
neo-python-core Documentation

Release 0.5.6

City of Zion

Jan 13, 2019

Contents

1	Getting started	3
1.1	Useful commands	3
1.2	Release checklist	3
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
4.4	Tips	11
5	History	13
5.1	0.5.6 2018-12-18	13
5.2	0.5.5 2018-12-18	13
5.3	0.5.4 2018-10-31	13
5.4	0.5.3 2018-10-02	13
5.5	0.5.2 (2018-08-28)	13
5.6	0.5.1 (2018-08-23)	14
5.7	0.5.0 (2018-08-21)	14
5.8	0.4.11 (2018-07-05)	14
5.9	0.4.10 (2018-06-25)	14
5.10	0.4.9 (2018-06-08)	14
5.11	0.4.8 (2018-05-31)	14
5.12	0.4.7 (2018-05-30)	14
5.13	0.4.6 (2018-04-30)	14
5.14	0.4.2 (2018-04-26)	15
5.15	0.4.1 (2018-04-26)	15
5.16	0.3.10 (2018-03-21)	15
5.17	0.3.8 (2018-03-14)	15
5.18	0.3.6 (2018-02-26)	15
5.19	0.3.5 (2018-02-15)	15
5.20	0.3.4 (2018-01-25)	15

5.21	0.3.3 (2018-01-25)	15
5.22	0.3.2 (2018-01-23)	15
5.23	0.3.1 (2018-01-09)	16
5.24	0.3.0 (2018-01-09)	16
5.25	0.2.4 + 0.2.5 (2018-01-03)	16
5.26	0.2.3 (2018-01-03)	16
5.27	0.2.1 (2018-01-02)	16
5.28	0.1.1 - 0.1.2 (2017-12-30)	16
5.29	0.1.0 (2017-12-28)	16
6	Indices and tables	17

Contents:

Library for working with NEO related data in Python, without database dependencies.

- Datatypes like UInt160, KeyPair, BigInteger and basic string to address and address to UInt160 methods
- Includes a useful cli-tool np-utils (see help with np-utils -h)
- Compatible with Python 3.5+
- Used by [neo-python](#)
- <https://pypi.python.org/pypi/neocore>

np-utils examples:

```
$ np-utils -h
usage: np-utils [-h] [--version] [--address-to-scripthash address]
               [--scripthash-to-address scripthash] [--create-wallet]

optional arguments:
-h, --help            show this help message and exit
--version             show program's version number and exit
--address-to-scripthash address
                     Convert an address to scripthash
--scripthash-to-address scripthash
                     Convert scripthash to address
--create-wallet       Create a wallet

$ np-utils --create-wallet
{
"private_key": "KwJqCbjsmGUCqbkp83Nxi9MJ9mA7F8EN4tebJVWjYZBEoWCNxCaF",
"address": "AHVvg26CNzlvxteJfeHy4R8P4VN8SydCM6"
}

$ np-utils --address-to-scripthash AK2nJJpJr6o664CWJKilQRXjqeic2zRp8y
Scripthash big endian: 0xe9eed8dc39332032dc22e5d6e86332c50327ba23
Scripthash little endian: 23ba2703c53263e8d6e522dc3220339dcd8eee9
Scripthash neo-python format: b'\xba\x03\x52\xe8\xd6\xe5"\xc2 39\xdc\xd8\xee\xe9
→ '

$ np-utils --scripthash-to-address 0xe9eed8dc39332032dc22e5d6e86332c50327ba23
AK2nJJpJr6o664CWJKilQRXjqeic2zRp8y

$ np-utils --scripthash-to-address 23ba2703c53263e8d6e522dc3220339dcd8eee9
Detected little endian scripthash. Converting to big endian for internal use.
Big endian scripthash: 0xe9eed8dc39332032dc22e5d6e86332c50327ba23
AK2nJJpJr6o664CWJKilQRXjqeic2zRp8y
```


CHAPTER 1

Getting started

You need Python 3.5 or higher!

You can install *neocore* from PyPI with `easy_install` or `pip`:

```
$ pip install -U neocore
```

Alternatively, if you want to work on the code, clone this repository and setup your venv:

- Clone the repo: `git clone https://github.com/CityOfZion/neo-python-core.git`
- Create a Python 3 virtual environment and activate it:

```
$ python3 -m venv venv  
$ source venv/bin/activate
```

- Then install the requirements:

```
$ pip install -e .  
$ pip install -r requirements_dev.txt
```

1.1 Useful commands

```
$ make lint  
$ make test  
$ make coverage
```

1.2 Release checklist

(Only for admins)

Releasing a new version on GitHub automatically uploads this release to PyPI. This is a checklist for releasing a new version:

```
# Only in case you want to increase the version number again (eg. scope changed from ↵
↵patch to minor):
bumpversion --no-tag minor|major

# Update ``HISTORY.rst`` with the new version number and the changes and commit this
vi HISTORY.rst
git commit -m "Updated HISTORY.rst" HISTORY.rst

# Set the release version number and create the tag
bumpversion release

# Increase patch number and add ``-dev``
bumpversion --no-tag patch

# Push to GitHub, which also updates the PyPI package
git push && git push --tags
```


2.1 Stable release

To install neo-python-core, run this command in your terminal:

```
$ pip install neocore
```

This is the preferred method to install neo-python-core, 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 neo-python-core can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/CityOfZion/neo-python-core
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/CityOfZion/neo-python-core/tarball/master
```

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

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


CHAPTER 3

Usage

To use neo-python-core in a project:

```
import neocore
```


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/CityOfZion/neo-python-core/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

neo-python-core could always use more documentation, whether as part of the official neo-python-core 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/CityOfZion/neo-python-core/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 *neo-python-core* for local development.

1. Fork the *neo-python-core* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:<your-name>/neo-python-core.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 neo-python-core
$ cd neo-python-core/
$ 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 neocore tests
$ python setup.py test or py.test
```

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/CityOfZion/neo-python-core/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_neocore
```


5.1 0.5.6 2018-12-18

- Updated dependencies
- Added `SafeReadBytes` to `BinaryReader`

5.2 0.5.5 2018-12-18

- Updated dependencies
- Added `IsValidPublicAddress()` to utility module

5.3 0.5.4 2018-10-31

- Updated *BigInteger* to properly support logical shifts
- Added tests for *np-utils*
- Updated dependencies

5.4 0.5.3 2018-10-02

- Updated the dependencies

5.5 0.5.2 (2018-08-28)

- *Fixed8.TryParse* fix for zero

- Updated dependencies

5.6 0.5.1 (2018-08-23)

- Change BigInteger divisor operation to use floordiv rather than truediv

5.7 0.5.0 (2018-08-21)

- `np-utils --address-to-scripthash` outputs now little-endian and big-endian scripthashes
- `np-utils --scripthash-to-address` detects input endianness and converts accordingly
- Updated dependencies

5.8 0.4.11 (2018-07-05)

- Added `Size()` method to *ECPoint* and *Fixed8* class.

5.9 0.4.10 (2018-06-25)

- Updated requirements: pycryptome

5.10 0.4.9 (2018-06-08)

- Updated dependencies, especially base58

5.11 0.4.8 (2018-05-31)

- Create wallets with `np-utils --create-wallet`

5.12 0.4.7 (2018-05-30)

- `BigInteger(0)` now is `b'\x00'` ([PR #50](#))

5.13 0.4.6 (2018-04-30)

- make `unhexlify` in `Crypto.VerifySignature` optional ([PR #48](#))

5.14 0.4.2 (2018-04-26)

- `np-utils` now supports `--scripthash-to-address` (thx @belane)

5.15 0.4.1 (2018-04-26)

- `np-utils` cli tool (see `cli.py`, [PR #40](#))
- alter initialization of Crypto signature curve

5.16 0.3.10 (2018-03-21)

- Fix formatting of `ToNeoJsonString()` which was cutting off trailing zeroes from integers.

5.17 0.3.8 (2018-03-14)

- Fix travis deploy to be compatible with recent neo-python changes
- Update `script` and `logzero` dependency versions

5.18 0.3.6 (2018-02-26)

- Enabled Python `>= 3.4` in `setup.py`

5.19 0.3.5 (2018-02-15)

- Bugfix: Dont unhex when writing var bytes ([PR #36](#))

5.20 0.3.4 (2018-01-25)

- Added `ParseString` method to `UInt160/UInt256` ([PR #35](#))

5.21 0.3.3 (2018-01-25)

- Added `Fixed8.ToJsonString()` ([PR #33](#))

5.22 0.3.2 (2018-01-23)

- Added `UInt ToHexString` method

5.23 0.3.1 (2018-01-09)

- Documentation update
- Moved the `cryptography` dependency to `requirements_dev.txt`

5.24 0.3.0 (2018-01-09)

- Added `neo.Cryptography` and `KeyPair`
- Changed signature of `neocore.Cryptography.Crypto.Sign()` to remove unused `public_key` argument
- Removed redundant `neocore.Cryptography.Helper.hash_to_wallet_address()` function, use `neocore.Cryptography.Helper.scripthash_to_address()` instead.
- Removed unused `neocore.Cryptography.Helper` functions: `random_string`, `bytes_to_hex_string`, `bin_sha256`, `sha256`, `random_key`.

5.25 0.2.4 + 0.2.5 (2018-01-03)

- Bugfix for deploying from Travis to PyPI/neocore

5.26 0.2.3 (2018-01-03)

- Bugfix for `BinaryWriter` ([PR #13](#))

5.27 0.2.1 (2018-01-02)

- Added `UInt*`, `Fixed8` and `neo.IO.Binary*` ([PR #9](#))

5.28 0.1.1 - 0.1.2 (2017-12-30)

- Testing of releases on PyPI with Travis CI.

5.29 0.1.0 (2017-12-28)

- First release on PyPI.

CHAPTER 6

Indices and tables

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