
PyMoneroWallet Documentation

Release 0.1

Carl Chenet <chaica@ohmytux.com.org>

October 15, 2016

1	Guide	3
1.1	How to install PyMoneroWallet	3
1.2	Use the PyMoneroWallet library	3
1.3	The <code>monerowallet</code> module	4
1.4	The <code>exceptions</code> module	8
1.5	Troubleshooting	8
1.6	License	9
1.7	Authors	9
2	Indices and tables	11
	Python Module Index	13



PyMoneroWallet is a Python library to query a Monero wallet.

You'll find in this documentation anything you need to install your Monero wallet and use the PyMoneroWallet library.

1.1 How to install PyMoneroWallet

1.1.1 From PyPI

```
$ pip3 install pmonerowallet
```

1.1.2 From sources

- You need at least Python 3.4.
- PyMoneroWallet 0.1 was only tested with Monero 0.10.0.0
- On some Linux Distribution **setuptools** package does not come with default python install, you need to install it.
- Install **PIP**:

```
$ wget https://bootstrap.pypa.io/get-pip.py -O - | sudo python3.4
```

- Install **setuptools** module:

```
$ wget https://bootstrap.pypa.io/ez_setup.py -O - | sudo python3.4
```

- Alternatively, Setuptools may be installed to a user-local path:

```
$ wget https://bootstrap.pypa.io/ez_setup.py -O - | python3.4 --user
```

- Untar the tarball and go to the source directory with the following commands:

```
$ tar zxvf pmonerowallet-0.1.tar.gz  
$ cd pmonerowallet
```

- Next, to install PyMoneroWallet on your computer, type the following command with the root user:

```
$ python3.4 setup.py install
```

1.2 Use the PyMoneroWallet library

The first thing you need to install the Monero wallet 0.10.0.0:

```
$ mkdir monero_0.10.0.0
$ cd monero_0.10.0.0
$ wget https://downloads.getmonero.org/monero.linux.x64.v0-10-0-0.tar.bz2
$ tar jxvf monero.linux.x64.v0-10-0-0.tar.bz2
```

Now you need to synchronize your Monero wallet:

```
$ monerod
```

Once you are synchronized, you need to launch your Monero wallet for the first time to define your wallet password:

```
$ monero-wallet-cli
```

After having creating your Monero wallet and define your password, you need to launch your Monero wallet activating the RPC calls:

```
$ monero-wallet-cli --wallet-file monerowallet --password 'v3rY S3cr3t P4sSw0rd' --rpc-bind-ip 127.0
```

Now we are ready to use the PyMoneroWallet library with Python3:

```
$ python3
>>> from monerowallet import MoneroWallet
>>> mw = MoneroWallet()
>>> mw.getbalance()
{'unlocked_balance': 2262265030000, 'balance': 2262265030000}
```

To get extensive details about available methods, see the documentation of the monerowallet module.

1.3 The monerowallet module

Provide pythonic way to request a Monero wallet.

Example

```
>>> import monerowallet
>>> mw = monerowallet.MoneroWallet()
>>> mw.getaddress()
'94EJSG4URLDVwzAgDvCLaRwFGHxv75DT5MvFp1YfAxQU9icGxjVJiY8Jr9YF1atXN7UFBDx3vJq2s3CzULkPrEAuEioqyrP'
```

```
class monerowallet.MoneroWallet (protocol='http', host='127.0.0.1', port=18082,
                                  path='/json_rpc')
```

The MoneroWallet class. Instantiate a MoneroWallet object with parameters to dialog with the RPC wallet server.

Parameters

- **protocol** (*str*) – Protocol for requesting the RPC server ('http' or 'https', defaults to 'http')
- **host** – The host for requesting the RPC server (defaults to '127.0.0.1')
- **port** (*str*) – The port for requesting the RPC server (defaults to 18082)
- **path** (*str*) – The path for requesting the RPC server (defaults to '/json_rpc')

Returns A MoneroWallet object

Return type *MoneroWallet*

Example


```
>>> mw = MoneroWallet()
>>> mw
<monerowallet.MoneroWallet object at 0x7fe09e4e8da0>
```

get_bulk_payments (*payment_ids, min_block_height*)

Get a list of incoming payments using a given payment id, or a list of payments ids, from a given height. This method is the preferred method over `get_payments` because it has the same functionality but is more extendable. Either is fine for looking up transactions by a single payment ID.

Parameters `payment_ids` (*list*) – A list of incoming payments

Returns A list of dictionaries with the details of the incoming payments

Return type dict

Example

```
>>> mw.get_bulk_payments(['94dd4c2613f5919d'], 1148609)
>>> mw.get_bulk_payments(['fdfcfd993482b58b'], 1157950)
[{'unlock_time': 0, 'amount': 1000000000, 'tx_hash': 'db3870905ce3c8ca349e224688c344371addca
```

get_payments (*payment_id*)

Get a list of incoming payments using a given payment id.

Parameters `payment_id` (*str*) – Payment id

Returns A list of dictionaries with the details of the incoming payments

Return type list

Example

```
>>> mw = MoneroWallet()
>>> mw.get_payments('fdfcfd993482b58b')
[{'unlock_time': 0, 'amount': 1000000000, 'tx_hash': 'db3870905ce3c8ca349e224688c344371addca
```

getaddress ()

Return the wallet's address.

Returns A string with the address of the wallet

Return type str

Example

```
>>> mw.getaddress()
'94EJSG4URLDVwzAgDvCLaRwFGHxv75DT5MvFp1YfAxQU9icGxjVJiY8Jr9YF1atXN7UFBDx3vJq2s3CzULkPrEAuEic
```

getbalance ()

Return the wallet's balance.

Returns A dictionary with the wallet balance and the unlocked balance

Return type dict

Example

```
>>> mw.getbalance()
{'unlocked_balance': 2262265030000, 'balance': 2262265030000}
```

getheight ()

Returns the wallet's current block height.

Returns An integer with the wallet's current block height

Return type int

Example

```
>>> mw.getheight()
1146043
```

incoming_transfers (transfer_type='all')

Return a list of incoming transfers to the wallet.

Parameters **transfer_type** (*str*) – The transfer type ('all', 'available' or 'unavailable')

Returns A list with the incoming transfers

Return type list

Example

```
>>> import pprint # just useful for a nice display of data
>>> pprint.pprint(mw.incoming_transfers())
[{'amount': 30000,
    'global_index': 4593,
    'spent': False,
    'tx_hash': '0a4562f0bfc4c5e7123e0ff212b1ca810c76a95fa45b18a7d7c4f',
    'tx_size': 606},
 {'amount': 5000000,
    'global_index': 23572,
    'spent': False,
    'tx_hash': '1a4567f0afc7e5e7123e0aa192b2ca101c75a95ba12b53a1d7c4f',
    'tx_size': 606},
 ]
```

make_integrated_address (payment_id='')

Make an integrated address from the wallet address and a payment id.

Parameters **payment_id** (*str*) – Specific payment id. Otherwise it is randomly generated

Returns A dictionary with both integrated address and payment id

Return type dict

Example

```
>>> mw.make_integrated_address()
{'integrated_address': '4JwWT4sy2bjFfzSxvRBUxTLftcNM98DT5MvFp4JNJRh3icqrjVJiY8Jr9YF1atXN7Uf'}
```

query_key (key_type='mnemonic')

Return the spend or view private key.

Parameters `key_type` (*str*) – Which key to retrieve ('mnemonic' or 'view_key', default is 'mnemonic')

Returns A string with either the mnemonic-format key either the hexadecimal-format key

Return type `str`

Example

```
>>> mw.query_key(key_type='mnemonic')
'adapt adapt nostril using suture tail faked relic huddle army gags bugs abyss wield tidy ja
>>> mw.query_key(key_type='view_key')
'49c087c10112eea3554d85bc9813c57f8bbd1cac1f3abb3b70d12cbea712c908'
```

split_integrated_address (*integrated_address*)

Retrieve the standard address and payment id corresponding to an integrated address.

Parameters `integrated_address` (*str*) – the integrated address to split

Returns a dictionary with the payment id and the standard address

Return type `dict`

Example

```
>>> mw.split_integrated_address('4JwWT4sy2bjFfzSxvRBUxTLftcNM98DT5MvFp4JNJRh3iCqrjVJiY8Jr9Y
{'standard_address': '12GLv8KzVhxehev712FWPTF7CSWuVjuBarFd17QP163uxMaFyoqwmDf1aiRtS5jWgCkRsk1
```

stop_wallet ()

Stops the wallet, storing the current state.

Returns An empty dictionary

Return type `dict`

Example

```
>>> mw.stop_wallet()
{}
```

store ()

Save the blockchain.

Returns An empty dictionary

Return type `dict`

Example

```
>>> mw.store()
{}
```

sweep_dust ()

Send all dust outputs back to the wallet's, to make them easier to spend (and mix).

Returns a list of the hashes of the transactions

Return type `list`

Example

```
>>> mw.sweep_dust ()
[]
```

transfer (*destinations*)

Send monero to a number of recipients.

Parameters **destinations** – a list of destinations to receive XMR

Returns a dict of with the hash and the key of the transaction

Example

```
>>> mw.transfer([{'amount': 10000000000, 'address': '51EqSG4URLDFfzSxvRBUxTLftcMM76DT3MvFp3J
{'tx_hash': 'd4d0048c275e816ae1f6f55b4b04f7d508662679c044741db2aeb7cd63452059', 'tx_key': ' '}
```

transfer_split (*destinations*)

Send monero to a number of recipients.

Parameters **destinations** – a list of destinations to receive XMR

Returns a list with the transaction hashes

Return type list

Example

```
>>> mw.transfer_split([{'amount': 10000000000, 'address': '59EqSG5UKBDFfzSxvRABxTLftcNM77DT3
['653a5da2dd541ab4b3d9811f84255bb243dd7338c1218c5e75036725b6ca123e']
```

1.4 The exceptions module

Exceptions raised by PyMoneroWallet.

exception `monerowallet.exceptions.Error`

General PyMoneroWallet exception

exception `monerowallet.exceptions.MethodNotFoundError`

Returned exception when the RCP server of the Monero wallet is not able to understand the request

exception `monerowallet.exceptions.StatusCodeError`

Returned exception when returned HTTP status code is different from 200

1.5 Troubleshooting

The main issue you can encounter is the following error:

```
>>> from monerowallet import MoneroWallet
>>> mw = MoneroWallet()
>>> mw.getaddress()
ConnectionRefusedError: [Errno 111] Connection refused
...
requests.exceptions.ConnectionError: HTTPConnectionPool(host='127.0.0.1', port=18082): Max retries ex
```

It means the RPC server of your Monero wallet is not listening of the dedicated port and so can not reply to requests. Launch it as described in the *Use the PyMoneroWallet library* section. For other errors, have a look at the *monerowallet.exceptions* or open a bug report.

1.6 License

This software comes under the terms of the **GPLv3+**. See the LICENSE file for the complete text of the license.

1.7 Authors

Carl Chenet <chaica@ohmytux.com>

Indices and tables

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

m

`monerowallet`, 4

`monerowallet.exceptions`, 8

E

Error, 8

G

get_bulk_payments() (monerowallet.MoneroWallet method), 5

get_payments() (monerowallet.MoneroWallet method), 5

getaddress() (monerowallet.MoneroWallet method), 5

getbalance() (monerowallet.MoneroWallet method), 5

getheight() (monerowallet.MoneroWallet method), 6

I

incoming_transfers() (monerowallet.MoneroWallet method), 6

M

make_integrated_address() (monerowallet.MoneroWallet method), 6

MethodNotFoundError, 8

MoneroWallet (class in monerowallet), 4

monerowallet (module), 4

monerowallet.exceptions (module), 8

Q

query_key() (monerowallet.MoneroWallet method), 6

S

split_integrated_address() (monerowallet.MoneroWallet method), 7

StatusCodeError, 8

stop_wallet() (monerowallet.MoneroWallet method), 7

store() (monerowallet.MoneroWallet method), 7

sweep_dust() (monerowallet.MoneroWallet method), 7

T

transfer() (monerowallet.MoneroWallet method), 8

transfer_split() (monerowallet.MoneroWallet method), 8