
MyTardis Documentation

Release 4.2

Sep 17, 2020

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	42
11.3	Development	64
11.4	Documentation for included Apps	245
11.5	Releases	252
12	Indices and tables	263
	Python Module Index	265
	Index	271

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 meta-data. 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 [Kubernetes](#) 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/>

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 bug fixes 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, the latest release is `4.2.0`, tagged from the `series-4.2` branch. Follow this branch for your production installation and perform version upgrades manually.

Each bug fix or set of fixes bumps the minor version and each new release is tagged, eg. `4.2.1`. 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.

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.

MyTardis Home About **My Data** Public Data Stats Help

Experiments

1 Experiment You Own

[+ Create](#)

[Collapse all](#) [Expand all](#)

Test Experiment	yesterday	1	0	Private
Download data as .tar				
Peter Capaldi				
Microscopy experiment				
Latest dataset in this experiment				
Dataset 1				

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

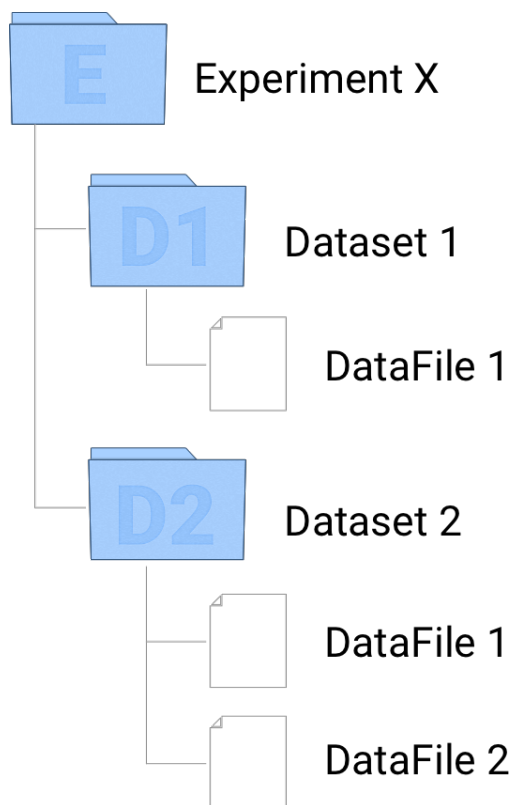


Fig. 1: Fig. 1. Data hierarchy in MyTardis

MyTardis

HomeAboutMy DataPublic DataStatsHelp

thedoctor

Create Experiment

Title

Test Experiment

Authors

Peter Capaldi

Comma-separated authors and optional emails/URLs

Institution name

Monash University

Description

Microscopy experiment

Save

Cancel

Powered by 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.

The screenshot shows the 'Test Experiment' page in MyTardis. The top navigation bar includes 'MyTardis', 'Home', 'About', 'My Data', 'Public Data', 'Stats', and 'Help'. The experiment details section shows 'Test Experiment' by Peter Capaldi. On the right, there is a '1 Dataset' section with a '+ Add New' button highlighted by a red box. The page also includes a navigation bar at the top and a footer indicating it is powered by MyTardis.

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

The screenshot shows the 'Add new dataset' form. The form has three input fields: 'Description' (containing 'Dataset 1'), 'Directory', and 'Instrument'. A 'Save' button is at the bottom. The page is powered by MyTardis.

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.

MyTardis
Home
About
My Data
Public Data
Stats
Help
thedoctor

Dataset
Dataset 1
From the experiment: [Test Experiment](#)

Edit Dataset Description

1 0 0 bytes

Datafile Information
Click the button next to any file to view details and actions.

Dataset Metadata
There is no metadata for this dataset.

Other Experiments
This dataset doesn't appear in any other experiments.

0 Files
Download Dataset: [TAR](#) [SFTP](#) [MD5](#)

Add files...

Search: Enter part or all of a filename, then press ↵

Select: [All](#) / [None](#) [Download Selected Files](#)

Powered by MyTardis

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:

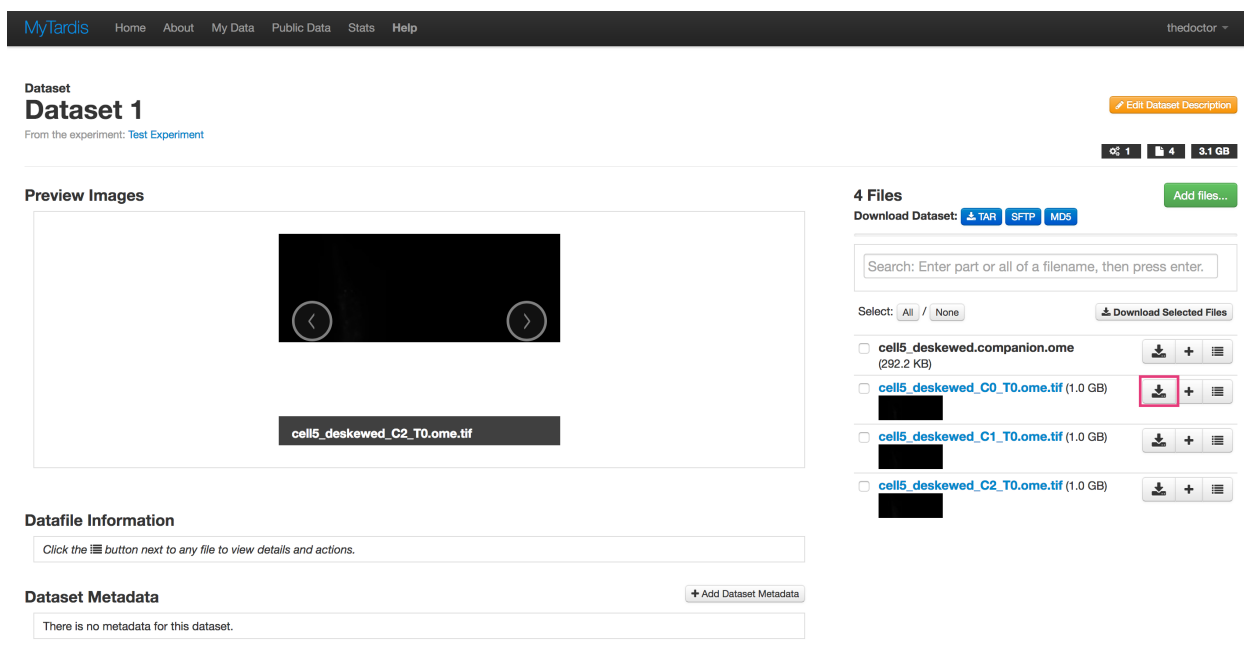
1. Navigate to the Dataset containing the Data File you want to download.
2. Click the **Download** button on the Data File you want to download.

MyTardis
Home
About
My Data
Public Data
Stats
Help
thedoctor

Dataset
Dataset 1
From the experiment: [Test Experiment](#)

Edit Dataset Description

1 4 3.1 GB

Preview Images


cell5_deskewed_C2_T0.ome.tif

Datafile Information
Click the button next to any file to view details and actions.

Dataset Metadata
There is no metadata for this dataset.

4 Files
Download Dataset: [TAR](#) [SFTP](#) [MD5](#)

Add files...

Search: Enter part or all of a filename, then press enter.

Select: [All](#) / [None](#) [Download Selected Files](#)

☐ cell5_deskewed.companion.ome (292.2 KB)

☐ cell5_deskewed_C0_T0.ome.tif (1.0 GB)

☐ cell5_deskewed_C1_T0.ome.tif (1.0 GB)

☐ cell5_deskewed_C2_T0.ome.tif (1.0 GB)

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.

The screenshot shows the MyTardis interface for 'Dataset 1'. The top navigation bar includes 'MyTardis', 'Home', 'About', 'My Data', 'Public Data', 'Stats', and 'Help'. The user 'thedoctor' is logged in. The dataset is titled 'Dataset 1' and is from the experiment 'Test Experiment'. It contains 4 files totaling 3.1 GB. The 'Preview Images' section shows a thumbnail of a file named 'cell5_deskewed_C2_T0.ome.tif'. The 'Datafile Information' section has a note: 'Click the [icon] button next to any file to view details and actions.' The 'Dataset Metadata' section states 'There is no metadata for this dataset.' On the right, the '4 Files' section shows download options: 'TAR' (highlighted), 'SFTP', and 'MD5'. Below this is a search bar and a list of files: 'cell5_deskewed.companion.ome' (292.2 KB), 'cell5_deskewed_C0_T0.ome.tif' (1.0 GB), 'cell5_deskewed_C1_T0.ome.tif' (1.0 GB), and 'cell5_deskewed_C2_T0.ome.tif' (1.0 GB). Each file has a download icon, a plus icon, and a list icon.

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.

MyTardis Home About My Data Public Data Stats Help thedoctor

Experiment
Test Experiment
Peter Capaldi

Create publication Edit Experiment

1 4 3.1 GB 5th December 2017 Private

Microscopy experiment

Description Metadata Sharing Transfer Datasets

Institution Monash University
Licensing Unspecified
Administrators thedoctor

Download All **TAR** SFTP

1 Dataset **Dataset 1** 3.1 GB

Powered by MyTardis

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	<i>none</i>

[↗️ 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 status.

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 status.

No tokens defined

[Create New Temporary Link](#)

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

User Sharing

Current User Permissions

[thedoctor] (doctor@tardis.org) Read Edit Owner Edit Delete

Add new user

Username:

Permissions:

View and Edit

Owners have the ability to change access controls and share experiments with others.

+ Add User

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.

User Sharing ×

Current User Permissions

[thedoctor] (doctor@tardis.org)	Read	Edit	Owner	Edit	Delete
[rosetyler] (rosetyler@tardis.org)	Read	Edit		Edit	Delete

Add new user

Username:

Permissions:

Owners have the ability to change access controls and share experiments with others.

[+ Add User](#)

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.

Links

This experiment is private. A temporary link can be created by its owner(s) and privately shared for direct access.

Temporary access links provide full access to recipients regardless of an experiment's public status.

No tokens defined




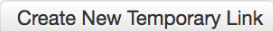
[Create New Temporary Link](#)

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.

Links

This experiment is private. A temporary link can be created by its owner(s) and privately shared for direct access.

Temporary access links provide full access to recipients regardless of an experiment's public status.

Temporary Link	Archive direct download	Expiry	Granted By
Page link  Copy to clipboard	Archive link  Copy to clipboard	10th January 2018	thedoctor  Delete
			

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.

Public Access

Step 1: Change Public Access:

Public access

Public

Step 2: Select a license:

Creative Commons Attribution 3.0 Australia (CC BY 3.0)

This licence lets others distribute, remix, tweak, and build upon your work, even commercially, as long as they credit you for the original creation. This is the most accommodating of licences offered under Creative Commons.


[Reselect License](#)

Step 3: Accept The Legal Agreement:

Monash University (ABN 12 377 614 012) ("University") maintains a digital repository ("Repository") in which it stores and makes available various works via the Internet or other electronic means.

By submitting material for the Repository ("Work") the person(s) submitting the Work ("Depositor") agree to the following terms and conditions:

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 in MyTardis. Importantly, the SFTP interface to MyTardis is read only i.e., it is a mechanism to access data in MyTardis but does **not** allow adding data to MyTardis.

Prerequisites for accessing data via SFTP

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 \(Win & Mac\)](#).

Registering and managing SSH keys in MyTardis

MyTardis uses SSH key pairs to authenticate you when connecting via SFTP. Using SSH keys to authenticate is more secure and private than password-based authentication. SSH keys generally come in a pair: (1) a private part and (2) a public part. Key-based authentication typically requires you to share your public key with the service, again which you use your private key to authenticate. Never share your private key!

MyTardis provide to mechanism for registering a public key:

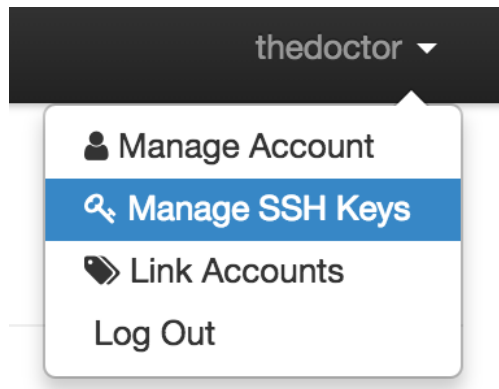
1. *Adding an existing public SSH key to MyTardis*
2. *Have MyTardis to generate an SSH key pair*

Adding an existing public SSH key to MyTardis

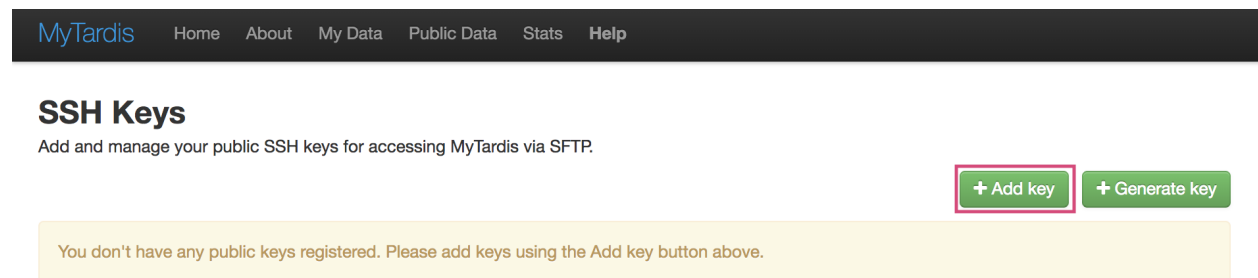
If you have an existing SSH keypair, you can register the public key in MyTardis. For instructions on how to create an SSH key pair, Gitlab provide some nice documentation on how to do so [here](#).

To register a public key in MyTardis:

1. Navigate to the SSH key management page using the *Manage SSH Keys* link in the user menu.



2. Click the + *Add Key* button:



3. Fill in the *Add SSH Keys* form by entering a name for the key and the public key text. The public key text should be in OpenSSH format e.g., `ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADYznXpTP6e/BsQw+...`

Add SSH Keys ✕

Add your own public SSH Key using the form below. **Important:** Don't add your private key.

Device Name:

Public Key:

```
ssh-rsa
AAAAB3NzaC1vc2EAAAADAQABAAQBAQDYZYnXpTP6e/BsQw+QEVdtpvdKJV2pxc9Q
DlSipqsV464bxerFs7+deNSHlpJiHy7zBO4zbzGd2vE2wuxXa02tGRwKqhQm3/fFRH6ZL4
pfJ/tkasYd7i5sAIKlkh0tPVVU+6HYbNbNjN+riIYh16ECrp7+YmtwPwVieeiTn69cSTkG+way
TFXwO6M3dolb/SMY8zE5JMotVnqxBJTwNwpQZHmQW4eh87eIHbv7QrtRF7ws/Ph6ME
Bpy1A9V118ysBKMG4YxybxGuwOOMDAOI/2RhqMGV1MPXlulshIOAy9vrYCeS40AukYjE
```

Close Add Key

4. Click the *Add Key* button.

If successful you should get a new entry in the SSH keys table.

MyTardis [Home](#) [About](#) [My Data](#) [Public Data](#) [Stats](#) [Help](#)

SSH Keys

Add and manage your public SSH keys for accessing MyTardis via SFTP.

+ Add key + Generate key

Key
<p>massive Delete</p> <p>Type: SSH-RSA</p> <p>Fingerprint: f68edca12548c068f8836d412230bdd4</p> <p>Date added: 2018-08-07</p>

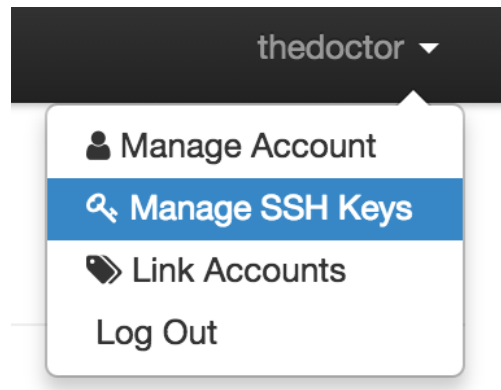
Have MyTardis to generate an SSH key pair

Creating SSH key pairs can be a bit of a challenge, particularly on Windows systems. For this reason, MyTardis provides another you another option for registering a public keys. In this case, MyTardis will generate the key pair for

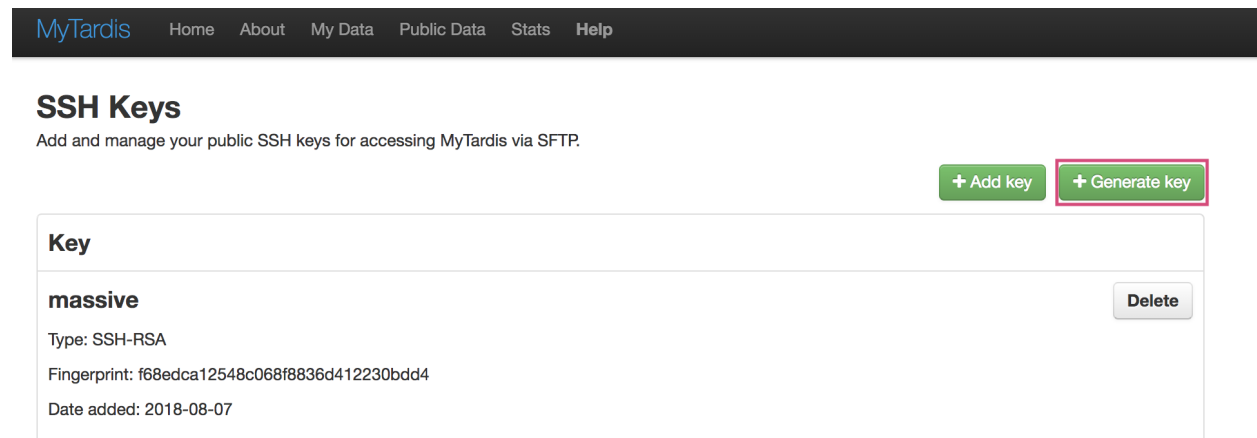
you, store the public part of the key and provide you with a one time opportunity to download the private part of the key.

It is important to note that MyTardis never stores your private key. If you lose/delete it, MyTardis cannot recover it for you. This isn't really a problem, since MyTardis can always generate another key pair for you; however, you should revoke the public key associated to the lost private key in your account to ensure that no one who gains access to your private key can access your data. See [Deleting previously registered public keys in MyTardis](#).

1. Navigate to the SSH key management page using the *Manage SSH Keys* link in the user menu.



2. Click the + *Generate key* button:



3. Give the key a name and click the *Generate & Download* button. The server will generate an SSH key pair, register the public key and trigger your browser to download the private key.

SFTP:

Generate an SSH key pair ×

MyTardis can generate an SSH key pair for you. Only the public part of the key pair will be stored by MyTardis and you will be provided with a one time opportunity to download the private part of the key pair.

Name:

Important: Keep your private key in a safe place. Anyone with access to the private key will be able to access your data on MyTardis.

- The SSH keys table will be updated with your newly registered public key.

MyTardis
Home
About
My Data
Public Data
Stats
Help

SSH Keys

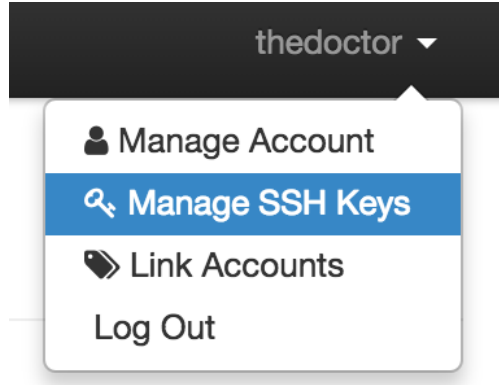
Add and manage your public SSH keys for accessing MyTardis via SFTP.

Key
<div> massive <input type="button" value="Delete"/> </div> <p>Type: SSH-RSA</p> <p>Fingerprint: f68edca12548c068f8836d412230bdd4</p> <p>Date added: 2018-08-07</p>
<div> TestKey <input type="button" value="Delete"/> </div> <p>Type: SSH-RSA</p> <p>Fingerprint: 3c4c3c3eb293d274fbeb687cc46aff76</p> <p>Date added: 2018-08-07</p>

Deleting previously registered public keys in MyTardis

Removing a previously registered public SSH key from MyTardis is straightforward.

1. Navigate to the SSH key management page using the *Manage SSH Keys* link in the user menu.



2. Click the *Delete* button on the key you wish to remove.

MyTardis Home About My Data Public Data Stats Help

SSH Keys

Add and manage your public SSH keys for accessing MyTardis via SFTP.

[+ Add key](#) [+ Generate key](#)

Key	
massive Type: SSH-RSA Fingerprint: f68edca12548c068f8836d412230bdd4 Date added: 2018-08-08	Delete
TestKey Type: SSH-RSA Fingerprint: e27936d5ccac5b7e5af83cefb29f01f0 Date added: 2018-08-09	Delete

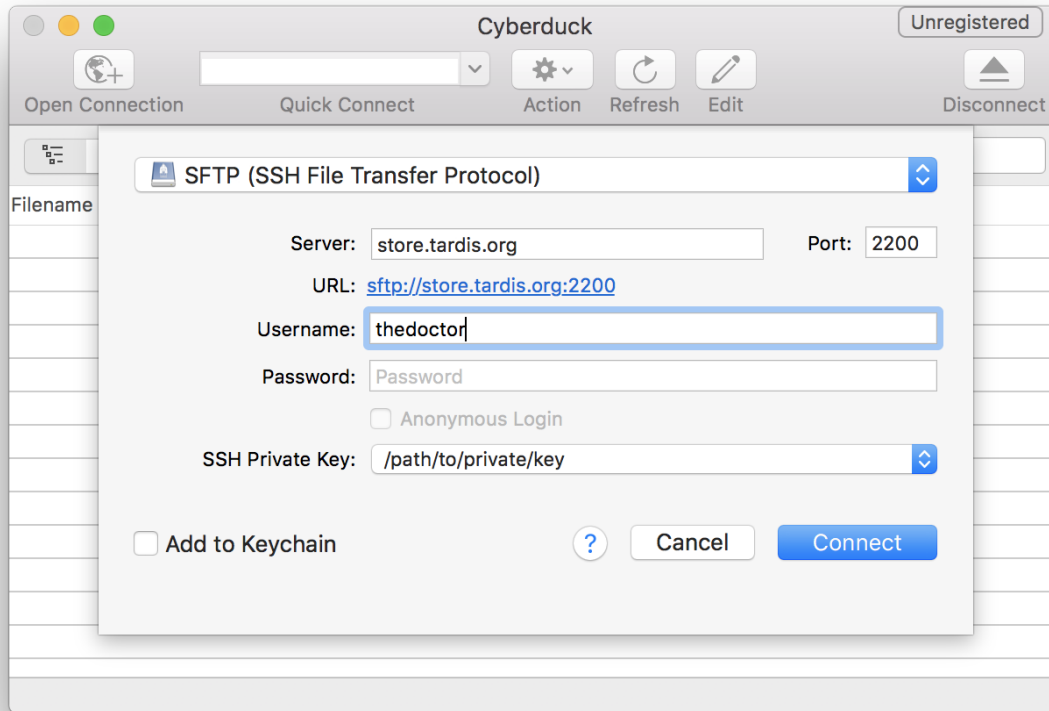
Attempting to connect to MyTardis via SFTP using the private sister key to the deleted key will no longer work.

Connecting to MyTardis via SFTP

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

Parameter	Value
Server:	<i>URL for your MyTardis deployment e.g. https://store.erc.monash.edu</i>
Port:	<i>Port on which SFTP server is running</i>
Username:	<i>Your MyTardis username</i>
Password:	<i>Leave this blank</i>
SSH Private Key	<i>Path to a private SSH key¹</i>

Note: substitute your credentials and details for your MyTardis deployment for the *italicised* values. If you are unsure about any of these value, please contact your system admin should be able to provide these to you.



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.

¹ You must register the public key in MyTardis first, see [Registering and managing SSH keys in MyTardis](#)

Experiment View

Experiment

Test Experiment

Peter Capaldi

Microscopy experiment

[Description](#) [Metadata](#) [Sharing](#) [Transfer Datasets](#)

Institution Monash University

Licensing Unspecified

Administrators Peter Capaldi

Download All [TAR](#) [SFTP](#)

Dataset View

4 Files

Download Dataset: [TAR](#) [SFTP](#) [MD5](#)[Add files...](#)

Search: Enter part or all of a filename, then press enter.

Select: [All](#) / [None](#)[Download Selected Files](#)

<input type="checkbox"/>	cell5_deskewed.companion.ome (292.2 KB)	Download + List
<input type="checkbox"/>	cell5_deskewed_C0_T0.ome.tif (1.0 GB)	Download + List
<input type="checkbox"/>	cell5_deskewed_C1_T0.ome.tif (1.0 GB)	Download + List
<input type="checkbox"/>	cell5_deskewed_C2_T0.ome.tif (1.0 GB)	Download + List

- 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 (18.04 LTS is recommended)

Run this script for Python 3:

```
sudo bash install-ubuntu-py3-requirements.sh
```

It will install required packages with these commands:

```
# for Ubuntu 16.04 or 18.04
# sudo bash install-ubuntu-py3-requirements.sh

apt-get update
apt-get install \
    git libldap2-dev libmagickwand-dev libsasl2-dev \
    libssl-dev libxml2-dev libxslt1-dev libmagic-dev curl gnupg \
    python3-dev python3-pip python3-venv zlib1g-dev libfreetype6-dev libjpeg-dev

curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
sudo apt-get install -y nodejs
```

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

If you want to use `virtualenvwrapper`, you can install it with `sudo pip3 install virtualenvwrapper` and set the `export VIRTUALENV_PYTHON=/usr/bin/python3` in your `~/.bashrc` or `~/.profile` to ensure that `mkvirtualenv` will make a Python 3 virtual environment. For more information on configuring `virtualenvwrapper`, see <https://virtualenvwrapper.readthedocs.io/en/latest/install.html#shell-startup-file>

To activate `virtualenvwrapper`:

For Ubuntu 18.04 with Python 3 (using `pip3` installed `virtualenvwrapper`):

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

Then create the `mytardis` virtual environment

```
mkvirtualenv 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
```

To install minimal Javascript dependencies for production:

```
npm install --production
```

To install Javascript dependencies for production and for testing:

```
npm install && npm test
```

Configuring MyTardis is done through a standard Django `settings.py` file. MyTardis comes with a set of default settings in its `tardis/default_settings/` package. 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.
```

(continues on next page)

(continued from previous page)

```
# 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\"' % ('from .default_settings import * \n\n' if not os.path.
↳ generated from build.sh\n' % ('from .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 manage.py migrate
python manage.py createcachetable default_cache
python manage.py createcachetable celery_lock_cache
```

This avoids creating a superuser before the MyTardis specific UserProfile table has been created. More information about the migrate commands can be found at [Configuration and Administration](#).

Next, create a superuser:

```
python manage.py createsuperuser
```

MyTardis can now be executed in its simplest form using:

```
python manage.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 *tardis package*.

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 manage.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.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, the Creative Commons Attribution 4.0 International licences are loaded.

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 specifying following settings in `settings.py`:

```
LEGAL_TEXT = "A sample legal Text"
```

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 responsible 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 [Apps and Contextual Views](#).

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>
"""

```

(continues on next page)

(continued from previous page)

```

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 manage.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;
}

```

(continues on next page)

(continued from previous page)

```

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 TLSv1.1 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: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;

    ## 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-Proto $scheme;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://mytardis;
        client_max_body_size 4G;
        client_body_buffer_size 8192k;
        proxy_connect_timeout 2000;
    }

```

(continues on next page)

(continued from previous page)

```
        proxy_send_timeout 2000;
        proxy_read_timeout 2000;
    }

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

The X-Forwarded-Proto header is explained in <http://docs.gunicorn.org/en/stable/deploy.html#id5>:

It is recommended to pass protocol information to Gunicorn. Many web frameworks use this information to generate URLs. Without this information, the application may mistakenly generate ‘http’ URLs in ‘https’ responses, leading to mixed content warnings or broken applications.

To tell MyTardis to set this header in its HTTP requests and redirects, you’ll need the following in your `settings.py`:

```
SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTO', 'https')
```

For more information, including warnings on the risks of misconfiguring this setting, see: <https://docs.djangoproject.com/en/2.2/ref/settings/#secure-proxy-ssl-header>

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 manage.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.

Creating Systemd Services for Gunicorn and Celery

Gunicorn is a Python WSGI HTTP Server which is suitable for production (when combined with NGINX). Gunicorn is typically run from a Systemd service (on Ubuntu 16.04 or Ubuntu 18.04), saved in `/etc/systemd/system/gunicorn.service`:

```
[Unit]
Description=gunicorn daemon
After=network.target

[Service]
User=mytardis
Group=mytardis
```

(continues on next page)

(continued from previous page)

```

WorkingDirectory=/home/mytardis/mytardis
ExecStart=/home/mytardis/.virtualenvs/mytardis/bin/gunicorn \
  -c gunicorn_settings.py -b unix:/tmp/gunicorn.socket \
  -b 127.0.0.1:8000 \
  --log-syslog \
  wsgi:application

[Install]
WantedBy=multi-user.target

```

On older systems (Ubuntu 14.04), Supervisor can be used instead of Systemd. In this case, the Gunicorn service would be configured in `/etc/supervisor/conf.d/gunicorn.conf`:

```

[program:gunicorn]
command=/home/mytardis/.virtualenvs/mytardis/bin/gunicorn
  -c /home/mytardis/mytardis/gunicorn_settings.py
  -b unix:/tmp/gunicorn.socket
  -b 127.0.0.1:8000
  --log-syslog
  wsgi:application
user=mytardis
stdout_logfile=/var/log/gunicorn.log
redirect_stderr=true

```

A single server MyTardis deployment requires only one Gunicorn service, but MyTardis can be installed on multiple web nodes, each running NGINX and Gunicorn to accomodate load balancing and high availability using HAProxy.

The Celery workers which run MyTardis asynchronous tasks also require a service configuration, which is typically implemented with Systemd (on Ubuntu 16.04 or Ubuntu 18.04), saved in `/etc/systemd/system/celeryworker.service`:

```

[Unit]
Description=celeryworker daemon
After=network.target

[Service]
User=mytardis
Group=mytardis
WorkingDirectory=/home/mytardis/mytardis
Environment=DJANGO_SETTINGS_MODULE=tardis.settings
ExecStart=/home/mytardis/.virtualenvs/mytardis/bin/celery worker \
  -A tardis.celery.tardis_app \
  -c 2 -Q celery,default -n "allqueues.%%h"

[Install]
WantedBy=multi-user.target

```

On older systems (Ubuntu 14.04), Supervisor can be used instead of Systemd. In this case, the Celery worker service would be configured in `/etc/supervisor/conf.d/celeryworker.conf`:

```

[program:celeryd]
environment=
  DJANGO_SETTINGS_MODULE=tardis.settings
command=/home/mytardis/.virtualenvs/mytardis/bin/celery worker
  -A tardis.celery.tardis_app
  -c 2 -Q celery,default -n "allqueues.%%h"
user=mytardis
directory=/home/mytardis/mytardis

```

(continues on next page)

(continued from previous page)

```
stdout_logfile=/var/log/celeryd.log
redirect_stderr=true
killasgroup=true
stopwaitsecs=600
```

For tasks scheduled by Celerybeat, the Systemd service configuration (for Ubuntu 16.04 or Ubuntu 18.04), is saved in */etc/systemd/system/celerybeat.service*:

```
[Unit]
Description=celerybeat daemon
After=network.target

[Service]
User=mytardis
Group=mytardis
WorkingDirectory=/home/mytardis/mytardis
Environment=DJANGO_SETTINGS_MODULE=tardis.settings
ExecStart=/home/mytardis/.virtualenvs/mytardis/bin/celery beat \
  -A tardis.celery.tardis_app --loglevel INFO

[Install]
WantedBy=multi-user.target
```

On older systems (Ubuntu 14.04), Supervisor can be used instead of Systemd. In this case, the Celerybeat service would be configured in */etc/supervisor/conf.d/celerybeat.conf*:

```
[program:celerybeat]
environment=
  DJANGO_SETTINGS_MODULE=tardis.settings
command=/home/mytardis/.virtualenvs/mytardis/bin/celery beat
  -A tardis.celery.tardis_app --loglevel INFO
user=mytardis
directory=/home/mytardis/mytardis
stdout_logfile=/var/log/celerybeat.log
redirect_stderr=true
```

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 invalid (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

With the `USE_FILTERS` option set to `True` in settings, filters will be called once a file object has been verified.

Filters allow post-processing of uploaded files and can be used to extract metadata from file headers and/or generate thumbnail images.

The `DataFileObject`'s `verify` method submits a task called `"mytardis.apply_filters"` to the message broker (RabbitMQ).

This task can be picked up by the `"mytardis-filters"` microservice: <https://github.com/mytardis/mytardis-filters> or by your own custom microservice.

11.2.4 Database

Initialising

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

Usage

```
python manage.py migrate
```

Migrating

Some of the upgrades to MyTardis will require that the database schema be upgraded to match the internal data model. The `migrate` command 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.

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

Usage

```
python manage.py migrate
```

If the model changes require it, run:

```
python manage.py update_permissions
```

creating superuser

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

Usage

```
python manage.py createsuperuser
```

Backup

Previous versions of MyTardis included a `backupdb` management command but it has been removed in 4.0. Please use the recommended backup tool for your database engine, e.g. `pg_dump` or `mysqldump`.

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

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

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

Selecting “Schemas” in the administrative 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, and Django Elasticsearch DSL library that allows indexing of django models in elasticsearch, 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.

MyTardis comes with a search app that allows indexing Experiments, Dataset and Datafile models and also provides view and api to perform elasticsearch query on these models. This can be enabled by adding 'tardis.apps.search' to the list of installed apps.

Other settings are shown below:

ELASTICSEARCH_DSL Settings

The default value is

```
INSTALLED_APPS += ('django_elasticsearch_dsl', 'tardis.apps.search',)
ELASTICSEARCH_DSL = {
    'default': {
        'hosts': 'http://localhost:9200'
    },
}
ELASTICSEARCH_DSL_INDEX_SETTINGS = {
    'number_of_shards': 1,
    'number_of_replicas': 0
}
```

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`) Elasticsearch DSL 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 instance, 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).

Elasticsearch DSL registers a number of management commands with the Django framework, these commands can be listed by running the following command

```
python manage.py search_index --help
```

the important one here being the `--rebuild` command. To rebuild, navigate to your checkout and call the following command

```
python manage.py search_index --rebuild
```

Elasticsearch DSL 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 instantiation 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.

The django-storages package is now included in MyTardis's requirements-base.txt

It can be used to configure an S3 storage box, or it can be used to store static assets in S3, as described at <https://django-storages.readthedocs.io/en/latest/backends/amazon-S3.html>

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	600
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.

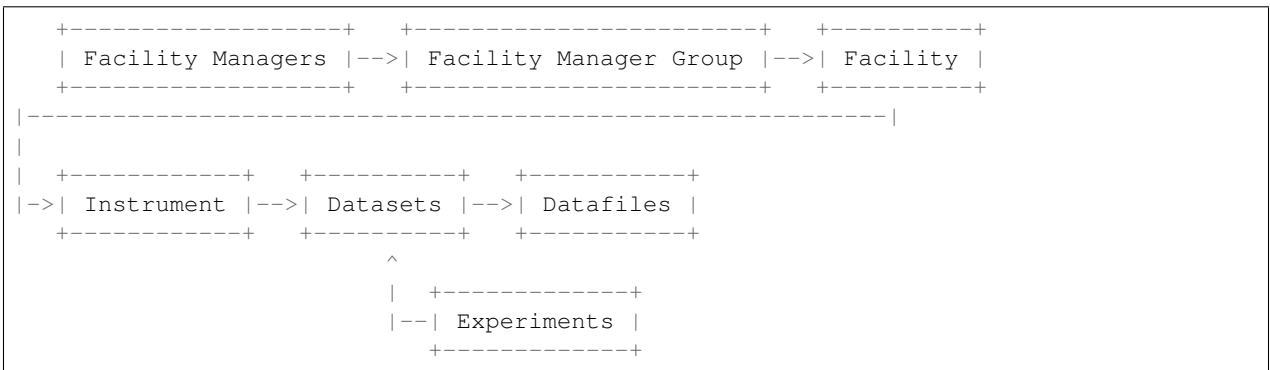
11.2.8 Facility Overview

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.

11.2.9 Task Priorities

Overview

From v4.1.0, MyTardis assumes the use of RabbitMQ as its message broker, and instructs Celery to create a “celery” queue with task priorities enabled. The queue’s maximum priority is set to 10 in `tardis/default_settings/celery_settings.py` which is used when setting up the Celery app in `tardis/celery.py`:

```
tardis/default_settings/celery_settings.py
-----

...
MAX_TASK_PRIORITY = 10
DEFAULT_TASK_PRIORITY = 5
DEFAULT_EMAIL_TASK_PRIORITY = 10

CELERY_DEFAULT_QUEUE = 'celery'
# The 'x-max-priority' argument will only be respected by the RabbitMQ broker,
# which is the recommended broker for MyTardis:
CELERY_QUEUES = (
    Queue('celery', Exchange('celery'),
          routing_key='celery',
          queue_arguments={'x-max-priority': MAX_TASK_PRIORITY}),
)

tardis/celery.py
-----

...
tardis_app = Celery('tardis')
tardis_app.config_from_object('django.conf:settings')
...
```

Celery's `apply_async` method's `shadow` argument is used to annotate storage box related task names with the location (storage box name) which they are running in, e.g. `dfo_verify` becomes `dfo_verify location:default-storage`:

```
tardis/tardis_portal/tasks.py
-----

...
def verify_dfos(**kwargs):
    ...
    for dfo in dfos_to_verify:
        kwargs['priority'] = dfo.priority
        kwargs['shadow'] = 'dfo_verify location:%s' % dfo.storage_box.name
        dfo_verify.apply_async(args=[dfo.id], **kwargs)
    ...
```

Monitoring

We can confirm that the `x-max-priority` argument was supplied when creating the queue by running `rabbitmqctl report` on our RabbitMQ server:

```
$ sudo rabbitmqctl report | grep 'x-max-priority'
...
Listing queues for vhost mytardis ...
name    durable auto_delete    arguments ...
...
celery   true     false    [{"x-max-priority",10}] ...
```

We can list the tasks currently running (and observe their priorities) by running `celery inspect active` from one of our Celery nodes:

```
(mytardis) mytardis@celery0 ~/mytardis$ DJANGO_SETTINGS_MODULE=tardis.settings celery_
↪-A tardis.celery.tardis_app inspect active
↪-> celery@allqueues.celery0: OK
```

(continues on next page)

(continued from previous page)

```

    * {u'args': u'[32933203]', u'time_start': 1548368931.7151582, u'name': u'dfo_
↪verify location:default-storage',
      u'delivery_info': {u'priority': 5, u'redelivered': False, u'routing_key': u
↪'celery', u'exchange': u''},
      u'hostname': u'celery@allqueues.celery0', u'acknowledged': False, u'kwargs': u'
↪{}',
      u'type': u'tardis_portal.dfo.verify', u'id': u'50c80b84-5d64-44c7-a6d4-
↪c551b6d14e22', u'worker_pid': 3730}
-> celery@allqueues.celery2: OK
    - empty -
...
-> celery@allqueues.celery7: OK
    * {u'args': u'[30476811]', u'time_start': 1548368924.2799926, u'name': u'dfo_
↪verify location:default-storage',
      u'delivery_info': {u'priority': 5, u'redelivered': False, u'routing_key': u
↪'celery', u'exchange': u''},
      u'hostname': u'celery@allqueues.celery7', u'acknowledged': False, u'kwargs': u'
↪{}',
      u'type': u'tardis_portal.dfo.verify', u'id': u'de7d0fe1-f386-4937-af7f-
↪a693e7630fb5', u'worker_pid': 9051}
...
-> celery@allqueues.celery14: OK
    * {u'args': u'[289]', u'time_start': 1548368427.9639463, u'name': u'sbox_cache_
↪files location:big-data1',
      u'delivery_info': {u'priority': 4, u'redelivered': False, u'routing_key': u
↪'celery', u'exchange': u''},
      u'hostname': u'celery@allqueues.celery14', u'acknowledged': False, u'kwargs': u
↪'{}',
      u'type': u'tardis_portal.storage_box.cache_files', u'id': u'8767709f-ae98-4735-
↪9d29-21bb5b13e230', u'worker_pid': 15906}

```

Notice that most tasks have the default priority of 5 (defined in `tardis/default_settings/celery_settings.py` as `DEFAULT_TASK_PRIORITY` and that the task operating on a file from the `big-data1` storage boxes has a lower priority of 4.

Default priorities for storage boxes can be configured via `StorageBox` Attributes. We can check which storage boxes have priorities specified in the Django shell as follows:

```

(myrtardis) myrtardis@celery0:~/myrtardis$ ./manage.py shell_plus

>>> StorageBoxAttribute.objects.filter(key='priority')
<QuerySet [<StorageBoxAttribute: big-data1-> priority: 4>, <StorageBoxAttribute: big-
↪data2-> priority: 4>]>

```

So in the example above, explicit priorities are only set for the “big-data1” and “big-data2” storage boxes.

Common Problems and Solutions

Celery workers time out when running `celery inspect active`

When running `celery inspect active` sometimes you will see this error:

```
Error: No nodes replied within time constraint.
```

Usually, running `celery inspect active` again will resolve the problem, i.e. it will just work without error on subsequent attempts.

If desired, you can specify a timeout e.g. `timeout 10` but usually running `celery inspect active` again does the trick.

Non-priority queue already exists

If an attempt to submit a task to the queue (with `apply_async` triggers an error like this:

```
PreconditionFailed: Queue.declare: (406) PRECONDITION_FAILED - inequivalent arg 'x-
↪max-priority' for queue 'celery' in vhost '/':
received the value '10' of type 'signedint' but current is none
```

This means that the MyTardis process attempting to submit the task is expecting the queue to have the `x-max-priority` argument, but it doesn't have that argument (see `rabbitmqctl` report above).

In this case, you can delete the `celery` queue, and allow Celery to recreate it with the `x-max-priority` argument:

```
(mytardis) mytardis@celery0 ~/mytardis$ DJANGO_SETTINGS_MODULE=tardis.settings celery_
↪-A tardis.celery.tardis_app amqp queue.delete celery
```

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 partitioning. 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.

Persistence 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 Framework*.

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 fine 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,
                        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"
        },
        {
          "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": [
```

(continues on next page)

(continued from previous page)

```

    "/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": [{

```

(continues on next page)

(continued from previous page)

```
    "schema": "http://datafiles.shop.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://datafiles.shop.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://datafiles.shop.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 specify 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 (Behaviour Driven Development) tests with the default settings and headless Chrome, download ChromeDriver from <http://chromedriver.chromium.org/downloads> and make it available in your PATH (e.g. in `/usr/local/bin/`) and run:

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

Running QUnit tests

The QUnit tests reside in the `js_tests/` directory.

`package.json` contains `devDependencies` required for running these tests.

Running `npm install` will install everything you need, whereas `npm install --production` will skip the `devDependencies`.

You can run the QUnit tests with:

```
npm test
```

Or by running a web server:

```
python -m SimpleHTTPServer
```

and opening http://127.0.0.1:8000/js_tests/tests.html in your browser.

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 analytics 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.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.filepicker package

Submodules

tardis.apps.filepicker.filepicker_settings module

tardis.apps.filepicker.urls module

tardis.apps.filepicker.utils module

taken from https://raw.githubusercontent.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 = re.compile('https?:\\/?\\/?www.filepicker.io\\/?api\\/?file\\/?.*')

    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.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

Raises **NotImplementedError** – not implemented

tardis.apps.oaipmh.provider.experiment module

class `tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider` (*site*)

Bases: `tardis.apps.oaipmh.provider.base.BaseProvider`

NS_CC = 'http://www.tardis.edu.au/schemas/creative_commons/2011/05/17'

getRecord (*metadataPrefix*, *identifier*)

Return record if we handle it.

static get_id (*obj*)

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

Return identifiers in range, provided we handle this metadata prefix.

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

Return records in range, provided we handle this metadata prefix.

listSets ()

No support for sets.

class `tardis.apps.oaipmh.provider.experiment.DcExperimentProvider` (*site*)

Bases: `tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider`

listMetadataFormats (*identifier=None*)

Return metadata format if no identifier, or identifier is a valid experiment.

```
class tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider (site)
    Bases: tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider

    class ExperimentWriter (root, metadata, site)
        Bases: object

        write ()

        writeRegistryObjectsWrapper ()

        writeRelatedAuthor (element, obj, relation)

        writeRelatedInfo (element, obj)

        writeRelatedUser (element, obj, relation)

        writeSubject (element, obj)

    static get_rifcs_id (id_, site_=None)

    listMetadataFormats (identifier=None)
        Return metadata format if no identifier, or identifier is a valid experiment.

    static writeExperimentMetadata (element, metadata, site=None, writer=None)
        Wrapper around experiment writer.

    static writeUserMetadata (element, metadata, site=None)
```

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

class tardis.apps.oaipmh.tests.provider.test_experiment.**AbstractExperimentProviderTC** (methodNames: `testGetRecord`, `testListRecords`, `testListSets`)
Bases: `object`

setUp ()

tearDown ()

testGetRecordHandlesInvalidIdentifiers ()

testIdentify ()

There can be only one provider that responds. This one does not.

testListIdentifiers ()

testListIdentifiersDoesNotHandleSets ()

testListMetadataFormats ()

testListSets ()

class tardis.apps.oaipmh.tests.provider.test_experiment.**DcExperimentProviderTestCase** (methodName: `testGetRecord`, `testListRecords`, `testListSets`)
Bases: `tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTC`, `django.test.testcases.TestCase`

testGetRecord ()

testListRecords ()

class tardis.apps.oaipmh.tests.provider.test_experiment.**RifCsExperimentProviderTestCase** (methodName: `testGetRecord`, `testListRecords`, `testListSets`)
Bases: `tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTC`, `django.test.testcases.TestCase`

tearDown ()

Hook method for deconstructing the test fixture after testing it.

testGetRecord ()

testListRecords ()

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 currently 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.openid_migration package

Subpackages

tardis.apps.openid_migration.migrations package

Submodules

tardis.apps.openid_migration.migrations.0001_initial module

```
class tardis.apps.openid_migration.migrations.0001_initial.Migration(name,  
                                                                    app_label)  
    Bases: django.db.migrations.migration.Migration  
  
    dependencies = [('tardis_portal', '0012_userauthentication_approved'), ('auth', '__first__')]  
    initial = True  
  
    operations = [CreateModel name='OpenidACLMigration', fields=[('id', <django.db.models
```

Module contents

tardis.apps.openid_migration.tests package

Submodules

tardis.apps.openid_migration.tests.test_forms module

Tests related to OpenID migration forms

```
class tardis.apps.openid_migration.tests.test_forms.OpenIDMigrationFormTestCase (methodName=''):
    Bases: django.test.testcases.TestCase

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

    test_init ()
```

tardis.apps.openid_migration.tests.test_migration module

Tests related to OpenID migration


```
class tardis.apps.openid_migration.tests.test_migration.OpenIDMigrationTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

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

    test_do_migration ()
```

tardis.apps.openid_migration.tests.test_models module

```
class tardis.apps.openid_migration.tests.test_models.ModelTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    create_OpenidUserMigration ()

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

    test_OpenidUserMigration_creation ()
```

tardis.apps.openid_migration.tests.test_views module

Tests related to OpenID migration views

```
class tardis.apps.openid_migration.tests.test_views.OpenIDMigrationViewTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

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

    test_migrate_accounts (mock_webpack_get_bundle)
```

Module contents

Submodules

tardis.apps.openid_migration.apps module

```
class tardis.apps.openid_migration.apps.OpenidMigrationConfig (app_name,
                                                             app_module)
    Bases: tardis.app_config.AbstractTardisAppConfig

    name = 'tardis.apps.openid_migration'

    verbose_name = 'OpenID migrations'
```

tardis.apps.openid_migration.context_processors module

```
tardis.apps.openid_migration.context_processors.openid_migration_processor (request)
    adds context for openid_migration
```

tardis.apps.openid_migration.default_settings module

tardis.apps.openid_migration.email_text module

```
tardis.apps.openid_migration.email_text.email_migration_success (user,
                                                                new_username,
                                                                auth_method)

tardis.apps.openid_migration.email_text.interpolate_template (template_name,
                                                             **kwargs)
```

tardis.apps.openid_migration.forms module

```
tardis.apps.openid_migration.forms.openid_user_migration_form()
    Create a user migration form with username and password field.
```

tardis.apps.openid_migration.migration module

```
tardis.apps.openid_migration.migration.acl_migration (userIdToBeReplaced,
                                                       replacementUserId,
                                                       user_migration_record)

tardis.apps.openid_migration.migration.confirm_migration (request)

tardis.apps.openid_migration.migration.do_migration (request)
    Migrating account from the account that the logged in user has provided in the Authentication Form. Migration
    involve relinking the UserAuthentication table entries, transferring ObjectACL entries to the migrated account,
    changing the Group memberships and making the old account inactive.
```

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

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

Return type HttpResponse

```
tardis.apps.openid_migration.migration.getSupportedAuthMethods ()
    Return the list of authentication methods.

tardis.apps.openid_migration.migration.get_api_key (user)

tardis.apps.openid_migration.migration.get_matching_auth_provider (backend)

tardis.apps.openid_migration.migration.migrate_api_key (old_user, new_user)

tardis.apps.openid_migration.migration.migrate_user_permissions (old_user,
                                                                new_user)

tardis.apps.openid_migration.migration.openid_migration_method (request)
```

tardis.apps.openid_migration.models module

```
class tardis.apps.openid_migration.models.OpenidACLMigration (id, user_migration,
                                                            acl_id)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
```

exception MultipleObjectsReturnedBases: `django.core.exceptions.MultipleObjectsReturned`**acl_id**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

acl_id_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

user_migration

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

user_migration_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.apps.openid_migration.models.OpenidACLMigrationAdmin(model, admin_site)
```

Bases: `django.contrib.admin.options.ModelAdmin`

list_display = `['user_migration_obj', 'acl_id']`

media

user_migration_obj (*obj*)

```
class tardis.apps.openid_migration.models.OpenidUserMigration(id, old_user,
                                                             old_user_auth_method,
                                                             new_user,
                                                             new_user_auth_method,
                                                             migration_timestamp,
                                                             migration_status)
```

Bases: `django.db.models.base.Model`

exception DoesNotExistBases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

get_next_by_migration_timestamp (*, *field*=<`django.db.models.fields.DateTimeField: migration_timestamp`>, *is_next*=True, **kwargs)

get_previous_by_migration_timestamp (*, *field*=<`django.db.models.fields.DateTimeField: migration_timestamp`>, *is_next*=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

migration_status

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

migration_timestamp

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

new_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

new_user_auth_method

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

new_user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

old_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

old_user_auth_method

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

old_user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

openidaclmigration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
class tardis.apps.openid_migration.models.OpenidUserMigrationAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ['__str__', 'old_user', 'old_user_auth_method', 'new_user_id', 'new_user_auth_method', 'media']
```

tardis.apps.openid_migration.tasks module

tardis.apps.openid_migration.urls module

tardis.apps.openid_migration.user_menu_modifiers module

```
tardis.apps.openid_migration.user_menu_modifiers.add_migrate_account_menu_item(request, user_menu)
```

Add a 'Migrate My Account' item to the user menu

Parameters

- **request** (`django.http.HttpRequest`) – an HTTP Request instance
- **user_menu** (`list`) – user menu context to modify

Returns user_menu list

Return type list

tardis.apps.openid_migration.utils module

```
tardis.apps.openid_migration.utils.rollback_migration(user_migration_obj)
```

tardis.apps.openid_migration.views module

```
tardis.apps.openid_migration.views.migrate_accounts(request)
```

Manage user migration using AJAX.

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 = [('auth', '__first__'), ('auth', '0006_require_contenttypes_0002')]
    operations = [<CreateModel name='Credential', fields=[('id', <django.db.models.fields.
```

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 = [('push_to', '0001_initial')]
    operations = [<AlterField model_name='credential', name='key_type', field=<django.db.m
```

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_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: `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)`

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

password

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

remote_hosts

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

remote_user

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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='id_%s',      prefix=None,
                                                  initial=None,         error_class=<class
                                                  'django.forms.utils.ErrorList'>,
                                                  label_suffix=None,
                                                  empty_permitted=False,
                                                  instance=None,
                                                  use_required_attribute=None,   ren-
                                                  derer=None)
```

Bases: django.forms.models.ModelForm

class Meta

Bases: object

fields = '__all__'

model

alias of *Credential*

widgets = {'password': <django.forms.widgets.PasswordInput object>}

base_fields = {'key_type': <django.forms.fields.CharField object>, 'password': <djan

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
```

```
    Bases: object
```

```
    abstract = False
```

```
key
```

Returns a subclass of PKey of the appropriate key type

Return type PKey

Raises `ValidationError` –

```
key_type
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
private_key
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
public_key
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

```
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
```

```
allow_for_all
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
allowed_groups
```

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

allowed_remote_hosts

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

allowed_users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

cert_signing_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nickname

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

oauth_authorize_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

oauth_check_token_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

oauth_client_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

oauth_client_secret

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

oauth_token_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

administrator_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

credential_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

host_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

logo_img

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nickname

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

oauthsshcertsigningservice_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

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

port

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
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.tasks.do_file_copy` (*credential_id*, *remote_host_id*, *datafile_map*,
base_dir=None)
`tardis.apps.push_to.tasks.make_dirs` (*sftp_client*, *dir_list*)
`tardis.apps.push_to.tasks.notify_user` (*user_id*, *remote_host_id*, *success=True*)

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

`tardis.apps.push_to.views.authorize_remote_access` (*request*, *remote_host_id*, *service_id=None*)
 Generates an SSH certificate using an OAuth2 SSH signing service :param Request request: request object
 :param basestring remote_host_id: remote host id :param basestring service_id: OAuth2 SSH certificate signing
 service id :return: an error message or OAuth2 redirects :rtype: HttpResponseRedirect

`tardis.apps.push_to.views.get_accessible_hosts` (*request*, *obj_type=None*,
push_obj_id=None)
 Retrieves all accessible hosts (i.e. hosts for which the user already has credentials for) including push-to trigger
 URLs if the object type and id are supplied :param Request request: request object :param object obj_type: data
 type to be copied

(experiment, dataset or datafile)

Parameters `push_obj_id` (*int*) – the database object id

Returns json object with accessible hosts

Return type HttpResponseRedirect

`tardis.apps.push_to.views.get_credential` (*request*, *remote_host*)
 Fetches a suitable credential for the remote host, or raises an exception if none found :param Request request:
 request object :param RemoteHost remote_host: the RemoteHost for which a credential

should be found

Returns the credential

Return type `object`

Raises `NoSuitableCredential` – raised when no credential is found

`tardis.apps.push_to.views.get_push_url_for_host(remote_host, obj_type, push_obj_id)`

Constructs a push-to URL to trigger data transfer :param RemoteHost remote_host: the RemoteHost to which data should be copied :param obj_type: data type to be copied (experiment, dataset or datafile) :type obj_type: object :param int push_obj_id: the database object id :return: a push-to URL :rtype: basestring

`tardis.apps.push_to.views.get_signing_services(request, obj_type=None, push_obj_id=None)`

Retrieves all certificate signing services and associated hosts including push-to trigger URLs if the object type and id are supplied :param Request request: request object :param class obj_type: data type to be copied (experiment, dataset or datafile) :param int push_obj_id: the database object id :return: json object with signing services and hosts :rtype: HttpResponse

`tardis.apps.push_to.views.oauth_callback(request)`

OAuth2 callback endpoint to continue the SSH certificate signing process :param Request request: request object :return: error message or redirect to the signing service with access token :rtype: HttpResponse

`tardis.apps.push_to.views.oauth_callback_url(request)`

Builds the oauth callback URL :param Request request: request object :return: callback URL :rtype: basestring

`tardis.apps.push_to.views.render_error_message(request, message, status=500)`

`tardis.apps.push_to.views.render_success_message(request, message, status=200)`

`tardis.apps.push_to.views.validate_remote_path(request, remote_host_id)`

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 (mock_webpack_get_bundle)

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 (mock_webpack_get_bundle)

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 (mock_webpack_get_bundle)

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 (mock_webpack_get_bundle)

    testAccessWithoutReadPerms (mock_webpack_get_bundle)

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 (mock_webpack_get_bundle)

```

Module contents

Submodules

tardis.apps.related_info.forms module

```
class tardis.apps.related_info.forms.RelatedInfoForm (data=None, files=None,  
                                                    auto_id='id_%s', pre-  
                                                    fix=None, initial=None,  
                                                    error_class=<class  
                                                    'django.forms.utils.ErrorList'>,  
                                                    label_suffix=None,  
                                                    empty_permitted=False,  
                                                    field_order=None,  
                                                    use_required_attribute=None,  
                                                    renderer=None)  
  
    Bases: django.forms.forms.Form  
  
    base_fields = {'identifier': <django.forms.fields.URLField object>, 'notes': <django  
    declared_fields = {'identifier': <django.forms.fields.URLField object>, 'notes': <dj  
    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.s3utils package

Subpackages

tardis.apps.s3utils.tests package

Submodules

tardis.apps.s3utils.tests.test_api module

Testing the s3util app's extensions to the tastypie-based mytardis api

```
class tardis.apps.s3utils.tests.test_api.S3UtilsAppApiTestCase (methodName='runTest')  
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase  
  
    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_dfo()

Test downloading a DataFileObject using the s3util app's extensions to the MyTardis REST API.

tardis.apps.s3utils.tests.test_app_config module

class tardis.apps.s3utils.tests.test_app_config.S3UtilsConfigTestCase (*methodName='runTest'*)

Bases: django.test.testcases.TestCase

test_app_config()

tardis.apps.s3utils.tests.test_checksums module

Testing the s3util app's ability to calculate checksums for S3 objects

class tardis.apps.s3utils.tests.test_checksums.S3UtilsAppChecksumsTestCase (*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_checksums()

Ensure that we can calculate an MD5 sum and a SHA512 sum for a file in S3 object storage

Module contents

Submodules

tardis.apps.s3utils.api module

Additions to MyTardis's REST API

class tardis.apps.s3utils.api.ReplicaAppResource (*api_name=None*)

Bases: *tardis.tardis_portal.api.ReplicaResource*

Extends MyTardis's API for DFOs, adding in a download method for S3 objects

class Meta

Bases: *tardis.tardis_portal.api.Meta*

authorization

Authorisation class for Tastypie.

queryset

resource_name = 'replica'

base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'datafile': <

declared_fields = {}

download_dfo (*request, **kwargs*)

Download DataFileObject from S3 Object Store

prepend_urls()

A hook for adding your own URLs or matching before the default URLs.

tardis.apps.s3utils.apps module

```
class tardis.apps.s3utils.apps.S3UtilsConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig

    name = 's3utils'
```

tardis.apps.s3utils.default_settings module

Default settings for s3utils app

```
tardis.apps.s3utils.default_settings.S3_SIGNED_URL_EXPIRY = 30
```

A short expiry (30 seconds) is used, because it is only intended to provide access long enough for an authenticated MyTardis user to be redirected to the signed URL.

tardis.apps.s3utils.urls module

Minimal urls.py, so we can do a reverse lookup for the 's3_api_download_dfo' URL pattern.

's3_api_download_dfo' is defined in the prepend_urls method of the ReplicaAppResource class in api.py

The API endpoint defined in this app is mapped to a URL in tardis/urls/api.py (along with API endpoints defined by other MyTardis apps).

tardis.apps.s3utils.utils module

Utilities for S3 objects

```
tardis.apps.s3utils.utils.calculate_checksums(dfo, compute_md5=True, compute_sha512=False)
```

Calculates checksums for an S3 DataFileObject instance. For files in S3, using the django-storages abstraction is inefficient - we end up with a clash of chunking algorithms between the download from S3 and MyTardis's Python-based checksum calculation. So for S3 files, we calculate checksums using external binaries (md5sum and shasum) instead.

:param dfo : The DataFileObject instance :type dfo: DataFileObject :param compute_md5: whether to compute md5 default=True :type compute_md5: bool :param compute_sha512: whether to compute sha512, default=True :type compute_sha512: bool

Returns the checksums as {'md5sum': result, 'sha512sum': result}

Return type dict

```
tardis.apps.s3utils.utils.generate_presigned_url(dfo, expiry=None)
```

Generate a presigned URL for an S3 object

boto3 must be installed if you are using the storages.backends.s3boto3.S3Boto3Storage storage class, defined in the django-storages package

dfo [DataFileObject] The DataFileObject to generate the pre-signed URL for

expiry [int] The signed URL's expiry in seconds

string The pre-signed URL

Module contents

tardis.apps.search package

Subpackages

tardis.apps.search.tests package

Submodules

tardis.apps.search.tests.test_api module

```
class tardis.apps.search.tests.test_api.SimpleSearchTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

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

    test_advance_search_authenticated_user()

    test_advance_search_unauthenticated_user()

    test_simple_search_authenticated_user()

    test_simple_search_unauthenticated_user()
```

tardis.apps.search.tests.test_index module

```
class tardis.apps.search.tests.test_index.IndexExperimentTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

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

    test_create_index()
```

Module contents

Submodules

tardis.apps.search.api module

RESTful API for MyTardis search. Implemented with Tastypie.

```
class tardis.apps.search.api.AdvanceSearchAppResource (api_name=None)
    Bases: tastypie.resources.Resource

    class Meta
        Bases: object

        always_return_data = True
```

```
authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>
list_allowed_methods = ['post']
object_class
    alias of SearchObject
resource_name = 'advance-search'
serializer = <tardis.apps.search.api.PrettyJSONSerializer object>
base_fields = {'hits': <tastypie.fields.ApiField object>, 'resource_uri': <tastypie.
declared_fields = {'hits': <tastypie.fields.ApiField object>}
```

detail_uri_kwargs (*bundle_or_obj*)

Given a *Bundle* or an object (typically a *Model* instance), it returns the extra kwargs needed to generate a detail URI.

By default, it uses this resource's *detail_uri_name* in order to create the URI.

get_object_list (*request*)

A hook to allow making returning the list of available objects.

This needs to be implemented at the user level.

ModelResource includes a full working version specific to Django's *Models*.

obj_create (*bundle, **kwargs*)

Creates a new object based on the provided data.

This needs to be implemented at the user level.

ModelResource includes a full working version specific to Django's *Models*.

obj_get_list (*bundle, **kwargs*)

Fetches the list of objects available on the resource.

This needs to be implemented at the user level.

ModelResource includes a full working version specific to Django's *Models*.

```
class tardis.apps.search.api.PrettyJSONSerializer (formats=None,                con-
                                                    tent_types=None,            date-
                                                    time_formatting=None)

Bases: tastypie.serializers.Serializer

json_indent = 2

to_json (data, options=None)
    Given some Python data, produces JSON output.
```

```
class tardis.apps.search.api.SearchAppResource (api_name=None)
Bases: tastypie.resources.Resource

Tastypie resource for simple-search

class Meta
    Bases: object

    always_return_data = True

    authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>

    list_allowed_methods = ['get']

    object_class
        alias of SearchObject
```

```

    resource_name = 'simple-search'

    serializer = <tardis.apps.search.api.PrettyJSONSerializer object>

    base_fields = {'hits': <tastypie.fields.ApiField object>, 'resource_uri': <tastypie.
    declared_fields = {'hits': <tastypie.fields.ApiField object>}

    detail_uri_kwargs (bundle_or_obj)
        Given a Bundle or an object (typically a Model instance), it returns the extra kwargs needed to generate
        a detail URI.

        By default, it uses this resource's detail_uri_name in order to create the URI.

    get_object_list (request)
        A hook to allow making returning the list of available objects.

        This needs to be implemented at the user level.

        ModelResource includes a full working version specific to Django's Models.

    obj_get_list (bundle, **kwargs)
        Fetches the list of objects available on the resource.

        This needs to be implemented at the user level.

        ModelResource includes a full working version specific to Django's Models.

class tardis.apps.search.api.SearchObject (hits=None, id=None)
    Bases: object

tardis.apps.search.api.simple_search_public_data (query_text)

```

tardis.apps.search.apps module

```

class tardis.apps.search.apps.SearchConfig (app_name, app_module)
    Bases: tardis.app_config.AbstractTardisAppConfig

    name = 'tardis.apps.search'

    verbose_name = 'Search'

```

tardis.apps.search.documents module

```

class tardis.apps.search.documents.DataFileDocument (related_instance_to_ignore=None,
                                                    **kwargs)
    Bases: django_elasticsearch_dsl.documents.DocType

    class Django
        Bases: object

        model
            alias of tardis.tardis_portal.models.datafile.DataFile

        queryset_pagination = 5000

        related_models = [<class 'tardis.tardis_portal.models.dataset.Dataset'>, <class 'ta

class Index
    Bases: object

    name = 'datafile'

```

```
        settings = {'number_of_replicas': 0, 'number_of_shards': 1}
django = {'model': <class 'tardis.tardis_portal.models.datafile.DataF...}
get_instances_from_related(related_instance)
get_queryset()
    Return the queryset that should be indexed by this doc type.
parallel_bulk(actions, **kwargs)
prepare_experiments(instance)
class tardis.apps.search.documents.DatasetDocument(related_instance_to_ignore=None,
                                                    **kwargs)
    Bases: django_elasticsearch_dsl.documents.DocType
    class Django
        Bases: object
        model
            alias of tardis.tardis_portal.models.dataset.Dataset
        related_models = [<class 'tardis.tardis_portal.models.experiment.Experiment'>, <cla
    class Index
        Bases: object
        name = 'dataset'
        settings = {'number_of_replicas': 0, 'number_of_shards': 1}
        django = {'model': <class 'tardis.tardis_portal.models.dataset.Datase...}
        get_instances_from_related(related_instance)
        parallel_bulk(actions, **kwargs)
class tardis.apps.search.documents.ExperimentDocument(related_instance_to_ignore=None,
                                                        **kwargs)
    Bases: django_elasticsearch_dsl.documents.DocType
    class Django
        Bases: object
        model
            alias of tardis.tardis_portal.models.experiment.Experiment
        related_models = [<class 'django.contrib.auth.models.User'>, <class 'tardis.tardis
    class Index
        Bases: object
        name = 'experiments'
        settings = {'number_of_replicas': 0, 'number_of_shards': 1}
        django = {'model': <class 'tardis.tardis_portal.models.experiment.Exp...}
        get_instances_from_related(related_instance)
        parallel_bulk(actions, **kwargs)
```

tardis.apps.search.urls module

tardis.apps.search.views module

views relevant to search

```
class tardis.apps.search.views.SearchView(**kwargs)
    Bases: django.views.generic.base.TemplateView
    template_name = 'search.html'
```

Module contents

tardis.apps.sftp package

Subpackages

tardis.apps.sftp.management package

Subpackages

tardis.apps.sftp.management.commands package

Submodules

tardis.apps.sftp.management.commands.sftpd module

```
class tardis.apps.sftp.management.commands.sftpd.Command(stdout=None,
                                                         stderr=None,
                                                         no_color=False,
                                                         force_color=False)
    Bases: django.core.management.base.BaseCommand
    add_arguments(parser)
        Entry point for subclassed commands to add custom arguments.
    handle(*args, **options)
        The actual logic of the command. Subclasses must implement this method.
```

Module contents

Module contents

tardis.apps.sftp.migrations package

Submodules

tardis.apps.sftp.migrations.0001_initial module

```
class tardis.apps.sftp.migrations.0001_initial.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('tardis_portal', '0011_auto_20160505_1643'), ('auth', '__first__')]
    initial = True
    operations = [<CreateModel name='SFTPPublicKey', fields=[('id', <django.db.models.fields
```

Module contents

tardis.apps.sftp.tests package

Submodules

tardis.apps.sftp.tests.test_sftp module

```
class tardis.apps.sftp.tests.test_sftp.SFTPDManagementTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    testSFTPDWithoutHostKey ()
        Attempting to start the SFTPD service without a host key should raise an SSHException

class tardis.apps.sftp.tests.test_sftp.SFTPTTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

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

    test_cyberduck_connection_window ()

    test_sftp ()

    test_sftp_dynamic_docs_dataset (mock_webpack_get_bundle)

    test_sftp_dynamic_docs_experiment (mock_webpack_get_bundle)

    test_sftp_key_connect ()
```

Module contents

Submodules

tardis.apps.sftp.admin module

tardis.apps.sftp.api module

```
class tardis.apps.sftp.api.SFTPAclAuthorization
    Bases: tastypie.authorization.Authorization

    create_detail (object_list, bundle)
        Returns either True if the user is allowed to create the object in question or throw Unauthorized if they are not.
```


Returns `True` by default.

delete_detail (*object_list*, *bundle*)

Returns either `True` if the user is allowed to delete the object in question or throw `Unauthorized` if they are not.

Returns `True` by default.

read_detail (*object_list*, *bundle*)

Returns either `True` if the user is allowed to read the object in question or throw `Unauthorized` if they are not.

Returns `True` by default.

read_list (*object_list*, *bundle*)

Returns a list of all the objects a user is allowed to read.

Should return an empty list if none are allowed.

Returns the entire list by default.

class `tardis.apps.sftp.api.SFTPPublicKeyAppResource` (*api_name=None*)

Bases: `tastypie.resources.ModelResource`

Tastypie model resource for `SFTPPublicKey` model

class `Meta`

Bases: `object`

`authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>`

`authorization`

`detail_allowed_methods = ['get', 'delete']`

`filtering = {'id': ('exact',), 'name': ('exact',)}`

`list_allowed_methods = ['get', 'post']`

`object_class`

alias of `tardis.apps.sftp.models.SFTPPublicKey`

`queryset`

`resource_name = 'publickey'`

`validation = <tastypie.validation.FormValidation object>`

`base_fields = {'added': <tastypie.fields.DateField object>, 'id': <tastypie.fields.IntegerField object>}`

`declared_fields = {}`

dehydrate (*bundle*)

A hook to allow a final manipulation of data once all fields/methods have built out the dehydrated data.

Useful if you need to access more than one dehydrated field or want to annotate on additional data.

Must return the modified bundle.

hydrate (*bundle*)

A hook to allow an initial manipulation of data before all methods/fields have built out the hydrated data.

Useful if you need to access more than one hydrated field or want to annotate on additional data.

Must return the modified bundle.

tardis.apps.sftp.apps module

```
class tardis.apps.sftp.apps.SFTPConfig(app_name, app_module)
    Bases: tardis.app_config.AbstractTardisAppConfig

    name = 'tardis.apps.sftp'

    verbose_name = 'SFTP'
```

tardis.apps.sftp.default_settings module

```
tardis.apps.sftp.default_settings.REQUIRE_SSL_TO_GENERATE_KEY = True
    Require a secure connection (i.e., HTTPS) to allow key generation.

tardis.apps.sftp.default_settings.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.apps.sftp.forms module

```
class tardis.apps.sftp.forms.KeyAddForm(data=None, files=None, auto_id='id_%s', pre-
                                         fix=None, initial=None, error_class=<class
                                         'django.forms.utils.ErrorList'>, la-
                                         bel_suffix=None, empty_permitted=False,
                                         field_order=None, use_required_attribute=None,
                                         renderer=None)

    Bases: django.forms.forms.Form

    base_fields = {'key_type': <django.forms.fields.CharField object>, 'name': <django.f

    clean()
        Hook for doing any extra form-wide cleaning after Field.clean() has been called on every field. Any
        ValidationError raised by this method will not be associated with a particular field; it will have a special-
        case association with the field named '__all__'.

    declared_fields = {'key_type': <django.forms.fields.CharField object>, 'name': <djan

    media

class tardis.apps.sftp.forms.KeyGenerateForm(data=None, files=None,
                                              auto_id='id_%s', prefix=None, ini-
                                              tial=None, error_class=<class
                                              'django.forms.utils.ErrorList'>,
                                              label_suffix=None,
                                              empty_permitted=False, field_order=None,
                                              use_required_attribute=None, ren-
                                              derer=None)

    Bases: django.forms.forms.Form

    base_fields = {'name': <django.forms.fields.CharField object>}

    declared_fields = {'name': <django.forms.fields.CharField object>}

    media
```

tardis.apps.sftp.models module

class tardis.apps.sftp.models.**SFTPPublicKey** (*args, **kwargs)

Bases: django.db.models.base.Model

Model for associated SFTP public keys with users

Parameters

- **user** (*ForeignKey for User*) – user who owns this public key
- **name** (*string*) – name for this public key
- **public_key** (*string*) – OpenSSH formatted public key
- **added** (*date*) – date the public key was added (Optional)

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

added

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_added (*, field=<django.db.models.fields.DateField: added>, is_next=True, **kwargs)

get_previous_by_added (*, field=<django.db.models.fields.DateField: added>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

key_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

public_key

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

tardis.apps.sftp.sftp module

SFTP Server

```
class tardis.apps.sftp.sftp.DynamicTree (host_obj=None)
```

Bases: `object`

add_child (*name, obj=None*)

add_path (*path*)

add_path_elems (*elems*)

clear_children ()

get_leaf (*path, update=False*)

update_all_files ()

update_dataset_files ()

update_datasets ()

update_experiments ()

update_nothing ()

```
class tardis.apps.sftp.sftp.MyTSFTPHandle (df, flags=0, optional_args=None)
```

Bases: `paramiko.sftp_handle.SFTPHandle`

SFTP File Handle

stat ()

Return an `L{SFTPAttributes}` object referring to this open file, or an error code. This is equivalent to `L{SFTPServerInterface.stat}`, except it's called on an open file instead of a path.

@return: an attributes object for the given file, or an SFTP error code (like `L{SFTP_PERMISSION_DENIED}`). @rtype: `L{SFTPAttributes}` I{or error code}

```
class tardis.apps.sftp.sftp.MyTSFTPRequestHandler (request, client_address, server)
```

Bases: `socketserver.BaseRequestHandler`

auth_timeout = 60

handle ()

handle_timeout ()

setup ()

timeout = 60

```
class tardis.apps.sftp.sftp.MyTSFTPServer (*args, **kwargs)
```

Bases: `paramiko.sftp_server.SFTPServer`

override SFTPServer to provide channel information to the SFTP subsystem

```
class tardis.apps.sftp.sftp.MyTSFTPServerInterface (server, *args, **kwargs)
```

Bases: `paramiko.sftp_si.SFTPServerInterface`

MyTardis data via SFTP

canonicalize (*path*)

Return the canonical form of a path on the server.

experiments**list_folder** (*path*)

Returns a list of files within a given folder. The C{path} will use posix notation (C{" / " separates folder names) and may be an absolute or relative path.

The list of files is expected to be a list of L{SFTPAttributes} objects, which are similar in structure to the objects returned by C{os.stat}. In addition, each object should have its C{filename} field filled in, since this is important to a directory listing and not normally present in C{os.stat} results.

In case of an error, you should return one of the C{SFTP_*} error codes, such as L{SFTP_PERMISSION_DENIED}.

@param path: the requested path (relative or absolute) to be listed. @type path: str @return: a list of the files in the given folder, using L{SFTPAttributes} objects. @rtype: list of L{SFTPAttributes} I{or error code}

lstat (*path*)

symbolic links are not supported

open (*path, flags, attr*)

Open a file on the server and create a handle for future operations on that file. On success, a new object subclassed from L{SFTPHandle} should be returned. This handle will be used for future operations on the file (read, write, etc). On failure, an error code such as L{SFTP_PERMISSION_DENIED} should be returned.

C{flags} contains the requested mode for opening (read-only, write-append, etc) as a bitset of flags from the C{os} module:

- C{os.O_RDONLY}
- C{os.O_WRONLY}
- C{os.O_RDWR}
- C{os.O_APPEND}
- C{os.O_CREAT}
- C{os.O_TRUNC}
- C{os.O_EXCL}

(One of C{os.O_RDONLY}, C{os.O_WRONLY}, or C{os.O_RDWR} will always be set.)

The C{attr} object contains requested attributes of the file if it has to be created. Some or all attribute fields may be missing if the client didn't specify them.

@note: The SFTP protocol defines all files to be in "binary" mode. There is no equivalent to python's "text" mode.

Parameters

- **path** (*basestring*) – the requested datafile path
- **flags** (*int*) – flags or'd together from the C{os} module indicating the requested mode for opening the file.
- **attr** (*SFTPAttributes*) – requested attributes of the file if it is newly created.

Returns a new L{SFTPHandle} I{or error code}.

Return type SFTPHandle

session_ended()

run cleanup on exceptions or disconnection. idea: collect stats and store them in this function

session_started()

run on connection initialisation

stat(path)

Return an `L{SFTPAttributes}` object for a path on the server, or an error code. If your server supports symbolic links (also known as “aliases”), you should follow them. (`L{lstat}` is the corresponding call that doesn’t follow symlinks/aliases.)

@param path: the requested path (relative or absolute) to fetch file statistics for. @type path: str

@return: an attributes object for the given file, or an SFTP error code (like `L{SFTP_PERMISSION_DENIED}`). @rtype: `L{SFTPAttributes}` I{or error code}

```
class tardis.apps.sftp.sftp.MyTSFTPTCPServer(address, host_key, RequestHandler-  
                                           Class=None)
```

Bases: `socketserver.TCPServer`

allow_reuse_address = True

close_request(request)

Called to clean up an individual request.

shutdown_request(request)

Called to shutdown and close an individual request.

```
class tardis.apps.sftp.sftp.MyTServerInterface
```

Bases: `paramiko.server.ServerInterface`

check_auth_interactive(username, submethods)

Begin an interactive authentication challenge, if supported. You should override this method in server mode if you want to support the “keyboard-interactive” auth type, which requires you to send a series of questions for the client to answer.

Return `AUTH_FAILED` if this auth method isn’t supported. Otherwise, you should return an *InteractiveQuery* object containing the prompts and instructions for the user. The response will be sent via a call to *check_auth_interactive_response*.

The default implementation always returns `AUTH_FAILED`.

Parameters

- **username** (*str*) – the username of the authenticating client
- **submethods** (*str*) – a comma-separated list of methods preferred by the client (usually empty)

Returns `AUTH_FAILED` if this auth method isn’t supported; otherwise an object containing queries for the user

Return type int or *InteractiveQuery*

check_auth_interactive_response(responses)

Continue or finish an interactive authentication challenge, if supported. You should override this method in server mode if you want to support the “keyboard-interactive” auth type.

Return `AUTH_FAILED` if the responses are not accepted, `AUTH_SUCCESSFUL` if the responses are accepted and complete the authentication, or `AUTH_PARTIALLY_SUCCESSFUL` if your authentication is stateful, and this set of responses is accepted for authentication, but more authentication is required. (In this latter case, *get_allowed_auths* will be called to report to the client what options it has for continuing the authentication.)

If you wish to continue interactive authentication with more questions, you may return an *InteractiveQuery* object, which should cause the client to respond with more answers, calling this method again. This cycle can continue indefinitely.

The default implementation always returns `AUTH_FAILED`.

Parameters `responses` – list of *str* responses from the client

Returns `AUTH_FAILED` if the authentication fails; `AUTH_SUCCESSFUL` if it succeeds; `AUTH_PARTIALLY_SUCCESSFUL` if the interactive auth is successful, but authentication must continue; otherwise an object containing queries for the user

Return type `int` or *InteractiveQuery*

check_auth_password (*username*, *password*)

Determine if a given username and password supplied by the client is acceptable for use in authentication.

Return `AUTH_FAILED` if the password is not accepted, `AUTH_SUCCESSFUL` if the password is accepted and completes the authentication, or `AUTH_PARTIALLY_SUCCESSFUL` if your authentication is stateful, and this key is accepted for authentication, but more authentication is required. (In this latter case, *get_allowed_auths* will be called to report to the client what options it has for continuing the authentication.)

The default implementation always returns `AUTH_FAILED`.

Parameters

- **username** (*str*) – the username of the authenticating client.
- **password** (*str*) – the password given by the client.

Returns `AUTH_FAILED` if the authentication fails; `AUTH_SUCCESSFUL` if it succeeds; `AUTH_PARTIALLY_SUCCESSFUL` if the password auth is successful, but authentication must continue.

Return type `int`

check_auth_publickey (*username*, *key*)

Determine if a given key supplied by the client is acceptable for use in authentication. You should override this method in server mode to check the username and key and decide if you would accept a signature made using this key.

Return `AUTH_FAILED` if the key is not accepted, `AUTH_SUCCESSFUL` if the key is accepted and completes the authentication, or `AUTH_PARTIALLY_SUCCESSFUL` if your authentication is stateful, and this password is accepted for authentication, but more authentication is required. (In this latter case, *get_allowed_auths* will be called to report to the client what options it has for continuing the authentication.)

Note that you don't have to actually verify any key signature here. If you're willing to accept the key, Paramiko will do the work of verifying the client's signature.

The default implementation always returns `AUTH_FAILED`.

Parameters

- **username** (*str*) – the username of the authenticating client
- **key** (*PKey*) – the key object provided by the client

Returns `AUTH_FAILED` if the client can't authenticate with this key; `AUTH_SUCCESSFUL` if it can; `AUTH_PARTIALLY_SUCCESSFUL` if it can authenticate with this key but must continue with authentication

Return type `int`

check_channel_request (*kind*, *chanid*)

Determine if a channel request of a given type will be granted, and return `OPEN_SUCCEEDED` or an error code. This method is called in server mode when the client requests a channel, after authentication is complete.

If you allow channel requests (and an ssh server that didn't would be useless), you should also override some of the channel request methods below, which are used to determine which services will be allowed on a given channel:

- *check_channel_ptty_request*
- *check_channel_shell_request*
- *check_channel_subsystem_request*
- *check_channel_window_change_request*
- *check_channel_x11_request*
- *check_channel_forward_agent_request*

The *chanid* parameter is a small number that uniquely identifies the channel within a *.Transport*. A *.Channel* object is not created unless this method returns `OPEN_SUCCEEDED` – once a *.Channel* object is created, you can call *.Channel.get_id* to retrieve the channel ID.

The return value should either be `OPEN_SUCCEEDED` (or 0) to allow the channel request, or one of the following error codes to reject it:

- `OPEN_FAILED_ADMINISTRATIVELY_PROHIBITED`
- `OPEN_FAILED_CONNECT_FAILED`
- `OPEN_FAILED_UNKNOWN_CHANNEL_TYPE`
- `OPEN_FAILED_RESOURCE_SHORTAGE`

The default implementation always returns `OPEN_FAILED_ADMINISTRATIVELY_PROHIBITED`.

Parameters

- **kind** (*str*) – the kind of channel the client would like to open (usually "session").
- **chanid** (*int*) – ID of the channel

Returns an *int* success or failure code (listed above)

get_allowed_auths (*username*)

Return a list of authentication methods supported by the server. This list is sent to clients attempting to authenticate, to inform them of authentication methods that might be successful.

The "list" is actually a string of comma-separated names of types of authentication. Possible values are "password", "publickey", and "none".

The default implementation always returns "password".

Parameters **username** (*str*) – the username requesting authentication.

Returns a comma-separated *str* of authentication types

myt_auth (*username*, *password*)

`tardis.apps.sftp.sftp.start_server` (*host=None*, *port=None*, *keyfile=None*)

The `SFTP_HOST_KEY` setting is required for configuring SFTP access. The `SFTP_PORT` setting defaults to 2200.

See: `tardis/default_settings/sftp.py`

tardis.apps.sftp.urls module

tardis.apps.sftp.user_menu_modifiers module

`tardis.apps.sftp.user_menu_modifiers.add_ssh_keys_menu_item(request, user_menu)`

Add a 'Manage SSH Keys' item to the user menu

Parameters

- **request** (`django.http.HttpRequest`) – an HTTP Request instance
- **user_menu** (`list`) – user menu context to modify

Returns user_menu list

Return type `list`

tardis.apps.sftp.views module

`tardis.apps.sftp.views.cybderduck_connection_window(request)`

`tardis.apps.sftp.views.sftp_access(request)`

Show dynamically generated instructions on how to connect to SFTP :param Request request: HttpRequest
:return: HttpResponse :rtype: HttpResponse

`tardis.apps.sftp.views.sftp_keys(request)`

Generate an RSA key pair for a user.

Generates a key pair, stores the public part of the key and provides a one time opportunity for the user to download the private part of the key.

Parameters **request** (`HttpRequest`) – http request

Returns either returns form on GET request or private key download on POST request

Return type `HttpResponse`

Module contents

tardis.apps.social_auth package

Subpackages

tardis.apps.social_auth.auth package

Submodules

tardis.apps.social_auth.auth.authorisation module

tardis.apps.social_auth.auth.social_auth module

`tardis.apps.social_auth.auth.social_auth.add_authentication_method(**kwargs)`

Creates an authentication record for OpenID authenticated user

`tardis.apps.social_auth.auth.social_auth.add_migration_permission(**kwargs)`
Adds permission to migrate account for OpenID authenticated user

`tardis.apps.social_auth.auth.social_auth.add_user_permissions(**kwargs)`
Adds default permission to OpenID authenticated user

`tardis.apps.social_auth.auth.social_auth.approve_user_auth(**kwargs)`
Sets approved status to True in user authentication This will add user permissions as well.

`tardis.apps.social_auth.auth.social_auth.configure_social_auth_user(**kwargs)`
Applies configuration used for external (non-Django) accounts.
Adds user to settings.NEW_USER_INITIAL_GROUPS and sets isDjangoAccount to False in their UserProfile, so that MyTardis won't allow them to change their password.

`tardis.apps.social_auth.auth.social_auth.get_auth_method(authenticatedBackendName)`
Return matching user authentication method from list of authentication methods in settings

`tardis.apps.social_auth.auth.social_auth.is_openid_migration_enabled()`

`tardis.apps.social_auth.auth.social_auth.migrate_user_message(**kwargs)`
Automatically detects if a user has an account with the same email address and prompts user to perform migration.

`tardis.apps.social_auth.auth.social_auth.requires_admin_approval(authenticationBackend)`

`tardis.apps.social_auth.auth.social_auth.send_admin_email(**kwargs)`
Sends MyTardis admins an email for approving account

Module contents

Submodules

`tardis.apps.social_auth.apps` module

```
class tardis.apps.social_auth.apps.SocialAuthConfig(app_name, app_module)
    Bases: tardis.app_config.AbstractTardisAppConfig

    name = 'tardis.apps.social_auth'
    verbose_name = 'Social Auth'
```

`tardis.apps.social_auth.default_settings` module

`tardis.apps.social_auth.urls` module

Module contents

Module contents

`tardis.default_settings` package

Submodules

tardis.default_settings.admins module

`tardis.default_settings.admins.ADMINS = []`

A list of all the people who get code error notifications. When `DEBUG=False` and `AdminEmailHandler` is configured in `LOGGING` (done by default), Django emails these people the details of exceptions raised in the request/response cycle.

Each item in the list should be a tuple of (Full name, email address). Example:

```
[(‘John’, ‘john@example.com’), (‘Mary’, ‘mary@example.com’)]
```

tardis.default_settings.analytics module

tardis.default_settings.apps module

`tardis.default_settings.apps.USER_MENU_MODIFIERS = ['tardis.apps.sftp.user_menu_modifiers..`

A list of methods which can modify the user menu defined in `tardis.tardis_portal.context_processors.user_menu_processor`. The modifications will be applied in order, so it is possible for one app to overwrite changes made by another app whose modifier method is earlier in the list.

Each modifier method should take a `django.http.HttpRequest` object and a list of user menu items, and return a modified list of user menu items.

An example from the SFTP app is below:

```
USER_MENU_MODIFIERS.extend([
    'tardis.apps.sftp.user_menu_modifiers.add_ssh_keys_menu_item'
])
```

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': {'BACKEND': 'django.core.cache.backends`

change the `CACHES` setting to memcached if you prefer. Requires additional dependencies.

tardis.default_settings.celery_settings module

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.io/en/develop/apps/contextual_views.html#dataset-and-experiment-views

e.g.:

```
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.io/en/develop/apps/contextual_views.html#dataset-and-experiment-views

e.g.:

```
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.io/en/develop/apps/contextual_views.html#custom-index-view

e.g.:

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

`tardis.default_settings.custom_views.LOGIN_VIEWS = {}`

A custom login 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.io/en/develop/apps/contextual_views.html#custom-login-view

e.g.:

```
LOGIN_VIEWS = {
    1: 'tardis.apps.my_custom_app.views.MyCustomLoginSubclass',
    'store.example.com': 'tardis.apps.myapp.AnotherCustomLoginSubclass'
}
```

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.debug.INTERNAL_IPS = ('127.0.0.1',)`

A list of IP addresses, as strings, that: Allow the debug() context processor to add some variables to the template context.

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.downloads.DOWNLOAD_URI_TEMPLATES = {}
```

When a file download is requested, by default, MyTardis will create a StreamingHttpResponse to serve the download which requires reading the file from a file-like object. For some storage backends, e.g. S3Boto3Storage provided by the django-storages package, it is more efficient to redirect the request directly to the storage provider.

Or the download request could be handled by an API which is aware of HSM (Hierarchical Storage Management) status for files which could be either on disk or on tape.

The DOWNLOAD_URI_TEMPLATES dictionary can be used to specify a URI template (e.g. '/api/v1/s3utils_replica/{dfo_id}/download/' which can be used to download a DataFileObject, instead of using the default /api/v1/dataset_file/[datafile_id]/download/ endpoint.

For example,

```
DOWNLOAD_URI_TEMPLATES = { 'storages.backends.s3boto3.S3Boto3Storage':
    '/api/v1/s3utils_replica/{dfo_id}/download/'
}
```

The '/api/v1/s3utils_replica/{dfo_id}/download/' endpoint is provided by the 's3utils' tardis app which needs to be in your INSTALLED_APPS

tardis.default_settings.email module

```
tardis.default_settings.email.DEFAULT_FROM_EMAIL = 'webmaster@localhost'
```

This can be set as : "MyTardis Admins <admins@mytardis.org>"

```
tardis.default_settings.email.EMAIL_HOST = 'localhost'
```

Set this to your local SMTP server, e.g. 'smtp.example.edu' or to a remote SMTP server, e.g. 'smtp.gmail.com'

```
tardis.default_settings.email.EMAIL_HOST_PASSWORD = ''
```

When using a local SMTP server, you probably don't need to authenticate, so you can leave this blank.

If using a remote SMTP server, this can be set to the password used to authenticate.

```
tardis.default_settings.email.EMAIL_HOST_USER = ''
```

When using a local SMTP server, you probably don't need to authenticate, so you can leave this blank.

If using a remote SMTP server, this can be set to the email address used to authenticate, e.g. 'bob@bobmail.com'

```
tardis.default_settings.email.EMAIL_PORT = 25
```

Some SMTP servers require a different port, e.g. 587. Django's default value for this setting is 25.

```
tardis.default_settings.email.EMAIL_USE_TLS = False
```

Some SMTP servers require this to be set to True. Django's default value for this setting is False.

tardis.default_settings.filters module

```
tardis.default_settings.filters.FILTERS_TASK_PRIORITY = 4
```

The default RabbitMQ task priority for messages sent to the filters microservice. Priority 4 is slightly less than

the overall default task priority of 5, defined in `tardis/default_settings/celery_settings.py`

`tardis.default_settings.filters.USE_FILTERS = False`

If enabled, a task will be sent to RabbitMQ after a file is saved and verified, requesting post-processing, e.g. extracting metadata from file headers and/or generating thumbnail images.

`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.frontend.MAX_IMAGES_IN_CAROUSEL = 100`

Max number of images in dataset view's carousel: zero means no limit

`tardis.default_settings.frontend.RENDER_IMAGE_DATASET_SIZE_LIMIT = 0`

Render image dataset size limit: zero means no limit

In order to display a dataset thumbnail image, MyTardis queries the dataset's files for image MIME types. These queries can be very slow for large datasets, making page load times slow. In future versions, the dataset thumbnails will be generated asynchronously (not at response time), but for now, we can set `RENDER_IMAGE_DATASET_SIZE_LIMIT` to the maximum number of files a dataset can have for MyTardis to scan it for image files at response time.

`tardis.default_settings.frontend.RENDER_IMAGE_SIZE_LIMIT = 0`

Render image file size limit: zero means no limit

`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

Settings for search

`tardis.default_settings.search.ELASTICSEARCH_PARALLEL_INDEX_SETTINGS = {'chunk_size': 500,`

Setting for running elastic search index in parallel mode to get indexing speed boost while indexing <https://django-elasticsearch-dsl.readthedocs.io/en/latest/settings.html#elasticsearch-dsl-parallel>

`tardis.default_settings.search.MAX_SEARCH_RESULTS = 100`

Limits the maximum number of search results for each model (Experiment, Dataset and DataFile). The default value of 100 means that a query could potentially return 300 results in total, i.e. 100 experiments, 100 datasets and 100 datafiles.

`tardis.default_settings.search.MIN_CUTOFF_SCORE = 0.0`

Filters results based on this value. The default value of 0.0 means that nothing will be excluded from search results. Set it to any number greater than 0.0 to filter out results.

```
tardis.default_settings.search.SINGLE_SEARCH_ENABLED = False
```

To enable search:

```
SINGLE_SEARCH_ENABLED = True INSTALLED_APPS += ('django_elasticsearch_dsl',
'tardis.app.search') ELASTICSEARCH_DSL = {
    'default': { 'hosts': os.environ.get('ELASTICSEARCH_URL', 'http://localhost:9200')
    }
} ELASTICSEARCH_DSL_INDEX_SETTINGS = {
    'number_of_shards': 1, 'number_of_replicas': 0
}
```

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.static_files module

tardis.default_settings.storage module

```
tardis.default_settings.storage.CALCULATE_CHECKSUMS_METHODS = {}
```

A custom method can be provided for calculating checksums for a storage class, e.g.

```
CALCULATE_CHECKSUMS_METHODS = {
    'storages.backends.s3boto3.S3Boto3Storage': 'tardis.apps.s3utils.utils.calculate_checksums'
}
```

The `DataFileObject` class's `calculate_checksums` method checks for a storage class match in the `CALCULATE_CHECKSUMS_METHODS` dict, and if one is not found, it calls the classic `compute_checksums` method which uses the `file_object` to calculate the checksums one chunk at a time. For some storage backends (e.g. S3), representing the file as file-like object with Django's file storage API is not the most efficient way to calculate the checksum.

```
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.storage.REUSE_DATASET_STORAGE_BOX = True
```

If a new `DataFile` is created in a `Dataset` whose files are all stored in the same storage box, then the new file will be stored in the same storage box, irrespective of the “default” `StorageBoxAttribute`.

The `mytardis-app-mydata` app has its own logic for determining the appropriate storage box for uploads from a `MyData` instance. When a `MyData` instance (an “Uploader”) is approved, it will be assigned a storage box which should be used for `DataFileObjects` created from `MyData` uploads.

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.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 (*request*)

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=<LazySettings "tardis.test_settings">*)

Bases: `object`

The AuthService provides an interface for querying the auth(nl/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 returned 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, group_id)
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.interfaces module

```
class tardis.tardis_portal.auth.interfaces.AuthProvider
```

Bases: `object`

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
```

Bases: `object`

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
```

Bases: `object`

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 = 'django_group'

searchGroups (***filter*)

return a list of groups that match the 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 = 'django_user'
```

tardis.tardis_portal.auth.token_auth module

token authentication module

```
class tardis.tardis_portal.auth.token_auth.TokenAuthMiddleware(get_response)  
    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)  
    return an iteration of the available groups.
```

```
name = '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 marking it as a not a Django account.

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.management package

Subpackages

tardis.tardis_portal.management.commands package

Submodules

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,
                                                                force_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments (parser)
        Entry point for subclassed commands to add custom arguments.

    handle (*args, **options)
        The actual logic of the command. Subclasses must implement this method.

    help = 'Used to create a MyTardis user.'
```

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,
                                                                force_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments (parser)
        Entry point for subclassed commands to add custom arguments.

    args = '[namespace...]'

    handle (*args, **options)
        The actual logic of the command. Subclasses must implement this method.

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

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,
                                                                force_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments (parser)
        Entry point for subclassed commands to add custom arguments.

    args = 'schema [schema ...]'
```

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

The actual logic of the command. Subclasses must implement this method.

```
help = 'Load soft schema definitions'
```

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,
                                                                    force_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments (parser)
        Entry point for subclassed commands to add custom arguments.

    args = '<MyTardis Exp ID>'

    handle (*args, **options)
        The actual logic of the command. Subclasses must implement this method.

    help = 'Delete the supplied MyTardis Experiment ID'
```

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 = [('contenttypes', '0002_remove_content_type_name'), ('auth', '__first__')]
    operations = [<CreateModel name='DataFile', fields=[('id', <django.db.models.fields.AutoField object>)]>]
```

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_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('contenttypes', '0002_remove_content_type_name'), ('auth', '__first__')]
    operations = [<CreateModel name='DataFile', fields=[('id', <django.db.models.fields.AutoField object>)]>]
```



```

    replaces = [('tardis_portal', '0001_initial'), ('tardis_portal', '0002_auto_20150528_1128')]
tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643.cast_string_to_integer

```

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 = [('tardis_portal', '0001_initial')]
    operations = [<AlterField model_name='userprofile', name='user', field=<django.db.models.fields.CharField object>]

```

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 = [('tardis_portal', '0002_auto_20150528_1128')]
    operations = [<AlterField model_name='parametername', name='data_type', field=<django.db.models.fields.CharField object>]

```

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 = [('tardis_portal', '0003_auto_20150907_1315')]
    operations = [<AddField model_name='storageboxoption', name='value_type', field=<django.db.models.fields.CharField object>]

```

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 = [('tardis_portal', '0004_storageboxoption_value_type')]
    operations = [<AddField model_name='datafile', name='_size', field=<django.db.models.fields.IntegerField object>]
tardis.tardis_portal.migrations.0005_datafile_add_size_int_column.cast_string_to_integer(app_label='tardis_portal', schema_name='tardis_portal', table_name='datafile', column_name='_size')

```

tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column module

```

class tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column.Migration(name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '0005_datafile_add_size_int_column')]

```

```
operations = [<RemoveField model_name='datafile', name='size'>, <RenameField model_name='datafile', name='size'>]
```

`tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index` module

```
class tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index.Migration(name, app_label):
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '0006_datafile_remove_size_string_column')]
    operations = [<AlterField model_name='datafileparameter', name='string_value', field=<django.db.models.CharField(max_length=255)>>]
```

`tardis.tardis_portal.migrations.0008_string_value_partial_index_postgres` module

```
class tardis.tardis_portal.migrations.0008_string_value_partial_index_postgres.Migration(name, app_label):
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '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 = [('tardis_portal', '0008_string_value_partial_index_postgres')]
    operations = [<AlterUniqueTogether name='instrument', unique_together={('name', 'facility')}>]
```

`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 = [('tardis_portal', '0009_auto_20160128_1119')]
    operations = [<AlterField model_name='datafile', name='mimetype', field=<django.db.models.CharField(max_length=255)>>]
```

`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 = [('tardis_portal', '0010_auto_20160503_1443')]
    operations = [<AlterField model_name='datafile', name='directory', field=<django.db.models.CharField(max_length=255)>>]
```

tardis.tardis_portal.migrations.0012_userauthentication_approved module

```
class tardis.tardis_portal.migrations.0012_userauthentication_approved.Migration(name,
                                         app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '0011_auto_20160505_1643')]
    operations = [<AddField model_name='userauthentication', name='approved', field=<django.db.migrations.fields.BooleanField>]
```

tardis.tardis_portal.migrations.0013_auto_20181002_1136 module

```
class tardis.tardis_portal.migrations.0013_auto_20181002_1136.Migration(name,
                                         app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '0012_userauthentication_approved')]
    operations = [<AlterModelOptions name='facility', options={'ordering': ('name',)}, 'verbose_name']
```

tardis.tardis_portal.migrations.0014_auto_20181002_1154 module

```
class tardis.tardis_portal.migrations.0014_auto_20181002_1154.Migration(name,
                                         app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '0013_auto_20181002_1136')]
    operations = [<AlterField model_name='storagebox', name='max_size', field=<django.db.models.fields.IntegerField>]
```

tardis.tardis_portal.migrations.0015_dataset_created_time module

```
class tardis.tardis_portal.migrations.0015_dataset_created_time.Migration(name,
                                         app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '0014_auto_20181002_1154')]
    operations = [<AddField model_name='dataset', name='created_time', field=<django.db.models.fields.DateTimeField>]
```

tardis.tardis_portal.migrations.0016_add_timestamps module

```
class tardis.tardis_portal.migrations.0016_add_timestamps.Migration(name,
                                         app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '0015_dataset_created_time')]
    operations = [<AddField model_name='facility', name='created_time', field=<django.db.models.fields.DateTimeField>]
```

tardis.tardis_portal.migrations.0017_add_cc_licenses module

```
class tardis.tardis_portal.migrations.0017_add_cc_licenses.Migration(name,
                                         app_label)
    Bases: django.db.migrations.migration.Migration
```

```
dependencies = [('tardis_portal', '0016_add_timestamps'), ('auth', '0008_alter_user_us
operations = [<RunPython <function forwards_func>, <function reverse_func>>]
tardis.tardis_portal.migrations.0017_add_cc_licenses.forwards_func(apps,
                                                                    schema_editor)
tardis.tardis_portal.migrations.0017_add_cc_licenses.load_licenses(apps,
                                                                    schema_editor)
tardis.tardis_portal.migrations.0017_add_cc_licenses.remove_licenses(apps,
                                                                    schema_editor)
tardis.tardis_portal.migrations.0017_add_cc_licenses.reverse_func(apps,
                                                                    schema_editor)
```

tardis.tardis_portal.migrations.0018_make_default_storage_box_status_online module

```
class tardis.tardis_portal.migrations.0018_make_default_storage_box_status_online.Migration
    Bases: django.db.migrations.migration.Migration
    dependencies = [('tardis_portal', '0017_add_cc_licenses')]
    operations = [<AlterField model_name='storagebox', name='status', field=<django.db.mod
```

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 foreign key to the django.contrib.auth.models.User
        Attribute group a foreign key to the django.contrib.auth.models.Group
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist
exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
group
    Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOne-
    ToOneDescriptor subclass) relation.
```

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

group_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

aclOwnershipType

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

canDelete

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

canRead

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

canWrite

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

content_object

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

content_type

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

content_type_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

effectiveDate

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

entityId

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

expiryDate

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_aclOwnershipType_display (*, *field=<django.db.models.fields.IntegerField: aclOwnershipType>*)

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.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

isOwner

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

object_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

openidaclmigration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

pluginId

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.access_control.UserAuthentication(id, user-
                                                                    Profile,
                                                                    user-
                                                                    name,
                                                                    authen-
                                                                    tication-
                                                                    Method,
                                                                    ap-
                                                                    proved)
```

Bases: django.db.models.base.Model

CHOICES = ()

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

approved

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

authenticationMethod

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

getAuthMethodDescription()

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

save (*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

userProfile

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

userProfile_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

username

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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 ()

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

isDjangoAccount

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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 necessarily have a last (or family) name.

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

rapidConnectEduPersonTargetedID

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

userauthentication_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
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`

A `DataFile` is a record of a file which includes its filename, its size in bytes, its relative directory, and various other meta-data. Each `DataFile` belongs to a `Dataset` which usually represents the files from one folder on an instrument PC.

The physical copy (or copies) of a file are described by distinct `DataFileObject` records.

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 Should be populated with the file's creation time from the instrument PC.

Attribute modification_time Should be populated with the file's last modification time from the instrument PC.

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()`

`created_time`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`datafileparameterset_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`dataset`

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

`dataset_id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`deleted`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`deleted_time`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`directory`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`file_object`

`file_objects`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

filename

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

getParameterSets()

Return datafile parametersets associated with this datafile.

get_absolute_filepath()

get_as_temporary_file(directory=None)

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.

Parameters `directory` (*basestring*) – 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 dataset or use global default

get_download_url()

get_file(verified_only=True)

Returns the file as a readable file-like object from the best available 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()

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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()

md5sum

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

mimetype

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modification_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

save(*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

sha512sum

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

size

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

Return True if at least one DataFileObject is verified

verify(reverify=False)

version

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class tardis.tardis_portal.models.datafile.DataFileObject(*args, **kwargs)

Bases: django.db.models.base.Model

The physical copy (or copies) of a *DataFile* are described by distinct *DataFileObject* records.

Attribute datafile The *DataFile* record which this DataFileObject is storing a copy of.

Attribute storage_box The *StorageBox* containing this copy of the file. The *StorageBox* could represent a directory on a mounted filesystem, or a bucket in an Object Store.

Attribute uri The relative path of the file location within the the *StorageBox*, e.g. dataset1-12345/file1.txt for a copy of a *DataFile* with filename file1.txt which belongs to a *Dataset* with a description of dataset1 and an ID of 12345.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

apply_filters()

cache_file()

calculate_checksums (*compute_md5=True, compute_sha512=False*)

Calculates checksums for a *DataFileObject* instance

Parameters

- **compute_md5** (*bool*) – whether to compute md5 default=True
- **compute_sha512** (*bool*) – whether to compute sha512, default=True

Returns the checksums as {'md5sum': result, 'sha512sum': result}

Return type `dict`

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 *bool* save: :return: :rtype: *basestring*

created_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datafile

Accessor to the related object on the forward side of a many-to-one or one-to-one (via *ForwardOneToOneDescriptor* subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a *ForwardManyToOneDescriptor* instance.

datafile_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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 (*, field=<django.db.models.fields.DateTimeField: created_time>, is_next=True, **kwargs)

get_previous_by_created_time (*, field=<django.db.models.fields.DateTimeField: created_time>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

last_verified_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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>

priority

Default priority for tasks which take this DFO as an argument

save (*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

storage_box

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

storage_box_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

storage_type

Returns storage_box type

Return type StorageBox type constant

uri

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

verified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

verify (*add_checksums=True, add_size=True*)

```
tardis.tardis_portal.models.datafile.compute_checksums (file_object,      com-
                                                         pute_md5=True,      com-
                                                         pute_sha512=False,
                                                         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)
Deletes the actual file / object, before deleting the database record
```

tardis.tardis_portal.models.dataset module

class tardis.tardis_portal.models.dataset.Dataset (*args, **kwargs)

Bases: `django.db.models.base.Model`

A dataset represents a collection files usually associated with a folder on an instrument PC. Each file within the dataset is represented by a `tardis.tardis_portal.models.DataFile` record. A dataset can appear in one or more *Experiment* records. Access controls are configured at the *Experiment* level by creating *ObjectACL* records. Each dataset can be associated with an *Instrument* record, but it is possible to create a dataset without specifying an instrument.

Attribute experiment A foreign key to the one ore more *Experiment* records which contain this dataset

Attribute instrument The foreign key to the instrument that generated this data

Attribute description Description of this dataset, which usually corresponds to the folder name on the instrument PC

Attribute immutable Whether this dataset is read-only

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

created_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datafile_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

datasetparameterset_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

directory

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

experiments

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

getParameterSets()

Return the dataset parametersets associated with this experiment.

get_absolute_url()

Return the absolute url to the current Dataset

get_all_storage_boxes_used()

get_datafiles()

get_dir_nodes(*dir_tuples*)

Return child node's subdirectories in format required for tree view

Given a list of ('subdir', 'path/to/subdir') tuples for a dataset directory node, return a list of {'name': 'subdir', 'children': []} dictionaries required for the tree view.

Like the `get_dir_tuples` method, the `get_dir_nodes` method only lists files and directories immediately within the supplied basedir, so any subdirectories will have an empty children array.

Continuing the example from the `_dirs` property method:

test files/subdir1/ test files/subdir2/ test files/subdir3/ test files/subdir3/subdir4/

List directories in the dataset's top level directory: `>>> dir_tuples = ds.get_dir_tuples("") >>> ds.get_dir_nodes(dir_tuples) [`


```

    { 'name': 'test files', 'path': '', 'children': []
    }
]

```

List directories within the dataset's "test files" directory: >>> dir_tuples = ds.get_dir_tuples("test files")
>>> ds.get_dir_nodes(dir_tuples) [

```

    { 'name': 'subdir1', 'path': 'test files/subdir1', 'children': []
    }, {
        'name': 'subdir2', 'path': 'test files/subdir2', 'children': []
    }, {
        'name': 'subdir3', 'path': 'test files/subdir3', 'children': []
    },
]

```

Request directories within a non-existent directory: >>> dir_tuples = ds.get_dir_tuples("test file") >>> ds.get_dir_nodes(dir_tuples) []

List directories within the dataset's "test files/subdir3" directory: >>> dir_tuples = ds.get_dir_tuples("test files3/subdir3") >>> ds.get_dir_nodes(dir_tuples) [

```

    { 'name': 'subdir4', 'path': 'test files/subdir3/subdir4', 'children': []
    }
]

```

get_dir_tuples (*basedir=""*)

List the directories immediately inside basedir.

If basedir is empty (the default), list directories with no separator, e.g. "subdir1"

If basedir is a string without a separator (e.g. "subdir1"), look for directory paths with one separator e.g. "subdir1/subdir2" and include a ".." for navigating back to the dataset's top-level directory.

Continuing the example from the `_dirs` property method:

```
test files/subdir1/ test files/subdir2/ test files/subdir3/ test files/subdir3/subdir4/
```

List directories in the dataset's top level directory: >>> ds.get_dir_tuples("") [('test files')]

List directories within the dataset's "test files" directory: >>> ds.get_dir_tuples("test files") [('..' , 'test files'), ('subdir1' , 'test /filessubdir1'),

```
( 'subdir2' , 'test files/subdir2' ), ( 'subdir3' , 'test files/subdir3' )]
```

Request directories within a non-existent directory: >>> ds.get_dir_tuples("test file") []

List directories within the dataset's "test files/subdir3" directory: >>> ds.get_dir_tuples("test files/subdir3") [('..' , 'test files/subdir3'), ('subdir4' , 'test files/subdir3/subdir4')]

List directories within the dataset's "test files/subdir3/subdir4" directory: >>> ds.get_dir_tuples("test files/subdir3/subdir4") [('..' , 'test files/subdir3/subdir4')]

get_download_urls ()

get_edit_url ()

Return the absolute url to the edit view of the current Dataset

get_first_experiment ()

get_images ()

get_path()

get_size()

get_thumbnail_url()

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

image

immutable

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

instrument

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

instrument_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_online

modified_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

save (*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

tardis.tardis_portal.models.experiment module

class tardis.tardis_portal.models.experiment.**Experiment** (*args, **kwargs)

Bases: django.db.models.base.Model

An Experiment is a collection of *Dataset* records. A *Dataset* record can appear in multiple Experiment records. Access controls are configured at the Experiment level by creating *ObjectACL* records.

Attribute url An optional URL associated with the data collection

Attribute approved An optional field indicating whether the collection is approved

Attribute title The title of the experiment.

Attribute description The description of the experiment.

Attribute institution_name The name of the institution who created the experiment.

Attribute start_time Undocumented

Attribute end_time Undocumented

Attribute created_time Undocumented

Attribute handle Undocumented

Attribute public Whether the experiment is publicly accessible

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`

approved

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created_by

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

created_by_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datasets

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

end_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

experimentauthor_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

experimentparameterset_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

getParameterSets()

Return the experiment parametersets associated with this experiment.

get_absolute_url()

Return the absolute url to the current Experiment

get_create_token_url()

Return the absolute url to the create token view of the current Experiment

get_ct()**get_datafiles()****get_download_urls()****get_edit_url()**

Return the absolute url to the edit view of the current Experiment

get_groups()**get_images()****get_next_by_created_time(*, field=<django.db.models.fields.DateTimeField: created_time>, is_next=True, **kwargs)**

```

get_next_by_update_time (*, field=<django.db.models.fields.DateTimeField: update_time>,
                        is_next=True, **kwargs)

get_or_create_directory ()

get_owners ()

get_previous_by_created_time (*, field=<django.db.models.fields.DateTimeField: created_time>, is_next=False, **kwargs)

get_previous_by_update_time (*, field=<django.db.models.fields.DateTimeField: update_time>, is_next=False, **kwargs)

get_public_access_display (*, field=<django.db.models.fields.PositiveSmallIntegerField: public_access>)

```

get_size ()

handle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

institution_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_publication ()

is_publication_draft ()

license

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```

class Child(Model):
    parent = ForeignKey(Parent, related_name='children')

```

Child.parent is a ForwardManyToOneDescriptor instance.

license_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

locked

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects

Accessor to the related objects manager on the one-to-many relation created by GenericRelation.

In the example:

```

class Post(Model):
    comments = GenericRelation(Comment)

```

post.comments is a ReverseGenericManyToOneDescriptor instance.

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

public_access

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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)

Save the current instance. Override this in a subclass if you want to control the saving process.

The 'force_insert' and 'force_update' parameters can be used to insist that the "save" must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

start_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

token_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

update_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.experiment.ExperimentAuthor(id, experiment,
                                                                author, institu-
                                                                tion, email, order,
                                                                url)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

author

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

experiment

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

experiment_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

institution

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

order

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save (*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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.

Each *Instrument* record must belong to exactly one facility. Many *Instrument* records can be associated with the same facility.

Attribute name The name of the facility, e.g. “Test Facility”

Attribute manager_group The group of users who can access the Facility Overview for this facility.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

created_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

instrument_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

manager_group

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

manager_group_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>**save (*args, **kwargs)**

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘`force_insert`’ and ‘`force_update`’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

`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.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`

created_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

dataset_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

facility

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

facility_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

getParameterSets()

Return the instrument parametersets associated with this instrument.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

instrumentparameterset_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

modified_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>**save (*args, **kwargs)**

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

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`

created_time

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_next_by_created_time(*, field=<django.db.models.fields.DateTimeField: created_time>,
                          is_next=True, **kwargs)
```

```
get_previous_by_created_time(*, field=<django.db.models.fields.DateTimeField: created_time>, is_next=False, **kwargs)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

jti

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
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`

```
allows_distribution
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
experiment_set
```

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
classmethod get_none_option_license()
```

```
classmethod get_suitable_licenses(public_access_method=None)
```

```
id
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
image_url
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
internal_description
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
is_active
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
name
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

tardis.tardis_portal.models.parameters module

```
class tardis.tardis_portal.models.parameters.DatafileParameter(id, name,  
                                                             string_value,  
                                                             numeri-  
                                                             cal_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`

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

link_ct

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

link_gfk

Provide a generic many-to-one relation through the content_type and object_id fields.

This class also doubles as an accessor to the related object (similar to ForwardManyToOneDescriptor) by adding itself as a model attribute.

name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

parameter_type = 'Datafile'**parameterset**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

parameterset_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
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

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

datafile_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datafileparameter_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

parameter_class

alias of `DatafileParameter`

schema

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

storage_box

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

```
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*

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

link_ct

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

link_gfk

Provide a generic many-to-one relation through the content_type and object_id fields.

This class also doubles as an accessor to the related object (similar to ForwardManyToOneDescriptor) by adding itself as a model attribute.

name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

parameter_type = 'Dataset'

parameterset

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

parameterset_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
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

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

dataset_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datasetparameter_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

parameter_class

alias of *DatasetParameter*

schema

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

storage_box

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
class tardis.tardis_portal.models.parameters.ExperimentParameter(id, name,
                                                                string_value,
                                                                numerical_value,
                                                                date_time_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*

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

link_ct

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

link_gfk

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

parameter_type = 'Experiment'

parameterset

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

parameterset_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save (*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

```
class tardis.tardis_portal.models.parameters.ExperimentParameterSet(id,
                                                                    schema,
                                                                    experi-
                                                                    ment)
```

Bases: `tardis.tardis_portal.models.parameters.ParameterSet`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

experiment

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

experiment_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

experimentparameter_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

parameter_class

alias of *ExperimentParameter*

schema

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

storage_box

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

```
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

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

parameter_name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

parameter_name_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.InstrumentParameter(id, name,
                                                                string_value,
                                                                numeri-
                                                                cal_value,
                                                                date-
                                                                time_value,
                                                                link_id,
                                                                link_ct, pa-
                                                                rameterset)
```

Bases: *tardis.tardis_portal.models.parameters.Parameter*

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

link_ct

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

link_gfk

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to ForwardManyToOneDescriptor) by adding itself as a model attribute.

name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

parameter_type = 'Instrument'

parameterset

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

parameterset_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.InstrumentParameterSet(id,
                                                                    schema,
                                                                    instru-
                                                                    ment)
```

Bases: `tardis.tardis_portal.models.parameters.ParameterSet`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

instrument

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

instrument_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

instrumentparameter_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

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

parameter_class

alias of `InstrumentParameter`

schema

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

storage_box

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
class tardis.tardis_portal.models.parameters.Parameter(*args, **kwargs)
```

Bases: `django.db.models.base.Model`

class Meta

Bases: `object`

abstract = `False`

app_label = `'tardis_portal'`

ordering = `['name']`

datetime_value

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get()

link_ct

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

link_ct_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

link_gfk

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

link_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

link_url**name**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

name_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

numerical_value

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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** –

string_value

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
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

choices

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

comparison_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

data_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datafileparameter_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

datasetparameter_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

experimentparameter_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

freetextsearchfield_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

full_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

getUniqueShortName()

get_comparison_type_display (*, field=<django.db.models.fields.IntegerField: comparison_type>)

get_data_type_display (*, field=<django.db.models.fields.IntegerField: data_type>)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

immutable

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

instrumentparameter_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

isDateTime()

isFilename()**isLink()****isLongString()****isNumeric()****isString()****isURL()****is_json()****is_searchable**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

natural_key()**objects** = <tardis.tardis_portal.managers.ParameterNameManager object>**order**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

schema

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

schema_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

units

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.ParameterSet(*args, **kwargs)
Bases: django.db.models.base.Model, tardis.tardis_portal.models.parameters.
ParameterSetManagerMixin
```

class Meta

Bases: object

abstract = False**app_label** = 'tardis_portal'**ordering** = ['id']**parameter_class** = None

save (*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

schema

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

schema_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

storage_box

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

```
class tardis.tardis_portal.models.parameters.ParameterSetManagerMixin(parameterset=None,
                                                                    par-
                                                                    en-
                                                                    tOb-
                                                                    ject=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, sub-
                                                    type, 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: `Exception`

datafileparameterset_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

datasetparameterset_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

experimentparameterset_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_type_display (*, *field*=<*django.db.models.fields.IntegerField: type*>)

hidden

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

immutable

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

instrumentparameterset_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

namespace

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

natural_key()

objects = `<tardis.tardis_portal.managers.SchemaManager object>`

parametername_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

subtype

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

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

child_boxes

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

copy_files (*dest_box=None*)**copy_to_master** ()**classmethod create_local_box** (*location=None*)**datafileparametersets**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

datasetparametersets

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

django_storage_class

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

experimentparametersets

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

file_objects

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

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** ()**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

instrumentparametersets

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

master_box

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

master_box_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

max_size

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

move_files (*dest_box=None*)

move_to_master ()

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

offline_types = [2]

online_types = [3, 1, 4, 6]

options

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

priority

Default priority for tasks which take this box as an argument

status

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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.

Attribute key The key used to look up the attribute (e.g. “type”).

Attribute value The value of the attribute e.g. “cache”.

The “type” key has three values support by the core MyTardis code: “permanent”, “receiving” and “cache”.

Adding an attribute with key “type” and value “permanent” preserves the default behaviour of a storage box, i.e. it is equivalent to not adding the attribute.

Adding an attribute with key “type” and value “receiving” means that the storage box will be treated as a staging area which receives files intended to be transfered to a permanent storage box. If a storage box has the “type: receiving” attribute, it must link to a permanent storage box (via its master_box foreign key).

Adding an attribute with key “type” and value “cache” means that the storage box will be used to copy data from slow-access storage to fast-access storage. If a storage box has the “type: cache” attribute, it must link to a permanent storage box (via its master_box foreign key).

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

key

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

storage_box

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

storage_box_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

value

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
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 DoesNotExistBases: `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** (`*, field=<django.db.models.fields.CharField: value_type>`)**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

key

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = `<django.db.models.manager.Manager object>`**storage_box**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

storage_box_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

unpickled_value**value**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

value_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

experiment_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

expiry_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_expiry_date (*, field=<django.db.models.fields.DateField: expiry_date>, is_next=True, **kwargs)

get_previous_by_expiry_date (*, field=<django.db.models.fields.DateField: expiry_date>, is_next=False, **kwargs)

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

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_expired()

objects = <tardis.tardis_portal.managers.OracleSafeManager object>

save_with_random_token()

token

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

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)
```

Module contents

Submodules

tardis.tardis_portal.publish.publishservice module

```
class tardis.tardis_portal.publish.publishservice.PublishService (providers, ex-
                                                                periment)
    Bases: object
    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.
```

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

tardis.tardis_portal templatetags package

Submodules

tardis.tardis_portal templatetags approved_user_tags module

tardis.tardis_portal templatetags approved_user_tags.**check_if_user_not_approved**(*request*)

Custom template filter to identify whether a user account is approved.

tardis.tardis_portal templatetags approved_user_tags.**get_matching_authmethod**(*backend*)

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 (odelist, varname)
    Bases: django.template.base.Node

    render (context)
        Return the node rendered as a string.

tardis.tardis_portal.templatetags.capture.capture (parser, token)
    {% capture as [foo] %}
```

tardis.tardis_portal.templatetags.dynurl module

```
class tardis.tardis_portal.templatetags.dynurl.DynUrlNode (*args)
    Bases: django.template.base.Node

    render (context)
        Return the node rendered as a string.

tardis.tardis_portal.templatetags.dynurl.dynurl (parser, token)
```

tardis.tardis_portal.templatetags.experiment_tags module

```
tardis.tardis_portal.templatetags.experiment_tags.experiment_authors (experiment,
                                                                    **kwargs)
    Displays an experiment's authors in an experiment list view

tardis.tardis_portal.templatetags.experiment_tags.experiment_download_link (experiment,
                                                                    **kwargs)
    Displays a download link for an experiment in a list view
```

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.pagination` module

`tardis.tardis_portal.templatetags.pagination.pagination` (*data_list*, *paginator*,
page_num, *query_string*)

Generates the series of links to the pages in a paginated list.

`tardis.tardis_portal.templatetags.pagination.paginator_number` (*data_list*, *paginator*,
page_num,
query_string,
page_index)

Generates an individual page index link in a paginated list.

`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.api` package

Submodules

`tardis.tardis_portal.tests.api.test_auth` module

Testing authentication and authorization in the Tastypie-based MyTardis REST API

class `tardis.tardis_portal.tests.api.test_auth.ACLAuthorizationTest` (*methodName='runTest'*)
Bases: `django.test.testcases.TestCase`

class `tardis.tardis_portal.tests.api.test_auth.MyTardisAuthenticationTest` (*methodName='runTest'*)
Bases: `tardis.tardis_portal.tests.api.MyTardisResourceTestCase`

`test_apikey_authentication()`

`test_bad_credentials()`

tardis.tardis_portal.tests.api.test_datafile_metadata_resources module

Testing the DatafileParameter and DatafileParameterSet resources in MyTardis's Tastypie-based REST API

```

class tardis.tardis_portal.tests.api.test_datafile_metadata_resources.DatafileParameterReso
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.api.test_datafile_metadata_resources.DatafileParameterSetI
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

```

tardis.tardis_portal.tests.api.test_datafile_resource module

Testing the DataFile resource in MyTardis's Tastypie-based REST API

```

class tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResourceTest (methodName=
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

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

    test_create_df_for_staging ()

    test_download_file ()
        Re-run the upload test in order to create a verified file to download - it will be verified immediately because
        CELERY_ALWAYS_EAGER is True in test_settings.py

        Then download the file, check the HTTP status code and check the file content.

    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 ()

```

tardis.tardis_portal.tests.api.test_dataset_metadata_resources module

Testing the DatasetParameter and DatasetParameterSet resources in MyTardis's Tastypie-based REST API

```

class tardis.tardis_portal.tests.api.test_dataset_metadata_resources.DatasetParameterResou
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.api.test_dataset_metadata_resources.DatasetParameterSetRes
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

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

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

    test_create_dataset_pset ()
        Test creating a dataset parameter set

    test_create_dataset_pset_no_auth ()
        Test attempting to create a dataset parameter set without access

    test_post_dataset_with_params ()
        Test creating a dataset with metadata

```

tardis.tardis_portal.tests.api.test_dataset_resource module

Testing the Dataset resource in MyTardis's Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

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

    test_get_child_dir_nodes()

    test_get_child_dir_nodes_no_files_in_root_dir()

    test_get_dataset_files()

    test_get_dataset_filter_instrument()

    test_get_dataset_no_instrument()

    test_get_dataset_with_instrument()

    test_get_root_dir_nodes()

    test_post_dataset()
```

tardis.tardis_portal.tests.api.test_experiment_metadata_resources module

Testing the ExperimentParameter and ExperimentParameterSet resources in MyTardis's Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_experiment_metadata_resources.ExperimentParameterSetResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.api.test_experiment_metadata_resources.ExperimentParameterResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase
```

tardis.tardis_portal.tests.api.test_experiment_resource module

Testing the Experiment resource in MyTardis's Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_experiment_resource.ExperimentResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

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

    test_get_experiment()

    test_get_experiment_author()

    test_post_experiment()
```

tardis.tardis_portal.tests.api.test_facility_resource module

Testing the Facility resource in MyTardis's Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_facility_resource.FacilityResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.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 `facilities_managed_by()` via the API

```
test_get_facility_by_name()
```

`tardis.tardis_portal.tests.api.test_group_resource` module

Testing the Group resource in MyTardis’s Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_group_resource.GroupResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase
```

```
test_get_group_by_id()
```

```
test_get_group_by_name()
```

`tardis.tardis_portal.tests.api.test_instrument_resource` module

Testing the Instrument resource in MyTardis’s Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase
```

```
test_get_instrument_by_id()
```

```
test_get_instrument_by_name()
```

```
test_post_instrument()
```

```
test_rename_instrument()
```

```
test_unauthorized_instrument_access_attempt()
```

`tardis.tardis_portal.tests.api.test_list_api_endpoints` module

Testing listing all of the endpoints in MyTardis’s Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_list_api_endpoints.ListEndpointsTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase
```

```
test_list_endpoints()
```

`tardis.tardis_portal.tests.api.test_replica_resource` module

Testing the Replica resource in MyTardis’s Tastypie-based REST API

“Replica” was the old name for what is now known as a `DataFileObject` in MyTardis. The API v1 endpoint is still `/api/v1/replica/`

```
class tardis.tardis_portal.tests.api.test_replica_resource.ReplicaResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase
```

tardis.tardis_portal.tests.api.test_schema_resource module

Testing the Schema resource in MyTardis's Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_schema_resource.SchemaResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

    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_schema_by_id()

    test_get_schema_by_namespace()
```

tardis.tardis_portal.tests.api.test_serializer module

Testing the serializer in the Tastypie-based MyTardis REST API

```
class tardis.tardis_portal.tests.api.test_serializer.SerializerTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    test_debug_serializer()

    test_pretty_serializer()
```

tardis.tardis_portal.tests.api.test_storagebox_resources module

Testing the StorageBox resources in MyTardis's Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_storagebox_resources.StorageBoxAttributeResourceTest
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

    test_get_storage_box_attr_list_from_box_id()

class tardis.tardis_portal.tests.api.test_storagebox_resources.StorageBoxOptionResourceTest
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

    test_get_storage_box_option_by_id()

    test_get_storage_box_option_list_from_box_id()

class tardis.tardis_portal.tests.api.test_storagebox_resources.StorageBoxResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

    test_get_storage_box_by_id()
```

tardis.tardis_portal.tests.api.test_user_resource module

Testing the User resource in MyTardis's Tastypie-based REST API

```
class tardis.tardis_portal.tests.api.test_user_resource.UserResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

    test_get_user_by_id()

    test_get_user_by_username()
```

Module contents

Testing the tastypie-based mytardis api

class tardis.tardis_portal.tests.api.**MyTardisResourceTestCase** (*methodName='runTest'*)
 Bases: tastypie.test.ResourceTestCaseMixin, django.test.testcases.TestCase

abstract class without tests to combine common settings in one place

get_admin_credentials ()

get_apikey_credentials ()

get_credentials ()

A convenience method for the user as a way to shorten up the often repetitious calls to create the same authentication.

Raises NotImplementedError by default.

Usage:

```
class MyResourceTestCase (ResourceTestCase) :
    def get_credentials (self) :
        return self.create_basic ('daniel', 'pass')

    # Then the usual tests...
```

setUp ()

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

tardis.tardis_portal.tests.auth package

Module contents

tardis.tardis_portal.tests.management package

Submodules

tardis.tardis_portal.tests.management.test_collectstatic module

class tardis.tardis_portal.tests.management.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.management.test_createuser module

class tardis.tardis_portal.tests.management.test_createuser.**CreateUserTestCase** (*methodName='runTest'*)
 Bases: django.test.testcases.TestCase

testInteractive()

Just test that we can run `./manage.py createuser` without any runtime exceptions by mocking the `raw_input` username and email entry

testNoInput()

Just test that we can run `./manage.py createuser --username testuser1 --email testuser1@example.com --noinput` without any runtime exceptions

tardis.tardis_portal.tests.management.test_dumpschemas module

class `tardis.tardis_portal.tests.management.test_dumpschemas.DumpSchemasTestCase` (*methodName=*

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.

testDumpSchemas()

Just test that we can run `./manage.py dumpschemas` without any runtime exceptions

tardis.tardis_portal.tests.management.test_loadschemas module

class `tardis.tardis_portal.tests.management.test_loadschemas.LoadSchemasTestCase` (*methodName=*

Bases: `django.test.testcases.TestCase`

testLoadSchemas()

Test that we can run `./manage.py loadschemas tardis/tardis_portal/fixtures/jeol_metadata_schema.json`

tardis.tardis_portal.tests.management.test_rmexperiment module

class `tardis.tardis_portal.tests.management.test_rmexperiment.RmExperimentTestCase` (*methodName=*

Bases: `django.test.testcases.TestCase`

testList()

testRemove()

Module contents

This package contains tests relating to management commands, i.e. command-line tools invoked with `manage.py`

tardis.tardis_portal.tests.models package

Submodules

tardis.tardis_portal.tests.models.test_authors module

`test_authors.py`

```
class tardis.tardis_portal.tests.models.test_authors.AuthorTestCase (methodName='runTest')
    Bases: tardis.tardis_portal.tests.models.ModelTestCase

    test_authors ()
```

tardis.tardis_portal.tests.models.test_datafile module

test_datafile.py

```
class tardis.tardis_portal.tests.models.test_datafile.DataFileTestCase (methodName='runTest')
    Bases: tardis.tardis_portal.tests.models.ModelTestCase

    test_datafile (mock_send_task)
```

tardis.tardis_portal.tests.models.test_dataset module

test_dataset.py

```
class tardis.tardis_portal.tests.models.test_dataset.DatasetTestCase (methodName='runTest')
    Bases: tardis.tardis_portal.tests.models.ModelTestCase

    test_dataset ()

    test_get_dir_tuples ()
```

tardis.tardis_portal.tests.models.test_dfo module

tests_datafileobject.py

```
class tardis.tardis_portal.tests.models.test_dfo.DataFileObjectTestCase (methodName='runTest')
    Bases: tardis.tardis_portal.tests.models.ModelTestCase

    test_deleting_dfo_without_uri ()
```

tardis.tardis_portal.tests.models.test_experiment module

test_experiment.py

```
class tardis.tardis_portal.tests.models.test_experiment.ExperimentTestCase (methodName='runTest')
    Bases: tardis.tardis_portal.tests.models.ModelTestCase

    test_experiment ()
```

tardis.tardis_portal.tests.models.test_generate_api_key module

test_generate_api_key.py

```
class tardis.tardis_portal.tests.models.test_generate_api_key.ApiKeyTestCase (methodName='runTest')
    Bases: tardis.tardis_portal.tests.models.ModelTestCase

    test_create_user_automatically_generate_api_key ()
        Verify that create a new user will generate an api_key automatically
```

tardis.tardis_portal.tests.models.test_instrument module

test_instrument.py

```
class tardis.tardis_portal.tests.models.test_instrument.InstrumentTestCase (methodName='runTest')
    Bases: tardis.tardis_portal.tests.models.ModelTestCase

    test_instrument ()
```

tardis.tardis_portal.tests.models.test_parameter module

test_parameter.py

```
class tardis.tardis_portal.tests.models.test_parameter.ParameterTestCase (methodName='runTest')
    Bases: tardis.tardis_portal.tests.models.ModelTestCase

    test_parameter ()
```

Module contents

tests/models/__init__.py <http://docs.djangoproject.com/en/dev/topics/testing/>

```
class tardis.tardis_portal.tests.models.ModelTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

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

tardis.tardis_portal.tests.views package

Submodules

tardis.tardis_portal.tests.views.test_auth_views module

test_auth_views.py

Tests for view methods relating to users, groups, access controls and authorization

```
class tardis.tardis_portal.tests.views.test_auth_views.ManageAccountTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    test_manage_account (mock_webpack_get_bundle)

class tardis.tardis_portal.tests.views.test_auth_views.RightsTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    test_rights_require_valid_owner ()

class tardis.tardis_portal.tests.views.test_auth_views.UserGroupListsTestCase (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_group_list ()
    test_get_user_list ()
class tardis.tardis_portal.tests.views.test_auth_views.UserListTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
    User lists are used for autocompleting the user-to-share-with field when granting access to an experiment
    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_user_list ()

```

tardis.tardis_portal.tests.views.test_contextual_views module

test_contextual_views.py

Tests for view methods supplying context data to templates

```

class tardis.tardis_portal.tests.views.test_contextual_views.ContextualViewTest (methodName='runTest')
    Bases: django.test.testcases.TestCase
    setUp ()
        setting up essential objects, copied from tests above
    tearDown ()
        Hook method for deconstructing the test fixture after testing it.
    test_details_display ()
        test display of view for an existing schema and no display for an undefined one.

```

tardis.tardis_portal.tests.views.test_experiment_views module

test_experiment_views.py

Tests for view methods relating to experiments

```

class tardis.tardis_portal.tests.views.test_experiment_views.ExperimentTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
    setUp ()
        Hook method for setting up the test fixture before exercising it.
    test_create_and_edit (mock_webpack_get_bundle)
    test_dataset_json ()

```

tardis.tardis_portal.tests.views.test_template_contexts module

test_template_contexts.py

Tests for view methods supplying context data to templates

```

class tardis.tardis_portal.tests.views.test_template_contexts.ExperimentListsTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

```

```
    setUp ()
        setting up essential objects, copied from tests above

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

    test_mydata_view (mock_webpack_get_bundle)
        Test My Data view

    test_shared_view (mock_webpack_get_bundle)
        Test Shared view

class tardis.tardis_portal.tests.views.test_template_contexts.ViewTemplateContextsTest (meth
    Bases: django.test.testcases.TestCase

    setUp ()
        setting up essential objects, copied from tests above

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

    test_dataset_view (mock_webpack_get_bundle)
        test some context parameters for a dataset view

    test_experiment_view (mock_webpack_get_bundle)
        test some template context parameters for an experiment view
```

tardis.tardis_portal.tests.views.test_upload_views module

test_upload_views.py

Tests for view methods relating to uploads

```
class tardis.tardis_portal.tests.views.test_upload_views.UploadTestCase (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_file_upload ()

    test_upload_complete ()
```

Module contents

Tests for view methods in tardis/tardis_portal/views/

Submodules

tardis.tardis_portal.tests.ldap_ldif module

tardis.tardis_portal.tests.slapped module

Utilities for starting up a test slapd server and talking to it with ldapsearch/ldapadd.


```
class tardis.tardis_portal.tests.slapped.Slapped
```

```
    Bases: object
```

Controller class for a slapped instance, OpenLDAP's server.

This class creates a temporary data store for slapped, 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 slapped 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/tmpq1slgamma'
```

```
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 slapped.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 slapped instance, passing it the ldif content

```
ldapsearch (base=None, filter='(objectClass=*)', attrs=[], scope='sub', extra_args=[])
```

```
restart ()
```

Restarts the slapped server; ERASING previous content. Starts the server even if it isn't already running.

```
set_debug ()
```

```
set_dn_suffix (dn)
```

```
set_port (port)
```

```
set_root_cn (cn)
```

```
set_root_password (pw)
```

```
set_slapped_debug_level (level)
```

```
set_tmpdir (path)
```

```
start ()
```

Starts the slapped 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_authentication module

Created on 19/01/2011

```
class tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase(methodName='runTest')
```

Bases: django.test.testcases.TestCase

setUp()

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

```
testManageAuthMethods(mock_webpack_get_bundle)
```

```
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.

```
testCantEditLockedExperiment(mock_webpack_get_bundle)
```

```
testChangeUserPermissions(mock_webpack_get_bundle)
```

```
testOwnedExperiments()
```

```
testReadAccess(mock_webpack_get_bundle)
```

```
testWriteAccess(mock_webpack_get_bundle)
```

```
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 ()

class tardis.tardis_portal.tests.test_authservice.MockAuthProvider
    Bases: object

    authenticate (request)

class tardis.tardis_portal.tests.test_authservice.MockGroupProvider
    Bases: 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.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_copy_move module

test_copy_move.py

Test copying and moving a file to another storage box

```

class tardis.tardis_portal.tests.test_copy_move.CopyMoveTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    Test copying and moving files between storage boxes

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

```

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

test_cache()
Test caching a file from a slow-access storage box

test_copy()
Test copying a file to another storage box

test_move()
Test moving a file to another storage box

tardis.tardis_portal.tests.test_df_save_metadata module

class tardis.tardis_portal.tests.test_df_save_metadata.**DatafileSaveMetadataTestCase** (*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_df_save_metadata()
Test the DataFile metadata saving task used by the post-save filters microservice

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 (*mock_webpack_get_bundle*)

testView (*mock_webpack_get_bundle*)

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='runTest'*)
Bases: tastypie.test.ResourceTestCaseMixin, 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_download_apikey()

tardis.tardis_portal.tests.test_facility_overview module

Tests relating to facility overview

```

class tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

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

    test_facility_overview_datafile_list ()

    test_facility_overview_experiments ()
        Despite the name of the test_facility_overview_experiments method, it actually returns a JSON list of
        datasets (not experiments)

```

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 ()

```

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_parameters module

test_parameters.py

```
class tardis.tardis_portal.tests.test_parameters.ParametersTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    Tests for the different parameter types, defined in the tardis.tardis_portal.models.parameters.ParameterName
    class:
```

NUMERIC, STRING, URL, LINK, FILENAME, DATETIME, LONGSTRING and JSON

setUp()

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

tearDown()

Hook method for deconstructing the test fixture after testing it.

test_experiment_parameter_get()

Test the Parameter class's get() method which should return an appropriate string representation of the parameter which depends on the parameter's data type:

NUMERIC, STRING, URL, LINK, FILENAME, DATETIME, LONGSTRING or JSON

test_image_filename_parameters()

When a FILENAME parameter refers to a thumbnil image file, MyTardis can generate a HTML for displaying that image.

This method tests the generation of the image HTML

test_link_urls()

Test URLs generated for parameters which link to MyTardis model records

test_permissions_checks()

Test permissions checks, used by tardis.tardis_portal.auth.authorisation

tardis.tardis_portal.tests.test_parametersets module

test_parametersets.py <http://docs.djangoproject.com/en/dev/topics/testing/>

class tardis.tardis_portal.tests.test_parametersets.**EditParameterSetTestCase** (*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_add_datafile_params()

test_add_dataset_params()

test_add_experiment_params()

test_edit_datafile_params()

test_edit_dataset_params()

test_edit_experiment_params()

class tardis.tardis_portal.tests.test_parametersets.**ParameterSetManagerTestCase** (*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_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_parameterset_as_string()

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)
        tardis.test_settings adds this to the template dirs: tardis/tardis_portal/tests/rifcs/

    is_schema_valid(experiment)

class tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase(methodName='runTest')
    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_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_get_receiving_box()

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)
    Bases: object

    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_create_token (mock_webpack_get_bundle)

    test_default_expiry ()

    test_get_session_expiry ()

```

```
test_get_session_expiry_expired_token()
test_get_session_expiry_near_expiry()
test_is_expired()
test_retrieve_access_list_tokens()
test_save_with_random_token()
test_save_with_random_token_failures()
test_save_with_random_token_gives_up()
test_token_delete(mock_webpack_get_bundle)
urls = 'tardis.tardis_portal.tests.urls'
```

tardis.tardis_portal.tests.tests module

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

```
class tardis.tardis_portal.tests.tests.UserInterfaceTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
    test_login()
    test_root(mock_webpack_get_bundle)
    test_urls(mock_webpack_get_bundle)
    test_urls_with_some_content(mock_webpack_get_bundle)
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

views that perform some action and don't return anything very useful

tardis.tardis_portal.views.ajax_json module

views that return JSON data

```
tardis.tardis_portal.views.ajax_json.get_experiment_list(request)
tardis.tardis_portal.views.ajax_json.retrieve_licenses(request)
```

tardis.tardis_portal.views.ajax_pages module

views that return HTML that is injected into pages

```
tardis.tardis_portal.views.ajax_pages.retrieve_owned_exps_list(request, template_name='tardis_portal/ajax/exps_list.html')
tardis.tardis_portal.views.ajax_pages.retrieve_shared_exps_list(request, template_name='tardis_portal/ajax/exps_list.html')
```

tardis.tardis_portal.views.authentication module

views that have to do with authentication

```
tardis.tardis_portal.views.authentication.create_user(request)
tardis.tardis_portal.views.authentication.login(request)
    handler for login page
tardis.tardis_portal.views.authentication.logout(request)
tardis.tardis_portal.views.authentication.manage_auth_methods(request)
    Manage the user's authentication methods using AJAX.
tardis.tardis_portal.views.authentication.manage_user_account(request)
tardis.tardis_portal.views.authentication.rcauth(request)
```

tardis.tardis_portal.views.authorisation module

views that have to do with authorisations

```
tardis.tardis_portal.views.authorisation.create_group(request)
tardis.tardis_portal.views.authorisation.manage_groups(request)
tardis.tardis_portal.views.authorisation.retrieve_access_list_group_readonly(request, experiment_id)
tardis.tardis_portal.views.authorisation.retrieve_access_list_user_readonly(request, experiment_id)
tardis.tardis_portal.views.authorisation.retrieve_group_list(request)
tardis.tardis_portal.views.authorisation.retrieve_group_list_by_user(request)
tardis.tardis_portal.views.authorisation.retrieve_group_userlist_readonly(request, group_id)
tardis.tardis_portal.views.authorisation.retrieve_user_list(request)
tardis.tardis_portal.views.authorisation.share(request, experiment_id)
    Choose access rights and licence.
tardis.tardis_portal.views.authorisation.token_delete(request, token_id)
```

tardis.tardis_portal.views.facilities module

views relevant for facilities and the facility view

`tardis.tardis_portal.views.facilities.dataset_aggregate_info(dataset)`

`tardis.tardis_portal.views.facilities.datetime_to_us(dt)`

The datetime objects are kept as None if they aren't set, otherwise they're converted to milliseconds so AngularJS can format them nicely.

`tardis.tardis_portal.views.facilities.facility_overview_data_count(request, facility_id)`

returns the total number of datasets for pagination in json format

`tardis.tardis_portal.views.facilities.facility_overview_datafile_list(dataset)`

`tardis.tardis_portal.views.facilities.facility_overview_dataset_detail(request, dataset_id)`

`tardis.tardis_portal.views.facilities.facility_overview_experiments(request, facility_id, start_index, end_index)`

json facility datasets

`tardis.tardis_portal.views.facilities.facility_overview_facilities_list(request)`

json list of facilities managed by the current user

tardis.tardis_portal.views.images module

views that return images or route to images

`tardis.tardis_portal.views.images.load_datafile_image(request, parameter_id)`

`tardis.tardis_portal.views.images.load_dataset_image(request, parameter_id)`

`tardis.tardis_portal.views.images.load_experiment_image(request, parameter_id)`

`tardis.tardis_portal.views.images.load_image(request, parameter)`

tardis.tardis_portal.views.machine module

views that return data useful only to other machines (but not JSON)

`tardis.tardis_portal.views.machine.site_settings(request)`

tardis.tardis_portal.views.pages module

views that render full pages

class `tardis.tardis_portal.views.pages.DatasetView(**kwargs)`

Bases: `django.views.generic.base.TemplateView`

find_custom_view_override(*request, dataset*)

Determines if any custom view overrides have been defined in settings.DATASET_VIEWS and returns

the view function if a match to one the schemas for the dataset is found. (DATASET_VIEWS is a list of (schema_namespace, view_function) tuples).

Parameters

- **request** –
- **dataset** –

Returns

Return type

get (*request*, *args, **kwargs)
View an existing dataset.

This default view can be overridden by defining a dictionary DATASET_VIEWS in settings.

Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **args** (*list*) –
- **kwargs** (*dict*) –

Returns The Django response object

Return type `django.http.HttpResponse`

get_context_data (*request*, *dataset*, **kwargs)
Prepares the values to be passed to the default dataset view, respecting authorization rules. Returns a dict of values (the context).

Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **dataset** (`tardis.tardis_portal.models.dataset.Dataset`) – the Dataset model instance
- **kwargs** (*dict*) –

Returns A dictionary of values for the view/template.

Return type `dict`

```
template_name = 'tardis_portal/view_dataset.html'
```

```
class tardis.tardis_portal.views.pages.ExperimentView(**kwargs)
Bases: django.views.generic.base.TemplateView
```

```
find_custom_view_override(request, experiment)
```

get (*request*, *args, **kwargs)
View an existing experiment.

This default view can be overridden by defining a dictionary EXPERIMENT_VIEWS in settings.

Parameters

- **request** (`django.http.HttpRequest`) – a HTTP Request instance
- **args** (*list*) –
- **kwargs** (*dict*) – In kwargs: param int experiment_id: the ID of the experiment

Returns an HttpResponse

Return type `django.http.HttpResponse`

get_context_data (*request*, *experiment*, ***kwargs*)

Prepares the values to be passed to the default experiment view, respecting authorization rules. Returns a dict of values (the context).

Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **experiment** (`tardis.tardis_portal.models.experiment.Experiment`) – the experiment model instance
- **kwargs** (*dict*) – kwargs

Returns A dictionary of values for the view/template.

Return type `dict`

`template_name = 'tardis_portal/view_experiment.html'`

class `tardis.tardis_portal.views.pages.IndexView` (***kwargs*)

Bases: `django.views.generic.base.TemplateView`

get (*request*, **args*, ***kwargs*)

The index view, intended to render the front page of the MyTardis site listing recent experiments.

This default view can be overridden by defining a dictionary `INDEX_VIEWS` in settings which maps `SITE_ID`'s or domain names to an alternative view function (similar to the `DATASET_VIEWS` or `EXPERIMENT_VIEWS` overrides).

Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **args** (*list*) –
- **kwargs** (*dict*) –

Returns The Django response object

Return type `django.http.HttpResponse`

get_context_data (**args*, ***kwargs*)

`template_name = 'tardis_portal/index.html'`

`tardis.tardis_portal.views.pages.about` (*request*)

`tardis.tardis_portal.views.pages.add_dataset` (*request*, *experiment_id*)

`tardis.tardis_portal.views.pages.create_experiment` (*request*, *template_name='tardis_portal/create_experiment.html'*)

Create a new experiment view.

Parameters

- **request** (`django.http.HttpRequest`) – a HTTP Request instance
- **template_name** (*string*) – the path of the template to render

Returns an `HttpResponse`

Return type `django.http.HttpResponse`

`tardis.tardis_portal.views.pages.edit_dataset` (*request*, *dataset_id*)

`tardis.tardis_portal.views.pages.facility_overview` (*request*)
summary of experiments in a facility

`tardis.tardis_portal.views.pages.healthz(request)`
returns that the server is alive

`tardis.tardis_portal.views.pages.my_data(request)`
show owned data with credential-based access

`tardis.tardis_portal.views.pages.public_data(request)`
list of public experiments

`tardis.tardis_portal.views.pages.shared(request)`
show shared data with credential-based access

`tardis.tardis_portal.views.pages.site_routed_view(request, _default_view, _site_mappings, *args, **kwargs)`

Allows a view to be overridden based on the Site (eg domain) for the current request. Takes a default fallback view (`_default_view`) and a dictionary mapping Django Sites (domain name or int `SITE_ID`) to views. If the current request matches a Site in the dictionary, that view is used instead of the default.

The intention is to define `{site: view}` mappings in `settings.py`, and use this wrapper view in `urls.py` to allow a single URL to be routed to different views depending on the Site in the request.

Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **_default_view** (`types.FunctionType` | `str`) – The default view if no Site in `_site_mappings` matches the current Site.
- **_site_mappings** (`dict`) – A dictionary mapping Django sites to views (sites are specified as either a domain name `str` or int `SITE_ID`).
- **args** –
- **kwargs** –

Returns A view function

Return type `types.FunctionType`

`tardis.tardis_portal.views.pages.stats(request)`

`tardis.tardis_portal.views.pages.use_rapid_connect(fn)`

A decorator that adds AAF Rapid Connect settings to a `get_context_data` method.

Parameters **fn** (`types.FunctionType`) – A `get_context_data` function/method.

Returns A `get_context_data` function that adds `RAPID_CONNECT_*` keys to its output context.

Return type `types.FunctionType`

`tardis.tardis_portal.views.pages.user_guide(request)`

`tardis.tardis_portal.views.parameters` module

views to do with metadata, parameters etc. Mostly ajax page inclusions

`tardis.tardis_portal.views.parameters.add_datafile_par(request, datafile_id)`

`tardis.tardis_portal.views.parameters.add_dataset_par(request, dataset_id)`

`tardis.tardis_portal.views.parameters.add_experiment_par(request, experiment_id)`

`tardis.tardis_portal.views.parameters.add_par(request, parentObject, otype, stype)`

`tardis.tardis_portal.views.parameters.edit_datafile_par(request, parameterset_id)`

```
tardis.tardis_portal.views.parameters.edit_dataset_par(request, parameterset_id)
tardis.tardis_portal.views.parameters.edit_experiment_par(request,      parameter-
                                                           set_id)
tardis.tardis_portal.views.parameters.edit_parameters(request, parameterset, otype)
```

tardis.tardis_portal.views.upload module

views for uploading files via HTTP

```
tardis.tardis_portal.views.upload.upload_complete(request,      tem-
                                                    plate_name='tardis_portal/upload_complete.html')
```

The ajax-loaded result of a file being uploaded

Parameters

- **request** (`django.http.HttpRequest`) – a HTTP Request instance
- **template_name** (`string`) – the path of the template to render

Returns an `HttpResponse`

Return type `django.http.HttpResponse`

tardis.tardis_portal.views.utils module

helper functions used by other views

```
class tardis.tardis_portal.views.utils.HttpResponseMethodNotAllowed(*args,
                                                                    **kwargs)
```

Bases: `django.http.response.HttpResponse`

status_code = 303

```
class tardis.tardis_portal.views.utils.HttpResponseSeeAlso(redirect_to,      *args,
                                                            **kwargs)
```

Bases: `django.http.response.HttpResponseRedirect`

status_code = 303

```
tardis.tardis_portal.views.utils.feedback(request)
```

```
tardis.tardis_portal.views.utils.get_dataset_info(dataset, include_thumbnail=False,
                                                    exclude=None)
```

```
tardis.tardis_portal.views.utils.remove_csrf_token(request)
```

rather than fixing the form code that loops over all POST entries indiscriminately, I am removing the csrf token with this hack. This is only required in certain form code and can be removed should this ever be fixed

Module contents

importing all views files here, so that any old code will work as expected, when importing from `tardis.tardis_portal.views`

Submodules

tardis.tardis_portal.ParameterSetManager module

```
class tardis.tardis_portal.ParameterSetManager.ParameterSetManager (parameterset=None,
                                                                    parentOb-
                                                                    ject=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 (params_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='id_%s', pre-
                                                            fix=None, initial=None,
                                                            error_class=<class
                                                            'django.forms.utils.ErrorList'>,
                                                            label_suffix=None,
                                                            empty_permitted=False,
                                                            instance=None,
                                                            use_required_attribute=None,
                                                            renderer=None)

    Bases: django.forms.models.ModelForm
```

```
class Meta
    Bases: object
    fields = '__all__'
    model
        alias of tardis.tardis_portal.models.datafile.DataFileObject
    widgets = {'uri': <django.forms.widgets.TextInput object>}
base_fields = {'datafile': <django.forms.models.ModelChoiceField object>, 'last_verif
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='id_%s',  prefix=None,
                                                    initial=None, error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=None,
                                                    empty_permitted=False,
                                                    instance=None,
                                                    use_required_attribute=None,
                                                    renderer=None)

    Bases: django.forms.models.ModelForm

class Meta
    Bases: object
    fields = '__all__'
    model
        alias of tardis.tardis_portal.models.datafile.DataFile
    widgets = {'directory': <django.forms.widgets.TextInput object>}
base_fields = {'created_time': <django.forms.fields.DateTimeField object>, 'dataset':
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,      ad-
                                                              min_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,          ad-
                                                              min_site)
    Bases: django.contrib.admin.options.ModelAdmin
    inlines = [<class 'tardis.tardis_portal.admin.ExperimentParameterInline'>]
    media
class tardis.tardis_portal.admin.FacilityAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    media
    search_fields = ['name']
class tardis.tardis_portal.admin.FreeTextSearchFieldAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    media
class tardis.tardis_portal.admin.InstrumentAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    media
    search_fields = ['name']
class tardis.tardis_portal.admin.InstrumentParameterInline (parent_model,      ad-
                                                              min_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,          ad-
                                                              min_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 = ['__str__', 'pluginId', 'entityId', 'canRead', 'canWrite', 'canDelete',

```

```
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
    search_fields = ['name']
class tardis.tardis_portal.admin.StorageBoxAttributeInline (parent_model,      ad-
                                                             min_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='id_%s',
                                                            pre-
                                                            fix=None, ini-
                                                            tial=None, er-
                                                            ror_class=<class
                                                            'django.forms.utils.ErrorList'>,
                                                            la-
                                                            bel_suffix=None,
                                                            empty_permitted=False,
                                                            instance=None,
                                                            use_required_attribute=None,
                                                            ren-
                                                            derer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object
    fields = '__all__'
    model
        alias of tardis.tardis_portal.models.storage.StorageBoxAttribute
    widgets = {'key': <django.forms.widgets.TextInput object>, 'value': <django.forms
base_fields = {'key': <django.forms.fields.CharField object>, 'storage_box': <django
declared_fields = {}
media

class tardis.tardis_portal.admin.StorageBoxForm(data=None,          files=None,
auto_id='id_%s',          prefix=None,
initial=None,          error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
instance=None,
use_required_attribute=None,    ren-
derer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object
    fields = '__all__'
    model
        alias of tardis.tardis_portal.models.storage.StorageBox
    widgets = {'description': <django.forms.widgets.TextInput object>, 'django_storage
base_fields = {'description': <django.forms.fields.CharField object>, 'django_storage
declared_fields = {}
media

class tardis.tardis_portal.admin.StorageBoxOptionInline(parent_model,          ad-
min_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='id_%s',
                                                             prefix=None,    ini-
                                                             tial=None,        er-
                                                             ror_class=<class
                                                             'django.forms.utils.ErrorList'>,
                                                             label_suffix=None,
                                                             empty_permitted=False,
                                                             instance=None,
                                                             use_required_attribute=None,
                                                             renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = '__all__'

    model
        alias of tardis.tardis_portal.models.storage.StorageBoxOption

    widgets = {'key': <django.forms.widgets.TextInput object>, 'value': <django.forms
base_fields = {'key': <django.forms.fields.CharField object>, 'storage_box': <django
declared_fields = {}

media

class tardis.tardis_portal.admin.UserAuthenticationAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    media

    search_fields = ['username', 'authenticationMethod', 'userProfile__user__username']
```

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)
        Returns either True if the user is allowed to create the object in question or throw Unauthorized if
        they are not.

        Returns True by default.
```

create_list (*object_list, bundle*)
 Unimplemented, as TastyPie never creates entire new lists, but present for consistency & possible extension.

delete_detail (*object_list, bundle*)
 Returns either `True` if the user is allowed to delete the object in question or throw `Unauthorized` if they are not.
 Returns `True` by default.

delete_list (*object_list, bundle*)
 Returns a list of all the objects a user is allowed to delete.
 Should return an empty list if none are allowed.
 Returns the entire list by default.

read_detail (*object_list, bundle*)
 Returns either `True` if the user is allowed to read the object in question or throw `Unauthorized` if they are not.
 Returns `True` by default.

read_list (*object_list, bundle*)
 Returns a list of all the objects a user is allowed to read.
 Should return an empty list if none are allowed.
 Returns the entire list by default.

update_detail (*object_list, bundle*)
 Latest TastyPie requires `update_detail` permissions to be able to create objects. Rather than duplicating code here, we'll just use the same authorization rules we use for `create_detail`.

update_list (*object_list, bundle*)
 Returns a list of all the objects a user is allowed to update.
 Should return an empty list if none are allowed.
 Returns the entire list by default.

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': ('exact', 'startswith')}

object_class
 alias of `tardis.tardis_portal.models.datafile.DataFile`

ordering = ['id', 'filename', 'modification_time']

queryset

resource_name = 'dataset_file'

base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'datafile': <tastypie.fields.FileField object>}

declared_fields = {'datafile': <tastypie.fields.FileField object>, 'dataset': <tastypie.fields.CharField object>}

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*)

A hook to allow an initial manipulation of data before all methods/fields have built out the hydrated data.

Useful if you need to access more than one hydrated field or want to annotate on additional data.

Must return the modified 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*)

Creates a new resource/object with the provided data.

Calls `obj_create` with the provided data and returns a response with the new resource's location.

If a new resource is created, return `HttpCreated` (201 Created). If `Meta.always_return_data = True`, there will be a populated body of serialized data.

prepend_urls ()

A hook for adding your own URLs or matching before the default 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>, 'id': <tastypie.fields.IntegerField object>}

declared_fields = {'parameterset': <tastypie.fields.ForeignKey object>}

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>, 'id': <tastypie.fields.IntegerField object>}

declared_fields = {'datafile': <tastypie.fields.ForeignKey object>, 'parameters': <tastypie.fields.ForeignKey object>}

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>, 'id': <tastypie.fields.IntegerField object>}
    declared_fields = {'parameterset': <tastypie.fields.ForeignKey object>}

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>, 'id': <tastypie.fields.IntegerField object>}
        declared_fields = {'dataset': <tastypie.fields.ForeignKey object>, 'parameters': <tastypie.fields.ManyToManyField object>}

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': ('exact',)}

        object_class
            alias of tardis.tardis_portal.models.dataset.Dataset

        ordering = ['id', 'description']

        queryset

        base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'description': <tastypie.fields.CharField object>}
        declared_fields = {'experiments': <tastypie.fields.ToManyField object>, 'instrument': <tastypie.fields.CharField object>}

    dehydrate (bundle)
        A hook to allow a final manipulation of data once all fields/methods have built out the dehydrated data.

        Useful if you need to access more than one dehydrated field or want to annotate on additional data.

        Must return the modified bundle.

    get_child_dir_files (request, **kwargs)
        Return a list of datafile Ids within a child subdirectory :param request: a HTTP Request instance :type request: django.http.HttpRequest :param kwargs: :return: a list of datafile IDs :rtype: JsonResponse

    get_child_dir_nodes (request, **kwargs)
        Return JSON-serialized list of filenames/folders within a child subdirectory

    get_datafiles (request, **kwargs)

    get_root_dir_nodes (request, **kwargs)
        Return JSON-serialized list of filenames/folders in the dataset's root directory

```

hydrate_m2m(*bundle*)

Create experiment-dataset associations first, because they affect authorization for adding other related resources, e.g. metadata

prepend_urls()

A hook for adding your own URLs or matching before the default URLs.

class tardis.tardis_portal.api.**ExperimentAuthorResource**(*api_name=None*)

Bases: *tardis.tardis_portal.api.MyTardisModelResource*

API for ExperimentAuthors

class Meta

Bases: *tardis.tardis_portal.api.Meta*

always_return_data = True

filtering = {'author': ('exact', 'iexact'), 'email': ('exact', 'iexact'), 'experiment': ('exact', 'iexact')}

object_class

alias of *tardis.tardis_portal.models.experiment.ExperimentAuthor*

ordering = ['id', 'author', 'email', 'order']

queryset

base_fields = {'author': <tastypie.fields.CharField object>, 'email': <tastypie.fields.CharField object>, 'experiment': <tastypie.fields.ForeignKey object>}

declared_fields = {'experiment': <tastypie.fields.ForeignKey object>}

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>, 'id': <tastypie.fields.IntegerField object>, 'parameterset': <tastypie.fields.ForeignKey object>}

declared_fields = {'parameterset': <tastypie.fields.ForeignKey object>}

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>, 'id': <tastypie.fields.IntegerField object>, 'parameters': <tastypie.fields.ManyToManyField object>}

declared_fields = {'experiment': <tastypie.fields.ForeignKey object>, 'parameters': <tastypie.fields.ManyToManyField object>}

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 = ['id', 'title', 'created_time', 'update_time']

    queryset

base_fields = {'approved': <tastypie.fields.BooleanField object>, 'created_by': <tas
declared_fields = {'created_by': <tastypie.fields.ForeignKey object>, 'parameter_sets
dehydrate(bundle)
    A hook to allow a final manipulation of data once all fields/methods have built out the dehydrated data.

    Useful if you need to access more than one dehydrated field or want to annotate on additional data.

    Must return the modified 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.

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

        ordering = ['id', 'name']

        queryset

base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'id': <tastyp
declared_fields = {'manager_group': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.GroupResource(api_name=None)
    Bases: tastypie.resources.ModelResource

    class Meta
        Bases: object

        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 = {'id': <tastypie.fields.IntegerField object>, 'name': <tastypie.fields
    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
        ordering = ['id', 'name']
        queryset

    base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'facility': <
    declared_fields = {'facility': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.LocationResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta
        queryset

    base_fields = {'resource_uri': <tastypie.fields.CharField object>}
    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
        Bases: object
        authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>
        authorization
            Authorisation class for Tastypie.
        object_class = None
```

```

        serializer = <tardis.tardis_portal.api.PrettyJSONSerializer object>
base_fields = {'resource_uri': <tastypie.fields.CharField object>}
declared_fields = {}

class tardis.tardis_portal.api.ObjectACLResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: object

        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
        ordering = ['id']
        queryset

    base_fields = {'aclOwnershipType': <tastypie.fields.IntegerField object>, 'canDelete'
declared_fields = {'content_object': <tastypie.contrib.contenttypes.fields.GenericFor
hydrate (bundle)
    A hook to allow an initial manipulation of data before all methods/fields have built out the hydrated data.
    Useful if you need to access more than one hydrated field or want to annotate on additional data.
    Must return the modified 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>, 'comparison_type': <ta
declared_fields = {'schema': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.ParameterResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    base_fields = {'name': <tastypie.fields.ForeignKey object>, 'resource_uri': <tastypi
declared_fields = {'name': <tastypie.fields.ForeignKey object>, 'value': <tastypie.f
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 = {'resource_uri': <tastypie.fields.CharField object>, 'schema': <tastypie.fields.CharField object>}
declared_fields = {'schema': <tastypie.fields.ForeignKey object>}
hydrate_schema(bundle)

class tardis.tardis_portal.api.PrettyJSONSerializer(formats=None, content_types=None, time_formatting=None,
                                                    content_types=None, date_time_formatting=None)

Bases: tastypie.serializers.Serializer

json_indent = 2

to_json(data, options=None)
    Given some Python data, produces JSON output.

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
    ordering = ['id']
    queryset

base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'datafile': <tastypie.fields.ForeignKey object>}
declared_fields = {'datafile': <tastypie.fields.ForeignKey object>}
dehydrate(bundle)
    A hook to allow a final manipulation of data once all fields/methods have built out the dehydrated data.
    Useful if you need to access more than one dehydrated field or want to annotate on additional data.
    Must return the modified bundle.

hydrate(bundle)
    A hook to allow an initial manipulation of data before all methods/fields have built out the hydrated data.
    Useful if you need to access more than one hydrated field or want to annotate on additional data.
    Must return the modified 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
    ordering = ['id']
    queryset

base_fields = {'hidden': <tastypie.fields.BooleanField object>, 'id': <tastypie.fields.CharField object>}
declared_fields = {}
```

```

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

        ordering = ['id']

        queryset

    base_fields = {'id': <tastypie.fields.IntegerField object>, 'key': <tastypie.fields.
    declared_fields = {'storage_box': <tastypie.fields.ForeignKey object>}

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

        ordering = ['id']

        queryset

    accessible_keys = ['location']

    base_fields = {'id': <tastypie.fields.IntegerField object>, 'key': <tastypie.fields.
    declared_fields = {'storage_box': <tastypie.fields.ForeignKey object>}

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

        ordering = ['id']

        queryset

    base_fields = {'attributes': <tastypie.fields.ToManyField object>, 'description': <t
    declared_fields = {'attributes': <tastypie.fields.ToManyField object>, 'options': <t

class tardis.tardis_portal.api.UserResource (api_name=None)
    Bases: tastypie.resources.ModelResource

    class Meta
        Bases: object

        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>, 'first_name': <tastypie.
declared_fields = {'groups': <tastypie.fields.ManyToManyField object>}
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.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.registration_processor (request)
tardis.tardis_portal.context_processors.single_search_processor (request)
tardis.tardis_portal.context_processors.user_details_processor (request)
tardis.tardis_portal.context_processors.user_menu_processor (request)
```

tardis.tardis_portal.deprecations module

Deprecation warnings

exception tardis.tardis_portal.deprecations.RemovedInMyTardis42Warning
Bases: [DeprecationWarning](#)
Used to raise warnings about deprecated functionality.
Usage:


```
import warnings

warnings.warn(
    "This method will be removed in MyTardis 4.2. "
    "Please use method2 instead.",
    RemovedInMyTardis42Warning
)
```

exception tardis.tardis_portal.deprecations.**RemovedInMyTardis43Warning**
 Bases: `PendingDeprecationWarning`

Used to raise warnings about deprecated functionality.

Usage:

```
import warnings

warnings.warn(
    "This method will be removed in MyTardis 4.3. "
    "Please use method2 instead.",
    RemovedInMyTardis43Warning
)
```

tardis.tardis_portal.download module

download.py

class tardis.tardis_portal.download.**UncachedTarStream**(*mapped_file_objs, filename, do_gzip=False, buffer-size=131072, comp_level=6, http_buffersize=65535*)

Bases: `tarfile.TarFile`

Stream files into a compressed tar stream on the fly

close_gzip()

compress(*buf*)

compute_size()

get_response(*tracker_data=None*)

make_tar()

main tar generator. until python 3 needs to be in one function because 'yield's don't bubble up.

prepare_output(*uc_buf, remainder*)

tarinfo_for_df(*df, name*)

tardis.tardis_portal.download.**classic_mapper**(*rootdir*)

tardis.tardis_portal.download.**download_api_key**(*request*)

tardis.tardis_portal.download.**download_datafile**(*request, datafile_id*)

tardis.tardis_portal.download.**get_download_organizations**()

tardis.tardis_portal.download.**make_mapper**(*organization, rootdir*)

`tardis.tardis_portal.download.streaming_download_datafiles(request)`

takes string parameter “comptype” for compression method. Currently implemented: “tgz” and “tar” The datafiles to be downloaded are selected using “datafile”, “dataset” or “url” parameters. An “expid” parameter may be supplied for use in the download archive name. If “url” is used, the “expid” parameter is also used to limit the datafiles to be downloaded to a given experiment.

`tardis.tardis_portal.download.view_datafile(request, datafile_id)`

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.fields module

class `tardis.tardis_portal.fields.MultiValueCommaSeparatedField(fields, *, require_all_fields=True, **kwargs)`

Bases: `django.forms.fields.MultiValueField`

compress (*data_list*)

Return a single value for the given list of values. The values can be assumed to be valid.

For example, if this MultiValueField was instantiated with fields=(DateField(), TimeField()), this might return a datetime object created by combining the date and time in data_list.

tardis.tardis_portal.forms module

forms module

class `tardis.tardis_portal.forms.AddUserPermissionsForm(data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=None, renderer=None)`

Bases: `django.forms.forms.Form`

base_fields = {'authMethod': <django.forms.fields.CharField object>, 'autocomp_user':

declared_fields = {'authMethod': <django.forms.fields.CharField object>, 'autocomp_us

media

```

class tardis.tardis_portal.forms.CreateUserPermissionsForm (data=None,
                                                            files=None,
                                                            auto_id='id_%s',
                                                            prefix=None,    ini-
                                                            tial=None,        er-
                                                            ror_class=<class
                                                            'django.forms.utils.ErrorList'>,
                                                            label_suffix=None,
                                                            empty_permitted=False,
                                                            field_order=None,
                                                            use_required_attribute=None,
                                                            renderer=None)

Bases: tardis.tardis_portal.forms.RegistrationForm

base_fields = {'authMethod': <django.forms.fields.CharField object>, 'email': <django.forms.fields.CharField object>}
declared_fields = {'authMethod': <django.forms.fields.CharField object>, 'email': <django.forms.fields.CharField object>}
media

class tardis.tardis_portal.forms.DatasetForm (data=None,
                                                files=None,
                                                auto_id='id_%s',    prefix=None,    ini-
                                                tial=None,        error_class=<class
                                                'django.forms.utils.ErrorList'>,
                                                label_suffix=None,
                                                empty_permitted=False,    instance=None,
                                                use_required_attribute=None,    ren-
                                                derer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['description', 'directory', 'instrument']

    model
        alias of tardis.tardis_portal.models.dataset.Dataset

base_fields = {'description': <django.forms.fields.CharField object>, 'directory': <django.forms.fields.CharField object>}
declared_fields = {'description': <django.forms.fields.CharField object>}
media

class tardis.tardis_portal.forms.ExperimentAuthor (data=None,
                                                    files=None,
                                                    auto_id='id_%s',    prefix=None,
                                                    initial=None,    error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=None,
                                                    empty_permitted=False,
                                                    instance=None,
                                                    use_required_attribute=None,
                                                    renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['author', 'institution', 'email', 'order', 'url']

    model
        alias of tardis.tardis_portal.models.experiment.ExperimentAuthor

```

```
base_fields = {'author': <django.forms.fields.CharField object>, 'email': <django.f
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, in-
stance=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: collections.UserDict

    This is a dict wrapper that store the values returned from the tardis.tardis_portal.forms.
    ExperimentForm.save() function. It provides a convenience method for saving the model objects.
```

```
save_m2m()
    {'experiment': experiment, 'experiment_authors': experiment_authors, 'authors': authors, 'datasets':
    datasets, 'datafiles': datafiles}
```

```
class Meta
    Bases: object

    fields = ('title', 'institution_name', 'description')

    model
        alias of tardis.tardis_portal.models.experiment.Experiment
```

```
base_fields = {'description': <django.forms.fields.CharField object>, 'institution_na
declared_fields = {'url': <django.forms.fields.CharField object>}
```

```
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 returned value is dependent on the validity of the nested forms.
```

This validity also takes into account foreign keys that might be dependent on an unsaved model.

Returns validity

Return type `bool`

media

```
save(commit=True)
    Save this form's self.instance object if commit=True. Otherwise, add a save_m2m() method to the form
    which can be called after the instance is saved manually at a later time. Return the model instance.
```

```

class tardis.tardis_portal.forms.ImportParamsForm(data=None, files=None,
auto_id='id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
field_order=None,
use_required_attribute=None,
renderer=None)

Bases: django.forms.forms.Form

base_fields = {'params': <django.forms.fields.FileField object>, 'password': <django
declared_fields = {'params': <django.forms.fields.FileField object>, 'password': <dj
media

class tardis.tardis_portal.forms.LoginForm(*args, **kwargs)
Bases: django.contrib.auth.forms.AuthenticationForm

base_fields = {'password': <django.forms.fields.CharField object>, 'username': <djan
declared_fields = {'password': <django.forms.fields.CharField object>, 'username': <
media

class tardis.tardis_portal.forms.ManageAccountForm(data=None, files=None,
auto_id='id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
instance=None,
use_required_attribute=None,
renderer=None)

Bases: django.forms.models.ModelForm

Form for changing account details.

class Meta
Bases: object

fields = ('first_name', 'last_name', 'email')

model
alias of django.contrib.auth.models.User

base_fields = {'email': <django.forms.fields.EmailField object>, 'first_name': <djan
declared_fields = {}

media

```

```
class tardis.tardis_portal.forms.ManageGroupPermissionsForm (data=None,  
                                                         files=None,  
                                                         auto_id='id_%s',  
                                                         prefix=None,    ini-  
                                                         tial=None,      er-  
                                                         ror_class=<class  
                                                         'django.forms.utils.ErrorList'>,  
                                                         label_suffix=None,  
                                                         empty_permitted=False,  
                                                         field_order=None,  
                                                         use_required_attribute=None,  
                                                         renderer=None)  
  
Bases: django.forms.forms.Form  
  
base_fields = {'adduser': <django.forms.fields.CharField object>, 'admin': <django.f  
  
declared_fields = {'adduser': <django.forms.fields.CharField object>, 'admin': <djan  
  
media  
  
class tardis.tardis_portal.forms.NoInput (attrs=None)  
Bases: django.forms.widgets.Widget  
  
media  
  
render (name, value, attrs=None)  
    Render the widget as an HTML string.  
  
class tardis.tardis_portal.forms.RegisterExperimentForm (data=None,    files=None,  
                                                         auto_id='id_%s',    pre-  
                                                         fix=None,    initial=None,  
                                                         error_class=<class  
                                                         'django.forms.utils.ErrorList'>,  
                                                         label_suffix=None,  
                                                         empty_permitted=False,  
                                                         field_order=None,  
                                                         use_required_attribute=None,  
                                                         renderer=None)  
  
Bases: django.forms.forms.Form  
  
base_fields = {'experiment_owner': <django.forms.fields.CharField object>, 'from_url'  
  
declared_fields = {'experiment_owner': <django.forms.fields.CharField object>, 'from_  
  
media  
  
class tardis.tardis_portal.forms.RegistrationForm (data=None,          files=None,  
                                                         auto_id='id_%s',    prefix=None,  
                                                         initial=None,    error_class=<class  
                                                         'django.forms.utils.ErrorList'>,  
                                                         label_suffix=None,  
                                                         empty_permitted=False,  
                                                         field_order=None,  
                                                         use_required_attribute=None,  
                                                         renderer=None)  
  
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>, 'password1': <django
```

```
clean()
```

Verify 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>, 'password1': <d
```

```
media
```

```
save (profile_callback=None)
```

```
class tardis.tardis_portal.forms.RightsForm (data=None, files=None,
                                             auto_id='id_%s', prefix=None, ini-
                                             tial=None, error_class=<class
                                             'django.forms.utils.ErrorList'>,
                                             label_suffix=None,
                                             empty_permitted=False, instance=None,
                                             use_required_attribute=None, ren-
                                             derer=None)
```

Bases: `django.forms.models.ModelForm`

Form for changing public access and licence.

```
class Meta
```

Bases: `object`

```
fields = ('public_access', 'license', 'legal_text')
```

```
model
```

alias of `tardis.tardis_portal.models.experiment.Experiment`

```
widgets = {'license': <django.forms.widgets.HiddenInput object>}
```

```
base_fields = {'legal_text': <django.forms.fields.CharField object>, 'license': <dja
```

```
clean()
```

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'legal_text': <django.forms.fields.CharField object>}
```

```
media
```

```
class tardis.tardis_portal.forms.StaticField (*, required=True, widget=None,
                                             label=None, initial=None,
                                             help_text="", error_messages=None,
                                             show_hidden_initial=False, valida-
                                             tors=(), localize=False, disabled=False,
                                             label_suffix=None)
```

Bases: `django.forms.fields.Field`

```
clean (value)
```

Validate the given value and return its “cleaned” value as an appropriate Python object. Raise `ValidationError` for any errors.

widgetalias 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.create_parameter_add_form(schema, parentObject, re-
                                                    quest=None)

tardis.tardis_portal.forms.create_parameterset_edit_form(parameterset, re-
                                                         quest=None)

tardis.tardis_portal.forms.getAuthMethodChoices()

tardis.tardis_portal.forms.save_parameter_add_form(schema, parentObject, request)

tardis.tardis_portal.forms.save_parameter_edit_form(parameterset, request)
```

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(*args, **kwargs)

tardis.tardis_portal.iiif.download_info(*args, **kwargs)
```

tardis.tardis_portal.logging_middleware module

```
class tardis.tardis_portal.logging_middleware.LoggingMiddleware(get_response)
    Bases: object

    process_exception(request, exception)

    process_response(request, response)

tardis.tardis_portal.logging_middleware.get_client_ip(request)
```

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 retrieval 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 ()

Return a new QuerySet object. Subclasses can override this method to 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 `use_system_timezone=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 `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: expected datetime, got str instead

```

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.redirect_back_with_error(request, message)`

`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 `
`.

```
tardis.tardis_portal.shortcuts.render_response_index(request, *args, **kwargs)
```

```
tardis.tardis_portal.shortcuts.render_to_file(template, filename, context)
```

Write the output of `render_to_string` to a file.

The `render_to_string()` method returns a unicode string, which can be written to a file with `locale.getpreferredencoding()`, usually UTF-8.

```
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.tasks module

tardis.tardis_portal.util module

```
tardis.tardis_portal.util.get_filesystem_safe_dataset_name(dataset)
```

Given a Dataset, return a filesystem safe string representing the dataset. Useful for filenames for dataset downloads, maybe URLs.

Parameters `dataset` (`tardis.tardis_portal.models.dataset.Dataset`) – A Dataset object.

Returns A filesystem safe string as a Dataset name.

Return type basestring

```
tardis.tardis_portal.util.get_filesystem_safe_experiment_name(experiment)
```

Given an Experiment, return a filesystem safe string representing the experiment. Useful for filenames for experiment downloads, maybe URLs.

Parameters `experiment` (`tardis.tardis_portal.models.experiment.Experiment`) – A Experiment object.

Returns A filesystem safe string as a Experiment name.

Return type basestring

```
tardis.tardis_portal.util.get_local_time(dt)
```

Ensure datetime is timezone-aware and in local time.

If the `USE_TZ` setting in the current dev version of Django comes in, this *should* keep providing correct behaviour.

```
tardis.tardis_portal.util.get_utc_time(dt)
```

Ensure datetime is timezone-aware and in UTC time.

If the `USE_TZ` setting in the current dev version of Django comes in, this *should* keep providing correct behaviour.

```
tardis.tardis_portal.util.split_path(p)
```

tardis.tardis_portal.widgets module

```
class tardis.tardis_portal.widgets.CommaSeparatedInput(attrs=None)
```

Bases: `django.forms.widgets.TextInput`

media

render (*name, value, attrs=None, renderer=None*)

Render the widget as an HTML string.

value_from_datadict (*data, files, name*)

Given a dictionary of data and this widget's name, return the value of this widget or None if it's not provided.

class tardis.tardis_portal.widgets.**Label** (*attrs=None*)

Bases: django.forms.widgets.Widget

media

render (*name, value, attrs=None*)

Render the widget as an HTML string.

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**

Bases: *object*

static write_template_to_dir (*dest_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** (*string*) – The relative path to the Django template to be rendered
- **context** (*django.template.context.Context*) – The Context object (dictionary of variables for template output)

Returns The full path to the created file

Return type string

static write_template_to_file (*prefix_dir, objectprefix, uniqueid, templatepath, context*)

Parameters

- **prefix_dir** (*string*) – The subdirectory off of the OAI_DOCS_PATH to store the resulting document in
- **objectprefix** (*string*) – The name prefix of the resulting file. Files are output in the format prefix-uniqueid.xml
- **uniqueid** (*string*) – The unique ID of the file to be output
- **templatepath** (*string*) – The relative path to the Django template to be rendered
- **context** (*django.template.context.Context*) – The Context object (dictionary of variables for template output)

Returns The full path to the created file

Return type string

static write_xml_to_file (*prefix_dir*, *objectprefix*, *uniqueid*, *xmlstring*)

Parameters

- **prefix_dir** (*string*) – The subdirectory off of the OAI_DOCS_PATH to store the resulting document in
- **objectprefix** (*string*) – The name prefix of the resulting file. Files are output in the format prefix-uniqueid.xml
- **uniqueid** (*string*) – The unique ID of the file to be output
- **xmlstring** (*string*) – The relative path to the Django template to be rendered

Returns The full path to the created file

Return type string

```
tardis.tardis_portal.xmlwriter.logger = <Logger tardis.tardis_portal.xmlwriter (WARNING)>  
XML Writer
```

A set of static methods for writing xml files.

Module contents

importing all views files here, so that any old code will work as expected, when importing from
tardis.tardis_portal.views

models/__init__.py

tardis.urls package

Submodules

tardis.urls.accounts module

Accounts URLs

tardis.urls.ajax module

AJAX URLs

tardis.urls.api module

URLs for MyTardis's RESTful API

tardis.urls.core module

Core URLs

tardis.urls.datafile module

DataFile URLs

tardis.urls.dataset module

Dataset URLs

tardis.urls.display module

Display URLs

tardis.urls.download module

Download URLs

tardis.urls.experiment module

URLs for experiments

tardis.urls.facility module

Facility URLs

tardis.urls.group module

Group URLs

tardis.urls.token module

Token URLs

Module contents**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.celery 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.views module

`tardis.views.error_handler (request, **kwargs)`

Module contents

importing all views files here, so that any old code will work as expected, when importing from `tardis.tardis_portal.views`

`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 and login pages. 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 overridden by apps on a per-schema basis, allowing custom views for specific 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 = [
    ('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`.

Custom Login View

Rationale

Specific sites or facilities may want to display a custom login page that which is more meaningful to their particular domain or application. MyTardis supports overriding the login page (/login) on a per-domain or per-Site basis.

User Guide

Example:

```
LOGIN_VIEWS = {
    1: 'tardis.apps.my_custom_app.views.MyCustomLoginViewClass',
    'facility.example.org': 'tardis.apps.myapp.AnotherCustomLoginViewClass'
}
```

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.

For an example MyTardis app which provides a login view, see <https://github.com/mytardis/mytardis-aaf-google-login>

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`, `login.html`, `view_dataset.html` and `view_experiment.html` templates can be 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

Raises **NotImplementedError** – not implemented

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 currently not implemented

11.4.3 MyTardis Social Authentication

Overview

The MyTardis social auth app allows MyTardis deployments to accept logins using OAuth and OpenID Connect. It builds on the [Python social auth package](#), and uses the [Django social auth app](#).

Usage

To enable the app, include `social_django` and `tardis.apps.social_auth` in settings. `INSTALLED_APPS`:

```
INSTALLED_APPS += (
    'social_django',
    'tardis.apps.social_auth',
)
```

Adding backends

You will need to add authentication backends that you want to enable. To enable Google authentication add following AUTHENTICATION_BACKENDS to *settings.py*

```
AUTHENTICATION_BACKENDS += (
    'social_core.backends.open_id.OpenIdAuth',
    'social_core.backends.google.GoogleOpenId',
    'social_core.backends.google.GoogleOAuth2',
)
```

To enable Australian Access federation(AAF) OpenID connect Provider(OIDC) authentication add following AUTHENTICATION_BACKENDS to *settings.py*

```
AUTHENTICATION_BACKENDS += (
    'tardis.apps.social_auth.auth.authorisation.AAFOpenId',
)
```

Adding authentication providers

You will need add authentication providers that you want to enable.

```
AUTH_PROVIDERS += (
    ('Google', 'Google',
     'social_core.backends.google.GoogleOAuth2'),
    ('AAF', 'AAF',
     'tardis.apps.social_auth.auth.authorisation.AAFOpenId'),
)
```

Adding Exception Middleware

You may want to add exception middleware provided by *python-social-auth*. To do this add following to *settings.py*

```
MIDDLEWARE += (
    'social_django.middleware.SocialAuthExceptionMiddleware',
)
```

Adding Context Processor

You will need to add following context processor to *settings.py*

```
TEMPLATES[0]['OPTIONS']['context_processors'].extend([
    'social_django.context_processors.backends',
    'social_django.context_processors.login_redirect'])
```

Application setup

Once the application is enabled and installed define the following settings to enable authentication behaviour.

```
SOCIAL_AUTH_AAF_PIPELINE = (  
'social_core.pipeline.social_auth.social_details',  
'social_core.pipeline.social_auth.social_uid',  
'social_core.pipeline.social_auth.social_user',  
'social_core.pipeline.user.get_username',  
'social_core.pipeline.user.create_user',  
'tardis.apps.social_auth.auth.social_auth.configure_social_auth_user',  
'tardis.apps.social_auth.auth.social_auth.add_authentication_method',  
'tardis.apps.social_auth.auth.social_auth.approve_user_auth',  
'tardis.apps.social_auth.auth.social_auth.add_user_permissions',  
'social_core.pipeline.social_auth.associate_user',  
'social_core.pipeline.social_auth.load_extra_data',  
'social_core.pipeline.user.user_details',  
)
```

```
SOCIAL_AUTH_GOOGLE_OAUTH2_PIPELINE = (  
'social_core.pipeline.social_auth.social_details',  
'social_core.pipeline.social_auth.social_uid',  
'social_core.pipeline.social_auth.social_user',  
'social_core.pipeline.user.get_username',  
'social_core.pipeline.user.create_user',  
'tardis.apps.social_auth.auth.social_auth.configure_social_auth_user',  
'tardis.apps.social_auth.auth.social_auth.add_authentication_method',  
'social_core.pipeline.social_auth.associate_user',  
'social_core.pipeline.social_auth.load_extra_data',  
'social_core.pipeline.user.user_details',  
'tardis.apps.social_auth.auth.social_auth.send_admin_email',  
)
```

Get key and secrets from the OIDC provider that you want to enable and add following settings.

```
SOCIAL_AUTH_URL_NAMESPACE_BEGIN = 'social:begin',  
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = 'Get this from Google'  
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = 'Get this from Google'  
SOCIAL_AUTH_AAF_KEY = 'Get this from AAF'  
SOCIAL_AUTH_AAF_SECRET = 'Get this from AAF'  
SOCIAL_AUTH_AAF_AUTH_URL = 'Get this from AAF'  
SOCIAL_AUTH_AAF_TOKEN_URL = 'Get this from AAF'  
SOCIAL_AUTH_AAF_USER_INFO_URL = 'Get this from AAF'
```

To override MyTardis's default login page (Username / Password) with a more appropriate page for AAF and Google authentication, you can use <https://github.com/mytardis/mytardis-aaf-google-login>

11.5 Releases

11.5.1 4.3.0

- Updated various Python and JS libraries
- Removed Mustache and Backbone JS libraries (replaced with React)
- No new session will be created on API call or health check

- Check DataFile permissions in API differently (performance gain)
- Check for dataset permissions during API calls (security bug)
- New uploader registration email to be send to managers group not admins

11.5.2 4.2.0

- Upgraded Django from 1.11.26 to 2.2.6
- Upgraded Bootstrap from 3.4.1 to 4.1.3
- Continuous Integration testing implemented for Python 3.5, 3.6, 3.7 and 3.8
- Dropped support for Python 2
- Dataset view now has a tree-based file browser
- Added a task which can be scheduled to clean up unverified files
- Added a task which can be scheduled to clean up DataFiles without DataFileObjects
- Bug fixes - Ensured thumbnail image files are opened in binary format, required for Python 3 - Fixed bug with downloads on Python 3 by removing use of `.__next__()` - Fixed bugs in Push To's encoding/decoding of SSH certificates in Python 3 - Fixed bug in MyTardis SFTP service relating to use of `gevent` with Django 2.2
- Dependency updates - Python and JS dependencies have been updated to address vulnerabilities.

11.5.3 4.1.5

- Update AngularJS to address the SNYK-JS-ANGULAR-534884 vulnerability.
- Update the handlebars version in package-lock.json to avoid having “npm install” report high severity vulnerabilities.
- Fix the dataset metadata API test which was failing on Python 3.5.

11.5.4 4.1.4

- Fixed duplicate form submission bugs for create experiment/dataset
- Fixed search bug which restricted instrument drop-down to 20 records
- Fixed some byte string encoding issues with LDAP auth in Python 3
- Fixed Python 3.5 unit tests
- Fixed pickled StorageBoxOption values for Python 3

11.5.5 4.1.3

- Update the https-proxy-agent version in package-lock.json to avoid having “npm install” display “found 1 high severity vulnerability”.

11.5.6 4.1.2

- Allow .jsx files to be included in assets/js/tardis_portal/ and ensure that they won't be linted using the jQuery ESLint configuration
- Switch back to the official version of the pyoi dependency

11.5.7 4.1.1

- Fix Python 3 bug with string encoding in deep download mapper which affected directory names in SFTP interface.

11.5.8 4.1

- Added React search components and django-elasticsearch-dsl backend
- Removed post-save filters middleware, replaced with microservice architecture
- Added RabbitMQ task priorities support, dropped support for Redis as a broker
- Upgraded Bootstrap CSS framework from v2.3.2 to v3.4.1
- Added Python 3 support
- Added webpack to collect static assets (JS / CSS), supporting ES6, JSX etc.
- Annotated storage box related tasks with their storage box name, visible in “celery inspect active”
- Added task for clearing Django sessions
- Added timestamps (created and modified) in facility and instrument models
- Updated built-in Creative Commons licenses to v4
- Added django-storages and boto3 to requirements to support S3 storage boxes and storing static assets in S3
- Improved efficiency of checksums and downloads for files in S3 storage
- COMPUTE_SHA512 now defaults to False. COMPUTE_MD5 still defaults to True.
- Legal text for publishing can now be specified in settings
- Now using Dataset created_time in facility overview instead of experiment created time
- Added a new setting to prevent large datasets (many files) from being scanned for image files at page load time
- API v1's instrument resource now allows any authenticated user to list the instrument names, which is used in the new search interface
- The ExperimentAuthor model now exposed in API v1
- MyTardis no longer tries to guess an appropriate storage box for new DataFileObjects unless REUSE_DATASET_STORAGE_BOX is True
- Improved BDD test coverage, now measuring template coverage with django-coverage-plugin
- Bug fixes (GitHub Issue numbers below) - Fixed #1503 - Fixed #1568 - Removed bob@bobmail.com from default ADMINS, fixing #1613 - Fixed #1664 - Fixed #1708 - Fixed #1857 - Fixed #1853 - Fixed concatenated messages issue in user sharing and group sharing dialogs - Fixed #1790 - Fixed truncated TAR download issue with unverified files - Fixed sharing with AAF/Google issue - Fixed some broken Font Awesome icons

11.5.9 4.0.2

- Upgraded Django to 1.11.23
- Upgraded vulnerable dependencies of JS dev dependencies
- Fixed #1844 (remove a hard-coded LDAP attribute)

11.5.10 4.0.1

- Removed ansrc_codes tardis app which contained a potentially insecure dependency (rdflib)
- Added created_time and modified_time fields in the Instrument and Facility models
- Updated Python and Javascript dependencies which had vulnerabilities reported since the v4.0 release.
- Fixed token authentication (#1531, 615d9df)
- Fixed some Font Awesome icons (1ac549d)
- Fixed an incomplete database migration for the Dataset created_time field. This fix is included in the tardis/tardis_portal/migrations/0016_add_timestamps.py migration which also adds the created_time and modified_time fields to the Instrument and Facility models (ec238b4)
- Removed hard-coded LDAP attributes (#1664, 96a0fbf)
- Fixed issue with get_accessible_datafiles_for_user potentially returning an empty list instead of an empty QuerySet (a13cefc)
- Fixed issue with Add/Save Experiment Metadata - added a form attribute to the Save button. (fd2393a)
- In S3 storage documentation, removed reference to old fork of django-storages. (f0c62d5)
- Fixed issue where MyTardis could try to verify SHA512 sums even when COMPUTE_SHA512 was set to False (#1419, 1da1b3b)
- In S3 storage documentation, removed reference to old fork of django-storages. (f0c62d5)
- Fixed issue where MyTardis could try to verify SHA512 sums even when COMPUTE_SHA512 was set to False (#1419, 1da1b3b)
- Fixed issue where downloading a TAR of a dataset including unverified files could result in a “Truncated tar archive” error (#1425, b4fa17c)
- Fixed issue where MyTardis tried to retrieve thumbnail images for non-image files, resulting in 404 errors (e261065)
- Fixed issue where failing to set ADMINS in tardis/settings.py could cause MyTardis to attempt to send emails to bob@bobmail.com (#1613, f8ed6dd)
- Fixed issue where Facility Overview’s “Load more” button was enabled while content was still loading. (a28a253)

11.5.11 4.0

- Django 1.11
- jQuery 3.3.1
- Improved test coverage
- Continuous Integration tests run against Ubuntu 18.04 (MyTardis v3.x used 14.04)

- ChromeDriver is used for BDD (Behaviour Driven Development) tests
- Social Auth, including AAF and Google Auth via OpenID Connect
- Migrating user accounts from LDAP or localdb to OpenID Connect
- Customizable user menu
- Using `message.level_tag` instead of `message.tags` in `portal_template`, so that extra tags can be added to Django messages without interfering with the Bootstrap alert class.
- My Data page (which previously contained Owned and Shared experiments) has been split into two pages - “My Data” and “Shared” - Each page loads thumbnails asynchronously for faster initial page load time - An improved pagination widget allows for a very large number of pages.
- Index page’s thumbnails are loaded asynchronously for faster initial page load time.
- Login page can be replaced with a site-specific page
- SFTP can now be used with keys instead of passwords
- Upgraded Bootstrap from 2.0.4 to 2.3.2 (further upgrades coming soon)
- Fixed some bugs in single search
- jQuery code is being moved out of HTML templates and into JS files which can be linted (with ESLint) and tested (with QUnit).
- **Removed old broken code and unnecessary code which is duplicated in other repositories.**
 - Import via staging with jsTree
 - Uploadify
- **Updated or removed (as appropriate) some out-of-date JS dependencies bundled within the MyTardis repository**
 - Most JS dependences are installed by npm now, so we can run security checks with npm audit
- `manage.py` can now be used instead of `mytardis.py` and `mytardis.py` will soon be deprecated
- New support email setting can be used in email templates or HTML templates.
- Updating loadschemas management command for Django 1.11 and adding test for it
- Updated the dumpschemas management command for Django 1.11 and added a test for it
- **Bug fixes (GitHub Issue numbers below)** Fixed #243. Bug in `tardis_acls.change_experiment` permissions check Fixed #516 - only show “Add files” button if user has permission to upload files Fixed #636 Fixed #637 - “()” is added to “Author” line every time an experiment is edited Fixed #779 Fixed #868 Fixed #893 Fixed #988 Fixed #1083 Fixed #1185
- Added docs on X-Forwarded-Proto HTTP header for HTTPS deployments
- Added docs on configuring services in systemd or supervisor
- Removed password length restriction in linked user authentication form
- Removed `settings_changeme` - use `default_settings` instead
- Removed backslash from set of characters used to generate secret key.
- Removed `django-celery` - it is no longer necessary to run Celery via Django
- Improved forwards compatibility with Python 3, but we’re not fully Python 3 compatible yet.
- Switched to PEP 328 relative imports
- Tests no longer require the unmaintained “compare” module

- Added a default value for DATA_UPLOAD_MAX_MEMORY_SIZE (required by Django 1.10+) to default settings
- Removed some unused dependencies, e.g. PyYAML
- Removed the createmysuperuser which is no longer needed
- Removed the checkhashes management command
- Removed the diffraction image filter
- Removed the backupdb management command
- Removed the old publication form - a new publication workflow is coming soon.

11.5.12 3.9

- Added deprecation warnings for functionality which will be removed in 4.0
- Added INTERNAL_IPS to default settings for template debugging on localhost
- Disabled the old publication forms app in default settings, and ensured that MyTardis didn't attempt to access its static content when disabled
- Removed apps code from ExperimentView's get_context_data which assumed that each app would provide a views module with an index
- Fixed a bug where creating a group which already existed gave a 500 error
- Fixed a bug where non-ASCII characters in experiment names could break SFTP
- Made dataset thumbnails optional - disabling them can improve page load times
- Fixed a bug which had made it difficult to delete a DataFileObject without a URI from the Django shell
- Fixed a bug which made search indexing fail when there were users with non-ASCII characters in their first or last name

11.5.13 3.8.1

- Fix regression in Push To app

11.5.14 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.

11.5.15 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

11.5.16 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

11.5.17 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.

11.5.18 3.0 - unreleased

- Twitter Bootstrap
- javascript templates
- backbone.js rendering of datasets
- UI for transferring datasets
- bpython shell
- celery queue

11.5.19 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 individual 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

11.5.20 1.07 - 01/06/2010

- Publish to tardis.edu.au interface created, though not implemented, pending legal text

11.5.21 1.06 - 15/03/2010

- Parameter import interface for creation of new parameter/schema definitions
- iPhone Interface

11.5.22 1.05 - 01/03/2010

- Images as parameters supported
- Data / metadata transfer from synchrotron is now 'threaded' using asynchronous web service transfers.

11.5.23 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

11.5.24 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

11.5.25 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`

t

tardis, 244

tardis.analytics, 76

tardis.analytics.apps, 75

tardis.analytics.ga, 75

tardis.analytics.tracker, 75

tardis.app_config, 243

tardis.apps, 118

tardis.apps.deep_storage_download_mapper, 76

tardis.apps.deep_storage_download_mapper.mapper, 76

tardis.apps.deep_storage_download_mapper.urls, 76

tardis.apps.dl_mapper_df_dir_only, 77

tardis.apps.dl_mapper_df_dir_only.mapper, 76

tardis.apps.dl_mapper_df_dir_only.urls, 77

tardis.apps.filepicker, 77

tardis.apps.filepicker.filepicker_settings, 77

tardis.apps.filepicker.urls, 77

tardis.apps.filepicker.utils, 77

tardis.apps.filepicker.views, 77

tardis.apps.oaipmh, 84

tardis.apps.oaipmh.models, 82

tardis.apps.oaipmh.provider, 80

tardis.apps.oaipmh.provider.base, 77

tardis.apps.oaipmh.provider.experiment, 79

tardis.apps.oaipmh.server, 82

tardis.apps.oaipmh.tests, 82

tardis.apps.oaipmh.tests.provider, 81

tardis.apps.oaipmh.tests.provider.test_base, 80

tardis.apps.oaipmh.tests.provider.test_experiment, 81

tardis.apps.oaipmh.tests.test_oai, 81

tardis.apps.oaipmh.urls, 84

tardis.apps.oaipmh.views, 84

tardis.apps.openid_migration, 89

tardis.apps.openid_migration.apps, 85

tardis.apps.openid_migration.context_processors, 85

tardis.apps.openid_migration.default_settings, 86

tardis.apps.openid_migration.email_text, 86

tardis.apps.openid_migration.forms, 86

tardis.apps.openid_migration.migration, 86

tardis.apps.openid_migration.migrations, 84

tardis.apps.openid_migration.migrations.0001_initial, 84

tardis.apps.openid_migration.models, 86

tardis.apps.openid_migration.tasks, 89

tardis.apps.openid_migration.tests, 85

tardis.apps.openid_migration.tests.test_forms, 84

tardis.apps.openid_migration.tests.test_migration, 84

tardis.apps.openid_migration.tests.test_models, 85

tardis.apps.openid_migration.tests.test_views, 85

tardis.apps.openid_migration.urls, 89

tardis.apps.openid_migration.user_menu_modifiers, 89

tardis.apps.openid_migration.utils, 89

tardis.apps.openid_migration.views, 89

tardis.apps.push_to, 98

tardis.apps.push_to.apps, 90

tardis.apps.push_to.exceptions, 91

tardis.apps.push_to.migrations, 90

tardis.apps.push_to.migrations.0001_initial, 90

tardis.apps.push_to.migrations.0002_auto_20160518_1

90
tardis.apps.push_to.models, 91
tardis.apps.push_to.oauth_tokens, 96
tardis.apps.push_to.ssh_authz, 97
tardis.apps.push_to.tasks, 97
tardis.apps.push_to.tests, 90
tardis.apps.push_to.tests.test_models, 90
tardis.apps.push_to.urls, 97
tardis.apps.push_to.utils, 97
tardis.apps.push_to.views, 97
tardis.apps.related_info, 100
tardis.apps.related_info.forms, 100
tardis.apps.related_info.models, 100
tardis.apps.related_info.settings, 100
tardis.apps.related_info.tests, 100
tardis.apps.related_info.tests.test_oaipmh, 98
tardis.apps.related_info.tests.tests, 99
tardis.apps.related_info.urls, 100
tardis.apps.related_info.views, 100
tardis.apps.s3utils, 103
tardis.apps.s3utils.api, 101
tardis.apps.s3utils.apps, 102
tardis.apps.s3utils.default_settings, 102
tardis.apps.s3utils.tests, 101
tardis.apps.s3utils.tests.test_api, 100
tardis.apps.s3utils.tests.test_app_config, 101
tardis.apps.s3utils.tests.test_checksums, 101
tardis.apps.s3utils.urls, 102
tardis.apps.s3utils.utils, 102
tardis.apps.search, 107
tardis.apps.search.api, 103
tardis.apps.search.apps, 105
tardis.apps.search.documents, 105
tardis.apps.search.tests, 103
tardis.apps.search.tests.test_api, 103
tardis.apps.search.tests.test_index, 103
tardis.apps.search.urls, 107
tardis.apps.search.views, 107
tardis.apps.sftp, 117
tardis.apps.sftp.admin, 108
tardis.apps.sftp.api, 108
tardis.apps.sftp.apps, 110
tardis.apps.sftp.default_settings, 110
tardis.apps.sftp.forms, 110
tardis.apps.sftp.management, 107
tardis.apps.sftp.management.commands, 107
tardis.apps.sftp.management.commands.sftpd, 107
tardis.apps.sftp.migrations, 108
tardis.apps.sftp.migrations.0001_initial, 108
tardis.apps.sftp.models, 111
tardis.apps.sftp.sftp, 112
tardis.apps.sftp.tests, 108
tardis.apps.sftp.tests.test_sftp, 108
tardis.apps.sftp.urls, 117
tardis.apps.sftp.user_menu_modifiers, 117
tardis.apps.sftp.views, 117
tardis.apps.social_auth, 118
tardis.apps.social_auth.apps, 118
tardis.apps.social_auth.auth, 118
tardis.apps.social_auth.auth.social_auth, 117
tardis.apps.social_auth.default_settings, 118
tardis.celery, 244
tardis.default_settings, 124
tardis.default_settings.admins, 119
tardis.default_settings.analytics, 119
tardis.default_settings.apps, 119
tardis.default_settings.auth, 119
tardis.default_settings.caches, 119
tardis.default_settings.celery_settings, 119
tardis.default_settings.custom_views, 119
tardis.default_settings.database, 120
tardis.default_settings.debug, 120
tardis.default_settings.downloads, 121
tardis.default_settings.email, 121
tardis.default_settings.filters, 121
tardis.default_settings.frontend, 122
tardis.default_settings.i18n, 122
tardis.default_settings.localisation, 122
tardis.default_settings.logging, 122
tardis.default_settings.middlewares, 122
tardis.default_settings.publication, 122
tardis.default_settings.search, 122
tardis.default_settings.sharing, 123
tardis.default_settings.site_customisations, 123
tardis.default_settings.static_files, 123
tardis.default_settings.storage, 123
tardis.default_settings.templates, 124
tardis.default_settings.uploads, 124
tardis.default_settings.urls, 124
tardis.tardis_portal, 242

tardis.tardis_portal.admin, 213	133
tardis.tardis_portal.api, 218	tardis.tardis_portal.migrations.0006_datafile_removal, 133
tardis.tardis_portal.auth, 130	133
tardis.tardis_portal.auth.authentication, 124	tardis.tardis_portal.migrations.0007_remove_parameter, 134
tardis.tardis_portal.auth.authorisation, 125	tardis.tardis_portal.migrations.0008_string_value_type, 134
tardis.tardis_portal.auth.authservice, 125	tardis.tardis_portal.migrations.0009_auto_20160128_134, 134
tardis.tardis_portal.auth.decorators, 126	tardis.tardis_portal.migrations.0010_auto_20160503_134, 134
tardis.tardis_portal.auth.fix_circular, 128	tardis.tardis_portal.migrations.0011_auto_20160505_134, 134
tardis.tardis_portal.auth.interfaces, 128	tardis.tardis_portal.migrations.0012_userauthentication, 135
tardis.tardis_portal.auth.localdb_auth, 129	tardis.tardis_portal.migrations.0013_auto_20181002_135, 135
tardis.tardis_portal.auth.token_auth, 130	tardis.tardis_portal.migrations.0014_auto_20181002_135, 135
tardis.tardis_portal.auth.utils, 130	tardis.tardis_portal.migrations.0015_dataset_create, 135
tardis.tardis_portal.context_processors, 228	tardis.tardis_portal.migrations.0016_add_timestamps, 135
tardis.tardis_portal.deprecations, 228	tardis.tardis_portal.migrations.0017_add_cc_license, 135
tardis.tardis_portal.download, 229	tardis.tardis_portal.migrations.0018_make_default_136, 136
tardis.tardis_portal.email, 230	tardis.tardis_portal.models, 182
tardis.tardis_portal.fields, 230	tardis.tardis_portal.models.access_control, 136
tardis.tardis_portal.forms, 230	tardis.tardis_portal.models.datafile, 141
tardis.tardis_portal.iiif, 236	tardis.tardis_portal.models.dataset, 147
tardis.tardis_portal.logging_middleware, 236	tardis.tardis_portal.models.experiment, 150
tardis.tardis_portal.management, 132	tardis.tardis_portal.models.facility, 155
tardis.tardis_portal.management.commands, 132	tardis.tardis_portal.models.hooks, 157
tardis.tardis_portal.management.commands.add_dataset, 131	tardis.tardis_portal.models.instrument, 157
tardis.tardis_portal.management.commands.add_experiment, 131	tardis.tardis_portal.models.jti, 158
tardis.tardis_portal.management.commands.rm_experiment, 132	tardis.tardis_portal.models.license, 159
tardis.tardis_portal.managers, 236	tardis.tardis_portal.models.parameters, 160
tardis.tardis_portal.migrations, 136	tardis.tardis_portal.models.storage, 176
tardis.tardis_portal.migrations.0001_initial, 132	tardis.tardis_portal.models.token, 181
tardis.tardis_portal.migrations.0001_square_and_install_20150114_132, 132	tardis.tardis_portal.ParameterSetManager, 135
tardis.tardis_portal.migrations.0002_auto_20150128_133, 133	tardis.tardis_portal.publish, 183
tardis.tardis_portal.migrations.0003_auto_20150907_133, 133	tardis.tardis_portal.publish.provider, 183
tardis.tardis_portal.migrations.0004_storage_expiration_value_type, 133	tardis.tardis_portal.publish.provider.rifcsprovider, 183
tardis.tardis_portal.migrations.0005_datafile_size_report_at_publish, 133	tardis.tardis_portal.publish.publishservice, 183

183
tardis.tardis_portal.rfc3339, 239
tardis.tardis_portal.shortcuts, 239
tardis.tardis_portal.storage, 184
tardis.tardis_portal.storage.file_system, 183
tardis.tardis_portal.tasks, 240
tardis.tardis_portal.template_tags, 186
tardis.tardis_portal.template_tags.approved_users, 184
tardis.tardis_portal.template_tags.basiccompared_filters, 184
tardis.tardis_portal.template_tags.bleach_tags, 184
tardis.tardis_portal.template_tags.capture_tags, 185
tardis.tardis_portal.template_tags.dynurl, 185
tardis.tardis_portal.template_tags.experiment_tags, 185
tardis.tardis_portal.template_tags.experiment_statuses, 185
tardis.tardis_portal.template_tags.facility_tags, 185
tardis.tardis_portal.template_tags.feed, 185
tardis.tardis_portal.template_tags.formfield_filters, 185
tardis.tardis_portal.template_tags.lookup_filters, 186
tardis.tardis_portal.template_tags.pagination, 186
tardis.tardis_portal.template_tags.xml_dates, 186
tardis.tardis_portal.tests, 206
tardis.tardis_portal.tests.api, 191
tardis.tardis_portal.tests.api.test_auth, 186
tardis.tardis_portal.tests.api.test_datafile, 187
tardis.tardis_portal.tests.api.test_datafile_resource, 187
tardis.tardis_portal.tests.api.test_datafile_report, 187
tardis.tardis_portal.tests.api.test_datafile_resource, 188
tardis.tardis_portal.tests.api.test_experiment, 188
tardis.tardis_portal.tests.api.test_experiment_source, 188
tardis.tardis_portal.tests.api.test_facility_resource, 188
tardis.tardis_portal.tests.api.test_group_resource, 189
tardis.tardis_portal.tests.api.test_instrument_resource, 189
tardis.tardis_portal.tests.api.test_list_api_endpoints, 189
tardis.tardis_portal.tests.api.test_replica_resource, 189
tardis.tardis_portal.tests.api.test_schema_resource, 190
tardis.tardis_portal.tests.api.test_serializer, 190
tardis.tardis_portal.tests.api.test_storagebox_resource, 190
tardis.tardis_portal.tests.api.test_user_resource, 190
tardis.tardis_portal.tests.auth, 191
tardis.tardis_portal.tests.ldap_ldif, 196
tardis.tardis_portal.tests.management, 192
tardis.tardis_portal.tests.management.test_collectors, 191
tardis.tardis_portal.tests.management.test_create_users, 191
tardis.tardis_portal.tests.management.test_dump_schema, 192
tardis.tardis_portal.tests.management.test_load_schema, 192
tardis.tardis_portal.tests.management.test_rm_experiments, 192
tardis.tardis_portal.tests.models, 194
tardis.tardis_portal.tests.models.test_authors, 192
tardis.tardis_portal.tests.models.test_datafile, 193
tardis.tardis_portal.tests.models.test_dataset, 193
tardis.tardis_portal.tests.models.test_dfo, 193
tardis.tardis_portal.tests.models.test_experiment, 193
tardis.tardis_portal.tests.models.test_generate_api, 193
tardis.tardis_portal.tests.models.test_instrument, 194
tardis.tardis_portal.tests.models.test_parameter, 194
tardis.tardis_portal.tests.models.test_resource, 196
tardis.tardis_portal.tests.test_authentication, 198
tardis.tardis_portal.tests.test_authorisation, 198
tardis.tardis_portal.tests.test_authservice, 199
tardis.tardis_portal.tests.test_copy_move,

[199](#)
 tardis.tardis_portal.tests.test_df_save_metadata, [208](#)
[200](#)
 tardis.tardis_portal.tests.test_download, [208](#)
[200](#)
 tardis.tardis_portal.tests.test_download_apikey, [211](#)
[200](#)
 tardis.tardis_portal.tests.test_facility, [212](#)
[201](#)
 tardis.tardis_portal.tests.test_forms, [241](#)
[201](#)
 tardis.tardis_portal.tests.test_iiif, [244](#)
[201](#)
 tardis.tardis_portal.tests.test_ldap, [244](#)
[202](#)
 tardis.tardis_portal.tests.test_parameter, [242](#)
[202](#)
 tardis.tardis_portal.tests.test_parameter, [242](#)
[203](#)
 tardis.tardis_portal.tests.test_publish, [243](#)
[204](#)
 tardis.tardis_portal.tests.test_storage, [243](#)
[204](#)
 tardis.tardis_portal.tests.test_tar_download, [243](#)
[205](#)
 tardis.tardis_portal.tests.test_tasks, [243](#)
[205](#)
 tardis.tardis_portal.tests.test_tokens, [243](#)
[205](#)
 tardis.tardis_portal.tests.tests, [206](#)
 tardis.tardis_portal.tests.views, [196](#)
 tardis.tardis_portal.tests.views.test_auth_views, [194](#)
 tardis.tardis_portal.tests.views.test_contextual_views, [195](#)
 tardis.tardis_portal.tests.views.test_experiment_views, [195](#)
 tardis.tardis_portal.tests.views.test_template_contexts, [195](#)
 tardis.tardis_portal.tests.views.test_upload_views, [196](#)
 tardis.tardis_portal.util, [240](#)
 tardis.tardis_portal.views, [212](#)
 tardis.tardis_portal.views.ajax_actions, [206](#)
 tardis.tardis_portal.views.ajax_json, [206](#)
 tardis.tardis_portal.views.ajax_pages, [207](#)
 tardis.tardis_portal.views.authentication, [207](#)
 tardis.tardis_portal.views.authorisation, [207](#)
 tardis.tardis_portal.views.facilities,

A

- `about()` (in module `tardis.tardis_portal.views.pages`), 210
- `abstract` (`tardis.apps.push_to.models.KeyPair.Meta` attribute), 93
- `abstract` (`tardis.tardis_portal.models.parameters.ParameterSet.Meta` attribute), 169
- `abstract` (`tardis.tardis_portal.models.parameters.ParameterSet.Meta` attribute), 173
- `AbstractExperimentProvider` (class in `tardis.apps.oaipmh.provider.experiment`), 79
- `AbstractExperimentProviderTC` (class in `tardis.apps.oaipmh.tests.provider.test_experiment`), 81
- `AbstractTardisAppConfig` (class in `tardis.app_config`), 243
- `accessible_keys` (`tardis.tardis_portal.api.StorageBoxOptionResource` attribute), 227
- `acl_id` (`tardis.apps.openid_migration.models.OpenidACLMigration` attribute), 87
- `acl_id_id` (`tardis.apps.openid_migration.models.OpenidACLMigration` attribute), 87
- `acl_migration()` (in module `tardis.apps.openid_migration.migration`), 86
- `ACLAuthorization` (class in `tardis.tardis_portal.api`), 218
- `ACLAuthorizationTest` (class in `tardis.tardis_portal.tests.api.test_auth`), 186
- `ACLAwareBackend` (class in `tardis.tardis_portal.auth.authorisation`), 125
- `aclOwnershipType` (`tardis.tardis_portal.models.access_control.ObjectACL` attribute), 137
- `add_arguments()` (`tardis.apps.sftp.management.commands.sftpd.Command` method), 107
- `add_arguments()` (`tardis.tardis_portal.management.commands.createuser.Command` method), 131
- `add_arguments()` (`tardis.tardis_portal.management.commands.dumpschemas.Command` method), 131
- `add_arguments()` (`tardis.tardis_portal.management.commands.loadschema.Command` method), 131
- `add_arguments()` (`tardis.tardis_portal.management.commands.rmexperiment.Command` method), 132
- `add_auth_method()` (in module `tardis.tardis_portal.auth.authentication`), 124
- `add_authentication_method()` (in module `tardis.apps.social_auth.auth.social_auth`), 117
- `add_child()` (`tardis.apps.sftp.sftp.DynamicTree` method), 112
- `add_datafile_par()` (in module `tardis.tardis_portal.views.parameters`), 211
- `add_dataset()` (in module `tardis.tardis_portal.views.pages`), 210
- `add_dataset_par()` (in module `tardis.tardis_portal.views.parameters`), 211
- `add_experiment_par()` (in module `tardis.tardis_portal.views.parameters`), 211
- `add_migrate_account_menu_item()` (in module `tardis.apps.openid_migration.user_menu_modifiers`), 89
- `add_migration_permission()` (in module `tardis.apps.social_auth.auth.social_auth`), 117
- `add_par()` (in module `tardis.tardis_portal.views.parameters`), 211
- `add_path()` (`tardis.apps.sftp.sftp.DynamicTree` method), 112
- `add_path_elems()` (`tardis.apps.sftp.sftp.DynamicTree` method), 112
- `add_ssh_keys_menu_item()` (in module `tardis.apps.sftp.user_menu_modifiers`), 117
- `add_user_permissions()` (in module `tardis.apps.social_auth.auth.social_auth`), 118
- `added` (`tardis.apps.sftp.models.SFTPPublicKey` attribute), 111
- `AddUserPermissionsForm` (class in `tardis.tardis_portal.forms`), 230

administrator (*tardis.apps.push_to.models.RemoteHost*
 attribute), 95
 administrator_id (*tardis.apps.push_to.models.RemoteHost*
 attribute), 95
 ADMINS (in module *tardis.default_settings.admins*), 119
 AdvanceSearchAppResource (class in *app_dependencies* (*tardis.app_config.AbstractTardisAppConfig*
 tardis.apps.search.api), 103
 AdvanceSearchAppResource.Meta (class in *app_label* (*tardis.tardis_portal.auth.authorisation.ACLAwareBackend*
 tardis.apps.search.api), 103
 all () (*tardis.tardis_portal.managers.ExperimentManager*
 method), 236
 allow_for_all (*tardis.apps.push_to.models.OAuthSSHCertSigningService* (*tardis.tardis_portal.models.parameters.ParameterSet.Meta*
 attribute), 93
 allow_reuse_address (*tardis.apps.sftp.sftp.MyTSFTPTCPServer*
 attribute), 114
 allowed_groups (*tardis.apps.push_to.models.OAuthSSHCertSigningService* (*tardis.tardis_portal.models.parameters.ParameterSet.Meta*
 attribute), 93
 ALLOWED_HOSTS, 45
 ALLOWED_HOSTS (in module *tardis.default_settings.debug*), 120
 allowed_methods (*tardis.tardis_portal.api.UserResource.Meta*
 attribute), 227
 allowed_remote_hosts (*tardis.apps.push_to.models.OAuthSSHCertSigningService* (*tardis.tardis_portal.models.parameters.ParameterSet.Meta*
 attribute), 93
 allowed_users (*tardis.apps.push_to.models.OAuthSSHCertSigningService* (*tardis.tardis_portal.models.parameters.ParameterSet.Meta*
 attribute), 94
 allows_distribution (*tardis.tardis_portal.models.license.License*
 attribute), 159
 always_return_data (*tardis.apps.search.api.AdvanceSearchAppResource.Meta*
 attribute), 103
 always_return_data (*tardis.apps.search.api.SearchAppResource.Meta*
 attribute), 104
 always_return_data (*tardis.tardis_portal.api.DatasetResource.Meta*
 attribute), 221
 always_return_data (*tardis.tardis_portal.api.ExperimentAuthorResource.Meta*
 attribute), 222
 always_return_data (*tardis.tardis_portal.api.ExperimentResource.Meta*
 attribute), 223
 always_return_data (*tardis.tardis_portal.api.FacilityResource.Meta*
 attribute), 223
 always_return_data (*tardis.tardis_portal.api.InstrumentResource.Meta*
 attribute), 224
 AnalyticsConfig (class in *tardis.analytics.apps*), 75
 ApiKeyDownloadTestCase (class in *tardis.tardis_portal.tests.test_download_apikey*), 200
 ApiKeyTestCase (class in *tardis.tardis_portal.tests.models.test_generate_api_key*), 193
 app_dependencies (*tardis.app_config.AbstractTardisAppConfig*
 attribute), 243
 app_label (*tardis.tardis_portal.auth.authorisation.ACLAwareBackend*
 attribute), 125
 app_label (*tardis.tardis_portal.models.parameters.Parameter.Meta*
 attribute), 169
 app_label (*tardis.tardis_portal.models.parameters.ParameterSet.Meta*
 attribute), 173
 apply_filters () (*tardis.tardis_portal.models.datafile.DataFileObject*
 method), 145
 approve_user_auth () (in module *tardis.tardis_portal.social_auth.auth.social_auth*), 118
 approved (*tardis.tardis_portal.models.access_control.UserAuthentication*
 attribute), 139
 approved (*tardis.tardis_portal.models.experiment.Experiment*
 attribute), 151
 args (*tardis.tardis_portal.management.commands.dumpschemas.Command*
 attribute), 131
 args (*tardis.tardis_portal.management.commands.loadschemas.Command*
 attribute), 131
 args (*tardis.tardis_portal.management.commands.rmexperiment.Command*
 attribute), 132
 attributes (*tardis.tardis_portal.models.storage.StorageBox*
 attribute), 177
 auth_timeout (*tardis.apps.sftp.sftp.MyTSFTPRequestHandler*
 attribute), 112
 authenticate () (*tardis.tardis_portal.auth.authorisation.ACLAwareBackend*
 method), 125
 authenticate () (*tardis.tardis_portal.auth.authservice.AuthService*
 method), 125
 authenticate () (*tardis.tardis_portal.auth.interfaces.AuthProvider*
 method), 128
 authenticate () (*tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend*
 method), 129
 authenticate () (*tardis.tardis_portal.tests.test_authservice.MockAuthService*
 method), 199
 authentication (*tardis.apps.search.api.AdvanceSearchAppResource.Meta*
 attribute), 103
 authentication (*tardis.apps.search.api.SearchAppResource.Meta*
 attribute), 104
 authentication (*tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta*
 attribute), 109
 authentication (*tardis.tardis_portal.api.GroupResource.Meta*
 attribute), 223
 authentication (*tardis.tardis_portal.api.MyTardisModelResource.Meta*
 attribute), 224
 authentication (*tardis.tardis_portal.api.ObjectACLResource.Meta*
 attribute), 225

authentication (*tardis.tardis_portal.api.UserResource*.*Meta* attribute), 227
 authenticationMethod (*tardis.tardis_portal.models.access_control.UserAuthentication* attribute), 139
 AuthenticationTestCase (class in *tardis.tardis_portal.tests.test_authentication*), 198
 author (*tardis.tardis_portal.models.experiment.Experiment* attribute), 154
 authorization (*tardis.apps.s3utils.api.ReplicaAppResource*.*Meta* attribute), 101
 authorization (*tardis.apps.sftp.api.SFTPPublicKeyAppResource*.*Meta* attribute), 109
 authorization (*tardis.tardis_portal.api.GroupResource*.*Meta* attribute), 223
 authorization (*tardis.tardis_portal.api.MyTardisModelResource*.*Meta* attribute), 224
 authorization (*tardis.tardis_portal.api.ObjectACLResource*.*Meta* attribute), 225
 authorization (*tardis.tardis_portal.api.UserResource*.*Meta* attribute), 227
 authorize_remote_access() (in module *tardis.apps.push_to.views*), 97
 AuthorTestCase (class in *tardis.tardis_portal.tests.models.test_authors*), 192
 AuthProvider (class in *tardis.tardis_portal.auth.interfaces*), 128
 AuthService (class in *tardis.tardis_portal.auth.authservice*), 125
 AuthServiceTestCase (class in *tardis.tardis_portal.tests.test_authservice*), 199
 autocache (*tardis.tardis_portal.models.storage.StorageBox* attribute), 177
 AUTOGENERATE_API_KEY (in module *tardis.default_settings.auth*), 119

B

BackgroundTaskTestCase (class in *tardis.tardis_portal.tests.test_tasks*), 205
 base_fields (*tardis.apps.push_to.models.CredentialForm* attribute), 92
 base_fields (*tardis.apps.related_info.forms.RelatedInfoForm* attribute), 100
 base_fields (*tardis.apps.s3utils.api.ReplicaAppResource* attribute), 101
 base_fields (*tardis.apps.search.api.AdvanceSearchAppResource* attribute), 104
 base_fields (*tardis.apps.search.api.SearchAppResource* attribute), 105
 base_fields (*tardis.apps.sftp.api.SFTPPublicKeyAppResource* attribute), 109
 base_fields (*tardis.apps.sftp.forms.KeyAddForm* attribute), 110
 base_fields (*tardis.apps.sftp.forms.KeyGenerateForm* attribute), 110
 base_fields (*tardis.tardis_portal.admin.DatafileAdminForm* attribute), 214
 base_fields (*tardis.tardis_portal.admin.DataFileObjectInlineForm* attribute), 214
 base_fields (*tardis.tardis_portal.admin.StorageBoxAttributeInlineForm* attribute), 217
 base_fields (*tardis.tardis_portal.admin.StorageBoxForm* attribute), 217
 base_fields (*tardis.tardis_portal.admin.StorageBoxOptionInlineForm* attribute), 218
 base_fields (*tardis.tardis_portal.api.DatafileParameterResource* attribute), 220
 base_fields (*tardis.tardis_portal.api.DatafileParameterSetResource* attribute), 220
 base_fields (*tardis.tardis_portal.api.DataFileResource* attribute), 219
 base_fields (*tardis.tardis_portal.api.DatasetParameterResource* attribute), 221
 base_fields (*tardis.tardis_portal.api.DatasetParameterSetResource* attribute), 221
 base_fields (*tardis.tardis_portal.api.DatasetResource* attribute), 221
 base_fields (*tardis.tardis_portal.api.ExperimentAuthorResource* attribute), 222
 base_fields (*tardis.tardis_portal.api.ExperimentParameterResource* attribute), 222
 base_fields (*tardis.tardis_portal.api.ExperimentParameterSetResource* attribute), 222
 base_fields (*tardis.tardis_portal.api.ExperimentResource* attribute), 223
 base_fields (*tardis.tardis_portal.api.FacilityResource* attribute), 223
 base_fields (*tardis.tardis_portal.api.GroupResource* attribute), 224
 base_fields (*tardis.tardis_portal.api.InstrumentResource* attribute), 224
 base_fields (*tardis.tardis_portal.api.LocationResource* attribute), 224
 base_fields (*tardis.tardis_portal.api.MyTardisModelResource* attribute), 225
 base_fields (*tardis.tardis_portal.api.ObjectACLResource* attribute), 225
 base_fields (*tardis.tardis_portal.api.ParameterNameResource* attribute), 225
 base_fields (*tardis.tardis_portal.api.ParameterResource* attribute), 225
 base_fields (*tardis.tardis_portal.api.ParameterSetResource* attribute), 225
 base_fields (*tardis.tardis_portal.api.ReplicaResource* attribute), 226

base_fields (*tardis.tardis_portal.api.SchemaResource* attribute), 226
 base_fields (*tardis.tardis_portal.api.StorageBoxAttributeResource* attribute), 227
 base_fields (*tardis.tardis_portal.api.StorageBoxOptionResource* attribute), 227
 base_fields (*tardis.tardis_portal.api.StorageBoxResource* attribute), 227
 base_fields (*tardis.tardis_portal.api.UserResource* attribute), 228
 base_fields (*tardis.tardis_portal.forms.AddUserPermissionsForm* attribute), 230
 base_fields (*tardis.tardis_portal.forms.CreateUserPermissionsForm* attribute), 231
 base_fields (*tardis.tardis_portal.forms.DatasetForm* attribute), 231
 base_fields (*tardis.tardis_portal.forms.ExperimentAuthor* attribute), 231
 base_fields (*tardis.tardis_portal.forms.ExperimentForm* attribute), 232
 base_fields (*tardis.tardis_portal.forms.ImportParamsForm* attribute), 233
 base_fields (*tardis.tardis_portal.forms.LoginForm* attribute), 233
 base_fields (*tardis.tardis_portal.forms.ManageAccountForm* attribute), 233
 base_fields (*tardis.tardis_portal.forms.ManageGroupPermissionsForm* attribute), 234
 base_fields (*tardis.tardis_portal.forms.RegisterExperimentForm* attribute), 234
 base_fields (*tardis.tardis_portal.forms.RegistrationForm* attribute), 235
 base_fields (*tardis.tardis_portal.forms.RightsForm* attribute), 235
 BaseProvider (class in *tardis.apps.oaipmh.provider.base*), 77, 247
 BaseProviderTestCase (class in *tardis.apps.oaipmh.tests.provider.test_base*), 80
 blank_param (*tardis.tardis_portal.ParameterSetManager.ParameterSetManager* attribute), 213
 BLEACH_ALLOWED_ATTRIBUTES (in module *tardis.default_settings.frontend*), 122
 BLEACH_ALLOWED_TAGS (in module *tardis.default_settings.frontend*), 122
 bleach_value () (in module *tardis.tardis_portal.templatetags.bleach_tag*), 184
 build_template_html () (in module *tardis.tardis_portal.email*), 230
 build_template_text () (in module *tardis.tardis_portal.email*), 230
 BUNDLE (*tardis.tardis_portal.models.storage.StorageBox* attribute), 176
 bytes_available () (in module *tardis.apps.push_to.utils*), 97
 C
 cache (*tardis.tardis_portal.models.storage.StorageBox* attribute), 176
 cache_box (*tardis.tardis_portal.models.storage.StorageBox* attribute), 177
 cache_file () (*tardis.tardis_portal.models.datafile.DataFile* method), 142
 cache_file () (*tardis.tardis_portal.models.datafile.DataFileObject* method), 145
 CACHES (in module *tardis.default_settings.caches*), 119
 calculate_checksums () (in module *tardis.apps.s3utils.utils*), 102
 calculate_checksums () (*tardis.tardis_portal.models.datafile.DataFileObject* method), 145
 CALCULATE_CHECKSUMS_METHODS (in module *tardis.default_settings.storage*), 123
 can_copy () (in module *tardis.apps.push_to.utils*), 97
 can_publish () (*tardis.tardis_portal.publish.provider.rifcsprovider.RifC* method), 183
 canDelete (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 138
 canonicalize () (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* method), 112
 canRead (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 138
 canWrite (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 138
 capture () (in module *tardis.tardis_portal.templatetags.capture*), 185
 CaptureNode (class in *tardis.tardis_portal.templatetags.capture*), 185
 cast_string_to_integer () (in module *tardis.tardis_portal.migrations.0001_squashed_0011_auto_2016*), 133
 cast_string_to_integer () (in module *tardis.tardis_portal.migrations.0005_datafile_add_size_int_column*), 133
 cert_signing_url (*tardis.apps.push_to.models.OAuthSSHCertSigning* attribute), 94
 check_app_dependencies () (in module *tardis.app_config*), 243
 check_auth_interactive () (*tardis.apps.sftp.sftp.MyTServerInterface* method), 114
 check_auth_interactive_response () (*tardis.apps.sftp.sftp.MyTServerInterface* method), 114
 check_auth_password ()

[\(tardis.apps.sftp.sftp.MyTServerInterface method\), 115](#)
[check_auth_publickey\(\) \(tardis.apps.sftp.sftp.MyTServerInterface method\), 115](#)
[check_channel_request\(\) \(tardis.apps.sftp.sftp.MyTServerInterface method\), 115](#)
[check_if_facility_manager\(\) \(in module tardis.tardis_portal.templatetags.facility_tags\), 185](#)
[check_if_user_not_approved\(\) \(in module tardis.tardis_portal.templatetags.approved_user_tags\), 184](#)
[check_paths\(\) \(tardis.tardis_portal.tests.slapd.Slapd class method\), 197](#)
[child_boxes\(tardis.tardis_portal.models.storage.StorageBox attribute\), 177](#)
[CHOICES\(tardis.tardis_portal.models.access_control.UserAuthentication attribute\), 139](#)
[choices\(tardis.tardis_portal.models.parameters.ParameterName attribute\), 171](#)
[classic_mapper\(\) \(in module tardis.tardis_portal.download\), 229](#)
[clean\(\) \(tardis.apps.sftp.forms.KeyAddForm method\), 110](#)
[clean\(\) \(tardis.tardis_portal.forms.RegistrationForm method\), 235](#)
[clean\(\) \(tardis.tardis_portal.forms.RightsForm method\), 235](#)
[clean\(\) \(tardis.tardis_portal.forms.StaticField method\), 235](#)
[clean_username\(\) \(tardis.tardis_portal.forms.RegistrationForm method\), 235](#)
[cleanup\(\) \(tardis.apps.filepicker.utils.FilepickerFile method\), 77](#)
[clear_children\(\) \(tardis.apps.sftp.sftp.DynamicTree method\), 112](#)
[close\(\) \(tardis.analytics.tracker.IteratorTracker method\), 75](#)
[close_gzip\(\) \(tardis.tardis_portal.download.UncachedTarStream method\), 229](#)
[close_request\(\) \(tardis.apps.sftp.sftp.MyTSFTPTCPServer method\), 114](#)
[CollectstaticTest \(class in tardis.tardis_portal.tests.management.test_collectstatic\), 191](#)
[Command\(class in tardis.apps.sftp.management.commands.sftpd\), 107](#)
[Command\(class in tardis.tardis_portal.management.commands.createuser\), 131](#)
[Command\(class in tardis.tardis_portal.management.commands.editpermissions\), 131](#)
[Command\(class in tardis.tardis_portal.management.commands.showschemas\), 131](#)
[Command\(class in tardis.tardis_portal.management.commands.rmexperiment\), 131](#)
[CommaSeparatedInput \(class in tardis.tardis_portal.widgets\), 240](#)
[comparison_type\(tardis.tardis_portal.models.parameters.ParameterName attribute\), 171](#)
[compliance_header\(\) \(in module tardis.tardis_portal.iiif\), 236](#)
[compress\(\) \(tardis.tardis_portal.download.UncachedTarStream method\), 229](#)
[compress\(\) \(tardis.tardis_portal.fields.MultiValueCommaSeparatedField method\), 230](#)
[compute_checksums\(\) \(in module tardis.tardis_portal.models.datafile\), 147](#)
[compute_etag\(\) \(in module tardis.tardis_portal.iiif\), 236](#)
[compute_size\(\) \(tardis.tardis_portal.download.UncachedTarStream method\), 229](#)
[configure\(\) \(tardis.tardis_portal.tests.slapd.Slapd method\), 197](#)
[configure_social_auth_user\(\) \(in module tardis.apps.social_auth.auth.social_auth\), 118](#)
[configure_user\(\) \(in module tardis.tardis_portal.auth.utils\), 130](#)
[confirm_migration\(\) \(in module tardis.apps.openid_migration.migration\), 86](#)
[CONTAINS_COMPARISON \(tardis.tardis_portal.models.parameters.ParameterName attribute\), 170](#)
[content_object\(tardis.tardis_portal.models.access_control.ObjectACL attribute\), 138](#)
[content_type\(tardis.tardis_portal.models.access_control.ObjectACL attribute\), 138](#)
[content_type_id\(tardis.tardis_portal.models.access_control.ObjectACL attribute\), 138](#)
[ContextualViewTest \(class in tardis.tardis_portal.tests.views.test_contextual_views\), 195](#)
[create_datafile\(\) \(tardis.tardis_portal.models.datafile.DataFileObject method\), 145](#)
[copy_files\(\) \(tardis.tardis_portal.models.storage.StorageBox method\), 177](#)
[copy_to_master\(\) \(tardis.tardis_portal.models.storage.StorageBox method\), 177](#)
[CopyMoveTestCase \(class in tardis.tardis_portal.tests.test_copy_move\), 199](#)
[create_acl\(\) \(tardis.apps.sftp.api.SFTPAclAuthorization method\), 108](#)
[create_schema\(\) \(tardis.tardis_portal.api.AclAuthorization method\), 218](#)
[create_schema_element\(\) \(in module](#)

[tardis.tardis_portal.views.pages](#)), 210
[create_group\(\)](#) (in module [tardis.tardis_portal.views.authorisation](#)), 207
[create_list\(\)](#) ([tardis.tardis_portal.api.ACLAuthorization](#) method), 218
[create_local_box\(\)](#) ([tardis.tardis_portal.models.storage.StorageBox](#) class method), 177
[create_OpenidUserMigration\(\)](#) ([tardis.apps.openid_migration.tests.test_models.ModelTestCases](#) method), 85
[create_parameter_add_form\(\)](#) (in module [tardis.tardis_portal.forms](#)), 236
[create_parameterset_edit_form\(\)](#) (in module [tardis.tardis_portal.forms](#)), 236
[create_set_uri\(\)](#) ([tardis.tardis_portal.models.datafile.DataFileObject](#) method), 145
[create_user\(\)](#) (in module [tardis.tardis_portal.auth.utils](#)), 130
[create_user\(\)](#) (in module [tardis.tardis_portal.views.authentication](#)), 207
[create_user_api_key\(\)](#) (in module [tardis.tardis_portal.models.access_control](#)), 141
[create_user_profile\(\)](#) (in module [tardis.tardis_portal.models.access_control](#)), 141
[created_by](#) ([tardis.tardis_portal.models.experiment.Experiment](#) attribute), 151
[created_by_id](#) ([tardis.tardis_portal.models.experiment.Experiment](#) attribute), 151
[created_time](#) ([tardis.tardis_portal.models.datafile.DataFileObject](#) attribute), 142
[created_time](#) ([tardis.tardis_portal.models.datafile.DataFileObject](#) attribute), 145
[created_time](#) ([tardis.tardis_portal.models.dataset.Dataset](#) attribute), 147
[created_time](#) ([tardis.tardis_portal.models.experiment.Experiment](#) attribute), 151
[created_time](#) ([tardis.tardis_portal.models.facility.Facility](#) attribute), 156
[created_time](#) ([tardis.tardis_portal.models.instrument.Instrument](#) attribute), 157
[created_time](#) ([tardis.tardis_portal.models.jti.JTI](#) attribute), 158
[createLinkedUserAuthenticationForm\(\)](#) (in module [tardis.tardis_portal.forms](#)), 236
[CreateTestCase](#) (class in [tardis.apps.related_info.tests.tests](#)), 99
[CreateUserPermissionsForm](#) (class in [tardis.tardis_portal.forms](#)), 230
[CreateUserTestCase](#) (class in [tardis.tardis_portal.tests.management.test_createuser](#)), 191
[Credential](#) (class in [tardis.apps.push_to.models](#)), 91
[Credential.DoesNotExist](#), 91
[Credential.MultipleObjectsReturned](#), 91
[credential_set](#) ([tardis.apps.push_to.models.RemoteHost](#) attribute), 95
[CredentialAdmin](#) (class in [tardis.apps.push_to.models](#)), 92
[CredentialForm](#) (class in [tardis.apps.push_to.models](#)), 92
[CredentialForm.Meta](#) (class in [tardis.apps.push_to.models](#)), 92
[cyberduck_connection_window\(\)](#) (in module [tardis.apps.sftp.views](#)), 117

D

[DataFileObject](#)
[data_type](#) ([tardis.tardis_portal.models.parameters.ParameterName](#) attribute), 171
[DATABASE_ENGINE](#) ([tardis.default_settings](#) attribute), 45
[DATABASE_HOST](#) ([tardis.default_settings](#) attribute), 45
[DATABASE_NAME](#) ([tardis.default_settings](#) attribute), 45
[DATABASE_PASSWORD](#) ([tardis.default_settings](#) attribute), 45
[DATABASE_PORT](#) ([tardis.default_settings](#) attribute), 45
[DATABASE_USER](#) ([tardis.default_settings](#) attribute), 45
[DataFile](#) (class in [tardis.tardis_portal.models.datafile](#)), 141
[DataFile](#) ([tardis.tardis_portal.models.datafile.DataFileObject](#) attribute), 145
[DataFileParameterSet](#) (class in [tardis.tardis_portal.models.parameters](#)), 161
[DATAFILE](#) ([tardis.tardis_portal.models.parameters.Schema](#) attribute), 174
[DataFile.DoesNotExist](#), 142
[DataFile.MultipleObjectsReturned](#), 142
[datafile_access_required\(\)](#) (in module [tardis.tardis_portal.auth.decorators](#)), 126
[datafile_id](#) ([tardis.tardis_portal.models.datafile.DataFileObject](#) attribute), 145
[datafile_id](#) ([tardis.tardis_portal.models.parameters.DatafileParameterSet](#) attribute), 161
[datafile_set](#) ([tardis.tardis_portal.models.dataset.Dataset](#) attribute), 147
[DatafileAdmin](#) (class in [tardis.tardis_portal.admin](#)), 214
[DatafileAdminForm](#) (class in [tardis.tardis_portal.admin](#)), 214
[DatafileAdminForm.Meta](#) (class in [tardis.tardis_portal.admin](#)), 214
[DataFileDocument](#) (class in [tardis.apps.search.documents](#)), 105

DataFileDocument.Django	(class in <i>tardis.apps.search.documents</i>), 105	<i>tardis.tardis_portal.tests.api.test_datafile_metadata_resources</i>), 187
DataFileDocument.Index	(class in <i>tardis.apps.search.documents</i>), 105	datafileparametersets (<i>tardis.tardis_portal.models.storage.StorageBox</i> attribute), 177
DataFileObject	(class in <i>tardis.tardis_portal.models.datafile</i>), 144	DataFileResource (class in <i>tardis.tardis_portal.api</i>), 219
DataFileObject.DoesNotExist	, 145	DataFileResource.Meta (class in <i>tardis.tardis_portal.api</i>), 219
DataFileObject.MultipleObjectsReturned	, 145	DataFileResourceTest (class in <i>tardis.tardis_portal.tests.api.test_datafile_resource</i>), 187
DataFileObjectInline	(class in <i>tardis.tardis_portal.admin</i>), 213	DatafileSaveMetadataTestCase (class in <i>tardis.tardis_portal.tests.test_df_save_metadata</i>), 200
DataFileObjectInlineForm	(class in <i>tardis.tardis_portal.admin</i>), 213	DataFileTestCase (class in <i>tardis.tardis_portal.tests.models.test_datafile</i>), 193
DataFileObjectInlineForm.Meta	(class in <i>tardis.tardis_portal.admin</i>), 213	Dataset (class in <i>tardis.tardis_portal.models.dataset</i>), 147
DataFileObjectTestCase	(class in <i>tardis.tardis_portal.tests.models.test_dfo</i>), 193	dataset (<i>tardis.tardis_portal.models.datafile.DataFile</i> attribute), 142
DatafileParameter	(class in <i>tardis.tardis_portal.models.parameters</i>), 160	dataset (<i>tardis.tardis_portal.models.parameters.DatasetParameterSet</i> attribute), 163
DatafileParameter.DoesNotExist	, 160	DATASET (<i>tardis.tardis_portal.models.parameters.Schema</i> attribute), 174
DatafileParameter.MultipleObjectsReturned	, 160	Dataset.DoesNotExist, 147
datafileparameter_set	(<i>tardis.tardis_portal.models.parameters.DatafileParameterSet</i> attribute), 161	Dataset.MultipleObjectsReturned, 147
datafileparameter_set	(<i>tardis.tardis_portal.models.parameters.ParameterSet</i> attribute), 171	dataset_access_required() (in module <i>tardis.tardis_portal.auth.decorators</i>), 126
DatafileParameterResource	(class in <i>tardis.tardis_portal.api</i>), 220	dataset_aggregate_info() (in module <i>tardis.tardis_portal.views.facilities</i>), 208
DatafileParameterResource.Meta	(class in <i>tardis.tardis_portal.api</i>), 220	dataset_download_required() (in module <i>tardis.tardis_portal.auth.decorators</i>), 126
DatafileParameterResourceTest	(class in <i>tardis.tardis_portal.tests.api.test_datafile_metadata_resources</i>), 187	dataset_id (<i>tardis.tardis_portal.models.datafile.DataFile</i> attribute), 142
DatafileParameterSet	(class in <i>tardis.tardis_portal.models.parameters</i>), 161	dataset_id (<i>tardis.tardis_portal.models.parameters.DatasetParameterSet</i> attribute), 163
DatafileParameterSet.DoesNotExist	, 161	dataset_set (<i>tardis.tardis_portal.models.instrument.Instrument</i> attribute), 157
DatafileParameterSet.MultipleObjectsReturned	, 161	DATASET_VIEWS (in module <i>tardis.default_settings.custom_views</i>), 119
datafileparameterset_set	(<i>tardis.tardis_portal.models.datafile.DataFile</i> attribute), 142	dataset_write_permissions_required() (in module <i>tardis.tardis_portal.auth.decorators</i>), 126
datafileparameterset_set	(<i>tardis.tardis_portal.models.parameters.Schema</i> attribute), 175	DatasetAdmin (class in <i>tardis.tardis_portal.admin</i>), 214
DatafileParameterSetResource	(class in <i>tardis.tardis_portal.api</i>), 220	DatasetDocument (class in <i>tardis.apps.search.documents</i>), 106
DatafileParameterSetResource.Meta	(class in <i>tardis.tardis_portal.api</i>), 220	DatasetDocument.Django (class in <i>tardis.apps.search.documents</i>), 106
DatafileParameterSetResourceTest	(class in <i>tardis.tardis_portal.tests.api.test_datafile_metadata_resources</i>), 187	DatasetDocument.Index (class in <i>tardis.apps.search.documents</i>), 106
		DatasetForm (class in <i>tardis.tardis_portal.forms</i>), 231

DatasetForm.Meta	(class in tardis.tardis_portal.forms), 231	DatasetView	(class in tardis.tardis_portal.views.pages), 208
DatasetParameter	(class in tardis.tardis_portal.models.parameters), 162	DATETIME	(tardis.tardis_portal.models.parameters.ParameterName attribute), 170
DatasetParameter.DoesNotExist	, 162	datetime_to_us()	(in module tardis.tardis_portal.views.facilities), 208
DatasetParameter.MultipleObjectsReturned	, 162	datetime_value	(tardis.tardis_portal.models.parameters.Parameter attribute), 169
datasetparameter_set	(tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 163	DefaultPolicy	(class in tardis.apps.push_to.models), 92
datasetparameter_set	(tardis.tardis_portal.models.parameters.ParameterName attribute), 171	DcExperimentProvider	(class in tardis.apps.oaipmh.provider.experiment), 79
DatasetParameterResource	(class in tardis.tardis_portal.api), 220	DcExperimentProviderTestCase	(class in tardis.apps.oaipmh.tests.provider.test_experiment), 81
DatasetParameterResource.Meta	(class in tardis.tardis_portal.api), 220	DEBUG	(in module tardis.default_settings.debug), 120
DatasetParameterResourceTest	(class in tardis.tardis_portal.tests.api.test_dataset_metadata_resource), 187	declared_fields	(tardis.apps.push_to.models.CredentialForm attribute), 92
DatasetParameterSet	(class in tardis.tardis_portal.models.parameters), 163	declared_fields	(tardis.apps.related_info.forms.RelatedInfoForm attribute), 100
DatasetParameterSet.DoesNotExist	, 163	declared_fields	(tardis.apps.s3utils.api.ReplicaAppResource attribute), 101
DatasetParameterSet.MultipleObjectsReturned	, 163	declared_fields	(tardis.apps.search.api.AdvanceSearchAppResource attribute), 104
datasetparameterset_set	(tardis.tardis_portal.models.dataset.Dataset attribute), 148	declared_fields	(tardis.apps.search.api.SearchAppResource attribute), 105
datasetparameterset_set	(tardis.tardis_portal.models.parameters.Schema attribute), 175	declared_fields	(tardis.apps.sftp.api.SFTPPublicKeyAppResource attribute), 109
DatasetParameterSetResource	(class in tardis.tardis_portal.api), 221	declared_fields	(tardis.apps.sftp.forms.KeyAddForm attribute), 110
DatasetParameterSetResource.Meta	(class in tardis.tardis_portal.api), 221	declared_fields	(tardis.apps.sftp.forms.KeyGenerateForm attribute), 110
DatasetParameterSetResourceTest	(class in tardis.tardis_portal.tests.api.test_dataset_metadata_resource), 187	declared_fields	(tardis.tardis_portal.admin.DatafileAdminForm attribute), 214
datasetparametersets	(tardis.tardis_portal.models.storage.StorageBox attribute), 177	declared_fields	(tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 214
DatasetResource	(class in tardis.tardis_portal.api), 221	declared_fields	(tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 217
DatasetResource.Meta	(class in tardis.tardis_portal.api), 221	declared_fields	(tardis.tardis_portal.admin.StorageBoxForm attribute), 217
DatasetResourceTest	(class in tardis.tardis_portal.tests.api.test_dataset_resource), 188	declared_fields	(tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 218
datasets	(tardis.tardis_portal.models.experiment.Experiment attribute), 151	declared_fields	(tardis.tardis_portal.api.DatafileParameterResource attribute), 220
DatasetTestCase	(class in tardis.tardis_portal.tests.models.test_dataset),	declared_fields	(tardis.tardis_portal.api.DatafileParameterSetResource attribute), 220
		declared_fields	(tardis.tardis_portal.api.DataFileResource attribute), 219
		declared_fields	(tardis.tardis_portal.api.DatasetParameterResource attribute), 221
		declared_fields	(tardis.tardis_portal.api.DatasetParameterSetResource attribute), 221

dependencies (`tardis.apps.openid_migration.migrations.0001_initial.Migration` methods
 attribute), 84 (tardis.apps.sftp.api.SFTTPublicKeyAppResource.Meta
 attribute), 109
 dependencies (`tardis.apps.push_to.migrations.0001_initial.Migration` methods
 attribute), 90 detail_uri_kwargs ()
 dependencies (`tardis.apps.push_to.migrations.0002_auto_20160518_1953.Migration` methods
 attribute), 90 AdvanceSearchAppResource
 method), 104
 dependencies (`tardis.apps.sftp.migrations.0001_initial.Migration` methods
 attribute), 108 detail_uri_kwargs ()
 dependencies (`tardis.tardis_portal.migrations.0001_initial.Migration` methods
 attribute), 105 (tardis.apps.search.api.SearchAppResource
 method), 105
 dependencies (`tardis.tardis_portal.migrations.0001_squashed_001650528_1128.Migration` methods
 attribute), 132 df_dir_only () (in module
 dependencies (`tardis.tardis_portal.migrations.0001_squashed_001650528_1128.Migration` methods
 attribute), 132 tardis_portal.models.datafile.DataFile
 attribute), 132 only_mapper), 76
 dependencies (`tardis.tardis_portal.migrations.0002_auto_20150528_1128.Migration` methods
 attribute), 133 attribute), 142
 dependencies (`tardis.tardis_portal.migrations.0003_auto_20150907_1318.Migration` methods
 attribute), 133 tardis_portal.models.dataset.Dataset
 attribute), 148
 dependencies (`tardis.tardis_portal.migrations.0004_storage_box_type.Migration` methods
 attribute), 133 tardis.storage.StorageBox
 attribute), 176
 dependencies (`tardis.tardis_portal.migrations.0005_datafile_and_search_index.Migration` methods
 attribute), 133 DataFileDocument
 attribute), 106
 dependencies (`tardis.tardis_portal.migrations.0006_datafile_and_search_index.Migration` methods
 attribute), 133 tardis.apps.search.index.DatasetDocument
 attribute), 106
 dependencies (`tardis.tardis_portal.migrations.0007_remove_old_experiment.Migration` methods
 attribute), 134 tardis.apps.experiment.ExperimentDocument
 attribute), 106
 dependencies (`tardis.tardis_portal.migrations.0008_string_value_partials.Migration` methods
 attribute), 134 postgres.Migration
 attribute), 134 (tardis.tardis_portal.models.storage.StorageBox
 attribute), 134 attribute), 134
 dependencies (`tardis.tardis_portal.migrations.0009_auto_20160128_1119.Migration` methods
 attribute), 134 DjangoAuthBackend (class in
 dependencies (`tardis.tardis_portal.migrations.0010_auto_20160503_1443.Migration` methods
 attribute), 134 auth.localdb_auth), 129
 dependencies (`tardis.tardis_portal.migrations.0011_auto_20160505_1643.Migration` methods
 attribute), 134 DjangoGroupProvider (class in
 dependencies (`tardis.tardis_portal.migrations.0012_user_authentication.Migration` methods
 attribute), 135 auth.localdb_auth), 129
 dependencies (`tardis.tardis_portal.migrations.0013_auto_20181002_1336.Migration` methods
 attribute), 135 do_file_copy () (in module
 dependencies (`tardis.tardis_portal.migrations.0014_auto_20181002_1336.Migration` methods
 attribute), 135 tardis_portal.tasks), 97
 dependencies (`tardis.tardis_portal.migrations.0014_auto_20181002_1336.Migration` methods
 attribute), 135 do_migration () (in module
 dependencies (`tardis.tardis_portal.migrations.0015_data_created_time.Migration` methods
 attribute), 135 tardis_portal.migration.migration), 86
 dependencies (`tardis.tardis_portal.migrations.0016_add_time_stamp.Migration` methods
 attribute), 135 (in module
 dependencies (`tardis.tardis_portal.migrations.0017_add_license.Migration` methods
 attribute), 135 tardis.tardis_portal.download), 229
 dependencies (`tardis.tardis_portal.migrations.0018_make_default_storage_backend.Migration` methods
 attribute), 136 tardis.tardis_portal.download), 229
 dependencies (`tardis.tardis_portal.migrations.0018_make_default_storage_backend.Migration` methods
 attribute), 136 tardis.apps.s3utils.api.ReplicaAppResource
 method), 101
 dependencies (`tardis.tardis_portal.migrations.0018_make_default_storage_backend.Migration` methods
 attribute), 136 tardis.tardis_portal.models.datafile.DataFileResource
 method), 219
 description (`tardis.tardis_portal.models.dataset.Dataset` download_image () (in module
 attribute), 148 tardis.tardis_portal.iiif), 236
 description (`tardis.tardis_portal.models.experiment.Experiment` download_info () (in module
 attribute), 152 tardis.tardis_portal.iiif), 236
 description (`tardis.tardis_portal.models.storage.StorageBox` DOWNLOAD_URI_TEMPLATES (in module
 attribute), 178 tardis.default_settings.downloads), 121
 deserialize () (`tardis.tardis_portal.api.DataFileResource` download_test_case (class in
 method), 219 tardis.tardis_portal.tests.test_download),

200

DummyStorage (class in *tardis.tardis_portal.storage*), 184

DumpSchemasTestCase (class in *tardis.tardis_portal.tests.management.test_dumpschemas*), 192

DynamicTree (class in *tardis.apps.sftp.sftp*), 112

dynurl () (in module *tardis.tardis_portal.templatetags.dynurl*), 185

DynUrlNode (class in *tardis.tardis_portal.templatetags.dynurl*), 185

E

edit_auth_method () (in module *tardis.tardis_portal.auth.authentication*), 124

edit_datafile_par () (in module *tardis.tardis_portal.views.parameters*), 211

edit_dataset () (in module *tardis.tardis_portal.views.pages*), 210

edit_dataset_par () (in module *tardis.tardis_portal.views.parameters*), 211

edit_experiment_par () (in module *tardis.tardis_portal.views.parameters*), 212

edit_parameters () (in module *tardis.tardis_portal.views.parameters*), 212

EditParameterSetTestCase (class in *tardis.tardis_portal.tests.test_parametersets*), 203

effectiveDate (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 138

ELASTICSEARCH_PARALLEL_INDEX_SETTINGS (in module *tardis.default_settings.search*), 122

email (*tardis.tardis_portal.models.experiment.ExperimentAuthor* attribute), 155

EMAIL_HOST (in module *tardis.default_settings.email*), 121

EMAIL_HOST_PASSWORD (in module *tardis.default_settings.email*), 121

EMAIL_HOST_USER (in module *tardis.default_settings.email*), 121

email_migration_success () (in module *tardis.apps.openid_migration.email_text*), 86

EMAIL_PORT (in module *tardis.default_settings.email*), 121

EMAIL_USE_TLS (in module *tardis.default_settings.email*), 121

email_user () (in module *tardis.tardis_portal.email*), 230

end_time (*tardis.tardis_portal.models.experiment.Experiment* attribute), 152

endpoint () (in module *tardis.apps.oaipmh.views*), 84

EndpointTestCase (class in *tardis.apps.oaipmh.tests.test_oai*), 81

entityId (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 138

error_handler () (in module *tardis.views*), 244

EXACT_VALUE_COMPARISON (in module *tardis.tardis_portal.models.parameters.ParameterName* attribute), 171

Experiment (class in *tardis.tardis_portal.models.experiment*), 150

experiment (*tardis.tardis_portal.models.experiment.ExperimentAuthor* attribute), 155

experiment (*tardis.tardis_portal.models.parameters.ExperimentParameter* attribute), 165

EXPERIMENT (*tardis.tardis_portal.models.parameters.Schema* attribute), 174

experiment (*tardis.tardis_portal.models.token.Token* attribute), 181

Experiment.DoesNotExist, 151

Experiment.MultipleObjectsReturned, 151

experiment_access_required () (in module *tardis.tardis_portal.auth.decorators*), 126

experiment_authors () (in module *tardis.tardis_portal.templatetags.experiment_tags*), 185

experiment_download_link () (in module *tardis.tardis_portal.templatetags.experiment_tags*), 185

experiment_download_required () (in module *tardis.tardis_portal.auth.decorators*), 126

experiment_file_count () (in module *tardis.tardis_portal.templatetags.experimentstats*), 185

experiment_id (*tardis.tardis_portal.models.experiment.ExperimentAuthor* attribute), 155

experiment_id (*tardis.tardis_portal.models.parameters.ExperimentParameter* attribute), 165

experiment_id (*tardis.tardis_portal.models.token.Token* attribute), 182

experiment_ownership_required () (in module *tardis.tardis_portal.auth.decorators*), 126

experiment_set (*tardis.tardis_portal.models.license.License* attribute), 159

EXPERIMENT_VIEWS (in module *tardis.default_settings.custom_views*), 120

ExperimentAdmin (class in *tardis.tardis_portal.admin*), 214

ExperimentAuthor (class in *tardis.tardis_portal.forms*), 231

ExperimentAuthor (class in *tardis.tardis_portal.models.experiment*), 154

ExperimentAuthor.DoesNotExist, 154
 ExperimentAuthor.Meta (class in *tardis.tardis_portal.forms*), 231
 ExperimentAuthor.MultipleObjectsReturned, 154
 experimentauthor_set (*tardis.tardis_portal.models.experiment.Experiment* attribute), 152
 ExperimentAuthorResource (class in *tardis.tardis_portal.api*), 222
 ExperimentAuthorResource.Meta (class in *tardis.tardis_portal.api*), 222
 ExperimentDocument (class in *tardis.apps.search.documents*), 106
 ExperimentDocument.Django (class in *tardis.apps.search.documents*), 106
 ExperimentDocument.Index (class in *tardis.apps.search.documents*), 106
 ExperimentForm (class in *tardis.tardis_portal.forms*), 232
 ExperimentForm.FullExperiment (class in *tardis.tardis_portal.forms*), 232
 ExperimentForm.Meta (class in *tardis.tardis_portal.forms*), 232
 ExperimentListsTest (class in *tardis.tardis_portal.tests.views.test_template_contexts*), 195
 ExperimentManager (class in *tardis.tardis_portal.managers*), 236
 ExperimentParameter (class in *tardis.tardis_portal.models.parameters*), 164
 ExperimentParameter.DoesNotExist, 164
 ExperimentParameter.MultipleObjectsReturned, 164
 experimentparameter_set (*tardis.tardis_portal.models.parameters.ExperimentParameterSet* attribute), 166
 experimentparameter_set (*tardis.tardis_portal.models.parameters.ParameterSet* attribute), 172
 ExperimentParameterInline (class in *tardis.tardis_portal.admin*), 215
 ExperimentParameterResource (class in *tardis.tardis_portal.api*), 222
 ExperimentParameterResource.Meta (class in *tardis.tardis_portal.api*), 222
 ExperimentParameterResourceTest (class in *tardis.tardis_portal.tests.api.test_experiment_metadata_resource*), 188
 ExperimentParameterSet (class in *tardis.tardis_portal.models.parameters*), 165
 ExperimentParameterSet.DoesNotExist, 165
 ExperimentParameterSet.MultipleObjectsReturned, 165
 experimentparameterset_set (*tardis.tardis_portal.models.parameters.Schema* attribute), 175
 ExperimentParameterSetAdmin (class in *tardis.tardis_portal.admin*), 215
 ExperimentParameterSetResource (class in *tardis.tardis_portal.api*), 222
 ExperimentParameterSetResource.Meta (class in *tardis.tardis_portal.api*), 222
 ExperimentParameterSetResourceTest (class in *tardis.tardis_portal.tests.api.test_experiment_metadata_resource*), 188
 experimentparametersets (*tardis.tardis_portal.models.storage.StorageBox* attribute), 178
 ExperimentResource (class in *tardis.tardis_portal.api*), 222
 ExperimentResource.Meta (class in *tardis.tardis_portal.api*), 223
 ExperimentResourceTest (class in *tardis.tardis_portal.tests.api.test_experiment_resource*), 188
 experiments (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* attribute), 113
 experiments (*tardis.tardis_portal.models.dataset.Dataset* attribute), 148
 ExperimentTestCase (class in *tardis.tardis_portal.tests.models.test_experiment*), 193
 ExperimentTestCase (class in *tardis.tardis_portal.tests.views.test_experiment_views*), 193
 ExperimentView (class in *tardis.tardis_portal.views.pages*), 209
 expiry_date (*tardis.tardis_portal.models.token.Token* attribute), 182
 expiryDate (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 138
 ext_groups (*tardis.tardis_portal.models.access_control.UserProfile* attribute), 140
 external_users () (*tardis.tardis_portal.managers.ExperimentManager* method), 237
 extra (*tardis.tardis_portal.admin.DataFileObjectInline* attribute), 213
 extra (*tardis.tardis_portal.admin.ExperimentParameterInline* attribute), 215
 extra (*tardis.tardis_portal.admin.InstrumentParameterInline* attribute), 215
 extra (*tardis.tardis_portal.admin.ObjectACLInline* at-

- tribute), 216
- extra (tardis.tardis_portal.admin.ParameterNameInline attribute), 216
- extra (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 216
- extra (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 217
- ExtraTestCases (class in tardis.tardis_portal.tests.test_iif), 201
- ## F
- facilities_managed_by() (in module tardis.tardis_portal.models.facility), 156
- Facility (class in tardis.tardis_portal.models.facility), 155
- facility (tardis.tardis_portal.models.instrument.Instrument attribute), 157
- Facility.DoesNotExist, 155
- Facility.MultipleObjectsReturned, 156
- facility_id (tardis.tardis_portal.models.instrument.Instrument attribute), 157
- facility_overview() (in module tardis.tardis_portal.views.pages), 210
- facility_overview_data_count() (in module tardis.tardis_portal.views.facilities), 208
- facility_overview_datafile_list() (in module tardis.tardis_portal.views.facilities), 208
- facility_overview_dataset_detail() (in module tardis.tardis_portal.views.facilities), 208
- facility_overview_experiments() (in module tardis.tardis_portal.views.facilities), 208
- facility_overview_facilities_list() (in module tardis.tardis_portal.views.facilities), 208
- FacilityAdmin (class in tardis.tardis_portal.admin), 215
- FacilityOverviewTestCase (class in tardis.tardis_portal.tests.test_facility_overview), 201
- FacilityResource (class in tardis.tardis_portal.api), 223
- FacilityResource.Meta (class in tardis.tardis_portal.api), 223
- FacilityResourceTest (class in tardis.tardis_portal.tests.api.test_facility_resource), 188
- feedback() (in module tardis.tardis_portal.views.utils), 212
- fields (tardis.apps.push_to.models.CredentialForm.Meta attribute), 92
- fields (tardis.apps.push_to.models.RemoteHostAdmin attribute), 96
- fields (tardis.tardis_portal.admin.DatafileAdminForm.Meta attribute), 214
- fields (tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta attribute), 214
- fields (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta attribute), 217
- fields (tardis.tardis_portal.admin.StorageBoxForm.Meta attribute), 217
- fields (tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute), 218
- fields (tardis.tardis_portal.api.UserResource.Meta attribute), 227
- fields (tardis.tardis_portal.forms.DatasetForm.Meta attribute), 231
- fields (tardis.tardis_portal.forms.ExperimentAuthor.Meta attribute), 231
- fields (tardis.tardis_portal.forms.ExperimentForm.Meta attribute), 232
- fields (tardis.tardis_portal.forms.ManageAccountForm.Meta attribute), 233
- fields (tardis.tardis_portal.forms.RightsForm.Meta attribute), 235
- file_object (tardis.tardis_portal.models.datafile.DataFile attribute), 142
- file_object (tardis.tardis_portal.models.datafile.DataFileObject attribute), 145
- file_objects (tardis.tardis_portal.models.datafile.DataFile attribute), 142
- file_objects (tardis.tardis_portal.models.storage.StorageBox attribute), 178
- filename (tardis.tardis_portal.models.datafile.DataFile attribute), 143
- FILENAME (tardis.tardis_portal.models.parameters.ParameterName attribute), 171
- filepicker_url_regex (tardis.apps.filepicker.utils.FilepickerFile attribute), 77
- FilepickerFile (class in tardis.apps.filepicker.utils), 77
- filtering (tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta attribute), 109
- filtering (tardis.tardis_portal.api.DataFileResource.Meta attribute), 219
- filtering (tardis.tardis_portal.api.DatasetResource.Meta attribute), 221
- filtering (tardis.tardis_portal.api.ExperimentAuthorResource.Meta attribute), 222
- filtering (tardis.tardis_portal.api.ExperimentResource.Meta attribute), 223
- filtering (tardis.tardis_portal.api.FacilityResource.Meta attribute), 223
- filtering (tardis.tardis_portal.api.GroupResource.Meta attribute), 223
- filtering (tardis.tardis_portal.api.InstrumentResource.Meta attribute), 223

[attribute](#)), 224
[filtering \(tardis.tardis_portal.api.ObjectACLResource.Meta attribute\)](#), 225
[filtering \(tardis.tardis_portal.api.ParameterNameResource.Meta attribute\)](#), 225
[filtering \(tardis.tardis_portal.api.ReplicaResource.Meta attribute\)](#), 226
[filtering \(tardis.tardis_portal.api.SchemaResource.Meta attribute\)](#), 226
[filtering \(tardis.tardis_portal.api.UserResource.Meta attribute\)](#), 227
[FILTERS_TASK_PRIORITY \(in module tardis.default_settings.filters\)](#), 121
[find_available_tcp_port\(\) \(in module tardis.tardis_portal.tests.slapped\)](#), 198
[find_custom_view_override\(\) \(tardis.tardis_portal.views.pages.DatasetView method\)](#), 208
[find_custom_view_override\(\) \(tardis.tardis_portal.views.pages.ExperimentView method\)](#), 209
[form \(tardis.apps.push_to.models.CredentialAdmin attribute\)](#), 92
[form \(tardis.tardis_portal.admin.DatafileAdmin attribute\)](#), 214
[form \(tardis.tardis_portal.admin.DataFileObjectInline attribute\)](#), 213
[form \(tardis.tardis_portal.admin.StorageBoxAdmin attribute\)](#), 216
[form \(tardis.tardis_portal.admin.StorageBoxAttributeInline attribute\)](#), 216
[form \(tardis.tardis_portal.admin.StorageBoxOptionInline attribute\)](#), 218
[format_app_name_for_url\(\) \(in module tardis.app_config\)](#), 244
[formfield_overrides \(tardis.tardis_portal.admin.ExperimentParameterInline attribute\)](#), 215
[forwards_func\(\) \(in module tardis.tardis_portal.migrations.0017_add_cc_licenses\)](#), 136
[FreeTextSearchField \(class in tardis.tardis_portal.models.parameters\)](#), 166
[FreeTextSearchField.DoesNotExist](#), 166
[FreeTextSearchField.MultipleObjectsReturned](#), 166
[freetextsearchfield_set \(tardis.tardis_portal.models.parameters.ParameterName attribute\)](#), 172
[FreeTextSearchFieldAdmin \(class in tardis.tardis_portal.admin\)](#), 215
[freeze_time\(\) \(tardis.tardis_portal.tests.test_tokens.FrozenTime class method\)](#), 205
[FrozenTime \(class in tardis.tardis_portal.tests.test_tokens\)](#), 205
[full_name \(tardis.tardis_portal.models.parameters.ParameterName attribute\)](#), 172
[generate_keypair_credential\(\) \(tardis.apps.push_to.models.Credential static method\)](#), 91
[generate_presigned_url\(\) \(in module tardis.apps.s3utils.utils\)](#), 102
[get\(\) \(tardis.tardis_portal.managers.ExperimentManager method\)](#), 237
[get\(\) \(tardis.tardis_portal.models.parameters.Parameter method\)](#), 169
[get\(\) \(tardis.tardis_portal.views.pages.DatasetView method\)](#), 209
[get\(\) \(tardis.tardis_portal.views.pages.ExperimentView method\)](#), 209
[get\(\) \(tardis.tardis_portal.views.pages.IndexView method\)](#), 210
[get_absolute_filepath\(\) \(tardis.tardis_portal.models.datafile.DataFile method\)](#), 143
[get_absolute_url\(\) \(tardis.tardis_portal.models.dataset.Dataset method\)](#), 148
[get_absolute_url\(\) \(tardis.tardis_portal.models.experiment.Experiment method\)](#), 152
[get_accessible_datafiles_for_user\(\) \(in module tardis.tardis_portal.auth.decorators\)](#), 127
[get_accessible_experiments\(\) \(in module tardis.tardis_portal.auth.decorators\)](#), 127
[get_accessible_experiments_for_dataset\(\) \(in module tardis.tardis_portal.auth.decorators\)](#), 127
[get_accessible_hosts\(\) \(in module tardis.apps.push_to.views\)](#), 97
[get_aclOwnershipType_display\(\) \(tardis.tardis_portal.models.access_control.ObjectACL method\)](#), 138
[get_address\(\) \(tardis.tardis_portal.tests.slapped.Slapped method\)](#), 197
[get_admin_credentials\(\) \(tardis.tardis_portal.tests.api.MyTardisResourceTestCase method\)](#), 191
[get_all_storage_boxes_used\(\) \(tardis.tardis_portal.models.dataset.Dataset method\)](#), 148
[get_all_tardis_apps\(\) \(in module tardis.test_settings\)](#), 244
[get_allowed_auths\(\)](#)

(*tardis.apps.sftp.sftp.MyTServerInterface* method), 116

get_api_key() (in module *tardis.apps.openid_migration.migration*), 86

get_apikey_credentials() (*tardis.tardis_portal.tests.api.MyTardisResourceTestCase* method), 191

get_as_temporary_file() (*tardis.tardis_portal.models.datafile.DataFile* method), 143

get_auth_method() (in module *tardis.apps.social_auth.auth.social_auth*), 118

get_available_signing_services() (*tardis.apps.push_to.models.OAuthSSHCertSigningService* static method), 94

get_beamline() (*tardis.tardis_portal.tests.test_publishservice.MachRefCephView* method), 204

get_by_natural_key() (*tardis.tardis_portal.managers.ParameterNameManager* method), 238

get_by_natural_key() (*tardis.tardis_portal.managers.SchemaManager* method), 238

get_child_dir_files() (*tardis.tardis_portal.api.DatasetResource* method), 221

get_child_dir_nodes() (*tardis.tardis_portal.api.DatasetResource* method), 221

get_client_for_host() (*tardis.apps.push_to.models.Credential* method), 91

get_client_ip() (in module *tardis.tardis_portal.logging_middleware*), 236

get_comparison_type_display() (*tardis.tardis_portal.models.parameters.ParameterName* method), 172

get_context() (*tardis.tardis_portal.publish.publishservice.PublishService* method), 183

get_context_data() (*tardis.tardis_portal.views.pages.DatasetView* method), 209

get_context_data() (*tardis.tardis_portal.views.pages.ExperimentView* method), 210

get_context_data() (*tardis.tardis_portal.views.pages.IndexView* method), 210

get_create_token_url() (*tardis.tardis_portal.models.experiment.Experiment* method), 152

get_credential() (in module *tardis.apps.push_to.views*), 97

get_credentials() (*tardis.tardis_portal.tests.api.MyTardisResourceTestCase* method), 191

get_ct() (*tardis.tardis_portal.models.experiment.Experiment* method), 152

get_data_type_display() (*tardis.tardis_portal.models.parameters.ParameterName* method), 172

get_datafiles() (*tardis.tardis_portal.api.DatasetResource* method), 221

get_datafiles() (*tardis.tardis_portal.models.dataset.Dataset* method), 148

get_datafiles() (*tardis.tardis_portal.models.experiment.Experiment* method), 152

get_dataset_info() (in module *tardis.tardis_portal.views.utils*), 212

get_default_push_location() (in module *tardis.apps.push_to.utils*), 97

get_default_storage() (*tardis.tardis_portal.models.storage.StorageBox* class method), 178

get_default_storage_box() (*tardis.tardis_portal.models.datafile.DataFile* method), 143

get_dir_nodes() (*tardis.tardis_portal.models.dataset.Dataset* method), 148

get_dir_tuples() (*tardis.tardis_portal.models.dataset.Dataset* method), 149

get_dn_suffix() (*tardis.tardis_portal.tests.slapped.Slapped* method), 197

get_download_organizations() (in module *tardis.tardis_portal.download*), 229

get_download_url() (*tardis.tardis_portal.models.datafile.DataFile* method), 143

get_download_urls() (*tardis.tardis_portal.models.dataset.Dataset* method), 149

get_download_urls() (*tardis.tardis_portal.models.experiment.Experiment* method), 152

get_edit_url() (*tardis.tardis_portal.models.dataset.Dataset* method), 149

get_edit_url() (*tardis.tardis_portal.models.experiment.Experiment* method), 152

get_effective_query() (*tardis.tardis_portal.models.access_control.ObjectACL* class method), 138

get_experiment_list() (in module *tardis.tardis_portal.views.ajax_json*), 206

get_experiment_referer() (in module *tardis.tardis_portal.shortcuts*), 239

```

get_file() (tardis.apps.filepicker.utils.FilepickerFile
method), 77
get_file() (tardis.tardis_portal.models.datafile.DataFile
method), 143
get_filesystem_safe_dataset_name() (in
module tardis.tardis_portal.util), 240
get_filesystem_safe_experiment_name()
(in module tardis.tardis_portal.util), 240
get_first_experiment()
(tardis.tardis_portal.models.dataset.Dataset
method), 149
get_full_path() (tardis.tardis_portal.models.datafile.DataFileObject
method), 145
get_git_version() (in module
tardis.default_settings), 124
get_groups() (tardis.tardis_portal.models.experiment.Experiment
method), 152
get_id() (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider
static method), 79
get_identifier() (tardis.tardis_portal.api.MyTardisAuthentication
method), 224
get_image_data() (tardis.tardis_portal.models.datafile.DataFile
method), 143
get_images() (tardis.tardis_portal.models.dataset.Dataset
method), 149
get_images() (tardis.tardis_portal.models.experiment.Experiment
method), 152
get_initialised_storage_instance()
(tardis.tardis_portal.models.storage.StorageBox
method), 178
get_instances_from_related()
(tardis.apps.search.documents.DataFileDocument
method), 106
get_instances_from_related()
(tardis.apps.search.documents.DatasetDocument
method), 106
get_instances_from_related()
(tardis.apps.search.documents.ExperimentDocument
method), 106
get_item() (in module
tardis.tardis_portal.templatetags.lookupfilters),
186
get_leaf() (tardis.apps.sftp.sftp.DynamicTree
method), 112
get_license_uri()
(tardis.tardis_portal.tests.test_publishservice.MockRefCProvider
method), 204
get_local_time() (in module
tardis.tardis_portal.util), 240
get_matching_auth_provider() (in module
tardis.apps.openid_migration.migration), 86
get_matching_authmethod() (in module
tardis.tardis_portal.templatetags.approved_user_tag),
184
get_mimetype() (tardis.tardis_portal.models.datafile.DataFile
method), 143
get_next_by_added()
(tardis.apps.sftp.models.SFTPPublicKey
method), 111
get_next_by_created_time()
(tardis.tardis_portal.models.datafile.DataFileObject
method), 145
get_next_by_created_time()
(tardis.tardis_portal.models.experiment.Experiment
method), 152
get_next_by_created_time()
(tardis.tardis_portal.models.jti.JTI
method),
158
get_next_by_expiry_date()
(tardis.tardis_portal.models.token.Token
method), 182
get_next_by_registration_timestamp()
(tardis.apps.openid_migration.models.OpenidUserMigration
method), 88
get_next_by_update_time()
(tardis.tardis_portal.models.experiment.Experiment
method), 152
get_none_option_license()
(tardis.tardis_portal.models.license.License
class method), 159
get_oauth_service()
(tardis.apps.push_to.models.OAuthSSHCertSigningService
static method), 94
get_object_list()
(tardis.apps.search.api.AdvanceSearchAppResource
method), 104
get_object_list()
(tardis.apps.search.api.SearchAppResource
method), 105
get_object_size() (in module
tardis.apps.push_to.utils), 97
get_options_as_dict()
(tardis.tardis_portal.models.storage.StorageBox
method), 178
get_or_create_directory()
(tardis.tardis_portal.models.experiment.Experiment
method), 153
get_or_create_user() (in module
tardis.tardis_portal.auth.utils), 130
get_or_create_user()
(tardis.tardis_portal.auth.authservice.AuthService
method), 126
get_owned_experiments() (in module
tardis.tardis_portal.auth.decorators), 127
get_owners() (tardis.tardis_portal.models.experiment.Experiment
method), 153
get_param() (tardis.tardis_portal.ParameterSetManager.ParameterSetM
method), 213

```

`get_params()` (`tardis.tardis_portal.ParameterSetManager.ParameterSetManager` method), 213
`get_path()` (`tardis.tardis_portal.models.dataset.Dataset` method), 149
`get_perm_bool()` (`tardis.tardis_portal.auth.authorisation.ACLAwareBackend` method), 125
`get_preferred_dfo()` (`tardis.tardis_portal.models.datafile.DataFile` method), 143
`get_previous_by_added()` (`tardis.apps.sftp.models.SFTPPublicKey` method), 111
`get_previous_by_created_time()` (`tardis.tardis_portal.models.datafile.DataFileObject` method), 146
`get_previous_by_created_time()` (`tardis.tardis_portal.models.experiment.Experiment` method), 153
`get_previous_by_created_time()` (`tardis.tardis_portal.models.jti.JTI` method), 158
`get_previous_by_expiry_date()` (`tardis.tardis_portal.models.token.Token` method), 182
`get_previous_by_migration_timestamp()` (`tardis.apps.openid_migration.models.OpenidUserMigration` method), 88
`get_previous_by_update_time()` (`tardis.tardis_portal.models.experiment.Experiment` method), 153
`get_public_access_display()` (`tardis.tardis_portal.models.experiment.Experiment` method), 153
`get_push_url_for_host()` (in module `tardis.apps.push_to.views`), 98
`get_queryset()` (`tardis.apps.search.documents.DataFileDocument` method), 106
`get_queryset()` (`tardis.tardis_portal.managers.OracleSafeManager` method), 238
`get_receiving_storage_box()` (`tardis.tardis_portal.models.datafile.DataFile` method), 143
`get_related_object()` (`tardis.tardis_portal.models.access_control.ObjectACL` method), 138
`get_related_object_group()` (`tardis.tardis_portal.models.access_control.ObjectACL` method), 138
`get_response()` (`tardis.tardis_portal.download.UncachedTarStream` method), 229
`get_rifcs_context()` (`tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider` method), 183
`get_rifcs_context()` (`tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider` method), 183
`get_rifcs_id()` (`tardis.apps.oaipmh.provider.experiment.RifCsExperiment` static method), 80
`get_root_dn()` (`tardis.tardis_portal.tests.slapped.Slapped` method), 197
`get_root_password()` (`tardis.tardis_portal.tests.slapped.Slapped` method), 197
`get_schema()` (`tardis.tardis_portal.ParameterSetManager.ParameterSetManager` method), 213
`get_server()` (in module `tardis.apps.oaipmh.server`), 83
`get_session_expiry()` (`tardis.tardis_portal.models.token.Token` method), 182
`get_shared_experiments()` (in module `tardis.tardis_portal.auth.decorators`), 127
`get_signing_services()` (in module `tardis.apps.push_to.views`), 98
`get_size()` (`tardis.tardis_portal.models.datafile.DataFile` method), 143
`get_size()` (`tardis.tardis_portal.models.dataset.Dataset` method), 150
`get_size()` (`tardis.tardis_portal.models.experiment.Experiment` method), 153
`get_size_and_sha512sum()` (in module `tardis.tardis_portal.tests.test_download`), 200
`get_suitable_credential()` (`tardis.apps.push_to.models.Credential` static method), 91
`get_suitable_licenses()` (`tardis.tardis_portal.models.license.License` class method), 159
`get_tardis_apps()` (in module `tardis.app_config`), 244
`get_template()` (`tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider` method), 183
`get_template()` (`tardis.tardis_portal.publish.publishservice.PublishService` method), 183
`get_template()` (`tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider` method), 204
`get_thumbnail_url()` (`tardis.tardis_portal.models.dataset.Dataset` method), 150
`get_tmpdir()` (`tardis.tardis_portal.tests.slapped.Slapped` method), 197
`get_token_data()` (in module `tardis.apps.push_to.oauth_tokens`), 96

tardis.apps.push_to.oauth_tokens), 96
get_type_display() (*tardis.tardis_portal.models.parameters.Schema* method), 175
get_url() (*tardis.tardis_portal.tests.slapi.Slapd* method), 197
get_user() (*tardis.tardis_portal.auth.interfaces.AuthProvider* method), 128
get_user() (*tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend* method), 129
get_utc_time() (in module *tardis.tardis_portal.util*), 240
get_value_type_display() (*tardis.tardis_portal.models.storage.StorageBoxOptions* method), 181
get_view_url() (*tardis.tardis_portal.models.datafile.DataFile* method), 143
getAuthMethodChoices() (in module *tardis.tardis_portal.forms*), 236
getAuthMethodDescription() (*tardis.tardis_portal.models.access_control.UserAuthentication* method), 139
getGroupById() (*tardis.tardis_portal.auth.interfaces.GroupProvider* method), 128
getGroupById() (*tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider* method), 129
getGroupById() (*tardis.tardis_portal.tests.test_authservice.MockGroupProvider* method), 199
getGroups() (in module *tardis.tardis_portal.auth.fix_circular*), 128
getGroups() (*tardis.tardis_portal.auth.authservice.AuthService* method), 126
getGroups() (*tardis.tardis_portal.auth.interfaces.GroupProvider* method), 128
getGroups() (*tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider* method), 129
getGroups() (*tardis.tardis_portal.auth.token_auth.TokenGroupProvider* method), 130
getGroups() (*tardis.tardis_portal.tests.test_authservice.MockGroupProvider* method), 199
getGroupsForEntity() (*tardis.tardis_portal.auth.authservice.AuthService* method), 126
getGroupsForEntity() (*tardis.tardis_portal.auth.interfaces.GroupProvider* method), 128
getGroupsForEntity() (*tardis.tardis_portal.tests.test_authservice.MockGroupProvider* method), 199
getParameterSets() (*tardis.tardis_portal.models.datafile.DataFile* method), 143
getParameterSets() (*tardis.tardis_portal.models.dataset.Dataset* method), 148
getParameterSets() (*tardis.tardis_portal.models.experiment.Experiment* method), 152
getParameterSets() (*tardis.tardis_portal.models.instrument.Instrument* method), 157
getRecord() (*tardis.apps.oaipmh.provider.base.BaseProvider* method), 77, 247
getRecord() (*tardis.apps.oaipmh.provider.experiment.AbstractExperiment* method), 79
getRecord() (*tardis.apps.oaipmh.server.ProxyingServer* method), 82, 249
getSupportedAuthMethods() (in module *tardis.apps.openid_migration.migration*), 86
GetTestCase (class in *tardis.apps.related_info.tests.tests*), 99
getUniqueShortName() (*tardis.tardis_portal.models.parameters.ParameterName* method), 172
getUser() (*tardis.tardis_portal.auth.authservice.AuthService* method), 126
getUserAuthentications() (*tardis.tardis_portal.models.access_control.UserProfile* method), 140
getUserById() (*tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider* method), 128
getUserByEmail() (*tardis.tardis_portal.auth.authservice.AuthService* method), 126
getUserByEmail() (*tardis.tardis_portal.auth.interfaces.AuthProvider* method), 128
getUserByEmail() (*tardis.tardis_portal.auth.interfaces.UserProvider* method), 128
global_contexts() (in module *tardis.tardis_portal.context_processors*), 228
google_analytics() (in module *tardis.tardis_portal.context_processors*), 228
GREATER_THAN_COMPARISON (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 171
GREATER_THAN_EQUAL_COMPARISON (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 171
group (*tardis.tardis_portal.models.access_control.GroupAdmin* attribute), 136
group_acls_system_owned()

(tardis.tardis_portal.managers.ExperimentManager
 method), 237
 group_acls_user_owned() (in module
 tardis.tardis_portal.managers.ExperimentManager
 method), 237
 group_id (tardis.tardis_portal.models.access_control.GroupAdmin
 attribute), 136
 group_ownership_required() (in module
 tardis.tardis_portal.auth.decorators), 127
 GroupAdmin (class in tardis.tardis_portal.models.access_control),
 136
 GroupAdmin.DoesNotExist, 136
 GroupAdmin.MultipleObjectsReturned, 136
 GroupProvider (class in tardis.tardis_portal.auth.interfaces), 128
 GroupResource (class in tardis.tardis_portal.api), 223
 GroupResource.Meta (class in tardis.tardis_portal.api), 223
 GroupResourceTest (class in tardis.tardis_portal.tests.api.test_group_resource),
 189
 gt() (in module tardis.tardis_portal.template_tags.basiccomparisonfilters), 82
 gte() (in module tardis.tardis_portal.template_tags.basiccomparisonfilters), 82
H
 handle (tardis.tardis_portal.models.experiment.Experiment
 attribute), 153
 handle() (tardis.apps.sftp.management.commands.sftpd.Command
 attribute), 131
 handle() (tardis.apps.sftp.sftp.MyTSFTPRequestHandler
 method), 107
 handle() (tardis.apps.sftp.sftp.MyTSFTPRequestHandler
 method), 112
 handle() (tardis.tardis_portal.management.commands.createuser.Command
 attribute), 132
 handle() (tardis.tardis_portal.management.commands.createuser.Command
 method), 131
 handle() (tardis.tardis_portal.management.commands.dumpschemas.Command
 attribute), 131
 handle() (tardis.tardis_portal.management.commands.dumpschemas.Command
 method), 131
 handle() (tardis.tardis_portal.management.commands.loadschemas.Command
 attribute), 131
 handle() (tardis.tardis_portal.management.commands.loadschemas.Command
 method), 131
 handle() (tardis.tardis_portal.management.commands.rmexperiment.Command
 attribute), 132
 handle() (tardis.tardis_portal.management.commands.rmexperiment.Command
 method), 132
 handle_timeout() (tardis.apps.sftp.sftp.MyTSFTPRequestHandler
 method), 112
 has_datafile_access() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_datafile_download_access() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_dataset_access() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_dataset_download_access() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_dataset_ownership() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_dataset_write() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_delete_permissions() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_experiment_access() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_experiment_download_access() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_experiment_ownership() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_experiment_write() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_image() (tardis.tardis_portal.models.datafile.DataFile
 method), 143
 has_perm() (tardis.tardis_portal.auth.authorisation.ACLAwareBackend
 method), 125
 has_read_or_owner_ACL() (in module
 tardis.tardis_portal.auth.decorators), 127
 has_write_permissions() (in module
 tardis.tardis_portal.auth.decorators), 128
 hasReader() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry
 method), 82
 hasWriter() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry
 method), 82
 healthz() (in module
 tardis.tardis_portal.views.pages), 211
 help (tardis.tardis_portal.management.commands.createuser.Command
 attribute), 131
 help (tardis.tardis_portal.management.commands.dumpschemas.Command
 attribute), 131
 help (tardis.tardis_portal.management.commands.loadschemas.Command
 attribute), 132
 help (tardis.tardis_portal.management.commands.rmexperiment.Command
 attribute), 132
 hidden (tardis.tardis_portal.models.parameters.Schema
 attribute), 175
 host_name (tardis.apps.push_to.models.RemoteHost
 attribute), 175
 HttpResponseRedirectNotAllowed (class in
 tardis.tardis_portal.views.utils), 212
 HttpResponseRedirect (class in
 tardis.tardis_portal.views.utils), 212
 hydrate() (tardis.apps.sftp.api.SFTPPublicKeyAppResource
 method), 109
 hydrate() (tardis.tardis_portal.api.DataFileResource
 method), 220
 hydrate() (tardis.tardis_portal.api.ObjectACLResource
 method), 225
 hydrate() (tardis.tardis_portal.api.ParameterResource
 method), 225
 hydrate() (tardis.tardis_portal.api.ReplicaResource
 method), 226

hydrate_m2m() (*tardis.tardis_portal.api.DatasetResource* attribute), 221

hydrate_m2m() (*tardis.tardis_portal.api.ExperimentResource* attribute), 223

hydrate_schema() (*tardis.tardis_portal.api.ParameterSetNameResource* attribute), 226

I

id (*tardis.apps.openid_migration.models.OpenidACLMigration* attribute), 87

id (*tardis.apps.openid_migration.models.OpenidUserMigration* attribute), 88

id (*tardis.apps.push_to.models.Credential* attribute), 91

id (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 94

id (*tardis.apps.push_to.models.RemoteHost* attribute), 95

id (*tardis.apps.sftp.models.SFTPPublicKey* attribute), 111

id (*tardis.tardis_portal.models.access_control.GroupAdmin* attribute), 137

id (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 138

id (*tardis.tardis_portal.models.access_control.UserAuthentication* attribute), 139

id (*tardis.tardis_portal.models.access_control.UserProfile* attribute), 140

id (*tardis.tardis_portal.models.datafile.DataFile* attribute), 143

id (*tardis.tardis_portal.models.datafile.DataFileObject* attribute), 146

id (*tardis.tardis_portal.models.dataset.Dataset* attribute), 150

id (*tardis.tardis_portal.models.experiment.Experiment* attribute), 153

id (*tardis.tardis_portal.models.experiment.ExperimentAuthor* attribute), 155

id (*tardis.tardis_portal.models.facility.Facility* attribute), 156

id (*tardis.tardis_portal.models.instrument.Instrument* attribute), 157

id (*tardis.tardis_portal.models.jti.JTI* attribute), 158

id (*tardis.tardis_portal.models.license.License* attribute), 159

id (*tardis.tardis_portal.models.parameters.DatafileParameterSet* attribute), 160

id (*tardis.tardis_portal.models.parameters.DatafileParameterSet* attribute), 161

id (*tardis.tardis_portal.models.parameters.DatasetParameterSet* attribute), 162

id (*tardis.tardis_portal.models.parameters.DatasetParameterSet* attribute), 163

id (*tardis.tardis_portal.models.parameters.ExperimentParameterSet* attribute), 164

id (*tardis.tardis_portal.models.parameters.ExperimentParameterSet* attribute), 166

id (*tardis.tardis_portal.models.parameters.FreeTextSearchField* attribute), 166

id (*tardis.tardis_portal.models.parameters.InstrumentParameterSet* attribute), 167

id (*tardis.tardis_portal.models.parameters.InstrumentParameterSet* attribute), 168

id (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 172

id (*tardis.tardis_portal.models.parameters.Schema* attribute), 175

id (*tardis.tardis_portal.models.storage.StorageBox* attribute), 178

id (*tardis.tardis_portal.models.storage.StorageBoxAttribute* attribute), 180

id (*tardis.tardis_portal.models.storage.StorageBoxOption* attribute), 181

id (*tardis.tardis_portal.models.token.Token* attribute), 182

identify() (*tardis.apps.oaipmh.provider.base.BaseProvider* method), 78, 248

identify() (*tardis.apps.oaipmh.server.ProxyingServer* method), 83, 249

image (*tardis.tardis_portal.models.dataset.Dataset* attribute), 150

image_url (*tardis.tardis_portal.models.license.License* attribute), 159

immutable (*tardis.tardis_portal.models.dataset.Dataset* attribute), 150

immutable (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 172

immutable (*tardis.tardis_portal.models.parameters.Schema* attribute), 175

ImportParamsForm (class in *tardis.tardis_portal.forms*), 232

INDEX_VIEWS (in module *tardis.default_settings.custom_views*), 120

IndexExperimentTestCase (class in *tardis.apps.search.tests.test_index*), 103

IndexView (class in *tardis.tardis_portal.views.pages*), 210

initial (*tardis.apps.openid_migration.migrations.0001_initial.Migration* attribute), 84

initial (*tardis.apps.sftp.migrations.0001_initial.Migration* attribute), 108

inlines (*tardis.tardis_portal.admin.DatafileAdmin* attribute), 214

inlines (*tardis.tardis_portal.admin.ExperimentAdmin* attribute), 214

inlines (*tardis.tardis_portal.admin.ExperimentParameterSetAdmin* attribute), 215

inlines (*tardis.tardis_portal.admin.InstrumentParameterSetAdmin* attribute), 215

`inlines (tardis.tardis_portal.admin.SchemaAdmin attribute), 216`
`inlines (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 216`
`institution (tardis.tardis_portal.models.experiment.ExperimentAdmin attribute), 155`
`institution_name (tardis.tardis_portal.models.experiment.ExperimentAdmin attribute), 153`
`Instrument (class in tardis.tardis_portal.models.instrument), 157`
`instrument (tardis.tardis_portal.models.dataset.Dataset attribute), 150`
`instrument (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 168`
`INSTRUMENT (tardis.tardis_portal.models.parameters.Schema attribute), 174`
`Instrument.DoesNotExist, 157`
`Instrument.MultipleObjectsReturned, 157`
`instrument_id (tardis.tardis_portal.models.dataset.Dataset attribute), 150`
`instrument_id (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 168`
`instrument_set (tardis.tardis_portal.models.facility.Facility attribute), 156`
`InstrumentAdmin (class in tardis.tardis_portal.admin), 215`
`InstrumentParameter (class in tardis.tardis_portal.models.parameters), 167`
`InstrumentParameter.DoesNotExist, 167`
`InstrumentParameter.MultipleObjectsReturned, 167`
`instrumentparameter_set (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 168`
`instrumentparameter_set (tardis.tardis_portal.models.parameters.ParameterSetName attribute), 172`
`InstrumentParameterInline (class in tardis.tardis_portal.admin), 215`
`InstrumentParameterSet (class in tardis.tardis_portal.models.parameters), 168`
`InstrumentParameterSet.DoesNotExist, 168`
`InstrumentParameterSet.MultipleObjectsReturned, 168`
`instrumentparameterset_set (tardis.tardis_portal.models.instrument.Instrument attribute), 158`
`instrumentparameterset_set (tardis.tardis_portal.models.parameters.Schema attribute), 175`
`InstrumentParameterSetAdmin (class in tardis.tardis_portal.admin), 215`
`instrumentparametersets (tardis.tardis_portal.models.storage.StorageBox attribute), 178`
`InstrumentResource (class in tardis.tardis_portal.api), 224`
`InstrumentResource.Meta (class in tardis.tardis_portal.api), 224`
`InstrumentResourceTest (class in tardis.tardis_portal.tests.api.test_instrument_resource), 189`
`InstrumentTestCase (class in tardis.tardis_portal.tests.models.test_instrument), 189`
`internal_description (tardis.tardis_portal.models.license.License attribute), 159`
`INTERNAL_IPS (in module tardis.default_settings.debug), 120`
`interpolate_template () (in module tardis.apps.openid_migration.email_text), 80`
`is_active (tardis.tardis_portal.models.license.License attribute), 159`
`is_authenticated () (tardis.tardis_portal.api.MyTardisAuthentication method), 224`
`is_directory () (in module tardis.apps.push_to.utils), 97`
`is_expired () (tardis.tardis_portal.models.token.Token method), 182`
`is_facility_manager () (in module tardis.tardis_portal.models.facility), 156`
`is_group_admin () (in module tardis.tardis_portal.auth.decorators), 128`
`is_image () (tardis.tardis_portal.models.datafile.DataFile method), 143`
`is_in_name () (tardis.tardis_portal.models.parameters.ParameterName method), 173`
`is_local () (tardis.tardis_portal.models.datafile.DataFile method), 143`
`is_online (tardis.tardis_portal.models.datafile.DataFile attribute), 144`
`is_online (tardis.tardis_portal.models.dataset.Dataset attribute), 150`
`is_openid_migration_enabled () (in module tardis.apps.social_auth.auth.social_auth), 118`
`is_public () (tardis.tardis_portal.models.datafile.DataFile method), 144`
`is_publication () (tardis.tardis_portal.models.experiment.Experiment method), 153`
`is_publication_draft () (tardis.tardis_portal.models.experiment.Experiment method), 153`
`is_schema_valid ()`

(tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider (class in tardis.tardis_portal.models.storage.StorageBoxOption attribute), 183
 is_schema_valid() (tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider (class in tardis.tardis_portal.models.storage.StorageBoxOption attribute), 93
 is_searchable (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 111
 is_tardis_app() (in module tardis.app_config), 244
 is_valid() (tardis.tardis_portal.forms.ExperimentForm (class in tardis.tardis_portal.forms), 110
 is_valid_email() (in module tardis.tardis_portal.management.commands.createuser), 93
 isDateTime() (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 172
 isDjangoAccount (tardis.tardis_portal.models.access_control.UserProfile (class in tardis.tardis_portal.models.access_control), 140
 isFilename() (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 146
 isLink() (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 173
 isLongString() (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 173
 isNumeric() (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 173
 isOwner (tardis.tardis_portal.models.access_control.ObjectACL (class in tardis.tardis_portal.models.access_control), 139
 isString() (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 173
 isURL() (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 173
 isValidPublicContact() (tardis.tardis_portal.models.access_control.UserProfile (class in tardis.tardis_portal.models.access_control), 140
 IteratorTracker (class in tardis.analytics.tracker), 75
J
 JSON (tardis.tardis_portal.models.parameters.ParameterName (class in tardis.tardis_portal.models.parameters.parameters), 171
 json_indent (tardis.apps.search.api.PrettyJSONSerializer (class in tardis.apps.search.api), 104
 json_indent (tardis.tardis_portal.api.PrettyJSONSerializer (class in tardis.tardis_portal.api), 226
 JTI (class in tardis.tardis_portal.models.jti), 158
 jti (tardis.tardis_portal.models.jti.JTI (class in tardis.tardis_portal.models.jti), 158
 JTI.DoesNotExist, 158
 JTI.MultipleObjectsReturned, 158
K
 key (tardis.apps.push_to.models.KeyPair (class in tardis.apps.push_to.models), 93
 key (tardis.tardis_portal.models.storage.StorageBoxAttribute (class in tardis.tardis_portal.models.storage), 180
 key_type (tardis.apps.push_to.models.KeyPair (class in tardis.apps.push_to.models), 93
 key_type (tardis.apps.sftp.models.SFTPPublicKey (class in tardis.apps.sftp.models), 111
 KeyAddForm (class in tardis.apps.sftp.forms), 110
 KeyGenerateForm (class in tardis.apps.sftp.forms), 110
 KeyPair (class in tardis.apps.push_to.models), 93
 KeyPair.Meta (class in tardis.apps.push_to.models), 93
 Label (class in tardis.tardis_portal.widgets), 241
 LDAP_BASE (tardis.default_settings (class in tardis.default_settings), 54
 LDAP_GROUP_ATTR_MAP (tardis.default_settings (class in tardis.default_settings), 53
 LDAP_USER_BASE (tardis.default_settings (class in tardis.default_settings), 54
 LDAP_USER_ID_ATTR (tardis.default_settings (class in tardis.default_settings), 53
 LDAP_TLS (tardis.default_settings (class in tardis.default_settings), 53
 LDAP_URL (tardis.default_settings (class in tardis.default_settings), 53
 LDAP_USER_LOGIN_ATTR (tardis.default_settings (class in tardis.default_settings), 53
 ldapadd() (tardis.tardis_portal.tests.slapped.Slapped (class in tardis.tardis_portal.tests.slapped), 197
 LDAPErrorTest (class in tardis.tardis_portal.tests.test_ldap), 202
 ldapsearch() (tardis.tardis_portal.tests.slapped.Slapped (class in tardis.tardis_portal.tests.slapped), 197
 LDAPTest (class in tardis.tardis_portal.tests.test_ldap), 202
 length_gt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 184
 length_gte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 184
 length_lt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 184
 length_lte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 184
 LESS_THAN_COMPARISON

(*tardis.tardis_portal.models.parameters.ParameterName* attribute), 171

LESS_THAN_EQUAL_COMPARISON (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 171

Level0TestCase (class in *tardis.tardis_portal.tests.test_iiif*), 201

Level1TestCase (class in *tardis.tardis_portal.tests.test_iiif*), 201

Level2TestCase (class in *tardis.tardis_portal.tests.test_iiif*), 202

License (class in *tardis.tardis_portal.models.license*), 159

license (*tardis.tardis_portal.models.experiment.Experiment* attribute), 153

License.DoesNotExist, 159

License.MultipleObjectsReturned, 159

license_id (*tardis.tardis_portal.models.experiment.Experiment* attribute), 153

LINK (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 171

link_ct (*tardis.tardis_portal.models.parameters.DatafileParameter* attribute), 160

link_ct (*tardis.tardis_portal.models.parameters.DatasetParameter* attribute), 162

link_ct (*tardis.tardis_portal.models.parameters.ExperimentParameter* attribute), 164

link_ct (*tardis.tardis_portal.models.parameters.InstrumentParameter* attribute), 167

link_ct (*tardis.tardis_portal.models.parameters.Parameter* attribute), 169

link_ct_id (*tardis.tardis_portal.models.parameters.Parameter* attribute), 169

link_gfk (*tardis.tardis_portal.models.parameters.DatafileParameter* attribute), 160

link_gfk (*tardis.tardis_portal.models.parameters.DatasetParameter* attribute), 162

link_gfk (*tardis.tardis_portal.models.parameters.ExperimentParameter* attribute), 164

link_gfk (*tardis.tardis_portal.models.parameters.InstrumentParameter* attribute), 167

link_gfk (*tardis.tardis_portal.models.parameters.Parameter* attribute), 170

link_id (*tardis.tardis_portal.models.parameters.Parameter* attribute), 170

link_url (*tardis.tardis_portal.models.parameters.Parameter* attribute), 170

list_allowed_methods (*tardis.apps.search.api.AdvanceSearchAppResource.Meta* attribute), 104

list_allowed_methods (*tardis.apps.search.api.SearchAppResource.Meta* attribute), 104

list_allowed_methods (*tardis.apps.search.api.SearchAppResource.Meta* attribute), 104

list_auth_methods () (in *tardis.tardis_portal.auth.authentication*), 124

list_display (*tardis.apps.openid_migration.models.OpenidACLMigration* attribute), 87

list_display (*tardis.apps.openid_migration.models.OpenidUserMigration* attribute), 89

list_display (*tardis.tardis_portal.admin.ObjectACLAdmin* attribute), 215

list_folder () (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* method), 113

list_subdirectories () (in *tardis.apps.push_to.utils*), 97

ListEndpointsTest (class in *tardis.tardis_portal.tests.api.test_list_api_endpoints*), 189

listIdentifiers () (*tardis.apps.oaipmh.provider.base.BaseProvider* method), 78, 248

listIdentifiers () (*tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider* method), 79

listIdentifiers () (*tardis.apps.oaipmh.server.ProxyingServer* method), 83, 249

listMetadataFormats () (*tardis.apps.oaipmh.provider.base.BaseProvider* method), 78, 248

listMetadataFormats () (*tardis.apps.oaipmh.provider.experiment.DcExperimentProvider* method), 79

listMetadataFormats () (*tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider* method), 80

listMetadataFormats () (*tardis.apps.oaipmh.server.ProxyingServer* method), 83, 250

listParameters () (*tardis.apps.oaipmh.provider.base.BaseProvider* method), 78, 248

listRecords () (*tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider* method), 79

listRecords () (*tardis.apps.oaipmh.server.ProxyingServer* method), 83, 250

listSets () (*tardis.apps.oaipmh.provider.base.BaseProvider* method), 79, 249

listSets () (*tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider* method), 79

listSets () (*tardis.apps.oaipmh.server.ProxyingServer* method), 83, 250

ListTestCase (class in *tardis.apps.related_info.tests.tests*), 99

load_datafile_image () (in *tardis.tardis_portal.models.parameters*), 171

tardis.tardis_portal.views.images), 208
 load_dataset_image() (in module *tardis.tardis_portal.views.images*), 208
 load_experiment_image() (in module *tardis.tardis_portal.views.images*), 208
 load_image() (in module *tardis.tardis_portal.views.images*), 208
 load_licenses() (in module *tardis.tardis_portal.migrations.0017_add_cc_licenses*), 136
 LoadSchemasTestCase (class in *tardis.tardis_portal.tests.management.test_loadschemas*), 192
 LocationResource (class in *tardis.tardis_portal.api*), 224
 LocationResource.Meta (class in *tardis.tardis_portal.api*), 224
 locked (*tardis.tardis_portal.models.experiment.Experiment* attribute), 153
 logger (in module *tardis.tardis_portal.xmlwriter*), 242
 LoggingMiddleware (class in *tardis.tardis_portal.logging_middleware*), 236
 login() (in module *tardis.tardis_portal.views.authentication*), 207
 LOGIN_VIEWS (in module *tardis.default_settings.custom_views*), 120
 LoginForm (class in *tardis.tardis_portal.forms*), 233
 logo_img (*tardis.apps.push_to.models.RemoteHost* attribute), 95
 logout() (in module *tardis.tardis_portal.views.authentication*), 207
 LONGSTRING (*tardis.tardis_portal.models.parameters.Parameter* attribute), 171
 lstat() (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* method), 113
 lt() (in module *tardis.tardis_portal.templatetags.basiccomparisonfilters*), 184
 lte() (in module *tardis.tardis_portal.templatetags.basiccomparisonfilters*), 184
M
 make_dirs() (in module *tardis.apps.push_to.tasks*), 97
 make_mapper() (in module *tardis.tardis_portal.download*), 229
 make_tar() (*tardis.tardis_portal.download.UncachedTarStream* method), 229
 manage_auth_methods() (in module *tardis.tardis_portal.views.authentication*), 207
 manage_groups() (in module *tardis.tardis_portal.views.authentication*), 207
 manage_rifcs() (*tardis.tardis_portal.publish.publishservice.PublishService* method), 183
 manage_user_account() (in module *tardis.tardis_portal.views.authentication*), 207
 ManageAccountForm (class in *tardis.tardis_portal.forms*), 233
 ManageAccountForm.Meta (class in *tardis.tardis_portal.forms*), 233
 ManageAccountTestCase (class in *tardis.tardis_portal.tests.views.test_auth_views*), 194
 ManageGroupPermissionsForm (class in *tardis.tardis_portal.forms*), 233
 manager_group (*tardis.tardis_portal.models.facility.Facility* attribute), 156
 manager_group_id (*tardis.tardis_portal.models.facility.Facility* attribute), 156
 master_box (*tardis.tardis_portal.models.storage.StorageBox* attribute), 179
 master_box_id (*tardis.tardis_portal.models.storage.StorageBox* attribute), 179
 MAX_IMAGES_IN_CAROUSEL (in module *tardis.default_settings.frontend*), 122
 MAX_SEARCH_RESULTS (in module *tardis.default_settings.search*), 122
 max_size (*tardis.tardis_portal.models.storage.StorageBox* attribute), 179
 md5sum (*tardis.tardis_portal.models.datafile.DataFile* attribute), 144
 media (*tardis.apps.openid_migration.models.OpenidACLMigrationAdmin* attribute), 87
 media (*tardis.apps.openid_migration.models.OpenidUserMigrationAdmin* attribute), 89
 media (*tardis.apps.push_to.models.CredentialAdmin* attribute), 92
 media (*tardis.apps.push_to.models.CredentialForm* attribute), 92
 media (*tardis.apps.push_to.models.RemoteHostAdmin* attribute), 96
 media (*tardis.apps.related_info.forms.RelatedInfoForm* attribute), 100
 media (*tardis.apps.sftp.forms.KeyAddForm* attribute), 110
 media (*tardis.apps.sftp.forms.KeyGenerateForm* attribute), 110
 media (*tardis.tardis_portal.admin.DatafileAdmin* attribute), 214
 media (*tardis.tardis_portal.admin.DatafileAdminForm* attribute), 214
 media (*tardis.tardis_portal.admin.DataFileObjectInline* attribute), 213

media (tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 214	media (tardis.tardis_portal.forms.ImportParamsForm attribute), 233
media (tardis.tardis_portal.admin.DatasetAdmin attribute), 214	media (tardis.tardis_portal.forms.LoginForm attribute), 233
media (tardis.tardis_portal.admin.ExperimentAdmin attribute), 214	media (tardis.tardis_portal.forms.ManageAccountForm attribute), 233
media (tardis.tardis_portal.admin.ExperimentParameterInlineForm attribute), 215	media (tardis.tardis_portal.forms.ManageGroupPermissionsForm attribute), 234
media (tardis.tardis_portal.admin.ExperimentParameterSetAdmin attribute), 215	media (tardis.tardis_portal.forms.NoInput attribute), 234
media (tardis.tardis_portal.admin.FacilityAdmin attribute), 215	media (tardis.tardis_portal.forms.RegisterExperimentForm attribute), 234
media (tardis.tardis_portal.admin.FreeTextSearchFieldAdmin attribute), 215	media (tardis.tardis_portal.forms.RegistrationForm attribute), 235
media (tardis.tardis_portal.admin.InstrumentAdmin attribute), 215	media (tardis.tardis_portal.forms.RightsForm attribute), 235
media (tardis.tardis_portal.admin.InstrumentParameterInlineForm attribute), 215	media (tardis.tardis_portal.widgets.CommaSeparatedInput attribute), 240
media (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 215	media (tardis.tardis_portal.widgets.Label attribute), 241
media (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 215	media (tardis.tardis_portal.widgets.Span attribute), 241
media (tardis.tardis_portal.admin.ObjectACLInline attribute), 216	merge_auth_method() (in module tardis.tardis_portal.auth.authentication), 124
media (tardis.tardis_portal.admin.ParameterNameAdmin attribute), 216	METADATA_STORE_PATH (in module tardis.default_settings.storage), 123
media (tardis.tardis_portal.admin.ParameterNameInline attribute), 216	migrate_accounts() (in module tardis.apps.openid_migration.views), 89
media (tardis.tardis_portal.admin.SchemaAdmin attribute), 216	migrate_api_key() (in module tardis.apps.openid_migration.migration), 86
media (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 216	migrate_user_message() (in module tardis.apps.social_auth.auth.social_auth), 118
media (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 216	migrate_user_permissions() (in module tardis.apps.openid_migration.migration), 86
media (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 217	Migration (class in tardis.apps.openid_migration.migrations.0001_initial), 84
media (tardis.tardis_portal.admin.StorageBoxForm attribute), 217	Migration (class in tardis.apps.push_to.migrations.0001_initial), 90
media (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 218	Migration (class in tardis.apps.push_to.migrations.0002_auto_20160518_1953), 90
media (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 218	Migration (class in tardis.apps.sftp.migrations.0001_initial), 108
media (tardis.tardis_portal.admin.UserAuthenticationAdmin attribute), 218	Migration (class in tardis.tardis_portal.migrations.0001_initial), 132
media (tardis.tardis_portal.forms.AddUserPermissionsForm attribute), 230	Migration (class in tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160518_1953), 132
media (tardis.tardis_portal.forms.CreateUserPermissionsForm attribute), 231	
media (tardis.tardis_portal.forms.DatasetForm attribute), 231	
media (tardis.tardis_portal.forms.ExperimentAuthor attribute), 232	
media (tardis.tardis_portal.forms.ExperimentForm attribute), 232	

Migration (class in `tardis.apps.openid_migration.models.OpenidUserMigration` attribute), 88
 133 `mimetype` (`tardis.tardis_portal.models.datafile.DataFile` attribute), 144
 Migration (class in `tardis.tardis_portal.migrations.0002_auto_20150528_1128` attribute), 144
 133 `ONLINE_SCORE` (in `tardis.default_settings.search`), 122
 Migration (class in `missing_host_key()` attribute), 92
 133 `tardis.tardis_portal.migrations.0004_storageboxoption_value_index` (`tardis.apps.push_to.models.DBHostKeyPolicy` method), 92
 Migration (class in `makedirs()` (in module `tardis.tardis_portal.tests.slapd`),
 133 `tardis.tardis_portal.migrations.0005_datafile_add_size_int_index`),
 Migration (class in `MockAuthProvider` (class in
 133 `tardis.tardis_portal.migrations.0006_datafile_remove_size_int_index`),
 Migration (class in `MockGroupProvider` (class in
 133 `tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index`),
 Migration (class in `MockRequest` (class in
 134 `tardis.tardis_portal.migrations.0008_string_value_partial_index_postgres`),
 Migration (class in `MockRifCsProvider` (class in
 134 `tardis.tardis_portal.migrations.0009_auto_20160128_1119`), 104
 Migration (class in `MockSettings` (class in
 134 `tardis.tardis_portal.migrations.0010_auto_20160503_1443`), 99
 Migration (class in `model` (`tardis.apps.push_to.models.CredentialForm.Meta`
 134 `tardis.tardis_portal.migrations.0011_auto_20160503_1443` attribute), 92
 Migration (class in `model` (`tardis.apps.search.documents.DataFileDocument.Django`
 134 `tardis.tardis_portal.migrations.0012_userauthentication_app_provider` attribute), 105
 Migration (class in `model` (`tardis.apps.search.documents.DatasetDocument.Django`
 135 `tardis.tardis_portal.migrations.0012_userauthentication_app_provider` attribute), 106
 Migration (class in `model` (`tardis.apps.search.documents.ExperimentDocument.Django`
 135 `tardis.tardis_portal.migrations.0013_auto_20181002_1136` attribute), 106
 Migration (class in `model` (`tardis.tardis_portal.admin.DatafileAdminForm.Meta`
 135 `tardis.tardis_portal.migrations.0013_auto_20181002_1136` attribute), 214
 Migration (class in `model` (`tardis.tardis_portal.admin.DataFileObjectInline`
 135 `tardis.tardis_portal.migrations.0014_auto_20181002_1154` attribute), 213
 Migration (class in `model` (`tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta`
 135 `tardis.tardis_portal.migrations.0015_dataset_created_online` attribute), 214
 Migration (class in `model` (`tardis.tardis_portal.admin.ExperimentParameterInline`
 135 `tardis.tardis_portal.migrations.0015_dataset_created_online` attribute), 215
 Migration (class in `model` (`tardis.tardis_portal.admin.InstrumentParameterInline`
 135 `tardis.tardis_portal.migrations.0016_add_timestamps` attribute), 215
 Migration (class in `model` (`tardis.tardis_portal.admin.ObjectACLInline` at-
 135 `tardis.tardis_portal.migrations.0017_add_cc_licenses` tribute), 216
 Migration (class in `model` (`tardis.tardis_portal.admin.ParameterNameInline`
 135 `tardis.tardis_portal.migrations.0017_add_cc_licenses` attribute), 216
 Migration (class in `model` (`tardis.tardis_portal.admin.StorageBoxAttributeInline`
 136 `tardis.tardis_portal.migrations.0018_make_default_storage_box_online` attribute), 216
 136 `model` (`tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta`
 migration_status (`tardis.apps.openid_migration.models.OpenidUserMigration`
 attribute), 88
 migration_timestamp (`model` (`tardis.tardis_portal.admin.StorageBoxForm.Meta`
 attribute), 217

`model (tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute), 218`
`model (tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute), 218`
`model (tardis.tardis_portal.forms.DatasetForm.Meta attribute), 231`
`model (tardis.tardis_portal.forms.ExperimentAuthor.Meta attribute), 231`
`model (tardis.tardis_portal.forms.ExperimentForm.Meta attribute), 232`
`model (tardis.tardis_portal.forms.ManageAccountForm.Meta attribute), 233`
`model (tardis.tardis_portal.forms.RightsForm.Meta attribute), 235`
`ModelsTestCase (class in tardis.apps.push_to.tests.test_models), 90`
`ModelTestCase (class in tardis.apps.openid_migration.tests.test_models), 85`
`ModelTestCase (class in tardis.tardis_portal.tests.models), 194`
`ModelTestCase (class in tardis.tardis_portal.tests.test_storage), 204`
`modification_time (tardis.tardis_portal.models.datafile.DataFile attribute), 144`
`modified_time (tardis.tardis_portal.models.datafile.DataFileObject attribute), 146`
`modified_time (tardis.tardis_portal.models.dataset.Dataset attribute), 150`
`modified_time (tardis.tardis_portal.models.facility.Facility attribute), 156`
`modified_time (tardis.tardis_portal.models.instrument.Instrument attribute), 158`
`move_file () (tardis.tardis_portal.models.datafile.DataFileObject method), 146`
`move_files () (tardis.tardis_portal.models.storage.StorageBox method), 179`
`move_to_master () (tardis.tardis_portal.models.storage.StorageBox method), 179`
`MultiValueCommaSeparatedField (class in tardis.tardis_portal.fields), 230`
`my_data () (in module tardis.tardis_portal.views.pages), 211`
`myt_auth () (tardis.apps.sftp.sftp.MyTServerInterface method), 116`
`MyTardisAuthentication (class in tardis.tardis_portal.api), 224`
`MyTardisAuthenticationTest (class in tardis.tardis_portal.tests.api.test_auth), 186`
`MyTardisLocalFileSystemStorage (class in tardis.tardis_portal.storage.file_system), 183`
`MyTardisModelResource (class in tardis.tardis_portal.api), 224`
`MyTardisModelResource.Meta (class in tardis.tardis_portal.api), 224`
`MyTardisModelResource.TestCase (class in tardis.tardis_portal.tests.api), 191`
`MyTServerInterface (class in tardis.apps.sftp.sftp), 114`
`MyTSFTPHandle (class in tardis.apps.sftp.sftp), 112`
`MyTSFTPRequestHandler (class in tardis.apps.sftp.sftp), 112`
`MyTSFTPServer (class in tardis.apps.sftp.sftp), 112`
`MyTSFTPServerInterface (class in tardis.apps.sftp.sftp), 112`
`MyTSFTPTCPServer (class in tardis.apps.sftp.sftp), 114`

N

`name (tardis.analytics.apps.AnalyticsConfig attribute), 75`
`name (tardis.apps.openid_migration.apps.OpenidMigrationConfig attribute), 85`
`name (tardis.apps.push_to.apps.PushToConfig attribute), 90`
`name (tardis.apps.s3utils.apps.S3UtilsConfig attribute), 102`
`name (tardis.apps.search.apps.SearchConfig attribute), 105`
`name (tardis.apps.search.documents.DataFileDocument.Index attribute), 105`
`name (tardis.apps.search.documents.DatasetDocument.Index attribute), 106`
`name (tardis.apps.search.documents.ExperimentDocument.Index attribute), 106`
`name (tardis.apps.sftp.apps.SFTPConfig attribute), 110`
`name (tardis.apps.sftp.models.SFTPPublicKey attribute), 111`
`name (tardis.apps.social_auth.apps.SocialAuthConfig attribute), 118`
`name (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider attribute), 129`
`name (tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider attribute), 130`
`name (tardis.tardis_portal.auth.token_auth.TokenGroupProvider attribute), 130`
`name (tardis.tardis_portal.models.facility.Facility attribute), 156`
`name (tardis.tardis_portal.models.instrument.Instrument attribute), 158`
`name (tardis.tardis_portal.models.license.License attribute), 159`
`name (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 160`
`name (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 162`

name (*tardis.tardis_portal.models.parameters.ExperimentParameter* attribute), 165

name (*tardis.tardis_portal.models.parameters.InstrumentParameter* attribute), 167

name (*tardis.tardis_portal.models.parameters.Parameter* attribute), 170

name (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 173

name (*tardis.tardis_portal.models.parameters.Schema* attribute), 176

name (*tardis.tardis_portal.models.storage.StorageBox* attribute), 179

name_id (*tardis.tardis_portal.models.parameters.Parameter* attribute), 170

namespace (*tardis.tardis_portal.models.parameters.Schema* attribute), 176

natural_key () (*tardis.tardis_portal.models.parameters.ParameterName* method), 173

natural_key () (*tardis.tardis_portal.models.parameters.Schema* method), 176

new_param () (*tardis.tardis_portal.ParameterSetManager.ParameterSetManager* method), 213

new_user (*tardis.apps.openid_migration.models.OpenidUserMigration* attribute), 88

new_user_auth_method (*tardis.apps.openid_migration.models.OpenidUserMigration* attribute), 88

new_user_id (*tardis.apps.openid_migration.models.OpenidUserMigration* attribute), 88

next () (*tardis.analytics.tracker.IteratorTracker* method), 75

nickname (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 94

nickname (*tardis.apps.push_to.models.RemoteHost* attribute), 95

NoInput (class in *tardis.tardis_portal.forms*), 234

NONE (*tardis.tardis_portal.models.parameters.Schema* attribute), 174

NoSuitableCredential, 91

NOT_EQUAL_COMPARISON (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 171

notify_user () (in module *tardis.apps.push_to.tasks*), 97

now () (*tardis.tardis_portal.tests.test_tokens.FrozenTime* class method), 205

NS_CC (*tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider* attribute), 79

NUMERIC (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 171

numerical_value (*tardis.tardis_portal.models.parameters.Parameter* attribute), 170

oauth_authorize_url (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 94

oauth_callback () (in module *tardis.apps.push_to.views*), 98

oauth_callback_url () (in module *tardis.apps.push_to.views*), 98

oauth_check_token_url (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 94

oauth_client_id (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 94

oauth_client_secret (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 94

oauth_token_url (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 94

OAuthSSHCertSigningService (class in *tardis.apps.push_to.models*), 93

OAuthSSHCertSigningService.DoesNotExist, 93

OAuthSSHCertSigningService.MultipleObjectsReturned, 93

oauthsshcertsigningservice_set (*tardis.apps.push_to.models.RemoteHost* attribute), 95

obj_create () (*tardis.apps.search.api.AdvanceSearchAppResource* method), 104

obj_create () (*tardis.tardis_portal.api.DataFileResource* method), 220

obj_create () (*tardis.tardis_portal.api.ExperimentResource* method), 223

obj_get_list () (*tardis.apps.search.api.AdvanceSearchAppResource* method), 104

obj_get_list () (*tardis.apps.search.api.SearchAppResource* method), 105

object_class (*tardis.apps.search.api.AdvanceSearchAppResource.Meta* attribute), 104

object_class (*tardis.apps.search.api.SearchAppResource.Meta* attribute), 104

object_class (*tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta* attribute), 109

object_class (*tardis.tardis_portal.api.DatafileParameterResource.Meta* attribute), 220

object_class (*tardis.tardis_portal.api.DatafileParameterSetResource.Meta* attribute), 220

object_class (*tardis.tardis_portal.api.DataFileResource.Meta* attribute), 219

object_class (*tardis.tardis_portal.api.DatasetParameterResource.Meta* attribute), 221

object_class (*tardis.tardis_portal.api.DatasetParameterSetResource.Meta* attribute), 221

object_class (*tardis.tardis_portal.api.DatasetResource.Meta* attribute), 221

[attribute](#)), 179
[objects \(tardis.tardis_portal.models.storage.StorageBoxAttribute](#)
[attribute](#)), 180
[objects \(tardis.tardis_portal.models.storage.StorageBoxAttribute](#)
[attribute](#)), 181
[objects \(tardis.tardis_portal.models.token.TokenAttribute](#)
[attribute](#)), 182
[offline_types \(tardis.tardis_portal.models.storage.StorageBoxAttribute](#)
[attribute](#)), 179
[old_user \(tardis.apps.openid_migration.models.OpenidUserMigration](#)
[attribute](#)), 88
[old_user_auth_method \(tardis.apps.openid_migration.models.OpenidUserMigration](#)
[attribute](#)), 88
[old_user_id \(tardis.apps.openid_migration.models.OpenidUserMigration](#)
[attribute](#)), 88
[online_types \(tardis.tardis_portal.models.storage.StorageBoxAttribute](#)
[attribute](#)), 179
[open\(\) \(tardis.apps.sftp.sftp.MyTSFTPServerInterface](#)
[method](#)), 113
[openid_migration_method\(\) \(in module](#)
[tardis.apps.openid_migration.migration](#)),
86
[openid_migration_processor\(\) \(in module](#)
[tardis.apps.openid_migration.context_processors](#)),
85
[openid_user_migration_form\(\) \(in module](#)
[tardis.apps.openid_migration.forms](#)), 86
[OpenidACLMigration \(class in](#)
[tardis.apps.openid_migration.models](#)), 86
[OpenidACLMigration.DoesNotExist](#), 86
[OpenidACLMigration.MultipleObjectsReturned](#)
86
[openidaclmigration_set \(tardis.apps.openid_migration.models.OpenidUserMigration](#)
[attribute](#)), 88
[openidaclmigration_set \(tardis.tardis_portal.models.access_control.ObjectACL](#)
[attribute](#)), 139
[OpenidACLMigrationAdmin \(class in](#)
[tardis.apps.openid_migration.models](#)), 87
[OpenidMigrationConfig \(class in](#)
[tardis.apps.openid_migration.apps](#)), 85
[OpenIDMigrationFormTestCase \(class in](#)
[tardis.apps.openid_migration.tests.test_forms](#)),
84
[OpenIDMigrationTestCase \(class in](#)
[tardis.apps.openid_migration.tests.test_migration](#)),
84
[OpenIDMigrationViewTestCase \(class in](#)
[tardis.apps.openid_migration.tests.test_views](#)),
85
[OpenidUserMigration \(class in](#)
[tardis.apps.openid_migration.models](#)), 87
[OpenidUserMigration.DoesNotExist](#), 87
[OpenidUserMigration.MultipleObjectsReturned](#)
87
[OpenidUserMigrationAdmin \(class in](#)
[tardis.apps.openid_migration.models](#)), 89
[operations \(tardis.apps.openid_migration.migrations.0001_initial.Migration](#)
[attribute](#)), 84
[operations \(tardis.apps.push_to.migrations.0001_initial.Migration](#)
[attribute](#)), 90
[operations \(tardis.apps.push_to.migrations.0002_auto_20160518_1953](#)
[attribute](#)), 90
[operations \(tardis.apps.sftp.migrations.0001_initial.Migration](#)
[attribute](#)), 108
[operations \(tardis.tardis_portal.migrations.0001_initial.Migration](#)
[attribute](#)), 132
[operations \(tardis.tardis_portal.migrations.0001_squashed_0011_auto](#)
[attribute](#)), 132
[operations \(tardis.tardis_portal.migrations.0002_auto_20150528_1128](#)
[attribute](#)), 133
[operations \(tardis.tardis_portal.migrations.0003_auto_20150907_1315](#)
[attribute](#)), 133
[operations \(tardis.tardis_portal.migrations.0004_storageboxoption_val](#)
[attribute](#)), 133
[operations \(tardis.tardis_portal.migrations.0005_datafile_add_size_int](#)
[attribute](#)), 133
[operations \(tardis.tardis_portal.migrations.0006_datafile_remove_size](#)
[attribute](#)), 133
[operations \(tardis.tardis_portal.migrations.0007_remove_parameter_s](#)
[attribute](#)), 134
[operations \(tardis.tardis_portal.migrations.0008_string_value_partial](#)
[attribute](#)), 134
[operations \(tardis.tardis_portal.migrations.0009_auto_20160128_1119](#)
[attribute](#)), 134
[operations \(tardis.tardis_portal.migrations.0010_auto_20160503_1443](#)
[attribute](#)), 134
[operations \(tardis.tardis_portal.migrations.0011_auto_20160505_1643](#)
[attribute](#)), 134
[operations \(tardis.tardis_portal.migrations.0012_userauthentication_ap](#)
[attribute](#)), 135
[operations \(tardis.tardis_portal.migrations.0013_auto_20181002_1136](#)
[attribute](#)), 135
[operations \(tardis.tardis_portal.migrations.0014_auto_20181002_1154](#)
[attribute](#)), 135
[operations \(tardis.tardis_portal.migrations.0015_dataset_created_time](#)
[attribute](#)), 135
[operations \(tardis.tardis_portal.migrations.0016_add_timestamps.Migr](#)
[attribute](#)), 135
[operations \(tardis.tardis_portal.migrations.0017_add_cc_licenses.Migr](#)
[attribute](#)), 136
[operations \(tardis.tardis_portal.migrations.0018_make_default_storage](#)
[attribute](#)), 136
[options \(tardis.tardis_portal.models.storage.StorageBoxAttribute](#)
[attribute](#)), 179
[OracleSafeManager \(class in](#)

[tardis.tardis_portal.managers\), 238](#)
[order \(tardis.tardis_portal.models.experiment.ExperimentAuthorResourceMeta attribute\), 155](#)
[order \(tardis.tardis_portal.models.parameters.ParameterNameMeta attribute\), 173](#)
[ordering \(tardis.tardis_portal.api.DataFileResource.MetaParallelBulk method\), 219](#)
[ordering \(tardis.tardis_portal.api.DatasetResource.MetaParameter \(class in tardis.tardis_portal.models.parameters\), attribute\), 221](#)
[ordering \(tardis.tardis_portal.api.ExperimentAuthorResource.MetaParameterMeta \(class in tardis.tardis_portal.models.parameters\), attribute\), 222](#)
[ordering \(tardis.tardis_portal.api.ExperimentResource.MetaParameterMeta \(class in tardis.tardis_portal.models.parameters\), attribute\), 223](#)
[ordering \(tardis.tardis_portal.api.FacilityResource.MetaParameterClass \(tardis.tardis_portal.models.parameters.DatafileParameterMeta attribute\), 223](#)
[ordering \(tardis.tardis_portal.api.InstrumentResource.MetaParameterClass \(tardis.tardis_portal.models.parameters.DatasetParameterMeta attribute\), 224](#)
[ordering \(tardis.tardis_portal.api.ObjectACLResource.MetaParameterClass \(tardis.tardis_portal.models.parameters.ExperimentParameterMeta attribute\), 225](#)
[ordering \(tardis.tardis_portal.api.ReplicaResource.MetaParameterClass \(tardis.tardis_portal.models.parameters.InstrumentParameterMeta attribute\), 226](#)
[ordering \(tardis.tardis_portal.api.SchemaResource.MetaParameterClass \(tardis.tardis_portal.models.parameters.ParameterSetMeta attribute\), 226](#)
[ordering \(tardis.tardis_portal.api.StorageBoxAttributeResourceMetaName \(tardis.tardis_portal.models.parameters.FreeTextSearchFieldMeta attribute\), 227](#)
[ordering \(tardis.tardis_portal.api.StorageBoxOptionResourceMetaNameId \(tardis.tardis_portal.models.parameters.FreeTextSearchFieldMeta attribute\), 227](#)
[ordering \(tardis.tardis_portal.api.StorageBoxResource.MetaParameterType \(tardis.tardis_portal.models.parameters.DatafileParameterMeta attribute\), 227](#)
[ordering \(tardis.tardis_portal.models.parameters.ParameterMeta attribute\), 160](#)
[ordering \(tardis.tardis_portal.models.parameters.ParameterSetMeta attribute\), 163](#)
[ordering \(tardis.tardis_portal.models.parameters.ParameterType \(tardis.tardis_portal.models.parameters.ExperimentParameterMeta attribute\), 173](#)
[owned \(\) \(tardis.tardis_portal.managers.ExperimentManager parameter_type \(tardis.tardis_portal.models.parameters.InstrumentParameterMeta attribute\), 237](#)
[owned_and_shared \(\) \(tardis.tardis_portal.managers.ExperimentManager parameter_type \(tardis.tardis_portal.models.parameters.ParameterMeta attribute\), 237](#)
[owned_by_group \(\) \(tardis.tardis_portal.managers.ExperimentManager parameter_type \(tardis.tardis_portal.models.parameters.ParameterMeta attribute\), 237](#)
[owned_by_user \(\) \(tardis.tardis_portal.managers.ExperimentManager parameter_type \(tardis.tardis_portal.models.parameters.ParameterMeta attribute\), 237](#)
[owned_by_user_id \(\) \(tardis.tardis_portal.managers.ExperimentManager parameter_type \(tardis.tardis_portal.models.parameters.ParameterMeta attribute\), 237](#)
[OWNER_OWNED \(tardis.tardis_portal.models.access_control.ObjectACL \(tardis.tardis_portal.templatetags.formfieldfilters\), attribute\), 137](#)
P
[pagination \(\) \(in module tardis.tardis_portal.templatetags.pagination\), 186](#)
[paginator_number \(\) \(in module tardis.tardis_portal.templatetags.pagination\), 186](#)

ParameterNameManager	(class	in	attribute), 197
<i>tardis.tardis_portal.managers</i>),	238		
ParameterNameResource	(class	in	attribute), 197
<i>tardis.tardis_portal.api</i>),	225		
ParameterNameResource.Meta	(class	in	attribute), 197
<i>tardis.tardis_portal.api</i>),	225		
ParameterResource	(class	in	attribute), 197
<i>tardis.tardis_portal.api</i>),	225		
parameters	(<i>tardis.tardis_portal.ParameterSetManager.ParameterSetManager</i> attribute),	213	
ParameterSet	(class	in	attribute), 197
<i>tardis.tardis_portal.models.parameters</i>),	173		
parameterset	(<i>tardis.tardis_portal.models.parameters.DataFileParameter</i> attribute),	160	
parameterset	(<i>tardis.tardis_portal.models.parameters.DatasetParameter</i> attribute),	163	
parameterset	(<i>tardis.tardis_portal.models.parameters.ExperimentParameter</i> attribute),	165	
parameterset	(<i>tardis.tardis_portal.models.parameters.InstrumentParameter</i> attribute),	168	
parameterset	(<i>tardis.tardis_portal.ParameterSetManager.ParameterSetManager</i> attribute),	213	
ParameterSet.Meta	(class	in	post_save_experimentauthor() (in module <i>tardis.tardis_portal.models.parameters</i>),
<i>tardis.tardis_portal.models.parameters</i>),	173		
parameterset_id	(<i>tardis.tardis_portal.models.parameters.DataFileParameter</i> attribute),	161	
parameterset_id	(<i>tardis.tardis_portal.models.parameters.DatasetParameter</i> attribute),	163	
parameterset_id	(<i>tardis.tardis_portal.models.parameters.ExperimentParameter</i> attribute),	165	
parameterset_id	(<i>tardis.tardis_portal.models.parameters.InstrumentParameter</i> attribute),	168	
ParameterSetManager	(class	in	prepend_urls() (<i>tardis.tardis_portal.api.DataFileResource</i> method),
<i>tardis.tardis_portal.ParameterSetManager</i>),	213		
ParameterSetManagerMixin	(class	in	prepend_urls() (<i>tardis.tardis_portal.api.DatasetResource</i> method),
<i>tardis.tardis_portal.models.parameters</i>),	174		
ParameterSetManagerTestCase	(class	in	PrettyJSONSerializer (class in <i>tardis.apps.search.api</i>),
<i>tardis.tardis_portal.tests.test_parametersets</i>),	203		
ParameterSetResource	(class	in	PrettyJSONSerializer (class in <i>tardis.tardis_portal.api</i>),
<i>tardis.tardis_portal.api</i>),	225		
ParametersTestCase	(class	in	priority (<i>tardis.tardis_portal.models.datafile.DataFileObject</i> attribute),
<i>tardis.tardis_portal.tests.test_parameters</i>),	202		
ParameterTestCase	(class	in	priority (<i>tardis.tardis_portal.models.storage.StorageBox</i> attribute),
<i>tardis.tardis_portal.tests.models.test_parameter</i>),	194		
password	(<i>tardis.apps.push_to.models.Credential</i> attribute),	91	
PATH_LDAPADD	(<i>tardis.tardis_portal.tests.slapi.Slapd</i>		private_key (<i>tardis.apps.push_to.models.KeyPair</i> attribute),
			93
			process_exception() (<i>tardis.tardis_portal.logging_middleware.LoggingMiddleware</i> method),
			236
			process_request() (<i>tardis.tardis_portal.auth.token_auth.TokenAuthMiddleware</i> method),
			130

[process_response\(\)](#) (class in `tardis.tardis_portal.logging_middleware.LoggingMiddleware`), 236

[ProxyingMetadataRegistry](#) (class in `tardis.apps.oaipmh.server`), 82

[ProxyingServer](#) (class in `tardis.apps.oaipmh.server`), 82, 249

[public\(\)](#) (`tardis.tardis_portal.managers.ExperimentManager` attribute), 101

[public_access\(\)](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 153

[PUBLIC_ACCESS_CHOICES](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 151

[PUBLIC_ACCESS_EMBARGO](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 151

[PUBLIC_ACCESS_FULL](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 151

[public_access_implies_distribution\(\)](#) (`tardis.tardis_portal.models.experiment.Experiment` class method), 154

[PUBLIC_ACCESS_METADATA](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 151

[PUBLIC_ACCESS_NONE](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 151

[public_data\(\)](#) (in module `tardis.tardis_portal.views.pages`), 211

[public_download_allowed\(\)](#) (`tardis.tardis_portal.models.experiment.Experiment` method), 154

[public_key\(\)](#) (`tardis.apps.push_to.models.KeyPair` attribute), 93

[public_key\(\)](#) (`tardis.apps.sftp.models.SFTPPublicKey` attribute), 111

[PUBLICATION_DETAILS_SCHEMA](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 151

[PUBLICATION_DRAFT_SCHEMA](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 151

[PUBLICATION_SCHEMA_ROOT](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 151

[publish_public_expt_rifcs\(\)](#) (in module `tardis.tardis_portal.models.hooks`), 157

[PublishService](#) (class in `tardis.tardis_portal.publish.publishservice`), 183

[PublishServiceTestCase](#) (class in `tardis.tardis_portal.tests.test_publishservice`),

[put_detail\(\)](#) (`tardis.tardis_portal.api.DataFileResource` method), 220

[queryset\(\)](#) (`tardis.apps.s3utils.api.ReplicaAppResource.Meta` attribute), 101

[queryset\(\)](#) (`tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta` attribute), 109

[queryset\(\)](#) (`tardis.tardis_portal.api.DatafileParameterResource.Meta` attribute), 220

[queryset\(\)](#) (`tardis.tardis_portal.api.DatafileParameterSetResource.Meta` attribute), 220

[queryset\(\)](#) (`tardis.tardis_portal.api.DataFileResource.Meta` attribute), 219

[queryset\(\)](#) (`tardis.tardis_portal.api.DatasetParameterResource.Meta` attribute), 221

[queryset\(\)](#) (`tardis.tardis_portal.api.DatasetParameterSetResource.Meta` attribute), 221

[queryset\(\)](#) (`tardis.tardis_portal.api.DatasetResource.Meta` attribute), 221

[queryset\(\)](#) (`tardis.tardis_portal.api.ExperimentAuthorResource.Meta` attribute), 222

[queryset\(\)](#) (`tardis.tardis_portal.api.ExperimentParameterResource.Meta` attribute), 222

[queryset\(\)](#) (`tardis.tardis_portal.api.ExperimentParameterSetResource.Meta` attribute), 222

[queryset\(\)](#) (`tardis.tardis_portal.api.ExperimentResource.Meta` attribute), 223

[queryset\(\)](#) (`tardis.tardis_portal.api.FacilityResource.Meta` attribute), 223

[queryset\(\)](#) (`tardis.tardis_portal.api.GroupResource.Meta` attribute), 224

[queryset\(\)](#) (`tardis.tardis_portal.api.InstrumentResource.Meta` attribute), 224

[queryset\(\)](#) (`tardis.tardis_portal.api.LocationResource.Meta` attribute), 224

[queryset\(\)](#) (`tardis.tardis_portal.api.ObjectACLResource.Meta` attribute), 225

[queryset\(\)](#) (`tardis.tardis_portal.api.ParameterNameResource.Meta` attribute), 225

[queryset\(\)](#) (`tardis.tardis_portal.api.ReplicaResource.Meta` attribute), 226

[queryset\(\)](#) (`tardis.tardis_portal.api.SchemaResource.Meta` attribute), 226

[queryset\(\)](#) (`tardis.tardis_portal.api.StorageBoxAttributeResource.Meta` attribute), 227

[queryset\(\)](#) (`tardis.tardis_portal.api.StorageBoxOptionResource.Meta` attribute), 227

[queryset\(\)](#) (`tardis.tardis_portal.api.StorageBoxResource.Meta` attribute), 227

[queryset\(\)](#) (`tardis.tardis_portal.api.UserResource.Meta` attribute), 228

[queryset_pagination](#) (*tardis.apps.search.documents.DataFileDocument*, *attribute*), 105
[quote\(\)](#) (in module *tardis.tardis_portal.tests.slapped*), 198
R
[RANGE_COMPARISON](#) (*tardis.tardis_portal.models.parameters.ParameterName*, *attribute*), 171
[rapidConnectEduPersonTargetedID](#) (*tardis.tardis_portal.models.access_control.UserProfile*, *attribute*), 140
[rcauth\(\)](#) (in module *tardis.tardis_portal.views.authentication*), 207
[read_detail\(\)](#) (*tardis.apps.sftp.api.SFTPAuthorization*, *method*), 109
[read_detail\(\)](#) (*tardis.tardis_portal.api.ACLAuthorization*, *method*), 219
[read_list\(\)](#) (*tardis.apps.sftp.api.SFTPAuthorization*, *method*), 109
[read_list\(\)](#) (*tardis.tardis_portal.api.ACLAuthorization*, *method*), 219
[readMetadata\(\)](#) (*tardis.apps.oaipmh.server.ProxyingMetadataRegistry*, *method*), 82
[redirect_back_with_error\(\)](#) (in module *tardis.tardis_portal.shortcuts*), 239
[RegisterExperimentForm](#) (class in *tardis.tardis_portal.forms*), 234
[registerReader\(\)](#) (*tardis.apps.oaipmh.server.ProxyingMetadataRegistry*, *method*), 82
[registerWriter\(\)](#) (*tardis.apps.oaipmh.server.ProxyingMetadataRegistry*, *method*), 82
[REGISTRATION_OPEN](#) (in module *tardis.default_settings.auth*), 119
[registration_processor\(\)](#) (in module *tardis.tardis_portal.context_processors*), 228
[RegistrationForm](#) (class in *tardis.tardis_portal.forms*), 234
[related_models](#) (*tardis.apps.search.documents.DataFileDocument*, *attribute*), 105
[related_models](#) (*tardis.apps.search.documents.DatasetDocument*, *attribute*), 106
[related_models](#) (*tardis.apps.search.documents.ExperimentDocument*, *attribute*), 106
[RelatedInfoForm](#) (class in *tardis.apps.related_info.forms*), 100
[remote_hosts](#) (*tardis.apps.push_to.models.Credential*, *attribute*), 91
[remote_user](#) (*tardis.apps.push_to.models.Credential*, *attribute*), 92
[RemoteHost](#) (class in *tardis.apps.push_to.models*), 95
[RemoteHost.DoesNotExist](#), 95
[RemoteHost.MultipleObjectsReturned](#), 95
[RemoteHostAdmin](#) (class in *tardis.apps.push_to.models*), 96
[remove_auth_method\(\)](#) (in module *tardis.tardis_portal.auth.authentication*), 125
[remove_csrf_token\(\)](#) (in module *tardis.tardis_portal.views.utils*), 212
[remove_licenses\(\)](#) (in module *tardis.tardis_portal.migrations.0017_add_cc_licenses*), 136
[RemovedInMyTardis42Warning](#), 228
[RemovedInMyTardis43Warning](#), 229
[render\(\)](#) (*tardis.tardis_portal.forms.NoInput*, *method*), 234
[render\(\)](#) (*tardis.tardis_portal.templatetags.capture.CaptureNode*, *method*), 185
[render\(\)](#) (*tardis.tardis_portal.templatetags.dynurl.DynUrlNode*, *method*), 185
[render\(\)](#) (*tardis.tardis_portal.widgets.CommaSeparatedInput*, *method*), 240
[render\(\)](#) (*tardis.tardis_portal.widgets.Label*, *method*), 241
[render_error_message\(\)](#) (in module *tardis.apps.push_to.views*), 98
[render_error_message\(\)](#) (in module *tardis.tardis_portal.shortcuts*), 239
[RENDER_IMAGE_DATASET_SIZE_LIMIT](#) (in module *tardis.default_settings.frontend*), 122
[RENDER_IMAGE_SIZE_LIMIT](#) (in module *tardis.default_settings.frontend*), 122
[render_response_index\(\)](#) (in module *tardis.tardis_portal.shortcuts*), 239
[render_success_message\(\)](#) (in module *tardis.apps.push_to.views*), 98
[render_to_file\(\)](#) (in module *tardis.tardis_portal.shortcuts*), 240
[replaces](#) (*tardis.tardis_portal.migrations.0001_squashed_0011_auto_2012-01-10_17_02*, *attribute*), 132
[ReplicaAppResource](#) (class in *tardis.apps.s3utils.api*), 101
[ReplicaAppResource.Meta](#) (class in *tardis.apps.s3utils.api*), 101
[ReplicaResource](#) (class in *tardis.tardis_portal.api*), 226
[ReplicaResource.Meta](#) (class in *tardis.tardis_portal.api*), 226
[ReplicaResourceTest](#) (class in *tardis.tardis_portal.tests.api.test_replica_resource*), 189
[REQUIRE_DATAFILE_CHECKSUMS](#) (*tardis.default_settings*, *attribute*), 45
[REQUIRE_DATAFILE_SIZES](#) (*tardis.default_settings*, *attribute*), 45

- [REQUIRE_SSL_TO_GENERATE_KEY](#) (in module [tardis.apps.sftp.default_settings](#)), 110
[REQUIRE_VALIDATION_ON_INGESTION](#) ([tardis.default_settings](#) attribute), 45
[requires_admin_approval\(\)](#) (in module [tardis.apps.social_auth.auth.social_auth](#)), 118
[resource_name](#) ([tardis.apps.s3utils.api.ReplicaAppResource.Meta](#) attribute), 101
[resource_name](#) ([tardis.apps.search.api.AdvanceSearchAppResource.Meta](#) attribute), 104
[resource_name](#) ([tardis.apps.search.api.SearchAppResource.Meta](#) attribute), 105
[resource_name](#) ([tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta](#) attribute), 109
[resource_name](#) ([tardis.tardis_portal.api.DataFileResource.Meta](#) attribute), 219
[restart\(\)](#) ([tardis.tardis_portal.tests.slapped.Slapped](#) method), 197
[RestfulExperimentParameterSet](#) (class in [tardis.tardis_portal.shortcuts](#)), 239
[retrieve_access_list_group_readonly\(\)](#) (in module [tardis.tardis_portal.views.authorisation](#)), 207
[retrieve_access_list_user_readonly\(\)](#) (in module [tardis.tardis_portal.views.authorisation](#)), 207
[retrieve_group_list\(\)](#) (in module [tardis.tardis_portal.views.authorisation](#)), 207
[retrieve_group_list_by_user\(\)](#) (in module [tardis.tardis_portal.views.authorisation](#)), 207
[retrieve_group_userlist_readonly\(\)](#) (in module [tardis.tardis_portal.views.authorisation](#)), 207
[retrieve_licenses\(\)](#) (in module [tardis.tardis_portal.views.ajax_json](#)), 206
[retrieve_owned_exps_list\(\)](#) (in module [tardis.tardis_portal.views.ajax_pages](#)), 207
[retrieve_shared_exps_list\(\)](#) (in module [tardis.tardis_portal.views.ajax_pages](#)), 207
[retrieve_user_list\(\)](#) (in module [tardis.tardis_portal.views.authorisation](#)), 207
[return_response_error\(\)](#) (in module [tardis.tardis_portal.shortcuts](#)), 240
[return_response_error_message\(\)](#) (in module [tardis.tardis_portal.shortcuts](#)), 240
[return_response_not_found\(\)](#) (in module [tardis.tardis_portal.shortcuts](#)), 240
[REUSE_DATASET_STORAGE_BOX](#) (in module [tardis.default_settings.storage](#)), 123
[reverse_func\(\)](#) (in module [tardis.tardis_portal.migrations.0017_add_cc_licenses](#)), 136
[RFC](#) RFC 3339, 239
[rfc3339\(\)](#) (in module [tardis.tardis_portal.rfc3339](#)), 239
[RifCsExperimentProvider](#) (class in [tardis.apps.oaipmh.provider.experiment](#)), 80
[RifCsExperimentProvider.ExperimentWriter](#) (class in [tardis.apps.oaipmh.provider.experiment](#)), 80
[RifCsExperimentProviderTestCase](#) (class in [tardis.apps.oaipmh.tests.provider.test_experiment](#)), 80
[RifCsProvider](#) (class in [tardis.tardis_portal.publish.provider.rifcsprovider](#)), 183
[RifCSTestCase](#) (class in [tardis.apps.related_info.tests.test_oaipmh](#)), 98
[RightsForm](#) (class in [tardis.tardis_portal.forms](#)), 235
[RightsForm.Meta](#) (class in [tardis.tardis_portal.forms](#)), 235
[RightsFormTestCase](#) (class in [tardis.tardis_portal.tests.test_forms](#)), 201
[RightsTestCase](#) (class in [tardis.tardis_portal.tests.views.test_auth_views](#)), 194
[RmExperimentTestCase](#) (class in [tardis.tardis_portal.tests.management.test_rmexperiment](#)), 192
[rollback_migration\(\)](#) (in module [tardis.apps.openid_migration.utils](#)), 89
- ## S
- [S3_SIGNED_URL_EXPIRY](#) (in module [tardis.apps.s3utils.default_settings](#)), 102
[S3UtilsAppApiTestCase](#) (class in [tardis.apps.s3utils.tests.test_api](#)), 100
[S3UtilsAppChecksumsTestCase](#) (class in [tardis.apps.s3utils.tests.test_checksums](#)), 101
[S3UtilsConfig](#) (class in [tardis.apps.s3utils.apps](#)), 102
[S3UtilsConfigTestCase](#) (class in [tardis.apps.s3utils.tests.test_app_config](#)), 101
[safe](#) ([tardis.tardis_portal.models.experiment.Experiment](#) attribute), 154
[sanitize_html\(\)](#) (in module [tardis.tardis_portal.templatetags.formfieldfilters](#)), 186
[save\(\)](#) ([tardis.apps.push_to.models.KeyPair](#) method), 93

`save()` (`tardis.tardis_portal.forms.ExperimentForm` `method`), 232
`save()` (`tardis.tardis_portal.forms.RegistrationForm` `method`), 235
`save()` (`tardis.tardis_portal.models.access_control.UserAuthenticationManager` `method`), 140
`save()` (`tardis.tardis_portal.models.datafile.DataFile` `method`), 144
`save()` (`tardis.tardis_portal.models.datafile.DataFileObject` `method`), 146
`save()` (`tardis.tardis_portal.models.dataset.Dataset` `method`), 150
`save()` (`tardis.tardis_portal.models.experiment.Experiment` `method`), 154
`save()` (`tardis.tardis_portal.models.experiment.ExperimentAuthor` `method`), 155
`save()` (`tardis.tardis_portal.models.facility.Facility` `method`), 156
`save()` (`tardis.tardis_portal.models.instrument.Instrument` `method`), 158
`save()` (`tardis.tardis_portal.models.parameters.ExperimentParameterSet` `method`), 165
`save()` (`tardis.tardis_portal.models.parameters.ParameterSet` `method`), 173
`save_m2m()` (`tardis.tardis_portal.forms.ExperimentForm.FullExperimentForm` `method`), 232
`save_parameter_add_form()` (in module `tardis.tardis_portal.forms`), 236
`save_parameter_edit_form()` (in module `tardis.tardis_portal.forms`), 236
`save_with_random_token()` (`tardis.tardis_portal.models.token.Token` `method`), 182
`Schema` (class in `tardis.tardis_portal.models.parameters`), 174
`schema` (`tardis.tardis_portal.models.parameters.DatafileParameterSet` `attribute`), 161
`schema` (`tardis.tardis_portal.models.parameters.DatasetParameterSet` `attribute`), 164
`schema` (`tardis.tardis_portal.models.parameters.ExperimentParameterSet` `attribute`), 166
`schema` (`tardis.tardis_portal.models.parameters.InstrumentParameterSet` `attribute`), 169
`schema` (`tardis.tardis_portal.models.parameters.ParameterName` `attribute`), 173
`schema` (`tardis.tardis_portal.models.parameters.ParameterSet` `attribute`), 174
`schema` (`tardis.tardis_portal.shortcuts.RestfulExperimentParameterSet` `attribute`), 239
`Schema.DoesNotExist`, 174
`Schema.MultipleObjectsReturned`, 174
`Schema.UnsupportedType`, 174
`schema_id` (`tardis.tardis_portal.models.parameters.ParameterName` `attribute`), 173
`schema_id` (`tardis.tardis_portal.models.parameters.ParameterSet` `attribute`), 174
`SchemaAdmin` (class in `tardis.tardis_portal.admin`), 216
`SchemaManager` (class in `tardis.tardis_portal.managers`), 238
`SchemaResource` (class in `tardis.tardis_portal.api`), 226
`SchemaResource.Meta` (class in `tardis.tardis_portal.api`), 226
`SchemaResourceTest` (class in `tardis.tardis_portal.tests.api.test_schema_resource`), 190
`search_fields` (`tardis.tardis_portal.admin.DatafileAdmin` `attribute`), 214
`search_fields` (`tardis.tardis_portal.admin.DatasetAdmin` `attribute`), 214
`search_fields` (`tardis.tardis_portal.admin.ExperimentAdmin` `attribute`), 215
`search_fields` (`tardis.tardis_portal.admin.FacilityAdmin` `attribute`), 215
`search_fields` (`tardis.tardis_portal.admin.InstrumentAdmin` `attribute`), 215
`search_fields` (`tardis.tardis_portal.admin.ObjectACLAdmin` `attribute`), 216
`search_fields` (`tardis.tardis_portal.admin.ParameterNameAdmin` `attribute`), 216
`search_fields` (`tardis.tardis_portal.admin.SchemaAdmin` `attribute`), 216
`search_fields` (`tardis.tardis_portal.admin.StorageBoxAdmin` `attribute`), 216
`search_fields` (`tardis.tardis_portal.admin.UserAuthenticationAdmin` `attribute`), 218
`SearchAppResource` (class in `tardis.apps.search.api`), 104
`SearchAppResource.Meta` (class in `tardis.apps.search.api`), 104
`SearchAppResource.config` (class in `tardis.apps.search.apps`), 105
`searchEntities()` (`tardis.tardis_portal.auth.authservice.AuthService` `method`), 126
`searchGroups()` (`tardis.tardis_portal.auth.authservice.AuthService` `method`), 126
`searchGroups()` (`tardis.tardis_portal.auth.interfaces.GroupProvider` `method`), 128
`searchGroups()` (`tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider` `method`), 129
`searchGroups()` (`tardis.tardis_portal.auth.token_auth.TokenGroupProvider` `method`), 130
`searchGroups()` (`tardis.tardis_portal.tests.test_authservice.MockGroupProvider` `method`), 199
`SearchObject` (class in `tardis.apps.search.api`), 105
`searchUsers()` (`tardis.tardis_portal.auth.authservice.AuthService` `method`), 126
`searchUsers()` (`tardis.tardis_portal.auth.interfaces.UserProvider` `method`), 126

[method](#)), 129
[SearchView](#) (class in [tardis.apps.search.views](#)), 107
[SECRET_KEY](#), 44
[send_admin_email\(\)](#) (in module [tardis.apps.social_auth.auth.social_auth](#)), 118
[serializer](#) ([tardis.apps.search.api.AdvanceSearchAppResource.Meta](#) attribute), 104
[serializer](#) ([tardis.apps.search.api.SearchAppResource.Meta](#) attribute), 105
[serializer](#) ([tardis.tardis_portal.api.MyTardisModelResource.Meta](#) attribute), 224
[serializer](#) ([tardis.tardis_portal.api.UserResource.Meta](#) attribute), 228
[SerializerTest](#) (class in [tardis.tardis_portal.tests.api.test_serializer](#)), 190
[service](#) (in module [tardis.analytics.tracker](#)), 75
[session_ended\(\)](#) ([tardis.apps.sftp.sftp.MyTSFTPServerInterface](#) method), 113
[session_started\(\)](#) ([tardis.apps.sftp.sftp.MyTSFTPServerInterface](#) method), 114
[set_debug\(\)](#) ([tardis.tardis_portal.tests.slapd.Slapd](#) method), 197
[set_dn_suffix\(\)](#) ([tardis.tardis_portal.tests.slapd.Slapd](#) method), 197
[set_param\(\)](#) ([tardis.tardis_portal.ParameterSetManager.ParameterSetManager](#) method), 213
[set_param_list\(\)](#) ([tardis.tardis_portal.ParameterSetManager.ParameterSetManager](#) method), 213
[set_params_from_dict\(\)](#) ([tardis.tardis_portal.ParameterSetManager.ParameterSetManager](#) method), 213
[set_port\(\)](#) ([tardis.tardis_portal.tests.slapd.Slapd](#) method), 197
[set_root_cn\(\)](#) ([tardis.tardis_portal.tests.slapd.Slapd](#) method), 197
[set_root_password\(\)](#) ([tardis.tardis_portal.tests.slapd.Slapd](#) method), 197
[set_slapd_debug_level\(\)](#) ([tardis.tardis_portal.tests.slapd.Slapd](#) method), 197
[set_tmpdir\(\)](#) ([tardis.tardis_portal.tests.slapd.Slapd](#) method), 197
[set_token\(\)](#) (in module [tardis.apps.push_to.oauth_tokens](#)), 96
[set_value\(\)](#) ([tardis.tardis_portal.models.parameters.Parameter](#) method), 170
[setPost\(\)](#) ([tardis.tardis_portal.tests.test_authservice.MockRequest](#) method), 199
[settings](#) ([tardis.apps.search.documents.DataFileDocument.Index](#) attribute), 105
[settings](#) ([tardis.apps.search.documents.DatasetDocument.Index](#) attribute), 106
[settings](#) ([tardis.apps.search.documents.ExperimentDocument.Index](#) attribute), 106
[setUp\(\)](#) ([tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase](#) method), 80
[setUp\(\)](#) ([tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentTestCase](#) method), 81
[setUp\(\)](#) ([tardis.apps.oaipmh.tests.test_oai.EndpointTestCase](#) method), 81
[setUp\(\)](#) ([tardis.apps.openid_migration.tests.test_forms.OpenIDMigrationTestCase](#) method), 84
[setUp\(\)](#) ([tardis.apps.openid_migration.tests.test_migration.OpenIDMigrationTestCase](#) method), 85
[setUp\(\)](#) ([tardis.apps.openid_migration.tests.test_models.ModelTestCase](#) method), 85
[setUp\(\)](#) ([tardis.apps.openid_migration.tests.test_views.OpenIDMigrationTestCase](#) method), 85
[setUp\(\)](#) ([tardis.apps.push_to.tests.test_models.ModelsTestCase](#) method), 90
[setUp\(\)](#) ([tardis.apps.related_info.tests.test_oaipmh.RifCSTestCase](#) method), 98
[setUp\(\)](#) ([tardis.apps.related_info.tests.tests.CreateTestCase](#) method), 99
[setUp\(\)](#) ([tardis.apps.related_info.tests.tests.DeleteTestCase](#) method), 99
[setUp\(\)](#) ([tardis.apps.related_info.tests.tests.GetTestCase](#) method), 99
[setUp\(\)](#) ([tardis.apps.related_info.tests.tests.ListTestCase](#) method), 99
[setUp\(\)](#) ([tardis.apps.related_info.tests.tests.TabTestCase](#) method), 99
[setUp\(\)](#) ([tardis.apps.related_info.tests.tests.UpdateTestCase](#) method), 99
[setUp\(\)](#) ([tardis.apps.s3utils.tests.test_api.S3UtilsAppApiTestCase](#) method), 100
[setUp\(\)](#) ([tardis.apps.s3utils.tests.test_checksums.S3UtilsAppChecksumsTestCase](#) method), 101
[setUp\(\)](#) ([tardis.apps.search.tests.test_api.SimpleSearchTestCase](#) method), 103
[setUp\(\)](#) ([tardis.apps.search.tests.test_index.IndexExperimentTestCase](#) method), 103
[setUp\(\)](#) ([tardis.apps.sftp.sftp.MyTSFTPRequestHandler](#) method), 112
[setUp\(\)](#) ([tardis.apps.sftp.tests.test_sftp.SFTPTTest](#) method), 108
[setUp\(\)](#) ([tardis.tardis_portal.tests.api.MyTardisResourceTestCase](#) method), 191
[setUp\(\)](#) ([tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResourceTestCase](#) method), 187
[setUp\(\)](#) ([tardis.tardis_portal.tests.api.test_dataset_metadata_resources.DatasetMetadataResourceTestCase](#) method), 187
[setUp\(\)](#) ([tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTestCase](#) method), 188

tardis.default_settings.search), 122
single_search_processor() (in module *tardis.tardis_portal.context_processors*), 228
site_routed_view() (in module *tardis.tardis_portal.views.pages*), 211
site_settings() (in module *tardis.tardis_portal.views.machine*), 208
SITE_TITLE (in module *tardis.default_settings.site_customisations*), 123
size (*tardis.tardis_portal.models.datafile.DataFile* attribute), 144
size() (in module *tardis.tardis_portal.templatetags.formfield_filters*), 186
Slapd (class in *tardis.tardis_portal.tests.slapd*), 196
SocialAuthConfig (class in *tardis.apps.social_auth.apps*), 118
Span (class in *tardis.tardis_portal.widgets*), 241
split_path() (in module *tardis.tardis_portal.util*), 240
SPONSORED_TEXT (in module *tardis.default_settings.site_customisations*), 123
start() (*tardis.tardis_portal.tests.slapd.Slapd* method), 197
start_server() (in module *tardis.apps.sftp.sftp*), 116
start_time (*tardis.tardis_portal.models.experiment.Experiment* attribute), 154
started() (*tardis.tardis_portal.tests.slapd.Slapd* method), 197
stat() (*tardis.apps.sftp.sftp.MyTSFTPHandle* method), 112
stat() (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* method), 114
STATIC_ROOT (*tardis.default_settings* attribute), 48
STATIC_URL (*tardis.default_settings* attribute), 48
StaticField (class in *tardis.tardis_portal.forms*), 235
stats() (in module *tardis.tardis_portal.views.pages*), 211
status (*tardis.tardis_portal.models.datafile.DataFile* attribute), 144
status (*tardis.tardis_portal.models.storage.StorageBox* attribute), 179
status_code (*tardis.tardis_portal.views.utils.HttpResponseMethodNotAllowed* attribute), 212
status_code (*tardis.tardis_portal.views.utils.HttpResponseServerError* attribute), 212
stop() (*tardis.tardis_portal.tests.slapd.Slapd* method), 198
storage_box (*tardis.tardis_portal.models.datafile.DataFileObject* attribute), 146
storage_box (*tardis.tardis_portal.models.parameters.DatasetParameterSet* attribute), 162
storage_box (*tardis.tardis_portal.models.parameters.ExperimentParameterSet* attribute), 164
storage_box (*tardis.tardis_portal.models.parameters.InstrumentParameterSet* attribute), 166
storage_box (*tardis.tardis_portal.models.parameters.ParameterSet* attribute), 174
storage_box (*tardis.tardis_portal.models.storage.StorageBoxAttribute* attribute), 180
storage_box (*tardis.tardis_portal.models.storage.StorageBoxOption* attribute), 181
storage_box_id (*tardis.tardis_portal.models.datafile.DataFileObject* attribute), 146
storage_box_id (*tardis.tardis_portal.models.storage.StorageBoxAttribute* attribute), 180
storage_box_id (*tardis.tardis_portal.models.storage.StorageBoxOption* attribute), 181
storage_type (*tardis.tardis_portal.models.datafile.DataFileObject* attribute), 146
storage_type (*tardis.tardis_portal.models.storage.StorageBox* attribute), 179
StorageBox (class in *tardis.tardis_portal.models.storage*), 176
StorageBox.DoesNotExist, 176
StorageBox.MultipleObjectsReturned, 176
StorageBoxAdmin (class in *tardis.tardis_portal.admin*), 216
StorageBoxAttribute (class in *tardis.tardis_portal.models.storage*), 180
StorageBoxAttribute.DoesNotExist, 180
StorageBoxAttribute.MultipleObjectsReturned, 180
StorageBoxAttributeInline (class in *tardis.tardis_portal.admin*), 216
StorageBoxAttributeInlineForm (class in *tardis.tardis_portal.admin*), 216
StorageBoxAttributeInlineForm.Meta (class in *tardis.tardis_portal.admin*), 217
StorageBoxAttributeResource (class in *tardis.tardis_portal.api*), 226
StorageBoxAttributeResource.Meta (class in *tardis.tardis_portal.api*), 227
StorageBoxAttributeResourceTest (class in *tardis.tardis_portal.tests.api.test_storagebox_resources*), 190
StorageBoxForm (class in *tardis.tardis_portal.admin*), 217
StorageBoxForm.Meta (class in *tardis.tardis_portal.admin*), 217
StorageBoxOption (class in *tardis.tardis_portal.models.storage*), 180
StorageBoxOption.DoesNotExist, 180

StorageBoxOption.MultipleObjectsReturned (module), 244

StorageBoxOptionInline (class in [tardis.tardis_portal.admin](#)), 217

StorageBoxOptionInlineForm (class in [tardis.tardis_portal.admin](#)), 218

StorageBoxOptionInlineForm.Meta (class in [tardis.tardis_portal.admin](#)), 218

StorageBoxOptionResource (class in [tardis.tardis_portal.api](#)), 227

StorageBoxOptionResource.Meta (class in [tardis.tardis_portal.api](#)), 227

StorageBoxOptionResourceTest (class in [tardis.tardis_portal.tests.api.test_storagebox_resources](#)), 190

StorageBoxResource (class in [tardis.tardis_portal.api](#)), 227

StorageBoxResource.Meta (class in [tardis.tardis_portal.api](#)), 227

StorageBoxResourceTest (class in [tardis.tardis_portal.tests.api.test_storagebox_resources](#)), 190

streaming_download_datafiles() (in module [tardis.tardis_portal.download](#)), 229

STRING ([tardis.tardis_portal.models.parameters.ParameterName](#) attribute), 171

STRING ([tardis.tardis_portal.models.storage.StorageBoxOption](#) attribute), 181

string_value ([tardis.tardis_portal.models.parameters.Parameter](#) attribute), 170

subtype ([tardis.tardis_portal.models.parameters.Schema](#) attribute), 176

suite() (in module [tardis.tardis_portal.tests.tests](#)), 206

sum_sizes() ([tardis.tardis_portal.models.datafile.DataFile](#) class method), 144

supports_anonymous_user ([tardis.tardis_portal.auth.authorisation.ACLAwareBackend](#) attribute), 125

supports_object_permissions ([tardis.tardis_portal.auth.authorisation.ACLAwareBackend](#) attribute), 125

SYSTEM_OWNED ([tardis.tardis_portal.models.access_control.ObjectACL](#) attribute), 137

system_owned_groups() ([tardis.tardis_portal.managers.ExperimentManager](#) method), 238

T

TabTestCase (class in [tardis.apps.related_info.tests.tests](#)), 99

tag ([tardis.tardis_portal.widgets.Label](#) attribute), 241

tag ([tardis.tardis_portal.widgets.Span](#) attribute), 241

TAPE ([tardis.tardis_portal.models.storage.StorageBox](#) attribute), 176

[tardis \(module\)](#), 244

[tardis.analytics \(module\)](#), 76

[tardis.analytics.apps \(module\)](#), 75

[tardis.analytics.ga \(module\)](#), 75

[tardis.analytics.tracker \(module\)](#), 75

[tardis.app_config \(module\)](#), 243

[tardis.apps \(module\)](#), 118

[tardis.apps.deep_storage_download_mapper \(module\)](#), 76

[tardis.apps.deep_storage_download_mapper.mapper \(module\)](#), 76

[tardis.apps.deep_storage_download_mapper.urls \(module\)](#), 76

[tardis.apps.dl_mapper_df_dir_only \(module\)](#), 77

[tardis.apps.dl_mapper_df_dir_only.mapper \(module\)](#), 76

[tardis.apps.dl_mapper_df_dir_only.urls \(module\)](#), 77

[tardis.apps.filepicker \(module\)](#), 77

[tardis.apps.filepicker.filepicker_settings \(module\)](#), 77

[tardis.apps.filepicker.urls \(module\)](#), 77

[tardis.apps.filepicker.utils \(module\)](#), 77

[tardis.apps.filepicker.views \(module\)](#), 77

[tardis.apps.oaipmh \(module\)](#), 84

[tardis.apps.oaipmh.models \(module\)](#), 82

[tardis.apps.oaipmh.provider \(module\)](#), 80

[tardis.apps.oaipmh.provider.base \(module\)](#), 77

[tardis.apps.oaipmh.provider.experiment \(module\)](#), 79

[tardis.apps.oaipmh.server \(module\)](#), 82

[tardis.apps.oaipmh.tests \(module\)](#), 82

[tardis.apps.oaipmh.tests.provider \(module\)](#), 81

[tardis.apps.oaipmh.tests.provider.test_base \(module\)](#), 80

[tardis.apps.oaipmh.tests.provider.test_experiment \(module\)](#), 81

[tardis.apps.oaipmh.tests.test_oai \(module\)](#), 81

[tardis.apps.oaipmh.urls \(module\)](#), 84

[tardis.apps.oaipmh.views \(module\)](#), 84

[tardis.apps.openid_migration \(module\)](#), 89

[tardis.apps.openid_migration.apps \(module\)](#), 85

[tardis.apps.openid_migration.context_processors \(module\)](#), 85

[tardis.apps.openid_migration.default_settings \(module\)](#), 86

[tardis.apps.openid_migration.email_text \(module\)](#), 86

tardis.apps.openid_migration.forms (*module*), 86
 tardis.apps.openid_migration.migration (*module*), 86
 tardis.apps.openid_migration.migrations (*module*), 84
 tardis.apps.openid_migration.migrations.0001_initial (*module*), 84
 tardis.apps.openid_migration.models (*module*), 86
 tardis.apps.openid_migration.tasks (*module*), 89
 tardis.apps.openid_migration.tests (*module*), 85
 tardis.apps.openid_migration.tests.test_forms (*module*), 84
 tardis.apps.openid_migration.tests.test_migrations (*module*), 84
 tardis.apps.openid_migration.tests.test_models (*module*), 85
 tardis.apps.openid_migration.tests.test_views (*module*), 85
 tardis.apps.openid_migration.urls (*module*), 89
 tardis.apps.openid_migration.user_menu (*module*), 89
 tardis.apps.openid_migration.utils (*module*), 89
 tardis.apps.openid_migration.views (*module*), 89
 tardis.apps.push_to (*module*), 98
 tardis.apps.push_to.apps (*module*), 90
 tardis.apps.push_to.exceptions (*module*), 91
 tardis.apps.push_to.migrations (*module*), 90
 tardis.apps.push_to.migrations.0001_initial (*module*), 90
 tardis.apps.push_to.migrations.0002_auto_20150518_1953 (*module*), 90
 tardis.apps.push_to.models (*module*), 91
 tardis.apps.push_to.oauth_tokens (*module*), 96
 tardis.apps.push_to.ssh_authz (*module*), 97
 tardis.apps.push_to.tasks (*module*), 97
 tardis.apps.push_to.tests (*module*), 90
 tardis.apps.push_to.tests.test_models (*module*), 90
 tardis.apps.push_to.urls (*module*), 97
 tardis.apps.push_to.utils (*module*), 97
 tardis.apps.push_to.views (*module*), 97
 tardis.apps.related_info (*module*), 100
 tardis.apps.related_info.forms (*module*), 100
 tardis.apps.related_info.models (*module*), 100
 tardis.apps.related_info.settings (*module*), 100
 tardis.apps.related_info.tests (*module*), 100
 tardis.apps.related_info.tests.test_oaipmh (*module*), 98
 tardis.apps.related_info.tests.tests (*module*), 99
 tardis.apps.related_info.urls (*module*), 100
 tardis.apps.related_info.views (*module*), 100
 tardis.apps.s3utils (*module*), 103
 tardis.apps.s3utils.api (*module*), 101
 tardis.apps.s3utils.apps (*module*), 102
 tardis.apps.s3utils.default_settings (*module*), 102
 tardis.apps.s3utils.tests (*module*), 101
 tardis.apps.s3utils.tests.test_api (*module*), 100
 tardis.apps.s3utils.tests.test_app_config (*module*), 101
 tardis.apps.s3utils.tests.test_checksums (*module*), 101
 tardis.apps.s3utils.urls (*module*), 102
 tardis.apps.s3utils.utils (*module*), 102
 tardis.apps.search (*module*), 107
 tardis.apps.search.api (*module*), 103
 tardis.apps.search.apps (*module*), 105
 tardis.apps.search.documents (*module*), 105
 tardis.apps.search.tests (*module*), 103
 tardis.apps.search.tests.test_api (*module*), 103
 tardis.apps.search.tests.test_index (*module*), 103
 tardis.apps.search.urls (*module*), 107
 tardis.apps.search.views (*module*), 107
 tardis.apps.sftp (*module*), 117
 tardis.apps.sftp.admin (*module*), 108
 tardis.apps.sftp.api (*module*), 108
 tardis.apps.sftp.apps (*module*), 110
 tardis.apps.sftp.default_settings (*module*), 110
 tardis.apps.sftp.forms (*module*), 110
 tardis.apps.sftp.management (*module*), 107
 tardis.apps.sftp.management.commands (*module*), 107
 tardis.apps.sftp.management.commands.sftpd (*module*), 107
 tardis.apps.sftp.migrations (*module*), 108
 tardis.apps.sftp.migrations.0001_initial (*module*), 108

`tardis.apps.sftp.models` (*module*), 111
`tardis.apps.sftp.sftp` (*module*), 112
`tardis.apps.sftp.tests` (*module*), 108
`tardis.apps.sftp.tests.test_sftp` (*module*), 108
`tardis.apps.sftp.urls` (*module*), 117
`tardis.apps.sftp.user_menu_modifiers` (*module*), 117
`tardis.apps.sftp.views` (*module*), 117
`tardis.apps.social_auth` (*module*), 118
`tardis.apps.social_auth.apps` (*module*), 118
`tardis.apps.social_auth.auth` (*module*), 118
`tardis.apps.social_auth.auth.social_auth` (*module*), 117
`tardis.apps.social_auth.default_settings` (*module*), 118
`tardis.celery` (*module*), 244
`tardis.default_settings` (*module*), 124
`tardis.default_settings.admins` (*module*), 119
`tardis.default_settings.analytics` (*module*), 119
`tardis.default_settings.apps` (*module*), 119
`tardis.default_settings.auth` (*module*), 119
`tardis.default_settings.caches` (*module*), 119
`tardis.default_settings.celery_settings` (*module*), 119
`tardis.default_settings.custom_views` (*module*), 119
`tardis.default_settings.database` (*module*), 120
`tardis.default_settings.debug` (*module*), 120
`tardis.default_settings.downloads` (*module*), 121
`tardis.default_settings.email` (*module*), 121
`tardis.default_settings.filters` (*module*), 121
`tardis.default_settings.frontend` (*module*), 122
`tardis.default_settings.i18n` (*module*), 122
`tardis.default_settings.localisation` (*module*), 122
`tardis.default_settings.logging` (*module*), 122
`tardis.default_settings.middlewares` (*module*), 122
`tardis.default_settings.publication` (*module*), 122
`tardis.default_settings.search` (*module*), 122
`tardis.default_settings.sharing` (*module*), 123
`tardis.default_settings.site_customisations` (*module*), 123
`tardis.default_settings.static_files` (*module*), 123
`tardis.default_settings.storage` (*module*), 123
`tardis.default_settings.templates` (*module*), 124
`tardis.default_settings.uploads` (*module*), 124
`tardis.default_settings.urls` (*module*), 124
`tardis.tardis_portal` (*module*), 242, 244
`tardis.tardis_portal.admin` (*module*), 213
`tardis.tardis_portal.api` (*module*), 218
`tardis.tardis_portal.auth` (*module*), 130
`tardis.tardis_portal.auth.authentication` (*module*), 124
`tardis.tardis_portal.auth.authorisation` (*module*), 125
`tardis.tardis_portal.auth.authservice` (*module*), 125
`tardis.tardis_portal.auth.decorators` (*module*), 126
`tardis.tardis_portal.auth.fix_circular` (*module*), 128
`tardis.tardis_portal.auth.interfaces` (*module*), 128
`tardis.tardis_portal.auth.localdb_auth` (*module*), 129
`tardis.tardis_portal.auth.token_auth` (*module*), 130
`tardis.tardis_portal.auth.utils` (*module*), 130
`tardis.tardis_portal.context_processors` (*module*), 228
`tardis.tardis_portal.deprecations` (*module*), 228
`tardis.tardis_portal.download` (*module*), 229
`tardis.tardis_portal.email` (*module*), 230
`tardis.tardis_portal.fields` (*module*), 230
`tardis.tardis_portal.forms` (*module*), 230
`tardis.tardis_portal.iiif` (*module*), 236
`tardis.tardis_portal.logging_middleware` (*module*), 236
`tardis.tardis_portal.management` (*module*), 132
`tardis.tardis_portal.management.commands` (*module*), 132
`tardis.tardis_portal.management.commands.createuser` (*module*), 131
`tardis.tardis_portal.management.commands.dumpschema` (*module*), 131

[tardis.tardis_portal.management.commands.addschmandis_portal.models.experiment \(module\), 131](#)
[tardis.tardis_portal.management.commands.addexperimentis_portal.models.facility \(module\), 132](#)
[tardis.tardis_portal.managers \(module\), 236](#)
[tardis.tardis_portal.migrations \(module\), 136](#)
[tardis.tardis_portal.migrations.0001_initiandis_portal.models.jti \(module\), \(module\), 132](#)
[tardis.tardis_portal.migrations.0001_squashedis_portal.models.parameters \(module\), 132](#)
[tardis.tardis_portal.migrations.0002_autha20150528d1328portal.models.parameters \(module\), 133](#)
[tardis.tardis_portal.migrations.0003_autha20150907d1315portal.models.storage \(module\), 133](#)
[tardis.tardis_portal.migrations.0004_storage_externdis_portal.models.token \(module\), 133](#)
[tardis.tardis_portal.migrations.0005_datafiledis_portal.models.token \(module\), 133](#)
[tardis.tardis_portal.migrations.0006_datafiledis_portal.models.token \(module\), 133](#)
[tardis.tardis_portal.migrations.0007_remove_paramdis_portal.models.token \(module\), 134](#)
[tardis.tardis_portal.migrations.0008_string_valdis_portal.models.token \(module\), 134](#)
[tardis.tardis_portal.migrations.0009_auto_20160110dis_portal.models.token \(module\), 134](#)
[tardis.tardis_portal.migrations.0010_autha20160503d1443portal.shortcuts \(module\), \(module\), 134](#)
[tardis.tardis_portal.migrations.0011_autha20160505d1643portal.storage \(module\), 134](#)
[tardis.tardis_portal.migrations.0012_userautherdis_portal.storage \(module\), 135](#)
[tardis.tardis_portal.migrations.0013_autha20181002d1336portal.templatetags \(module\), 135](#)
[tardis.tardis_portal.migrations.0014_autha20181002d1354portal.templatetags.approved_user_tag \(module\), 135](#)
[tardis.tardis_portal.migrations.0015_datafiledis_portal.templatetags.approved_user_tag \(module\), 135](#)
[tardis.tardis_portal.migrations.0016_addtimedis_portal.templatetags.approved_user_tag \(module\), 135](#)
[tardis.tardis_portal.migrations.0017_addtimedis_portal.templatetags.approved_user_tag \(module\), 135](#)
[tardis.tardis_portal.migrations.0018_makedis_portal.templatetags.approved_user_tag \(module\), 136](#)
[tardis.tardis_portal.models \(module\), 182](#)
[tardis.tardis_portal.models.access_contrbdis_portal.templatetags.approved_user_tag \(module\), 136](#)
[tardis.tardis_portal.models.datafile \(module\), 141](#)
[tardis.tardis_portal.models.dataset \(module\), 147](#)

(module), 204
 tardis.tardis_portal.tests.test_tar_download (module), 205
 tardis.tardis_portal.tests.test_tasks (module), 205
 tardis.tardis_portal.tests.test_tokens (module), 205
 tardis.tardis_portal.tests.tests (module), 206
 tardis.tardis_portal.tests.views (module), 196
 tardis.tardis_portal.tests.views.test_authentication (module), 194
 tardis.tardis_portal.tests.views.test_content_disposition (module), 195
 tardis.tardis_portal.tests.views.test_experiment_download (module), 195
 tardis.tardis_portal.tests.views.test_template_texts (module), 195
 tardis.tardis_portal.tests.views.test_upload_vim (module), 196
 tardis.tardis_portal.util (module), 240
 tardis.tardis_portal.views (module), 212, 242, 244
 tardis.tardis_portal.views.ajax_actions (module), 206
 tardis.tardis_portal.views.ajax_json (module), 206
 tardis.tardis_portal.views.ajax_pages (module), 207
 tardis.tardis_portal.views.authentication (module), 207
 tardis.tardis_portal.views.authorization (module), 207
 tardis.tardis_portal.views.facilities (module), 208
 tardis.tardis_portal.views.images (module), 208
 tardis.tardis_portal.views.machine (module), 208
 tardis.tardis_portal.views.pages (module), 208
 tardis.tardis_portal.views.parameters (module), 211
 tardis.tardis_portal.views.upload (module), 212
 tardis.tardis_portal.views.utils (module), 212
 tardis.tardis_portal.widgets (module), 240
 tardis.tardis_portal.xmlwriter (module), 241
 tardis.test_on_mysql_settings (module), 244
 tardis.test_on_postgresql_settings (module), 244
 tardis.test_settings (module), 244
 tardis.urls.accounts (module), 242
 tardis.urls.ajax (module), 242
 tardis.urls.api (module), 242
 tardis.urls.core (module), 242
 tardis.urls.datafile (module), 243
 tardis.urls.dataset (module), 243
 tardis.urls.display (module), 243
 tardis.urls.download (module), 243
 tardis.urls.experiment (module), 243
 tardis.urls.facility (module), 243
 tardis.urls.group (module), 243
 tardis.urls.historical_views (module), 243
 tardis.views (module), 244
 TardisDownloadTestCase (class in `tardis.tardis_portal.tests.test_tar_download`), 205
 tarinfo_for_df () (`tardis.tardis_portal.download.UncachedTarStream` method), 229
 tearDown () (`tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTest` method), 80
 tearDown () (`tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentTest` method), 81
 tearDown () (`tardis.apps.oaipmh.tests.provider.test_experiment.RifCsExperimentTest` method), 81
 tearDown () (`tardis.apps.oaipmh.tests.test_oai.EndpointTestCase` method), 81
 tearDown () (`tardis.apps.s3utils.tests.test_api.S3UtilsAppApiTestCase` method), 100
 tearDown () (`tardis.apps.s3utils.tests.test_checksums.S3UtilsAppChecksumsTestCase` method), 101
 tearDown () (`tardis.tardis_portal.tests.api.test_dataset_metadata_resource.DatasetMetadataResourceTest` method), 187
 tearDown () (`tardis.tardis_portal.tests.api.test_schema_resource.SchemaResourceTest` method), 190
 tearDown () (`tardis.tardis_portal.tests.management.test_collectstatic.CollectStaticTest` method), 191
 tearDown () (`tardis.tardis_portal.tests.management.test_dumpschemas.DumpSchemasTest` method), 192
 tearDown () (`tardis.tardis_portal.tests.test_authorisation.ObjectACLTest` method), 198
 tearDown () (`tardis.tardis_portal.tests.test_authservice.AuthServiceTest` method), 199
 tearDown () (`tardis.tardis_portal.tests.test_copy_move.CopyMoveTest` method), 199
 tearDown () (`tardis.tardis_portal.tests.test_df_save_metadata.DatafileSaveMetadataTest` method), 200
 tearDown () (`tardis.tardis_portal.tests.test_download.DownloadTestCase` method), 200
 tearDown () (`tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownloadTest` method), 200
 tearDown () (`tardis.tardis_portal.tests.test_ldap.LDAPTest` method), 202

test_dataset_view() (tardis.tardis_portal.tests.views.test_template_contexts.ViewTemplateContextsTest method), 196	test_experiment_view() (tardis.tardis_portal.tests.views.test_template_contexts.ViewTemplateContextsTest method), 196
test_debug_serializer() (tardis.tardis_portal.tests.api.test_serializer.SerializerTest method), 190	test_facility_overview_datafile_list() (tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTest method), 201
test_default_expiry() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 205	test_facility_overview_experiments() (tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTest method), 201
test_deleting_dfo_without_uri() (tardis.tardis_portal.tests.models.test_dfo.DataFileObjectTest method), 193	test_file_upload() (tardis.tardis_portal.tests.views.test_upload_views.UploadTestCases method), 196
test_details_display() (tardis.tardis_portal.tests.views.test_contextual_views.ContextualViewsTest method), 195	test_get_allowed_signing_services() (tardis.tardis_portal.tests.test_models.ModelsTestCase method), 90
test_df_save_metadata() (tardis.tardis_portal.tests.test_df_save_metadata.DatafileSaveMetadataTest method), 200	test_get_child_dir_nodes() (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest method), 188
test_djangoauth() (tardis.tardis_portal.tests.test_authentication.AuthenticationTest method), 198	test_get_child_dir_nodes_no_files_in_root_dir() (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest method), 188
test_do_migration() (tardis.apps.openid_migration.tests.test_migration.OpenIDMigrationTest method), 85	test_get_dataset_files() (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest method), 188
test_download_apikey() (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownloadTest method), 200	test_get_dataset_filter_instrument() (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest method), 188
test_download_dfo() (tardis.apps.s3utils.tests.test_api.S3UtilsAppApiTestCase method), 101	test_get_dataset_no_instrument() (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest method), 188
test_download_file() (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResourceTest method), 187	test_get_dataset_with_instrument() (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest method), 188
test_edit_datafile_params() (tardis.tardis_portal.tests.test_parametersets.EditParametersTest method), 203	test_get_dir_tuples() (tardis.tardis_portal.tests.models.test_dataset.DatasetTestCase method), 193
test_edit_dataset_params() (tardis.tardis_portal.tests.test_parametersets.EditParametersTest method), 203	test_get_experiment() (tardis.tardis_portal.tests.api.test_experiment_resource.ExperimentResourceTest method), 188
test_edit_experiment_params() (tardis.tardis_portal.tests.test_parametersets.EditParametersTest method), 203	test_get_experiment_author() (tardis.tardis_portal.tests.api.test_experiment_resource.ExperimentResourceTest method), 188
test_ensures_suitable_license() (tardis.tardis_portal.tests.test_forms.RightsFormTestCase method), 201	test_get_facility_by_id() (tardis.tardis_portal.tests.api.test_facility_resource.FacilityResourceTest method), 188
test_existing_parameterset() (tardis.tardis_portal.tests.test_parametersets.ParameterSetTest method), 203	test_get_facility_by_manager_group_id() (tardis.tardis_portal.tests.api.test_facility_resource.FacilityResourceTest method), 188
test_experiment() (tardis.tardis_portal.tests.models.test_experiment.ExperimentTest method), 193	test_get_facility_by_name() (tardis.tardis_portal.tests.api.test_facility_resource.FacilityResourceTest method), 189
test_experiment_parameter_get() (tardis.tardis_portal.tests.test_parameters.ParametersTest method), 203	test_get_group_by_id() (tardis.tardis_portal.tests.api.test_group_resource.GroupResourceTest method), 189

test_get_group_by_name()	test_get_user_list()
(tardis.tardis_portal.tests.api.test_group_resource.GroupResourceTest method), 189	(tardis.tardis_portal.tests.views.test_auth_views.UserListTestCase method), 195
test_get_group_list()	test_getgroupbyid()
(tardis.tardis_portal.tests.views.test_auth_views.UserGroupListTestCase method), 194	(tardis.tardis_portal.tests.test_ldap.LDAPTest method), 202
test_get_instrument_by_id()	test_getgroups()
(tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest method), 189	(tardis.tardis_portal.tests.test_ldap.LDAPTest method), 202
test_get_instrument_by_name()	test_getgroupsforentity()
(tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest method), 189	(tardis.tardis_portal.tests.test_ldap.LDAPTest method), 202
test_get_receiving_box()	test_getuserid()
(tardis.tardis_portal.tests.test_storage.ModelTestCase method), 205	(tardis.tardis_portal.tests.test_ldap.LDAPTest method), 202
test_get_root_dir_nodes()	test_image_filename_parameters()
(tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest method), 188	(tardis.tardis_portal.tests.test_parameters.ParametersTestCase method), 193
test_get_schema_by_id()	test_init()
(tardis.tardis_portal.tests.api.test_schema_resource.SchemaResourceTest method), 190	(tardis.apps.openid_migration.tests.test_forms.OpenIDMigrationTest method), 84
test_get_schema_by_namespace()	test_instrument()
(tardis.tardis_portal.tests.api.test_schema_resource.SchemaResourceTest method), 190	(tardis.tardis_portal.tests.models.test_instrument.InstrumentTestCase method), 194
test_get_session_expiry()	test_instrument_resource()
(tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 205	(tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 206
test_get_session_expiry_expired_token()	test_keys_from_private_key_only()
(tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 205	(tardis.apps.push_to.tests.test_models.ModelsTestCase method), 90
test_get_session_expiry_near_expiry()	test_link_parameter_type()
(tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 206	(tardis.tardis_portal.tests.test_parametersets.ParameterSetManager method), 203
test_get_storage_box_attr_list_from_box_id()	test_link_parameter_type_extra()
(tardis.tardis_portal.tests.api.test_storagebox_resource.StorageBoxAttributeTestCase method), 190	(tardis.tardis_portal.tests.test_parametersets.ParameterSetManager method), 204
test_get_storage_box_by_id()	test_list_endpoints()
(tardis.tardis_portal.tests.api.test_storagebox_resource.StorageBoxResourceTest method), 190	(tardis.tardis_portal.tests.test_parameters.ParametersTestCase method), 203
test_get_storage_box_option_by_id()	test_login()
(tardis.tardis_portal.tests.api.test_storagebox_resource.StorageBoxOptionResourceTest method), 190	(tardis.tardis_portal.tests.tests.UserInterfaceTestCase method), 206
test_get_storage_box_option_list_from_box_id()	test_manage_account()
(tardis.tardis_portal.tests.api.test_storagebox_resource.StorageBoxOptionResourceTest method), 190	(tardis.tardis_portal.tests.views.test_auth_views.ManageAccountTestCase method), 194
test_get_user_by_id()	test_migrate_accounts()
(tardis.tardis_portal.tests.api.test_user_resource.UserResourceTest method), 190	(tardis.apps.openid_migration.tests.test_views.OpenIDMigrationTest method), 85
test_get_user_by_username()	test_move()
(tardis.tardis_portal.tests.api.test_user_resource.UserResourceTest method), 190	(tardis.tardis_portal.tests.test_copy_move.CopyMoveTestCase method), 200
test_get_user_list()	test_new_view()
(tardis.tardis_portal.tests.views.test_auth_views.UserGroupListTestCase method), 195	(tardis.tardis_portal.tests.views.test_template_contexts.ExperimentTemplateContextTestCase method), 196
	test_parameterset()
	(tardis.tardis_portal.tests.test_parametersets.ParameterSetManager method), 203

method), 204
 test_OpenidUserMigration_creation() (tardis.apps.openid_migration.tests.test_models.ModelTestCase method), 85
 test_parameter() (tardis.tardis_portal.tests.models.test_parameters.ParametersTestCase method), 194
 test_parameterset_as_string() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase method), 204
 test_permissions_checks() (tardis.tardis_portal.tests.test_parameters.ParametersTestCase method), 203
 test_pkey_to_credential() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 90
 test_post_dataset() (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTestCase method), 188
 test_post_dataset_with_params() (tardis.tardis_portal.tests.api.test_dataset_metadata_resource.DatasetMetadataParameterSetResourceTest method), 187
 test_post_experiment() (tardis.tardis_portal.tests.api.test_experiment_resource.ExperimentResourceTest method), 188
 test_post_instrument() (tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest method), 189
 test_post_single_file() (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResourceTestCase method), 187
 test_pretty_serializer() (tardis.tardis_portal.tests.api.test_serializer.SerializerTest method), 190
 test_rename_instrument() (tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest method), 189
 test_retrieve_access_list_tokens() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 206
 test_rights_require_valid_owner() (tardis.tardis_portal.tests.views.test_auth_views.RightsTestCase method), 194
 test_root() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase method), 206
 test_save_with_random_token() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 206
 test_save_with_random_token_failures() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 206
 test_save_with_random_token_gives_up() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 206
 test_search() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 202
 test_search() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 202
 test_searchgroups() (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 202
 test_sftp() (tardis.apps.sftp.tests.test_sftp.SFTPTest method), 108
 test_sftp_dynamic_docs_dataset() (tardis.apps.sftp.tests.test_sftp.SFTPTest method), 108
 test_sftp_dynamic_docs_experiment() (tardis.apps.sftp.tests.test_sftp.SFTPTest method), 108
 test_sftp_key_connect() (tardis.apps.sftp.tests.test_sftp.SFTPTest method), 108
 test_shared_fs_many_files() (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResourceTestCase method), 187
 test_shared_fs_single_file() (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResourceTestCase method), 187
 test_shared_view() (tardis.tardis_portal.tests.views.test_template_contexts.ExperimentResourceTest method), 190
 test_simple_search_authenticated_user() (tardis.apps.search.tests.test_api.SimpleSearchTest method), 103
 test_simple_search_unauthenticated_user() (tardis.apps.search.tests.test_api.SimpleSearchTest method), 103
 test_storageboxoption() (tardis.tardis_portal.tests.test_storage.ModelTestCase method), 206
 test_tar_experiment_download() (tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase method), 205
 test_token_delete() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 206
 test_tz_aware_date_handling() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase method), 204
 test_tz_naive_date_handling() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase method), 204
 test_unauthorized_instrument_access_attempt() (tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest method), 189
 test_unresolvable_link_parameter() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase method), 204
 test_upload_complete() (tardis.tardis_portal.tests.test_upload.UploadCompleteTest method), 190

`(tardis.tardis_portal.tests.views.test_upload_views.UploadTestCases.BadInput ()`
`method), 196` `(tardis.apps.related_info.tests.tests.UpdateTestCase`
`test_urls () (tardis.tardis_portal.tests.tests.UserInterfaceTestCase.method), 99`
`method), 206` `testDownload () (tardis.tardis_portal.tests.test_download.DownloadTe`
`test_urls_with_some_content ()` `method), 200`
`(tardis.tardis_portal.tests.tests.UserInterfaceTestCase`
`method), 206` `testDumpSchemas ()`
`TEST_UTILS_DIR (tardis.tardis_portal.tests.slapi.Slapd` `method), 192`
`attribute), 197` `testExistsInOaipmh ()`
`test_wrong_size_verification ()` `(tardis.apps.related_info.tests.test_oaipmh.RifCTestCase`
`(tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase.method), 98`
`method), 205` `testGetGroupsForEntity ()`
`testAccessWithoutReadPerms ()` `(tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase`
`(tardis.apps.related_info.tests.tests.TabTestCase` `method), 199`
`method), 99` `testGetRecord () (tardis.apps.oaipmh.tests.provider.test_base.BasePro`
`testAccessWithReadPerms ()` `method), 80`
`(tardis.apps.related_info.tests.tests.TabTestCase` `testGetRecord () (tardis.apps.oaipmh.tests.provider.test_experiment.D`
`method), 99` `method), 81`
`testAuthenticate ()` `testGetRecord () (tardis.apps.oaipmh.tests.provider.test_experiment.R`
`(tardis.tardis_portal.tests.test_authservice.AuthServiceTestCases.method), 81`
`method), 199` `testGetRecord () (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase`
`testCanCreate () (tardis.apps.related_info.tests.tests.CreateTestCase.method), 82`
`method), 99` `testGetRecordHandlesInvalidIdentifiers ()`
`testCanDelete () (tardis.apps.related_info.tests.tests.DeleteTestCase` `(tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExper`
`method), 99` `method), 81`
`testCanGetInfoAsJSON ()` `testGroupProvider ()`
`(tardis.tardis_portal.tests.test_iiif.Level0TestCase` `(tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase`
`method), 201` `method), 199`
`testCanGetInfoAsXML ()` `testGroupSearch ()`
`(tardis.tardis_portal.tests.test_iiif.Level0TestCase` `(tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase`
`method), 201` `method), 199`
`testCanGetJpegFormat ()` `testHandleRegions ()`
`(tardis.tardis_portal.tests.test_iiif.Level1TestCase` `(tardis.tardis_portal.tests.test_iiif.Level1TestCase`
`method), 202` `method), 202`
`testCanGetOriginalImage ()` `testHandleRotation ()`
`(tardis.tardis_portal.tests.test_iiif.Level0TestCase` `(tardis.tardis_portal.tests.test_iiif.Level1TestCase`
`method), 201` `method), 202`
`testCanGetRequiredFormats ()` `testHandlesEmptySet ()`
`(tardis.tardis_portal.tests.test_iiif.Level2TestCase` `(tardis.apps.related_info.tests.tests.ListTestCase`
`method), 202` `method), 99`
`testCantEditLockedExperiment ()` `testHandlesFound ()`
`(tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCases` `(tardis.apps.related_info.tests.tests.GetTestCase`
`method), 198` `method), 99`
`testChangeUserPermissions ()` `testHandleSizing ()`
`(tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCases` `(tardis.tardis_portal.tests.test_iiif.Level1TestCase`
`method), 198` `method), 202`
`testContext () (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase`
`method), 204` `(tardis.tardis_portal.tests.test_iiif.Level2TestCase`
`testDatasetFile ()` `method), 202`
`(tardis.tardis_portal.tests.test_download.DownloadTestCases`
`method), 200` `testHandlesMultipleEntries ()`
`(tardis.apps.related_info.tests.tests.ListTestCase`
`testDetectsBadInput ()` `method), 99`
`(tardis.apps.related_info.tests.tests.CreateTestCase`
`method), 99` `testHandlesNotFound ()`
`(tardis.apps.related_info.tests.tests.GetTestCase`

`method)`, 99
`testHandlesSingleEntry()` (`tardis.apps.related_info.tests.tests.ListTestCase` `method)`, 99
`testIdentify()` (`tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase` `method)`, 80
`testIdentify()` (`tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTC` `method)`, 81
`testIdentify()` (`tardis.apps.oaipmh.tests.test_oai.EndpointTestCase` `method)`, 82
`testImageCacheControl()` (`tardis.tardis_portal.tests.test_iiif.ExtraTestCases` `method)`, 201
`testImageHasEtags()` (`tardis.tardis_portal.tests.test_iiif.ExtraTestCases` `method)`, 201
`testInfoHasEtags()` (`tardis.tardis_portal.tests.test_iiif.ExtraTestCases` `method)`, 201
`testInitialisation()` (`tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase` `method)`, 199
`testInitialisation()` (`tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase` `method)`, 204
`testInitialisationNoProvider()` (`tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase` `method)`, 204
`testInteractive()` (`tardis.tardis_portal.tests.management.test_createuser.CreateUserTestCase` `method)`, 191
`testList()` (`tardis.tardis_portal.tests.management.test_rmexperiment.RmExperimentTestCase` `method)`, 192
`testListIdentifiers()` (`tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase` `method)`, 80
`testListIdentifiers()` (`tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTC` `method)`, 81
`testListIdentifiers()` (`tardis.apps.oaipmh.tests.test_oai.EndpointTestCase` `method)`, 82
`testListIdentifiersDoesNotHandleSets()` (`tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTC` `method)`, 81
`testListMetadataFormats()` (`tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase` `method)`, 80
`testListMetadataFormats()` (`tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTC` `method)`, 81
`testListMetadataFormats()` (`tardis.apps.oaipmh.tests.test_oai.EndpointTestCase` `method)`, 82
`testListRecords()` (`tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase` `method)`, 80
`testListRecords()` (`tardis.apps.oaipmh.tests.provider.test_experiment.DcExperimentProviderTC` `method)`, 81
`testListRecords()` (`tardis.apps.oaipmh.tests.provider.test_experiment.RifCsExperimentProviderTC` `method)`, 81
`testListRecords()` (`tardis.apps.oaipmh.tests.test_oai.EndpointTestCase` `method)`, 82
`testListSets()` (`tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase` `method)`, 81
`testListSets()` (`tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTC` `method)`, 81
`testLoadSchemas()` (`tardis.tardis_portal.tests.management.test_loadschemas.LoadSchemasTestCase` `method)`, 192
`testLocalFile()` (`tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase` `method)`, 205
`testManageAuthMethods()` (`tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase` `method)`, 198
`testManageRifCsCheckContent()` (`tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase` `method)`, 204
`testManageRifCsCreateAndRemove()` (`tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase` `method)`, 204
`testMustHaveWrite()` (`tardis.tardis_portal.tests.management.test_createuser.CreateUserTestCase` `method)`, 191
`testMustHaveWrite()` (`tardis.tardis_portal.tests.management.test_rmexperiment.RmExperimentTestCase` `method)`, 192
`testMustHaveWrite()` (`tardis.tardis_portal.tests.management.test_createuser.CreateUserTestCase` `method)`, 191
`testMustHaveWrite()` (`tardis.tardis_portal.tests.management.test_rmexperiment.RmExperimentTestCase` `method)`, 192
`testMustHaveWrite()` (`tardis.tardis_portal.tests.management.test_createuser.CreateUserTestCase` `method)`, 191
`testMustHaveWrite()` (`tardis.tardis_portal.tests.management.test_rmexperiment.RmExperimentTestCase` `method)`, 192
`testMustHaveWrite()` (`tardis.tardis_portal.tests.management.test_createuser.CreateUserTestCase` `method)`, 191
`testMustHaveWrite()` (`tardis.tardis_portal.tests.management.test_rmexperiment.RmExperimentTestCase` `method)`, 192
`testNoInput()` (`tardis.tardis_portal.tests.management.test_createuser.CreateUserTestCase` `method)`, 192
`testOwnedExperiments()` (`tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase` `method)`, 193
`testReadAccess()` (`tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase` `method)`, 198
`testReadAccess()` (`tardis.tardis_portal.tests.management.test_rmexperiment.RmExperimentTestCase` `method)`, 192
`testSFTPDWithoutHostKey()` (`tardis.tardis_portal.tests.test_sftp.SFTPDManagementTestCase` `method)`, 108
`testSimpleAuthenticate()` (`tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase` `method)`, 198

[testView\(\)](#) ([tardis.tardis_portal.tests.test_download.DownloadTestCase](#) value ([tardis.tardis_portal.models.storage.StorageBoxOptions](#) attribute), 200
[testWriteAccess\(\)](#) ([tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase](#) value ([tardis.apps.sftp.sftp.DynamicTree](#) attribute), 181
[timeout](#) ([tardis.apps.sftp.sftp.MyTSFTPRequestHandler](#) attribute), 112
[title](#) ([tardis.tardis_portal.models.experiment.Experiment](#) attribute), 154
[to_json\(\)](#) ([tardis.apps.search.api.PrettyJSONSerializer](#) attribute), 104
[to_json\(\)](#) ([tardis.tardis_portal.api.PrettyJSONSerializer](#) attribute), 226
[todatetime\(\)](#) ([tardis.tardis_portal.templatetags.feed](#) module), 185
[Token](#) (class in [tardis.tardis_portal.models.token](#)), 181
[token](#) ([tardis.tardis_portal.models.token.Token](#) attribute), 182
[Token.DoesNotExist](#), 181
[Token.MultipleObjectsReturned](#), 181
[token_delete\(\)](#) ([tardis.tardis_portal.views.authorisation](#) module), 207
[token_set](#) ([tardis.tardis_portal.models.experiment.Experiment](#) attribute), 154
[TokenAuthMiddleware](#) (class in [tardis.tardis_portal.auth.token_auth](#)), 130
[TokenGroupProvider](#) (class in [tardis.tardis_portal.auth.token_auth](#)), 130
[TokenTestCase](#) (class in [tardis.tardis_portal.tests.test_tokens](#)), 205
[toxmldatetime\(\)](#) ([tardis.tardis_portal.templatetags.xmldate](#) module), 186
[track_download\(\)](#) ([tardis.analytics.ga](#) module), 75
[track_login\(\)](#) ([tardis.analytics.ga](#) module), 75
[track_logout\(\)](#) ([tardis.analytics.ga](#) module), 75
[type](#) ([tardis.tardis_portal.models.parameters.Schema](#) attribute), 176
[TYPE_CHOICES](#) ([tardis.tardis_portal.models.storage.StorageBoxOptions](#) attribute), 181
[type_order](#) ([tardis.tardis_portal.models.storage.StorageBoxOptions](#) attribute), 180
[TYPE_UNKNOWN](#) ([tardis.tardis_portal.models.storage.StorageBoxOptions](#) attribute), 177
[TYPES](#) ([tardis.tardis_portal.models.storage.StorageBoxOptions](#) attribute), 176

U
[UncachedTarStream](#) (class in [tardis.tardis_portal.download](#)), 229
[units](#) ([tardis.tardis_portal.models.parameters.ParameterName](#) attribute), 173

[update_all_files\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), 112
[update_dataset_files\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), 112
[update_datasets\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), 112
[update_detail\(\)](#) ([tardis.tardis_portal.api.ACLAuthorization](#) method), 219
[update_experiments\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), 112
[update_list\(\)](#) ([tardis.tardis_portal.api.ACLAuthorization](#) method), 219
[update_mimetype\(\)](#) ([tardis.tardis_portal.models.datafile.DataFile](#) method), 144
[update_nothing\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), 112
[update_time](#) ([tardis.tardis_portal.models.experiment.Experiment](#) attribute), 154
[UpdateTestCase](#) (class in [tardis.apps.related_info.tests.tests](#)), 99
[upload_auth\(\)](#) ([tardis.tardis_portal.auth.decorators](#) module), 128
[upload_complete\(\)](#) ([tardis.tardis_portal.views.upload](#) module), 212
[UPLOAD_METHOD](#) ([tardis.default_settings.uploads](#) module), 124
[UploadTestCase](#) (class in [tardis.tardis_portal.tests.views.test_upload_views](#)), 196
[uri](#) ([tardis.tardis_portal.models.datafile.DataFileObject](#) attribute), 146
[url](#) ([tardis.tardis_portal.models.experiment.Experiment](#) attribute), 154
[url_option](#) ([tardis.tardis_portal.models.experiment.ExperimentAuthor](#) attribute), 155
[url_rapid_connect\(\)](#) ([tardis.tardis_portal.models.license.License](#) attribute), 159
[url_rapid_connect\(\)](#) ([tardis.tardis_portal.models.parameters.ParameterName](#) attribute), 171
[urls](#) ([tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase](#) attribute), 198
[urls](#) ([tardis.tardis_portal.tests.test_tokens.TokenTestCase](#) attribute), 206
[USE_FILTERS](#) ([tardis.default_settings.filters](#) module), 122
[use_rapid_connect\(\)](#) ([tardis.tardis_portal.views.pages](#) module), 211

[user \(tardis.apps.push_to.models.Credential attribute\), 92](#)
[user \(tardis.apps.sftp.models.SFTPPublicKey attribute\), 111](#)
[user \(tardis.tardis_portal.models.access_control.GroupAdmin attribute\), 137](#)
[user \(tardis.tardis_portal.models.access_control.UserProfile attribute\), 141](#)
[user \(tardis.tardis_portal.models.token.Token attribute\), 182](#)
[user_acls\(\) \(tardis.tardis_portal.managers.ExperimentManager method\), 238](#)
[user_details_processor\(\) \(in module tardis.tardis_portal.context_processors\), 228](#)
[user_guide\(\) \(in module tardis.tardis_portal.views.pages\), 211](#)
[user_id \(tardis.apps.push_to.models.Credential attribute\), 92](#)
[user_id \(tardis.apps.sftp.models.SFTPPublicKey attribute\), 111](#)
[user_id \(tardis.tardis_portal.models.access_control.GroupAdmin attribute\), 137](#)
[user_id \(tardis.tardis_portal.models.access_control.UserProfile attribute\), 141](#)
[user_id \(tardis.tardis_portal.models.token.Token attribute\), 182](#)
[USER_MENU_MODIFIERS \(in module tardis.default_settings.apps\), 119](#)
[user_menu_processor\(\) \(in module tardis.tardis_portal.context_processors\), 228](#)
[user_migration \(tardis.apps.openid_migration.models.OpenidACLMigration attribute\), 87](#)
[user_migration_id \(tardis.apps.openid_migration.models.OpenidACLMigration attribute\), 87](#)
[user_migration_obj\(\) \(tardis.apps.openid_migration.models.OpenidACLMigrationAdmin method\), 87](#)
[user_owned_groups\(\) \(tardis.tardis_portal.managers.ExperimentManager method\), 238](#)
[UserAuthentication \(class in tardis.tardis_portal.models.access_control\), 139](#)
[UserAuthentication.DoesNotExist, 139](#)
[UserAuthentication.MultipleObjectsReturned, 139](#)
[userauthentication_set \(tardis.tardis_portal.models.access_control.UserProfile attribute\), 141](#)
[UserAuthenticationAdmin \(class in tardis.tardis_portal.admin\), 218](#)
[UserGroupListsTestCase \(class in tardis.tardis_portal.tests.views.test_auth_views\), 194](#)
[UserInterfaceTestCase \(class in tardis.tardis_portal.tests.tests\), 206](#)
[UserListTestCase \(class in tardis.tardis_portal.tests.views.test_auth_views\), 195](#)
[username \(tardis.tardis_portal.models.access_control.UserAuthentication attribute\), 140](#)
[UserProfile \(class in tardis.tardis_portal.models.access_control\), 140](#)
[UserProfile.DoesNotExist, 140](#)
[UserProfile.MultipleObjectsReturned, 140](#)
[UserProfile_id \(tardis.tardis_portal.models.access_control.UserAuthentication attribute\), 140](#)
[UserProvider \(class in tardis.tardis_portal.auth.interfaces\), 128](#)
[UserResource \(class in tardis.tardis_portal.api\), 227](#)
[UserResource.Meta \(class in tardis.tardis_portal.api\), 227](#)
[UserResourceTest \(class in tardis.tardis_portal.tests.api.test_user_resource\), 190](#)
[users\(\) \(tardis.tardis_portal.managers.ExperimentManager method\), 238](#)
V
[validate_remote_path\(\) \(in module tardis.apps.push_to.views\), 98](#)
[validation \(tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta attribute\), 109](#)
[value \(tardis.tardis_portal.models.storage.StorageBoxAttribute attribute\), 180](#)
[value \(tardis.tardis_portal.models.storage.StorageBoxOption attribute\), 181](#)
[value_from_datadict\(\) \(tardis.tardis_portal.widgets.CommaSeparatedInput method\), 241](#)
[value_type \(tardis.tardis_portal.models.storage.StorageBoxOption attribute\), 181](#)
[verbose_name \(tardis.analytics.apps.AnalyticsConfig attribute\), 75](#)
[verbose_name \(tardis.apps.openid_migration.apps.OpenidMigrationConfig attribute\), 85](#)
[verbose_name \(tardis.apps.push_to.apps.PushToConfig attribute\), 91](#)
[verbose_name \(tardis.apps.search.apps.SearchConfig attribute\), 105](#)
[verbose_name \(tardis.apps.sftp.apps.SFTPConfig attribute\), 110](#)

`verbose_name` (`tardis.apps.social_auth.apps.SocialAuthConfig` attribute), 118
`verified` (`tardis.tardis_portal.models.datafile.DataFile` attribute), 144
`verified` (`tardis.tardis_portal.models.datafile.DataFileObject` attribute), 146
`verify` () (`tardis.tardis_portal.models.datafile.DataFile` method), 144
`verify` () (`tardis.tardis_portal.models.datafile.DataFileObject` method), 146
`verify_file` () (`tardis.tardis_portal.api.DataFileResource` method), 220
`verify_remote_access` () (`tardis.apps.push_to.models.Credential` method), 92
`version` (`tardis.tardis_portal.models.datafile.DataFile` attribute), 144
`view_datafile` () (in module `tardis.tardis_portal.download`), 230
`view_functions` (`tardis.tardis_portal.shortcuts.RestfulExperimentProvider` attribute), 239
`ViewTemplateContextsTest` (class in `tardis.tardis_portal.tests.views.test_template_contexts`), 196
W
`wait` () (`tardis.tardis_portal.tests.slapped.Slapped` method), 198
`which` () (in module `tardis.tardis_portal.tests.slapped`), 198
`widget` (`tardis.tardis_portal.forms.StaticField` attribute), 235
`widgets` (`tardis.apps.push_to.models.CredentialForm.Meta` attribute), 92
`widgets` (`tardis.tardis_portal.admin.DatafileAdminForm.Meta` attribute), 214
`widgets` (`tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta` attribute), 214
`widgets` (`tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta` attribute), 217
`widgets` (`tardis.tardis_portal.admin.StorageBoxForm.Meta` attribute), 217
`widgets` (`tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta` attribute), 218
`widgets` (`tardis.tardis_portal.forms.RightsForm.Meta` attribute), 235
`write` () (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider.ExperimentWriter` method), 80
`write_permissions_required` () (in module `tardis.tardis_portal.auth.decorators`), 128
`write_template_to_dir` () (`tardis.tardis_portal.xmlwriter.XMLWriter` static method), 241
`write_template_to_file` () (`tardis.tardis_portal.xmlwriter.XMLWriter` static method), 241
`write_xml_to_file` () (`tardis.tardis_portal.xmlwriter.XMLWriter` static method), 241
`writeExperimentMetadata` () (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider` static method), 80
`writeMetadata` () (`tardis.apps.oaipmh.provider.base.BaseProvider` method), 79, 249
`writeMetadata` () (`tardis.apps.oaipmh.server.ProxyingMetadataRegistry` method), 82
`writeRegistryObjectsWrapper` () (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider` method), 80
`writeRelatedAuthor` () (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider` method), 80
`writeRelatedInfo` () (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider` method), 80
`writeRelatedUser` () (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider` method), 80
`writeSubject` () (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider` method), 80
`writeUserMetadata` () (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider` static method), 80
X
`XMLWriter` (class in `tardis.tardis_portal.xmlwriter`), 241