
MyTardis Documentation

Release 4.6.0

Jul 11, 2022

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	54
11.3 Development	77
11.4 Documentation for included Apps	295
11.5 Releases	306
12 Indices and tables	317
Python Module Index	319
Index	325

CHAPTER 1

Overview

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

[Read more...](#)

CHAPTER 2

Key features for users

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

[Read more...](#)

CHAPTER 3

Key features for instrument facilities

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

[Read more...](#)

CHAPTER 4

Developing for MyTardis

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

To set up and manage these services we employ the [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/>

CHAPTER 7

Related projects and repositories

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

CHAPTER 8

Releases

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

The `master` branch is the current stable release with all the latest 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 `4.6.0`, tagged from the `series-4.6` 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. `4.6.1`. 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](#).

CHAPTER 11

Documentation

11.1 User Guide

11.1.1 Basics

Organising Data

Data Structure Overview

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

Creating Experiments

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

The screenshot shows the 'My Data' page with the 'Experiments' tab selected. At the top right, there is a yellow button labeled '+ Create'. Below it are two green buttons: 'Collapse all' and 'Expand all'. A blue button labeled 'yesterday' shows statistics: 1 experiment, 0 datasets, and 0 data files. To the right of these buttons is a link 'Download data as .tar'. The main content area displays one experiment titled 'Test Experiment' created by Peter Capaldi. It is described as a 'Microscopy experiment'. A note at the bottom says 'Latest dataset in this experiment'. A blue button labeled 'Dataset 1' is visible at the bottom left of the experiment card.

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

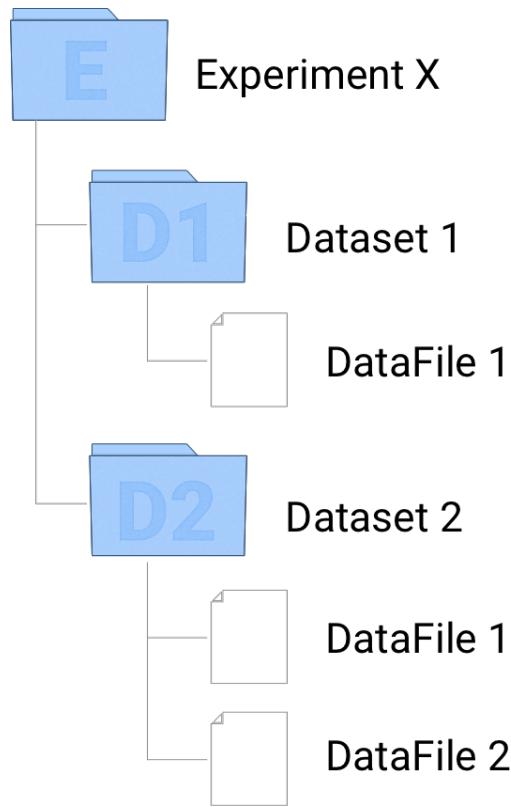


Fig. 1: Fig. 1. Data hierarchy in MyTardis

MyTardis Home About My Data Public Data Stats Help thedoctor ▾

Create Experiment

Title

Authors
Comma-separated authors and optional emails/URLs

Institution name

Description

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. At the top, there are links for Home, About, My Data, Public Data, Stats, Help, and a user account dropdown for 'thedoc'. Below the header, the experiment title 'Test Experiment' is displayed along with the author 'Peter Capaldi'. A 'Create publication' and 'Edit Experiment' button are visible. The main content area shows '1 Dataset' with a link to 'Download Selected'. A search bar allows filtering by description. Below the search is a card for 'Dataset 1', showing its status (0 files, 0 bytes, yesterday), a preview icon, and a 'Delete' button. At the bottom, there's a 'Download All' button with TAR and SFTP options, and a note that it's 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. It includes fields for 'Description' (set to 'Dataset 1'), 'Directory' (empty), and 'Instrument' (a dropdown menu). A large text area for notes is present, and a 'Save' button is located at the bottom left. 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.

The screenshot shows the 'Dataset 1' page from MyTardis. At the top, there's a navigation bar with links for Home, About, My Data, Public Data, Stats, and Help. A user account dropdown is also present. Below the navigation, the dataset title 'Dataset 1' is displayed, along with a note that it's from the experiment 'Test Experiment'. There are sections for 'Datafile Information' (with a note to click the icon to view details), 'Dataset Metadata' (empty), and 'Other Experiments' (empty). On the right, there's a summary of files (0 Files) and download options (TAR, SFTP, MDS). A green button labeled 'Add files...' is visible. At the bottom right, it says '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.

This screenshot shows the same 'Dataset 1' page as above, but with more content. It includes a 'Preview Images' section with a large black placeholder image and arrows for navigating through files. Below this is a 'Datafile Information' section with a note to click the icon to view details. The 'Dataset Metadata' section is still empty. On the right, there's a detailed file list showing 4 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 entry has a download icon, a plus sign for adding to a collection, and a grid icon for viewing details. A green 'Add files...' button is also present in this section.

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 'Dataset 1' page on MyTardis. At the top, there's a navigation bar with links for Home, About, My Data, Public Data, Stats, Help, and a user account dropdown. Below the navigation is the dataset title 'Dataset 1' and a note 'From the experiment: Test Experiment'. To the right is an 'Edit Dataset Description' button. In the center, there's a preview section titled 'Preview Images' with a large black placeholder image and arrows for navigation. Below it is a file list titled '4 Files' with a 'Download Dataset' button (labeled 'TAR') highlighted in red. The file list includes four entries: '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 entry has a checkbox, a download icon, and a more options icon. On the left, there's a 'Datafile Information' section with a note about viewing details and actions. On the right, there's a 'Dataset Metadata' section with a note that there is no metadata for this dataset.

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.

The screenshot shows the 'Test Experiment' page. At the top, there's a navigation bar with links for Home, About, My Data, Public Data, Stats, Help, and a dropdown for 'thedocor'. Below the navigation is the experiment title 'Test Experiment' by 'Peter Capaldi'. To the right are buttons for 'Create publication' and 'Edit Experiment'. Below the title, it says '1 Dataset' with a link to 'Download Selected'. A search bar allows filtering datasets based on descriptions. The main content area shows 'Dataset 1' which contains four files (indicated by small icons) and has a total size of 3.1 GB. Below the dataset list are download links for 'TAR' and 'SFTP'. At the bottom right, it says '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	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](#)

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.

The screenshot shows a user sharing interface. At the top, it displays the current user's information: [thedoctor] (doctor@tardis.org) with permissions Read, Edit, Owner, and a row of buttons for Edit and Delete. Below this, a section titled "Add new user" contains fields for "Username" (an empty input field) and "Permissions" (a dropdown menu set to "View and Edit"). A note below the permissions dropdown states: "Owners have the ability to change access controls and share experiments with others." A blue button labeled "+ Add User" is located at the bottom left of the form area.

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:

rosetyler

Permissions:

View and Edit

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
Create New Temporary Link				

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 two mechanisms for registering a public key:

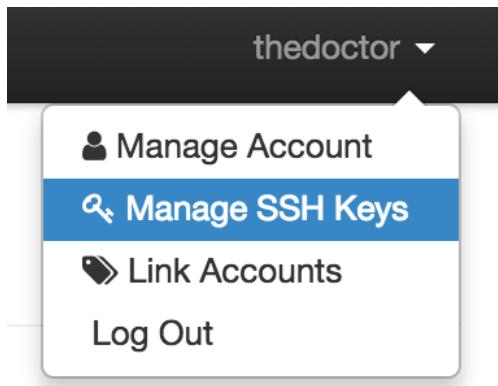
1. *Adding an existing public SSH key to MyTardis*
2. *Have MyTardis to generate an SSH key pair*

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:

A screenshot of the "SSH Keys" management page. At the top, there is a navigation bar with links: Home, About, My Data, Public Data, Stats, and Help. Below the navigation bar, the title "SSH Keys" is displayed, followed by the sub-instruction "Add and manage your public SSH keys for accessing MyTardis via SFTP.". At the bottom of the page, there are two green buttons: "+ Add key" and "+ Generate key". A yellow callout box contains the text "You don't have any public keys registered. Please add keys using the Add key button above.".

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 AAAAB3NzaC1yc2EAAAQABAAQDYZYnXpTP6e/BsQw+....`

FTP

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
AAAAAB3NzaC1yc2EAAAQABAAQDYZYnXpTP6e/BsQw+QEvdtpvdKJV2pxc9Q
DISjpqsV464bxerFs7+deNShlpJiHy7zBO4zbzGd2vE2wuxXa02tGRwKghQm3/fFRH6ZL4
pfJ/tkasYd7i5sAIKIkh0tPVVU+6HYbnbNjN+rIYh16ECrp7+YmtwPwVieeiTn69cSTkG+way
TFXwO6M3dolb/SMY8zE5JM0tVnaxBJTwNwpQZHmQW4eh87elHbv7QrtRF7ws/Ph6ME
Bpy1A9VI18ysBKMG4YxybxGuwOOMDAOI/2RhqMGV1MPXlushIOAy9vrYCeS40AukYiE
```

[Close](#) [Add Key](#)

4. Click the *Add Key* button.

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

SSH Keys						
Add and manage your public SSH keys for accessing MyTardis via SFTP.						
						+ Add key
						+ Generate key
Key						Delete
massive						Delete
Type: SSH-RSA						
Fingerprint: f68edca12548c068f8836d412230bdd4						
Date added: 2018-08-07						

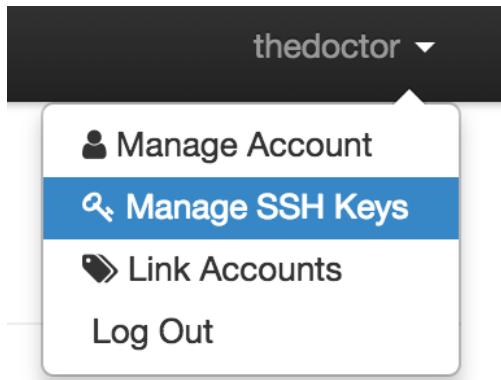
Have MyTardis to generate an SSH key pair

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.

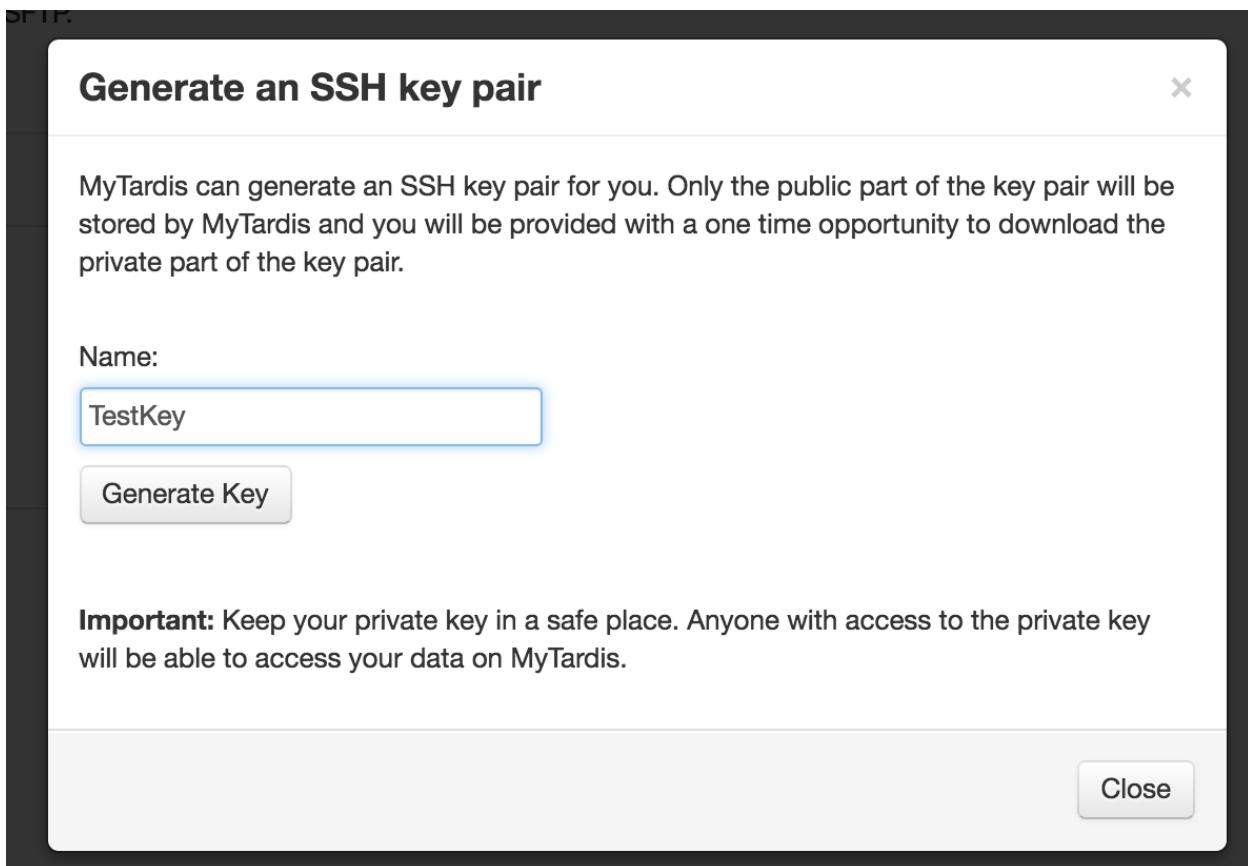


2. Click the + *Generate key* button:

A screenshot of the "SSH Keys" management page. At the top, there's a navigation bar with "MyTardis" and links for "Home", "About", "My Data", "Public Data", "Stats", and "Help". Below that is a section titled "SSH Keys" with the sub-instruction "Add and manage your public SSH keys for accessing MyTardis via SFTP.". There are two buttons at the top right: "+ Add key" and "+ Generate key" (which is highlighted with a red box). The main area shows a table with one row:

Key	
massive Type: SSH-RSA Fingerprint: f68edca12548c068f8836d412230bdd4 Date added: 2018-08-07	Delete

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.



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

MyTardis Home About My Data Public Data Stats Help

SSH Keys

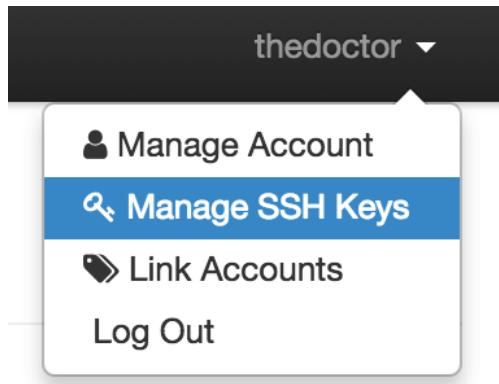
Add and manage your public SSH keys for accessing MyTardis via SFTP.

Key	+ Add key	+ Generate key
massive Type: SSH-RSA Fingerprint: f68edca12548c068f8836d412230bdd4 Date added: 2018-08-07	Delete	
TestKey Type: SSH-RSA Fingerprint: 3c4c3c3eb293d274fbebe687cc46aff76 Date added: 2018-08-07	Delete	

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.

A screenshot of the "SSH Keys" management page. At the top, there's a navigation bar with "MyTardis" and links for "Home", "About", "My Data", "Public Data", "Stats", and "Help". Below that is a section titled "SSH Keys" with the sub-instruction "Add and manage your public SSH keys for accessing MyTardis via SFTP.". There are two green buttons: "+ Add key" and "+ Generate key". The main area shows a table of keys:

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

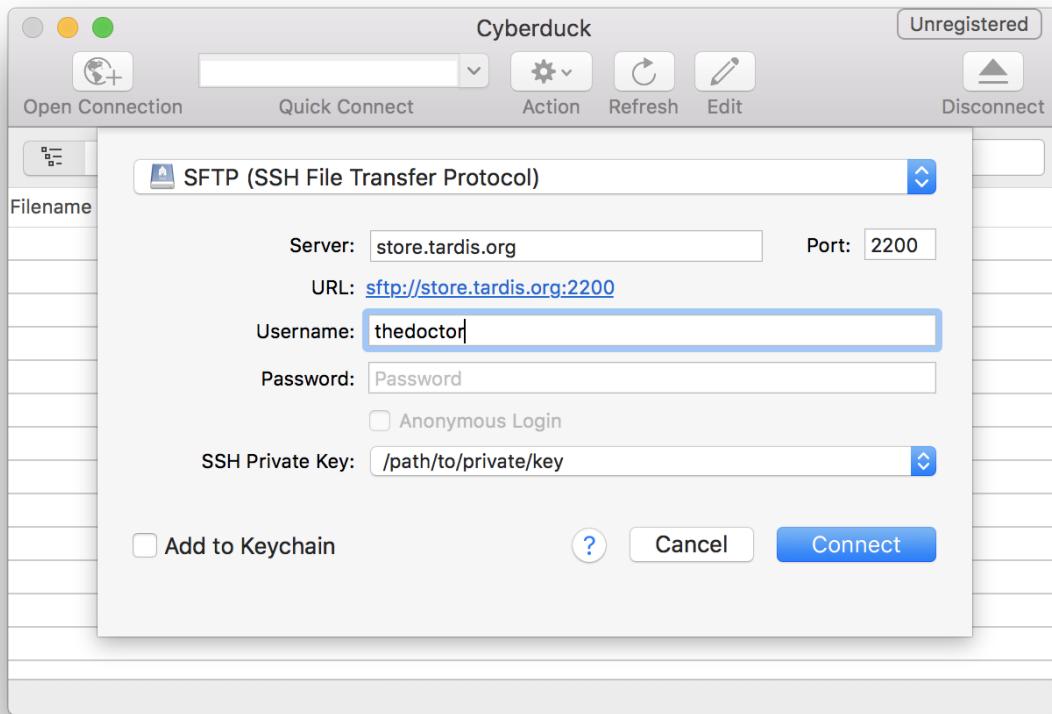
Attempting to connect to MyTardis via SFTP using the private sister key to the deleted key will no longer work.

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](#)

[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)



- 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.

11.1.3 Publication Workflow

Basic Usage

Listing Existing Publications and Drafts

Click on “My Publications” in the menu bar to see existing publications and drafts which you have access to:

My Publications

[Create publication](#)

Drafts

ID	Publication Title	DOI	Release Date	Actions
105	My Publication Title		2017-11-01	Resume draft Delete draft Mint DOI Share

Scheduled

You do not have access to any scheduled publications.

Released

You do not have access to any released publications.

Creating a Draft Publication

From the “My Publications” page, click “Create publication” to begin a draft publication:

[Create publication](#)

The publication form should appear, showing whatever introductory text has been configured by your MyTardis administrator, using the PUBLICATION_INTRODUCTION setting in `tardis/settings.py`:

MyTardis Publishing Form

This form will help MyTardis to describe and provide online access to your data.

Once data has been released for publication, you will receive a **DOI** link to this data for citation.

Follow the form to identify datasets and provide additional information.

Data will be released under a **Creative Commons BY** license: Whenever a work is copied or redistributed under a Creative Commons BY licence, the original creator (and any other nominated parties) must be credited and the source linked to.

For further enquiries, please contact: store.star.help@monash.edu.

[Back](#) 1 of 4 [Agree](#)

After acknowledging the introductory text, you can begin selecting datasets to include in your publication:

Select datasets

Select your data! Below is all the data you've generated. Select the datasets you want to include in this publication.

Publication title	My Publication Title			
Description	My publication description			
Select experiment	Added datasets			
Experiment 99	Experiment	Description		
Select datasets	<table border="1"> <thead> <tr> <th>Dataset 1</th> </tr> </thead> <tbody> <tr> <td>+ Add dataset</td> </tr> </tbody> </table>		Dataset 1	+ Add dataset
Dataset 1				
+ Add dataset				
Filter				

[Back](#) 2 of 4 [Save and continue](#)

Selected datasets appear on the right:

Select datasets

Select your data! Below is all the data you've generated. Select the datasets you want to include in this publication.

Publication title
My Publication Title

Description
My publication description

Select experiment
Experiment 99

Select datasets
Filter

+ Add dataset

Added datasets	
Experiment	Description
Experiment 99	Dataset 1

Back 2 of 4 Save and continue

You can then describe each included dataset (beyond what is already in the dataset's description field):

Extra information

The following extra information is required based on your dataset selection:

Dataset description
Please provide a description for each dataset.

Dataset 1
Some extra info about dataset 1.

Back 3 of 4 Save and continue

The final page allows you to specify the authors, license, acknowledgements and release date. **The last page of the form will be split into two pages in the next version of the publication form.**

Attribution and licensing

Name	Institution	Email
My Name	Monash University	my.name@example.com

Acknowledgements

Example acknowledgements

Select an option from the list above for example text.

Thanks to everyone who helped me to collect my data!

Select a license

Creative Commons Attribution 3.0 Australia (CC BY 3.0)

[Read the full license here.](#)

Release date

What is the earliest date that this publication may be released, subject to any additional restrictions?

01/11/2017 [today](#)

I am authorised to publish the selected datasets with the license and embargo period selected and have received the appropriate consent from co-authors and other relevant parties.

Almost done, but make sure all details are correct. Once submitted, you cannot make changes.

[Back](#) 4 of 4 [Save and finish later](#) [Submit](#)

After clicking “Save and finish Later”, you can see the publication in the Drafts section of the “My Publications” page:

The screenshot shows the 'My Publications' section of the MyTardis interface. At the top, there's a navigation bar with links for Home, About, My Data, My Publications (which is the active tab), Public Data, Stats, Facility Overview, and Help. A user profile 'wettenhj' is shown on the right. Below the navigation is a header 'My Publications' with a 'Create publication' button. A 'Drafts' section follows, featuring a table with columns: ID, Publication Title, DOI, Release Date, and Actions. One row is present: ID 105, Publication Title 'My Publication Title', DOI (not visible), Release Date '2017-11-01', and Actions buttons for 'Resume draft' (blue), 'Delete draft' (red), 'Mint DOI' (blue), and 'Share' (blue).

Scheduled

You do not have access to any scheduled publications.

Released

You do not have access to any released publications.

Resuming a Draft Publication

The “Resume draft” button below opens the publication form, allowing you to make further changes:

This screenshot is identical to the one above, showing the 'My Publications' page with the 'Drafts' section. The 'Resume draft' button for the single listed publication is highlighted with a yellow box, indicating it is the target of the user's action.

Scheduled

You do not have access to any scheduled publications.

Released

You do not have access to any released publications.

When resuming a draft, you will be taken straight to the second page of the publication form below, skipping the introductory text:

Select datasets

Select your data! Below is all the data you've generated. Select the datasets you want to include in this publication.

Publication title	My Publication Title							
Description	My publication description							
Select experiment	Experiment 99 <input type="button" value="▼"/>							
Select datasets	Filter <div style="border: 1px solid #ccc; height: 150px; margin-top: 10px;"></div>							
<input type="button" value="+ Add dataset"/>								
Added datasets <table border="1"> <thead> <tr> <th>Experiment</th> <th>Description</th> <th>Remove</th> </tr> </thead> <tbody> <tr> <td>Experiment 99</td> <td>Dataset 1</td> <td>-</td> </tr> </tbody> </table>			Experiment	Description	Remove	Experiment 99	Dataset 1	-
Experiment	Description	Remove						
Experiment 99	Dataset 1	-						
<input type="button" value="Back"/>	2 of 4	<input type="button" value="Save and continue"/>						

Deleting a Draft Publication

The “Delete draft” button below allowd you to delete a draft publication:

[MyTardis](#) Home About My Data **My Publications** Public Data Stats Facility Overview Help wettenhj ▾

My Publications

Drafts

ID	Publication Title	DOI	Release Date	Actions
105	My Publication Title		2017-11-01	<input type="button" value="Resume draft"/> <input type="button" value="Delete draft"/> <input type="button" value="Mint DOI"/> <input type="button" value="Share"/>

Scheduled
You do not have access to any scheduled publications.

Released
You do not have access to any released publications.

You will be asked to confirm that you really want to delete the draft publication:

Are you sure you want to delete Publication ID 105?

You cannot undo this action!

[NO, KEEP IT.](#) [YES, DELETE IT!](#)

Minting a DOI (Digital Object Identifier)

The “Mint DOI” button below allowd you to mint a DOI (Digital Object Identifier):

The screenshot shows the "My Publications" section of the MyTardis interface. At the top, there is a navigation bar with links for Home, About, My Data, My Publications (which is the active tab), Public Data, Stats, Facility Overview, and Help. A user profile icon for "wettenhj" is also present. Below the navigation, the title "My Publications" is displayed, along with a "Create publication" button. The main content area is titled "Drafts" and contains a table with one row. The table columns are labeled "ID", "Publication Title", "DOI", "Release Date", and "Actions". The single row shows "105" as the ID, "My Publication Title" as the title, "2017-11-01" as the release date, and four buttons in the "Actions" column: "Resume draft" (blue), "Delete draft" (red), "Mint DOI" (blue with a DOI icon), and "Share" (blue).

Scheduled

You do not have access to any scheduled publications.

Released

You do not have access to any released publications.

You will be asked to confirm that you really want to mint a DOI:

Are you sure you want to mint a DOI for Publication ID 105?

You cannot undo this action!

[NO, DON'T MINT A DOI.](#) [YES, MINT A DOI!](#)

Sharing a Draft Publication

The “Share” button below opens temporary links dialog, allowing you grant short-term access (3 months) to the draft publication via a temporary URL:

The screenshot shows the 'My Publications' section of the MyTardis interface. At the top, there's a navigation bar with links for Home, About, My Data, My Publications (which is the active tab), Public Data, Stats, Facility Overview, and Help. A user profile dropdown 'wettenhj' is also present. Below the navigation, a 'Create publication' button is visible. The main area is titled 'Drafts' and lists one item: '105 My Publication Title' from '2017-11-01'. For this item, there are four buttons: 'Resume draft' (blue), 'Delete draft' (red), 'Mint DOI' (blue), and 'Share' (blue).

Scheduled

You do not have access to any scheduled publications.

Released

You do not have access to any released publications.

The first time you click Share, you won't have any tokens for granting temporary access:

A modal window titled 'No tokens defined' contains a single button labeled 'Create New Temporary Link'.

After clicking the "Create New Temporary Link" button, you should see a temporary link:

Temporary Link	Archive direct download	Expiry	Granted By
Page link Copy to clipboard	Archive link Copy to clipboard	2017-12-01	wettenhj Delete
Create New Temporary Link			

Submitting a Publication for Scheduled Release

When you are ready to submit your data publication for release, you can click the "Submit" button after checking that you have entered all of the required information, including the date on which the publication should be released.

Release date
What is the earliest date that this publication may be released, subject to any additional restrictions?

27/01/2018 today

I am authorised to publish the selected datasets with the license and embargo period selected and have received the appropriate consent from co-authors and other relevant parties.

Almost done, but make sure all details are correct. Once submitted, you cannot make changes.

Back 4 of 4 Save and finish later Submit

At this point, a DOI (Digital Object Identifier) will be minted and your publication will be moved from the Drafts section of the My Publications page to the Scheduled or the Released section, depending on the Release Date you entered. In the example below, the Release Date was in the future, so the publication appears in the Scheduled section.

Scheduled

↓ ID	Publication Title	DOI	Release Date	Actions
114	My Test Publication	10.5072/20/TEST_DOI_5a694a42b9214	2018-01-27	Retract Share

Submitting a Publication for Immediate Release

When you are ready to submit your data publication for immediate release, you can click the “Submit” button after checking that you have entered all of the required information, including the date on which the publication should be released (today).

Release date
What is the earliest date that this publication may be released, subject to any additional restrictions?

25/01/2018 today

I am authorised to publish the selected datasets with the license and embargo period selected and have received the appropriate consent from co-authors and other relevant parties.

Almost done, but make sure all details are correct. Once submitted, you cannot make changes.

Back 4 of 4 Save and finish later Submit

At this point, a DOI (Digital Object Identifier) will be minted and your publication will be moved from the Drafts section of the My Publications page to the Scheduled or the Released section, depending on the Release Date you

entered. In the example below, the Release Date was set to today, so the publication appears in the Released section.

Released

ID	Publication Title	DOI	Release Date	Actions
115	My Test Publication2	10.5072/20/TEST_DOI_5a694dbf1ff25	2018-01-25	Retract

For MyTardis System Administrators

Enabling or Disabling the Publication Workflow

The publication workflow is enabled by default in a new MyTardis installation:

tardis/default_settings/apps.py:

```
...
INSTALLED_APPS = (
    ...
    'tardis.apps.publication_workflow',
    ...
)
```

If necessary, it can be disabled in tardis/settings.py as shown below:

```
disabled_apps = ['publication_workflow']
for app in disabled_apps:
    INSTALLED_APPS = filter(lambda a: a != app, INSTALLED_APPS)
```

When the publication workflow is enabled, you should see the “My Publications” link in the menu bar:

Publication Workflow Settings

To use the default publication workflow settings, add the following to your tardis/settings.py:

```
from tardis.apps.publication_workflow.default_settings import *
```

Installing Licenses

The publication workflow app contains Creative Commons licenses in tardis/apps/publication_workflow/fixtures/licenses.json which can be installed with:

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

The screenshot shows the Django administration interface for managing licenses. At the top, it says "Django administration" and "Welcome, wettenhj. View site / Documentation / Change password / Log out". Below that, the URL "Home > Tardis_Portal > Licenses" is shown. The main title is "Select license to change". There is a "Action:" dropdown set to "-----" and a "Go" button. A message "0 of 6 selected" is displayed. A list of 6 licenses is shown in a table:

<input type="checkbox"/>	License
<input type="checkbox"/>	Creative Commons Attribution–NonCommercial–NoDerivatives 4.0 International (CC BY–NC–ND 4.0)
<input type="checkbox"/>	Creative Commons Attribution–NonCommercial–ShareAlike 4.0 International (CC BY–NC–SA 4.0)
<input type="checkbox"/>	Creative Commons Attribution–NoDerivatives 4.0 International (CC BY–ND 4.0)
<input type="checkbox"/>	Creative Commons Attribution–NonCommercial 4.0 International (CC BY–NC 4.0)
<input type="checkbox"/>	Creative Commons Attribution–ShareAlike 4.0 International (CC BY–SA 4.0)
<input type="checkbox"/>	Creative Commons Attribution 4.0 International (CC BY 4.0)

At the bottom left, it says "6 licenses". On the right, there is a "Add license +"

Minting DOIs

From Wikipedia, “a Digital Object Identifier or DOI is a persistent identifier or handle used to uniquely identify objects, standardized by the International Organization for Standardization”. For example, a DOI of “10.2196/mhealth.3314” can be resolved at <http://dx.doi.org/10.2196/mhealth.3314>, leading to <http://mhealth.jmir.org/2014/4/e50/>. DOIs are commonly used to provide permanent links to published articles, but they can also be used to provide permanent links to published data.

MyTardis’s publication workflow provides functionality for minting DOIs using Monash University’s DOI minting service at <https://mod.monash.edu> and its corresponding test service at <https://vera186.its.monash.edu/modc/>. DOIs can be minted manually by MyTardis system administrators, or from the user interface provided in the “My Publications” view.

To get started, you need to register your MyTardis service as a “Webservice App” in Monash’s DOI-minting service at <https://mod.monash.edu> or in its corresponding test service at <https://vera186.its.monash.edu/modc/>.

The screenshot shows the Monash University Library DOI minting service interface. At the top, it has the Monash University logo and the text "MONASH University Library Digital Object Identifier Service". On the right, there is a "Logout" link. Below that is a navigation bar with links for "Home", "DOI", "Webservice App", "Users", and "About Us".

The main content area shows a table titled "Total 4 Web Service Apps". It includes a "Filter by: -- select --" dropdown and a "+ Add New App" button. The table columns are: Name, Service Type, DOI Minting Prefix, App ID, App Password, and Description. One row is visible:

Name	Service Type	DOI Minting Prefix	App ID	App Password	Description
mytardis-demo	doi	Monash	514a087f-30eb-4860-a0ad-79a92e875564	[REDACTED]	DOI minting for MyTardis Demo Server

There is also a "View" button next to the row.

Then you need to configure your MyTardis’s `tardis/settings.py` to use the DOI minting service. For the “mytardis-demo” DOI-minting application registered above on <https://vera186.its.monash.edu/modc/>, we can use the following settings:

```

MODC_DOI_ENABLED = True
MODC_DOI_API_ID = '514a087f-30eb-4860-a0ad-79a92e875564'
# Use the real API password here:
MODC_DOI_API_PASSWORD = '*****'
MODC_DOI_MINT_DEFINITION = 'https://vera186.its.monash.edu/modc/ws/MintDoiService.wsdl'
↳
MODC_DOI_ACTIVATE_DEFINITION = 'https://vera186.its.monash.edu/modc/ws/' \
    'ActivateDoiService.wsdl'
MODC_DOI_DEACTIVATE_DEFINITION = 'https://vera186.its.monash.edu/modc/ws/' \
    'DeactivateDoiService.wsdl'
MODC_DOI_ENDPOINT = 'https://vera186.its.monash.edu/modc/ws/'
# The MODC_DOI_MINT_URL_ROOT should be set to the base URL of your publicly
# accessible MyTardis server. You can't test the DOI-minting functionality
# from a local dev environment (http://127.0.0.1:8000), because the URL you
# are minting a DOI for needs to be publicly accessible:
MODC_DOI_MINT_URL_ROOT = 'https://mytardisdemo.erc.monash.edu'
```

A DOI can be minted from the Django shell (mytardis.py shell_plus) as follows:

```

>>> from django.core.urlresolvers import reverse
>>> from tardis.apps.publication_workflow.doi import DOI
>>> doi = DOI()
>>> experiment_id = 105
>>> doi.mint(experiment_id, reverse('tardis_portal.view_experiment', args=(experiment_
    ↴id,)))
10.5072/20/TEST_DOI_5a14bcdd0f309
```

Now http://dx.doi.org/10.5072/20/TEST_DOI_5a14bcdd0f309 should resolve to <https://mytardisdemo.erc.monash.edu/experiment/view/105/>.

We can then deactivate this DOI until the required publication criteria are met:

```

>>> doi.deactivate()
(reply){
    serviceId = "514a087f-30eb-4860-a0ad-79a92e875564"
    responsecode = "MT003"
    message = "DOI 10.5072/20/TEST_DOI_5a14bcdd0f309 was successfully deactivated."
    doi = "10.5072/20/TEST_DOI_5a14bcdd0f309"
    url = None
    verbosemessage = None
}
```

If we need to look up the same DOI later, e.g. after exiting the Django shell and starting a new one, we can create a DOI instance with an existing DOI:

```

>>> from tardis.apps.publication_workflow.doi import DOI
>>> doi = DOI('10.5072/20/TEST_DOI_5a14bcdd0f309')
>>> doi.activate()
(reply){
    serviceId = "514a087f-30eb-4860-a0ad-79a92e875564"
    responsecode = "MT004"
    message = "DOI 10.5072/20/TEST_DOI_5a14bcdd0f309 was successfully activated."
    doi = "10.5072/20/TEST_DOI_5a14bcdd0f309"
    url = None
    verbosemessage = None
}
```

For a complete list of the possible response codes, see: <https://documentation.ands.org.au/display/DOC/Cite+My+>

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)

Login as `ubuntu` user, not root.

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
```

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

Make sure you are running Python 3.x:

```
$ python -V
Python 3.6.9
```

Now upgrade pip and setup tools:

```
pip install -U pip setuptools
```

Check pip version:

```
$ pip -V
pip 21.0.1 from /home/ubuntu/mytardis/mytardis/lib/python3.6/site-packages/pip_
  ↪(python 3.6)
```

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
```

To compile Webpack assets:

```
npm run-script build
```

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.

# 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=%"%" #_
˓→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/>.

Running in a Docker Container

Installation steps from above can be summarised as a following *Dockerfile*:

```
FROM ubuntu:18.04

ENV APT_KEY_DONT_WARN_ON_DANGEROUS_USAGE DontWarn
ENV DEBIAN_FRONTEND noninteractive
```

(continues on next page)

(continued from previous page)

```

ENV PYTHONUNBUFFERED 1

ENV LANG C.UTF-8

RUN apt-get update -y

RUN apt-get install -y \
    sudo 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

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

RUN pip3 install virtualenvwrapper

RUN groupadd -g 1001 ubuntu
RUN useradd -rm -d /home/ubuntu -s /bin/bash -g ubuntu -G sudo -u 1001 ubuntu
USER ubuntu
WORKDIR /home/ubuntu

RUN git clone -b develop git://github.com/mytardis/mytardis.git

ENV VIRTUAL_ENV=/home/ubuntu/mytardis/mytardis
RUN python3 -m venv $VIRTUAL_ENV
ENV PATH="$VIRTUAL_ENV/bin:$PATH"

RUN pip install -U pip setuptools

WORKDIR /home/ubuntu/mytardis

RUN pip install -U -r requirements.txt
RUN npm install --production
RUN npm run-script build

RUN echo "from .default_settings import *\n\
DEBUG = True\n\
DATABASES['default']['ENGINE'] = 'django.db.backends.sqlite3'\n\
DATABASES['default']['NAME'] = 'tardis_db'\n" >> tardis/settings.py

RUN python -c "import os; from random import choice; key_line = '%sSECRET_KEY=%" "%\n' % ('from .default_settings import *\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()"

RUN python manage.py migrate
RUN python manage.py createcachetable default_cache
RUN python manage.py createcachetable celery_lock_cache

EXPOSE 8000

CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]

```

This is a minimum configuration for MyTardis, which is not suitable for running in production.

To build and run Docker container locally you will need to run following commands:

```
docker build . --tag mytardis
docker run --name mytardis -p 8000:8000 mytardis
```

Alternatively you can use pre-built Docker image from DockerHub:

```
docker run --name mytardis -p 8000:8000 mytardis/mytardis:develop
```

If you are keep in MyTardis development, we recommended you to visit <https://github.com/mytardis/mytardis-dev> for a full stack development environment, which runs using Docker.

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>
"""
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:



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

```

upstream mytardis {
    server unix:/var/run/gunicorn/mytardis/socket;
    server 127.0.0.1:8000 backup;
}

server {
    listen 80 default_server;
    server_name demo.mytardis.org;
    return 301 https://$server_name$request_uri;
}

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

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

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

    # intermediate configuration. tweak to your needs.
    ssl_protocols TLSv1.1 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;

```

(continues on next page)

(continued from previous page)

```

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;
    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:r 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
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
```

(continues on next page)

(continued from previous page)

```
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
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

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

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

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

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

The *Schema* fields are:

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

The MyTardis naming convention is:

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

Name The display name of the schema.

Type Experiment, Dataset or Datafile.

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

The *ParameterName* fields are:

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

Name The identifier used to ingest parameters.

Full Name The display name of the parameter.

Units The display name of the units for numerical values.

Data Type One of:

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

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

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

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

Is Searchable Flag whether this parameter is searchable.

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

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

11.2.6 Setting Up Search

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

The single search box uses Elasticsearch, 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 `djangostorage_class` attribute of a StorageBox

These parameters are string types by default. However, by setting the optional parameter `value_type` to '`pickle`', any pickleable 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	
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	SET8
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	

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.

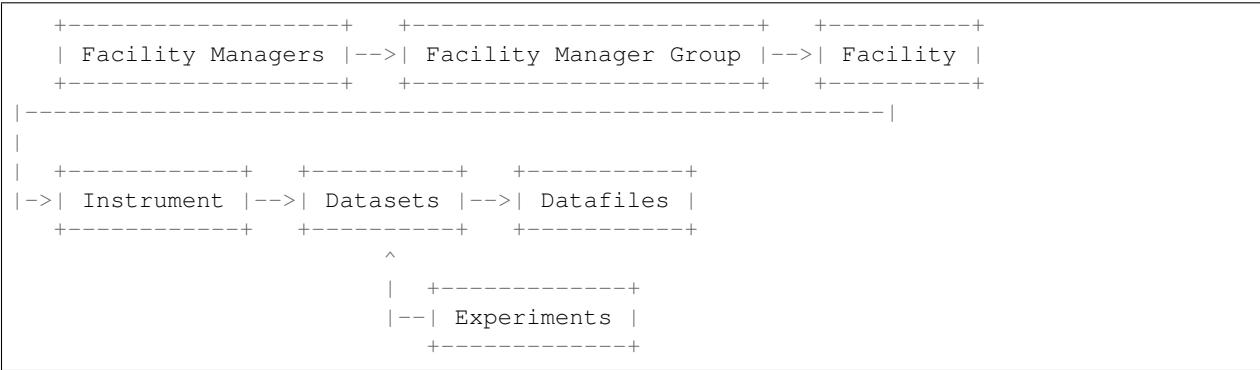
11.2.8 Facility Overview

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_TASK_DEFAULT_QUEUE = 'celery'
# The 'x-max-priority' argument will only be respected by the RabbitMQ broker,
# which is the recommended broker for MyTardis:
CELERY_TASK_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', namespace='CELERY')
...
```

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:

```
(mytardis) mytardis@celery0:~/mytardis$ ./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 annotations, raw data, versioned data and derived data.

Persistance and Data Architecture

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

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

Security Model

Authentication

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

Authorisation

MyTardis includes an extensible authorisation engine, documented in [ref-authframework](#) and [Authorisation 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.ACIAwareBackend',
)
```

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(realme=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://datafileshop.com/fileinfo/v1",
    "parameters": [
        {
            "name": "fileparameter1",
            "value": "123"
        },
        {
            "name": "fileparameter2",
            "value": "1234"
        }
    ]
}

```

Example requests script:

```

import requests
from requests.auth import HTTPBasicAuth

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

```

Via staging location

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

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

Example JSON input:

```

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

```

Via shared permanent storage location

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

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

Examples JSON:

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

urllib2 POST example script

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

```
import urllib2  
url = "http://localhost:8000/api/v1/MODEL/"  
headers = {'Accept': 'application/json',  
           'Content-Type': 'application/json'}  
auth_handler = urllib2.HTTPBasicAuthHandler()  
auth_handler.add_password(realms="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.eventlog package

Subpackages

tardis.apps.eventlog.migrations package

Submodules

tardis.apps.eventlog.migrations.0001_initial module

```
class tardis.apps.eventlog.migrations.0001_initial.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('auth', '__first__'), ('contenttypes', '0002_remove_content_type_name')]
    initial = True
    operations = [CreateModel name='Action', fields=[('id', AutoField()), ('timestamp', DateTimeField()), ('user', ForeignKey(User)), ('action', CharField(max_length=255)), ('extra', TextField())]]
```

Module contents

Submodules

tardis.apps.eventlog.admin module

```
class tardis.apps.eventlog.admin.LogAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ['timestamp', 'user', 'action', 'extra']
    list_filter = ['action__name', 'timestamp']
    media
    raw_id_fields = ['user']
    search_fields = ['user__username', 'user__email', 'extra']
```

tardis.apps.eventlog.default_settings module

tardis.apps.eventlog.fields module

```
class tardis.apps.eventlog.fields.JSONField(verbose_name=None, name=None, primary_key=False, max_length=None, unique=False, blank=False, null=False, db_index=False, rel=None, default=<class 'django.db.models.fields.NOT_PROVIDED'>, editable=True, serialize=True, unique_for_date=None, unique_for_month=None, unique_for_year=None, choices=None, help_text='', db_column=None, db_tablespace=None, auto_created=False, validators=(), error_messages=None)
```

Bases: django.db.models.fields.Field

Simple class to make *JSONField* available to sqlite backends.

db_type (*connection*)

Return the database column data type for this field, for the provided connection.

from_db_value (*value, expression, connection*)

get_prep_value (*value*)

Perform preliminary non-db specific value checks and conversions.

to_python (*value*)

Convert the input value into the expected Python data type, raising django.core.exceptions.ValidationError if the data can't be converted. Return the converted value. Subclasses should override this.

value_to_string (*obj*)

Return a string value of this field from the passed obj. This is used by the serialization framework.

tardis.apps.eventlog.models module

```
class tardis.apps.eventlog.models.Action(id, 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.

log_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.

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

```
class tardis.apps.eventlog.models.Log(id, timestamp, action, user, content_type, object_id,
                                         extra)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

action

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.

action_id

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

extra

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

```
get_previous_by_timestamp(*, field=<django.db.models.fields.DateTimeField: 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.

obj

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.

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>**timestamp**

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**tardis.apps.eventlog.signals module****tardis.apps.eventlog.urls module****tardis.apps.eventlog.utils module**

`tardis.apps.eventlog.utils.get_request_data(request)`

`tardis.apps.eventlog.utils.log(action, user=None, obj=None, extra=None, request=None)`

Module contents**tardis.apps.filepicker package****Submodules****tardis.apps.filepicker.filepicker_settings module****tardis.apps.filepicker.urls module****tardis.apps.filepicker.utils module**

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

`class tardis.apps.filepicker.utils.FilepickerFile(url)`

Bases: `object`

`cleanup()`

Removes any downloaded objects and closes open files.

`filepicker_url_regex = re.compile('https?:\/\/www\.filepicker\.io\/api\/file\/\w+')`

`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.hsm package

Subpackages

tardis.apps.hsm.migrations package

Submodules

tardis.apps.hsm.migrations.0001_initial_data module

Django migration to provide initial schema data for the hsm app.

```
class tardis.apps.hsm.migrations.0001_initial_data.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    HSM Schema migrations
    dependencies = [('tardis_portal', '0016_add_timestamps')]
    operations = [RunPython <function forward_func>, <function reverse_func>]
tardis.apps.hsm.migrations.0001_initial_data.forward_func(apps, schema_editor)
    Create HSM Schemas
tardis.apps.hsm.migrations.0001_initial_data.reverse_func(apps, schema_editor)
    Remove HSM Schemas
```

Module contents

tardis.apps.hsm.tests package

Submodules

tardis.apps.hsm.tests.test_api module

Testing the hsm app's extensions to the tastypie-based mytardis api .. moduleauthor:: James Wettenhall <james.wettenhall@monash.edu>

```
class tardis.apps.hsm.tests.test_api.HsmAppApiTestCase(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_ds_check()
    Test the task for updating a dataset's Online Status metadata

test_online_check_unsupported_storage_class()
    Test API endpoint for HSM online check with unsupported storage class

test_online_check_unverified_file()
    Test API endpoint for HSM online check with unverified file

test_online_check_valid_storage_class()
    Test API endpoint for HSM online check with valid storage class

test_online_check_with_bad_password()
    Test API endpoint for HSM online check without ACL access to DFO

test_online_check_without_acl_access_to_dfo()
    Test API endpoint for HSM online check without ACL access to DFO

test_online_count()
    Test counting the number of online files in a dataset

    This method (designed to be fast) looks directly at the files on disk without checking if each file is verified
    in the database. Since the underlying method checks for hidden external attribute on mount.cifs mounted
    filesystem, it may return offline status for a file in test environment

test_recall()
    Test the API endpoint for recalling a file from tape on an HSM system. The “recall” just attempts to read
    the first bit of the file which on most HSM systems will automatically trigger a recall.

test_stat_subprocess()
    If the Python os.stat function can't determine the number of blocks used by the file, then the hsm app falls
    back to using a subprocess

```

tardis.apps.hsm.tests.test_app_config module

```

class tardis.apps.hsm.tests.test_app_config.HsmAppConfigTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    test_app_config()

```

tardis.apps.hsm.tests.test_email_templates module

Tests for HSM app’s email templates

```

class tardis.apps.hsm.tests.test_email_templates.HsmAppEmailTemplateTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

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

    test_recall_complete_email()
        Test using the email template for a successful recall

    test_recall_failed_email()
        Test using the email template for a failed recall

```

tardis.apps.hsm.tests.test_migrations module

Tests for HSM app's migrations

```
class tardis.apps.hsm.tests.test_migrations.HsmAppMigrationTestCase(methodName='runTest')
    Bases: django.test.testcases.TestCase

    connection = <django.db.backends.sqlite3.base.DatabaseWrapper object>

    test_migration()
        Test unapplying and reapplying the migration which creates the metadata for the HSM app
```

Module contents

Submodules

tardis.apps.hsm.api module

Additions to MyTardis's REST API

```
class tardis.apps.hsm.api.DatasetAppResource(api_name=None)
    Bases: tardis.tardis_portal.api.DatasetResource

    Extends MyTardis's API for Datasets, adding in a method to count online files in a Hierarchical Storage Management (HSM) system

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        authorization
            Authorisation class for Tastypie.

        queryset
            resource_name = 'dataset'

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

        dataset_recall(request, **kwargs)
            Send and email to Site admin to recall Dataset from HSM system

        declared_fields = {}

        online_count(request, **kwargs)
            Return the number of online files and the total number of files in a dataset stored in a Hierarchical Storage Management (HSM) system

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

class tardis.apps.hsm.api.ReplicaAppResource(api_name=None)
    Bases: tardis.tardis_portal.api.ReplicaResource

    Extends MyTardis's API for DFOs, adding in a recall method and an online check method for files in a Hierarchical Storage Management (HSM) system

    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 = {}
    dfo_is_online(request, **kwargs)
        Return the online status of a DataFileObject stored in a Hierarchical Storage Management (HSM) system
    prepend_urls()
        A hook for adding your own URLs or matching before the default URLs.
    recall_dfo(request, **kwargs)
        Recall archived DataFileObject from HSM system
```

tardis.apps.hsm.apps module

```
class tardis.apps.hsm.apps.HsmConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig
    name = 'tardis.apps.hsm'
```

tardis.apps.hsm.check module

HSM check module. Method for detecting whether a MyTardis DataFileObject in Hierarchical Storage Management is online or offline (on tape).

```
tardis.apps.hsm.check.dataset_online_count(dataset)
    Checks how many of a dataset's files are online
```

dataset [Dataset] The Dataset for which to check the status

int The number of online files in this dataset

```
tardis.apps.hsm.check.dfo_online(dfo)
    Checks whether the underlying file of a DataFileObject is online
```

dfo [DataFileObject] The DataFileObject for which to check the status

bool Status for whether dfo is online.

DataFileObjectNotVerified If dfo is unverified

StorageClassNotSupportedError If the *django_storage_class* for the StorageBox of the input DataFileObject is not supported

tardis.apps.hsm.default_settings module

```
tardis.apps.hsm.default_settings.HSM_MAX_INODE_FILE_SIZE = 384
    The maximum size of files that can be stored within the inode (with stat reporting 0 blocks).
```

To determine the right value for your filesystem, you can create some small test files and check how small they have to be to have stat report 0 blocks.

<https://en.wikipedia.org/wiki/Inode#Inlining>

tardis.apps.hsm.email_text module

```
tardis.apps.hsm.email_text.email_dataset_recall_requested(dataset, user)
tardis.apps.hsm.email_text.email_dfo_recall_complete(dfo, user)
tardis.apps.hsm.email_text.email_dfo_recall_failed(dfo, user)
tardis.apps.hsm.email_text.email_dfo_recall_requested(dfo_id, user)
tardis.apps.hsm.email_text.interpolate_template(template_name, **kwargs)
```

tardis.apps.hsm.exceptions module

Exceptions related to Hierarchical Storage Management (HSM)

```
exception tardis.apps.hsm.exceptions.DataFileObjectNotVerified
    Bases: tardis.apps.hsm.exceptions.HsmException
```

Exception raised when an operation is attempted on an unverified DataFile

```
exception tardis.apps.hsm.exceptions.HsmException
    Bases: Exception
```

Base class for other exceptions to inherit from

```
exception tardis.apps.hsm.exceptions.StorageClassNotSupportedError
    Bases: tardis.apps.hsm.exceptions.HsmException
```

Exception raised when a storage class is not supported

tardis.apps.hsm.storage module

```
class tardis.apps.hsm.storage.HsmFileSystemStorage(location=None, base_url=None,
                                                       file_permissions_mode=None,
                                                       direc-
                                                       tory_permissions_mode=None)
Bases: django.core.files.storage.FileSystemStorage
```

This storage class is used to describe Hierarchical Storage Management filesystems where files may be offline (only or tape), but can be recalled to disk on demand

tardis.apps.hsm.tasks module

Tasks for recalling data from tape in Hierarchical Storage Management (HSM) systems

tardis.apps.hsm.urls module

Minimal urls.py, so we can do a reverse lookup for the ‘hsm_api_download_dfo’ URL pattern.

‘hsm_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.hsm.utils module

HSM utils module. Utilities for detecting whether files in Hierarchical Storage Management are online or offline (on tape).

`tardis.apps.hsm.utils.file_is_online(path)`

Detects whether a file is online or offline (on tape).

path [str] Path to the file for which we want to determine online/offline status.

bool specifies whether the file is online i.e., not on tape.

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

Return type `frozense`

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.IntegerField>), ('user', <django.db.models.ForeignKey>), ('method', <django.db.models.CharField>), ('accesstoken', <django.db.models.TextField>), ('refresh_token', <django.db.models.TextField>), ('access_expires', <django.db.models.DateTimeField>), ('refresh_expires', <django.db.models.DateTimeField>), ('is_valid', <django.db.models.BooleanField>)]), CreateIndex(name='idx_OpenidACLMigration_accesstoken', fields=['accesstoken'])]
```

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='runTest')
    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='')
    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 MultipleObjectsReturned

Bases: `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

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`

```
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 DoesNotExist`

Bases: 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`

```
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`

`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_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.publication_workflow package

Subpackages

tardis.apps.publication_workflow.migrations package

Submodules

tardis.apps.publication_workflow.migrations.0001_initial module

```
class tardis.apps.publication_workflow.migrations.0001_initial.Migration(name,  
                           app_label)  
    Bases: django.db.migrations.migration.Migration  
    dependencies = ['tardis_portal', '0011_auto_20160505_1643'], ('auth', '0008_alter_user')  
    operations = [<RunPython <function forwards_func>, <function reverse_func>]  
  
tardis.apps.publication_workflow.migrations.0001_initial.create_metadata_schemas(apps,  
                               schema_editor)  
  
tardis.apps.publication_workflow.migrations.0001_initial.forwards_func(apps,  
                               schema_editor)  
  
tardis.apps.publication_workflow.migrations.0001_initial.load_licenses(apps,  
                               schema_editor)  
  
tardis.apps.publication_workflow.migrations.0001_initial.remove_licenses(apps,  
                               schema_editor)  
  
tardis.apps.publication_workflow.migrations.0001_initial.remove_metadata_schemas(apps,  
                               schema_editor)  
  
tardis.apps.publication_workflow.migrations.0001_initial.reverse_func(apps,  
                               schema_editor)
```

Module contents

tardis.apps.publication_workflow.tests package

Submodules

tardis.apps.publication_workflow.tests.test_email_templates module

```
class tardis.apps.publication_workflow.tests.test_email_templates.EmailTemplates (methodName='r
Bases: django.test.testcases.TestCase

test_pub_rejected_template()
test_pub_released_template()
test_pub_reverted_to_draft_template()
```

tardis.apps.publication_workflow.tests.test_form module

Tests relating to publication form

```
class tardis.apps.publication_workflow.tests.test_form.PublicationFormTestCase (methodName='r
Bases: django.test.testcases.TestCase

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

test_create_draft_publication()
    Test creating draft publication

test_fetch_experiments_and_datasets()
    Test fetching experiments and datasets for publication form

test_resume_draft_publication()
    Test resuming draft publication

test_submit_form()
    Test submitting publication form

test_update_dataset_selection()
    Test updating dataset selection in publication form

test_update_extra_info()
    Test updating extra info in publication form
```

tardis.apps.publication_workflow.tests.test_my_publications module

Tests relating to My Publications view

```
class tardis.apps.publication_workflow.tests.test_my_publications.MyPublicationsTestCase (m
Bases: django.test.testcases.TestCase

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

test_retrieve_draft_pubs_list()
    Test retrieving list of draft publications
```

```
test_retrieve_released_pubs_list()
    Test retrieving list of released publications

test_retrieve_retracted_pubs_list()
    Test retrieving list of retracted publications

test_retrieve_scheduled_pubs_list()
    Test retrieving list of scheduled publications
```

tardis.apps.publication_workflow.tests.test_revert_finalize module

Tests relating to reverting a publication to a draft and finalizing a publication

```
class tardis.apps.publication_workflow.tests.test_revert_finalize.RevertFinalizeTestCase(m
    Bases: django.test.testcases.TestCase

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

    test_finalize_publication()
        Test finalizing a publication

    test_revert_publication_to_draft()
        Test reverting a publication to a draft
```

tardis.apps.publication_workflow.tests.test_tokens module

Tests relating to tokens (temporary links) used by My Publications view

```
class tardis.apps.publication_workflow.tests.test_tokens.PublicationTokensTestCase(methodName
    Bases: django.test.testcases.TestCase

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

    test_create_token()
        Test creating a temporary link for a draft publications

    test_delete_token()
        Test deleting a temporary link for a draft publications

    test_retrieve_access_list_tokens_json()
        Test retrieving a JSON list of tokens
```

Module contents

Submodules

tardis.apps.publication_workflow.apps module

```
class tardis.apps.publication_workflow.apps.PublicationWorkflowConfig(app_name,
    app_module)
    Bases: tardis.app_config.AbstractTardisAppConfig

    name = 'tardis.apps.publication_workflow'
    verbose_name = 'Publication Workflow'
```

tardis.apps.publication_workflow.default_settings module

default settings for publication app override in main settings

tardis.apps.publication_workflow.doi module

```
class tardis.apps.publication_workflow.doi.DOI (doi=None)
    Bases: object

    activate (doi=None)
    deactivate (doi=None)
    mint (experiment_id, uri, publisher='Monash University')
```

tardis.apps.publication_workflow.email_text module

```
tardis.apps.publication_workflow.email_text.email_pub_rejected (pub_title, message=None)
tardis.apps.publication_workflow.email_text.email_pub_released (pub_title, doi=None)
tardis.apps.publication_workflow.email_text.email_pub_retracted (pub_title, message=None)
tardis.apps.publication_workflow.email_text.email_pub_reverted_to_draft (pub_title, message=None)
tardis.apps.publication_workflow.email_text.interpolate_template (template_name, **kwargs)
```

tardis.apps.publication_workflow.managers module

```
class tardis.apps.publication_workflow.managers.PublicationManager
    Bases: tardis.tardis_portal.managers.ExperimentManager
```

Extends the ExperimentManager class, added methods for draft, scheduled and published publications.

create_draft_publication (*user*, *publication_title*, *publication_description*)
Create a new draft publication

draft_publications (*user*)
Return all draft publications which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters *user* (*djongo.contrib.auth.models.User*) – the user who we are retrieving draft publications for

Returns A QuerySet of experiments representing the draft publications

Return type QuerySet

released_publications (*user*)
Return all released publications which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters `user` (`django.contrib.auth.models.User`) – the user who we are retrieving draft publications for

Returns A QuerySet of experiments representing the draft publications

Return type QuerySet

retracted_publications (`user`)

Return all retracted publications which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters `user` (`django.contrib.auth.models.User`) – the user who we are retrieving retracted publications for

Returns A QuerySet of experiments representing the retracted publications

Return type QuerySet

scheduled_publications (`user`)

Return all scheduled publications which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters `user` (`django.contrib.auth.models.User`) – the user who we are retrieving draft publications for

Returns A QuerySet of experiments representing the draft publications

Return type QuerySet

tardis.apps.publication_workflow.models module

class `tardis.apps.publication_workflow.models.Publication(*args, **kwargs)`

Bases: `tardis.tardis_portal.models.experiment.Experiment`

Publication records are just experiment records with some metadata, so they can be represented by the Experiment model. However, it is useful to have a publication-specific manager, so we can retrieve publication records with Publication.safe.draft_publications(user)

exception DoesNotExist

Bases: `tardis.tardis_portal.models.experiment.DoesNotExist`

exception MultipleObjectsReturned

Bases: `tardis.tardis_portal.models.experiment.MultipleObjectsReturned`

class PublicationManager

Bases: `tardis.tardis_portal.managers.ExperimentManager`

Extends the ExperimentManager class, added methods for draft, scheduled and published publications.

create_draft_publication (`user, publication_title, publication_description`)

Create a new draft publication

draft_publications (`user`)

Return all draft publications which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters `user` (`django.contrib.auth.models.User`) – the user who we are retrieving draft publications for

Returns A QuerySet of experiments representing the draft publications

Return type QuerySet

`released_publications (user)`

Return all released publications which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters `user` (`django.contrib.auth.models.User`) – the user who we are retrieving draft publications for

Returns A QuerySet of experiments representing the draft publications

Return type QuerySet

`retracted_publications (user)`

Return all retracted publications which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters `user` (`django.contrib.auth.models.User`) – the user who we are retrieving retracted publications for

Returns A QuerySet of experiments representing the retracted publications

Return type QuerySet

`scheduled_publications (user)`

Return all scheduled publications which are owned by a particular user, including those shared with a group of which the user is a member.

Parameters `user` (`django.contrib.auth.models.User`) – the user who we are retrieving draft publications for

Returns A QuerySet of experiments representing the draft publications

Return type QuerySet

`add_acknowledgements (acknowledgements)`

Add the acknowledgements

`add_root_schema_pset ()`**`static get_details_schema ()`**

Get the publication details schema

`get_details_schema_pset ()`

Get the publication details schema parameter set

`get_form_state_parameter ()`

Get the form state database object

`objectacl`

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.

`remove_draft_status ()`**`retract ()`**

Retract a publication

`revert_to_draft (message=None)`

This method is not currently exposed to users.

In a previous version of the publication workflow, this method was used when proposed publications were rejected by the publication admins.

Anything with the `form_state` parameter can be reverted to draft

```
safe = <tardis.apps.publication_workflow.managers.PublicationManager object>
```

```
set_embargo_release_date(release_date)
```

tardis.apps.publication_workflow.tasks module

```
tardis.apps.publication_workflow.tasks.get_release_date(publication)
```

```
tardis.apps.publication_workflow.tasks.process_embargos()
```

tardis.apps.publication_workflow.urls module

```
tardis.apps.publication_workflow.urls.re_path(route, view,  
name=None, *, Pattern=<class  
'django.urls.resolvers.RegexPattern'>)
```

tardis.apps.publication_workflow.utils module

```
tardis.apps.publication_workflow.utils.get_pub_admin_email_addresses()
```

```
tardis.apps.publication_workflow.utils.send_mail_to_authors(publication,  
subject, message,  
fail_silently=True)
```

tardis.apps.publication_workflow.views module

```
tardis.apps.publication_workflow.views.delete_publication(request,  
experiment_id)
```

Delete the publication draft with the supplied experiment ID

```
tardis.apps.publication_workflow.views.fetch_experiments_and_datasets(request)
```

```
tardis.apps.publication_workflow.views.finalize_publication(request, publication,  
send_email=True)
```

```
tardis.apps.publication_workflow.views.form_view(request)
```

```
tardis.apps.publication_workflow.views.get_draft_publication(user,  
publication_id)
```

```
tardis.apps.publication_workflow.views.get_licenses(request)
```

```
tardis.apps.publication_workflow.views.is_publication(request, experiment_id)
```

Return a JSON response indicating whether the exp ID is a publication

```
tardis.apps.publication_workflow.views.is_publication_draft(request,  
experiment_id)
```

Return a JSON response indicating whether the exp ID is a publication draft

```
tardis.apps.publication_workflow.views.map_form_to_schemas(extraInfo,  
publication)
```

```
tardis.apps.publication_workflow.views.mint_doi_and_deactivate(request, experiment_id)
```

```
tardis.apps.publication_workflow.views.my_publications(request)
```

Show drafts of public data collections, scheduled publications and published data.

```
tardis.apps.publication_workflow.views.process_form(request)
```

```
tardis.apps.publication_workflow.views.retract_publication(request, publication_id)
    Retract the publication with the supplied publication ID

tardis.apps.publication_workflow.views.retrieve_access_list_tokens_json(request, experiment_per_i-ment_id)
    json list of tokens associated with a given experiment

tardis.apps.publication_workflow.views.retrieve_draft_pubs_list(request)
    json list of draft pubs accessible by the current user

tardis.apps.publication_workflow.views.retrieve_released_pubs_list(request)
    json list of released pubs accessible by the current user

tardis.apps.publication_workflow.views.retrieve_retracted_pubs_list(request)
    json list of retracted pubs accessible by the current user

tardis.apps.publication_workflow.views.retrieve_scheduled_pubs_list(request)
    json list of scheduled pubs accessible by the current user

tardis.apps.publication_workflow.views.select_forms(datasets)
    This method was designed to return form templates for the publication form's Extra Information section, depending on a PUBLICATION_FORM_MAPPINGS setting which supported custom mappings between form templates, dataset schemas and publication schemas.

    These custom mappings have been removed for now, leaving only the generic Extra Information form (a textarea for each dataset). This is because the way the Extra Information was updated each time the form was processed required deleting all of the publication draft's metadata (except for the form_state) and repopulating it. This clashes with the My Publications workflow's desire to add metadata (like a DOI) outside of the form.

tardis.apps.publication_workflow.views.set_publication_authors(author_list, publication)

tardis.apps.publication_workflow.views.submit_form(request, form_state, publication)

tardis.apps.publication_workflow.views.tokens(request, experiment_id)

tardis.apps.publication_workflow.views.update_attribution_and_licensing(request, form_state, publication)
    update_attribution_and_licensing(request, form_state, publication)

tardis.apps.publication_workflow.views.update_data_selection(request, form_state, publication)

tardis.apps.publication_workflow.views.update_dataset_basic_info(request, form_state, publication)

tardis.apps.publication_workflow.views.update_extra_info(request, form_state, publication)

tardis.apps.publication_workflow.views.validation_error(error=None)
```

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..


tardis.apps.push_to.migrations.0003_auto_20210216_1322 module
```

```
class tardis.apps.push_to.migrations.0003_auto_20210216_1322.Migration(name,
    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('push_to', '0002_auto_20160518_1953')]
    operations = [CreateModel name='Request', fields=[('id', django.db.models.fields.Aut..


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 remote_hosts: 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.

request_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.

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

```
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, renderer=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': <django.forms.widgets.PasswordInput object>}
    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.Progress(*args, **kwargs)

Bases: django.db.models.base.Model

Files copy progress

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**id**

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

message

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>**request**

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.

request_id**retry**

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

status

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

timestamp

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.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**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.

request_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.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

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

PushTo request

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

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

credential
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.

credential_id
get_next_by_timestamp(*,
    field=<django.db.models.fields.DateTimeField: timestamp>,
    is_next=True, **kwargs)
get_previous_by_timestamp(*,
    field=<django.db.models.fields.DateTimeField: timestamp>,
    is_next=False, **kwargs)

host
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.

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

message
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.

```

object_type

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>
```

progress_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.

timestamp

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

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.complete_request(request_id)`

`tardis.apps.push_to.tasks.make_dirs(sftp_client, dir_list)`

tardis.apps.push_to.urls module

tardis.apps.push_to.utils module

`tardis.apps.push_to.utils.can_copy(sftp, object_type, object_id, path)`

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

`tardis.apps.push_to.utils.is_directory(sftp, 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: HttpResponseRedirect

`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)`

`tardis.apps.push_to.views.verify_redirect(request, redirect_url)`

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', 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.Form

base_fields = {'identifier': <django.forms.fields.URLField object>, 'notes': <django.forms.fields.URLField object>}
declared_fields = {'identifier': <django.forms.fields.URLField object>, 'notes': <django.forms.fields.URLField object>}
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.S3UtilsAppTestCase(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 = 'tardis.apps.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,      com-
                                                pute_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, indent_types=None, time_formatting=None)
    con-
    date-
    time

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.

log_search_event (request, query_text, result_dict)
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'>, <class 'tardis.tardis_portal.models.dataset.Dataset'>]

class Index
    Bases: object

    name = 'dataset'

    settings = {'number_of_replicas': 0, 'number_of_shards': 1}

django = {'model': <class 'tardis.tardis_portal.models.dataset.Data...>
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_portal.models.dataset.Dataset'>]

class Index
    Bases: object

    name = 'experiments'

    settings = {'number_of_replicas': 0, 'number_of_shards': 1}

django = {'model': <class 'tardis.tardis_portal.models.experiment.Experiment'>
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.AutoField>, {'auto_created': true}), ('key', <django.db.models.fields.TextField>, {}), ('dataset', <django.db.models.fields.related.ForeignKey>, {'related_name': 'public_keys', 'to': 'tardis_portal.Dataset'}), ('experiment', <django.db.models.fields.related.ForeignKey>, {'related_name': 'public_keys', 'to': 'tardis_portal.Experiment'})]]
```

Module contents

tardis.apps.sftp.tests package

Submodules

tardis.apps.sftp.tests.test_sftp module

```
class tardis.apps.sftp.tests.test_sftp.SFTPManagementTestCase(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': <django.f
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

tardis.apps.sftp.sftp module

SFTP Server

```
class tardis.apps.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.MySFTPHandle(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.MySFTPRequestHandler(request, client_address, server)
    Bases: socketserver.BaseRequestHandler

    auth_timeout = 60
    handle()
    handle_timeout()
    setup()
    timeout = 60
```

```
class tardis.apps.sftp.MySFTPServer(*args, **kwargs)
    Bases: paramiko.sftp_server.SFTPServer
```

override SFTPServer to provide channel information to the SFTP subsystem

```
class tardis.apps.sftp.MySFTPServerInterface(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} |{or error code}

class tardis.apps.sftp.sftp.MyTSFTPTCPServer(address, host_key, RequestHandlerClass=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_pty_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)
```

Keeping this to prevent to avoid issues social auth

```
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.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.add_ssh_keys_menu_item']`

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.apps_default_settings module

tardis.default_settings.auth module

`tardis.default_settings.auth.AUTOGENERATE_API_KEY = False`

Generate a tastypie API key with user post_save (`tardis/tardis_portal/models/hooks.py`)

`tardis.default_settings.auth.REGISTRATION_OPEN = True`

Enable/disable the self-registration link and form in the UI. Note - this does not actually disable the URL endpoints for registration. You must also remove the `registration` app from `INSTALLED_APPS` to disable registration.

tardis.default_settings.caches module

`tardis.default_settings.caches.CACHES = {'celery-locks': {'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache', 'LOCATION': '127.0.0.1:11211'}}`

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.downloads.PROXY_DOWNLOADS = False
```

Enable proxying of downloads to NGINX (or another web server)

```
tardis.default_settings.downloads.PROXY_DOWNLOAD_PREFIXES = {}
```

The PROXY_DOWNLOAD_PREFIXES dictionary describes the mapping between each directory prefix and the corresponding internal URL prefix:

```
PROXY_DOWNLOAD_PREFIXES = { '/srv/': '/protected/' }
```

For downloads not handled by DOWNLOAD_URI_TEMPLATES, an alternative to streaming the downloads from MyTardis is to proxy the download request to NGINX using an X-Accel-Redirect header. You can choose a directory or mountpoint prefix e.g. '/srv/' or '/mnt/' and have MyTardis redirect all download requests to NGINX for files within that directory.

For this to work, the NGINX worker processes need to be able to read the data directories. One way to do this is to configure NGINX to run as the mytardis user, by changing the following in /etc/nginx/nginx.conf:

```
# user www-data; user mytardis
```

and restarting the NGINX service.

When changing the NGINX user, you also need to delete or change ownership of files in NGINX's proxy_temp_path (usually /var/lib/nginx/proxy/).

The NGINX location configuration for proxied downloads could look like this:

```
location /protected/ { internal; alias /srv/;
```

which corresponds to the following PROXY_DOWNLOAD_PREFIXES setting:

```
PROXY_DOWNLOAD_PREFIXES = { '/srv/': '/protected/' }
```

```
tardis.default_settings.downloads.RECALL_URI_TEMPLATES = {}
```

When a file recall (from tape/archive) is requested, MyTardis can direct the request to an API endpoint provided by the tardis.apps.hsm app, or to another API endpoint which can take a DataFileObject ID and trigger a recall for the corresponding file on the HSM (Hierarchical Storage Management) system.

For example,

```
RECALL_URI_TEMPLATES = { 'tardis.apps.hsm.storage.HsmFileSystemStorage':  
    '/api/v1/hsm_replica/{dfo_id}/recall/'  
}
```

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', 'block
```

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.graphql module

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_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.ACIAwareBackend
    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(nlz) 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 search 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,      experi-  
                           ment_id)
```

```
tardis.tardis_portal.auth.decorators.has_experiment_access(request,      experi-  
                           ment_id)
```

```
tardis.tardis_portal.auth.decorators.has_experiment_download_access(request,  
                           experi-  
                           ment_id)
```

```
tardis.tardis_portal.auth.decorators.has_experiment_ownership(request,      experi-  
                           ment_id)
```

```
tardis.tardis_portal.auth.decorators.has_experiment_write(request,      experi-  
                           ment_id)
```

```
tardis.tardis_portal.auth.decorators.has_read_or_owner_ACL(request,      experi-  
                           ment_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 return 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,      experi-  
                           ment_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

    getUserId(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.graphql package

Submodules

tardis.tardis_portal.graphql.datafile module

class *tardis.tardis_portal.graphql.datafile.CreateDataFile* (**args*, ***kwargs*)
Bases: *graphene_django_plus.mutations.ModelCreateMutation*

Input

alias of *graphene.relay.mutation.CreateDataFileInput*

class *tardis.tardis_portal.graphql.datafile.DataFileType* (**args*, ***kwargs*)
Bases: *graphene_django_plus.types.ModelType*

pk = <*graphene.types.scalars.Int* object>

```
class tardis.tardis_portal.graphql.datafile.DataTypeFilter(data=None,
    query-
    set=None,      *,
    request=None,
    prefix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object

    fields = {'created_time': ['lte', 'gte'], 'directory': ['exact', 'contains'], 'fi
    model
        alias of tardis.tardis_portal.models.datafile.DataFile

    base_filters = {'created_time__gte': <django_filters.filters.DateTimeFilter object>,
    declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.datafile.UpdateDataFile(*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateDataFileInput

class tardis.tardis_portal.graphql.datafile.UploadDataFile(*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.UploadDataFileInput

classmethod after_save(info, instance, cleaned_input=None)
    Perform “after save” operations.

    Override this to perform any operation on the instance after its .save() method is called.

classmethod clean_input(info, instance, data)
    Clear and normalize the input data.
```

tardis.tardis_portal.graphql.datafileobject module

```
class tardis.tardis_portal.graphql.datafileobject.CreateDataFileObject(*args,
    **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateDataFileObjectInput

class tardis.tardis_portal.graphql.datafileobject.DataFileObjectType(*args,
    **kwargs)
Bases: graphene_django_plus.types.ModelType

    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.datafileobject.DataFileObjectTypeFilter(data=None,
    query-
    set=None,      *,
    request=None,
    pre-
    fix=None)
```

```
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object

        fields = {'created_time': ['lte', 'gte'], 'id': ['exact'], 'verified': ['exact']}
        model
            alias of tardis.tardis_portal.models.datafile.DataFileObject
        base_filters = {'created_time__gte': <django_filters.filters.DateTimeFilter object>,
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.datafileobject.UpdateDataFileObject(*args,
                                                               **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateDataFileObjectInput
```

tardis.tardis_portal.graphql.dataset module

```
class tardis.tardis_portal.graphql.dataset.CreateDataset(*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateDatasetInput

class tardis.tardis_portal.graphql.dataset.DatasetType(*args, **kwargs)
Bases: graphene_django_plus.types.ModelType

    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.dataset.DatasetTypeFilter(data=None, query-
                                                               set=None, *,
                                                               request=None,
                                                               prefix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object

        fields = {'created_time': ['lte', 'gte'], 'description': ['exact', 'contains']}
        model
            alias of tardis.tardis_portal.models.dataset.Dataset
        base_filters = {'created_time__gte': <django_filters.filters.DateTimeFilter object>,
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.dataset.UpdateDataset(*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateDatasetInput
```

tardis.tardis_portal.graphql.experiment module

```
class tardis.tardis_portal.graphql.experiment.CreateExperiment (*args,
                                                               **kwargs)
    Bases: graphene_django_plus.mutations.ModelCreateMutation

    Input
        alias of graphene.relay.mutation.CreateExperimentInput

    classmethod after_save (info, instance, cleaned_input=None)
        Perform “after save” operations.

        Override this to perform any operation on the instance after its .save() method is called.

    classmethod before_save (info, instance, cleaned_input=None)
        Perform “before save” operations.

        Override this to perform any operation on the instance before its .save() method is called.

    classmethod clean_instance (instance, clean_input)
        Validate the instance by calling its .full_clean() method.

class tardis.tardis_portal.graphql.experiment.CreateExperimentAuthor (*args,
                                                                     **kwargs)
    Bases: graphene_django_plus.mutations.ModelCreateMutation

    Input
        alias of graphene.relay.mutation.CreateExperimentAuthorInput

class tardis.tardis_portal.graphql.experiment.ExperimentAuthorType (*args,
                                                                     **kwargs)
    Bases: graphene_django_plus.types.ModelType

    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.experiment.ExperimentAuthorTypeFilter (data=None,
                                                                           query-
                                                                           set=None,
                                                                           *;
                                                                           re-
                                                                           quest=None,
                                                                           pre-
                                                                           fix=None)
    Bases: django_filters.filterset.FilterSet

    class Meta
        Bases: object

        fields = {'author': ['exact', 'contains'], 'id': ['exact']}
        model
            alias of tardis.tardis_portal.models.experiment.ExperimentAuthor

        base_filters = {'author': <django_filters.filters.CharFilter object>, 'author__contains': <django_filters.filters.ContainsFilter object>}
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.experiment.ExperimentType (*args, **kwargs)
    Bases: graphene_django_plus.types.ModelType

    pk = <graphene.types.scalars.Int object>
```

```

class tardis.tardis_portal.graphql.experiment.ExperimentTypeFilter (data=None,
query-
set=None,
*, re-
quest=None,
pre-
fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object

        fields = {'created_time': ['lte', 'gte'], 'id': ['exact'], 'title': ['exact', 'co']}
        model
            alias of tardis.tardis_portal.models.experiment.Experiment
        base_filters = {'created_time__gte': <django_filters.filters.DateTimeFilter object>,
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.experiment.UpdateExperiment (*args,
                                                               **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateExperimentInput

class tardis.tardis_portal.graphql.experiment.UpdateExperimentAuthor (*args,
                                                               **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateExperimentAuthorInput

```

tardis.tardis_portal.graphql.facility module

```

class tardis.tardis_portal.graphql.facility.CreateFacility (*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateFacilityInput

class tardis.tardis_portal.graphql.facility.FacilityType (*args, **kwargs)
Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.facility.FacilityTypeFilter (data=None,
                                                               query-
set=None, *,
request=None,
prefix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object

        fields = {'created_time': ['lte', 'gte'], 'id': ['exact'], 'name': ['exact', 'co']}
        model
            alias of tardis.tardis_portal.models.facility.Facility

```

```
base_filters = {'created_time__gte': <django_filters.filters.DateTimeFilter object>,
declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.facility.UpdateFacility (*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateFacilityInput
```

tardis.tardis_portal.graphql.group module

```
class tardis.tardis_portal.graphql.group.CreateGroup (*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateGroupInput

class tardis.tardis_portal.graphql.group.GroupType (*args, **kwargs)
Bases: graphene_django_plus.types.ModelType

pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.group.GroupTypeFilter (data=None,      query-
                                                               set=None,      *,      re-
                                                               quest=None,      pre-
                                                               fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object

    fields = {'id': ['exact'], 'name': ['exact', 'contains']}
    model
        alias of django.contrib.auth.models.Group

base_filters = {'id': <django_filters.filters.NumberFilter object>, 'name': <django_}
declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.group.UpdateGroup (*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateGroupInput
```

tardis.tardis_portal.graphql.instrument module

```
class tardis.tardis_portal.graphql.instrument.CreateInstrument (*args,
                                                               **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateInstrumentInput

class tardis.tardis_portal.graphql.instrument.InstrumentType (*args, **kwargs)
Bases: graphene_django_plus.types.ModelType

pk = <graphene.types.scalars.Int object>
```

```

class tardis.tardis_portal.graphql.instrument.InstrumentTypeFilter(data=None,
    query-
    set=None,
    *, re-
    quest=None,
    pre-
    fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object

        fields = {'created_time': ['lte', 'gte'], 'id': ['exact'], 'name': ['exact', 'co']}
        model
            alias of tardis.tardis_portal.models.instrument.Instrument
        base_filters = {'created_time__gte': <django_filters.filters.DateTimeFilter object>,
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.instrument.UpdateInstrument(*args,
    **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateInstrumentInput

```

tardis.tardis_portal.graphql.parameters module

```

class tardis.tardis_portal.graphql.parameters.CreateDatafileParameter(*args,
    **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateDatafileParameterInput

class tardis.tardis_portal.graphql.parameters.CreateDatafileParameterSet(*args,
    **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateDatafileParameterSetInput

class tardis.tardis_portal.graphql.parameters.CreateDatasetParameter(*args,
    **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateDatasetParameterInput

class tardis.tardis_portal.graphql.parameters.CreateDatasetParameterSet(*args,
    **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateDatasetParameterSetInput

class tardis.tardis_portal.graphql.parameters.CreateExperimentParameter(*args,
    **kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

```

```
Input
    alias of graphene.relay.mutation.CreateExperimentParameterInput

class tardis.tardis_portal.graphql.parameters.CreateExperimentParameterSet (*args,
                                         **kwargs)
    Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateExperimentParameterSetInput

class tardis.tardis_portal.graphql.parameters.CreateParameterName (*args,
                                         **kwargs)
    Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateParameterNameInput

class tardis.tardis_portal.graphql.parameters.CreateSchema (*args, **kwargs)
    Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateSchemaInput

class tardis.tardis_portal.graphql.parameters.DatafileParameterSetType (*args,
                                         **kwargs)
    Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.parameters.DatafileParameterSetTypeFilter (data=None,
                                         query-
                                         set=None,
                                         *,
                                         re-
                                         quest=None,
                                         pre-
                                         fix=None)
    Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object
    fields = {}
    model
        alias of tardis.tardis_portal.models.parameters.DatafileParameterSet
    base_filters = {'order_by': <django_filters.filters.OrderingFilter object>}
    declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.parameters.DatafileParameterType (*args,
                                         **kwargs)
    Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>
```

```

class tardis.tardis_portal.graphql.parameters.DatafileParameterTypeFilter (data=None,
query-
set=None,
*,
re-
quest=None,
pre-
fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object
        fields = {}
        model
            alias of tardis.tardis_portal.models.parameters.DatafileParameter
        base_filters = {'order_by': <django_filters.filters.OrderingFilter object>}
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.parameters.DatasetParameterSetType (*args,
**kwargs)
Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.parameters.DatasetParameterSetTypeFilter (data=None,
query-
set=None,
*,
re-
quest=None,
pre-
fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object
        fields = {}
        model
            alias of tardis.tardis_portal.models.parameters.DatasetParameterSet
        base_filters = {'order_by': <django_filters.filters.OrderingFilter object>}
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.parameters.DatasetParameterType (*args,
**kwargs)
Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>

```

```
class tardis.tardis_portal.graphql.parameters.DatasetParameterTypeFilter (data=None,
    query-
    set=None,
    *,
    re-
    quest=None,
    pre-
    fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object
    fields = {}
    model
        alias of tardis.tardis_portal.models.parameters.DatasetParameter
    base_filters = {'order_by': <django_filters.filters.OrderingFilter object>}
    declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.parameters.ExperimentParameterSetType (*args,
    **kwargs)
Bases: graphene_django_plus.types.ModelType
pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.parameters.ExperimentParameterSetTypeFilter (data=None,
    query-
    set=None,
    *,
    re-
    quest=None,
    pre-
    fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object
    fields = {}
    model
        alias of tardis.tardis_portal.models.parameters.ExperimentParameterSet
    base_filters = {'order_by': <django_filters.filters.OrderingFilter object>}
    declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.parameters.ExperimentParameterType (*args,
    **kwargs)
Bases: graphene_django_plus.types.ModelType
pk = <graphene.types.scalars.Int object>
```

```

class tardis.tardis_portal.graphql.parameters.ExperimentParameterTypeFilter (data=None,
query-
set=None,
*,
re-
quest=None,
pre-
fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object
        fields = {}
        model
            alias of tardis.tardis_portal.models.parameters.ExperimentParameter
        base_filters = {'order_by': <django_filters.filters.OrderingFilter object>}
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.parameters.ParameterNameType (*args,
                                                               **kwargs)
Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.parameters.ParameterNameTypeFilter (data=None,
query-
set=None,
*, re-
quest=None,
pre-
fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object
        fields = {'id': ['exact'], 'name': ['exact', 'contains'], 'schema': ['exact']}
        model
            alias of tardis.tardis_portal.models.parameters.ParameterName
        base_filters = {'id': <django_filters.filters.NumberFilter object>, 'name': <django_...>}
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.parameters.TardisSchemaType (*args,
                                                               **kwargs)
Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.parameters.TardisSchemaTypeFilter (data=None,
query-
set=None,
*, re-
quest=None,
pre-
fix=None)
Bases: django_filters.filterset.FilterSet

```

```
class Meta
    Bases: object

    fields = {'id': ['exact'], 'name': ['exact', 'contains'], 'namespace': ['exact'],
              model
                  alias of tardis.tardis_portal.models.parameters.Schema

    base_filters = {'id': <django_filters.filters.NumberFilter object>, 'name': <django_
                   filters.filters.CharFilter object>}
    declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.parameters.UpdateDatafileParameter(*args,
                                                                     **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateDatafileParameterInput

class tardis.tardis_portal.graphql.parameters.UpdateDatafileParameterSet(*args,
                                                                       **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateDatafileParameterSetInput

class tardis.tardis_portal.graphql.parameters.UpdateDatasetParameter(*args,
                                                                     **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateDatasetParameterInput

class tardis.tardis_portal.graphql.parameters.UpdateDatasetParameterSet(*args,
                                                                       **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateDatasetParameterSetInput

class tardis.tardis_portal.graphql.parameters.UpdateExperimentParameter(*args,
                                                                       **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateExperimentParameterInput

class tardis.tardis_portal.graphql.parameters.UpdateExperimentParameterSet(*args,
                                                                           **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateExperimentParameterSetInput

class tardis.tardis_portal.graphql.parameters.UpdateParameterName(*args,
                                                                **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateParameterNameInput

class tardis.tardis_portal.graphql.parameters.UpdateSchema(*args, **kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation
```

Input

alias of graphene.relay.mutation.UpdateSchemaInput

tardis.tardis_portal.graphql.schema module

```
class tardis.tardis_portal.graphql.schema.tardisMutation(*args, **kwargs)
Bases: graphene.types.objecttype.ObjectType
    api_key_sign_in = <graphene.types.field.Field object>
    create_datafile = <graphene.types.field.Field object>
    create_datafileobject = <graphene.types.field.Field object>
    create_datafileparameter = <graphene.types.field.Field object>
    create_datafileparameterset = <graphene.types.field.Field object>
    create_dataset = <graphene.types.field.Field object>
    create_datasetparameter = <graphene.types.field.Field object>
    create_datasetparameterset = <graphene.types.field.Field object>
    create_experiment = <graphene.types.field.Field object>
    create_experimentauthor = <graphene.types.field.Field object>
    create_experimentparameter = <graphene.types.field.Field object>
    create_experimentparameterset = <graphene.types.field.Field object>
    create_facility = <graphene.types.field.Field object>
    create_group = <graphene.types.field.Field object>
    create_instrument = <graphene.types.field.Field object>
    create_parametername = <graphene.types.field.Field object>
    create_schema = <graphene.types.field.Field object>
    create_storagebox = <graphene.types.field.Field object>
    create_storageboxattribute = <graphene.types.field.Field object>
    create_storageboxoption = <graphene.types.field.Field object>
    refresh_token = <graphene.types.field.Field object>
    revoke_token = <graphene.types.field.Field object>
    update_datafile = <graphene.types.field.Field object>
    update_datafileobject = <graphene.types.field.Field object>
    update_datafileparameter = <graphene.types.field.Field object>
    update_datafileparameterset = <graphene.types.field.Field object>
    update_dataset = <graphene.types.field.Field object>
    update_datasetparameter = <graphene.types.field.Field object>
    update_datasetparameterset = <graphene.types.field.Field object>
    update_experiment = <graphene.types.field.Field object>
```

```
update_experimentauthor = <graphene.types.field.Field object>
update_experimentparameter = <graphene.types.field.Field object>
update_experimentparameterset = <graphene.types.field.Field object>
update_facility = <graphene.types.field.Field object>
update_group = <graphene.types.field.Field object>
update_instrument = <graphene.types.field.Field object>
update_parametername = <graphene.types.field.Field object>
update_schema = <graphene.types.field.Field object>
update_storagebox = <graphene.types.field.Field object>
update_storageboxattribute = <graphene.types.field.Field object>
update_storageboxoption = <graphene.types.field.Field object>
upload_datafile = <graphene.types.field.Field object>
user_sign_in = <graphene.types.field.Field object>
verify_token = <graphene.types.field.Field object>

class tardis.tardis_portal.graphql.schema.tardisQuery(*args, **kwargs)
Bases: graphene.types.objecttype.ObjectType

datafileobjects = <graphene_django.filter.fields.DjangoFilterConnectionField object>
datafiles = <graphene_django.filter.fields.DjangoFilterConnectionField object>
datasets = <graphene_django.filter.fields.DjangoFilterConnectionField object>
experiments = <graphene_django.filter.fields.DjangoFilterConnectionField object>
facilities = <graphene_django.filter.fields.DjangoFilterConnectionField object>
groups = <graphene_django.filter.fields.DjangoFilterConnectionField object>
instruments = <graphene_django.filter.fields.DjangoFilterConnectionField object>
parameternames = <graphene_django.filter.fields.DjangoFilterConnectionField object>
resolve_datafileobjects(info, **kwargs)
resolve_datafiles(info, **kwargs)
resolve_datasets(info, **kwargs)
resolve_experiments(info, **kwargs)
resolve_facilities(info, **kwargs)
resolve_groups(info, **kwargs)
resolve_instruments(info, **kwargs)
resolve_parameternames(info, **kwargs)
resolve_schemas(info, **kwargs)
resolve_storageboxattributes(info, **kwargs)
resolve_storageboxes(info, **kwargs)
resolve_storageboxoptions(info, **kwargs)
```

```

resolve_user(info, **kwargs)
schemas = <graphene_django.filter.fields.DjangoFilterConnectionField object>
storageboxattributes = <graphene_django.filter.fields.DjangoFilterConnectionField object>
storageboxes = <graphene_django.filter.fields.DjangoFilterConnectionField object>
storageboxoptions = <graphene_django.filter.fields.DjangoFilterConnectionField object>
user = <graphene.types.field.Field object>

```

tardis.tardis_portal.graphql.storagebox module

```

class tardis.tardis_portal.graphql.storagebox.CreateStorageBox(*args,
**kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateStorageBoxInput

class tardis.tardis_portal.graphql.storagebox.StorageBoxType(*args, **kwargs)
Bases: graphene_django_plus.types.ModelType

    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.storagebox.StorageBoxTypeFilter(data=None,
query-
set=None,
*,      re-
quest=None,
pre-
fix=None)
Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object

        fields = {'id': ['exact'], 'name': ['exact', 'contains']}
        model
            alias of tardis.tardis_portal.models.storage.StorageBox

        base_filters = {'id': <django_filters.filters.NumberFilter object>, 'name': <django_>}
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.storagebox.UpdateStorageBox(*args,
**kwargs)
Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateStorageBoxInput

```

tardis.tardis_portal.graphql.storageboxattribute module

```

class tardis.tardis_portal.graphql.storageboxattribute.CreateStorageBoxAttribute(*args,
**kwargs)
Bases: graphene_django_plus.mutations.ModelCreateMutation

```

Input

alias of graphene.relay.mutation.CreateStorageBoxAttributeInput

```
class tardis.tardis_portal.graphql.storageboxattribute.StorageBoxAttributeType(*args,
                                                                           **kwargs)
    Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.storageboxattribute.StorageBoxAttributeTypeFilter(data=None,
                                                                                      query_set=None,
                                                                                      *, re-
                                                                                      quest=None,
                                                                                      pre-
                                                                                      fix=None)
    Bases: django_filters.filterset.FilterSet

class Meta
    Bases: object
    fields = {'id': ['exact'], 'key': ['exact', 'contains'], 'value': ['exact', 'co-
        mpare']}
    model
        alias of tardis.tardis_portal.models.storage.StorageBoxAttribute
    base_filters = {'id': <django_filters.filters.NumberFilter object>, 'key': <django_fi-
        lters.filters.CharFilter object>}
    declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}

class tardis.tardis_portal.graphql.storageboxattribute.UpdateStorageBoxAttribute(*args,
                                                                                 **kwargs)
    Bases: graphene_django_plus.mutations.ModelUpdateMutation

Input
    alias of graphene.relay.mutation.UpdateStorageBoxAttributeInput
```

tardis.tardis_portal.graphql.storageboxoption module

```
class tardis.tardis_portal.graphql.storageboxoption.CreateStorageBoxOption(*args,
                                                                           **kwargs)
    Bases: graphene_django_plus.mutations.ModelCreateMutation

Input
    alias of graphene.relay.mutation.CreateStorageBoxOptionInput

class tardis.tardis_portal.graphql.storageboxoption.StorageBoxOptionType(*args,
                                                                           **kwargs)
    Bases: graphene_django_plus.types.ModelType
    pk = <graphene.types.scalars.Int object>

class tardis.tardis_portal.graphql.storageboxoption.StorageBoxOptionTypeFilter(data=None,
                                                                                 query_set=None,
                                                                                 *, re-
                                                                                 quest=None,
                                                                                 pre-
                                                                                 fix=None)
```

```
Bases: django_filters.filterset.FilterSet
class Meta
    Bases: object
        fields = {'id': ['exact'], 'key': ['exact', 'contains'], 'value': ['exact', 'contains']}
        model
            alias of tardis.tardis_portal.models.storage.StorageBoxOption
        base_filters = {'id': <django_filters.filters.NumberFilter object>, 'key': <django_filters.filters.CharFilter object>}
        declared_filters = {'order_by': <django_filters.filters.OrderingFilter object>}
class tardis.tardis_portal.graphql.storageboxoption.UpdateStorageBoxOption (*args, **kwargs)
    Bases: graphene_django_plus.mutations.ModelUpdateMutation
    Input
        alias of graphene.relay.mutation.UpdateStorageBoxOptionInput
```

tardis.tardis_portal.graphql.user module

```
class tardis.tardis_portal.graphql.user.ApiSignIn (*args, **kwargs)
    Bases: graphene.types.mutation.Mutation
    class Arguments
        Bases: object
            input = <tardis.tardis_portal.graphql.user.ApiSignInInput object>
    mutate (info, input=None)
    token = <graphene.types.scalars.String object>
    user = <graphene.types.field.Field object>
class tardis.tardis_portal.graphql.user.ApiSignInInput (*args, **kwargs)
    Bases: graphene.types.inputobjecttype.InputObjectType
        key = <graphene.types.scalars.String object>
class tardis.tardis_portal.graphql.user.UserSignIn (*args, **kwargs)
    Bases: graphql_jwt.relay.JSONWebTokenMutation
    Input
        alias of graphene.relay.mutation.UserSignInInput
    classmethod resolve (self, info, **kwargs)
        user = <graphene.types.field.Field object>
class tardis.tardis_portal.graphql.user.UserType (*args, **kwargs)
    Bases: graphene_django_plus.types.ModelType
```

tardis.tardis_portal.graphql.utils module

```
class tardis.tardis_portal.graphql.utils.ExtendedConnection (*args, **kwargs)
    Bases: graphene.relay.connection.Connection
        edge_count = <graphene.types.scalars.Int object>
        resolve_edge_count (info, **kwargs)
```

```
    resolve_total_count(info, **kwargs)
        total_count = <graphene.types.scalars.Int object>
tardis.tardis_portal.graphql.utils.get_accessible_datafiles(user)
tardis.tardis_portal.graphql.utils.get_accessible_experiments(user)
```

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.rmxperiment 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.

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

```
class tardis.tardis_portal.management.commands.rmxperiment.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(name='id', primary_key=True, db_index=True, blank=True),), ('name', <django.db.models.fields.CharField(max_length=255, db_index=True, blank=True),), ('size', <django.db.models.fields.IntegerField(db_index=True, blank=True),), ('modified_time', <django.db.models.fields.DateTimeField(db_index=True, blank=True),), ('created_time', <django.db.models.fields.DateTimeField(db_index=True, blank=True),), ('content_type', <django.db.models.fields.related.ForeignKey(to='contenttypes.ContentType', db_index=True, blank=True),), ('content_object', <django.db.models.fields.related.GenericForeignKey(db_index=True, related_name='content_type_set_for_file'),)], db_table='tardis_portal_DataFile')]
```

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(name='id', primary_key=True, db_index=True, blank=True),), ('name', <django.db.models.fields.CharField(max_length=255, db_index=True, blank=True),), ('size', <django.db.models.fields.IntegerField(db_index=True, blank=True),), ('modified_time', <django.db.models.fields.DateTimeField(db_index=True, blank=True),), ('created_time', <django.db.models.fields.DateTimeField(db_index=True, blank=True),), ('content_type', <django.db.models.fields.related.ForeignKey(to='contenttypes.ContentType', db_index=True, blank=True),), ('content_object', <django.db.models.fields.related.GenericForeignKey(db_index=True, related_name='content_type_set_for_file'),)], db_table='tardis_portal_DataFile')]
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 = True
```

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(max_length=255, db_index=True, blank=True),>, <AlterField model_name='parameter', name='value', field=<django.db.models.fields.CharField(max_length=255, db_index=True, blank=True),>]
```

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(max_length=255, db_index=True, blank=True),>, <AlterField model_name='parameter', name='value_type', field=<django.db.models.fields.CharField(max_length=255, db_index=True, blank=True),>]
```

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(max_length=255, db_index=True, blank=True),>]
```

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>()]
tardis.tardis_portal.migrations.0005_datafile_add_size_int_column.cast_string_to_integer(app_label)
sc
```

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'>, <AlterField model_name='datafile', name='size', field=<django.db.models.fields.IntegerField>()]
sc
```

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.fields.CharField>()]
sc
```

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 = []
sc
```

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_id')}>()]
sc
```

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.fields.CharField>()]
sc
```

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.m...]
```

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.m...]
```

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',), 've...]
```

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.m...]
```

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.m...]
```

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.m
```

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 ForwardOneToOneDescriptor subclass) relation.

In the example:

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

Child.parent is a ForwardManyToOneDescriptor instance.

group_id

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

```
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 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
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`

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()

```
get_authenticationMethod_display(*, field=<django.db.models.fields.CharField: authenticationMethod>)
```

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

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`

`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,      in-
                                                               stance, created,
                                                               **kwargs)

tardis.tardis_portal.models.access_control.get_auth_method_choices()
```

`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

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.

download_url

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_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()`

`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>**progress_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.

recall_url

Get a URL to request a recall from tape

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.

view_url

```
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 book 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**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**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,  
                                         compute_md5=True,  
                                         compute_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.

getParametersSets (*schemaType=None*)

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

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>

online_files_count

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.mytardis.org/schemas/publication/details/'
    PUBLICATION_DRAFT_SCHEMA = 'http://www.mytardis.org/schemas/publication/draft/'
    PUBLICATION_RETRACTED_SCHEMA = 'http://www.mytardis.org/schemas/publication/retracted/'
    PUBLICATION_SCHEMA_ROOT = 'http://www.mytardis.org/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

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()
is_retracted_publication()

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`

`locked`

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

`objectacl`

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, institution, 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**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

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**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,  
                                         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 = '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

```
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

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

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

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

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

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

```
class tardis.tardis_portal.models.parameters.InstrumentParameter(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 = '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

```
class tardis.tardis_portal.models.parameters.InstrumentParameterSet(id,
    schema,
    instrument)
```

Bases: *tardis.tardis_portal.models.parameters.ParameterSet*

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

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

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**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**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

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

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

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 transferred 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

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 DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`PICKLE = 'pickle'`

`STRING = 'string'`

```
TYPE_CHOICES = (('string', 'String value'), ('pickle', 'Pickled value'))  
get_value_type_display(*, 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  
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
```

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

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, experiment)
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**(nodelist, 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.ACIAuthorizationTest(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.DatafileParameterRes
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase
```

```
class tardis.tardis_portal.tests.api.test_datafile_metadata_resources.DatafileParameterSet
    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_TASK_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.DatasetParameterResource
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.api.test_dataset_metadata_resources.DatasetParameterSetResource
    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='r')
    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.ExperimentParameterTest
    Bases: tardis.tardis_portal.tests.api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.api.test_experiment_metadata_resources.ExperimentParameterSetTest
    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='GET')
    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='GET')
    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 (method...
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.graphql package

Submodules

tardis.tardis_portal.tests.graphql.test_auth module

```
class tardis.tardis_portal.tests.graphql.test_auth.userSignInTestCase(methodName='runTest')
Bases: graphene_django.utils.testing.GraphQLTestCase
```

```
GRAPHQL_SCHEMA = <graphene.types.schema.Schema object>
```

`setUp()`

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

`tearDown()`

Hook method for deconstructing the test fixture after testing it.

`test_user_sign_in()`

tardis.tardis_portal.tests.graphql.test_facilities module

```
class tardis.tardis_portal.tests.graphql.test_facilities.createFacilityTestCase(methodName='')
Bases: tardis.tardis_portal.tests.graphql.test_facilities.facilitiesTestCase
```

`setUp()`

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

`test_create_facility()`

```
class tardis.tardis_portal.tests.graphql.test_facilities.facilitiesTestCase(methodName='runTe')
Bases: graphql_jwt.testcases.JSONWebTokenTestCase
```

setUp()

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

tearDown()

Hook method for deconstructing the test fixture after testing it.

class tardis.tardis_portal.tests.graphql.test_facilities.**getFacilitiesTestCase** (*methodName*=’runTest’)
Bases: *tardis.tardis_portal.tests.graphql.test_facilities.facilitiesTestCase*

setUp()

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

test_get_facilities()

class tardis.tardis_portal.tests.graphql.test_facilities.**updateFacilityTestCase** (*methodName*=’runTest’)
Bases: *tardis.tardis_portal.tests.graphql.test_facilities.facilitiesTestCase*

setUp()

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

test_update_facility()

tardis.tardis_portal.tests.graphql.test_user module

class tardis.tardis_portal.tests.graphql.test_user.**userSignInTestCase** (*methodName*=’runTest’)
Bases: *graphql_jwt.testcases.JSONWebTokenTestCase*

setUp()

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

tearDown()

Hook method for deconstructing the test fixture after testing it.

test_user_authenticated()**test_user_not_authenticated()**

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='r'
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 (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_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_Idif module

tardis.tardis_portal.tests.slapd module

Utilities for starting up a test slapd server and talking to it with ldapsearch/ldapadd.

class `tardis.tardis_portal.tests.slapd.Slapd`

Bases: `object`

Controller class for a slapd instance, OpenLDAP's server.

This class creates a temporary data store for slapd, runs it on a private port, and initialises it with a top-level dc and the root user.

When a reference to an instance of this class is lost, the slapd server is shut down.

`PATH_LDAPADD = None`

`PATH_LDAPSEARCH = None`

`PATH_SCHEMA_DIR = '/home/docs/checkouts/readthedocs.org/user_builds/mytardis/checkouts/`

`PATH_SLAUDP = None`

`PATH_SLAPTEST = None`

`PATH_TMPDIR = '/tmp/tmpqq1kto4c'`

`TEST_UTILS_DIR = '/home/docs/checkouts/readthedocs.org/user_builds/mytardis/checkouts/`

classmethod `check_paths()`

Checks that the configured executable paths look valid. If they don't, then logs warning messages (not errors).

configure (`cfg`)

Appends slapd.conf configuration lines to `cfg`. Also re-initializes any backing storage. Feel free to subclass and override this method.

get_address()

get_dn_suffix()

get_root_dn()

get_root_password()

get_tmpdir()

get_url()

ldapadd (`ldif`, `extra_args=[]`)

Runs ldapadd on this slapd instance, passing it the ldif content

ldapsearch (`base=None`, `filter='(objectClass=*)'`, `attrs=[]`, `scope='sub'`, `extra_args=[]`)

restart()

Restarts the slapd server; ERASING previous content. Starts the server even if it isn't already running.

set_debug()

set_dn_suffix (`dn`)

set_port (`port`)

set_root_cn (`cn`)

set_root_password (`pw`)

set_slapd_debug_level (`level`)

```
set_tmpdir(path)
start()
    Starts the slapd server process running, and waits for it to come up.

started()
    This method is called when the LDAP server has started up and is empty. By default, this method adds the two initial objects, the domain object and the root user object.

stop()
    Stops the slapd server, and waits for it to terminate

wait()
    Waits for the slapd process to terminate by itself.

tardis.tardis_portal.tests.slapd.delete_directory_content(path)
tardis.tardis_portal.tests.slapd.find_available_tcp_port(host='127.0.0.1')
tardis.tardis_portal.tests.slapd.mkdirs(path)
    Creates the directory path unless it already exists

tardis.tardis_portal.tests.slapd.quote(s)
    Quotes the " and characters in a string and surrounds with "...""

tardis.tardis_portal.tests.slapd.which(executable)
```

tardis.tardis_portal.tests.test_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_iif module

```
class tardis.tardis_portal.tests.test_iif.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_iif.Level0TestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
```

As per: <http://library.stanford.edu/iiif/image-api/compliance.html>

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

    testCanGetInfoAsJSON()

    testCanGetInfoAsXML()
```

```
    testCanGetOriginalImage()

class tardis.tardis_portal.tests.test_iiif.Level1TestCase (methodName='runTest')
Bases: django.test.testcases.TestCase

As per: http://library.stanford.edu/iiif/image-api/compliance.html

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

testCanGetJpegFormat()
testHandleRegions()
testHandleRotation()
testHandleSizing()

class tardis.tardis_portal.tests.test_iiif.Level2TestCase (methodName='runTest')
Bases: django.test.testcases.TestCase

As per: http://library.stanford.edu/iiif/image-api/compliance.html

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

testCanGetRequiredFormats()
testHandleSizing()
```

tardis.tardis_portal.tests.test_ldap module

```
class tardis.tardis_portal.tests.test_ldap.LDAPErrorTest (methodName='runTest')
Bases: django.test.testcases.TestCase

test_search()

class tardis.tardis_portal.tests.test_ldap.LDAPTest (methodName='runTest')
Bases: django.test.testcases.TestCase

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

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

test_authenticate()
test_getgroupbyid()
test_getgroups()
test_getgroupsforentity()
test_getuserbyid()
test_search()
test_searchgroups()
```

tardis.tardis_portal.tests.test_models module**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 thumbnail 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 moduletest_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='')

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

```
tardis.tardis_portal.tests.urls.groups_view(request)
    Dummy view for remote user tests
```

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,  
                           ex-  
                           per-  
                           i-  
                           ment_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 overriden 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 overriden 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 HttpResponseRedirect

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 overriden 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 facilitytardis.tardis_portal.views.pages.**healthz** (*request*)
returns that the server is alivetardis.tardis_portal.views.pages.**my_data** (*request*)
show owned data with credential-based accesstardis.tardis_portal.views.pages.**public_data** (*request*)
list of public experimentstardis.tardis_portal.views.pages.**shared** (*request*)
show shared data with credential-based accesstardis.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,
                                                 template_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.HttpRequestMethodNotAllowed(*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,
                                                               parentObject=None,
                                                               schema=None))
Bases: object
blank_param = None
delete_all_params()
delete_params(parname)
get_param(parname, value=False)
get_params(parname, value=False)
get_schema()
new_param(parname, value, fullparname=None)
parameters = None
parameterset = None
set_param(parname, value, fullparname=None, example_value=None)
set_param_list(parname, value_list, fullparname=None)
set_params_from_dict(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', 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.DataFileObject

    widgets = {'uri': <django.forms.widgets.TextInput object>}

    base_fields = {'datafile': <django.forms.models.ModelChoiceField object>, 'last_verified': None}

    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': None}

    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': <classe

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',
                   'modified']
    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',
                                                               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.storage.StorageBoxAttribute

widgets = {'key': <django.forms.widgets.TextInput object>, 'value': <django.forms.widgets.Textarea object>}

base_fields = {'key': <django.forms.fields.CharField object>, 'storage_box': <django.forms.fields.ModelChoiceField object>}

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, renderer=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, 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, 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.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.fields.CharField object>,
base_fields = {'key': <django.forms.fields.CharField object>, 'storage_box': <django.forms.widgets.TextInput object>},
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.ACIAuthorization
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.ForeignKey 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.DictField 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.DictField 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': ('exact',), 'directory': ('exact',), 'experiments': ('exact',), 'instrument': ('exact',), 'parameters': ('exact',), 'parameterset': ('exact',), 'dataset': ('exact',), 'id': <tastypie.fields.IntegerField object>}

declared_fields = {'experiments': <tastypie.fieldsToManyField object>, 'instrument': <tastypie.fields.ManyToManyField object>, 'parameters': <tastypie.fields.DictField object>, 'parameterset': <tastypie.fields.ForeignKey object>, 'dataset': <tastypie.fields.ForeignKey object>, 'id': <tastypie.fields.IntegerField 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
    :class: django.http.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'), 'experi
        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.field
        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': <tast
        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>}

declared_fields = {'experiment': <tastypie.fields.ForeignKey object>, 'parameters': <tastypie.fields.DictField 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': <tastypie.fields.ForeignKey object>}

declared_fields = {'created_by': <tastypie.fields.ForeignKey object>, 'parameter_sets': <tastypie.fields.DictField 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_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': <tastypie.fields.IntegerField object>}

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.CharField object>}

    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': <tastypie.fields.ForeignKey object>, 'name': <tastypie.fields.CharField object>}

    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': <tastypie.fields.BooleanField object>}
    declared_fields = {'content_object': <tastypie.contrib.contenttypes.fields.GenericForeignKey object>}

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': <tastypie.fields.CharField object>}
    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': <tastypie.f...}
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.f...}
declared_fields = {'schema': <tastypie.fields.ForeignKey object>}

hydrate_schema(bundle)

class tardis.tardis_portal.api.PrettyJSONSerializer(formats=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.f...}
    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.IntegerField 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.CharField object>}

        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.CharField object>}

        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.fieldsToManyField object>, 'description': <tastypie.fields.CharField object>}

        declared_fields = {'attributes': <tastypie.fieldsToManyField object>, 'options': <tastypie.fields.CharField object>}

```

```
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.forms.

declared_fields = {}

media

class tardis.tardis_portal.forms.ExperimentForm(data=None, files=None,
                                                auto_id='%', prefix=None, initial=None, error_class=<class
                                                'django.forms.utils.ErrorList'>, label_suffix=':',
                                                empty_permitted=False, instance=None, extra=0)

Bases: django.forms.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 convience 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 returnd value is dependent on the validity of the nested forms.

        This validity also takes into account forign keys that might be dependent on an unsaved model.

        Returns validity
        Return type bool

    media

    save(commit=True)
        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': <django...}

media

class tardis.tardis_portal.forms.LoginForm(*args, **kwargs)
Bases: django.contrib.auth.forms.AuthenticationForm

base_fields = {'password': <django.forms.fields.CharField object>, 'username': <django...}

declared_fields = {'password': <django.forms.fields.CharField object>, 'username': <django...}

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': <django...}

declared_fields = {}

media

```

```
class tardis.tardis_portal.forms.ManageGroupPermissionsForm(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.Form

base_fields = {'adduser': <django.forms.fields.CharField object>, 'admin': <django.f
declared_fields = {'adduser': <django.forms.fields.CharField object>, 'admin': <django.
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',     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.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.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.forms.fields.CharField object>}
```

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': <django.forms.fields.CharField object>}
```

media

save(profile_callback=None)

```
class tardis.tardis_portal.forms.RightsForm(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 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': <django.forms.fields.CharField object>}
```

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, validators=(), 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.

`widget`

alias of `NoInput`

```
tardis.tardis_portal.forms.createLinkedUserAuthenticationForm(authMethods)
    Create a LinkedUserAuthenticationForm and use the contents of authMethods to the list of options in the drop-
    down menu for authenticationMethod.

tardis.tardis_portal.forms.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 system-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_time=False` don't use the local timezone and consider the offset to UTC to be zero:

```
>>> rfc3339(date, use_system_timezone=False)
'2008-04-02T20:00:00+00:00'
```

`date` must be a a `datetime.datetime`, `datetime.date` or a timestamp as returned by `time.time()`:

```
>>> rfc3339(0, utc=True, use_system_timezone=False)
'1970-01-01T00:00:00Z'
>>> rfc3339(datetime.date(2008, 9, 6), use_system_timezone=False)
'2008-09-06T00:00:00+00:00'
>>> rfc3339('foo bar')
Traceback (most recent call last):
...
TypeError: 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.schema module

`class tardis.schema.Mutation(*args, **kwargs)`

Bases: `tardis.tardis_portal.graphql.schema.tardisMutation`, `graphene.types.objecttype.ObjectType`

`class tardis.schema.Query(*args, **kwargs)`

Bases: `tardis.tardis_portal.graphql.schema.tardisQuery`, `graphene.types.objecttype.ObjectType`

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 MyTardis Hierarchical Storage Management(HSM) App

Overview

The MyTardis Hierarchical Storage Management app allows MyTardis deployments to view online/offline(disk/tape) status for a file on a HSM file storage system. This app also allows users to request recalling individual datafile or a dataset that has been moved to tape.

Usage

To enable the app, include `tardis.apps.hsm` in `settings.INSTALLED_APPS`:

```
INSTALLED_APPS += (
    'tardis.apps.hsm',
)
```

This app will only try to find online/offline status for files backed by `tardis.apps.hsm.storage.HsmFileSystemStorage`.

To configure a `tardis.tardis_portal.models.storage.StorageBox` as a type of HSM file system, update following details for a storage box:

Django storage class as `tardis.apps.hsm.storage.HsmFileSystemStorage`

Also update following Storage Box Attributes

- Key as `remote_path` and value as path to remote HSM filesystem share e.g: `vault-v2.erc.monash.edu/abc/vault`
- Key as `mount` and value as mount point for above remote path e.g: `/srv/abc_vault`

Default settings for this app is available at `tardis.apps.hsm.default_settings`

11.4.3 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.aoipmh.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 *caipmh.error.NoSetHierarchyError* – because set hierarchies are currently not implemented

11.4.4 Publication Forms

Introduction

MyTardis supports a simple, configurable, publication workflow allowing researchers to curate collections of data sets for public release. This workflow represents an alternative to simply setting an experiment to “public access” that encourages researchers to provide a higher standard of metadata and bibliographic information, including authorship and acknowledgements without altering the original experiment. Additional features include embargo support, admin publication approvals, and PDB metadata extraction from the Protein Data Bank. DOI minting is also supported but depends on a third-party DOI minting service. Set up of the minting service is beyond the scope of this document.

The publication workflow is a three-stage process:

1. The user selects data sets to include in the publication, along with a title and description
2. Based on the selected data sets, domain-specific fields are presented
3. Authorship, acknowledgements and embargo information is collected

After stage one is started, the publication remains in draft until it is submitted.

Under the hood, this is achieved by creating a new experiment object that is differentiated from regular experiments by attaching a special publication schema. This new experiment links to the selected data sets, which may be from one or many original experiments. Publication related tasks (e.g. embargo processing) filter based on the publication schema and act accordingly.

Basic setup

A minimum setup requires the `tardis.apps.publication_workflow` app to be added to `INSTALLED_APPS` and the following settings to be defined:

- `PUBLICATION_DATA_ADMIN` set to the user name of the person responsible for research data management
- `PUBLICATION_NOTIFICATION_SENDER_EMAIL` set to the email from which publication related notifications should be sent
- `PUBLICATION_INTRODUCTION` set to an HTML formatted introductory text that is displayed to the user before begins the publication process
- the `apps.publication_workflow.update_publication_records` task should be added to `CELERYBEAT_SCHEDULE` and set to a reasonable interval for embargo processing

To disable the approval workflow and have publications automatically released, `PUBLICATIONS_REQUIRE_APPROVAL` can be set to False (defaults to True).

DOI support

These additional settings are required for DOI minting support:

- `MODC_DOI_ENABLED` set to True to enable support
- `MODC_DOI_API_ID` set to the API id assigned to this app by the minting service
- `MODC_DOI_API_PASSWORD` set to the password corresponding to the API id
- `MODC_DOI_MINT_DEFINITION` set to the WSDL file for DOI minting
- `MODC_DOI_ACTIVATE_DEFINITION` set to the WSDL file for DOI activation
- `MODC_DOI_DEACTIVATE_DEFINITION` set to the WSDL file for DOI deactivation
- `MODC_DOI_ENDPOINT` set to the API SOAP endpoint
- `MODC_DOI_MINT_URL_ROOT` set to the URL root for registered

DOIs, e.g. <http://tardis.example.com/>

Email configuration

There are a number of different notifications that are sent during the publication process. Very basic templates are pre-configured, but may be overridden by defining the `PUBLICATION_EMAIL_MESSAGES` dictionary in the `settings.py` file. Review the `default_settings.py` file for this app to see defaults. The following keys in the dictionary must be defined if

overriding the defaults:

- `PUBLICATION_DATA_ADMIN['requires_authorisation']`
 - Sent to publication admins to authorise/approve a publication for release
 - `{user_name}`: the submitting user's user name
 - `{pub_url}`: the direct link to the publication for review
 - `{approvals_url}`: the link to the admin approvals page
- `PUBLICATION_DATA_ADMIN['awaiting_approval']`
 - Sent to all authors immediately after submission

- *{pub_title}*: the title of the publication
- **PUBLICATION_DATA_ADMIN['approved']**
 - Sent to all authors after the publication has been approved but before being released (e.g. by embargo expiry)
 - *{pub_title}*: the title of the publication
 - *{pub_url}*: the direct link to the publication
- **PUBLICATION_DATA_ADMIN['approved_with_doi']**
 - As above, but contains a DOI
 - *{doi}*: the publication's DOI
- **PUBLICATION_DATA_ADMIN['rejected']**
 - Sent to all authors if the publication is rejected
 - *{pub_title}*: the title of the publication
- **PUBLICATION_DATA_ADMIN['reverted_to_draft']**
 - Sent to all authors when the publication requires amendments by the submitting user
 - *{pub_title}*: the title of the publication
- **PUBLICATION_DATA_ADMIN['released']**
 - Sent to all authors when the publication has been released following approval
 - *{pub_title}*: the title of the publication
- **PUBLICATION_DATA_ADMIN['released_with_doi']**
 - As above, but contains a DOI
 - *{doi}*: the publication's DOI

Domain-specific metadata

Domain-specific metadata is collected in stage two of the publication process. The forms to display are determined by inspecting the attached data set schemas. Schema to form mappings are defined in the PUBLICATION_FORM_MAPPINGS setting, an example of which is shown below:

```
PUBLICATION_FORM_MAPPINGS = [
    {'dataset_schema': r'^http://synchrotron.org.au/mx/',
     'publication_schema': PDB_PUBLICATION_SCHEMA_ROOT,
     'form_template': '/static/publication-form/mx-pdb-template.html'},
    {'dataset_schema': r'^http://synchrotron.org.au/mx/',
     'publication_schema': MX_PUBLICATION_DATASET_SCHEMA,
     'form_template':
         '/static/publication-form/mx-dataset-description-template.html'}]
```

dataset_schema is a regular expression applied to each schema attached to each data set, and if matched, the corresponding *publication_schema* is added to the resulting publication, and its parameters are provided by the form template. Care could be taken in constructing the *dataset_schema* regex keeping in mind that the expression could match one or more data sets.

Constructing the form template HTML files are somewhat challenging; examples are provided in the *static/publication-form* directory. These forms require special syntax that is defined using *AngularJs*. All forms have access to a *formTemplate* variable, which includes the publication schema name (*formTemplate.name*) a list of affected

data sets (*formTemplate.datasets*). Each form will populate *formData.extraInfo[x]* with the user supplied data, where *x* is a unique key.

The forms themselves must be enclosed in a *<tardis-form>* tag, which requires a *my-model* attribute (set to *formData.extraInfo[x]*), and a *schema* (set to *formTemplate.name*). Any form field tags must also include the *tardis-form-field* attribute, in addition to *parameter-name*, which is set to the publication schema's parameter to populate. The value given to *x* as the *formData.extraInfo[x]* key must be unique over all included forms. In our example below, the form and data set indices are concatenated to form this unique key. Each entry in *formData.extraInfo* is given its own parameter set in the final publication metadata.

Here is an example that collects some information for each dataset that matches the *dataset_schema* regex:

```
<div ng-repeat="ds in formTemplate.datasets"
    ng-init="f = $parent.$index; formData.extraInfo[f+'.'+$index]['dataset'] = ds.
    ↪description;">
    <h4>{{ ds.description }}</h4>
    <tardis-form my-model="formData.extraInfo[f+'.'+$index]" schema="formTemplate.
    ↪name">
        <textarea tardis-form-field parameter-name="additional-information"
            rows="3"
            style="width:100%;"
            placeholder="Information about this dataset.">
        </textarea>
    </tardis-form>
</div>
```

The above form works as follows:

1. The code inside the outer *<div>* tag is repeated for each data set, which is provided by *formTemplate.datasets*. The outer *<div>* tag also initialises a form index, *f*, obtained from the parent scope's *\$index* variable, which is used in conjunction with the inner scope's *\$index* variable to form the unique key for *formData.extraInfo*; namely, *formData.extraInfo[f+'.'+\$index]*.
2. *ng-init* in the outer *<div>* tag also saves a ‘dataset’ field, which is included in *formData.extraInfo*. This functions exactly as a hidden HTML input field would.
3. The corresponding dataset name is displayed to the user in the *<h4>* tag
4. A *<tardis-form>* is started, using *formData.extraInfo[f+'.'+\$index]* as its model, linking it to the schema provided by *formTemplate.name*.
5. A *<textarea>* collects a data set description that will be added to the parameter named ‘additional-information’, as defined in the corresponding schema

Please note that it is extremely important to ensure that the *parameter-name* attribute for each form field matches exactly the schema parameter. Any fields that do not match are silently ignored!

11.4.5 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 4.6.0

- #3153 Move static libs from PushTo app to Webpack
- #3115 Download single file via Nginx
- #3114 Update HSM app docs

- #3113 Update Celery settings to a new format
- #3111 Send datafile recall acknowledgement email
- #3091 Google auth user approval fix
- #3089 Events log update (switch to a native JSONField)
- #3028 and #3029 Security fixes as per CodeQL advice
- #3014 Track HSM app recalls with events log
- Dependency updates for Python and JS
- NodeJS has been switched from 10.x to LTS version (14.x)
- Dockerfile for tests has been switched to Ubuntu 20.04

11.5.2 4.5.0

- Support for displaying online/offline files status for file in HSM
- Support for requesting recall of an offline dataset
- Support for requesting recall of an offline datafile
- Updated various Python and JS libraries
- Bugfixes - Fixed #2901 - PushTo related fixes #2910 - Fix redirect to next page - Fixes related to tree view #2742

11.5.3 4.4.0

- User activity logging app
- Updated various Python and JS libraries
- Fix to Google Chrome installation
- Bugfixes - add SSH key

11.5.4 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

11.5.5 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.

11.5.6 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.

11.5.7 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

11.5.8 4.1.3

- Update the https-proxy-agent version in package-lock.json to avoid having “npm install” display “found 1 high severity vulnerability”.

11.5.9 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 pyoai dependency

11.5.10 4.1.1

- Fix Python 3 bug with string encoding in deep download mapper which affected directory names in SFTP interface.

11.5.11 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

11.5.12 4.0.2

- Upgraded Django to 1.11.23
- Upgraded vulnerable dependencies of JS dev dependencies
- Fixed #1844 (remove a hard-coded LDAP attribute)

11.5.13 4.0.1

- Removed anzsrc_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)

11.5.14 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.

11.5.15 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

11.5.16 3.8.1

- Fix regression in Push To app

11.5.17 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 @avrjk
- Added setting that allows datasets ordered by id on the Experiment page.
- Added setting to make sha512 checksums optional.

11.5.18 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

11.5.19 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

11.5.20 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.

11.5.21 3.0 - unreleased

- Twitter Bootstrap
- javascript templates
- backbone.js rendering of datasets
- UI for transferring datasets
- bpython shell
- celery queue

11.5.22 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 user 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 [Russell, Uli]
- Logging framework [Uli]
- Django 1.3

11.5.23 1.07 - 01/06/2010

- Publish to tardis.edu.au interface created, though not implemented, pending legal text

11.5.24 1.06 - 15/03/2010

- Parameter import interface for creation of new parameter/schema definitions
- iPhone Interface

11.5.25 1.05 - 01/03/2010

- Images as parameters supported
- Data / metadata transfer from synchrotron is now ‘threaded’ using asynchronous web service transfers.

11.5.26 1.0 - 01/02/2010

- MyTardis created from existing 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

11.5.27 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

11.5.28 0.1 - 2007

- Federated index (php) running on Apache HTTP Server
- Crystallography data deposition and packaging tools for Fedora Commons (java swing desktop)
- Search Interface via web

CHAPTER 12

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

tardis, 295
tardis.analytics, 89
tardis.analytics.apps, 88
tardis.analytics.ga, 88
tardis.analytics.tracker, 88
tardis.app_config, 293
tardis.apps, 150
tardis.apps.deep_storage_download_mapper, 89
tardis.apps.deep_storage_download_mapper.mapper, 89
tardis.apps.deep_storage_download_mapper.urls, 89
tardis.apps.dl_mapper_df_dir_only, 90
tardis.apps.dl_mapper_df_dir_only.mapper, 89
tardis.apps.dl_mapper_df_dir_only.urls, 90
tardis.apps.eventlog, 93
tardis.apps.eventlog.admin, 90
tardis.apps.eventlog.default_settings, 90
tardis.apps.eventlog.fields, 91
tardis.apps.eventlog.migrations, 90
tardis.apps.eventlog.migrations.0001_initial, 90
tardis.apps.eventlog.models, 91
tardis.apps.eventlog.signals, 93
tardis.apps.eventlog.urls, 93
tardis.apps.eventlog.utils, 93
tardis.apps.filepicker, 94
tardis.apps.filepicker.filepicker_settings, 93
tardis.apps.filepicker.urls, 93
tardis.apps.filepicker.utils, 93
tardis.apps.filepicker.views, 94
tardis.apps.hsm, 99
tardis.apps.hsm.api, 96
tardis.apps.hsm.apps, 97
tardis.apps.hsm.check, 97
tardis.apps.hsm.default_settings, 97
tardis.apps.hsm.email_text, 98
tardis.apps.hsm.exceptions, 98
tardis.apps.hsm.migrations, 94
tardis.apps.hsm.migrations.0001_initial_data, 94
tardis.apps.hsm.storage, 98
tardis.apps.hsm.tasks, 98
tardis.apps.hsm.tests, 96
tardis.apps.hsm.tests.test_api, 94
tardis.apps.hsm.tests.test_app_config, 95
tardis.apps.hsm.tests.test_email_templates, 95
tardis.apps.hsm.tests.test_migrations, 96
tardis.apps.hsm.urls, 98
tardis.apps.hsm.utils, 99
tardis.apps.oaipmh, 105
tardis.apps.oaipmh.models, 103
tardis.apps.oaipmh.provider, 102
tardis.apps.oaipmh.provider.base, 99
tardis.apps.oaipmh.provider.experiment, 101
tardis.apps.oaipmh.server, 104
tardis.apps.oaipmh.tests, 103
tardis.apps.oaipmh.tests.provider, 103
tardis.apps.oaipmh.tests.provider.test_base, 102
tardis.apps.oaipmh.tests.provider.test_experiment, 102
tardis.apps.oaipmh.tests.test_oai, 103
tardis.apps.oaipmh.urls, 105
tardis.apps.oaipmh.views, 105
tardis.apps.openid_migration, 111
tardis.apps.openid_migration.apps, 107
tardis.apps.openid_migration.context_processors, 107

tardis.apps.openid_migration.default_settings.apps.publication_workflow.tests.test_revert_107
tardis.apps.openid_migration.email_text,tardis.apps.publication_workflow.tests.test_tokens,107
tardis.apps.openid_migration.forms,107 tardis.apps.publication_workflow.urls,
tardis.apps.openid_migration.migration, 107 tardis.apps.publication_workflow.utils,
tardis.apps.openid_migration.migrations, 106 tardis.apps.publication_workflow.views,
tardis.apps.openid_migration.migrations.0001_initial, 106 tardis.apps.push_to,130
tardis.apps.openid_migration.models,108 tardis.apps.push_to.apps,120
tardis.apps.openid_migration.tasks,110 tardis.apps.push_to.exceptions,120
tardis.apps.openid_migration.tests,107 tardis.apps.push_to.migrations,119
tardis.apps.openid_migration.tests.test_farm,106 tardis.apps.push_to.migrations.0001_initial,
106
tardis.apps.openid_migration.tests.test_magda,106 tardis.apps.push_to.migrations.0002_auto_20160518_119
tardis.apps.openid_migration.tests.test_maddis,106 tardis.apps.push_to.migrations.0003_auto_20210216_119
tardis.apps.openid_migration.tests.test_taed,107 tardis.apps.push_to.models,120
tardis.apps.openid_migration.urls,110 tardis.apps.push_to.oauth_tokens,128
tardis.apps.openid_migration.user_menu_mbardi,110 tardis.apps.push_to.ssh_authz,129
tardis.apps.push_to.tasks,129 tardis.apps.push_to.tests,120
tardis.apps.openid_migration.utils,111 tardis.apps.push_to.tests.test_models,
tardis.apps.openid_migration.views,111 119
tardis.apps.publication_workflow,118 tardis.apps.push_to.urls,129
tardis.apps.publication_workflow.apps, 113 tardis.apps.push_to.utils,129
tardis.apps.push_to.views,129
tardis.apps.publication_workflow.default_tt,114 tardis.apps.related_info,132
tardis.apps.publication_workflow.doi, 114 tardis.apps.related_info.forms,132
tardis.apps.publication_workflow.email_t,114 tardis.apps.related_info.models,132
tardis.apps.related_info.settings,132
tardis.apps.publication_workflow.managers, 114 tardis.apps.related_info.tests,test_oaipmh,
tardis.apps.publication_workflow.migrations, 112 tardis.apps.related_info.tests.tests,
131 tardis.apps.related_info.urls,132
tardis.apps.publication_workflow.migratibas,111 tardis.apps.s3utils,135
tardis.apps.publication_workflow.models,tardis.apps.s3utils.api,133
115 tardis.apps.s3utils.apps,134
tardis.apps.publication_workflow.tasks, 117 tardis.apps.s3utils.default_settings,
134
tardis.apps.publication_workflow.tests, 113 tardis.apps.s3utils.tests,133
tardis.apps.publication_workflow.tests.t,112 tardis.apps.s3utils.tests.test_api,132
tardis.apps.publication_workflow.tests.t,112 tardis.apps.s3utils.tests.tests,test_app_config,
133
tardis.apps.publication_workflow.tests.t,112 tardis.apps.s3utils.tests.test_checksums,
133
tardis.apps.publication_workflow.tests.t,112 tardis.apps.s3utils.tests.test_dmy,134
tardis.apps.publication_workflow.tests.t,112 tardis.apps.s3utils.urls,134
tardis.apps.publication_workflow.views,112 tardis.apps.s3utils.utils,134

tardis.apps.search, 139
tardis.apps.search.api, 135
tardis.apps.search.apps, 137
tardis.apps.search.documents, 137
tardis.apps.search.tests, 135
tardis.apps.search.tests.test_api, 135
tardis.apps.search.tests.test_index, 135
tardis.apps.search.urls, 139
tardis.apps.search.views, 139
tardis.apps.sftp, 149
tardis.apps.sftp.admin, 140
tardis.apps.sftp.api, 140
tardis.apps.sftp.apps, 142
tardis.apps.sftp.default_settings, 142
tardis.apps.sftp.forms, 142
tardis.apps.sftp.management, 139
tardis.apps.sftp.management.commands,
139
tardis.apps.sftp.management.commands.sftp, 139
tardis.apps.sftp.migrations, 140
tardis.apps.sftp.migrations.0001_initial, 140
tardis.apps.sftp.models, 143
tardis.apps.sftp.sftp, 144
tardis.apps.sftp.tests, 140
tardis.apps.sftp.tests.test_sftp, 140
tardis.apps.sftp.urls, 149
tardis.apps.sftp.user_menu_modifiers,
149
tardis.apps.sftp.views, 149
tardis.apps.social_auth, 150
tardis.apps.social_auth.apps, 150
tardis.apps.social_auth.auth, 150
tardis.apps.social_auth.auth.social_auth,
149
tardis.apps.social_auth.default_settings,
150
tardis.celery, 294
tardis.default_settings, 157
tardis.default_settings.admins, 150
tardis.default_settings.analytics, 151
tardis.default_settings.apps, 151
tardis.default_settings.apps_default_settings,
151
tardis.default_settings.auth, 151
tardis.default_settings.caches, 151
tardis.default_settings.celery_settings,
151
tardis.default_settings.custom_views,
151
tardis.default_settings.database, 152
tardis.default_settings.debug, 152
tardis.default_settings.downloads, 153
tardis.default_settings.email, 154
tardis.default_settings.filters, 154
tardis.default_settings.frontend, 154
tardis.default_settings.graphql, 155
tardis.default_settings.i18n, 155
tardis.default_settings.localisation,
155
tardis.default_settings.logging, 155
tardis.default_settings.middlewares, 155
tardis.default_settings.publication, 155
tardis.default_settings.search, 155
tardis.default_settings.sharing, 156
tardis.default_settings.site_customisations,
156
tardis.default_settings.static_files,
156
tardis.default_settings.storage, 156
tardis.default_settings.templates, 156
tardis.tardis.default_settings.uploads, 156
tardis.default_settings.urls, 156
tardis.schema, 294
tardis.tardis_portal, 292
tardis.tardis_portal.admin, 263
tardis.tardis_portal.api, 268
tardis.tardis_portal.auth, 163
tardis.tardis_portal.auth.authentication,
157
tardis.tardis_portal.auth.authorisation,
158
tardis.tardis_portal.auth.authservice,
158
tardis.tardis_portal.auth.decorators,
159
tardis.tardis_portal.auth.fix_circular,
161
tardis.tardis_portal.auth.interfaces,
161
tardis.tardis_portal.auth.localdb_auth,
162
tardis.tardis_portal.auth.token_auth,
162
tardis.tardis_portal.auth.utils, 163
tardis.tardis_portal.context_processors,
278
tardis.tardis_portal.deprecations, 279
tardis.tardis_portal.download, 279
tardis.tardis_portal.email, 280
tardis.tardis_portal.fields, 280
tardis.tardis_portal.forms, 280
tardis.tardis_portal.graphql, 180
tardis.tardis_portal.graphql.datafile,
163
tardis.tardis_portal.graphql.datafileobject,
164

tardis.tardis_portal.graphql.dataset, tardis.tardis_portal.migrations.0009_auto_20160128_165 183
tardis.tardis_portal.graphql.experiment, tardis.tardis_portal.migrations.0010_auto_20160503_166 183
tardis.tardis_portal.graphql.facility, tardis.tardis_portal.migrations.0011_auto_20160505_167 184
tardis.tardis_portal.graphql.group, 168 tardis.tardis_portal.migrations.0012_userauthenticat...
tardis.tardis_portal.graphql.instrument, 184 tardis.tardis_portal.migrations.0013_auto_20181002_168
tardis.tardis_portal.graphql.parameters, 184 tardis.tardis_portal.migrations.0014_auto_20181002_169
tardis.tardis_portal.graphql.schema, 175 184
tardis.tardis_portal.graphql.storagebox, tardis.tardis_portal.migrations.0015_dataset_create_177 184
tardis.tardis_portal.graphql.storageboxatardis_portal.migrations.0016_add_timestamps_177 184
tardis.tardis_portal.graphql.storageboxoparidns.tardis_portal.migrations.0017_add_cc_license_178 185
tardis.tardis_portal.graphql.user, 179 tardis.tardis_portal.migrations.0018_make_default_s...
tardis.tardis_portal.graphql.utils, 179 185
tardis.tardis_portal.iif, 286 tardis.tardis_portal.models, 231
tardis.tardis_portal.logging_middleware, tardis.tardis_portal.models.access_control, 286 185
tardis.tardis_portal.management, 181 tardis.tardis_portal.models.datafile,
tardis.tardis_portal.management.commands, 190 tardis.tardis_portal.models.dataset, 196
181 tardis.tardis_portal.models.dataset, 196
tardis.tardis_portal.management.commands.tardisdataset, tardis.tardis_portal.models.experiment, 180 200
tardis.tardis_portal.management.commands.tardisdataset, tardis.tardis_portal.models.facility, 180 205
tardis.tardis_portal.management.commands.tardisdataset, tardis.tardis_portal.models.hooks, 206
181 tardis.tardis_portal.models.instrument,
tardis.tardis_portal.management.commands.rmxperiment, 205 tardis.tardis_portal.models.jti, 207
tardis.tardis_portal.managers, 286 tardis.tardis_portal.models.license, 208
tardis.tardis_portal.migrations, 185 tardis.tardis_portal.models.parameters,
tardis.tardis_portal.migrations.0001_initial, 209 tardis.tardis_portal.models.storage, 225
181 tardis.tardis_portal.models.storage, 225
tardis.tardis_portal.migrations.0001_squashed001atdastpo200050de1648open, 230
182 tardis.tardis_portal.ParameterSetManager,
tardis.tardis_portal.migrations.0002_auto_20150208_1128, 230
182 tardis.tardis_portal.publish, 232
tardis.tardis_portal.migrations.0003_auto_20150007_d131portal.publish.provider, 232
182
tardis.tardis_portal.migrations.0004_storageexportdns_palteltypublish.provider.rifcsprovider, 232
182
tardis.tardis_portal.migrations.0005_datafilesatardisepontatopublish.publishservice, 232
182
tardis.tardis_portal.migrations.0006_datafilesremovedsspoetairifig3000um, 289
183 tardis.tardis_portal.shortcuts, 289
tardis.tardis_portal.migrations.0007_rembadparameters_pbrtag_stonagen20x, 289
183 tardis.tardis_portal.storage.file_system,
tardis.tardis_portal.migrations.0008_string_val120_partial_index_postgres, 290
183 tardis.tardis_portal.tasks, 290

tardis.tardis_portal.templatetags, 235
tardis.tardis_portal.templatetags.approve, 239
tardis.tardis_portal.templatetags.basic, 239
tardis.tardis_portal.templatetags.bleach, 239
tardis.tardis_portal.templatetags.capture, 240
tardis.tardis_portal.templatetags.dynurl, 240
tardis.tardis_portal.templatetags.experiment, 240
tardis.tardis_portal.templatetags.experiment, 241
tardis.tardis_portal.templatetags.facility, 246
tardis.tardis_portal.templatetags.feed, 242
tardis.tardis_portal.templatetags.formfile, 241
tardis.tardis_portal.templatetags.lookup, 242
tardis.tardis_portal.templatetags.pagination, 242
tardis.tardis_portal.templatetags.xmldate, 242
tardis.tardis_portal.tests, 256
tardis.tardis_portal.tests.api, 239
tardis.tardis_portal.tests.api.test_auth, 244
tardis.tardis_portal.tests.api.test_datafile, 243
tardis.tardis_portal.tests.api.test_datafile, 243
tardis.tardis_portal.tests.api.test_dataset, 243
tardis.tardis_portal.tests.api.test_dataset, 243
tardis.tardis_portal.tests.api.test_experiment, 243
tardis.tardis_portal.tests.api.test_experiment, 243
tardis.tardis_portal.tests.api.test_experiment, 243
tardis.tardis_portal.tests.api.test_facility, 244
tardis.tardis_portal.tests.api.test_group, 244
tardis.tardis_portal.tests.api.test_instrument, 248
tardis.tardis_portal.tests.api.test_list, 248
tardis.tardis_portal.tests.api.test_replies, 249
tardis.tardis_portal.tests.api.test_schema, 249

```
    249                               tardis.tardis_portal.views.authorization,
tardis.tardis_portal.tests.test_df_save_metadata 257
    250                               tardis.tardis_portal.views.facilities,
tardis.tardis_portal.tests.test_download,          258
    250                               tardis.tardis_portal.views.images, 258
tardis.tardis_portal.tests.test_downloadkeytardis_portal.views.machine, 259
    250                               tardis.tardis_portal.views.pages, 259
tardis.tardis_portal.tests.test_facilitytardisviewtardis_portal.views.parameters,
    251                               262
tardis.tardis_portal.tests.test_forms,             tardis.tardis_portal.views.upload, 262
    251                               tardis.tardis_portal.views.utils, 262
tardis.tardis_portal.tests.test_iiif,              tardis.tardis_portal.widgets, 290
    251                               tardis.tardis_portal.xmlwriter, 291
tardis.tardis_portal.tests.test_ldap,              tardis.test_on_mysql_settings, 294
    252                               tardis.test_on_postgresql_settings, 294
tardis.tardis_portal.tests.test_models,            tardis.test_settings, 294
    253                               tardis.urls, 293
tardis.tardis_portal.tests.test_parameteretardis.urls.accounts, 292
    253                               tardis.urls.ajax, 292
tardis.tardis_portal.tests.test_parameteretardis.urls.api, 292
    253                               tardis.urls.core, 292
tardis.tardis_portal.tests.test_publishstardis.urls.datafile, 293
    254                               tardis.urls.dataset, 293
tardis.tardis_portal.tests.test_storage,            tardis.urls.display, 293
    255                               tardis.urls.download, 293
tardis.tardis_portal.tests.test_tar_downloadtardis.urls.experiment, 293
    255                               tardis.urls.facility, 293
tardis.tardis_portal.tests.test_tasks,              tardis.urls.group, 293
    255                               tardis.urls.token, 293
tardis.tardis_portal.tests.test_tokens,             tardis.views, 294
    255
tardis.tardis_portal.tests.tests, 256
tardis.tardis_portal.tests.urls, 256
tardis.tardis_portal.tests.views, 246
tardis.tardis_portal.tests.views.test_auth_views,
    244
tardis.tardis_portal.tests.views.test_contextual_views,
    245
tardis.tardis_portal.tests.views.test_experiment_views,
    245
tardis.tardis_portal.tests.views.test_template_contexts,
    245
tardis.tardis_portal.tests.views.test_upload_views,
    246
tardis.tardis_portal.util, 290
tardis.tardis_portal.views, 263
tardis.tardis_portal.views.ajax_actions,
    257
tardis.tardis_portal.views.ajax_json,
    257
tardis.tardis_portal.views.ajax_pages,
    257
tardis.tardis_portal.views.authentication,
    257
```

Index

A

about () (in module *tardis.tardis_portal.views.pages*), 260
abstract (*tardis.apps.push_to.models.KeyPair.Meta attribute*), 122
abstract (*tardis.tardis_portal.models.parameters.Parameter.Meta attribute*), 218
abstract (*tardis.tardis_portal.models.parameters.ParameterSet.Meta attribute*), 222
AbstractExperimentProvider (class in *tardis.apps.oaipmh.provider.experiment*), 101
AbstractExperimentProviderTC (class in *tardis.apps.oaipmh.tests.provider.test_experiment*), 102
AbstractTardisAppConfig (class in *tardis.app_config*), 293
accessible_keys (*tardis.tardis_portal.api.StorageBoxOptionResource* attribute), 277
acl_id (*tardis.apps.openid_migration.models.OpenidACLMigration attribute*), 108
acl_id_id (*tardis.apps.openid_migration.models.OpenidACLMigration attribute*), 108
acl_migration () (in module *tardis.apps.openid_migration.migration*), 107
ACLAutorization (class in *tardis.tardis_portal.api*), 268
ACLAutorizationTest (class in *tardis.tardis_portal.tests.api.test_auth*), 235
ACLAwareBackend (class in *tardis.tardis_portal.auth.authorisation*), 158
aclOwnershipType (*tardis.tardis_portal.models.access_control.ObjectACLE* attribute), 187
Action (class in *tardis.apps.eventlog.models*), 91
action (*tardis.apps.eventlog.models.Log attribute*), 92
Action.DoesNotExist, 91
Action.MultipleObjectsReturned, 91
action_id (*tardis.apps.eventlog.models.Log* at- tribute), 92
activate () (*tardis.apps.publication_workflow.doi.DOI method*), 114
add_acknowledgements () (*tardis.apps.publication_workflow.models.Publication* add_arguments () (*tardis.apps.sftp.management.commands.sftpd.Command* method), 180
add_arguments () (*tardis.tardis_portal.management.commands.createuser Command* method), 180
add_arguments () (*tardis.tardis_portal.management.commands.dumps Command* method), 180
add_arguments () (*tardis.tardis_portal.management.commands.loads Command* method), 181
add_arguments () (*tardis.tardis_portal.management.commands.rmexp Command* method), 181
add_auth_method () (in module *tardis.tardis_portal.views.parameters*), 262
add_dataset () (in module *tardis.tardis_portal.views.parameters*), 262
add_experiment_par () (in module *tardis.tardis_portal.views.parameters*), 262
add_migrate_account_menu_item () (in module *tardis.tardis_portal.views.parameters*), 262
add_migration_permission () (in module *tardis.apps.openid_migration.user_menu_modifiers*), 110
add_par () (in module *tardis.tardis_portal.views.parameters*), 262
add_path () (*tardis.apps.sftp.sftp.DynamicTree* method), 144

```

add_path_elems() (tardis.apps.sftp.sftp.DynamicTree
    method), 144
add_root_schema_pset()
    (tardis.apps.publication_workflow.models.Publication
        method), 116
add_ssh_keys_menu_item() (in module
    tardis.apps.sftp.user_menu_modifiers), 149
add_user_permissions() (in module
    tardis.apps.social_auth.auth.social_auth),
    150
added (tardis.apps.sftp.models.SFTPPublicKey
    attribute), 143
AddUserPermissionsForm (class
    tardis.tardis_portal.forms), 280
administrator (tardis.apps.push_to.models.RemoteHost
    attribute), 125
administrator_id (tardis.apps.push_to.models.RemoteHost
    attribute), 125
ADMINs (in module tardis.default_settings.admins), 150
AdvanceSearchAppResource (class
    tardis.apps.search.api), 135
AdvanceSearchAppResource.Meta (class
    tardis.apps.search.api), 135
after_save() (tardis.tardis_portal.graphql.datafile.UploadDataFilter
    class method), 164
after_save() (tardis.tardis_portal.graphql.experiment.Experiment
    class method), 166
all() (tardis.tardis_portal.managers.ExperimentManager
    Arguments
        method), 286
allow_for_all (tardis.apps.push_to.models.OAuthSSHCertSigningService
    attribute), 123
allow_reuse_address
    (tardis.apps.sftp.sftp.MySFTPTCPServer
        attribute), 146
allowed_groups (tardis.apps.push_to.models.OAuthSSHCertSigningService
    attribute), 123
ALLOWED_HOSTS, 58
ALLOWED_HOSTS (in module
    tardis.default_settings.debug), 152
allowed_methods (tardis.tardis_portal.api.UserResource
    attribute), 278
allowed_remote_hosts
    (tardis.apps.push_to.models.OAuthSSHCertSigningService
        attribute), 123
allowed_users (tardis.apps.push_to.models.OAuthSSHCertSigningService
    attribute), 123
allows_distribution
    (tardis.tardis_portal.models.license.License
        attribute), 208
always_return_data
    (tardis.apps.search.api.AdvanceSearchAppResource
        attribute), 135
always_return_data
    (tardis.apps.search.api.SearchAppResource
        attribute), 135
attribute), 136
always_return_data
    (tardis.tardis_portal.api.DatasetResource.Meta
        attribute), 271
always_return_data
    (tardis.tardis_portal.api.ExperimentAuthorResource.Meta
        attribute), 272
always_return_data
    (tardis.tardis_portal.api.ExperimentResource.Meta
        attribute), 273
always_return_data
    (tardis.tardis_portal.api.FacilityResource.Meta
        attribute), 273
always_return_data
    (tardis.tardis_portal.api.InstrumentResource.Meta
        attribute), 274
AnalyticsConfig (class in tardis.analytics.apps), 88
api_key_sign_in (tardis.tardis_portal.graphql.schema.tardisMutation
    attribute), 175
ApiKeyDownloadTestCase (class
    in tardis.tardis_portal.tests.test_download_apikey),
    250
ApiKeyTestCase (class
    in tardis.tardis_portal.tests.test_generate_api_key),
    243
Experiment (class in tardis.tardis_portal.graphql.user),
    179
Arguments (class
    in tardis.tardis_portal.graphql.user), 179
CertSigningService (class
    in tardis.tardis_portal.graphql.user), 179
app_dependencies (tardis.app_config.AbstractTardisAppConfig
    attribute), 293
app_label (tardis.tardis_portal.auth.authorisation.ACIAwareBackend
    attribute), 218
Parameter.Meta (tardis.tardis_portal.models.parameters.Parameter
    attribute), 218
ParameterSet.Meta (tardis.tardis_portal.models.parameters.ParameterSet
    attribute), 222
DataFileObject (tardis.tardis_portal.models.datafile.DataFileObject
    method), 194
approve_user_auth() (in module
    tardis.apps.social_auth.auth.social_auth),
    150
UserAuthentication (tardis.tardis_portal.models.access_control.UserAuthentication
    attribute), 188
approved (tardis.tardis_portal.models.experiment.Experiment
    attribute), 200
args (tardis.tardis_portal.management.commands.dumpschemas.Command
    attribute), 180
Command (tardis.tardis_portal.management.commands.loadschemas.Command
    attribute), 181
args (tardis.tardis_portal.management.commands.rmxperiment.Command
    attribute), 181

```

attributes (*tardis.tardis_portal.models.storage.StorageAttribute*) (class in *tardis.tardis_portal.tests.models.test_authors*), 225
 auth_timeout (*tardis.apps.sftp.sftp.MyTSFTPRequestHandler*) (attribute), 243
 authenticate () (*tardis.tardis_portal.auth.authorisation.ACIAwareBackend*) (method), 158
 authenticate () (*tardis.tardis_portal.auth.authservice.AuthService*) (method), 158
 authenticate () (*tardis.tardis_portal.auth.interfaces.AuthProvider*) (method), 158
 authenticate () (*tardis.tardis_portal.auth.interfaces.AuthProvider*) (method), 161
 authenticate () (*tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend*) (method), 249
 authenticate () (*tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend*) (method), 249
 authenticate () (*tardis.tardis_portal.tests.test_authservice.NOCMEMBER_PROVIDERS_KEY*) (method), 151
 authentication (*tardis.apps.search.api.AdvanceSearchAppResource.Meta*)
 authentication (*tardis.apps.search.api.SearchAppResource.Meta*)
 authentication (*tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta*)
 authentication (*tardis.tardis_portal.api.GroupResourceMeta*)
 authentication (*tardis.tardis_portal.api.MyTardisModelResourceMeta*)
 authentication (*tardis.tardis_portal.api.ObjectACLResourceMeta*)
 authentication (*tardis.tardis_portal.api.UserResourceMeta*)
 authenticationMethod
 AuthenticationTestCase (class in *tardis.tardis_portal.tests.test_authentication*), 248
 author (*tardis.tardis_portal.models.experiment.ExperimentAuthor*)
 authorization (*tardis.apps.hsm.api.DatasetAppResourceMeta*)
 authorization (*tardis.apps.hsm.api.ReplicaAppResourceMeta*)
 authorization (*tardis.apps.s3utils.api.ReplicaAppResourceMeta*)
 authorization (*tardis.apps.sftp.api.SFTPPublicKeyAppResourceMeta*)
 authorization (*tardis.tardis_portal.api.GroupResourceMeta*)
 authorization (*tardis.tardis_portal.api.MyTardisModelResourceMeta*)
 authorization (*tardis.tardis_portal.api.ObjectACLResourceMeta*)
 authorization (*tardis.tardis_portal.api.UserResourceMeta*)
 authorize_remote_access () (in module *tardis.push_to.views*), 129
B
 BackgroundTaskTestCase (class in *tardis.tardis_portal.tests.test_tasks*), 255
 base_fields (*tardis.apps.push_to.models.Request*) (attribute), 127
 base_fields (*tardis.apps.hsm.api.DatasetAppResource*) (attribute), 96
 base_fields (*tardis.apps.hsm.api.ReplicaAppResource*) (attribute), 97
 base_fields (*tardis.apps.push_to.models.CredentialForm*) (attribute), 122
 base_fields (*tardis.apps.related_info.forms.RelatedInfoForm*) (attribute), 132
 base_fields (*tardis.apps.s3utils.api.ReplicaAppResource*) (attribute), 133
 base_fields (*tardis.apps.search.api.AdvanceSearchAppResource*) (attribute), 136
 base_fields (*tardis.apps.search.api.SearchAppResource*) (attribute), 137
 base_fields (*tardis.apps.sftp.api.SFTPPublicKeyAppResource*) (attribute), 141
 base_fields (*tardis.apps.sftp.forms.KeyAddForm*) (attribute), 142
 base_fields (*tardis.apps.sftp.forms.KeyGenerateForm*) (attribute), 142
 base_fields (*tardis.tardis_portal.admin.DatafileAdminForm*) (attribute), 264
 base_fields (*tardis.tardis_portal.admin.DataFileObjectInlineForm*) (attribute), 264
 base_fields (*tardis.tardis_portal.admin.StorageBoxAttributeInlineForm*) (attribute), 267
 base_fields (*tardis.tardis_portal.admin.StorageBoxForm*) (attribute), 268
 base_fields (*tardis.tardis_portal.admin.StorageBoxOptionInlineForm*) (attribute), 268
 base_fields (*tardis.tardis_portal.api.DatafileParameterResource*) (attribute), 270
 base_fields (*tardis.tardis_portal.api.DatafileParameterSetResource*) (attribute), 271

base_fields (*tardis.tardis_portal.api.DataFileResource*.base_fields (*tardis.tardis_portal.forms.ExperimentForm*.attribute), 270
base_fields (*tardis.tardis_portal.api.DatasetParameterResource*.base_fields (*tardis.tardis_portal.forms.ImportParamsForm*.attribute), 271
base_fields (*tardis.tardis_portal.api.DatasetParameterSetResource*.base_fields (*tardis.tardis_portal.forms.LoginForm*.attribute), 271
base_fields (*tardis.tardis_portal.api.DatasetResource*.base_fields (*tardis.tardis_portal.forms.ManageAccountForm*.attribute), 271
base_fields (*tardis.tardis_portal.api.ExperimentAuthorResource*.base_fields (*tardis.tardis_portal.forms.ManageGroupPermissionsForm*.attribute), 272
base_fields (*tardis.tardis_portal.api.ExperimentParameterResource*.base_fields (*tardis.tardis_portal.forms.RegisterExperimentForm*.attribute), 272
base_fields (*tardis.tardis_portal.api.ExperimentParameterSetResource*.base_fields (*tardis.tardis_portal.forms.RegistrationForm*.attribute), 273
base_fields (*tardis.tardis_portal.api.ExperimentResource*.base_fields (*tardis.tardis_portal.forms.RightsForm*.attribute), 273
base_fields (*tardis.tardis_portal.api.FacilityResource*.base_filters (*tardis.tardis_portal.graphql.datafile.DataTypeFilter*.attribute), 273
base_fields (*tardis.tardis_portal.api.GroupResource*.base_filters (*tardis.tardis_portal.graphql.datafileobject.DataFileObject*.attribute), 274
base_fields (*tardis.tardis_portal.api.InstrumentResource*.base_filters (*tardis.tardis_portal.graphql.dataset.DatasetTypeFilter*.attribute), 274
base_fields (*tardis.tardis_portal.api.LocationResource*.base_filters (*tardis.tardis_portal.graphql.experiment.ExperimentAuth*.attribute), 274
base_fields (*tardis.tardis_portal.api.MyTardisModelResource*.base_filters (*tardis.tardis_portal.graphql.experiment.ExperimentType*.attribute), 275
base_fields (*tardis.tardis_portal.api.ObjectACLResource*.base_filters (*tardis.tardis_portal.graphql.facility.FacilityTypeFilter*.attribute), 275
base_fields (*tardis.tardis_portal.api.ParameterNameResource*.base_filters (*tardis.tardis_portal.graphql.group.GroupTypeFilter*.attribute), 275
base_fields (*tardis.tardis_portal.api.ParameterResource*.base_filters (*tardis.tardis_portal.graphql.instrument.InstrumentTypeF*.attribute), 276
base_fields (*tardis.tardis_portal.api.ParameterSetResource*.base_filters (*tardis.tardis_portal.graphql.parameters.DatafileParam*.attribute), 276
base_fields (*tardis.tardis_portal.api.ReplicaResource*.base_filters (*tardis.tardis_portal.graphql.parameters.DatafileParam*.attribute), 276
base_fields (*tardis.tardis_portal.api.SchemaResource*.base_filters (*tardis.tardis_portal.graphql.parameters.DatasetParam*.attribute), 277
base_fields (*tardis.tardis_portal.api.StorageBoxAttributeResource*.base_filters (*tardis.tardis_portal.graphql.parameters.DatasetParam*.attribute), 277
base_fields (*tardis.tardis_portal.api.StorageBoxOptionResource*.base_filters (*tardis.tardis_portal.graphql.parameters.ExperimentPar*.attribute), 277
base_fields (*tardis.tardis_portal.api.StorageBoxResource*.base_filters (*tardis.tardis_portal.graphql.parameters.ExperimentPar*.attribute), 277
base_fields (*tardis.tardis_portal.api.UserResource*.base_filters (*tardis.tardis_portal.graphql.parameters.ParameterName*.attribute), 278
base_fields (*tardis.tardis_forms.AddUserPermissionsForm*.base_filters (*tardis.tardis_portal.graphql.parameters.TardisSchemaTy*.attribute), 280
base_fields (*tardis.tardis_forms.CreateUserPermissionsForm*.base_filters (*tardis.tardis_portal.graphql.storagebox.StorageBoxType*.attribute), 281
base_fields (*tardis.tardis_forms.DatasetForm*.base_filters (*tardis.tardis_portal.graphql.storageboxattribute.StorageB*.attribute), 281
base_fields (*tardis.tardis_forms.ExperimentAuthorForm*.base_filters (*tardis.tardis_portal.graphql.storageboxoption.StorageB*.attribute), 282

BaseProvider (class *tardis.apps.oaipmh.provider.base*), 99, 298
BaseProviderTestCase (class *tardis.apps.oaipmh.tests.provider.test_base*), 102
before_save () (*tardis.tardis_portal.graphql.experiment.CreateExperiment* class method), 166
blank_param (*tardis.tardis_portal.ParameterSetManager.ParameterSetManager* attribute), 263
BLEACH_ALLOWED_ATTRIBUTES (in module *tardis.default_settings.frontend*), 154
BLEACH_ALLOWED_TAGS (in module *tardis.default_settings.frontend*), 154
bleach_value () (in module *tardis.tardis_portal.templatetags.bleach_tag*), 233
build_template_html () (in module *tardis.tardis_portal.email*), 280
build_template_text () (in module *tardis.tardis_portal.email*), 280
BUNDLE (*tardis.tardis_portal.models.storage.StorageBox* attribute), 225

C

CACHE (*tardis.tardis_portal.models.storage.StorageBox* attribute), 225
cache_box (*tardis.tardis_portal.models.storage.StorageBox* attribute), 226
cache_file () (*tardis.tardis_portal.models.datafile.DataFile* method), 191
cache_file () (*tardis.tardis_portal.models.datafile.DataFileObject* method), 194
CACHES (in module *tardis.default_settings.caches*), 151
calculate_checksums () (in module *tardis.apps.s3utils.utils*), 134
calculate_checksums () (*tardis.tardis_portal.models.datafile.DataFileObject* method), 194
CALCULATE_CHECKSUMS_METHODS (in module *tardis.default_settings.storage*), 156
can_copy () (in module *tardis.apps.push_to.utils*), 129
can_publish () (*tardis.tardis_portal.publish.provider.RFCSPublisher* method), 232
canDelete (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 187
canonicalize () (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* method), 144
canRead (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 187
canWrite (*tardis.tardis_portal.models.access_control.ObjectACL* attribute), 187
capture () (in module *tardis.tardis_portal.templatetags.capture*), 233
CaptureNode (class *tardis.tardis_portal.templatetags.capture*), 233
cast_string_to_integer () (in module *tardis.tardis_portal.migrations.0001_squashed_0011_auto_2016* 183
cert_signing_url (*tardis.apps.push_to.models.OAuthSSHCertSigning* attribute), 123
check_app_dependencies () (in module *tardis.app_config*), 293
check_auth_interactive () (*tardis.apps.sftp.sftp.MyTServerInterface* method), 146
check_auth_interactive_response () (*tardis.apps.sftp.sftp.MyTServerInterface* method), 146
check_auth_password () (*tardis.apps.sftp.sftp.MyTServerInterface* method), 147
check_auth_publickey () (*tardis.apps.sftp.sftp.MyTServerInterface* method), 147
check_channel_request () (*tardis.apps.sftp.sftp.MyTServerInterface* method), 147
check_if_facility_manager () (in module *tardis.tardis_portal.templatetags.facility_tags*), 234
check_if_user_not_approved () (in module *tardis.tardis_portal.templatetags.approved_user_tags*), 233
check_paths () (*tardis.tardis_portal.tests.slapd.Slapd* class method), 247
child_boxes (*tardis.tardis_portal.models.storage.StorageBox* attribute), 226
choices (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 220
classic_mapper () (in module *tardis.tardis_portal.download*), 279
clean () (*tardis.apps.sftp.forms.KeyAddForm* method), 285
clean () (*tardis.tardis_portal.forms.RegistrationForm* method), 285
clean () (*tardis.tardis_portal.forms.RightsForm* method), 285
clean () (*tardis.tardis_portal.forms.StaticField* method), 285
clean_input () (*tardis.tardis_portal.graphql.datafile.UploadDataFile* class method), 164
clean_instance () (*tardis.tardis_portal.graphql.experiment.CreateExperiment* class method), 166

clean_username () (*tardis.tardis_portal.forms.RegistrationForm*)
 method), 285

cleanup () (*tardis.apps.filepicker.utils.FilepickerFile*)
 method), 93

clear_children () (*tardis.apps.sftp.sftp.DynamicTree*)
 method), 144

close () (*tardis.analytics.tracker.IteratorTracker*)
 method), 88

close_gzip () (*tardis.tardis_portal.download.UncachedTarStream*)
 method), 187

close_request () (*tardis.apps.sftp.sftp.MyTSFTPTCPServer*)
 method), 146

CollectstaticTest (class in *tardis.tardis_portal.tests.management.test_collectstatic*)
 241

Command (class in *tardis.apps.sftp.management.commands.sftpd*), 245
 139

Command (class in *tardis.tardis_portal.management.commands.createuser*), 194
 180

Command (class in *tardis.tardis_portal.management.commands.dump*), 226
 180

Command (class in *tardis.tardis_portal.management.commands.loadschema*), 226
 181

Command (class in *tardis.tardis_portal.management.commands.rmxpand*), 249
 181

CommaSeparatedInput (class in *tardis.tardis_portal.widgets*), 290

comparison_type (*tardis.tardis_portal.models.parameters.ParameterName*)
 attribute), 220

complete_request () (in module *tardis.apps.push_to.tasks*), 129

compliance_header () (in module *tardis.tardis_portal.iiif*), 286

compress () (*tardis.tardis_portal.download.UncachedTarStream*)
 method), 279

compress () (*tardis.tardis_portal.fields.MultiValueCommaSeparatedField*), 175
 method), 280

compute_checksums () (in module *tardis.tardis_portal.models.datafile*), 196

compute_etag () (in module *tardis.tardis_portal.iiif*), 286

compute_size () (*tardis.tardis_portal.download.UncachedTarStream*)
 method), 279

configure () (*tardis.tardis_portal.tests.slapd.Slapd*)
 method), 247

configure_social_auth_user () (in module *tardis.apps.social_auth.auth.social_auth*), 150

configure_user () (in module *tardis.tardis_portal.auth.utils*), 163

confirm_migration () (in module *tardis.apps.openid_migration.migration*), 108

connection (*tardis.apps.hsm.tests.test_migrations.HsmAppMigrati*), 115
 attribute), 96

DATUM_COMPARISON (*tardis.tardis_portal.models.parameters.ParameterName*)
 attribute), 219

content_object (*tardis.tardis_portal.models.access_control.ObjectAC*)
 attribute), 187

content_type (*tardis.apps.eventlog.models.Log*)
 attribute), 92

content_type (*tardis.tardis_portal.models.access_control.ObjectACL*)
 attribute), 187

content_type_id (*tardis.apps.eventlog.models.Log*)
 attribute), 92

content_type_id (*tardis.tardis_portal.models.access_control.ObjectAC*)
 attribute), 187

ContextualViewTest (class in *tardis.tardis_portal.tests.views.test_contextual_views*), 245

copy_file () (*tardis.tardis_portal.models.datafile.DataFileObject*)
 create_datafile (tardis.tardis_portal.graphql.schema.tardisMutation)
 attribute), 175

copy_files () (*tardis.tardis_portal.models.storage.StorageBox*)
 copy_to_master () (tardis.tardis_portal.models.storage.StorageBox)
 create_datafileparameter (tardis.tardis_portal.graphql.schema.tardisMutation)
 attribute), 175

CopyMoveTestCase (class in *tardis.tardis_portal.tests.test_copy_move*), 249

CreateDatasetParameter (tardis.tardis_portal.graphql.schema.tardisMutation)
 attribute), 175

create_dataset (*tardis.tardis_portal.graphql.schema.tardisMutation*)
 attribute), 175

create_datasetparameter (tardis.tardis_portal.graphql.schema.tardisMutation)
 attribute), 175

create_detail () (*tardis.apps.sftp.api.SFTPACLAuthorization*)
 method), 140

create_detail () (*tardis.tardis_portal.api.ACALAuthorization*)
 method), 269

create_draft_publication () (*tardis.apps.publication_workflow.managers.PublicationManager*)
 method), 114

create_draft_publication () (*tardis.tardis_portal.models.Publication*)
 method), 115

create_experiment
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.create_user()
 (*in module tardis.tardis_portal.auth.utils*), 163
 attribute), 175
 create_experiment()
 (*in module tardis.tardis_portal.views.pages*), 261
 create_experimentauthor
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.create_user_api_key()
 (*in module tardis.tardis_portal.models.access_control*),
 attribute), 175
 create_experimentparameter
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.create_user_profile()
 (*in module tardis.tardis_portal.models.access_control*),
 attribute), 175
 create_experimentparameterset
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.create_by(*tardis.tardis_portal.models.experiment.Experiment*
 attribute), 175
 attribute), 200
 create_facility(*tardis.tardis_portal.graphql.schema.tardisMutation*.id (*tardis.tardis_portal.models.experiment.Experiment*
 attribute), 175
 create_group(*tardis.tardis_portal.graphql.schema.tardisMutation*.created_time (*tardis.tardis_portal.models.datafile.DataFile*
 attribute), 175
 create_group()
 (*in module tardis.tardis_portal.views.authorisation*),
 257
 created_time (*tardis.tardis_portal.models.dataset.Dataset*
 attribute), 190
 create_instrument
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.created_time (*tardis.tardis_portal.models.experiment.Experiment*
 attribute), 175
 attribute), 201
 create_list()
 (*tardis.tardis_portal.api.ACIAuthorization*.create_time (*tardis.tardis_portal.models.facility.Facility*
 method), 269
 attribute), 205
 create_local_box()
 (*tardis.tardis_portal.models.storage.StorageBox*
 class method), 226
 created_time (*tardis.tardis_portal.models.instrument.Instrument*
 attribute), 206
 create_metadata_schemas()
 (*tardis.apps.publication_workflow.migrations.0001_initial*.CreateDataFile
 (class *tardis.tardis_portal.graphql.datafile*), 163
 attribute), 111
 create_OpenidUserMigration()
 (*tardis.apps.openid_migration.tests.test_models.ModelTest*.CreateDataFileObject
 (class *tardis.tardis_portal.graphql.datafileobject*),
 method), 106
 164
 create_parameter_add_form()
 (*in module tardis.tardis_portal.forms*), 286
 CreateDatafileParameter
 (class *tardis.tardis_portal.graphql.parameters*),
 create_parametername
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.CreateDatafileParameterSet
 (class *tardis.tardis_portal.graphql.parameters*),
 attribute), 175
 create_parameterset_edit_form()
 (*in module tardis.tardis_portal.forms*), 286
 CreateDataset
 (class *tardis.tardis_portal.graphql.parameters*),
 create_schema(*tardis.tardis_portal.graphql.schema.tardisMutation*.CreateDatasetParameter
 (class *tardis.tardis_portal.graphql.dataset*), 165
 attribute), 175
 create_set_uri()
 (*tardis.tardis_portal.models.datafile.DataFileObject*.CreateDatasetParameter
 (class *tardis.tardis_portal.graphql.parameters*),
 method), 194
 169
 create_storagebox
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.CreateDatasetParameterSet
 (class *tardis.tardis_portal.graphql.parameters*),
 attribute), 175
 create_storageboxattribute
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.CreateExperiment
 (class *tardis.tardis_portal.graphql.experiment*),
 attribute), 175
 166
 create_storageboxoption
 (*tardis.tardis_portal.graphql.schema.tardisMutation*.CreateExperimentAuthor
 (class *tardis.tardis_portal.graphql.experiment*),
 attribute), 175
 167

166				
CreateExperimentParameter (class <i>tardis.tardis_portal.graphql.parameters</i>), 169	in	cybderduck_connection_window() (in module <i>tardis.apps.sftp.views</i>), 149		
CreateExperimentParameterSet (class <i>tardis.tardis_portal.graphql.parameters</i>), 170	in	D		
CreateFacility (class <i>tardis.tardis_portal.graphql.facility</i>), 167	in	data_type (<i>tardis.tardis_portal.models.parameters.ParameterName</i> attribute), 220		
createFacilityTestCase (class <i>tardis.tardis_portal.tests.graphql.test_facilities</i>), 240	in	DATABASE_ENGINE (<i>tardis.default_settings</i> attribute), 58		
CreateGroup (class <i>tardis.tardis_portal.graphql.group</i>), 168	in	DATABASE_HOST (<i>tardis.default_settings</i> attribute), 58		
CreateInstrument (class <i>tardis.tardis_portal.graphql.instrument</i>), 168	in	DATABASE_NAME (<i>tardis.default_settings</i> attribute), 58		
createLinkedUserAuthenticationForm() (in module <i>tardis.tardis_portal.forms</i>), 286		DATABASE_PASSWORD (<i>tardis.default_settings</i> attribute), 58		
CreateParameterName (class <i>tardis.tardis_portal.graphql.parameters</i>), 170	in	DATABASE_PORT (<i>tardis.default_settings</i> attribute), 58		
CreateSchema (class <i>tardis.tardis_portal.graphql.parameters</i>), 170	in	DATABASE_USER (<i>tardis.default_settings</i> attribute), 58		
CreateStorageBox (class <i>tardis.tardis_portal.graphql.storagebox</i>), 177	in	DataFile (class in <i>tardis.tardis_portal.models.datafile</i>), 190		
CreateStorageBoxAttribute (class <i>tardis.tardis_portal.graphql.storageboxattribute</i>), 177	in	datafile (<i>tardis.apps.push_to.models.Progress</i> attribute), 124		
CreateStorageBoxOption (class <i>tardis.tardis_portal.graphql.storageboxoption</i>), 178	in	datafile (<i>tardis.tardis_portal.models.datafile.DataFileObject</i> attribute), 194		
CreateTestCase (class <i>tardis.apps.related_info.tests.tests</i>), 131	in	datafile (<i>tardis.tardis_portal.models.parameters.DatafileParameterSet</i> attribute), 210		
CreateUserPermissionsForm (class <i>tardis.tardis_portal.forms</i>), 281	in	DATAFILE (<i>tardis.tardis_portal.models.parameters.Schema</i> attribute), 223		
CreateUserTestCase (class <i>tardis.tardis_portal.tests.management.test_createuser</i>), 242	in	DataFile.DoesNotExist, 191		
Credential (class in <i>tardis.apps.push_to.models</i>), 120		DataFile.MultipleObjectsReturned, 191		
credential (<i>tardis.apps.push_to.models.Request</i> attribute), 127		datafile_access_required() (in module <i>tardis.tardis_portal.auth.decorators</i>), 159		
Credential.DoesNotExist, 120		datafile_id (<i>tardis.apps.push_to.models.Progress</i> attribute), 124		
Credential.MultipleObjectsReturned, 120		datafile_id (<i>tardis.tardis_portal.models.datafile.DataFileObject</i> attribute), 194		
credential_id (<i>tardis.apps.push_to.models.Request</i> attribute), 127	in	datafile_id (<i>tardis.tardis_portal.models.parameters.DatafileParameter</i> attribute), 210		
credential_set (<i>tardis.apps.push_to.models.RemoteHost</i> attribute), 125	in	datafile_set (<i>tardis.tardis_portal.models.dataset.Dataset</i> attribute), 197		
CredentialAdmin (class <i>tardis.apps.push_to.models</i>), 121	in	DatafileAdmin (class in <i>tardis.tardis_portal.admin</i>), 264		
CredentialForm (class <i>tardis.apps.push_to.models</i>), 121	in	DatafileAdminForm (class <i>tardis.tardis_portal.admin</i>), 264	in	
CredentialForm.Meta (class <i>tardis.tardis_portal.admin</i>), 121	in	DatafileAdminForm.Meta (class <i>tardis.tardis_portal.admin</i>), 264	in	
		DatafileDocument (class <i>tardis.apps.search.documents</i>), 137	in	
		DatafileDocument.Django (class <i>tardis.apps.search.documents</i>), 137	in	
		DatafileDocument.Index (class <i>tardis.apps.search.documents</i>), 137	in	
		DatafileObject (class <i>tardis.tardis_portal.models.datafile</i>), 194	in	
		DatafileObject.DoesNotExist, 194		
		DatafileObject.MultipleObjectsReturned, 194		

DataFileObjectInline (class in *tardis.tardis_portal.admin*), 263
DataFileObjectInlineForm (class in *tardis.tardis_portal.admin*), 263
DataFileObjectInlineForm.Meta (class in *tardis.tardis_portal.admin*), 264
DataFileObjectNotVerified, 98
datafileobjects (*tardis.tardis_portal.graphql.schema.tardisQuery* attribute), 226
attribute), 176
DataFileObjectTestCase (class in *tardis.tardis_portal.tests.models.test_dfo*), 243
DataFileObjectType (class in *tardis.tardis_portal.graphql.datafileobject*), 164
DataFileObjectTypeFilter (class in *tardis.tardis_portal.graphql.datafileobject*), 164
DataFileObjectTypeFilter.Meta (class in *tardis.tardis_portal.graphql.datafileobject*), 165
DatafileParameter (class in *tardis.tardis_portal.models.parameters*), 209
DatafileParameter.DoesNotExist, 209
DatafileParameter.MultipleObjectsReturned, 209
datafileparameter_set (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 210
datafileparameter_set (tardis.tardis_portal.models.parameters.ParameterName attribute), 220
DatafileParameterResource (class in *tardis.tardis_portal.api*), 270
DatafileParameterResource.Meta (class in *tardis.tardis_portal.api*), 270
DatafileParameterResourceTest (class in *tardis.tardis_portal.tests.api.test_datafile_metadata*), 235
DatafileParameterSet (class in *tardis.tardis_portal.models.parameters*), 210
DatafileParameterSet.DoesNotExist, 210
DatafileParameterSet.MultipleObjectsReturned, 210
datafileparameterset_set (tardis.tardis_portal.models.datafile.DataFile attribute), 191
datafileparameterset_set (tardis.tardis_portal.models.parameters.Schema attribute), 223
DatafileParameterSetResource (class in *tardis.tardis_portal.api*), 270
in DatafileParameterSetResource.Meta (class in *tardis.tardis_portal.api*), 270
in DatafileParameterSetResourceTest (class in *tardis.tardis_portal.tests.api.test_datafile_metadata_resources*), 235
datafileparametersets
(*tardis.tardis_portal.models.storage.StorageBox* attribute), 226
DatafileParameterSetType (class in *tardis.tardis_portal.graphql.parameters*), 170
DatafileParameterSetTypeFilter (class in *tardis.tardis_portal.graphql.parameters*), 170
DatafileParameterSetTypeFilter.Meta (class in *tardis.tardis_portal.graphql.parameters*), 170
DatafileParameterType (class in *tardis.tardis_portal.graphql.parameters*), 170
DatafileParameterTypeFilter (class in *tardis.tardis_portal.graphql.parameters*), 170
DatafileParameterTypeFilter.Meta (class in *tardis.tardis_portal.graphql.parameters*), 171
DataFileResource (class in *tardis.tardis_portal.api*), 269
DataFileResourceTest (class in *tardis.tardis_portal.tests.api.test_datafile_resource*), 236
datafiles (*tardis.tardis_portal.graphql.schema.tardisQuery* attribute), 176
DatafileSaveMetadataTestCase (class in *tardis.tardis_portal.tests.test_df_save_metadata*), 250
DataFileTestCase (class in *tardis.tardis_portal.tests.models.test_datafile*), 243
DataFileType (class in *tardis.tardis_portal.graphql.datafile*), 163
DataFileTypeFilter (class in *tardis.tardis_portal.graphql.datafile*), 163
DataFileTypeFilter.Meta (class in *tardis.tardis_portal.graphql.datafile*), 164
dataset (class in *tardis.tardis_portal.models.dataset*), 196
dataset (*tardis.tardis_portal.models.datafile.DataFile* attribute), 191
dataset (*tardis.tardis_portal.models.parameters.DatasetParameterSet* attribute), 212
DATASET (*tardis.tardis_portal.models.parameters.Schema* attribute), 223
Dataset.DoesNotExist, 196
Dataset.MultipleObjectsReturned, 196

dataset_access_required() (in module tardis.tardis_portal.auth.decorators), 159

dataset_aggregate_info() (in module tardis.tardis_portal.views.facilities), 258

dataset_download_required() (in module tardis.tardis_portal.auth.decorators), 159

dataset_id (tardis.tardis_portal.models.datafile.DataFile attribute), 191

dataset_id (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 212

dataset_online_count () (in module tardis.apps.hsm.check), 97

dataset_recall () (tardis.apps.hsm.api.DatasetAppResource method), 96

dataset_set (tardis.tardis_portal.models.instrument.Instrument attribute), 206

DATASET_VIEWS (in module tardis.default_settings.custom_views), 151

dataset_write_permissions_required() (in module tardis.tardis_portal.auth.decorators), 159

DatasetAdmin (class in tardis.tardis_portal.admin), 265

DatasetAppResource (class in tardis.apps.hsm.api), 96

DatasetAppResource.Meta (class in tardis.apps.hsm.api), 96

DatasetDocument (class in tardis.apps.search.documents), 138

DatasetDocument.Django (class in tardis.apps.search.documents), 138

DatasetDocument.Index (class in tardis.apps.search.documents), 138

DatasetForm (class in tardis.tardis_portal.forms), 281

DatasetForm.Meta (class in tardis.tardis_portal.forms), 281

DatasetParameter (class in tardis.tardis_portal.models.parameters), 211

DatasetParameter.DoesNotExist, 211

DatasetParameter.MultipleObjectsReturned, 211

datasetparameter_set (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 212

datasetparameter_set (tardis.tardis_portal.models.parameters.ParameterSets attribute), 220

DatasetParameterResource (class in tardis.tardis_portal.api), 271

DatasetParameterResource.Meta (class in tardis.tardis_portal.api), 271

DatasetParameterResourceTest (class in tardis.tardis_portal.tests.api.test_dataset_resource), 236

datasetparametersets (tardis.tardis_portal.models.storage.StorageBox attribute), 226

DatasetParameterSet (class in tardis.tardis_portal.models.dataset.Dataset attribute), 197

datasetparameterset_set (tardis.tardis_portal.models.parameters.Schema attribute), 224

DatasetParameterSetResource (class in tardis.tardis_portal.api), 271

DatasetParameterSetResource.Meta (class in tardis.tardis_portal.api), 271

DatasetParameterSetResourceTest (class in tardis.tardis_portal.tests.api.test_dataset_metadata_resources), 236

datasetparametersets (tardis.tardis_portal.models.storage.StorageBox attribute), 226

DatasetParameterSetType (class in tardis.tardis_portal.graphql.parameters), 171

DatasetParameterSetTypeFilter (class in tardis.tardis_portal.graphql.parameters), 171

DatasetParameterSetTypeFilter.Meta (class in tardis.tardis_portal.graphql.parameters), 171

DatasetParameterType (class in tardis.tardis_portal.graphql.parameters), 171

DatasetParameterTypeFilter (class in tardis.tardis_portal.graphql.parameters), 171

DatasetParameterTypeFilter.Meta (class in tardis.tardis_portal.graphql.parameters), 172

DatasetResource (class in tardis.tardis_portal.api), 271

DatasetResource.Meta (class in tardis.tardis_portal.api), 271

DatasetResourceTest (class in tardis.tardis_portal.tests.api.test_dataset_resource), 236

datasets (tardis.tardis_portal.models.experiment.Experiment attribute), 201

DatasetTestCase (class in tardis.tardis_portal.tests.models.test_dataset), 243

datasetparameterstype (tardis.tardis_portal.models.parameters.ParameterSets attribute), 176

tardis.tardis_portal.graphql.dataset), 165
 DatasetTypeFilter (class in declared_fields (*tardis.tardis_portal.admin.StorageBoxOptionInlineForm*.attribute)), 268
 tardis.tardis_portal.graphql.dataset), 165
 DatasetTypeFilter.Meta (class in declared_fields (*tardis.tardis_portal.api.DatafileParameterResource*.attribute)), 270
 tardis.tardis_portal.graphql.dataset), 165
 DatasetView (class in declared_fields (*tardis.tardis_portal.api.DatafileParameterSetResource*.attribute)), 271
 tardis.tardis_portal.views.pages), 259
 DATETIME (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 270
 attribute), 219
 datetime_to_us () (in module declared_fields (*tardis.tardis_portal.api.DatasetParameterResource*.attribute)), 271
 tardis.tardis_portal.views.facilities), 258
 datetime_value (*tardis.tardis_portal.models.parameters.Parameter* attribute), 271
 attribute), 218
 db_type () (*tardis.apps.eventlog.fields.JSONField* method), 91
 DBHostKeyPolicy (class in declared_fields (*tardis.tardis_portal.api.DatasetResource*.attribute)), 271
 tardis.apps.push_to.models), 122
 declared_fields (*tardis.tardis_portal.api.ExperimentAuthorResource*.attribute)), 272
 DcExperimentProvider (class in declared_fields (*tardis.tardis_portal.api.ExperimentParameterResource*.attribute)), 272
 tardis.apps.oaipmh.provider.experiment), 101
 declared_fields (*tardis.tardis_portal.api.ExperimentParameterSetResource*.attribute)), 273
 DcExperimentProviderTestCase (class in declared_fields (*tardis.tardis_portal.api.ExperimentResource*.attribute)), 273
 tardis.apps.oaipmh.tests.provider.test_experiment), 103
 declared_fields (*tardis.tardis_portal.api.FacilityResource*.attribute)), 273
 deactivate () (*tardis.apps.publication_workflow.doi.DOI* attribute), 273
 method), 114
 DEBUG (in module *tardis.default_settings.debug*), 152
 declared_fields (*tardis.apps.hsm.api.DatasetAppResource*.attribute), 274
 declared_fields (*tardis.tardis_portal.api.InstrumentResource*.attribute)), 274
 declared_fields (*tardis.apps.hsm.api.ReplicaAppResource*.attribute), 274
 declared_fields (*tardis.tardis_portal.api.LocationResource*.attribute)), 274
 declared_fields (*tardis.apps.push_to.models.Credential*.attribute), 275
 declared_fields (*tardis.tardis_portal.api.MyTardisModelResource*.attribute)), 275
 declared_fields (*tardis.apps.related_info.forms.RelatedInfoForm*.attribute), 275
 declared_fields (*tardis.tardis_portal.api.ObjectACLResource*.attribute)), 275
 declared_fields (*tardis.apps.s3utils.api.ReplicaAppResource*.attribute), 275
 declared_fields (*tardis.tardis_portal.api.ParameterNameResource*.attribute)), 275
 declared_fields (*tardis.apps.search.api.AdvanceSearchAppResource*.attribute), 276
 declared_fields (*tardis.tardis_portal.api.ParameterResource*.attribute)), 276
 declared_fields (*tardis.apps.search.api.SearchAppResource*.attribute), 276
 declared_fields (*tardis.tardis_portal.api.ParameterSetResource*.attribute)), 276
 declared_fields (*tardis.apps.sftp.api.SFTPPublicKeyAppResource*.attribute), 276
 declared_fields (*tardis.tardis_portal.api.ReplicaResource*.attribute)), 276
 declared_fields (*tardis.apps.sftp.forms.KeyAddForm*.attribute), 277
 declared_fields (*tardis.tardis_portal.api.SchemaResource*.attribute)), 277
 declared_fields (*tardis.apps.sftp.forms.KeyGenerateForm*.attribute), 277
 declared_fields (*tardis.tardis_portal.api.StorageBoxAttributeResource*.attribute)), 277
 declared_fields (*tardis.tardis_portal.admin.DatafileAdminForm*.attribute), 277
 declared_fields (*tardis.tardis_portal.api.StorageBoxOptionResource*.attribute)), 277
 declared_fields (*tardis.tardis_portal.admin.DataFileObjectInlineForm*.attribute), 277
 declared_fields (*tardis.tardis_portal.api.StorageBoxResource*.attribute)), 277
 declared_fields (*tardis.tardis_portal.admin.StorageBoxAttributeInlineForm*.attribute), 278
 declared_fields (*tardis.tardis_portal.admin.StorageBoxForm*.attribute), 278
 declared_fields (*tardis.tardis_portal.forms.AddUserPermissionsForm*.attribute)), 280

```

declared_fields (tardis.tardis_portal.forms.CreateUserPermissionsFormers (tardis.tardis_portal.graphql.storagebox.StorageBox
    attribute), 281                                         attribute), 177
declared_fields (tardis.tardis_portal.forms.DatasetForm declared_filters (tardis.tardis_portal.graphql.storageboxattribute.StorageBoxAttribute
    attribute), 281                                         attribute), 178
declared_fields (tardis.tardis_portal.forms.ExperimentAuthorised_filters (tardis.tardis_portal.graphql.storageboxoption.StorageBoxOption
    attribute), 282                                         attribute), 179
declared_fields (tardis.tardis_portal.forms.ExperimentForm storage_mapper ()      (in      module
    attribute), 282                                         tardis.apps.deep_storage_download_mapper.mapper),
declared_fields (tardis.tardis_portal.forms.ImportParamsForm89
    attribute), 283                                         DEFAULT_ARCHIVE_FORMATS      (in      module
declared_fields (tardis.tardis_portal.forms.LoginForm      tardis.default_settings.downloads), 153
    attribute), 283                                         DEFAULT_FROM_EMAIL      (in      module
declared_fields (tardis.tardis_portal.forms.ManageAccountFor tardis.default_settings.email), 154
    attribute), 283                                         DEFAULT_STORAGE_BASE_DIR
declared_fields (tardis.tardis_portal.forms.ManageGroupPerm (tardis.default_settings attribute), 59
    attribute), 284                                         dehydrate () (tardis.apps.sftp.api.SFTPPublicKeyAppResource
declared_fields (tardis.tardis_portal.forms.RegisterExperimentMethod), 141
    attribute), 284                                         dehydrate () (tardis.tardis_portal.api.DatasetResource
declared_fields (tardis.tardis_portal.forms.RegistrationForm method), 271
    attribute), 285                                         dehydrate () (tardis.tardis_portal.api.ExperimentResource
declared_fields (tardis.tardis_portal.forms.RightsForm       method), 273
    attribute), 285                                         dehydrate () (tardis.tardis_portal.api.ReplicaResource
declared_filters (tardis.tardis_portal.graphql.datafile.DataFileTypeFilter), 276
    attribute), 164                                         dehydrate () (tardis.tardis_portal.api.UserResource
declared_filters (tardis.tardis_portal.graphql.datafileobject.DataFileObjectFilter), 278
    attribute), 165                                         delete_all_params ()
declared_filters (tardis.tardis_portal.graphql.dataset.DatasetTypeFilter), 279
    attribute), 165                                         tardis.tardis_portal.ParameterSetManager.ParameterSetManager
method), 263
declared_filters (tardis.tardis_portal.graphql.experiment.ExperimentAuthorisedTypeFilter), 280
    attribute), 166                                         portal.models.datafile.DataFileObject
method), 195
declared_filters (tardis.tardis_portal.graphql.experiment.ExperimentAuthorisedTypeFilter), 281
    attribute), 167                                         tardis.apps.sftp.api.SFTPAclAuthorization
method), 141
declared_filters (tardis.tardis_portal.graphql.facility.FacilityTypeFilter) (tardis.tardis_portal.api.ACIAuthorization
    attribute), 168                                         method), 269
declared_filters (tardis.tardis_portal.graphql.group.GroupTypeFilter)      (in      module
    attribute), 168                                         tardis.tardis_portal.models.datafile), 196
declared_filters (tardis.tardis_portal.graphql.instrument.InstrumentTypeFilter content ()      (in      module
    attribute), 169                                         tardis.tardis_portal.tests.slapd), 248
declared_filters (tardis.tardis_portal.graphql.parameter.DatafileParameterSetTypeFilter), 282
    attribute), 170                                         tardis.tardis_portal.api.ACIAuthorization
method), 269
declared_filters (tardis.tardis_portal.graphql.parameter.DatafileParameterSetTypeFilter), 283
    attribute), 171                                         tardis.tardis_portal.ParameterSetManager.ParameterSetManager
method), 263
declared_filters (tardis.tardis_portal.graphql.parameter.DatasetParameterSetTypeFilter), 284
    attribute), 171                                         tardis.tardis_portal.auth.decorators), 159
declared_filters (tardis.tardis_portal.graphql.parameter.DatasetParameterTypeFilter) (in      module
    attribute), 172                                         tardis.tardis_portal.models.datafile.DataFile
method), 263
declared_filters (tardis.tardis_portal.graphql.parameters.ExperimentParameterSetTypeFilter
    attribute), 172                                         deleted (tardis.tardis_portal.models.datafile.DataFile
method), 263
declared_filters (tardis.tardis_portal.graphql.parameters.ExperimentParameterTypeFilter
    attribute), 173                                         deleted_time (tardis.tardis_portal.models.datafile.DataFile
method), 263
declared_filters (tardis.tardis_portal.graphql.parameters.ParameterSetNameTypeFilter
    attribute), 173                                         DeleteTestCase      (class
DeleteTestCase      (class
    attribute), 174                                         in
declared_filters (tardis.tardis_portal.graphql.parameters.TardisFileUploadTypeFilter), 285
    attribute), 174                                         dependencies (tardis.apps.eventlog.migrations.0001_initial.Migration
method), 263

```

attribute), 90
 dependencies (*tardis.apps.hsm.migrations.0001_initial_dataMigration* (*tardis.tardis_portal.models.experiment.Experiment attribute*), 94
 dependencies (*tardis.apps.openid_migration.migrations.0001_initial_Migration* (*tardis.tardis_portal.models.storage.StorageBox attribute*), 106
 dependencies (*tardis.apps.publication_workflow.migrations.0001_initial_Migration* (*tardis.tardis_portal.api.DataFileResource method*), 111
 dependencies (*tardis.apps.push_to.migrations.0001_initial_Migration* (*allowed_methods attribute*), 119
 dependencies (*tardis.apps.push_to.migrations.0002_auto_20160518_1818_Migration* (*detail_uri_kwarg*s ()
 dependencies (*tardis.apps.push_to.migrations.0003_auto_20210216rdi32q_Migration* (*api.AdvanceSearchAppResource method*), 136
 dependencies (*tardis.apps.sftp.migrations.0001_initial_Migration* (*uri_kwarg*s ()
 dependencies (*tardis.tardis_portal.migrations.0001_initial_Migration* (*method*), 137
 dependencies (*tardis.tardis_portal.migrations.0001_initial_Migration* (*attribute*), 181
 dependencies (*tardis.tardis_portal.migrations.0001_squashed_001_kbd1utapp20d160505pd64f_Migration* (*df_dir_only ()* (*in module*
 dependencies (*tardis.tardis_portal.migrations.0002_auto_20150528_1128_Migration* (*attribute*), 182
 dependencies (*tardis.tardis_portal.migrations.0003_auto_20150907_1315_Migration* (*attribute*), 182
 dependencies (*tardis.tardis_portal.migrations.0004_storageboxopitributeduel9type_Migration* (*attribute*), 182
 dependencies (*tardis.tardis_portal.migrations.0005_datafile_add_attributes_column_Migration* (*attribute*), 182
 dependencies (*tardis.tardis_portal.migrations.0006_datafile_remove_attributes_striking_column_Migration* (*attribute*), 183
 dependencies (*tardis.tardis_portal.migrations.0007_remove_parameter_indexing_value_index_Migration* (*attribute*), 183
 dependencies (*tardis.tardis_portal.migrations.0008_string_value_attributes_index_postgres_Migration* (*attribute*), 183
 dependencies (*tardis.tardis_portal.migrations.0009_auto_20160128_ribul0_Migration* (*attribute*), 183
 dependencies (*tardis.tardis_portal.migrations.0010_auto_20160509rdi414_Migration* (*attribute*), 183
 dependencies (*tardis.tardis_portal.migrations.0011_auto_20160505_H44_Migration* (*attribute*), 184
 dependencies (*tardis.tardis_portal.migrations.0012_useAuthenticationApproved_Migration* (*attribute*), 184
 dependencies (*tardis.tardis_portal.migrations.0013_auto_20180327_1736_Migration* (*attribute*), 184
 dependencies (*tardis.tardis_portal.migrations.0014_auto_20181002_1d54_Migration* (*attribute*), 184
 dependencies (*tardis.tardis_portal.migrations.0015_dataset_create08_time_Migration* (*attribute*), 184
 dependencies (*tardis.tardis_portal.migrations.0016_add_timestamps_Migration* (), (*in module tardis.tardis_portal.download*), 280
 dependencies (*tardis.tardis_portal.migrations.0017_add_overlicenses_Migration* (), (*in module tardis.tardis_portal.download*), 280
 dependencies (*tardis.tardis_portal.migrations.0018_make_default_storage_by_tardisapp until Migration* (*ReplicaAppResource method*), 133
 description (*tardis.tardis_portal.models.dataset.Dataset.download_file ()* (*tardis.tardis_portal.api.DataFileResource attribute*), 197
 attribute), 201
 attribute), 226
 method), 270
 detail_uri_kwarg
 df_dir_only ()
 89
 method), 97
 directory (*tardis.tardis_portal.models.datafile.DataFile attribute*), 97
 directory (*tardis.tardis_portal.models.dataset.Dataset attribute*), 97
 DISK (*tardis.tardis_portal.models.storage.StorageBox attribute*), 97
 django (*tardis.apps.search.documents.DataFileDocument attribute*), 97
 django (*tardis.apps.search.documents.DatasetDocument attribute*), 97
 django (*tardis.apps.search.documents.ExperimentDocument attribute*), 97
 django_storage_class
 227
 tardis.tardis_portal.auth.localdb_auth), 162
 tardis.tardis_portal.auth.localdb_auth), 162
 tardis.tardis_portal.auth.localdb_auth), 162
 tardis.tardis_portal.auth.localdb_auth), 162
 tardis.apps.openid_migration.migration), 114
 DOI (class in *tardis.apps.publication_workflow.doi*), 114
 tardis.tardis_portal.download), 280
 tardis.tardis_portal.download), 280
 tardis.tardis_portal.download), 280
 method), 133
 tardis.tardis_portal.download_file () (tardis.tardis_portal.api.DataFileResource

method), 270
download_image() (in module *tardis.tardis_portal.iiif*), 286
download_info() (in module *tardis.tardis_portal.iiif*), 286
DOWNLOAD_URI_TEMPLATES (in module *tardis.default_settings.downloads*), 153
download_url(*tardis.tardis_portal.models.datafile.DataFile*.attribute), 191
DownloadTestCase (class in *tardis.tardis_portal.tests.test_download*), 250
draft_publications() (in module *tardis.apps.publication_workflow.managers.PublicationManager*.*default_settings.email*), 154
method), 114
draft_publications() (in module *tardis.apps.publication_workflow.models.PublicationManager*.*success*), 154
method), 115
DummyStorage (class in *tardis.tardis_portal.storage*), 232
DumpSchemasTestCase (class in *tardis.tardis_portal.tests.management.test_dumpschemas*), 242
DynamicTree (class in *tardis.apps.sftp.sftp*), 144
dynurl() (in module *tardis.tardis_portal.templatetags.dynurl*), 234
DynUrlNode (class in *tardis.tardis_portal.templatetags.dynurl*), 234

E

edge_count (*tardis.tardis_portal.graphql.utils.ExtendedConnection*.attribute), 14
attribute), 179
edit_auth_method() (in module *tardis.tardis_portal.auth.authentication*), 157
edit_datafile_par() (in module *tardis.tardis_portal.views.parameters*), 262
edit_dataset() (in module *tardis.tardis_portal.views.pages*), 261
edit_dataset_par() (in module *tardis.tardis_portal.views.parameters*), 262
edit_experiment_par() (in module *tardis.tardis_portal.views.parameters*), 262
edit_parameters() (in module *tardis.tardis_portal.views.parameters*), 262
EditParameterSetTestCase (class in *tardis.tardis_portal.tests.test_paramtersets*), 253
effectiveDate (*tardis.tardis_portal.models.access_control.ObjectACL*.attribute), 187
ELASTICSEARCH_PARALLEL_INDEX_SETTINGS (in module *tardis.default_settings.search*), 155
email (*tardis.tardis_portal.models.experiment.ExperimentAuthor*.attribute), 204
email_dataset_recall_requested() (in module *tardis.apps.hsm.email_text*), 98
email_dfo_recall_complete() (in module *tardis.apps.hsm.email_text*), 98
email_dfo_recall_failed() (in module *tardis.apps.hsm.email_text*), 98
email_dfo_recall_requested() (in module *tardis.apps.hsm.email_text*), 98
EMAIL_HOST (in module *tardis.default_settings.email*), 154
EMAIL_HOST_PASSWORD (in module *tardis.default_settings.email*), 154
EMAIL_HOST_USER (in module *tardis.default_settings.email*), 154
EMAIL_PORT (in module *tardis.default_settings.email*), 154
tardis.apps.hsm_pub_rejected() (in module *tardis.apps.publication_workflow.email_text*), 114
email_pub_released() (in module *tardis.apps.publication_workflow.email_text*), 114
email_pub_retracted() (in module *tardis.apps.publication_workflow.email_text*), 114
email_pub_reverted_to_draft() (in module *tardis.apps.publication_workflow.email_text*), 114
EMAIL_USE_TLS (in module *tardis.default_settings.email*), 154
email_user() (in module *tardis.tardis_portal.email*), 280
EmailTemplates (class in *tardis.apps.publication_workflow.tests.test_email_templates*), 112
end_time (*tardis.tardis_portal.models.experiment.Experiment*.attribute), 201
endpoint() (in module *tardis.apps.oaipmh.views*), 105
EndpointTestCase (class in *tardis.apps.oaipmh.tests.test_oai*), 103
entityId (*tardis.tardis_portal.models.access_control.ObjectACL*.attribute), 187
error_handler() (in module *tardis.views*), 294
EXACT_VALUE_COMPARISON
tardis.tardis_portal.models.parameters.ParameterName.attribute), 219
Experiment (class in *tardis.tardis_portal.models.experiment*), 155

200
 experiment (*tardis.tardis_portal.models.experiment.ExperimentAuthor*) AuthorType (class in attribute), 204
 experiment (*tardis.tardis_portal.models.parameters.ExperimentParameterSet*) ExperimentAuthorTypeFilter (class in attribute), 214
 EXPERIMENT (*tardis.tardis_portal.models.parameters.Schema*) tardis.tardis_portal.graphql.experiment), 166
 experiment (*tardis.tardis_portal.models.token.Token*) ExperimentAuthorTypeFilter.Meta (class in attribute), 230
 Experiment.DoesNotExist, 200
 Experiment.MultipleObjectsReturned, 200
 experiment_access_required() (in module tardis.tardis_portal.auth.decorators), 159
 experiment_authors() (in module tardis.tardis_portal.templatetags.experiment_tags), 234
 experiment_download_link() (in module tardis.tardis_portal.templatetags.experiment_tags) ExperimentForm (class in tardis.tardis_portal.forms), 282
 experiment_download_required() (in module tardis.tardis_portal.auth.decorators), 159
 experiment_file_count() (in module tardis.tardis_portal.templatetags.experimentstats), 234
 experiment_id (*tardis.tardis_portal.models.experiment.Experiment*) ExperimentManager (class in attribute), 204
 experiment_id (*tardis.tardis_portal.models.parameters.ExperimentParameterSet*) ExperimentParameterSet (class in attribute), 215
 experiment_id (*tardis.tardis_portal.models.token.Token*) ExperimentParameter.DoesNotExist, 213
 experiment_ownership_required() (in module tardis.tardis_portal.auth.decorators), 159
 experiment_set (*tardis.tardis_portal.models.license.License*) Experimentparameter_Set (tardis.tardis_portal.models.parameters.ExperimentParameterSet), 213
 EXPERIMENT_VIEWS (in module tardis.default_settings.custom_views), 152
 ExperimentAdmin (class in tardis.tardis_portal.admin), 265
 ExperimentAuthor (class in tardis.tardis_portal.forms), 281
 ExperimentAuthor (class in tardis.tardis_portal.models.experiment), 204
 ExperimentAuthor.DoesNotExist, 204
 ExperimentAuthor.Meta (class in tardis.tardis_portal.forms), 281
 ExperimentAuthor.MultipleObjectsReturned, 204
 experimentauthor_set (*tardis.tardis_portal.models.experiment.Experiment*) ExperimentParameterSet (class in attribute), 214
 ExperimentAuthorResource (class in tardis.tardis_portal.api), 272
 ExperimentAuthorResource.Meta (class in experimentparameterset_Set)

(*tardis.tardis_portal.models.experiment.Experiment attribute*), 201
 experimentparameterset_set (*tardis.tardis_portal.models.parameters.Schema attribute*), 224
 ExperimentParameterSetAdmin (class in *tardis.tardis_portal.admin*), 265
 ExperimentParameterSetResource (class in *tardis.tardis_portal.api*), 272
 ExperimentParameterSetResource.Meta (class in *tardis.tardis_portal.api*), 272
 ExperimentParameterSetResourceTest (class in *tardis.tardis_portal.tests.api.test_experiment_metadata_restrictions*), 187
 237
 experimentparametersets (*tardis.tardis_portal.models.storage.StorageBox attribute*), 227
 ExperimentParameterSetType (class in *tardis.tardis_portal.graphql.parameters*), 172
 ExperimentParameterSetTypeFilter (class in *tardis.tardis_portal.graphql.parameters*), 172
 ExperimentParameterSetTypeFilter.Meta (class in *tardis.tardis_portal.graphql.parameters*), 172
 ExperimentParameterType (class in *tardis.tardis_portal.graphql.parameters*), 172
 ExperimentParameterTypeFilter (class in *tardis.tardis_portal.graphql.parameters*), 172
 ExperimentParameterTypeFilter.Meta (class in *tardis.tardis_portal.graphql.parameters*), 173
 ExperimentResource (class in *tardis.tardis_portal.api*), 273
 ExperimentResource.Meta (class in *tardis.tardis_portal.api*), 273
 ExperimentResourceTest (class in *tardis.tardis_portal.tests.api.test_experiment_resource_restrictions*), 237
 experiments (*tardis.apps.sftp.sftp.MyTSFTPServerInterface attribute*), 144
 experiments (*tardis.tardis_portal.graphql.schema.tardisQuery attribute*), 176
 experiments (*tardis.tardis_portal.models.dataset.Dataset attribute*), 197
 ExperimentTestCase (class in *tardis.tardis_portal.tests.models.test_experiment*), 243
 ExperimentTestCase (class in *tardis.tardis_portal.tests.views.test_experiment_views*), 245
 ExperimentType (class in *tardis.tardis_portal.graphql.experiment*), 166
 ExperimentTypeFilter (class in *tardis.tardis_portal.graphql.experiment*), 166
 ExperimentTypeFilter.Meta (class in *tardis.tardis_portal.graphql.experiment*), 167
 ExperimentView (class in *tardis.tardis_portal.views.pages*), 259
 expiry_date (*tardis.tardis_portal.models.token.Token attribute*), 230
 expiryDate (*tardis.tardis_portal.models.access_control.ObjectACL attribute*), 187
 ext_groups (*tardis.tardis_portal.models.access_control.UserProfile attribute*), 189
 ExtendedConnection (class in *tardis.tardis_portal.graphql.utils*), 179
 external_users () (*tardis.tardis_portal.managers.ExperimentManager method*), 287
 extra (*tardis.apps.eventlog.models.Log attribute*), 92
 extra (*tardis.tardis_portal.admin.DataFileObjectInline attribute*), 263
 extra (*tardis.tardis_portal.admin.ExperimentParameterInline attribute*), 265
 extra (*tardis.tardis_portal.admin.InstrumentParameterInline attribute*), 265
 extra (*tardis.tardis_portal.admin.ObjectACLInline attribute*), 266
 extra (*tardis.tardis_portal.admin.ParameterNameInline attribute*), 266
 extra (*tardis.tardis_portal.admin.StorageBoxAttributeInline attribute*), 266
 extra (*tardis.tardis_portal.admin.StorageBoxOptionInline attribute*), 268
 ExtraTestCases (class in *tardis.tardis_portal.tests.test_iif*), 251
 F
 facilities (*tardis.tardis_portal.graphql.schema.tardisQuery attribute*), 176
 facilities_managed_by () (in module *tardis.tardis_portal.models.facility*), 206
 facilitiesTestCase (class in *tardis.tardis_portal.tests.graphql.test_facilities*), 240
 Facility (class in *tardis.tardis_portal.models.facility*), 205
 facility (*tardis.tardis_portal.models.instrument.Instrument attribute*), 206
 Facility.DoesNotExist, 205
 Facility.MultipleObjectsReturned, 205
 facility_id (*tardis.tardis_portal.models.instrument.Instrument attribute*), 207

facility_overview() (in module tardis.tardis_portal.views.pages), 261
 facility_overview_data_count() (in module tardis.tardis_portal.views.facilities), 258
 facility_overview_datafile_list() (in module tardis.tardis_portal.views.facilities), 258
 facility_overview_dataset_detail() (in module tardis.tardis_portal.views.facilities), 258
 facility_overview_experiments() (in module tardis.tardis_portal.views.facilities), 258
 facility_overview_facilities_list() (in module tardis.tardis_portal.views.facilities), 258
 FacilityAdmin (class in tardis.tardis_portal.admin), 265
 FacilityOverviewTestCase (class in tardis.tardis_portal.tests.test_facility_overview), 251
 FacilityResource (class in tardis.tardis_portal.api), 273
 FacilityResource.Meta (class in tardis.tardis_portal.api), 273
 FacilityResourceTest (class in tardis.tardis_portal.tests.api.test_facility_resource), 237
 FacilityType (class in tardis.tardis_portal.graphql.facility), 167
 FacilityTypeFilter (class in tardis.tardis_portal.graphql.facility), 167
 FacilityTypeFilter.Meta (class in tardis.tardis_portal.graphql.facility), 167
 feedback() (in module tardis.tardis_portal.views.utils), 262
 fetch_experiments_and_datasets() (in module tardis.apps.publication_workflow.views), 117
 fields (tardis.apps.push_to.models.CredentialForm.Meta attribute), 122
 fields (tardis.apps.push_to.models.RemoteHostAdmin attribute), 126
 fields (tardis.tardis_portal.admin.DatafileAdminForm.Meta attribute), 264
 fields (tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta attribute), 264
 fields (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta attribute), 267
 fields (tardis.tardis_portal.admin.StorageBoxForm.Meta attribute), 267
 fields (tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute), 268
 fields (tardis.tardis_portal.api.UserResource.Meta attribute), 278
 fields (tardis.tardis_portal.forms.DatasetForm.Meta attribute), 281
 fields (tardis.tardis_portal.forms.ExperimentAuthor.Meta attribute), 281
 fields (tardis.tardis_portal.forms.ExperimentForm.Meta attribute), 282
 fields (tardis.tardis_portal.forms.ManageAccountForm.Meta attribute), 283
 fields (tardis.tardis_portal.forms.RightsForm.Meta attribute), 285
 fields (tardis.tardis_portal.graphql.datafile.DataFileTypeFilter.Meta attribute), 164
 fields (tardis.tardis_portal.graphql.datafileobject.DataFileObjectTypeFilter.Meta attribute), 165
 fields (tardis.tardis_portal.graphql.dataset.DatasetTypeFilter.Meta attribute), 165
 fields (tardis.tardis_portal.graphql.experiment.ExperimentAuthorTypeFilter.Meta attribute), 166
 fields (tardis.tardis_portal.graphql.experiment.ExperimentTypeFilter.Meta attribute), 167
 fields (tardis.tardis_portal.graphql.facility.FacilityTypeFilter.Meta attribute), 167
 fields (tardis.tardis_portal.graphql.group.GroupTypeFilter.Meta attribute), 168
 fields (tardis.tardis_portal.graphql.instrument.InstrumentTypeFilter.Meta attribute), 169
 fields (tardis.tardis_portal.graphql.parameters.DatafileParameterSetTypeFilter.Meta attribute), 170
 fields (tardis.tardis_portal.graphql.parameters.DatafileParameterTypeFilter.Meta attribute), 171
 fields (tardis.tardis_portal.graphql.parameters.DatasetParameterSetTypeFilter.Meta attribute), 171
 fields (tardis.tardis_portal.graphql.parameters.DatasetParameterTypeFilter.Meta attribute), 172
 fields (tardis.tardis_portal.graphql.parameters.ExperimentParameterSetTypeFilter.Meta attribute), 172
 fields (tardis.tardis_portal.graphql.parameters.ExperimentParameterTypeFilter.Meta attribute), 173
 fields (tardis.tardis_portal.graphql.parameters.ParameterNameTypeFilter.Meta attribute), 173
 fields (tardis.tardis_portal.graphql.parameters.TardisSchemaTypeFilter.Meta attribute), 174
 fields (tardis.tardis_portal.graphql.storagebox.StorageBoxTypeFilter.Meta attribute), 177
 fields (tardis.tardis_portal.graphql.storageboxattribute.StorageBoxAttributeTypeFilter.Meta attribute), 178
 fields (tardis.tardis_portal.graphql.storageboxoption.StorageBoxOptionTypeFilter.Meta attribute), 179
 file_is_online() (in module tardis.apps.hsm.utils), 99
 file_object (tardis.tardis_portal.models.datafile.DataFile attribute), 191
 file_object (tardis.tardis_portal.models.datafile.DataFileObject attribute), 195

```

file_objects (tardis.tardis_portal.models.datafile.DataFile) (tardis.tardis_portal.admin.DatafileAdmin attribute), 191
file_objects (tardis.tardis_portal.models.storage.StorageBox) (tardis.tardis_portal.admin.DataFileObjectInline attribute), 227
filename (tardis.tardis_portal.models.datafile.DataFile) form (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 192
FILENAME (tardis.tardis_portal.models.parameters.ParameterName) (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 219
filepicker_url_regex (tardis.apps.filepicker.utils.FilepickerFile attribute), 93
FilepickerFile (class in tardis.apps.filepicker.utils), 93
filtering (tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta) app_name_for_url () (in module tardis.app_config), 294
filtering (tardis.tardis_portal.api.DataFileResource.Meta) formfield_overrides (attribute), 269
filtering (tardis.tardis_portal.api.DatasetResource.Meta) forward_func () (attribute), 265
filtering (tardis.tardis_portal.api.ExperimentAuthorResource.Meta) forward_func () (attribute), 271
filtering (tardis.tardis_portal.api.ExperimentResource.Meta) forwards_func () (attribute), 273
filtering (tardis.tardis_portal.api.FacilityResource.Meta) forwards_func () (attribute), 273
filtering (tardis.tardis_portal.api.GroupResource.Meta) forwards_func () (attribute), 274
filtering (tardis.tardis_portal.api.InstrumentResource.Meta) freeTextSearchField (attribute), 274
filtering (tardis.tardis_portal.api.ObjectACLResource.Meta) freeTextSearchField (attribute), 275
filtering (tardis.tardis_portal.api.ParameterNameResource.Meta) freeTextSearchField..DoesNotExist, 215
filtering (tardis.tardis_portal.api.ReplicaResource.Meta) freeTextSearchField.MultipleObjectsReturned, 215
filtering (tardis.tardis_portal.api.SchemaResource.Meta) freeTextSearchfield_set (attribute), 276
filtering (tardis.tardis_portal.api.UserResource.Meta) freezeTextSearchFieldAdmin (attribute), 221
FILTERS_TASK_PRIORITY (in module tardis.default_settings.filters), 154
finalize_publication () (in module tardis.apps.publication_workflow.views), 117
find_available_tcp_port () (in module tardis.tardis_portal.tests.slapd), 248
find_custom_view_override () (tardis.tardis_portal.views.pages.DatasetView method), 259
find_custom_view_override () (tardis.tardis_portal.views.pages.ExperimentView method), 260
form (tardis.apps.push_to.models.CredentialAdmin attribute), 121
form (tardis.tardis_portal.admin.DatafileAdmin attribute), 264
form (tardis.tardis_portal.admin.DataFileObjectInline attribute), 263
form (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 266
form (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 267
form (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 268
form_view () (in module tardis.apps.publication_workflow.views), 117
form (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 268
form_view () (in module tardis.apps.publication_workflow.views), 117
form (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 267
form (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 265
form (tardis.tardis_portal.admin.ExperimentParameterInline attribute), 269
form (tardis.tardis_portal.admin.ExperimentParameterInline attribute), 269
forward_func () (in module tardis.apps.hsm.migrations.0001_initial_data), 94
forwards_func () (in module tardis.apps.publication_workflow.migrations.0001_initial), 111
forwards_func () (in module tardis.tardis_portal.migrations.0017_add_cc_licenses), 185
forwards_func () (in module tardis.tardis_portal.models.parameters), 215
FreeTextSearchField.DoesNotExist, 215
FreeTextSearchField.MultipleObjectsReturned, 215
FrozenTime (class in tardis.tardis_portal.models.parameters.ParameterName attribute), 221
FrozenTime (class in tardis.tardis_portal.admin), 265
FrozenTime (class in tardis.tardis_portal.tests.test_tokens.FrozenTime class method), 255
FrozenTime (class in tardis.tardis_portal.tests.test_tokens), 255
from_db_value () (tardis.apps.eventlog.fields.JSONField method), 91
full_name (tardis.tardis_portal.models.parameters.ParameterName attribute), 221
generate_keypair_credential () (tardis.apps.push_to.models.Credential static method), 120
generate_presigned_url () (in module tardis.apps.s3utils.utils), 134

```

G

```

generate_keypair_credential () (tardis.apps.push_to.models.Credential static method), 120
generate_presigned_url () (in module tardis.apps.s3utils.utils), 134

```


`tardis.apps.push_to.views), 129`

`get_credentials()`

`(tardis.tardis_portal.tests.api.MyTardisResourceTestCase method), 240`

`get_ct () (tardis.tardis_portal.models.experiment.Experiment method), 202`

`get_data_type_display ()`

`(tardis.tardis_portal.models.parameters.ParameterName method), 221`

`get_datafiles () (tardis.tardis_portal.api.DatasetReso`

`get_datafiles () (tardis.tardis_portal.models.dataset.Dataset method), 197`

`get_datafiles () (tardis.tardis_portal.models.experiment.Experiment method), 202`

`get_dataset_info () (in module tardis.tardis_portal.views.utils), 262`

`get_default_push_location () (in module tardis.apps.push_to.utils), 129`

`get_default_storage ()`

`(tardis.tardis_portal.models.storage.StorageBox class method), 227`

`get_default_storage_box ()`

`(tardis.tardis_portal.models.datafile.DataFile method), 192`

`get_details_schema ()`

`(tardis.apps.publication_workflow.models.Publication static method), 116`

`get_details_schema_pset ()`

`(tardis.apps.publication_workflow.models.Publication method), 116`

`get_dir_nodes () (tardis.tardis_portal.models.dataset.Dataset method), 197`

`get_dir_tuples () (tardis.tardis_portal.models.dataset.Dataset method), 198`

`get_dn_suffix () (tardis.tardis_portal.tests.slapd.Slapd method), 247`

`get_download_organizations () (in module tardis.tardis_portal.download), 280`

`get_download_urls ()`

`(tardis.tardis_portal.models.dataset.Dataset method), 199`

`get_download_urls ()`

`(tardis.tardis_portal.models.experiment.Experiment method), 202`

`get_draft_publication () (in module tardis.apps.publication_workflow.views), 117`

`get_edit_url () (tardis.tardis_portal.models.dataset.Dataset method), 199`

`get_edit_url () (tardis.tardis_portal.models.experiment.Experiment method), 202`

`get_effective_query ()`

`(tardis.tardis_portal.models.access_control.ObjectACL class method), 187`

`get_experiment_list () (in module tardis.tardis_portal.views.ajax_json), 257`

`get_experiment_referer () (in module tardis.tardis_portal.shortcuts), 289`

`get_file () (tardis.apps.filepicker.utils.FilepickerFile method), 93`

`get_filesystem_safe_dataset_name () (in module tardis.tardis_portal.util), 290`

`get_filesystem_safe_experiment_name () (in module tardis.tardis_portal.util), 290`

`get_filesystem_safe_experiment ()`

`(tardis.tardis_portal.models.dataset.Dataset method), 199`

`get_form_state_parameter ()`

`(tardis.apps.publication_workflow.models.Publication method), 116`

`get_full_path () (tardis.tardis_portal.models.datafile.DataFileObject method), 195`

`get_git_version () (in module tardis.default_settings), 157`

`get_groups () (tardis.tardis_portal.models.experiment.Experiment method), 202`

`get_id () (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider static method), 101`

`get_identifier () (tardis.tardis_portal.api.MyTardisAuthentication method), 274`

`get_image_data () (tardis.tardis_portal.models.datafile.DataFile method), 192`

`get_images () (tardis.tardis_portal.models.dataset.Dataset method), 199`

`get_instances_from_related ()`

`(tardis.apps.search.documents.DataFileDocument method), 138`

`get_instances_from_related ()`

`(tardis.apps.search.documents.DatasetDocument method), 138`

`get_instances_from_related ()`

`(tardis.apps.search.documents.ExperimentDocument method), 138`

`get_item () (in module tardis.tardis_portal.templatetags.lookupfilters), 235`

`get_license_uri ()`

`(tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider method), 144`

```

        method), 254
get_licenses()           (in      module get_or_create_directory()
                        tardis.apps.publication_workflow.views),
                        117
get_local_time()         (in      module get_or_create_user()       (in      module
                        tardis.tardis_portal.util), 290
get_matching_auth_provider() (in      module get_or_create_user()
                            tardis.apps.openid_migration.migration), 108
get_matching_authmethod() (in      module get_or_create_user()
                           tardis.tardis_portal.templatetags.approved_user_tags),
                           159
                           get_owned_experiments() (in      module
                           233
                           tardis.tardis_portal.auth.decorators), 160
get_mimetype()           (tardis.tardis_portal.models.datafile.DataFileOwners()
                           method), 192
get_next_by_added()       (tardis.apps.sftp.models.SFTPPublicKey
                           method), 143
get_next_by_created_time() (tardis.tardis_portal.models.datafile.DataFileObject
                           method), 195
get_next_by_created_time() (tardis.tardis_portal.models.experiment.Experiment
                           method), 202
get_next_by_created_time() (tardis.tardis_portal.models.jti.JTI
                           method), 208
get_next_by_expiry_date() (tardis.tardis_portal.models.token.Token
                           method), 231
get_next_by_migration_timestamp() (tardis.apps.openid_migration.models.OpenidUserMigration
                                    method), 109
get_next_by_timestamp()    (tardis.apps.eventlog.models.Log
                           method), 92
get_next_by_timestamp()    (tardis.apps.push_to.models.Request
                           method), 127
get_next_by_update_time()  (tardis.tardis_portal.models.experiment.Experiment
                           method), 202
get_none_option_license() (tardis.tardis_portal.models.license.License
                           class method), 208
get_oauth_service()        (tardis.apps.push_to.models.OAuthSSHCertSigningService
                           static method), 124
get_object_list()          (tardis.apps.search.api.AdvanceSearchAppResource
                           method), 136
get_object_list()          (tardis.apps.search.api.SearchAppResource
                           method), 137
get_options_as_dict()      (tardis.tardis_portal.models.storage.StorageBox
                           method), 202
method), 227
(tardis.tardis_portal.models.experiment.Experiment
method), 202
(tardis.tardis_portal.auth.utils), 163
(tardis.tardis_portal.auth.authservice.AuthService
method), 159
(tardis.tardis_portal.auth.decorators), 160
method), 202
get_params()              (tardis.tardis_portal.ParameterSetManager.ParameterSetM
                           method), 263
get_params()              (tardis.tardis_portal.ParameterSetManager.ParameterSetM
                           method), 263
get_path()                (tardis.tardis_portal.models.dataset.Dataset
                           method), 199
get_perm_bool()            (tardis.tardis_portal.auth.authorisation.ACIAwareB
                           method), 158
get_preferred_dfo()        (tardis.tardis_portal.models.datafile.DataFile
                           method), 192
get_prep_value()           (tardis.apps.eventlog.fields.JSONField
                           method), 91
get_previous_by_added()   (tardis.apps.sftp.models.SFTPPublicKey
                           method), 143
get_previous_by_created_time() (tardis.tardis_portal.models.datafile.DataFileObject
                           method), 195
get_previous_by_created_time() (tardis.tardis_portal.models.experiment.Experiment
                           method), 202
get_previous_by_created_time() (tardis.tardis_portal.models.jti.JTI
                           method), 208
get_previous_by_expiry_date() (tardis.tardis_portal.models.token.Token
                           method), 231
get_previous_by_migration_timestamp() (tardis.apps.openid_migration.models.OpenidUserMigration
                           method), 109
get_previous_by_timestamp() (tardis.apps.eventlog.models.Log
                           method), 92
get_previous_by_timestamp() (tardis.apps.push_to.models.Request
                           method), 127
get_previous_by_update_time() (tardis.tardis_portal.models.experiment.Experiment
                           method), 202
get_pub_admin_email_addresses() (in mod-
```

```

ule      tardis.apps.publication_workflow.utils),  get_size() (tardis.tardis_portal.models.datafile.DataFile
117                               method), 192
get_public_access_display()          get_size() (tardis.tardis_portal.models.dataset.Dataset
                                         method), 199
                                         get_size() (tardis.tardis_portal.models.experiment.Experiment
                                         method), 202
get_push_url_for_host()   (in      module      method), 202
                           tardis.push_to.views), 130  get_size_and_sha512sum()   (in      module
get_queryset() (tardis.apps.search.documents.DataFileDocument)  tardis.tardis_portal.tests.test_download),
                                         250
                                         method), 138
get_queryset() (tardis.tardis_portal.managers.OracleSafeManager)
                                         credential()  (tardis.push_to.models.Credential  static
                                         method), 288
                                         method), 120
get_receiving_storage_box()        (tardis.tardis_portal.models.datafile.DataFile
                                         method), 192  get_suitable_licenses()
                                         (tardis.tardis_portal.models.license.License
                                         class method), 208
get_related_object()              (tardis.tardis_portal.models.access_control.ObjectACL)
                                         tardis_apps() (in module tardis.app_config),
                                         method), 187  294
get_related_object_group()        (tardis.tardis_portal.models.access_control.ObjectACL
                                         method), 187  get_template() (tardis.tardis_portal.publish.provider.rifcsprovider.Rif
                                         method), 232
                                         method), 187  get_template() (tardis.tardis_portal.publish.publishservice.PublishSe
get_release_date()    (in      module      method), 232
                           tardis.publication_workflow.tasks),
                                         117  get_template() (tardis.tardis_portal.tests.test_publishservice.MockRif
                                         method), 254
get_request_data()    (in      module      get_thumbnail_url()
                           tardis.eventlog.utils), 93  (tardis.tardis_portal.models.dataset.Dataset
                                         method), 254
get_response() (tardis.tardis_portal.download.UncachedTarStream)
                                         method), 199  get_tmpdir() (tardis.tardis_portal.tests.slapd.Slapd
                                         method), 279  method), 247
get_rifcs_context()   (tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider)
                                         (in      module
                                         method), 232  tardis.push_to.oauth_tokens), 128
get_rifcs_context()   (tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider)
                                         (in      module
                                         method), 254  tardis.push_to.oauth_tokens), 128
get_rifcs_id() (tardis.apps.oaipmh.provider.experiment.RifCsExperiment)
                                         tardis.tardis_portal.models.parameters.Schema
                                         static method), 101  method), 224
get_root_dir_nodes() (tardis.tardis_portal.api.DatasetResource
                                         method), 272  get_url() (tardis.tardis_portal.tests.slapd.Slapd
                                         method), 247
get_root_dn() (tardis.tardis_portal.tests.slapd.Slapd
                                         method), 247  get_user() (tardis.tardis_portal.auth.interfacesAuthProvider
                                         method), 161
get_root_password() (tardis.tardis_portal.tests.slapd.Slapd method),
                                         247  get_user() (tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend
                                         method), 162
get_schema() (tardis.tardis_portal.ParameterSetManager)
                                         ParameterSetManager
                                         method), 263  get_utc_time() (in module tardis.tardis_portal.util),
                                         290
get_server() (in module tardis.apps.oaipmh.server),
                                         105  getParameterSetManager()
                                         display()
                                         (tardis.tardis_portal.models.storage.StorageBoxOption
                                         method), 230
get_session_expiry() (tardis.tardis_portal.models.token.Token
                                         method), 231  getAuthMethodChoices() (in      module
                                         tardis.tardis_portal.forms), 286
get_shared_experiments() (in      module
                                         tardis.tardis_portal.auth.decorators), 160  getAuthMethodDescription()
                                         (tardis.tardis_portal.models.access_control.UserAuthentication
                                         method), 188
get_signing_services() (in      module
                                         tardis.push_to.views), 130  getFacilitiesTestCase (class      in
                                         tardis.tardis_portal.tests.graphql.test_facilities),
                                         241

```

```
getGroupById() (tardis.tardis_portal.auth.interfaces.GroupProvider method), 159
    method), 161
        getUserAuthentications()
getGroupById() (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider portal.models.access_control.UserProfile
    method), 162
        method), 189
getGroupById() (tardis.tardis_portal.tests.test_authservice.MockGroupProvider tardis.tardis_portal.auth.interfaces.UserProvider
    method), 161
        method), 161
getGroups() (in module tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider
    tardis.tardis_portal.auth.fix_circular), 161
        method), 162
getGroups() (tardis.tardis_portal.auth.authservice.AuthService
    method), 158
        getUsernameByEmail()
getGroups() (tardis.tardis_portal.auth.interfaces.GroupProvider method), 159
    method), 161
        getUsernameByEmail()
getGroups() (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider tardis.tardis_portal.auth.interfacesAuthProvider
    method), 162
        method), 161
getGroups() (tardis.tardis_portal.auth.token_auth.TokenGroupProvider
    method), 163
        getEmailByProvider()
getGroups() (tardis.tardis_portal.tests.test_authservice.MockGroupProvider
    method), 161
        global_contexts() (in module
    tardis.tardis_portal.context_processors),
getGroupsForEntity() (tardis.tardis_portal.auth.authservice.AuthService
    method), 158
        278
getGroupsForEntity() (tardis.tardis_portal.auth.interfaces.GroupProvider
    method), 161
        google_analytics() (in module
    tardis.tardis_portal.context_processors),
getGroupsForEntity() (tardis.tardis_portal.tests.test_authservice.MockGroupProvider
    method), 249
        GRAPHQL_SCHEMA (tardis.tardis_portal.tests.graphql.test_auth.userSignIn
    attribute), 240
getParameterSets() (tardis.tardis_portal.models.datafile.DataFile
    method), 192
        GREATER_THAN_COMPARISON
getParameterSets() (tardis.tardis_portal.models.dataset.Dataset
    method), 197
        (tardis.tardis_portal.models.parameters.ParameterName
    attribute), 219
getParameterSets() (tardis.tardis_portal.models.experiment.Experiment
    method), 201
        group (tardis.tardis_portal.models.access_control.GroupAdmin
    attribute), 185
getParameterSets() (tardis.tardis_portal.models.instrument.Instrument
    method), 207
        group_acls_system_owned()
getRecord() (tardis.apps.oaipmh.provider.base.BaseProvider
    method), 99, 298
        group_acls_user_owned()
getRecord() (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider
    method), 101
        required() (in module
    tardis.tardis_portal.auth.decorators), 160
getRecord() (tardis.apps.oaipmh.server.ProxyingServer
    method), 104, 300
        GroupAdmin (class
    in tardis.tardis_portal.models.access_control),
getSupportedAuthMethods() (in module
    tardis.apps.openid_migration.migration),
        185
        GroupAdmin.DoesNotExist, 185
        GroupAdmin.MultipleObjectsReturned, 185
GetTestCase (class
    tardis.apps.related_info.tests.tests), 131
        in GroupProvider (class
    in tardis.tardis_portal.auth.interfaces), 161
getUniqueShortName() (tardis.tardis_portal.models.parameters.ParameterName
    method), 221
        GroupResource (class in tardis.tardis_portal.api),
        273
        GroupResource.Meta (class
    in tardis.tardis_portal.api), 274
getUser() (tardis.tardis_portal.auth.authservice.AuthService
```

```

GroupResourceTest      (class      in has_experiment_ownership()      (in module
                           tardis.tardis_portal.tests.api.test_group_resource),
                           has_experiment_write()      (in module
                           tardis.tardis_portal.auth.decorators), 160
                           has_image(tardis.tardis_portal.models.datafile.DataFile
                           attribute), 192
                           has_perm() (tardis.tardis_portal.auth.authorisation.ACIAwareBackend
                           method), 158
                           has_read_or_owner_ACL()      (in module
                           tardis.tardis_portal.auth.decorators), 160
                           has_write_permissions()      (in module
                           tardis.tardis_portal.auth.decorators), 160
                           hasReader() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry
                           method), 104
                           hasWriter() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry
                           method), 104
                           healthz()      (in module
                           tardis.tardis_portal.views.pages), 261
                           help(tardis.tardis_portal.management.commands.createuser.Command
                           attribute), 180
                           help(tardis.tardis_portal.management.commands.dumpschemas.Command
                           attribute), 180
                           help(tardis.tardis_portal.management.commands.loadschemas.Command
                           attribute), 181
                           help(tardis.tardis_portal.management.commands.rmexperiment.Command
                           attribute), 181
                           hidden(tardis.tardis_portal.models.parameters.Schema
                           attribute), 224
                           host (tardis.apps.push_to.models.Request attribute),
                           host_id (tardis.apps.push_to.models.Request attribute),
                           host_name (tardis.apps.push_to.models.RemoteHost
                           attribute), 127
                           handle(tardis.tardis_portal.management.commands.createuser.Command
                           attribute), 180
                           handle(tardis.tardis_portal.management.commands.dumpschemas.Command
                           attribute), 180
                           handle(tardis.tardis_portal.management.commands.loadschemas.Command
                           attribute), 181
                           handle(tardis.tardis_portal.management.commands.rmexperiment.Command
                           attribute), 181
                           handle_timeout(tardis.tardis_portal.management.commands.loadschemas.Command
                           attribute), 126
                           handle_timeout(tardis.tardis_portal.management.commands.loadschemas.Command
                           attribute), 126
                           HSM_MAX_INODE_FILE_SIZE      (in module
                           tardis.apps.hsm.default_settings), 97
                           HsmAppTestCase      (class      in
                           tardis.apps.hsm.tests.test_api), 94
                           HsmAppConfigTestCase      (class      in
                           tardis.apps.hsm.tests.test_app_config), 95
                           HsmAppEmailTemplateTestCase (class      in
                           tardis.apps.hsm.tests.test_email_templates), 95
                           HsmAppMigrationTestCase     (class      in
                           tardis.apps.hsm.tests.test_migrations), 96
                           HsmConfig (class in tardis.apps.hsm.apps), 97
                           HsmException, 98
                           HsmFileSystemStorage      (class      in
                           tardis.apps.hsm.storage), 98
                           HttpResponseMethodNotAllowed (class      in
                           tardis.tardis_portal.views.utils), 262
                           HttpResponseSeeAlso      (class      in
                           tardis.tardis_portal.views.utils), 262
                           hydrate() (tardis.apps.sftp.api.SFTPPublicKeyAppResource
                           method), 141

```

hydrate() (*tardis.tardis_portal.api.DataFileResource method*), 270
 hydrate() (*tardis.tardis_portal.api.ObjectACLResource method*), 275
 hydrate() (*tardis.tardis_portal.api.ParameterResource method*), 276
 hydrate() (*tardis.tardis_portal.api.ReplicaResource method*), 276
 hydrate_m2m() (*tardis.tardis_portal.api.DatasetResource method*), 272
 hydrate_m2m() (*tardis.tardis_portal.api.ExperimentResource method*), 273
 hydrate_schema() (*tardis.tardis_portal.api.ParameterSetResource attribute*), 213
 id (*tardis.tardis_portal.models.jti.JTI attribute*), 208
 id (*tardis.tardis_portal.models.license.License attribute*), 208
 id (*tardis.tardis_portal.models.parameters.DatafileParameter attribute*), 209
 id (*tardis.tardis_portal.models.parameters.DatafileParameterSet attribute*), 210
 id (*tardis.tardis_portal.models.parameters.DatasetParameter attribute*), 211
 id (*tardis.tardis_portal.models.parameters.DatasetParameterSet attribute*), 213
 id (*tardis.tardis_portal.models.parameters.ExperimentParameter attribute*), 215
 id (*tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute*), 215
 id (*tardis.tardis_portal.models.parameters.FreeTextSearchField attribute*), 215
 id (*tardis.tardis_portal.models.parameters.InstrumentParameter attribute*), 216
 id (*tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute*), 217
 id (*tardis.tardis_portal.models.parameters.ParameterName attribute*), 221
 id (*tardis.tardis_portal.models.parameters.Schema attribute*), 224
 id (*tardis.tardis_portal.models.storage.StorageBox attribute*), 227
 id (*tardis.tardis_portal.models.storage.StorageBoxAttribute attribute*), 229
 id (*tardis.tardis_portal.models.storage.StorageBoxOption attribute*), 230
 id (*tardis.tardis_portal.models.token.Token attribute*), 231
 identify() (*tardis.apps.oaipmh.provider.base.BaseProvider method*), 99, 299
 identify() (*tardis.apps.oaipmh.server.ProxyingServer method*), 104, 300
 image (*tardis.tardis_portal.models.dataset.Dataset attribute*), 199
 image_url (*tardis.tardis_portal.models.license.License attribute*), 208
 immutable (*tardis.tardis_portal.models.dataset.Dataset attribute*), 199
 immutable (*tardis.tardis_portal.models.parameters.ParameterName attribute*), 221
 immutable (*tardis.tardis_portal.models.parameters.Schema attribute*), 224
 ImportParamsForm (*class in tardis.tardis_portal.forms*), 282
 INDEX_VIEWS (*in module tardis.default_settings.custom_views*), 152
 IndexExperimentTestCase (*class in tardis.apps.search.tests.test_index*), 135
 IndexView (*class in tardis.tardis_portal.views.pages*), 135

260
initial (*tardis.apps.eventlog.migrations.0001_initial.Migration*) (tardis.tardis_portal.graphql.parameters.CreateDatafileParameterS attribute), 169
initial (*tardis.apps.openid_migration.migrations.0001_initial.Migration*) (tardis.tardis_portal.graphql.parameters.CreateDatasetParameter attribute), 169
initial (*tardis.apps.sftp.migrations.0001_initial.Migration*) (tardis.tardis_portal.graphql.parameters.CreateDatasetParameterS attribute), 169
inlines (*tardis.tardis_portal.admin.DatafileAdmin* attribute), 264 Input (tardis.tardis_portal.graphql.parameters.CreateExperimentParame attribute), 169
inlines (*tardis.tardis_portal.admin.ExperimentAdmin* attribute), 265 Input (tardis.tardis_portal.graphql.parameters.CreateExperimentParame attribute), 170
inlines (*tardis.tardis_portal.admin.ExperimentParameter* attribute), 265 SetAttribut tardis.tardis_portal.graphql.parameters.CreateParameterName attribute), 170
inlines (*tardis.tardis_portal.admin.InstrumentParameter* attribute), 266 SetAttribut tardis.tardis_portal.graphql.parameters.CreateSchema attribute), 170
inlines (*tardis.tardis_portal.admin.SchemaAdmin* attribute), 266 Input (tardis.tardis_portal.graphql.parameters.UpdateDatafileParameter attribute), 174
inlines (*tardis.tardis_portal.admin.StorageBoxAdmin* attribute), 266 Input (tardis.tardis_portal.graphql.parameters.UpdateDatafileParameter attribute), 174
Input (*tardis.tardis_portal.graphql.datafile.CreateDataFile*) Input (tardis.tardis_portal.graphql.parameters.UpdateDatasetParameter attribute), 163
Input (*tardis.tardis_portal.graphql.datafile.UpdateDataFile*) Input (tardis.tardis_portal.graphql.parameters.UpdateDatasetParameter attribute), 174
Input (*tardis.tardis_portal.graphql.datafile.UploadDataFile*) Input (tardis.tardis_portal.graphql.parameters.UpdateExperimentParame attribute), 174
Input (*tardis.tardis_portal.graphql.datafileobject.CreateDatafileObject*) Input (tardis.tardis_portal.graphql.parameters.UpdateExperimentParame attribute), 174
Input (*tardis.tardis_portal.graphql.datafileobject.UpdateDatafileObject*) Input (tardis.tardis_portal.graphql.parameters.UpdateParameterName attribute), 174
Input (*tardis.tardis_portal.graphql.dataset.CreateDataset*) Input (tardis.tardis_portal.graphql.parameters.UpdateSchema attribute), 174
Input (*tardis.tardis_portal.graphql.dataset.UpdateDataset*) Input (tardis.tardis_portal.graphql.storagebox.CreateStorageBox attribute), 177
Input (*tardis.tardis_portal.graphql.experiment.CreateExperiment*) Input (tardis.tardis_portal.graphql.storagebox.UpdateStorageBox attribute), 177
Input (*tardis.tardis_portal.graphql.experiment.CreateExperimentAt*) Input (tardis.tardis_portal.graphql.storageboxattribute.CreateStorageBox attribute), 177
Input (*tardis.tardis_portal.graphql.experiment.UpdateExperiment*) Input (tardis.tardis_portal.graphql.storageboxattribute.UpdateStorageBo attribute), 178
Input (*tardis.tardis_portal.graphql.experiment.UpdateExperiment*) Input (tardis.tardis_portal.graphql.storageboxoption.CreateStorageBoxO attribute), 178
Input (*tardis.tardis_portal.graphql.facility.CreateFacility*) Input (tardis.tardis_portal.graphql.storageboxoption.UpdateStorageBoxO attribute), 179
Input (*tardis.tardis_portal.graphql.facility.UpdateFacility*) Input (tardis.tardis_portal.graphql.user.ApiSignIn.Arguments attribute), 179
Input (*tardis.tardis_portal.graphql.group.CreateGroup*) Input (tardis.tardis_portal.graphql.user.UserSignIn attribute), 179
Input (*tardis.tardis_portal.graphql.group.UpdateGroup*) institution (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 204
Input (*tardis.tardis_portal.graphql.instrument.CreateInstrument*) institution_name (tardis.tardis_portal.models.experiment.Experiment attribute), 202
Input (*tardis.tardis_portal.graphql.instrument.UpdateInstrument*) instrument (class in tardis.tardis_portal.models.instrument), 206
Input (*tardis.tardis_portal.graphql.parameters.CreateDatafileParameter*) (tardis.tardis_portal.models.dataset.Dataset

attribute), 199
instrument (tardis.tardis_portal.models.parameters.InstrumentParameterSet Case (class in
attribute), 217
INSTRUMENT (tardis.tardis_portal.models.parameters.Schema 244
attribute), 223 InstrumentType (class in
Instrument.DoesNotExist, 206 tardis.tardis_portal.graphql.instrument),
Instrument.MultipleObjectsReturned, 206 168
instrument_id (tardis.tardis_portal.models.dataset.Dataset InstrumentTypeFilter (class in
attribute), 199 tardis.tardis_portal.graphql.instrument),
instrument_id (tardis.tardis_portal.models.parameters.InstrumentParameterSet 178
attribute), 217 InstrumentTypeFilter.Meta (class in
instrument_set (tardis.tardis_portal.models.facility.Facility tardis.tardis_portal.graphql.instrument),
attribute), 205 169
InstrumentAdmin (class in internal_description
tardis.tardis_portal.admin), 265 (tardis.tardis_portal.models.license.License
InstrumentParameter (class in attribute), 209
tardis.tardis_portal.models.parameters), INTERNAL_IPS (in module
216 tardis.default_settings.debug), 152
InstrumentParameter.DoesNotExist, 216 interpolate_template() (in module
InstrumentParameter.MultipleObjectsReturned, tardis.apps.hsm.email_text), 98
216 interpolate_template() (in module
instrumentparameter_set tardis.appsopenid_migration.email_text),
(tardis.tardis_portal.models.parameters.InstrumentParameterSet
attribute), 217 interpolate_template() (in module
instrumentparameter_set tardis.apps.publication_workflow.email_text),
(tardis.tardis_portal.models.parameters.ParameterName 114
attribute), 221 is_active(tardis.tardis_portal.models.license.License
InstrumentParameterInline (class in attribute), 209
tardis.tardis_portal.admin), 265 is_authenticated()
InstrumentParameterSet (class in (tardis.tardis_portal.api.MyTardisAuthentication
tardis.tardis_portal.models.parameters), method), 274
217 is_directory() (in module
InstrumentParameterSet.DoesNotExist, 217 tardis.apps.push_to.utils), 129
InstrumentParameterSet.MultipleObjectsReturned is_expired() (tardis.tardis_portal.models.token.Token
217 method), 231
instrumentparameterset_set is_facility_manager() (in module
(tardis.tardis_portal.models.instrument.Instrument tardis.tardis_portal.models.facility), 206
attribute), 207 is_group_admin() (in module
instrumentparameterset_set tardis.tardis_portal.auth.decorators), 160
(tardis.tardis_portal.models.parameters.Schema is_image() (tardis.tardis_portal.models.datafile.DataFile
attribute), 224 method), 192
InstrumentParameterSetAdmin (class in is_json() (tardis.tardis_portal.models.parameters.ParameterName
tardis.tardis_portal.admin), 265 method), 222
instrumentparametersets is_local() (tardis.tardis_portal.models.datafile.DataFile
(tardis.tardis_portal.models.storage.StorageBox method), 192
attribute), 227 is_online(tardis.tardis_portal.models.datafile.DataFile
InstrumentResource (class in attribute), 192
tardis.tardis_portal.api), 274 is_online(tardis.tardis_portal.models.dataset.Dataset
InstrumentResource.Meta (class in attribute), 199
tardis.tardis_portal.api), 274 is_public() (tardis.tardis_portal.models.datafile.DataFile
InstrumentResourceTest (class in method), 193
tardis.tardis_portal.tests.api.test_instrument_resource publication() (in module
238 tardis.apps.publication_workflow.views),
instruments (tardis.tardis_portal.graphql.schema.tardisQuery 117

LDAPTest (class in `tardis.tardis_portal.tests.test_ldap`), `link_gfk` (`tardis.tardis_portal.models.parameters.InstrumentParameter` attribute), 216
 252

`length_gt()` (in module `link_gfk` (`tardis.tardis_portal.models.parameters.Parameter` attribute)), 218
`tardis.tardis_portal.templatetags.basiccomparisonfilters`, 218
 233

`length_gte()` (in module `link_id` (`tardis.tardis_portal.models.parameters.Parameter` attribute)), 219
`tardis.tardis_portal.templatetags.basiccomparisonfilters`, 219
 233

`length_lt()` (in module `list_allowed_methods` (`tardis.tardis_portal.templatetags.basiccomparisonfilters`), `(tardis.apps.search.api.AdvanceSearchAppResource.Meta` attribute)), 136
 233

`length_lte()` (in module `list_allowed_methods` (`tardis.tardis_portal.templatetags.basiccomparisonfilters`), `(tardis.apps.search.api.SearchAppResource.Meta` attribute)), 136
 233

`LESS_THAN_COMPARISON` `list_allowed_methods` (`tardis.tardis_portal.models.parameters.ParameterName` attribute), 220

`LESS_THAN_EQUAL_COMPARISON` `list_auth_methods()` (in module `tardis.tardis_portal.auth.authentication`), 157

`Level0TestCase` (class in `tardis.tardis_portal.tests.test_iif`), 251

`Level1TestCase` (class in `tardis.tardis_portal.tests.test_iif`), 252

`Level2TestCase` (class in `tardis.tardis_portal.tests.test_iif`), 252

`License` (class in `tardis.tardis_portal.models.license`), `list_display` (`tardis.tardis_portal.admin.ObjectACLAdmin` attribute), 266
 208

`license` (`tardis.tardis_portal.models.experiment.Experiment` attribute), `list_filter` (`tardis.apps.eventlog.admin.LogAdmin` attribute), 90
 202

`License.DoesNotExist`, 208

`License.MultipleObjectsReturned`, 208

`license_id` (`tardis.tardis_portal.models.experiment.Experiment` attribute), `list_subdirectories()` (in module `tardis.push_to.utils`), 129
 203

`LINK` (`tardis.tardis_portal.models.parameters.ParameterName` attribute), 220

`listEndpointsTest` (class in `tardis.tardis_portal.tests.api.test_list_api_endpoints`), 238

`link_ct` (`tardis.tardis_portal.models.parameters.DatafileParameter` attribute), 209

`listIdentifiers()`

`link_ct` (`tardis.tardis_portal.models.parameters.DatasetParameter` attribute), 211

`listIdentifiers()` (`tardis.apps.oaipmh.provider.base.BaseProvider` method), 99, 299

`link_ct` (`tardis.tardis_portal.models.parameters.ExperimentParameter` attribute), 213

`listIdentifiers()` (`tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider` method), 213

`link_ct` (`tardis.tardis_portal.models.parameters.InstrumentParameter` attribute), 216

`listIdentifiers()`

`link_ct` (`tardis.tardis_portal.models.parameters.Parameter` attribute), 218

`(tardis.apps.oaipmh.server.ProxyingServer` method), 104, 300

`link_ct_id` (`tardis.tardis_portal.models.parameters.Parameter` attribute), 218

`listMetadataFormats()` (`tardis.apps.oaipmh.provider.base.BaseProvider` method), 100, 299

`link_gfk` (`tardis.tardis_portal.models.parameters.DatafileParameter` attribute), 209

`listMetadataFormats()`

`link_gfk` (`tardis.tardis_portal.models.parameters.DatasetParameter` attribute), 211

`(tardis.apps.oaipmh.provider.experiment.DcExperimentProvider` method), 101

`link_gfk` (`tardis.tardis_portal.models.parameters.ExperimentParameter` attribute), 214

`listMetadataFormats()` (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider` method), 101

```

        method), 101
listMetadataFormats()                               login()           (in      module
    (tardis.apps.oaipmh.server.ProxyingServer     tardis.tardis_portal.views.authentication),
        method), 105, 300                           257
listRecords() (tardis.apps.oaipmh.provider.base.BaseProvider_VIEWS (in      module
        method), 100, 299                         tardis.default_settings.custom_views), 152
listRecords() (tardis.apps.oaipmh.provider.experimentAbsenceExperimentHandler) (in      module
        method), 101                                tardis.tardis_portal.forms), 283
listRecords() (tardis.apps.oaipmh.server.ProxyingServer logo_img (tardis.apps.push_to.models.RemoteHost at-
        method), 105, 301                           tribute), 126
listSets() (tardis.apps.oaipmh.provider.base.BaseProvider logout()           (in      module
        method), 100, 300                         tardis.tardis_portal.views.authentication),
listSets() (tardis.apps.oaipmh.provider.experiment.AbstractExperiment tardis.tardis_portal.models.parameters.ParameterName
        method), 101                                attribute), 220
listSets() (tardis.apps.oaipmh.server.ProxyingServer lstat() (tardis.apps.sftp.sftp.MyTSFTPServerInterface
        method), 105, 301                           method), 145
ListTestCase (class      in      lt () (in module tardis.tardis_portal.templatetags.basiccomparisonfilters),
    tardis.related_info.tests.tests), 131
load_datafile_image() (in      module      lte () (in module tardis.tardis_portal.templatetags.basiccomparisonfilters
    tardis.tardis_portal.views.images), 258          233
load_dataset_image() (in      module      M
    tardis.tardis_portal.views.images), 258
load_experiment_image() (in      module      make_dirs () (in module tardis.apps.push_to.tasks),
    tardis.tardis_portal.views.images), 258          129
load_image() (in      module      make_mapper ()           (in      module
    tardis.tardis_portal.views.images), 258          tardis.tardis.download), 280
load_licenses() (in      module      make_tar () (tardis.tardis.download.UncachedTarStream
    tardis.apps.publication_workflow.migrations.0001_initial), method), 279
        111                                         manage_auth_methods ()           (in      module
        185                                         tardis.tardis_portal.views.authentication),
load_licenses() (in      module      manage_groups ()           (in      module
    tardis.tardis_portal.migrations.0017_add_cc_licenses), 257
        185                                         tardis.tardis_portal.publish.publishservice.PublishSe
LoadSchemasTestCase (class      in      manage_rifcs () (tardis.tardis_portal.publish.publishservice.PublishSe
    tardis.tardis_portal.tests.management.test_loadschemas), 257
        242                                         method), 232
LocationResource (class      in      manage_user_account ()           (in      module
    tardis.tardis_portal.api), 274
LocationResource.Meta (class      in      tardis.tardis_portal.views.authentication),
    tardis.tardis_portal.api), 274                           257
locked (tardis.tardis_portal.models.experiment.ExperimentManageAccountForm (class      in
        attribute), 203                                tardis.tardis_portal.forms), 283
Log (class in tardis.apps.eventlog.models), 92
log () (in module tardis.apps.eventlog.utils), 93
Log.DoesNotExist, 92
Log.MultipleObjectsReturned, 92
log_search_event ()                               ManageAccountForm.Meta (class      in
    (tardis.apps.search.api.SearchAppResource tardis.tardis_portal.forms), 283
        method), 137
log_set (tardis.apps.eventlog.models.Action attribute), 91
LogAdmin (class in tardis.apps.eventlog.admin), 90
logger (in module tardis.tardis_portal.xmlwriter), 292
LoggingMiddleware (class      in      ManageAccountTestCase (class      in
    tardis.tardis_portal.logging_middleware), map_form_to_schemas ()           (in      module
        91                                         tardis.apps.publication_workflow.views),
        244                                         tardis.tardis_portal.tests.views.test_auth_views),
        244                                         tardis.tardis_portal.forms), 283
manager_group (tardis.tardis_portal.models.facility.Facility
        attribute), 205
manager_group_id (tardis.tardis_portal.models.facility.Facility
        attribute), 205
map_form_to_schemas ()           (in      module
    tardis.apps.publication_workflow.views),
        244

```

117
master_box (*tardis.tardis_portal.models.storage.StorageBox* attribute), 228
master_box_id (*tardis.tardis_portal.models.storage.StorageBox* attribute), 228
MAX_IMAGES_IN_CAROUSEL (in *tardis.default_settings.frontend*), 154
MAX_SEARCH_RESULTS (in *tardis.default_settings.search*), 155
max_size (*tardis.tardis_portal.models.storage.StorageBox* attribute), 228
md5sum (*tardis.tardis_portal.models.datafile.DataFile* attribute), 193
media (*tardis.apps.eventlog.admin.LogAdmin* attribute), media (*tardis.tardis_portal.admin.StorageBoxAdmin* attribute), 266
media (*tardis.apps.openid_migration.models.OpenidACLMigration* attribute), 109
media (*tardis.apps.openid_migration.models.OpenidUserMigration* attribute), 110
media (*tardis.apps.push_to.models.CredentialAdmin* attribute), 121
media (*tardis.apps.push_to.models.CredentialForm* attribute), 122
media (*tardis.apps.push_to.models.RemoteHostAdmin* attribute), 127
media (*tardis.apps.related_info.forms.RelatedInfoForm* attribute), 132
media (*tardis.apps.sftp.forms.KeyAddForm* attribute), 142
media (*tardis.apps.sftp.forms.KeyGenerateForm* attribute), 142
media (*tardis.tardis_portal.admin.DatafileAdmin* attribute), 264
media (*tardis.tardis_portal.admin.DatafileAdminForm* attribute), 264
media (*tardis.tardis_portal.admin.DataFileObjectInline* attribute), 263
media (*tardis.tardis_portal.admin.DataFileObjectInlineForm* attribute), 264
media (*tardis.tardis_portal.admin.DatasetAdmin* attribute), 265
media (*tardis.tardis_portal.admin.ExperimentAdmin* attribute), 265
media (*tardis.tardis_portal.admin.ExperimentParameterInline* attribute), 265
media (*tardis.tardis_portal.admin.ExperimentParameterSetAdmin* attribute), 265
media (*tardis.tardis_portal.admin.FacilityAdmin* attribute), 265
media (*tardis.tardis_portal.admin.FreeTextSearchFieldAdmin* attribute), 265
media (*tardis.tardis_portal.admin.InstrumentAdmin* attribute), 265
media (*tardis.tardis_portal.admin.InstrumentParameterInline* attribute), 265
media (*tardis.tardis_portal.admin.InstrumentParameterSetAdmin* attribute), 266
media (*tardis.tardis_portal.admin.ObjectACLAdmin* attribute), 266
media (*tardis.tardis_portal.admin.ObjectACLIInline* attribute), 266
media (*tardis.tardis_portal.admin.ParameterNameAdmin* attribute), 266
media (*tardis.tardis_portal.admin.ParameterNameInline* attribute), 266
media (*tardis.tardis_portal.admin.SchemaAdmin* attribute), 266
media (*tardis.tardis_portal.admin.StorageBoxAttributeInline* attribute), 267
media (*tardis.tardis_portal.admin.StorageBoxAttributeInlineForm* attribute), 267
media (*tardis.tardis_portal.admin.StorageBoxForm* attribute), 268
media (*tardis.tardis_portal.admin.StorageBoxOptionInline* attribute), 268
media (*tardis.tardis_portal.admin.StorageBoxOptionInlineForm* attribute), 268
media (*tardis.tardis_portal.admin.UserAuthenticationAdmin* attribute), 268
media (*tardis.tardis_portal.forms.AddUserPermissionsForm* attribute), 280
media (*tardis.tardis_portal.forms.CreateUserPermissionsForm* attribute), 281
media (*tardis.tardis_portal.forms.DatasetForm* attribute), 281
media (*tardis.tardis_portal.forms.ExperimentAuthor* attribute), 282
media (*tardis.tardis_portal.forms.ExperimentForm* attribute), 282
media (*tardis.tardis_portal.forms.ImportParamsForm* attribute), 283
media (*tardis.tardis_portal.forms.LoginForm* attribute), 283
media (*tardis.tardis_portal.forms.ManageAccountForm* attribute), 283
media (*tardis.tardis_portal.forms.ManageGroupPermissionsForm* attribute), 284
media (*tardis.tardis_portal.forms.NoInput* attribute), 284
media (*tardis.tardis_portal.forms.RegisterExperimentForm* attribute), 284
media (*tardis.tardis_portal.forms.RegistrationForm* attribute), 285
media (*tardis.tardis_portal.forms.RightsForm* attribute), 285
media (*tardis.tardis_portal.widgets.CommaSeparatedInput*

attribute), 290
 media (*tardis.tardis_portal.widgets.Label* attribute), 291
 media (*tardis.tardis_portal.widgets.Span* attribute), 291
 merge_auth_method() (in module *tardis.tardis_portal.auth.authentication*), 157
 message (*tardis.apps.push_to.models.Progress* attribute), 125
 message (*tardis.apps.push_to.models.Request* attribute), 127
 METADATA_STORE_PATH (in module *tardis.default_settings.storage*), 156
 migrate_accounts() (in module *tardis.apps.openid_migration.views*), 111
 migrate_api_key() (in module *tardis.apps.openid_migration.migration*), 108
 migrate_user_permissions() (in module *tardis.apps.openid_migration.migration*), 108
 Migration (class in *tardis.apps.eventlog.migrations.0001_initial*), 90
 Migration (class in *tardis.apps.hsm.migrations.0001_initial_data*), 94
 Migration (class in *tardis.apps.openid_migration.migrations.0001_initial*), 106
 Migration (class in *tardis.apps.publication_workflow.migrations.0001_initial*), 111
 Migration (class in *tardis.apps.push_to.migrations.0001_initial*), 119
 Migration (class in *tardis.apps.push_to.migrations.0002_auto_20160518_1953*), 119
 Migration (class in *tardis.apps.push_to.migrations.0003_auto_20210216_1322*), 119
 Migration (class in *tardis.apps.sftp.migrations.0001_initial*), 140
 Migration (class in *tardis.tardis_portal.migrations.0001_initial*), 181
 Migration (class in *tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_10643*), 182
 Migration (class in *tardis.tardis_portal.migrations.0002_auto_20150528_1128*), 182
 Migration (class in *tardis.tardis_portal.migrations.0003_auto_20150907_1315*), 182
 Migration (class in *tardis.tardis_portal.migrations.0004_storageboxoption_value_type*), 182
 Migration (class in *tardis.tardis_portal.migrations.0005_datafile_add_size_int_column*), 182
 Migration (class in *tardis.tardis_portal.migrations.0006_datafile_remove_size_string*), 183
 Migration (class in *tardis.tardis_portal.migrations.0007_remove_parameter_string_value*), 183
 Migration (class in *tardis.tardis_portal.migrations.0008_string_value_partial_index*), 183
 Migration (class in *tardis.tardis_portal.migrations.0009_auto_20160128_1119*), 183
 Migration (class in *tardis.tardis_portal.migrations.0010_auto_20160503_1443*), 183
 Migration (class in *tardis.tardis_portal.migrations.0011_auto_20160505_1643*), 184
 Migration (class in *tardis.tardis_portal.migrations.0012_userauthentication_approve*), 184
 Migration (class in *tardis.tardis_portal.migrations.0013_auto_20181002_1136*), 184
 Migration (class in *tardis.tardis_portal.migrations.0014_auto_20181002_1154*), 184
 Migration (class in *tardis.tardis_portal.migrations.0015_dataset_created_time*), 184
 Migration (class in *tardis.tardis_portal.migrations.0016_add_timestamps*), 184
 Migration (class in *tardis.tardis_portal.migrations.0017_add_cc_licenses*), 185
 Migration (class in *tardis.tardis_portal.migrations.0018_make_default_storage_box*), 185
 migration_status (*tardis.apps.openid_migration.models.OpenidUser* attribute), 109
 migration_timestamp
 Mimetypes (*tardis.tardis_portal.models.datafile.DataFile* attribute), 193

MIN_CUTOFF_SCORE (in module `tardis.default_settings.search`), 155

mint () (in module `tardis.apps.publication_workflow.doi.DOI method`), 114

mint_doi_and_deactivate () (in module `tardis.apps.publication_workflow.views`), 117

missing_host_key () (in module `tardis.apps.push_to.models.DBHostKeyPolicy method`), 122

mkdirs () (in module `tardis.tardis_portal.tests.slapd`), 248

MockAuthProvider (class in `tardis.tardis_portal.tests.test_authservice`), 249

MockGroupProvider (class in `tardis.tardis_portal.tests.test_authservice`), 249

MockRequest (class in `tardis.tardis_portal.tests.test_authservice`), 249

MockRifCsProvider (class in `tardis.tardis_portal.tests.test_publishservice`), 254

MockSettings (class in `tardis.tardis_portal.tests.test_authservice`), 249

model (`tardis.apps.push_to.models.CredentialForm.Meta attribute`), 122

model (`tardis.apps.search.documents.DataFileDocument.Django model attribute`), 137

model (`tardis.apps.search.documents.DatasetDocument.Django model attribute`), 138

model (`tardis.apps.search.documents.ExperimentDocument.Django model attribute`), 138

model (`tardis.tardis_portal.admin.DatafileAdminForm.Meta attribute`), 264

model (`tardis.tardis_portal.admin.DataFileObjectInline attribute`), 263

model (`tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta attribute`), 264

model (`tardis.tardis_portal.admin.ExperimentParameterInline attribute`), 265

model (`tardis.tardis_portal.admin.InstrumentParameterInline attribute`), 265

model (`tardis.tardis_portal.admin.ObjectACLInline attribute`), 266

model (`tardis.tardis_portal.admin.ParameterNameInline attribute`), 266

model (`tardis.tardis_portal.admin.StorageBoxAttributeInline attribute`), 267

model (`tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta attribute`), 267

model (`tardis.tardis_portal.admin.StorageBoxForm.Meta attribute`), 267

model (`tardis.tardis_portal.admin.StorageBoxOptionInline attribute`), 268

model (`tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute`), 268

model (`tardis.tardis_portal.forms.DatasetForm.Meta attribute`), 281

model (`tardis.tardis_portal.forms.ExperimentAuthor.Meta attribute`), 281

model (`tardis.tardis_portal.forms.ExperimentForm.Meta attribute`), 282

model (`tardis.tardis_portal.forms.ManageAccountForm.Meta attribute`), 283

model (`tardis.tardis_portal.forms.RightsForm.Meta attribute`), 285

model (`tardis.tardis_portal.graphql.datafile.DataTypeFilter.Meta attribute`), 164

model (`tardis.tardis_portal.graphql.datafileobject.DataTypeObjectTypeFilter attribute`), 165

model (`tardis.tardis_portal.graphql.dataset.DatasetTypeFilter.Meta attribute`), 165

model (`tardis.tardis_portal.graphql.experiment.ExperimentAuthorTypeFilter attribute`), 166

model (`tardis.tardis_portal.graphql.experiment.ExperimentTypeFilter.Meta attribute`), 167

model (`tardis.tardis_portal.graphql.facility.FacilityTypeFilter.Meta attribute`), 167

model (`tardis.tardis_portal.graphql.group.GroupTypeFilter.Meta attribute`), 168

model (`tardis.tardis_portal.graphql.instrument.InstrumentTypeFilter.Meta attribute`), 169

model (`tardis.tardis_portal.graphql.parameters.DatafileParameterSetTypeFilter attribute`), 170

model (`tardis.tardis_portal.graphql.parameters.DatafileParameterTypeFilter attribute`), 171

model (`tardis.tardis_portal.graphql.parameters.DatasetParameterSetTypeFilter attribute`), 171

model (`tardis.tardis_portal.graphql.parameters.DatasetParameterTypeFilter attribute`), 172

model (`tardis.tardis_portal.graphql.parameters.ExperimentParameterSetTypeFilter attribute`), 172

model (`tardis.tardis_portal.graphql.parameters.ExperimentParameterTypeFilter attribute`), 173

model (`tardis.tardis_portal.graphql.parameters.InstrumentParameterSetTypeFilter attribute`), 173

model (`tardis.tardis_portal.graphql.parameters.ObjectACLSetTypeFilter attribute`), 174

model (`tardis.tardis_portal.graphql.parameters.ParameterNameTypeFilter attribute`), 173

model (`tardis.tardis_portal.graphql.storagebox.StorageBoxTypeFilter.Meta attribute`), 177

model (`tardis.tardis_portal.graphql.storageboxattribute.StorageBoxAttributeTypeFilter attribute`), 178

model (`tardis.tardis_portal.graphql.storageboxoption.StorageBoxOptionTypeFilter attribute`), 179

ModelTestCase (class in `tardis.tardis_portal.admin`), 268

```

    tardis.apps.push_to.tests.test_models), 119
ModelTestCase           (class      in MyTSFTPHandle (class in tardis.apps.sftp.sftp), 144
    tardis.apps.openid_migration.tests.test_models),
    106                  (class      in MyTSFTPRequestHandler (class      in
                                         tardis.apps.sftp.sftp), 144
                                         MyTSFTPServer (class in tardis.apps.sftp.sftp), 144
                                         MyTSFTPServerInterface (class      in
                                         tardis.apps.sftp.sftp), 144
                                         MyTSFTPTCPServer (class in tardis.apps.sftp.sftp),
                                         146
ModelTestCase           (class      in MyTSFTPTCPServer (class in tardis.apps.sftp.sftp),
    tardis.tardis_portal.tests.models), 244
ModelTestCase           (class      in MyTSFTPTCPServer (class in tardis.apps.sftp.sftp),
    tardis.tardis_portal.tests.test_storage), 255
modification_time
    (tardis.tardis_portal.models.datafile.DataFile
     attribute), 193
modified_time (tardis.tardis_portal.models.datafile.DataFileObject
     attribute), 195
modified_time (tardis.tardis_portal.models.dataset.Dataset
     attribute), 199
modified_time (tardis.tardis_portal.models.facility.Facility
     attribute), 107
modified_time (tardis.tardis_portal.models.instrument.Instrument
     attribute), 113
move_file () (tardis.tardis_portal.models.datafile.DataFileObject
     method), 195
move_files () (tardis.tardis_portal.models.storage.StorageBox
     method), 228
move_to_master () (tardis.tardis_portal.models.storage.StorageBox
     method), 228
MultiValueCommaSeparatedField (class      in name (tardis.analytics.apps.AnalyticsConfig
    tardis.tardis_portal.fields), 280
mutate () (tardis.tardis_portal.graphql.user.ApiSignIn
     method), 179
Mutation (class in tardis.schema), 294
my_data ()           (in      module name (tardis.apps.sftp.apps.SFTPConfig
    tardis.tardis_portal.views.pages), 261
my_publications ()  (in      module name (tardis.apps.sftp.models.SFTPPublicKey
    tardis.apps.publication_workflow.views),
     117
MyPublicationsTestCase (class      in name (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider
    tardis.apps.publication_workflow.tests.test_my_publications
     attribute), 162
     112
myt_auth ()          (tardis.apps.sftp.sftp.MyTServerInterface
     method), 148
MyTardisAuthentication (class      in name (tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider
    tardis.tardis_portal.api), 274
MyTardisAuthenticationTest (class      in name (tardis.tardis_portal.auth.token_auth.TokenGroupProvider
    tardis.tardis_portal.tests.api.test_auth), 235
     attribute), 163
MyTardisLocalStorageSystemStorage (class      in name (tardis.tardis_portal.models.facility.Facility
    tardis.tardis_portal.storage.file_system), 232
     attribute), 205
MyTardisModelErrorResource (class      in name (tardis.tardis_portal.models.instrument.Instrument
    tardis.tardis_portal.api), 274
     attribute), 207
MyTardisModelErrorResource.Meta (class      in name (tardis.tardis_portal.models.license.License
    tardis.tardis_portal.api), 275
     attribute), 209
MyTardisResourceTestCase (class      in name (tardis.tardis_portal.models.parameters.DatafileParameter
    tardis.tardis_portal.tests.api), 239
     attribute), 209
MyTServerInterface (class in tardis.apps.sftp.sftp),
     146                  (class      in name (tardis.tardis_portal.models.parameters.DatasetParameter
                                         attribute), 212
                                         name (tardis.tardis_portal.models.parameters.ExperimentParameter
                                         attribute), 214
                                         N
                                         name (tardis.analytics.apps.AnalyticsConfig
                                         attribute),
                                         name (tardis.apps.eventlog.models.Action
                                         attribute), 92
                                         name (tardis.apps.hsm.apps.HsmConfig
                                         attribute), 97
                                         name (tardis.apps.openid_migration.apps.OpenidMigrationConfig
                                         attribute),
                                         name (tardis.apps.publication_workflow.apps.PublicationWorkflowConfig
                                         attribute),
                                         name (tardis.apps.push_to.apps.PushToConfig
                                         attribute),
                                         name (tardis.apps.s3utils.apps.S3UtilsConfig
                                         attribute),
                                         name (tardis.apps.search.apps.SearchConfig
                                         attribute),
                                         name (tardis.apps.search.documents.DataFileDocument.Index
                                         attribute), 138
                                         name (tardis.apps.search.documents.DatasetDocument.Index
                                         attribute), 138
                                         name (tardis.apps.search.documents.ExperimentDocument.Index
                                         attribute), 138
                                         name (tardis.apps.sftp.apps.SFTPConfig
                                         attribute), 142
                                         name (tardis.apps.sftp.models.SFTPPublicKey
                                         attribute),
                                         143
                                         name (tardis.apps.social_auth.apps.SocialAuthConfig
                                         attribute), 150
                                         name (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider
                                         attribute), 162
                                         name (tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider
                                         attribute), 162
                                         name (tardis.tardis_portal.auth.token_auth.TokenGroupProvider
                                         attribute), 163
                                         name (tardis.tardis_portal.models.facility.Facility
                                         attribute), 205
                                         name (tardis.tardis_portal.models.instrument.Instrument
                                         attribute), 207
                                         name (tardis.tardis_portal.models.license.License
                                         attribute), 209
                                         name (tardis.tardis_portal.models.parameters.DatafileParameter
                                         attribute), 209
                                         name (tardis.tardis_portal.models.parameters.DatasetParameter
                                         attribute), 212
                                         name (tardis.tardis_portal.models.parameters.ExperimentParameter
                                         attribute), 214

```

name (*tardis.tardis_portal.models.parameters.InstrumentParameter*.callback () (in module *tardis.apps.push_to.views*), 130
 attribute), 216
 name (*tardis.tardis_portal.models.parameters.Parameter* oauth_callback_url () (in module *tardis.apps.push_to.views*), 130
 attribute), 219
 name (*tardis.tardis_portal.models.parameters.ParameterName* oauth_check_token_url
 attribute), 222
 name (*tardis.tardis_portal.models.parameters.Schema* oauth_client_id (*tardis.apps.push_to.models OAuthSSHCertSigningService*
 attribute), 124
 attribute), 224
 name (*tardis.tardis_portal.models.storage.StorageBox* oauth_client_secret
 attribute), 228
 name_id (*tardis.tardis_portal.models.parameters.Parameter* oauth_token_url (*tardis.apps.push_to.models OAuthSSHCertSigningService*
 attribute), 219
 attribute), 124
 namespace (*tardis.tardis_portal.models.parameters.Schema* oauth_token_url (*tardis.apps.push_to.models OAuthSSHCertSigningService*
 attribute), 224
 attribute), 124
 natural_key () (*tardis.tardis_portal.models.parameters.PartitionNone* SigningService (class in
 method), 222
tardis.apps.push_to.models), 123
 natural_key () (*tardis.tardis_portal.models.parameters.Schema* SHCertSigningService.DoesNotExist,
 method), 225
 123
 new_param () (*tardis.tardis_portal.ParameterSetManager* PartitionNone SigningService.MultipleObjectsReturned,
 method), 263
 123
 new_user (*tardis.apps.openid_migration.models.OpenidUserMigration*ertsigningservice_set
 attribute), 109
 attribute), 126
 new_user_auth_method
 (*tardis.apps.openid_migration.models.OpenidUserMigrat*
 attribute), 109
 (*tardis.apps.eventlog.models.Log* attribute), 92
 attribute), 109
 obj_create () (*tardis.apps.search.api.AdvanceSearchAppResource*
 method), 270
 new_user_id (*tardis.apps.openid_migration.models.OpenidUserMigrat*hod), 136
 attribute), 110
 obj_create () (*tardis.tardis_portal.api.DataFileResource*
 method), 270
 next () (*tardis.analytics.tracker.IteratorTracker* object_create () (*tardis.tardis_portal.api.ExperimentResource*
 method), 88
 method), 270
 nickname (*tardis.apps.push_to.models.OAuthSSHCertSigningService*nickname (attribute), 124
 attribute), 124
 nickname (*tardis.apps.push_to.models.RemoteHost* attribute), 126
 obj_get_list () (*tardis.apps.search.api.AdvanceSearchAppResource*
 method), 136
 obj_get_list () (*tardis.apps.search.api.SearchAppResource*
 method), 137
 NoInput (class in *tardis.tardis_portal.forms*), 284
 object_class (*tardis.apps.search.api.AdvanceSearchAppResource*.Meta
 attribute), 136
 NONE (*tardis.tardis_portal.models.parameters.Schema* attribute), 223
 object_class (*tardis.apps.search.api.SearchAppResource*.Meta
 attribute), 136
 NoSuitableCredential, 120
 object_class (*tardis.apps.search.api.SearchAppResource*.Meta
 attribute), 136
 NOT_EQUAL_COMPARISON
 (*tardis.tardis_portal.models.parameters.Parameter*.*Object*_class (*tardis.apps.sftp.api.SFTPPublicKeyAppResource*.Meta
 attribute), 220
 attribute), 141
 now () (*tardis.tardis_portal.tests.test_tokens.FrozenTime* object_class (*tardis.tardis_portal.api.DatafileParameterResource*.Meta
 class method), 255
 attribute), 270
 NS_CC (*tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider* (*tardis.tardis_portal.api.DatafileParameterSetResource*.*Object*
 attribute), 101
 attribute), 270
 NUMERIC (*tardis.tardis_portal.models.parameters.Parameter*.*Object*_class (*tardis.tardis_portal.api.DataFileResource*.Meta
 attribute), 220
 attribute), 269
 numerical_value (*tardis.tardis_portal.models.parameters.Parameter*.*Object*_class (*tardis.tardis_portal.api.DatasetParameterResource*.Meta
 attribute), 219
 attribute), 271
 object_class (*tardis.tardis_portal.api.DatasetParameterSetResource*.Meta
 attribute), 271
 object_class (*tardis.tardis_portal.api.DatasetResource*.Meta
 attribute), 271
 oauth_authorize_url
 (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 271
 attribute), 124
 object_class (*tardis.tardis_portal.api.ExperimentAuthorResource*.Meta
 attribute), 272

O

oauth_authorize_url
 (*tardis.apps.push_to.models.OAuthSSHCertSigningService* attribute), 271
 attribute), 124
 object_class (*tardis.tardis_portal.api.ExperimentAuthorResource*.Meta
 attribute), 272

object_class (tardis.tardis_portal.api.ExperimentParameterResourceMeta)	203
attribute), 272	ObjectACLTestCase (class in
object_class (tardis.tardis_portal.api.ExperimentParameterSetResourceMeta)	248
attribute), 272	tardis.tardis_portal.tests.test_authorisation),
object_class (tardis.tardis_portal.api.ExperimentResourceMeta)	92
attribute), 273	(tardis.apps.eventlog.models.Action attribute),
object_class (tardis.tardis_portal.api.FacilityResourceMeta)	93
attribute), 273	(tardis.apps.eventlog.models.Log attribute),
object_class (tardis.tardis_portal.api.GroupResourceMeta)	108
attribute), 274	(tardis.apps.openid_migration.models.OpenidACLMigration attribute),
object_class (tardis.tardis_portal.api.InstrumentResourceMeta)	110
attribute), 274	(tardis.apps.openid_migration.models.OpenidUserMigration attribute),
object_class (tardis.tardis_portal.api.MyTardisModelResourceMeta)	120
attribute), 275	(tardis.apps.push_to.models.Credential attribute),
object_class (tardis.tardis_portal.api.ObjectACLResourceMeta)	124
attribute), 275	(tardis.apps.push_to.models.OAuthSSHCertiSigningService attribute),
object_class (tardis.tardis_portal.api.ParameterNameResourceMeta)	125
attribute), 275	(tardis.apps.push_to.models.Progress attribute),
object_class (tardis.tardis_portal.api.ReplicaResourceMeta)	126
attribute), 276	(tardis.apps.push_to.models.RemoteHost attribute),
object_class (tardis.tardis_portal.api.SchemaResourceMeta)	128
attribute), 277	(tardis.apps.push_to.models.Request attribute),
object_class (tardis.tardis_portal.api.StorageBoxAttributeResourceMeta)	143
attribute), 277	(tardis.tardis_portal.models.sftp.models.SFTPPublicKey attribute),
object_class (tardis.tardis_portal.api.StorageBoxOptionResourceMeta)	186
attribute), 277	(tardis.tardis_portal.models.access_control.GroupAdmin attribute),
object_class (tardis.tardis_portal.api.StorageBoxResourceMeta)	188
attribute), 277	(tardis.tardis_portal.models.access_control.ObjectACL attribute),
object_class (tardis.tardis_portal.api.UserResourceMeta)	189
attribute), 278	(tardis.tardis_portal.models.access_control.UserAuthentication attribute),
object_id (tardis.apps.eventlog.models.Log attribute), 92	objects (tardis.tardis_portal.models.access_control.UserProfile attribute), 189
object_id (tardis.apps.push_to.models.Request attribute), 127	objects (tardis.tardis_portal.models.datafile.DataFile attribute), 193
object_id (tardis.tardis_portal.models.access_control.ObjectACLs attribute), 188	objects (tardis.tardis_portal.models.datafile.DataFileObject attribute), 195
object_type (tardis.apps.push_to.models.Request attribute), 127	objects (tardis.tardis_portal.models.dataset.Dataset attribute), 199
ObjectACL (class tardis.tardis_portal.models.access_control), 186	objects (tardis.tardis_portal.models.experiment.Experiment attribute), 203
ObjectACL.DoesNotExist, 186	objects (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 204
ObjectACL.MultipleObjectsReturned, 186	objects (tardis.tardis_portal.models.facility.Facility attribute), 206
ObjectACLAdmin (class tardis.tardis_portal.admin), 266	objects (tardis.tardis_portal.models.instrument.Instrument attribute), 207
ObjectACLInline (class tardis.tardis_portal.admin), 266	objects (tardis.tardis_portal.models.jti.JTI attribute), 208
ObjectACLResource (class tardis.tardis_portal.api), 275	objects (tardis.tardis_portal.models.license.License attribute), 209
ObjectACLResource.Meta (class tardis.tardis_portal.api), 275	objects (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 211
objectacl (tardis.apps.publication_workflow.models.Publication attribute), 116	objects (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 213
objectacls (tardis.tardis_portal.models.experiment.Experiment attribute), 213	

objects (*tardis.tardis_portal.models.parameters.ExperimentParameter*, 215) *ExperimentParameterIntegration_set*
attribute), 215 (*tardis.tardis_portal.models.access_control.ObjectACL*)
objects (*tardis.tardis_portal.models.parameters.FreeTextSearchField*, 188)
attribute), 216
objects (*tardis.tardis_portal.models.parameters.InstrumentParameter*, 217) *InstrumentParameterSet*
attribute), 217 (*tardis.apps.openid_migration.models*), 109
objects (*tardis.tardis_portal.models.parameters.Parameter*, 219) *Parameter*
attribute), 219 (*tardis.apps.openid_migration.apps*), 107
objects (*tardis.tardis_portal.models.parameters.ParameterName*, 222) *ParameterName*
attribute), 222 (*tardis.apps.openid_migration.tests.test_forms*), 106
objects (*tardis.tardis_portal.models.parameters.Schema*, 225) *Schema*
attribute), 225 (*OpenIDMigrationTestCase*, 109) *OpenIDMigrationTestCase*
objects (*tardis.tardis_portal.models.storage.StorageBox*, 228) *StorageBox*
attribute), 228 (*OpenIDMigrationViewTestCase*, 106) *OpenIDMigrationViewTestCase*
objects (*tardis.tardis_portal.models.storage.StorageBoxAttribute*, 229) *StorageBoxAttribute*
attribute), 229 (*tardis.apps.openid_migration.tests.test_views*), 107
objects (*tardis.tardis_portal.models.storage.StorageBox*, 230) *StorageBox*
attribute), 230 (*OpenidUserMigration*, 109) *OpenidUserMigration*
objects (*tardis.tardis_portal.models.token.Token*, 231) *Token*
attribute), 231 (*OpenidUserMigration.DoesNotExist*, 109) *OpenidUserMigration.DoesNotExist*
OpenidUserMigration.MultipleObjectsReturned,
offline_types (*tardis.tardis_portal.models.storage.StorageBox*, 228) *StorageBox*
attribute), 228 (*OpenidUserMigrationAdmin*, 109) *OpenidUserMigrationAdmin*
old_user (*tardis.apps.openid_migration.models.OpenidUserMigration*, 110) *OpenidUserMigration*
attribute), 110 (*tardis.apps.eventlog.migrations.0001_initial.Migration*
operations (*tardis.apps.hsm.migrations.0001_initial_data*, 90) *attribute*, 90
old_user_auth_method (*tardis.apps.openid_migration.models.OpenidUserMigration*, 110) *OpenidUserMigration*
attribute), 110 (*tardis.apps.hsm.migrations.0001_initial_data*, 94) *attribute*, 94
old_user_id (*tardis.apps.openid_migration.models.OpenidUserMigration*, 110) *OpenidUserMigration*
attribute), 110 (*tardis.apps.openid_migration.migrations.0001_initial.Migration*
operations (*tardis.apps.hsm.api.DatasetAppResource*, 96) *attribute*, 106
online_count () (*tardis.apps.hsm.api.DatasetAppResource*, 96) *attribute*, 111
operations (*tardis.apps.push_to.migrations.0001_initial.Migration*
attribute), 111
online_files_count (*tardis.tardis_portal.models.dataset.Dataset*, 199) *attribute*, 119
operations (*tardis.apps.push_to.migrations.0002_auto_20160518_1953*, 119) *attribute*, 119
online_types (*tardis.tardis_portal.models.storage.StorageBox*, 228) *StorageBox*
attribute), 228 (*tardis.apps.publication_workflow.migrations.0001_initial.Migration*
operations (*tardis.apps.push_to.migrations.0003_auto_20210216_1322*, 119) *attribute*, 119
open () (*tardis.apps.sftp.sftp.MyTSFTPServerInterface*, 145) *method*, 145
openid_migration_method () (in module *tardis.apps.openid_migration.migration*), 108
openid_migration_processor () (in module *tardis.apps.openid_migration.context_processors*), 107
openid_user_migration_form () (in module *tardis.apps.openid_migration.forms*), 107
OpenidUserMigration (class in *tardis.apps.openid_migration.models*), 108
OpenidUserMigration.DoesNotExist, 108
OpenidUserMigration.MultipleObjectsReturned
operations (*tardis.tardis_portal.migrations.0001_squashed_0011_auto_20150528_1128*, 182) *attribute*, 182
operations (*tardis.tardis_portal.migrations.0002_auto_20150528_1128*, 182) *attribute*, 182
operations (*tardis.tardis_portal.migrations.0003_auto_20150907_1315*, 182) *attribute*, 182
operations (*tardis.tardis_portal.migrations.0004_storageboxoption_value*, 182) *attribute*, 182
operations (*tardis.tardis_portal.migrations.0005_datafile_add_size_int*, 182) *attribute*, 182
openidaclmigration_set (*tardis.apps.openid_migration.models.OpenidUserMigration*, 110) *attribute*, 110
operations (*tardis.tardis_portal.migrations.0006_datafile_remove_size*, 183) *attribute*, 183
operations (*tardis.tardis_portal.migrations.0007_remove_parameter*, 108) *attribute*, 108

attribute), 183
operations (tardis.tardis_portal.migrations.0008_string~~order~~*partial_id*~~partial_id~~*tardis_portal.models.parameters.Parameter.Meta*
attribute), 183
operations (tardis.tardis_portal.migrations.0009_auto_20160128_1109~~Migration~~*tardis_portal.models.parameters.ParameterSet.Meta*
attribute), 183
operations (tardis.tardis_portal.migrations.0010_auto_20160503_1442~~Migration~~*tardis_portal.managers.ExperimentManager*
attribute), 183
operations (tardis.tardis_portal.migrations.0011_auto_20160505_1643~~Migration~~*(tardis.tardis_portal.managers.ExperimentManager*
attribute), 184
operations (tardis.tardis_portal.migrations.0012_userauthentication~~method~~*, 187**.Migration*
attribute), 184
owned_by_group () (tardis.tardis_portal.managers.ExperimentManager
attribute), 184
operations (tardis.tardis_portal.migrations.0013_auto_20181002_1146~~Migration~~*owned_by_user () (tardis.tardis_portal.managers.ExperimentManager*
attribute), 184
operations (tardis.tardis_portal.migrations.0014_auto_20181002_1147~~Migration~~*owned_by_user_id ()*
attribute), 184
operations (tardis.tardis_portal.migrations.0015_dataset_created~~Migration~~*tardis.tardis_portal.managers.ExperimentManager*
attribute), 184
method), 287
operations (tardis.tardis_portal.migrations.0016_add_to_wiki_page~~Migration~~*tardis.tardis_portal.models.access_control.ObjectACL*
attribute), 185
attribute), 185
P
operations (tardis.tardis_portal.migrations.0018_make_default_storage_box_status_online~~Migration~~*module*
attribute), 185
paginator_number () (in
*options (tardis.tardis_portal.models.storage.StorageBox**tardis.tardis_portal.templates.pagination),*
attribute), 228
235
OracleSafeManager
(class
in
tardis.tardis_portal.managers), 288
235
order (tardis.tardis_portal.models.experiment.ExperimentAuthor~~parallel_bulk~~*(tardis.apps.search.documents.DataFileDocument*
attribute), 204
method), 138
order (tardis.tardis_portal.models.parameters.ParameterName~~parallel_bulk~~*(tardis.apps.search.documents.DatasetDocument*
attribute), 222
method), 138
ordering (tardis.tardis_portal.api.DataFileResource.Meta~~parallel_bulk~~*(tardis.apps.search.documents.ExperimentDocument*
attribute), 269
method), 138
ordering (tardis.tardis_portal.api.DatasetResource.Meta~~Parameter~~
attribute), 271
(class
tardis.tardis_portal.models.parameters),
ordering (tardis.tardis_portal.api.ExperimentAuthorResource.Meta~~Parameter~~
attribute), 272
Parameter.Meta
(class
in
ordering (tardis.tardis_portal.api.ExperimentResource.Meta~~Parameter~~
attribute), 273
Parameter.Meta
(class
tardis.tardis_portal.models.parameters),
218
ordering (tardis.tardis_portal.api.FacilityResource.Meta~~parameter_class~~
attribute), 273
(tardis.tardis_portal.models.parameters.DatafilePar
attribute), 211
ordering (tardis.tardis_portal.api.InstrumentResource.Meta~~parameter_class~~
attribute), 274
(tardis.tardis_portal.models.parameters.DatasetPar
attribute), 213
ordering (tardis.tardis_portal.api.ObjectACLResource.Meta~~parameter_class~~
attribute), 275
(tardis.tardis_portal.models.parameters.Experiment
attribute), 215
ordering (tardis.tardis_portal.api.ReplicaResource.Meta~~parameter_class~~
attribute), 276
(tardis.tardis_portal.models.parameters.InstrumentP
attribute), 218
ordering (tardis.tardis_portal.api.SchemaResource.Meta~~parameter_class~~
attribute), 277
(tardis.tardis_portal.models.parameters.ParameterS
attribute), 222
ordering (tardis.tardis_portal.api.StorageBoxAttributeResource.Meta~~parameter_name~~
attribute), 277
(tardis.tardis_portal.models.parameters.FreeTextSearc
attribute), 216
ordering (tardis.tardis_portal.api.StorageBoxOptionResource.Meta~~parameter_name_id~~
attribute), 277
(tardis.tardis_portal.models.parameters.FreeTextSearchField
attribute), 216

parameter_type (*tardis.tardis_portal.models.parameters.DatafileParameter*,
 attribute), 214
 parameterset (*tardis.tardis_portal.models.parameters.InstrumentParameter*,
 attribute), 217
 parameterset (*tardis.tardis_portal.ParameterSetManager.ParameterSet*,
 attribute), 218
 ParameterSet.Meta (class) in
 parameterset (*tardis.tardis_portal.models.parameters.InstrumentParameter*,
 attribute), 222
 parameterset (*tardis.tardis_portal.models.parameters.ExperimentParameter*,
 attribute), 223
 ParameterSet.id (*tardis.tardis_portal.models.parameters.DatafileParameter*,
 attribute), 210
 ParameterSet.id (*tardis.tardis_portal.models.parameters.DatasetParameter*,
 attribute), 212
 parameterset (*tardis.tardis_portal.models.parameters.ExperimentParameter*,
 attribute), 214
 ParameterSetName (class) in
 ParameterSetName (*tardis.tardis_portal.models.parameters*),
 219
 ParameterName.DoesNotExist, 219
 ParameterName.MultipleObjectsReturned,
 220
 parametername_form() (in module
 tardis.tardis_portal.templatetags.formfieldfilters),
 234
 parametername_set
 (*tardis.tardis_portal.models.parameters.Schema*,
 attribute), 225
 ParameterNameAdmin (class)
 (*tardis.tardis_portal.admin*), 266
 ParameterNameInline (class)
 (*tardis.tardis_portal.admin*), 266
 ParameterNameManager (class)
 (*tardis.tardis_portal.managers*), 288
 ParameterNameResource (class)
 (*tardis.tardis_portal.api*), 275
 ParameterNameResource.Meta (class) in ParameterSetManager (class) in
 (*tardis.tardis_portal.api*), 275
 ParameterNames (*tardis.tardis_portal.graphql.schema.tardisQuery*,
 attribute), 244
 ParameterNameType (class) in ParameterSetManagerTestCase (class) in
 (*tardis.tardis_portal.graphql.parameters*),
 173
 ParameterNameTypeFilter (class) in PATH_LDAPADD (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 ParameterNameTypeFilter.Meta (class) in PATH_LDAPSEARCH (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 ParameterResource (class) in PATH_SCHEMA_DIR (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 parameters (*tardis.tardis_portal.ParameterSetManager.ParameterSetManager*,
 attribute), 247
 ParameterSet (class) in PATH_SLAPTEST (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 ParameterSet (class) in PATH_TMPDIR (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 parameterset (*tardis.tardis_portal.models.parameters.DatafileParameter*,
 attribute), 210
 parameterset (*tardis.tardis_portal.models.parameters.DatasetParameter*,
 attribute), 212
 parameterset (*tardis.tardis_portal.models.parameters.ExperimentParameter*,
 attribute), 214
 ParameterSetName (class) in
 ParameterSetName (*tardis.tardis_portal.models.parameters*),
 223
 ParameterSetManager (class) in ParameterSetManagerTestCase (class) in
 (*tardis.tardis_portal.tests.test_parametersets*),
 253
 ParameterSetManagerMixin (class) in
 (*tardis.tardis_portal.models.parameters*),
 223
 ParameterSetManagerTestCase (class) in
 (*tardis.tardis_portal.tests.test_parameter*),
 244
 password (*tardis.apps.push_to.models.Credential*,
 attribute), 121
 PATH_LDAPADD (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 PATH_LDAPSEARCH (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 PATH_SCHEMA_DIR (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 PATH_SLAPTEST (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 PATH_TMPDIR (*tardis.tardis_portal.tests.slapd.Slapd*,
 attribute), 247
 PICKLE (*tardis.tardis_portal.models.storage.StorageBoxOption*,
 attribute), 229
 parameterset (*tardis.tardis_portal.models.parameters.DatafileParameter*,
 attribute), 163
 parameterset (*tardis.tardis_portal.models.parameters.DatasetParameter*,
 attribute), 164
 parameterset (*tardis.tardis_portal.models.parameters.ExperimentParameter*,
 attribute), 165

```

    tribute), 165
pk (tardis.tardis_portal.graphql.experiment.ExperimentAuthorType attribute), 166
pk (tardis.tardis_portal.graphql.experiment.ExperimentType attribute), 166
pk (tardis.tardis_portal.graphql.facility.FacilityType attribute), 167
pk (tardis.tardis_portal.graphql.group.GroupType attribute), 168
pk (tardis.tardis_portal.graphql.instrument.InstrumentType attribute), 168
pk (tardis.tardis_portal.graphql.parameters.DatafileParameterSetType attribute), 170
pk (tardis.tardis_portal.graphql.parameters.DatasetParameterType attribute), 170
pk (tardis.tardis_portal.graphql.parameters.DatasetParameterSetType attribute), 171
pk (tardis.tardis_portal.graphql.parameters.DatasetParameterType attribute), 171
pk (tardis.tardis_portal.graphql.parameters.ExperimentParameterSetType attribute), 172
pk (tardis.tardis_portal.graphql.parameters.ExperimentParameterType exception () attribute), 172
pk (tardis.tardis_portal.graphql.parameters.ParameterNameType attribute), 173
pk (tardis.tardis_portal.graphql.parameters.TardisSchemaType attribute), 173
pk (tardis.tardis_portal.graphql.storagebox.StorageBoxType attribute), 177
pk (tardis.tardis_portal.graphql.storageboxattribute.StorageBoxAttributeType attribute), 178
pk (tardis.tardis_portal.graphql.storageboxoption.StorageBoxOptionType attribute), 178
pluginId (tardis.tardis_portal.models.access_control.ObjectAccess attribute), 188
port (tardis.apps.push_to.models.RemoteHost attribute), 126
post_list () (tardis.tardis_portal.api.DataFileResource method), 270
post_save_experiment () (in module tardis.tardis_portal.models.hooks), 206
post_save_experiment_parameter () (in module tardis.tardis_portal.models.hooks), 206
post_save_experimentauthor () (in module tardis.tardis_portal.models.hooks), 206
POST_SAVE_FILTERS (tardis.default_settings attribute), 59
prepare_experiments () (tardis.apps.search.documents.DataFileDocument public () method), 138
prepare_output () (tardis.tardis_portal.download.UncachedTarStream attribute), 279
prepend_urls () (tardis.apps.hsm.api.DatasetAppResource method), 96
prepend_urls () (tardis.apps.hsm.api.ReplicaAppResource method), 97
prepend_urls () (tardis.apps.s3utils.api.ReplicaAppResource method), 134
prepend_urls () (tardis.tardis_portal.api.DataFileResource method), 270
prepend_urls () (tardis.tardis_portal.api.DatasetResource method), 272
PrettyJSONSerializer (class in tardis.apps.search.api), 136
PrettyJSONSerializer (class in tardis.tardis_portal.api), 276
priority (tardis.tardis_portal.models.datafile.DataFileObject priority), 195
priority (tardis.tardis_portal.models.storage.StorageBox priority), 228
private_key (tardis.apps.push_to.models.KeyPair attribute), 122
process_embargos () (in module tardis.tardis_portal.models.publication_workflow.tasks), 117
process_form () (in module tardis.tardis_portal.logging_middleware.LoggingMiddleware method), 286
process_request () (tardis.tardis_portal.auth.token_auth.TokenAuthMiddleware attribute), 163
process_response () (tardis.tardis_portal.logging_middleware.LoggingMiddleware method), 286
Progress.DoesNotExist (class in tardis.apps.push_to.models), 124
Progress.DoesNotExist, 124
Progress.MultipleObjectsReturned, 124
progress_set (tardis.apps.push_to.models.Request attribute), 128
progress_set (tardis.tardis_portal.models.datafile.DataFile attribute), 193
PROXY_DOWNLOAD_PREFIXES (in module tardis.default_settings.downloads), 153
PROXY_DOWNLOADS (in module tardis.default_settings.downloads), 153
ProxyingMetadataRegistry (class in tardis.apps.ooipmh.server), 104
ProxyingServer (class in tardis.apps.ooipmh.server), 104, 300
PUBLIC_ACCESS_CHOICES (tardis.tardis_portal.models.experiment.Experiment attribute), 203

```

attribute), 200

PUBLIC_ACCESS_EMBARGO
(tardis.tardis_portal.models.experiment.Experiment attribute), 200

PUBLIC_ACCESS_FULL
(tardis.tardis_portal.models.experiment.Experiment attribute), 200

public_access_implies_distribution()
(tardis.tardis_portal.models.experiment.Experiment class method), 203

PUBLIC_ACCESS_METADATA
(tardis.tardis_portal.models.experiment.Experiment attribute), 200

PUBLIC_ACCESS_NONE
(tardis.tardis_portal.models.experiment.Experiment attribute), 200

public_data() *(in module tardis.tardis_portal.views.pages), 261*

public_download_allowed()
(tardis.tardis_portal.models.experiment.Experiment method), 203

public_key *(tardis.apps.push_to.models.KeyPair attribute), 122*

public_key *(tardis.apps.sftp.models.SFTPPublicKey attribute), 143*

Publication *(class in tardis.apps.publication_workflow.models), 115*

Publication.DoesNotExist, 115

Publication.MultipleObjectsReturned, 115

Publication.PublicationManager *(class in tardis.apps.publication_workflow.models), 115*

PUBLICATION_DETAILS_SCHEMA
(tardis.tardis_portal.models.experiment.Experiment attribute), 200

PUBLICATION_DRAFT_SCHEMA
(tardis.tardis_portal.models.experiment.Experiment attribute), 200

PUBLICATION_RETRACTED_SCHEMA
(tardis.tardis_portal.models.experiment.Experiment attribute), 200

PUBLICATION_SCHEMA_ROOT
(tardis.tardis_portal.models.experiment.Experiment attribute), 200

PublicationFormTestCase *(class in tardis.apps.publication_workflow.tests.test_form), 112*

PublicationManager *(class in tardis.apps.publication_workflow.managers), 114*

PublicationTokensTestCase *(class in tardis.apps.publication_workflow.tests.test_tokens), 113*

PublicationWorkflowConfig *(class in tardis.apps.publication_workflow.apps), 113*

publish_public_expt_rifcs() *(in module tardis.tardis_portal.models.hooks), 206*

PublishService *(class in tardis.tardis_portal.publish.publishservice), 232*

PublishServiceTestCase *(class in tardis.tardis_portal.tests.test_publishservice), 254*

PushToConfig *(class in tardis.apps.push_to.apps), 120*

put_detail() *(tardis.tardis_portal.api.DataFileResource method), 270*

Q

Query *(class in tardis.schema), 294*

queryset *(tardis.apps.hsm.api.DatasetAppResource.Meta attribute), 96*

queryset *(tardis.apps.hsm.api.ReplicaAppResource.Meta attribute), 96*

queryset *(tardis.apps.s3utils.api.ReplicaAppResource.Meta attribute), 133*

queryset *(tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta attribute), 141*

queryset *(tardis.tardis_portal.api.DatafileParameterResource.Meta attribute), 270*

queryset *(tardis.tardis_portal.api.DatafileParameterSetResource.Meta attribute), 271*

queryset *(tardis.tardis_portal.api.DataFileResource.Meta attribute), 269*

queryset *(tardis.tardis_portal.api.DatasetParameterResource.Meta attribute), 271*

queryset *(tardis.tardis_portal.api.DatasetParameterSetResource.Meta attribute), 271*

queryset *(tardis.tardis_portal.api.DatasetResource.Meta attribute), 271*

queryset *(tardis.tardis_portal.api.ExperimentAuthorResource.Meta attribute), 272*

queryset *(tardis.tardis_portal.api.ExperimentParameterResource.Meta attribute), 272*

queryset *(tardis.tardis_portal.api.ExperimentParameterSetResource.Meta attribute), 273*

queryset *(tardis.tardis_portal.api.ExperimentResource.Meta attribute), 273*

queryset *(tardis.tardis_portal.api.FacilityResource.Meta attribute), 273*

queryset *(tardis.tardis_portal.api.GroupResource.Meta attribute), 274*

queryset *(tardis.tardis_portal.api.InstrumentResource.Meta attribute), 274*

queryset *(tardis.tardis_portal.api.LocationResource.Meta attribute), 274*

queryset *(tardis.tardis_portal.api.ObjectACLResource.Meta attribute), 275*

```

queryset (tardis.tardis_portal.api.ParameterNameResource.Meta registerReader () (tardis.apps.oaipmh.server.ProxyingMetadataRegis
attribute), 275 terExperimentForm (class in tardis.tardis_portal.forms), 284
queryset (tardis.tardis_portal.api.ReplicaResource.Meta registerWriter () (tardis.apps.oaipmh.server.ProxyingMetadataRegis
attribute), 276 terExperimentForm (class in tardis.tardis_portal.forms), 284
queryset (tardis.tardis_portal.api.SchemaResource.Meta registerWriter () (tardis.apps.oaipmh.server.ProxyingMetadataRegis
attribute), 277 terExperimentForm (class in tardis.tardis_portal.forms), 284
queryset (tardis.tardis_portal.api.StorageBoxAttributeReserREPRESENTATION (in module tardis.default_settings.auth), 151
attribute), 277
queryset (tardis.tardis_portal.api.StorageBoxOptionReserREPRESENTATION (in module tardis.tardis_portal.context_processors),
attribute), 277
queryset (tardis.tardis_portal.api.StorageBoxResource.Meta registerWriter () (tardis.apps.oaipmh.server.ProxyingMetadataRegis
attribute), 278 terExperimentForm (class in tardis.tardis_portal.forms), 284
queryset (tardis.tardis_portal.api.UserResource.Meta registerWriter () (tardis.apps.oaipmh.server.ProxyingMetadataRegis
attribute), 278 terExperimentForm (class in tardis.tardis_portal.forms), 284
queryset_pagination (tardis.apps.search.documents.DataFileDocument.Djangorelated_models (in module tardis.tardis_portal.context_processors),
attribute), 137
quote () (in module tardis.tardis_portal.tests.slapd), related_models (tardis.apps.search.documents.ExperimentDocument.Django
248 attribute), 138
RelatedInfoForm (class in tardis.apps.related_info.forms), 132
R
RANGE_COMPARISON (tardis.tardis_portal.models.parameters.ParameterValueMeta registerWriter () (tardis.apps.publication_workflow.managers.PublicationManager
attribute), 220
rapidConnectEduPersonTargetedID (tardis.tardis_portal.models.access_control.UserProfile) eased_publications ()
attribute), 189 (tardis.apps.publication_workflow.models.Publication.Publication
raw_id_fields (tardis.apps.eventlog.admin.LogAdmin method), 115
attribute), 90 remote_hosts (tardis.apps.push_to.models.Credential
rcauth () (in module tardis.tardis_portal.views.authentication), attribute), 121
attribute), 257 remote_user (tardis.apps.push_to.models.Credential
read_detail () (tardis.apps.sftp.api.SFTPACLAuthorization method), 141
attribute), 121
RemoteHost (class in tardis.apps.push_to.models), 125
RemoteHost.DoesNotExist, 125
read_detail () (tardis.tardis_portal.api.ACLOperation) RemoteHost.MultipleObjectsReturned, 125
method), 141 RemoteHostAdmin (class in tardis.tardis_portal.push_to.models), 126
remove_auth_method () (in module tardis.tardis_portal.auth.authentication),
method), 141 157
read_list () (tardis.tardis_portal.api.ACLOperation) remove_csrf_token () (in module
method), 269 tardis.tardis_portal.views.utils), 263
readMetadata () (tardis.apps.aoipmh.server.ProxyingMetadataRegisREPRESENTATION (in module tardis.tardis_portal.push_to.models), 126
method), 104
recall_dfo () (tardis.apps.hsm.api.ReplicaAppResource remove_licenses () (in module
method), 97 tardis.apps.publication_workflow.migrations.0001_initial),
RECALL_URI_TEMPLATES (in module tardis.default_settings.downloads), 153 111
recall_url (tardis.tardis_portal.models.datafile.DataFile) remove_licenses () (in module
attribute), 193 tardis.tardis_portal.migrations.0017_add_cc_licenses),
redirect_back_with_error () (in module tardis.tardis_portal.shortcuts), 289 185
refresh_token (tardis.tardis_portal.graphql.schema.tardisMutate) remove_metadata_schemas () (in module
attribute), 175 tardis.apps.publication_workflow.migrations.0001_initial),
111
RemovedInMyTardis42Warning, 279

```

RemovedInMyTardis43Warning, 279
 render() (*tardis.tardis_portal.forms.NoInput method*), 284
 render() (*tardis.tardis_portal.templatetags.capture.CaptureNode method*), 233
 render() (*tardis.tardis_portal.templatetags.dynurl.DynURLNode method*), 234
 render() (*tardis.tardis_portal.widgets.CommaSeparatedInput method*), 290
 render() (*tardis.tardis_portal.widgets.Label method*), 291
 render_error_message() (*in module tardis.apps.push_to.views*), 130
 render_error_message() (*in module tardis.tardis_portal.shortcuts*), 289
 RENDER_IMAGE_DATASET_SIZE_LIMIT (*in module tardis.default_settings.frontend*), 154
 RENDER_IMAGE_SIZE_LIMIT (*in module tardis.default_settings.frontend*), 155
 render_response_index() (*in module tardis.tardis_portal.shortcuts*), 289
 render_success_message() (*in module tardis.apps.push_to.views*), 130
 render_to_file() (*in module tardis.tardis_portal.shortcuts*), 290
 replaces (*tardis.tardis_portal.migrations.0001_squashed_0011_attributes*), 182
 ReplicaAppResource (*class in tardis.apps.hsm.api*), 96
 ReplicaAppResource (*class in tardis.apps.s3utils.api*), 133
 ReplicaAppResource.Meta (*class in tardis.apps.hsm.api*), 96
 ReplicaAppResource.Meta (*class in tardis.apps.s3utils.api*), 133
 ReplicaResource (*class in tardis.tardis_portal.api*), 276
 ReplicaResource.Meta (*class in tardis.tardis_portal.api*), 276
 ReplicaResourceTest (*class in tardis.tardis_portal.tests.api.test_replica_resource*), 238
 Request (*class in tardis.apps.push_to.models*), 127
 request (*tardis.apps.push_to.models.Progress attribute*), 125
 Request.DoesNotExist, 127
 Request.MultipleObjectsReturned, 127
 request_id (*tardis.apps.push_to.models.Progress attribute*), 125
 request_set (*tardis.apps.push_to.models.Credential attribute*), 121
 request_set (*tardis.apps.push_to.models.RemoteHost attribute*), 126
 REQUIRE_DATAFILE_CHECKSUMS
 REQUIRE_DATAFILE_SIZES (*tardis.default_settings attribute*), 59
 REQUIRE_SSL_TO_GENERATE_KEY (*in module tardis.apps.sftp.default_settings*), 142
 REQUIRE_VALIDATION_ON_INGESTION (*tardis.default_settings attribute*), 59
 requires_admin_approval() (*in module tardis.apps.social_auth.auth.social_auth*), 150
 resolve() (*tardis.tardis_portal.graphql.user.UserSignIn class method*), 179
 resolve_datafileobjects() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_datafiles() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_datasets() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_edge_count() (*tardis.tardis_portal.graphql.utils.ExtendedConnection method*), 179
 resolve_experiments()
 resolve_facilities() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_groups() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_instruments() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_parameternames() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_schemas() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_storageboxattributes() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_storageboxes() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_storageboxoptions() (*tardis.tardis_portal.graphql.schema.tardisQuery method*), 176
 resolve_total_count() (*tardis.tardis_portal.graphql.utils.ExtendedConnection method*), 180
 resolve_user() (*tardis.tardis_portal.graphql.schema.tardisQuery*)

```

        method), 176
resource_name (tardis.apps.hsm.api.DatasetAppResource.Meta)  retrace_retracted_pubs_list () (in module
attribute), 96                                     tardis.apps.publication_workflow.views), 118
resource_name (tardis.apps.hsm.api.ReplicaAppResource.Meta) retrace_scheduled_pubs_list () (in module
attribute), 97                                     tardis.apps.publication_workflow.views), 118
resource_name (tardis.apps.s3utils.api.ReplicaAppResource.Meta) retrace_shared_exps_list () (in module
attribute), 133                                     tardis.tardis_portal.views.ajax_pages), 257
resource_name (tardis.apps.search.api.AdvanceSearchAppResource.Meta) retrace_list () (in module
attribute), 136                                     tardis.tardis_portal.views.authorization),
resource_name (tardis.apps.search.api.SearchAppResource.Meta) 258
attribute), 137                                     retry (tardis.apps.push_to.models.Progress attribute),
resource_name (tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta)
attribute), 141                                     return_response_error () (in module
resource_name (tardis.tardis_portal.api.DataFileResource.Meta) tardis.tardis_portal.shortcuts), 290
attribute), 269                                     return_response_error_message () (in mod-
attribute), 269                                     ule tardis.tardis_portal.shortcuts), 290
restart () (tardis.tardis_portal.tests.slapd.Slapd
method), 247                                     return_response_not_found () (in module
RestfulExperimentParameterSet (class in
tardis.tardis_portal.shortcuts), 289             tardis.tardis_portal.shortcuts), 290
REUSE_DATASET_STORAGE_BOX (in module
retract () (tardis.apps.publication_workflow.models.Publication tardis.default_settings.storage), 156
method), 116                                     reverse_func () (in module
retract_publication () (in module
tardis.apps.publication_workflow.views),
117                                     tardis.apps.hsm.migrations.0001_initial_data),
retracted_publications ()                                     94
(tardis.apps.publication_workflow.managers.PublicationMahläger
method), 115                                     reverse_func () (in module
retracted_publications ()                                     tardis.tardis_portal.migrations.0017_add_cc_lenses),
(tardis.apps.publication_workflow.models.Publication.PublitionManager
method), 116                                     revert_to_draft ()
retrieve_access_list_group_READONLY () (in module
tardis.tardis_portal.views.authorization),
257                                     (tardis.apps.publication_workflow.models.Publication
method), 116                                     RevertFinalizeTestCase (class
in
tardis.tardis_portal.views.authorization), 118             tardis.apps.publication_workflow.tests.test_revert_finalize),
118                                     113
retrieve_access_list_user_READONLY () (in module
tardis.tardis_portal.views.authorization), RFC
257                                     revoke_token (tardis.tardis_portal.graphql.schema.tardisMutation
attribute), 175
RFC 3339, 289
retrieve_draft_pubs_list () (in module rfc3339 () (in module tardis.tardis_portal.rfc3339),
tardis.apps.publication_workflow.views), 118             289
retrieve_group_list () (in module RifCsExperimentProvider (class
tardis.tardis_portal.views.authorization),
258             tardis.apps.oaipmh.provider.experiment),
101
retrieve_group_list_by_user () (in module RifCsExperimentProvider.ExperimentWriter
tardis.tardis_portal.views.authorization), 258             (class in tardis.apps.oaipmh.provider.experiment),
101
retrieve_group_userlist_READONLY () (in module RifCsExperimentProviderTestCase (class in
tardis.tardis_portal.views.authorization), 258             tardis.apps.oaipmh.tests.provider.test_experiment),
103
retrieve_licenses () (in module RifCsProvider (class
tardis.tardis_portal.views.ajax_json), 257             in
tardis.tardis_portal.publish.provider.rifcsprovider),
232
retrieve_owned_exps_list () (in module RifCSTestCase (class
tardis.tardis_portal.views.ajax_pages), 257             in
tardis.tardis_portal.publish.provider.rifcsprovider),
232
retrieve_released_pubs_list () (in module RifCSTestCase (class
tardis.tardis_portal.publish.provider.rifcsprovider),
232

```

tardis.apps.related_info.tests.test_oaipmh), 130
RightsForm (class in tardis.tardis_portal.forms), 285
RightsForm.Meta (class in tardis.tardis_portal.forms), 285
RightsFormTestCase (class in tardis.tardis_portal.tests.test_forms), 251
RightsTestCase (class in tardis.tardis_portal.tests.views.test_auth_views), 244
RmExperimentTestCase (class in tardis.tardis_portal.tests.management.test_rmexperiment), 242
rollback_migration () (in module tardis.appsopenid_migration.utils), 111

S

S3_SIGNED_URL_EXPIRY (in module tardis.apps.s3utils.default_settings), 134
S3UtilsAppApiTestCase (class in tardis.apps.s3utils.tests.test_api), 132
S3UtilsAppChecksumsTestCase (class in tardis.apps.s3utils.tests.test_checksums), 133
S3UtilsConfig (class in tardis.apps.s3utils.apps), 134
S3UtilsConfigTestCase (class in tardis.apps.s3utils.tests.test_app_config), 133
safe (tardis.apps.publication_workflow.models.Publication attribute), 116
safe (tardis.tardis_portal.models.experiment.Experiment attribute), 203
sanitize_html () (in module tardis.tardis_portal.templatetags.formfieldfilters), 234
save () (tardis.apps.push_to.models.KeyPair method), 122
save () (tardis.tardis_portal.forms.ExperimentForm method), 282
save () (tardis.tardis_portal.forms.RegistrationForm method), 285
save () (tardis.tardis_portal.models.access_control.UserAuthentication method), 189
save () (tardis.tardis_portal.models.datafile.DataFile method), 193
save () (tardis.tardis_portal.models.datafile.DataFileObject method), 195
save () (tardis.tardis_portal.models.dataset.Dataset method), 199
save () (tardis.tardis_portal.models.experiment.Experiment method), 203
save () (tardis.tardis_portal.models.experiment.ExperimentAuthor method), 204
save () (tardis.tardis_portal.models.facility.Facility method), 206
save () (tardis.tardis_portal.models.instrument.Instrument method), 207
save () (tardis.tardis_portal.models.parameters.ExperimentParameter method), 214
save () (tardis.tardis_portal.models.parameters.ParameterSet method), 222
save_m2m () (tardis.tardis_portal.forms.ExperimentForm.FullExperiment method), 282
save_parameter_add_form () (in module tardis.tardis_portal.forms), 286
save_parameter_edit_form () (in module tardis.tardis_portal.forms), 286
save_with_random_token () (tardis.tardis_portal.models.token.Token method), 231
scheduled_publications () (tardis.apps.publication_workflow.managers.PublicationManager method), 115
scheduled_publications () (tardis.apps.publication_workflow.models.Publication.Published method), 116
Schema (class in tardis.tardis_portal.models.parameters), 223
schema (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 211
schema (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 213
schema (tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute), 215
schema (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 218
schema (tardis.tardis_portal.models.parameters.ParameterName attribute), 222
schema (tardis.tardis_portal.models.parameters.ParameterSet attribute), 222
schema (tardis.tardis_portal.shortcuts.RestfulExperimentParameterSet attribute), 289
Schema.DoesNotExist, 223
Schema.MultipleObjectsReturned, 223
Schema.UnsupportedType, 223
schema_id (tardis.tardis_portal.models.parameters.ParameterName attribute), 222
schema_id (tardis.tardis_portal.models.parameters.ParameterSet attribute), 223
SchemaAdmin (class in tardis.tardis_portal.admin), 266
SchemaManager (class in tardis.tardis_portal.managers), 288
SchemaResource (class in tardis.tardis_portal.api), 276
SchemaResource.Meta (class in tardis.tardis_portal.api), 276

```

SchemaResourceTest      (class      in    send_admin_email()      (in      module
    tardis.tardis_portal.tests.api.test_schema_resource),      tardis.apps.social_auth.auth.social_auth),
    238
schemas (tardis.tardis_portal.graphql.schema.tardisQuery.send_mail_to_authors()      (in      module
    attribute), 177
search_fields (tardis.apps.eventlog.admin.LogAdmin serializer (tardis.apps.search.api.AdvanceSearchAppResource.Meta
    attribute), 90
search_fields (tardis.tardis_portal.admin.DatafileAdmin serializer (tardis.apps.search.api.SearchAppResource.Meta
    attribute), 264
search_fields (tardis.tardis_portal.admin.DatasetAdmin serializer (tardis.tardis_portal.api.MyTardisModelResource.Meta
    attribute), 265
search_fields (tardis.tardis_portal.admin.ExperimentAdmin serializer (tardis.tardis_portal.api.UserResource.Meta
    attribute), 265
search_fields (tardis.tardis_portal.admin.FacilityAdmin serializerTest (class      in
    attribute), 265
search_fields (tardis.tardis_portal.admin.InstrumentAdmin      239
    attribute), 265
search_fields (tardis.tardis_portal.admin.ObjectACLAdmin      service (in module tardis.analytics.tracker), 88
    attribute), 266
search_fields (tardis.tardis_portal.admin.ParameterNameAdmin_started()
    attribute), 266
search_fields (tardis.tardis_portal.admin.SchemaAdmin      (tardis.apps.sftp.sftp.MyTSFTPServerInterface
    attribute), 266
search_fields (tardis.tardis_portal.admin.StorageBoxAdmin      method), 145
    attribute), 266
search_fields (tardis.tardis_portal.admin.UserAuthenticationAdmin.method), 247
    attribute), 268
set_debug ()      (tardis.tardis_portal.tests.slapd.Slapd
SearchAppResource      (class      in      (tardis.apps.publication_workflow.models.Publication
    tardis.apps.search.api), 136
method), 116
SearchAppResource.Meta      (class      in      set_param() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager
    tardis.apps.search.api), 136
method), 263
SearchConfig (class in tardis.apps.search.apps), 137
set_param_list () (tardis.tardis_portal.ParameterSetManager.ParameterSetManager
searchEntities () (tardis.tardis_portal.auth.authservice.AuthService
    method), 263
method), 159
set_params_from_dict ()
searchGroups () (tardis.tardis_portal.auth.authservice.AuthService
    method), 159
set_param()
searchGroups () (tardis.tardis_portal.auth.interfaces.GroupProvider)      (tardis.tardis_portal.tests.slapd.Slapd
    method), 247
searchGroups () (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider.authors()      (in      module
    method), 162
tardis.apps.publication_workflow.views),
searchGroups () (tardis.tardis_portal.auth.token_auth.TokenGroupProvider
    method), 163
set_root_cn ()
searchGroups () (tardis.tardis_portal.tests.test_authservice.MockGroupProvider
    method), 249
set_root_password()
SearchObject (class in tardis.apps.search.api), 137
(tardis.tardis_portal.tests.slapd.Slapd method),
searchUsers () (tardis.tardis_portal.auth.authservice.AuthService
    method), 247
method), 159
set_slapd_debug_level()
searchUsers () (tardis.tardis_portal.auth.interfaces.UserProvider(tardis.tardis_portal.tests.slapd.Slapd
    method), 161
method), 247
set_token()      (in      module
SearchView (class in tardis.apps.search.views), 139
SECRET_KEY, 58
select_forms ()      (in      module
    tardis.apps.publication_workflow.views),
    118
set_value() (tardis.tardis_portal.models.parameters.Parameter

```

method), 219
setPost () (tardis.tardis_portal.tests.test_authservice.MockRequest (tardis.apps.search.tests.test_api.SimpleSearchTest method), 135
settings (tardis.apps.search.documents.DataFileDocumentIndex (tardis.apps.search.tests.test_index.IndexExperimentTestCase attribute), 138
settings (tardis.apps.search.documents.DatasetDocumentIndex () (tardis.apps.sftp.sftp.MyTSFTPRequestHandler method), 144
settings (tardis.apps.search.documents.ExperimentDocumentIndex (tardis.apps.sftp.tests.test_sftp.SFTPTTest attribute), 138
setUp () (tardis.apps.hsm.tests.test_api.HsmAppApiTestCase setUp () (tardis.tardis_portal.tests.api.MyTardisResourceTestCase method), 240
setUp () (tardis.apps.hsm.tests.test_email_templates.HsmAppEmailTemplate (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResource method), 236
setUp () (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase (tardis.tardis_portal.tests.api.test_dataset_metadata_resources.List method), 236
setUp () (tardis.apps.oaipmh.tests.provider.test_experiments.AbstractExperimentAndProviderTests.api.test_dataset_resource.DatasetResource method), 236
setUp () (tardis.apps.oaipmh.tests.provider.test_oai.EndpointTestCase setUp () (tardis.tardis_portal.tests.api.test_experiment_resource.ExperimentResource method), 237
setUp () (tardis.apps.openid_migration.tests.test_forms.OpenIDMigrationForm (tardis.tardis_portal.tests.api.test_schema_resource.SchemaResource method), 238
setUp () (tardis.apps.openid_migration.tests.test_migration.OpenIDMigrationList (tardis.tardis_portal.tests.graphql.test_auth.userSignInTestCase method), 240
setUp () (tardis.apps.openid_migration.tests.test_models.ModelTestCase (tardis.tardis_portal.tests.graphql.test_facilities.createFacilityTestCase method), 240
setUp () (tardis.apps.openid_migration.tests.test_views.OpenIDMigrationView (tardis.tardis_portal.tests.graphql.test_facilities.facilitiesTestCase method), 240
setUp () (tardis.apps.publication_workflow.tests.test_forms.PublicationForm (tardis.tardis_portal.tests.graphql.test_facilities.getFacilitiesTestCase method), 241
setUp () (tardis.apps.publication_workflow.tests.test_my_publication.MyPublication (tardis.tardis_portal.tests.graphql.test_facilities.updateFacilityTestCase method), 241
setUp () (tardis.apps.publication_workflow.tests.test_reversefulltext.ReverseFullText (tardis.tardis_portal.tests.graphql.test_user.userSignInTestCase method), 241
setUp () (tardis.apps.publication_workflow.tests.test_tokens.PublicationToken (tardis.tardis_portal.tests.management.test_collectstatic.Collectors method), 241
setUp () (tardis.apps.push_to.tests.test_models.ModelsTestCase setUp () (tardis.tardis_portal.tests.management.test_dumpschemas.DumpSchema method), 242
setUp () (tardis.apps.related_info.tests.test_oaipmh.RifCSTestCase) (tardis.tardis_portal.tests.models.ModelTestCase method), 244
setUp () (tardis.apps.related_info.tests.tests.CreateTestCase setUp () (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase method), 248
setUp () (tardis.apps.related_info.tests.tests.DeleteTestCase setUp () (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase method), 248
setUp () (tardis.apps.related_info.tests.tests.GetTestCase setUp () (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase method), 249
setUp () (tardis.apps.related_info.tests.tests.ListTestCase setUp () (tardis.tardis_portal.tests.test_copy_move.CopyMoveTestCase method), 250
setUp () (tardis.apps.related_info.tests.tests.TabTestCase setUp () (tardis.tardis_portal.tests.test_df_save_metadata.DatafileSaveMethod), 250
setUp () (tardis.apps.related_info.tests.tests.UpdateTestCase setUp () (tardis.tardis_portal.tests.test_download.DownloadTestCase method), 250
setUp () (tardis.apps.s3utils.tests.test_api.S3UtilsAppApiTestCase) (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownload method), 250
setUp () (tardis.apps.s3utils.tests.test_checksums.S3UtilsAppChecksumList (tardis.tardis_portal.tests.test_facility_overview.FacilityOverview method), 250

```

        method), 251
setUp() (tardis.tardis_portal.tests.test_forms.RightsForm  SFTPPublicKey.DoesNotExist, 143
        method), 251
setUp() (tardis.tardis_portal.tests.test_iif.ExtraTestCase  SFTPPublicKey.MultipleObjectsReturned,
        method), 251
setUp() (tardis.tardis_portal.tests.test_iif.Level0TestCase  SFTPPublicKeyAppResource (class      in
        method), 251
setUp() (tardis.tardis_portal.tests.test_iif.Level1TestCase  SFTPPublicKeyAppResource.Meta (class   in
        method), 252
setUp() (tardis.tardis_portal.tests.test_iif.Level2TestCase  SFTPPublicKeyAppResource.attribute), 193
        method), 252
setUp() (tardis.tardis_portal.tests.test_ldap.LDAPTest  share() (in           module
        method), 252
setUp() (tardis.tardis_portal.tests.test_parameters.ParametersTestCase  tardis.tardis_portal.views.authorization),
        method), 253
setUp() (tardis.tardis_portal.tests.test_parametersets.EditParameterSetsTestCase  tardis.tardis_portal.managers.ExperimentManager
        method), 253
setUp() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase  (in           module
        method), 254
setUp() (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase  tardis.apps.push_to.utils), 129
        method), 254
setUp() (tardis.tardis_portal.tests.test_storage.ModelTestCase  (tardis.apps.sftp.sftp.MyTSFTPTCPServer
        method), 255
setUp() (tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase  sign_certificate() (in           module
        method), 255
setUp() (tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase  tardis.apps.push_to.ssh_authz), 129
        method), 255
setUp() (tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase  simple_search_public_data() (in           module
        method), 255
setUp() (tardis.tardis_portal.tests.test_tokens.TokenTestCase  tardis.apps.search.tests.test_api), 135
        method), 255
setUp() (tardis.tardis_portal.tests.views.test_auth_views.UserGroupTestCase  SINGLE_SEARCH_ENABLED (in           module
        method), 244
setUp() (tardis.tardis_portal.tests.views.test_auth_views.UserListTestCase  tardis.default_settings.settings.search), 155
        method), 244
setUp() (tardis.tardis_portal.tests.views.test_auth_views.UserListTestCase  single_search_processor() (in           module
        method), 245
setUp() (tardis.tardis_portal.tests.views.test_contextual_views.ContextedViewTestCase  tardis.tardis_portal.context_processors),
        method), 245
setUp() (tardis.tardis_portal.tests.views.test_contextual_views.ContextedViewTestCase  tardis.tardis_portal.views.pages), 261
        method), 245
setUp() (tardis.tardis_portal.tests.views.test_experiment_views.ExperimentTestCase  (in           module
        method), 245
setUp() (tardis.tardis_portal.tests.views.test_template_contexts.ViewTemplateContextsTest  tardis.tardis_portal.views.machine), 259
        method), 246
setUp() (tardis.tardis_portal.tests.views.test_upload_views.UploadTestCase  tardis.default_settings.site_customisations),
        method), 246
sftp_access() (in module tardis.apps.sftp.views), 149
sftp_keys() (in module tardis.apps.sftp.views), 149
SFTP_USERNAME_ATTRIBUTE (in module tardis.apps.sftp.default_settings), 142
SFTPAuthorization (class      in tardis.apps.sftp.api), 140
SFTPConfig (class in tardis.apps.sftp.apps), 142
SFTPManagementTestCase (class      in tardis.apps.sftp.tests.test_sftp), 140
SFTPPublicKey (class in tardis.apps.sftp.models), 143

```

method), 248
start_server() (in module tardis.apps.sftp.sftp), 148
start_time (tardis.tardis_portal.models.experiment.Experiment attribute), 203
started() (tardis.tardis_portal.tests.slapd.Slapd method), 248
stat() (tardis.apps.sftp.sftp.MyTSFTPHandle method), 144
stat() (tardis.apps.sftp.sftp.MyTSFTPServerInterface method), 146
STATIC_ROOT (tardis.default_settings attribute), 61
STATIC_URL (tardis.default_settings attribute), 61
StaticField (class in tardis.tardis_portal.forms), 285
stats() (in module tardis.tardis_portal.views.pages), 261
status (tardis.apps.push_to.models.Progress attribute), 125
status (tardis.tardis_portal.models.datafile.DataFile attribute), 193
status (tardis.tardis_portal.models.storage.StorageBox attribute), 228
status_code (tardis.tardis_portal.views.utils.HttpResponseMethod attribute), 262
status_code (tardis.tardis_portal.views.utils.HttpResponseMethod attribute), 262
stop() (tardis.tardis_portal.tests.slapd.Slapd method), 248
storage_box (tardis.tardis_portal.models.datafile.DataFileObject attribute), 195
storage_box (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 211
storage_box (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 213
storage_box (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 215
storage_box (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 218
storage_box (tardis.tardis_portal.models.parameters.Parameter attribute), 223
storage_box (tardis.tardis_portal.models.storage.StorageBoxAttribute attribute), 229
storage_box (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 230
storage_box_id (tardis.tardis_portal.models.datafile.DataFileObject attribute), 195
storage_box_id (tardis.tardis_portal.models.storage.StorageBoxAttribute attribute), 229
storage_box_id (tardis.tardis_portal.models.storage.StorageBoxAttribute attribute), 230
storage_type (tardis.tardis_portal.models.datafile.DataFileObject attribute), 195
storage_type (tardis.tardis_portal.models.storage.StorageBox attribute), 228
StorageBox (class in tardis.tardis_portal.models.storage), 225
StorageBox.DoesNotExist, 225
StorageBox.MultipleObjectsReturned, 225
StorageBoxAdmin (class in tardis.tardis_portal.admin), 266
StorageBoxAttribute (class in tardis.tardis_portal.models.storage), 228
StorageBoxAttribute.DoesNotExist, 229
StorageBoxAttribute.MultipleObjectsReturned, 229
StorageBoxAttributeInline (class in tardis.tardis_portal.admin), 266
StorageBoxAttributeInlineForm (class in tardis.tardis_portal.admin), 267
StorageBoxAttributeInlineForm.Meta (class in tardis.tardis_portal.admin), 267
StorageBoxAttributeResource (class in tardis.tardis_portal.api), 277
StorageBoxAttributeResource.Meta (class in tardis.tardis_portal.api), 277
StorageBoxAttributeResourceTest (class in tardis.tardis_portal.tests.api.test_storagebox_resources), 239
StorageBoxAttributeType (class in tardis.tardis_portal.graphql.storageboxattribute), 178
StorageBoxAttributeTypeFilter (class in tardis.tardis_portal.graphql.storageboxattribute), 178
StorageBoxAttributeTypeFilter.Meta (class in tardis.tardis_portal.graphql.storageboxattribute), 178
StorageBoxAttributeTypeFilter.Meta (class in tardis.tardis_portal.graphql.storageboxattribute), 178
StorageBoxAttributeTypeForm (class in tardis.tardis_portal.admin), 267
StorageBoxAttributeTypeForm.Meta (class in tardis.tardis_portal.admin), 267
StorageBoxOption (class in tardis.tardis_portal.models.storage), 229
StorageBoxOption.DoesNotExist, 229
StorageBoxOption.MultipleObjectsReturned, 229
StorageBoxOptionInline (class in tardis.tardis_portal.admin), 268
StorageBoxOptionInlineForm (class in tardis.tardis_portal.admin), 268
StorageBoxOptionInlineForm.Meta (class in tardis.tardis_portal.admin), 268
StorageBoxOptionResource (class in tardis.tardis_portal.admin), 268
StorageBoxOptionResource (class in tardis.tardis_portal.admin), 268

```

    tardis.tardis_portal.api), 277
StorageBoxOptionResource.Meta (class in supports_object_permissions
    tardis.tardis_portal.api), 277 (tardis.tardis_portal.auth.authorisation.ACIAwareBackend
                                     attribute), 158
StorageBoxOptionResourceTest (class in SYSTEM OWNED (tardis.tardis_portal.models.access_control.ObjectACL
    tardis.tardis_portal.tests.api.test_storagebox_resources), attribute), 187
    239 system_owned_groups ()
storageboxoptions (tardis.tardis_portal.managers.ExperimentManager
    attribute), 177 method), 288
StorageBoxOptionType (class in T
    tardis.tardis_portal.graphql.storageboxoption), TabTestCase (class in
    178 tardis.apps.related_info.tests.tests), 131
StorageBoxOptionTypeFilter (class in tag (tardis.tardis_portal.widgets.Label attribute), 291
    tardis.tardis_portal.graphql.storageboxoption), tag (tardis.tardis_portal.widgets.Span attribute), 291
    178 TAPE (tardis.tardis_portal.models.storage.StorageBox
    attribute), 225
StorageBoxOptionTypeFilter.Meta (class in tardis (module), 295
    tardis.tardis_portal.graphql.storageboxoption), tardis.analytics (module), 89
    179 StorageBoxResource (class in tardis.analytics.apps (module), 88
    tardis.tardis_portal.api), 277 tardis.analytics.ga (module), 88
StorageBoxResource.Meta (class in tardis.analytics.tracker (module), 88
    tardis.tardis_portal.api), 277 tardis.app_config (module), 293
StorageBoxResourceTest (class in tardis.apps (module), 150
    tardis.tardis_portal.tests.api.test_storagebox_resources) tardis.apps.deep_storage_download_mapper
    239 (module), 89
StorageBoxType (class in tardis.apps.deep_storage_download_mapper.mapper
    tardis.tardis_portal.graphql.storagebox), (module), 89
    177 StorageBoxTypeFilter (class in tardis.apps.deep_storage_download_mapper.urls
    tardis.tardis_portal.graphql.storagebox), (module), 89
    177 StorageBoxTypeFilter.Meta (class in tardis.apps.dl_mapper_df_dir_only (mod-
    tardis.tardis_portal.graphql.storagebox), ule), 90
    177 StorageClassNotFoundError, 98 tardis.apps.dl_mapper_df_dir_only.mapper
    tardis.tardis_portal.download, 280 (module), 89
    tardis.apps.eventlog (module), 93
    tardis.apps.eventlog.admin (module), 90
    STRING (tardis.tardis_portal.models.parameters.ParameterName tardis.apps.eventlog.default_settings
    attribute), 220 (module), 90
    STRING (tardis.tardis_portal.models.storage.StorageBoxOption tardis.apps.eventlog.fields (module), 91
    attribute), 229 tardis.apps.eventlog.migrations (module),
    string_value (tardis.tardis_portal.models.parameters.Parameter 90
    attribute), 219 tardis.apps.eventlog.migrations.0001_initial
    submit_form () (in module tardis.apps.publication_workflow.views), (module), 90
    118 subtype (tardis.tardis_portal.models.parameters.Schema tardis.apps.eventlog.models (module), 91
    attribute), 225 tardis.apps.eventlog.signals (module), 93
    suite () (in module tardis.tardis_portal.tests.tests), 256 tardis.apps.eventlog.urls (module), 93
    sum_sizes () (tardis.tardis_portal.models.datafile.DataFile tardis.apps.eventlog.utils (module), 93
    class method), 193 tardis.apps.filepicker (module), 94
    supports_anonymous_user (tardis.tardis_portal.auth.authorisation.ACIAwareBackend
    attribute), 158 tardis.apps.filepicker.filepicker_settings
                                     (module), 93
                                     tardis.apps.filepicker.urls (module), 93
                                     tardis.apps.filepicker.utils (module), 93
                                     tardis.apps.filepicker.views (module), 94

```

tardis.apps.hsm (*module*), 99
tardis.apps.hsm.api (*module*), 96
tardis.apps.hsm.apps (*module*), 97
tardis.apps.hsm.check (*module*), 97
tardis.apps.hsm.default_settings (*module*), 97
tardis.apps.hsm.email_text (*module*), 98
tardis.apps.hsm.exceptions (*module*), 98
tardis.apps.hsm.migrations (*module*), 94
tardis.apps.hsm.migrations.0001_initial (*module*), 94
tardis.apps.hsm.storage (*module*), 98
tardis.apps.hsm.tasks (*module*), 98
tardis.apps.hsm.tests (*module*), 96
tardis.apps.hsm.tests.test_api (*module*), 94
tardis.apps.hsm.tests.test_app_config (*module*), 95
tardis.apps.hsm.tests.test_email_template (*module*), 95
tardis.apps.hsm.tests.test_migrations (*module*), 96
tardis.apps.hsm.urls (*module*), 98
tardis.apps.hsm.utils (*module*), 99
tardis.apps.oaipmh (*module*), 105
tardis.apps.oaipmh.models (*module*), 103
tardis.apps.oaipmh.provider (*module*), 102
tardis.apps.oaipmh.provider.base (*module*), 99
tardis.apps.oaipmh.provider.experiment (*module*), 101
tardis.apps.oaipmh.server (*module*), 104
tardis.apps.oaipmh.tests (*module*), 103
tardis.apps.oaipmh.tests.provider (*module*), 103
tardis.apps.oaipmh.tests.provider.test_base (*module*), 102
tardis.apps.oaipmh.tests.provider.test_provider (*module*), 102
tardis.apps.oaipmh.tests.test_oai (*module*), 103
tardis.apps.oaipmh.urls (*module*), 105
tardis.apps.oaipmh.views (*module*), 105
tardis.appsopenid_migration (*module*), 111
tardis.appsopenid_migration.apps (*module*), 107
tardis.appsopenid_migration.context_protocols (*module*), 107
tardis.appsopenid_migration.default_settings (*module*), 107
tardis.appsopenid_migration.email_text (*module*), 107
tardis.appsopenid_migration.forms (*module*), 107
tardis.appsopenid_migration.migration (*module*), 107
tardis.appsopenid_migration.migrations (*module*), 106
tardis.appsopenid_migration.migrations.0001_initial (*module*), 106
tardis.appsopenid_migration.models (*module*), 108
tardis.appsopenid_migration.tasks (*module*), 110
tardis.appsopenid_migration.tests (*module*), 107
tardis.appsopenid_migration.tests.test_forms (*module*), 106
tardis.appsopenid_migration.tests.test_migration (*module*), 106
tardis.appsopenid_migration.tests.test_models (*module*), 106
tardis.appsopenid_migration.tests.test_views (*module*), 107
tardis.appsopenid_migration.urls (*module*), 110
tardis.appsopenid_migration.user_menu_modifiers (*module*), 110
tardis.appsopenid_migration.utils (*module*), 111
tardis.appsopenid_migration.views (*module*), 111
tardis.appspublication_workflow (*module*), 118
tardis.appspublication_workflow.apps (*module*), 113
tardis.appspublication_workflow.default_settings (*module*), 114
tardis.appspublication_workflow.doi_base (*module*), 114
tardis.appspublication_workflow.email_text_experiments (*module*), 114
tardis.appspublication_workflow.managers (*module*), 114
tardis.appspublication_workflow.migrations (*module*), 112
tardis.appspublication_workflow.migrations.0001_initial (*module*), 111
tardis.appspublication_workflow.models (*module*), 115
tardis.appspublication_workflow.tasks (*module*), 117
tardis.appspublication_workflow.tests (*module*), 113
tardis.appspublication_workflow.tests.test_email_text (*module*), 112
tardis.appspublication_workflow.tests.test_form (*module*), 112

tardis.apps.publication_workflow.tests.tearDown()
(module), 133
tardis.apps.s3utils.apps(module), 134

tardis.apps.publication_workflow.tests.tearDown()
(module), 133
tardis.apps.s3utils.apps(module), 134

tardis.apps.publication_workflow.urls
(module), 117

tardis.apps.publication_workflow.utils
(module), 117

tardis.apps.publication_workflow.views
(module), 117

tardis.apps.push_to(module), 130

tardis.apps.push_to.apps(module), 120

tardis.apps.push_to.exceptions
(module), 120

tardis.apps.push_to.migrations
(module), 119

tardis.apps.push_to.migrations.0001_init
(module), 119

tardis.apps.push_to.migrations.0002_auto_20160518095000
(module), 119

tardis.apps.push_to.migrations.0003_auto_20210216032000
(module), 119

tardis.apps.push_to.models(module), 120

tardis.apps.push_to.oauth_tokens
(module), 128

tardis.apps.push_to.ssh_authz
(module), 129

tardis.apps.push_to.tasks(module), 129

tardis.apps.push_to.tests(module), 120

tardis.apps.push_to.tests.test_models
(module), 119

tardis.apps.push_to.urls(module), 129

tardis.apps.push_to.utils(module), 129

tardis.apps.push_to.views(module), 129

tardis.apps.related_info(module), 132

tardis.apps.related_info.forms
(module), 132

tardis.apps.related_info.models
(module), 132

tardis.apps.related_info.settings
(module), 132

tardis.apps.related_info.tests
(module), 132

tardis.apps.related_info.tests.test_oai_pmh
(module), 130

tardis.apps.related_info.tests.tests
(module), 131

tardis.apps.related_info.urls
(module), 132

tardis.apps.related_info.views
(module), 132

tardis.apps.s3utils(module), 135

tardis.apps.social_auth.default_settings
(*module*), 150

tardis.celery (*module*), 294

tardis.default_settings (*module*), 157

tardis.default_settings.admins (*module*),
150

tardis.default_settings.analytics (*mod-
ule*), 151

tardis.default_settings.apps (*module*), 151

tardis.default_settings.apps_default_sett
and\$
(*module*), 151

tardis.default_settings.auth (*module*), 151

tardis.default_settings.caches (*module*),
151

tardis.default_settings.celery_settings
(*module*), 151

tardis.default_settings.custom_views
(*module*), 151

tardis.default_settings.database (*mod-
ule*), 152

tardis.default_settings.debug (*module*),
152

tardis.default_settings.downloads (*mod-
ule*), 153

tardis.default_settings.email (*module*),
154

tardis.default_settings.filters (*module*),
154

tardis.default_settings.frontend (*mod-
ule*), 154

tardis.default_settings.graphql (*module*),
155

tardis.default_settings.i18n (*module*), 155

tardis.default_settings.localisation
(*module*), 155

tardis.default_settings.logging (*module*),
155

tardis.default_settings.middlewares
(*module*), 155

tardis.default_settings.publication
(*module*), 155

tardis.default_settings.search (*module*),
155

tardis.default_settings.sharing (*module*),
156

tardis.default_settings.site_customisat
asdis.tardis_portal.graphql.group (*mod-
ule*), 156

tardis.default_settings.static_files
(*module*), 156

tardis.default_settings.storage (*module*),
156

tardis.default_settings.templates (*mod-
ule*), 156

tardis.default_settings.uploads (*module*),
156

156

tardis.default_settings.urls (*module*), 156

tardis.schema (*module*), 294

tardis.tardis_portal (*module*), 292, 295

tardis.tardis_portal.admin (*module*), 263

tardis.tardis_portal.api (*module*), 268

tardis.tardis_portal.auth (*module*), 163

tardis.tardis_portal.auth.authentication
(*module*), 157

tardis.tardis_portal.auth.authorization
(*module*), 158

tardis.tardis_portal.auth.authservice
(*module*), 158

tardis.tardis_portal.auth.decorators
(*module*), 159

tardis.tardis_portal.auth.fix_circular
(*module*), 161

tardis.tardis_portal.auth.interfaces
(*module*), 161

tardis.tardis_portal.auth.localdb_auth
(*module*), 162

tardis.tardis_portal.auth.token_auth
(*module*), 162

tardis.tardis_portal.auth.utils (*module*),
163

tardis.tardis_portal.context_processors
(*module*), 278

tardis.tardis_portal.deprecations (*mod-
ule*), 279

tardis.tardis_portal.download (*module*),
279

tardis.tardis_portal.email (*module*), 280

tardis.tardis_portal.fields (*module*), 280

tardis.tardis_portal.forms (*module*), 280

tardis.tardis_portal.graphql (*module*), 180

tardis.tardis_portal.graphql.datafile
(*module*), 163

tardis.tardis_portal.graphql.datafileobject
(*module*), 164

tardis.tardis_portal.graphql.dataset
(*module*), 165

tardis.tardis_portal.graphql.experiment
(*module*), 166

tardis.tardis_portal.graphql.facility
(*module*), 167

tardis.tardis_portal.graphql.group (*mod-
ule*), 168

tardis.tardis_portal.graphql.instrument
(*module*), 168

tardis.tardis_portal.graphql.parameters
(*module*), 169

tardis.tardis_portal.graphql.schema
(*module*), 175

tardis.tardis_portal.graphql.storagebox

(*module*), 177 tardis.tardis_portal.migrations.0013_auto_20181002_0014_auto_20181002_0015_dataset_create (*module*), 184
tardis.tardis_portal.graphql.storageboxattribute (*module*), 177 tardis.tardis_portal.migrations.0014_auto_20181002_0015_dataset_create (*module*), 184
tardis.tardis_portal.graphql.storageboxoption (*module*), 178 tardis.tardis_portal.migrations.0015_dataset_create (*module*), 184
tardis.tardis_portal.graphql.user (*module*), 179 tardis.tardis_portal.migrations.0016_add_timestamps (*module*), 184
tardis.tardis_portal.graphql.utils (*module*), 179 tardis.tardis_portal.migrations.0017_add_cc_license (*module*), 184
tardis.tardis_portal.iif (*module*), 286 tardis.tardis_portal.migrations.0018_make_default_s3_bucket (*module*), 185
tardis.tardis_portal.logging_middleware (*module*), 286 tardis.tardis_portal.migrations.0018_make_default_s3_bucket (*module*), 185
tardis.tardis_portal.management (*module*), 181 tardis.tardis_portal.models (*module*), 231, 292, 295
tardis.tardis_portal.management.commands (*module*), 181 tardis.tardis_portal.models.access_control (*module*), 185
tardis.tardis_portal.management.commands.dataset (*module*), 180 tardis.tardis_portal.models.datafile (*module*), 190
tardis.tardis_portal.management.commands.datasetadd (*module*), 180 tardis.tardis_portal.models.dataset (*module*), 196
tardis.tardis_portal.management.commands.datasetaddschema (*module*), 181 tardis.tardis_portal.models.experiment (*module*), 200
tardis.tardis_portal.management.commands.datasetaddschemaadd (*module*), 181 tardis.tardis_portal.models.facility (*module*), 205
tardis.tardis_portal.managers (*module*), 286 tardis.tardis_portal.models.hooks (*module*), 206
tardis.tardis_portal.migrations (*module*), 185 tardis.tardis_portal.models.instrument (*module*), 206
tardis.tardis_portal.migrations.0001_init (*module*), 181 tardis.tardis_portal.models.jti (*module*), 207
tardis.tardis_portal.migrations.0001_squashed_001_fixes (*module*), 182 tardis.tardis_portal.models.license (*module*), 208
tardis.tardis_portal.migrations.0002_auto_20150528_1200 (*module*), 182 tardis.tardis_portal.models.parameters (*module*), 209
tardis.tardis_portal.migrations.0003_auto_20150607_1515 (*module*), 182 tardis.tardis_portal.models.storage (*module*), 225
tardis.tardis_portal.migrations.0004_storageboxprivilieges_patched (*module*), 182 tardis.tardis_portal.models.token (*module*), 230
tardis.tardis_portal.migrations.0005_datafilesatardisepontatoParameterSetManager (*module*), 182 tardis.tardis_portal.publish.parameter_set_manager (*module*), 263
tardis.tardis_portal.migrations.0006_datafilesremoved (*module*), 183 tardis.tardis_portal.publish.provider (*module*), 232
tardis.tardis_portal.migrations.0007_remove_parameter_set_index (*module*), 183 tardis.tardis_portal.publish.provider.rifcsprovider (*module*), 230
tardis.tardis_portal.migrations.0008_string_val (*module*), 183 tardis.tardis_portal.publish.publishservice (*module*), 231
tardis.tardis_portal.migrations.0009_auto_20160414_1200 (*module*), 183 tardis.tardis_portal.rfc3339 (*module*), 289
tardis.tardis_portal.migrations.0010_auto_20160503_1448 (*module*), 183 tardis.tardis_portal.shortcuts (*module*), 289
tardis.tardis_portal.migrations.0011_auto_20160505_1648 (*module*), 184 tardis.tardis_portal.storage (*module*), 232
tardis.tardis_portal.migrations.0012_userauthen (*module*), 184 tardis.tardis_portal.storage.file_system (*module*), 232
tardis.tardis_portal.migrations.0012_userapproved (*module*), 184 tardis.tardis_portal.tasks (*module*), 290

tardis.tardis_portal.templatetags (module), 238
 tardis.tardis_portal.templatetags.approved_user (module), 238
 tardis.tardis_portal.templatetags.basiccomparis (module), 239
 tardis.tardis_portal.templatetags.bleach_tag (module), 239
 tardis.tardis_portal.templatetags.capture (module), 239
 tardis.tardis_portal.templatetags.dynurl (module), 240
 tardis.tardis_portal.templatetags.experiment_t (module), 241
 tardis.tardis_portal.templatetags.experimentsta (module), 240
 tardis.tardis_portal.templatetags.facility_tags (module), 240
 tardis.tardis_portal.templatetags.feed (module), 241
 tardis.tardis_portal.templatetags.formfieldfilt (module), 246
 tardis.tardis_portal.templatetags.lookupfilters (module), 242
 tardis.tardis_portal.templatetags.pagination (module), 241
 tardis.tardis_portal.templatetags.xmldate (module), 242
 tardis.tardis_portal.tests (module), 256
 tardis.tardis_portal.tests.api (module), 239
 tardis.tardis_portal.tests.api.test_auth (module), 235
 tardis.tardis_portal.tests.api.test_datafa (module), 235
 tardis.tardis_portal.tests.api.test_datafa (module), 236
 tardis.tardis_portal.tests.api.test_datafa (module), 236
 tardis.tardis_portal.tests.api.test_exper (module), 237
 tardis.tardis_portal.tests.api.test_exper (module), 237
 tardis.tardis_portal.tests.api.test_facilit (module), 237
 tardis.tardis_portal.tests.api.test_groupa (module), 238
 tardis.tardis_portal.tests.api.test_instrum (module), 238
 tardis.tardis_portal.tests.api.test_listta (module), 238
 tardis.tardis_portal.tests.api.test_repli (module), 238
 tardis.tardis_portal.tests.api.test_schema_ (module), 235
 tardis.tardis_portal.tests.api.test_serializer (module), 233
 tardis.tardis_portal.tests.api.test_storag (module), 239
 tardis.tardis_portal.tests.api.test_user_r (module), 233
 tardis.tardis_portal.tests.auth (module), 240
 tardis.tardis_portal.tests.graphql (modu (module), 234
 tardis.tardis_portal.tests.graphql.test_a (module), 234
 tardis.tardis_portal.tests.graphql.test_f (module), 234
 tardis.tardis_portal.tests.graphql.test_u (module), 234
 tardis.tardis_portal.tests.ldap_ldif (module), 246
 tardis.tardis_portal.tests.management (module), 242
 tardis.tardis_portal.tests.management.test_c (module), 235
 tardis.tardis_portal.tests.management.test_c (module), 235
 tardis.tardis_portal.tests.management.test_d (module), 242
 tardis.tardis_portal.tests.management.test_e (module), 242
 tardis.tardis_portal.tests.management.test_f (module), 242
 tardis.tardis_portal.tests.management.test_g (module), 242
 tardis.tardis_portal.tests.management.test_h (module), 242
 tardis.tardis_portal.tests.management.test_i (module), 242
 tardis.tardis_portal.tests.management.test_j (module), 242
 tardis.tardis_portal.tests.management.test_k (module), 242
 tardis.tardis_portal.tests.management.test_l (module), 242
 tardis.tardis_portal.tests.management.test_m (module), 242
 tardis.tardis_portal.tests.management.test_n (module), 242
 tardis.tardis_portal.tests.management.test_o (module), 242
 tardis.tardis_portal.tests.management.test_p (module), 242
 tardis.tardis_portal.tests.management.test_q (module), 242
 tardis.tardis_portal.tests.management.test_r (module), 242
 tardis.tardis_portal.tests.management.test_s (module), 242
 tardis.tardis_portal.tests.management.test_t (module), 242
 tardis.tardis_portal.tests.management.test_u (module), 242
 tardis.tardis_portal.tests.management.test_v (module), 242
 tardis.tardis_portal.tests.management.test_w (module), 242
 tardis.tardis_portal.tests.management.test_x (module), 242
 tardis.tardis_portal.tests.management.test_y (module), 242
 tardis.tardis_portal.tests.management.test_z (module), 242
 tardis.tardis_portal.tests.models (module), 244
 tardis.tardis_portal.tests.models.test_authors (module), 243
 tardis.tardis_portal.tests.models.test_datafile (module), 243
 tardis.tardis_portal.tests.models.test_dataset (module), 243
 tardis.tardis_portal.tests.models.test_dfo (module), 243
 tardis.tardis_portal.tests.models.test_experiment (module), 243
 tardis.tardis_portal.tests.models.test_generate_ap (module), 243
 tardis.tardis_portal.tests.models.test_instrument (module), 244
 tardis.tardis_portal.tests.models.test_parameter (module), 244
 tardis.tardis_portal.tests.slapd (module), 247
 tardis.tardis_portal.tests.test_authentication

```

    (module), 248
tardis.tardis_portal.tests.test_authorisation (module), 248
tardis.tardis_portal.tests.test_authservice (module), 249
tardis.tardis_portal.tests.test_copy_move (module), 249
tardis.tardis_portal.tests.test_df_save_metadata (module), 250
tardis.tardis_portal.tests.test_download (module), 250
tardis.tardis_portal.tests.test_download_apikey (module), 250
tardis.tardis_portal.tests.test_facility_overview (module), 251
tardis.tardis_portal.tests.test_forms (module), 251
tardis.tardis_portal.tests.test_iiif (module), 251
tardis.tardis_portal.tests.test_ldap (module), 252
tardis.tardis_portal.tests.test_models (module), 253
tardis.tardis_portal.tests.test_parameters (module), 253
tardis.tardis_portal.tests.test_parametersets (module), 253
tardis.tardis_portal.tests.test_publish (module), 254
tardis.tardis_portal.tests.test_storage (module), 255
tardis.tardis_portal.tests.test_tar_downloads (module), 255
tardis.tardis_portal.tests.test_tasks (module), 255
tardis.tardis_portal.tests.test_tokens (module), 255
tardis.tardis_portal.tests.tests (module), 256
tardis.tardis_portal.tests.urls (module), 256
tardis.tardis_portal.tests.views (module), 246
tardis.tardis_portal.tests.views.test_authentication (module), 244
tardis.tardis_portal.tests.views.test_content_type (module), 245
tardis.tardis_portal.tests.views.test_experiment (module), 245
tardis.tardis_portal.tests.views.test_facility (module), 245
tardis.tardis_portal.tests.views.test_graphql (module), 245
tardis.tardis_portal.tests.views.test_upload_view (module), 246
tardis.tardis_portal.util (module), 290
tardis.tardis_portal.views (module), 263
tardis.tardis_portal.views.ajax_actions (module), 292, 295
tardis.tardis_portal.views.ajax_json (module), 257
tardis.tardis_portal.views.ajax_pages (module), 257
tardis.tardis_portal.views.authentication (module), 257
tardis.tardis_portal.views.authorization (module), 257
tardis.tardis_portal.views.facilities (module), 258
tardis.tardis_portal.views.images (module), 258
tardis.tardis_portal.views.machine (module), 259
tardis.tardis_portal.views.pages (module), 259
tardis.tardis_portal.views.parameters (module), 262
tardis.tardis_portal.views.upload (module), 262
tardis.tardis_portal.views.utils (module), 262
tardis.tardis_portal.widgets (module), 290
tardis.tardis_portal.xmlwriter (module), 291
tardis.tardis_portal.test_on_mysql_settings (module), 294
tardis.tardis_portal.test_on_postgresql_settings (module), 294
tardis.test_settings (module), 294
tardis.urls (module), 293
tardis.urls.accounts (module), 292
tardis.urls.ajax (module), 292
tardis.urls.api (module), 292
tardis.urls.core (module), 292
tardis.urls.datafile (module), 293
tardis.urls.dataset (module), 293
tardis.urls.display (module), 293
tardis.urls.download (module), 293
tardis.urls.experiment (module), 293
tardis.urls.facility (module), 293
tardis.urls.graphql (module), 293
tardis.urls.token (module), 293
tardis.urls.view (module), 294
tardisMutation (class), in
tardisQuery (class), in
TardisSchemaType (class), in
tardis.tardis_portal.graphql.parameters),

```



```

(tardis.apps.hsm.tests.test_app_config.HsmAppConfigTestCase.method), 119
    method), 95
test_app_config()                               test_cybderduck_connection_window()
    (tardis.apps.s3utils.tests.test_app_config.S3UtilsConfigTestCase.method), 140
    method), 133
test_authenticate()                           test_datafile() (tardis.tardis_portal.tests.models.test_datafile.DataFile
    (tardis.tardis_portal.tests.test_ldap.LDAPTest   method), 243
    method), 252
test_authors() (tardis.tardis_portal.tests.models.test_authors.AuthorTestCase)
    method), 243
test_bad_credentials()                      (tardis.tardis_portal.tests.views.test_experiment_views.ExperimentView
    (tardis.tardis_portal.tests.api.test_auth.MyTardisAuthenticationTest_view()
    method), 245
method), 235
test_cache() (tardis.tardis_portal.tests.test_copy_move.CopyMoveTestCase)
    method), 246
    method), 250
test_debug_serializer()
test_checksums() (tardis.apps.s3utils.tests.test_checksums.S3UtilsChecksumTestCase)
    method), 239
test_collectstatic()                         test_default_expiry()
    (tardis.tardis_portal.tests.management.test_collectstatic.CollectionStaticListPortal
    method), 256
method), 241
test_copy() (tardis.tardis_portal.tests.test_copy_move.CopyMoveTestCase)
    method), 250
    method), 250
test_create_and_edit()                      (tardis.tardis_portal.tests.views.test_experiment_views.ExperimentTestCase)
    (tardis.tardis_portal.tests.views.test_experiment_views.ExperimentTestCase_without_uri()
    method), 245
method), 245
test_create_dataset_pset()                  (tardis.tardis_portal.tests.models.test_dfo.DataFileObjectTestCase)
    (tardis.tardis_portal.tests.api.test_dataset_metadata_resources.DatasetParameterSetResourceTest
    method), 243
method), 243
test_create_dataset_pset_no_auth()          (tardis.tardis_portal.tests.api.test_dataset_metadata_resources.DatasetParameterSetResourceTest
    (tardis.tardis_portal.tests.test_df_save_metadata.DatafileSaveMe
    method), 236
method), 236
test_create_df_for_staging()                (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResourceTest)
    (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase)
    method), 236
method), 250
test_create_draft_publication()             (tardis.apps.publication_workflow.tests.test_form.PublicationFormTestCase)
    (tardis.apps.openid_migration.tests.test_migration.OpenIDMigration
    method), 112
method), 106
test_create_facility()                     (tardis.tardis_portal.tests.graphql.test_facilities.createFacilityTestCase)
    (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownload
    method), 240
method), 251
test_create_index()                        (tardis.apps.search.tests.test_index.IndexExperimentTestCase)
    (tardis.apps.s3utils.tests.test_api.S3UtilsAppApiTestCase)
    method), 135
method), 135
test_create_token()                        (tardis.apps.publication_workflow.tests.test_tokens.PublicationTokensTestCase)
    (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResource
    method), 113
method), 236
test_create_token()                        (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
    (tardis.tardis_portal.tests.test_ds_check())
    method), 256
method), 95
test_create_user Automatically_generate_api_key() (tardis.apps.hsm.tests.test_api.HsmAppApiTestCase)
    (tardis.tardis_portal.tests.models.test_generate_api_key.ApiKeyTestCase)
    method), 243
method), 253
test_credential_generation()               test_edit_dataset_params()
    (tardis.apps.push_to.tests.test_models.ModelsTestCase)
    (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase)

```

```

        method), 253
test_edit_experiment_params()           test_get_dir_tuples()
        (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase
        method), 253
test_ensures_suitable_license()         test_get_experiment()
        (tardis.tardis_portal.tests.test_forms.RightsFormTestCase
        method), 251
test_existing_parameterset()           test_get_experiment_author()
        (tardis.tardis_portal.tests.test_parametersets.ParameterSetTestCase
        method), 254
test_experiment()                      test_get_facilities()
        (tardis.tardis_portal.tests.models.test_experiment.ExperimentTestCase
        method), 243
test_experiment_parameter_get()         test_get_facility_by_id()
        (tardis.tardis_portal.tests.test_parameters.ParametersTestCase
        method), 253
test_experiment_view()                 test_get_facility_by_manager_group_id()
        (tardis.tardis_portal.tests.views.test_template_contexts.ViewTestCase
        method), 246
test_facility_overview_datafile_list()  test_get_facility_by_name()
        (tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTestCase
        method), 251
test_facility_overview_experiments()    test_get_group_by_id()
        (tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTestCase
        method), 251
test_fetch_experiments_and_datasets()   test_get_group_by_name()
        (tardis.apps.publication_workflow.tests.test_form.PublicationFormTestCase
        method), 112
test_file_upload()                     test_get_group_list()
        (tardis.tardis_portal.tests.views.test_upload_views.UploadFormTestCase
        method), 246
test_finalize_publication()            test_get_instrument_by_id()
        (tardis.apps.publication_workflow.tests.test_revert_finalize.RevertFinalizeTestCase
        method), 113
test_get_allowed_signing_services()    test_get_instrument_by_name()
        (tardis.apps.push_to.tests.test_models.ModelsTestCase
        method), 119
test_get_child_dir_nodes()             test_get_receiving_box()
        (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetTestCase
        method), 237
test_get_child_dir_nodes_no_files_in_rooted_dir() test_get_root_dir_nodes()
        (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetTestCase
        method), 237
test_get_dataset_files()               test_get_schema_by_id()
        (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetTestCase
        method), 237
test_get_dataset_filter_instrument()   test_get_schema_by_namespace()
        (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetTestCase
        method), 237
test_get_dataset_no_instrument()       test_get_session_expiry()
        (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetTestCase
        method), 237
test_get_dataset_with_instrument()     test_get_session_expiry_expired_token()
        (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetTestCase
        method), 237

```

```
method), 256
test_get_session_expiry_near_expiry()
    (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
        test_link_parameter_type_extra()
            method), 256
method), 256
test_get_storage_box_attr_list_from_box_id()
    (tardis.tardis_portal.tests.api.test_storagebox_resources.StorageBoxAttributeTest)
        method), 239
method), 239
test_get_storage_box_by_id()
    (tardis.tardis_portal.tests.api.test_storagebox_resources.StorageBoxResourceTest)
        method), 239
method), 239
test_get_storage_box_option_by_id()
    (tardis.tardis_portal.tests.api.test_storagebox_resources.StorageBoxOptionResourceTest)
        method), 239
method), 239
test_get_storage_box_option_list_from_box_id()
    (tardis.tardis_portal.tests.views.test_auth_views.ManageAccountTest)
        (tardis.tardis_portal.tests.api.test_storagebox_resources.StorageBoxOptionResourceTest)
            method), 239
method), 239
test_get_user_by_id()
    (tardis.tardis_portal.tests.api.test_user_resource.UserResourceTest)
        method), 239
method), 239
test_get_user_by_username()
    (tardis.tardis_portal.tests.api.test_user_resource.UserResourceTest)
        method), 239
method), 239
test_get_user_list()
    (tardis.tardis_portal.tests.views.test_auth_views.UserGroupListTestCase)
        method), 245
method), 245
test_get_user_list()
    (tardis.tardis_portal.tests.views.test_auth_views.UserListTestCase)
        (tardis.tardis_portal.tests.test_parametersets.ParameterSetManager)
            method), 245
method), 245
test_getgroupbyid()
    (tardis.tardis_portal.tests.test_ldap.LDAPTest)
        method), 252
method), 252
test_getgroups()
    (tardis.tardis_portal.tests.test_ldap.LDAPTest)
        method), 252
method), 252
test_getgroupsforentity()
    (tardis.tardis_portal.tests.test_ldap.LDAPTest)
        method), 252
method), 252
test_getuserbyid()
    (tardis.tardis_portal.tests.test_ldap.LDAPTest)
        method), 252
method), 252
test_image_filename_parameters()
    (tardis.tardis_portal.tests.test_parameters.ParametersTest)
        method), 253
method), 253
test_init()
    (tardis.apps.openid_migration.tests.test_forms.OpenIDMigrationFormTestCase)
        method), 106
method), 106
test_instrument()
    (tardis.tardis_portal.tests.models.test_instrument.InstrumentTestCase)
        method), 244
method), 244
test_is_expired()
    (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
        method), 256
method), 256
test_keys_from_private_key_only()
    (tardis.apps.push_to.tests.test_models.ModelsTestCase)
        method), 120
method), 120
test_link_parameter_type()
    (tardis.tardis_portal.tests.test_parametersets.ParameterSetManager)
        method), 254
method), 254
test_list_endpoints()
    (tardis.tardis_portal.tests.api.test_list_api_endpoints.ListEndpoints)
        method), 238
method), 238
test_login()
    (tardis.tardis_portal.tests.tests.UserInterfaceTestCase)
        test_manage_account()
            (tardis.tardis_portal.tests.views.test_auth_views.ManageAccountTest)
                (tardis.tardis_portal.tests.api.test_storagebox_resources.StorageBoxOptionResourceTest)
                    method), 239
method), 239
test_migrate_accounts()
    (tardis.apps.openid_migration.tests.test_views.OpenIDMigration)
        method), 96
method), 96
test_mydata_view()
    (tardis.tardis_portal.tests.views.test_template_contexts.Experiment)
        method), 246
method), 246
test_new_parameterset()
    (tardis.tardis_portal.tests.test_parametersets.ParameterSetManager)
        method), 254
method), 254
test_online_check_unsupported_storage_class()
    (tardis.apps.hsm.tests.test_api.HsmAppApiTestCase)
        method), 95
method), 95
test_online_check_unverified_file()
    (tardis.apps.hsm.tests.test_api.HsmAppApiTestCase)
        method), 95
method), 95
test_online_check_valid_storage_class()
    (tardis.apps.hsm.tests.test_api.HsmAppApiTestCase)
        method), 95
method), 95
test_online_check_with_bad_password()
    (tardis.apps.hsm.tests.test_api.HsmAppApiTestCase)
        method), 95
method), 95
test_online_count()
    (tardis.apps.hsm.tests.test_api.HsmAppApiTestCase)
        method), 95
method), 95
test_OpenidUserMigration_creation()
    (tardis.apps.openid_migration.tests.test_models.ModelTestCase)
        method), 106
method), 106
test_parameter()
    (tardis.tardis_portal.tests.models.test_parameter.Parameter)
        method), 244
method), 244
test_parameterset_as_string()
    (tardis.tardis_portal.tests.test_parametersets.ParameterSetManager)
        method), 254
method), 254
```

```

test_permissions_checks() (tardis.apps.publication_workflow.tests.test_my_publications.MyPublicationsTest)
    (tardis.tardis_portal.tests.test_parameters.ParametersTestCase.method), 112
    method), 253
test_pkey_to_credential() test_retrieve_released_pubs_list() (tardis.apps.publication_workflow.tests.test_my_publications.MyPublicationsTest)
    (tardis.apps.push_to.tests.test_models.ModelsTestCase.method), 112
    method), 120
test_post_dataset() test_retrieve_retracted_pubs_list() (tardis.apps.publication_workflow.tests.test_my_publications.MyPublicationsTest)
    (tardis.tardis_portal.tests.api.test_dataset_resource.DatasetResourceTest)
    method), 237
test_post_dataset_with_params() test_retrieve_scheduled_pubs_list() (tardis.apps.publication_workflow.tests.test_my_publications.MyPublicationsTest)
    (tardis.tardis_portal.tests.api.test_dataset_metadata_resource.DatasetParameterSetResourceTest)
    method), 236
test_post_experiment() test_revert_publication_to_draft() (tardis.apps.publication_workflow.tests.test_revert_finalize.RevertFinalizeTest)
    (tardis.tardis_portal.tests.api.test_experiment_resource.ExperimentResourceTest)
    method), 237
test_post_instrument() test_rights_require_valid_owner() (tardis.tardis_portal.tests.views.test_auth_views.RightsTestCase)
    (tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest)
    method), 238
test_post_single_file() test_root() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase)
    (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResourceTest)
    method), 236
method), 256
test.pretty_serializer() test_save_with_random_token() (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
    (tardis.tardis_portal.tests.api.test_serializer.SerializerTest)
    method), 239
method), 256
test_pub_rejected_template() test_email_template_email_token_gives_up() (tardis.tardis_portal.tests.test_email.EmailTemplateTestCase)
    (tardis.apps.publication_workflow.tests.test_email_email_templates.EmailTemplate)
    method), 112
method), 256
test_pub_released_template() test_email_template_email_token_gives_up() (tardis.tardis_portal.tests.test_email.EmailTemplateTestCase)
    (tardis.apps.publication_workflow.tests.test_email_email_templates.EmailTemplate)
    method), 112
method), 256
test_pub_reverted_to_draft_template() test_search() (tardis.tardis_portal.tests.test_ldap.LDAPTest)
    (tardis.apps.publication_workflow.tests.test_email_email_templates.EmailTemplate)
    method), 112
method), 256
test_recall() (tardis.apps.hsm.tests.test_api.HsmAppApiTestCase)
    (tardis.tardis_portal.tests.test_ldap.LDAPTest)
    method), 95
method), 252
test_recall_complete_email() test_sftp() (tardis.apps.sftp.tests.test_sftp.SFTPTest)
    (tardis.apps.hsm.tests.test_email_email_templates.HsmAppEmailTemplate)
    method), 95
method), 256
test_recall_failed_email() test_sftp_dynamic_docs_dataset() (tardis.apps.sftp.tests.test_sftp.SFTPTest)
    (tardis.apps.hsm.tests.test_email_email_templates.HsmAppEmailTemplate)
    method), 95
method), 256
test_rename_instrument() test_sftp_dynamic_docs_experiment() (tardis.apps.sftp.tests.test_sftp.SFTPTest)
    (tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest)
    method), 238
method), 256
test_resume_draft_publication() test_sftp_key_connect() (tardis.apps.sftp.tests.test_sftp.SFTPTest)
    (tardis.apps.publication_workflow.tests.test_form.PublicationFormTestCase)
    method), 112
method), 256
test_retrieve_access_list_tokens() test_shared_fs_many_files() (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResource)
    (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
    method), 256
method), 236
test_retrieve_access_list_tokens_json() test_shared_fs_single_file() (tardis.tardis_portal.tests.api.test_datafile_resource.DataFileResource)
    (tardis.tardis_portal.tests.test_tokens.PublicationTokenTestCase)
    method), 113
method), 256
test_retrieve_draft_pubs_list() test_shared_view() (tardis.tardis_portal.tests.views.test_template_contexts.ExperimentTemplateContext)
    (tardis.tardis_portal.tests.test_email.EmailTemplate)
    method), 256
method), 256

```

```

        method), 246
test_simple_search_authenticated_user()           test_user_not_authenticated()
                                                (tardis.tardis_portal.tests.graphql.test_user.userSignInTestCase
         (tardis.apps.search.tests.test_api.SimpleSearchTest   method), 241
         method), 135
test_simple_search_unauthenticated_user()          test_user_sign_in()
                                                (tardis.tardis_portal.tests.graphql.test_auth.userSignInTestCase
         (tardis.apps.search.tests.test_api.SimpleSearchTest   method), 240
         method), 135
test_stat_subprocess()                           TEST_UTILS_DIR (tardis.tardis_portal.tests.slacd.Slapd
                                                attribute), 247
(test.tardis.portal.tests.test_storage.ModelTestCase
method), 255
test_storageboxoption()                         test_wrong_size_verification()
                                                (tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase
         (tardis.tardis_portal.tests.test_storage.ModelTestCase
method), 255
method), 255
test_submit_form()                            testAccessWithoutReadPerms()
                                                (tardis.apps.related_info.tests.tests.TabTestCase
         (tardis.apps.publication_workflow.tests.test_form.PublicationFormTestCase
method), 131
method), 112
test_tar_experiment_download()                 testAccessWithReadPerms()
                                                (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase
         (tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase
method), 131
method), 255
test_token_delete()                           testCanCreate()
                                                (tardis.tardis_portal.tests.test_tokens.TokenTestCase
method), 249
method), 256
test_tz_aware_date_handling()                 testCanDelete()
                                                (tardis.apps.related_info.tests.tests.DeleteTestCase
         (tardis.tardis_portal.tests.test_parametersets.ParameterSetMutagedTestCase
method), 254
method), 254
test_tz_naive_date_handling()                 testCanGetInfoAsJSON()
                                                (tardis.tardis_portal.tests.test_iif.Level0TestCase
         (tardis.tardis_portal.tests.test_parametersets.ParameterSetMutagedTestCase
method), 254
method), 254
test_unauthorized_instrument_access_attempt() testCanGetInfoAsXML()
                                                (tardis.tardis_portal.tests.test_iif.Level0TestCase
         (tardis.tardis_portal.tests.api.test_instrument_resource.InstrumentResourceTest
method), 238
method), 238
test_unresolvable_link_parameter()            testCanGetJpegFormat()
                                                (tardis.tardis_portal.tests.test_iif.Level1TestCase
         (tardis.tardis_portal.tests.test_parametersets.ParameterSetMutagedTestCase
method), 254
method), 254
test_update_dataset_selection()               testCanGetOriginalImage()
                                                (tardis.tardis_portal.tests.test_iif.Level0TestCase
         (tardis.apps.publication_workflow.tests.test_form.PublicationMethodTestCase
method), 112
method), 112
test_update_extra_info()                     testCanGetRequiredFormats()
                                                (tardis.tardis_portal.tests.test_iif.Level2TestCase
         (tardis.apps.publication_workflow.tests.test_form.PublicationMethodTestCase
method), 112
method), 112
test_update_facility()                      testChangeUserPermissions()
                                                (tardis.tardis_portal.tests.test_authorisation.ObjectACLTTestCase
         (tardis.tardis_portal.tests.graphql.test_facilities.updateFacilityTest
method), 248
method), 241
test_upload_complete()                      testContext()
                                                (tardis.tardis_portal.tests.test_authorisation.ObjectACLTTestCase
         (tardis.tardis_portal.tests.views.test_upload_views.UploadTestCase
method), 248
method), 246
test_urls() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase
method), 254
method), 256
test_urls_with_some_content()               testDatasetFile()
                                                (tardis.tardis_portal.tests.test_download.DownloadTestCase
         (tardis.tardis_portal.tests.tests.UserInterfaceTestCase
method), 250
method), 256
test_user_authenticated()                  testDetectsBadInput()
                                                (tardis.apps.related_info.tests.tests.CreateTestCase
         (tardis.tardis_portal.tests.graphql.test_user.userSignInTestCase
method), 131
method), 241
method), 241

```

`(tardis.apps.related_info.tests.tests.UpdateTestCase testHandlesSingleEntry()
 method), 131` `(tardis.apps.related_info.tests.tests.ListTestCase
 method), 131`
`testDownload() (tardis.tardis_portal.tests.test_download.DownloadTestCase
 method), 250` `testIdentify() (tardis.apps.oaipmh.tests.provider.test_base.BaseProv
 method), 102`
`testDumpSchemas() (tardis.tardis_portal.tests.management.test_dumpschemas.DumpSchemasTestCase
 method), 242` `(tardis.apps.oaipmh.tests.provider.test_experiment.Abs
 method), 102`
`testExistsInOaipmh() (tardis.apps.related_info.tests.test_oaipmh.RifCSTestCase
 method), 103` `testIdentify() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase
 method), 130`
`testGetGroupsForEntity() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase
 method), 251` `testImageCacheControl()`
`testGetRecord() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase
 method), 251` `(tardis.tardis_portal.tests.test_iiif.ExtraTestCases
 method), 251`
`testGetRecord() (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTestCase
 method), 103` `(tardis.tardis_portal.tests.test_iiif.ExtraTestCases
 method), 251`
`testGetRecord() (tardis.apps.oaipmh.tests.provider.test_experiment.RifCsExperimentProviderTestCase
 method), 103` `testInitialisation()`
`testGetRecord() (tardis.apps.oaipmh.tests.test_oai.EndpointTestCase
 method), 249` `(tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase
 method), 249`
`testGetRecordHandlesInvalidIdentifiers() (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTestCase
 method), 102` `testInitialisation()`
`(tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase
 method), 249` `testInitialisationNoProvider()`
`testGroupProvider() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase
 method), 249` `(tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase
 method), 254`
`testGroupSearch() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase
 method), 249` `testInteractive()`
`testHandleRegions() (tardis.tardis_portal.tests.test_iiif.Level1TestCase
 method), 252` `testList() (tardis.tardis_portal.tests.management.test_createuser.CreateUser
 method), 242`
`testHandleRotation() (tardis.tardis_portal.tests.test_iiif.Level1TestCase
 method), 252` `testListIdentifiers()`
`(tardis.tardis_portal.tests.test_iiif.Level1TestCase
 method), 252` `(tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase
 method), 102`
`testHandlesEmptySet() (tardis.apps.related_info.tests.tests.ListTestCase
 method), 131` `testListIdentifiers()`
`(tardis.apps.related_info.tests.tests.ListTestCase
 method), 131` `(tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperiment
 method), 102`
`testHandlesFound() (tardis.apps.related_info.tests.tests.GetTestCase
 method), 131` `testListIdentifiers()`
`(tardis.apps.related_info.tests.tests.GetTestCase
 method), 131` `(tardis.apps.oaipmh.tests.test_oai.EndpointTestCase
 method), 103`
`testHandleSizing() (tardis.tardis_portal.tests.test_iiif.Level1TestCase
 method), 252` `testListIdentifiersDoesNotHandleSets()`
`(tardis.tardis_portal.tests.test_iiif.Level1TestCase
 method), 252` `(tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperiment
 method), 103`
`testHandleSizing() (tardis.tardis_portal.tests.test_iiif.Level2TestCase
 method), 252` `testListMetadataFormats()`
`(tardis.tardis_portal.tests.test_iiif.Level2TestCase
 method), 252` `(tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase
 method), 102`
`testHandlesMultipleEntries() (tardis.apps.related_info.tests.tests.ListTestCase
 method), 131` `testListMetadataFormats()`
`(tardis.apps.related_info.tests.tests.ListTestCase
 method), 131` `(tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperiment
 method), 103`
`testHandlesNotFound() (tardis.apps.related_info.tests.tests.GetTestCase
 method), 131` `testListMetadataFormats()`
`(tardis.apps.related_info.tests.tests.GetTestCase
 method), 131` `(tardis.apps.oaipmh.tests.test_oai.EndpointTestCase
 method), 103`
`testListRecords()`

```

(tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase), 250
    method), 102
testListRecords ()                               testWriteAccess ()
    (tardis.apps.oaipmh.tests.provider.test_experiment.DcExperimentProviderTestCase
        method), 103
testListRecords ()                               timeout (tardis.apps.sftp.sftp.MyTSFTPRequestHandler
    (tardis.apps.oaipmh.tests.provider.test_experiment.RifCExperimentProviderTestCase
        method), 103
testListRecords ()                               attribute), 144
    (tardis.apps.oaipmh.tests.provider.test_experiment.RifCExperimentProviderTestCase
        method), 103
testListRecords ()                               attribute), 93
    timestamp (tardis.apps.push_to.models.Progress
        method), 103
testListSets () (tardis.apps.oaipmh.tests.provider.test_base.BaseExperimentProviderTestCase
    method), 102
testListSets () (tardis.apps.oaipmh.tests.provider.test_experiment.AlbumExperimentProviderTC
    method), 103
testLoadSchemas ()                               to_json () (tardis.apps.search.api.PrettyJSONSerializer
    (tardis.tardis_portal.tests.management.test_loadschemas.LoadSchemaTestCase
        method), 242
testLocalFile () (tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCases
    method), 255
testManageAuthMethods ()                         timestamp (in module
    (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase
        method), 248
testManageRifCsCheckContent ()                  todatetime () (in module
    (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase
        method), 254
testManageRifCsCreateAndRemove ()                Token (class in tardis.tardis_portal.models.token),
    (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase
        method), 254
testMustHaveWrite ()                            token (tardis.tardis_portal.graphql.user.ApiSignIn
    (tardis.apps.related_info.tests.tests.CreateTestCase
        method), 131
testMustHaveWrite ()                            attribute), 231
    (tardis.apps.related_info.tests.tests.DeleteTestCase
        method), 131
testMustHaveWrite ()                            token_set (tardis.tardis_portal.models.experiment.Experiment
    (tardis.apps.related_info.tests.tests.UpdateTestCase
        method), 131
testMustHaveWrite ()                            attribute), 203
    (tardis.apps.related_info.tests.tests.UpdateTestCase
        method), 131
testNoInput () (tardis.tardis_portal.tests.management.test_createuser.CreateUserTestCase
    method), 242
testOwnedExperiments ()                        TokenAuthMiddleware (class in
    (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase
        method), 248
testReadAccess () (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase
    method), 248
testRemove () (tardis.tardis_portal.tests.management.test_removeExperiment.RemoveExperimentTestCase
    method), 242
testSFTPDWithoutHostKey ()                     tardis.tardis_portal.auth.token_auth), 162
    (tardis.apps.sftp.tests.test_sftp.SFTPDManagementTestCase
        method), 140
testSimpleAuthenticate ()                      tardis.tardis_portal.auth.token_auth), 163
    (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase
        method), 248
testView () (tardis.tardis_portal.tests.test_download.DownloadTestCase
    method), 225

```

TYPE_CHOICES (*tardis.tardis_portal.models.storage.StorageBoxOption* attribute), 229
 type_order (*tardis.tardis_portal.models.storage.StorageBox* attribute), 228
 TYPE_UNKNOWN (*tardis.tardis_portal.models.storage.StorageBox* attribute), 225
 TYPES (*tardis.tardis_portal.models.storage.StorageBox* attribute), 225

U

UncachedTarStream (class *tardis.tardis_portal.download*), 279
 units (*tardis.tardis_portal.models.parameters.ParameterName* attribute), 222
 unpickled_value (*tardis.tardis_portal.models.storage.StorageBox* attribute), 230
 update_all_files () (tardis.apps.sftp.sftp.DynamicTree method), 144
 update_attribution_and_licensing () (in module tardis.apps.publication_workflow.views), 118
 update_data_selection () (in module tardis.apps.publication_workflow.views), 118
 update_datafile (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_datafileobject (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_datafileparameter (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_datafileparameterset (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_dataset (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_dataset_basic_info () (in module tardis.apps.publication_workflow.views), 118
 update_dataset_files () (tardis.apps.sftp.sftp.DynamicTree method), 144
 update_datasetparameter (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_datasetparameterset (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_datasets () (tardis.apps.sftp.sftp.DynamicTree method), 144
 update_detail () (tardis.tardis_portal.api.ACIAuthorization method), 269

experiment (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_experimentauthor (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 175
 update_experimentparameter (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_experimentparameterset (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_experiments () (tardis.apps.sftp.sftp.DynamicTree method), 144
 update_extra_info () (in module tardis.apps.publication_workflow.views), 118
 update_facility (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_group (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_instrument (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_mimetype () (tardis.tardis_portal.models.datafile.DataFile method), 193
 update_nothing () (tardis.apps.sftp.sftp.DynamicTree method), 144
 update_parametername (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_schema (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_storagebox (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_storageboxattribute (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_storageboxoption (tardis.tardis_portal.graphql.schema.tardisMutation attribute), 176
 update_time (tardis.tardis_portal.models.experiment.Experiment attribute), 203

UpdateDataFile (class tardis.tardis_portal.graphql.datafile), 164
 UpdateDataFileObject (class tardis.tardis_portal.graphql.datafileobject), 164
 UpdateDatafileParameter (class in

```

    tardis.tardis_portal.graphql.parameters),
    174
UpdateDatafileParameterSet (class
    tardis.tardis_portal.graphql.parameters),
    174
UpdateDataset (class
    tardis.tardis_portal.graphql.dataset), 165
UpdateDatasetParameter (class
    tardis.tardis_portal.graphql.parameters),
    174
UpdateDatasetParameterSet (class
    tardis.tardis_portal.graphql.parameters),
    174
UpdateExperiment (class
    tardis.tardis_portal.graphql.experiment),
    167
UpdateExperimentAuthor (class
    tardis.tardis_portal.graphql.experiment),
    167
UpdateExperimentParameter (class
    tardis.tardis_portal.graphql.parameters),
    174
UpdateExperimentParameterSet (class
    tardis.tardis_portal.graphql.parameters), 174
UpdateFacility (class
    tardis.tardis_portal.graphql.facility), 168
updateFacilityTestCase (class
    tardis.tardis_portal.tests.graphql.test_facilities),
    241
UpdateGroup (class
    tardis.tardis_portal.graphql.group), 168
UpdateInstrument (class
    tardis.tardis_portal.graphql.instrument),
    169
UpdateParameterName (class
    tardis.tardis_portal.graphql.parameters),
    174
UpdateSchema (class
    tardis.tardis_portal.graphql.parameters),
    174
UpdateStorageBox (class
    tardis.tardis_portal.graphql.storagebox),
    177
UpdateStorageBoxAttribute (class
    tardis.tardis_portal.graphql.storageboxattribute),
    178
UpdateStorageBoxOption (class
    tardis.tardis_portal.graphql.storageboxoption),
    179
UpdateTestCase (class
    tardis.apps.related_info.tests.tests), 131
upload_auth () (in module
    tardis.tardis_portal.auth.decorators), 160
upload_complete () (in module
    tardis.tardis_portal.views.upload), 262
in upload_datafile (tardis.tardis_portal.graphql.schema.tardisMutation
    attribute), 176
    UPLOAD_METHOD (in module
        tardis.default_settings.uploads), 156
in UploadDataFile (class
    tardis.tardis_portal.graphql.datafile), 164
in UploadTestCase (class
    tardis.tardis_portal.tests.views.test_upload_views),
    246
in uri (tardis.tardis_portal.models.datafile.DataFileObject
    attribute), 196
url (tardis.tardis_portal.models.experiment.Experiment
    attribute), 204
url (tardis.tardis_portal.models.experiment.ExperimentAuthor
    attribute), 204
in url (tardis.tardis_portal.models.license.License
    attribute), 209
URL (tardis.tardis_portal.models.parameters.ParameterName
    attribute), 220
urls (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase
    attribute), 249
urls (tardis.tardis_portal.tests.test_tokens.TokenTestCase
    attribute), 256
in USE_FILTERS (in module
    tardis.default_settings.filters), 154
in use_rapid_connect () (in module
    tardis.tardis_portal.views.pages), 261
user (tardis.apps.eventlog.models.Log attribute), 93
in user (tardis.apps.push_to.models.Credential attribute),
    121
in user (tardis.apps.push_to.models.Request attribute),
    128
user (tardis.apps.sftp.models.SFTPPublicKey attribute),
    143
user (tardis.tardis_portal.graphql.schema.tardisQuery
    attribute), 177
in user (tardis.tardis_portal.graphql.user.ApiSignIn
    attribute), 179
user (tardis.tardis_portal.graphql.user.UserSignIn
    attribute), 179
user (tardis.tardis_portal.models.access_control.GroupAdmin
    attribute), 186
in user (tardis.tardis_portal.models.access_control.UserProfile
    attribute), 190
user (tardis.tardis_portal.models.token.Token
    attribute), 231
user_acls () (tardis.tardis_portal.managers.ExperimentManager
    method), 288
in user_details_processor () (in module
    tardis.tardis_portal.context_processors),
    278
user_guide () (in module
    tardis.tardis_portal.views.pages), 262

```

user_id (*tardis.apps.eventlog.models.Log* attribute), 93
 user_id (*tardis.apps.push_to.models.Credential* attribute), 121
 user_id (*tardis.apps.push_to.models.Request* attribute), 128
 user_id (*tardis.apps.sftp.models.SFTPPublicKey* attribute), 143
 user_id (*tardis.tardis_portal.models.access_control.GroupAdmin* attribute), 186
 user_id (*tardis.tardis_portal.models.access_control.UserProfile* attribute), 190
 user_id (*tardis.tardis_portal.models.token.Token* attribute), 231
 USER_MENU_MODIFIERS (in module *tardis.default_settings.apps*), 151
 user_menu_processor() (in module *tardis.tardis_portal.context_processors*), 278
 user_migration (in module *tardis.apps.openid_migration.models*.*OpenidACLMigration* attribute), 108
 user_migration_id (in module *tardis.apps.openid_migration.models*.*OpenidACLMigration* attribute), 109
 user_migration_obj() (in module *tardis.apps.openid_migration.models*.*OpenidACLMigration* method), 109
 user_owned_groups() (in module *tardis.tardis_portal.managers.ExperimentManager* method), 288
 user_sign_in (*tardis.tardis_portal.graphql.schema.tardisMutation*.*mutation_remote_path* () attribute), 176
 UserAuthentication (class in *tardis.tardis_portal.models.access_control*), 188
 UserAuthentication.DoesNotExist, 188
 UserAuthentication.MultipleObjectsReturned, 188
 userauthentication_set (in module *tardis.tardis_portal.models.access_control.UserProfile*.*UserAuthentication* attribute), 190
 UserAuthenticationAdmin (class in *tardis.tardis_portal.admin*), 268
 UserGroupListsTestCase (class in *tardis.tardis_portal.tests.views.test_auth_views*), 244
 UserInterfaceTestCase (class in *tardis.tardis_portal.tests.tests*), 256
 UserListTestCase (class in *tardis.tardis_portal.tests.views.test_auth_views*), 245
 username (in module *tardis.tardis_portal.models.access_control.UserAuthentication*.*UserAuthentication* attribute), 189
 UserProfile (class in *tardis.tardis_portal.models.access_control*.*UserAuthentication* attribute), 189
 userProfile (in module *tardis.tardis_portal.models.access_control.UserAuthentication*.*UserAuthentication* attribute), 189
 userProfile_provider (class in *tardis.tardis_portal.auth.interfaces*), 161
 userProfileResource (class in *tardis.tardis_portal.api*), 277
 UserResource.Meta (class in *tardis.tardis_portal.api*), 278
 UserResourceTest (class in *tardis.tardis_portal.tests.api.test_user_resource*), 239
 users() (in module *tardis.tardis_portal.managers.ExperimentManager* method), 288
 UserSignIn (class in *tardis.tardis_portal.tests.graphql.test_auth*), 240
 userSignInTestCase (class in *tardis.tardis_portal.tests.graphql.test_user*), 179
 userSignInTestCase (class in *tardis.tardis_portal.tests.graphql.test_user*), 241
 UserType (class in *tardis.tardis_portal.graphql.user*), 179
 V
 validation (in module *tardis.apps.sftp.api.SFTPPublicKeyAppResource*.*Meta* attribute), 141
 validation_error() (in module *tardis.apps.publication_workflow.views*), 118
 value (in module *tardis.tardis_portal.models.storage.StorageBoxAttribute* attribute), 229
 value (in module *tardis.tardis_portal.models.storage.StorageBoxOption* attribute), 230
 value_from_datadict () (in module *tardis.tardis_portal.widgets.CommaSeparatedInput* method), 291
 value_to_string () (in module *tardis.apps.eventlog.fields.JSONField* method), 91
 value_type (in module *tardis.tardis_portal.models.storage.StorageBoxOption* attribute), 230
 verbose_name (in module *tardis.analytics.apps.AnalyticsConfig* attribute), 88
 attribute), 107

verbose_name (*tardis.apps.publication_workflow.apps.PublicationWorkflowConfig.portal.admin.StorageBoxForm.Meta attribute*), 113
 verbose_name (*tardis.apps.push_to.apps.PushToConfig widgets* (*tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta attribute*), 268
 verbose_name (*tardis.apps.search.apps.SearchConfig widgets* (*tardis.tardis_portal.forms.RightsForm.Meta attribute*), 285
 verbose_name (*tardis.apps.sftp.apps.SFTPConfig attribute*), 142
 verbose_name (*tardis.apps.social_auth.apps.SocialAuthConfig permissions_required()* (in module *tardis.tardis_portal.auth.decorators*), 160
 verified (*tardis.tardis_portal.models.datafile.DataFile write_template_to_dir()* (attribute), 193
 verified (*tardis.tardis_portal.models.datafile.DataFileObject static method*), 291
 verify () (*tardis.tardis_portal.models.datafile.DataFile write_template_to_file()* (method), 291
 verify () (*tardis.tardis_portal.models.datafile.DataFileObject write_xml_to_file()* (method), 291
 verify_file () (*tardis.tardis_portal.api.DataFileResource static method*), 292
 verify_redirect () (in module *tardis.apps.push_to.views*), 130
 verify_remote_access () (*tardis.apps.push_to.models.Credential method*), 121
 verify_token (*tardis.tardis_portal.graphql.schema.tardisMutation method*), 104
 version (*tardis.tardis_portal.models.datafile.DataFile attribute*), 193
 view_datafile () (in module *tardis.tardis_portal.download*), 280
 view_functions (*tardis.tardis_portal.shortcuts.RestfulExperimentPathFinderSet attribute*), 289
 view_url (*tardis.tardis_portal.models.datafile.DataFile attribute*), 194
 ViewTemplateContextsTest (class in *tardis.tardis_portal.tests.views.test_template_contexts*), 246
W
 wait () (*tardis.tardis_portal.tests.slacd.Slapd method*), 248
 which () (in module *tardis.tardis_portal.tests.slacd*), 248
 widget (*tardis.tardis_portal.forms.StaticField attribute*), 285
 widgets (*tardis.apps.push_to.models.CredentialForm.Meta attribute*), 122
 widgets (*tardis.tardis_portal.admin.DatafileAdminForm.Meta attribute*), 264
 widgets (*tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta attribute*), 264
 widgets (*tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta attribute*), 267
X
 XMLWriter (class in *tardis.tardis_portal.xmlwriter*), 291