MOGI Documentation

Tom Lawson, Ralf Weber, Martin Jones, Mark Viant

Nov 21, 2018

Contents

CHAPTER 1

Django applications & projects

The MOGI suite is a collection of Django applications to manage and organise metabolomic studies using Galaxy, ISA and Django frameworks.

The suite consists of the following reusable applications:

- *django-mogi*
- *django-misa*
- *django-galaxy*
- *django-mbrowse*
- *django-gfiles*

The **django-mogi** application inherits all of the detailed applications below to create ready to go framework for organising metabolomics studies with Galaxy and ISA. The full suite of applications can be used in either a docker container or directly as Django package run. See *User docs* for details about how to get up and running with the full suite of applications.

Additionally, each application has been developed to be reusable. This allows the more general functionality to be used in other Django projects and applications. See the individual package documentation for more details.

The figure below shows how the applications fit within Django project website

CHAPTER 2

What the MOGI suite can do

- Organise metabolomics studies following ISA framework
- Interface with Galaxy (initiate workflows, manage history, inport/export data)
- Search metabolomics spectra across multiple projects (spectral matching, mass search)
- Summarise compound annotations (e.g. from different annotation approaches like MetFrag, CSI:FingerID, spectral matching)

CHAPTER 3

User Documentation

3.1 User docs

3.1.1 Installation

Running the example project with Docker

The quickest way to try out the MOGI suite of Django applications is to use the docker image. This should be run with alongside the docker of a modified Worfklow4Metabolomics Galaxy docker that contains all the tools and workflows that are required. See below:

1. Start the Galaxy instance through docker: `docker run -p 8080:80 -p 8022:22 -p 8021:21 tomnl/w4m-docker-mogi` For full integration with mogi the w4m-docker-mogi should be used (but is still in development). However any Galaxy docker could be used e.g. workflow4metabolomics/galaxy-workflow4metabolomics
2. Start the Django-Celery-rabbitmq-redis services through docker (currently this has to be done with docker-compose)
    - `git clone https://github.com/computational-metabolomics/django-mogisite`
    - `cd django-mogisite`
    - `docker-compose up`
3. By default a user on the django-mogisite website called *admin* with password *admin* is already linked to the Galaxy user *admin* with password *admin*
4. An example ISA projects should also be pre-loaded

Running the example project locally

The following project site can also be used as a standard Django project. See below:

1. Optionally create virtual environment with venv or conda (e.g. `conda create -n mogi python=3.6.5`)
2. The functionality for the MOGI applications has been tested using MySQL and SQLite databases, however MySQL is the preferred database backend to use. Please ensure either mysqlclient is installed (e.g. `conda install mysqlcient`) if using the MySQL backend.
3. `pip install django-mogi`
4. `git clone https://github.com/computational-metabolomics/django-mogisite`
5. `cd django-mogisite`
6. `python manage.py migrate` to create the mogi models.
7. Start the development server and visit http://127.0.0.1:8000/
8. Register http://127.0.0.1:8000/register/ and login http://127.0.0.1:8000/login/

If using outside of the docker, to get the full functionality you will need install and configure the messaging broker **rabbitmq** and the result backend **redis** for the distributed task queue system **Celery**.

1. Celery should be installed alongside the mogi python packages but can also just be installed via via pip or conda e.g.:bash:*pip install celery* or `conda install celery`
2. Install erlang (required for rabbitmq) `sudo apt-get install -y erlang`
3. Install rabbitmq `sudo apt-get install rabbitmq-server`
4. Add user (needs to match to settings.py file) `sudo rabbitmqctl add_user admin mypass`
5. Add host (needs to match to settings.py file) `sudo rabbitmqctl add_vhost myvhost`
6. Add tags (optional) `sudo rabbitmqctl set_user_tags myuser mytag`
7. Give permission to user `sudo rabbitmqctl set_permissions -p myvhost admin ".*" ".*" ".*"`
8. Restart rabbitmq `sudo service rabbitmq-server restart`
9. Install redis (easiest way is through conda) `conda install redis-py`
10. Start redis (if not already started) `redis-server`
11. Finally, start celery from the django-mogisite directory `celery -A mogi_site.celery worker -l DEBUG -E`

Different backends can be used with Celery and should still be compatible with the MOGI applications. See the celery documentation for more details.

Incorporating the packages into existing projects

Alternatively, individual applications can be installed into existing Django projects as well, see the **quick start** section of the documentation for each Django application.

3.1.2 Manage Metabolomic ISA projects

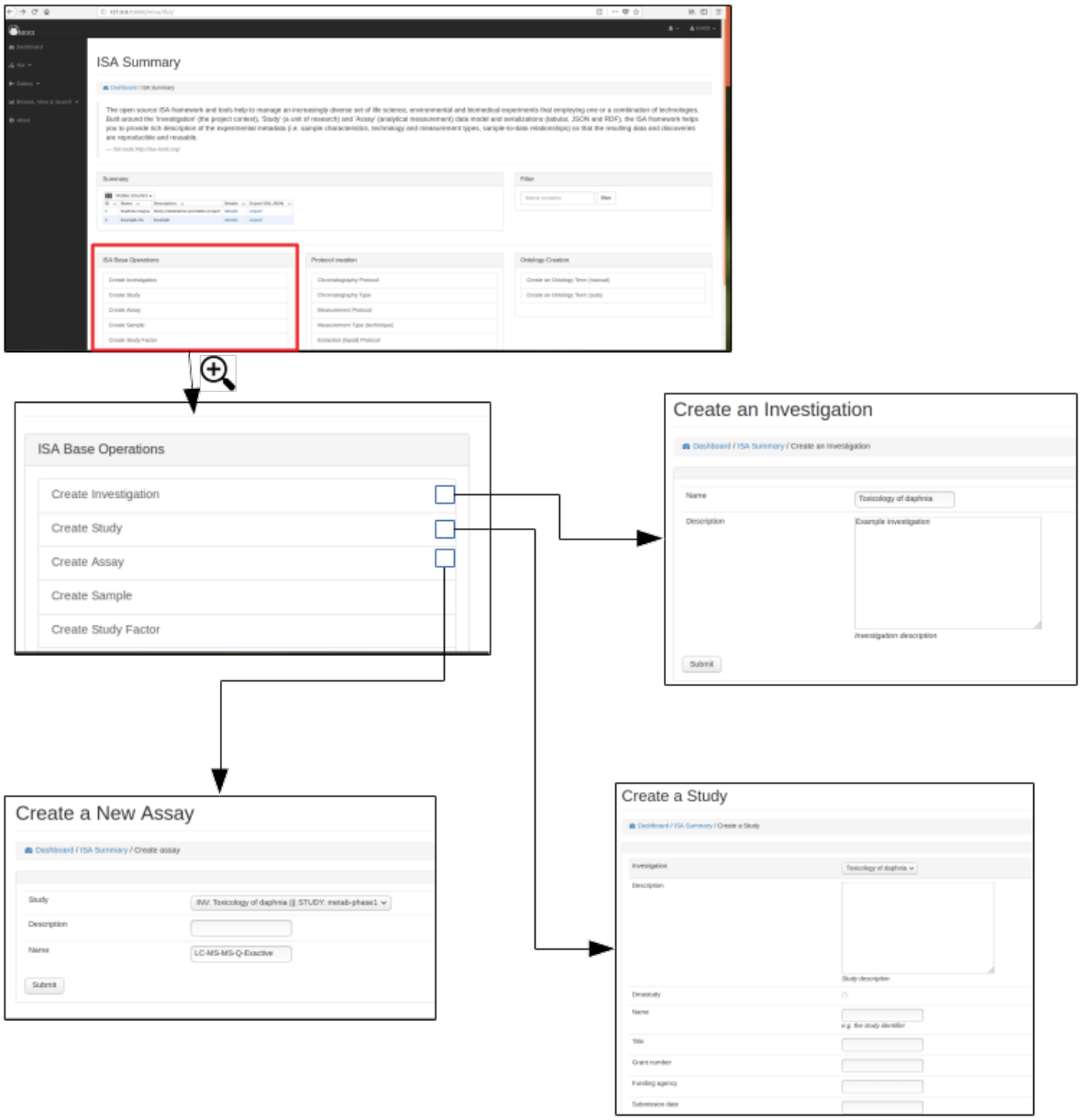
The functionality is summarised along with any current ISA projects at http://127.0.0.1:8000/misa/

Create ISA backbone

To initiate an ISA project the backbone of the ISA project has to be created. e.g. the investigation, study and assay details.

First an investigation has to be created, this only requires the name of the overall ISA project and a short description. One or more studies can then be created and assigned to an investigation. Further details are added at this stage including any study descriptors.

Multiple assays can then be added to each study.

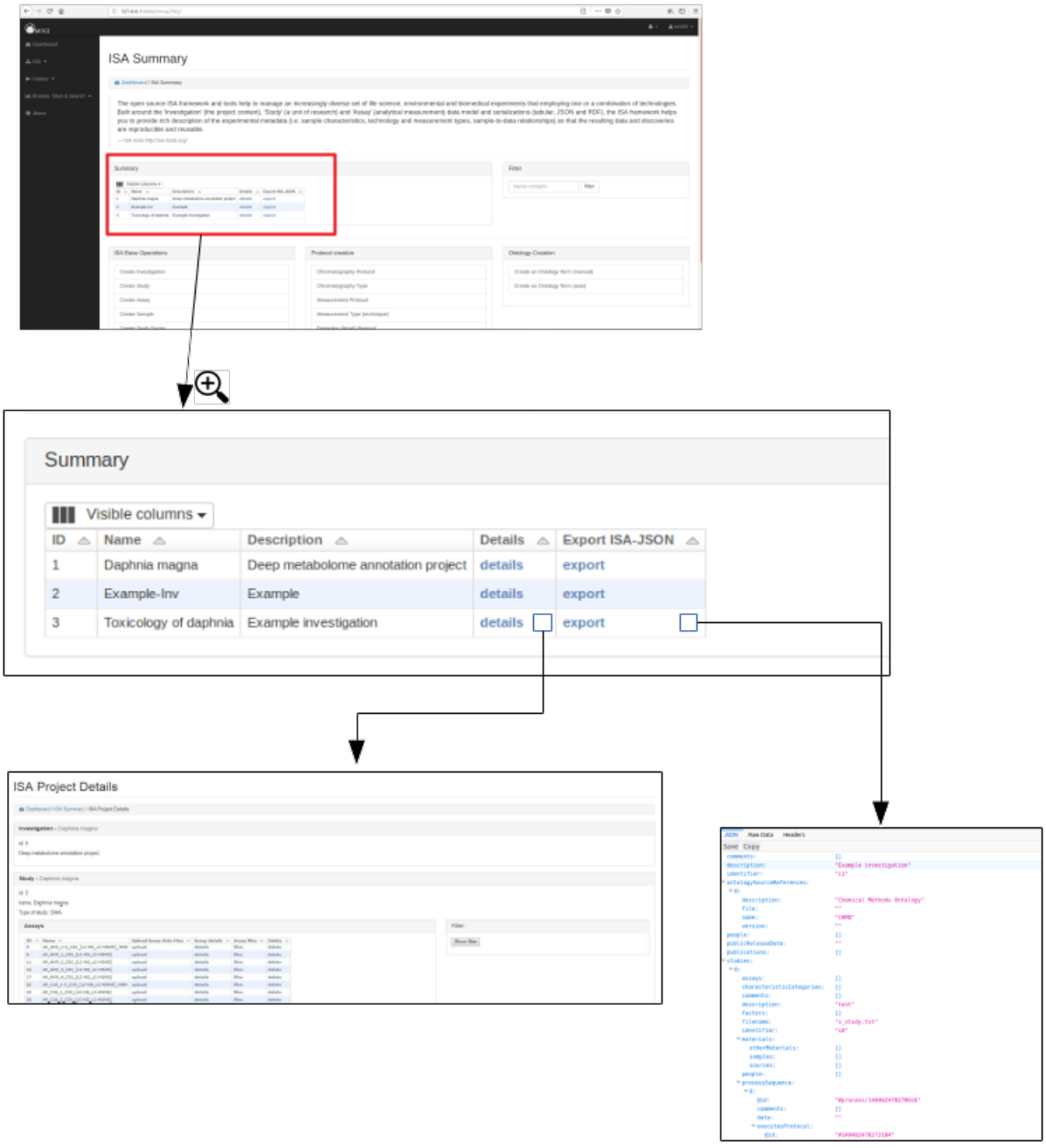


Browse and export ISA projects

Once an investigation has been created. The ISA project will be visible in the summary table. Further details can then be shown by clicking on the details section.

Preliminary work has been made so that the ISA project can be exported as an ISA-JSON file.

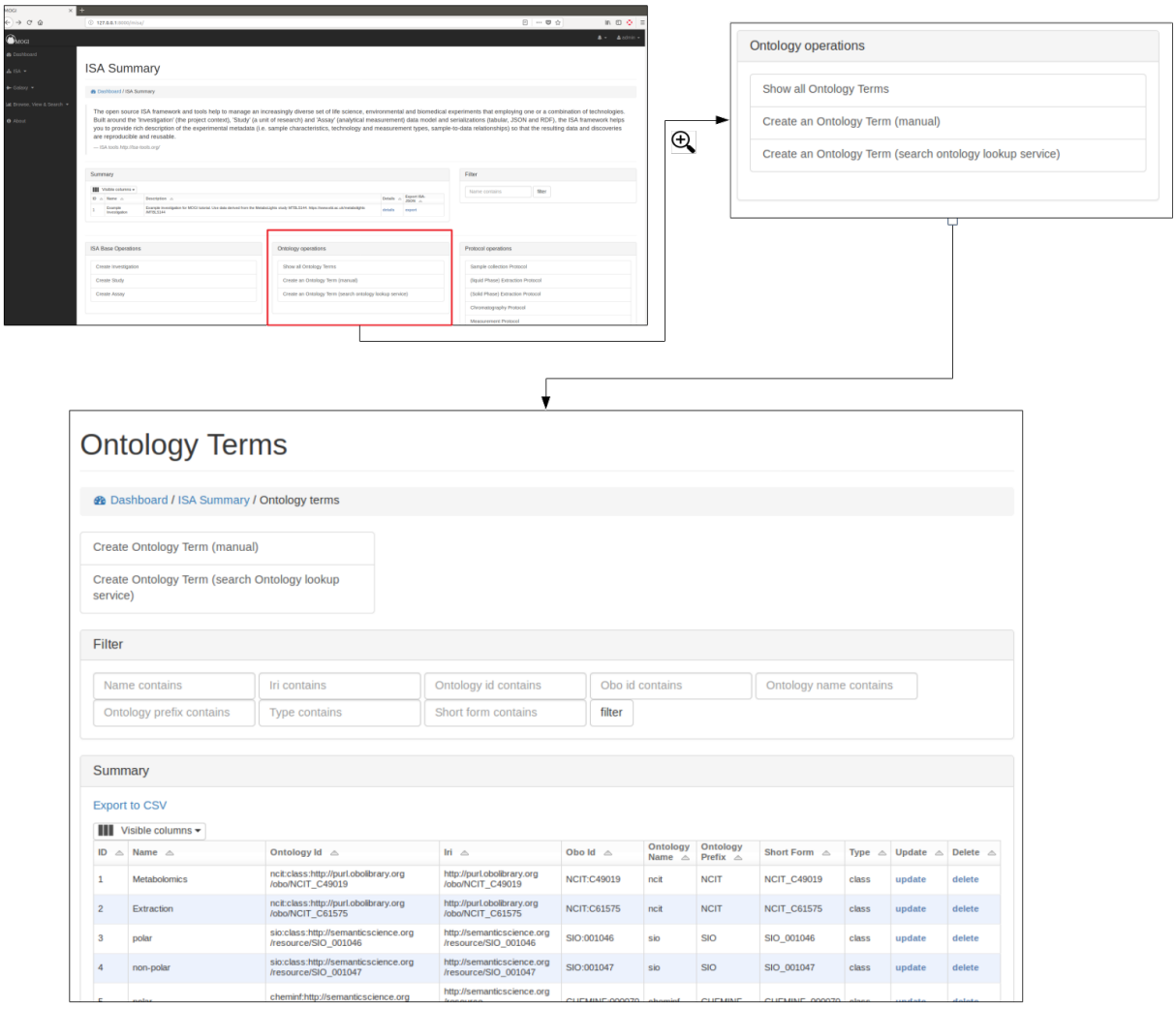
Currently only admin users can delete the projects via the standard Django admin interface (e.g. at 127.0.0.1:8000/admin).



Ontologies

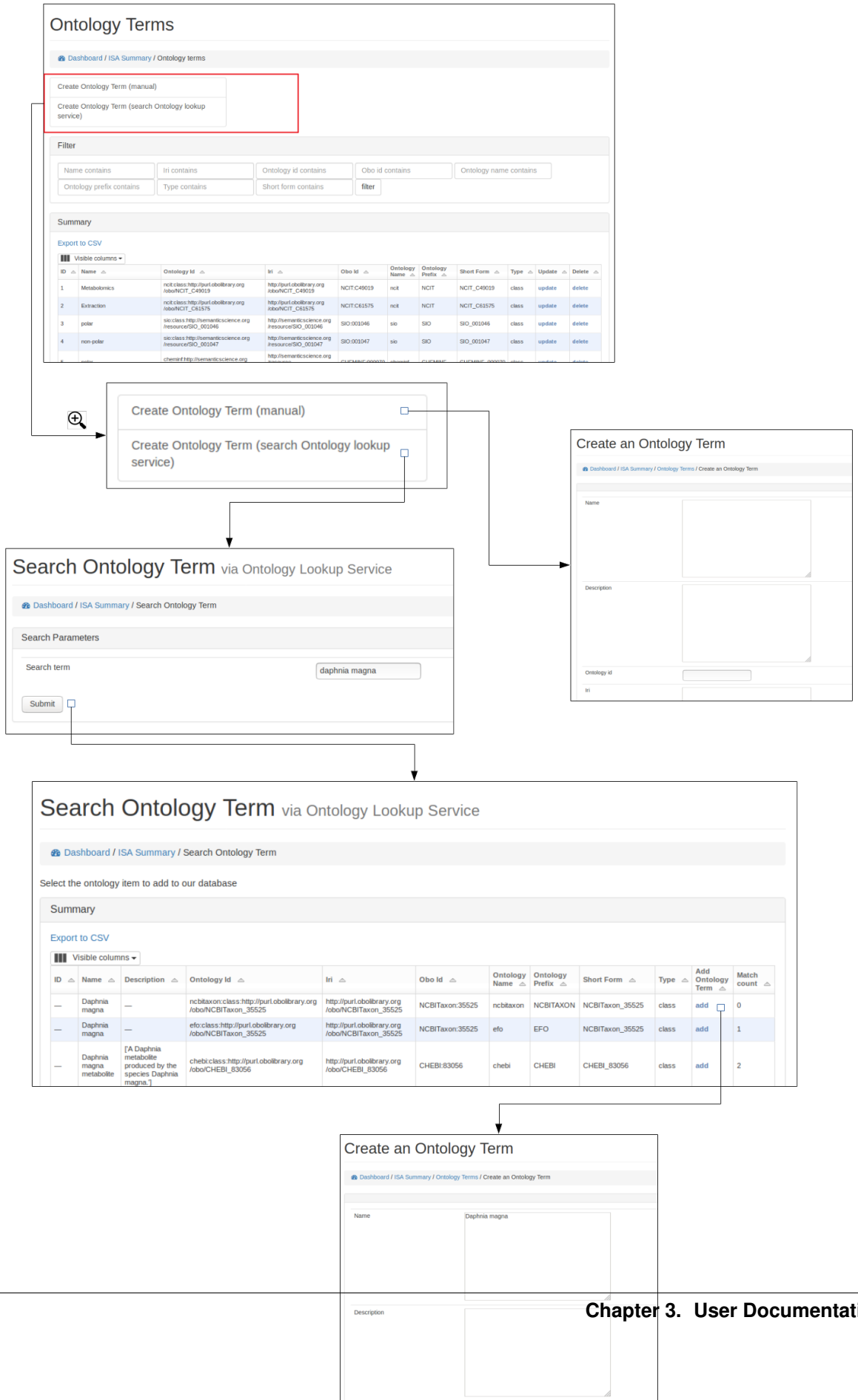
Browse, update and delete ontology terms

Where possible ontological terms are used to describe components of the ISA project. The full list of ontology terms used for a project can be searched, updated and viewed (see below)



Add new terms

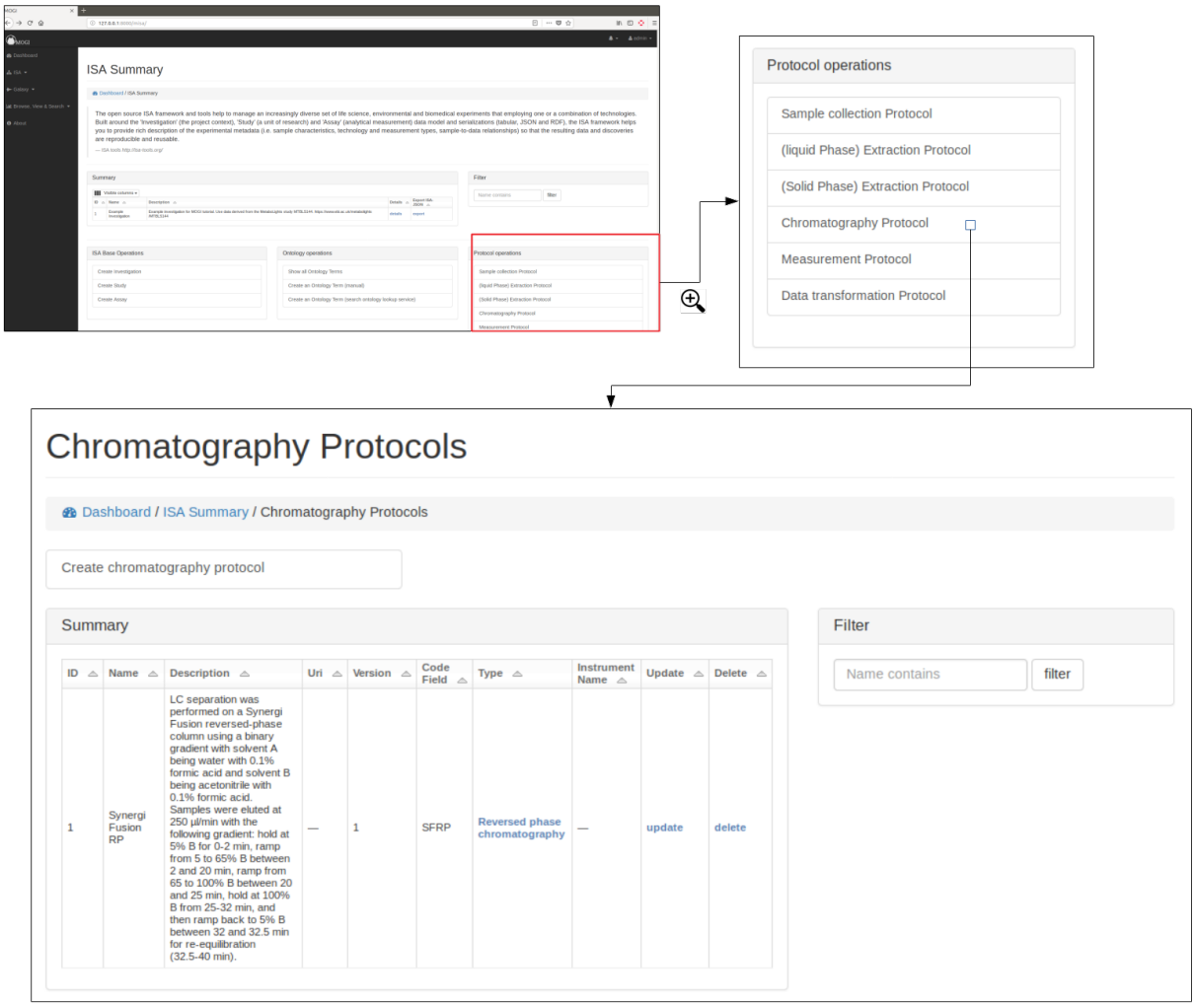
New ontological terms can be added manually or can be searched using the EBI ontology Lookup service and the best term can be added into the local database of ontology terms.


Ontology Terms
Dashboard / ISA Summary / Ontology terms
Create Ontology Term (manual)
Create Ontology Term (search Ontology lookup service)
Filter
Summary
Export to CSV
Create Ontology Term (manual)
Create Ontology Term (search Ontology lookup service)
Create an Ontology Term
Search Ontology Term via Ontology Lookup Service
Dashboard / ISA Summary / Search Ontology Term
Search Parameters
Search term
daphnia magna
Submit
Search Ontology Term via Ontology Lookup Service
Dashboard / ISA Summary / Search Ontology Term
Select the ontology item to add to our database
Summary
Export to CSV
Visible columns
Create an Ontology Term
Dashboard / ISA Summary / Ontology Terms / Create an Ontology Term
Name
Daphnia magna
Description

Protocols

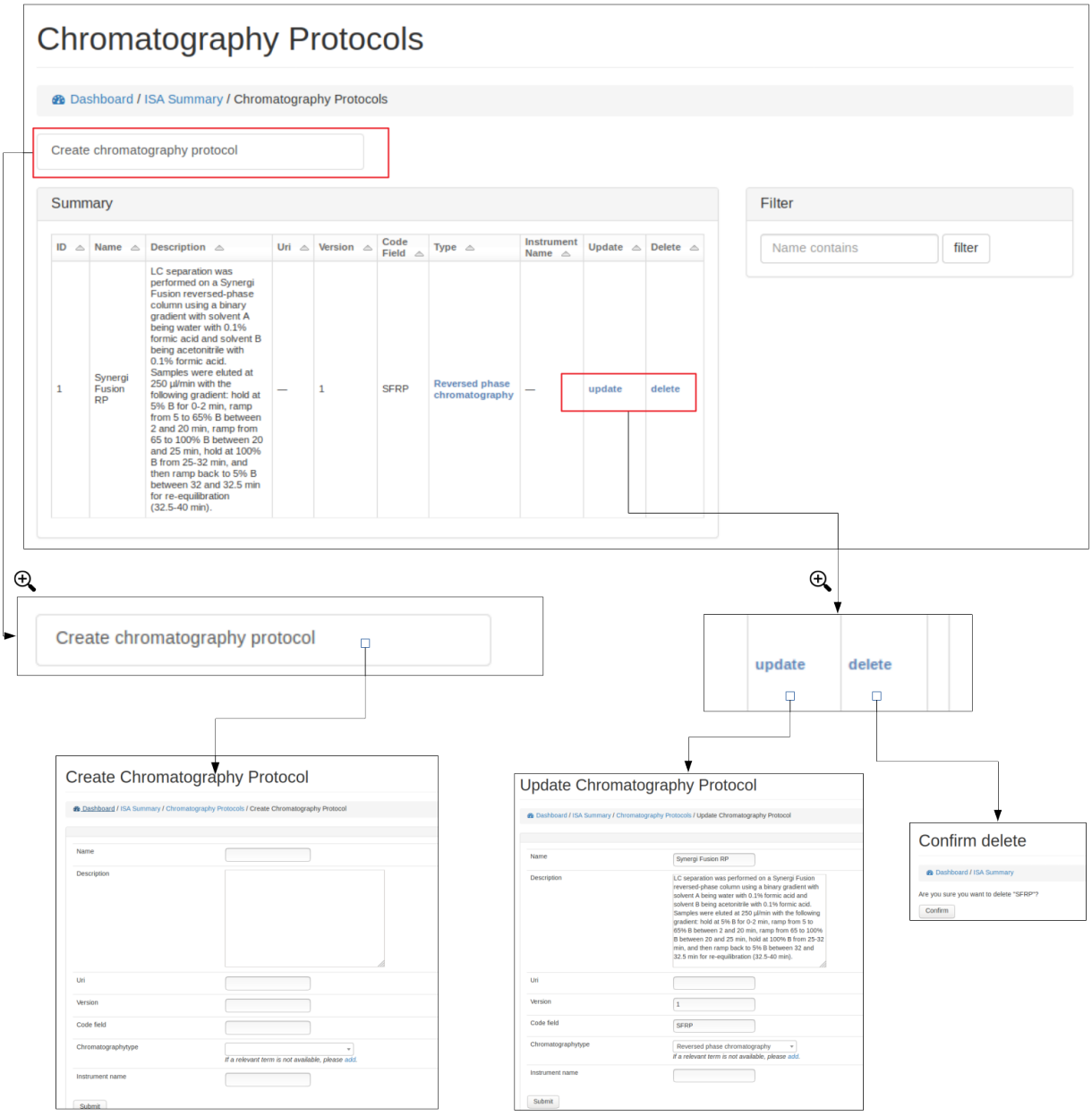
Browse protocols

Protocol models have been created based on the MetaboLights description for metabolomics projects and consists of sample collection, (liquid phase)-extraction, (solid phase)-extraction, chromatography, measurement (i.e. mass spectrometry type), data transformation and metabolite identification.



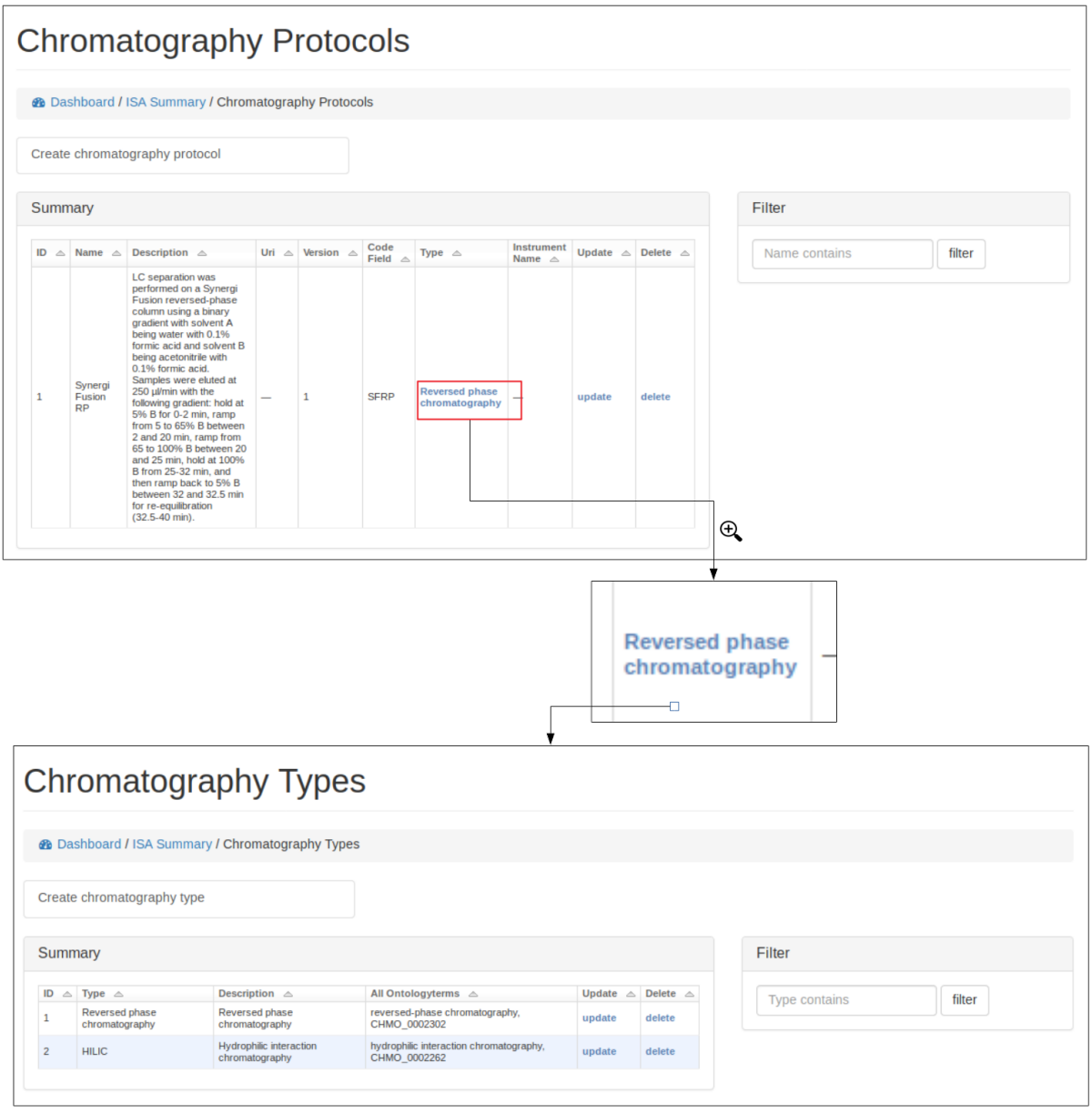
Create, edit and delete protocols

A user can view, edit and delete protocols, see below for example using the Chromatography protocol. An important field for a protocol is the **code_field** this is a unique small code for a protocol that can be used later when linking uploaded data files to a specific protocol.


Chromatography Protocols
Dashboard / ISA Summary / Chromatography Protocols
Create chromatography protocol
Summary
Filter
Name contains
filter
ID
Name
Description
Uri
Version
Code Field
Type
Instrument Name
Update
Delete
1
Synergi Fusion RP
LC separation was performed on a Synergi Fusion reversed-phase column using a binary gradient with solvent A being water with 0.1% formic acid and solvent B being acetonitrile with 0.1% formic acid. Samples were eluted at 250 µl/min with the following gradient: hold at 5% B for 0-2 min, ramp from 5 to 65% B between 2 and 20 min, ramp from 65 to 100% B between 20 and 25 min, hold at 100% B from 25-32 min, and then ramp back to 5% B between 32 and 32.5 min for re-equilibration (32.5-40 min).
—
1
SFRP
Reversed phase chromatography
—
update
delete
Create chromatography protocol
update
delete
Create Chromatography Protocol
Update Chromatography Protocol
Confirm delete
Are you sure you want to delete "SFRP"?
Confirm

Protocol types

Where possible, protocols can be grouped together by different protocol types. For example a Chromatography protocol can be associated with either the Chromatography type 'HILIC' or 'reverse phase chromatography' (see below). A user can add and edit as many different protocol types as necessary.



Study samples

Browse, create, update and delete study samples

Study samples can be browsed, created and edited.


Sample operations
Study Samples
Study Factors
Organisms
Organism parts
Study Samples
Dashboard / ISA Summary / Create Study Samples
Create Study Sample
Batch Create Study Samples
Summary
Filter
Sample name contains
filter
Update
Delete
update
delete
Create a Study Sample
Dashboard / ISA Summary / Study Samples / Create a Study Sample
Study
INV: Example Investigation ||| STUDY...
Sample name
daph-t1
The sample name has to unique for each study
Studyfactor
Time: 1
If factor not available then please add.
Organism
Daphnia magna
If factor not available then please add.
Organism part
Whole Organism
If organism part not available then please add.
Sampletype
ANIMAL
Submit
Update a Study Sample
Dashboard / ISA Summary / Study Samples / Update a Study Sample
Study
INV: Example Investigation ||| STUDY: MTBLS144-TEST-CASE
Sample name
Mtab_FT_012611_13
The sample name has to unique for each study
Studyfactor
Control: with Thalassiosira
Time: 1
If factor not available then please add.
Organism
Thalassiosira pseudonana
If factor not available then please add.
Organism part
exometabolome
If organism part not available then please add.
Sampletype
ANIMAL
Submit
Confirm delete
Dashboard / ISA Summary
Are you sure you want to delete "Mtab_FT_012611_13"?
Confirm

Study factors, organisms and organism parts

Each study factor can be associated with an organism and organism part as well as multiple study factors.


Study Samples
Dashboard / ISA Summary / Create Study Samples
Create Study Sample
Batch Create Study Samples
Summary
Filter
Sample name contains
filter
ID
Investigation
Study
Sample Name
Study Factors
Organism
Organism Part
Update
Delete
1 Example Investigation MTBLS144-TEST-CASE Mtab_FT_012611_13 • Control: with Thalassiosira • Time: 1 Thalassiosira pseudonana exometabolome update delete
2 Example Investigation MTBLS144-TEST-CASE Mtab_FT_012611_14 • Time: 1 • Control: cell-free control Thalassiosira pseudonana exometabolome update delete
3 Example Investigation MTBLS144-TEST-CASE Mtab_FT_012611_15 • Control: with Thalassiosira • Time: 1 Thalassiosira pseudonana exometabolome update delete
4 Example Investigation MTBLS144-TEST-CASE Mtab_FT_012611_16 • Control: with Thalassiosira • Time: 1 Thalassiosira pseudonana endometabolome update delete
5 Example Investigation MTBLS144-TEST-CASE Mtab_FT_012611_17 • Time: 1 • Control: cell-free control Thalassiosira pseudonana endometabolome update delete
Study Factors
Organism
Organism Part
• Control: with Thalassiosira • Time: 1 Thalassiosira pseudonana exometabolome
• Time: 1 • Control: cell-free control Thalassiosira pseudonana exometabolome
Study Factors
Dashboard / ISA Summary / Study Factors
Create Study Factor
Summary
Organism part list
Dashboard / ISA Summary / Organism parts
Create organism part
Summary
ID Ontologyterm Name Update Delete
1 Whole Organism Whole Organism update delete
2 exometabolome exometabolome update delete
3 endometabolome endometabolome update delete
Organisms
Dashboard / ISA Summary / Organisms
Create organism
Summary
ID Ontologyterm Name Update Delete
1 Thalassiosira pseudonana Thalassiosira pseudonana update delete
2 Daphnia magna Daphnia magna update delete
3 Daphnia pulex Daphnia pulex update delete
4 Homo sapiens Homo sapiens update delete
5 Mus musculus Mus musculus update delete

Add study samples (batch)

For studies with many samples it is recommended to batch upload the study samples. Relevant ontology terms will be automatically uploaded based on the sample list provided. The columns of the sample list should consist of the following column titles:

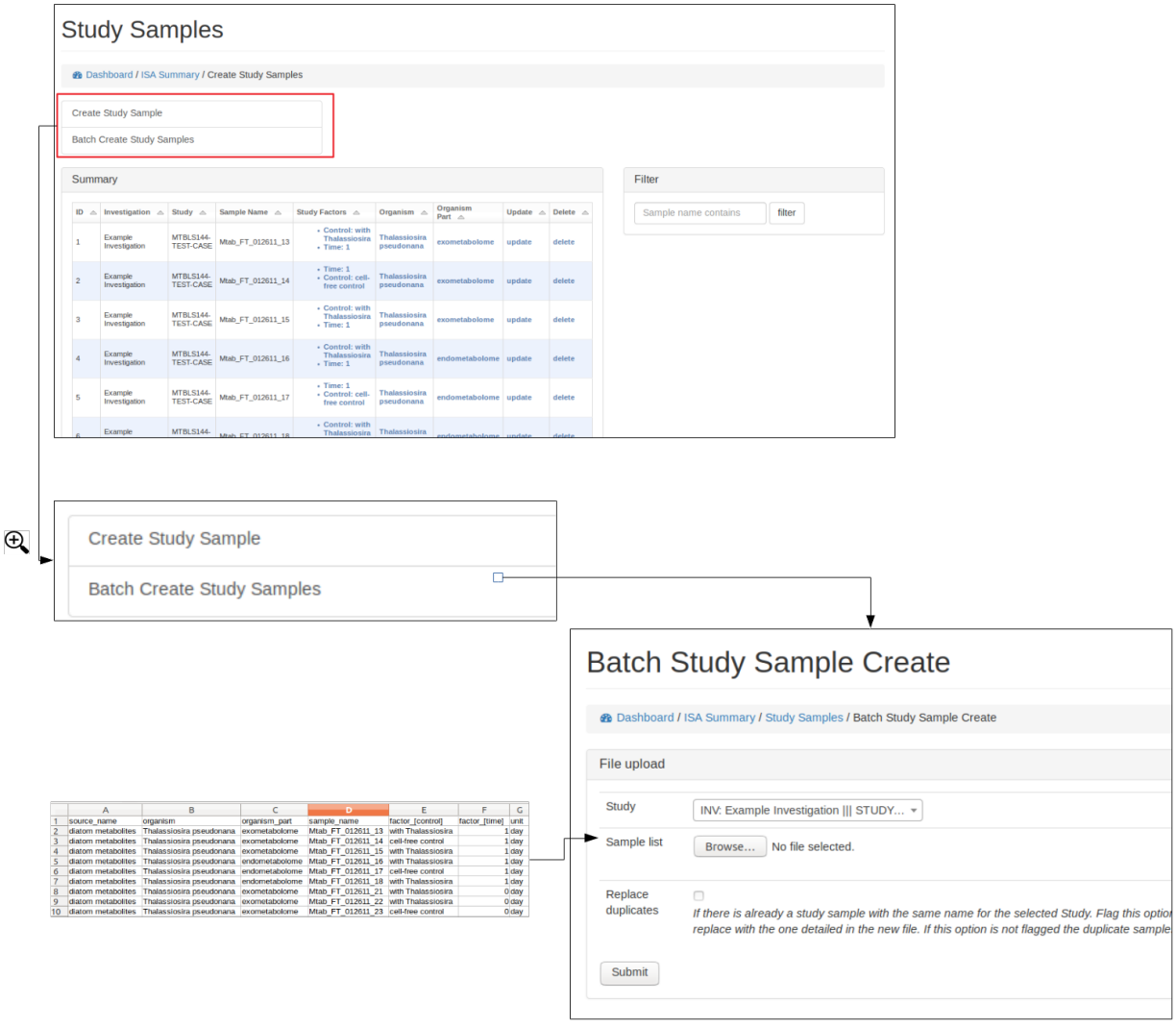
- **source_name**: Source of the sample (e.g. Acme laboratories).
- **sample_name**: Study sample name (needs to be unique for the study).
- **organism**: Organism name (e.g. *Daphnia magna*). Best matching ontological term will be automatically searched.
- **organism_part**: The part of the organism for the sample (e.g. heart, lung). Best matching ontological term will be automatically searched.
- **factor_[<–name of study factor–>]**: A study factor for the sample. The column name will change based on the factor type e.g. factor_[time]. Relevant ontological terms will be automatically searched.
- **factor_[<–name of study factor–>]_unit**: Optionally a unit for the study factor can be used. The column should match the study factor type e.g factor_[time]_unit. Relevant ontological terms will be automatically searched.

Multiple factor columns can be used e.g.

source_name	sample_name	organism	organism_part	factor_[time]	factor_[time]_unit	factor_[control]
Acme laboratories	Mtab_FT_012611_T3	Thalassiosira pseudonana	exometabolome	1	day	with Thalassiosira
Acme laboratories	Mtab_FT_012611_T4	Thalassiosira pseudonana	exometabolome	2	day	cell-free control

Alternatively, if an ISA tab file has already been created. The study.txt file can be uploaded and the relevant details will be extracted.

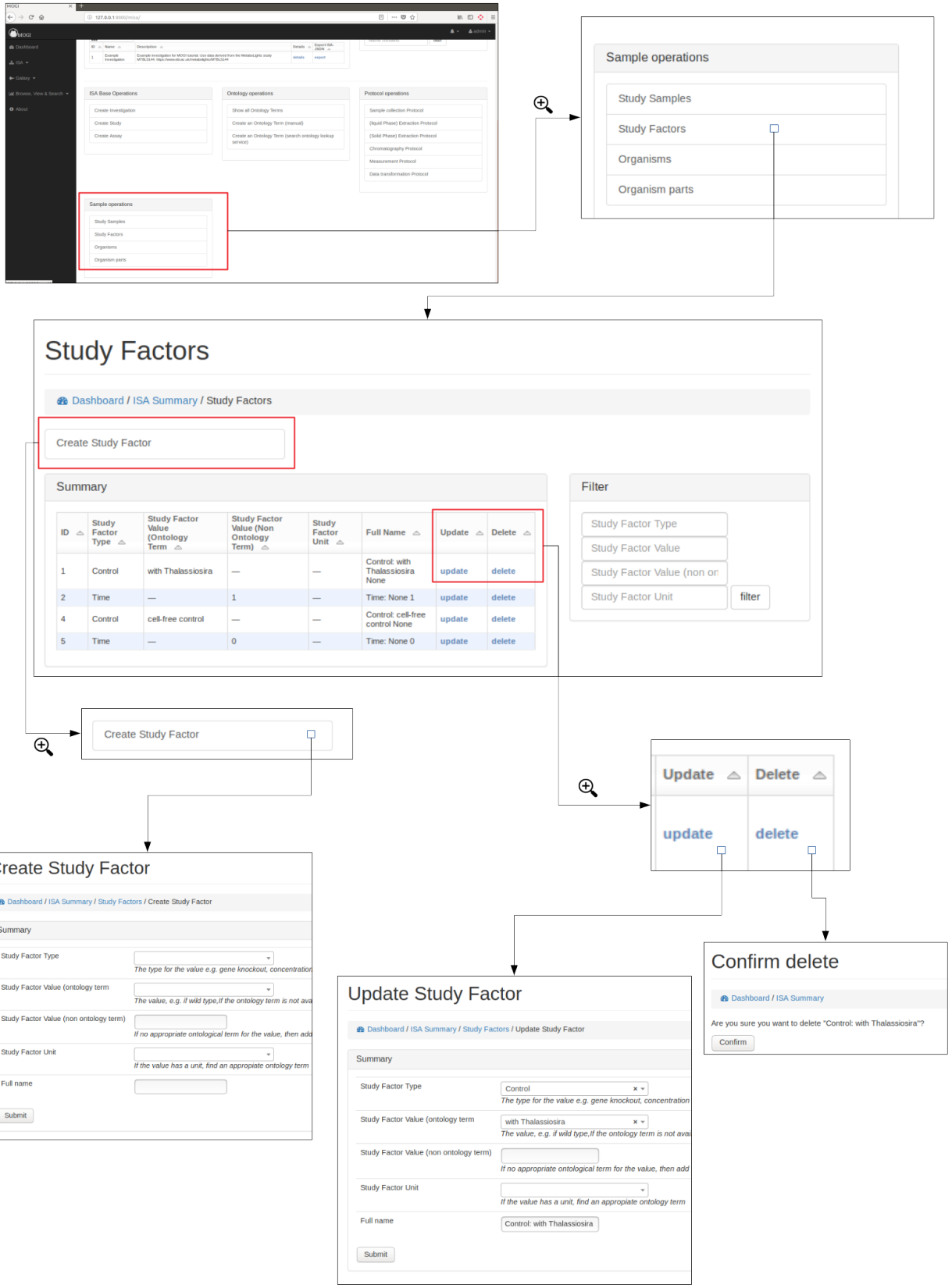
See below for how to upload study samples as a batch:



Study Factors

Browse, create, update and delete study factors

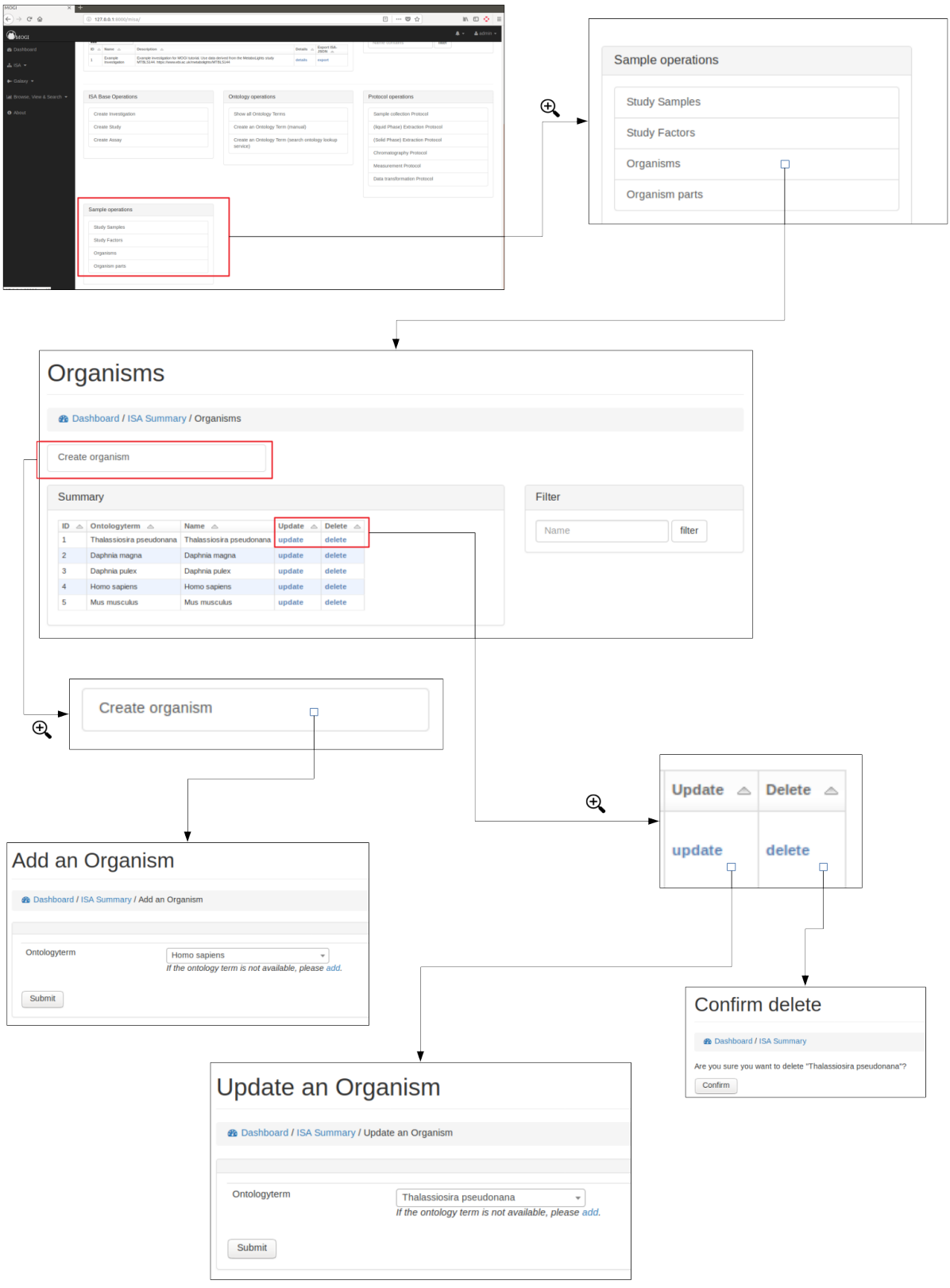
Study factors can be either added manually (see below) or added with study samples in a batch process (see section **Add study samples (batch)** above)


Sample operations
Study Samples
Study Factors
Organisms
Organism parts
Study Factors
Dashboard / ISA Summary / Study Factors
Create Study Factor
Summary
ID
Study Factor Type
Study Factor Value (Ontology Term)
Study Factor Value (Non Ontology Term)
Study Factor Unit
Full Name
Update
Delete
1 Control with Thalassiosira — — Control: with Thalassiosira None update delete
2 Time — 1 — Time: None 1 update delete
4 Control cell-free control — — Control: cell-free control None update delete
5 Time — 0 — Time: None 0 update delete
Filter
Study Factor Type
Study Factor Value
Study Factor Value (non or
Study Factor Unit
filter
Create Study Factor
Update
Delete
update
delete
Create Study Factor
Dashboard / ISA Summary / Study Factors / Create Study Factor
Summary
Study Factor Type
Study Factor Value (ontology term
Study Factor Value (non ontology term)
Study Factor Unit
Full name
Submit
Update Study Factor
Dashboard / ISA Summary / Study Factors / Update Study Factor
Summary
Study Factor Type
Control
Study Factor Value (ontology term
with Thalassiosira
Study Factor Value (non ontology term)
Study Factor Unit
Full name
Control: with Thalassiosira
Submit
Confirm delete
Dashboard / ISA Summary
Are you sure you want to delete "Control: with Thalassiosira"?
Confirm

Organisms

Browse, create, update and delete organisms

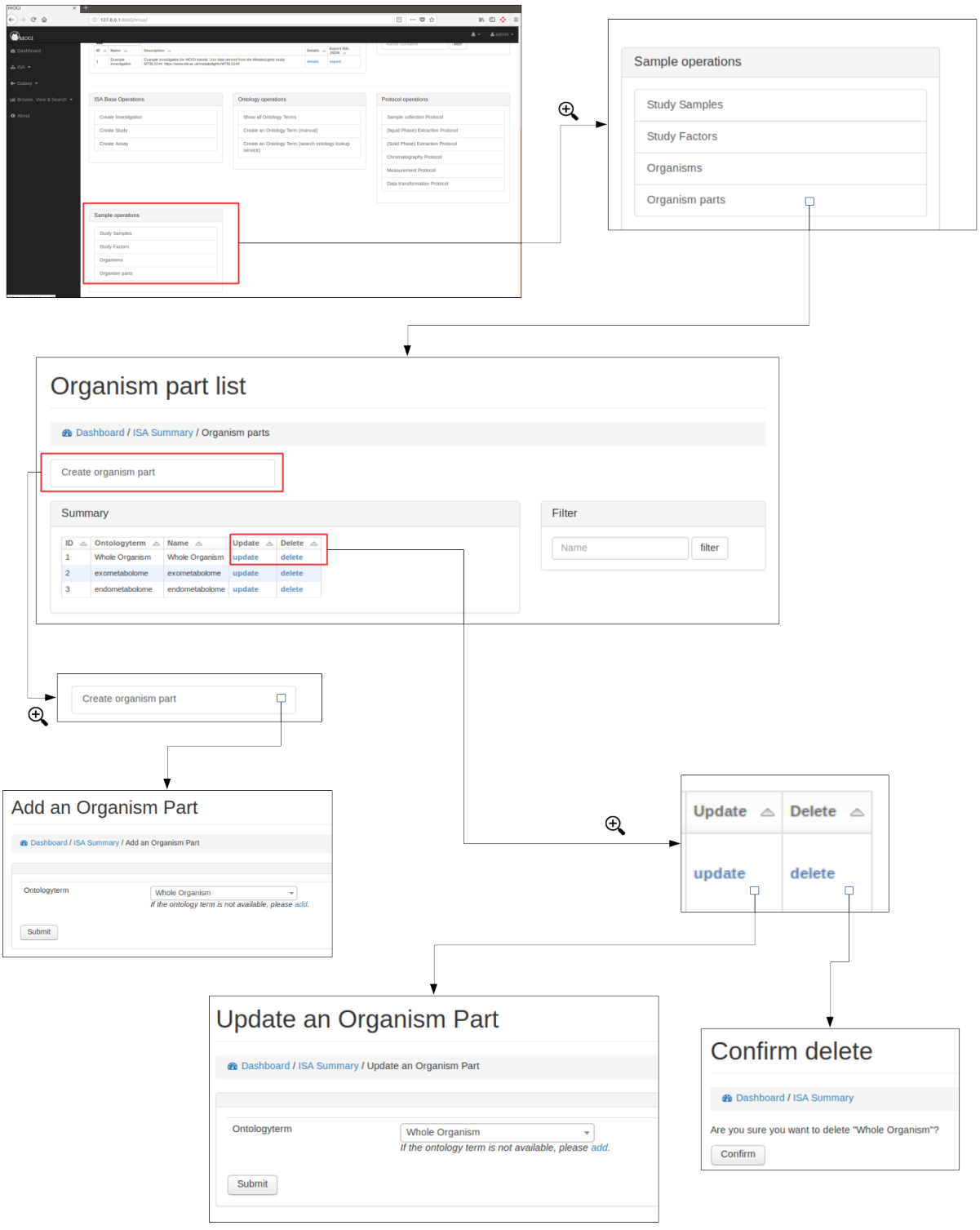
Organisms can be either added manually (see below) or added with study samples in a batch process (see section **Add study samples (batch)** above)


Sample operations
Study Samples
Study Factors
Organisms
Organism parts
Organisms
Dashboard / ISA Summary / Organisms
Create organism
Summary
Filter
Name
filter
ID
Ontologyterm
Name
Update
Delete
1 Thalassiosira pseudonana Thalassiosira pseudonana update delete
2 Daphnia magna Daphnia magna update delete
3 Daphnia pulex Daphnia pulex update delete
4 Homo sapiens Homo sapiens update delete
5 Mus musculus Mus musculus update delete
Create organism
Update
Delete
update
delete
Add an Organism
Dashboard / ISA Summary / Add an Organism
Ontologyterm
Homo sapiens
If the ontology term is not available, please add.
Submit
Confirm delete
Dashboard / ISA Summary
Are you sure you want to delete "Thalassiosira pseudonana"?
Confirm
Update an Organism
Dashboard / ISA Summary / Update an Organism
Ontologyterm
Thalassiosira pseudonana
If the ontology term is not available, please add.
Submit

Organism parts

Browse, create, update and delete organism parts

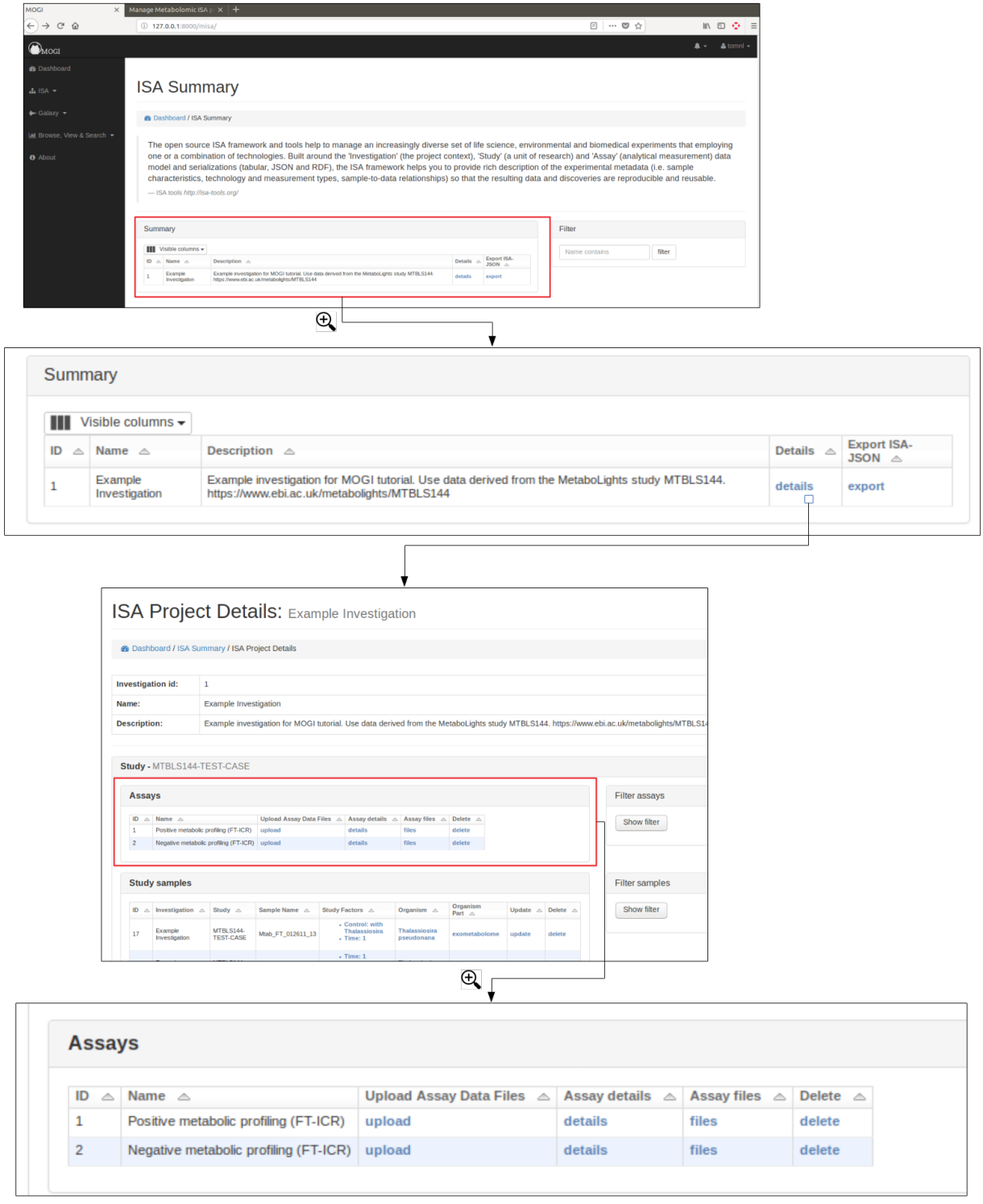
Organisms parts can be either added manually (see below) or added with study samples in a batch process (see section **Add study samples (batch)** above)


Sample operations
Study Samples
Study Factors
Organisms
Organism parts
Organism part list
Dashboard / ISA Summary / Organism parts
Create organism part
Summary
Filter
Name
filter
ID
Ontologyterm
Name
Update
Delete
1
Whole Organism
Whole Organism
update
delete
2
exometabolome
exometabolome
update
delete
3
endometabolome
endometabolome
update
delete
Create organism part
Add an Organism Part
Dashboard / ISA Summary / Add an Organism Part
Ontologyterm
Whole Organism
If the ontology term is not available, please add.
Submit
Update
Delete
update
delete
Update an Organism Part
Dashboard / ISA Summary / Update an Organism Part
Ontologyterm
Whole Organism
If the ontology term is not available, please add.
Submit
Confirm delete
Dashboard / ISA Summary
Are you sure you want to delete "Whole Organism"?
Confirm

Assay details and data files

Select assay

To upload data files (e.g. mzML) to an assay, first an assay needs to have been created for the appropriate study (see section **Create ISA backbone**. The assay should then be viewable in the Investigation details section. See below for example:


ISA Summary
Summary
Visible columns
ID
Name
Description
Details
Export ISA-JSON
1
Example Investigation
Example investigation for MOGI tutorial. Use data derived from the MetaboLights study MTBLS144. https://www.ebi.ac.uk/metabolights/MTBLS144
details
export
ISA Project Details: Example Investigation
Investigation id: 1
Name: Example Investigation
Description: Example investigation for MOGI tutorial. Use data derived from the MetaboLights study MTBLS144.
Study - MTBLS144-TEST-CASE
Assays
Filter assays
Show filter
Study samples
Filter samples
Show filter
Assays
ID
Name
Upload Assay Data Files
Assay details
Assay files
Delete
1
Positive metabolic profiling (FT-ICR)
upload
details
files
delete
2
Negative metabolic profiling (FT-ICR)
upload
details
files
delete

Upload data files and mapping

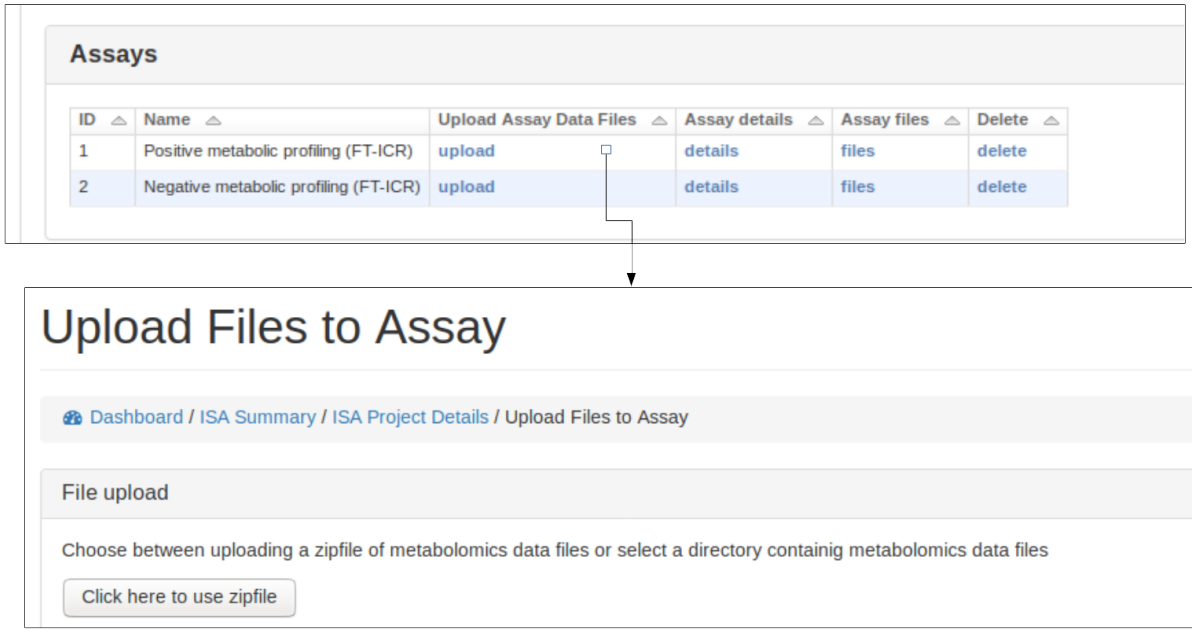
Data files (e.g. mzML, raw) should be uploaded to the appriopiate assay for a study. This can either be done by uploading a zip file of the data files or providing a path to the data files if the file system is available.

A mapping file should be provided that links each file to sample and protocols. Each protocol has a 'code_field' to be used to reference the correct protocol that was performed. See below for accepted column details

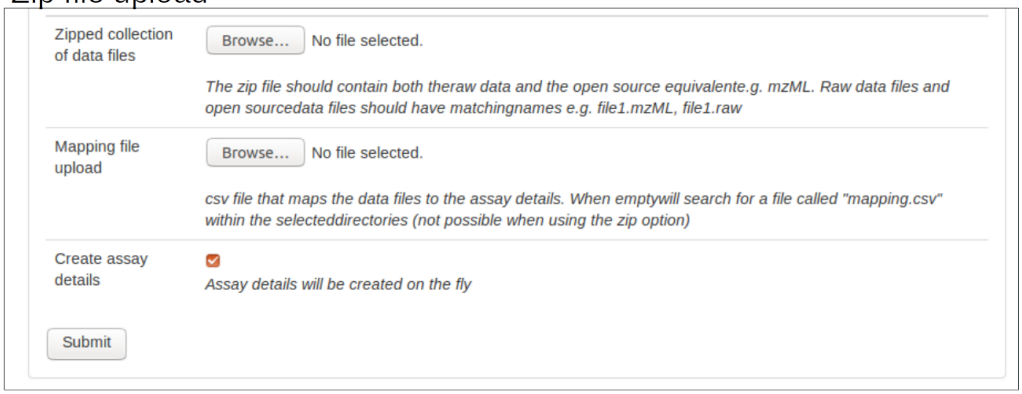
- **filename**: Name of the data file (e.g. sample1_lcms.mzML)
- **sample**: The sample name (should correspond to the sample name provided at earlier stages)
- **sample_collection**: The code_field for the sample collection protocol performed
- **extraction**: The code_field for the liquid phase extraction protocol performed
- **spe**: The code_field for the solid phase extraction protocol performed
- **spe_frac**: If SPE fractionation was performed resulting in multiple fractions, this column is to indicate the fraction number
- **chromatography**: The code_field for the chromatography protocol performed
- **chromatography_frac**: If chromatography fractionation was performed resulting in multiple fractions, this column is to indicate the fraction number
- **measurement**: The code_field for the measurement protocol performed
- **polarity**: The code_field for the polarity setting of the instrument
- **technical_replicate**: Numerical value indicating technical replicate
- **fileformat**: suffix of the file being uploaded (currently supports mzML and raw)

If a protocol was not used (e.g. if Chromatography was not performed) the column in the mapping file should be NA. See below for example format:

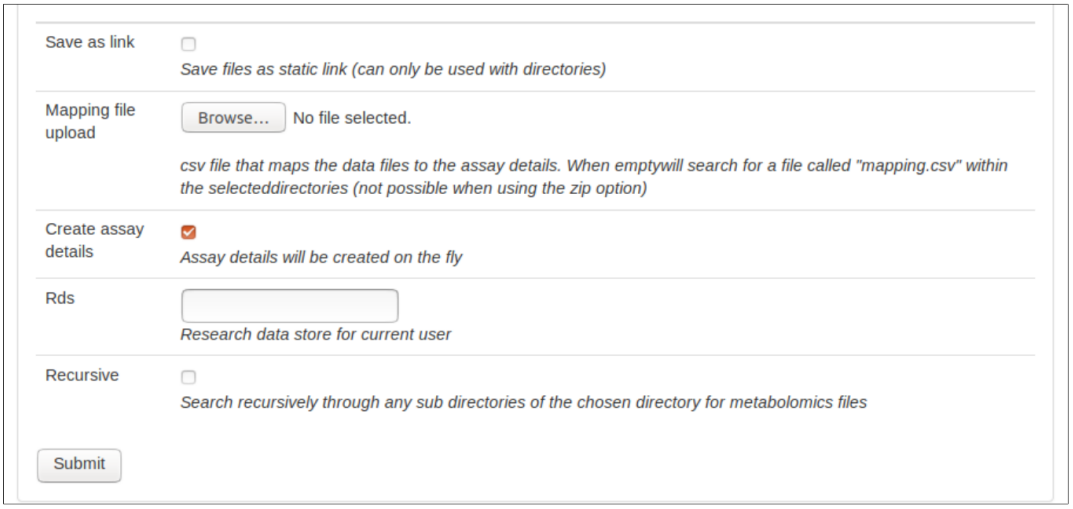
filename	sample	sample_collection	extraction	spe	spe_frac	chromatography	chromatography_frac	measurement	polarity	technical_replicate	fileformat
Mtab_FT_0126	Mtab_FT_0126	DOM_11_13 ATOM	DOM	DOM	NA	SFRP	NA	FT-ICR	POS-I-TIVE	1	mzml
Mtab_FT_0126	Mtab_FT_0126	DOM_11_13 ATOM	DOM	DOM	NA	SFRP	NA	FT-ICR	POS-I-TIVE	2	mzml



Zip file upload



Directory file upload



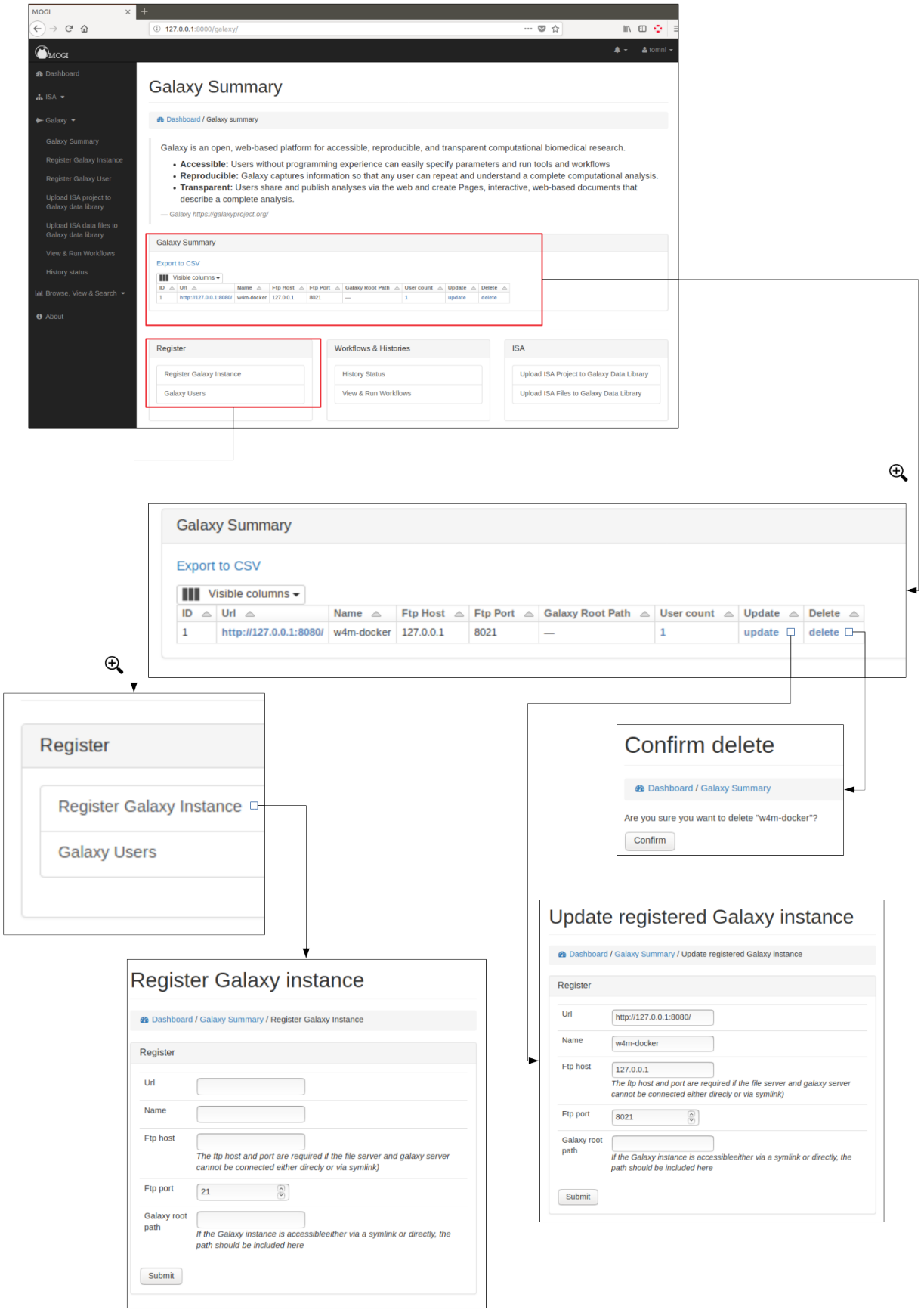
3.1.3 Interact with Galaxy

The functionality is summarised at http://127.0.0.1:8000/galaxy/

Register Galaxy details

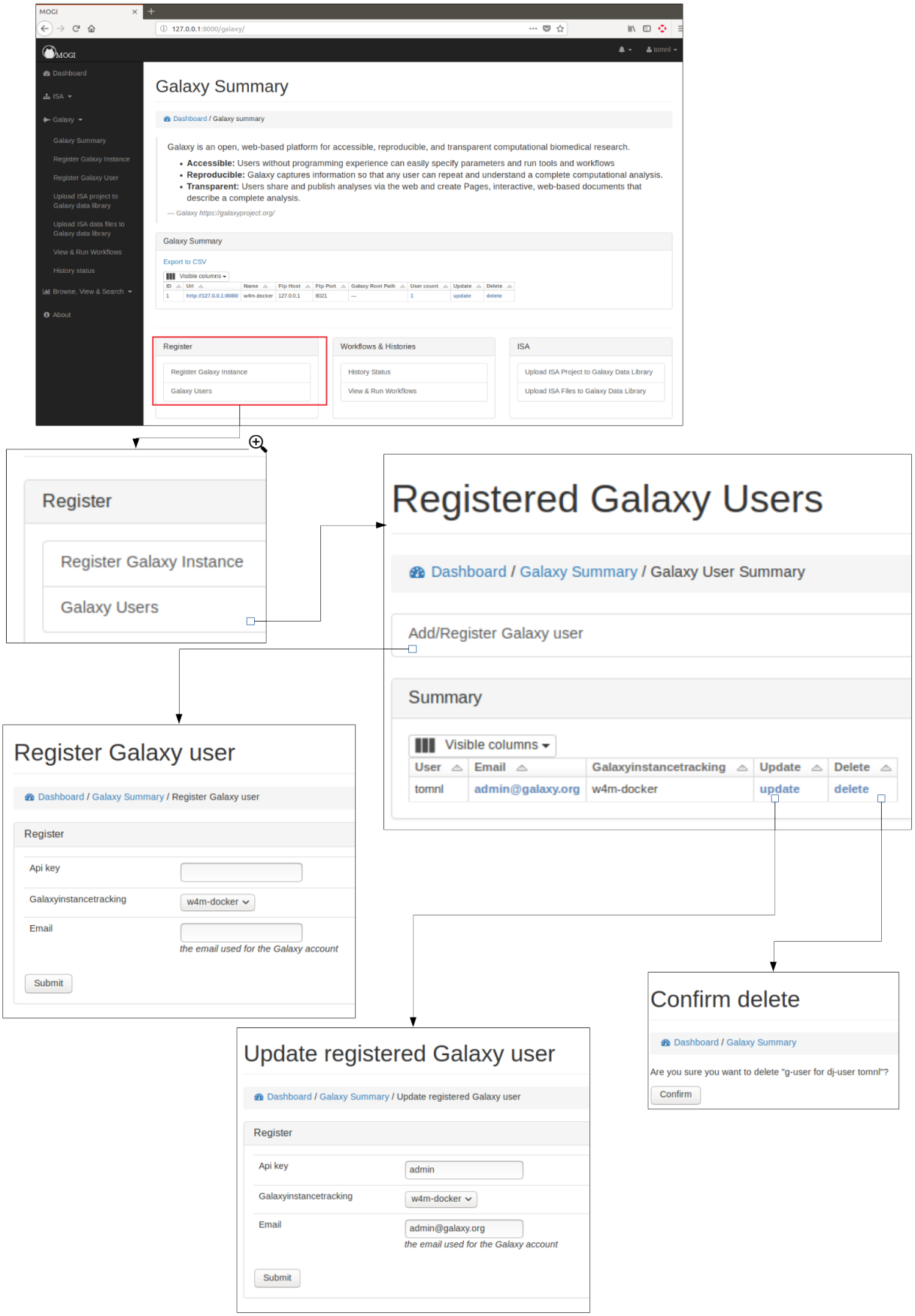
Register Galaxy instance

A Galaxy instance needs to be registered before any of the django-galaxy functionality can be used. If the Galaxy instance is accessible directly via the files system of a symbolic link on the file system then a the root of the Galaxy path should be included when registering. Alternatively the FTP site for the Galaxy instance can be recorded.


MOGI
127.0.0.1:8000/galaxy/
Dashboard
ISA
Galaxy
Galaxy Summary
Register Galaxy Instance
Register Galaxy User
Upload ISA project to Galaxy data library
Upload ISA data files to Galaxy data library
View & Run Workflows
History status
Browse, View & Search
About
Galaxy Summary
Dashboard / Galaxy summary
Galaxy is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research.
Accessible: Users without programming experience can easily specify parameters and run tools and workflows
Reproducible: Galaxy captures information so that any user can repeat and understand a complete computational analysis.
Transparent: Users share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis.
— Galaxy https://galaxyproject.org/
Export to CSV
Visible columns
ID
Url
Name
Ftp Host
Ftp Port
Galaxy Root Path
User count
Update
Delete
1
http://127.0.0.1:8080/
w4m-docker
127.0.0.1
8021
—
1
update
delete
Register
Register Galaxy Instance
Galaxy Users
Workflows & Histories
History Status
View & Run Workflows
ISA
Upload ISA Project to Galaxy Data Library
Upload ISA Files to Galaxy Data Library
Confirm delete
Dashboard / Galaxy Summary
Are you sure you want to delete "w4m-docker"?
Confirm
Register Galaxy instance
Dashboard / Galaxy Summary / Register Galaxy Instance
Register
Url
Name
Ftp host
The ftp host and port are required if the file server and galaxy server cannot be connected either direcly or via symlink)
Ftp port
21
Galaxy root path
If the Galaxy instance is accessibleeither via a symlink or directly, the path should be included here
Submit
Update registered Galaxy instance
Dashboard / Galaxy Summary / Update registered Galaxy instance
Register
Url
http://127.0.0.1:8080/
Name
w4m-docker
Ftp host
127.0.0.1
The ftp host and port are required if the file server and galaxy server cannot be connected either direcly or via symlink)
Ftp port
8021
Galaxy root path
If the Galaxy instance is accessibleeither via a symlink or directly, the path should be included here
Submit

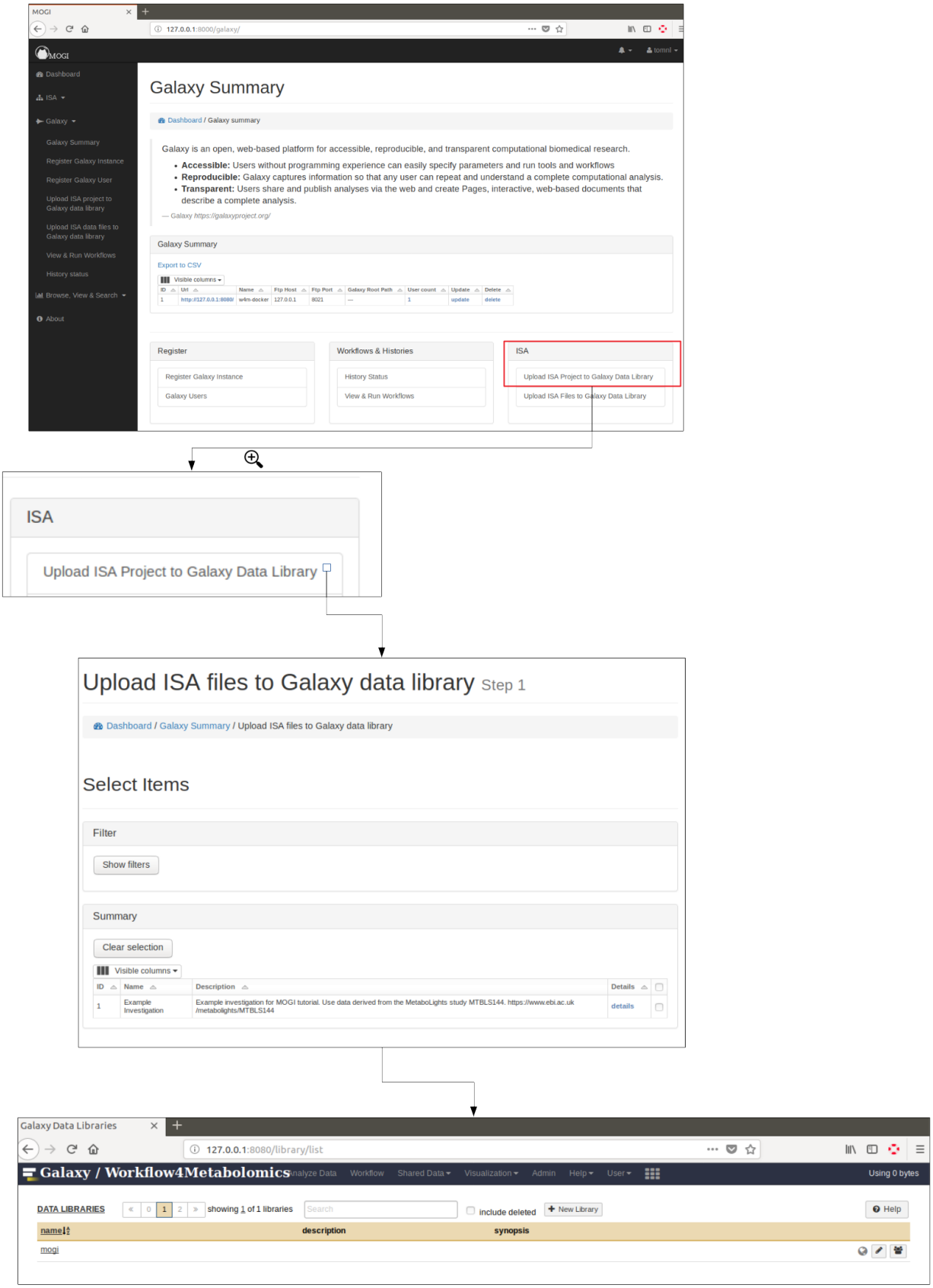
Register Galaxy user

Each MOGI user (e.g. Django user) can be registered to any of the registered Galaxy instance. The API key that has is provided by the user allows permission of the Galaxy instance API to be used.


MOGI
127.0.0.1:8000/galaxy/
Dashboard
ISA
Galaxy
Galaxy Summary
Register Galaxy Instance
Register Galaxy User
Upload ISA project to Galaxy data library
Upload ISA data files to Galaxy data library
View & Run Workflows
History status
Browse, View & Search
About
Galaxy Summary
Dashboard / Galaxy summary
Galaxy is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research.
Accessible: Users without programming experience can easily specify parameters and run tools and workflows
Reproducible: Galaxy captures information so that any user can repeat and understand a complete computational analysis.
Transparent: Users share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis.
— Galaxy https://galaxyproject.org/
Galaxy Summary
Export to CSV
Visible columns
Register
Register Galaxy Instance
Galaxy Users
Workflows & Histories
History Status
View & Run Workflows
ISA
Upload ISA Project to Galaxy Data Library
Upload ISA Files to Galaxy Data Library
Register
Register Galaxy Instance
Galaxy Users
Registered Galaxy Users
Dashboard / Galaxy Summary / Galaxy User Summary
Add/Register Galaxy user
Summary
Visible columns
User
Email
Galaxyinstancetracking
Update
Delete
tomnl
admin@galaxy.org
w4m-docker
update
delete
Register Galaxy user
Dashboard / Galaxy Summary / Register Galaxy user
Register
Api key
Galaxyinstancetracking
w4m-docker
Email
the email used for the Galaxy account
Submit
Confirm delete
Dashboard / Galaxy Summary
Are you sure you want to delete "g-user for dj-user tomnl"?
Confirm
Update registered Galaxy user
Dashboard / Galaxy Summary / Update registered Galaxy user
Register
Api key
admin
Galaxyinstancetracking
w4m-docker
Email
admin@galaxy.org
the email used for the Galaxy account
Submit

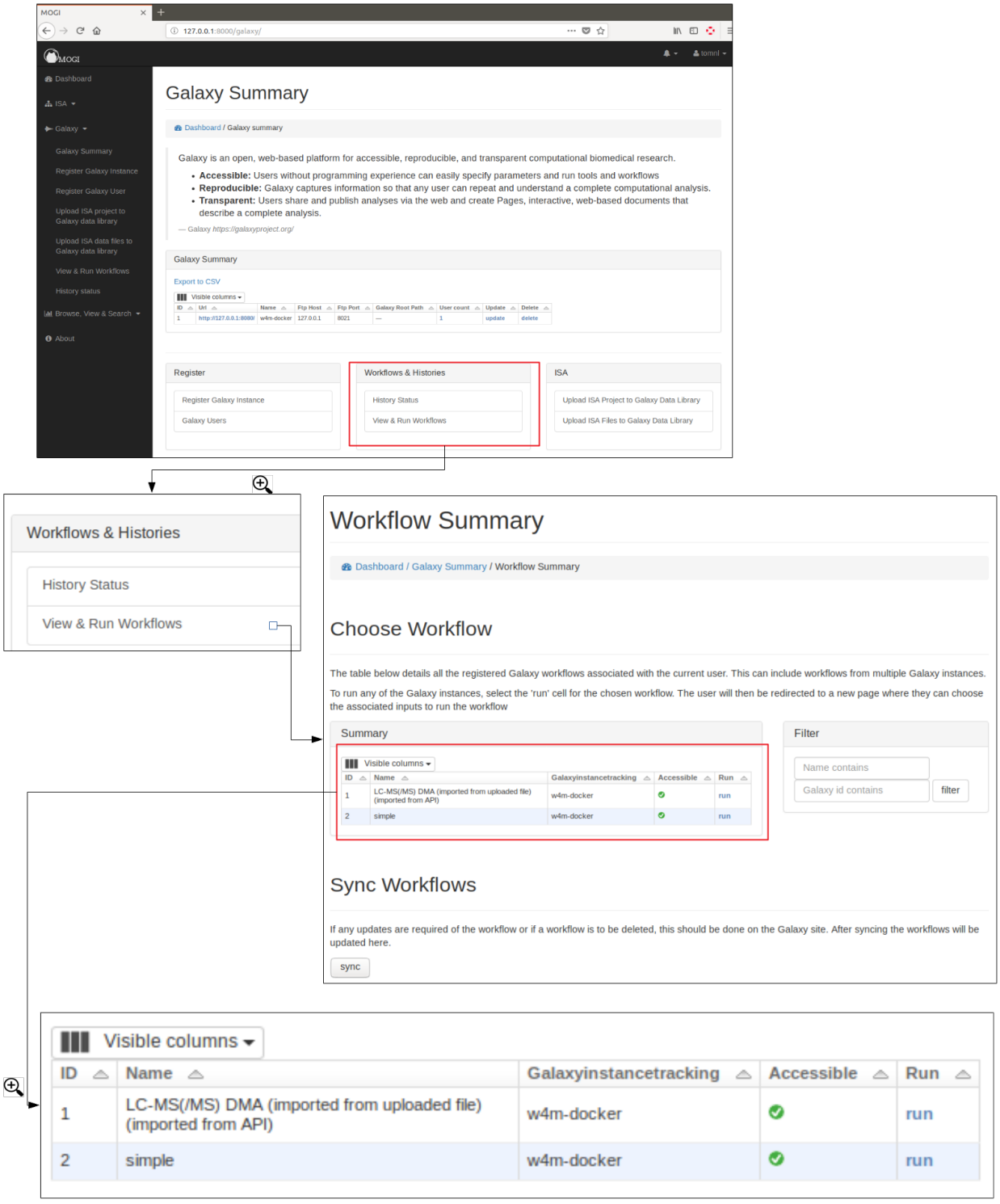
Upload ISA projects to Galaxy data library

If a ISA project has be created and data has been uploaded into an assay. This data can be uploaded into the Galaxy Data libraries. If the Galaxy instance is on the same file system (or accessible by a symbolic link) then the files can be uploaded as symbolic links. This is useful if there is limited space available for Galaxy instance. Alternatively the files can be uploaded via FTP.

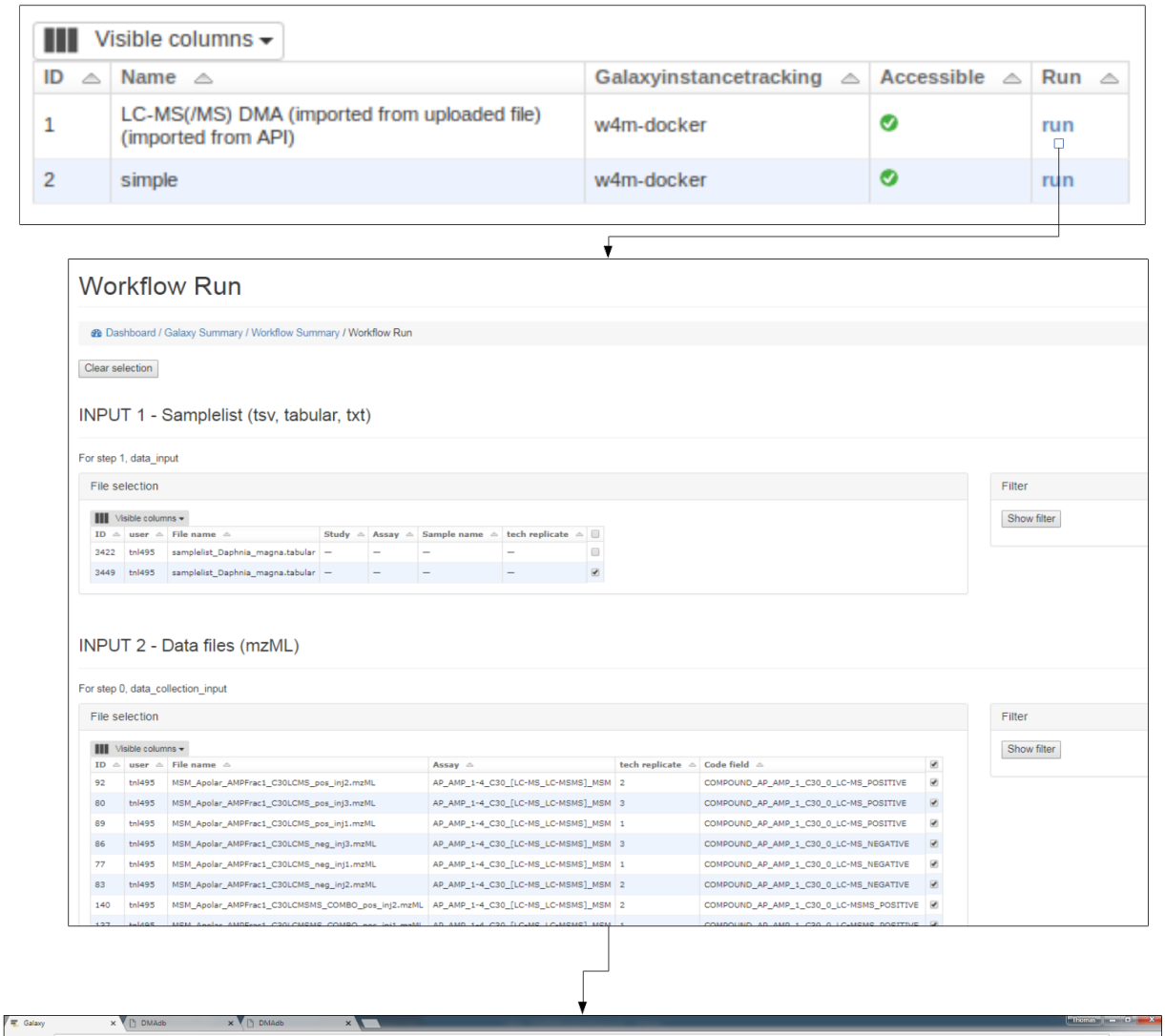

MOGI
127.0.0.1:8000/galaxy/
Dashboard
ISA
Galaxy
Galaxy Summary
Register Galaxy Instance
Register Galaxy User
Upload ISA project to Galaxy data library
Upload ISA data files to Galaxy data library
View & Run Workflows
History status
Browse, View & Search
About
Galaxy Summary
Dashboard / Galaxy summary
Galaxy is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research.
Accessible: Users without programming experience can easily specify parameters and run tools and workflows
Reproducible: Galaxy captures information so that any user can repeat and understand a complete computational analysis.
Transparent: Users share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis.
— Galaxy https://galaxyproject.org/
Galaxy Summary
Export to CSV
Visible columns
Register
Register Galaxy Instance
Galaxy Users
Workflows & Histories
History Status
View & Run Workflows
ISA
Upload ISA Project to Galaxy Data Library
Upload ISA Files to Galaxy Data Library
ISA
Upload ISA Project to Galaxy Data Library
Upload ISA files to Galaxy data library Step 1
Dashboard / Galaxy Summary / Upload ISA files to Galaxy data library
Select Items
Filter
Show filters
Summary
Clear selection
Visible columns
ID
Name
Description
Details
1
Example Investigation
Example investigation for MOGI tutorial. Use data derived from the MetaboLights study MTBLS144. https://www.ebi.ac.uk/metabolights/MTBLS144
details
Galaxy Data Libraries
127.0.0.1:8080/library/list
Galaxy / Workflow4Metabolomics
Analyze Data
Workflow
Shared Data
Visualization
Admin
Help
User
Using 0 bytes
DATA LIBRARIES
showing 1 of 1 libraries
Search
include deleted
New Library
Help
name
description
synopsis
mogi

Run Workflows

All workflows for each Galaxy instances can be synced with MOGI (Django) so they can be run directly from the MOGI interface. If any new Galaxy workflow has been added to an instance make sure the **sync** button is pressed.

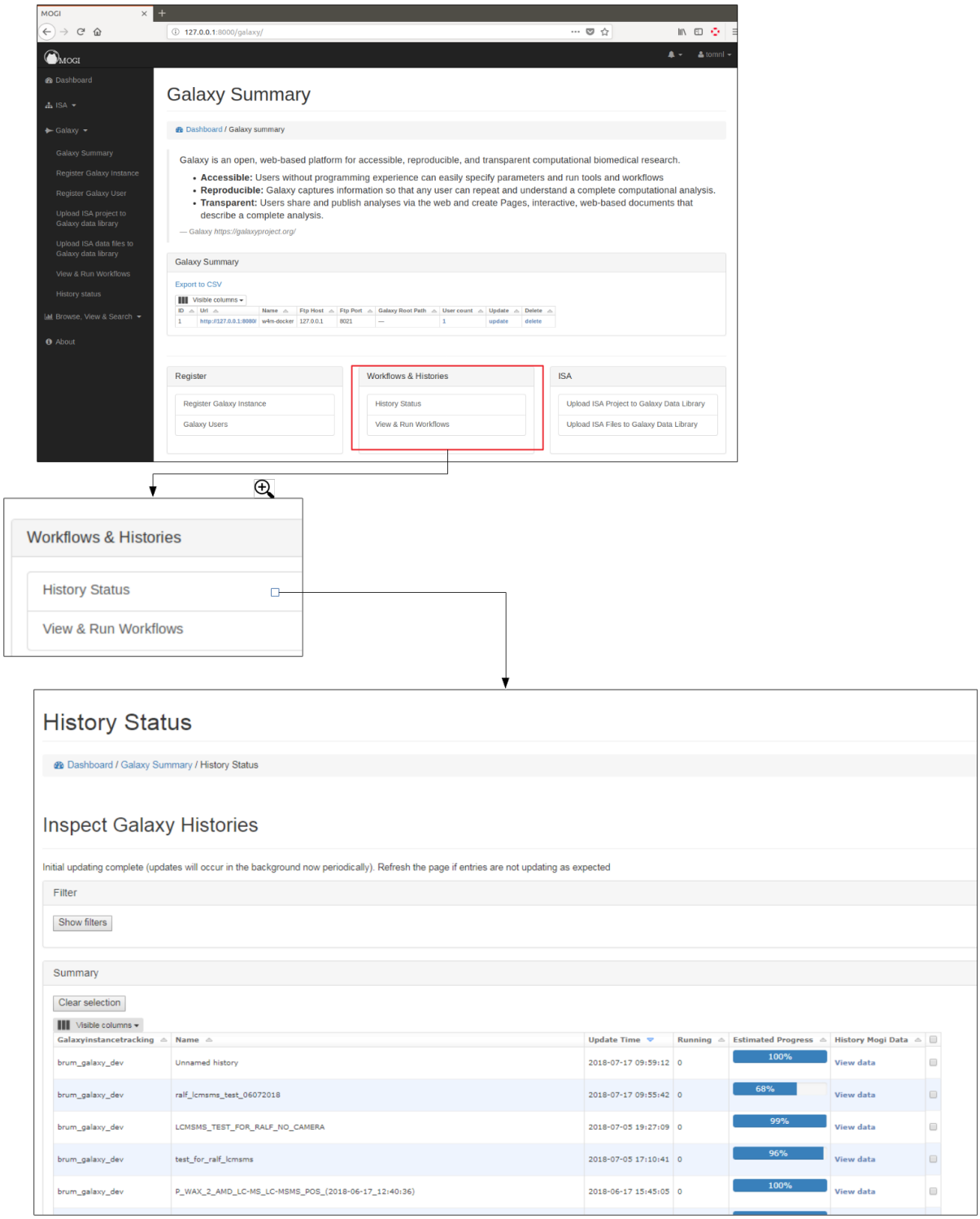


Once the data files have been uploaded onto a Galaxy instance (see section **Upload ISA projects to Galaxy data library**. The files will be visible when a user chooses to perform a workflow. The relevant files can be filtered based on what protocol or sample was used. A samplelist is automatically when an ISA project is uploaded into the Galaxy data libraries. This can be useful if a samplelist is needed for the data analysis.


Visible columns
ID
Name
Galaxyinstancetracking
Accessible
Run
1
LC-MS(/MS) DMA (imported from uploaded file) (imported from API)
w4m-docker
run
2
simple
w4m-docker
run
Workflow Run
Dashboard / Galaxy Summary / Workflow Summary / Workflow Run
Clear selection
INPUT 1 - Samplelist (tsv, tabular, txt)
For step 1, data_input
File selection
Filter
Show filter
INPUT 2 - Data files (mzML)
For step 0, data_collection_input
File selection
Filter
Show filter
Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Admin
Help
User
Tools
History
Hello, Galaxy is running!
To customize this page edit static/welcome.html
Configuring Galaxy »
Installing Tools »
Take an interactive tour:
Galaxy is an open platform for supporting data intensive research. Galaxy is developed by The Galaxy Team with the support of many contributors.

View Galaxy histories

All histories for each Galaxy instances are summarised within MOGI.



LC-MS(/MS) annotation workflow and importing data to MOGI

An LC-MS(/MS) annotation workflow is provided within the mogi docker. The worklfow performs various data processing and annotation steps and generates an SQLite database containing all the relevant peak and annotation information. The final step of the workflow allows this SQLite database to be exported to the MOGI database.

More details to come!

3.1.4 Interact with Galaxy

The functionality is summarised at http://127.0.0.1:8000/mbrowse/

Upload LC-MS(/MS) dataset

Browse and view peaklists

Browse and view annotations

Search m/z and fragmentation spectra

CHAPTER 4

django-mogi

Metabolome Organisation with Galaxy and ISA (imports all of the django-applications within the Django MOGI application suite)

4.1 django-mogi

Metabolomics organisation with Galaxy and ISA

4.1.1 Quick start

1. Add "mogi" and django application dependencies to your INSTALLED_APPS setting like this (mogi should come before gfiles, galaxy, mbrowse and misa):

```
INSTALLED_APPS = [
    ...
    'mogi',
    'misa',
    'mbrowse',
    'galaxy',
    'gfiles',

    'django_tables2',
    'django_tables2_column_shifter',
    'django_filters',
    'bootstrap3',
    'django_sb_admin',
    'dal',
    'dal_select2',
]
```

2. Include the URLconf in your project urls.py like this:

```
url(r'^', include('gfiles.urls')),
url('mogi/', include('mogi.urls')),
url('mbrowse/', include('mbrowse.urls')),
url('misa/', include('misa.urls')),
url('galaxy/', include('galaxy.urls')),
```

3. Run *python manage.py migrate* to create the mogi models.
4. Start the development server and visit http://127.0.0.1:8000/
5. Register http://127.0.0.1:8000/register/ and login http://127.0.0.1:8000/login/
6. General overview http://127.0.0.1:8000
7. Create, edit, view and export ISA projects http://127.0.0.1:8000/misa/ilist/
8. Upload to Galaxy, run Galaxy workflows and view Galaxy histories http://127.0.0.1:8000/misa/ilist/
9. Browse, view and search metabolomic datasets http://127.0.0.1:8000/mbrowse/general_summary/

4.1.2 API reference

django-mogi models

class mogi.models.**AnnotationSummary**(*id*, *compound*, *lcms_ann_level*, *best_score*, *assays*, *mzmin*, *mzmax*, *rtmin*, *rtmax*)

exception DoesNotExist

exception MultipleObjectsReturned

class mogi.models.**CAnnotationMOGI**(*id*, *cannotation*)

exception DoesNotExist

exception MultipleObjectsReturned

class mogi.models.**CPeakGroupMetaMOGI**(*id*, *date*, *metabinputdata*, *cpeakgroupmeta_ptr*, *historydatamogi*)

exception DoesNotExist

exception MultipleObjectsReturned

class mogi.models.**HistoryDataMOGI**(*id*, *data_file*, *original_filename*, *user*, *genericfile_ptr*, *history*, *name*, *historydata_ptr*, *investigation*)

exception DoesNotExist

exception MultipleObjectsReturned

class mogi.models.**ISAGalaxyTrack**(*id*, *galaxyinstancetracking*, *isatogalaxyparam*, *investigation*, *galaxy_id*)

exception DoesNotExist

exception MultipleObjectsReturned

class mogi.models.**IncomingGalaxyData**(*id*, *galaxy_url*, *galaxy_name*, *galaxy_data_id*, *galaxy_history_id*, *galaxy_history_name*, *other_details*)

```
exception DoesNotExist
exception MultipleObjectsReturned
```

django-mogi filters

django-mogi views

django-mogi filters

django-mogi tables

django-mogi utils

CHAPTER 5

django-galaxy

Django interfacing with Galaxy. Backend using the bioblend API.

5.1 django-galaxy

Django interfacing with Galaxy. Backend using the bioblend API.

Galaxy is a web based workflow platform that can be used to perform bioinformatics in a reproducible and sharable environment.

5.1.1 Quick start

1. Add "galaxy" and django application dependencies to your INSTALLED_APPS setting like this (galaxy should come before gfiles):

```
INSTALLED_APPS = [
    ...
    'galaxy',
    'gfiles',

    'django_tables2',
    'bootstrap3',
    'django_tables2_column_shifter',
    'django_sb_admin',
    'django_filter'
]
```

2. Include the polls URLconf in your project urls.py like this:

```
url(r'^', include('gfiles.urls')),
path('galaxy/', include('galaxy.urls')),
```

3. Run *python manage.py migrate* to create the models.
4. Start the development server and visit http://127.0.0.1:8000
5. Register http://127.0.0.1:8000/register/ and login http://127.0.0.1:8000/login/
6. View summary of Galaxy instances and options http://127.0.0.1:8000/galaxy/galaxy_summary/
7. Register Galaxy instance http://127.0.0.1:8000/galaxy/addgi/
8. Register Galaxy user http://127.0.0.1:8000/galaxy/addguser/
9. Upload file(s) to Django (for bulk upload of files see django-misa and django-mogi) http://127.0.0.1:8000/upload_gfile/
10. Upload file(s) to Galaxy data library (for bulk upload of files see django-misa and django-mogi) http://127.0.0.1:8000/galaxy/files_to_galaxy_datalib/
11. Run workflow (for running workflows with ISA data see django-misa and django-mogi) http://127.0.0.1:8000/galaxy/workflow_summary/
12. View histories http://127.0.0.1:8000/galaxy/history_status/

5.1.2 API reference

django-galaxy models

class galaxy.models.**FilesToGalaxyDataLibraryParam**(*id*, *added_by*, *folder_name*, *galaxyinstancetracking*, *link2files*, *local_path*, *ftp*, *remove*)

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**GalaxyFileLink**(*id*, *galaxy_id*, *galaxy_library*, *genericfile*, *galaxyinstancetracking*, *removed*)

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**GalaxyInstanceTracking**(**args*, ***kwargs*)

Model for tracking Galaxy instances and associated ftp sites

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**GalaxyUser**(**args*, ***kwargs*)

Model for linking a Galaxy user to a Django user

A django user can be linked to many Galaxy instances and each Galaxy User HAS to be linked to Galaxy instance

django-user [1 — *] galaxy-users

galaxy-user [* — 1] galaxy-instances

However, a django user can't be linked to multiple of the same galaxy instances

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**GenericFilesToGalaxyHistoryParam**(*id*, *added_by*, *history_name*, *galaxyinstancetracking*)

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**History**(*id*, *update_time*, *galaxyinstancetracking*, *name*, *empty*, *error*, *failed_metadata*, *new*, *ok*, *paused*, *running*, *queued*, *setting_metadata*, *upload*, *galaxy_id*, *estimated_progress*)

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**HistoryData**(*id*, *data_file*, *original_filename*, *user*, *genericfile_ptr*, *history*, *name*)

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**Workflow**(**args*, ***kwargs*)
Model for Galaxy workflows. The workflow needs to be associated with a Valid Galaxy instance

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**WorkflowInput**(*id*, *name*, *step*, *datatype*, *workflow*)

exception DoesNotExist

exception MultipleObjectsReturned

class galaxy.models.**WorkflowRun**(*id*, *rundate*, *ran_by*, *workflow*, *library*, *history_name*)

exception DoesNotExist

exception MultipleObjectsReturned

django-galaxy views

class galaxy.views.**FilesToGalaxyDataLib**(***kwargs*)
Select Files to be added to Galaxy data Library

Inherit the GFileListView that view that shows current files and allows some basic filtering

class galaxy.views.**GalaxyInstanceCreateView**(***kwargs*)
Create a Galaxy instance to track in django. Note that the Galaxy needs to be accessible at the point of initialisation.

If file transfer is required to Galaxy that is not located on the same server as the Django server then the associated FTP host & post details need to be added as well, see galaxy docs

User login required

form_class
alias of `galaxy.forms.GalaxyInstanceTrackingForm`

model
alias of *galaxy.models.GalaxyInstanceTracking*

class galaxy.views.**GalaxyInstanceTrackingDeleteView**(***kwargs*)

model
alias of *galaxy.models.GalaxyInstanceTracking*

class galaxy.views.**GalaxyInstanceTrackingUpdateView**(***kwargs*)

form_class
alias of galaxy.forms.GalaxyInstanceTrackingForm

model
alias of *galaxy.models.GalaxyInstanceTracking*

class galaxy.views.**GalaxySummaryView**(***kwargs*)

model
alias of *galaxy.models.GalaxyInstanceTracking*

class galaxy.views.**GalaxySync**(***kwargs*)
Sync workflows from available galaxy instances for the current user.

This will add any workflows to the django database that have not already been added. And will update any that have been updated on the galaxy instance.

todo: It might be worth changing this view to a more general GalaxySync option. That syncs workflows & files

class galaxy.views.**GalaxyUserCreateView**(***kwargs*)
Register a Galaxy user to a Galaxy instance that we have tracked.

A django user can be linked to many Galaxy instances and each Galaxy User HAS to be linked to Galaxy instance

django-user [1 — *] galaxy-users

galaxy-user [* — 1] galaxy-instances

However, a django user can't be linked to multiple of the same galaxy instances

form_class
alias of galaxy.forms.GalaxyUserForm

form_valid(*form*)
If the form is valid, save the associated model.

get_form_kwargs()
Returns the keyword arguments for instantiating the form.

get_initial()
Returns the initial data to use for forms on this view.

model
alias of *galaxy.models.GalaxyUser*

class galaxy.views.**GalaxyUserDeleteView**(***kwargs*)

model
alias of *galaxy.models.GalaxyUser*

class galaxy.views.**GalaxyUserListView**(***kwargs*)

model
alias of *galaxy.models.GalaxyUser*

class galaxy.views.**GalaxyUserUpdateView**(***kwargs*)

form_class
alias of galaxy.forms.GalaxyUserForm

model
alias of *galaxy.models.GalaxyUser*

class galaxy.views.**GenericFilesToGalaxyHistory**(***kwargs*)

class galaxy.views.**HistoryDataBioBlendListView**(***kwargs*)

class galaxy.views.**HistoryDataCreateView**(***kwargs*)

form_valid(*form*)
If the form is valid, save the associated model.

get_initial()
Returns the initial data to use for forms on this view.

model
alias of *galaxy.models.HistoryData*

class galaxy.views.**HistoryListView**(***kwargs*)
View and initiate a run for all registered workflows.

Workflows can also be synced here as well

model
alias of *galaxy.models.History*

class galaxy.views.**TableFileSelectMixin**
General class for file selection with ajax multipage

class galaxy.views.**WorkflowCreateView**(***kwargs*)

form_valid(*form*)
If the form is valid, save the associated model.

get_form_kwargs(***kwargs*)
Returns the keyword arguments for instantiating the form.

get_initial()
Returns the initial data to use for forms on this view.

model
alias of *galaxy.models.Workflow*

class galaxy.views.**WorkflowListView**(***kwargs*)
View and initiate a run for all registered workflows.

Workflows can also be synced here as well

model
alias of *galaxy.models.Workflow*

class galaxy.views.**WorkflowRunView**(***kwargs*)
Run a registered workflow

filter_class
alias of *gfiles.filter.GFileFilter*

table_class
alias of *gfiles.tables.GFileTable*

class galaxy.views.**WorkflowStatus**(***kwargs*)
View available Galaxy. If any new workflows are added to a Galaxy instance the user should sync first before they can be seen in the table.

django-galaxy filters

django-galaxy tables

django-galaxy utils

CHAPTER 6

django-misa

ISA organisation for metabolomic studies with Django

6.1 django-misa

ISA organisation for metabolomic studies with Django

6.1.1 Quick start

1. Add “mbrowse” and django application dependencies to your INSTALLED_APPS setting like this (mbrowse should come before gfiles):

```
INSTALLED_APPS = [
    ...
    'mbrowse',
    'gfiles',

    'django_tables2',
    'bootstrap3',
    'django_tables2_column_shifter',
    'django_sb_admin',
    'django_filter'
    'dal',
    'dal_select2',
]
```

2. Include the polls URLconf in your project urls.py like this:

```
url(r'^', include('gfiles.urls')),
path('mbrowse/', include('mbrowse.urls')),
```

3. Run *python manage.py migrate* to create the mbrowse models.
4. Start the development server and visit http://127.0.0.1:8000/mbrowse/general_summary
5. Register http://127.0.0.1:8000/register/ and login http://127.0.0.1:8000/login/
6. Upload metabolomics mzML files (can also be done with djang-misa and django-mogi) http://127.0.0.1:8000/mbrowse/upload_mfiles_batch/
7. Upload LC-MS data set (can be done through galaxy, see django-mogi) http://127.0.0.1:8000/mbrowse/upload_lcms_dataset/
8. Browse and view the datasets http://127.0.0.1:8000/mbrowse/cpeakgroupmeta_summary/
9. Browse and view the annotations http://127.0.0.1:8000/mbrowse/cpeakgroupmeta_summary/
10. Search the datasets http://127.0.0.1:8000/mbrowse/search_result_summary/

6.1.2 API reference

django-misa models

class misa.models.**Assay**(*id*, *study*, *description*, *name*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**AssayDetail**(*id*, *assay*, *code_field*, *studysample*, *samplecollectionprocess*, *extractionprocess*, *speprocess*, *chromatographyprocess*, *measurementprocess*, *datatransformationprocess*, *metaboliteidentifcationprocess*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**AssayRun**(*id*, *assaydetail*, *run*, *technical_replicate*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**ChromatographyProcess**(*id*, *chromatographyprotocol*, *date*, *details*, *chromatographyfrac*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**ChromatographyProtocol**(*id*, *name*, *description*, *uri*, *version*, *code_field*, *chromatographytype*, *instrument_name*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**ChromatographyType**(*id*, *type*, *description*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**DataTransformationProcess**(*id*, *datatransformationprotocol*, *details*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**DataTransformationProtocol**(*id*, *name*, *description*, *uri*, *version*, *code_field*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**ExtractionProcess**(*id*, *extractionprotocol*, *date*, *details*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**ExtractionProtocol**(*id*, *name*, *description*, *uri*, *version*, *code_field*, *extractiontype*, *postextraction*, *derivitisation*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**ExtractionType**(*id*, *type*, *description*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**Investigation**(*id*, *name*, *description*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**MISAFile**(*id*, *data_file*, *original_filename*, *user*, *genericfile_ptr*, *investigation*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**MeasurementProcess**(*id*, *measurementprotocol*, *date*, *details*, *polaritytype*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**MeasurementProtocol**(*id*, *name*, *description*, *uri*, *version*, *code_field*, *measurementtechnique*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**MeasurementTechnique**(*id*, *type*, *description*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**MetaboliteIdentificationProcess**(*id*, *metaboliteidentificationprotocol*, *details*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**MetaboliteIdentificationProtocol**(*id*, *name*, *description*, *uri*, *version*, *code_field*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**OntologyTerm**(*id*, *name*, *description*, *ontology_id*, *iri*, *obo_id*, *ontology_name*, *ontology_prefix*, *short_form*, *type*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**Organism**(*id*, *ontologyterm*, *name*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**OrganismPart**(*id*, *ontologyterm*, *name*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**PolarityType**(*id*, *type*, *description*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**SampleCollectionProcess**(*id*, *samplecollectionprotocol*, *details*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**SampleCollectionProtocol**(*id*, *name*, *description*, *uri*, *version*, *code_field*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**SampleType**(*id*, *type*, *ontologyterm*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**SpeProcess**(*id*, *speprotocol*, *date*, *details*, *spefrac*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**SpeProtocol**(*id*, *name*, *description*, *uri*, *version*, *code_field*, *spetype*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**SpeType**(*id*, *type*, *description*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**Study**(*id*, *investigation*, *description*, *dmastudy*, *name*, *title*, *grant_number*, *funding_agency*, *submission_date*, *public_release_date*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**StudyFactor**(*id*, *ontologyterm_type*, *ontologyterm_value*, *value*, *ontologyterm_unit*, *full_name*)

exception DoesNotExist

exception MultipleObjectsReturned

class misa.models.**StudySample**(*id*, *study*, *sample_name*, *source_name*, *organism*, *organism_part*, *sampletype*, *samplecollectionprocess*)

exception DoesNotExist

exception MultipleObjectsReturned

django-misa filters

django-misa views

django-misa filters

django-misa tables

django-misa utils

CHAPTER 7

django-mbrowse

Browse, view and search metabolomic datasets

7.1 django-mbrowse

Browse, view and search metabolomic datasets

7.1.1 Quick start

1. Add "mbrowse" and django application dependencies to your INSTALLED_APPS setting like this (mbrowse should come before gfiles):

```
INSTALLED_APPS = [
    ...
    'mbrowse',
    'gfiles',

    'django_tables2',
    'bootstrap3',
    'django_tables2_column_shifter',
    'django_sb_admin',
    'django_filter'
    'dal',
    'dal_select2',
]
```

2. Include the polls URLconf in your project urls.py like this:

```
url(r'^', include('gfiles.urls')),
path('mbrowse/', include('mbrowse.urls')),
```

3. Run *python manage.py migrate* to create the mbrowse models.
4. Start the development server and visit http://127.0.0.1:8000/mbrowse/general_summary
5. Register http://127.0.0.1:8000/register/ and login http://127.0.0.1:8000/login/
6. Upload metabolomics mzML files (can also be done with djang-misa and django-mogi) http://127.0.0.1:8000/mbrowse/upload_mfiles_batch/
7. Upload LC-MS data set (can be done through galaxy, see django-mogi) http://127.0.0.1:8000/mbrowse/upload_lcms_dataset/
8. Browse and view the datasets http://127.0.0.1:8000/mbrowse/cpeakgroupmeta_summary/
9. Browse and view the annotations http://127.0.0.1:8000/mbrowse/cpeakgroupmeta_summary/
10. Search the datasets http://127.0.0.1:8000/mbrowse/search_result_summary/

7.1.2 API reference

django-mbrowse models

class mbrowse.models.models_general.**AdductRule**(*id*, *adduct_type*, *nmol*, *charge*, *massdiff*, *oidscore*, *quasi*, *ips*, *frag_score*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_general.**MFile**(*id*, *data_file*, *original_filename*, *user*, *generic_file_ptr*, *run*, *mfilesuffix*, *prefix*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_general.**MFileSuffix**(*id*, *suffix*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_general.**MetabInputData**(*id*, *date*, *gfile*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_general.**Polarity**(*id*, *polarity*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_general.**Run**(*id*, *prefix*, *polarity*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**Adduct**(*id*, *idi*, *adductrule*, *cpeakgroup*, *neutralmass*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**CPeak**(*id*, *idi*, *mz*, *mzmin*, *mzmax*, *rt*, *rtmin*, *rtmax*, *_into*, *intb*, *maxo*, *sn*, *rtminraw*, *rtmaxraw*, *xcmsfileinfo*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**CPeakGroup**(*id*, *idi*, *mzmed*, *mzmin*, *mzmax*, *rtmed*, *rtmin*, *rtmax*, *npeaks*, *isotopes*, *adducts*, *pcgroup*, *msms_count*, *cpeakgroupmeta*, *best_annotation*, *best_score*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**CPeakGroupLink**(*id*, *cpeak*, *cpeakgroup*, *best_feature*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**CPeakGroupMeta**(*id*, *date*, *metabinputdata*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**Eic**(*id*, *idi*, *scan*, *intensity*, *rt_raw*, *rt_corrected*, *purity*, *cpeak*, *cpeakgroup*, *eicmeta*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**EicMeta**(*id*, *metabinputdata*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**Isotope**(*id*, *idi*, *cpeakgroup1*, *cpeakgroup2*, *iso*, *charge*, *metabinputdata*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**NeutralMass**(*id*, *idi*, *nm*, *ips*, *metabinputdata*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**SPeakMetaCPeakFragLink**(*id*, *cpeak*, *speakmeta*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_cpeaks.**XCMSFileInfo**(*id*, *idi*, *filename*, *classname*, *mfile*, *metabinputdata*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_speaks.**SPeak**(*id*, *speakmeta*, *mz*, *i*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_speaks.**SPeakMeta**(*id*, *run*, *idi*, *scan_idi*, *precursor_mz*, *precursor_i*, *scan_num*, *precursor_scan_num*, *precursor_nearest*, *precursor_rt*, *ms_level*, *metabinputdata*, *a_mz*, *a_purity*, *a_pknm*, *i_mz*, *i_purity*, *i_pknm*, *in_purity*, *in_pknm*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_libraries.**LibrarySpectra**(*id*, *library_spectra_meta*, *mz*, *i*, *other*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_libraries.**LibrarySpectraMeta**(*id*, *name*, *collision_energy*, *ms_level*, *accession*, *resolution*, *polarity*, *fragmentation_type*, *precursor_mz*, *precursor_type*, *instrument_type*, *instrument*, *copyright*, *column*, *mass_accuracy*, *mass_error*, *origin*, *library_spectra_source*, *inchikey*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_libraries.**LibrarySpectraSource**(*id*, *name*, *description*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_compounds.**Compound**(*created_at*, *updated_at*, *inchikey_id*, *name*, *systematic_name*, *iupac_name*, *trade_name*, *other_names*, *molecular_formula*, *smiles*, *pubchem_id*, *chemspider_id*, *kegg_id*, *hmdb_id*, *lmdb_id*, *lbdb_id*, *humancyc_id*, *chebi_id*, *metlin_id*, *foodb_id*, *monoisotopic_mass*, *exact_mass*, *molecular_weight*, *xlogp*, *category*, *compound_class*, *sub_class*, *FA*, *brite*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_annotations.**CAnnotation**(*id*, *compound*, *cpeakgroup*, *spectral_matching_average_score*, *metfrag_average_score*, *mzcloud_average_score*, *sirius_csifingerid_average_score*, *ms1_average_score*, *weighted_score*, *rank*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_annotations.**CAnnotationWeight**(*id*, *spectral_matching_weight*, *mzcloud_weight*, *sirius_csifingerid_weight*, *ms1_weight*, *metfrag_weight*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_annotations.**CSIFingerIDAnnotation**(*id*, *idi*, *s_peak_meta*, *inchikey2d*, *molecular_formula*, *rank*, *score*, *name*, *links*, *smiles*, *rank_score*, *csifingeridmeta*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_annotations.**CSIFingerIDMeta**(*id*, *datetime_stamp*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_annotations.**MetFragAnnotation**(*id*, *idi*, *s_peak_meta*, *explained_peaks*, *formula_explained_peaks*, *maximum_tree_depth*, *fragmentor_score*, *fragmentor_score_values*, *number_peaks_used*, *score*, *compound*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_annotations.**ProbmetabAnnotation**(*id*, *idi*, *cpeakgroup*, *compound*, *prob*)

exception DoesNotExist

exception MultipleObjectsReturned

class mbrowse.models.models_annotations.**SpectralMatching**(*id*, *idi*, *s_peak_meta*, *score*, *percentage_match*, *match_num*, *library_spectra_meta*, *accession*, *name*)

exception DoesNotExist

exception MultipleObjectsReturned

django-mbrowse filters

django-mbrowse views

class mbrowse.views.views_general.**GeneralSummaryView**(***kwargs*)

class mbrowse.views.views_general.**MFileAutocomplete**(***kwargs*)

get_queryset()
Filter the queryset with GET['q'].

class mbrowse.views.views_general.**RunCreateView**(***kwargs*)

model
alias of *mbrowse.models.models_general.Run*

class mbrowse.views.views_general.**UploadAdductRules**(***kwargs*)

class mbrowse.views.views_general.**UploadMFilesBatch**(***kwargs*)

class mbrowse.views.views_cpeaks.**CAnnotationsListAllView**(***kwargs*)

model
alias of *mbrowse.models.models_annotations.CAnnotation*

class mbrowse.views.views_cpeaks.**CAnnotationsListView**(***kwargs*)

model
alias of *mbrowse.models.models_annotations.CAnnotation*

class mbrowse.views.views_cpeaks.**CPeakGroupAllListView**(***kwargs*)

model
alias of *mbrowse.models.models_cpeaks.CPeakGroup*

class mbrowse.views.views_cpeaks.**CPeakGroupListView**(***kwargs*)

model
alias of *mbrowse.models.models_cpeaks.CPeakGroup*

class mbrowse.views.views_cpeaks.**CPeakGroupMetaListView**(***kwargs*)

model
alias of *mbrowse.models.models_cpeaks.CPeakGroupMeta*

class mbrowse.views.views_cpeaks.**EicListView**(***kwargs*)

model
alias of *mbrowse.models.models_cpeaks.Eic*

class mbrowse.views.views_cpeaks.**Frag4FeatureListView**(***kwargs*)

model
alias of *mbrowse.models.models_speaks.SPeak*

class mbrowse.views.views_cpeaks.**UploadLCMSDataset**(***kwargs*)

form_valid(*form*)
If the form is valid, save the associated model.

model
alias of *mbrowse.models.models_general.MetabInputData*

class mbrowse.views.views_libraries.**CPeakGroupSpectralMatchingListView**(***kwargs*)

model
alias of *mbrowse.models.models_annotations.SpectralMatching*

class mbrowse.views.views_libraries.**LibrarySpectraSourceCreateView**(***kwargs*)

form_valid(*form*)
If the form is valid, save the associated model.

model
alias of *mbrowse.models.models_libraries.LibrarySpectraSource*

class mbrowse.views.views_libraries.**SMatchView**(***kwargs*)

model
alias of *mbrowse.models.models_speaks.SPeak*

class mbrowse.views.views_search.**SearchFragParamCreateView**(***kwargs*)

form_valid(*form*)
If the form is valid, save the associated model.

model
alias of mbrowse.models.models_search.SearchFragParam

class mbrowse.views.views_search.**SearchFragResultListView**(***kwargs*)

model
alias of mbrowse.models.models_search.SearchFragResult

class mbrowse.views.views_search.**SearchMzParamCreateView**(***kwargs*)

form_valid(*form*)
If the form is valid, save the associated model.

model
alias of mbrowse.models.models_search.SearchMzParam

class mbrowse.views.views_search.**SearchMzResultListView**(***kwargs*)

model
alias of mbrowse.models.models_search.SearchMzResult

class mbrowse.views.views_search.**SearchNmParamCreateView**(***kwargs*)

form_valid(*form*)
If the form is valid, save the associated model.

model
alias of mbrowse.models.models_search.SearchNmParam

class mbrowse.views.views_search.**SearchNmResultListView**(***kwargs*)

model
alias of mbrowse.models.models_search.SearchNmResult

class mbrowse.views.views_search.**SearchResultSummaryView**(***kwargs*)

django-mbrowse filters

class mbrowse.filter.**MFileFilter**(*data=None*, *queryset=None*, *prefix=None*, *strict=None*, *request=None*)

django-mbrowse tables

class mbrowse.tables.**MFileTable**(**args*, ***kwargs*)

django-mbrowse utils

CHAPTER 8

django-gfiles

Simple file management for generic files in Django

8.1 django-gfiles

Simple file management of generic files in Django. Allows files to be saved as symlinks, tracks the original filename, tracks the user that uploaded the file.

The uploaded files can be viewed and searched using django-tables2 and django-filter applications.

8.1.1 Quick start

1. Add "gfiles" and django application dependencies to your INSTALLED_APPS setting like this:

```
INSTALLED_APPS = [
    ...
    'gfiles',

    'django_tables2',
    'bootstrap3',
    'django_tables2_column_shifter',
    'django_sb_admin',
    'django_filter'
]
```

2. Include the polls URLconf in your project urls.py like this:

```
url(r'^', include('gfiles.urls')),
```

3. Run *python manage.py migrate* to create the models.

4. Start the development server and visit http://127.0.0.1:8000

5. Register http://127.0.0.1:8000/register/ and login http://127.0.0.1:8000/login/
6. Add files (need to be logged in) http://127.0.0.1:8000/upload_gfile/
7. View and filter files http://127.0.0.1:8000/gfile_summary/

8.1.2 API reference

django-gfiles models

class gfiles.models.**GenericFile**(**args*, ***kwargs*)
Model for managing generic files. Files can either be saved as symlink or copied to the data file store based on what has been set in the settings file MEDIA_ROOT parameter

exception DoesNotExist

exception MultipleObjectsReturned

django-gfiles views

class gfiles.views.**GFileCreateView**(***kwargs*)
Class to create a save a generic file using the GenericFile model.

Inherits the CreateView class and uses the LoginRequiredMixin

form_class
alias of gfiles.forms.GFileForm

form_valid(*form*)
If the form is valid, save the associated model.

model
alias of *gfiles.models.GenericFile*

class gfiles.views.**GFileListView**(***kwargs*)
Class to view a table and filter all of the currently saved GenericFiles

Inherits the FilterView class and uses the SingleTableMixin for viewing the django-tables2 table and uses the ExportMixin so that the table can be exported as a csv file

filterset_class
alias of *gfiles.filter.GFileFilter*

model
alias of *gfiles.models.GenericFile*

table_class
alias of *gfiles.tables.GFileTableWithCheck*

gfiles.views.**index**(*request*)
basic index view

gfiles.views.**status_update**(*request*)
Updates for tracking status of long processes via celery

gfiles.views.**success**(*request*)
basic success view

django-gfiles filters

class gfiles.filter.**GFileFilter**(*data=None*, *queryset=None*, *prefix=None*, *strict=None*, *request=None*)

Class for filtering Generic Files using django_filters.FilterSet

Filters available for the username of the user who uploaded the file and the original filename (the original name can be different to the name of the file that is stored if multiple files of the same name are uploaded at the same time)

django-gfiles tables

class gfiles.tables.**GFileTable**(**args*, ***kwargs*)

Class for django-tables2 table of `gfiles.models.GenericFiles`.

Inherits the ColumnShiftTable, that is a modified django-tables2 class that allows columns to be dynamically selected

class gfiles.tables.**GFileTableWithCheck**(**args*, ***kwargs*)

Class for django-tables2 table of `gfiles.models.GenericFiles` with a check box column

Inherits `gfiles.models.GFileTable` and just adds checkbox.

django-gfiles utils

gfiles.utils.save_as_symlink.**save_as_symlink**(*abs_pth*, *name*, *generic_file_obj*)

Takes a `gfiles.models.GenericFiles` object and saves a symlink of the file to the absolute path (abs_pth).

Parameters

- **`abs_pth`** – path to the file to symlink to
- **`name`** – Name to save to the file as
- **`generic_file_obj`** – `gfiles.models.GenericFiles` object

Returns updated `gfiles.models.GenericFiles` object with symlink

CHAPTER 9

About

9.1 Contacts

9.1.1 Tom Lawson

Computational metabolomics
Metabolomics and System Toxicology Laboratory
University of Birmingham, United Kingdom
tnl495@bham.ac.uk

9.1.2 Ralf Weber

Computational metabolomics
Phenome Centre Birmingham (PCB)
School of Bioscience, University of Birmingham, United Kingdom
r.j.weber@bham.ac.uk

9.1.3 Martin Jones

Computational metabolomics
Metabolomics and System Toxicology Laboratory
University of Birmingham, United Kingdom
m.r.jones.1@bham.ac.uk

9.1.4 Mark Viant

Metabolomics and System Toxicology Laboratory
Phenome Centre Birmingham (PCB)
School of Bioscience, University of Birmingham, United Kingdom
m.viant@bham.ac.uk

Python Module Index

g

m

Index

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

W

X