libchain Documentation

Release 0.1 - Alpha

@Bleznudd - Matteo Carrirolo

Jun 17, 2018

Contents

`libchain` requires Python 3.6 or above. If you do not already have a Python environment configured on your computer, please see getting python.

Below we assume you have the default Python environment already configured on your computer and you intend to install `libchain` inside of it.

First, make sure you have the latest version of `pip` (the Python package manager) installed. If you do not, refer to the pip documentation and install `pip` first.

CHAPTER 1

Install the released version

Install the curent release of `libchain` with `pip`:

```
$ pip install libchain
```

To upgrade to a newer release use the `--upgrade` flag:

```
$ pip install --upgrade libchain
```

If you do not have permission to install software systemwide, you can install into your user directory using the `--user` flag:

```
$ pip install --user libchain
```

CHAPTER 2

Install the development version

Before installing the development version, you may need to uninstall the standard version of `libchain` (if installed) using `pip`:

```
$  pip uninstall libchain
```

Then, if you have git installed on your system, clone this repository:

```
$ git clone https://github.com/Bleznudd/libchain.git
$ pip install setup.py
```

Warning: Since you're installing the git-version be aware is up to you to take precaution against damages resulting from this software. Software in testing should for example not be used on sensitive and/or valuable data. The usage of this software is at your own risk and may void warranty on your products. If you are not an experienced user, or otherwise in in any doubt of your capabilities to make such a decision, please do not download and/or use this software.

CHAPTER 3

Classes and their methods

Here we want to give an overview of libchain's classes and the methods avaiable.

Note: In the methods documentation `1bitcoinaddress`, `txhash` and `blockhash` are used instead of real addresses, transaction hashes and blocks hashes. If you want to try the examples listed below, you have to use real values.

3.1 user

class libchain.**user**(*_addres=0*, *_value=0*)

Defines an address, its balance, its num of tx, and the amount wich has sent or received in a tx

balance()

Give the balance of the address of the user

Parameters **none** –

Returns the amount of BTC owned by the user

Return type float

Raises

- `AddressError` – if the address is wrong
- `Bech32AddressError` – if the address a bech32, this kind of address is not yet supported by blockchain.info

Examples:

```
>>>> u = lc.user('1bitcoinaddresshash')
>>>> u.balance()
1.0284
```

parse(*_json={}*, **args*)

When a json is given, it is loaded in the user

Parameters **response** (*json*) – a requests' json encoded response from blockchain.info. An example can be found here address sample

Returns none

Examples:

```
>>>> import requests as rq
>>>> data = rq.get('https://blockchain.info/rawaddr/1bitcoinaddress')
>>>> data = data.json()
>>>> u = lc.user()
>>>> u.parse(data)
```

Warning: This method shouldn't be used, use `fromhash` instead

totalreceived()

Gives the amount of BTC wich has been received by the user

Parameters **none** –

Returns the total amount of BTC received in all history by the address of the user

Return type float

Raises

- `AddressError` – if the address is wrong
- `Bech32AddressError` – if the address a bech32, this kind of address is not yet supported by blockchain.info

Examples:

```
>>>> u = lc.user('1bitcoinaddresshash')
>>>> u.totalreceived()
12.084
```

totaltx()

Gives the number of transactions in wich the user is involved

See also:

It's the same as `len(u.transactions())`

Parameters **none** –

Returns total number of transactions in wich the address of the user is involved

Return type int

Raises

- `AddressError` – if the address is wrong

- `Bech32AddressError` – if the address a bech32, this kind of address is not yet supported by blockchain.info

Examples:

```
>>>> u = lc.user('1bitcoinaddresshash')
>>>> u.totaltx()
3
```

transactions()

Gives the list of tx in wich the user is involved

Parameters **none** –

Returns a list of all the tx hashes in wich is involved the address of the user

Return type string list

Raises

- `AddressError` – if the address is wrong
- `Bech32AddressError` – if the address a bech32, this kind of address is not yet supported by blockchain.info

Examples:

```
>>>> u = lc.user('1bitcoinaddresshash')
>>>> u.transactions()
['txhash1','txhash2','txhash3']
```

3.2 transaction

class libchain.**tx**(*_hash=0*, *_index=0*, *_senders=[]*, *_recipients=[]*, **args*)

Defines a transaction in a block of the bitcoin's blockchain

fromhash(*_hash*)

When a hash is given, the corrisponding transaction is loaded

Parameters **string** – the hash of a valid transaction

Returns none

Raises `TxHashError` – if the hash argument is invalid

Examples:

```
>>>> t = lc.tx()
>>>> t.fromhash('txhash')
```

parse(*_json={}*, **args*)

When a json is given, it is loaded in the transaction.

- `---------- mined ----------` is assigned for newly created BTC
- `------- op return / unknown ------` is assigned for transactions with no recepient

Parameters **response** (*json*) – a requests' json encoded response from blockchain.info. An example can be found here: tx sample

Returns none

Examples:

```
>>>> import requests as rq
>>>> data = rq.get('https://blockchain.info/rawtx/1bitcoinaddress')
>>>> data = data.json()
>>>> t = lc.tx()
>>>> t.parse(data)
```

Warning: This method shouldn't be used, use `fromhash` instead

simplelinks()

Gives a list of tuple (from addr, to addr) from the transaction

Parameters **none** –

Returns a list of tuple (from addr, to addr) from the transaction. Each sender is counted once for each recepients

Return type tuple list

Examples:

```
>>>> t = lc.tx('txhash')
>>>> t.simplelinks()
[('1bitcoinaddress','3bitcoinaddress'),('1bitcoinaddress','4bitcoinaddress'),
('2bitcoinaddress','3bitcoinaddress'),('2bitcoinaddress','4bitcoinaddress')]
```

value()

Gives the amount of BTC moved within the transaction

Parameters **none** –

Returns the amount of BTC moved within the transaction

Return type float

Examples:

```
>>>> t = lc.tx('txhash')
>>>> t.value()
1051.13479
```

3.3 block

class libchain.**block**(*_hash=0*, *_prevblock=0*, *_time=0*, *_index=0*, *_height=0*, *_tx=[]*, **args*)

Defines a block of the bitcoin's blockchain

fromhash(*_hash*)
When a hash is given, the corrisponding block is loaded

Parameters **string** – the hash of a valid block

Returns none

Raises BlockHashError – if the hash argument is invalid

Examples:

```
>>>> b = lc.b()
>>>> b.fromhash('blockhash')
```

fromheight(*_height*)
When a height is given, the corrisponding block is loaded

Parameters **string** – the hash of a valid block

Returns none

Raises BlockHashError – if the hash argument is invalid

Examples:

```
>>>> t = lc.tx()
>>>> t.fromheight('blockheight')
```

parse(*_json={}*, **args*)
When a json file is given, it is loaded in the block

Parameters **response** (*json*) – a requests' json encoded response from blockchain.info. An example can be found here block sample

Returns none

Examples:

```
>>>> import requests as rq
>>>> data = rq.get('https://blockchain.info/rawblock/blockhash')
>>>> data = data.json()
>>>> b = lc.block()
>>>> b.parse(data)
```

Warning: This method shouldn't be used, use fromhash instead

value()
Give the amount of BTC moved within the block

Parameters **none** –

Returns the amount of BTC moved within the block

Return type float

Examples:

```
>>>> b = lc.b('blockhash')
>>>> b.value()
101951.179
```

CHAPTER 4

Functions

In this section you can find non-calss related functions and their description.

Note: In the methods documentation `1bitcoinaddress`, `txhash` and `blockhash` are used instead of real addresses, transaction hashes and blocks hashes. If you want to try the examples listed below, you have to use real values.

`libchain.`**`tobtc`**`(`*_currency*, *_value*`)`

When a currency symbol (USD, EUR, JPY, ...) and a value are given, the amount is converted in bitcoin at the actual exchange rate

Examples:

```
>>>> libchain.tobtc('EUR', 100)
0.01783487
```

`libchain.`**`exchangerate`**`()`

Return a dictionary with the actual exchange rate of the most famous currencies

Examples:

```
>>>> libchain.exchangerate()
{'USD': {'15m': 6508.15, 'last': 6508.15, 'buy': 6508.15, 'sell': 6508.15, 'symbol
↪': '$'},
 'AUD': {'15m': 8740.46, 'last': 8740.46, 'buy': 8740.46, 'sell': 8740.46, 'symbol
↪': '$'},
 ...}
```

`libchain.`**`lasthash`**`()`

Return the hash of the last block added to the chain

Examples:

```
>>>> libchain.lasthash()
blockhash
```

libchain.**poolblocks**(*_pool_name*)

When the name (case sensitive) of a mining pool is given, the latest blocks mined in that pool are returned as list

Examples:

```
>>>> libchain.poolblocks('AntPool')
['00000000000000000195caa1b5c7aa8bd94b0b8866e8fecf1d9ef61b1d5c97d',
 '00000000000000000034fce45130415766495ad4ee58daa8a1404d499b2c2203',
 '00000000000000000000616c5ae17d51c5d903c9b79c187c189849cd1b8bce161',
 ...]
```

libchain.**dayblocks**(*_time*)

When the time (in milliseconds) of a day is given, the blocks mined in that day are returned as list

Examples:

```
>>>> lc.dayblocks('1529176182')
[]
```

Note: I'm not sure this is working right now, need more investigation

libchain.**unconfirmedtx**()

Return a list of the latest unconfirmed transactions

Examples:

```
>>>> for tx in lc.unconfirmedtx():
...     print(tx)
...
hash:
2c6c97f5ee17fefbb01e6140458c039808ad803f0dccb4b02090fc4302af12cc
index:
354886645
senders:
1LQWp7eZRryr4DjsZHhGopV5aL1fSS94oX 7616394
13i994ZyvNqGEJxFoDcWbNjMqZjqoi2zFA 166133
recipients:
14rRznVHsaBYKed7iSBAP9vJRbu8YTaGhD 7714187
.
.
.
```

CHAPTER 5

License

libchain is distributed with the Apache v2 license:

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
```

(continues on next page)

(continued from previous page)

```
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
```

(continues on next page)

(continued from previous page)

```
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
```

(continues on next page)

(continued from previous page)

```
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

In addition, while a stable version is not ready, remember that this is an Alpha version so

Warning: Be aware that is up to you to take precaution against damages resulting from this software. Software in testing should for example not be used on sensitive and/or valuable data. The usage of this software is at your own risk and may void warranty on your products. If you are not an experienced user, or otherwise in in any doubt of your capabilities to make such a decision, please do not download and/or use this software.

5.1 Credits

I need to thank the developer of requests and the people of blockchain.info that made this library possible.

CHAPTER 6

libchain's documentation

Welcome to libchain's documentation, here you can find a minimal description of this library's classes and methods.

Hope you find what you're looking for and happy coding!

Note: This is a first release of the documentation, I haven't had enough time to complete it yet, but I hope it is enough to get started using **libchain**. Future and improved versions of **libchain** and it's documentation are planned.

Warning: **libchain** is still in **Alpha**, bugs are expected and documentation is not complete yet. Be aware that is up to you to take precaution against damages resulting from this software. Software in testing should for example not be used on sensitive and/or valuable data. The usage of this software is at your own risk and may void warranty on your products. If you are not an experienced user, or otherwise in in any doubt of your capabilities to make such a decision, please do not download and/or use this software.

Python Module Index

l

Index