Company Returns API Specification

Version:	8
Date Modified:	29 March 2018

1. The context

We are a company specialising in commodities trading. We are developing a complex system to provide an analytics platform for our traders. We wish to open our system to third party software houses, offering them ability to “plug-in” independent software modules that implement particular functions. We therefore request all interested companies to provide an independent software module that implements an API as specified in the rest of this document.

2. Functionality of the Company Returns API

The calculating returns is a simple but powerful method that can help researchers, traders and fund managers assess the variation of a firm’s value. Using this method, they determine whether there is an "abnormal" stock price effect. A method to calculate returns is relatively easy to implement, because the only data necessary are the names of publicly traded firms, a date of interest, and stock prices.

1.1. Stock Return

In finance, return is the profit on any investment. For instance, an investment of $100 on 1 Jan 2017, that was worth $120 on 31 Dec 2017 had a profit of $(120-100) = $20 in 2017. Return can also be measured as the percentage of profit over the initial investment. The later definition is also referred to as rate of return. In the above example the return (rate of return) in percentage is $20/$100 = 20%. In this assignment, when we refer to return we mean return in percentage. Investment period for which a return is measured, could be daily, monthly, annually, etc.
Stock prices can fluctuate substantially during a day. However, in this assignment we are not focused on intraday prices. We intend to focus on daily price. For daily prices, we consider the *closing price* or *last price* as the price for a certain day.
Most markets operate around 250 days each year. So, markets do not provide any data for stock on the days that markets are closed. The data that we provide to you is raw, and on the days that market has been closed data is missing. You need to simply assume that price did not change on the days that market is closed.
The first three columns of the table below show a snapshot of raw data that you will use. The columns in red are outputs that we have produced based on the above explanations for adjusting daily prices and calculating returns. Note that returns are calculated on a daily basis.

Company Code	Date	ClosingPrice	Closing Price (adjusted)	Return	Return (%)
CBA.AX	5-Jan-00	25.35	$25.35		
CBA.AX	6-Jan-00	24.85	$24.85	$(0.50)	-1.97%
CBA.AX	7-Jan-00	25.1	$25.10	$0.25	1.01%
CBA.AX	8-Jan-00		$25.10	$-	0.00%
CBA.AX	9-Jan-00		$25.10	$-	0.00%
CBA.AX	10-Jan-00	25.7	$25.70	$0.60	2.39%
CBA.AX	11-Jan-00	25.285	$25.29	$(0.41)	-1.61%
CBA.AX	12-Jan-00	25.45	$25.45	$0.16	0.65%

CBA.AX	13-Jan-00	25.25	$25.25	$(0.20)	-0.79%
CBA.AX	14-Jan-00	25.583	$25.58	$0.33	1.32%

1.2. Average Returns for list of stocks

In order to calculate average returns over a given period of time, you simply need to average the returns values. The average return value for the time period given in above table is listed below.

Average Return	Average Return (%)
$0.02	**0.07%**

$$Average\ Return = \frac{\sum_{t=-m}^{n} Return_{it}}{m+n}$$

1.3. Cumulative Returns

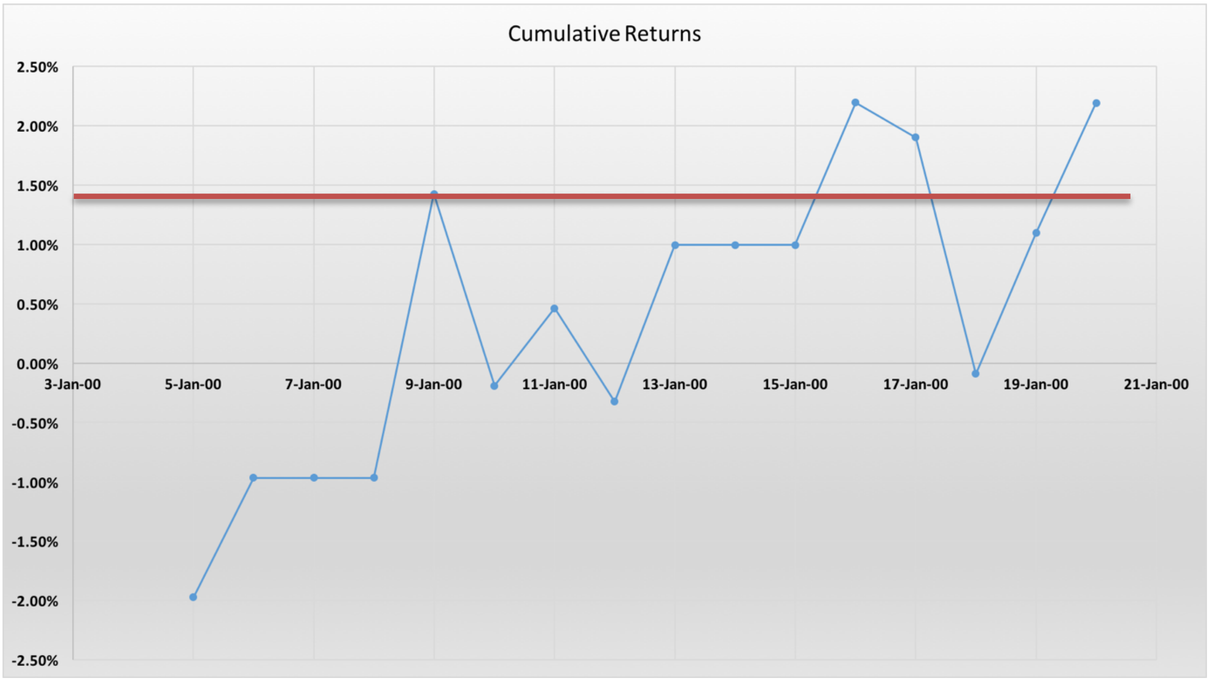
Cumulative return is the summation of returns over a given period. The formula for calculating cumulative return is shown below:

$$Cumulative\ Return = \sum_{t=-m}^{n} Return_{it}$$

where $Return_{it}$ is the return of Stock I at time t and -m and n represent number of days considered in past and future. The cumulative return for above table (for a period of 10 days) is

Cum. Return	Cum. Return (%)
$0.25	**1.10%**

1.4. Variation of Cumulative Returns, around a Date of Interest



We intend to inspect how cumulative returns (%) and average returns behave, around a particular date of interest. This will help us to behaviours of market data. The cumulative returns for the above table are shown in a graph below. The blue graph in the picture above

show how cumulative return have moved since the beginning of data (3 Jan) and changed at 9th Jan.
Assume that use is interested on 9 January, the redline on the graph is a reference line which starts at the observed cumulative return. We observe by following the blue line that the from 9th of January there is a reasonable negative impact on the returns, leading to around 2% drop following the event date.

How API Works

Figure 1 shows how our system interacts with the Company Returns API.

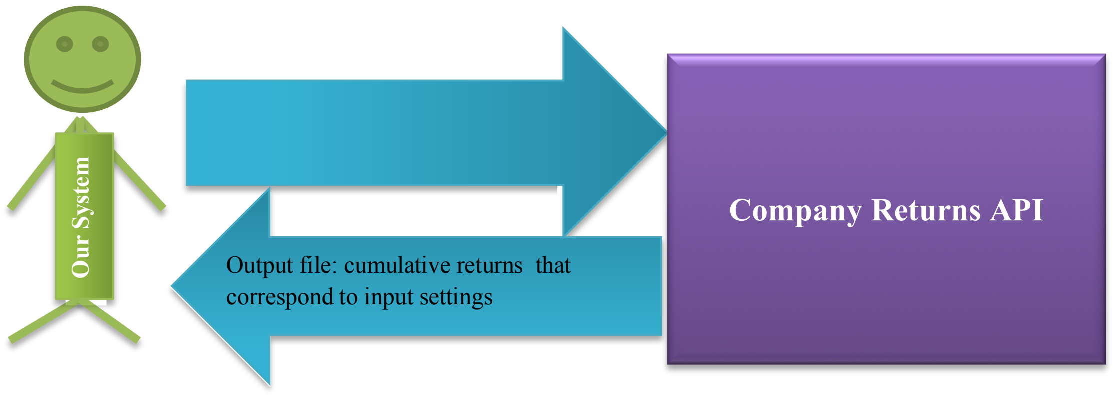
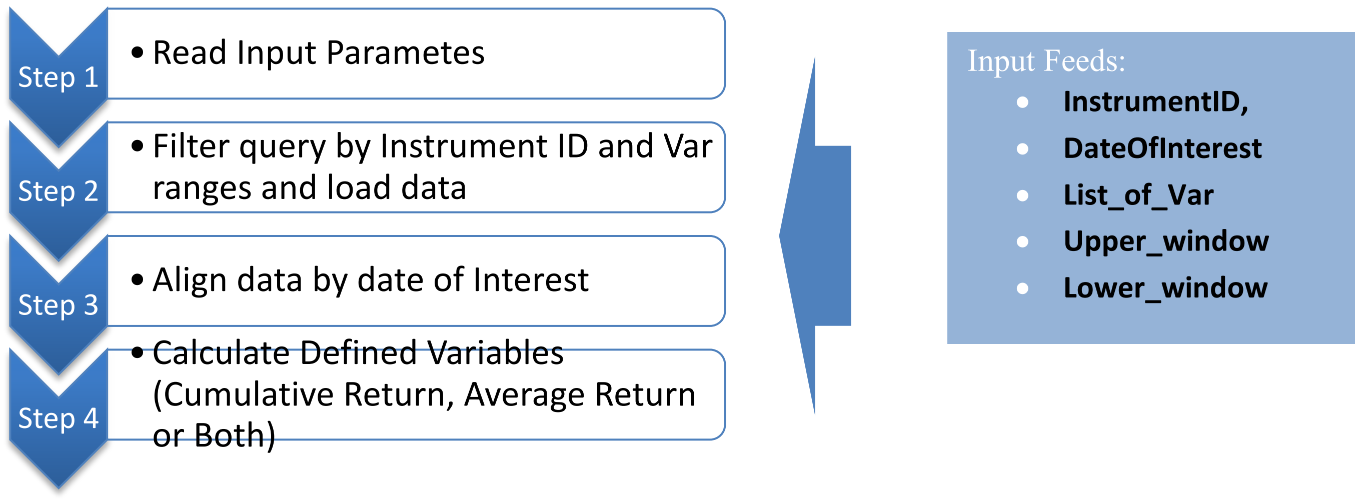

Our System
Output file: cumulative returns that correspond to input settings
Company Returns API

Figure 1. High level view

The language in which the module is to be written is not important as long as there is some way to invoke it from our system. Also, all exchanged data is in the form of text files to avoid any specific encoding. In other words, our system calls the module and supplies any data in the form of text files. After execution is finished, all results are visualised. It is important that the module can be used without its source code being revealed.
API's purpose is to facilitate calculation of return relative to a company, by external programs. The interface hides the complexity behind a calculation and external programs can treat it like a "black box" supply the required data and getting results back.
API requires the following input to run:

- **Inputs: InstrumentID, DateOfInterest, List_of_Var (CM_Return,AV_Return), upper_window(n), lower_window(m)**

Company Returns API gives external programs control over the generated output, so that they are able to calculate different return values with their own range of variables.


Step 1
• Read Input Parametes
Step 2
• Filter query by Instrument ID and Var ranges and load data
Step 3
• Align data by date of Interest
Step 4
• Calculate Defined Variables (Cumulative Return, Average Return or Both)
Input Feeds:
InstrumentID,
DateOfInterest
List_of_Var
Upper_window
Lower_window

Step 1
API parse the input parameters provided by user

Step 2
API send a query to get daily price data, filtered by the Instrument ID. As the API should calculated returns for each day in the defined date range, it needs to acquire daily price data for a range of dates before and after the DateOfIntrest.
As API have to calculate cumulative and average return for each day within the upper and lower window, minimally the range should be between (DateOfIntrest+2*UpperWindow) and (DateOfIntrest-2*LowerWindow-1), in order to have sufficient data for calculation. Refer to appendix 6.2 table where sample calculation was conducted, to understand the data range necessary.

Step 3
API build a table of price data, align with the given date.

Step 4
Company Returns API calculates the average returns and cumulative returns and shows them versus the event date. The formula for calculating cumulative return and average return at time (T) is shown below:

$$Average\ Return(i,T) = \frac{\sum_{t=T-m}^{T+n} Return_{it}}{m+n}$$

$$Cumulative\ Return(i,T) = \sum_{t=T-m}^{T+n} Return_{it}$$

where $Return_{it}$ is the return of Stock I at time t. The API must show how the cumulative returns have changed over time as we pass through the given date of interest. The output file must provide the cumulative return for each stock over the given period.

3. Input Parameters for API
As mentioned earlier, other than data acquired through the REST service, user will define input parameters for calculation returns.

- **Parameters:** InstrumentID, DateOfIntrest , List_of_Var (CM_Return,AV_Return), upper_window(n), lower_window(m). Remember that you only need to output the matrix that belongs to user-specified ranges.

Example set of input parameters:

InstrumentID	**ABP.AX**
ListOfVar	CM_Return,AV_Return
Upper Window	5
Lower Window	3
DateOfInterest	**10/12/2012**

4. Output Format

The output file returned by the API should contains the user requested variables for the defined Instrument ID. The output file must provide cumulative returns, average returns or both for every date relative to the date of interest. It should be formatted in a JSON file as shown in appendix 6.3.
For example, for the example input parameters the output values will be as given in table below, but in JSON format.

InstrumentID	RelativeDate	Date	Return	CM_Return	AV_Return
ABP.AX	-3	7/12/2012	0	0.024820106	0.003102513
ABP.AX	-2	9/12/2012	0	0.044121755	0.005515219
ABP.AX	-1	9/12/2012	-0.014593536	0.029312093	0.003664012
ABP.AX	**0**	**10/12/2012**	**0.014809661**	**0.029312093**	**0.003664012**
ABP.AX	1	11/12/2012	0.009794319	0.05300607	0.006625759
ABP.AX	2	12/12/2012	0	0.039109435	0.004888679
ABP.AX	3	13/12/2012	0.019301649	0.063082242	0.00788528
ABP.AX	4	14/12/2012	0	0.052941987	0.006617748
ABP.AX	5	15/12/2012	0	0.038499964	0.004812495

Appendix 6.2 contains an example table of calculation, showing the steps on how to obtain output.
Another output of the API is a log file must contain the following information:

- Developer team
- Module name and version
- Parameters passed
- An indication if execution has been successful or there is an error
- If error, indicate the nature of the error
- If successful, need to supply
  - Start date and time of execution

- ➢ End date and time of execution
- ➢ Elapsed time
- ➢ Output file name

5. Acquiring stock prices for the API.

To get the daily stock price data, you can use free REST API provided by Alpha Vintage.
Alpha Vintage documentation URL : https://www.alphavantage.co/documentation/

You can get the historical daily price data related to each Instrument ID by querying the API.

Given below is a URL of the REST service call for accessing company data.
https://www.alphavantage.co/query?**function**=TIME_SERIES_DAILY_ADJUSTED&**symbol**=BHP.AX&**apikey=<aquired_key>&**outputsize=full
This example URL, request to get full set of data of BHP.AX. You can get your own api key at https://www.alphavantage.co/support/#api-key and try this URL on a web browser to observe result data file in JSON fomrat. The URL contains set of parameters which you can configure, each separated by '&' sign. Values can be assigned to the parameters after the "=" sign.

6. Appendices

6.1 Additional Information

Teams have the choice of running their system on two different platforms:

- Standalone Program
  - ➢ PC running Windows
  - ➢ Unix/Linux platform
- Web service (accessible via a REST interface)

Throughout the workshop, each team will need to have a Web page. As a minimum, the page is showing:

- The team name and members
- Consecutive releases of their module. Each release page must include a link to download the module and information about:
  - o The date and version of the release
  - o What has been implemented so far
  - o Differences with previous version
  - o Clear instructions on how to run the module in standalone mode
  - o Guidelines on how to integrate the module with other systems
  - o Any test software or data

6.2 Calculation Example

InstrumentID	Relative Date	Date	Close Price	Close Price (Adjusted)	Return	Return(%)	CM_Return	AV_Return
ABP.AX		3/12/12	2.0122	2.0122				
ABP.AX		4/12/12	2.0122	2.0122	0	0		
ABP.AX		5/12/12	2.042	2.042	0.0298	0.014809661		
ABP.AX		6/12/12	2.042	2.042	0	0		
ABP.AX	-3	7/12/12		2.042	0	0	0.024820106	0.003102513
ABP.AX	-2	9/12/12		2.042	0	0	0.044121755	0.005515219
ABP.AX	-1	9/12/12	2.0122	2.0122	-0.0298	-0.014593536	0.029312093	0.003664012
ABP.AX	0	10/12/12	2.042	2.042	0.0298	0.014809661	0.029312093	0.003664012
ABP.AX	1	11/12/12	2.062	2.062	0.02	0.009794319	0.05300607	0.006625759
ABP.AX	2	12/12/12	2.062	2.062	0	0	0.039109435	0.004888679
ABP.AX	3	13/12/12	2.1018	2.1018	0.0398	0.019301649	0.063082242	0.00788528
ABP.AX	4	14/12/12		2.1018	0	0	0.052941987	0.006617748
ABP.AX	5	15/12/12		2.1018	0	0	0.038499964	0.004812495
ABP.AX		16/12/12	2.1516	2.1516	0.0498	0.023693977	0.038499964	0.004812495
ABP.AX		17/12/12	2.1217	2.1217	-0.0299	-0.013896635		
ABP.AX		18/12/12	2.1416	2.1416	0.0199	0.009379271		
ABP.AX		19/12/12	2.1516	2.1516	0.01	0.004669406		
ABP.AX		20/12/12	2.1416	2.1416	-0.01	-0.004647704		

6.3 Output JSON File Format

```
{
   "CompanyReturns": [
          {“InstrumentID”:<>,
          “Data”:[
                   { RelativeDate:<>,
                   Date:<>,
                   Return:<>;
                   CM_Return:<>,
                   AV_Return:<>}
                   { RelativeDate:<>,
                   Date:<>,
                   CM_Return:<>,
                   AV_Return:<>}
                   .
                   .
                   ]
          }
          {“InstrumentID”:<>,
          “Data”:[
                   { RelativeDate:<>,
                   Date:<>,
                   Return:<>,
                   CM_Return:<>,
                   AV_Return:<>}
                   { RelativeDate:<>,
                   Date:<>,
                   Return:<>,
                   CM_Return:<>,
                   AV_Return:<>}
                   .
                   .
                   .
                   ]
          }
]}
```

6.3 References

Brown, S., & Warner, J. 1985. Using daily stock returns: The case of event studies. ***Journal of Financial Economics***, 14: 3-3 1.
MacKinlay, AC. 1997. Event studies in economics and finance, ***Journal of Economic Literature***, vol. 35, no. 1, pp. 13-39.