**Superpy User Guide**

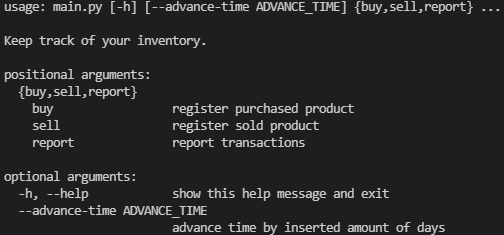
1. **Introduction**

SuperPy is a command-line tool which helps us to keepi track of our inventory, it can be run through the terminal by using various command-lines. Additional required files will automatically be created after running SuperPy for the first time.

We can execute the following command-line to review our command line options:



*Output:*



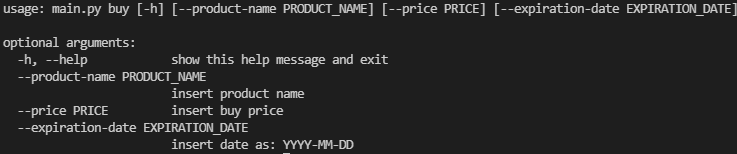
All these options will be further explained in this guide with the inclusion of examples.

1. **Buying products**

We can execute the following command-line to review the required buy arguments:

****

*Output:*



**Example 1**

If we want to buy an apple with a price of 3 euros and an expiration date of 2021-05-11, we can execute the following command-line:

**

*Output:*



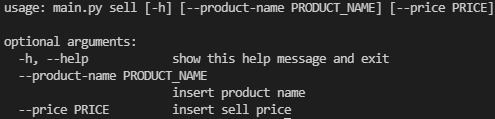
Note how the buy data is stored in file ‘bought.csv’.

1. **Selling products**

We can execute the following command-line to review the required sell arguments:

****

*Output:*

****

**Example 1**

If we want to sell an apple for 4 euros, we can execute the following command-line:

****

*Output:*

****

Note how the sell data is stored in file ‘sold.csv’.

**Example 2**

But what will happen if we want to sell a product that’s currently not in stock? Or what if the product we do have in stock has already expired? SuperPy will return the following output:

****

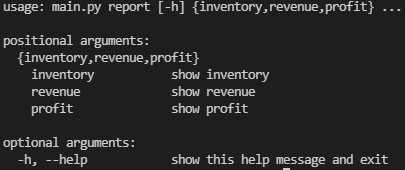
1. **Reports**

SuperPy is also able to generate various reports: inventory, revenue and profit.

We can execute the following command-line to review the required report arguments/options:



*Output:*

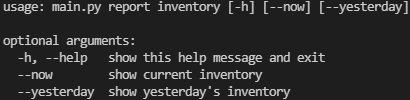
****

* 1. **Inventory**

We can execute the following command-line to review the required inventory arguments/options:

****

*Output:*

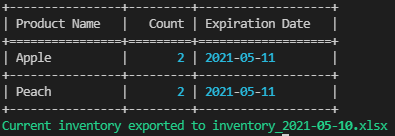
****

**Example 1**

If we want to generate a report of our current inventory, we can execute the following command-line:



*Output:*

****

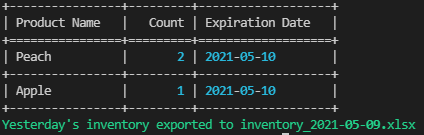
Note how the inventory report has also been exported to an .xlsx file with today’s date as filename.

**Example 2**

If we want to generate a report of yesterday’s inventory, we can execute the following command-line:

****

*Output:*



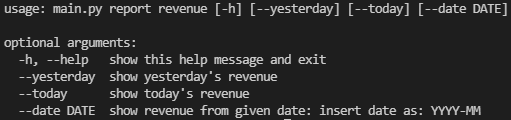
Note how the inventory report has also been exported to an .xlsx file with yesterday’s date as filename.

* 1. **Revenue**

We can execute the following command-line to review the required revenue arguments/options:



*Output:*

****

**Example 1**

If we want to know today’s revenue, we can execute the following command-line:



*Output:*

****

**Example 2**

If we want to know yesterday’s revenue, we can execute the following command-line:



*Output:*

****

**Example 3**

If we want to know the revenue of a certain month, we can execute the following command-line:



*Output:*

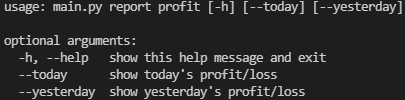


* 1. **Profit**

We can execute the following command-line to review the required inventory arguments/options:

****

*Output:*

****

**Example 1**

If we want to know today’s profit/loss, we can execute the following command-line:



*Output 1:*

**

*Output 2:*



**Example 2**

If we want to know yesterday’s profit/loss, we can execute the following command-line:

****

*Output 1:*



*Output 2:*



1. **Advance time**

SuperPy also has the ability to time travel. It can change the system date by a user inserted amount of days.

**Example**

If we want to jump to a week from now for whatever reason, we can execute the following command-line:



*Output*

**