

---

# py-flexpoolapi

Jan 25, 2021



---

## Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
<b>2</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



Pythonic library for easy interfacing with Flexpool Public API.



## 1.1 Quickstart

**py-flexpoolapi** is a pythonic library for easy interfacing with Flexpool Public API. The library has minimal dependencies which means it can be easily used inside every environment.

### 1.1.1 Installation

The recommended way is to install our PyPI package using **pip**.

```
pip3 install flexpoolapi
```

---

**Note:** You need to use `pip` instead of `pip3` if you're using Windows.

---

However, you can always build it from source by

```
git clone https://github.com/flexpool/py-flexpoolapi
cd py-flexpoolapi
make install
```

### 1.1.2 Usage

Great! The **py-flexpoolapi** is installed successfully, you can check if it works by

```
>>> import flexpoolapi
>>> flexpoolapi.pool.hashrate()
{'EU1': 21818049812367, 'US1': 19274829582345, 'total': 41092879394712}
```

### 1.1.3 Quick example

```
>>> import flexpoolapi

# Pool
>>> flexpoolapi.pool.hashrate()
{'EU1': 21818049812367, 'US1': 19274829582345, 'total': 41092879394712}
>>> flexpoolapi.pool.miners_online()
47192
>>> flexpoolapi.pool.workers_online()
253194

# Miner
>>> miner = flexpoolapi.miner("0x8B82eE62Ae306BF1bE085458a08241759d1d7E20")
>>> miner.balance()
575311819007598793
>>> effective_hashrate, reported_hashrate = miner.current_hashrate()
(532256937, 497730709)
```

For better understanding, we recommend reading the documentation fully. If you don't like reading documentation, you can always refer to the [Examples Directory on GitHub](#).

## 1.2 Pool Module

**class** flexpoolapi.pool  
The API's Pool Module bindings.

### 1.2.1 General statistics

pool.hashrate()

- Delegates to /pool/hashrate API Method

Returns pool's current effective hashrate in H/s (hashes per second). All data is splitted into servers (e.g. EU1, US2)

```
>>> flexpoolapi.pool.hashrate()
{'EU1': 21818049812367, 'US1': 19274829582345, 'total': 41092879394712}
```

pool.miners\_online()

- Delegates to /pool/minersOnline API Method

Returns the amount of miners currently mining on our pool.

```
>>> flexpoolapi.pool.miners_online()
47192
```

pool.workers\_online()

- Delegates to /pool/workersOnline API Method

Returns the amount of workers currently mining on our pool.

```
>>> flexpoolapi.pool.workers_online()
253194
```



`pool.block_count()`

- Delegates to `/pool/blockCount` API Method

Returns the amount of blocks that were mined by pool.

```
>>> flexpoolapi.pool.block_count()
528191
```

`pool.current_luck()`

- Delegates to `/pool/currentLuck` API Method

Returns current luck.

```
>>> flexpoolapi.pool.current_luck()
2.911822938883974 # 291%
```

`pool.avg_luck_roundtime()`

- Delegates to `/pool/avgLuckRoundtime` API Method

Returns average luck and round time.

---

**Hint:** Round time is the time between pool's blocks

---

```
>>> luck, round_time = flexpoolapi.pool.avg_luck_roundtime()
>>> luck
1.297540974974846 # 129%
>>> round_time
55.4 # 55.4 secs
```

## 1.2.2 Top Statistics

`pool.top_miners()`

- Delegates to `/pool/topMiners` API method

Returns top miners by hashrate (descending order).

```
>>> top_miners = flexpoolapi.pool.top_miners()
[<flexpoolapi.pool.TopMiner object 0xD7557BcC922E16D5248231Ee85919F5b01c97d12: 4.8 TH/s>, <flexpoolapi.pool.TopMiner object 0x31bfB275184Ce145B689ea79963c7b8ba5Fc5C99: 983.2 GH/s>, ...]

>>> top_miners[0]
<flexpoolapi.pool.TopMiner object 0xD7557BcC922E16D5248231Ee85919F5b01c97d12: 4.8 TH/s>

>>> top_miners[0].address
0xD7557BcC922E16D5248231Ee85919F5b01c97d12
>>> top_miners[0].hashrate
4832143791236
>>> top_miners[0].pool_donation
0.05
>>> top_miners[0].total_workers
24821
```

(continues on next page)

```
>>> top_miners[0].first_joined
datetime.datetime(2020, 5, 13, 20, 8, 7)
```

**References:**

*pool.TopMiner*

`pool.top_donators()`

- Delegates to `/pool/topDonators` API method

Returns top miners by total donated amount (descending order).

```
>>> top_donators = flexpoolapi.pool.top_donators()
[<flexpoolapi.pool.TopDonator object 0xD7557BcC922E16D5248231Ee85919F5b01c97d12:
↳534.1283 ETH>, <flexpoolapi.pool.TopDonator object
↳0xD7557BcC922E16D5248231Ee85919F5b01c97d12: 277.7074 ETH>, ...]

>>> top_donators[0]
<flexpoolapi.pool.TopDonator object 0xD7557BcC922E16D5248231Ee85919F5b01c97d12:
↳534.1283 ETH>

>>> top_donators[0].address
0xD7557BcC922E16D5248231Ee85919F5b01c97d12
>>> top_donators[0].pool_donation
0.05
>>> top_donators[0].total_donated
534.128394767847103826
>>> top_donators[0].first_joined
datetime.datetime(2020, 5, 13, 20, 8, 7)
```

**References:**

*pool.TopDonator*

## 1.2.3 Blocks

`pool.last_blocks(count=10)`

- Wraps paged `/pool/blocks` API method

Returns last N blocks mined by pool (descending order).

```
>>> last_blocks = flexpoolapi.pool.last_blocks(5)
[<flexpoolapi.shared.Block object Block #10208094 (0x4a916...0be99)>,
↳<flexpoolapi.shared.Block object Uncle #10156606 (0x262bb...1134d)>, ...
↳]
```

**References:**

*shared.Block*

`pool.blocks_paged(page: int)`

- Delegates to `/pool/blocks` API method

Returns paged response wrapped into `PageResponse` class (descending order).

---

**Hint:** There are 10 blocks per one page

---

```
>>> blocks_page_0 = flexpoolapi.pool.blocks_paged(page=0) # Get first 10 blocks
<flexpoolapi.shared.PageResponse object [ <flexpoolapi.shared.Block object Block
↪ #10208094 (0x4a916...0be99)>, <flexpoolapi.shared.Block object Uncle #10156606
↪ (0x262bb...1134d)>, <flexpoolapi.shared.Block object Block #9994360
↪ (0x1251a...6dad9)>, ... ]>
>>> blocks_page_0.contents
[ <flexpoolapi.shared.Block object Block #10208094 (0x4a916...0be99)>,
↪ <flexpoolapi.shared.Block object Uncle #10156606 (0x262bb...1134d)>,
↪ <flexpoolapi.shared.Block object Block #9994360 (0x1251a...6dad9)>, ... ]
blocks_page_0.total_items
528191
>>> blocks_page_0.total_pages
52820
>> blocks_page_0.items_per_page
10
```

**References:**

*shared.PageResponse*

*shared.Block*

## 1.2.4 Other

pool.**hashrate\_chart** ()

- Delegates to /pool/hashrateChart API Method

Returns history of pool hashrate wrapped into flexpoolapi.pool.HashrateChartItem classes.

**Example**

```
[
  <flexpoolapi.pool.HashrateChartItem (T)>,
  <flexpoolapi.pool.HashrateChartItem (T - 10m)>,
  <flexpoolapi.pool.HashrateChartItem (T - 20m)>,
  <flexpoolapi.pool.HashrateChartItem (T - 30m)>,
  ...
]
```

```
>>> hashrate_chart = flexpoolapi.pool.hashrate_chart()
[ <flexpoolapi.pool.HashrateChartItem object EU1 (21.7 TH/s), US1 (19.1 TH/s)>,
↪ <flexpoolapi.pool.HashrateChartItem object EU1 (21 TH/s), US1 (19.8 TH/s)>, ... ]

>>> hashrate_chart[0]
<flexpoolapi.pool.HashrateChartItem object EU1 (21.8 TH/s), US1 (19.1 TH/s)>
>>> hashrate_chart[0].servers
{'EU1': 21818049812367, 'US1': 19274829582345} # Pool's hashrate splitted by
↪ servers
>>> hashrate_chart[0].total_hashrate
41092879394712 # Total pool's hashrate
>>> hashrate_chart[0].timestamp
1592321400 # Chart item's Unix timestamp
```

**References:**

*pool.HashrateChartItem*

## 1.3 Miner Module

**class** flexpoolapi.miner

The API's Miner Module bindings.

### 1.3.1 Introduction

The API is wrapped into class for better interfacing. First you need to initialize your miner class by calling `miner = flexpoolapi.miner(<ADDRESS>)`, and then you would be able to access the API.

```
>>> miner = flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD")
>>> effective, reported = miner.current_hashrate()
>>> effective
532256937
>>> reported
497730709
```

---

**Note:** If the given address hasn't mined and doesn't exist in our database, the `flexpoolapi.exceptions.MinerDoesNotExist` exception would be risen.

---

### 1.3.2 Statistics

`miner.balance()`

- Delegates to `/miner/<ADDRESS>/balance` API Method

Returns miner's unpaid balance in weis (minimal ETH unit)

```
>>> flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").balance()
575311819007598793 # 0.57 ETH
```

`miner.current_hashrate()`

- Delegates to `/miner/<ADDRESS>/current` API Method

Returns miner's current hashrate in H/s (hashes per second).

```
>>> effective, reported = flexpoolapi.miner(
↳ "0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").current_hashrate()
>>> effective
532256937
>>> reported
497730709
```

`miner.daily_average_stats()`

- Delegates to `/miner/<ADDRESS>/daily` API Method

Returns miner's daily average hashrate and the amount of shares submitted during the day.

```

>>> stats = flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").daily_
↳average_stats()
<flexpoolapi.shared.DailyAverageStats object 486.3 MH/s>
>>> stats.effective_hashrate
486282826.8471824
>>> stats.reported_hashrate
497730709.1734889
>>> stats.valid_shares
10503
>>> stats.stale_shares
58
>> stats.invalid_shares
0

```

**References:**

*shared.DailyAverageStats*

miner.**stats**()

- Delegates to /miner/<ADDRESS>/stats API Method

Returns miner's current and daily average hashrate, and the amount of shares submitted during the day.

```

>>> stats = flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").
↳stats()
<flexpoolapi.shared.Stats object 486.3 MH/s>
>>> stats.current_effective_hashrate
486282826.8471824
>>> stats.average_effective_hashrate
466831513.7732951
>>> stats.current_reported_hashrate
517639937.54042846
>>> stats.average_reported_hashrate
497730709.1734889
>>> stats.valid_shares
10503
>>> stats.stale_shares
58
>> stats.invalid_shares
0

```

**References:**

*shared.Stats*

miner.**block\_count**()

- Delegates to /miner/<ADDRESS>/blockCount API Method

Returns the count of blocks mined by miner.

```

>>> flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").block_count()
2

```

miner.**details**()

- Delegates to /miner/<ADDRESS>/details API Method

Returns the miner details.

```
>>> details = flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").
↳details()
<flexpoolapi.miner.MinerDetails object_
↳(0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD)>
>>> details.min_payout_threshold
200000000000000000 # 0.2 ETH
>>> details.pool_donation
0.02
>>> details.first_joined_date
datetime.datetime(2020, 4, 30, 20, 50)
>>> details.censored_email
'mai*@exa****.com'
>>> details.censored_ip
'*. *.*.1'
```

**References:**

*miner.MinerDetails*

`miner.total_paid()`

- Delegates to `/miner/<ADDRESS>/totalPaid` API Method

Returns the value of weis paid to the miner.

```
>>> flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").total_paid()
719802600880251629
```

`miner.total_donated()`

- Delegates to `/miner/<ADDRESS>/totalDonated` API Method

Returns the value of weis donated by miner. (<3)

```
>>> flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").total_donated()
35990130044012584
```

`miner.estimated_daily_revenue()`

- Delegates to `/miner/<ADDRESS>/estimatedDailyRevenue` API Method

Returns miner's approximate daily profit (in weis).

```
>>> flexpoolapi.miner("0xAB1a87C524b5B302954a3ceE2685445C38b6B916").estimated_
↳daily_revenue()
128129842556298557
```

`miner.round_share()`

- Delegates to `/miner/<ADDRESS>/roundShare` API Method

Returns miner's round share (%).

```
>>> flexpoolapi.miner("0x25bC7A927ff4a2C4243752e0C577c6aFB740c291").round_share()
0.024 # 0.024%
```

### 1.3.3 Payments

`miner.payment_count()`

- Delegates to `/miner/<ADDRESS>/paymentCount` API Method

Returns the amount of payments done.

```
>>> stats = flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").
↳payment_count()
47
```

`miner.payments_paged` (*page: int*)

- Delegates to `/miner/<ADDRESS>/payments` API method

Returns paged response wrapped into `PagedResponse` class (descending order, latest first).

---

**Hint:** There are 10 payments per one page.

---

```
>>> payments_page_0 = flexpoolapi.miner(
↳"0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").payments_paged(page=0)
<flexpoolapi.shared.PageResponse object [
<flexpoolapi.miner.Transaction object 1.
↳61075 ETH (2020 Jun 06 14:12)>,
<flexpoolapi.miner.Transaction object 1.38525_
↳ETH (2020 May 30 00:20)>, ...]>
>>> blocks_page_0.contents
[<flexpoolapi.miner.Transaction object 1.61075 ETH (2020 Jun 06 14:12)>,
↳<flexpoolapi.miner.Transaction object 1.38525 ETH (2020 May 30 00:20)>, ...]
blocks_page_0.total_items
47
>>> blocks_page_0.total_pages
5
>> blocks_page_0.items_per_page
10
```

#### References:

*shared.PageResponse*

*miner.Transaction*

### 1.3.4 Blocks

`miner.blocks_paged` (*page: int*)

- Delegates to `/miner/<ADDRESS>/blocks` API method

Returns paged response wrapped into `PagedResponse` class (descending order, latest first).

---

**Hint:** There are 10 blocks per one page

---

```
>>> payments_page_0 = flexpoolapi.miner(
↳"0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").blocks_paged(page=0)
<flexpoolapi.shared.PageResponse object [
<flexpoolapi.shared.Block object Uncle
↳#10156606 (0x262bb...1134d)>,
<flexpoolapi.shared.Block object Block #9994360_
↳(0x1251a...6dad9)>, ...]>
>>> blocks_page_0.contents
[<flexpoolapi.shared.Block object Uncle #10156606 (0x262bb...1134d)>,
↳<flexpoolapi.shared.Block object Block #9994360 (0x1251a...6dad9)>]]
blocks_page_0.total_items
```

(continues on next page)

```

2
>>> blocks_page_0.total_pages
1
>> blocks_page_0.items_per_page
10

```

**References:***shared.PageResponse**shared.Block*

### 1.3.5 Other

`miner.chart()`

- Delegates to `/miner/<ADDRESS>/chart` API Method

Returns history of miner hashrate and shares wrapped into `flexpoolapi.shared.StatChartItem` classes.

**Example**

```

[
  <flexpoolapi.shared.StatChartItem (T)>,
  <flexpoolapi.shared.StatChartItem (T - 10m)>,
  <flexpoolapi.shared.StatChartItem (T - 20m)>,
  <flexpoolapi.shared.StatChartItem (T - 30m)>,
  ...
]

```

```

>>> chart = flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").
↳chart ()
[<flexpoolapi.shared.StatChartItem object (2020 Jun 17 12:40)>, <flexpoolapi.
↳shared.StatChartItem object (2020 Jun 17 12:30)>, ...]

>>> chart[0]
<flexpoolapi.shared.StatChartItem object (2020 Jun 17 12:40)>
>>> hashrate_chart[0].effective
497730709
>>> hashrate_chart[0].reported
532256937
>>> hashrate_chart[0].valid_shares
72
>>> hashrate_chart[0].stale_shares
1
>>> hashrate_chart[0].invalid_shares
0

```

**References:***shared.StatChartItem*



## 1.4 Worker Module

**class** flexpoolapi.worker

The API's Worker Module bindings.

### 1.4.1 Introduction

As py-flexpoolapi is coded in Pythonic Style, the worker API is mounted into the Miner module directly.

```
>>> miner = flexpoolapi.miner("0xa598f8fb0a44eF74357815e318dC1C48719Fc3AD")
>>> workers = miner.workers()
[<flexpoolapi.worker.Worker object rig1 (0xa598f8fb0a44eF74357815e318dC1C48719Fc3AD)>,
 ↪ <flexpoolapi.worker.Worker object rig2_
 ↪ (0xa598f8fb0a44eF74357815e318dC1C48719Fc3AD)>, ...]
>>> workers[0].worker_name
'rig1'
>>> workers[0].is_online
True
>>> workers[0].last_seen_date
time.datetime(2020, 6, 17, 12, 13, 25)
```

**Warning:** The flexpoolapi.worker.Worker class is intended to be initialized ONLY by miner.workers().

### 1.4.2 Statistics

worker.**current\_hashrate** (*page: int*)

- Delegates to /worker/<MINER\_ADDRESS>/<WORKER\_NAME>/current API method

Returns worker's current hashrate.

```
>>> worker = flexpoolapi.miner("0xa598f8fb0a44eF74357815e318dC1C48719Fc3AD").
 ↪workers()[0]
<flexpoolapi.worker.Worker object rig1_
 ↪ (0xa598f8fb0a44eF74357815e318dC1C48719Fc3AD)>
>>> effective, reported = worker.current_hashrate()
>>> effective
164963909 # 165 MH/s
>>> reported
196102107 # 196.1 MH/s
```

worker.**daily\_average\_stats**()

- Delegates to /woker/<MINER\_ADDRESS>/<WORKER\_NAME>/daily API Method

Returns workers's daily average hashrate and the amount of shares submitted during the day. Same as miner.daily\_average\_stats().

```
>>> stats = flexpoolapi.miner("0xa598f8fb0a44eF74357815e318dC1C48719Fc3AD").
 ↪workers()[0].daily_average_stats()
<flexpoolapi.shared.DailyAverageStats object 121.6 MH/s>
>>> stats.effective_hashrate
121570706.7117956
```

(continues on next page)

(continued from previous page)

```

>>> stats.reported_hashrate
124432677.29337223
>>> stats.valid_shares
2625
>>> stats.stale_shares
12
>> stats.invalid_shares
0

```

flexpoolapi.shared.DailyAverageStats reference: <TODO/TBD>

worker.**stats**()

- Delegates to /worker/<MINER\_ADDRESS>/<WORKER\_NAME>/stats API Method

Returns workers's current and daily average hashrate, and the amount of shares submitted during the day. Same as miner.stats().

```

>>> stats = flexpoolapi.miner("0xa598f8fb0a44eF74357815e318dC1C48719Fc3AD").
↳workers()[0].stats()
<flexpoolapi.shared.Stats object 121.6 MH/s>
>>> stats.current_effective_hashrate
121592946.2467181
>>> stats.average_effective_hashrate
121570706.7117956
>>> stats.current_reported_hashrate
128733972.73389934
>>> stats.average_reported_hashrate
124432677.29337223
>>> stats.valid_shares
2625
>>> stats.stale_shares
12
>> stats.invalid_shares
0

```

flexpoolapi.shared.Stats reference: <TODO/TBD>

### 1.4.3 Other

worker.**chart**()

- Delegates to /worker/<MINER\_ADDRESS>/<WORKER\_NAME>/chart API Method

Returns history of miner hashrate and shares wrapped into flexpoolapi.shared.StatChartItem classes.

#### Example

```

[
  <flexpoolapi.shared.StatChartItem (T)>,
  <flexpoolapi.shared.StatChartItem (T - 10m)>,
  <flexpoolapi.shared.StatChartItem (T - 20m)>,
  <flexpoolapi.shared.StatChartItem (T - 30m)>,
  ...
]

```

```

>>> chart = flexpoolapi.miner("0xa598f8fB0a44eF74357815e318dC1C48719Fc3AD").
↳chart()
[<flexpoolapi.shared.StatChartItem object (2020 Jun 17 12:40)>, <flexpoolapi.
↳shared.StatChartItem object (2020 Jun 17 12:30)>, ...]

>>> chart[0]
<flexpoolapi.shared.StatChartItem object (2020 Jun 17 12:40)>
>>> hashrate_chart[0].effective
124432677
>>> hashrate_chart[0].reported
133064234
>>> hashrate_chart[0].valid_shares
15
>>> hashrate_chart[0].stale_shares
1
>>> hashrate_chart[0].invalid_shares
0

```

flexpoolapi.shared.StatChartItem reference: <TODO/TBD>

## 1.5 Classes

For easy interfacing, all structured responses are wrapped into classes.

### 1.5.1 Shared

These classes are used all around the **py-flexpoolapi**

### 1.5.2 shared.Block

**class** flexpoolapi.shared.**Block**

#### Properties

- **number:** *int*  
Block number (aka block height).
- **hash:** *str*  
Block hash.
- **type:** *str*  
Block type. One of ['block', 'uncle', 'orphan']
- **miner:** *str*  
Block miner's checksummed Ethereum address.  
(e.g 0x3f7d727B73A30E8BF801760F59962C15a56DbA74)
- **difficulty:** *int*  
Block difficulty.  
(e.g. 2312863022593449)

- **time:** *datetime.datetime*  
Date when block was mined.
- **is\_confirmed:** *bool*  
Block confirmation.
- **round\_time:** *int*  
Round time in seconds (e.g 153)
- **luck:** *float*  
Block luck.  
(e.g. 1.743672971, which is 174%)
- **server\_name:** *str*  
Name of server that mined this block.  
(e.g. EU1, US1)
- **block\_reward:** *int*  
Block reward in wei.  
(e.g. 20000000000000000000)
- **block\_fees:** *int*  
Block fees in wei.
- **uncle\_inclusion\_rewards:** *int*  
Uncle inclusion rewards (e.g 625000000000000000, which is 0.0625 ETH)
- **total\_rewards:** *int*  
Sum of all rewards.

### 1.5.3 shared.Stats

**class** flexpoolapi.shared.Stats

#### Properties

- **current\_effective\_hashrate:** *int*  
Current effective hashrate
- **current\_reported\_hashrate:** *int*  
Current reported hashrate
- **average\_effective\_hashrate:** *int*  
24h average effective hashrate
- **average\_reported\_hashrate:** *int*  
24h average reported hashrate
- **valid\_shares:** *int*  
Total amount of valid shares
- **stale\_shares:** *int*

Total amount of stale shares

- **invalid\_shares:** *int*

Total amount of invalid shares

### 1.5.4 shared.DailyAverageStats

**class** flexpoolapi.shared.DailyAverageStats

- **effective\_hashrate:** *int*

24h average effective hashrate

- **reported\_hashrate:** *int*

24h average reported hashrate

- **valid\_shares:** *int*

Amount of valid shares submitted during the day.

- **stale\_shares:** *int*

Amount of stale shares submitted during the day.

- **invalid\_shares:** *int*

Amount of invalid shares submitted during the day.

### 1.5.5 shared.StatChartItem

**class** flexpoolapi.shared.StatChartItem

- **effective\_hashrate:** *int*

Effective hashrate

- **reported\_hashrate:** *int*

Reported hashrate

- **valid\_shares:** *int*

Amount of valid shares submitted during the item time (10min).

- **stale\_shares:** *int*

Amount of stale shares submitted during the item time (10min).

- **invalid\_shares:** *int*

Amount of invalid shares submitted during the item time (10min).

### 1.5.6 shared.PageResponse

**class** flexpoolapi.shared.PageResponse

---

**Note:** This class can be used as a regular list.

```
>>> len(page_response)
'<amount of items>'
>>> page_response[0]
'<item with index 0>'
>>> page_response[1]
'<item with index 1>'
```

### Properties

- **contents:** [*<any>*]  
Stores the PageResponse items.
- **total\_items:** *int*  
The amount of items in contents. Cannot be bigger than `items_per_page`.
- **total\_pages:** *int*  
The total amount of pages.
- **items\_per\_page:** *int*  
Amount of items per page. Normally equals to **10**.

## 1.5.7 Pool

Classes used by `flexpoolapi.pool` namespace.

## 1.5.8 pool.TopMiner

**class** `flexpoolapi.pool.TopMiner`

### Properties

- **address:** *str*  
Checksummed Miner's Ethereum address.
- **hashrate:** *int*  
Miner's current effective hashrate.
- **pool\_donation:** *int*  
Miner's Pool Donation.  
(e.g. 0.05, which is 5%)
- **total\_workers:** *int*  
The amount of workers.
- **first\_joined:** *datetime.datetime*  
Date when the miner firstly mined on pool.

### 1.5.9 pool.TopDonator

```
class flexpoolapi.pool.TopDonator
```

#### Properties

- **address:** *str*  
Checksummed Miner's Ethereum address.
- **total\_donated:** *int*  
Total donated amount (weis).  
(e.g. 528498812374981273489, which is 528.5 ETH)
- **pool\_donation:** *int*  
Miner's Pool Donation.  
(e.g. 0.05, which is 5%)
- **hashrate:** *int*  
Miner's current effective hashrate.
- **first\_joined:** *datetime.datetime*  
Date when the miner firstly mined on pool.

### 1.5.10 pool.HashrateChartItem

```
class flexpoolapi.pool.HashrateChartItem
```

#### Properties

- **servers:** *{str: int}*  
Pool hashrate splitted by servers. A dictionary with {"server": <hashrate>} scheme.
- **total\_hashrate:** *int*  
Total pool's hashrate.
- **timestamp:** *int*  
Unix timestamp of item.

### 1.5.11 Miner

Classes used by `flexpoolapi.miner` class.

### 1.5.12 miner.MinerDetails

```
class flexpoolapi.miner.MinerDetails
```

#### Properties

- **addresss:** *str*  
The Miner's Ethereum address.
- **min\_payout\_threshold:** *int*

Miner's minimal payout threshold (represented in wei).

(e.g. 200000000000000000, which is 0.2 ETH)

- **pool\_donation:** *int*

Miner's pool donation.

(e.g. 0.05, which is 5%)

- **censored\_email:** *int*

Miner's censored email.

(e.g. mai\*@exa\*\*\*\*.com)

- **censored\_ip:** *int*

Miner's censored IP address.

(e.g. \*.\*.\*.1)

- **first\_joined\_date:** *datetime.datetime*

Date when miner's firstly mined on the pool.

### 1.5.13 miner.Transaction

**class** flexpoolapi.miner.Transaction

#### Properties

- **amount:** *int*

Transaction value (represented in wei).

(e.g. 912347012097312304, which is 0.91 ETH)

- **time:** *datetime.datetime*

Time when the transaction was sent.

- **duration:** *int*

The duration between current and previous payout (secs). Equals to 0 if it is the first payout.

- **txid:** *str*

Transaction hash.

(e.g. 3d02b5f888169e8ab55ae39a8f93eeab1f24703081798c61ac1a390d1b2e909b)



## CHAPTER 2

---

### Indices and tables

---

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



**A**

avg\_luck\_roundtime() (*pool method*), 5

**B**

balance() (*miner method*), 8

block\_count() (*miner method*), 9

block\_count() (*pool method*), 4

blocks\_paged() (*miner method*), 11

blocks\_paged() (*pool method*), 6

**C**

chart() (*miner method*), 12

chart() (*worker method*), 14

current\_hashrate() (*miner method*), 8

current\_hashrate() (*worker method*), 13

current\_luck() (*pool method*), 5

**D**

daily\_average\_stats() (*miner method*), 8

daily\_average\_stats() (*worker method*), 13

details() (*miner method*), 9

**E**

estimated\_daily\_revenue() (*miner method*), 10

**F**

flexpoolapi.miner (*built-in class*), 8

flexpoolapi.miner.MinerDetails (*built-in class*), 19

flexpoolapi.miner.Transaction (*built-in class*), 20

flexpoolapi.pool (*built-in class*), 4

flexpoolapi.pool.HashrateChartItem (*built-in class*), 19

flexpoolapi.pool.TopDonator (*built-in class*), 19

flexpoolapi.pool.TopMiner (*built-in class*), 18

flexpoolapi.shared.Block (*built-in class*), 15

flexpoolapi.shared.DailyAverageStats (*built-in class*), 17

flexpoolapi.shared.PageResponse (*built-in class*), 17

flexpoolapi.shared.StatChartItem (*built-in class*), 17

flexpoolapi.shared.Stats (*built-in class*), 16

flexpoolapi.worker (*built-in class*), 13

**H**

hashrate() (*pool method*), 4

hashrate\_chart() (*pool method*), 7

**L**

last\_blocks() (*pool method*), 6

**M**

miners\_online() (*pool method*), 4

**P**

payment\_count() (*miner method*), 10

payments\_paged() (*miner method*), 11

**R**

round\_share() (*miner method*), 10

**S**

stats() (*miner method*), 9

stats() (*worker method*), 14

**T**

top\_donators() (*pool method*), 6

top\_miners() (*pool method*), 5

total\_donated() (*miner method*), 10

total\_paid() (*miner method*), 10

**W**

workers\_online() (*pool method*), 4