UTC 2021 - Setup Instructions & Case Details

Hello everyone! We're the UChicago Trading Competition casewriting and platform development team. This document contains all of the information you need to set up your environment and start developing bots for cases 1 and 2. **We recommend that you read this entire document from start to finish and follow the instructions carefully. If you have any questions about anything in this document, please do not hesitate to contact us via Piazza**.

Supported Software

We will be providing technical support only for competitors who write their bots in the Python programming language and use Visual Studio Code as their editor. Instructions on how to install these two pieces of software are below. It is possible to use other programming languages, but significantly more difficult.

We also expect a very basic level of familiarity with the command prompt (if you have a Windows computer) or terminal (if you have a Mac). If you are not yet familiar with these tools, there are a number of very helpful tutorials available online.

Python 3.7+ Installation

We ask that you install Python version 3.7 or later. To get the download links for your operating system, visit https://www.python.org/downloads/.

After installing Python, you should open a new command prompt/terminal and type

```
python3 --version
```

NOTE: on Windows, you may need to replace `python3` with `py`.

> - **If you see an error message instead**, it is likely that you need to add Python to your PATH environment variable. There are a number of resources online that detail how to do this. If the error persists, contact the platform developers.
> - **If the version number you see is less than 3.7.0**, you will need to upgrade your Python installation. In this case, download the latest version of Python from the link above, and the installer should automatically upgrade your version of Python.

Visual Studio Code Installation

To download VS Code, please visit https://code.visualstudio.com/.

After downloading the editor, open the folder containing this file. The editor should automatically suggest that the Python and Pylance extensions be installed. We strongly recommend that you install these--they will offer a much easier development experience once you get started working on your bots.

Platform Setup

The platform for the trading competition consists of two main parts:

- The *clients*, which are the bots that you will write that will trade against each other and the market
- The *exchange*, which the clients will connect to send order request/receive data feeds about the market

Some notes on how to set up and use these parts follow.

Client Setup & Usage

When you first download a new version of the platform, you will need to set it up before using it. To do this, navigate to the directory containing the platform (using the `cd` command) and run

```
python3 setup-xchange.py
```

NOTE: on Windows, you may need to replace `python3` with `py`.

If all goes well, a message saying **Everything is set up!** should appear.

> **If you see red text that says Error: ...**, then read the error message and follow the corresponding instructions. If errors persist, please contact the platform team via Piazza.

After running the above command successfully, a new *virtual environment* will have been set up in the directory. For more details on what a virtual environment is and how to use them, please visit https://docs.python.org/3/tutorial/venv.html.

To activate this virtual environment, type:

- `.\venv\Scripts\activate` (if you're on Windows)
- `source venv/bin/activate` (if you're on Mac/Linux)

If this is successful, there should be a `(venv)` before your command line, which indicates that the environment is currently active. **You should make sure to always run your bots with the environment active**.

This virtual environment comes pre-installed with some useful packages:

- `numpy` - A very popular scientific computing package
- `pandas` - A very popular data analysis package
- `scipy` - A popular scientific computing package with extended functionality
- `py_vollib` - A package for computing option prices, greeks, and implied volatilities
- `betterproto` - An interface with protocol buffers, which the bot uses under the hood to communicate with the platform

While in the virtual environment, you can run a bot by running

```
python bot_name.py
```

Exchange Details

This folder contains 3 copies of the exchange, one for each major operating system. **For the remainder of the document, we will use `<xchange>` to refer to**:

- `.\xchange-win.exe` if you're on Windows
- `./xchange-mac` if you're on Mac
- `./xchange-linux` if you're on Linux

For details on how each of these applications work, run

```
<xchange> -help
```

Details about how to use the platform for the cases will follow.

Getting Started!

In order to test your bots before the competition, you will need to run both your bot and the exchange *at the same time* on your computer. This means that you will have to open two different command line windows, one in which the virtual environment is activated (in which you can run your bot), and one where you will run the exchange. While the exchange is running, you can visit http://localhost:8080 in your browser for a real-time display of what is occurring in the markets.

Details on how to run the exchange for each case are in the following sections

Case 1

Instructions

Before the exchange has started, begin the execution of your bot in a separate command line window. To test that everything is working, you can start by running the example bot by typing

```
python clients/example_bot_case1.py
```

After your bot has been started, it will wait to connect to the exchange. To run the exchange for Case 1, open a separate command line window and execute

```
<xchange> case1
```

so that both your bot and the exchange are running at the same time. Soon after this, your bot should print a message saying that it connected to the exchange, and it will start running.

Case 1 comes with different example price/announcement paths that you can test your bots on. Your options range from 2011 to 2020 inclusive, which correspond to the interest rate and announcement data of that year. The price path of choice can be specified with the `-pricepath` flag, e.g.

```
<xchange> -pricepath=2012 case1
```

will run the price path of 2012.

We strongly recommend that you start your bot development by taking a close look at the example bot and understanding how it works. It provides a good example of how to interact with the exchange for this case.

Data

The data file will contain two important CSVs. All times in both CSVs are indexed with the number of days since January 1, 2011 (e.g. day 0 is January 1, 2011, day 253 is January 2, 2012)

- `/data/announcements.csv` - This contains all FOMC announcements spanning from from 2011 to 2020. Note that there are 50, 60, and 80 observations for ROR, HAP, and USD respectively per year (review the case packet if this is confusing). The first observation is the FOMC announcement from December 31, 2010 revealed on January 1, 2011. The announcement for December 31, 2020 (not in csv) will be revealed on January 1, 2021 at the start of the competition.
- `/data/interest_rates.csv` - This contains the daily realized Federal Funds interest rates with the same correspondence. These are critical to predict the exchange rates and futures contract rates. As of December 31, 2010, the currency exchange rates are

  ```
  ROR/USD = 0.25
  HAP/USD = 0.5
  HAP/ROR = 2
  ```

You can read these files, but **do not modify them!!** Doing so will cause the exchange for case 1 to stop working, and you will have to redownload the original data.

During the case, your updates will come in two forms. One will be central bank announcements and the other will be daily interest rate announcements.

Central Bank announcements are of the format: `"{Currency Name} New Federal Funds Target {Float FF Target}"` Interest Rate Announcements are of the format: `"{Day of Year}, {ROR Interest Rate}, {HAP Interest Rate}, {USD Interest Rate}"`

Case 2

Instructions

Before the exchange has started, begin the execution of your bot in a separate command line window. To test that everything is working, you can start by running the example bot by typing

```
python clients/example_bot_case2.py
```

After your bot has been started, it will wait to connect to the exchange. To run the exchange for Case 2, open a separate command line window and execute

```
<xchange> case2
```

so that both your bot and the exchange are running at the same time. Soon after this, your bot should print a message saying that it connected to the exchange, and it will start running.

Case 2 comes with 100 different example underlying paths that you can test your bots on. These paths are numbered 0 through 99, inclusive. The price path of choice can be specified with the `-pricepath` flag, e.g.

```
<xchange> -pricepath=45 case2
```

Will run price path 45.

We strongly recommend that you start your bot development by taking a close look at the example bot and understanding how it works. It provides a good example of how to interact with the exchange for this case.

Data

The price paths used by the exchange can be found as CSVs in the `data/case2/` folder. They numbered using the same scheme as is used to specify the price paths.

Questions

If you encounter any issues at all with the platform or cases while working on the competition, please do not hesitate to contact the casewriters and/or platform team via Piazza.

If the question or problem is technical in nature, please run the `<xchange>` executable with the `-debug` flag. Once the problem has been reproduced, use `Ctrl+C` to terminate the program, zip the `xchange-logs` folder, and attach it to your Piazza post. This will allow the team to walk through what is going on with the exchange and diagnose any problems that may be occurring.