filters.test_middleware.FilterInitTestCase method), 132
setUp() (tardis.tardis_portal.tests.test_ands_doi.ANDSDOITestCase method), 134
setUp() (tardis.tardis_portal.tests.test_api.DataFileResourceTest method), 134
setUp() (tardis.tardis_portal.tests.test_api.DatasetResourceTest method), 134
setUp() (tardis.tardis_portal.tests.test_api.ExperimentResourceTest method), 135
setUp() (tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase method), 136
setUp() (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase method), 136
setUp() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 136
setUp() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase method), 137
setUp() (tardis.tardis_portal.tests.test_collectstatic.CollectstaticTest method), 137
setUp() (tardis.tardis_portal.tests.test_download.DownloadTestCase method), 138
setUp() (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownloadTestCase method), 138
setUp() (tardis.tardis_portal.tests.test_forms.RightsFormTestCase method), 139
setUp() (tardis.tardis_portal.tests.test_iiif.ExtraTestCases method), 139
setUp() (tardis.tardis_portal.tests.test_iiif.Level0TestCase method), 139
setUp() (tardis.tardis_portal.tests.test_iiif.Level1TestCase method), 139
setUp() (tardis.tardis_portal.tests.test_iiif.Level2TestCase method), 140
setUp() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
setUp() (tardis.tardis_portal.tests.test_models.ModelTestCase method), 140
setUp() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase method), 141
setUp() (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase method), 141
setUp() (tardis.tardis_portal.tests.test_staging.TestStagingFiles method), 142
setUp() (tardis.tardis_portal.tests.test_storage.ModelTestCase method), 142
setUp() (tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase method), 142
setUp() (tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase method), 143
setUp() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
setUp() (tardis.tardis_portal.tests.tests.SearchTestCase method), 144
setUpClass() (tardis.tardis_portal.tests.test_fetcher.PrivilegedOpenerTestCase class method), 138
SFTP_USERNAME_ATTRIBUTE (in module tardis.default_settings.sftp), 89
shared() (tardis.tardis_portal.managers.ExperimentManager method), 171
shell_escape() (in module tardis.apps.push_to.utils), 85
sign_certificate() (in module tardis.apps.push_to.ssh_authz), 84
single_search_processor() (in module tardis.tardis_portal.context_processors), 160
SITE_TITLE (in module tardis.default_settings.site_customisations), 89
size() (in module tardis.tardis_portal.templatetags.formfieldfilters), 130
SlapdTest (class in tardis.tardis_portal.tests.slapd), 132
Span (class in tardis.tardis_portal.widgets), 175
spectral_oscillation() (tardis.tardis_portal.filters.diffractionimage.DiffractionImage method), 100
spectral_output() (tardis.tardis_portal.filters.diffractionimage.DiffractionImage

method), 100
SPONSORED_TEXT (in module tardis.default_settings.site_customisations), 89
squashfs_dirs (tardis.tardis_portal.storage.squashfs.SquashFSStorage attribute), 128
SquashFSStorage (class in tardis.tardis_portal.storage.squashfs), 128
staging_list() (in module tardis.tardis_portal.staging), 174
STAGING_PATH (tardis.default_settings attribute), 41
start() (tardis.tardis_portal.tests.slapd.Slapd method), 133
start() (tardis.tardis_portal.tests.test_fetcher.TestWebServer method), 139
started() (tardis.tardis_portal.tests.slapd.Slapd method), 133
STATIC_ROOT (tardis.default_settings attribute), 43
STATIC_URL (tardis.default_settings attribute), 43
StaticField (class in tardis.tardis_portal.forms), 169
status (tardis.tardis_portal.models.datafile.DataFile attribute), 111
stop() (tardis.tardis_portal.tests.slapd.Slapd method), 133
stop() (tardis.tardis_portal.tests.test_fetcher.TestWebServer method), 139
storage_box (tardis.tardis_portal.models.datafile.DataFileObject attribute), 112
storage_box (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 118
storage_box (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 119
storage_box (tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute), 120
storage_box (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 121
storage_box (tardis.tardis_portal.models.parameters.ParameterSet attribute), 123
storage_box (tardis.tardis_portal.models.storage.StorageBoxAttribute attribute), 126
storage_box (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 126
storage_type (tardis.tardis_portal.models.datafile.DataFileObject attribute), 112
storage_type (tardis.tardis_portal.models.storage.StorageBox attribute), 125
StorageBox (class in tardis.tardis_portal.models.storage), 124
StorageBox.DoesNotExist, 124
StorageBox.MultipleObjectsReturned, 124
StorageBoxAdmin (class in tardis.tardis_portal.admin), 149
StorageBoxAttribute (class in tardis.tardis_portal.models.storage), 125
StorageBoxAttribute.DoesNotExist, 125
StorageBoxAttribute.MultipleObjectsReturned, 126
StorageBoxAttributeInline (class in tardis.tardis_portal.admin), 149
StorageBoxAttributeInlineForm (class in tardis.tardis_portal.admin), 149
StorageBoxAttributeInlineForm.Meta (class in tardis.tardis_portal.admin), 149
StorageBoxAttributeResource (class in tardis.tardis_portal.api), 158
StorageBoxAttributeResource.Meta (class in tardis.tardis_portal.api), 158
StorageBoxForm (class in tardis.tardis_portal.admin), 150
StorageBoxForm.Meta (class in tardis.tardis_portal.admin), 150
StorageBoxOption (class in tardis.tardis_portal.models.storage), 126
StorageBoxOption.DoesNotExist, 126
StorageBoxOption.MultipleObjectsReturned, 126
StorageBoxOptionInline (class in tardis.tardis_portal.admin), 150
StorageBoxOptionInlineForm (class in tardis.tardis_portal.admin), 150
StorageBoxOptionInlineForm.Meta (class in tardis.tardis_portal.admin), 150
StorageBoxOptionResource (class in tardis.tardis_portal.api), 158
StorageBoxOptionResource.Meta (class in tardis.tardis_portal.api), 158
StorageBoxResource (class in tardis.tardis_portal.api), 158
StorageBoxResource.Meta (class in tardis.tardis_portal.api), 158
STRING (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
STRING (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 126
suite() (in module tardis.apps.equipment.tests), 71
suite() (in module tardis.tardis_portal.tests.tests), 144
sum_sizes() (tardis.tardis_portal.models.datafile.DataFile class method), 111
supports_anonymous_user (tardis.tardis_portal.auth.authorisation.ACLAwareBackend attribute), 93
supports_object_permissions (tardis.tardis_portal.auth.authorisation.ACLAwareBackend attribute), 93
synchrotron_search_epn() (in module tardis.apps.publication_forms.views), 80
SYSTEM_OWNED (tardis.tardis_portal.forms.ChangeUserPermissionsFor attribute), 163
SYSTEM_OWNED (tardis.tardis_portal.models.access_control.ObjectACL attribute), 109
system_owned_groups() (tardis.tardis_portal.managers.ExperimentManager method), 172

T

TabTestCase (class in tardis.apps.anzsrc_codes.tests.tests), 67
TabTestCase (class in tardis.apps.related_info.tests.tests), 86
tag (tardis.tardis_portal.widgets.Label attribute), 175
tag (tardis.tardis_portal.widgets.Span attribute), 175
TAPE (tardis.tardis_portal.models.storage.StorageBox attribute), 124
tardis (module), 177
tardis.analytics (module), 67
tardis.analytics.apps (module), 66
tardis.analytics.ga (module), 66
tardis.analytics.tracker (module), 66
tardis.app_config (module), 176
tardis.apps (module), 87
tardis.apps.anzsrc_codes (module), 68
tardis.apps.anzsrc_codes.forms (module), 68
tardis.apps.anzsrc_codes.models (module), 68
tardis.apps.anzsrc_codes.settings (module), 68
tardis.apps.anzsrc_codes.tests (module), 68
tardis.apps.anzsrc_codes.tests.test_oaipmh (module), 67
tardis.apps.anzsrc_codes.tests.tests (module), 67
tardis.apps.anzsrc_codes.urls (module), 68
tardis.apps.anzsrc_codes.views (module), 68
tardis.apps.deep_storage_download_mapper (module), 69
tardis.apps.deep_storage_download_mapper.mapper (module), 68
tardis.apps.deep_storage_download_mapper.urls (module), 69
tardis.apps.dl_mapper_df_dir_only (module), 69
tardis.apps.dl_mapper_df_dir_only.mapper (module), 69
tardis.apps.dl_mapper_df_dir_only.urls (module), 69
tardis.apps.equipment (module), 71
tardis.apps.equipment.admin (module), 70
tardis.apps.equipment.forms (module), 70
tardis.apps.equipment.migrations (module), 70
tardis.apps.equipment.migrations.0001_initial (module), 69
tardis.apps.equipment.models (module), 70
tardis.apps.equipment.south_migrations (module), 70
tardis.apps.equipment.tests (module), 70
tardis.apps.equipment.urls (module), 71
tardis.apps.equipment.views (module), 71
tardis.apps.filepicker (module), 72
tardis.apps.filepicker.filepicker_settings (module), 71
tardis.apps.filepicker.urls (module), 71
tardis.apps.filepicker.utils (module), 71
tardis.apps.filepicker.views (module), 71
tardis.apps.mx_views (module), 72
tardis.apps.mx_views.urls (module), 72
tardis.apps.mx_views.views (module), 72
tardis.apps.oaipmh (module), 77
tardis.apps.oaipmh.models (module), 75
tardis.apps.oaipmh.provider (module), 74
tardis.apps.oaipmh.provider.base (module), 72
tardis.apps.oaipmh.server (module), 75
tardis.apps.oaipmh.tests (module), 75
tardis.apps.oaipmh.tests.provider (module), 75
tardis.apps.oaipmh.tests.provider.test_base (module), 74
tardis.apps.oaipmh.tests.test_oai (module), 75
tardis.apps.oaipmh.urls (module), 77
tardis.apps.oaipmh.views (module), 77
tardis.apps.publication_forms (module), 80
tardis.apps.publication_forms.default_settings (module), 77
tardis.apps.publication_forms.doi (module), 78
tardis.apps.publication_forms.email_text (module), 78
tardis.apps.publication_forms.pub_form_config (module), 78
tardis.apps.publication_forms.tasks (module), 78
tardis.apps.publication_forms.tests (module), 77
tardis.apps.publication_forms.tests.test_email_templates (module), 77
tardis.apps.publication_forms.utils (module), 79
tardis.apps.publication_forms.views (module), 79
tardis.apps.push_to (module), 85
tardis.apps.push_to.apps (module), 81
tardis.apps.push_to.exceptions (module), 81
tardis.apps.push_to.migrations (module), 81
tardis.apps.push_to.migrations.0001_initial (module), 80
tardis.apps.push_to.migrations.0002_auto_20160518_1953 (module), 80
tardis.apps.push_to.models (module), 81
tardis.apps.push_to.oauth_tokens (module), 84
tardis.apps.push_to.ssh_authz (module), 84
tardis.apps.push_to.tests (module), 81
tardis.apps.push_to.tests.test_models (module), 81
tardis.apps.push_to.utils (module), 84
tardis.apps.related_info (module), 87
tardis.apps.related_info.forms (module), 86
tardis.apps.related_info.models (module), 87
tardis.apps.related_info.settings (module), 87
tardis.apps.related_info.tests (module), 86
tardis.apps.related_info.tests.test_oaipmh (module), 85
tardis.apps.related_info.tests.tests (module), 85
tardis.apps.related_info.urls (module), 87
tardis.apps.related_info.views (module), 87
tardis.apps.slideshow_view (module), 87
tardis.apps.slideshow_view.urls (module), 87
tardis.default_settings (module), 90
tardis.default_settings.admins (module), 87
tardis.default_settings.analytics (module), 87
tardis.default_settings.apps (module), 87
tardis.default_settings.auth (module), 87
tardis.default_settings.caches (module), 87
tardis.default_settings.celery (module), 88

tardis.default_settings.custom_views (module), 88
tardis.default_settings.database (module), 88
tardis.default_settings.debug (module), 88
tardis.default_settings.downloads (module), 89
tardis.default_settings.email (module), 89
tardis.default_settings.filters (module), 89
tardis.default_settings.frontend (module), 89
tardis.default_settings.i18n (module), 89
tardis.default_settings.localisation (module), 89
tardis.default_settings.logging (module), 89
tardis.default_settings.middlewares (module), 89
tardis.default_settings.publication (module), 89
tardis.default_settings.search (module), 89
tardis.default_settings.sftp (module), 89
tardis.default_settings.sharing (module), 89
tardis.default_settings.site_customisations (module), 89
tardis.default_settings.staging (module), 90
tardis.default_settings.static_files (module), 90
tardis.default_settings.storage (module), 90
tardis.default_settings.templates (module), 90
tardis.default_settings.uploads (module), 90
tardis.default_settings.urls (module), 90
tardis.manage (module), 177
tardis.search (module), 92
tardis.search.forms (module), 90
tardis.search.search_indexes (module), 90
tardis.search.utils (module), 92
tardis.settings_changeme (module), 177
tardis.tardis_portal (module), 176, 177
tardis.tardis_portal.admin (module), 146
tardis.tardis_portal.ands_doi (module), 151
tardis.tardis_portal.api (module), 151
tardis.tardis_portal.auth (module), 99
tardis.tardis_portal.auth.authentication (module), 92
tardis.tardis_portal.auth.authorisation (module), 93
tardis.tardis_portal.auth.authservice (module), 93
tardis.tardis_portal.auth.decorators (module), 95
tardis.tardis_portal.auth.fix_circular (module), 96
tardis.tardis_portal.auth.httpbasicendpoint_auth (module), 96
tardis.tardis_portal.auth.interfaces (module), 97
tardis.tardis_portal.auth.localdb_auth (module), 98
tardis.tardis_portal.auth.token_auth (module), 99
tardis.tardis_portal.auth.utils (module), 99
tardis.tardis_portal.constants (module), 159
tardis.tardis_portal.context_processors (module), 159
tardis.tardis_portal.creativecommonshandler (module), 160
tardis.tardis_portal.email (module), 160
tardis.tardis_portal.errors (module), 160
tardis.tardis_portal.fetcher (module), 161
tardis.tardis_portal.fields (module), 161
tardis.tardis_portal.filters (module), 101
tardis.tardis_portal.filters.diffractionimage (module), 99
tardis.tardis_portal.filters.jeolsem (module), 101
tardis.tardis_portal.forms (module), 161
tardis.tardis_portal.hacks (module), 170
tardis.tardis_portal.iiif (module), 170
tardis.tardis_portal.logging_middleware (module), 170
tardis.tardis_portal.management (module), 105
tardis.tardis_portal.management.commands (module), 105
tardis.tardis_portal.management.commands.backupdb (module), 101
tardis.tardis_portal.management.commands.checkhashes (module), 102
tardis.tardis_portal.management.commands.chownexperiment (module), 102
tardis.tardis_portal.management.commands.cleanuptokens (module), 103
tardis.tardis_portal.management.commands.createmysuperuser (module), 103
tardis.tardis_portal.management.commands.createtokenuser (module), 103
tardis.tardis_portal.management.commands.createuser (module), 103
tardis.tardis_portal.management.commands.dumpschemas (module), 104
tardis.tardis_portal.management.commands.loadschemas (module), 104
tardis.tardis_portal.management.commands.rmexperiment (module), 104
tardis.tardis_portal.management.commands.runfilters (module), 105
tardis.tardis_portal.managers (module), 170
tardis.tardis_portal.migrations (module), 108
tardis.tardis_portal.migrations.0001_initial (module), 105
tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643 (module), 105
tardis.tardis_portal.migrations.0002_auto_20150528_1128 (module), 106
tardis.tardis_portal.migrations.0003_auto_20150907_1315 (module), 106
tardis.tardis_portal.migrations.0004_storageboxoption_value_type (module), 106
tardis.tardis_portal.migrations.0005_datafile_add_size_int_column (module), 106
tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column (module), 106
tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index (module), 107
tardis.tardis_portal.migrations.0008_string_value_partial_index_postgres (module), 107
tardis.tardis_portal.migrations.0009_auto_20160128_1119 (module), 107
tardis.tardis_portal.migrations.0010_auto_20160503_1443 (module), 107
tardis.tardis_portal.migrations.0011_auto_20160505_1643

(module), 107
tardis.tardis_portal.models (module), 127, 176, 177
tardis.tardis_portal.models.access_control (module), 108
tardis.tardis_portal.models.datafile (module), 110
tardis.tardis_portal.models.dataset (module), 113
tardis.tardis_portal.models.experiment (module), 114
tardis.tardis_portal.models.facility (module), 116
tardis.tardis_portal.models.hooks (module), 116
tardis.tardis_portal.models.instrument (module), 116
tardis.tardis_portal.models.jti (module), 117
tardis.tardis_portal.models.license (module), 117
tardis.tardis_portal.models.parameters (module), 118
tardis.tardis_portal.models.storage (module), 124
tardis.tardis_portal.models.token (module), 126
tardis.tardis_portal.MultiPartForm (module), 145
tardis.tardis_portal.ParameterSetManager (module), 146
tardis.tardis_portal.publish (module), 128
tardis.tardis_portal.publish.provider (module), 127
tardis.tardis_portal.publish.provider.rifcsprovider (module), 127
tardis.tardis_portal.publish.publishservice (module), 127
tardis.tardis_portal.rfc3339 (module), 172
tardis.tardis_portal.search_backend (module), 173
tardis.tardis_portal.search_query (module), 173
tardis.tardis_portal.shortcuts (module), 173
tardis.tardis_portal.signals (module), 174
tardis.tardis_portal.staging (module), 174
tardis.tardis_portal.storage (module), 128
tardis.tardis_portal.storage.file_system (module), 128
tardis.tardis_portal.storage.squashfs (module), 128
tardis.tardis_portal.tasks (module), 175
tardis.tardis_portal.templatetags (module), 131
tardis.tardis_portal.templatetags.basiccomparisonfilters (module), 129
tardis.tardis_portal.templatetags.bleach_tag (module), 129
tardis.tardis_portal.templatetags.capture (module), 129
tardis.tardis_portal.templatetags.dynurl (module), 129
tardis.tardis_portal.templatetags.experimentstats (module), 130
tardis.tardis_portal.templatetags.facility_tags (module), 130
tardis.tardis_portal.templatetags.feed (module), 130
tardis.tardis_portal.templatetags.formfieldfilters (module), 130
tardis.tardis_portal.templatetags.lookupfilters (module), 130
tardis.tardis_portal.templatetags.uploadify_tags (module), 130
tardis.tardis_portal.templatetags.xmldate (module), 131
tardis.tardis_portal.tests (module), 145
tardis.tardis_portal.tests.auth (module), 131
tardis.tardis_portal.tests.filters (module), 132
tardis.tardis_portal.tests.filters.test_jeolsem (module), 131
tardis.tardis_portal.tests.filters.test_middleware (module), 131
tardis.tardis_portal.tests.ldap_ldif (module), 132
tardis.tardis_portal.tests.mock_vbl_auth (module), 132
tardis.tardis_portal.tests.mock_vbl_download (module), 132
tardis.tardis_portal.tests.slapd (module), 132
tardis.tardis_portal.tests.test_ands_doi (module), 134
tardis.tardis_portal.tests.test_api (module), 134
tardis.tardis_portal.tests.test_authentication (module), 136
tardis.tardis_portal.tests.test_authorisation (module), 136
tardis.tardis_portal.tests.test_authservice (module), 137
tardis.tardis_portal.tests.test_collectstatic (module), 137
tardis.tardis_portal.tests.test_download (module), 138
tardis.tardis_portal.tests.test_download_apikey (module), 138
tardis.tardis_portal.tests.test_fetcher (module), 138
tardis.tardis_portal.tests.test_forms (module), 139
tardis.tardis_portal.tests.test_iiif (module), 139
tardis.tardis_portal.tests.test_ldap (module), 140
tardis.tardis_portal.tests.test_models (module), 140
tardis.tardis_portal.tests.test_parametersetmanager (module), 141
tardis.tardis_portal.tests.test_publishservice (module), 141
tardis.tardis_portal.tests.test_rmexperiment (module), 142
tardis.tardis_portal.tests.test_staging (module), 142
tardis.tardis_portal.tests.test_storage (module), 142
tardis.tardis_portal.tests.test_tar_download (module), 142
tardis.tardis_portal.tests.test_tasks (module), 143
tardis.tardis_portal.tests.test_tokens (module), 143
tardis.tardis_portal.tests.tests (module), 143
tardis.tardis_portal.widgets (module), 175
tardis.tardis_portal.xmlwriter (module), 175
tardis.test_on_mysql_settings (module), 177
tardis.test_on_postgresql_settings (module), 177
tardis.test_settings (module), 177
tardis.views (module), 177
TarDownloadTestCase (class in tardis.tardis_portal.tests.test_tar_download), 142
tearDown() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestC method), 74
tearDown() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase method), 75
tearDown() (tardis.tardis_portal.tests.test_ands_doi.ANDSDOITestCase method), 134
tearDown() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 136

tearDown() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase method), 137
tearDown() (tardis.tardis_portal.tests.test_collectstatic.CollectstaticTest method), 137
tearDown() (tardis.tardis_portal.tests.test_download.DownloadTestCase method), 138
tearDown() (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownloadTestCase method), 138
tearDown() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
tearDown() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase method), 141
tearDown() (tardis.tardis_portal.tests.test_staging.TestStagingFiles method), 142
tearDown() (tardis.tardis_portal.tests.test_storage.ModelTestCase method), 142
tearDown() (tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase method), 142
tearDown() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
tearDown() (tardis.tardis_portal.tests.tests.SearchTestCase method), 144
tearDownClass() (tardis.tardis_portal.tests.test_fetcher.PrivilegedOpenerTestCase class method), 138
temp_url (tardis.tardis_portal.api.DataFileResource attribute), 152
TEMPORARY (tardis.tardis_portal.models.storage.StorageBox attribute), 124
test_apikey_authentication() (tardis.tardis_portal.tests.test_api.MyTardisAuthenticationTest method), 135
test_authenticate() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
test_authors() (tardis.tardis_portal.tests.test_models.ModelTestCase method), 140
test_bad_credentials() (tardis.tardis_portal.tests.test_api.MyTardisAuthenticationTest method), 135
test_collectstatic() (tardis.tardis_portal.tests.test_collectstatic.CollectstaticTest method), 137
test_create_user_automatically_generate_api_key() (tardis.tardis_portal.tests.test_models.ModelTestCase method), 140
test_credential_generation() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 81
test_datafile() (tardis.tardis_portal.tests.test_models.ModelTestCase method), 140
test_dataset() (tardis.tardis_portal.tests.test_models.ModelTestCase method), 141
test_debug_serializer() (tardis.tardis_portal.tests.test_api.SerializerTest method), 136
test_default_expiry() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
test_djangoauth() (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase method), 136
test_download_apikey() (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownloadTestCase method), 138
test_ensures_suitable_license() (tardis.tardis_portal.tests.test_forms.RightsFormTestCase method), 139
test_experiment() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase method), 141
test_get_allowed_signing_services() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 81
test_get_dataset_filter_instrument() (tardis.tardis_portal.tests.test_api.DatasetResourceTest method), 135
test_get_dataset_no_instrument() (tardis.tardis_portal.tests.test_api.DatasetResourceTest method), 135
test_get_dataset_with_instrument() (tardis.tardis_portal.tests.test_api.DatasetResourceTest method), 135
test_get_doi_none() (tardis.tardis_portal.tests.test_ands_doi.ANDSDOITest method), 134
test_get_experiment() (tardis.tardis_portal.tests.test_api.ExperimentResource method), 135
test_get_facility_by_id() (tardis.tardis_portal.tests.test_api.FacilityResource method), 135
test_get_facility_by_manager_group_id() (tardis.tardis_portal.tests.test_api.FacilityResourceTest method), 135
test_get_facility_by_name() (tardis.tardis_portal.tests.test_api.FacilityResourceTest method), 135
test_get_group_by_id() (tardis.tardis_portal.tests.test_api.GroupResourceTest method), 135
test_get_group_by_name() (tardis.tardis_portal.tests.test_api.GroupResourceTest method), 135
test_get_instrument_by_id() (tardis.tardis_portal.tests.test_api.InstrumentResourceTest method), 135
test_get_instrument_by_name() (tardis.tardis_portal.tests.test_api.InstrumentResourceTest method), 135
test_get_session_expiry() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
test_get_session_expiry_expired_token() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
test_get_session_expiry_near_expiry() (tardis.tardis_portal.tests.test_tokens.TokenTestCase

method), 143
test_getgroupbyid() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
test_getgroups() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
test_getgroupsforentity() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
test_getuserbyid() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
test_init() (tardis.tardis_portal.tests.test_ands_doi.ANDSDOITestCase method), 134
test_is_expired() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
test_key_type_detection() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 81
test_keys_from_private_key_only() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 81
test_link_parameter_type() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase method), 141
test_link_parameter_type_extra() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase method), 141
test_login() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase method), 144
test_needs_confirmation() (tardis.tardis_portal.tests.test_forms.RightsFormTestCase method), 139
test_new_parameterset() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase method), 141
test_parameter() (tardis.tardis_portal.tests.test_models.ModelTestCase method), 141
test_pkey_to_credential() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 81
test_post_dataset() (tardis.tardis_portal.tests.test_api.DatasetResourceTest method), 135
test_post_experiment() (tardis.tardis_portal.tests.test_api.ExperimentResourceTest method), 135
test_post_instrument() (tardis.tardis_portal.tests.test_api.InstrumentResourceTest method), 135
test_post_single_file() (tardis.tardis_portal.tests.test_api.DataFileResourceTest method), 134
test_pretty_serializer() (tardis.tardis_portal.tests.test_api.SerializerTest method), 136
test_pub_approved_template() (tardis.apps.publication_forms.tests.test_email_templates.EmailTemplates method), 77
test_pub_awaiting_approval_template() (tardis.apps.publication_forms.tests.test_email_templates.EmailTemplates method), 77
test_pub_rejected_template() (tardis.apps.publication_forms.tests.test_email_templates.EmailTemplates method), 77
test_pub_released_template() (tardis.apps.publication_forms.tests.test_email_templates.EmailTemplates method), 77
test_pub_requires_authorisation_template() (tardis.apps.publication_forms.tests.test_email_templates.EmailTemplates method), 77
test_pub_reverted_to_draft_template() (tardis.apps.publication_forms.tests.test_email_templates.EmailTemplates method), 77
test_rename_instrument() (tardis.tardis_portal.tests.test_api.InstrumentResourceTest method), 135
test_reroot() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase method), 144
test_save_with_random_token() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
test_save_with_random_token_failures() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
test_save_with_random_token_gives_up() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 143
test_search() (tardis.tardis_portal.tests.test_ldap.LDAPErrorTest method), 140
test_search() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
test_search_urls() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase method), 144
test_searchgroups() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 140
test_shared_fs_many_files() (tardis.tardis_portal.tests.test_api.DataFileResourceTest method), 134
test_shared_fs_single_file() (tardis.tardis_portal.tests.test_api.DataFileResourceTest method), 134
test_staging_replica() (tardis.tardis_portal.tests.test_staging.TestStagingFiles method), 142
test_storage_exception() (tardis.tardis_portal.tests.test_storage.ModelTestCase method), 142
test_streaming_experiment_download() (tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase method), 142
test_tz_aware_date_handling() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase method), 141
test_tz_naive_date_handling() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase method), 141
test_unresolvable_link_parameter() (tardis.tardis_portal.tests.test_parametersetmanager.ParameterSetManagerTestCase

method), 141
test_urls() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase method), 144
test_urls_with_some_content() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase method), 144
TEST_UTILS_DIR (tardis.tardis_portal.tests.slapd.Slapd attribute), 133
test_wrong_size_verification() (tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase method), 143
testAccessWithoutReadPerms() (tardis.apps.anzsrc_codes.tests.tests.TabTestCase method), 67
testAccessWithoutReadPerms() (tardis.apps.related_info.tests.tests.TabTestCase method), 86
testAccessWithReadPerms() (tardis.apps.anzsrc_codes.tests.tests.TabTestCase method), 67
testAccessWithReadPerms() (tardis.apps.related_info.tests.tests.TabTestCase method), 86
testAuthenticate() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase method), 137
testCanCreate() (tardis.apps.related_info.tests.tests.CreateTestCase method), 85
testCanDelete() (tardis.apps.related_info.tests.tests.DeleteTestCase method), 85
testCanGetInfoAsJSON() (tardis.tardis_portal.tests.test_iiif.Level0TestCase method), 139
testCanGetInfoAsXML() (tardis.tardis_portal.tests.test_iiif.Level0TestCase method), 139
testCanGetJpegFormat() (tardis.tardis_portal.tests.test_iiif.Level0TestCase method), 139
testCanGetOriginalImage() (tardis.tardis_portal.tests.test_iiif.Level0TestCase method), 139
testCanGetRequiredFormats() (tardis.tardis_portal.tests.test_iiif.Level2TestCase method), 140
testCantAddTokenuserToGroups() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 136
testCantEditLockedExperiment() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 136
testChangeUserPermissions() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 136
testContext() (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase method), 141
testCreateNewAccount() (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase method), 136
testCredentials() (tardis.tardis_portal.tests.test_fetcher.PrivilegedOpenerTestCase method), 138
testDatasetFile() (tardis.tardis_portal.tests.test_download.DownloadTestCase method), 138
testDetectsBadInput() (tardis.apps.related_info.tests.tests.CreateTestCase method), 85
testDetectsBadInput() (tardis.apps.related_info.tests.tests.UpdateTestCase method), 86
testDownload() (tardis.tardis_portal.tests.test_download.DownloadTestCase method), 138
testEquipmentDetail() (tardis.apps.equipment.tests.EquipmentTestCase method), 71
testEquipmentList() (tardis.apps.equipment.tests.EquipmentTestCase method), 71
testExistsInOaipmh() (tardis.apps.anzsrc_codes.tests.test_oaipmh.RifCSTestCase method), 67
testExistsInOaipmh() (tardis.apps.related_info.tests.test_oaipmh.RifCSTestCase method), 85
testFiltering() (tardis.tardis_portal.tests.filters.test_middleware.FilterInitTestCase method), 132
testGetGroupsForEntity() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase method), 137
testGetRecord() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase method), 74
testGetRecord() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase method), 75
testGroupProvider() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase method), 137
testGroupSearch() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase method), 137
testHandleRegions() (tardis.tardis_portal.tests.test_iiif.Level1TestCase method), 140
testHandleRotation() (tardis.tardis_portal.tests.test_iiif.Level1TestCase method), 140
testHandlesEmptySet() (tardis.apps.related_info.tests.tests.ListTestCase method), 86
testHandlesFound() (tardis.apps.related_info.tests.tests.GetTestCase method), 86
testHandleSizing() (tardis.tardis_portal.tests.test_iiif.Level1TestCase method), 140
testHandleSizing() (tardis.tardis_portal.tests.test_iiif.Level2TestCase method), 140
testHandlesMultipleEntries() (tardis.apps.related_info.tests.tests.ListTestCase method), 86
testHandlesNotFound() (tardis.apps.related_info.tests.tests.GetTestCase method), 86
testHandlesSingleEntry() (tardis.apps.related_info.tests.tests.ListTestCase method), 86
testIdentify() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase

method), 74
testIdentify() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase method), 75
testImageCacheControl() (tardis.tardis_portal.tests.test_iiif.ExtraTestCases method), 139
testImageHasEtags() (tardis.tardis_portal.tests.test_iiif.ExtraTestCases method), 139
testInfoHasEtags() (tardis.tardis_portal.tests.test_iiif.ExtraTestCases method), 139
testInitialisation() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase method), 137
testInitialisation() (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase method), 141
testInitialisationNoProvider() (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase method), 142
testJEOLComplex() (tardis.tardis_portal.tests.filters.test_jeolsem.JEOLSEMFilterTestCase method), 131
testJEOLSimple() (tardis.tardis_portal.tests.filters.test_jeolsem.JEOLSEMFilterTestCase method), 131
testList() (tardis.tardis_portal.tests.test_rmexperiment.RmExperimentTestCase method), 142
testListIdentifiers() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase method), 74
testListIdentifiers() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase method), 75
testListMetadataFormats() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase method), 74
testListMetadataFormats() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase method), 75
testListRecords() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase method), 74
testListRecords() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase method), 75
testListSets() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase method), 74
testLocalFile() (tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase method), 143
testManageAuthMethods() (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase method), 136
testManageRifCsCheckContent() (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase method), 142
testManageRifCsCreateAndRemove() (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase method), 142
testMustHaveWrite() (tardis.apps.related_info.tests.tests.CreateTestCase method), 85
testMustHaveWrite() (tardis.apps.related_info.tests.tests.DeleteTestCase method), 85
testMustHaveWrite() (tardis.apps.related_info.tests.tests.UpdateTestCase method), 86
testOwnedExperiments() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 136
testReadAccess() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 136
testRemove() (tardis.tardis_portal.tests.test_rmexperiment.RmExperimentTestCase method), 142
testSearchDatafileAuthentication() (tardis.tardis_portal.tests.tests.SearchTestCase method), 144
testSearchDatafileForm() (tardis.tardis_portal.tests.tests.SearchTestCase method), 144
testSearchEquipmentForm() (tardis.apps.equipment.tests.EquipmentTestCase method), 71
testSearchEquipmentResult() (tardis.apps.equipment.tests.EquipmentTestCase method), 71
testSearchExperimentAuthentication() (tardis.tardis_portal.tests.tests.SearchTestCase method), 144
testSearchExperimentForm() (tardis.tardis_portal.tests.tests.SearchTestCase method), 144
testSearchExperimentResults() (tardis.tardis_portal.tests.tests.SearchTestCase method), 144
testSimpleAuthenticate() (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase method), 136
TestStagingHelper (class in tardis.tardis_portal.tests.test_staging), 142
TestWeb() (tardis.tardis_portal.tests.test_download.DownloadTestCase method), 138
TestWebServer (class in tardis.tardis_portal.tests.test_fetcher), 138
TestWebServer.QuietSimpleHTTPRequestHandler (class in tardis.tardis_portal.tests.test_fetcher), 138
TestWebServer.ThreadedTCPServer (class in tardis.tardis_portal.tests.test_fetcher), 138
testWriteAccess() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 137
text (tardis.search.search_indexes.DataFileIndex attribute), 91
text (tardis.search.search_indexes.DatasetIndex attribute), 91
text (tardis.search.search_indexes.ExperimentIndex attribute), 92
to_json() (tardis.tardis_portal.api.PrettyJSONSerializer method), 157
todatetime() (in module

tardis.tardis_portal.templatetags.feed), 130
Token (class in tardis.tardis_portal.models.token), 126
Token.DoesNotExist, 126
Token.MultipleObjectsReturned, 126
token_set (tardis.tardis_portal.models.experiment.Experiment attribute), 115
TokenAuthMiddleware (class in tardis.tardis_portal.auth.token_auth), 99
TokenGroupProvider (class in tardis.tardis_portal.auth.token_auth), 99
TokenTestCase (class in tardis.tardis_portal.tests.test_tokens), 143
tokenuser_processor() (in module tardis.tardis_portal.context_processors), 160
toxmldatetime() (in module tardis.tardis_portal.templatetags.xmldate), 131
track_download() (in module tardis.analytics.ga), 66
track_login() (in module tardis.analytics.ga), 66
track_logout() (in module tardis.analytics.ga), 66
tuples (tardis.tardis_portal.tests.filters.test_middleware.Filter1 attribute), 131
tuples (tardis.tardis_portal.tests.filters.test_middleware.Filter2 attribute), 131
TYPE_CHOICES (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 126
type_order (tardis.tardis_portal.models.storage.StorageBox attribute), 125
TYPE_UNKNOWN (tardis.tardis_portal.models.storage.StorageBox attribute), 124
TYPES (tardis.tardis_portal.models.storage.StorageBox attribute), 124

U

unpickled_value (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 126
UnsupportedSearchQueryTypeError, 161
update_detail() (tardis.tardis_portal.api.ACLAuthorization method), 152
update_list() (tardis.tardis_portal.api.ACLAuthorization method), 152
update_mimetype() (tardis.tardis_portal.models.datafile.DataFile method), 111
UpdateTestCase (class in tardis.apps.related_info.tests.tests), 86
upload_auth() (in module tardis.tardis_portal.auth.decorators), 96
UPLOAD_METHOD (in module tardis.default_settings.uploads), 90
URL (tardis.tardis_portal.models.parameters.ParameterName attribute), 122
url_safe_query() (tardis.search.utils.SearchQueryString method), 92
urls (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase attribute), 137
urls (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase attribute), 137
urls (tardis.tardis_portal.tests.test_tokens.TokenTestCase attribute), 143
user (tardis.apps.push_to.models.Credential attribute), 82
user (tardis.tardis_portal.models.access_control.GroupAdmin attribute), 108
user (tardis.tardis_portal.models.access_control.UserProfile attribute), 110
user (tardis.tardis_portal.models.token.Token attribute), 126
user_acls() (tardis.tardis_portal.managers.ExperimentManager method), 172
user_details_processor() (in module tardis.tardis_portal.context_processors), 160
user_owned_groups() (tardis.tardis_portal.managers.ExperimentManager method), 172
UserAuthentication (class in tardis.tardis_portal.models.access_control), 109
UserAuthentication.DoesNotExist, 109
UserAuthentication.MultipleObjectsReturned, 109
userAuthentication_set (tardis.tardis_portal.models.access_control.UserProfile attribute), 110
UserInterfaceTestCase (class in tardis.tardis_portal.tests.tests), 144
UserProfile (class in tardis.tardis_portal.models.access_control), 109
userProfile (tardis.tardis_portal.models.access_control.UserAuthentication attribute), 109
UserProfile.DoesNotExist, 109
UserProfile.MultipleObjectsReturned, 109
UserProvider (class in tardis.tardis_portal.auth.interfaces), 97
UserResource (class in tardis.tardis_portal.api), 158
UserResource.Meta (class in tardis.tardis_portal.api), 159
users() (tardis.tardis_portal.managers.ExperimentManager method), 172

V

validate_digest() (in module tardis.tardis_portal.management.commands.checkhashes), 102
ValidationFailed, 102
value_from_datadict() (tardis.tardis_portal.forms.ChangeGroupPermissions method), 162
value_from_datadict() (tardis.tardis_portal.forms.ChangeUserPermissionsF method), 163
value_from_datadict() (tardis.tardis_portal.widgets.CommaSeparatedInput method), 175
verbose_name (tardis.analytics.apps.AnalyticsConfig attribute), 66

verbose_name (tardis.apps.push_to.apps.PushToConfig attribute), 81
verified (tardis.tardis_portal.models.datafile.DataFile attribute), 111
verify() (tardis.tardis_portal.models.datafile.DataFile method), 111
verify() (tardis.tardis_portal.models.datafile.DataFileObject method), 112
verify_file() (tardis.tardis_portal.api.DataFileResource method), 152
verify_remote_access() (tardis.apps.push_to.models.Credential method), 82
view_functions (tardis.tardis_portal.shortcuts.RestfulExperimentParameterSet attribute), 174
view_id() (in module tardis.apps.equipment.views), 71
view_key() (in module tardis.apps.equipment.views), 71

W

wait() (tardis.tardis_portal.tests.slapd.Slapd method), 133
walk() (tardis.tardis_portal.storage.squashfs.SquashFSStorage method), 128
which() (in module tardis.tardis_portal.tests.slapd), 134
widget (tardis.tardis_portal.forms.StaticField attribute), 169
widgets (tardis.apps.push_to.models.CredentialForm.Meta attribute), 82
widgets (tardis.tardis_portal.admin.DatafileAdminForm.Meta attribute), 147
widgets (tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta attribute), 147
widgets (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta attribute), 150
widgets (tardis.tardis_portal.admin.StorageBoxForm.Meta attribute), 150
widgets (tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute), 151
widgets (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta attribute), 164
widgets (tardis.tardis_portal.forms.RightsForm.Meta attribute), 169
write_permissions_required() (in module tardis.tardis_portal.auth.decorators), 96
write_template_to_dir() (tardis.tardis_portal.xmlwriter.XMLWriter static method), 175
write_template_to_file() (tardis.tardis_portal.xmlwriter.XMLWriter static method), 175
write_uploaded_file_to_dataset() (in module tardis.tardis_portal.staging), 174
write_xml_to_file() (tardis.tardis_portal.xmlwriter.XMLWriter static method), 176
writeMetadata() (tardis.apps.oaipmh.provider.base.BaseProvider method), 73, 181
writeMetadata() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry method), 75

X

XMLWriter (class in tardis.tardis_portal.xmlwriter), 175

Y

year_field (tardis.tardis_portal.forms.ChangeGroupPermissionsForm.Select attribute), 162
year_field (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.Se attribute), 163