

---

# Python

Dec 02, 2019



---

## Contents

---

<b>1</b>	<b>Deployment</b>	<b>3</b>
1.1	User . . . . .	3
1.2	Developer . . . . .	4
<b>2</b>	<b>Usage</b>	<b>7</b>
<b>3</b>	<b>Use with external endpoint</b>	<b>9</b>
3.1	Build external endpoint abstraction . . . . .	9
3.2	Deploy AskOmics . . . . .	9
3.3	Integrate external endpoint . . . . .	10
<b>4</b>	<b>Contribute to AskOmics</b>	<b>11</b>
4.1	Issues . . . . .	11
4.2	Pull requests . . . . .	11
4.3	Tests . . . . .	12
4.4	Coding style guidelines . . . . .	12
4.5	Contribute to docs . . . . .	12
<b>5</b>	<b>askomics package</b>	<b>13</b>
5.1	Subpackages . . . . .	13
5.2	Submodules . . . . .	48
5.3	askomics.app module . . . . .	48
5.4	askomics.tasks module . . . . .	48
5.5	Module contents . . . . .	48
<b>6</b>	<b>Indices and tables</b>	<b>49</b>
	<b>Python Module Index</b>	<b>51</b>
	<b>Index</b>	<b>53</b>



AskOmics is a visual SPARQL query interface supporting both intuitive data integration and querying while shielding the user from most of the technical difficulties underlying RDF and SPARQL



## 1.1 User

### 1.1.1 Manual installation

#### Dependencies

##### Virtuoso

AskOmics work with an RDF triplestore.

Compile [virtuoso](#) or install it with [docker](#)

##### Python3/venv/npm

AskOmics is build in python3 and javascript. Install the following packages

```
# ubuntu
apt install -y python3 python3-venv npm
# Fedora
dnf install -y python3 python3-venv npm
```

#### Installation with scripts

Install AskOmics with `install.sh` and run it with `run_all.sh`. AskOmics will be available at [localhost:5000](#).

### 1.1.2 Installation with docker-compose

Clone the [askomics-docker-compose](#) repository

```
# clone
git clone https://github.com/askomics/flaskomics-docker-compose.git
# cd
cd flaskomics-docker-compose/standalone
# Or, if you want to use federated queries
cd flaskomics-docker-compose/federated
```

Update config (see [README](#))

Run

```
sudo docker-compose up -d
```

AskOmics will be available at [localhost](#).

### 1.1.3 Installation with a single docker

Docker image [askomics/flaskomics-with-dependencies](#) contain AskOmics with all his dependencies (Redis, virtuoso, celery ...).

```
# Pull image
docker pull askomics/flaskomics-with-dependencies
# run image
docker run -d askomics/flaskomics-with-dependencies
```

If you need a persistent volume, run

```
docker run -d -v ./flaskomics-data:/tmp/flaskomics askomics/flaskomics-with-
↳dependencies
```

The image create a default user at the first run. You can update this user by setting the following environment variables:

ENV	User field	default value
USER_FIRST_NAME	First name	Ad
USER_LAST_NAME	Last name	Min
USERNAME	Username	admin
USER_EMAIL	Email	admin@example.org
USER_PASSWORD	Password (clear)	admin
USER_APIKEY	User api key	admin
GALAXY_URL	Galaxy url	(optional)
GALAXY_API_KEY	Galaxy api key	(optional)

For example:

```
docker run -d -v ./flaskomics-data:/tmp/flaskomics -e USER_FIRST_NAME="John" -e USER_
↳LAST_NAME="Wick" -e USERNAME="jwick" askomics/flaskomics-with-dependencies
```

## 1.2 Developer

### 1.2.1 AskOmics

Fork the AskOmics repository

then, clone your fork

```
git clone https://github.com/USERNAME/flaskomics.git # replace USERNAME with your_
↳github username
```



### *Install AskOmics*

Run it with dev mod

```
./run_all.sh -d dev
```

AskOmics will be available at [localhost:5000](http://localhost:5000)

## 1.2.2 Dependencies

Clone the `askomics-docker-compose` repository and go into the `dev` directory.

```
# clone
git clone https://github.com/askomics/flaskomics-docker-compose.git
# cd
cd flaskomics-docker-compose/dev
```

The `dev` directory contain a `docker-compose.yml` file who can launch all dependencies needed by AskOmics (Virtuoso, Redis, Corese, Galaxy).

```
# clone
docker-compose up -d
```

Use the `ci` directory to launch dockers for testing AskOmics

```
cd ../ci
docker-compose up -d
```



## CHAPTER 2

---

### Usage

---

TODO: Write the docs ;-)



---

## Use with external endpoint

---

AskOmics can be use to explore external endpoint such as [NeXtProt](#).

### 3.1 Build external endpoint abstraction

First, build AskOmics abstraction of the external endpoint with [Abstractor](#).

Install Abstractor in a python virtual env

```
# Create and source venv
python3 -m venv venv
source venv/bin/activate
# Install Abstractor
pip install abstractor
```

Generate AskOmics abstraction. The tool explore the entire endpoint, so it can be long.

```
# Need help ?
abstractor -h
# Generate abstraction
abstractor -e <endpoint_url> -p <main_prefix> -o abstraction.ttl
```

### 3.2 Deploy AskOmics

Deploy AskOmics with a special configuration to use it with an external endpoint. The following ini entry have to be updated:

- askomics
  - disable\_integration: true
- triplestore

- `external_endpoint: <endpoint_url>`
- `prefix = <main_prefix>`
- `namespace = <main_prefix>`

### 3.3 Integrate external endpoint

Create an account on the AskOmics instance. The first account is an admin account who can integrate datasets. Load the `asbtraction.ttl` and integrate it has a public dataset.

Other users can't integrate datasets, but they can explore the external endpoint.

---

## Contribute to AskOmics

---

### 4.1 Issues

If you have an idea for a feature to add or an approach for a bugfix, it is best to communicate with developers early. The most common venues for this are [GitHub issues](#).

### 4.2 Pull requests

All changes to AskOmics should be made through pull requests to this repository.

For the [askomics repository](#) to your account. To keep your copy up to date, you need to frequently [sync your fork](#):

```
git remote add upstream https://github.com/askomics/flaskomics
git fetch upstream
git checkout master
git merge upstream/master
```

Then, create a new branch for your new feature

```
git checkout -b my_new_feature
```

Commit and push your modification to your [fork](#). If your changes modify code, please ensure that is conform to *AskOmics style*

Write tests for your changes, and make sure that they *passes*.

Open a pull request against the master branch of askomics. The message of your pull request should describe your modifications (why and how).

The pull request should pass all the continuous integration tests which are automatically run by Github using Travis CI. The coverage must be at least remain the same (but it's better if it increases)

## 4.3 Tests

Use `test.sh` script to lint and test the code. Don't PR if linting or testing don't pass.

## 4.4 Coding style guidelines

### 4.4.1 General

Ensure all user-enterable strings are unicode capable. Use only English language for everything (code, documentation, logs, comments, ...)

### 4.4.2 Python

We follow [PEP-8](#), with particular emphasis on the parts about knowing when to be inconsistent, and readability being the ultimate goal.

- Whitespace around operators and inside parentheses
- 4 spaces per indent, spaces, not tabs
- Include docstrings on your modules, class and methods
- Avoid from module import \*. It can cause name collisions that are tedious to track down.
- Class should be in `CamelCase`, methods and variables in `lowercase_with_underscore`

### 4.4.3 Javascript

We follow [W3 JavaScript Style Guide and Coding Conventions](#)

## 4.5 Contribute to docs

all the documentation (including what you are reading) can be found [here](#). Files are on the [AskOmics repository](#).

To preview the docs, run

```
cd flaskomics
# source the askomics virtual env
source venv/bin/activate
cd docs
make html
```

html files are in `build` directory.



## 5.1 Subpackages

### 5.1.1 askomics.api package

#### Submodules

#### askomics.api.admin module

Admin routes

`askomics.api.admin.get_users()`  
Get all users

**Returns** users: list of all users info error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.admin.set_admin()`  
change admin status of a user

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.admin.set_blocked()`  
Change blocked status of a user

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.admin.set_quota()`  
Change quota of a user

**Returns** users: updated users error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

## askomics.api.auth module

Authentication routes

`askomics.api.auth.admin_required(f)`  
Login required function

`askomics.api.auth.login()`  
Log a user

**Returns** Information about the logged user

**Return type** json

`askomics.api.auth.login_api_key(key)`  
Log user with his API key

**Parameters** `key` (*string*) – User API key

**Returns** Information about the logged user

**Return type** json

`askomics.api.auth.login_required(f)`  
Login required function

`askomics.api.auth.logout()`  
Logout the current user

**Returns** no username and logged false

**Return type** json

`askomics.api.auth.signup()`  
Register a new user

**Returns** Info about the user

**Return type** json

`askomics.api.auth.update_apikey()`  
Update the user apikey

**Returns** The user with his new apikey

**Return type** json

`askomics.api.auth.update_galaxy()`  
Update the user apikey

**Returns** The user with his new apikey

**Return type** json

`askomics.api.auth.update_password()`  
Update the user password

**Returns** The user

**Return type** json

---

```
askomics.api.auth.update_profile()
```

Update user profile (names and email)

**Returns** The updated user

**Return type** json

### askomics.api.catch\_url module

Catch\_all route

```
askomics.api.catch_url.catch_all(path)
```

Return all routes to home

**Parameters** *path* (*str*) – Original path

**Returns** Redirect to route /

**Return type** redirect

### askomics.api.datasets module

Api routes

```
askomics.api.datasets.delete_datasets()
```

Delete some datasets (db and triplestore) with a celery task

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

```
askomics.api.datasets.get_datasets()
```

Get datasets information

**Returns** datasets: list of all datasets of current user error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

```
askomics.api.datasets.toggle_public()
```

Toggle public status of a dataset

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

### askomics.api.file module

Api routes

```
askomics.api.file.delete_files()
```

Delete files

**Returns** files: list of all files of current user error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.file.get_files()`

Get files info of the logged user

**Returns** files: list of all files of current user error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.file.get_preview()`

Get files preview

**Returns** previewFiles: preview of selected files error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.file.integrate()`

Integrate a file

**Returns** task\_id: celery task id error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.file.serve_file(path, user_id, username)`

Serve a static ttl file of a user

**Parameters**

- **path** (*string*) – The file path to serve
- **user\_id** (*int*) – user id
- **username** (*string*) – username

**Returns** the file

**Return type** file

`askomics.api.file.upload_chunk()`

Upload a file chunk

**Returns** path: name of the local file. To append the next chunk into it error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.file.upload_url()`

Upload a distant file with an URL

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

## askomics.api.galaxy module

Api routes

`askomics.api.galaxy.get_dataset_content()`

Download a galaxy datasets into AskOmics

**Returns** dataset\_content: content of the requested Galaxy dataset error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.galaxy.get_datasets()`

Get galaxy datasets and histories of a user

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.galaxy.get_queries()`

Get galaxy queries (json datasets)

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.galaxy.upload_datasets()`

Download a galaxy datasets into AskOmics

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

## askomics.api.query module

`askomics.api.query.get_abstraction()`

Get abstraction

**Returns** abstraction: abstraction error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.query.get_preview()`

Get a preview of query

**Returns** resultsPreview: Preview of the query results headerPreview: Header of the results table error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.query.query()`

Get start points

**Returns** startpoint: list of start points error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.query.save_result()`

Save a query in filesystem and db, using a celery task

**Returns** task\_id: celery task id error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

**askomics.api.results module**

`askomics.api.results.delete_result()`

Summary

**Returns** files: list of all files of current user error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.results.download_result()`

Download result file

`askomics.api.results.get_graph_state()`

Summary

**Returns** preview: list of result preview header: result header error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.results.get_preview()`

Summary

**Returns** preview: list of result preview header: result header error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.results.get_results()`

Get ...

**Returns** files: list of all files of current user error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.results.get_sparql_query()`

Get sparql query of result for the query editor

**Returns** query: the sparql query error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.results.publish_query()`

Publish a query template from a result

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.results.send2galaxy()`

Send a result file into Galaxy

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.results.set_description()`

Update a result description

**Returns** files: all files error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

`askomics.api.results.set_public()`

Change public status of a file, and return all files

**Returns** files: list of all files of current user error: True if error, else False errorMessage: the error message of error, else an empty string

**Return type** json

## askomics.api.sparql module

`askomics.api.sparql.prefix()`

Get the default sparql query

**Returns** default query

**Return type** json

`askomics.api.sparql.query()`

Perform a sparql query

**Returns** query results

**Return type** json

`askomics.api.sparql.save_query()`

Perform a sparql query

**Returns** query results

**Return type** json

## askomics.api.start module

`askomics.api.start.hello()`

Dummy routes

**Returns** error: True if error, else False errorMessage: the error message of error, else an empty string message: a welcome message

**Return type** json

`askomics.api.start.start()`

Starting route

**Returns** Information about a eventually logged user, and the AskOmics version and a footer message

**Return type** json

## askomics.api.view module

Render route

`askomics.api.view.home(path)`

Render the html of AskOmics

**Returns** Html code of AskOmics

**Return type** html

## Module contents

### 5.1.2 askomics.libaskomics package

#### Submodules

#### askomics.libaskomics.BedFile module

**class** askomics.libaskomics.BedFile.**BedFile** (*app, session, file\_info, host\_url=None, external\_endpoint=None, custom\_uri=None*)

Bases: *askomics.libaskomics.File.File*

Bed File

**public**

Public or private dataset

**Type** bool

**generate\_rdf\_content** ()

Generate RDF content of the BED file

**Yields** *Graph* – RDF content

**get\_preview** ()

Get file preview

**Returns** bed file preview

**Return type** dict

**integrate** (*entity\_name, public=True*)

Integrate BeD file

**Parameters**

- **entities** (*List*) – Entities to integrate
- **public** (*bool, optional*) – Insert in public dataset

**set\_preview** ()

Set entity name preview

**set\_rdf\_abstraction\_domain\_knowledge** ()

Set the abstraction and domain knowledge

#### askomics.libaskomics.CsvFile module

**class** askomics.libaskomics.CsvFile.**CsvFile** (*app, session, file\_info, host\_url=None, external\_endpoint=None, custom\_uri=None*)

Bases: *askomics.libaskomics.File.File*

CSV file

**category\_values**

Category values

**Type** dict



**columns\_type**  
Columns type  
**Type** list

**header**  
Header  
**Type** list

**preview**  
Previex  
**Type** list

**public**  
Public  
**Type** bool

**check\_columns\_types()**  
Check all columns type after detection and correct them

**dialect**  
Like @property on a member function, but also cache the calculation in self.\_\_dict\_\_[function name]. The function is called only once since the cache stored as an instance attribute override the property residing in the class attributes. Following accesses cost no more than standard Python attribute access. If the instance attribute is deleted the next access will re-evaluate the function. Source: <https://blog.ionelmc.ro/2014/11/04/an-interesting-python-descriptor-quirk/>

```
class Shape(object):
    @cached_property def area(self):
        # compute value return value
```

**func**  
Description  
**Type** TYPE

**force\_columns\_type(forced\_columns\_type)**  
Set the columns type without detecting them  
**Parameters** **forced\_columns\_type** (*list*) – columns type

**generate\_rdf\_content()**  
Generator of the rdf content  
**Yields** *Graph* – Rdf content

**get\_preview()**  
Get a preview of the file  
**Returns** File preview  
**Return type** dict

**guess\_column\_type(values, header\_index)**  
Guess the columns type  
**Parameters**

- **values** (*list*) – columns preview
- **header\_index** (*int*) – Header index

**Returns** The guessed type

**Return type** string

**integrate** (*forced\_columns\_type*, *public=False*)

Integrate the file

**Parameters**

- **forced\_columns\_type** (*list*) – columns type
- **public** (*bool*, *optional*) – True if dataset will be public

**is\_category** (*values*)

Check if a list of values are categories

**Parameters** **values** (*list*) – List of values

**Returns** True if values are categories

**Return type** bool

**static is\_decimal** (*value*)

Guess if a variable is a number

**Parameters** **value** – The var to test

**Returns** True if it's decimal

**Return type** boolean

**set\_columns\_type** ()

Set the columns type by guessing them

**set\_preview** ()

Set preview, header and columns type by sniffing the file

**set\_preview\_and\_header** ()

Set the preview and header by looking in the first lines of the file

**set\_rdf\_abstraction** ()

Set the abstraction

**set\_rdf\_abstraction\_domain\_knowledge** ()

Set intersection of abstraction and domain knowledge

**set\_rdf\_domain\_knowledge** ()

Set the domain knowledge

**transposed\_preview**

Transpose the preview

**Returns** Transposed preview

**Return type** list

## askomics.libaskomics.Database module

Contain the Database class

**class** askomics.libaskomics.Database.Database (*app*, *session*)

Bases: *askomics.libaskomics.Params.Params*

Manage Database connection

**database\_path**  
Path to the database file  
**Type** str

**create\_datasets\_table()**  
Create the datasets table

**create\_endpoints\_table()**  
Create the endpoints table

**create\_files\_table()**  
Create the files table

**create\_galaxy\_table()**  
Create the galaxy table

**create\_integration\_table()**  
Create the integration table

**create\_results\_table()**  
Create the results table

**create\_user\_table()**  
Create the user table

**execute\_sql\_query**(*query*, *variables=[]*, *get\_id=False*)  
Execute an sql query to the database

**Parameters**

- **query** (*str*) – The sql query
- **variables** (*List*, *optional*) – Sql variables
- **get\_id** (*bool*, *optional*) – Return the last row id

**Returns** Result of the query, or last row id

**Return type** List

**init\_database()**  
Create all tables

**update\_datasets\_table()**  
Add cols on the datasets table

**update\_results\_table()**  
Add the size and sparql\_query cols on the results table

**update\_users\_table()**  
Add the quota col on the users table  
Update the users table for the instance who don't have this column

### askomics.libaskomics.Dataset module

**class** askomics.libaskomics.Dataset.**Dataset**(*app*, *session*, *dataset\_info={}*)  
Bases: *askomics.libaskomics.Params.Params*

**celery\_id**  
celery id  
**Type** string

**file\_id**  
database file id  
**Type** int

**graph\_name**  
graph name  
**Type** string

**id**  
database dataset id  
**Type** int

**name**  
dataset name  
**Type** string

**public**  
Public  
**Type** bool

**delete\_from\_db()**  
Delete a dataset from the database

**save\_in\_db()**  
Save the dataset into the database

**set\_info\_from\_db()**  
Set the info in from the database

**toggle\_public(new\_status)**  
Change public status of a dataset (triplestore and db)  
**Parameters** **new\_status** (*bool*) – True if public

**update\_celery(celery\_id)**  
Update celery id of dataset in database  
**Parameters** **celery\_id** (*string*) – Describe the celery id option

**update\_in\_db(status, update\_celery=False, error=False, error\_message=None, ntriples=0, traceback=None)**  
Update the dataset when integration is done  
**Parameters**

- **error** (*bool, optional*) – True if error during integration
- **error\_message** (*None, optional*) – Error string if error is True
- **ntriples** (*int, optional*) – Number of triples integrated

### askomics.libaskomics.DatasetsHandler module

**class** askomics.libaskomics.DatasetsHandler.**DatasetsHandler** (*app, session, datasets\_info=[]*)  
Bases: *askomics.libaskomics.Params.Params*  
Summary

**datasets**  
Description  
**Type** list

**datasets\_info**  
Description  
**Type** TYPE

**delete\_datasets()**  
delete the datasets from the database and the triplestore

**delete\_datasets\_in\_db()**  
Delete datasets of the database

**get\_datasets()**  
Get info about the datasets  
**Returns** Datasets informations  
**Return type** list of dict

**handle\_datasets()**  
Handle datasets

**update\_status\_in\_db(status)**  
Update the status of a datasets in the database  
**Parameters** **status** (*string*) – The new status (started, success or deleting)  
**Returns** Remaining datasets  
**Return type** list

### askomics.libaskomics.File module

**class** askomics.libaskomics.File.**File**(*app, session, file\_info, host\_url=None, external\_endpoint=None, custom\_uri=None*)  
Bases: *askomics.libaskomics.Params.Params*

Summary

**askomics\_namespace**  
AskOmics namespace askomics:  
**Type** Namespace

**askomics\_prefix**  
AskOmics prefix :  
**Type** Namespace

**dc**  
dc namespace  
**Type** Namespace

**default\_graph**  
Default rdf graph  
**Type** string

**faldo**  
faldo namespace

**Type** Namespace

**faldo\_entity**  
True if entity is a faldo entity

**Type** bool

**file\_graph**  
File graph containing the file

**Type** string

**host\_url**  
AskOmics url

**Type** string

**id**  
database file id

**Type** int

**max\_chunk\_size**  
Max number of triple to insert in one Load or insert

**Type** int

**method**  
Load or insert

**Type** int

**name**  
Name of the file

**Type** string

**now**  
timestamp of the current time

**Type** datetime

**ntriples**  
Number of triples

**Type** int

**path**  
Path of the file

**Type** string

**prov**  
prov namespace

**Type** Namespace

**public**  
True if the file is public

**Type** bool

**size**  
file size

**Type** int

**timestamp**  
Description

**Type** TYPE

**ttl\_dir**  
path to the ttl directory

**Type** string

**type**  
file type

**Type** string

**user\_graph**  
User graph

**Type** string

**convert\_type** (*value*)  
Convert a value to a int or float or text

**Parameters** **value** (*string*) – The value to convert

**Returns** the converted value

**Return type** string/float/int

**format\_uri** (*string, remove\_space=False*)  
remove space and quote

**get\_faldo\_strand** (*raw\_strand*)  
Get faldo strand

**Parameters** **raw\_strand** (*string*) – raw value of strand

**Returns** Faldo “Foward”, “Reverse” or “Both” uri

**Return type** rdf term

**get\_metadata** ()  
Get a rdflib graph of the metadata

**Returns** graph containing metadata of the file

**Return type** Graph

**get\_rdf\_type** (*value*)  
get xsd type of a value

**Parameters** **value** – The value to get type

**Returns** rdflib.XSD.string or rdflib.XSD.decimal

**Return type** TYPE

**integrate** ()  
Integrate the file into the triplestore

**load\_graph** (*rdf\_graph, tmp\_file\_name*)  
Load a rdflib graph into the triplestore

Write rdf to a tmp file, and send the url to this file to the triplestore with a LOAD request

**Parameters**

- **rdf\_graph** (*Graph*) – rdf graph to load

- **tmp\_file\_name** (*string*) – Path to a tmp file

**rdsize** (*string*)  
Rdfize a string  
Return the literal is string is an url, else, prefix it with askomics prefix

**Parameters** **string** (*string*) – Term to rdfize

**Returns** Rdfized term

**Return type** rdflib.???

**rollback** ()  
Drop the dataset from the triplestore in case of error

**set\_triples\_number** ()  
Set graph triples number by requesting the triplestore

### askomics.libaskomics.FilesHandler module

**class** askomics.libaskomics.FilesHandler.**FilesHandler** (*app, session, host\_url=None, external\_endpoint=None, custom\_uri=None*)

Bases: *askomics.libaskomics.FilesUtils.FilesUtils*

Handle files

**files**  
list of File

**Type** list

**host\_url**  
AskOmics url, for the triplestore

**Type** string

**upload\_path**  
Upload path

**Type** string

**delete\_file\_from\_db** (*file\_id*)  
remove a file for the database

**Parameters** **file\_id** (*int*) – the file id to remove

**delete\_file\_from\_fs** (*file\_path*)  
Delete a file from filesystem

**Parameters** **file\_path** (*string*) – Path to the file

**delete\_files** (*files\_id*)  
Delete files from database and filesystem

**Parameters** **files\_id** (*list*) – list of file id

**Returns** list of files info

**Return type** list

**download\_url** (*url*)  
Download a file from an URL and insert info in database



**Parameters** `url` (*string*) – The file url

**get\_file\_name** ()  
Get a random file name

**Returns** file name

**Return type** string

**get\_file\_path** (*file\_id*)  
Get the file path with id

**Parameters** `file_id` (*int*) – the file id

**Returns** file path

**Return type** string

**get\_files\_infos** (*files\_id=None, return\_path=False*)  
Get files info

**Parameters**

- `files_id` (*None, optional*) – list of files id
- `return_path` (*bool, optional*) – return the path if True

**Returns** list of files info

**Return type** list

**get\_type** (*file\_ext*)  
Get files type, based on extension

TODO: sniff file to get type

**Parameters** `file_ext` (*string*) – file extension

**Returns** file type

**Return type** string

**handle\_files** (*files\_id*)  
Handle file

**Parameters** `files_id` (*list*) – id of files to handle

**persist\_chunk** (*chunk\_info*)  
Persist a file by chunk. Store info in db if the chunk is the last

**Parameters** `chunk_info` (*dict*) – Info about the chunk

**Returns** local filename

**Return type** str

**store\_file\_info\_in\_db** (*name, filetype, file\_name, size*)  
Store the file info in the database

**Parameters**

- `name` (*string*) – Name of the file
- `filetype` (*string*) – Type (csv ...)
- `file_name` (*string*) – Local file name
- `size` (*string*) – Size of file

**write\_data\_into\_file** (*data*, *file\_name*, *mode*)

Write data into a file

**Parameters**

- **data** (*string*) – data to write
- **file\_name** (*string*) – Local file name
- **mode** (*string*) – open mode (w or a)

## askomics.libaskomics.FilesUtils module

**class** askomics.libaskomics.FilesUtils.**FilesUtils** (*app*, *session*)

Bases: *askomics.libaskomics.Params.Params*

Contain methods usefull in FilesHandler and ResultsdHandler

**get\_size\_occupied\_by\_user** ()

Get disk size occupied by file user (uploaded files and results)

**Returns** size un bytes

**Return type** int

## askomics.libaskomics.Galaxy module

**class** askomics.libaskomics.Galaxy.**Galaxy** (*app*, *session*, *url=None*, *apikey=None*)

Bases: *askomics.libaskomics.Params.Params*

Connection with a Galaxy account

**apikey**

Galaxy API key

**Type** string

**url**

Galaxy url

**Type** string

**check\_galaxy\_instance** ()

Check the Galaxy credentials

**Returns** True if URL and Key exists

**Return type** Boolean

**download\_datasets** (*datasets\_id*)

Download galaxy datasets into AskOmics

**Parameters** **datasets\_id** (*list*) – List of Galaxy datasets id

**get\_dataset\_content** (*dataset\_id*)

Get Galaxy dataset content

**Parameters** **dataset\_id** (*string*) – dataset ID

**Returns** Content of the dataset

**Return type** string

**get\_datasets\_and\_histories** (*history\_id=None, query=False*)

Get Galaxy datasets of the current history and all histories

**Parameters**

- **history\_id** (*int, optional*) – A history id
- **query** (*bool, optional*) – Get Datasets, or json datasets for query

**Returns** Datasets and histories

**Return type** dict

## askomics.libaskomics.GffFile module

**class** askomics.libaskomics.GffFile.**GffFile** (*app, session, file\_info, host\_url=None, external\_endpoint=None, custom\_uri=None*)

Bases: *askomics.libaskomics.File.File*

GFF File

**public**

Public or private dataset

**Type** bool

**generate\_rdf\_content** ()

Generator of the rdf content

**Yields** *Graph* – Rdf content

**get\_preview** ()

Get gff file preview (list of entities)

**Returns** Return info about the file

**Return type** dict

**integrate** (*entities, public=True*)

Integrate GFF file

**Parameters**

- **entities** (*List*) – Entities to integrate
- **public** (*bool, optional*) – Insert in public dataset

**set\_preview** ()

Summary

**set\_rdf\_abstraction\_domain\_knowledge** ()

Set the abstraction and domain knowledge

## askomics.libaskomics.LocalAuth module

Contain the Database class

**class** askomics.libaskomics.LocalAuth.**LocalAuth** (*app, session*)

Bases: *askomics.libaskomics.Params.Params*

Manage user authentication

**add\_galaxy\_account** (*user, url, apikey*)

Add a Galaxy account

**Parameters**

- **user** (*dict*) – Previous user info
- **url** (*string*) – Galaxy URL
- **apikey** (*string*) – Galaxy API key

**Returns** Updated user

**Return type** dict

**authenticate\_user** (*inputs*)

check if the password is the good password associate with the email

**Parameters** **inputs** (*dict*) – login and password

**Returns** user info if authentication success

**Return type** dict

**authenticate\_user\_with\_apikey** (*apikey*)

Return the user associated with the API key

**Parameters** **inputs** (*string*) – API key

**Returns** user info if authentication success

**Return type** dict

**check\_inputs** (*inputs*)

Check user inputs

Check if inputs are not empty, if passwords are identical, and if username and email are not already in the database

**Parameters** **inputs** (*dict*) – User inputs

**create\_directory** (*directory\_path*)

Create a directory

**Parameters** **directory\_path** (*string*) – Path

**create\_user\_directories** (*user\_id, username*)

Create the User directory

**Parameters**

- **user\_id** (*int*) – User id
- **username** (*string*) – username

**get\_all\_users** ()

Get all user info

**Returns** All user info

**Return type** list

**get\_number\_of\_users** ()

get the number of users in the DB

**Returns** Number of user in the Database

**Return type** int

**get\_user** (*username*)

Get a specific user by his username

**Parameters** **username** (*string*) – User username

**Returns** The corresponding user

**Return type** dict

**is\_email\_in\_db** (*email*)

Check if the email is present in the database

**Parameters** **email** (*str*) – Email

**Returns** True if the email exist

**Return type** bool

**is\_username\_in\_db** (*username*)

Check if the username is present in the database

**Parameters** **username** (*str*) – Username

**Returns** True if the user exist

**Return type** bool

**persist\_user** (*inputs, ldap=False*)

Persist user in the TS

**Parameters**

- **inputs** (*dict*) – User infos
- **ldap** (*bool, optional*) – If True, user is ldap

**Returns** The user

**Return type** dict

**set\_admin** (*new\_status, username*)

Set a new admin status to a user

**Parameters**

- **new\_status** (*boolean*) – True for an admin
- **username** (*string*) – The concerned username

**set\_blocked** (*new\_status, username*)

Set a new blocked status to a user

**Parameters**

- **new\_status** (*boolean*) – True for blocked
- **username** (*string*) – The concerned username

**set\_quota** (*quota, username*)

Set a new quota to a user

**Parameters**

- **quota** (*int*) – New quota
- **username** (*string*) – The concerned username

**update\_apikey** (*user*)

Create a new api key and store in the database

**Parameters** **user** (*dict*) – The current user

**Returns** error, error message and updated user

**Return type** dict

**update\_galaxy\_account** (*user, url, apikey*)

Update a Galaxy account

**Parameters**

- **user** (*dict*) – Previous user info
- **url** (*string*) – Galaxy URL
- **apikey** (*string*) – Galaxy API key

**Returns** Updated user

**Return type** dict

**update\_password** (*inputs, user*)

Update the password of a user

**Parameters**

- **inputs** (*dict*) – Current password and the new one (and confirmation)
- **user** (*dict*) – The current user

**Returns** error, error message and updated user

**Return type** dict

**update\_profile** (*inputs, user*)

Update the profile of a user

**Parameters**

- **inputs** (*dict*) – fields to update
- **user** (*dict*) – The current user

**Returns** error, error message and updated user

**Return type** dict

## askomics.libaskomics.Params module

Contain the Params class

**class** askomics.libaskomics.Params.**Params** (*app, session*)

Bases: object

Mother of all libaskomics classes

**app**

flask app

**log**

flask logger

**session**

flask session

**settings**

askomics settings (from ini)

**get\_error** ()

**get\_error\_message()**

**logged\_user()**  
Check if a user is logged

**Returns** True if a user is logged

**Return type** bool

**str\_to\_bool(bool\_str)**  
Convert a true/false string to a boolean value

**Parameters** **bool\_str** (*str*) – boolean string

**Returns** True or False

**Return type** bool

### askomics.libaskomics.PrefixManager module

**class** askomics.libaskomics.PrefixManager.**PrefixManager** (*app, session*)  
Bases: *askomics.libaskomics.Params.Params*

Manage sparql prefixes

**askomics\_namespace**  
askomics namespace, from config file

**Type** str

**askomics\_prefix**  
askomics prefix, from config file

**Type** str

**prefix**  
dict of all prefixes

**Type** dict

**get\_prefix()**  
Get all prefixes

**Returns** prefixes

**Return type** str

### askomics.libaskomics.RdfFile module

**class** askomics.libaskomics.RdfFile.**RdfFile** (*app, session, file\_info, host\_url=None, external\_endpoint=None, custom\_uri=None*)  
Bases: *askomics.libaskomics.File.File*

RDF (turtle) File

**public**  
Public or private dataset

**Type** bool

**get\_preview()**  
Get a preview of the frist 100 lines of a ttl file

**Returns** Description

**Return type** TYPE

**integrate** (*public=False*)

Integrate the file into the triplestore

**Parameters** **public** (*bool, optional*) – Integrate in private or public graph

**set\_preview** ()

Summary

## askomics.libaskomics.RdfGraph module

**class** askomics.libaskomics.RdfGraph.**RdfGraph** (*app, session*)

Bases: *askomics.libaskomics.Params.Params*

rdflib.graph wrapper

**askomics\_namespace**

AskOmics namespace

**Type** Namespace

**askomics\_prefix**

AskOmics prefix

**Type** Namespace

**graph**

rdflib graph

**Type** Graph

**ntriple**

Number of triple in the graph

**Type** int

**add** (*triple*)

Add a triple into the rdf graph

**Parameters** **triple** (*tuple*) – triple to add

**bind** (*a, b*)

Bind a namespace

**Parameters**

- **a** (*string*) – prefix
- **b** (*string*) – namespace

**get\_triple** ()

Get all triple

**merge** (*other\_graph*)

Merge a graph into this graph

**Parameters** **other\_graph** (*RdfGraph*) – The graph to merge

**serialize** (*destination=None, format='xml', base=None, encoding=None, \*\*args*)

Serialize the graph into a file

**Parameters**

- **format** (*string*) – rdf syntaxe



- **encoding** (*string*) – Encoding
- **destination** (*string*) – File destination

### askomics.libaskomics.Result module

**class** askomics.libaskomics.Result.**Result** (*app, session, result\_info, force\_no\_db=False*)

Bases: *askomics.libaskomics.Params.Params*

Result represent a query result file

**celery\_id**

Celery job id

**Type** str

**file\_name**

file name

**Type** str

**file\_path**

file path

**Type** str

**graph\_state**

The json query graph state

**Type** dict

**id**

database id

**Type** int

**result\_path**

results directory path

**Type** str

**clean\_link** (*link*)

Clean a link by removing coordinates and other stuff

**Parameters** **link** (*dict*) – A graph link

**Returns** Cleaned link

**Return type** dict

**clean\_node** (*node*)

Clean a node by removing coordinates and other stuff

**Parameters** **node** (*dict*) – A graph node

**Returns** Cleaned node

**Return type** dict

**delete\_db\_entry** ()

Delete results from db

**delete\_file\_from\_filesystem** ()

Remove result file from filesystem

**delete\_result()**  
Remove results from db and filesystem

**format\_graph\_state**(*d3\_graph\_state*)  
Format Graph state  
Remove coordinates and other things  
**Parameters** *d3\_graph\_state* (*dict*) – The d3 graph state  
**Returns** formatted graph state  
**Return type** dict

**get\_dir\_path()**  
Get directory path  
**Returns** directory path  
**Return type** str

**get\_file\_name()**  
Get file name  
**Returns** file name  
**Return type** str

**get\_file\_preview()**  
Get a preview of the results file  
**Returns** headers and preview  
**Return type** list, list

**get\_graph\_state**(*formatted=False*)  
Get get\_graph\_state  
**Returns** graph state  
**Return type** dict

**get\_sparql\_query()**  
Get the sparql query if exists  
**Returns** The sparql query  
**Return type** string

**publish\_query**(*public*)  
Insert query id and desc in the published\_query table

**rollback()**  
Delete file

**save\_in\_db()**  
Save results file info into the database

**save\_result\_in\_file**(*headers, results*)  
Save query results in a csv file  
**Parameters**

- **headers** (*list*) – List of results headers
- **results** (*list*) – Query results

**Returns** File size

**Return type** int

**send2galaxy** (*file2send*)

Send files to Galaxy

**send\_query\_to\_galaxy** ()

Send the json query to a galaxy dataset

**send\_result\_to\_galaxy** ()

Send a result file to Galaxy

**set\_celery\_id** (*celery\_id*)

Set celery id

**Parameters** **celery\_id** (*string*) – The celery id

**set\_info\_from\_db\_with\_id** ()

Set result info from the db

**update\_celery** (*celery\_id*)

Update celery id of result in database

**Parameters** **celery\_id** (*string*) – Describe the celery id option

**update\_db\_status** (*status*, *size=None*, *update\_celery=False*, *error=False*, *error\_message=None*, *traceback=None*)

Update status of results in db

**Parameters**

- **error** (*bool*, *optional*) – True if error during integration
- **error\_message** (*bool*, *optional*) – Error string if error is True

**update\_description** (*description*)

Change the result description

**update\_public\_status** (*public*)

Change public status

**Parameters** **public** (*bool*) – New public status

## askomics.libaskomics.ResultsHandler module

**class** askomics.libaskomics.ResultsHandler.**ResultsHandler** (*app*, *session*)

Bases: *askomics.libaskomics.Params.Params*

Handle results

**delete\_results** (*files\_id*)

Delete files

**Parameters** **files\_id** (*list*) – list of file id to delete

**Returns** list of remaining files

**Return type** list

**get\_files\_info** ()

Get files info of the user

**Returns** list of file info

**Return type** list

**get\_public\_queries()**

Get id and description of published queries

**Returns** List of published queries (id and description)

**Return type** List

### askomics.libaskomics.SparqlQueryBuilder module

**class** askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder(*app, session*)

Bases: *askomics.libaskomics.Params.Params*

Format a sparql query

**private\_graphs**

all user private graph

**Type** list

**public\_graphs**

all public graph

**Type** list

**build\_query\_from\_json**(*json\_query, preview=False, for\_editor=False*)

Build a sparql query for the json dict of the query builder

**Parameters** **json\_query** (*dict*) – The json query from the query builder

**Returns** SPARQL query

**Return type** str

**format\_endpoint\_name**(*endpoint*)

Replace local url by “local triplestore”

**Parameters** **endpoint** (*string*) – The endpoint name

**Returns** Formated endpoint name

**Return type** string

**format\_graph\_name**(*graph*)

Format graph name by removing base graph and timestamp

**Parameters** **graph** (*string*) – The graph name

**Returns** Formated graph name

**Return type** string

**format\_query**(*query, limit=30, replace\_froms=True, federated=False*)

Format the Sparql query

- remove all FROM
- add FROM <graph> (public graph and user graph)
- set a limit if not (or if its to big)

**Parameters**

- **query** (*string*) – sparql query to format
- **limit** (*int, optional*) – Description

**Returns** formatted sparql query

**Return type** string

**format\_sparql\_variable** (*name*)

Format a name into a sparql variable by remove spacial char and add a ?

**Parameters** **name** (*string*) – name to convert

**Returns** The corresponding sparql variable

**Return type** string

**get\_checked\_asked\_graphs** (*asked\_graphs*)

Check if asked graphs are present in public and private graphs

**Parameters** **asked\_graphs** (*list*) – list of graphs asked by the user

**Returns** list of graphs asked by the user, in the public and private graphs

**Return type** list

**get\_default\_query** ()

Get the default query

**Returns** the default query

**Return type** str

**get\_default\_query\_with\_prefix** ()

Get default query with the prefixes

**Returns** default query with prefixes

**Return type** str

**get\_endpoints\_string** ()

get endpoint strngs for the federated query engine

**Returns** the endpoint string

**Return type** string

**get\_federated\_froms** ()

Get @from string fir the federated query engine

**Returns** The from string

**Return type** string

**get\_federated\_froms\_from\_graphs** (*graphs*)

Get @from string fir the federated query engine

**Returns** The from string

**Return type** string

**get\_federated\_line** ()

Get federtated line

**Returns** @federate <endpoint1> <endpoint1> ...

**Return type** string

**get\_froms** ()

Get FROM string

**Returns** FROM string

**Return type** string

**get\_froms\_from\_graphs** (*graphs*)

Get FROM's form a list of graphs

**Parameters** **graphs** (*list*) – List of graphs

**Returns** from string

**Return type** str

**get\_graphs\_and\_endpoints** (*selected\_graphs=None, selected\_endpoints=None*)

get graphs and endpoints (uri and names)

**Returns** List of dict uri name

**Return type** list

**is\_bnode** (*uri, entities*)

Check if a node uri is a blank node

**Parameters**

- **uri** (*string*) – node uri
- **entities** (*list*) – all the entities

**Returns** True if uri correspond to a blank node

**Return type** Bool

**is\_federated** ()

Return True if there is more than 1 endpoint

**Returns** True or False

**Return type** bool

**prefix\_query** (*query*)

Add prefix and dedent a sparql query string

**Parameters** **query** (*string*) – The sparql query

**Returns** Formatted query

**Return type** string

**replace\_froms** ()

True if not federated and endpoint is local

**Returns** True or False

**Return type** bool

**set\_endpoints** (*endpoints*)

Set endpoints

**Parameters** **endpoints** (*list*) – Endpoints

**set\_graphs** (*graphs*)

Set graphs

**Parameters** **graphs** (*list*) – graphs

**set\_graphs\_and\_endpoints** (*entities=None, graphs=None, endpoints=None*)

Get all public and private graphs containing the given entities

**Parameters** **entities** (*list, optional*) – list of entity uri

**toggle\_public** (*graph*, *public*)

Change public status of data into the triplestore

**Parameters**

- **graph** (*string*) – Graph to update public status
- **public** (*string*) – true or false (string)

**triple\_dict\_to\_string** (*triple\_dict*)

Convert a triple dict into a triple string

**Parameters** **triple\_dict** (*dict*) – The triple dict

**Returns** The triple string

**Return type** string

## askomics.libaskomics.SparqlQueryLauncher module

```
class askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher (app,
                                                                    session,
                                                                    get_result_query=False,
                                                                    federated=False,
                                                                    end-points=None)
```

Bases: *askomics.libaskomics.Params.Params*

**endpoint**

The triplestore endpoint

**Type** SPARQLWrapper

**query\_time**

Query execution time

**Type** time

**triplestore**

triplestore (virtuoso, fuseki ...)

**Type** string

**drop\_dataset** (*graph*)

Drop the datasets of the triplestore and its metadata

**Parameters** **graph** (*string*) – graph name to remove

**execute\_query** (*query*)

Execute a sparql query

**Parameters** **query** (*string*) – Query to perform

**Returns** result

**Return type** TYPE

**get\_triples\_from\_graph** (*graph*)

Get triples from a rdflib graph

**Parameters** **graph** (*Graph*) – rdf graph

**Returns** ttl string

**Return type** string

**insert\_data** (*ttl*, *graph*, *metadata=False*)

Insert data into the triplesotre using INSERT

**Parameters**

- **ttl** (*Graph*) – rdflib graph
- **graph** (*string*) – graph name
- **metadata** (*bool*, *optional*) – metadatas?

**Returns** query result

**Return type** TYPE

**insert\_ttl\_string** (*ttl\_string*, *graph*)

Insert ttl into the triplestore

**Parameters**

- **ttl\_string** (*string*) – ttl triples to insert
- **graph** (*string*) – Insert in the named graph

**Returns** query result

**Return type** dict?

**load\_data** (*file\_name*, *graph*, *host\_url*)

Load data in function of the triplestore

**Parameters**

- **file\_name** (*string*) – File name to load
- **graph** (*string*) – graph name
- **host\_url** (*string*) – AskOmics url

**load\_data\_fuseki** (*file\_name*, *graph*)

Load data using fuseki load request

**Parameters**

- **file\_name** (*string*) – File name to load
- **graph** (*string*) – graph name

**Returns** Response of request

**Return type** response

**load\_data\_virtuoso** (*file\_name*, *graph*, *host\_url*)

Load data using virtuoso load query

**Parameters**

- **file\_name** (*string*) – File name to load
- **graph** (*string*) – graph name
- **host\_url** (*string*) – AskOmics url

**Returns** result of query

**Return type** TYPE



**parse\_results** (*json\_results*)  
Parse result of sparql query

**Parameters** **json\_results** (*dict*) – Query result

**Returns** Header and data

**Return type** list, list

**parse\_results\_old** (*json\_results*)  
Parse result of sparql query

**Parameters** **json\_results** – Result of the query

**Returns** parsed results

**Return type** list

**process\_query** (*query*)  
Execute a query and return parsed results

**Parameters** **query** (*string*) – The query to execute

**Returns** Parsed results

**Return type** list

### askomics.libaskomics.Start module

Contain the Start classe

**class** askomics.libaskomics.Start.**Start** (*app, session*)  
Bases: *askomics.libaskomics.Params.Params*

Initialize the data directory and the database

**data\_directory**  
Path to the data directory

**Type** str

**database\_path**  
Path to the database file

**Type** str

**create\_data\_directory** ()  
Create the data directory if it not exists

**create\_database** ()  
Initialize the database file

**start** ()  
Create the data diretory and initialize the database file

### askomics.libaskomics.TriplestoreExplorer module

**class** askomics.libaskomics.TriplestoreExplorer.**TriplestoreExplorer** (*app, session*)  
Bases: *askomics.libaskomics.Params.Params*

Explore the triplestore

**check\_presence** (*uri*, *list\_of\_things*)

Check if an uri is present in a list of dict

**Parameters**

- **uri** (*string*) – the uri to test
- **list\_of\_things** (*list*) – the list of dict['uri']

**Returns** True if the uri is present

**Return type** bool

**get\_abstraction** ()

Get AskOmics Abstraction

**Returns** AskOmics abstraction

**Return type** dict

**get\_abstraction\_attributes** ()

Get user abstraction attributes from the triplestore

**Returns** AskOmics attributes

**Return type** list

**get\_abstraction\_entities** ()

Get abstraction entities

**Returns** List of entities available

**Return type** list

**get\_abstraction\_relations** ()

Get user abstraction relations from the triplestore

**Returns** Relations

**Return type** list

**get\_attribute\_index** (*uri*, *attribute\_list*)

Get attribute index

**Parameters**

- **uri** (*string*) – uri of the attribute
- **attribute\_list** (*list*) – list of attributes

**Returns** Index of the given uri in the list

**Return type** int

**get\_startpoints** ()

Get public and user startpoints

**Returns** Startpoints

**Return type** list

## askomics.libaskomics.Utils module

**class** askomics.libaskomics.Utils.Utils

Bases: object

Contain utils fonction and classes

**static** `get_random_string(number)`

return a random string of n character

**Parameters** `number` (*int*) – number of character of the random string

**Returns** a random string of n chars

**Return type** str

**static** `humansize_to_bytes(hsize)`

Convert human-readable string into bytes

**Parameters** `hsize` (*string*) – Human readable string

**Returns** Bytes

**Return type** int

**static** `intersect(a, b)`

return the intersection of two lists

**static** `is_url(url)`

Check is string is an url

**Parameters** `url` (*string*) – string to test

**Returns** True if string is url

**Return type** bool

**static** `is_valid_url(url)`

Test if a string an url

**Parameters** `url` (*string*) – The url to test

**Returns** True is url is valid

**Return type** bool

**static** `union(a, b)`

return the union of two lists

**static** `unique(l)`

return the list with duplicate elements removed and keep order

**class** `askomics.libaskomics.Utils.cached_property(func)`

Bases: object

Like @property on a member function, but also cache the calculation in self.\_\_dict\_\_[function name]. The function is called only once since the cache stored as an instance attribute override the property residing in the class attributes. Following accesses cost no more than standard Python attribute access. If the instance attribute is deleted the next access will re-evaluate the function. Source: <https://blog.ionelmc.ro/2014/11/04/an-interesting-python-descriptor-quirk/>

`class Shape(object):`

`@cached_property def area(self):`

`# compute value return value`

**func**

Description

**Type** TYPE

**func**

## Module contents

## 5.2 Submodules

## 5.3 askomics.app module

AskOmics app

`askomics.app.BLUEPRINTS`

Flask blueprints

**Type** Tuple

`askomics.app.create_app` (*config*='config/askomics.ini', *app\_name*='askomics', *blueprints*=None)

Create the AskOmics app

**Parameters**

- **config** (*str*, *optional*) – Path to the config file
- **app\_name** (*str*, *optional*) – Application name
- **blueprints** (*None*, *optional*) – Flask blueprints

**Returns** AskOmics Flask application

**Return type** Flask

`askomics.app.create_celery` (*app*)

Create the celery object

**Parameters** **app** (*Flask*) – AskOmics Flask application

**Returns** Celery object

**Return type** Celery

## 5.4 askomics.tasks module

## 5.5 Module contents

## CHAPTER 6

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### a

- askomics, 48
- askomics.api, 20
  - askomics.api.admin, 13
  - askomics.api.auth, 14
  - askomics.api.catch\_url, 15
  - askomics.api.datasets, 15
  - askomics.api.file, 15
  - askomics.api.galaxy, 16
  - askomics.api.query, 17
  - askomics.api.results, 18
  - askomics.api.sparql, 19
  - askomics.api.start, 19
  - askomics.api.view, 19
- askomics.app, 48
- askomics.libaskomics, 48
  - askomics.libaskomics.BedFile, 20
  - askomics.libaskomics.CsvFile, 20
  - askomics.libaskomics.Database, 22
  - askomics.libaskomics.Dataset, 23
  - askomics.libaskomics.DatasetsHandler, 24
  - askomics.libaskomics.File, 25
  - askomics.libaskomics.FilesHandler, 28
  - askomics.libaskomics.FilesUtils, 30
  - askomics.libaskomics.Galaxy, 30
  - askomics.libaskomics.GffFile, 31
  - askomics.libaskomics.LocalAuth, 31
  - askomics.libaskomics.Params, 34
  - askomics.libaskomics.PrefixManager, 35
  - askomics.libaskomics.RdfFile, 35
  - askomics.libaskomics.RdfGraph, 36
  - askomics.libaskomics.Result, 37
  - askomics.libaskomics.ResultsHandler, 39
  - askomics.libaskomics.SparqlQueryBuilder, 40
  - askomics.libaskomics.SparqlQueryLauncher, 43
  - askomics.libaskomics.Start, 45
  - askomics.libaskomics.TriplestoreExplorer, 45
  - askomics.libaskomics.Utills, 46





## A

- `add()` (*askomics.libaskomics.RdfGraph.RdfGraph* method), 36
- `add_galaxy_account()` (*askomics.libaskomics.LocalAuth.LocalAuth* method), 31
- `admin_required()` (in module *askomics.api.auth*), 14
- `apikey` (*askomics.libaskomics.Galaxy.Galaxy* attribute), 30
- `app` (*askomics.libaskomics.Params.Params* attribute), 34
- askomics* (module), 48
- askomics.api* (module), 20
- askomics.api.admin* (module), 13
- askomics.api.auth* (module), 14
- askomics.api.catch\_url* (module), 15
- askomics.api.datasets* (module), 15
- askomics.api.file* (module), 15
- askomics.api.galaxy* (module), 16
- askomics.api.query* (module), 17
- askomics.api.results* (module), 18
- askomics.api.sparql* (module), 19
- askomics.api.start* (module), 19
- askomics.api.view* (module), 19
- askomics.app* (module), 48
- askomics.libaskomics* (module), 48
- askomics.libaskomics.BedFile* (module), 20
- askomics.libaskomics.CsvFile* (module), 20
- askomics.libaskomics.Database* (module), 22
- askomics.libaskomics.Dataset* (module), 23
- askomics.libaskomics.DatasetsHandler* (module), 24
- askomics.libaskomics.File* (module), 25
- askomics.libaskomics.FilesHandler* (module), 28
- askomics.libaskomics.FilesUtils* (module), 30
- askomics.libaskomics.Galaxy* (module), 30
- askomics.libaskomics.GffFile* (module), 31
- askomics.libaskomics.LocalAuth* (module), 31
- askomics.libaskomics.Params* (module), 34
- askomics.libaskomics.PrefixManager* (module), 35
- askomics.libaskomics.RdfFile* (module), 35
- askomics.libaskomics.RdfGraph* (module), 36
- askomics.libaskomics.Result* (module), 37
- askomics.libaskomics.ResultsHandler* (module), 39
- askomics.libaskomics.SparqlQueryBuilder* (module), 40
- askomics.libaskomics.SparqlQueryLauncher* (module), 43
- askomics.libaskomics.Start* (module), 45
- askomics.libaskomics.TriplestoreExplorer* (module), 45
- askomics.libaskomics.Utils* (module), 46
- askomics\_namespace* (*askomics.libaskomics.File.File* attribute), 25
- askomics\_namespace* (*askomics.libaskomics.PrefixManager.PrefixManager* attribute), 35
- askomics\_namespace* (*askomics.libaskomics.RdfGraph.RdfGraph* attribute), 36
- askomics\_prefix* (*askomics.libaskomics.File.File* attribute), 25
- askomics\_prefix* (*askomics.libaskomics.PrefixManager.PrefixManager* attribute), 35
- askomics\_prefix* (*askomics.libaskomics.RdfGraph.RdfGraph* attribute), 36
- `authenticate_user()` (*askomics.libaskomics.LocalAuth.LocalAuth* method), 32
- `authenticate_user_with_apikey()` (*askomics.libaskomics.LocalAuth.LocalAuth* method), 32

## B

BedFile (class in askomics.libaskomics.BedFile), 20  
 bind() (askomics.libaskomics.RdfGraph.RdfGraph method), 36  
 BLUEPRINTS (in module askomics.app), 48  
 build\_query\_from\_json() (askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder method), 40

## C

cached\_property (class in askomics.libaskomics.Utils), 47  
 catch\_all() (in module askomics.api.catch\_url), 15  
 category\_values (askomics.libaskomics.CsvFile.CsvFile attribute), 20  
 celery\_id (askomics.libaskomics.Dataset.Dataset attribute), 23  
 celery\_id (askomics.libaskomics.Result.Result attribute), 37  
 check\_columns\_types() (askomics.libaskomics.CsvFile.CsvFile method), 21  
 check\_galaxy\_instance() (askomics.libaskomics.Galaxy.Galaxy method), 30  
 check\_inputs() (askomics.libaskomics.LocalAuth.LocalAuth method), 32  
 check\_presence() (askomics.libaskomics.TriplestoreExplorer.TriplestoreExplorer method), 45  
 clean\_link() (askomics.libaskomics.Result.Result method), 37  
 clean\_node() (askomics.libaskomics.Result.Result method), 37  
 columns\_type (askomics.libaskomics.CsvFile.CsvFile attribute), 20  
 convert\_type() (askomics.libaskomics.File.File method), 27  
 create\_app() (in module askomics.app), 48  
 create\_celery() (in module askomics.app), 48  
 create\_data\_directory() (askomics.libaskomics.Start.Start method), 45  
 create\_database() (askomics.libaskomics.Start.Start method), 45  
 create\_datasets\_table() (askomics.libaskomics.Database.Database method), 23  
 create\_directory() (askomics.libaskomics.LocalAuth.LocalAuth method), 32  
 create\_endpoints\_table() (askomics.libaskomics.Database.Database method), 23

create\_files\_table() (askomics.libaskomics.Database.Database method), 23  
 create\_galaxy\_table() (askomics.libaskomics.Database.Database method), 23  
 create\_integration\_table() (askomics.libaskomics.Database.Database method), 23  
 create\_results\_table() (askomics.libaskomics.Database.Database method), 23  
 create\_user\_directories() (askomics.libaskomics.LocalAuth.LocalAuth method), 32  
 create\_user\_table() (askomics.libaskomics.Database.Database method), 23  
 CsvFile (class in askomics.libaskomics.CsvFile), 20

## D

data\_directory (askomics.libaskomics.Start.Start attribute), 45  
 Database (class in askomics.libaskomics.Database), 22  
 database\_path (askomics.libaskomics.Database.Database attribute), 22  
 database\_path (askomics.libaskomics.Start.Start attribute), 45  
 Dataset (class in askomics.libaskomics.Dataset), 23  
 datasets (askomics.libaskomics.DatasetsHandler.DatasetsHandler attribute), 24  
 datasets\_info (askomics.libaskomics.DatasetsHandler.DatasetsHandler attribute), 25  
 DatasetsHandler (class in askomics.libaskomics.DatasetsHandler), 24  
 dc (askomics.libaskomics.File.File attribute), 25  
 default\_graph (askomics.libaskomics.File.File attribute), 25  
 delete\_datasets() (askomics.libaskomics.DatasetsHandler.DatasetsHandler method), 25  
 delete\_datasets() (in module askomics.api.datasets), 15  
 delete\_datasets\_in\_db() (askomics.libaskomics.DatasetsHandler.DatasetsHandler method), 25  
 delete\_db\_entry() (askomics.libaskomics.Result.Result method), 37  
 delete\_file\_from\_db() (askomics.libaskomics.FilesHandler.FilesHandler method), 28  
 delete\_file\_from\_filesystem()

(*askomics.libaskomics.Result.Result* method), *FilesUtils* (class in *askomics.libaskomics.FilesUtils*), 30  
 37  
*delete\_file\_from\_fs()* (*askomics.libaskomics.FilesHandler.FilesHandler* method), 28  
*delete\_files()* (*askomics.libaskomics.FilesHandler.FilesHandler* method), 28  
*delete\_files()* (in module *askomics.api.file*), 15  
*delete\_from\_db()* (*askomics.libaskomics.Dataset.Dataset* method), 24  
*delete\_result()* (*askomics.libaskomics.Result.Result* method), 37  
*delete\_result()* (in module *askomics.api.results*), 18  
*delete\_results()* (*askomics.libaskomics.ResultsHandler.ResultsHandler* method), 39  
*delete\_results()* (*askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder* method), 40  
*dialect* (*askomics.libaskomics.CsvFile.CsvFile* attribute), 21  
*download\_datasets()* (*askomics.libaskomics.Galaxy.Galaxy* method), 30  
*download\_result()* (in module *askomics.api.results*), 18  
*download\_url()* (*askomics.libaskomics.FilesHandler.FilesHandler* method), 28  
*drop\_dataset()* (*askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher* method), 43

## G

*Galaxy* (class in *askomics.libaskomics.Galaxy*), 30  
*generate\_rdf\_content()* (*askomics.libaskomics.BedFile.BedFile* method), 20  
*execute\_query()* (*askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher* method), 43  
*execute\_sql\_query()* (*askomics.libaskomics.Database.Database* method), 23  
*generate\_rdf\_content()* (*askomics.libaskomics.GffFile.GffFile* method), 31

## F

*faldo* (*askomics.libaskomics.File.File* attribute), 25  
*faldo\_entity* (*askomics.libaskomics.File.File* attribute), 26  
*File* (class in *askomics.libaskomics.File*), 25  
*file\_graph* (*askomics.libaskomics.File.File* attribute), 26  
*file\_id* (*askomics.libaskomics.Dataset.Dataset* attribute), 23  
*file\_name* (*askomics.libaskomics.Result.Result* attribute), 37  
*file\_path* (*askomics.libaskomics.Result.Result* attribute), 37  
*files* (*askomics.libaskomics.FilesHandler.FilesHandler* attribute), 28  
*FilesHandler* (class in *askomics.libaskomics.FilesHandler*), 28  
*get\_abstraction()* (*askomics.libaskomics.TriplestoreExplorer.TriplestoreExplorer* method), 46  
*get\_abstraction()* (in module *askomics.api.query*), 17  
*get\_abstraction\_attributes()* (*askomics.libaskomics.TriplestoreExplorer.TriplestoreExplorer* method), 46  
*get\_abstraction\_entities()* (*askomics.libaskomics.TriplestoreExplorer.TriplestoreExplorer* method), 46  
*get\_abstraction\_relations()* (*askomics.libaskomics.TriplestoreExplorer.TriplestoreExplorer* method), 46  
*get\_all\_users()* (*askomics.libaskomics.LocalAuth.LocalAuth* method), 32  
*get\_attribute\_index()*

~~(askomics.libaskomics.TriplestoreExplorer.TriplestoreExplorer~~  
~~method), 46~~  
get\_checked\_asked\_graphs() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 39~~  
~~method), 41~~  
get\_dataset\_content() ~~(askomics.libaskomics.Galaxy.Galaxy method),~~  
~~30~~  
get\_dataset\_content() (in module askomics.api.galaxy), 16  
get\_datasets() (askomics.libaskomics.DatasetsHandler.DatasetsHandler  
~~method), 25~~  
get\_datasets() (in module askomics.api.datasets), 15  
get\_datasets() (in module askomics.api.galaxy), 17  
get\_datasets\_and\_histories() ~~(askomics.libaskomics.Galaxy.Galaxy method),~~  
~~30~~  
get\_default\_query() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 41~~  
get\_default\_query\_with\_prefix() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 41~~  
get\_dir\_path() (askomics.libaskomics.Result.Result  
~~method), 38~~  
get\_endpoints\_string() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 41~~  
get\_error() (askomics.libaskomics.Params.Params  
~~method), 34~~  
get\_error\_message() ~~(askomics.libaskomics.Params.Params~~  
~~method), 34~~  
get\_faldo\_strand() ~~(askomics.libaskomics.File.File method),~~  
~~27~~  
get\_federated\_froms() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 41~~  
get\_federated\_froms\_from\_graphs() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 41~~  
get\_federated\_line() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 41~~  
~~method), 46~~  
get\_file\_name() (askomics.libaskomics.FilesHandler.FilesHandler  
~~method), 29~~  
get\_file\_name() (askomics.libaskomics.Result.Result  
~~method), 38~~  
get\_file\_path() (askomics.libaskomics.FilesHandler.FilesHandler  
~~method), 29~~  
get\_file\_preview() ~~(askomics.libaskomics.Result.Result method),~~  
~~38~~  
get\_files() (in module askomics.api.file), 15  
get\_files\_info() (askomics.libaskomics.ResultsHandler.ResultsHandler  
~~method), 39~~  
get\_files\_infos() ~~(askomics.libaskomics.FilesHandler.FilesHandler~~  
~~method), 29~~  
get\_froms() (askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder  
~~method), 41~~  
get\_froms\_from\_graphs() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 42~~  
get\_graph\_state() ~~(askomics.libaskomics.Result.Result method),~~  
~~38~~  
get\_graph\_state() (in module askomics.api.results), 18  
get\_graphs\_and\_endpoints() ~~(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder~~  
~~method), 42~~  
get\_graph\_state() (askomics.libaskomics.Result.Result method), 38  
get\_graph\_state() (in module askomics.api.results), 18  
get\_graphs\_and\_endpoints() (askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder  
~~method), 42~~  
get\_metadata() (askomics.libaskomics.File.File  
~~method), 27~~  
get\_number\_of\_users() ~~(askomics.libaskomics.LocalAuth.LocalAuth~~  
~~method), 32~~  
get\_prefix() (askomics.libaskomics.PrefixManager.PrefixManager  
~~method), 35~~  
get\_preview() (askomics.libaskomics.BedFile.BedFile  
~~method), 20~~  
get\_preview() (askomics.libaskomics.CsvFile.CsvFile  
~~method), 21~~  
get\_preview() (askomics.libaskomics.GffFile.GffFile  
~~method), 31~~  
get\_preview() (askomics.libaskomics.RdfFile.RdfFile  
~~method), 35~~  
get\_preview() (in module askomics.api.file), 16  
get\_preview() (in module askomics.api.query), 17  
get\_preview() (in module askomics.api.results), 18  
get\_queries() ~~(askomics.libaskomics.ResultsHandler.ResultsHandler~~  
~~method), 39~~  
get\_random\_string() ~~(askomics.libaskomics.Utills.Utills static~~  
~~method), 46~~  
get\_rdf\_type() (askomics.libaskomics.File.File  
~~method), 27~~  
get\_results() (in module askomics.api.results), 18  
get\_size\_occupied\_by\_user() ~~(askomics.libaskomics.FilesUtils.FilesUtils~~  
~~method), 30~~  
get\_sparql\_query() ~~(askomics.libaskomics.Result.Result method),~~  
~~38~~

---

```

get_sparql_query() (in module askomics.libaskomics.BedFile.BedFile
askomics.api.results), 18
get_startpoints() (askomics.libaskomics.TriplestoreExplorer.TriplestoreExplorer
method), 46
get_triple() (askomics.libaskomics.RdfGraph.RdfGraph
method), 36
get_triples_from_graph() (askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher
method), 43
get_type() (askomics.libaskomics.FilesHandler.FilesHandler
method), 29
get_user() (askomics.libaskomics.LocalAuth.LocalAuth
method), 32
get_users() (in module askomics.api.admin), 13
GffFile (class in askomics.libaskomics.GffFile), 31
graph (askomics.libaskomics.RdfGraph.RdfGraph
attribute), 36
graph_name (askomics.libaskomics.Dataset.Dataset
attribute), 24
graph_state (askomics.libaskomics.Result.Result
attribute), 37
guess_column_type() (askomics.libaskomics.CsvFile.CsvFile
method), 21

H
handle_datasets() (askomics.libaskomics.DatasetsHandler.DatasetsHandler
method), 25
handle_files() (askomics.libaskomics.FilesHandler.FilesHandler
method), 29
header (askomics.libaskomics.CsvFile.CsvFile
attribute), 21
hello() (in module askomics.api.start), 19
home() (in module askomics.api.view), 19
host_url (askomics.libaskomics.File.File
attribute), 26
host_url (askomics.libaskomics.FilesHandler.FilesHandler
attribute), 28
humansize_to_bytes() (askomics.libaskomics.Utils.Utils
static
method), 47

I
id (askomics.libaskomics.Dataset.Dataset
attribute), 24
id (askomics.libaskomics.File.File
attribute), 26
id (askomics.libaskomics.Result.Result
attribute), 37
init_database() (askomics.libaskomics.Database.Database
method), 23
insert_data() (askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher
method), 44
insert_ttl_string() (askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher
method), 44

integrate() (askomics.libaskomics.BedFile.BedFile
method), 20
integrate() (askomics.libaskomics.CsvFile.CsvFile
method), 22
integrate() (askomics.libaskomics.File.File
method), 27
integrate() (askomics.libaskomics.GffFile.GffFile
method), 31
integrate() (askomics.libaskomics.RdfFile.RdfFile
method), 36
integrate() (in module askomics.api.file), 16
intersect() (askomics.libaskomics.Utils.Utils
static
method), 47
is_bnode() (askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder
method), 42
is_category() (askomics.libaskomics.CsvFile.CsvFile
method), 22
is_decimal() (askomics.libaskomics.CsvFile.CsvFile
static
method), 22
is_email_in_db() (askomics.libaskomics.LocalAuth.LocalAuth
method), 33
is_federated() (askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder
method), 42
is_url() (askomics.libaskomics.Utils.Utils
static
method), 47
is_username_in_db() (askomics.libaskomics.LocalAuth.LocalAuth
method), 33
is_valid_url() (askomics.libaskomics.Utils.Utils
static
method), 47

load_data() (askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher
method), 44
load_data_fuseki() (askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher
method), 44
load_data_virtuoso() (askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher
method), 44
load_graph() (askomics.libaskomics.File.File
method), 27
LocalAuth (class in askomics.libaskomics.LocalAuth),
31
log (askomics.libaskomics.Params.Params
attribute), 34
logged_user() (askomics.libaskomics.Params.Params
method), 35
login() (in module askomics.api.auth), 14
login_api_key() (in module askomics.api.auth), 14
login_api_token() (in module askomics.api.auth),
14
logout() (in module askomics.api.auth), 14

```



## M

`max_chunk_size` (*askomics.libaskomics.File.File* attribute), 26  
`merge()` (*askomics.libaskomics.RdfGraph.RdfGraph* method), 36  
`method` (*askomics.libaskomics.File.File* attribute), 26

## N

`name` (*askomics.libaskomics.Dataset.Dataset* attribute), 24  
`name` (*askomics.libaskomics.File.File* attribute), 26  
`now` (*askomics.libaskomics.File.File* attribute), 26  
`ntriple` (*askomics.libaskomics.RdfGraph.RdfGraph* attribute), 36  
`ntriples` (*askomics.libaskomics.File.File* attribute), 26

## P

`Params` (class in *askomics.libaskomics.Params*), 34  
`parse_results()` (*askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher* method), 44  
`parse_results_old()` (*askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher* method), 45  
`path` (*askomics.libaskomics.File.File* attribute), 26  
`persist_chunk()` (*askomics.libaskomics.FilesHandler.FilesHandler* method), 29  
`persist_user()` (*askomics.libaskomics.LocalAuth.LocalAuth* method), 33  
`prefix` (*askomics.libaskomics.PrefixManager.PrefixManager* attribute), 35  
`prefix()` (in module *askomics.api.sparql*), 19  
`prefix_query()` (*askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder* method), 42  
`PrefixManager` (class in *askomics.libaskomics.PrefixManager*), 35  
`preview` (*askomics.libaskomics.CsvFile.CsvFile* attribute), 21  
`private_graphs` (*askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder* attribute), 40  
`process_query()` (*askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher* method), 45  
`prov` (*askomics.libaskomics.File.File* attribute), 26  
`public` (*askomics.libaskomics.BedFile.BedFile* attribute), 20  
`public` (*askomics.libaskomics.CsvFile.CsvFile* attribute), 21  
`public` (*askomics.libaskomics.Dataset.Dataset* attribute), 24  
`public` (*askomics.libaskomics.File.File* attribute), 26  
`public` (*askomics.libaskomics.GffFile.GffFile* attribute), 31  
`public` (*askomics.libaskomics.RdfFile.RdfFile* attribute), 35  
`public_graphs` (*askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder* attribute), 40  
`publish_query()` (*askomics.libaskomics.Result.Result* method), 38  
`publish_query()` (in module *askomics.api.results*), 18  
`query()` (in module *askomics.api.query*), 17  
`query()` (in module *askomics.api.sparql*), 19  
`query_time` (*askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher* attribute), 43  
`RdfFile` (class in *askomics.libaskomics.RdfFile*), 35  
`RdfGraph` (class in *askomics.libaskomics.RdfGraph*), 36  
`rdftime()` (*askomics.libaskomics.File.File* method), 28  
`replace_froms()` (*askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder* method), 42  
`Result` (class in *askomics.libaskomics.Result*), 37  
`result_path` (*askomics.libaskomics.Result.Result* attribute), 37  
`ResultsHandler` (class in *askomics.libaskomics.ResultsHandler*), 39  
`rollback()` (*askomics.libaskomics.File.File* method), 28  
`rollback()` (*askomics.libaskomics.Result.Result* method), 38  
`SavesInLocal` (class in *askomics.libaskomics.Dataset.Dataset*), 24  
`save_in_db()` (*askomics.libaskomics.Result.Result* method), 38  
`save_query()` (in module *askomics.api.sparql*), 19  
`save_result()` (in module *askomics.api.query*), 17  
`save_result_to_file()` (*askomics.libaskomics.Result.Result* method), 38  
`send2galaxy()` (*askomics.libaskomics.Result.Result* method), 39  
`send2galaxy()` (in module *askomics.api.results*), 18  
`send_query_to_galaxy()` (*askomics.libaskomics.Result.Result* method), 39  
`send_result_to_galaxy()` (*askomics.libaskomics.Result.Result* method), 39  
`serialize()` (*askomics.libaskomics.RdfGraph.RdfGraph* method), 36  
`serve_file()` (in module *askomics.api.file*), 16  
`session` (*askomics.libaskomics.Params.Params* attribute), 34

## Q

`query()` (in module *askomics.api.query*), 17  
`query()` (in module *askomics.api.sparql*), 19  
`query_time` (*askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher* attribute), 43

## R

`RdfFile` (class in *askomics.libaskomics.RdfFile*), 35  
`RdfGraph` (class in *askomics.libaskomics.RdfGraph*), 36  
`rdftime()` (*askomics.libaskomics.File.File* method), 28  
`replace_froms()` (*askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder* method), 42  
`Result` (class in *askomics.libaskomics.Result*), 37  
`result_path` (*askomics.libaskomics.Result.Result* attribute), 37  
`ResultsHandler` (class in *askomics.libaskomics.ResultsHandler*), 39  
`rollback()` (*askomics.libaskomics.File.File* method), 28  
`rollback()` (*askomics.libaskomics.Result.Result* method), 38

## S

`SavesInLocal` (class in *askomics.libaskomics.Dataset.Dataset*), 24  
`save_in_db()` (*askomics.libaskomics.Result.Result* method), 38  
`save_query()` (in module *askomics.api.sparql*), 19  
`save_result()` (in module *askomics.api.query*), 17  
`save_result_to_file()` (*askomics.libaskomics.Result.Result* method), 38  
`send2galaxy()` (*askomics.libaskomics.Result.Result* method), 39  
`send2galaxy()` (in module *askomics.api.results*), 18  
`send_query_to_galaxy()` (*askomics.libaskomics.Result.Result* method), 39  
`send_result_to_galaxy()` (*askomics.libaskomics.Result.Result* method), 39  
`serialize()` (*askomics.libaskomics.RdfGraph.RdfGraph* method), 36  
`serve_file()` (in module *askomics.api.file*), 16  
`session` (*askomics.libaskomics.Params.Params* attribute), 34

[set\\_admin\(\) \(askomics.libaskomics.LocalAuth.LocalAuth method\), 33](#)  
[set\\_admin\(\) \(in module askomics.api.admin\), 13](#)  
[set\\_blocked\(\) \(askomics.libaskomics.LocalAuth.LocalAuth method\), 33](#)  
[set\\_blocked\(\) \(in module askomics.api.admin\), 13](#)  
[set\\_celery\\_id\(\) \(askomics.libaskomics.Result.Result method\), 39](#)  
[set\\_columns\\_type\(\) \(askomics.libaskomics.CsvFile.CsvFile method\), 22](#)  
[set\\_description\(\) \(in module askomics.api.results\), 18](#)  
[set\\_endpoints\(\) \(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder method\), 42](#)  
[set\\_graphs\(\) \(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder method\), 42](#)  
[set\\_graphs\\_and\\_endpoints\(\) \(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder method\), 42](#)  
[set\\_info\\_from\\_db\(\) \(askomics.libaskomics.Dataset.Dataset method\), 24](#)  
[set\\_info\\_from\\_db\\_with\\_id\(\) \(askomics.libaskomics.Result.Result method\), 39](#)  
[set\\_preview\(\) \(askomics.libaskomics.BedFile.BedFile method\), 20](#)  
[set\\_preview\(\) \(askomics.libaskomics.CsvFile.CsvFile method\), 22](#)  
[set\\_preview\(\) \(askomics.libaskomics.GffFile.GffFile method\), 31](#)  
[set\\_preview\(\) \(askomics.libaskomics.RdfFile.RdfFile method\), 36](#)  
[set\\_preview\\_and\\_header\(\) \(askomics.libaskomics.CsvFile.CsvFile method\), 22](#)  
[set\\_public\(\) \(in module askomics.api.results\), 19](#)  
[set\\_quota\(\) \(askomics.libaskomics.LocalAuth.LocalAuth method\), 33](#)  
[set\\_quota\(\) \(in module askomics.api.admin\), 13](#)  
[set\\_rdf\\_abstraction\(\) \(askomics.libaskomics.CsvFile.CsvFile method\), 22](#)  
[set\\_rdf\\_abstraction\\_domain\\_knowledge\(\) \(askomics.libaskomics.BedFile.BedFile method\), 20](#)  
[set\\_rdf\\_abstraction\\_domain\\_knowledge\(\) \(askomics.libaskomics.CsvFile.CsvFile method\), 22](#)  
[set\\_rdf\\_abstraction\\_domain\\_knowledge\(\) \(askomics.libaskomics.GffFile.GffFile method\), 31](#)  
[set\\_rdf\\_domain\\_knowledge\(\)](#)

[\(askomics.libaskomics.CsvFile.CsvFile method\), 22](#)  
[set\\_triples\\_number\(\) \(askomics.libaskomics.File.File method\), 28](#)  
[settings \(askomics.libaskomics.Params.Params attribute\), 34](#)  
[signup\(\) \(in module askomics.api.auth\), 14](#)  
[size \(askomics.libaskomics.File.File attribute\), 26](#)  
[SparqlQueryBuilder \(class in askomics.libaskomics.SparqlQueryBuilder\), 40](#)  
[SparqlQueryLauncher \(class in askomics.libaskomics.SparqlQueryLauncher\), 40](#)  
[Start \(class in askomics.libaskomics.Start\), 45](#)  
[start\(\) \(in module askomics.api.start\), 19](#)  
[store\\_file\\_info\\_in\\_db\(\) \(askomics.libaskomics.FilesHandler.FilesHandler method\), 29](#)  
[str\\_to\\_bool\(\) \(askomics.libaskomics.Params.Params method\), 35](#)

## T

[timestamp \(askomics.libaskomics.File.File attribute\), 26](#)  
[toggle\\_public\(\) \(askomics.libaskomics.Dataset.Dataset method\), 24](#)  
[toggle\\_public\(\) \(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder method\), 42](#)  
[toogle\\_public\(\) \(in module askomics.api.datasets\), 15](#)  
[transposed\\_preview \(askomics.libaskomics.CsvFile.CsvFile attribute\), 22](#)  
[triple\\_dict\\_to\\_string\(\) \(askomics.libaskomics.SparqlQueryBuilder.SparqlQueryBuilder method\), 43](#)  
[triplestore \(askomics.libaskomics.SparqlQueryLauncher.SparqlQueryLauncher attribute\), 43](#)  
[TriplestoreExplorer \(class in askomics.libaskomics.TriplestoreExplorer\), 45](#)  
[ttl\\_dir \(askomics.libaskomics.File.File attribute\), 27](#)  
[type \(askomics.libaskomics.File.File attribute\), 27](#)

## U

[union\(\) \(askomics.libaskomics.Utils.Utils static method\), 47](#)  
[unique\(\) \(askomics.libaskomics.Utils.Utils static method\), 47](#)  
[update\\_apikey\(\) \(askomics.libaskomics.LocalAuth.LocalAuth method\), 33](#)  
[update\\_apikey\(\) \(in module askomics.api.auth\), 14](#)

`update_celery()` (*askomics.libaskomics.Dataset.Dataset* *method*), 29

`update_celery()` (*askomics.libaskomics.Result.Result* *method*), 39

`update_datasets_table()`  
(*askomics.libaskomics.Database.Database* *method*), 23

`update_db_status()`  
(*askomics.libaskomics.Result.Result* *method*), 39

`update_description()`  
(*askomics.libaskomics.Result.Result* *method*), 39

`update_galaxy()` (*in module askomics.api.auth*), 14

`update_galaxy_account()`  
(*askomics.libaskomics.LocalAuth.LocalAuth* *method*), 34

`update_in_db()` (*askomics.libaskomics.Dataset.Dataset* *method*), 24

`update_password()`  
(*askomics.libaskomics.LocalAuth.LocalAuth* *method*), 34

`update_password()` (*in module askomics.api.auth*), 14

`update_profile()` (*askomics.libaskomics.LocalAuth.LocalAuth* *method*), 34

`update_profile()` (*in module askomics.api.auth*), 14

`update_public_status()`  
(*askomics.libaskomics.Result.Result* *method*), 39

`update_results_table()`  
(*askomics.libaskomics.Database.Database* *method*), 23

`update_status_in_db()`  
(*askomics.libaskomics.DatasetsHandler.DatasetsHandler* *method*), 25

`update_users_table()`  
(*askomics.libaskomics.Database.Database* *method*), 23

`upload_chunk()` (*in module askomics.api.file*), 16

`upload_datasets()` (*in module askomics.api.galaxy*), 17

`upload_path` (*askomics.libaskomics.FilesHandler.FilesHandler* *attribute*), 28

`upload_url()` (*in module askomics.api.file*), 16

`url` (*askomics.libaskomics.Galaxy.Galaxy* *attribute*), 30

`user_graph` (*askomics.libaskomics.File.File* *attribute*), 27

`Utils` (*class in askomics.libaskomics.Utils*), 46

## W

`write_data_into_file()`  
(*askomics.libaskomics.FilesHandler.FilesHandler*