

---

# **Kuha2 Documentation**

*Release 0.1.0*

**Toni Sissala**

**Apr 11, 2019**



---

## Contents:

---

<b>1</b>	<b>Links</b>	<b>3</b>
<b>2</b>	<b>License</b>	<b>5</b>
<b>3</b>	<b>Installation</b>	<b>7</b>
3.1	User guide . . . . .	7
3.2	Installation . . . . .	10
3.3	Kuha Document Store . . . . .	14
3.4	Kuha OAI-PMH Repo Handler . . . . .	25
3.5	Kuha OSMH Repo Handler . . . . .	28
3.6	Kuha Client . . . . .	31
3.7	Kuha Common . . . . .	32
3.8	Developer Documentation . . . . .	33
3.9	Versions . . . . .	128
<b>4</b>	<b>Indices and tables</b>	<b>141</b>
	<b>Python Module Index</b>	<b>143</b>
	<b>HTTP Routing Table</b>	<b>145</b>



Kuha2 is a metadata server that provides descriptive social science research metadata for harvesting via multiple protocols and a growing variety of metadata standards. The software is a collection of applications and consists of three server applications, a client application and a database.

The development was initiated by [CESSDA SaW -project](#), but will continue as an Open Source project lead by [FSD](#).



Source code repositories and issue trackers are hosted in Bitbucket.

- [Kuha Document Store Latest release: 0.7.1](#)
- [Kuha OAI-PMH Repo Handler Latest release: 0.6.0](#)
- [Kuha OSMH Repo Handler Latest release: 0.3.0](#)
- [Kuha Client Latest release: 0.5.1](#)
- [Kuha Common Latest release: 0.9.1](#)
- [Kuha Documentation](#)

Documentation for Kuha2 applications is hosted at ReadTheDocs.

- [Kuha Document Store](#)
- [Kuha OAI-PMH Repo Handler](#)
- [Kuha OSMH Repo Handler](#)
- [Kuha Client](#)
- [Kuha Common](#)





## CHAPTER 2

---

### License

---

Kuha2 software components are available under the EUPL.



The supported operating system is Ubuntu 16.04. Every application is written in Python and requires Python 3.5. It is recommended to run the applications in a python virtual environment.

Each software component needs to be installed separately. Refer to the *Installation* chapter.

### 3.1 User guide

---

**Note:** This chapter is a work in progress.

---

#### 3.1.1 Architecture

In a typical usage scenario, the repository owner submits DDI files to the Document Store using Kuha Client. Document Store then stores the metadata into a database. Repository handlers implement different harvesting protocols such as OAI-PMH and query the Document Store accordingly. Only repository handlers should be exposed to external use (that is, to harvesters). If access control or traffic shaping is required, Kuha2 can be deployed behind an API gateway or some other proxy.

Kuha2 components communicate to each other through RESTful APIs and the use of the Document Store is not restricted to DDI. It is possible to bypass Kuha Client and use the Document Store API directly to submit metadata to the Store.

#### 3.1.2 Getting started

Once the *installation* is complete, you may wish to populate Document Store with some example data in order to see how the software works. This guide will demonstrate how to populate Document Store with example data. The guide assumes all Kuha2 software components are installed on localhost and using default configuration values.

Also refer to [OAI-PMH](#) documentation and [OSMH](#) documentation for information about the protocols.

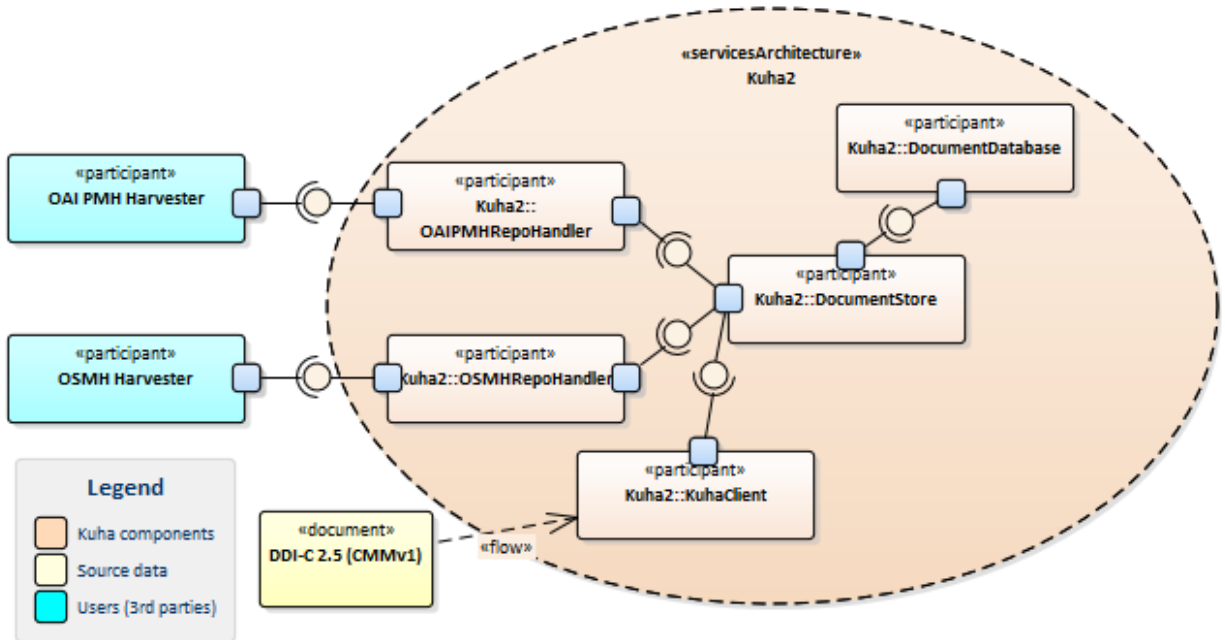


Fig. 3.1: Fig. Kuha2 service architecture diagram.

**Note:** The commands in this guide may change in future versions of Kuha Client. Refer to *Kuha Client documentation* for current commands and configuration parameters.

### Importing metadata

Import Study in DDI 1.2.2. to create a single Study with study number `study_1` and a single StudyGroup with identifier `serie_1`. The Study and StudyGroup contain some localized content marked with `xml:lang` attributes.

Import the XML metadata to Document Store

```
python -m kuha_client.kuha_import --document-store-url=http://localhost:6001/v0 --
↪source-file-type=ddi_122_nesstar ddi122_minimal_study.xml
```

The metadata is now available via OAI-PMH.

```
curl "http://localhost:6003/v0/oai?verb=GetRecord&metadataPrefix=ddi_c&
↪identifier=study_1"
```

And OSMH.

```
curl "http://localhost:6002/v0/GetRecord/Study/study_1"
curl "http://localhost:6002/v0/GetRecord/StudyGroup/serie_1"
```

Study in DDI 2.5. is a similar example, but serialized in DDI 2.5. It creates a single Study with study number `study_2` and a single StudyGroup with identifier `serie_2`. The Study and StudyGroup also contain some localized content.

Import the XML metadata to Document Store.

```
python -m kuha_client.kuha_import --document-store-url=http://localhost:6001/v0 --
↳source-file-type=ddi_c ddi25_minimal_study.xml
```

The metadata is now available via OAI-PMH.

```
curl "http://localhost:6003/v0/oai?verb=GetRecord&metadataPrefix=ddi_c&
↳identifier=study_2"
```

And OSMH.

```
curl "http://localhost:6002/v0/GetRecord/Study/study_2"
curl "http://localht:6002/v0/GetRecord/StudyGroup/serie_2"
```

Variables in DDI 2.5. contains a Study with three Variables and Questions.

Import the metadata.

```
python -m kuha_client.kuha_import --document-store-url=http://localhost:6001/v0 --
↳source-file-type=ddi_c ddi25_minimal_variables.xml
```

See it in OAI-PMH.

```
curl "http://localhost:6003/v0/oai?verb=GetRecord&metadataPrefix=ddi_c&
↳identifier=study_2"
```

And in OSMH.

```
curl "http://localhost:6002/v0/GetRecord/Study/study_3"
curl "http://localht:6002/v0/GetRecord/Variable/study_3:VAR_1"
curl "http://localht:6002/v0/GetRecord/Variable/study_3:VAR_2"
curl "http://localht:6002/v0/GetRecord/Variable/study_3:VAR_3"
curl "http://localhost:6002/v0/GetRecord/Question/study_3:QUESTION_1"
curl "http://localhost:6002/v0/GetRecord/Question/study_3:QUESTION_2"
curl "http://localhost:6002/v0/GetRecord/Question/study_3:QUESTION_3"
```

## Updating metadata

To keep the Document Store up-to-date with your DDI metadata, Kuha Client provides a `kuha_upsert` -module. The use is similar to the `kuha_import` module, except that `upsert` provides an optional command line parameter which instructs the client to remove records that are not found in current run. This is used in batch operations, when you wish to sync a directory full of DDI-files to Document Store.

Updated Study in DDI 2.5. has the same study number `study_2`, so it will update the already imported study. This file contains a new distributor (`distrbrt`-element) and has removed the elements referring to secondary study titles (`partitl`-elements).

To update the Study to Document Store.

```
python -m kuha_client.kuha_upsert --document-store-url=http://localhost:6001/v0 --
↳source-file-type=ddi_c ddi25_minimal_study_updated.xml
```

See the updated study in OAI-PMH.

```
curl "http://localhost:6003/v0/oai?verb=GetRecord&metadataPrefix=ddi_c&
↳identifier=study_2"
```

And OSMH.

```
curl "http://localhost:6002/v0/GetRecord/Study/study_2"
```

If you run the upsert command with the command line option `--remove-absent`, the other documents imported earlier will be removed, since they are not found from the DDI file.

First assure that the documents imported earlier are still served via OAI-PMH

```
curl "http://localhost:6003/v0/oai?verb=ListIdentifiers&metadataPrefix=ddi_c"
```

Now run the upsert command with `--remove-absent`.

```
python -m kuha_client.kuha_upsert --remove-absent --document-store-url=http://  
→localhost:6001/v0 --source-file-type=ddi_c ddi25_minimal_study_updated.xml
```

Then look at ListIdentifiers again.

```
curl "http://localhost:6003/v0/oai?verb=ListIdentifiers&metadataPrefix=ddi_c"
```

ListIdentifiers should only return the a single `study_2`.

### Deleting all records

To remove the example data from Document Store you can issue a delete command with Kuha Client, which will delete all documents from all collections.

```
python -m kuha_client.kuha_delete --document-store-url=http://localhost:6001/v0 ALL_  
→ALL
```

## 3.2 Installation

This chapter describes the installation of each application.

### 3.2.1 Installing Kuha Document Store

This guide will provide step-by-step instructions in installing Kuha Document Store and MongoDB database. The supported operating system is Ubuntu 16.04, but other modern Linux variants may be used.

In this guide the installation of the database is done on a separate server, which will also be Ubuntu 16.04. However, Document Store and MongoDB may be installed on the same server.

#### Install MongoDB

---

**Note:** These actions should be done on the MongoDB server.

---

It is recommended to use the latest version of MongoDB which can be obtained from MongoDB's own repository. Refer to [MongoDB manual](#) on how to install MongoDB to Ubuntu 16.04. At the time of writing the installation was done as follows.

1. Obtain MongoDB public key.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv_  
→0C49F3730359A14518585931BC711F9BA15703C6
```

## 2. Add MongoDB source.

```
echo "deb [ arch=amd64,arm64 ] http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/
↳3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list
```

## 3. Update indexes and install.

```
sudo apt-get update && sudo apt-get install -y mongodb-org
```

## 4. Configure MongoDB to accept incoming connections. Use IP of your MongoDB server in <mongodb-ip>.

```
sudo sed -i 's/ bindIp: 127.0.0.1/ bindIp: <mongodb-ip>/' /etc/mongod.conf
```

## 5. Start MongoDB.

```
sudo service mongod start
```

Now MongoDB is running and accepting incoming connections. Note that the MongoDB instance is now accepting incoming connections without authentication. Authentication will be enabled later in this guide. For up-to-date information on configuration and security see MongoDB manual.

The next step is to create administrator credentials and setup databases, collections and database users. This can be done with a script bundled with Kuha Document Store.

First it is required to install the Document Store package.

### Install Document Store

---

**Note:** These actions should be done on the Document Store server.

---

## 1. Create directory for document store and Python virtualenv.

```
mkdir kuha2
```

## 2. Clone package to subdirectory.

```
hg clone -r releases https://bitbucket.org/tietoarkisto/kuha_document_store kuha2/
↳kuha_document_store
```

## 3. Install Python virtual environment.

```
sudo apt install -y python3-venv
```

## 4. Make installation script executable.

```
chmod +x ./kuha2/kuha_document_store/scripts/install_kuha_document_store_virtualenv.sh
```

## 5. Install Kuha Document Store to virtual environment.

```
./kuha2/kuha_document_store/scripts/install_kuha_document_store_virtualenv.sh
```

### Setup MongoDB for Document Store

Document store provides a script which will help setup MongoDB. The script will prompt for administrator credentials, which will be created. Give hostname/IP of your MongoDB server as command line parameter.

The script will create needed collections and database users. It will also setup indexes for the collections to speed up database queries.

**Note:** You may wish to provide DB credentials for editor and reader. Give parameter `--help` to see how.

---

1. Make the setup script executable.

```
chmod +x ./kuha2/kuha_document_store/scripts/setup_mongodb.sh
```

2. Run the MongoDB setup script. **Replace <mongodb-ip> with the IP of your MongoDB server.**

```
./kuha2/kuha_document_store/scripts/setup_mongodb.sh --database-host=<mongodb-ip>
```

Now the database is ready to be used with Document Store. Care should be taken to secure the MongoDB instance. For Kuha2 the only IP that needs access to the database is Kuha Document Store's IP. Authentication should also be enabled.

### Enable authentication to MongoDB

**Note:** These actions should be done on the MongoDB server.

---

1. Enable authentication.

```
sudo sed -i 's/#security:/security:\n  authorization: enabled/' /etc/mongod.conf
```

2. Restart MongoDB.

```
sudo service mongod restart
```

### Running the Document Store

**Note:** This action should be done on the Document Store server.

---

Start serving Document Store. **Replace <mongodb-ip> with the IP of your MongoDB server.**

```
./kuha2/kuha_document_store/scripts/run_kuha_document_store.sh --database-host=  
↪<mongodb-ip>
```

## 3.2.2 Installing Kuha OSMH Repo Handler

1. Create directory for OSMH Repo Handler and Python virtualenv.

```
mkdir kuha2
```

2. Clone package to subdirectory.

```
hg clone -r releases https://bitbucket.org/tietoarkisto/kuha_osmh_repo_handler kuha2/  
↪kuha_osmh_repo_handler
```

3. Install Python virtual environment.

```
sudo apt install -y python3-venv
```

4. Make install script executable.



```
chmod +x ./kuha2/kuha_osmh_repo_handler/scripts/install_kuha_osmh_repo_handler_
↳virtualenv.sh
```

#### 5. Install Kuha OSMH Repo Handler to virtual environment.

```
./kuha2/kuha_osmh_repo_handler/scripts/install_kuha_osmh_repo_handler_virtualenv.sh
```

To run Kuha OSMH Repo Handler you need access to Kuha Document Store. First you will need to make run script executable.

```
chmod +x ./kuha2/kuha_osmh_repo_handler/scripts/run_kuha_osmh_repo_handler.sh
```

Run by calling the script. **Replace <document-store-url> with the URL to the Document Store.**

```
./kuha2/kuha_osmh_repo_handler/scripts/run_kuha_osmh_repo_handler.sh --document-store-
↳url=<document-store-url>
```

### 3.2.3 Installing Kuha OAI-PMH Repo Handler

#### 1. Create directory for OAI-PMH Repo Handler and Python virtualenv.

```
mkdir kuha2
```

#### 2. Clone package to subdirectory.

```
hg clone -r releases https://bitbucket.org/tietoarkisto/kuha_oai_pmh_repo_handler_
↳kuha2/kuha_oai_pmh_repo_handler
```

#### 3. Install Python virtual environment.

```
sudo apt install -y python3-venv
```

#### 4. Make install script executable.

```
chmod +x ./kuha2/kuha_oai_pmh_repo_handler/scripts/install_kuha_oai_pmh_repo_handler_
↳virtualenv.sh
```

#### 5. Install Kuha OAI-PMH Repo Handler to virtual environment.

```
./kuha2/kuha_oai_pmh_repo_handler/scripts/install_kuha_oai_pmh_repo_handler_
↳virtualenv.sh
```

To run Kuha OAI-PMH Repo Handler you need access to Kuha Document Store. First make the run script executable.

```
chmod +x ./kuha2/kuha_oai_pmh_repo_handler/scripts/run_kuha_oai_pmh_repo_handler.sh
```

Run by calling the script. **Replace <document-store-url> with the URL to the Document Store.** You also need to specify few configuration values for OAI-PMH: *base\_url* and *admin\_email*.

```
./kuha2/kuha_oai_pmh_repo_handler/scripts/run_kuha_oai_pmh_repo_handler.sh --document-
↳store-url=<document-store-url> --oai-pmh-base-url=<base_url> --oai-pmh-admin-email=
↳<email>
```

## 3.2.4 Installing Kuha Client

1. Create directory for Kuha Client and Python virtualenv.

```
mkdir kuha2
```

2. Clone package to subdirectory.

```
hg clone -r releases https://bitbucket.org/tietoarkisto/kuha_client kuha2/kuha_client
```

3. Install Python virtual environment.

```
sudo apt install -y python3-venv
```

4. Install Kuha Client to virtual environment

```
cd kuha2
python3 -m venv kuha_client-env
source ./kuha_client-env/bin/activate
cd kuha_client
pip install -r requirements.txt
python setup.py install
```

Batch import files to Document Store. Python virtual environment must be active.

```
python -m kuha_client.kuha_import --document-store-url=<document-store-url> --file-
↳log-path=file_log /path/to/directory
```

Batch upsert records to Document Store and remove records not in current batch. Python virtual environment must be active.

```
python -m kuha_client.kuha_upsert --document-store-url=<document-store-url> --remove-
↳absent --file-log-path=file_log /path/to/directory
```

## 3.3 Kuha Document Store

Kuha Document Store is a HTTP backend API written in Python for serving Document Store records to multiple repo handlers. The Document Store uses MongoDB as a persistent storage and provides multiple endpoints for managing the database documents.

Kuha Document Store is a part of Open Source software bundle Kuha2.

### 3.3.1 Features

#### Import records from DDI XML

Kuha Document Store provides an easy way to import multiple records all at once by simply submitting a DDI file to an import-endpoint. The Document Store imports all records found from the file and handles inserts and updates correctly.

#### REST API for full control

Kuha Document Store has a REST API that gives end users full control of the records stored in the Document Store. The REST API may be used to build functionality for specific needs, for example, to automatically update a record when a record is changed in a 3rd party storage system.

With the REST API, end users are not tied to using DDI, but may use arbitrary metadata formats and submit their records to Document Store using HTTP with JSON payload.

### Flexible query support

Kuha Document Store provides an endpoint for selectively querying stored records. The Query API is used by client applications which are a part of the Kuha2 software bundle.

## 3.3.2 Dependencies & requirements

Versions specified here are the ones the software has been developed with. Newer versions may be compatible.

- Python 3.5
- MongoDB 3.4 (License: GNU AGPL v3.0)
- **Recommended:** python3-venv 3.5.1

### Python packages

The following can be obtained from Python package index.

- motor (License: Apache License 2.0)
- pymongo (License: Apache License 2.0)
- tornado (License: Apache License 2.0)
- Cerberus (License: ICS)
- python-dateutil (License: Simplified BSD)

Kuha Common is a library used with Kuha2 software. It can be obtained from [https://bitbucket.org/tietoarkisto/kuha\\_common](https://bitbucket.org/tietoarkisto/kuha_common)

- kuha\_common (License: EUPL)

## 3.3.3 License

Kuha Document Store is available under the EUPL. See LICENSE.txt for the full license.

## 3.3.4 Configuration

The application can be configured with a configuration file, via command line arguments or by environment variables. All configuration options have default values. If a configuration option is specified in more than one place, then command line values override environment variables which override configuration file values which override defaults.

The following configuration options are available:

**-h, --help**

Show help message and exit.

**--print-configuration**

Print active configuration and exit.

**--document-store-port** <port>

Port of Kuha document store database. Defaults to "6001". May also be controlled by setting environment variable: KUHA\_DS\_PORT.

- document-store-api-version** <api\_version>  
Api version for document store. This gets prepended to the URL path. Defaults to v0. May also be controlled by setting environment variable: KUHA\_DS\_API\_VERSION.
- database-host** <database\_host>  
Host/IP of the Document Store database. Defaults to localhost. May also be controlled by setting environment variable: KUHA\_DS\_DBHOST
- database-port** <port>  
Port of the Document Store database. Defaults to 27017. May also be controlled by setting environment variable: KUHA\_DS\_DBPORT
- database-name** <name>  
Name of Document Store database. Defaults to kuha\_document\_store. May also be controlled by setting environment variable: KUHA\_DS\_DBMAME
- database-user-reader** <user>  
Username for database user having read-only rights. Defaults to reader. May also be controlled by setting environment variable: KUHA\_DS\_DBUSER\_READER
- database-pass-reader** <password>  
Password for database user having read-only rights. Defaults to reader. May also be controlled by setting environment variable: KUHA\_DS\_DBPASS\_READER
- database-user-editor** <user>  
Username for database user having editing rights. Defaults to editor. May also be controlled by setting environment variable: KUHA\_DS\_DBUSER\_EDITOR
- database-pass-editor** <password>  
Password for database user having editing rights. Defaults to editor. May also be controlled by setting environment variable: KUHA\_DS\_DBPASS\_EDITOR
- loglevel** <loglevel>  
Lowest logging level of log messages that get output. Valid values are logging levels supported by Python's `logging` [CRITICAL, ERROR, WARNING, INFO, DEBUG]. Defaults to INFO. May also be controlled by setting environment variable: KUHA\_LOGLEVEL
- logformat** <logformat>  
Logging format supported by `logging`. Defaults to `%(asctime)s %(levelname)s %(name)s : %(message)s` May also be controlled by setting environment variable: KUHA\_LOGFORMAT

### Configuration file

Args that start with '-' (eg. `--document-store-port`) can also be set in a config file. The configuration file lookup searches the file from current working directory and from the package directory. The name of the configuration file is `kuha_document_store.ini`.

---

**Note:** Invoke with `--help` to print out config file lookup paths.

---

### Environment variables

If the program will be run by using the scripts provided in `scripts` subdirectory, the runtime environment can be controlled via `scripts/runtime_env`, which will be created by copying from `scripts/runtime_env.dist` at installation time by `scripts/install_kuha_document_store_virtualenv.sh`.

### 3.3.5 Running the program

This guide will use convenience scripts from `scripts` subdirectory. It is assumed that the program was installed by using `scripts/install_kuha_document_store_virtualenv.sh`.

Run Document Store server:

```
./scripts/run_kuha_document_store.sh
```

The script will source `scripts/runtime_env` and activate the installed virtualenv. Finally it calls `kuha_ds_serve`, with given command line arguments.

### 3.3.6 HTTP API endpoints

Root for the requests is configurable and defaults to `localhost:6001/v0`. Every endpoint will return HTTP status code 500 on internal errors.

---

**Note:** Responses with multiple objects will be streamed one object at a time.

---

## REST API

Document Store REST API provides **CRUD** support to the underlying documents.

**GET** `/ (collection) /`

`document_id` Get object from `collection` with optional `document_id`. If `document_id` is not given, endpoint will return all objects in collection.

#### Parameters

- **collection** (*str*) – Document collection. One of *studies*, *variables*, *questions* or *study\_groups*.
- **document\_id** (*str*) – Optional document ID. 24-character hex string.

#### Status Codes

- 200 OK – Success
- 400 Bad Request – Invalid parameters
- 404 Not Found – Resource not found

**POST** `/studies`

Create a new object to studies-collection from JSON request body.

**Example request:**

```
POST /studies HTTP/1.1
Content-Type: application/json

{"study_number": "study_1"}
```

**Example response:**

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=UTF-8
```

```
{
  "result": "insert_successful",
  "error": null,
  "affected_resource": "5a82e76e6fb71d06fef00e69"
}
```

**Request Headers**

- Content-Type – application/json

**Request JSON Object**

- **study\_number** (*string*) – Required study number. Used as an identifier. Must be unique within collection.

**Response JSON Object**

- **result** (*string*) – Operation outcome.
- **error** (*string*) – Errors during operation.
- **affected\_resource** (*string*) – *document\_id* of the created object.

**Status Codes**

- 201 Created – Created successfully.
- 415 Unsupported Media Type – Invalid content type.
- 400 Bad Request – Invalid JSON, Validation failed, Duplicate unique value.

**POST /variables**

Create a new object to variables-collection from JSON request body.

**Example request:**

```
POST /variables HTTP/1.1
Content-Type: application/json

{
  "study_number": "study_1",
  "variable_name": "variable_1"
}
```

**Example response:**

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=UTF-8

{
  "result": "insert_successful",
  "error": null,
  "affected_resource": "5a82ecf16fb71d06fef00e6a"
}
```

**Request Headers**

- Content-Type – application/json

**Request JSON Object**

- **study\_number** (*string*) – Required study number. Used as an identifier combined with *variable\_name*. Their combination must be unique within collection.
- **variable\_name** (*string*) – Required variable name. Used as an identifier combined with *study\_number*. Their combination must be unique within collection.

#### Response JSON Object

- **result** (*string*) – Operation outcome.
- **error** (*string*) – Errors during operation.
- **affected\_resource** (*string*) – *document\_id* of the created object.

#### Status Codes

- 201 **Created** – Created successfully.
- 415 **Unsupported Media Type** – Invalid content type.
- 400 **Bad Request** – Invalid JSON, Validation failed, Duplicate unique value.

### POST /questions

Create a new object to questions-collection from JSON request body.

#### Example request:

```
POST /questions HTTP/1.1
Content-Type: application/json

{
  "study_number": "study_1",
  "question_identifier": "question_1"
}
```

#### Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=UTF-8

{
  "result": "insert_successful",
  "error": null,
  "affected_resource": "5a82ee1a6fb71d06fef00e6b"
}
```

#### Request Headers

- **Content-Type** – application/json

#### Request JSON Object

- **study\_number** (*string*) – Required study number. Used as an identifier combined with *question\_identifier*. Their combination must be unique within collection.
- **question\_identifier** (*string*) – Required variable name. Used as an identifier combined with *study\_number*. Their combination must be unique within collection.

#### Response JSON Object

- **result** (*string*) – Operation outcome.
- **error** (*string*) – Errors during operation.

- **affected\_resource** (*string*) – *document\_id* of the created object.

#### Status Codes

- 201 Created – Created successfully.
- 415 Unsupported Media Type – Invalid content type.
- 400 Bad Request – Invalid JSON, Validation failed, Duplicate unique value.

#### POST /study\_groups

Create a new object to study\_groups-collection from JSON request body.

##### Example request:

```
POST /study_groups HTTP/1.1
Content-Type: application/json

{
  "study_group_identifier": "study_group_1"
}
```

##### Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=UTF-8

{
  "result": "insert_successful",
  "error": null,
  "affected_resource": "5a82ee876fb71d06fef00e6c"
}
```

#### Request Headers

- Content-Type – application/json

#### Request JSON Object

- **study\_group\_identifier** (*string*) – Required. Used as an identifier and must be unique within collection.

#### Response JSON Object

- **result** (*string*) – Operation outcome.
- **error** (*string*) – Errors during operation.
- **affected\_resource** (*string*) – *document\_id* of the created object.

#### Status Codes

- 201 Created – Created successfully.
- 415 Unsupported Media Type – Invalid content type.
- 400 Bad Request – Invalid JSON, Validation failed, Duplicate unique value.

#### DELETE / (collection) /

*document\_id* Delete document or all documents within collection. If optional *document\_id* is left out, will delete all documents within collection.

#### Response JSON Object



- **result** (*string*) – Operation outcome.
- **error** (*string*) – Errors during operation.
- **affected\_resource** (*string*) – *document\_id* of the created object or number of deleted documents if *document\_id* is not given in request.

#### Status Codes

- **200 OK** – Delete successful.
- **404 Not Found** – Resource not found.

#### PUT / (collection) /

*document\_id* Replace document within collection. :see: [Documents](#) for information on payload.

**Note** Leave `_metadata` field out of the request, to let document store handle updated-timestamp automatically.

#### Request Headers

- **Content-Type** – application/json

#### Response JSON Object

- **result** (*string*) – Operation outcome.
- **error** (*string*) – Errors during operation.
- **affected\_resource** (*string*) – *document\_id* of the created object or number of deleted documents if *document\_id* is not given in request.

#### Status Codes

- **200 OK** – Replace successful
- **404 Not Found** – Resource not found
- **400 Bad Request** – Invalid JSON, Validation failed, Duplicate unique value.

## Import API

Documents may be imported to Document Store by using the import API. See [Importers](#) for more information on how documents are parsed.

#### POST /import/ (importer\_id) /

*collection* Import document using importer specified with *importer\_id*. Optional *collection* may be given to limit the import to a specific collection. If *collection* is not given the importer will import to collections that are applicable to the posted file.

**Note** Importer API may only be used to initially import documents to the database. After the initial import the documents may be updated via the REST API.

#### Parameters

- **importer\_id** (*str*) – Mandatory parameter to select the importer.
  - `ddi_31` imports DDI 3.1 file.
  - `ddi_c` imports DDI 2.5 file.
  - `ddi_122_nesstar` imports DDI 1.2.2. Nesstar file.
- **collection** (*str*) – Optional parameter to limit the import to specific collection. One of *studies*, *variables*, *questions* or *study\_groups*.

### Request Headers

- `Content-Type` – text/xml

**Request body** DDI file contents

### Response JSON Object

- **result** (*string*) – Operation result
- **imported\_docs** (*array*) – Imported document IDs
- **error** (*string*) – Errors found during import

### Status Codes

- 200 OK – Import successful
- 400 Bad Request – Empty or invalid request body
- 404 Not Found – Invalid importer id
- 415 Unsupported Media Type – Invalid content type

## Query API

Query documents or information on documents from collection.

**POST** `/query/` (*collection*)

Execute query against *collection* and return results in JSON.

### Request Headers

- `Content-Type` – application/json

### Query Parameters

- **query\_type** (*string*) – Optional query parameter to select the query type.
  - `select` is the default query type. It returns all documents found by filter.
  - `count` returns the number of documents which match the filter.
  - `distinct` return distinct results for certain field which match the filter.

### Request JSON Object

- **\_filter** (*object*) – Query filter. Used for all query types. Request may specify multiple filter conditions.

**Example request with multiple filter conditions:**

```
POST /query/variables HTTP/1.1
Content-Type: application/json

{
  "_filter": {
    "study_number": "study_1",
    "variable_name": "variable_1"
  }
}
```

### Request JSON Object

- **fields** (*array*) – Optional. Select returned fields. Used in select query type. `_id` will always be returned. If not set, full document will be returned.
- **skip** (*int*) – Optional. Skip documents from the beginning. Used in select query type.
- **limit** (*int*) – Optional. Limit the number of returned documents. Used in select query type.
- **sort\_by** (*string*) – Optional. Sort the queried documents by field. Used in select query type.
- **sort\_order** (*int*) – Optional. Sort order of the queried documents. Used in select query type.
  - 1: Ascending sort order.
  - -1: Descending sort order.
- **fieldname** (*string*) – Mandatory for distinct query type. Return distinct values for this field.

Result depends on the requested *query\_type*.

#### JSON response for select query-type

Results will be streamed one object at a time. The object is a document with requested *fields*.

#### JSON response for count query-type

##### Response JSON Object

- **count** (*int*) – Number of documents found with `_filter`.

#### JSON response for distinct query-type

If *fieldname* points to document's leaf node the response is in the following format.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8

{
  "<fieldname>": ["<list-of-distinct-values>"]
}
```

If *fieldname* points to document's branch node the response is in the following format.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8

{
  "<fieldname>": ["<list-of-distinct-objects>"]
}
```

#### Example requests and responses for distinct query-type

```
POST /query/studies?query_type=distinct HTTP/1.1
Content-Type: application/json

{
  "fieldname": "_metadata.updated"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8

{
  "_metadata.updated": [
    "2018-02-13T13:49:37Z",
    "2018-02-08T10:55:41Z"
  ]
}
```

```
POST /query/studies?query_type=distinct HTTP/1.1
Content-Type: application/json

{
  "fieldname": "_metadata"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8

{
  "_metadata": [
    {
      "updated": "2017-11-09T12:07:48Z",
      "cmm_type": "study",
      "created": "2017-11-09T11:06:03Z"
    },
    {
      "updated": "2017-11-09T11:37:16Z",
      "cmm_type": "study",
      "created": "2017-11-09T11:37:16Z"
    }
  ]
}
```

---

**Note:** Distinct queries for datetime-fields will not work as expected, due to different precision in MongoDB and Document Store JSON. MongoDB stores `datetimes` in millisecond's precision, while Document Store JSON supports second's precision.

---

#### Status Codes

- 200 OK – OK
- 400 Bad Request – Message body empty, invalid *query\_type*, invalid *query parameters* for query type.
- 415 Unsupported Media Type – Invalid Content-Type

### 3.3.7 Documents

Documents are objects stored in a collection. Documents support four different types of fields:

1. key-value pair:

```
{"study_number": "1200"}
```

## 2. contained key-value pairs:

```
{ "_metadata": {
  "updated": "2018-01-31T11:37:34Z",
  "cmm_type": "study",
  "created": "2018-01-31T11:37:27Z"
}}
```

## 3. localized contained key-value pairs:

```
{"study_titles": [
  {
    "language": "en",
    "study_title": "Study 1983"
  },
  {
    "language": "fi",
    "study_title": "Tutkimus 1983"
  }
]}
```

## 4. list of unique values:

```
{"study_numbers": ["1210", "3134", "1175", "2290", "2498"]}
```

### 3.3.8 Importers

There are importers for DDI3.1., DDI 2.5. and DDI 1.2.2., which can be used to initially import DDI-XML files to document store.

Importer tries to update documents if they are already found from the database. However it is not guaranteed to work properly in cases where an ID element for a field is not found from the DDI. Therefore it is best to use the importer only for initial import of records and afterwards use the REST API to update the documents.

Importer reads `xml:lang` attributes from the XML-elements to get the language of the element's content. If an element should have no `xml:lang` attribute, the language is read from the root XML-element's `xml:lang`. If the root element has no `xml:lang` attribute the content is assumed to be in english, and `en` is used for the language.

## 3.4 Kuha OAI-PMH Repo Handler

Kuha OAI-PMH Repo Handler is a HTTP API written in Python for serving Kuha Document Store records through OAI-PMH.

Kuha OAI-PMH Repo Handler is a part of Open Source software bundle Kuha2.

### 3.4.1 Features

OAI-PMH features:

- Selective harvesting with Sets & Datestamps.

- List request sequence with ResumptionTokens.
- OAI-Identifiers.

Supported metadata standards:

- DDI-C 2.5
- OAI-DC

### 3.4.2 Dependencies & requirements

Versions specified here are the ones the software has been developed with. Newer versions may be compatible.

- Python 3.5
- **Recommended:** python3-venv 3.5.1

#### Python packages

The following can be obtained from Python package index.

- tornado (License: Apache License 2.0)
- Genshi (License: BSD)

Kuha Common is a library used with Kuha2 software. It can be obtained from [https://bitbucket.org/tietoarkisto/kuha\\_common](https://bitbucket.org/tietoarkisto/kuha_common)

- kuha\_common (License: EUPL)

### 3.4.3 License

Kuha OAI-PMH Repo Handler is available under the EUPL. See LICENSE.txt for the full license.

### 3.4.4 Configuration

The application can be configured with a configuration file, via command line arguments or by environment variables. If a configuration option is specified in more than one place, then command line values override environment variables which override configuration file values which override defaults.

---

**Note:** Configuration options for `-oai-pmh-base-url` and `-oai-pmh-admin-email` are required.

---

Some of the configuration options configure the OAI-PMH repository. Refer to [OAI-PMH](#) protocol description for more information.

The following configuration options are available:

**-h, --help**

Show help message and exit.

**--print-configuration**

Print active configuration and exit.

**--port** <port>

Port for serving OAI-PMH Repo Handler. Defaults to 6003 May also be controlled by setting environment variable: `KUHA_OPRH_PORT`.

- 
- template-folder** <folder>  
 Absolute path to Genshi templates. Defaults to <package-installation-dir>/templates. May also be controlled by setting environment variable: KUHA\_OPRH\_TEMPLATES.
- oai-pmh-repo-name** <repo\_name>  
 OAI-PMH repository name. Defaults to `Kuha2 oai-pmh repository`. May also be controlled by setting environment variable: KUHA\_OPRH\_OP\_REPO\_NAME.
- oai-pmh-base-url** <base\_url>  
 OAI-PMH base url. **Required** configuration value. May also be controlled by setting environment variable: KUHA\_OPRH\_OP\_BASE\_URL.
- oai-pmh-namespace-identifier** <namespace\_id>  
 Namespace identifier to use with **OAI-Identifiers**. Set `None` to disable use of OAI-Identifiers. Defaults to `None`. May also be controlled by setting environment variable: KUHA\_OPRH\_OP\_NAMESPACE\_ID.
- oai-pmh-protocol-version** <version>  
 OAI-PMH protocol version. Note that currently only 2.0 is supported. Defaults to `2.0`. May also be controlled by setting environment variable: KUHA\_OPRH\_OP\_PROTO\_VERSION.
- oai-pmh-results-per-list** <results\_per\_list>  
 Set maximum number of results for each list response. Defaults to `500`. May also be controlled by setting environment variable: KUHA\_OPRH\_OP\_LIST\_SIZE.
- oai-pmh-admin-email** <email>  
 OAI-PMH administrator email address. **Required** configuration value. Repeat to give multiple addresses. May also be controlled by setting environment variable: KUHA\_OPRH\_OP\_EMAIL\_ADMIN.
- oai-pmh-api-version** <api\_version>  
 Api version for OAI-PMH Repo Handler. This gets prepended to the URL path. Defaults to `v0`. May also be controlled by setting environment variable: KUHA\_OPRH\_API\_VERSION.
- document-store-host** <host>  
 Host & scheme of Kuha Document Store. Defaults to `http://localhost`. May also be controlled by setting environment variable: KUHA\_DS\_HOST.
- document-store-port** <port>  
 Port of Kuha document store database. Defaults to `6001`. May also be controlled by setting environment variable: KUHA\_DS\_PORT.
- document-store-api-version** <api\_version>  
 Api version for document store. This gets prepended to the URL path. Defaults to `v0`. May also be controlled by setting environment variable: KUHA\_DS\_API\_VERSION.
- document-store-client-request-timeout** <timeout>  
 Request timeout (in seconds) for Document Store client. Defaults to `120`. May also be controlled by setting environment variable: KUHA\_DOCUMENT\_STORE\_CLIENT\_REQUEST\_TIMEOUT.
- document-store-client-connect-timeout** <timeout>  
 Connect timeout (in seconds) for Document Store client. Defaults to `120`. May also be controlled by setting environment variable: KUHA\_DOCUMENT\_STORE\_CLIENT\_CONNECT\_TIMEOUT.
- document-store-client-max-clients** <max\_clients>  
 Maximum number of simultaneous client connections for Document Store client. Defaults to `10`. May also be controlled by setting environment variable: KUHA\_DOCUMENT\_STORE\_CLIENT\_MAX\_CLIENTS.
- loglevel** <loglevel>  
 Lowest logging level of log messages that get output. Valid values are logging levels supported by Python's `logging` [`CRITICAL`, `ERROR`, `WARNING`, `INFO`, `DEBUG`]. Defaults to `INFO`. May also be controlled by setting environment variable: KUHA\_LOGLEVEL

**--logformat** <logformat>

Logging format supported by `logging`. Defaults to `%(asctime)s %(levelname)s %(name)s : %(message)s` May also be controlled by setting environment variable: `KUHA_LOGFORMAT`

### Configuration file

Args that start with '-' (eg. `-document-store-port`) can also be set in a config file. The configuration file lookup searches the file from current working directory and from the package directory. The name of the configuration file is `kuha_oai_pmh_repo_handler.ini`.

---

**Note:** Invoke with `--help` to print out config file lookup paths.

---

### Environment variables

If the program will be run by using the scripts provided in `scripts` subdirectory, the runtime environment can be controlled via `scripts/runtime_env`, which will be created by copying from `scripts/runtime_env.dist` at installation time by `scripts/install_kuha_oai_pmh_repo_handler_virtualenv.sh`.

## 3.4.5 Running the program

This guide will use convenience scripts from `scripts` subdirectory. It is assumed that the program was installed by using `scripts/install_kuha_oai_pmh_repo_handler_virtualenv.sh`.

Run OAI-PMH Repo Handler server:

```
./scripts/run_kuha_oai_pmh_repo_handler.sh --oai-pmh-base-url=<base-url> --oai-pmh-  
↪admin-email=<admin-email>
```

The script will source `scripts/runtime_env` and activate the installed `virtualenv`. Finally it calls `kuha_oai_serve`, with given command line arguments.

## 3.5 Kuha OSMH Repo Handler

Kuha OSMH Repo Handler is a HTTP API written in Python for serving Kuha Document Store records through OSMH.

Kuha OSMH Repo Handler is a part of Open Source software bundle Kuha2.

### 3.5.1 Features

OSMH record types supported:

- Study
- Variable
- Question
- StudyGroup

OSMH Repo Handler supports streaming responses. Note that the requesting party must also support streaming. Streaming is disabled by default.



### 3.5.2 Dependencies & requirements

Versions specified here are the ones the software has been developed with. Newer versions may be compatible.

- Python 3.5
- **Recommended:** python3-venv 3.5.1

#### Python packages

The following can be obtained from Python package index.

- tornado 4.5 (License: Apache License 2.0)

Kuha Common is a library used with Kuha2 software. It can be obtained from [https://bitbucket.org/tietoarkisto/kuha\\_common](https://bitbucket.org/tietoarkisto/kuha_common)

- kuha\_common (License: EUPL)

### 3.5.3 License

Kuha OSMH Repo Handler is available under the EUPL. See LICENSE.txt for the full license.

### 3.5.4 Configuration

The application can be configured with a configuration file, via command line arguments or by environment variables. All configuration options have default values. If a configuration option is specified in more than one place, then command line values override environment variables which override configuration file values which override defaults.

The following configuration options are available:

**-h, --help**

Show help message and exit.

**--print-configuration**

Print active configuration and exit.

**--stream-response**

Switch to enable streaming from ListRecordHeaders endpoint. Defaults to `False`. May also be controlled by setting environment variable: `KUHA_OSMH_STREAM_RESPONSE`.

**--port <port>**

Port for serving OSMH Repo Handler. Defaults to `6002` May also be controlled by setting environment variable: `KUHA_ORH_PORT`.

**--osmh-repo-handler-api-version <api\_version>**

Api version for OSMH Repo Handler. This gets prepended to the URL path. Defaults to `v0`. May also be controlled by setting environment variable: `KUHA_OSMH_API_VERSION`.

**--query-limit <limit>**

Optional query limit for configuring the limit of records per query when fetching multiple records from Document Store. Set `0` to disable. Defaults to `0`. May also be controlled by setting environment variable: `KUHA_OSMH_QUERY_LIMIT`.

**--document-store-host <host>**

Host & scheme of Kuha Document Store. Defaults to `http://localhost`. May also be controlled by setting environment variable: `KUHA_DS_HOST`.

- document-store-port** <port>  
Port of Kuha document store database. Defaults to 6001. May also be controlled by setting environment variable: `KUHA_DS_PORT`.
- document-store-api-version** <api\_version>  
Api version for document store. This gets prepended to the URL path. Defaults to `v0`. May also be controlled by setting environment variable: `KUHA_DS_API_VERSION`.
- document-store-client-request-timeout** <timeout>  
Request timeout (in seconds) for Document Store client. Defaults to 120. May also be controlled by setting environment variable: `KUHA_DOCUMENT_STORE_CLIENT_REQUEST_TIMEOUT`.
- document-store-client-connect-timeout** <timeout>  
Connect timeout (in seconds) for Document Store client. Defaults to 120. May also be controlled by setting environment variable: `KUHA_DOCUMENT_STORE_CLIENT_CONNECT_TIMEOUT`.
- document-store-client-max-clients** <max\_clients>  
Maximum number of simultaneous client connections for Document Store client. Defaults to 10. May also be controlled by setting environment variable: `KUHA_DOCUMENT_STORE_CLIENT_MAX_CLIENTS`.
- loglevel** <loglevel>  
Lowest logging level of log messages that get output. Valid values are logging levels supported by Python's `logging` [`CRITICAL`, `ERROR`, `WARNING`, `INFO`, `DEBUG`]. Defaults to `INFO`. May also be controlled by setting environment variable: `KUHA_LOGLEVEL`.
- logformat** <logformat>  
Logging format supported by `logging`. Defaults to `%(asctime)s %(levelname)s %(name)s : %(message)s`. May also be controlled by setting environment variable: `KUHA_LOGFORMAT`.

### Configuration file

Args that start with '-' (eg. `--document-store-port`) can also be set in a config file. The configuration file lookup searches the file from current working directory and from the package directory. The name of the configuration file is `kuha_osmh_repo_handler.ini`.

---

**Note:** Invoke with `--help` to print out config file lookup paths.

---

### Environment variables

If the program will be run by using the scripts provided in `scripts` subdirectory, the runtime environment can be controlled via `scripts/runtime_env`, which will be created by copying from `scripts/runtime_env.dist` at installation time by `scripts/install_kuha_osmh_repo_handler_virtualenv.sh`.

### 3.5.5 Running the program

This guide will use convenience scripts from `scripts` subdirectory. It is assumed that the program was installed by using `scripts/install_kuha_osmh_repo_handler_virtualenv.sh`.

Run OSMH Repo Handler server:

```
./scripts/run_kuha_osmh_repo_handler.sh
```

The script will source `scripts/runtime_env` and activate the installed virtualenv. Finally it calls `kuha_osmh_serve`, with given command line arguments.

## 3.6 Kuha Client

Kuha Client is used to submit records to Kuha Document Store. Kuha Client is written in Python and uses HTTP to communicate with Document Store.

### 3.6.1 Features

- Support for DDI 3.1, DDI 2.5 and DDI 1.2.2 metadata standards.
- Import records to Document Store.
- Update records already stored in Document Store.
- Delete records in Document Store.
- Batch process DDI files by recursing into directories:
  - Option to remove records from Document Store not found in the current batch.
  - Option to keep track of previously processed files and bypass processing if modification times have not changed.

### 3.6.2 Dependencies & requirements

Versions specified here are the ones that the software has been developed with. Newer versions may be compatible.

- Python 3.5.
- **Recommended:** python3-venv 3.5.1

#### Python packages

Kuha Common is a library used with Kuha2 software. It can be obtained from [https://bitbucket.org/tietoarkisto/kuha\\_common](https://bitbucket.org/tietoarkisto/kuha_common)

- kuha\_common (License: EUPL)

### 3.6.3 License

Kuha Client is available under the EUPL. See LICENSE.txt for the full license.

### 3.6.4 Configuration

Most common configuration options are described here. Use `--help` to print all available options.

#### paths

**Required** positional argument. Absolute path to file or directory. Repeat to process multiple paths.

#### -h, --help

Show help and exit.

#### --collection <collection>

**Only for upsert and import run.** Limits the import to a specific document type. Valid values are [studies, variables, questions, study\_groups]. Set None to import all document types. Defaults to None.

**--document-store-url** <document\_store\_url>

**Required.** Full URL to Document Store, for example `http://localhost:6001/v0`. May also be controlled by setting environment variable: `KUHA_DS_URL`.

**--file-log-path** <path>

**Only for upsert and import run.** Store processed files to file log. Compare modification times on subsequent run. Bypass if modification times have not changed.

**--remove-absent**

**Only for upsert run.** Remove records from Document Store not present in current batch.

### 3.6.5 Running the program

If installed to a Python virtual environment, the environment must be activated before running the program.

**Import records** to Document Store by scanning a directory tree for `.xml` files to submit and create a file-log to keep track of processed files:

```
python -m kuha_client.kuha_import --file-log-path=file_log /path/to/directory
```

**Upsert records** (insert and update) to Document Store by scanning a directory tree for `.xml` files and comparing found files to the ones store in file-log. If a file's modification time is newer than the one stored in file-log, the file gets processed. When using the `--remove-absent` flag, any ID found from document store, but not from the current batch, gets removed:

```
python -m kuha_client.kuha_upsert --file-log-path=file_log --remove-absent /path/to/  
↪directory
```

**Delete record from collection:**

```
python -m kuha_client.kuha_delete studies 5af94ff06fb71d7646160bd4
```

**Delete all records from collection:**

```
python -m kuha_client.kuha_delete studies ALL
```

**Delete all records from all collections:**

```
python -m kuha_client.kuha_delete ALL ALL
```

## 3.7 Kuha Common

Kuha Common is a Python library used with Kuha2 software bundle.

### 3.7.1 Dependencies & requirements

Versions specified here are the ones that the software has been developed with. Newer versions may be compatible.

- Python 3.5
- **Recommended:** `python3-venv 3.5.1`

#### Python packages

The following can be obtained from Python package index.

- ConfigArgParse (License: MIT)
- Tornado (License: Apache License 2.0)

### 3.7.2 License

Kuha Common is available under the EUPL. See LICENSE.txt for the full license.

## 3.8 Developer Documentation

### 3.8.1 kuha\_common

High-level modules common for Kuha applications.

#### server.py

Common server functions & classes used in Kuha.

`kuha_common.server.log_request(handler)`

Log request output to JSON. Gets called after each successful response.

**Parameters** `handler` (subclass of `tornado.web.RequestHandler`) – handler for the current request.

`kuha_common.server.log_exception(typ, value, tb, correlation_id)`

Log exception output to JSON. Gets called from `RequestHandler`.

#### Parameters

- `typ` – type of exception
- `value` – caught exception
- `tb` – traceback
- `correlation_id` – correlation id of the request that ended in exception.

`kuha_common.server.str_api_endpoint(api_version, suffix=None)`

Helper function to prepend endpoints with `api_version`.

#### Parameters

- `api_version` (`str`) – version of the api.
- `suffix` (`str`) – api endpoint.

**Returns** `str` – endpoint prepended with `api_version`

`kuha_common.server.serve(web_app, port, process_count=0, on_exit=None)`

Serve web application.

#### Parameters

- `web_app` (`tornado.web.Application`) – application to serve
- `port` (`int`) – Port to listen to.
- `process_count` (`int`) – number of processes to spawn. 0 = forks one process per cpu.
- `on_exit` (`function`) – callback on server/ioloop stop.

**class** `kuha_common.server.RequestHandler` (*\*args, \*\*kwargs*)

Common request handler for kuha server applications. Subclass in application specific handlers.

**prepare** ()

Prepare each request.

Look for correlation id; create one if not found. Set correlation id to response header.

**set\_output\_content\_type** (*ct, charset='UTF-8'*)

Sets content type for responses.

**Parameters**

- **ct** (*str*) – content type for response.
- **charset** (*str*) – charset definition for response content type.

**log\_exception** (*typ, value, tb*)

Overrides tornado's exception logging. Sends HTTP errors as responses. Calls customised `log_exception()` which outputs JSON encoded log messages with request correlation ids. For easier debugging it also calls `tornado.web.RequestHandler.log_exception` to output full traceback.

**Parameters**

- **typ** – type of exception
- **value** – caught exception
- **tb** – traceback

**assert\_request\_content\_type** (*supported\_content\_type*)

Assert request has correct content type header.

**Parameters** **supported\_content\_type** (*str*) – content type supported by endpoint.

**Raises** `InvalidContentType` – if request has invalid content type.

**write\_error** (*status\_code, \*\*kwargs*)

Overrides `tornado.web.RequestHandler.write_error`. Outputs error messages in preferred content type.

**Parameters**

- **status\_code** (*int*) – HTTP status code.
- **\*\*kwargs** – keyword arguments are passed to `tornado.web.RequestHandler.write_error` if output content type is not `application/json`

**class** `kuha_common.server.WebApplication` (*handlers*)

Override `tornado.web.Application` to make sure server applications are using correct initialization parameters.

**log\_request** (*handler*)

Override `tornado.web.Application.log_request`. Server uses it's own implementation of `log_request`.

**Parameters** **handler** (Subclass of `tornado.web.RequestHandler`) – Handler of current request.

**exception** `kuha_common.server.KuhaServerError` (*msg, status\_code=500, context=None*)

Base class for common HTTP-exceptions

**exception** `kuha_common.server.InvalidContentType` (*requested\_content\_type, supported\_content\_type*)

Invalid content type HTTP-exception.

**exception** `kuha_common.server.BadRequest` (*msg=None*)  
Bad request HTTP-exception.

**exception** `kuha_common.server.ResourceNotFound` (*msg=None, context=None*)  
Resource not found HTTP-exception.

## query.py

Perform query operations against Kuha Document Store.

Offers High-level query methods to facilitate an easy access point with all necessary actions and properties needed to perform queries. To query the Document Store the caller only needs to use methods defined in class `QueryController` and records defined in `kuha_common.document_store.records`

**class** `kuha_common.query.ResultHandler` (*record\_constructor, on\_record=None*)  
Class which handles results and correct calls to callbacks, if any. Stores the result for later use.

Dynamically creates callable method `handle()` which receives result payload and calls `on_record` correctly.

### Parameters

- **record\_constructor** (*kuha\_common.document\_store.records.Study* or *kuha\_common.document\_store.records.Variable* or *kuha\_common.document\_store.records.Question* or *kuha\_common.document\_store.records.StudyGroup*) – Class to construct Document Store record.
- **on\_record** (*function or coroutinefunction or None*) – Callback called with constructed record instance as parameter.

**Returns** *ResultHandler*

**class** `kuha_common.query.QueryController` (*headers=None, record\_limit=0*)  
Asynchronous controller to query the Document Store.

Use to build queries and automatically fetch responses using HTTP as a protocol and JSON as exchange format.

Optional `record_limit` parameter may be given at initialization time to limit the number of records that are requested in a single HTTP request.

### Parameters

- **headers** (*dict*) – Optional headers parameter to store headers that are used in each query application wide as HTTP headers.
- **record\_limit** (*int*) – Optional `record_limit` parameter which is used to limit the number of records requested in a single HTTP request.

Example:

```
from kuha_common.document_store import Study
query_ctrl = QueryController()
study = await query_ctrl.query_single(
    Study,
    fields=[Study._metadata, Study.study_number],
    _filter={Study.study_number: 1234}
)
```

### fk\_constants

alias of `FilterKeyConstants`

**query\_single** (*record*, *on\_record=None*, *headers=None*, *\*\*kwargs*)

Query single record.

#### Parameters

- **record** (Subclass of `kuha_common.document_store.records.RecordBase`) – class used to construct the record instance.
- **on\_record** (*function or coroutinefunction*) – Optional callback function that gets passed the returned and instantiated record object.
- **headers** (*dict*) – Optional headers for this query. Headers get added to headers given for QueryController at initialization time. Note that it will overwrite headers with same key.
- **\*\*kwargs** – Keyword arguments contain parameters for the query. They get passed to `kuha_common.document_store.query.Query.construct()`

**Returns** None if passed *on\_record* callback, else returns the initiated record object.

**Raises** `QueryException` – if query parameters given as keyword arguments contain limit-parameter.

**query\_multiple** (*record*, *on\_record*, *headers=None*, *\*\*kwargs*)

Query multiple records.

Queries the document store for multiple records. Behaviour depends on whether `record_limit` has been set:

#### If there is a record\_limit

- the queries are split in multiple HTTP requests and queued.
- This method returns a `kuha_common.document_store.client.JSONStreamClient.run_queued_requests()`, which is to be called without arguments when the queries are to be executed.
- *on\_record* must be a normal function that takes the constructed record instance as parameter.

#### If there is no record\_limit

- this method returns nothing.
- The *on\_record* callback gets called with each instantiated record object.
- *on\_record* may be a normal function or a coroutine.

#### Parameters

- **record** (Subclass of `kuha_common.document_store.records.RecordBase`) – class used to construct the record instance.
- **on\_record** (*function or coroutine*) – callback that gets called with each instantiated record instance.
- **headers** (*dict*) – optional headers used for this query.
- **\*\*kwargs** – Keyword arguments contain parameters for the query. They get passed to `kuha_common.document_store.query.Query.construct()`

**Returns** None if no `record_limit`, else `kuha_common.document_store.client.JSONStreamClient.run_queued_requests()`

**query\_count** (*record*, *headers=None*, *\*\*kwargs*)

Query the number of records.



**Parameters**

- **record** (Subclass of `kuha_common.document_store.records.RecordBase`) – class used to construct the record instance.
- **headers** (*dict*) – optional headers used for this query.
- **\*\*kwargs** – Keyword arguments contain parameters for the query. They get passed to `kuha_common.document_store.query.Query.construct()`

**Returns** Number of records**Return type** `int`**query\_distinct** (*record*, *headers=None*, *\*\*kwargs*)

Query distinct values.

**Parameters**

- **record** (Subclass of `kuha_common.document_store.records.RecordBase`) – record to query for.
- **headers** (*dict*) – optional headers used for this query.
- **\*\*kwargs** – Keyword arguments contain parameters for the query. They get passed to `kuha_common.document_store.query.Query.construct_distinct()`

**Returns** distinct values: {fieldname : [value1, value2, value3]}. Note that contained values may be dictionaries or strings, depending on what is stored in requested field.**Return type** `dict`**cli\_setup.py**

Command line setup for Kuha applications

Parse command line for common configuration options and store loaded settings.

Load modules for querying Document Store:

```
import os
from kuha_common import cli_setup
cli_setup.load(os.getcwd())
settings = cli_setup.setup(cli_setup.MOD_DS_CLIENT, cli_setup.MOD_DS_QUERY)
```

`kuha_common.cli_setup.MOD_DS_CLIENT = 'document_store.client'`Constant for configuring `kuha_common.document_store.client``kuha_common.cli_setup.MOD_DS_QUERY = 'document_store.query'`Constant for configuring `kuha_common.document_store.query``kuha_common.cli_setup.MOD_LOGGING = 'logging'`Constant for configuring `logging``kuha_common.cli_setup.add_kuha_loglevel(parser)`

Add loglevel to parser

**Parameters** `parser` (*ArgumentParser instance*) – command line parser.`kuha_common.cli_setup.add_kuha_logformat(parser)`

Add logformat to parser

**Parameters** `parser` (*ArgumentParser instance*) – command line parser.

`kuha_common.cli_setup.add_document_store_url (parser, **kw)`  
Add document store url to parser

**Parameters**

- **parser** (*ArgumentParser instance*) – command line parser.
- **\*\*kw** – keyword arguments get passes to parser.add

`kuha_common.cli_setup.add_document_store_host (parser)`  
Add document store host to parser

**Parameters** **parser** (*ArgumentParser instance*) – command line parser.

`kuha_common.cli_setup.add_document_store_port (parser)`  
Add document store port to parser

**Parameters** **parser** (*ArgumentParser instance*) – command line parser.

`kuha_common.cli_setup.add_document_store_api_version (parser)`  
Add document store api-version to parser

**Parameters** **parser** (*ArgumentParser instance*) – command line parser.

`kuha_common.cli_setup.add_document_store_client_request_timeout (parser)`  
Add document store client request timeout to parser

**Parameters** **parser** (*ArgumentParser instance*) – command line parser.

`kuha_common.cli_setup.add_document_store_client_connect_timeout (parser)`  
Add document store client connect timeout to parser

**Parameters** **parser** (*ArgumentParser instance*) – command line parser

`kuha_common.cli_setup.add_document_store_client_max_clients (parser)`  
Add document store client max clients timeout to parser

**Parameters** **parser** (*ArgumentParser instance*) – command line parser.

`kuha_common.cli_setup.add_print_configuration (parser)`  
Add print configuration helper for testing configuration options.

**Parameters** **parser** (*ArgumentParser instance*) – command line parser.

**class** `kuha_common.cli_setup.KuhaConfigFileParser`  
Inherit to override `configargparse.DefaultConfigFileParser`.  
`get_syntax_description()`

**get\_syntax\_description()**  
Override syntax description of `configargparse.DefaultConfigFileParser`

**Returns** Syntax description for configuration file.

**Return type** `str`

**class** `kuha_common.cli_setup.Settings`  
Class for command line settings.

**is\_parser\_loaded()**  
Check is parser loaded.

**Return type** `bool`

**is\_settings\_loaded()**  
Check is settings loaded.

**Return type** `bool`

**set\_abs\_dir\_path** (*path*)

Set absolute directory path of configurable kuha application.

**Parameters** **path** (*str*) – absolute path to kuha application directory.

**get\_abs\_dir\_path** ()

Return absolute directory path of configurable kuha application.

**Returns** absolute path to directory.

**Return type** *str*

**add\_logging\_configs** ()

Wrapper to add logging-module configuration.

**add\_document\_store\_query\_configs** ()

Wrapper to add document\_store\_query configuration.

**add\_document\_store\_client\_configs** ()

Wrapper to add document\_store\_client configuration.

**setup\_logging** ()

Setup *logging* module.

**setup\_document\_store\_query** ()

Setup *kuha\_common.document\_store.query* module.

**setup\_document\_store\_client** ()

Setup *kuha\_common.document\_store.client* module.

**load\_parser** (*config\_file=None, \*\*kw*)

Load command line parser.

**Parameters**

- **description** (*str*) – Description to print for configurable module. It is passed to `configargparse.ArgumentParser`.
- **config\_file** (*str*) – Name of configuration file.

**set** (*parsed\_opts*)

Assign parser options to settings.

**Parameters** **parsed\_opts** (`argparse.Namespace`) – parser options.

**load\_cli\_args** ()

Load command line arguments.

**get** ()

Return active settings.

**Returns** active settings.

**Return type** `argparse.Namespace`

**add** (*\*args, \*\*kwargs*)

Add item for parser. Settings must not yet be loaded but parser must be loaded.

**Parameters**

- **\*args** – arguments passed to `configargparse.ArgumentParser`
- **\*\*kwargs** – keyword arguments passed to `configargparse.ArgumentParser`

`kuha_common.cli_setup.setup(*modules)`  
Setup command line parser.

Load modules, parse command line arguments, return loaded settings in `argparse.Namespace`

**Parameters** `*modules` (*str*) – common Kuha modules to load and include in parsing of command line arguments.

**Returns** Loaded settings.

**Return type** `argparse.Namespace`

`kuha_common.cli_setup.get_settings()`  
Get loaded settings stored in `Settings`.

**Returns** Loaded settings.

**Return type** `argparse.Namespace`

`kuha_common.cli_setup.add(*args, **kwargs)`  
Module level function to add items to be parsed in `Settings` singleton.

**Parameters**

- `*args` – arguments passed to `Settings.add()`.
- `**kwargs` – keyword arguments passed to `Settings.add()`

`kuha_common.cli_setup.load(abs_dir_path, **kwargs)`  
Module level function to load parser to `Settings` singleton.

**Parameters**

- `abs_dir_path` (*str*) – absolute path to directory of the kuha application to be configured.
- `**kwargs` – keyword arguments passed to `Settings.load_parser()`.

`kuha_common.cli_setup.prepend_abs_dir_path(path)`  
Helper function to prepend the stored absolute directory path to given argument.

**Parameters** `path` – end of the path.

**Returns** absolute path ending to `path`.

**Return type** `str`

## document\_store

Contains modules for interacting with Document Store.

### document\_store/client.py

`kuha_common.document_store.client` provides a http client interface to communicate with Kuha Document Store.

**class** `kuha_common.document_store.client.JSONStreamClient`

Base class used for requests. Implements own queue to store requests, since `tornado.httpclient.AsyncHTTPClient` starts timers for `request_timeout` and `connect_timeout` at the moment we call `client.fetch()`. See <https://github.com/tornadoweb/tornado/issues/1400> for more details.

Handles JSON decoding of incoming chunks and the encoding of body to JSON.

**max\_clients = 10**

Controls the maximum number of concurrent clients.

**request\_timeout = 120**

Sets timeout for a request.

**connect\_timeout = 120**

Sets timeout for establishing a connection.

**sleep\_on\_queue = 5**

Sets sleep timer for queue.

**classmethod set\_max\_clients** (*max\_clients*)

Set maximum concurrent clients

**classmethod set\_request\_timeout** (*request\_timeout*)

Set timeout per request

**classmethod set\_connect\_timeout** (*connect\_timeout*)

Set timeout per connection

**classmethod request** (*url*, *\*\*kwargs*)

Constructs a streaming request.

#### Parameters

- **url** (*str*) – url to request.
- **kwargs** – keyword arguments passed to `tornado.httpclient.HTTPRequest`

**Returns** `tornado.httpclient.HTTPRequest`

**wrap\_streaming\_callback** (*callback*)

Wrap streaming callback to support chunked JSON responses.

**Parameters** **callback** (*callable.*) – streaming callback. Gets called with response which is decoded to python object from JSON.

**Returns** Wrapped callback

**Return type** `functools.partial`

**execute\_stored\_callbacks** ()

Executes asynchronous callbacks stored in `_callbacks`

**run\_queued\_requests** (*queued\_requests=None*)

Run queued requests.

Calls queued requests asynchronously. Sleeps for `sleep_on_queue` if `max_clients` reached.

**Parameters** **queued\_requests** (`collections.deque`) – Optionally pass `queued_requests` to run.

**get\_streaming\_request** (*streaming\_callback*, *url*, *body=None*, *method=None*, *headers=None*, *\*\*kw*)

Get a streaming request.

Sets default headers `Content-Type: application/json` if not already given. Encodes body to JSON if given and is not string or bytes. If response is empty (for example query with no results) the streaming callback doesn't get called.

Subclass and override to support arbitrary requests.

#### Parameters

- **streaming\_callback** (*callable*) – callback which receives the response if any.

- **url** (*str*) – URL to send request to.
- **body** (*str, dict, list, tuple, integer, float or None*) – Optional request body. String will be supplied as is. Other values will be encoded to JSON.
- **method** (*str or None*) – HTTP method. Defaults to POST.
- **headers** (*dict or None*) – optional request headers. if Content-Type is not set, will set 'Content-Type': 'application/json' as default.

**Returns** HTTP request

**Return type** `tornado.httpclient.HTTPRequest`

**queue\_request** (*\*args, \*\*kwargs*)

Queue request to be run asynchronously by calling `run_queued_requests`.

**Parameters**

- **\*args** – arguments passed to `get_streaming_request`
- **\*\*kwargs** – keyword arguments passed to `get_streaming_request`.

**Returns** `run_queued_requests()` method to call to run the queued requests.

**fetch** (*\*args, \*\*kwargs*)

Run single query.

**Parameters**

- **\*args** – arguments passed to `queue_requests`.
- **\*\*kwargs** – keyword arguments passed to `queue_requests`

## document\_store/query.py

Access query properties by convenience methods to help build valid queries against the Document Store.

**class** `kuha_common.document_store.query.FilterKeyConstants`

Class used as a namespace to contain constants used in query filter.

**exception** `kuha_common.document_store.query.QueryException` (*msg, context=None*)

Exception for errors raised by `Query`. Has an optional context parameter for giving some additional information about the context of the exception.

**class** `kuha_common.document_store.query.Query` (*query, query\_document, query\_type='select'*)

Manipulate query properties without compromising the validity of the constructed query. Build the correct url for different query types.

**Note** This class provides low-level operations. Use `kuha_common.query.QueryController` for easy access to query actions and properties.

Example:

```
from kuha_common.document_store import Study, Query
query = Query(Query.construct(_filter={Study.study_number:'123'}), Study.
↳collection)
```

**Parameters**

- **query** (*dict*) – Actual query containing the properties such as `_filter`, `fields`, `sort_by` etc.

- **query\_document** (*str*) – One of the supported query documents declared in `Query.supported_query_documents` and specified in `kuha_common.document_store.records.py`
- **query\_type** (*str*) – Optional `query_type` parameter. Defaults to `Query.query_type_select`. Other valid values are `Query.query_type_count` and `Query.query_type_distinct`.

**k\_filter** = `'_filter'`

Query parameter for filtering results.

**k\_fields** = `'fields'`

Query parameter for fields to contain in results.

**k\_limit** = `'limit'`

Query parameters for limiting returned results.

**k\_skip** = `'skip'`

Query parameter for skipping number of results from the beginning of the resultset.

**k\_sort\_order** = `'sort_order'`

Query parameter for sort order.

**k\_sort\_by** = `'sort_by'`

Query parameter for sorting by a certain field.

**k\_fieldname** = `'fieldname'`

Query parameter for distinct queries. Specifies a field from which the distinct values are to be fetched.

**query\_type\_select** = `'select'`

Query type for `select` queries. Using this query type gets records as a response.

**query\_type\_count** = `'count'`

Query type for count queries. Using this query type the query returns an integer.

**query\_type\_distinct** = `'distinct'`

Query type for distinct queries. Using this query type the returning object contains all distinct values for a certain field.

**classmethod set\_base\_url** (*base\_url*)

Configure document store url used as a base when constructing the endpoint url for queries.

**classmethod as\_supported\_datetime\_str** (*datetime\_obj*)

Get datetime object as a supported datetime string.

**Parameters** *datetime\_obj* (*datetime-object.*) – Python datetime-object to be converted to str.

**Returns** String representation of the datetime-object that is suitable for querying.

**Return type** `str`

**classmethod construct** (*\*\*kwargs*)

Construct valid query parameters.

Example:

```
from kuha_common.document_store import Query, Study
params = Query.construct(_filter={Study.study_number:'123'},
                        fields=[Study._metadata, Study._id, Study.abstract],
                        sort_by=Study._id)
query = Query(params, Study.collection)
```

**Parameters** **\*\*kwargs** – keys should be valid query properties, while values should hold corresponding query values supported by the key.

**Returns** Valid query, ready to be sent to Document Store.

**Return type** `dict`

**classmethod** **construct\_distinct** (*\*\*kwargs*)

Construct valid query parameters for distinct queries.

**Parameters** **\*\*kwargs** – keys should be valid query properties, while values should hold corresponding query values supported by the key.

**Returns** Valid query, ready to be sent to Document Store.

**Return type** `dict`

**classmethod** **build\_query\_for\_date\_range** (*from\_=None, until=None*)

Build query filter for date-range.

**Parameters**

- **from** (*datetime-object*) – start of the date-range:
- **until** (*datetime-object*) – end of the date-range:

**Returns** date-range query-filter with datetime-objects converted into string representation.

**Return type** `dict`

**classmethod** **build\_query\_for\_exists** (*exists*)

Build query for exists-query.

**Parameters** **exists** (*bool*) – whether the field should exists or not.

**Returns** valid exists query for filter.

**Return type** `dict`

**Raises** `ValueError` for invalid boolean values in exists-parameter.

**classmethod** **get\_valid\_params** (*query\_type=None*)

Return valid query parameters for the query type.

**Parameters** **query\_type** (*str*) – Optional query\_type for which the query-parameters should be valid for.

**classmethod** **is\_valid\_query** (*query, query\_type*)

Check the validity of query parameters.

**Parameters**

- **query** (*dict*) – Full query to validate.
- **query\_type** (*str*) – Query type to validate against.

**Returns** Whether or not the query-parameters given are valid.

**Return type** `bool`

**classmethod** **is\_valid\_query\_type** (*query\_type*)

Check the validity of query\_type.

**Parameters** **query\_type** (*str*) – Query type to validate.

**Returns** Whether or not the query type is valid.

**Return type** `bool`



**is\_valid\_query\_document** (*query\_document*)

Check the validity of query document.

**Parameters** **query\_document** (*str*) – Query document to validate.

**Returns** Whether or not the query document is valid.

**Return type** `bool`

**is\_valid\_param** (*parameter*)

Check the validity of a single query parameter.

**Parameters** **parameter** (*str*) – Query parameter to validate.

**Returns** Whether or not the parameter is valid.

**Return type** `bool`

**validate\_query** (*query*)

Validate query parameters.

Checks parameters' validity for chosen query type. Raises *QueryException* if invalid.

**Parameters** **query** (*dict*) – Query parameters.

**Returns** Query parameters.

**Return type** `dict`

**Raises** *QueryException* if query parameters are invalid.

**validate\_query\_type** (*query\_type*)

Validate query type.

Checks that the query type is supported by Document Store. Raises *QueryException* for invalid query type.

**Parameters** **query\_type** (*str*) – Query type to validate.

**Returns** Query type.

**Return type** `str`

**Raises** *QueryException* if query type is invalid.

**validate\_query\_document** (*query\_document*)

Validates query document.

Checks that the query document is supported by Document Store. Raises *QueryException* if invalid.

**Parameters** **query\_document** (*str*) – Query document to validate.

**Returns** Query document

**Return type** `str`

**Raises** *QueryException* if query document is invalid.

**get\_endpoint** ()

Get correct endpoint for querying the Document Store.

Builds the endpoint by consulting configured values and the instantiated query for *query\_type* and *query\_document*

**Returns** Full url to Document Store endpoint which handles the constructed query.

**Return type** `str`

**get\_query** (*strip\_invalid\_params=True*)

Returns the constructed query parameters.

If the query type has been changed after initialization, for example to get the count of records, this method strips the invalid query parameters from the returned query. When doing so, it does not change the stored query parameters, but rather makes a copy of them for manipulating and returning.

**Parameters** **strip\_invalid\_params** (*bool*) – Whether to strip the unsupported (=invalid) query parameters out of the returned query.

**Returns** Constructed query parameters ready to submit to Document Store.

**Return type** `dict`

**get\_limit** ()

Get query limit parameter.

**Returns** Query limit (*int*) if set. None if not set.

**Return type** `int` or `None`

**get\_skip** ()

Get query skip parameter.

**Returns** Query skip (*int*) if set. None if not set.

**Return type** `int` or `None`

**set\_limit** (*limit*)

Set limit parameter for query.

Limit controls how many results should be returned.

**Parameters** **limit** (*int*) – Limit parameter for query.

**Returns** self for easy aggregation of manipulation methods.

**Return type** instantiated `Query()`

**set\_skip** (*skip*)

Set skip parameter for query.

Skip controls how many results should be skipped from the start (offset).

**Parameters** **skip** (*int*) – Skip parameter for query.

**Returns** self for easy aggregation of manipulation methods.

**Return type** instantiated `Query()`

**set\_fields** (*fields*)

Set fields parameter for query.

Field controls which fields of the record should be returned. *fields* can be a list of strings in the form used by MongoDB or a list of `kuha_common.document_store.records` class-variables.

Example:

```
from kuha_common.document_store import Query, Study
_params = Query.construct(_filter={Study.study_number:'123'})
_query = Query(_params, Study.collection)
_query.set_fields([Study.abstract, Study.study_number])
```

**Parameters** **fields** (*list*) – Fields parameter for query.

**Returns** self for easy aggregation of manipulation methods.

**Return type** instantiated *Query()*

**set\_sort\_by** (*sort\_by*)

Set *sort\_by* parameter for query.

Determines sorting of the returned results. *sort\_by* can be a string in the form used by MongoDB or a *kuha\_common.document\_store.records* class-variables.

**Parameters** *sort\_by* (*srt* or *class-variable of a record.*) – Sort by parameter for query.

**Returns** self for easy aggregation of manipulation methods.

**Return type** instantiated *Query()*

**set\_sort\_order** (*order*)

Set sort order for the query.

Determines the order which the returned results are to be sorted by.

**Note** Valid values come from *pymongo*. They actually depend on the *mongodb* driver, but since this is a caller API we don't want to make *pymongo* a dependency.

**Parameters** *order* (*int*) – Sort order. Must be either 1 or -1.

**Returns** self for easy aggregation of manipulation methods.

**Return type** instantiated *Query()*

**Raises** *QueryException* for invalid order values.

**set\_query\_type** (*query\_type*)

Set query type.

**Parameters** *query\_type* (*str*) – Valid query type for the query to be constructed.

**Returns** self for easy aggregation of manipulation methods.

**Return type** instantiated *Query()*

**add\_query\_statement** (*field*, *statement*)

Add query statement.

Manipulates the *\_filter* parameter of the query parameters. Raises a *QueryException* if the field already has a statement declared in *\_filter*.

**Parameters**

- **field** (*str*) – Field to target the statement to.
- **statement** (*str*) – Statement to filter the results by.

**Returns** self for easy aggregation of manipulation methods.

**Return type** instantiated *Query()*

**add\_query\_statements** (*\*\*kwargs*)

Add multiple query statements to filter the returned results.

Manipulates the *\_filter* parameter of the query parameters.

**Parameters** *\*\*kwargs* – key-value pairs that are to be added to the *\_filter* parameter.

**Returns** self for easy aggregation of manipulation methods.

**Return type** instantiated *Query()*

## document\_store/field\_types.py

Properties and actions for field types supported by records defined in `kuha_common.document_store.records`

Provides field types to be used not only for the construction of new records and updating existing records, but also to provide a format for fields of records that is interchangeable in a way that a receiver does not need to know the specifics of a field beforehand, but may use the field to gain knowledge of the properties of the field.

This module also provides factories which are used to fabricate the fieldtypes. The instantiated factories hold knowledge of the fields even though the fields themselves are not yet instantiated. This knowledge is used for querying records, but also to dynamically fabricate fieldtypes for records in `kuha_common.document_store.records`

**exception** `kuha_common.document_store.field_types.FieldTypeException`

Exception to raise on field type errors. Used for programming errors.

**class** `kuha_common.document_store.field_types.Value` (*name*, *value=None*)

Value is the most simple type of field.

Field type with name and single value. Serves also as a baseclass for other field types.

### Parameters

- **name** (*str*) – Name of the field.
- **value** – Optional value for the field.

**set\_value** (*value*)

Set initial value.

**Parameters** **value** – Value to set.

**add\_value** (*value*)

Add value for the field.

**Note** Overrides existing value.

**Parameters** **value** – The value to set.

**get\_value** ()

Get the value of the field.

**Returns** Value.

**get\_name** ()

Get name of the field.

**Returns** The name of the field.

**Return type** `str`

**export\_dict** ()

Exports Value as dictionary.

**Returns** {name:value}

**Return type** `dict`

**import\_records** (*record*)

Import record to field.

**Parameters** **record** – record to import.

**updates** (*secondary\_values*)

Update value.

**Parameters** `secondary_values` (*str*) – Value to update to.

**class** `kuha_common.document_store.field_types.Set` (*name*, *value=None*)  
Set is a field type with name and list of unique values.

Derived from `Value` Implements methods that differ from parent class.

#### Parameters

- **name** (*str*) – Name of the field.
- **value** (*list* or *None*) – Optional value for the field.

**set\_value** (*value*)  
Sets value.

**Parameters** `value` (*list*) – Value for field.

**Raises** `FieldTypeException` if submitted value is not a list.

**add\_value** (*value*)  
Add value to field.

Appends a value to the list of values already set. Makes sure that the list holds no duplicates by silently discarding them.

**Parameters** `value` (*list* or *str* or *None*) – value or values to be appended. If value is `None`, empties the list.

**import\_records** (*record*)  
Import records by adding the submitted records to contained values.

**Parameters** `record` (*list* or *str* or *None*) – Hold the values to be imported.

**updates** (*secondary\_values*)  
Updates old values with values contained in this Set.

Looks for combination of `secondary_values` and values in this set. Discards duplicates and stores the updated values to `value`.

**Parameters** `secondary_values` (*list*) – list of old values to be updated with new ones.

**class** `kuha_common.document_store.field_types.Element` (*name*, *attribute\_names=None*)  
Element is a field type with name, value and attributes.

Derived from `Value`.

Element is used to store fields that contain attributes in addition to a value. Each attribute in itself is an instance of `Value` and is dynamically stored in instance variable `attr_<name>`. When instantiated and populated with values and attributes, the element-instance can be used to get it's value, value's name, but also to get the attributes and their names, even though the caller does not know the attribute names a priori.

Example of constructing an element (the source):

```
>>> from kuha_common.document_store.field_types import Element
>>> animal = Element('animal', ['color', 'weight', 'height'])
>>> animal.add_value('cat', color='yellow', weight=10, height=5)
```

Example of reading from an unknown element (the receiver):

```
>>> unknown_element.get_name()
'animal'
>>> unknown_element.get_value()
'cat'
>>> for att in unknown_element.iterate_attributes():
```

```

...     att.get_name() + ' : ' + str(att.get_value())
...
'height : 5'
'color : yellow'
'weight : 10'
>>> unknown_element.attr_color.get_value()
'yellow'

```

This is especially useful when using as an interchange format. The receiver does not need to know the attribute names beforehand. Instead the receiver can iterate through every attribute to get their name-value pairs or if the receiver is interested in a single attribute, it may be called by the dynamically constructed instance-variable prefixed with *attr\_*.

#### Parameters

- **name** (*str*) – Name of the field.
- **attribute\_names** (*list*) – Optional parameter for attribute names.

Raises *FieldTypeException* if *attribute\_names* has duplicates.

#### **is\_pending()**

Is the element pending for values.

**Returns** True if pending, False if not.

**Return type** *bool*

#### **new()**

Create a new element-instance with same name and attributes but without values.

Instantiates a new instance of itself. The new instance is pending for values.

Example:

```

>>> animal = Element('animal', ['color', 'weight', 'height'])
>>> animal.add_value('cat', color='yellow', weight=10, height=5)
>>> another_animal = animal.new()
>>> another_animal.add_value('dog', color='white', weight=30, height=15)

```

**Returns** new element.

**Return type** *Element*

#### **add\_value** (*value=None, \*\*attributes*)

Add value with attributes as keyword arguments.

**Note** This may only be called once for each instance.

Example:

```

>>> from kuha_common.document_store.field_types import Element
>>> animal = Element('animal', ['color', 'weight', 'height'])
>>> animal.add_value('cat', color='yellow', weight=10, height=5)

```

#### Parameters

- **value** (*str or int or None*) – Value for the element.
- **\*\*attributes** – keyword arguments for attributes of the element.

**Raises** *FieldTypeException* if the element already has values or if submitted value is None and no attributes are given.

**iterate\_attributes** ()

Generator function. Iterates element attributes.

**Returns** a generator object for iterating attributes.

**get\_attribute** (*name*)

Get attribute by attribute name.

**Parameters** **name** (*str*) – Name of the attribute to look for.

**Returns** attribute of the element or None if not found.

**Return type** *Value* or None

**set\_attribute** (*name, value*)

Sets new value for attribute.

**Note** The element must have an attribute with the *name*.

**Parameters**

- **name** (*str*) – attribute name.
- **value** (*str or int or None*) – new value.

**Raises** *FieldTypeException* if element does not have an attribute with submitted *name*.

**export\_attributes\_as\_dict** ()

Export element's attributes as a dictionary.

**Returns** dictionary representing the attributes

**Return type** *dict*

**export\_dict** ()

Export the element as a dictionary.

Returns a dictionary with key-value pairs given wrapped inside a another dictionary with the elements key as name.

Example:

```
>>> from kuha_common.document_store.field_types import Element
>>> animal = Element('animal', ['color', 'weight', 'height'])
>>> animal.add_value('cat', color='yellow', weight=10, height=5)
>>> animal.export_dict()
{'animal': {'color': 'yellow', 'weight': 10, 'height': 5, 'animal': 'cat'}}
```

**Returns** dictionary representing the *Element*

**Return type** *dict*

**import\_records** (*record*)

This object does not support importing records.

**Raises** *FieldTypeException*

**updates** (*secondary\_values*)

Updates attributes not found in this element with the ones found from *secondary\_values*.

**Parameters** **secondary\_values** (*dict*) – Attributes from old element.

**class** `kuha_common.document_store.field_types.LocalizableElement` (*name*, *attribute\_names=None*)

LocalizableElement is a field type with name, value, language and attributes.

Derived from `Element`. Has an additional attribute for language. The language is special attribute that is used when updating elements.

**Seealso** `Element`

**Parameters**

- **name** (*str*) – Name of the element.
- **attribute\_names** (*list*) – Optional list of attribute names.

**Raises** `FieldTypeException` if *attribute\_names* contain a name that is reserved for language.

**set\_language** (*language*)

Set language for element.

**Parameters** **language** (*str*) – language to set.

**Raises** `FieldTypeException` if language already set.

**get\_language** ()

Get language of element.

**Returns** language

**Return type** `str` or `None`

**add\_value** (*value=None, language=None, \*\*attributes*)

Add values for element.

**Note** This may only be called once for each instance.

**Seealso** `Element.add_value()`

**Parameters**

- **value** (*str* or *int*) – value to set.
- **language** (*str*) – language of the element.
- **\*\*attributes** – keyword arguments for attributes of the element.

**Raises** `TypeError` if language is not given or is `None`.

**export\_dict** ()

Export the element as a dictionary.

**Seealso** `Element.export_dict()`

**Returns** dictionary representation of the element.

**Return type** `dict`

**class** `kuha_common.document_store.field_types.ElementContainer` (*name*, *sub\_element*)

ElementContainer contains a list of single type of `Element/LocalizableElement` field types.

Receives mandatory parameters for *name* and *sub\_element*. The *sub\_element* describes the element types that this container can store. Every new element that a container can create will be an instance created from this *sub\_element*.

Example:



```

>>> from kuha_common.document_store.field_types import ElementContainer, LocalizableElement
>>> animal = LocalizableElement('animal', ['color', 'width', 'height'])
>>> animals = ElementContainer('animals', animal)
>>> animals.add_value('cat', 'en', color='yellow', width=10, height=5)
>>> animals.add_value('kissa', 'fi', color='keltainen', width=10, height=5)
>>> animals.export_dict() # result formatted for better readability
{'animals': [
  {'width': 10,
   'language': 'en',
   'color': 'yellow',
   'height': 5,
   'animal': 'cat'},
  {'width': 10,
   'language': 'fi',
   'color': 'keltainen',
   'height': 5,
   'animal': 'kissa'}]}

```

Elements can be iterated:

```

>>> for animal in animals:
...     animal.attr_color.get_value() + " for language: " + animal.get_language()
...
'yellow for language: en'
'keltainen for language: fi'

```

And updated with containers sharing name and attribute names:

```

>>> another_animal = LocalizableElement('animal', ['color', 'width', 'height'])
>>> more_animals = ElementContainer('animals', another_animal)
>>> more_animals.add_value('dog', 'en', color='white', width=20, height=10)
>>> more_animals.add_value('koira', 'fi', color='valkoinen', width=20, height=10)
>>> animals.update(more_animals)
>>> animals.export_dict() # result formatted for better readability
{'animals': [
  {'language': 'en',
   'height': 5,
   'color': 'yellow',
   'animal': 'cat',
   'width': 10},
  {'language': 'fi',
   'height': 5,
   'color': 'keltainen',
   'animal': 'kissa',
   'width': 10},
  {'language': 'en',
   'height': 10,
   'color': 'white',
   'animal': 'dog',
   'width': 20},
  {'language': 'fi',
   'height': 10,
   'color': 'valkoinen',
   'animal': 'koira',
   'width': 20}]}

```

**Parameters**

- **name** (*str*) – name of the container.
- **sub\_element** (*LocalizableElement* or *Element*) – element to contain.

**Raises** *FieldTypeException* for invalid *sub\_element*.

**import\_records** (*record*)

Imports records from a list of dictionaries.

**Note** The dictionaries will lose information.

**Parameters** **record** (*list*) – list of dictionaries with records to import.

**add\_value** (*value=None, language=None, \*\*kwargs*)

Add new element to list of elements

**Parameters**

- **value** (*str or int or None*) – value for the new element.
- **language** (*str or None*) – language for the new element.
- **\*\*kwargs** – key-value pairs for attributes of the new element.

**Raises** *FieldTypeException* for invalid language parameter depending on whether the *sub\_element* is localizable.

**export\_dict** ()

Export container as dictionary.

**Returns** dictionary representing the container.

**Return type** *dict*

**iterate\_values\_for\_language** (*language*)

Generator for iterating contained elements by language.

**Parameters** **language** (*str*) – language which is used to filter yielded results.

**Returns** a generator object for iterating elements

**get\_available\_languages** ()

Get list of languages for this container.

**Returns** list of distinct languages.

**Return type** *list*

**updates** (*secondary\_values*)

Updates contained values with *secondary\_values*.

Looks for values that are not currently contained, and appends them as contained values. Also appends different language versions. If a language version has the same value, looks for differences in attributes. If new value has not the same attributes as the old one, adds these attributes to the new value. If old value has same attribute name, it will be discarded.

**Note** Document Store uses MongoDB as a backend. MongoDB deals with JSON-like objects, which in turn are best represented in Python as dictionaries. The purpose of *kuha\_common.document\_store.records* is to be used as a global (in Kuha context) interchange format and so it will be best to support both dictionaries and ElementContainers for this operation. Therefore there is some flexibility in the type of parameter that this method accepts.

**Note** There is a logical difference in which type of parameter is submitted to this method. When using other types than `ElementContainers`, the parameter's content will be changed.

**Parameters** `secondary_values` (instance of `ElementContainer` or dict or list) – Old values known to have the same container (must have the same name). If `secondary_values` is a list, it is assumed that the caller has explicitly checked that the parameter represents old values for this container. Otherwise the name of the container will be checked here and `KeyError` exceptions will be raised.

```
class kuha_common.document_store.field_types.FieldAttribute(name,          par-
                                                         ent=None)
```

Common attributes for each field type.

Stores fields name, parent fields name and constructs a path for the field. This path can be used when building queries against Document Store. The name can be used to lookup values from objects returned from Document Store.

Used by `FieldTypeFactory` to store information of fields that can be used before the field has been fabricated.

#### Parameters

- **name** (`str`) – name of the field.
- **parent** (`str`) – optional parameter parent. Used for sub-elements.

```
value_from_dict (_dict)
```

Get value or values corresponding to `path` from parameter.

**Note** Returned values cannot be separated by language afterwards.

**Parameters** `_dict` (`dict`) – dictionary to lookup for path.

**Returns** value or values stored in path of the `_dict`.

**Return type** `str` or `list` or `None`

```
class kuha_common.document_store.field_types.FieldTypeFactory(name,
                                                         sub_name=None,
                                                         attrs=None,
                                                         localiz-
                                                         able=True,  sin-
                                                         gle_value=False)
```

Factory for field types.

Stores information for each field, that can be used before the field actually has been initiated. This is useful for building queries against Document Store, because the caller needs to know the names and paths of the fields about to be queried.

The attributes stored here are also used to fabricate each field type. This means that each of the field types supported by `kuha_common.document_store.records` are to be initiated through this factory.

**Seealso** `ElementContainer`

Example:

```
>>> from kuha_common.document_store.field_types import FieldTypeFactory
>>> animals_factory = FieldTypeFactory('animals', 'animal', ['color', 'width',
↳ 'height'])
>>> animals_factory.attr_color.name
'color'
>>> animals_factory.attr_color.path
'animals.color'
```

```
>>> animals = animals_factory.fabricate()
>>> animals.add_value('cat', 'en', color='yellow', height=10, width=5)
>>> animals.export_dict()
{'animals': [{'color': 'yellow', 'animal': 'cat', 'height': 10, 'width': 5,
↪ 'language': 'en'}]}
```

### Parameters

- **name** (*str*) – name of the field.
- **sub\_name** (*str*) – name of the sub field, if any.
- **attrs** (*list or str*) – field attributes, if any. Multiple attributes in list.
- **localizable** (*bool*) – is the field localizable.
- **single\_value** (*bool*) – The fabricated field can contain only a single value.

**Raises** `ValueError` if attribute has same name as the element or sub\_element.

**Raises** `FieldTypeException` for parameter combinations that are not supported.

### **fabricate** ()

Fabricate field type by factory attributes.

Returns the correct type of field type based on attributes given to the factory at initialization time.

**Returns** Instance of one of the fields types.

## document\_store/records.py

Models for records supported by Document Store.

Due to its schemaless design, the document store relies heavily on these models. Use these models when building importers.

`kuha_common.document_store.records.datetime_to_datestamp(_datetime)`

Convert datetime object to datestamp string supported by Document Store.

**Parameters** `datetime` (`datetime.datetime`) – datetime to convert.

**Returns** converted datestamp.

**Return type** `str`

`kuha_common.document_store.records.datestamp_to_datetime(datestamp)`

Convert datestamp string to `datetime.datetime` object.

**Parameters** `datestamp` (*str*) – datestamp to convert.

**Returns** converted datetime.

**Return type** `datetime.datetime`

`kuha_common.document_store.records.datetime_now()`

Get current datetime in supported format.

**Returns** Supported datetime object representing current time.

**Return type** `datetime.datetime`

**class** `kuha_common.document_store.records.RecordBase` (*document\_store\_dictionary=None*)  
 Baseclass for each record.

Provides methods used to import, export, create and update records. Dynamically fabricates each class variable of type `FieldTypeFactory` into an instance variable overriding the class variable.

**Note** Use this class through subclasses only.

**Parameters** `document_store_dictionary` (*dict*) – Optional parameter for creating a record at initialization time. Note that this dictionary will be iterated destructively.

**classmethod** `get_collection` ()

Get record collection.

Collection is used for queries against the Document Store.

**Returns** collection of the record.

**Return type** `str`

**classmethod** `iterate_record_fields` ()

Iterate class attributes used as record fields.

Iteration returns tuples: (`attribute_name`, `attribute`)

**Returns** generator for iterating record fields.

**export\_metadata\_dict ()**

Export record metadata as dictionary.

**Returns** record's metadata

**Return type** `dict`

**export\_dict** (*include\_metadata=True, include\_id=True*)

Return dictionary representation of record.

**Parameters**

- **include\_metadata** (*bool*) – export includes metadata
- **include\_id** (*bool*) – export includes id

**Returns** record

**Return type** `dict`

**set\_updated** (*value=None*)

Set updated timestamp.

Sets updated metadata attribute.

**Note** The timestamp is always stored as `datetime.datetime`, but for convenience it is accepted as a string that is formatted accordingly.

**Parameters** **value** (`datetime.datetime` or `str`) – Optional timestamp to set.

**set\_created** (*value=None*)

Set created timestamp.

Sets created metadata attribute.

**Note** The timestamp is always stored as `datetime.datetime`, but for convenience it is accepted as a string that is formatted accordingly.

**Parameters** **value** (`datetime.datetime` or `str`) – Optional timestamp to set.

**set\_cmm\_type** (*value=None*)

Set cmm type.

**Parameters** **value** (*str*) – Optional type to set.

**set\_id** (*value*)

Set ID.

**Parameters** **value** (*str*) – id to set.

**get\_updated** ()

Get updated value.

**Note** The timestamp is stored as a `datetime.datetime` in `_metadata.attr_updated`, but is returned as a string datestamp when using this method. If there is need to access the `datetime.datetime` object, use `get_value()` of the field.

**Returns** updated timestamp.

**Return type** `str`

**get\_created** ()

Get created value.

**Note** The timestamp is stored as a `datetime.datetime` in `_metadata.attr_created`, but is returned as a string datestamp when using this method. If there is need to access the `datetime.datetime` object, use `get_value()` of the field.

**Returns** created timestamp.

**Return type** `str`

**get\_id** ()

Get record ID.

Id comes from the backend storage system.

**Returns** record ID in storage.

**Return type** `str` or `None`

**bypass\_update** (*\*fields*)

Add fields to be bypassed on update operation.

**Parameters** **\*fields** (*str*) – fieldnames to bypass.

**bypass\_create** (*\*fields*)

Add fields to be bypassed on create operation.

**Parameters** **\*fields** (*str*) – fieldnames to bypass.

**updates\_record** (*old\_record\_dict*)

Update record by appending old values that are not present in current record. Use old record's `_id` and `_metadata.created` if present.

**Note** parameter is a dictionary since MongoDB returns records as JSON-like objects, which in turn are best represented as dictionaries in python.

**Parameters** **old\_record\_dict** (*dict*) – Old record as a dictionary.

**updates** (*secondary\_record*)

Update record by appending values from secondary which are not present in this record.

**Parameters** **secondary\_record** (Record instance subclassed from `RecordBase`) – lookup values from this record.

**class** `kuha_common.document_store.records.Study` (*study\_dict=None*)  
Study record.

Derived from *RecordBase*. Used to store and manipulate Study records. Study number is a special attribute and it cannot be updated.

All attributes of the record are declared as class variables initiated from *kuha\_common.document\_store.field\_types.FieldTypeFactory*. Instance methods defined in this class are used to add/set values to record attributes. The signatures of the methods are dynamically constructed by the definition of the FieldTypeFactory instances. If, for example, there is a class variable definition:

```
animals = FieldTypeFactory('animals', 'animal', ['color', 'weight', 'height'])
```

The correct method signature should be:

```
def add_animals(self, value, language, color=None, weight=None, height=None):
```

For the dynamic nature of the record-model these signatures are left open, and python's `*args` and `**kwargs` are used instead. Note that the field type used will raise exceptions if keyword argument key is not found in the initial definition of the field type.

Create a new study record:

```
>>> study = Study()
>>> study.add_study_number(1234)
>>> study.add_study_titles('Study about animals', 'en')
>>> study.add_principal_investigators('investigator', 'en', organization='Big_
↳organization ltd.')
```

Import existing study record from dictionary:

```
>>> study_dict = {'study_number': 1234,
... 'study_titles': [{'study_title': 'Study about animals', 'language': 'en'}],
... 'principal_investigators': [{'principal_investigator': 'investigator',
... 'language': 'en', 'organization': 'Big organization ltd.'}]}
>>> study = Study(study_dict)
```

Iterate attributes:

```
>>> for pi in study.principal_investigators:
...     pi.attr_organization.get_value()
...
'Big organization ltd.'
```

**Seealso** *RecordBase* and *kuha\_common.document\_store.field\_types*

**Parameters** `study_dict` (*dict*) – Optional study record as dictionary used for constructing a record instance.

**study\_number** = <*kuha\_common.document\_store.field\_types.FieldTypeFactory* object>  
Study number is used to identify a study. It must be unique within records, not localizable and contain only a single value. It cannot be updated.

**persistent\_identifiers** = <*kuha\_common.document\_store.field\_types.FieldTypeFactory* object>  
Persistent identifiers. Multivalue-field with unique values.

**identifiers** = <*kuha\_common.document\_store.field\_types.FieldTypeFactory* object>  
Identifiers. Localizable field with agency-attribute. This needs to be localizable for the sake of agency-

attribute. Note that two identical identifiers with same locale cannot exist at same time. The latter agency will overwrite the former on update.

**study\_titles** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Study titles. Localizable, multivalued-field without attributes.

**parallel\_titles** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Parallel study titles. Localizable, multivalued-field without attributes.

**principal\_investigators** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Principal investigators. Localizable, multivalued-field with organization-attribute.

**publishers** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Publishers. Localizable, multivalued-field with abbreviation-attribute.

**distributors** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Distributors. Localizable, multivalued-field with abbreviation and uri attributes.

**document\_uris** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Document URIs. Localizable, multivalued-field with location and description attributes.

**publication\_dates** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Publication dates. Localizable, multivalued-field without attributes. Note that these are treated as strings, not datetime-objects.

**publication\_years** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Publication years. Localizable, multivalued-field with distribution date attribute.

**abstract** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Abstract. Localizable, multivalued-field.

**classifications** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Classifications. Localizable, multivalued-field with system name, uri and description attributes.

**keywords** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Keywords. Localizable, multivalued-field with system name, uri and description attributes.

**time\_methods** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Time methods. Localizable, multivalued-field with system name, uri and description attribute.

**sampling\_procedures** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Sampling procedures. Localizable, multivalued-field with description, system name and uri attributes.

**collection\_modes** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Collection modes. Localizable, multivalued-field with system name and uri attributes.

**analysis\_units** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Analysis units. Localizable, multivalued-field with system name, uri and description attributes.

**collection\_periods** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Collection periods. Localizable, multivalued-field with event-attribute.

**study\_area\_countries** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Study area countries. Localizable, multivalued-field with abbreviation attribute.

**universes** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Universes. Localizable, multivalued-field with included attribute.

**data\_access** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Data access. Localizable, multivalued-field.

**data\_access\_descriptions** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Data access descriptions. Localizable, multivalued-field.



**file\_names** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
File names. Localizable, multivalue-field.

**instruments** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Instruments. Localizable, multivalue-field with instrument name attribute.

**study\_groups** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Study groups. Localizable, multivalue-field with name attribute.

**copyrights** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
Copyrights. Localizable, multivalue-field.

**collection** = 'studies'  
Database collection (table) for persistent storage.

**cmm\_type** = 'study'  
CMM type for Study.

**add\_study\_number** (*value*)  
Add study number.

**Note** despite the name, the value does not need to be a number.

**Parameters** *value* (*str* or *int*) – study number.

**add\_persistent\_identifiers** (*value*)  
Add persistent identifiers

**Parameters** *value* (*str* or *int*) – persistent identifier

**add\_identifiers** (*value*, \**args*, \*\**kwargs*)  
Add identifiers.

**Parameters**

- **value** (*str* or *int*) – identifier
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_study\_titles** (*value*, \**args*, \*\**kwargs*)  
Add study titles.

**Parameters**

- **value** (*str*) – study title.
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_parallel\_titles** (*value*, \**args*, \*\**kwargs*)  
Add parallel titles.

**Parameters**

- **value** (*str*) – title.
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_principal\_investigators** (*value*, \*args, \*\*kwargs)

Add principal investigators.

**Parameters**

- **value** (*str*) – investigators.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_publishers** (*value*, \*args, \*\*kwargs)

Add publishers.

**Parameters**

- **value** (*str*) – publishers.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_distributors** (*value*, \*args, \*\*kwargs)

Add distributors.

**Parameters**

- **value** (*str*) – distributor.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_document\_uris** (*value*, \*args, \*\*kwargs)

Add document URIs.

**Parameters**

- **value** (*str*) – URI.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_publication\_dates** (*value*, \*args, \*\*kwargs)

Add publication dates.

**Parameters**

- **value** (*str*) – date.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_publication\_years** (*value*, \*args, \*\*kwargs)

Add publication dates.

#### Parameters

- **value** (*str*) – date.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_abstract** (*value*, \*args, \*\*kwargs)

Add abstract.

#### Parameters

- **value** (*str*) – abstract.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_classifications** (*value*, \*args, \*\*kwargs)

Add classifications.

#### Parameters

- **value** (*str*) – classifications.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_keywords** (*value*, \*args, \*\*kwargs)

Add keywords.

#### Parameters

- **value** (*str*) – keyword.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_time\_methods** (*value*, \*args, \*\*kwargs)

Add time methods.

#### Parameters

- **value** (*str*) – time method
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_sampling\_procedures** (*value*, \**args*, \*\**kwargs*)

Add sampling procedures

**Parameters**

- **value** (*str*) – sampling procedure
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_collection\_modes** (*value*, \**args*, \*\**kwargs*)

Add collection modes

**Parameters**

- **value** (*str*) – collection mode
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_analysis\_units** (*value*, \**args*, \*\**kwargs*)

Add analysis units.

**Parameters**

- **value** (*str*) – analysis unit.
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_collection\_periods** (*value*, \**args*, \*\**kwargs*)

Add collection periods.

**Parameters**

- **value** (*str*) – collection period.
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_study\_area\_countries** (*value*, \**args*, \*\**kwargs*)

Add study area countries.

**Parameters**

- **value** (*str*) – country.
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_universes** (*value*, \*args, \*\*kwargs)

Add universes.

#### Parameters

- **value** (*str*) – universe.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_data\_access** (*value*, \*args, \*\*kwargs)

Add data access.

#### Parameters

- **value** (*str*) – data access.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_data\_access\_descriptions** (*value*, \*args, \*\*kwargs)

Add data access descriptions.

#### Parameters

- **value** (*str*) – access description.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_file\_names** (*value*, \*args, \*\*kwargs)

Add file name.

#### Parameters

- **value** (*str*) – file name.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_instruments** (*value*, \*args, \*\*kwargs)

Add instrument.

#### Parameters

- **value** (*str*) – instrument.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_study\_groups** (*value*, \*args, \*\*kwargs)

Add study group.

**Parameters**

- **value** (*str*) – study group.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_copyrights** (*value*, \*args, \*\*kwargs)

Add copyright.

**Parameters**

- **value** (*str*) – copyright.
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**updates** (*secondary*)

Check that records have common unique keys. Update record by appending values from secondary which are not present in this record.

**Parameters** **secondary** (*Study*) – Lookup values to update from secondary record.

**Returns** True if record updated, False if not.

**Return type** `bool`

**class** `kuha_common.document_store.records.Variable` (*variable\_dict=None*)

Variable record.

Derived from `RecordBase`. Used to store and manipulate variable records. Study number and variable name are special attributes and cannot be updated.

**Seealso** `Study` documentation for more information.

**Parameters** **variable\_dict** (*dict*) – Optional variable record as dictionary used for constructing a record instance.

**study\_number** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`

Study number and variable name are used to identify a variable within variable records. Their combination must be unique withing variable records, they cannot be localizable, and they can only contain a single value. They also cannot be updated.

**variable\_name** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`

Variable name within a study. See also `study_number`

**question\_identifiers** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`

Question identifiers, if variable refers to a question. Not localizable, multiple unique values.

**variable\_labels** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`

Variable labels. Localizable, multivalue-field.

**codelist\_codes** = <kuha\_common.document\_store.field\_types.FieldTypeFactory object>  
 Codelist codes. Localizable, multivalue-field with label and missing attributes.

**collection** = 'variables'  
 Database collection for persistent storage.

**cmm\_type** = 'variable'  
 CMM type for variable.

**add\_study\_number** (*value*)  
 Add study number.

**Parameters** **value** (*str* or *int.*) – study number.

**add\_variable\_name** (*value*)  
 Add variable name.

**Parameters** **value** (*str*) – variable name.

**add\_question\_identifiers** (*value*)  
 Add question identifier

**Parameters** **value** (*str* or *int.*) – question identifier.

**add\_variable\_labels** (*value*, \**args*, \*\**kwargs*)  
 Add variable label

**Parameters**

- **value** (*str*) – variable label.
- \***args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- \*\***kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_codelist\_codes** (*value*, \**args*, \*\**kwargs*)  
 Add codelist code

**Parameters**

- **value** (*str*) – codelist code.
- \***args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- \*\***kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**updates** (*secondary*)

Check that records have common unique keys. Update record by appending values from secondary which are not present in this record.

**Parameters** **secondary** (*Variable*) – Lookup values to update from secondary record.

**Returns** True if record updated, False if not.

**Return type** `bool`

**class** `kuha_common.document_store.records.Question` (*question\_dict=None*)  
 Question record.

Derived from *RecordBase*. Used to store and manipulate question records. *study\_number* and *question\_idenntifier* are special attributes and cannot be updated.

**Seealso** *Study* documentation for more information.

**Parameters** `question_dict` (*dict*) – Optional question record as dictionary used for constructing a record instance.

**study\_number** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
 Study number and question identifier are used to identify a question. Their combination must be unique withing records, they must not be localizable and they can only contain a single value. They also cannot be updated.

**question\_identifier** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
 Question identifier within a study. See also *study\_number*

**variable\_name** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
 Variable name that specifies the variable for the question. Not localizable, single value.

**question\_texts** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
 Question texts. Localizable, multivalue-field.

**research\_instruments** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
 Research instruments. Localizable, multivalue-field.

**codelist\_references** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
 Codelist references. Localizable, multivalue-field.

**collection** = `'questions'`  
 Database collection for persistent storage.

**cmm\_type** = `'question'`  
 CMM type for question

**add\_study\_number** (*value*)  
 Add study number.

**Parameters** `value` (*str* or *int.*) – study number.

**add\_question\_identifier** (*value*)  
 Add question identifier

**Parameters** `value` (*str* or *int.*) – question identifier.

**add\_variable\_name** (*value*)  
 Add variable name.

**Parameters** `value` (*str*) – variable name.

**add\_question\_texts** (*value*, *\*args*, *\*\*kwargs*)  
 Add question text

**Parameters**

- **value** (*str*) – question text.
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_research\_instruments** (*value*, *\*args*, *\*\*kwargs*)  
 Add research instrument

**Parameters**

- **value** (*str*) – research instrument.



- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**add\_codelist\_references** (*value*, *\*args*, *\*\*kwargs*)

Add codelist reference

#### Parameters

- **value** (*str*) – codelist reference
- **\*args** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`
- **\*\*kwargs** – defined by the parameters given to `kuha_common.document_store.field_types.FieldTypeFactory`

**updates** (*secondary*)

Check that records have common unique keys. Update record by appending values from secondary which are not present in this record.

**Parameters** **secondary** (*Question*) – Lookup values to update from secondary record.

**Returns** True if record updated, False if not.

**Return type** `bool`

**class** `kuha_common.document_store.records.StudyGroup` (*study\_group\_dict=None*)

Study group record.

Derived from `RecordBase`. Used to store and manipulate study group records. `study_group_identifier` is a special attribute and cannot be updated.

**Seealso** `Study` documentation for more information.

**Parameters** **study\_group\_dict** (*dict*) – Optional study group record as dictionary used for constructing a record instance.

**study\_group\_identifier** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
Study group identifier. Used to identify study group. Must be unique within study groups, cannot be localizable and can contain only a single value. This value cannot be updated.

**study\_group\_names** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
Study group names. Localizable, multivalue-field.

**study\_numbers** = `<kuha_common.document_store.field_types.FieldTypeFactory object>`  
Study numbers. Multivalue-field with unique values.

**collection** = `'study_groups'`  
Database collection for persistent storage.

**cmm\_type** = `'study_group'`  
CMM type for study groups.

**add\_study\_group\_identifier** (*value*)  
Add study group identifier.

**Parameters** **value** (*str or int*) – Study group identifier.

**add\_study\_group\_names** (*value*, *\*args*, *\*\*kwargs*)  
Add study group names

#### Parameters

- **value** (*str*) – study group name.
- **\*args** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*
- **\*\*kwargs** – defined by the parameters given to *kuha\_common.document\_store.field\_types.FieldTypeFactory*

**add\_study\_numbers** (*value*)

Add study number.

**Parameters** **value** (*str* or *int*) – study number.

**updates** (*secondary*)

Check that records have common unique keys. Update record by appending values from secondary which are not present in this record.

**Parameters** **secondary** (*StudyGroup*) – Lookup values to update from secondary record.

**Returns** True if record updated, False if not.

**Return type** *bool*

*kuha\_common.document\_store.records.record\_factory* (*ds\_record\_dict*)

Dynamically construct record instance based on given document store dictionary.

Looks up the correct record by the cmm type found from *ds\_record\_dict* metadata.

**Parameters** **ds\_record\_dict** (*dict*) – record received from Document Store.

**Returns** Record instance.

**Return type** *Study* or *Variable* or *Question* or *StudyGroup*

*kuha\_common.document\_store.records.record\_by\_collection* (*collection*)

Finds a record class by the given collection.

**Parameters** **collection** (*str*) – collection of the record.

**Returns** record class

**Return type** *Study* or *Variable* or *Question* or *StudyGroup*

**Raises** *KeyError* if collection is not found in any record.

## document\_store/mappings

### document\_store/mappings/exceptions.py

Exceptions for mapping package.

**exception** *kuha\_common.document\_store.mappings.exceptions.InvalidMapperParameters*

Raise for invalid configuration of a mapper - Coding error.

For example trying to add attributes to a mapper which does not support attributes. These errors are coding errors and should be treated as such.

**exception** *kuha\_common.document\_store.mappings.exceptions.MappingError*

Raise for errors while mapping input - User error.

Subclass to create more precise error classes.

**exception** *kuha\_common.document\_store.mappings.exceptions.ParseError*

Unable to parse source XML.

**Note** Mask ElementTree ParseError so caller may use MappingError to catch all user-errors when mapping.

**exception** `kuha_common.document_store.mappings.exceptions.UnknownXMLRoot` (*expected=None, un-known=None*)  
Unknown root element.

**exception** `kuha_common.document_store.mappings.exceptions.MissingRequiredAttribute` (*\*xpath, msg=None*)  
Source does not contain a required attribute.

**exception** `kuha_common.document_store.mappings.exceptions.InvalidContent`  
Attribute found but contains invalid data.

### document\_store/mappings/xmlbase.py

Components to use for XML parsing & mapping to Document Store records.

Contains a base class to use for parsing XML to Document Store records. Provides common functions useful in parsing XML data.

**class** `kuha_common.document_store.mappings.xmlbase.MappedParams` (*value*)  
Contains parameters ready to pass to record's add-methods.

*XMLMapper* retrieves parameters from XML record and stores them in an instance of this class. The record instances add-methods get called with stored parameters by using tuple and dict unpacking.

Example:

```
mapped_params = MappedParams('study_identifier')
mapped_params.set_language('en')
mapped_params.keyword_arguments.update({'agency': 'archive'})
study = Study()
study.add_identifiers(*mapped_params.arguments, **mapped_params.keyword_arguments)
```

**Parameters** `value` (*str or None*) – value used as the first argument

**has\_language** ()

True if MappedParams has language argument

**Returns** True if has language, False if not.

**Return type** `bool`

**set\_language** (*language*)

Set language argument. Will overwrite if previously set.

**Parameters** `language` (*str*) – Language to set.

**get\_language** ()

Get language argument.

**Returns** Language

**Return type** `str`

**get\_value** ()

Get value argument.

**Returns** value.

**Return type** `str` or `None`

**copy ()**

Make a copy of the object with contents and return the copy.

**Returns** copy of this *MappedParams*

**Return type** *MappedParams*

**has\_arguments ()**

Return True if *MappedParams* has arguments or keyword\_arguments.

**Returns** True if object has arguments or keyword\_arguments.

**Return type** `bool`

**class** `kuha_common.document_store.mappings.xmlbase.XMLMapper` (*xpath*,  
*from\_attribute=None*,  
*required=False*,  
*localizable=True*)

XMLMapper populates *MappedParams* instances from XML.

**Parameters**

- **xpath** (*str*) – XPath where to look for element containing value.
- **from\_attribute** (*str* or *None*) – Look value from attribute of the element.
- **required** (*bool*) – raises `MissingRequiredAttribute` if value is not found.
- **localizable** (*bool*) – True if the value is localizable, False if not.

**set\_value\_conversion** (*conv\_func*)

Set conversion callable.

**Note** *conv\_func* must accept a string or `None` as a parameter and return the converted value.

**Parameters** **conv\_func** (*callable.*) – Callable used for conversion.

**Returns** `self`

**set\_value\_getter** (*getter\_func*)

Set value getter callable.

**Note** *getter\_func* must accept an XML element `xml.etree.ElementTree.Element` as a parameter and return the value.

**Parameters** **getter\_func** (*callable.*) – Callable used for getting a value from XML element.

**Returns** `self`

**expect\_single\_value ()**

This mapper will be expected to return a single value.

**Returns** `self`

**expect\_multiple\_values ()**

This mapper will be expected to return multiple values.

**Returns** `self`

**disable\_attributes ()**

This mapper will not contain any attributes.

**Returns** `self`

**iterate\_attributes** (\*relations)

Iterate attributes to map.

**Parameters** \*relations (*str*) – optional parameters to iterate only attributes with a certain relation.

**Returns** A generator yielding tuples of each attribute in the format: (attribute\_name, attribute\_mapper, attribute\_provides\_main\_lang)

**Return type** generator

**as\_params** (element, default\_language, xml\_namespaces)

Use mapping to construct a *MappedParams* from XML element.

Use mapper's `_value_getter` and `_value_conversion` to get value from XML element. Construct a *MappedParams* from the value. If mapping `localizable` is `True` add language from XML elements `xml:lang` attribute.

**Parameters**

- **element** (`xml.etree.ElementTree.Element`) – XML element.
- **default\_language** (*str*) – default language if element has none.
- **xml\_namespaces** (*dict*) – XML namespaces for the element.

**Returns** mapped parameters ready to pass to records add-method.

**Return type** *MappedParams*

**add\_attribute** (att\_name, mapper, relative=True, provides\_main\_lang=False)

Add attribute to mapper.

Counts the correct xpath if attribute's mapper's xpath is a parent element (starting with '/..'). Includes all needed information of the attribute to a list of tuples contained in `attributes`.

**Parameters**

- **att\_name** (*str*) – attribute name
- **mapper** (*XMLMapper*) – mapper instance for mapping value for the attribute.
- **relative** (*bool*) – Is the attribute map's xpath relative to this element. Defaults to `True`.
- **provides\_main\_lang** (*bool*) – Should the language of the attribute be used as a language when mapping this value. Defaults to `False`.

**Raises** `InvalidMapperParameters` for conflicting parameters such as: 1. Calling this method on a mapper which has disabled use of attributes. 2. Using `provides_main_lang` for a non-localizable mapper. 3. Setting `relative` to `False` on a mapper whose `xpath` refers to parent element.

**Returns** `self`

**value\_params** (source\_xml\_element, default\_language, xml\_namespaces, position=None)

Generate single *MappedParams* object from source XML.

**Parameters**

- **source\_xml\_element** (`xml.etree.ElementTree.Element`) – XML element.
- **default\_language** (*str*) – Default language.
- **xml\_namespaces** (*dict*) – XML namespaces.
- **position** (*int* or *None*) – Optional position for parent xpaths.

**Returns** generator yielding *MappedParams*.

**Return type** generator

**Raises** `MissingRequiredAttribute` if mapper's `required` is `True`, but `xpath` provides no element or the element provides no value.

**values\_params** (*source\_xml\_element, default\_language, xml\_namespaces*)

Generate multiple *MappedParams* objects from source XML.

The generated *MappedParams* will contain attributes as `keyword_arguments`.

**Parameters**

- **source\_xml\_element** (`xml.etree.ElementTree.Element`) – XML element.
- **default\_language** (*str*) – Default language.
- **xml\_namespaces** (*dict*) – XML namespaces.

**Returns** generator yielding *MappedParams*.

**Return type** generator

**Raises** `MissingRequiredAttribute` if mapper's `required` is `True`, but `xpath` provides no element or the element provides no value.

**class** `kuha_common.document_store.mappings.xmlbase.XMLParserBase` (*root\_element*)

Base class where parsers get derived from.

Declares the public API to be used in callers.

**Input:**

- `from_file()`
- `from_string()`

**Output:**

- `studies`
- `variables`
- `questions`
- `study_groups`
- `all`
- `select(collection=None)`

Provides common functionality to be used within subclasses which map XML-data to Document Store records. Subclasses must implement necessary generators that generate document store records.

Use in subclass:

```
class XMLRecordParser(XMLParserBase):
    @property
    def studies(self):
        maps = [(Study.add_study_number, self._map_single(xpath_to_study_number,
↳required=True)),
                (Study.add_study_titles, self._map_multi(xpath_to_study_title))]
        for study_element in self.root_element.findall(xpath_to_study_element,
↳self.NS):
            study = Study()
```

```
self._map_to_record(study, study_element, maps)
yield study
```

**Parameters** `root_element` (`xml.etree.ElementTree.Element`) – XML root.

`NS = {'xsi': 'http://www.w3.org/2001/XMLSchema-instance', 'xml': 'http://www.w3.org/'}
XML namespaces`

`default_language = 'en'`  
Default language.

**classmethod** `from_string` (`xml_body`)  
Get parser that iteratively parses XML and generates populated Document Store record instances.

**Parameters** `xml_body` (`str`) – XML Document as a string. This may come directly from HTTP request body.

**Returns** parser for iteratively parsing XML and generating Document Store records.

**Return type** `XMLParserBase`

**classmethod** `from_file` (`filepath`)  
Get parser that iteratively parses XML and generates populated Document Store record instances.

**Parameters** `filepath` (`str`) – Path for the XML file.

**Returns** parser for iteratively parsing XML and generating Document Store records.

**Return type** `XMLParserBase`

**classmethod** `child_text` (`xpath`)  
Returns a function which will lookup a child element from given xpath. The returned function takes a single element as a parameter which should be an `xml.etree.ElementTree.Element` or similar. When executed the function returns the child element's text contents or None if child element cannot be found.

**Parameters** `xpath` – xpath to child. relative to parent.

**Returns** function which accepts the parent element as a parameter.

**Return type** function

**root\_element**  
Get root element.

**Returns** Root element

**Return type** `xml.etree.ElementTree.Element`

**root\_language**  
Get language of the root element. If root does not have a language, returns `self.default_language`.

**Returns** root element language.

**Return type** `str`

**study\_number**  
Get study number as formatted in source XML.

**Seealso** `self.study_number_identifier`

**Returns** Study number from source XML.

**Return type** `str`

**study\_number\_identifier**

Get study number converted as a valid Document Store identifier.

**Returns** Study number as valid Document Store identifier.

**Return type** `str`

**studies**

Studies generator. Must be implemented in subclass.

**Returns** Generator which yields Document Store studies.

**variables**

Variables generator. Must be implemented in subclass.

**Returns** Generator which yields Document Store variables.

**questions**

Questions generator. Must be implemented in subclass.

**Returns** Generator which yields Document Store questions.

**study\_groups**

Study groups generator. Must be implemented in subclass.

**Returns** Generator which yields Document Store study groups.

**all**

Iterate all records found from source XML.

**Returns** Generator which yields Document Store records.

**Return type** Generator

**select** (*collection=None*)

Returns a selective parser. Call with a Document Store collection as parameter to select records only for certain collection.

---

**Note:** The returned attributes are defined in subclasses, so they may or may not be generators.

---

**Parameters** **collection** (*str or None*) – Document Store collection to select only records belonging to this collection.

**Returns** Generator which yields Document Store records.

**Return type** Generator

`kuha_common.document_store.mappings.xmlbase.as_valid_identifier(candidate)`

Convert candidate to a string that conforms the rules of validation.

Identifier must match regex: `[a-zA-Z0-9]+[a-zA-Z0-9?_(-).]*`

---

**Note:** Regex is defined in Document Store. Should it be moved to `kuha_common`?

---

**Returns** identifier which conforms the rules of validation.

**Return type** `str`



`kuha_common.document_store.mappings.xmlbase.str_equals` (*correct*, *default=None*)  
Conversion function wrapper to compare strings for equality.

Wrapper function that formats comparison value and default value for returned comparison function.

Check if string found from element value or element attribute equals to *correct*.

**Parameters**

- **correct** (*str*) – comparison string.
- **default** (*str*) – If the value parameter of the comparison function is None, return this value.

**Returns** function which accepts a single parameter for comparison. Returns True or False, or *default* if the parameter is None.

**Return type** function

`kuha_common.document_store.mappings.xmlbase.fixed_value` (*fixed*)  
Fixed value.

**Parameters** **fixed** – Use this value

**Returns** function which accepts a single argument value. The function always returns fixed.

**Return type** function

`kuha_common.document_store.mappings.xmlbase.element_remove_whitespace` (*element*)  
Conversion function to remove extra whitespace from end of element text.

Iterates element's inner text using `xml.etree.ElementTree.Element.itertext()` which iterates over this element and all subelements. Removes extra whitespaces so paragraphs of text will only have one separating whitespace character.

**Parameters** **element** (`xml.etree.ElementTree.Element`) – Element from which to get text.

**Returns** Element's inner text without extra whitespace.

**Return type** `str`

`kuha_common.document_store.mappings.xmlbase.element_strip_descendant_text` (*element*)  
Conversion function to remove inner elements and their contents.

**Parameters** **element** (`xml.etree.ElementTree.Element`) – Element for lookup.

**Returns** Element's inner text without text from descendants and without extra whitespace.

**Return type** `str`

## document\_store/mappings/ddi.py

Mapping profiles for DDI.

---

**Note:** has strict dependency to `kuha_common.document_store.records`

---

**class** `kuha_common.document_store.mappings.ddi.DDI122RecordParser` (*root\_element*)  
Parse Document Store records from DDI 1.2.2. XML.

**studies**

Parse XML to create and populate `kuha_common.document_store.records.Study`.

**Returns** Generator to Populate Document Store Study record.

**variables**

Parse XML to create and populate multiple *kuha\_common.document\_store.records.Variable* instances.

**Returns** Generator to populate multiple Document Store Variable records.

**questions**

Parse XML to create and populate multiple *kuha\_common.document\_store.records.Question* instances.

**Returns** Generator to populate multiple Document Store Question records.

**study\_groups**

Parse XML to create and populate multiple *kuha\_common.document\_store.records.StudyGroup* instances.

**Returns** Generator to populate multiple Document Store StudyGroup records.

**class** *kuha\_common.document\_store.mappings.ddi.DDI25RecordParser* (*root\_element*)

Parse Document Store records from DDI 2.5 XML.

**NS** = {'xsi': 'http://www.w3.org/2001/XMLSchema-instance', 'ddi': 'ddi:codebook:2\_5',  
XML namespaces

**studies**

Parse XML to create and populate *kuha\_common.document\_store.records.Study*.

**Returns** Generator to Populate Document Store Study record.

**variables**

Parse XML to create and populate multiple *kuha\_common.document\_store.records.Variable* instances.

**Returns** Generator to populate multiple Document Store Variable records.

**questions**

Parse XML to create and populate multiple *kuha\_common.document\_store.records.Question* instances.

**Returns** Generator to populate multiple Document Store Question records.

**study\_groups**

Parse XML to create and populate multiple *kuha\_common.document\_store.records.StudyGroup* instances.

**Returns** Generator to populate multiple Document Store StudyGroup records.

**class** *kuha\_common.document\_store.mappings.ddi.DDI31RecordParser* (*root\_element*)

Parse Document Store records from DDI 3.1. XML

Check the root element. Expects either ddi:DDIInstance or s:StudyUnit. Currently supports only single s:StudyUnit element within the root.

**Parameters** *root\_element* (*xml.etree.ElementTree.Element*) – XML root element.

**Raises** *UnknownXMLRoot* for unexpected root element.

**Raises** *MappingError* if root contains more or less that exactly one s:StudyUnit child.

**NS** = {'a': 'ddi:archive:3\_1', 'pd': 'ddi:physicaldataproduct:3\_1', 'r': 'ddi:reusab',  
XML namespaces

**studies**

Parse XML to create and populate `kuha_common.document_store.records.Study`.

**Returns** Generator to Populate Document Store Study record.

**variables**

Parse XML to create and populate `kuha_common.document_store.records.Variable`.

**Returns** Generator to Populate Document Store Variable records.

**questions**

Parse XML to create and populate `kuha_common.document_store.records.Question`.

**Returns** Generator to Populate Document Store Question records.

**study\_groups**

Parse XML to create and populate `kuha_common.document_store.records.StudyGroup`.

**Returns** Generator to Populate Document Store StudyGroup records.

**testing**

Package for common testing functions and classes.

`kuha_common.testing.time_me` (*func*)

Decorate function to print its execution time to stdout.

**Note** test runner may capture the output.

**Parameters** `func` – Function to decorate. Count execution time of function.

**class** `kuha_common.testing.MockCoro` (*dummy\_rval=None, func=None*)

Mock out a coroutine function.

Mock out coroutine and set return value:

```
>>> mock_coro = MockCoro('return_value')
>>> rval = await mock_coro()
>>> assert rval == 'return_value'
```

Mock out coroutine with custom function:

```
>>> async def custom_func(mock_coro, callback):
>>>     callback(mock_coro.dummy_rval)
>>> mock_coro = MockCoro('return_value', custom_func)
>>> await mock_coro()
```

Use as a `side_effect` when patching:

```
>>> @mock.patch.object(pkg.Class, 'async_method', side_effect=MockCoro())
>>> def test_something(mock_method):
>>>     something_calls_async_method()
>>>     mock_method.assert_called_once_with()
```

**Parameters**

- **dummy\_rval** – return value of dummy function.
- **func** – function to call instead of original function.

**dummy** (\*args, \*\*kwargs)  
Dummy coroutine function which returns self.dummy\_rval  
**Returns** self.dummy\_rval

## testing/testcases.py

Test cases for Kuha

**class** kuha\_common.testing.testcases.**KuhaUnitTestCase** (methodName='runTest')  
Base class for unittests.

- Assertion methods to check record equality.
- Helper methods to provide access to dummydata.

**dummydata\_dir** = '/home/docs/checkouts/readthedocs.org/user\_builds/kuha2/envs/latest/src'  
Override in subclass to lookup dummydata from different directory.

**classmethod** **get\_dummydata\_path** (path)  
Get absolute path to dummydatafile  
**Parameters** path – Path. Gets turned into an absolute if it isn't  
**Returns** Absolute path.  
**Return type** str

**classmethod** **get\_dummydata** (path)  
Get dummydata by reading file from path  
**Parameters** path – path to file.  
**Returns** Contents of the file.

**classmethod** **remove\_dummyfile\_if\_exists** (path)  
Remove dummyfile from path if it exists.  
**Parameters** path – path to dummyfile.  
**Returns** None

**classmethod** **set\_val** (value)  
Assign value as dummyvalue.  
**Parameters** value – Value to assign  
**Returns** value

**classmethod** **gen\_val** (length=None, unique=False, chars=None)  
Generate & assign dummyvalue.  
**Parameters**

- **length** (int or None) – length of the value
- **unique** (bool) – should the value be unique
- **chars** (str or None.) – use specific characters.

**Returns** generated value  
**Return type** str

**classmethod** **gen\_id** ()  
Generate Id.

**Returns** Generated id.

**Return type** `str`

**classmethod** `generate_dummy_study()`

Generate and return a Study with dummydata.

**Returns** study with dummydata

**Return type** `kuha_common.document_store.records.Study`

**classmethod** `generate_dummy_variable()`

Generate and return a Variable with dummydata.

**Returns** variable with dummydata

**Return type** `kuha_common.document_store.records.Variable`

**classmethod** `generate_dummy_question()`

Generate and return a Question with dummydata.

**Returns** question with dummydata

**Return type** `kuha_common.document_store.records.Question`

**classmethod** `generate_dummy_studygroup()`

Generate and return a StudyGroup with dummydata.

**Returns** studygroup with dummydata.

**Return type** `kuha_common.document_store.records.StudyGroup`

**setUp()**

Format testcase values and initialize event loop.

Call asynchronous code synchronously:

```
self._loop.run_until_complete(coro())
```

**tearDown()**

Stop patchers.

**await\_and\_store\_result(coro)**

Await coroutine and store returning result.

Example:

```
self._loop.run_until_complete(self.await_future_and_store_result(coro()))
```

**Parameters** `coro` – Coroutine or Future to await

**init\_patcher(patcher)**

Initialize patcher, store for later use, return it.

**Parameters** `patcher` (`unittest.mock._patch`) – Patch to start.

**Returns** MagicMock acting as patched object.

**Return type** `unittest.mock.MagicMock`

**assert\_records\_are\_equal(first, second, msg=None)**

Assert two Document Store records are equal.

**Parameters**

- **first** – First record to compare.
- **second** – Second record to compare.
- **msg** – Optional message to output on assertion.

**assert\_records\_are\_not\_equal** (*first, second, msg=None*)

Assert two Document Store records are not equal.

**Parameters**

- **first** – First record to compare.
- **second** – Second record to compare.
- **msg** – Optional message to output on assertion.

**class** `kuha_common.testing.testcases.KuhaEndToEndTestCase` (*methodName='runTest'*)

Base class for end-to-end tests.

- HTTPClient for interacting with Document Store.
- Assertion methods to check returning payload and status codes.

**static load\_cli\_args** (*sysexit\_to\_skiptest=False*)

Load command line arguments. Setup Document Store URL.

**Parameters** **sysexit\_to\_skiptest** (*bool*) – Mask SystemExit as `unittest.SkipTest`. Useful when missing command line arguments should not terminate the test run, but skip tests requiring the arguments.

**Returns** arguments not known to `kuha_common.cli_setup.settings` (= arguments external to Kuha)

**Return type** `list`

**static get\_record\_url** (*rec\_or\_coll, \_id=None*)

Get URL to Document Store records or single record.

**Parameters**

- **rec\_or\_coll** – record, record class or collection
- **\_id** (*str or None*) – Optional record ID.

**Returns** URL to Document Store collection or single record.

**Return type** `str`

**static get\_query\_url** (*rec\_or\_coll, query\_type=None*)

Get URL to Document Store query endpoint for collection

**Parameters**

- **rec\_or\_coll** (*str, record, or record class*) – Collection to query.
- **query\_type** – Optional query type

**Returns** URL to query endpoint.

**Return type** `str`

**classmethod GET\_to\_document\_store** (*rec\_or\_coll, \_id=None*)

GET to Document Store returns record(s).

**Parameters**

- **rec\_or\_coll** – record or collection to get.

- `_id` – Optional ObjectId. Will take precedence over `rec_or_coll` id.

**Returns** response body

**classmethod** `POST_to_document_store` (*record*)

POST to Document Store creates record.

**Parameters** `record` – Record to post.

**Returns** response body

**classmethod** `DELETE_to_document_store` (*rec\_or\_coll=None, \_id=None*)

DELETE to Document Store deletes record(s).

Call without arguments to delete all records from all collections.

**Parameters**

- `rec_or_coll` (*str or None*) – Collection to delete from.
- `_id` (*str or None*) – ID of the record to delete.

**Returns** None

**classmethod** `query_document_store` (*rec\_or\_coll, query, query\_type=None*)

Execute query against Document Store query API.

**Parameters**

- `rec_or_coll` (*str or record class or record instance*) – Collection to query.
- `query` – Query.
- `query_type` – Type of Query.

**Returns** query results

**Return type** None if query returned no results, dict for results.

**classmethod** `get_collection_record_count` (*rec\_or\_coll*)

Return number of records for collection in Document Store.

**Parameters** `rec_or_coll` – Document Store record, Document Store record class or collection.

**Returns** record count in Document Store.

**Return type** `int`

**assert\_document\_store\_is\_empty** ()

Assert Document Store contains no records.

**Raises** `AssertionError` if Document Store has records.

### 3.8.2 kuha\_document\_store

Kuha Document Store application

Query, manipulate and import Document Store records via HTTP API.

## configure.py

Configure Document Store.

`kuha_document_store.configure.add_database_configs()`

Add database configuration values to be parsed.

`kuha_document_store.configure.configure()`

Get settings for application configuration.

Declares application specific configuration options and some common options declared in `kuha_common.cli_setup`

Configure application with arguments specified in configuration file, environment variables and command line arguments.

**Note** Calling this function multiple times will not initiate new settings to be parsed, but will return previously parsed settings instead.

**Returns** settings

**Return type** `argparse.Namespace`

## serve.py

Main entry point for starting Document Store server.

`kuha_document_store.serve.get_app(api_version, app_settings=None)`

Setup routes and return initialized Tornado web application.

### Parameters

- **api\_version** (*str*) – HTTP Api version gets prepended to routes.
- **app\_settings** (*dict or None.*) – Settings to store to application.

**Returns** Tornado web application.

**Return type** `tornado.web.Application`

`kuha_document_store.serve.main()`

Application main function.

Parse commandline for settings. Initialize database and web application. Start serving via `kuha_common.server.serve()`. Exit on exceptions propagated at this level.

**Returns** exit code, 1 on error, 0 on success.

**Return type** `int`

## handlers.py

Define handlers for responding to HTTP-requests.

**class** `kuha_document_store.handlers.BaseHandler(*args, **kwargs)`

BaseHandler to derive from.

Provides common methods for subclasses.

**Note** use from a subclass



**prepare** ()

Prepare for each request.

Set output content type.

**get\_db** ()

Get database object stored in settings.

**Returns** database object.

**Return type** *kuha\_document\_store.database.DocumentStoreDatabase*

**assert\_body\_not\_empty** (*msg=None*)

Assert that request body contains data.

*kuha\_common.server.BadRequest* is raised if body is empty.

**Parameters** *msg* (*str*) – Optional message for exception.

**Raises** *kuha\_common.server.BadRequest* if body is empty.

**class** *kuha\_document\_store.handlers.RestApiHandler* (*\*args, \*\*kwargs*)

Handle requests to REST api.

**get** (*collection, resource\_id=None*)

HTTP-GET to REST api endpoint.

Respond with single record or multiple records, depending on whether *resource\_id* is requested.

**Note** Results will be streamed.

**Parameters**

- **collection** (*str*) – type of the requested collection.
- **resource\_id** (*str or None*) – optional ID of the requested resource. If left out of request, will return all records of requested type.

**Raises** *kuha\_common.server.BadRequest* if there are recoverable errors in database operation. The error message is passed to *BadRequest*. See: *kuha\_document\_store.database.DocumentStoreDatabase.recoverable\_errors*

**Raises** *kuha\_common.server.ResourceNotFound* if requested *resource\_id* does not return results.

**post** (*collection, resource\_id=None*)

HTTP-POST to REST api endpoint.

Create new resource from data submitted in request body.

**Parameters**

- **collection** (*str*) – collection type to create.
- **resource\_id** (*str or None*) – receives *resource\_id* for completeness in handler configuration. It is however a *kuha\_common.server.BadRequest* if one is submitted.

**Raises** *kuha\_common.server.BadRequest* if request contains *resource\_id* or if database operations raise recoverable errors. See: *kuha\_document\_store.database.DocumentStoreDatabase.recoverable\_errors*

**put** (*collection, resource\_id=None*)

HTTP-PUT to REST api endpoint.

Replace existing resource with data in request body.

### Parameters

- **collection** (*str*) – collection type to replace.
- **resource\_id** (*str* or *None*) – resource ID to replace. Optional for completeness in handler configuration. It is however a *kuha\_common.server.BadRequest* if not submitted.

**Raises** *kuha\_common.server.BadRequest* if requested endpoint does not contain *resource\_id* or if database operation raises one of *kuha\_document\_store.database.DocumentStoreDatabase.recoverable\_errors*

**Raises** *kuha\_common.server.ResourceNotFound* if *resource\_id* returns no results.

**delete** (*collection*, *resource\_id=None*)  
 HTTP-DELETE to REST api endpoint.

Delete resource or all resources of certain type.

### Parameters

- **collection** (*str*) – type of collection
- **resource\_id** (*str* or *None*) – resource ID to delete.

**Raises** *kuha\_common.server.BadRequest* if database operation raises one of *kuha\_document\_store.database.DocumentStoreDatabase.recoverable\_errors*

**Raises** *kuha\_common.server.ResourceNotFound* if *resource\_id* returns no results.

**class** *kuha\_document\_store.handlers.ImportHandler* (*\*args*, *\*\*kwargs*)  
 Handle request to import endpoint.

**prepare** ()  
 Prepare for each request.

All requests must define content type for XML. All requests must contain body data.

**post** (*importer\_id*, *collection=None*)  
 HTTP-POST to import endpoint.

Lookup correct importer. Load iterative parser. Pass iterative parser to database for processing.

### Parameters

- **importer\_id** (*str*) – importer to use for importing.
- **collection** (*str* or *None*) – Optional parameter limits the import to a specific collection (resource type).

**class** *kuha\_document\_store.handlers.QueryHandler* (*\*args*, *\*\*kwargs*)  
 Handle request to query endpoint.

**Note** Results will be streamed.

**prepare** ()  
 Prepare for each request.

Request content type must be JSON. Request body must not be empty. Requested query type must be supported and query must have valid parameters.

**post** (*collection*)  
 HTTP-POST to query endpoint.

Streams the results one JSON document at a time. Thus, the result of a response for multiple records will not a a valid JSON document.

**Note** Body must be a JSON object.

**Parameters** `collection` (*str*) – collection (resource type) to query.

## database.py

Database module provides access to MongoDB database.

MongoDB Database is accessed through this module. The module also provides convenience methods for easy access and manipulation via Document Store records defined in `kuha_common.document_store.records`

Database can be used directly, via records or with JSON representation of records.

**note** This module has strict dependency to `kuha_common.document_store.records`

```
class kuha_document_store.database.RecordsCollection (record_class,           in-
                                                    indexes_unique=None,         in-
                                                    indexes=None,                valida-
                                                    validators=None)
```

Database collection.

**Note** Relational Database term *table* is called a *collection* in MongoDB.

Contains properties for Document Store collections. Has strict dependency to `kuha_common.document_store.records`

### Parameters

- **record\_class** (`kuha_common.document_store.records.Study` or `kuha_common.document_store.records.Variable` or `kuha_common.document_store.records.Question` or `kuha_common.document_store.records.StudyGroup`) – Class of a record that belongs to this collection.
- **indexes\_unique** (*list* or *None*) – declare unique indexes.
- **indexes** (*list* or *None*) – additional indexes
- **validators** (*list* or *None*) – additional validators

**Returns** `RecordsCollection`

```
isodate_fields = ['_metadata.created', '_metadata.updated']
```

List common isodate fields

```
object_id_fields = ['_id']
```

Fields containing MongoDB ObjectIDs

```
index_updated = [('_metadata.updated', -1)]
```

Declare updated field as index.

```
classmethod bson_to_json (_dict)
```

Encode BSON dictionary to JSON.

Encodes special type of dictionary that comes from MongoDB queries to JSON representation. Also converts datetimes to strings.

**Parameters** `_dict` (*dict*) – Source object containing BSON.

**Returns** Source object converted to JSON.

**Return type** `str`

**get\_validator()**

Get defined database-level validators.

**Note** All validators are combined with AND operator.

**Returns** Database level validators to be used on DB setup.

**Return type** *dict*

**process\_json\_for\_upsert** (*json\_document*, *old\_metadata=None*)

Preprocess JSON for insert/update operations.

Decodes JSON to Python dictionary. Validates the result. Creates metadata for the document if the document has none, otherwise uses the submitted metadata. Decodes submitted metadata timestamps to date-time objects.

**Parameters**

- **json\_document** (*str*) – JSON representation of a record.
- **old\_metadata** (*dict or None*) – old metadata if updating existing record.

**Returns** Document ready to be submitted to database.

**Return type** *dict*

`kuha_document_store.database.RECORD_COLLECTIONS = [<kuha_document_store.database.RecordsCo`  
 Define Record Collections

**class** `kuha_document_store.database.Database` (*settings*)

MongoDB database.

Provides access to low-level database operations. For fine access control uses two database credentials, one for read-only operations, one for write operations. Chooses the correct credentials to authenticate based on the operation to be performed.

**Note** Does not authenticate or connect to the database before actually performing operations that need connecting. Therefore connection/authentication issues will raise when performing operations and not when initiating the database.

**Parameters** **settings** (*argparse.Namespace*) – settings for database connections

**Returns** *Database*

**close()**

Close open sockets to database.

**query\_single** (*collection\_name*, *query*, *fields=None*, *callback=None*)

Query for a single database document.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.
- **query** (*dict*) – Database query.
- **fields** (*list or None*) – Fields to select. None selects all.
- **callback** (*function or None*) – Result callback. Called with result as parameter. If None this method will return the result.

**Returns** A single document or None if no matching document is found. or if callback is given.

**Return type** *dict or None*

**query\_multiple** (*collection\_name*, *query*, *callback*, *fields=None*, *skip=0*, *sort\_by=None*,  
*sort\_order=1*, *limit=0*)  
 Query for multiple database documents.

**Note** has mandatory callback parameter.

#### Parameters

- **collection\_name** (*str*) – Name of database collection.
- **query** (*dict*) – Database query.
- **callback** (*Function that receives single record result as argument.*) – Result callback. Called with each document as parameter.
- **fields** (*list or None*) – Fields to select. None selects all.
- **skip** (*int*) – Skip documents from the beginning of query.
- **sort\_by** (*str*) – Sort by field.
- **sort\_order** (*int*) – Sort by ascending or descending order. MongoDB users 1 to sort ascending -1 to sort descending.
- **limit** (*int*) – Limit the number of returning documents. 0 returns all documents.

**query\_distinct** (*collection\_name*, *fieldname*, *filter\_=None*)  
 Query for distinct values in collection field.

#### Parameters

- **collection\_name** (*str*) – Name of database collection.
- **fieldname** (*str*) – Field to query for distinct values.
- **filter** (*dict or None*) – Optional filter to use with query.

**Returns** distinct values.

**Return type** *list*

**count** (*collection\_name*, *filter\_=None*)  
 Query for document count.

#### Parameters

- **collection\_name** (*str*) – Name of database collection.
- **filter** (*dict or None*) – Optional filter to use for query.

**Returns** Count of documents.

**Return type** *int*

**insert** (*collection\_name*, *document*)  
 Insert single document to database.

#### Parameters

- **collection\_name** (*str*) – Name of database collection.
- **document** (*dict*) – Document to insert.

**Returns** Insert result

**Return type** `pymongo.results.InsertOneResult`

**replace** (*collection\_name*, *oid*, *document*)  
 Replace single document in database.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.
- **oid** (*str*) – MongoDB object ID as string.
- **document** (*dict*) – Document to store.

**Returns** Update result.

**Return type** `pymongo.results.UpdateResult`

**insert\_or\_replace** (*collection\_name, query, document*)

Insert or replace a single document in database.

Uses special MongoDB method which will replace an existing document if one is found via query. Otherwise it will insert a new document.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.
- **query** (*dict*) – Database query.
- **document** (*dict*) – Document to store.

**Returns** The document that was stored.

**Return type** `dict`

**delete\_one** (*collection\_name, query*)

Delete single document.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.
- **query** (*dict*) – Database query.

**Returns** Delete result

**Return type** `pymongo.results.DeleteResult`

**delete\_many** (*collection\_name, query*)

Delete multiple documents.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.
- **query** (*dict*) – Database query.

**Returns** Delete result

**Return type** `pymongo.results.DeleteResult`

**class** `kuha_document_store.database.DocumentStoreDatabase` (*settings*)

Subclass of `Database`

Provides specialized methods extending the functionality of `Database`. Combines database operations with properties of `RecordsCollection`. Defines exceptions that, when raised, the HTTP-response operation can continue.

**recoverable\_errors** = (`<class 'pymongo.errors.WriteError'>`, `<class 'json.decoder.JSONDecodeError'>`)

These are exceptions that may be raised in normal database operation, so they are not exceptions that should terminate the HTTP-response process. As such, the caller may want to catch these errors.

**static json\_decode** (*json\_object*)

Helper method for converting HTTP input JSON to python dictionary.

**Parameters** **json\_object** (*str*) – json to convert.

**Returns** JSON object converted to python dictionary.

**Return type** *dict*

**query\_multiple** (*collection\_name, query, callback, \*\*kwargs*)

Query multiple documents with callback.

Converts resulting BSON to JSON. Calls callback with each resulting record JSON.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.
- **query** (*dict*) – Database query.
- **callback** (*function*) – Result callback. Called with each document as parameter.
- **\*\*kwargs** – additional keyword arguments passed to super method.

**query\_by\_oid** (*collection\_name, oid, callback, fields=None, not\_found\_exception=None*)

Query single record by ObjectID with callback.

Converts BSON result to JSON. Calls the callback with resulting JSON. If parameter for *not\_found\_exception* is given, will raise the exception if query ObjectID points to no known database object.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.
- **oid** (*str*) – ObjectID to query for.
- **callback** (*function*) – function to call with resulting JSON.
- **fields** (*list or None*) – Fields to select. None selects all.
- **not\_found\_exception** (*Exception class.*) – Raised if ObjectID not found.

**query\_distinct** (*collection\_name, fieldname, filter\_=None*)

Query for distinct values in collection field.

If *fieldname* points to a leaf node, returns a list of values, if it points to a branch node, returns a list of dictionaries.

If *fieldname* points to leaf node of isodate representations, or to branch node that contains isodates, converts datetimes to timestamps which are JSON serializable.

If ‘fieldname’ points to a leaf node containing MongoDB ObjectID values, cast those values to string.

**Note** Requires changes to logic if `collection.object_id_fields` should contain paths with multiple components, for example ‘some.path.with.id’. In that case distinct queries that point to branch nodes with OIDs will fail with Exception `TypeError: ObjectId(...)` is not JSON serializable.

**Note** Distinction will not work as expected on timestamp-fields that are stored as signed 64-bit integers with millisecond precision. The returned timestamps are not as precise since they have second precision.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.

- **fieldname** (*str*) – Field to query for distinct values.
- **filter** (*dict or None*) – Optional filter to use with query.

**Returns** distinct values from database

**Return type** *list*

**insert\_or\_update\_record** (*record*)

Insert or update database document by Document Store record.

Special method that takes a Document Store record instance as parameter and determines whether to insert or update the given record.

Makes a query to MongoDB to determine if the record is already in database. If there is a record, calls the record instance's `updates_record` method to update the instance with values that are present in database but not in the submitted instance.

Afterwards calls `insert_or_replace()` with record instances dictionary representation.

**Parameters** **record** (*kuha\_common.document\_store.records.Study or kuha\_common.document\_store.records.Variable or kuha\_common.document\_store.records.Question or kuha\_common.document\_store.records.StudyGroup*) – Document Store record instance.

**Returns** operation details: {'operation': 'insert'|'update', 'id': <ObjectID>, <records-unique-values>}

**Return type** *dict*

**bulk\_insert\_or\_update\_record** (*records*)

Run bulk insert/update operations for Document Store records.

Method that takes an iterable parameter yielding Document Store records. Then calls `insert_or_update_record()` with each record instance.

**Parameters** **records** (*iterable*) – Document Store records.

**Returns** list of `insert_or_update_record` methods operation details.

**Return type** *list*

**insert\_json** (*collection\_name, json\_object*)

Insert JSON-encoded document to Database.

Special method that takes a JSON object that is then inserted to database.

**Parameters**

- **collection\_name** (*str*) – Name of database collection.
- **json\_object** (*str*) – JSON object representing collection document.

**Returns** Insert result.

**Return type** `pymongo.results.InsertOneResult`

**replace\_json** (*collection\_name, oid, json\_object, not\_found\_exception*)

Replace JSON-encoded document in Database.

Special method that replaces a document in database with document given as parameter *json\_object*. The document to be replaced is queried by given *oid*.

This method also takes a *not\_found\_exception* as mandatory parameter. The exception is raised if a document with given *oid* cannot be found.



**Note** if the submitted JSON does not contain metadata for the document. the metadata gets calculated by `RecordsCollection.process_json_for_upsert()`

#### Parameters

- **collection\_name** (*str*) – Name of database collection.
- **oid** (*str*) – MongoDB object ID as string.
- **json\_object** (*str*) – JSON object representing collection document.
- **not\_found\_exception** (*Exception class.*) – exception to raise if document is not found with *oid*

**Returns** Update result.

**Return type** `pymongo.results.UpdateResult`

**delete\_by\_oid** (*collection\_name, oid*)

Delete database document with ObjectID.

#### Parameters

- **collection\_name** (*str*) – Name of database collection.
- **oid** (*str*) – MongoDB object ID as string.

**Returns** Delete result

**Return type** `pymongo.results.DeleteResult`

## validation.py

Simple validation for dictionary representation of document store records.

**note** This module has strict dependency to `kuha_common.document_store.records`

Validate study record dictionary:

```
>>> from kuha_common.document_store.records import Study
>>> from kuha_document_store.validation import validate
>>> validate(Study.get_collection(), Study().export_dict(include_metadata=False))
Traceback (most recent call last):
[...]
def validate(collection, document, raise_error=True, update=False):
kuha_document_store.validation.RecordValidationError: ('Validation of studies failed',
{'study_number': ['null value not allowed']})
)
```

**class** `kuha_document_store.validation.RecordValidator` (*\*args, \*\*kwargs*)

Subclass `cerberus.Validator` to customize validation.

JSON does not support sets. Therefore a rule to validate list items for uniqueness is needed.

For the sake of simplicity in raising and handling validation errors this class also overrides `cerberus.Validator.validate()`.

**validate** (*document, \*\*kwargs*)

Override `cerberus.Validator.validate()`

Handle unvalidated `_id`-field here to simplify error message flow and enable validation messages.

If document is to be updated it is allowed to have an `_id` field. If document is being inserted it is an error to have an `_id` field.

**Parameters**

- **document** (*dict*) – Document to be validated.
- **\*\*kwargs** – keyword arguments passed to `cerberus.Validator.validate()`. Here it is only checked if keyword argument `updated` is present and `True`.

**Returns** `True` if validation passes, `False` if not.

**Return type** `bool`

**exception** `kuha_document_store.validation.RecordValidationError` (*collection*, *validation\_errors*, *msg=None*)

Raised on validation errors.

**Parameters**

- **collection** (*str*) – Collection that got validated.
- **validation\_errors** (*dict*) – Validation errors from `cerberus.Validator.errors`. These are stored in `RecordValidationError.validation_errors` for later processing.
- **msg** (*str*) – Optional message.

**Returns** `RecordValidationError`

**class** `kuha_document_store.validation.RecordValidationSchema` (*record\_class*, \**args*)  
Create validation schema from records in `kuha_common.document_store.records` to validate user-submitted data.

Schema items are built dynamically by consulting record's field types.

- For single value fields the type is string and null values are not accepted.
- For localizable fields it is required to have a `kuha_common.document_store.constants.REC_FIELDNAME_LANGUAGE` attribute.
- Field attributes are strings and they may be null.
- Subfield values are strings and not nullable.
- Fallback to string, not null.

Record's metadata is accepted as input but not required.

**Note** `kuha_common.document_store.RecordBase._metadata` and `kuha_common.document_store.RecordBase._id` are also validated at database level.

**Seealso** `kuha_document_store.database.RecordsCollection.get_validator()`

Every dynamically built schema item may be overridden by a custom schema item given as a parameter for class constructor.

**Parameters**

- **record\_class** (`kuha_common.document_store.records.Study` or `kuha_common.document_store.records.Variable` or `kuha_common.document_store.records.Question` or `kuha_common.document_store.records.StudyGroup`) – class which holds record attributes.
- **\*args** – Custom schema items to override dynamically built schema items.

**Returns** `RecordValidationSchema`

**get\_schema()**  
Get Schema.

**Returns** Validation schema supported by cerberus

**Return type** dict

`kuha_document_store.validation.validate(collection, document, raise_error=True, update=False)`

Validate document against collection schema.

**Parameters**

- **collection** (*str*) – Collection the document belongs to.
- **document** (*dict*) – Document to validate. Document is a dictionary representation of a document store record.
- **raise\_error** (*bool*) – Should a *RecordValidationError* be raised if validation fails.
- **update** (*bool*) – Validate for an update/replace operation of an existing record?

**Returns** True if document passed validation, False if fails.

**Return type** bool

**Raises** *RecordValidationError* if `raise_error` is True and document fails validation.

## db\_setup.py

Script to help setup Document Store database.

Database administrator may use this script to setup MongoDB instance for usage with Document Store.

`kuha_document_store.db_setup.setup_admin_user(admin_username, admin_password, db)`  
Setup administrator credentials.

**Note** authentication must be disabled in MongoDB to use this operation.

**Parameters**

- **admin\_username** (*str*) – administrator username.
- **admin\_password** (*str*) – administrator password.
- **db** (`pymongo.database.Database`) – MongoDB database

`kuha_document_store.db_setup.setup_users(settings, client)`  
Setup database users for Document Store.

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.remove_users(settings, client)`  
Remove Document Store users from database.

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.setup_database(settings, client)`  
Create Document Store database.

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.delete_database(settings, client)`  
Delete Document Store database.

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.list_databases(settings, client)`  
List (print) databases.

**Note** Database won't show in list before it has a collection

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.setup_collections(settings, client)`  
Setup Document Store collections (tables).

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.delete_collections(settings, client)`  
Delete Document Store collections (tables).

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.list_collections(settings, client)`  
List (print) Document Store collections (tables).

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.list_db_users(settings, client)`  
List (print) database users.

**Parameters**

- **settings** (`argparse.Namespace`) – Document Store settings.
- **client** (`pymongo.mongo_client.MongoClient`) – MongoDB client

`kuha_document_store.db_setup.OPERATIONS = {'list_databases': <function list_databases>, 'list_collections': <function list_collections>, 'list_db_users': <function list_db_users>, 'delete_collections': <function delete_collections>, 'delete_database': <function delete_database>, 'setup_collections': <function setup_collections>, 'setup_database': <function setup_database>, 'delete_database': <function delete_database>}`  
Supported operations.

`kuha_document_store.db_setup.main()`  
Script main entry point.

## importers

Supported importers are defined in this package.

Declare importers here.

`kuha_document_store.importers.importers = {'ddi_31': <bound method XMLParserBase.from_str>`  
Register importers here. {importer\_id: importer\_function} Importer\_id must be unique within importers. Importer\_function must accept XML body as string for first argument and Document Store collection as an optional second argument. The importer function must return a generator that will iteratively return populated Document Store record instances.

### 3.8.3 kuha\_oai\_pmh\_repo\_handler

Kuha OAI-PMH Repo Handler application.

Serve records from Kuha Document Store through OAI-PMH protocol.

#### serve.py

Main entry point for starting OAI-PMH Repo Handler.

`kuha_oai_pmh_repo_handler.serve.get_app(api_version, app_settings=None)`  
Setup routes and return initialized Tornado web application.

##### Parameters

- **api\_version** (*str*) – HTTP Api version gets prepended to routes.
- **app\_settings** (*dict or None.*) – Settings to store to application.

**Returns** Tornado web application.

**Return type** `tornado.web.Application`

`kuha_oai_pmh_repo_handler.serve.main()`  
Application main function.

Parse commandline for settings. Setup and serve webapp. Exit on exceptions propagated at this level.

**Returns** exit code, 1 on error, 0 on success.

**Return type** `int`

#### configure.py

Configure OAI-PMH Repo Handler

`kuha_oai_pmh_repo_handler.configure.configure()`  
Get settings for application configuration.

Declares application specific configuration options and some common options declared in `kuha_common.cli_setup`

Configure application with arguments specified in configuration file, environment variables and command line arguments.

**Note** Calling this function multiple times will not initiate new settings to be parsed, but will return previously parsed settings instead.

**Returns** settings

**Return type** `argparse.Namespace`

## genshi\_loader.py

Load genshi templates.

Configure:

```
from genshi_loader import add_template_folder, set_template_writer
add_template_folder(settings.oai_pmh_template_folder)
set_template_writer(handler.template_writer)
```

Use as decorator:

```
from genshi_loader import OAITemplate

class Handler:
    ...
    @OAITemplate('error.xml')
    async def build_error_message(self):
        return {'msg': 'there was an error'}
```

`kuha_oai_pmh_repo_handler.genshi_loader.FOLDERS = []`

Template folders. There can be multiple.

`kuha_oai_pmh_repo_handler.genshi_loader.add_template_folder(folder)`

Add folder to lookup for templates.

**Parameters** `folder` (*str*) – absolute path to folder containing genshi templates.

`kuha_oai_pmh_repo_handler.genshi_loader.get_template_folder()`

Get template folder.

**Returns** template folders.

**Return type** *list*

`kuha_oai_pmh_repo_handler.genshi_loader.WRITER = []`

Template writer. Function which accepts an iterator as parameter.

`kuha_oai_pmh_repo_handler.genshi_loader.set_template_writer(writer)`

Set template writer.

**Note** Supports only one template writer.

**Parameters** `writer` – Function that writes the template. Must accept an iterator as a parameter.

`kuha_oai_pmh_repo_handler.genshi_loader.get_template_writer()`

Get template writer.

**Returns** template writer

**Return type** *function*

**class** `kuha_oai_pmh_repo_handler.genshi_loader.OAITemplate(template_file)`  
OAITemplate class.

Decorate functions that should write output to genshi-templates. The decorated function must be an asynchronous function and it must return a dictionary.

Example:

```
from genshi_loader import OAITemplate
class Handler:
    @OAITemplate('error.xml')
    async def build_error_message(self):
        ...
        return {'msg': 'there was an error'}
```

#### Parameters

- **template\_file** (*str*) – filename of the template to use.
- **template\_folder** (*str*) – optional parameter to use a different template folder to lookup for given template\_file.

**Raises** `ValueError` if decorated function returns invalid type.

## handlers.py

Define handlers for responding to HTTP-requests.

**class** `kuha_oai_pmh_repo_handler.handlers.OAIRouteHandler` (\*args, \*\*kwargs)

Handle requests to OAI endpoint.

*OAIRouteHandler* extends *kuha\_common.server.RequestHandler*.

Input and output goes through this class. It is responsible for accepting requests via HTTP and routing the requests to OAI-protocol and to the correct verb-handler. Verb-handlers are defined in this class.

Verb-handlers are responsible for calling the *kuha\_common.query.QueryController* and again routing the records to OAI-protocol.

Verb-handlers also define the templates used to serialize XML, which is then sent as HTTP-response via *template\_writer()*.

The oai protocol is defined in *kuha\_oai\_pmh\_repo\_handler.oai.protocol*.

**prepare** ()

Prepare each response.

Initialize response. Load query controller. Set output content type.

**template\_writer** (*generator*)

Writes the output from genshi template.

**Parameters** **generator** (*generator*) – generator object containing the XML-serialization.

**get** ()

HTTP-GET handler

Gathers request arguments. Calls router. Finishes the response.

“URLs for GET requests have keyword arguments appended to the base URL”

—<http://www.openarchives.org/OAI/openarchivesprotocol.html#ProtocolFeatures>

**Note** Does not support multiple values for same key.

**post** ()

HTTP-POST handler

Validates request content type. Gathers request arguments. Calls router. Finishes the response.

“Keyword arguments are carried in the message body of the HTTP POST. The Content-Type of the request must be application/x-www-form-urlencoded.”

—<http://www.openarchives.org/OAI/openarchivesprotocol.html#ProtocolFeatures>

## oai

Defines OAI-PMH protocol.

Provides classes for handling requests and responses supported by the protocol. Builds records from *kuha\_common.document\_store.records*.

### oai/errors.py

Errors for OAI-protocol

**exception** *kuha\_oai\_pmh\_repo\_handler.oai.errors.OAIError* (*msg=None, context=None*)

Base for OAI errors

**get\_code** ()

Get OAI error code

**get\_msg** ()

Get OAI error message

**get\_context** ()

Get error context

**get\_contextual\_message** ()

Get error message with possible context.

**Returns** message with context.

**Return type** *str*

**exception** *kuha\_oai\_pmh\_repo\_handler.oai.errors.MissingVerb* (*msg=None, context=None*)

OAIError for missing verb

**exception** *kuha\_oai\_pmh\_repo\_handler.oai.errors.BadVerb* (*msg=None, context=None*)

OAIError for bad verb

**exception** *kuha\_oai\_pmh\_repo\_handler.oai.errors.NoMetadataFormats* (*msg=None, context=None*)

OAIError for no metadata formats

**exception** *kuha\_oai\_pmh\_repo\_handler.oai.errors.IdDoesNotExist* (*msg=None, context=None*)

OAIError for no such id

**exception** *kuha\_oai\_pmh\_repo\_handler.oai.errors.BadArgument* (*msg=None, context=None*)

OAIError for bad argument



**exception** `kuha_oai_pmh_repo_handler.oai.errors.CannotDisseminateFormat` (*msg=None, con-  
text=None*)

OAIError for cannot disseminate format

**exception** `kuha_oai_pmh_repo_handler.oai.errors.NoRecordsMatch` (*msg=None, con-  
text=None*)

OAIError for no records match

**exception** `kuha_oai_pmh_repo_handler.oai.errors.BadResumptionToken` (*msg=None, con-  
text=None*)

OAIError for bad resumption token

## **oai/constants.py**

OAI constants

`kuha_oai_pmh_repo_handler.oai.constants.REGEX_OAI_IDENTIFIER = 'oai:[a-zA-Z][a-zA-Z0-9\-\_]'`  
 Regex to validate oai-identifier. Note that it is more strict than required in <http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm>

`kuha_oai_pmh_repo_handler.oai.constants.REGEX_SETSPEC = '([A-Za-z0-9\-\_\.!\~\*\'\(\)\ ])+'`  
 Sets not complying with this regular expression are invalid according to OAI-PMH schema: see: <http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd>

## **oai/metadata\_formats.py**

Define supported metadata formats.

**class** `kuha_oai_pmh_repo_handler.oai.metadata_formats.MetadataFormatBase`  
 Base class for metadata formats.

Defines common attributes and methods.

**Note** This class must be subclassed and the class attributes overridden.

**prefix = None**

Prefix for metadata format. Override in subclass.

**schema = None**

Schema URL for metadata format. Override in subclass.

**namespace = None**

Namespace for metadata format. Override in subclass.

**get\_prefix()**

Get metadata prefix.

**Returns** metadata prefix.

**Return type** `str`

**get\_schema()**

Get metadata schema URL.

**Returns** URL to metadata schema.

**Return type** `str`

**get\_namespace()**

Get metadata namespace.

**Returns** Metadata namespace.

**Return type** `str`

**get\_relative\_records** ()

Get document store records required by this schema.

These fields are required to represent the record in this metadata schema.

**Note** Implement in subclass.

**Raises** `NotImplementedError`

**get\_record\_fields** ()

Get fields for querying Document Store.

These fields are required to represent the record in this metadata schema.

**Note** Implement in subclass.

**Raises** `NotImplementedError`

**as\_dict** ()

Return metadata attributes in dictionary representation.

**Returns** metadata attributes.

**Return type** `dict`

**class** `kuha_oai_pmh_repo_handler.oai.metadata_formats.DCMetadataFormat`

Metadata format for OAI-DC.

**prefix** = `'oai_dc'`

Metadata prefix for OAI-DC

**schema** = `'http://www.openarchives.org/OAI/2.0/oai_dc.xsd'`

Metadata schema url for OAI-DC

**namespace** = `'http://www.openarchives.org/OAI/2.0/oai_dc/'`

Namespace for OAI-DC

**get\_record\_fields** (*record*=<class 'kuha\_common.document\_store.records.Study'>)

Return Document Store fields required by this schema.

**Parameters** *record* (`kuha_common.document_store.records.Study`) –  
Get fields for this Document Store record. `DCMetadataFormat` only supports  
`kuha_common.document_store.records.Study`

**Returns** document store record fields

**Return type** `list`

**class** `kuha_oai_pmh_repo_handler.oai.metadata_formats.DDIMetadataFormat`

Metadata format for DDI-C.

**prefix** = `'ddi_c'`

Metadata prefix for DDI-C

**schema** = `'http://www.ddialliance.org/Specification/DDI-Codebook/2.5/XMLSchema/codebook'`

Metadata schema url for DDI-C

**namespace** = `'ddi:codebook:2_5'`

Namespace for DDI-C

**get\_record\_fields** (*record*=<class 'kuha\_common.document\_store.records.Study'>)

Return Document Store fields required by this schema.

**Parameters** **record** (*kuha\_common.document\_store.records.Study* or *kuha\_common.document\_store.records.Variable* or *kuha\_common.document\_store.records.Question*) – Get fields for this Document Store record.

**Returns** document store record fields

**Return type** *list*

**class** *kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.CDCDDI25MetadataFormat*  
Metadata format for Cessda Data Catalogue DDI 2.5

**prefix** = `'oai_ddi25'`

Metadata prefix for CESSDA Data Catalogue

**schema** = `'http://www.ddialliance.org/Specification/DDI-Codebook/2.5/XMLSchema/codebook'`

Metadata schema url for DDI-C

**namespace** = `'ddi:codebook:2_5'`

Namespace for DDI-C

**get\_record\_fields** (*record*=<class *'kuha\_common.document\_store.records.Study'*>)

Return Document Store fields required by this schema.

**Parameters** **record** (*kuha\_common.document\_store.records.Study*) – Get fields for this Document Store record.

**Returns** document store record fields

**Return type** *list*

## **oai/protocol.py**

Defines the protocol

*kuha\_oai\_pmh\_repo\_handler.oai.protocol.as\_supported\_datetime* (*datetime\_str*,  
*raise\_oai\_exc=True*)

Convert string representation of datetime to *datetime*.

**Note** If the *datetime\_str* does not come from HTTP-Request, set *raise\_oai\_exc* to False.

**Note** The legitimate formats are YYYY-MM-DD and YYYY-MM-DDThh:mm:ssZ.

### **Parameters**

- **datetime\_str** (*str*) – datetime to convert
- **raise\_oai\_exc** (*bool*) – Catch datetime.strptime errors and reraise as oai-error.

**Returns** converted datetime.

**Return type** *datetime*

**Raises** *kuha\_oai\_pmh\_repo\_handler.oai.errors.BadArgument* for invalid format if *raise\_oai\_exc* is True.

*kuha\_oai\_pmh\_repo\_handler.oai.protocol.as\_supported\_datestring* (*datetime\_obj*)  
Convert *datetime* to string representation.

The target format is YYYY-MM-DDThh:mm:ssZ

**Parameters** **datetime\_obj** (*datetime*) – datetime to convert.

**Returns** string representation of *datetime\_obj*.

**Return type** `str``kuha_oai_pmh_repo_handler.oai.protocol.encode_uri (string)`

Encode uri string.

Replace special characters in string using `urllib.parse.quote()`. Return resulting string.**Parameters** `string (str)` – value to encode.**Returns** encoded value**Return type** `str``kuha_oai_pmh_repo_handler.oai.protocol.decode_uri (uri)`

Decode uri string.

Replace uri encoded special characters in string using `urllib.parse.unquote()`. Return resulting string.**Parameters** `string (str)` – value to decode.**Returns** decoded value**Return type** `str``kuha_oai_pmh_repo_handler.oai.protocol.min_increment_step (datetime_str)`

Count smallest increment step from datetime string.

**Parameters** `datetime_str (str)` – string representation of a datetime. Datetime must be represented either by day's precision or by second's precision.**Returns** smallest increment step.**Type** `datetime.timedelta`**Raises** `ValueError` if string length is invalid.

```
class kuha_oai_pmh_repo_handler.oai.protocol.ResumptionToken (cursor=0,
                                                         from_=None,
                                                         until=None, complete_list_size=None,
                                                         meta-
                                                         data_prefix=None)
```

Class representing OAI-PMH Resumption Token.

Holds attributes of the resumption token. Creates a new resumption token with initial values or takes a dictionary of resumption token arguments. Validates the token based on records list size. If the list size has been changed between requests asserts that the token is invalid by raising a `kuha_oai_pmh_repo_handler.oai.errors.BadResumptionToken` exception.

**Note** Since `OAIArgument.set_` is not supported by resumption token, changing the requested set may result in falsely valid resumption token. But changing the requested set in the middle of a list request sequence should be seen as bad behaviour by the requester/harvester.

**Parameters**

- **cursor** (`int`) – Optional parameter for the current position in list.
- **from** (`str`) – Optional parameter for from datestamp.
- **until** (`str`) – Optional parameter for until datestamp.
- **complete\_list\_size** (`int`) – Optional parameter for the number of records in the complete list.
- **metadata\_prefix** (`str`) – Optional parameter for the requested metadata prefix.

```

class Attribute (key, value)
    Store ResumptionToken attribute keys and values.

    key
        Alias for field number 0

    value
        Alias for field number 1

response_list_size = 100
    Configurable value for the size of the list response.

classmethod set_response_list_size (size)
    Configure response list size.

    Parameters size (int) – Number of records in list response.

classmethod load_resumption_token_argument (argument)
    Create new resumption token from arguments.

    Use to load resumption token from OAI request.

    Parameters argument (str) – Resumption token argument. This comes from HTTP-request.

    Returns New ResumptionToken

set_complete_list_size (size)
    Set the number of records in the complete query response.

    Note Resumption token is invalid if the number of records for the complete query response has
    been changed between requests.

    Parameters size (int) – Number of records for the complete query response.

    Raises kuha_oai_pmh_repo_handler.oai.errors.BadResumptionToken if list
    sizes don't match.

get_complete_list_size ()
    Return the complete list size.

    Returns Number of records in the complete query response.

    Return type int

get_cursor ()
    Get current position in list.

    Returns Cursor representing the current position.

    Return type int

get_from ()
    Get from datestamp.

    Returns datestamp of from attribute.

    Return type datetime.datetime

get_until ()
    Get until datestamp.

    Returns datestamp of until attribute.

    Return type datetime.datetime

get_query_param_until ()
    Get until datestamp for querying.

```

**Note** This is until + smallest increment step.

**Returns** datestamp of query\_param\_until attribute.

**Return type** `datetime.datetime`

**get\_metadata\_prefix()**

Get metadata prefix.

**Returns** metadata prefix from resumption token.

**Return type** `str` or `None`

**get\_encoded()**

Get encoded Resumption Token.

Returns uri-encoded representation of the resumption token if the list request sequence is ongoing. If the list request sequence is over, returns `None`.

**Returns** uri-encoded representation of the token, or `None`

**Return type** `str` or `None`

**class** `kuha_oai_pmh_repo_handler.oai.protocol.OAIResponse` (*request\_url=None*)

Represents the response.

The response is stored in a dictionary which then gets submitted to XML-templates. Thus it is required that the dictionary built within this class is supported by the templates.

**Parameters** `request_url` (*str* or *None*) – Optional requested url. Leave empty to use base url.

**classmethod** `set_repository_name` (*name*)

Set repository name.

**Parameters** `name` (*str*) – repository name.

**classmethod** `set_base_url` (*url*)

Set base url

**Parameters** `url` (*str*) – url.

**classmethod** `set_admin_email` (*email*)

Set admin email address.

**Parameters** `email` (*list*) – Admin email(s)

**classmethod** `set_protocol_version` (*version*)

Set protocol version

**Parameters** `version` (*float*) – OAI-PMH protocol version.

**add\_record** (*record*)

Add record to response

**Parameters** `record` (*kuha\_oai\_pmh\_repo\_handler.oai.records.OAIRecord*)  
– `OAIRecord` to add.

**has\_records** ()

Return True if response has records.

**Return type** `bool`

**assert\_single\_record** ()

Assert the response has a single record.

**Raises** `AssertionError` if there is more or less than a single record.

**set\_earliest\_datestamp** (*datestamp*)

Set earliest datestamp.

**Parameters** **datestamp** (*str*) – datestamp in finest granularity ISO8601

**set\_deleted\_records\_declaration** (*declaration*)

Set deleted records declaration.

**Parameters** **declaration** (*str*) – declare support for deleted records

**set\_granularity** (*granularity*)

Set datestamp granularity.

**Parameters** **granularity** (*str*) – datestamp format for finest granularity supported by this repository.

**set\_metadata\_formats** (*metadata\_formats*)

Set supported metadata formats.

**Parameters** **metadata\_formats** (*list*) – supported metadata formats

**set\_resumption\_token** (*token*)

Set resumption token.

**Parameters** **token** (*ResumptionToken*) – resumption token.

**set\_error** (*oai\_error*)

Set OAI-PMH error.

**Note** These are the errors that are defined in the OAI-protocol. Programming errors are handled separately in higher levels.

**Parameters** **oai\_error** (Subclass of *kuha\_oai\_pmh\_repo\_handler.oai.errors.OAIError*) – OAI error.

**add\_sets\_element** (*spec, name*)

Add new sets element.

**Parameters**

- **spec** (*str*) – setSpec-sublement value.
- **name** – setName-sublement value.

**extend\_sets\_element** (*sets\_list*)

Add multiple sets elements

**Note** Parameter may come directly from *kuha\_oai\_pmh\_repo\_handler.oai.records.Sets.get\_sets\_list\_from\_records()*

**Parameters** **sets\_list** (*list*) – list of sets-elements.

**set\_request\_params** (*oai\_request*)

Gather response parameters from request.

**Note** These are common response parameters that can be added to each response.

**Parameters** **oai\_request** (*OAIRequest*) – Current OAI-request.

**get\_response** ()

Get dictionary representation of the response.

The response attributes are gathered in a dictionary that is to be parsed in the templates.

**Note** The dictionary will contain python objects, so it is not serializable to JSON or arbitrary formats as is.

**Returns** Response ready to pass to templates.

**Return type** `dict`

```
class kuha_oai_pmh_repo_handler.oai.protocol.OAIArguments (verb, resump-
                                                    tion_token=None,
                                                    identifier=None, meta-
                                                    data_prefix=None,
                                                    set_=None,
                                                    from_=None, un-
                                                    til=None)
```

Arguments of OAI-protocol.

Store arguments. Convert timestamps string to datetime objects. Validate arguments for each verb.

**Parameters**

- **verb** (*str*) – requested OAI verb.
- **resumption\_token** (*str*) – requested resumption token.
- **identifier** (*str*) – requested identifier.
- **metadata\_prefix** (*str*) – requested metadata prefix.
- **set** (*str*) – requested set.
- **from** (*str*) – requested timestamp for from attribute.
- **until** (*str*) – requested timestamp for until attribute.

**Raises** `kuha_oai_pmh_repo_handler.oai.errors.OAIError` for OAI errors.

**supported\_verbs** = ['Identify', 'ListSets', 'ListMetadataFormats', 'ListIdentifiers', 'ListRecords']  
 Define supported verbs

**resumable\_verbs** = ['ListSets', 'ListIdentifiers', 'ListRecords']  
 Define resumption token verbs

**supported\_metadata\_formats** = [`<class 'kuha_oai_pmh_repo_handler.oai.metadata_formats.D'`]  
 Define supported metadata formats

**is\_verb\_resumable** ()  
 Is the requested verb a resumable list request?

**Returns** True if verb is resumable False otherwise

**Return type** `bool`

**get\_verb** ()  
 Get requested OAI-verb.

**Returns** requested OAI-verb.

**Return type** `str`

**get\_resumption\_token** ()  
 Get resumption token for request.

The resumption token is either submitted in the request or created automatically.

**Returns** resumption token.

**Return type** `ResumptionToken`

**get\_identifier** ()  
 Get requested identifier.



**Returns** requested identifier if any.

**Return type** `str` or `None`

**get\_local\_identifier()**

Get requested local identifier.

Local identifier does not have prefixes for oai and namespace. It is used to identify records locally.

**Returns** Local identifier if applicable for the request.

**Return type** `str`

**Raises** `kuha_oai_pmh_repo_handler.oai.errors.IdDoesNotExist` for invalid identifier.

**get\_metadata\_format()**

Get requested metadata format.

This is one of the supported metadata formats defined in `OAIArguments.supported_metadata_formats`

**Returns** requested metadata format if any.

**Return type** Subclass of `kuha_oai_pmh_repo_handler.oai.metadata_formats.MetadataFormatBase` or `None`

**get\_set()**

Get requested set.

**Returns** requested set.

**Return type** `str`

**is\_selective()**

Return True if request is selective.

Selective refers to selective harvesting supported by OAI-PMH.

**Returns** True if selective, False if not.

**Return type** `bool`

**has\_set()**

Return True if the request contained set.

**Return type** `bool`

**iterate\_supported\_metadata\_formats()**

Generator for iterating through supported metadata formats.

**Returns** Generator object for iterating supported metadata formats.

```
class kuha_oai_pmh_repo_handler.oai.protocol.OAIRequest (verb, resump-
                                     tion_token=None,
                                     identifier=None, meta-
                                     data_prefix=None,
                                     set_=None, from_=None,
                                     until=None)
```

Represents the OAI request.

Subclass of `OAIArguments`. Defines keys for OAI arguments.

**static from\_request\_args(args)**

Create a request-object from requested arguments.

**Parameters** `args(dict)` – requested arguments.

**Returns** *OAIRequest* object.

## **oai/records.py**

Define OAI records.

**note** This module has a strict dependency to *kuha\_common.document\_store.records*

Contains information for querying records from document store and appending them to responses with *OAIHeaders*, *OAIRecord* and *SETS*.

*kuha\_oai\_pmh\_repo\_handler.oai.records.SetAttribute*  
Attribute to store set configuration

alias of Set

*kuha\_oai\_pmh\_repo\_handler.oai.records.SET\_STUDY\_GROUP = Set(setname='Study group', setspec=)*  
Configuration for study\_group set

*kuha\_oai\_pmh\_repo\_handler.oai.records.SET\_LANGUAGE = Set(setname='Language', setspec='lang)*  
Configuration for language set

*kuha\_oai\_pmh\_repo\_handler.oai.records.SETS = [Set(setname='Study group', setspec='study\_gr)*  
Supported sets

*kuha\_oai\_pmh\_repo\_handler.oai.records.REGEX\_VALID\_SETSPEC = re.compile("([A-Za-z0-9\\-\_\\.)*  
Validation regex for setSpec

*kuha\_oai\_pmh\_repo\_handler.oai.records.is\_valid\_setspec(candidate)*  
Validates setSpec value.

**Parameters** *candidate* (*str*) – setSpec value to validate.

**Returns** True if valid, False if not.

**Return type** *bool*

*kuha\_oai\_pmh\_repo\_handler.oai.records.get\_record\_query\_field\_by\_setspec(setspec)*  
Get document store field to query for set value.

**Parameters** *setspec* (*str*) – setSpec field of the requested set.

**Returns** document store field or None

**Return type** *kuha\_common.document\_store.field\_types.FieldAttribute* or  
None

*kuha\_oai\_pmh\_repo\_handler.oai.records.get\_set\_specs\_from\_ds\_record(ds\_record)*  
Get set specs from document store record.

**Parameters** *ds\_record* (Record object from *kuha\_common.document\_store.records*)  
– One of the document store records. Currently only Study is supported.

**Returns** set specs for use in oai-headers.

**Return type** *dict*

*kuha\_oai\_pmh\_repo\_handler.oai.records.get\_sets\_list\_from\_query\_result(set\_,  
query\_result)*  
Get sets list from query results.

Query is built on the basis of set attributes defined in this class. It is a distinct type of query, so the returned object is not a document store record. This function accepts the results and builds a sets list with each cell containing setName and setSpec keys with their values.

**Parameters**

- **set** (*SetAttribute*) – set-attribute used for the query.
- **query\_result** (*dict*) – results from the query.

**Returns** list of sets to be used in list sets response.

**Return type** *list*

```
class kuha_oai_pmh_repo_handler.oai.records.OAIHeaders (identifier, datestamp,
                                                    **set_specs)
```

Represents OAI-PMH record headers.

Store information of a single record's headers and document store fields to include in query. Provides methods to validate OAI-Identifiers and to iterate set specs list.

**Parameters**

- **identifier** (*str*) – local identifier of a record.
- **datestamp** (*str*) – last modified/updated datestamp.
- **\*\*set\_specs** – key-value pairs of set specs for the record.

**namespace\_identifier** = `None`

Namespace identifier used to construct an OAI-Identifier Use `None` if wish to use local identifiers in OAI-responses.

**identifier\_oai\_prefix** = `'oai'`

Prefix for all identifiers when constructing an OAI-Identifier.

**valid\_oai\_identifier** = `re.compile('oai:[a-zA-Z][a-zA-Z0-9\\-]* (\\. [a-zA-Z][a-zA-Z0-9\\-]`

Validation regex for OAI-Identifier

**valid\_identifier** = `re.compile('[a-zA-Z0-9]+[a-zA-Z0-9?_()-]*')`

Validation regex for local identifier (a subset of oai-identifier)

**classmethod from\_ds\_record** (*ds\_record*)

Return *OAIHeaders* constructed from document store record.

**Note** Currently supports only Study

**Parameters** *ds\_record* (Record object defined in *kuha\_common.document\_store.records*) – Document Store record.

**Returns** headers constructed from Document Store record.

**Return type** *OAIHeaders*

**classmethod set\_namespace\_identifier** (*ns\_id*)

Set namespace identifier for all instances.

**Note** this will be validated afterwards in *set\_identifier()*

**Parameters** *ns\_id* (*str*) – namespace identifier

**classmethod as\_local\_id** (*identifier*)

Get local identifier part of OAI-Identifier.

**Parameters** *identifier* (*str*) – records identifier.

**Returns** local identifier or `None` for invalid identifier.

**Return type** *str* or `None`

**static get\_header\_fields** ()

Get header fields to query.

**Note** currently supports only Study.

**Returns** list of fields to contain in query.

**Return type** list

**set\_identifier** (*identifier*)

Set identifier.

If namespace\_identifier is not None, will build an OAI-Identifier. The identifier will be validated and `ValueError` will be raised if the validation fails.

**Parameters** **identifier** (*str*) – Record’s local identifier.

**Raises** `ValueError` if validation fails.

**get\_identifier** ()

Get identifier

**Returns** record’s identifier.

**Return type** str

**get\_datestamp** ()

Get records datestamp

**Returns** record’s datestamp

**Return type** str

**iterate\_set\_specs** ()

Iterate over setSpec key-value pairs.

**Returns** Generator object for iterating over setSpec key-value pairs.

**Return type** Generator

**class** `kuha_oai_pmh_repo_handler.oai.records.OAIRecord` (*study*)

Class stores record and headers.

**Parameters** **study** (`kuha_common.document_store.records.Study`) – Document Store study record.

**add\_variable** (*variable*)

Add variable to OAIRecord.

**Parameters** **variable** (`kuha_common.document_store.records.Variable`) – Document Store variable.

**add\_question** (*question*)

Add question to OAIRecord.

Question lookup is done by variable name. Therefore it makes sense to use a dictionary with variable\_name as key. The key content will be a list, since a variable may refer multiple questions.

**Note** questions without variable\_name will be discarded and a warning will be logged.

**Parameters** **question** (`kuha_common.document_store.records.Question`) – Document Store question.

**get\_questions\_by\_variable** (*variable*)

Get questions for OAIRecord by variable.

Lookup questions by variable’s variable\_name.

**Parameters** **variable** (`kuha_common.document_store.records.Variable`) – Document Store variable.

**Returns** List of `kuha_common.document_store.records.Question`

**Return type** `list`

### 3.8.4 kuha\_osmh\_repo\_handler

Kuha OSMH Repo Handler application.

Serve records from Kuha Document Store through OSMH protocol.

#### serve.py

Main entry point for starting OSMH Repo Handler

`kuha_osmh_repo_handler.serve.get_app(api_version, app_settings=None)`  
Setup routes and return initialized Tornado web application.

#### Parameters

- **api\_version** (`str`) – HTTP Api version gets prepended to routes.
- **app\_settings** (`dict` or `None.`) – Settings to store to application.

**Returns** Tornado web application.

**Return type** `tornado.web.Application`

`kuha_osmh_repo_handler.serve.main()`  
Application main function.

Parse commandline for settings. Pass settings to `kuha_osmh_repo_handler.server.main()`. Exit on exceptions propagated at this level.

**Returns** exit code, 1 on error, 0 on success.

**Return type** `int`

#### configure.py

Configure OSMH Repo Handler

`kuha_osmh_repo_handler.configure.configure()`  
Get settings for application configuration.

Declares application specific configuration options and some common options declared in `kuha_common.cli_setup`

Configure application with arguments specified in configuration file, environment variables and command line arguments.

**Note** Calling this function multiple times will not initiate new settings to be parsed, but will return previously parsed settings instead.

**Returns** settings

**Return type** `argparse.Namespace`

## handlers.py

Define handlers for responding to HTTP-requests.

**note** OSMH protocol only supports HTTP-GET

**class** `kuha_osmh_repo_handler.handlers.BaseHandler (*args, **kwargs)`

BaseHandler class for handling OSMH requests.

Derived from `kuha_common.server.RequestHandler`. Defines common attributes.

**prepare ()**

Prepare for responding to request.

Set output content type. Initiate `kuha_osmh_repo_handler.response.RecordsResponse` and `kuha_common.query.QueryController` as instance attributes.

**class** `kuha_osmh_repo_handler.handlers.ListRecordHeadersHandler (*args, **kwargs)`

Handle list responses.

**prepare ()**

Prepare for each request.

Depending on stream-configuration choose which response callback to use. If streaming is enabled write output once a single record has been built. Otherwise store all records in a list and write output when all records are built.

**Note** With streaming enabled memory consumption will be lower since the records will not be gathered in a single list and encoded to JSON all at once. When dealing with large repositories the amount of memory consumed without streaming could be an issue.

**get (record\_type=None)**

Handles HTTP-GET requests to endpoint.

**Parameters** `record_type (str or None)` – Optional `record_type` parameter defines if the request limits the list to a certain OSMH type. If the parameter is None, shall output every record in repository. Valid values are defined in handler configuration.

**class** `kuha_osmh_repo_handler.handlers.GetRecordHandler (*args, **kwargs)`

Handle responses for single record.

**get (record\_type, identifier)**

Handle HTTP-GET requests to endpoint.

Gathers the needed information by querying Document Store. Builds the OSMH record response.

**Parameters**

- **record\_type (str)** – requested OSMH record type.
- **identifier (str)** – requested record identifier.

**class** `kuha_osmh_repo_handler.handlers.ListSupportedRecordTypesHandler (*args, **kwargs)`

Handle response to endpoint that lists supported record types.

**get ()**

Handle HTTP-GET request.

**Note** returning object will be a list, so will encode to JSON by using build-in json-module.

**class** `kuha_osmh_repo_handler.handlers.SupportedVersionsHandler (*args, **kwargs)`

Handle response to endpoint that lists supported api version.

**Note** Currently only single version is supported at a time.

**get** ()

Handle HTTP-GET request.

**Note** returning object will be a list, so will encode to JSON by using build-in json-module.

## response.py

Build responses containing OSMH records.

**class** `kuha_osmh_repo_handler.response.RecordsResponse`

Response containing records.

Provides methods that can be used as callbacks to `kuha_common.query.QueryController`. Uses OSMH records defined in `kuha_osmh_repo_handler.osmh.records`. Stores these records for later processing.

**iterate\_records** ()

Iterate through records gathered to response.

**Returns** generator for iterating records.

**get\_record\_appender** (*record\_constructor*)

Get record appender function.

Records that are appended are constructed by using the `record_constructor` parameter.

**Returns** callback function for appending records that get constructed by submitting the receiver parameter to given `record_constructor`.

**Return type** `functools.partial`

**get\_payload\_appender** (*record\_constructor*)

Get payload appender function.

Records that are appended are constructed by using the `record_constructor` parameter. After constructing the record, calls the constructed record-objects `get_payload()` method.

**Returns** callback function for appending records that get constructed by submitting the receiver parameter to given `record_constructor`.

**Return type** `functools.partial`

**get\_response** ()

Build and return the response as a list of dictionary payloads.

**Returns** record payloads

**Return type** `list`

**get\_single\_response** ()

Get single response payload.

**Note** After calling this method the payloads list will be empty.

**Returns** single record's payload.

**Return type** `dict`

**Raises** `ValueError` if contained payloads list has more than a single cell.

## osmh

Defines OSMH records and payload.

### osmh/records.py

Build OSMH payload from Document Store record objects. Provide mapping between these two record formats. Provide Document Store fields for querying.

**note** This module has strict dependency to `kuha_common.document_store.records`

**class** `kuha_osmh_repo_handler.osmh.records.Payload` (*identifier, last\_modified*)  
Represents OSMH record's payload.

Provides methods for manipulating the payload. Stores the payload in a dictionary, which can be easily encoded to JSON.

Example:

```
>>> from kuha_osmh_repo_handler.osmh.records import Payload
>>> payload = Payload('1', '2017-01-01')
>>> payload.insert_localized_value('study_title', 'en', 'Household Survey')
>>> payload.insert_localized_value('study_title', 'fi', 'Kotitalouskysely')
>>> payload.get() # Indent for better readability
{'identifier': '1',
 'lastModified': '2017-01-01',
 'study_title':
   {'fi': 'Kotitalouskysely',
    'en': 'Household Survey'}}
```

#### Parameters

- **identifier** (*str*) – Record's OSMH-identifier. Must uniquely identify the record within other records of the same OSMH record type in the repository.
- **last\_modified** (*str*) – timestamp of the last modification made to the record.

**Returns** *Payload*

**classmethod** `join_values` (*\*args*)  
Join values together using `_join_character`

**Parameters** *\*args* (*str*) – values to join

**classmethod** `split_value` (*value*)  
Split value using `_join_character`

**Parameters** *value* (*str*) – value to split

**Returns** splitted values

**Return type** *list*

**insert** (*key, value*)  
Insert a value to payload.

Insert a value for given key to the payload. If the key is not present in the payload, creates one.

#### Parameters

- **key** (*str*) – payload key for the value.



- **value** (*str*) – value to be inserted.

**insert\_localized\_value** (*key, locale, value*)

Insert a localized value to payload.

Insert value for given locale into the given payload key. If the key is not present in the payload, creates one.

#### Parameters

- **key** (*str*) – payload key
- **locale** (*str*) – values locale
- **value** (*str*) – payload value

**append** (*key, value, unique=False*)

Insert list item to given payload key

If key is not in payload, creates it and inserts a list with a single cell containing value. If parameter unique is True, will not append duplicate values to list.

#### Parameters

- **key** (*str*) – payload key
- **value** (*str*) – value to insert as list item
- **unique** (*bool*) – whether to keep the list of values unique (no duplicates)

**header** (*osmh\_type*)

Create record header to payload

**Note** Header is common for all record types. The only changing value is the record type.

**Parameters** **osmh\_type** (*str*) – OSMH record type

**get** ()

Return the constructed payload

**Returns** OSMH payload

**Return type** dict

**class** `kuha_osmh_repo_handler.osmh.records.OSMHRecord` (*payload*)

Abstract Base class for OSMH record.

Use from a subclass.

Provides common properties and methods to be used in OSMH records.

**Parameters** **payload** (*Payload*) – payload of the record.

**Raises** `TypeError` if subclass does not define class attributes.

**osmh\_type**

OSMH type. Declare in subclass.

**query\_document**

Document Store record to query. Declare in subclass.

**relative\_queries\_for\_record**

Does the record-response require relative records queried from Document Store. Declare in subclass.

**static fields\_for\_header** ()

Get fields to query that are required to build the record header.

Override in subclass.

**static fields\_for\_record()**

Get fields to query that are required to build the record.

Override in subclass.

**static query\_filter\_for\_record(*identifier*)**

Get filter which queries the correct record from Document Store.

Override in subclass.

**classmethod for\_header\_response(*ds\_record*)**

Create a record for response that only contains headers for records.

**Parameters** *ds\_record* (Record defined in *kuha\_common.document\_store.records*) – Document Store record.

**Returns** Instantiated OSMH record object.

**classmethod for\_record\_response(*ds\_record*)**

Create record for response containing the actual record.

**Parameters** *ds\_record* (Record defined in *kuha\_common.document\_store.records*) – Document Store record.

**Returns** Instantiated OSMH record object.

**classmethod get\_query\_document()**

Return the Document Store record used for Querying.

**Returns** Document Store record used for querying.

**classmethod requires\_relative\_queries\_for\_record()**

Does the record require querying for relative records from Document Store to construct the full record response.

**Returns** True or False.

**Return type** `bool`

**build\_header\_payload()**

Builds the common header payload.

**build\_record\_payload()**

Builds the common record payload.

**get\_payload()**

Get the built payload.

**Returns** record payload.

**Return type** `dict`

**class** `kuha_osmh_repo_handler.osmh.records.StudyRecord(study)`

Represents OSMH Study.

Derived from *OSMHRecord*.

**Parameters** *study* (*kuha\_common.document\_store.records.Study*) – Study from Document Store.

**Returns** Instantiated OSMH Study record

**Return type** *StudyRecord*

**query\_document**

alias of *Study*

**static fields\_for\_header()**

Get fields to query that are required to build the record header.

**Returns** Study fields required to build record header.

**Return type** *list*

**static fields\_for\_record()**

Get fields to query that are required to build the record.

**Returns** Study fields required to build record header.

**Return type** *list*

**static query\_filter\_for\_record(*identifier*)**

Get filter which queries the correct record from Document Store.

**Parameters** **identifier** (*str*) – study identifier (study number).

**Returns** filter to use for query.

**Return type** *dict*

**static get\_secondary\_query\_fields\_for\_record()**

Get fields to query that are required to build the relative record (Variable).

**Returns** Variable fields.

**Return type** *list*

**static get\_secondary\_query\_document()**

Get secondary query document (Document Store record).

**Returns** Document Store variable record.

**Return type** *kuha\_common.document\_store.records.Variable*

**get\_secondary\_query\_filter\_for\_record()**

Get filter which queries the correct record from Document Store.

**Returns** filter to use for query.

**Return type** *dict*

**build\_relative\_record\_payload(*relative\_record*)**

Build payload for relative record.

**Parameters** **relative\_record** (*kuha\_common.document\_store.records.Variable*) – Relative record instance.

**build\_record\_payload()**

Build payload for record.

**class** *kuha\_osmh\_repo\_handler.osmh.records.VariableRecord*(*variable*)

Represents OSMH Variable.

Derived from *OSMHRecord*.

**Parameters** **variable** (*kuha\_common.document\_store.records.Variable*) – Variable from Document Store.

**Returns** Instantiated OSMH Variable record

**Return type** *VariableRecord*

**query\_document**

alias of *Variable*

**static fields\_for\_header()**

Get fields to query that are required to build the record header.

**Returns** Variable fields required to build record header.

**Return type** *list*

**static fields\_for\_record()**

Get fields to query that are required to build the record.

**Returns** Variable fields required to build record header.

**Return type** *list*

**static query\_filter\_for\_record(identifier)**

Get filter which queries the correct record from Document Store.

**Parameters** **identifier** (*str*) – variable identifier.

**Returns** filter to use for query.

**Return type** *dict*

**build\_record\_payload()**

Build payload for record.

**class** `kuha_osmh_repo_handler.osmh.records.QuestionRecord(question)`

Represents OSMH Question.

Derived from *OSMHRecord*.

**Parameters** **question** (*kuha\_common.document\_store.records.Question*) – Question from Document Store.

**Returns** Instantiated OSMH Question record

**Return type** *QuestionRecord*

**query\_document**

alias of *Question*

**static fields\_for\_header()**

Get fields to query that are required to build the record header.

**Returns** Question fields required to build record header.

**Return type** *list*

**static fields\_for\_record()**

Get fields to query that are required to build the record.

**Returns** Question fields required to build record header.

**Return type** *list*

**static query\_filter\_for\_record(identifier)**

Get filter which queries the correct record from Document Store.

**Parameters** **identifier** (*str*) – question identifier.

**Returns** filter to use for query.

**Return type** *dict*

**build\_record\_payload()**

Build record payload.

---

**class** `kuha_osmh_repo_handler.osmh.records.StudyGroupRecord` (*study\_group*)  
 Represents OSMH StudyGroup.

Derived from *OSMHRecord*.

**Parameters** `study_group` (*kuha\_common.document\_store.records.StudyGroup*)  
 – StudyGroup from Document Store.

**Returns** Instantiated OSMH StudyGroup record

**Return type** *StudyGroupRecord*

**query\_document**  
 alias of *StudyGroup*

**static fields\_for\_header** ()  
 Get fields to query that are required to build the record header.

**Returns** StudyGroup fields required to build record header.

**Return type** *list*

**static fields\_for\_record** ()  
 Get fields to query that are required to build the record.

**Returns** StudyGroup fields required to build record header.

**Return type** *list*

**static query\_filter\_for\_record** (*identifier*)  
 Get filter which queries the correct record from Document Store.

**Parameters** `identifier` (*str*) – Study group identifier.

**Returns** filter to use for query.

**Return type** *dict*

**build\_record\_payload** ()  
 Build record payload.

`kuha_osmh_repo_handler.osmh.records.get_osmh_record_for_type` (*osmh\_record\_type*)  
 Return the OSMH record class representing *osmh\_record\_type*.

**Parameters** `osmh_record_type` (*str*) – Supported OSMH record type.

**Returns** One of the OSMH records defined in this module.

**Return type** *StudyRecord* or *VariableRecord* or *QuestionRecord* or *StudyGroupRecord*

### 3.8.5 kuha\_client

#### kuha\_client.py

Kuha Client communicates with Document Store and provides a simple way of importing, updating and deleting records by reading a batch of XML files stored in filesystem.

**class** `kuha_client.kuha_client.SourceFile` (*path*)  
 File used as a source for Document Store records.

**Parameters** `path` – Path to source file

**class** `kuha_client.kuha_client.FileLog` (*path*)  
Keep track of processed files.

**Parameters** `path` – Path to filelog

**set\_current** (*\_file*)  
Set current source file.

**Parameters** `_file` (*SourceFile*) – source file currently processed.

**add\_pending\_study\_group** (*study\_group\_identifier*)  
Add StudyGroup record to queue waiting to be processed.

**Parameters** `study_group_identifier` – Id of pending StudyGroup

**pop\_pending\_study\_group\_files** (*study\_group\_identifier*)  
Return and remove source files containing references to `study_group_identifier`.

**Parameters** `study_group_identifier` – StudyGroup identifier.

**Returns** source files referencing `study_group_identifier`

**Return type** `list`

**add\_id** (*collection, \_id*)  
Add id to current source file’s collection of Document Store record IDs.

**Parameters**

- `collection` (*str*) – Document Store collection the ID belongs to.
- `_id` – ObjectId (ID in Document Store) of the Record.

**get\_ids** (*collection*)  
Return list of ObjectIds for `collection` in current file.

**Parameters** `collection` (*str*) – Document Store collection.

**Returns** ObjectIds

**Return type** `list`

**get\_filepaths** ()  
Get paths from `self.files`

Iterate through each *SourceFile* in `self.files` and gather their paths. Return the paths.

**Returns** List of filepaths

**Return type** `list`

**load** ()  
Load FileLog from `self.path`. Populates `self.timestamp` and `self.files`.

**save** ()  
Save FileLog to `self.path`.

**has\_match** (*path*)  
Does the sourcefile found from `path` have a match with `path` and modified timestamp in this filelog.

**Parameters** `path` – Path to source file.

**Returns** True if `path` and timestamps match.

**Return type** `bool`

**remove\_files\_by\_path\_difference** (*paths*)

Remove each *SourceFile* from `self.files` whose path is not in `paths`.

Compare difference in contained source file paths to `paths`. Remove sourcefiles from `self.files` whose paths are not found. Every sourcefile whose paths is not in `paths` gets removed.

**Parameters** `paths` – list of filepaths to compare.

**exception** `kuha_client.kuha_client.DocumentStoreHTTPError` (*error\_response*)

Raise if DocumentStore response payload contains errors.

`kuha_client.kuha_client.get_import_url` (*collection=None, importer=None*)

Construct URL to Document Store import endpoint.

**Parameters**

- **collection** (*str*) – Optional parameter to limit the import to certain collection.
- **importer** (*str*) – Optional parameter to set importer. Defaults to ‘`ddi_c`’

**Returns** Constructed URL

**Return type** *str*

`kuha_client.kuha_client.query_record` (*record*)

Query single record by unique fields.

**Parameters** **record** (*kuha\_common.document\_store.records.Study* or *kuha\_common.document\_store.records.Variable* or *kuha\_common.document\_store.records.Question* or *kuha\_common.document\_store.records.StudyGroup*) – record to query.

**Returns** found record if any.

**Return type** *kuha\_common.document\_store.records.Study* or *kuha\_common.document\_store.records.Variable* or *kuha\_common.document\_store.records.Question* or *kuha\_common.document\_store.records.StudyGroup*

`kuha_client.kuha_client.query_distinct_ids` (*collection*)

Query collection for distinct ObjectIds

**Parameters** **collection** (*str*) – record’s collection.

**Returns** set of distinct ids.

**Return type** *set*

`kuha_client.kuha_client.iterate_xml_directory` (*directory*)

Recursively iterate over XML-files in directory.

**Parameters** **directory** (*str*) – Absolute path to directory.

**Returns** generator for iterating XML-files.

`kuha_client.kuha_client.iterate_xml_files_recursively` (*\*paths*)

Helper for batch processing XML-files.

Check each path. If a path points to a file yield its absolute path. If it points to a directory, recursively iterate paths to each XML-file found and yield each file’s absolute path.

**Parameters** **paths** (*list*) – Paths to file or directory.

**Returns** generator for iterating absolute paths to xml-files

**class** `kuha_client.kuha_client.BatchProcessor` (*collections=None, file\_log=None, sourcefiletype=None*)

Processor for operations performed in a single run.

Keep record of what gets done. Collect StudyGroups from records and update accordingly. Facilitate access to operations needed to perform tasks against Document Store API.

**Parameters**

- **collections** (*list or None*) – List of collections to process. Use None to process all collections.
- **file\_log** (*FileLog*) – Keep track of processed source files and records ObjectIDs related to them.
- **sourcefiletype** (*str or None*) – Controls how the mapping from sourcefile to Document Store records is done. None sets the default SOURCEFILETYPE\_DDIC

**classmethod** `get_supported_sourcefiletypes` ()

Get supported source file types.

**Returns** supported source file types.

**Return type** *list*

**classmethod** `with_file_log` (*file\_log\_path, collections=None, sourcefiletype=None*)

Initiate BatchProcessor with File Log.

**Parameters**

- **file\_log\_path** (*str*) – path to file log.
- **collections** (*list or None*) – collection to process.
- **sourcefiletype** (*str or None*) – file type of source file.

**sourcefileparser** (*path*)

Initiate sourcefileparser, which depends on `self.sourcefiletype`

**Parameters** **path** – path to source file to be parsed.

**Returns** iterative parser

**create** (*record*)

Create new record.

**Parameters** **record** (*kuha\_common.document\_store.records.Study or kuha\_common.document\_store.records.Variable or kuha\_common.document\_store.records.Question or kuha\_common.document\_store.records.StudyGroup*) – populated record instance which gets created.

**Returns** ObjectID of the new record.

**Return type** *str*

**upsert** (*record*)

Upsert record.

If record exists, compare with existing. If records differ, discard the existing record and store the new one to DocumentStore with the existing ObjectID. If record does not exist, insert it to DocumentStore.

**Parameters** **record** (*kuha\_common.document\_store.records.Study or kuha\_common.document\_store.records.Variable or kuha\_common.document\_store.records.Question or kuha\_common.document\_store.records.StudyGroup*) – populated record instance which gets created.



**Returns** ObjectId of the record.

**Return type** `str`

**upsert\_from\_parser** (*parser*)

Upsert records to `self.collections` from parser.

**Parameters** **parser** – Parser generates Document Store records.

**upsert\_paths** (*\*paths*)

Upsert records found recursively from paths.

**Parameters** **\*paths** – one or more paths to recurse to look for files to parse.

**upsert\_study\_groups** ()

Upsert collected StudyGroups.

**add\_study\_group** (*study\_group*)

Add StudyGroup for later processing.

Lookup the StudyGroup if it has been stored before and update its contents. If it's not found, store it as a new one.

**Parameters** **study\_group** (*kuha\_common.document\_store.records.StudyGroup*) – StudyGroup to add for later processing.

**import\_source** (*source\_data*)

Import source data to Document Store.

**import\_source\_files** (*\*paths*)

Import files from paths.

**Parameters** **\*paths** – one or more paths to lookup for source files.

**remove\_absent** (*collection*)

Remove records from collection which were not present in current upsert run.

**Parameters** **collections** (*str*) – collection to process.

**remove\_absent\_records** ()

Remove records which were not present in current upsert run.

**remove\_record** (*record*)

Remove record or records.

If record is an instance of DocumentStore record, remove it from DocumentStore. If record is a record class, remove all records from it's collection.

**Parameters** **record** (*DocumentStore record instance or class.*) – Record to remove or class whose records will be removed.

**remove\_study\_and\_relatives\_by\_studyid** (*study\_id*)

Remove study and relative records.

For a single study the process should remove:

- Actual Study,
- Variable referenced from the Study,
- Questions referenced from the Study,
- Remove references to the Study from StudyGroups.

**Note** Does not remove StudyGroup even if all references to studies are removed.

**Parameters** `study_id` (*str*) – ObjectId of the study to remove.

`import_run` (*lookup\_paths*)

Main entry point for import run.

**Parameters** `lookup_paths` (*list*) – list of paths to lookup for source files.

`upsert_run` (*lookup\_paths*, *remove\_absent=False*)

Main entry point for upsert run.

Upsert records found from `lookup_paths`. Remove absent records if `remove_absent` is True.

**Parameters**

- `lookup_paths` (*sequence*) – list of paths to lookup for source files.
- `remove_absent` (*bool*) – True will remove all records not found from `lookup_paths`.

`remove_run` (*record\_or\_class=None*)

Main entry point for remove run.

**Parameters** `record_or_class` – Record or RecordClass to remove. If None will remove every record in every collection.

## kuha\_import.py

Callable module serves as entry point to import records from DocumentStore.

Example run from command line. Import records from /some/path:

```
python -m kuha_client.kuha_import --document-store-url=http://localhost:6001/v0 /some/  
↪path
```

Print help:

```
python -m kuha_client.kuha_import -h
```

`kuha_client.kuha_import.import_run` (*paths*, *file\_log\_path=None*, *\*\*kwargs*)

Import run with arguments.

**Parameters**

- `paths` – Lookup source files from paths.
- `file_log_path` – Path to file log.
- `**kwargs` – Additional keyword arguments get passed to BatchProcessor.

**Returns** 0 on success.

**Return type** `int`

`kuha_client.kuha_import.cli` ()

Parse command line arguments. Call `import_run` ().

**Returns** Return value of `import_run` ()

## kuha\_upsert.py

Callable module serves as entry point to upsert (insert or update) records from DocumentStore.

Use Document Store's Query API to see if document exists. If it exists, fetch it, update it, submit it back to Document Store via REST API.

Example run from command line. Upsert records from /some/path:

```
python -m kuha_client.kuha_upsert --document-store-url=http://localhost:6001/v0 /some/
↳path
```

Print help:

```
python -m kuha_client.kuha_upsert -h
```

`kuha_client.kuha_upsert.upsert_run` (*paths*, *collections=None*, *file\_log\_path=None*, *remove\_absent=False*, *sourcefiletype=None*)

Upsert run with arguments.

#### Parameters

- **paths** – Lookup source files from paths.
- **collections** – Limit run to collections.
- **file\_log\_path** – Path to file log.
- **remove\_absent** – Should upsert run remove records, which are found from Document Store but not from source files in current run.
- **sourcefiletype** – File type of source files.

**Returns** 0 on success.

**Return type** int

`kuha_client.kuha_upsert.cli()`  
Parse command line arguments. Call `upsert_run()`.

**Returns** Return value of `upsert_run()`

### kuha\_delete.py

Callable module serves as entry point to delete records from DocumentStore.

Example run from command line. Delete study with ID:

```
python -m kuha_client.kuha_delete --document-store-url=http://localhost:6001/v0
↳studies 5afa741d6fb71d7b2d333982
```

Print help:

```
python -m kuha_client.kuha_delete -h
```

`kuha_client.kuha_delete.cli()`  
Parse command line arguments and call `BatchProcessor.remove_run()`

**Returns** 0 on success.

**Return type** int

## 3.9 Versions

### 3.9.1 Kuha Common Changelog

#### 0.9.1 (2019-04-03)

- DDI mappings: `kuha_common.document_store.mappings.xmlbase.MappedParams.has_arguments()` treat MappedParams objects which don't have a value and all keyword arguments' values are None as MappedParams objects that don't have any arguments. This way all empty XML elements will be discarded in the mapping phase before they get turned into Document Store records. (Fixes #26)
- Treat attributes-dictionaries with only None values as invalid if the Element's value is also None in `kuha_common.document_store.field_types.Element.add_value()`.
- Servers exception logging `kuha_common.server.log_exception()` will now log all traceback lines, instead of the last four.

#### 0.9.0 (2019-03-14)

- Maintenance release with no new functionality besides bug fixes.
- Support tornado > 5
  - Remove callbacks used with tornado client.
  - requirements.txt declares tornado==6.0.1, which is currently the latest.
- `kuha_common.query.QueryController._query_multiple_with_limit()` now correctly handles limit and skip query parameters.
- `kuha_common.query.QueryController.query_single()` now correctly raises `kuha_common.document_store.query.QueryException` if passed a limit parameter.
- Add validation functions to `kuha_common.document_store.query` to validate query parameters.
- Add common testcase functionality to `kuha_common.testing.testcases.KuhaUnitTestCase`.
- Fix `kuha_common.document_store.client.JSONStreamClient.fetch()` so it won't interfere with queue created by `kuha_common.document_store.client.JSONStreamClient.queue_request()`.
- Update copyright headers to 2019.

#### 0.8.0 (2018-12-18)

- Refactor `kuha_common.server`.
  - `kuha_common.server.serve()` replaces `run_server()` function. It takes the web-application as a parameter and does not handle application settings. Setting up the application should be handled by the caller who instantiates the application and knows what settings the application needs in order to operate.
- Add `kuha_common.testing.MockCoro` to help mocking out coroutine functions.

### 0.7.1 (2018-11-20)

- `kuha_common.document_store.mappings.xmlbase.XMLParserBase._get_study_number_from_study_unit_element()` Change priority of elements when looking up `Study.study_number`:
  1. `a:Archive/a:ArchiveSpecific/a:Collection/a:CallNumber`
  2. `a:Archive/a:ArchiveSpecific/a:Item/a:CallNumber`
  3. `s:StudyUnit/r:UserID`

### 0.7.0 (2018-11-19)

- **Breaks backward compatibility** Refactored DDI mapping profiles' API. Package structure of `kuha_common.document_store.mappings` changed.
  - `kuha_common.document_store.mappings.xmlbase.XMLParserBase` now mandates how subclasses should be built by providing the public API for callers. The implementation of the actual mapping of resources is an implementation detail of subclasses and does not need to be public.
  - Refactor all DDI mapping profiles to a single module `kuha_common.document_store.mappings.ddi`
  - Add more mapping exceptions to differentiate error conditions in `kuha_common.document_store.mappings.exceptions`
  - Add mapping profile for DDI 3.1. to a new class `kuha_common.document_store.mapping.ddi.DDI31RecordParser` (Implements #24)

### 0.6.0 (2018-07-18)

- Add methods to `kuha_common.testing.testcases`
- Add support for parsing StudyGroups from DDI 1.2.2. in `kuha_common.document_store.mappings.ddi_122_nesstar.DDI122RecordParser.p_study_groups()` (Implements #20)
- Omit logging of request body to request log if the body is larger than `kuha_common.server.REQUEST_LOG_BODY_MAXLEN`. (Implements #22)
- Fix possible `JSONDecodeError` in `kuha_common.server.log_request()` (Fixes #23)

### 0.5.1 (2018-07-11)

- Declare `testing.testcases.KuhaUnitTestCase.gen_id()` as a classmethod, since it uses class's method.

### 0.5.0 (2018-07-10)

- Package for common test functions and classes `kuha_common.testing`. (Implements #12)

### 0.4.1 (2018-07-04)

- `kuha_common.document_store.mappings.xmlbase.XMLMapper._values_from_parent()` resets the state of the attribute mapper if it needs to manipulate the mapper. (Fixes #19)

#### 0.4.0 (2018-07-03)

- Relax record schema by decreasing the number of mandatory attributes. It should be possible to populate an element with attributes but have no value. (Implements #7)
  - `kuha_common.document_store.field_types.Element.add_value()` can be called without a value-parameter.
  - `kuha_common.document_store.field_types.Value.export_dict()` returns an empty dict if value is None.
  - `kuha_common.document_store.field_types.ElementContainer._updates()` If `sec_value_value` is None and no match is found, create and append new sub-element.
  - `kuha_common.document_store.field_types.ElementContainer._updates()` If `sec_value_value` is None and a matching value and attributes is found, discard the `sec_value` completely. (Implements #15)
  - Fix possible `KeyError` in `kuha_common.document_store.field_types.FieldAttribute.value_from_dict()` This was regression introduced by #7. (Fixes #16)
  - `kuha_common.document_store.field_types.FieldAttribute.value_from_dict()` Now always returns None if nothing was found. Previously it was possible to get an empty list.
- Relax DDI-C mappings: allow record field's value to be None, if there are attributes for the field. (Implements #8)
- Add support for DDI from Nesstar Publisher. (Implements #9)
  - New mapping file `ddi_122_nesstar.py` to `kuha_common.document_store.mappings` package.
  - Refactor `document_store/mappings/ddi_c.py` and separate common XML classes & functions to a new module: `xmlbase.py`.
    - \* `kuha_common.document_store.mappings.xmlbase.Value.from_element()` converts empty string values to None.
- Correctly handle mapping from DDI-C if Codebook instance contains multiple series separated by ID. (Fixes #10)
- Fix DDI-C mapping for localized codelists for variables. (Fixes #11)
- DDI-C mapping for Question now removes extra whitespace around `research_instruments`.
- Fix fetching multiple parent elements for mapped parameters which have no main element. (Fixes #18)

#### Known Issues

- Mapping is unable to handle DDI-XML (DDI 2.5 and DDI 1.2.2 Nesstar) which contains inconsistent use of conceptual elements. See issue #17. For instance the following structure for `anlyUnit`:

```
<anlyUnit>Description for analysis unit.
  <concept>concept.of.analysis.unit</concept>
</anlyUnit>
<anlyUnit>Description for another analysis unit.</anlyUnit>
```

Will be interpreted as:

```
[{'analysis_unit': 'concept.of.analysis.unit',
  'description': 'Description for analysis unit.',
  'language': 'en'}]
```

### 0.3.1 (2018-04-19)

- Add optional keyword arguments to `kuha_common.cli_setup.add_document_store_url()` that are passed to `parser.add()`.
- Declare `kuha_common.cli_setup.Settings.set()` as public function.
- `kuha_common.document_store.field_types.Value.updates()` now actually updates the value.
- Support query filter for record's `_id`.
- Use document-store-url with full document store base url for `kuha_common.document_store.query.Query`. (Fixes #5)
- Include request body for PUT and PATCH in `kuha_common.server.log_request()`.
- Add `kuha_common.document_store.records.RecordBase.get_id()`.
- Add `kuha_common.document_store.records.RecordBase.updates()`.
- Add updates method for each subclass of `kuha_common.document_store.records.RecordBase`.
- Changes in `kuha_common.document_store.mappings.ddi_c`:
  - Fix default value for `universe.attr_clusion`.
  - Introduce new class `kuha_common.document_store.mappings.ddi_c.RecordParser` which stores default language; this way parsing can be done simultaneously to multiple XML-documents.
  - Default language is now a mandatory parameter for mapping methods:
    - \* `kuha_common.document_store.mappings.ddi_c.Value.from_element()`
    - \* `kuha_common.document_store.mappings.ddi_c.ValueMap.__call__()`
    - \* `kuha_common.document_store.mappings.ddi_c.MultiValueMap.__call__()`
  - Remove `kuha_common.document_store.mappings.ddi_c.parse()` to streamline the process.
  - Add global to store each parser and corresponding record collection.
  - Declare `kuha_common.document_store.mappings.ddi_c.check_root()` as public function.
  - Add `kuha_common.document_store.mappings.ddi_c.get_root_language()` function.

### 0.3.0 (2018-03-06)

- Move `ddi_c.py` mapping module (DDI-C -> Document Store records) from `kuha_document_store` to `kuha_common.document_store.mappings` package.
- Forward keyword arguments from `Settings.load_parser` to `configargparse.get_arg_parser` in `cli_setup.py`.
- Make `JSONStreamClient._get_request` a public method `JSONStreamClient.get_request`

- Forward keyword arguments from `JSONStreamClient.get_request` to `DocumentStoreClient.streaming_query_request` to support more options specifically more HTTP-methods than POST.
- Assert `_log_request()` in `server.py` will not raise `UnicodeDecodeError` if `request.body` is not utf-8 encoded.
- Add `Study.document_uris`
- Add abbreviation-attribute to `Study.publishers`.
- Add DDI-C mappings to `Study.document_uris` and `Study.publishers.attr_abbreviation`.

### 0.2.3 (2018-01-26)

- Implement support for non-localizable containerized elements.
- Add more fields to Study record.
- Add more unit & integration tests.

### 0.2.2 (2017-11-10)

- Update documentation

### 0.2.1 (2017-11-09)

- Fix referring variables to questions. Variable may refer multiple questions.
- Fix `server.py log_request` function. Call `RequestHandler.CONTENT_TYPE_JSON`, rather than handler-object.
- Partial support for coroutine callbacks in `QueryController`

### 0.2.0 (2017-11-01)

- Support referring variables to questions and vice versa.
- Add `tox.ini` to support running tests with `tox`.

### 0.1.0 (2017-10-25)

- Initial release

## 3.9.2 Kuha Document Store Changelog

### 0.7.1 (2019-04-05)

- Kuha Common 0.9.1 improves handling of empty XML elements. Use it as a requirement in `requirements.txt`.
- Add end-to-end test to make sure empty elements are handled correctly when importing.

### 0.7.0 (2019-03-14)

- Support for `kuha_common` 0.9.0
- Update copyright headers to 2019.



**0.6.0 (2018-12-18)**

- Initial support for importing DDI 3.1. metadata.
  - Require `kuha_common>=0.8.0`

**0.5.0 (2018-07-19)**

- Support importing StudyGroups from DDI 1.2.2 metadata.
  - Require `kuha_common>=0.6.0`
- Refactor end-to-end tests: Use `kuha_common.testing` package. (Implements #12)

**0.4.2 (2018-07-04)**

- Require `kuha_common>=0.4.1`, because 0.4.0 had critical bug.

**0.4.1 (2018-07-04)**

- Use tag 0.4.0 for `kuha_common` in `requirements.txt` rather than the release branch. This way it is easier to revert to older releases of Document Store.

**0.4.0 (2018-07-03)**

- Add importer for DDI 1.2.2 Nesstar.

**0.3.2 (2018-05-15)**

- Relax identifier validation to allow dots.
- Cast MongoDB ObjectId to string for distinct query return values. (Fixes #11)

**0.3.1 (2018-03-14)**

- Remove extra file `kuha_document_store.ini.dist`.
- `scripts/install_kuha_document_store_virtualenv.sh`: install with pip and use `-upgrade` flag to support upgrading.

**0.3.0 (2018-03-06)**

- Import API now returns HTTP status code 400, if `DataImportError` is risen.
- Import API gives result (`'import failed'`) if import has failed, instead of null.
- Fix `TypeError` on `datestamp` fields for distinct query type. (Fixes #8)
- Move `ddi_c.py` mapping module to `kuha_common` library.
- Fix validation for localizable fields regarding the language attribute. It is now mandatory as it should be. (Fixes #9)
- `scripts/install_kuha_document_store_virtualenv.sh`: add `-upgrade` flag to pip install cmd.
- Write some more documentation.

### 0.2.3 (2018-01-30)

- DDI-C importer: add support for
  - Study.identifiers
  - Study.distributors
  - Study.classifications.attr\_uri,
  - Study.classifications.attr\_description
  - Study.keywords.attr\_uri
  - Study.keywords.attr\_description
  - Study.collection\_periods
  - Study.analysis\_units
  - Study.time\_methods
  - Study.sampling\_procedures
  - Study.collection\_modes
  - Study.data\_access\_descriptions
- Add validation for Study.persistent\_identifiers.
- Separate tests in subfolders *end\_to\_end*, *integration*, *unit*.
- Add some unit-tests against DDI-C importer. Coverage is still lacking.

### 0.2.2 (2017-11-10)

- Update documentation.

### 0.2.1 (2017-11-09)

- **Breaks backward compatibility to 0.2.0. Users should rebuild variable collections.**
- Fix referring variables to questions. Variable may refer multiple questions.

### 0.2.0 (2017-11-01)

- Support referring variables to questions and vice versa.

### 0.1.2 (2017-11-01)

- Support querying without created and updated -fields.
- Add tox.ini to support running tests with tox.

### 0.1.1 (2017-10-27)

- Update documentation.

**0.1.0 (2017-10-25)**

- Initial version.

**3.9.3 Kuha OAI-PMH Repo Handler Changelog****0.6.0 (2019-03-14)**

- Support for kuha\_common 0.9.0
- Update copyright headers to 2019.

**0.5.0 (2018-12-18)**

- Require kuha\_common 0.8.0. for better support for testing tornado handlers.
- Decouple setup of `template_folder` from importing handlers.

**0.4.0 (2018-09-13)**

- DDI-C template: Handle the possibility of None values in `variable.codelist_codes[n].code`. (Fixes #14)
- Support using `base_url` for OAI-PMH request element. (Implements #6)
  - Add configuration option `--oai-pmh-respond-with-requested-url` which defaults to False. Using this computes the base url for OAI-PMH request element from the HTTP request.
- Require `kuha_common >= 0.6.0`.

**0.3.0 (2018-07-12)**

- Pin requirement to `kuha_common` version in `requirements.txt`. This way it is easier to use older releases.
- Fix possible `TypeError` in `kuha_oai_pmh_repo_handler.oai.records.is_valid_setspec()`, by explicit check for `None` in `kuha_oai_pmh_repo_handler.oai.records.get_set_specs_from_ds_record()`. (Fixes #11)
- Conceptual container elements should be presented even if no concept is found. (Implements #10)
- Fix possible `NameError` in `kuha_oai_pmh_repo_handler.oai.OAIRecord.add_question()`. (Fixes #12)
- Serve `topcClas` and keyword DDI 2.5 elements even if no `@ID` attribute can be found. (Implements #13)
- Refactor end-to-end tests: Use `kuha_common.testing` package.

**0.2.5 (2018-03-14)**

- Add DDI 2.5 metadata format for CESSDA Data Catalogue. (Resolves #8)
- `scripts/install_kuha_oai_pmh_repo_handler_virtualenv.sh`: Use `pip` to install & upgrade.

**0.2.4 (2018-03-09)**

- `ListIdentifiers` verb handler should not retrieve relative records. (Fixes #7)

### 0.2.3 (2018-03-07)

- DDI-C metadata & template:
  - Add support for Study.document\_uris
  - Add support for Study.publishers.attr\_abbreviation
  - Add missing vocabURI-attribute to topClas element
  - Add missing vocabURI-attribute to keyword element
- OAI-DC metadata & template:
  - Use Study.document\_uris for identifier.
- Don't require metadataPrefix when request contains resumptionToken. (Fixes #5)
- Add upgrade flag to pip install command.

### 0.2.2 (2018-01-31)

- DDI-C metadata & template: add support for:
  - Study.identifiers
  - Study.distributors
  - Study.time\_methods
  - Study.sampling\_procedures
  - Study.collection\_modes
  - Study.analysis\_units
  - Study.collection\_periods
  - Study.data\_access\_descriptions
- Add support for Study.identifiers in OAI-DC metadata & template.
- Fix incorrect handling of requested identifier when using OAI namespace-identifier. (Fixes #4)
- Separate OAI-PMH error messages for *no known item* and *invalid identifier structure*

### 0.2.1 (2017-11-16)

- Add ID-attribute to qstn-element in DDI-C template

### 0.2.0 (2017-11-10)

- Support for variable level metadata in DDI-C

### 0.1.1 (2017-10-27)

- Update documentation

#### 0.1.0 (2017-10-25)

- Initial release

### 3.9.4 Kuha OSMH Repo Handler Changelog

#### 0.3.0 (2019-03-20)

- requirements.txt: kuha\_common 0.9.0, tornado 6.0.1, configargparse 0.14.0.
- handlers.py: Make sure flush\_queue callable is set prior calling.
- handlers.py: Fix bug when streaming from empty collection.
- osmh/records.py: Don't write analysis units with empty description to response.

#### 0.2.0 (2018-12-18)

- Require kuha\_common 0.8.0 for better support for testing tornado handlers.

#### 0.1.4 (2018-03-14)

- scripts/install\_kuha\_osmh\_repo\_handler\_virtualenv.sh: use pip to install and upgrade.

#### 0.1.3 (2018-03-07)

- Add analysis unit to study record.
- Use keyword.attr\_description for study record subject.
- Add upgrade flag to pip install command.
- Add some tests.
- Change records.Payload.\_join\_character to ':', since it is illegal character in XML attributes. This reduces the risk of possible collisions.

#### 0.1.2 (2017-11-10)

- Update documentation: CHANGES.rst, remove version from kuha\_common at README.rst

#### 0.1.1 (2017-10-27)

- Update documentation: configuration.rst, running.rst

#### 0.1.0 (2017-10-25)

- Initial release

## 3.9.5 Kuha Client Changelog

### 0.5.1 (2019-04-11)

- Kuha Common 0.9.1 improves handling of empty XML elements. Use it as a requirement in requirements.txt.
- Add end-to-end test to ensure empty elements are handled correctly on upsert.
- Call `kuha_common.cli_setup.settings.setup_document_store_query()` in `kuha_client.kuha_delete` to set base url to `kuha_common.document_store.query.Query` (Fixes #10)

### 0.5.0 (2019-03-20)

- Require `kuha_common 0.9.0`.
  - Remove `kuha_client.kuha_client.AsyncKuhaClient` since its functionality can now be achieved with `kuha_common.document_store.client.JSONStreamClient`. Refactor HTTP functions to use that instead.
- Make sure Document Store errors are properly logged out. (Fixes #9)
- Update copyright headers to 2019.

### 0.4.0 (2019-01-17)

- Support for DDI 3.1
- Require `kuha_common 0.8.0`

### 0.3.1 (2018-08-30)

- Check that the `ObjectId` that gets logged is not one that has been deleted before adding the `ObjectId` to `FileLog`. (Fixes #8)
- Add debug log messages.

### 0.3.0 (2018-07-19)

- Pin requirement to `kuha_common` version in requirements.txt. This way it is easier to use older releases.
- Support for DDI 1.2.2 from Nesstar Publisher. (Implements #7)
  - Require `kuha_common>=0.6.0`
- Remove files from file log which are not found in current batch. (Fixes #2)

### 0.2.4 (2018-05-25)

- Fix call for `FileLog.add_id()`. (Fixes #6)

### 0.2.3 (2018-05-22)

- Fix regression introduced by 0.2.2. (Fixes #5)

### 0.2.2 (2018-05-21)

- Fix removing of absent StudyGroups when using file log. (Fixes #4)

### 0.2.1 (2018-05-16)

- Fix callable module prog paths from help message.

### 0.2.0 (2018-05-16)

- Complete rewrite of application logic.
- Add tests.
- Support updating and deleting Document Store records.
- Implement file log for keeping track of previously processed files.
- Use clients from kuha\_common.client rather than requests-module.
- Update all documentation to match current behaviour.

### 0.1.2 (2017-11-10)

- Update documentation: CHANGES.rst

### 0.1.1 (2017-10-27)

- Update documentation: configuration.rst, running.rst

### 0.1.0 (2017-10-25)

- Initial release





## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**k**

kuha\_client.kuha\_client, 121  
kuha\_client.kuha\_delete, 127  
kuha\_client.kuha\_import, 126  
kuha\_client.kuha\_upsert, 126  
kuha\_common, 33  
kuha\_common.cli\_setup, 37  
kuha\_common.document\_store, 40  
kuha\_common.document\_store.client, 40  
kuha\_common.document\_store.field\_types, 48  
kuha\_common.document\_store.mappings, 70  
kuha\_common.document\_store.mappings.ddi, 77  
kuha\_common.document\_store.mappings.exceptions, 70  
kuha\_common.document\_store.mappings.xmlbase, 71  
kuha\_common.document\_store.query, 42  
kuha\_common.document\_store.records, 56  
kuha\_common.query, 35  
kuha\_common.server, 33  
kuha\_common.testing, 79  
kuha\_common.testing.testcases, 80  
kuha\_document\_store, 83  
kuha\_document\_store.configure, 84  
kuha\_document\_store.database, 87  
kuha\_document\_store.db\_setup, 95  
kuha\_document\_store.handlers, 84  
kuha\_document\_store.importers, 97  
kuha\_document\_store.serve, 84  
kuha\_document\_store.validation, 93  
kuha\_oai\_pmh\_repo\_handler, 97  
kuha\_oai\_pmh\_repo\_handler.configure, 97  
kuha\_oai\_pmh\_repo\_handler.genshi\_loader, 98  
kuha\_oai\_pmh\_repo\_handler.handlers, 99  
kuha\_oai\_pmh\_repo\_handler.oai, 100  
kuha\_oai\_pmh\_repo\_handler.oai.constants, 101  
kuha\_oai\_pmh\_repo\_handler.oai.errors, 100  
kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats, 101  
kuha\_oai\_pmh\_repo\_handler.oai.protocol, 103  
kuha\_oai\_pmh\_repo\_handler.oai.records, 110  
kuha\_oai\_pmh\_repo\_handler.serve, 97  
kuha\_osmh\_repo\_handler, 113  
kuha\_osmh\_repo\_handler.configure, 113  
kuha\_osmh\_repo\_handler.handlers, 114  
kuha\_osmh\_repo\_handler.osmh, 116  
kuha\_osmh\_repo\_handler.osmh.records, 116  
kuha\_osmh\_repo\_handler.response, 115  
kuha\_osmh\_repo\_handler.serve, 113



---

## HTTP Routing Table

---

### **/(collection)**

GET /(collection)/(document\_id),??  
PUT /(collection)/(document\_id),??  
DELETE /(collection)/(document\_id),??

### **/import**

POST /import/(importer\_id)/(collection),  
??

### **/query**

POST /query/(collection),??

### **/questions**

POST /questions,??

### **/studies**

POST /studies,??

### **/study\_groups**

POST /study\_groups,??

### **/variables**

POST /variables,??



## Symbols

- collection <collection>  
command line option, 31
- database-host <database\_host>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16
- database-name <name>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16
- database-pass-editor <password>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16
- database-pass-reader <password>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16
- database-port <port>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16
- database-user-editor <user>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16
- database-user-reader <user>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16
- document-store-api-version <api\_version>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 15  
kuha\_oai\_serve command line option, 27  
kuha\_osmh\_serve command line option, 30
- document-store-client-connect-timeout <timeout>  
kuha\_oai\_serve command line option, 27  
kuha\_osmh\_serve command line option, 30
- document-store-client-max-clients <max\_clients>  
kuha\_oai\_serve command line option, 27  
kuha\_osmh\_serve command line option, 30
- document-store-client-request-timeout <timeout>  
kuha\_oai\_serve command line option, 27  
kuha\_osmh\_serve command line option, 30
- document-store-host <host>  
kuha\_oai\_serve command line option, 27  
kuha\_osmh\_serve command line option, 29
- document-store-port <port>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 15  
kuha\_oai\_serve command line option, 27  
kuha\_osmh\_serve command line option, 29
- document-store-url <document\_store\_url>  
command line option, 31
- file-log-path <path>  
command line option, 32
- logformat <logformat>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16  
kuha\_oai\_serve command line option, 27  
kuha\_osmh\_serve command line option, 30
- loglevel <loglevel>  
kuha\_ds\_serve,-kuha\_db\_setup command line option, 16  
kuha\_oai\_serve command line option, 27  
kuha\_osmh\_serve command line option, 30
- oai-pmh-admin-email <email>  
kuha\_oai\_serve command line option, 27
- oai-pmh-api-version <api\_version>  
kuha\_oai\_serve command line option, 27
- oai-pmh-base-url <base\_url>  
kuha\_oai\_serve command line option, 27
- oai-pmh-namespace-identifier <namespace\_id>  
kuha\_oai\_serve command line option, 27
- oai-pmh-protocol-version <version>  
kuha\_oai\_serve command line option, 27
- oai-pmh-repo-name <repo\_name>  
kuha\_oai\_serve command line option, 27
- oai-pmh-results-per-list <results\_per\_list>  
kuha\_oai\_serve command line option, 27
- osmh-repo-handler-api-version <api\_version>  
kuha\_osmh\_serve command line option, 29
- port <port>  
kuha\_oai\_serve command line option, 26  
kuha\_osmh\_serve command line option, 29

-print-configuration  
     kuha\_ds\_serve,-kuha\_db\_setup command line option, 15  
     kuha\_oai\_serve command line option, 26  
     kuha\_osmh\_serve command line option, 29  
 -query-limit <limit>  
     kuha\_osmh\_serve command line option, 29  
 -remove-absent  
     command line option, 32  
 -stream-response  
     kuha\_osmh\_serve command line option, 29  
 -template-folder <folder>  
     kuha\_oai\_serve command line option, 26  
 -h, -help  
     command line option, 31  
     kuha\_ds\_serve,-kuha\_db\_setup command line option, 15  
     kuha\_oai\_serve command line option, 26  
     kuha\_osmh\_serve command line option, 29

## A

abstract (kuha\_common.document\_store.records.Study attribute), 60  
 add() (in module kuha\_common.cli\_setup), 40  
 add() (kuha\_common.cli\_setup.Settings method), 39  
 add\_abstract() (kuha\_common.document\_store.records.Study method), 63  
 add\_analysis\_units() (kuha\_common.document\_store.records.Study method), 64  
 add\_attribute() (kuha\_common.document\_store.mappings.xml\_base.XMLParser method), 73  
 add\_classifications() (kuha\_common.document\_store.records.Study method), 63  
 add\_codelist\_codes() (kuha\_common.document\_store.records.Study method), 67  
 add\_codelist\_references() (kuha\_common.document\_store.records.Question method), 69  
 add\_collection\_modes() (kuha\_common.document\_store.records.Study method), 64  
 add\_collection\_periods() (kuha\_common.document\_store.records.Study method), 64  
 add\_copyrights() (kuha\_common.document\_store.records.Study method), 66  
 add\_data\_access() (kuha\_common.document\_store.records.Study method), 65  
 add\_data\_access\_descriptions() (kuha\_common.document\_store.records.Study method), 65  
 add\_database\_configs() (in module kuha\_document\_store.configure), 84  
 add\_distributors() (kuha\_common.document\_store.records.Study method), 62

add\_document\_store\_api\_version() (in module kuha\_common.cli\_setup), 38  
 add\_document\_store\_client\_configs() (kuha\_common.cli\_setup.Settings method), 39  
 add\_document\_store\_client\_connect\_timeout() (in module kuha\_common.cli\_setup), 38  
 add\_document\_store\_client\_max\_clients() (in module kuha\_common.cli\_setup), 38  
 add\_document\_store\_client\_request\_timeout() (in module kuha\_common.cli\_setup), 38  
 add\_document\_store\_host() (in module kuha\_common.cli\_setup), 38  
 add\_document\_store\_port() (in module kuha\_common.cli\_setup), 38  
 add\_document\_store\_query\_configs() (kuha\_common.cli\_setup.Settings method), 39  
 add\_document\_store\_url() (in module kuha\_common.cli\_setup), 37  
 add\_document\_uris() (kuha\_common.document\_store.records.Study method), 62  
 add\_file\_names() (kuha\_common.document\_store.records.Study method), 65  
 add\_id() (kuha\_client.kuha\_client.FileLog method), 122  
 add\_identifiers() (kuha\_common.document\_store.records.Study method), 61  
 add\_instruments() (kuha\_common.document\_store.records.Study method), 65  
 add\_keywords() (kuha\_common.document\_store.records.Study method), 63  
 add\_kuha\_log\_level() (in module kuha\_common.cli\_setup), 37  
 add\_kuha\_loglevel() (in module kuha\_common.cli\_setup), 37  
 add\_logging\_configs() (kuha\_common.cli\_setup.Settings method), 39  
 add\_parallel\_titles() (kuha\_common.document\_store.records.Study method), 61  
 add\_pending\_study\_group() (kuha\_client.kuha\_client.FileLog method), 122  
 add\_persistent\_identifiers() (kuha\_common.document\_store.records.Study method), 61  
 add\_principalInvestigators() (kuha\_common.document\_store.records.Study method), 62  
 add\_print\_configuration() (in module kuha\_common.cli\_setup), 38  
 add\_publication\_dates() (kuha\_common.document\_store.records.Study method), 62  
 add\_publication\_years() (kuha\_common.document\_store.records.Study method), 63  
 add\_publishers() (kuha\_common.document\_store.records.Study method), 62



add\_query\_statement() (kuha\_common.document\_store.query.Query method), 47  
 add\_query\_statements() (kuha\_common.document\_store.query.Query method), 47  
 add\_question() (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIRecord method), 112  
 add\_question\_identifier() (kuha\_common.document\_store.records.Question method), 68  
 add\_question\_identifiers() (kuha\_common.document\_store.records.Variable method), 67  
 add\_question\_texts() (kuha\_common.document\_store.records.OAIRecord method), 68  
 add\_record() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIRecord method), 106  
 add\_research\_instruments() (kuha\_common.document\_store.records.Question method), 68  
 add\_sampling\_procedures() (kuha\_common.document\_store.records.Study method), 64  
 add\_sets\_element() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIRecord method), 107  
 add\_study\_area\_countries() (kuha\_common.document\_store.records.Study method), 64  
 add\_study\_group() (kuha\_client.kuha\_client.BatchProcessor method), 125  
 add\_study\_group\_identifier() (kuha\_common.document\_store.records.StudyGroup method), 69  
 add\_study\_group\_names() (kuha\_common.document\_store.records.StudyGroup method), 69  
 add\_study\_groups() (kuha\_common.document\_store.records.Study method), 66  
 add\_study\_number() (kuha\_common.document\_store.records.Question method), 68  
 add\_study\_number() (kuha\_common.document\_store.records.Study method), 61  
 add\_study\_number() (kuha\_common.document\_store.records.Variable method), 67  
 add\_study\_numbers() (kuha\_common.document\_store.records.Study method), 70  
 add\_study\_titles() (kuha\_common.document\_store.records.Study method), 61  
 add\_template\_folder() (in module kuha\_oai\_pmh\_repo\_handler.genshi\_loader), 98  
 add\_time\_methods() (kuha\_common.document\_store.records.Study method), 63  
 add\_universes() (kuha\_common.document\_store.records.Study method), 65  
 add\_value() (kuha\_common.document\_store.field\_types.Element method), 50  
 add\_value() (kuha\_common.document\_store.field\_types.ElementContainer method), 54  
 add\_value() (kuha\_common.document\_store.field\_types.LocalizableElement method), 52  
 add\_value() (kuha\_common.document\_store.field\_types.Set method), 49  
 add\_value() (kuha\_common.document\_store.field\_types.Value method), 48  
 add\_variable() (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIRecord method), 112  
 add\_variable\_labels() (kuha\_common.document\_store.records.Variable method), 67  
 add\_variable\_name() (kuha\_common.document\_store.records.Question method), 68  
 add\_variable\_name() (kuha\_common.document\_store.records.Variable method), 67  
 all (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase attribute), 76  
 analysis\_units (kuha\_common.document\_store.records.Study attribute), 60  
 apply() (kuha\_oai\_pmh\_repo\_handler.osmh.records.Payload method), 117  
 as\_dict() (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.MetadataForm method), 102  
 as\_local\_id() (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIHeaders class method), 111  
 as\_params() (kuha\_common.document\_store.mappings.xmlbase.XMLMap method), 73  
 as\_supported\_datestring() (in module kuha\_oai\_pmh\_repo\_handler.oai.protocol), 103  
 as\_supported\_datetime() (in module kuha\_oai\_pmh\_repo\_handler.oai.protocol), 103  
 as\_supported\_datetime\_str() (in module kuha\_oai\_pmh\_repo\_handler.oai.protocol), 103  
 class method), 43  
 (in module kuha\_common.document\_store.mappings.xmlbase), 76  
 (in module kuha\_document\_store.handlers.BaseHandler), 85  
 (in module kuha\_common.testing.testcases.KuhaEndToEndTestCase), 83  
 (in module kuha\_common.testing.testcases.KuhaUnitTestCase), 81  
 (in module kuha\_common.testing.testcases.KuhaUnitTestCase), 82

assert\_request\_content\_type() (kuha\_common.server.RequestHandler method), 34

assert\_single\_record() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 106

await\_and\_store\_result() (kuha\_common.testing.testcases.KuhaUnitTestCase method), 81

## B

BadArgument, 100

BadRequest, 35

BadResumptionToken, 101

BadVerb, 100

BaseHandler (class in kuha\_document\_store.handlers), 84

BaseHandler (class in kuha\_osmh\_repo\_handler.handlers), 114

BatchProcessor (class in kuha\_client.kuha\_client), 123

bson\_to\_json() (kuha\_document\_store.database.RecordsCollection class method), 87

build\_header\_payload() (kuha\_osmh\_repo\_handler.osmh.records.OSMRecord method), 118

build\_query\_for\_date\_range() (kuha\_common.document\_store.query.Query class method), 44

build\_query\_for\_exists() (kuha\_common.document\_store.query.Query class method), 44

build\_record\_payload() (kuha\_osmh\_repo\_handler.osmh.records.OSMRecord method), 118

build\_record\_payload() (kuha\_osmh\_repo\_handler.osmh.records.QuestionRecord method), 120

build\_record\_payload() (kuha\_osmh\_repo\_handler.osmh.records.StudyGroupRecord method), 121

build\_record\_payload() (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord method), 119

build\_record\_payload() (kuha\_osmh\_repo\_handler.osmh.records.VariableRecord method), 120

build\_relative\_record\_payload() (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord method), 119

bulk\_insert\_or\_update\_record() (kuha\_document\_store.database.DocumentStoreDatabase method), 92

bypass\_create() (kuha\_common.document\_store.records.RecordBase method), 58

bypass\_update() (kuha\_common.document\_store.records.RecordBase method), 58

## C

CannotDisseminateFormat, 100

CDCDDI25MetadataFormat (class in kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats), 103

child\_text() (kuha\_common.document\_store.mappings.xmlbase.XMLParser class method), 75

classifications (kuha\_common.document\_store.records.StudyRecord attribute), 60

cli() (in module kuha\_client.kuha\_delete), 127

cli() (in module kuha\_client.kuha\_import), 126

cli() (in module kuha\_client.kuha\_upsert), 127

close() (kuha\_document\_store.database.Database method), 88

cmm\_type (kuha\_common.document\_store.records.Question attribute), 68

cmm\_type (kuha\_common.document\_store.records.Study attribute), 61

cmm\_type (kuha\_common.document\_store.records.StudyGroup attribute), 69

cmm\_type (kuha\_common.document\_store.records.Variable attribute), 67

codelist\_codes (kuha\_common.document\_store.records.Variable attribute), 66

codelist\_references (kuha\_common.document\_store.records.Question attribute), 68

collection (kuha\_common.document\_store.records.Question attribute), 68

collection (kuha\_common.document\_store.records.Study attribute), 61

collection (kuha\_common.document\_store.records.StudyGroup attribute), 69

collection (kuha\_common.document\_store.records.Variable attribute), 67

collection\_modes (kuha\_common.document\_store.records.Study attribute), 60

collection\_periods (kuha\_common.document\_store.records.Study attribute), 60

command\_line\_option -collection <collection>, 31

command\_line\_option -document-store-url <document\_store\_url>, 31

command\_line\_option -file-log-path <path>, 32

command\_line\_option -remove-absent, 32

command\_line\_option -h, -help, 31

command\_line\_option paths, 31

configure() (in module kuha\_document\_store.configure), 84

configure() (in module kuha\_oai\_pmh\_repo\_handler.configure), 97

configure() (in module kuha\_osmh\_repo\_handler.configure), 113

connect\_timeout (kuha\_common.document\_store.client.JSONStreamClient attribute), 41

construct() (kuha\_common.document\_store.query.Query class method), 43

construct\_distinct() (kuha\_common.document\_store.query.Query class method), 44

copy() (kuha\_common.document\_store.mappings.xmlbase.MappedParams

method), 72

copyrights (kuha\_common.document\_store.records.Study attribute), 61

count() (kuha\_document\_store.database.Database method), 89

create() (kuha\_client.kuha\_client.BatchProcessor method), 124

## D

data\_access (kuha\_common.document\_store.records.Study attribute), 60

data\_access\_descriptions (kuha\_common.document\_store.records.Study attribute), 60

Database (class in kuha\_document\_store.database), 88

datestamp\_to\_datetime() (in module kuha\_common.document\_store.records), 56

datetime\_now() (in module kuha\_common.document\_store.records), 56

datetime\_to\_datestamp() (in module kuha\_common.document\_store.records), 56

DCMetadataFormat (class in kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats), 102

DDI122RecordParser (class in kuha\_common.document\_store.mappings.ddi), 77

DDI25RecordParser (class in kuha\_common.document\_store.mappings.ddi), 78

DDI31RecordParser (class in kuha\_common.document\_store.mappings.ddi), 78

DDIMetadataFormat (class in kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats), 102

decode\_uri() (in module kuha\_oai\_pmh\_repo\_handler.oai.protocol), 104

default\_language (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase attribute), 75

delete() (kuha\_document\_store.handlers.RestApiHandler method), 86

delete\_by\_oid() (kuha\_document\_store.database.DocumentStoreDatabase method), 93

delete\_collections() (in module kuha\_document\_store.db\_setup), 96

delete\_database() (in module kuha\_document\_store.db\_setup), 96

delete\_many() (kuha\_document\_store.database.Database method), 90

delete\_one() (kuha\_document\_store.database.Database method), 90

DELETE\_to\_document\_store() (kuha\_common.testing.testcases.KuhaEndToEndTestCase class method), 83

disable\_attributes() (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase method), 72

distributors (kuha\_common.document\_store.records.Study attribute), 60

document\_uris (kuha\_common.document\_store.records.Study attribute), 60

DocumentStoreDatabase (class in kuha\_document\_store.database), 90

DocumentStoreHTTPError, 123

dummy() (kuha\_common.testing.MockCoro method), 79

dummydata\_dir (kuha\_common.testing.testcases.KuhaUnitTestCase attribute), 80

## E

Element (class in kuha\_common.document\_store.field\_types), 49

element\_remove\_whitespaces() (in module kuha\_common.document\_store.mappings.xmlbase), 77

element\_strip\_descendant\_text() (in module kuha\_common.document\_store.mappings.xmlbase), 77

ElementContainer (class in kuha\_common.document\_store.field\_types), 52

encode\_uri() (in module kuha\_oai\_pmh\_repo\_handler.oai.protocol), 104

execute\_stored\_callbacks() (kuha\_common.document\_store.client.JSONStreamClient method), 41

expect\_multiple\_values() (kuha\_common.document\_store.mappings.xmlbase.XMLMapper method), 72

expect\_single\_value() (kuha\_common.document\_store.mappings.xmlbase.XMLMapper method), 72

export\_attributes\_as\_dict() (kuha\_common.document\_store.field\_types.Element method), 51

export\_dict() (kuha\_common.document\_store.field\_types.Element method), 51

export\_dict() (kuha\_common.document\_store.field\_types.ElementContainer method), 54

export\_dict() (kuha\_common.document\_store.field\_types.LocalizableElement method), 52

export\_dict() (kuha\_common.document\_store.field\_types.Value method), 48

export\_dict() (kuha\_common.document\_store.records.RecordBase method), 57

export\_metadata\_dict() (kuha\_common.document\_store.records.RecordBack class method), 57

extend\_sets\_element() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse class method), 107

## F

fabricate() (kuha\_common.document\_store.field\_types.FieldType class method), 56

fetch() (kuha\_common.document\_store.client.JSONStreamClient class method), 42

FieldAttribute (class in kuha\_common.document\_store.field\_types), 55

fields\_for\_header() (kuha\_osmh\_repo\_handler.osmh.records.OSMHRRecord class method), 117

fields\_for\_header() (kuha\_osmh\_repo\_handler.osmh.records.QuestionRecord class method), 120

fields\_for\_header() (kuha\_osmh\_repo\_handler.osmh.records.StudyGroupRecord class method), 121

fields\_for\_header() (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord class method), 118

fields\_for\_header() (kuha\_osmh\_repo\_handler.osmh.records.VariableRecord class method), 119

fields\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.OSMHRRecord class method), 117

fields\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.QuestionRecord class method), 120

fields\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.StudyGroupRecord class method), 121

fields\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord class method), 119

fields\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.VariableRecord class method), 120

FieldTypeException, 48

FieldTypeFactory (class in kuha\_common.document\_store.field\_types), 55

file\_names (kuha\_common.document\_store.records.StudyRecord attribute), 60

FileLog (class in kuha\_client.kuha\_client), 121

FilterKeyConstants (class in kuha\_common.document\_store.query), 42

fixed\_value() (in module kuha\_common.document\_store.mappings.xmlbase), 77

fk\_constants (kuha\_common.query.QueryController attribute), 35

## FOLDERS

(in module kuha\_oai\_pmh\_repo\_handler.genshi\_loader), 98

for\_header\_response() (kuha\_osmh\_repo\_handler.osmh.records.OSMHRRecord class method), 118

for\_record\_response() (kuha\_osmh\_repo\_handler.osmh.records.OSMHRRecord class method), 118

from\_request\_args() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse static method), 109

from\_string() (kuha\_common.document\_store.mappings.xmlbase.XMLParser class method), 75

## G

gen\_id() (kuha\_common.testing.testcases.KuhaUnitTestCase class method), 80

gen\_val() (kuha\_common.testing.testcases.KuhaUnitTestCase class method), 80

generate\_dummy\_question() (kuha\_common.testing.testcases.KuhaUnitTestCase class method), 81

generate\_dummy\_studygroup() (kuha\_common.testing.testcases.KuhaUnitTestCase class method), 81

generate\_dummy\_variable() (kuha\_common.testing.testcases.KuhaUnitTestCase class method), 81

get() (kuha\_common.cli\_setup.Settings method), 39

get() (kuha\_oai\_pmh\_repo\_handler.handlers.OAIRouteHandler method), 99

get() (kuha\_oai\_pmh\_repo\_handler.handlers.GetRecordHandler method), 114

get() (kuha\_oai\_pmh\_repo\_handler.handlers.ListRecordHeadersHandler method), 114

get() (kuha\_oai\_pmh\_repo\_handler.handlers.ListSupportedRecordTypesHandler method), 114

get() (kuha\_oai\_pmh\_repo\_handler.handlers.SupportedVersionsHandler method), 115

get() (kuha\_osmh\_repo\_handler.osmh.records.Payload method), 117

get\_abs\_dir\_path() (kuha\_common.cli\_setup.Settings method), 39

get\_app() (in module kuha\_document\_store.serve), 84

get\_app() (in module kuha\_oai\_pmh\_repo\_handler.serve), 97

get\_app() (in module kuha\_osmh\_repo\_handler.serve), 113

get\_attribute() (kuha\_common.document\_store.field\_types.Element method), 51

get\_languages() (kuha\_common.document\_store.field\_types.ElementContainer method), 54

[get\\_code\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.errors.OAIError method\), 109](#)  
[get\\_code\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.errors.OAIError method\), 100](#)  
[get\\_collection\(\) \(kuha\\_common.document\\_store.records.RecordBase class method\), 57](#)  
[get\\_collection\\_record\\_count\(\) \(kuha\\_common.testing.testcases.KuhaEndToEndTestCase class method\), 83](#)  
[get\\_complete\\_list\\_size\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.ResumptionToken class method\), 105](#)  
[get\\_context\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.errors.OAIError method\), 100](#)  
[get\\_contextual\\_message\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.errors.OAIError method\), 100](#)  
[get\\_created\(\) \(kuha\\_common.document\\_store.records.RecordBase class method\), 58](#)  
[get\\_cursor\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.ResumptionToken class method\), 105](#)  
[get\\_datestamp\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.records.OAIRecord class method\), 112](#)  
[get\\_db\(\) \(kuha\\_document\\_store.handlers.BaseHandler class method\), 85](#)  
[get\\_dummydata\(\) \(kuha\\_common.testing.testcases.KuhaUnitTestCase class method\), 80](#)  
[get\\_dummydata\\_path\(\) \(kuha\\_common.testing.testcases.KuhaUnitTestCase class method\), 80](#)  
[get\\_encoded\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.ResumptionToken class method\), 106](#)  
[get\\_endpoint\(\) \(kuha\\_common.document\\_store.query.Query class method\), 45](#)  
[get\\_filepaths\(\) \(kuha\\_client.kuha\\_client.FileLog class method\), 122](#)  
[get\\_from\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.ResumptionToken class method\), 105](#)  
[get\\_header\\_fields\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.records.OAIRecord class method\), 111](#)  
[get\\_id\(\) \(kuha\\_common.document\\_store.records.RecordBase class method\), 58](#)  
[get\\_identifier\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.OAIArgument class method\), 108](#)  
[get\\_identifier\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.records.OAIRecord class method\), 112](#)  
[get\\_ids\(\) \(kuha\\_client.kuha\\_client.FileLog class method\), 122](#)  
[get\\_import\\_url\(\) \(in module kuha\\_client.kuha\\_client\), 123](#)  
[get\\_language\(\) \(kuha\\_common.document\\_store.field\\_types.LocalizableRecords class method\), 52](#)  
[get\\_language\(\) \(kuha\\_common.document\\_store.mappings.xbase.Mapping class method\), 71](#)  
[get\\_limit\(\) \(kuha\\_common.document\\_store.query.Query class method\), 46](#)  
[get\\_local\\_identifier\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.OAIArgument class method\), 109](#)  
[get\\_metadata\\_format\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.OAIArgument class method\), 109](#)  
[get\\_metadata\\_prefix\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.ResumptionToken class method\), 101](#)  
[get\\_msg\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.errors.OAIError class method\), 100](#)  
[get\\_name\(\) \(kuha\\_common.document\\_store.field\\_types.Value class method\), 48](#)  
[get\\_name\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.metadata\\_formats.Metadata class method\), 101](#)  
[get\\_osmsh\\_record\\_for\\_type\(\) \(in module kuha\\_osmsh\\_repo\\_handler.osmh.records\), 121](#)  
[get\\_payload\(\) \(kuha\\_osmsh\\_repo\\_handler.osmh.records.OSMHRRecord class method\), 118](#)  
[get\\_payload\\_appender\(\) \(kuha\\_osmsh\\_repo\\_handler.response.RecordsResponse class method\), 115](#)  
[get\\_prefix\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.metadata\\_formats.Metadata class method\), 101](#)  
[get\\_query\(\) \(kuha\\_common.document\\_store.query.Query class method\), 45](#)  
[get\\_query\\_document\(\) \(kuha\\_osmsh\\_repo\\_handler.osmh.records.OSMHRRecord class method\), 118](#)  
[get\\_query\\_param\\_until\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.ResumptionToken class method\), 105](#)  
[get\\_query\\_url\(\) \(kuha\\_common.testing.testcases.KuhaEndToEndTestCase class method\), 82](#)  
[get\\_questions\\_by\\_variable\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.records.OAIRecord class method\), 112](#)  
[get\\_record\\_appender\(\) \(kuha\\_osmsh\\_repo\\_handler.response.RecordsResponse class method\), 115](#)  
[get\\_record\\_fields\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.metadata\\_formats.CD class method\), 103](#)  
[get\\_record\\_fields\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.metadata\\_formats.DC class method\), 102](#)  
[get\\_record\\_fields\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.metadata\\_formats.DD class method\), 102](#)  
[get\\_record\\_fields\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.metadata\\_formats.Me class method\), 102](#)  
[get\\_headers\\_query\\_field\\_by\\_setspec\(\) \(in module kuha\\_oai\\_pmh\\_repo\\_handler.oai.records\), 110](#)  
[get\\_record\\_url\(\) \(kuha\\_common.testing.testcases.KuhaEndToEndTestCase class method\), 82](#)  
[get\\_relatable\\_records\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.metadata\\_formats.Me class method\), 102](#)  
[get\\_response\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.OAIResponse class method\), 107](#)  
[get\\_response\(\) \(kuha\\_osmsh\\_repo\\_handler.response.RecordsResponse class method\), 115](#)  
[get\\_resumption\\_token\(\) \(kuha\\_oai\\_pmh\\_repo\\_handler.oai.protocol.OAIArgument class method\), 108](#)  
[get\\_schema\(\) \(kuha\\_common.document\\_store.validation.RecordValidationSchema class method\), 101](#)

method), 94

get\_schema() (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_format\_handler.FormatBase (class in kuha\_osmh\_repo\_handler.handlers), 114 method), 101

get\_secondary\_query\_document() (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord static method), 119

get\_secondary\_query\_fields\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord static method), 119

get\_secondary\_query\_filter\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord method), 119

get\_set() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments method), 109

get\_set\_specs\_from\_ds\_record() (in module kuha\_oai\_pmh\_repo\_handler.oai.records), 110

get\_sets\_list\_from\_query\_result() (in module kuha\_oai\_pmh\_repo\_handler.oai.records), 110

get\_settings() (in module kuha\_common.cli\_setup), 40

get\_single\_response() (kuha\_osmh\_repo\_handler.response.RecordsResponse method), 115

get\_skip() (kuha\_common.document\_store.query.Query method), 46

get\_streaming\_request() (kuha\_common.document\_store.client.JSONStreamClient method), 41

get\_supported\_sourcefiletypes() (kuha\_client.kuha\_client.BatchProcessor class method), 124

get\_syntax\_description() (kuha\_common.cli\_setup.KuhaConfigFileParser method), 38

get\_template\_folder() (in module kuha\_oai\_pmh\_repo\_handler.genshi\_loader), 98

get\_template\_writer() (in module kuha\_oai\_pmh\_repo\_handler.genshi\_loader), 98

GET\_to\_document\_store() (kuha\_common.testing.testcases.KuhaEndToEndTestCase class method), 82

get\_until() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.ResumptionToken method), 105

get\_updated() (kuha\_common.document\_store.records.RecordBase method), 58

get\_valid\_params() (kuha\_common.document\_store.query.Query class method), 44

get\_validator() (kuha\_document\_store.database.RecordsCollection method), 87

get\_value() (kuha\_common.document\_store.field\_types.Value method), 48

get\_value() (kuha\_common.document\_store.mappings.xmlbase.MappedParams method), 71

get\_verb() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments method), 108

GetRecordHandler (class in kuha\_document\_store.handlers), 86

has\_arguments() (kuha\_common.document\_store.mappings.xmlbase.MappedParams method), 72

has\_language() (kuha\_common.document\_store.mappings.xmlbase.MappedParams method), 71

has\_match() (kuha\_client.kuha\_client.FileLog method), 122

has\_records() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 106

has\_set() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments method), 109

header() (kuha\_osmh\_repo\_handler.osmh.records.Payload method), 117

**I**

IdDoesNotExist, 100

identifier\_oai\_prefix (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIHeaders attribute), 111

identifiers (kuha\_common.document\_store.records.StudyRecord attribute), 59

import\_records() (kuha\_common.document\_store.field\_types.Element method), 51

import\_records() (kuha\_common.document\_store.field\_types.ElementContainer method), 54

import\_records() (kuha\_common.document\_store.field\_types.Set method), 49

import\_records() (kuha\_common.document\_store.field\_types.Value method), 48

import\_run() (in module kuha\_client.kuha\_import), 126

import\_run() (kuha\_client.kuha\_client.BatchProcessor method), 126

import\_source() (kuha\_client.kuha\_client.BatchProcessor method), 125

import\_source\_files() (kuha\_client.kuha\_client.BatchProcessor method), 125

importers (in module kuha\_document\_store.importers), 97

ImportHandler (class in kuha\_document\_store.handlers), 86

index\_updated (kuha\_document\_store.database.RecordsCollection attribute), 87

init\_patcher() (kuha\_common.testing.testcases.KuhaUnitTestCase method), 81

insert() (kuha\_document\_store.database.Database method), 89

insert() (kuha\_osmh\_repo\_handler.osmh.records.Payload method), 116

insert\_json() (kuha\_document\_store.database.DocumentStoreDatabase method), 92

insert\_localized\_value() (kuha\_osmh\_repo\_handler.osmh.records.Payload method), 117

insert\_or\_replace() (kuha\_document\_store.database.Database method), 90

insert\_or\_update\_record() (kuha\_document\_store.database.DocumentStoreDatabase class method), 92

instruments (kuha\_common.document\_store.records.Study attribute), 61

InvalidContent, 71

InvalidContentType, 34

InvalidMapperParameters, 70

is\_parser\_loaded() (kuha\_common.cli\_setup.Settings method), 38

is\_pending() (kuha\_common.document\_store.field\_types.Element method), 50

is\_selective() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments method), 109

is\_settings\_loaded() (kuha\_common.cli\_setup.Settings method), 38

is\_valid\_param() (kuha\_common.document\_store.query.Query method), 45

is\_valid\_query() (kuha\_common.document\_store.query.Query class method), 44

is\_valid\_query\_document() (kuha\_common.document\_store.query.Query method), 44

is\_valid\_query\_type() (kuha\_common.document\_store.query.Query class method), 44

is\_valid\_setspec() (in module kuha\_oai\_pmh\_repo\_handler.oai.records), 110

is\_verb\_resumable() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments method), 108

isodate\_fields (kuha\_document\_store.database.RecordsCollection attribute), 87

iterate\_attributes() (kuha\_common.document\_store.field\_types.Element method), 51

iterate\_attributes() (kuha\_common.document\_store.mappings.xmlbase.XMLMapper method), 72

iterate\_record\_fields() (kuha\_common.document\_store.records.RecordBase class method), 57

iterate\_records() (kuha\_osmh\_repo\_handler.response.RecordsResponse method), 115

iterate\_set\_specs() (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIHeaders method), 112

iterate\_supported\_metadata\_formats() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments method), 109

iterate\_values\_for\_language() (kuha\_common.document\_store.field\_types.ElementContainer method), 54

iterate\_xml\_directory() (in module kuha\_client.kuha\_client), 123

insert\_payload\_files\_recursively() (in module kuha\_client.kuha\_client), 123

join\_values() (kuha\_osmh\_repo\_handler.osmh.records.Payload Database class method), 116

json\_decode() (kuha\_document\_store.database.DocumentStoreDatabase static method), 90

JSONStreamClient (class in kuha\_common.document\_store.client), 40

**K**

k\_fieldname (kuha\_common.document\_store.query.Query attribute), 43

k\_fields (kuha\_common.document\_store.query.Query attribute), 43

k\_filter (kuha\_common.document\_store.query.Query attribute), 43

k\_limit (kuha\_common.document\_store.query.Query attribute), 43

k\_skip (kuha\_common.document\_store.query.Query attribute), 43

k\_sort\_by (kuha\_common.document\_store.query.Query attribute), 43

k\_sort\_order (kuha\_common.document\_store.query.Query attribute), 43

key (kuha\_oai\_pmh\_repo\_handler.oai.protocol.ResumptionToken.Attribute attribute), 105

keywords (kuha\_common.document\_store.records.Study attribute), 60

kuha\_client.kuha\_client (module), 121

kuha\_client.kuha\_delete (module), 127

kuha\_client.kuha\_import (module), 126

kuha\_client.kuha\_upsert (module), 126

kuha\_common (module), 33

kuha\_common.cli\_setup (module), 37

kuha\_common.document\_store (module), 40

kuha\_common.document\_store.client (module), 40

kuha\_common.document\_store.field\_types (module), 48

kuha\_common.document\_store.mappings (module), 70

kuha\_common.document\_store.mappings.ddi (module), 77

kuha\_common.document\_store.mappings.exceptions (module), 70

kuha\_common.document\_store.mappings.xmlbase (module), 71

kuha\_common.document\_store.query (module), 42

kuha\_common.document\_store.records (module), 56

kuha\_common.query (module), 35

kuha\_common.server (module), 33

kuha\_common.testing (module), 79

kuha\_common.testing.testcases (module), 80

kuha\_document\_store (module), 83

kuha\_document\_store.configure (module), 84

- kuha\_document\_store.database (module), 87
  - kuha\_document\_store.db\_setup (module), 95
  - kuha\_document\_store.handlers (module), 84
  - kuha\_document\_store.importers (module), 97
  - kuha\_document\_store.serve (module), 84
  - kuha\_document\_store.validation (module), 93
  - kuha\_ds\_serve.-kuha\_db\_setup command line option
    - database-host <database\_host>, 16
    - database-name <name>, 16
    - database-pass-editor <password>, 16
    - database-pass-reader <password>, 16
    - database-port <port>, 16
    - database-user-editor <user>, 16
    - database-user-reader <user>, 16
    - document-store-api-version <api\_version>, 15
    - document-store-port <port>, 15
    - logformat <logformat>, 16
    - loglevel <loglevel>, 16
    - print-configuration, 15
    - h, -help, 15
  - kuha\_oai\_pmh\_repo\_handler (module), 97
  - kuha\_oai\_pmh\_repo\_handler.configure (module), 97
  - kuha\_oai\_pmh\_repo\_handler.genshi\_loader (module), 98
  - kuha\_oai\_pmh\_repo\_handler.handlers (module), 99
  - kuha\_oai\_pmh\_repo\_handler.oai (module), 100
  - kuha\_oai\_pmh\_repo\_handler.oai.constants (module), 101
  - kuha\_oai\_pmh\_repo\_handler.oai.errors (module), 100
  - kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats (module), 101
  - kuha\_oai\_pmh\_repo\_handler.oai.protocol (module), 103
  - kuha\_oai\_pmh\_repo\_handler.oai.records (module), 110
  - kuha\_oai\_pmh\_repo\_handler.serve (module), 97
  - kuha\_oai\_serve command line option
    - document-store-api-version <api\_version>, 27
    - document-store-client-connect-timeout <timeout>, 27
    - document-store-client-max-clients <max\_clients>, 27
    - document-store-client-request-timeout <timeout>, 27
    - document-store-host <host>, 27
    - document-store-port <port>, 27
    - logformat <logformat>, 27
    - loglevel <loglevel>, 27
    - oai-pmh-admin-email <email>, 27
    - oai-pmh-api-version <api\_version>, 27
    - oai-pmh-base-url <base\_url>, 27
    - oai-pmh-namespace-identifier <namespace\_id>, 27
    - oai-pmh-protocol-version <version>, 27
    - oai-pmh-repo-name <repo\_name>, 27
    - oai-pmh-results-per-list <results\_per\_list>, 27
    - port <port>, 26
    - print-configuration, 26
    - template-folder <folder>, 26
    - h, -help, 26
  - kuha\_osmh\_repo\_handler (module), 113
  - kuha\_osmh\_repo\_handler.configure (module), 113
  - kuha\_osmh\_repo\_handler.handlers (module), 114
  - kuha\_osmh\_repo\_handler.osmh (module), 116
  - kuha\_osmh\_repo\_handler.osmh.records (module), 116
  - kuha\_osmh\_repo\_handler.response (module), 115
  - kuha\_osmh\_repo\_handler.serve (module), 113
  - kuha\_osmh\_serve command line option
    - document-store-api-version <api\_version>, 30
    - document-store-client-connect-timeout <timeout>, 30
    - document-store-client-max-clients <max\_clients>, 30
    - document-store-client-request-timeout <timeout>, 30
    - document-store-host <host>, 29
    - document-store-port <port>, 29
    - logformat <logformat>, 30
    - loglevel <loglevel>, 30
    - osmh-repo-handler-api-version <api\_version>, 29
    - port <port>, 29
    - print-configuration, 29
    - query-limit <limit>, 29
    - stream-response, 29
    - h, -help, 29
  - KuhaConfigFileParser (class in kuha\_common.cli\_setup), 38
  - KuhaEndToEndTestCase (class in kuha\_common.testing.testcases), 82
  - KuhaServerError, 34
  - KuhaUnitTestCase (class in kuha\_common.testing.testcases), 80
- L**
- list\_collections() (in module kuha\_document\_store.db\_setup), 96
  - list\_databases() (in module kuha\_document\_store.db\_setup), 96
  - list\_db\_users() (in module kuha\_document\_store.db\_setup), 96
  - ListRecordHeadersHandler (class in kuha\_osmh\_repo\_handler.handlers), 114
  - ListSupportedRecordTypesHandler (class in kuha\_osmh\_repo\_handler.handlers), 114
  - load() (in module kuha\_common.cli\_setup), 40
  - load() (kuha\_client.kuha\_client.FileLog method), 122
  - load\_cli\_args() (kuha\_common.cli\_setup.Settings method), 39
  - load\_cli\_args() (kuha\_common.testing.testcases.KuhaEndToEndTestCase static method), 82
  - load\_parser() (kuha\_common.cli\_setup.Settings method), 39





- post() (kuha\_document\_store.handlers.QueryHandler method), 86
- post() (kuha\_document\_store.handlers.RestApiHandler method), 85
- post() (kuha\_oai\_pmh\_repo\_handler.handlers.OAIRouteHandler method), 99
- POST\_to\_document\_store() (kuha\_common.testing.testcases.KuhaEndToEndTestCase class method), 83
- prefix (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.CDCDDI25MetadataFormat attribute), 103
- prefix (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.DCMetadataFormat attribute), 102
- prefix (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.DDIMetadataFormat attribute), 102
- prefix (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.MetadataFormat attribute), 101
- prepare() (kuha\_common.server.RequestHandler method), 34
- prepare() (kuha\_document\_store.handlers.BaseHandler method), 84
- prepare() (kuha\_document\_store.handlers.ImportHandler method), 86
- prepare() (kuha\_document\_store.handlers.QueryHandler method), 86
- prepare() (kuha\_oai\_pmh\_repo\_handler.handlers.OAIRouteHandler method), 99
- prepare() (kuha\_osmh\_repo\_handler.handlers.BaseHandler method), 114
- prepare() (kuha\_osmh\_repo\_handler.handlers.ListRecordHeadersHandler method), 114
- prepend\_abs\_dir\_path() (in module kuha\_common.cli\_setup), 40
- principal\_investigators (kuha\_common.document\_store.records.Study attribute), 60
- process\_json\_for\_upsert() (kuha\_document\_store.database.RecordsCollection method), 88
- publication\_dates (kuha\_common.document\_store.records.Study attribute), 60
- publication\_years (kuha\_common.document\_store.records.Study attribute), 60
- publishers (kuha\_common.document\_store.records.Study attribute), 60
- put() (kuha\_document\_store.handlers.RestApiHandler method), 85
- Q**
- Query (class in kuha\_common.document\_store.query), 42
- query\_by\_oid() (kuha\_document\_store.database.DocumentStoreDatabase method), 91
- query\_count() (kuha\_common.query.QueryController method), 36
- query\_distinct() (kuha\_common.query.QueryController method), 37
- query\_distinct() (kuha\_document\_store.database.Database method), 89
- query\_distinct() (kuha\_document\_store.database.DocumentStoreDatabase method), 91
- query\_distinct\_ids() (in module kuha\_client.kuha\_client), 123
- query\_document (kuha\_osmh\_repo\_handler.osmh.records.OSMHRRecord attribute), 118
- query\_document (kuha\_osmh\_repo\_handler.osmh.records.QuestionRecord attribute), 120
- query\_document (kuha\_osmh\_repo\_handler.osmh.records.StudyGroupRecord attribute), 121
- query\_document (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord attribute), 118
- query\_document (kuha\_osmh\_repo\_handler.osmh.records.VariableRecord attribute), 119
- query\_document\_store() (kuha\_common.testing.testcases.KuhaEndToEndTestCase class method), 83
- query\_filter\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.OSMHRRecord static method), 118
- query\_filter\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.QuestionRecord static method), 120
- query\_filter\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.StudyGroupRecord static method), 121
- query\_filter\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.StudyRecord static method), 119
- query\_filter\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.VariableRecord static method), 120
- query\_multiple() (kuha\_common.query.QueryController method), 36
- query\_multiple() (kuha\_document\_store.database.Database method), 88
- query\_multiple() (kuha\_document\_store.database.DocumentStoreDatabase method), 91
- query\_record() (in module kuha\_client.kuha\_client), 123
- query\_single() (kuha\_common.query.QueryController method), 35
- query\_single() (kuha\_document\_store.database.Database method), 88
- query\_type\_count (kuha\_common.document\_store.query.Query attribute), 43
- query\_type\_distinct (kuha\_common.document\_store.query.Query attribute), 43
- query\_type\_select (kuha\_common.document\_store.query.Query attribute), 43
- QueryController (class in kuha\_common.query), 35
- QueryException, 42

QueryHandler (class in kuha\_document\_store.handlers), 86  
 Question (class in kuha\_common.document\_store.records), 67  
 question\_identifier (kuha\_common.document\_store.records.Question attribute), 68  
 question\_identifiers (kuha\_common.document\_store.records.Question attribute), 66  
 question\_texts (kuha\_common.document\_store.records.Question attribute), 68  
 QuestionRecord (class in kuha\_osmh\_repo\_handler.osmh.records), 120  
 questions (kuha\_common.document\_store.mappings.ddi.DDI122Record attribute), 78  
 questions (kuha\_common.document\_store.mappings.ddi.DDI25Record attribute), 78  
 questions (kuha\_common.document\_store.mappings.ddi.DDI31Record attribute), 79  
 questions (kuha\_common.document\_store.mappings.xmlbase.XMLBase attribute), 76  
 queue\_request() (kuha\_common.document\_store.client.JSONStreamClient method), 42  
**R**  
 record\_by\_collection() (in module kuha\_common.document\_store.records), 70  
 RECORD\_COLLECTIONS (in module kuha\_document\_store.database), 88  
 record\_factory() (in module kuha\_common.document\_store.records), 70  
 RecordBase (class in kuha\_common.document\_store.records), 56  
 RecordsCollection (class in kuha\_document\_store.database), 87  
 RecordsResponse (class in kuha\_osmh\_repo\_handler.response), 115  
 RecordValidationError, 94  
 RecordValidationSchema (class in kuha\_document\_store.validation), 94  
 RecordValidator (class in kuha\_document\_store.validation), 93  
 recoverable\_errors (kuha\_document\_store.database.DocumentStoreDatabase attribute), 90  
 REGEX\_OAI\_IDENTIFIER (in module kuha\_oai\_pmh\_repo\_handler.oai.constants), 101  
 REGEX\_SETSPEC (in module kuha\_oai\_pmh\_repo\_handler.oai.constants), 101  
 REGEX\_VALID\_SETSPEC (in module kuha\_oai\_pmh\_repo\_handler.oai.records), 101  
 relative\_queries\_for\_record (kuha\_osmh\_repo\_handler.osmh.records.OSMHRRecord attribute), 117  
 Question absent() (kuha\_client.kuha\_client.BatchProcessor method), 125  
 Variable absent\_records() (kuha\_client.kuha\_client.BatchProcessor method), 125  
 remove\_dummyfile\_if\_exists() (kuha\_common.testing.testcases.KuhaUnitTestCase class method), 80  
 remove\_files\_by\_path\_difference() (kuha\_client.kuha\_client.FileLog method), 122  
 RecordBase parse() (kuha\_client.kuha\_client.BatchProcessor method), 125  
 RecordBase parse() (kuha\_client.kuha\_client.BatchProcessor method), 126  
 XMLBase parse\_base\_relatives\_by\_studyid() (kuha\_client.kuha\_client.BatchProcessor method), 125  
 remove\_users() (in module kuha\_document\_store.db\_setup), 95  
 replace() (kuha\_document\_store.database.Database method), 89  
 replace\_json() (kuha\_document\_store.database.DocumentStoreDatabase method), 92  
 request() (kuha\_common.document\_store.client.JSONStreamClient class method), 41  
 request\_timeout (kuha\_common.document\_store.client.JSONStreamClient attribute), 41  
 RequestHandler (class in kuha\_common.server), 33  
 requires\_relative\_queries\_for\_record() (kuha\_osmh\_repo\_handler.osmh.records.OSMHRRecord class method), 118  
 research\_instruments (kuha\_common.document\_store.records.Question attribute), 68  
 ResourceNotFound, 35  
 response\_list\_size (kuha\_oai\_pmh\_repo\_handler.oai.protocol.ResumptionToken attribute), 105  
 RestApiHandler (class in kuha\_document\_store.handlers), 85  
 ResultHandler (class in kuha\_common.query), 35  
 resumable\_verbs (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments attribute), 108  
 ResumptionToken (class in kuha\_oai\_pmh\_repo\_handler.oai.protocol), 104  
 ResumptionToken.Attribute (class in kuha\_oai\_pmh\_repo\_handler.oai.protocol), 104  
 root\_element (kuha\_common.document\_store.mappings.xmlbase.XMLParse attribute), 75

root\_language (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase.document\_store.records.RecordBase attribute), 75  
 run\_queued\_requests() (kuha\_common.document\_store.client.JSONStreamClient.set\_language() (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIHeaders method), 41  
 SET\_LANGUAGE (in module kuha\_oai\_pmh\_repo\_handler.oai.records), 110  
 S  
 sampling\_procedures (kuha\_common.document\_store.records.Study attribute), 60  
 save() (kuha\_client.kuha\_client.FileLog method), 122  
 schema (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.DCIMetadataFormat attribute), 103  
 schema (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.DCIMetadataFormat attribute), 102  
 schema (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.DCIMetadataFormat attribute), 102  
 schema (kuha\_oai\_pmh\_repo\_handler.oai.metadata\_formats.MetadataFormat attribute), 101  
 select() (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase.select() (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIHeaders class method), 111  
 serve() (in module kuha\_common.server), 33  
 Set (class in kuha\_common.document\_store.field\_types), 49  
 set() (kuha\_common.cli\_setup.Settings method), 39  
 set\_abs\_dir\_path() (kuha\_common.cli\_setup.Settings method), 38  
 set\_admin\_email() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse class method), 106  
 set\_attribute() (kuha\_common.document\_store.field\_types.Element method), 51  
 set\_base\_url() (kuha\_common.document\_store.query.Query class method), 43  
 set\_base\_url() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse class method), 106  
 set\_cmm\_type() (kuha\_common.document\_store.records.RecordBase method), 57  
 set\_complete\_list\_size() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse class method), 105  
 set\_connect\_timeout() (kuha\_common.document\_store.client.JSONStreamClient.set\_connect\_timeout() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse class method), 107  
 set\_created() (kuha\_common.document\_store.records.RecordBase method), 57  
 set\_current() (kuha\_client.kuha\_client.FileLog method), 122  
 set\_deleted\_records\_declaration() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 107  
 set\_earliest\_datestamp() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 106  
 set\_error() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 107  
 set\_fields() (kuha\_common.document\_store.query.Query method), 46  
 set\_granularity() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 107  
 set\_language() (kuha\_common.document\_store.field\_types.LocalizableElement method), 52  
 set\_languages() (kuha\_common.document\_store.mappings.xmlbase.MappedElement method), 71  
 set\_limit() (kuha\_common.document\_store.query.Query method), 46  
 set\_login\_clients() (kuha\_common.document\_store.client.JSONStreamClient class method), 41  
 set\_metadata\_formats() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 107  
 set\_namespace\_identifier() (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIHeaders class method), 111  
 set\_output\_content\_type() (kuha\_common.server.RequestHandler method), 34  
 set\_protocol\_version() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse class method), 106  
 set\_query\_type() (kuha\_common.document\_store.query.Query method), 47  
 set\_repository\_name() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse class method), 106  
 set\_request\_params() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 107  
 set\_response\_timeout() (kuha\_common.document\_store.client.JSONStreamClient class method), 41  
 set\_response\_list\_size() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse class method), 105  
 set\_resumption\_token() (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIResponse method), 107  
 set\_sort() (kuha\_common.document\_store.query.Query method), 46  
 set\_sort\_by() (kuha\_common.document\_store.query.Query method), 47  
 set\_sort\_order() (kuha\_common.document\_store.query.Query method), 47  
 SET\_STUDY\_GROUP (in module kuha\_oai\_pmh\_repo\_handler.oai.records), 110  
 set\_template\_writer() (in module kuha\_oai\_pmh\_repo\_handler.genshi\_loader), 98  
 set\_updated() (kuha\_common.document\_store.records.RecordBase method), 57  
 set\_val() (kuha\_common.testing.testcases.KuhaUnitTestCase class method), 80  
 set\_value() (kuha\_common.document\_store.field\_types.Set

method), 49

set\_value() (kuha\_common.document\_store.field\_types.Value attribute), 48

set\_value\_conversion() (kuha\_common.document\_store.mappings.xmlbase.XMLMapper attribute), 72

set\_value\_getter() (kuha\_common.document\_store.mappings.xmlbase.XMLMapper attribute), 72

SetAttribute (in module kuha\_oai\_pmh\_repo\_handler.oai.records), 110

SETS (in module kuha\_oai\_pmh\_repo\_handler.oai.records), 110

Settings (class in kuha\_common.cli\_setup), 38

setup() (in module kuha\_common.cli\_setup), 39

setUp() (kuha\_common.testing.testcases.KuhaUnitTestCase method), 81

setup\_admin\_user() (in module kuha\_document\_store.db\_setup), 95

setup\_collections() (in module kuha\_document\_store.db\_setup), 96

setup\_database() (in module kuha\_document\_store.db\_setup), 95

setup\_document\_store\_client() (kuha\_common.cli\_setup.Settings method), 39

setup\_document\_store\_query() (kuha\_common.cli\_setup.Settings method), 39

setup\_logging() (kuha\_common.cli\_setup.Settings method), 39

setup\_users() (in module kuha\_document\_store.db\_setup), 95

sleep\_on\_queue (kuha\_common.document\_store.client.JSONStreamClient attribute), 41

SourceFile (class in kuha\_client.kuha\_client), 121

sourcefileparser() (kuha\_client.kuha\_client.BatchProcessor method), 124

split\_value() (kuha\_osmh\_repo\_handler.osmh.records.Payload class method), 116

str\_api\_endpoint() (in module kuha\_common.server), 33

str\_equals() (in module kuha\_common.document\_store.mappings.xmlbase), 76

studies (kuha\_common.document\_store.mappings.ddi.DDI122Record attribute), 77

studies (kuha\_common.document\_store.mappings.ddi.DDI25Record attribute), 78

studies (kuha\_common.document\_store.mappings.ddi.DDI31Record attribute), 78

studies (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase attribute), 76

Study (class in kuha\_common.document\_store.records), 58

study\_area\_countries (kuha\_common.document\_store.records.Study attribute), 60

study\_group\_identifier (kuha\_common.document\_store.records.StudyGroup attribute), 69

study\_group\_names (kuha\_common.document\_store.records.StudyGroup attribute), 69

studies\_groups (kuha\_common.document\_store.mappings.ddi.DDI122Record attribute), 78

studies\_groups (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase attribute), 78

study\_groups (kuha\_common.document\_store.mappings.ddi.DDI31Record attribute), 79

study\_groups (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase attribute), 76

study\_groups (kuha\_common.document\_store.records.Study attribute), 61

study\_number (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase attribute), 75

study\_number (kuha\_common.document\_store.records.Question attribute), 68

study\_number (kuha\_common.document\_store.records.Study attribute), 59

study\_number (kuha\_common.document\_store.records.Variable attribute), 66

study\_number\_identifier (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase attribute), 75

study\_numbers (kuha\_common.document\_store.records.StudyGroup attribute), 69

study\_titles (kuha\_common.document\_store.records.Study attribute), 60

StudyGroup (class in kuha\_common.document\_store.records), 69

StudyGroupRecord (class in kuha\_oai\_pmh\_repo\_handler.osmh.records), 120

StudyRecord (class in kuha\_osmh\_repo\_handler.osmh.records), 118

supported\_metadata\_formats (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments attribute), 108

supported\_verbs (kuha\_oai\_pmh\_repo\_handler.oai.protocol.OAIArguments attribute), 108

SupportedVersionsHandler (class in kuha\_osmh\_repo\_handler.handlers), 114

## U

universes (kuha\_common.document\_store.records.StudyGroup attribute), 60

attribute), 60

UnknownXMLRoot, 71

updates() (kuha\_common.document\_store.field\_types.Element method), 51

updates() (kuha\_common.document\_store.field\_types.Element method), 54

updates() (kuha\_common.document\_store.field\_types.Set method), 49

updates() (kuha\_common.document\_store.field\_types.Value method), 48

updates() (kuha\_common.document\_store.records.Question method), 69

updates() (kuha\_common.document\_store.records.RecordBase method), 58

updates() (kuha\_common.document\_store.records.Study method), 66

updates() (kuha\_common.document\_store.records.StudyGroup method), 70

updates() (kuha\_common.document\_store.records.Variable method), 67

updates\_record() (kuha\_common.document\_store.records.RecordBase method), 58

upsert() (kuha\_client.kuha\_client.BatchProcessor method), 124

upsert\_from\_parser() (kuha\_client.kuha\_client.BatchProcessor method), 125

upsert\_paths() (kuha\_client.kuha\_client.BatchProcessor method), 125

upsert\_run() (in module kuha\_client.kuha\_upsert), 127

upsert\_run() (kuha\_client.kuha\_client.BatchProcessor method), 126

upsert\_study\_groups() (kuha\_client.kuha\_client.BatchProcessor method), 125

**V**

valid\_identifier (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIHeader attribute), 111

valid\_oai\_identifier (kuha\_oai\_pmh\_repo\_handler.oai.records.OAIHeader attribute), 111

validate() (in module kuha\_document\_store.validation), 95

validate() (kuha\_document\_store.validation.RecordValidator method), 93

validate\_query() (kuha\_common.document\_store.query.Query method), 45

validate\_query\_document() (kuha\_common.document\_store.query.Query method), 45

validate\_query\_type() (kuha\_common.document\_store.query.Query method), 45

Value (class in kuha\_common.document\_store.field\_types), 48

value (kuha\_oai\_pmh\_repo\_handler.oai.protocol.ResumptionToken.Attribute attribute), 105

value\_from\_dict() (kuha\_common.document\_store.field\_types.FieldAttribute method), 55

value\_params() (kuha\_common.document\_store.mappings.xmlbase.XMLMapper method), 73

value\_params() (kuha\_common.document\_store.mappings.xmlbase.XMLMapper method), 74

Variable (class in kuha\_common.document\_store.records), 66

variable\_labels (kuha\_common.document\_store.records.Variable attribute), 66

variable\_name (kuha\_common.document\_store.records.Question attribute), 68

variable\_name (kuha\_common.document\_store.records.Variable attribute), 66

VariableRecord (class in kuha\_osmh\_repo\_handler.osmh.records), 119

variables (kuha\_common.document\_store.mappings.ddi.DDI122RecordParser attribute), 78

variables (kuha\_common.document\_store.mappings.ddi.DDI25RecordParser attribute), 78

variables (kuha\_common.document\_store.mappings.ddi.DDI31RecordParser attribute), 79

variables (kuha\_common.document\_store.mappings.xmlbase.XMLParserBase attribute), 76

**W**

WebApplication (class in kuha\_common.server), 34

with\_file\_log() (kuha\_client.kuha\_client.BatchProcessor class method), 124

wrap\_streaming\_callback() (kuha\_common.document\_store.client.JSONStreamClient method), 41

write\_error() (kuha\_common.server.RequestHandler method), 34

**XMLPATTERNS** (in module kuha\_oai\_pmh\_repo\_handler.genshi\_loader), 98

**X**

XMLMapper (class in kuha\_common.document\_store.mappings.xmlbase), 72

XMLParserBase (class in kuha\_common.document\_store.mappings.xmlbase), 74