
pystardog Documentation

Pedro Oliveira, John Bresnahan, Stephen Nowell

Jun 23, 2023

CONTENTS:

1	pystardog	1
1.1	What is it?	1
1.2	Installation	1
1.3	Documentation	1
1.4	Tests	1
1.5	Format	2
1.6	Quick Example	2
1.7	Interactive Tutorial	3
2	Modules	5
2.1	stardog.connection	5
2.2	stardog.admin	5
2.3	stardog.content	5
2.4	stardog.exceptions	5
3	Indices and tables	7

PYSTARDOG

Python wrapper for communicating with the Stardog HTTP server.

1.1 What is it?

This framework wraps all the functionality of a client for the Stardog Knowledge Graph, and provides access to a full set of functions such as executing SPARQL queries, administrative tasks on Stardog, and the use of the Reasoning API.

The implementation uses the HTTP protocol, since most of Stardog functionality is available using this protocol. For more information, go to the Stardog's [HTTP Programming](#) documentation.

1.2 Installation

pystardog is on PyPI so all you need is: `pip install pystardog`

1.3 Documentation

Documentation is readable at [Read the Docs](#) or can be built using Sphinx:

```
cd docs
pip install -r requirements.txt
make html
```

1.4 Tests

To run the tests locally, a valid Stardog license is required and placed in the `dockerfiles/stardog-license-key.bin`. Docker and docker-compose are also required.

1. Bring a stardog instance using docker-compose. For testing about 90% of the pystardog features, just a single node is sufficient, although we also provide a cluster set up for further testing.

```
# Bring a single node instance plus a bunch of Virtual Graphs for testing (Recommended).
docker-compose -f docker-compose.single-node.yml up -d
```

(continues on next page)

(continued from previous page)

```
# A cluster set up is also provided, if cluster only features are to be implemented and
↳ tested.
docker-compose -f docker-compose.cluster.yml up -d
```

Run the test suite. Create a virtual environment with the necessary dependencies:

```
# Create a virtualenv and activate it
virtualenv -p $(which python3) venv
source venv/bin/activate

# Install the dependencies
pip install -r requirements.txt -r test-requirements.txt

# Run the basic test suite (covers most of the pystardog functionalities)
pytest test/test_admin_basic.py test/test_connection.py test/test_utils.py -s
```

1.5 Format

To run a format of all the files

```
virtualenv -p $(which python3) venv
. venv/bin/activate
pip install -r test-requirements.txt
black .
```

1.6 Quick Example

```
import stardog

conn_details = {
    'endpoint': 'http://localhost:5820',
    'username': 'admin',
    'password': 'admin'
}

with stardog.Admin(**conn_details) as admin:
    db = admin.new_database('db')

    with stardog.Connection('db', **conn_details) as conn:
        conn.begin()
        conn.add(stardog.content.File('./test/data/example.ttl'))
        conn.commit()
        results = conn.select('select * { ?a ?p ?o }')

    db.drop()
```

1.7 Interactive Tutorial

There is a Jupyter notebook and instructions in the `notebooks` directory of this repository.

MODULES

2.1 stardog.connection

2.2 stardog.admin

2.3 stardog.content

2.4 stardog.exceptions

INDICES AND TABLES

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