Company Returns API Specification

Version:	3.3
Date Modified:	29 March 2017

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 2015, that was worth $120 on 31 Dec 2015 had a profit of $(120-100) = $20 in 2015. 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%**

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

1.3. Cumulative Returns

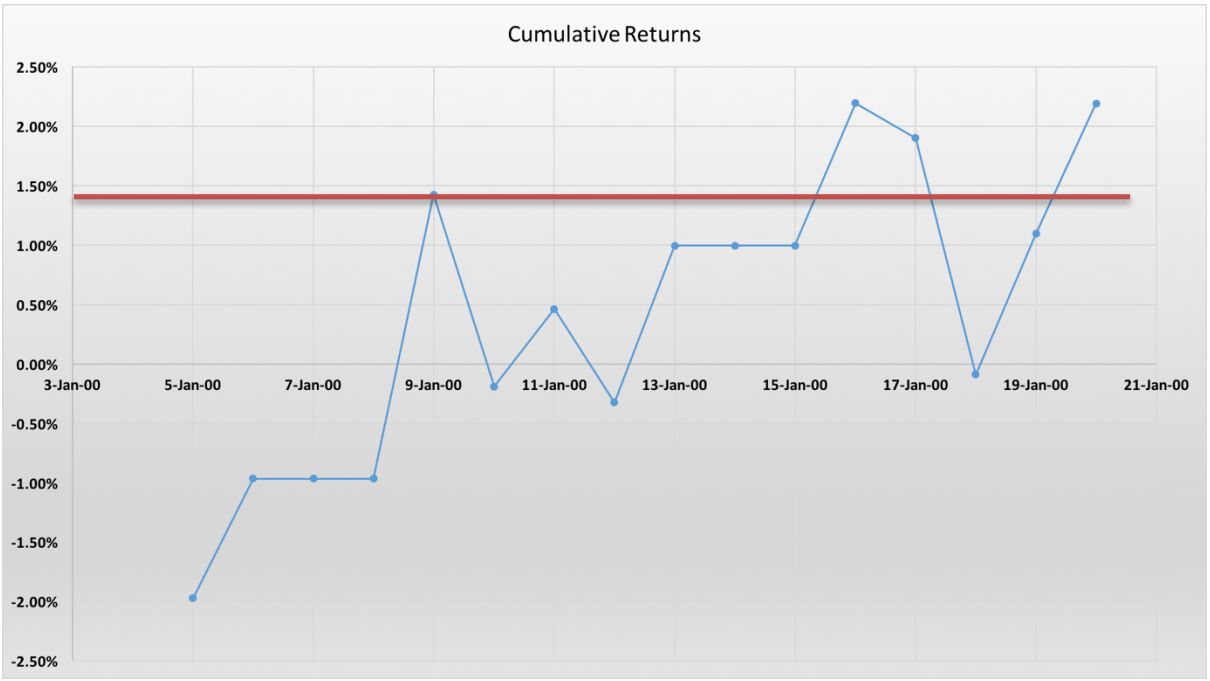
Cumulative return is the summation of returns over a given period. The formula for calculating cumulative return is shown below:

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

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.

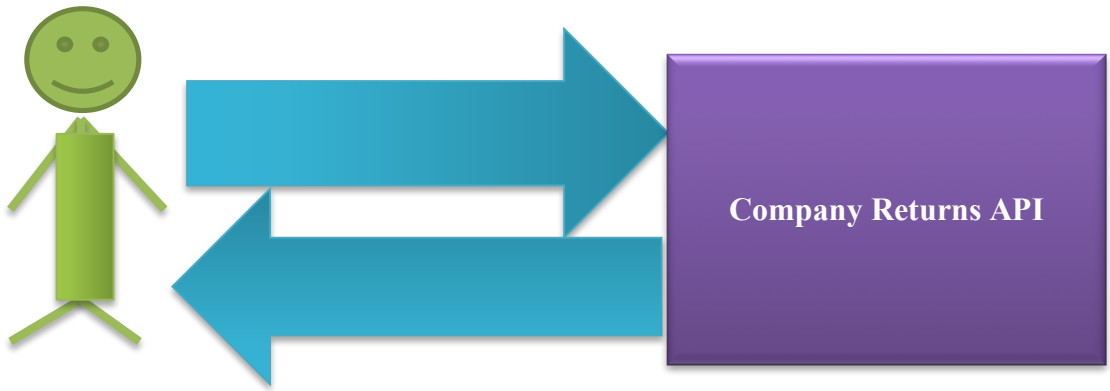


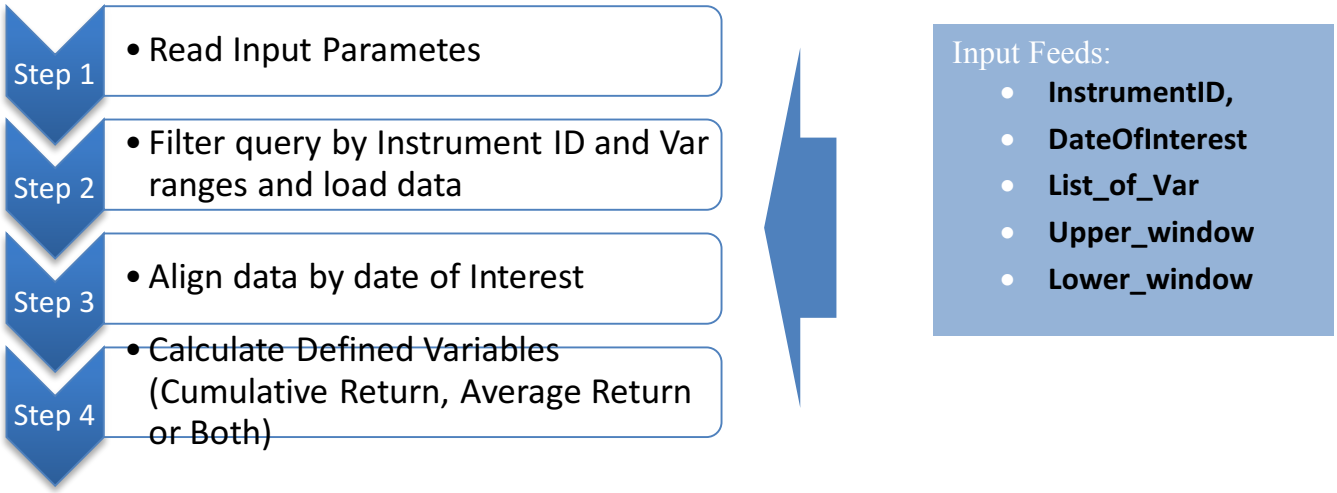
Figure 1. High level view

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

API parse the input parameters provided by user

Step 2

API send a query to get daily price data, filtered by the Instrument ID and Date rage. 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	0.009225843	0.00115323
ABP.AX	-2	9/12/2012	0	0.024080601	0.003010075
ABP.AX	-1	9/12/2012	0	0.043035782	0.005379473
ABP.AX	**0**	**10/12/2012**	**-0.014854757**	**0.028398459**	**0.003549807**
ABP.AX	1	11/12/2012	0.014637323	0.028398459	0.003549807
ABP.AX	2	12/12/2012	0.009660711	0.051548565	0.006443571
ABP.AX	3	13/12/2012	0	0.037465717	0.004683215
ABP.AX	4	14/12/2012	0.018955181	0.061621714	0.007702714
ABP.AX	5	15/12/2012	0	0.051613483	0.006451685

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 are provided with a REST API. You can get the price data related to each Instrument ID and for a defined time period, by querying the database. Given below is a URL of the REST service call for accessing Yahoo company data. This example URL, request to get data of two companies ABP.AX and BHP.AX from 2012-01-01 to 2013-01-01. You can copy and paste it on a web browser and observe result data file. The URL contains set of parameters which you can configure, each separated by '&' sign. Values can be assigned to the parameters after the "=" sign. You can copy and paste the given link and it will download a file with data from companies ABP.AX and BHP.AX.

http://adage.cse.unsw.edu.au:8080/ImportEventDataset/v1/data?Datasource=External:Yahoo&startDate=2012-01-01T00:00:00z&endDate=2013-01-01T23:59:59z&InstrumentID=Yahoo:ABP.AX;AAPL&DatasetType=EndOfDay

Hint: Changing the ';' seperated *InstrumentID* (Instrument Code) list after "Yahoo:" prefix and appropriate startDate and endDate variables are sufficient to get data from a new company.
The output will be a CSV file with pair of date and price values for the company and the period of time. Table below shows an example of a data file returned by the REST service.

ABP.AX	29/10/2015 11:34					
Date	Open	High	Low	Close	Volume	Adj Close
10/28/2015	116.93	119.300003	116.059998	119.269997	85023300	119.269997
10/27/2015	115.400002	116.540001	113.989998	114.550003	57953600	114.550003
10/26/2015	118.080002	118.129997	114.919998	115.279999	66019500	115.279999
10/23/2015	116.699997	119.230003	116.330002	119.080002	59139600	119.080002
10/22/2015	114.330002	115.5	114.099998	115.5	41272700	115.5

10/21/2015	114	115.580002	113.699997	113.760002	41795200	113.760002
10/20/2015	111.339996	114.169998	110.82	113.769997	48778800	113.769997
10/19/2015	110.800003	111.75	110.110001	111.730003	29606100	111.730003
10/16/2015	111.779999	112	110.529999	111.040001	38236300	111.040001
10/15/2015	110.93	112.099998	110.489998	111.860001	37341000	111.860001
10/14/2015	111.290001	111.519997	109.559998	110.209999	44325600	110.209999
10/13/2015	110.82	112.449997	110.68	111.790001	32424000	111.790001
10/12/2015	112.730003	112.75	111.440002	111.599998	30114400	111.599998
10/09/2015	110	112.279999	109.489998	112.120003	52533800	112.120003
10/08/2015	110.190002	110.190002	108.209999	109.5	61698500	109.5
10/07/2015	111.739998	111.769997	109.410004	110.779999	46602600	110.779999
10/06/2015	110.629997	111.739998	109.769997	111.309998	48196800	111.309998

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:
  - The date and version of the release
  - What has been implemented so far
  - Differences with previous version
  - Clear instructions on how to run the module in standalone mode
  - Guidelines on how to integrate the module with other systems
  - Any test software or data

6.2 Calculation Example

InstrumentID	Relative Date	Date	Close Price	ClosePrice (Adjusted)	Return	Return(%)	CM_Return	AV_Return
ABP.AX		3/12/2012	2.04204	2.04204				
ABP.AX		4/12/2012	2.01215	2.01215	-0.02989	-0.014637323		
ABP.AX		5/12/2012	2.01215	2.01215	0	0		
ABP.AX		6/12/2012	2.04204	2.04204	0.02989	0.014854757		
ABP.AX	-3	7/12/2012	2.04204	2.04204	0	0	0.010189819	0.001273727
ABP.AX	-2	9/12/2012		2.04204	0	0	0.024827142	0.003103393
ABP.AX	-1	9/12/2012		2.04204	0	0	0.044148565	0.005518571
ABP.AX	0	10/12/2012	2.01215	2.01215	-0.02989	-0.014637323	0.029293807	0.003661726
ABP.AX	1	11/12/2012	2.04204	2.04204	0.02989	0.014854757	0.029293807	0.003661726
ABP.AX	2	12/12/2012	2.06196	2.06196	0.01992	0.009754951	0.052992542	0.006624068
ABP.AX	3	13/12/2012	2.06196	2.06196	0	0	0.039105266	0.004888158
ABP.AX	4	14/12/2012	2.1018	2.1018	0.03984	0.019321422	0.063131155	0.007891394
ABP.AX	5	15/12/2012		2.1018	0	0	0.052927017	0.006615877
ABP.AX		16/12/2012		2.1018	0	0	0.043172066	0.005396508
ABP.AX		17/12/2012	2.15161	2.15161	0.04981	0.023698734		
ABP.AX		18/12/2012	2.12173	2.12173	-0.02988	-0.013887275		
ABP.AX		19/12/2012	2.14165	2.14165	0.01992	0.009388565		
ABP.AX		20/12/2012	2.15161	2.15161	0.00996	0.00465062		

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.