
Liquimigrate

Release 0.5.2

Marek Wywiał

October 22, 2015

1	Liquimigrate	3
2	Installation	5
3	Usage	7
3.1	Quick start	7
3.2	CLI	7
3.3	Configuration settings	7
3.4	How it works	7
3.5	Development	8
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
7	0.5.2 (2015-10-22)	17
8	0.5.1 (2015-10-22)	19
9	0.5.0 (2015-08-06)	21
10	0.4.1 (2015-08-06)	23
11	0.4.0 (2014-09-28)	25
12	0.3.0 (2014-02-17)	27
13	0.2.8 (2012-02-28)	29
14	0.2.7 (2011-06-01)	31
15	0.2.6 (2011-05-20)	33

16	0.2.5 (2011-05-18)	35
17	0.2.4 (2011-05-17)	37
18	0.2.3 (2011-05-17)	39
19	0.2.2 (2011-05-17)	41
20	0.2.1 (2011-05-17)	43
21	0.2.0 (2011-04-29)	45
22	0.1.2 (2011-04-28)	47
23	0.1.1 (2011-04-28)	49
24	0.1.0 (2011-04-28)	51
25	Indices and tables	53

Contents:

Liquimigrate

Liquibase migrations for Django

- License: BSD
- Documentation: <https://liquimigrate.readthedocs.org>.

Installation

At the command line:

```
$ pip install liquimigrate
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv liquimigrate  
$ pip install liquimigrate
```

Usage

3.1 Quick start

- install liquimigrate
- add 'liquimigrate' to **INSTALLED_APPS**
- configure **LIQUIMIGRATE_CHANGELOG_FILES** = { 'default': os.path.join(os.path.dirname(__file__), "migrations", "migrations.xml") in settings }
- ensure that you have java on your path, liquibase and java drivers are embedded into package

3.2 CLI

Just run ./management.py liquibase update or ./management.py liquibase -h

To learn how to use liquibase look at liquibase documentation:

- <http://www.liquibase.org/quickstart>

3.3 Configuration settings

- **LIQUIMIGRATE_CHANGELOG_FILES** - dictionary with paths to change log files f.e. 'os.path.join(os.path.dirname(__file__), "migrations", "migrations.xml")' for every database connection you need to maintain using liquimigrate
- old method is still supported: **LIQUIMIGRATE_CHANGELOG_FILE** - path to change log file f.e. 'os.path.join(os.path.dirname(__file__), "migrations", "migrations.xml")'
- **LIQUIMIGRATE_DATABASE** - selected database - default 'default'

3.4 How it works

Database settings are got from **DATABASES**, from key 'default' or key configured in settings.py as **LIQUIMIGRATE_DATABASE**.

Changelog path are got from **LIQUIMIGRATE_CHANGELOG_FILES** or from command line (check -h).

Supported drivers:

- postgresql
- mysql
- more in future

3.5 Development

Whole command code is put in:

- liquimigrate/management/commands/liquibase.py
- java connectors are stored in liquimigrate/vendor/connectors
- available drivers mapping is stored in liquimigrate/__init__.py
- mapping to create db url for drivers is stored in liquimigrate/management/commands/liquibase.py in **DB_DEFAULTS**

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/onjin/liquimigrate/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” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

Liquimigrate could always use more documentation, whether as part of the official Liquimigrate 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/onjin/liquimigrate/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 *liquimigrate* for local development.

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

```
$ git clone git@github.com:your_name_here/liquimigrate.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 liquimigrate
$ cd liquimigrate/
$ 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 liquimigrate tests
$ python setup.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, and 3.4, and for PyPy. Check https://travis-ci.org/onjin/liquimigrate/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_liquimigrate
```

Credits

5.1 Development Lead

- Marek Wywiał <onjinx@gmail.com>

5.2 Contributors

- Marcin Nowak <marcin.j.nowak@gmail.com>
- Bartek Radwan <bartek@sq9mev.info>

History

0.5.2 (2015-10-22)

- added *management* to *MANIFEST.in*

0.5.1 (2015-10-22)

- handle liquibase return codes as an exception

0.5.0 (2015-08-06)

- add documentation and better package layout

0.4.1 (2015-08-06)

- fix for disabling signals
- add possibility to disable emitting pre- and postmigrate signals
- add example project configured with liquibase migrations

0.4.0 (2014-09-28)

- more detailed messages in makemigrations and squashmigrations commands
- Django 1.7+ compatibility + wrapped migration commands

0.3.0 (2014-02-17)

- Update README.rst
- removed version from __init__
- Update README.rst
- back to previous versioning scheme
- cleaned up settings header
- Changed Makefile:pypi to not use iw.dist
- Update README.rst
- possibility to change liquibase jar and connectors via settings
- db name, host, port were overridden by defaults

0.2.8 (2012-02-28)

- Updated documentation about new LIQUIMIGRATE_CHANGELOG_FILES directive Renamed README.txt to README.rst & included into MANIFEST.in Print additional info only in `-verbosity 1` model Minor changes for PEP8
- full multidb support

0.2.7 (2011-06-01)

- emit post_sync signal, call loaddata and initial_data
- support for singledb django versions
- fixed driver option

0.2.6 (2011-05-20)

- accepting `-driver` option from command line

0.2.5 (2011-05-18)

- Better README.txt - link to liquibase

0.2.4 (2011-05-17)

- FIXED: project path

0.2.3 (2011-05-17)

- Updated README.txt about mysql driver & about BSD license

0.2.2 (2011-05-17)

- ADDED: pass other args as liquibase args

0.2.1 (2011-05-17)

- FIXED: too small tuple when using not supported db driver

0.2.0 (2011-04-29)

- ADDED: mysql connector

0.1.2 (2011-04-28)

- ADDED: custom syncdb command
- ADDED: ensure command was given & more README.txt
- FIXED: run cmdline than just print them ;)
- FIXED: missing vendor in egg

0.1.1 (2011-04-28)

- REMOVED: old dependencies/namespaces

0.1.0 (2011-04-28)

Indices and tables

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