Homework 4: Server-side Scripting

1. Objectives

- Get experience with the PHP programming language
- Get experience with the Markit on Demand APIs
- Get experience using JSON parsers in PHP.

2. Description

In this exercise, you are asked to create a webpage that allows you to search for stock information using the *Markit on Demand* APIs, and the results will be displayed in tabular format.

2.1. Description of the Search Form

A user first opens a page, called **stock.php**, where he/she can enter a company name or symbol. As shown in Figure 1. Providing a value for the "*Company Name or Symbol*" field is mandatory (the validation should be done using HTML5/Javascript). The form should include a proper disclaimer "Powered by Markit on Demand", linking to: 'http://www.markit.com/product/markit-on-demand'.

Figure 1: Initial Search Screen

The search form has two buttons:

- **Search** button: If the user clicks on the search button without providing a value in the "Company Name or Symbol" field, HTML5 should validate it (i.e Figure 2). An example of valid input is shown in Figure 3. Once the user has provided valid data, your client script should send a request to your web server for **stock.php** with the form data. You can use either GET or POST to transfer the form data to the web server. A PHP script will retrieve the data and send it to the *Markit on Demand* restful Web Service.
- **Clear** button: This button must clear the result area and the text field using a JavaScript.

Figure 2: An Error Message when there is no input

Figure 3: An Example of Valid Input

2.2. Displaying Results

In this section, we outline how to use the form data to construct the calls to restful web services from *Markit on Demand* APIs and display the result in the web page. Mainly we use two web services: Lookup and Quote. The Lookup web service returns an array of company matches for a given input, which contains the name, symbol and the trading exchange of the company while the Quote web service returns the current value of a corresponding company stock when user inputs a symbol of a company stock

The search operation is based on two main steps:

1. Use the Lookup web service in the *Markit on Demand* API to find the stock symbol corresponding to the user input.
2. Use the symbol returned from the Lookup web service to call the Quote web service to get the company stock information.

The PHP script (i.e., **stock.php**) uses the input information (company name or symbol) to construct a restful web service URL to retrieve all companies matching the query input:

http://dev.markitondemand.com/MODApis/Api/v2/**Lookup**/json?input=**APPL**

The Lookup web service URL has a parameter called *input*. The value of the *input* parameter should be the text entered in the "Company Name or Symbol" edit box. The above link is an example of calling lookup service with the input "AAPL". The response of this URL is an JSON object.

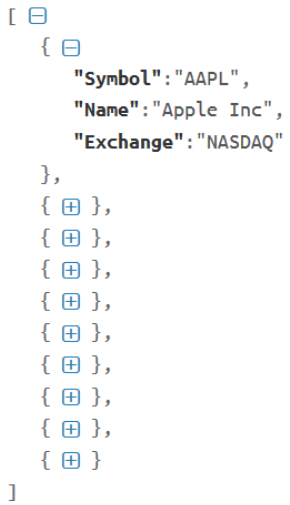

Figure 4: A Sample Result of the lookup service

The PHP script should parse the returned JSON object and extract the necessary fields. After extracting the data, the PHP script should display the data in a tabular format below the search form (see Figure 5). All returned data is used to render a result table. The symbol, name and exchange values should be displayed in the result table. Please note that the symbol value will be used for the next step to retrieve the quote information.

Name	Symbol	Exchange	Details
Apple Inc	AAPL	NASDAQ	More Info
Applied Industrial Technologies Inc	AIT	NYSE	More Info
Applied Materials Inc	AMAT	NASDAQ	More Info
Applied Micro Circuits Corp	AMCC	NASDAQ	More Info
Science Applications International Corp	SAIC	NYSE	More Info
	APLE	NYSE	More Info
GCP Applied Technologies Inc	GCP	NYSE	More Info
Appliance Recycling Centers of America Inc	ARCI	NASDAQ	More Info
Applied Industrial Technologies Inc	AIT	BATS Trading Inc	More Info
Science Applications International Corp	SAIC	BATS Trading Inc	More Info

Figure 5: An Example of Search Result

If the Lookup service returns an empty result set, the page should return "*No Record has been found*". Figure 6 shows an example when searching for a company name "*BLAHBLAH*".

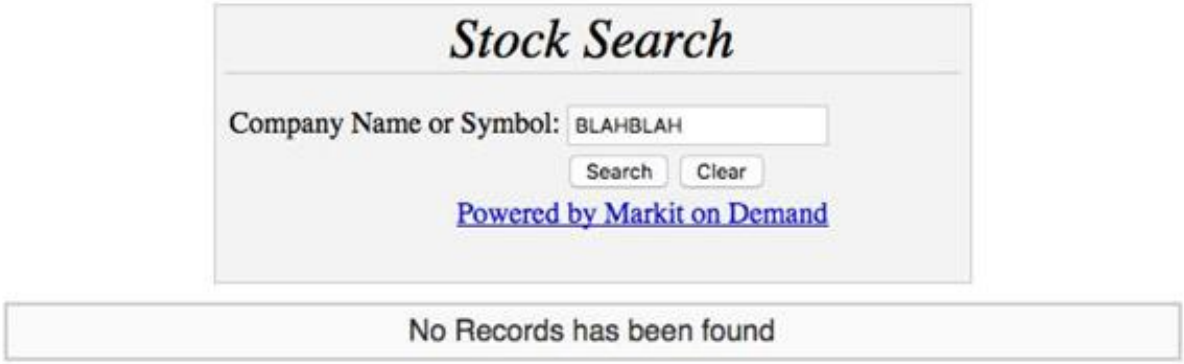

Figure 6: Search Result when there is no matching result

When the search result has at least one record, you need to map the data extracted from the lookup service result to render the result table as follows.

Table Column	Lookup service response
Name	<Name> tag under LookupResult
Symbol	<Symbol> tag under LookupResult
Exchange	<Exchange> tag under LookupResult

The details column contains a “More Info” link and when users click on the link of a certain company, the PHP script should use the **corresponding** symbol value to construct another restful web service URL to query the quote service to get the company quote info for the given symbol:

http://dev.markitondemand.com/MODApis/Api/v2/Quote/json?symbol=**SYMBOLVALUE**

The query of the quote method returns a JSON object as shown below for AAPL symbol:

http://dev.markitondemand.com/MODApis/Api/v2/Quote/json?symbol=***AAPL***

Figure 7 shows the returned JSON file to be parsed by the PHP script extract required fields.


```
{
   "Status":"SUCCESS",
   "Name":"Apple Inc",
   "Symbol":"AAPL",
   "LastPrice":95,
   "Change":0.980000000000004,
   "ChangePercent":1.04233141884706,
   "Timestamp":"Mon Feb 8 15:59:00 UTC-05:00 2016",
   "MSDate":42408.6659722222,
   "MarketCap":526735385000,
   "Volume":5125894,
   "ChangeYTD":105.26,
   "ChangePercentYTD":-9.74729241877257,
   "High":95.69,
   "Low":93.05,
   "Open":93.24
}
```


Figure 7: A Sample JSON-formatted object returned from *Markit on Demand* quote service

After extracting the data, the PHP script should display the data in a tabular format below the search form. A sample output is shown in Figure 8.

Name	Apple Inc
Symbol	AAPL
Last Price	94.85
Change	-0.16▼
Change Percent	-0.17%▼
Timestamp	2016-02-09 09:11 AM
Market Cap	525.9 B
Volume	1,842,552
Change YTD	(-10.41)▼
Change Percent YTD	-9.89%▼
High	95.38
Low	93.94
Open	94.3

Figure 8: Search Result When clicking the More Info Link

When the status value is “SUCCESS”, the PHP script should map the data retrieved from the quote service response to render the company stock table using the following mapping:

Table Column	Quote Service Response
Name	The value of *Name*
Symbol	The value of *Symbol*
Last Price	The value of *LastPrice*
Change	The value of *Change* rounded to two decimal points followed by a marker representing the price change trend of the stock
Change Percent	The value of *ChangePercent* rounded to two decimal points followed by a percentage “%”character and a marker representing the price change trend of the stock.
Timestamp	The value of *Timestamp* displayed in YYYY-MM-DD HH:MM AM/PM format.
Market Cap	The value of *MarketCap should be divided by a billion (1,000,000,000) and rounded* to two decimal points. The unit is B (stands for billion).
Volume	The value of *Volume* in Integer format. The value should be rendered with a thousand operator format (e.g., 9,876,543)
Change YTD	The value of *LastPrice* minus *ChangeYTD*, in parentheses if a negative value. The output value should be rounded to two decimal points followed by a mark representing the YTD change trend of the stock.
High	The value of *High*. Display the original data, no round is needed.
Low	The value of *Low*. Display the original data, no round is needed
Open	The value of *Open*. Display the original data, no round is needed.

Regarding the marker icon rendered beside the values of *Change*, *Change Percent*, *Change YTD*, and *Change Percent YTD*, a red-down arrow icon is displayed if the value is negative while a green-up arrow icon is displayed if the value is positive. The icons can be found on Canvas under HW4 folder.

If the returned result of the quote service has the “FAILURE” status value, the PHP script should display a message “*There is no stock information available*”. For example, searching for stock information using the symbol “GRNREG”, it returns the JSON output shown in Figure 9. Figure 10 shows how the PHP script handles this case.

```
{
   "Status":"Failure|APP_SPECIFIC_ERROR",
   "Name":"NASD REN ENER GE",
   "Symbol":"GRNREG",
   "LastPrice":0,
   "Change":0,
   "ChangePercent":0,
   "Timestamp":null,
   "MSDate":0,
   "MarketCap":0,
   "Volume":0,
   "ChangeYTD":0,
   "ChangePercentYTD":0,
   "High":0,
   "Low":0,
   "Open":0
}
```


Figure 9: An Example of the JSON response of Quote Service when the status is "FAILURE"

Figure 10: Search result when the Quote Service returns a response with "FAILURE" status

In summary, the search mechanism to be implemented behaves as follows:

- Based on the input data in the search form, construct a web service URL to retrieve the output from the lookup service.
- Parse the returned JSON and extract the symbol values.
- Call the quote service and retrieve the JSON-formatted output.
- Parse the returned JSON-formatted output and extract the quote.
- Display the stock information in tabular format.

2.3. Saving Previous Inputs

In addition to displaying the results, the PHP page should maintain the provided values to display the current result. For example, if a user searches for "*Company Name or symbol: GE*", the user should see what was provided in the search form when displaying the results. Specifically, when clicking on the "Search" button, the page should display the result retrieved from the Lookup web service and keep the value provided in the search form. In addition, when clicking on the "More Info" link, the page should display the result retrieved from the Quote web service and keep the value provided in the search form. It follows that you need to keep the whole search box/input fields and buttons even while displaying results/errors.

3. Hints

3.1. Markit on demand API Documentation

For information about the Markit on Demand API, please go to:
http://dev.markitondemand.com/MODApis/

3.2 Parsing JSON-formatted data in PHP

In PHP 5, you can parse JSON-formatted data using the "*json_decode*" function. For more information, please go to http://php.net/manual/en/function.json-decode.php.

To read the contents of a JSON-formatted object, you can use the "*file_get_contents*" function.

4. Submission

Your page should be hosted on your Amazon AWS instance. In addition, submit your files (likely only a single .php file) electronically to Canvas account so that they can be graded.
A readme file should accompany your submission that includes the URL for your Amazon AWS instance for the grader to check out your page.