JasperReports Tools Documentation

Release 0.3.1

Erick Navarro

Oct 24, 2018

Contents

Contents:

CHAPTER 1

JasperReports Tools

A collection of tools to handle Jasper Reports with python

- Free software: MIT license
- Documentation: http://jr-tools.readthedocs.io.

Tested with JasperServer CE 6.4

1.1 Features

- Client to get reports in API available formats(PDF, xls, etc)
- CLI: run `jr_tools --help` to get the list of available commands
- CLI: load resources from yaml file `jr_tools load path_to_yaml_file`

1.2 Development

For development there is a docker-compose based configuration to start jasper server and mysql.

Use the below commands to handle the docker setup:

- `make docker_up`: this will launch docker-compose services, it's going to take a few minutes to download the required images and setup everything.
- `make docker_down`: this will shutdown the launched containers.
- `make mysql_shell`: this will launch a mysql console to interact with the database, by default it connects to `demo` database.
- `make mysql_shell_root`: the same as above but use the `root` user.

Credentials:

Jasper Server:

- username: `jasperadmin`
- password: `jasperadmin`

MySQL:

- username: `demo`
- password: `demo`
- root password: `root`
- default database: `demo`

After the setup is complete you can enter to http://localhost:8080 and login using the credentials from above.

1.3 TODO

- Django helper to consume reports and converto to Django responses

1.4 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

2.1 Stable release

To install JasperReports Tools, run this command in your terminal:

```
$ pip install jr_tools
```

This is the preferred method to install JasperReports Tools, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for JasperReports Tools can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/erickgnavar/jr_tools
```

Or download the tarball:

```
$ curl  -OL https://github.com/erickgnavar/jr_tools/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

3.1 From python code

Run a report and get the binary result

```
from jr_tools.client import Client

client = Client(url='http://localhost:8080/jasperserver', username='jasperadmin',␣
↪password='secret')
result = client.run_report('/path/to/report', {'id': 1}, 'pdf')
```

3.2 From command line

Get all available commands

```
$ jr_tools --help
```

Run and save a report

```
$ jr_tools run_report /path/to/report result_file.pdf --format pdf
```

To get more info about the optional arguments run:

```
$ jr_tools run_report --help
```

Load resources from yaml file

```
$ jr_tools load resources.yml
```

Resources sample yml file

```
files:
- uri: /Files/report.jrxml
  path: /path/to/jrxml/file/on/disk
  type: jrxml

reports:
- uri: /Reports/report
  params:
    - label: param_id
      type: text
      mandatory: true
  jrxml_uri: /Files/report.jrxml
  data_source_uri: /DataSources/demo
```

The datasource must be configured previously

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/erickgnavar/jr_tools/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

4.1.4 Write Documentation

JasperReports Tools could always use more documentation, whether as part of the official JasperReports Tools docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/erickgnavar/jr_tools/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *jr_tools* for local development.

1. Fork the *jr_tools* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/jr_tools.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv jr_tools
$ cd jr_tools/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 jr_tools tests
$ python setup.py test or py.test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/erickgnavar/jr_tools/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_jr_tools
```

CHAPTER 5

Credits

5.1 Development Lead

- Erick Navarro <erick@navarro.io>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.3.1 (2018-10-24)

- Pin Click version to 6.X to use underscore command names

6.2 0.3.0 (2017-11-06)

- Add option to choose if a parameters must be mandatory

6.3 0.2.0 (2017-10-31)

- Add suport to upload and configure files and reports to JasperServer using a yml file

6.4 0.1.0 (2017-07-30)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search