

---

# Wealthsimple-Trade-Python

*Release 1.0.0*

Mar 14, 2021



---

## Contents

---

<b>1</b>	<b>The Wealthsimple-Trade-Python Project:</b>	<b>3</b>
1.1	Wealthsimple Trade Python . . . . .	3
1.2	Contributing . . . . .	4
1.3	Authors . . . . .	5
1.4	Changelog . . . . .	6
<b>2</b>	<b>Code Documentation</b>	<b>7</b>
2.1	wealthsimple package . . . . .	7
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



A convenient Python wrapper for the Wealthsimple Trade API. Note that this wrapper is Unofficial and is not in any way affiliated with Wealthsimple. Please use at your own risk.



---

## The Wealthsimple-Trade-Python Project:

---

### 1.1 Wealthsimple Trade Python

A convenient Python wrapper for the Wealthsimple Trade API. Note that this wrapper is Unofficial and is not in any way affiliated with Wealthsimple. Please use at your own risk.

docs	
tests	
package	

#### 1.1.1 Installation

```
pip install wealthsimple-trade-python
```

You can also install the in-development version with:

```
pip install https://github.com/seansullivan44/Wealthsimple-Trade-Python/archive/  
↪master.zip
```

*Note:* `node` is a dependency of this project. See [here](<https://github.com/VeNoMouS/cloudscraper#dependencies>) for more information.

## 1.1.2 Getting Started

Download the Wealthsimple Trade app for iOS or Android and create an account. This API wrapper will use your Wealthsimple Trade login credentials to make successful API calls. After creating an account, use your login credentials to create a *WSTrade* object:

```
import wealthsimple
WS = wealthsimple.WSTrade('email', 'password')
```

If your Wealthsimple Trade account uses two-factor authentication then you must provide the *WSTrade* object with a callback function as shown in the following example:

```
import wealthsimple

def my_two_factor_function():
    MFACode = ""
    while not MFACode:
        # Obtain user input and ensure it is not empty
        MFACode = input("Enter 2FA code: ")
    return MFACode

ws = wealthsimple.WSTrade(
    "email",
    "password",
    two_factor_callback=my_two_factor_function,
)
```

## 1.1.3 Documentation

<https://Wealthsimple-Trade-Python.readthedocs.io/>

## 1.2 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

### 1.2.1 Bug reports

When 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.

### 1.2.2 Documentation improvements

Wealthsimple-Trade-Python could always use more documentation, whether as part of the official Wealthsimple-Trade-Python docs, in docstrings, or even on the web in blog posts, articles, and such.



### 1.2.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/seansullivan44/Wealthsimple-Trade-Python/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 code contributions are welcome :)

### 1.2.4 Development

To set up *Wealthsimple-Trade-Python* for local development:

1. Fork [Wealthsimple-Trade-Python](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:seansullivan44/Wealthsimple-Trade-Python.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. 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
```

5. Submit a pull request through the GitHub website.

### Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Update documentation when there’s new API, functionality etc.
2. Add a note to `CHANGELOG.rst` about the changes.
3. Add yourself to `AUTHORS.rst`.

## 1.3 Authors

### 1.3.1 Sean Sullivan (Project Creator)

- Carleton University - Electrical Engineering
- Email: [seansullivan3@cmail.carleton.ca](mailto:seansullivan3@cmail.carleton.ca)
- Website: [seanmcsullivan.ca](http://seanmcsullivan.ca)

### 1.3.2 NicBT

- GitHub: <https://github.com/NicBT>

### 1.3.3 Joey Orlando

- GitHub: <https://github.com/joeyorlando>

## 1.4 Changelog

### 1.4.1 1.1.0 (2021-03-14)

- Swap out *requests.Session* with *cloudscraper.CloudScraper* instance to overcome recent security changes related to Cloudflare

### 1.4.2 1.0.0 (2020-06-24)

- First release on PyPI.

## 2.1 wealthsimple package

### 2.1.1 Submodules

### 2.1.2 wealthsimple.requestor module

**class** wealthsimple.requestor.**APIRequestor** (*session*, *APIMainURL*)

Bases: object

A class to simplify request calls to REST API

**session** [session] A requests Session object to be associated with the class

**APIMainURL** [str] Main URL endpoint for API

**makeRequest(method, endpoint, params=None, returnValue=None)** Make a request to a given API endpoint

**post(URL, params=None)** Make a POST request to a given API endpoint

**get(URL, params=None)** Make a GET request to a given API endpoint

**get** (*URL*, *params=None*)

Make a GET request to a given API endpoint

**URL** [str] Full URL endpoint of API

**params** [dict] Dictionary of parameters to be passed with request

**Response** [Response] A requests response object

**makeRequest** (*method*, *endpoint*, *params=None*, *returnValue=None*)

Make a request to a given API endpoint

**method** [str] Specify POST or GET request

**endpoint** [str] URL endpoint of API (Does not include base URL)

**params** [dict] Dictionary of parameters to be passed with request

**Response** [Response] A requests response object

**post** (*URL, params=None*)

Make a POST request to a given API endpoint

**URL** [str] Full URL endpoint of API

**params** [dict] Dictionary of parameters to be passed with request

**Response** [Response] A requests response object

### 2.1.3 wealthsimple.wealthsimple module

**class** `wealthsimple.wealthsimple.WSTrade` (*email: str, password: str, two\_factor\_callback: callable = None*)

Bases: `object`

Wealthsimple Trade API wrapper

**session** [session] A requests Session object to be associated with the class

**APIMainURL** [str] Main URL endpoint for API

**TradeAPI** [APIRequester] APIRequester object to handle API calls

**login(email=None, password=None, two\_factor\_callback=None)** Login to Wealthsimple Trade account

**get\_accounts()** Get Wealthsimple Trade accounts

**get\_account\_ids()** Get Wealthsimple Trade account ids

**get\_account(id)** Get a Wealthsimple Trade account given an id

**get\_account\_history(id, time="all")** Get Wealthsimple Trade account history

**get\_activities()** Get Wealthsimple Trade activities

**get\_orders(symbol=None)** Get Wealthsimple Trade orders

**get\_security(symbol)** Get information about a security

**get\_positions(id)** Get positions

**get\_person()** Get Wealthsimple Trade person object

**get\_me()** Get Wealthsimple Trade user object

**get\_bank\_accounts():** Get list of bank accounts tied to Wealthsimple Trade account

**get\_deposits()** Get list of deposits

**get\_forex()** Get foreign exchange rate

**get\_account** (*id: str*) → dict

Get a Wealthsimple Trade account given an id

**id** [str] Wealthsimple Trade account id

**dict** A dictionary containing the Wealthsimple Trade account

**get\_account\_history** (*id: str, time: str = 'all'*) → dict  
Get Wealthsimple Trade account history

**id** [str] Wealthsimple Trade account id

**time** [str] String containing time interval for history

**dict** A dictionary containing the historical Trade account data

**get\_account\_ids** () → list  
Get Wealthsimple Trade account ids

**list** A list of Wealthsimple Trade account ids

**get\_accounts** () → list  
Get Wealthsimple Trade accounts

**list** A list of Wealthsimple Trade account dictionary objects

**get\_activities** () → list  
Get Wealthsimple Trade activities

**list** A list of dictionaries containing Wealthsimple Trade activities

**get\_bank\_accounts** () → list  
Get list of bank accounts tied to Wealthsimple Trade account

**list** A list of dictionaries containing bank account objects

**get\_deposits** () → list  
Get list of deposits

**list** A list of dictionaries containing deposit objects

**get\_forex** () → dict  
Get foreign exchange rate

**dict** A dictionary containing foreign exchange rates

**get\_me** () → dict  
Get Wealthsimple Trade user object

**dict** A dictionary containing a user object

**get\_orders** (*symbol: str = None*) → list  
Get Wealthsimple Trade orders

**symbol** [str] Symbol for security to filter orders on

**list** A list containing Wealthsimple Trade order dictionaries

**get\_person** () → dict  
Get Wealthsimple Trade person object

**dict** A dictionary containing a person object

**get\_positions** (*id: str*) → list  
Get positions

**id** [str] Wealthsimple Trade account id

**list** A list containing positions

**get\_securities\_from\_ticker** (*symbol: str*) → list  
Get information about a securitys with matching ticker symbols

**symbol** [str] Symbol for security to search on

**list** A list containing matching securities

**get\_security** (*id: str*) → dict  
Get information about a security

**id** [str] Wealthsimple Security ID to search on

**dict** Dictionary containing information for security

**login** (*email: str = None, password: str = None, two\_factor\_callback: callable = None*) → None  
Login to Wealthsimple Trade account

**email** [str] Wealthsimple Trade account email

**password** [str] Wealthsimple Trade account password

**two\_factor\_callback: function** Callback function that returns user input for 2FA code

None

## 2.1.4 Module contents

### W

`wealthsimple`, [10](#)

`wealthsimple.requestor`, [7](#)

`wealthsimple.wealthsimple`, [8](#)





## A

APIRequestor (class in *wealthsimple.requestor*), 7

## G

get () (*wealthsimple.requestor.APIRequestor* method), 7

get\_account () (*wealthsimple.wealthsimple.WSTrade* method), 8

get\_account\_history () (*wealthsimple.wealthsimple.WSTrade* method), 8

get\_account\_ids () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_accounts () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_activities () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_bank\_accounts () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_deposits () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_forex () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_me () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_orders () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_person () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_positions () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_securities\_from\_ticker () (*wealthsimple.wealthsimple.WSTrade* method), 9

get\_security () (*wealthsimple.wealthsimple.WSTrade* method), 10

## L

login () (*wealthsimple.wealthsimple.WSTrade* method), 10

## M

makeRequest () (*wealthsimple.requestor.APIRequestor* method), 7

## P

post () (*wealthsimple.requestor.APIRequestor* method), 8

## W

wealthsimple (module), 10

wealthsimple.requestor (module), 7

wealthsimple.wealthsimple (module), 8

WSTrade (class in *wealthsimple.wealthsimple*), 8