

# SWE Project

RESTful APIs

Fall 2019

Team 7

# Team 7 Roster

- Lateef
- Anthony
- Austin
- Christine
- David
- Andrew

# Technology Overview

- XAMPP
- Windows OS
- PHP
- MySQL
- MariaDB
- myPHPadmin
- HTML/CSS/Bootstrap
- Postman
- Windows Task Scheduler
- SQLify.io (third party tool/website)

# Downloaded XAMPP

- <https://www.apachefriends.org/index.html>
- Downloaded XAMPP
- [https://www.youtube.com/watch?time\\_continue=68&v=h6DEDm7C37A&feature=emb\\_title](https://www.youtube.com/watch?time_continue=68&v=h6DEDm7C37A&feature=emb_title)

[Apache Friends](#) [Download](#) [Add-ons](#) [Hosting](#) [Community](#) [About](#)  [Search](#) [EN](#) [▼](#)

 **XAMPP** Apache + MariaDB + PHP + Perl

### What is XAMPP?

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.



[Download](#)  
Click here for other versions

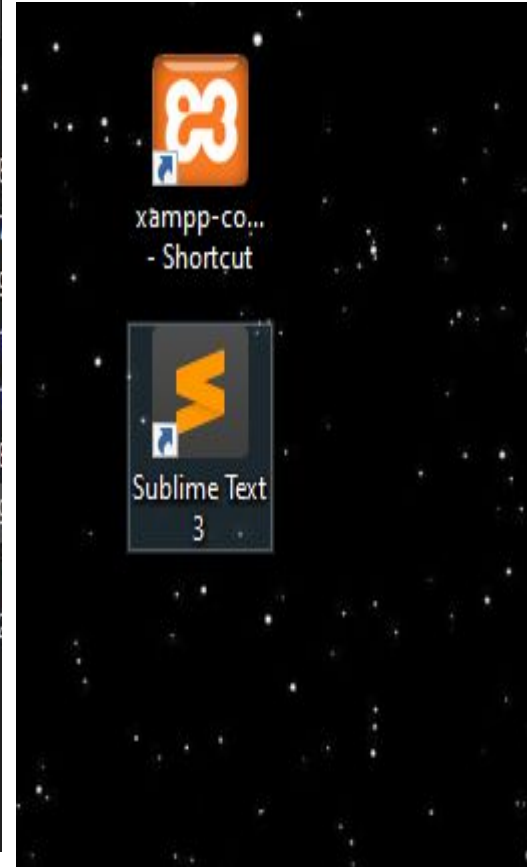
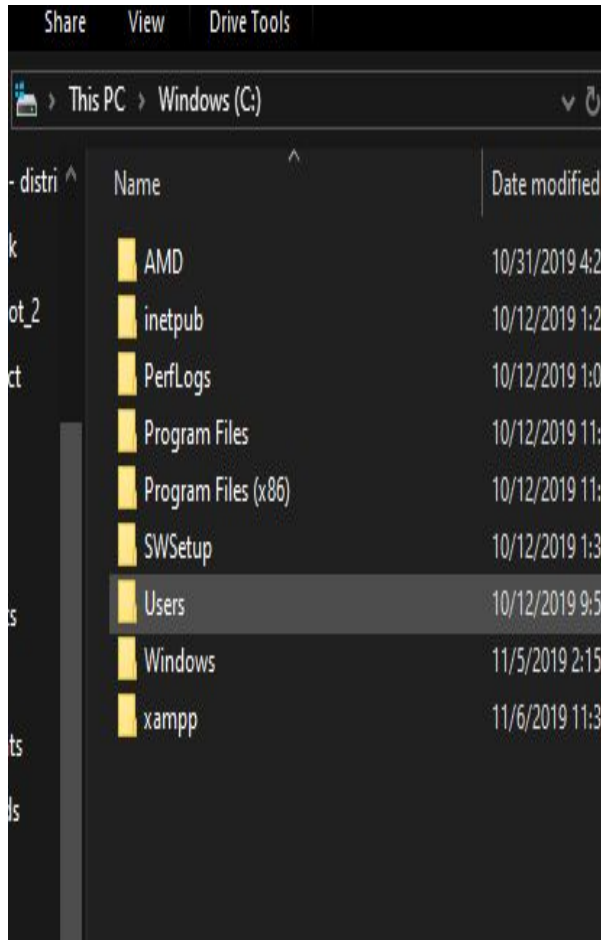
 **XAMPP for Windows**  
7.3.11 (PHP 7.3.11)

 **XAMPP for Linux**  
7.3.11 (PHP 7.3.11)

 **XAMPP for OS X**  
7.3.11 (PHP 7.3.11)

## Setting it up

- Download is simple
- When down, created shortcut of Control Panel and XAMPP Folder from C drive. Control panel was inside xampp folder in C drive.
- Sublime Text 3 editor would be the editor of choice.

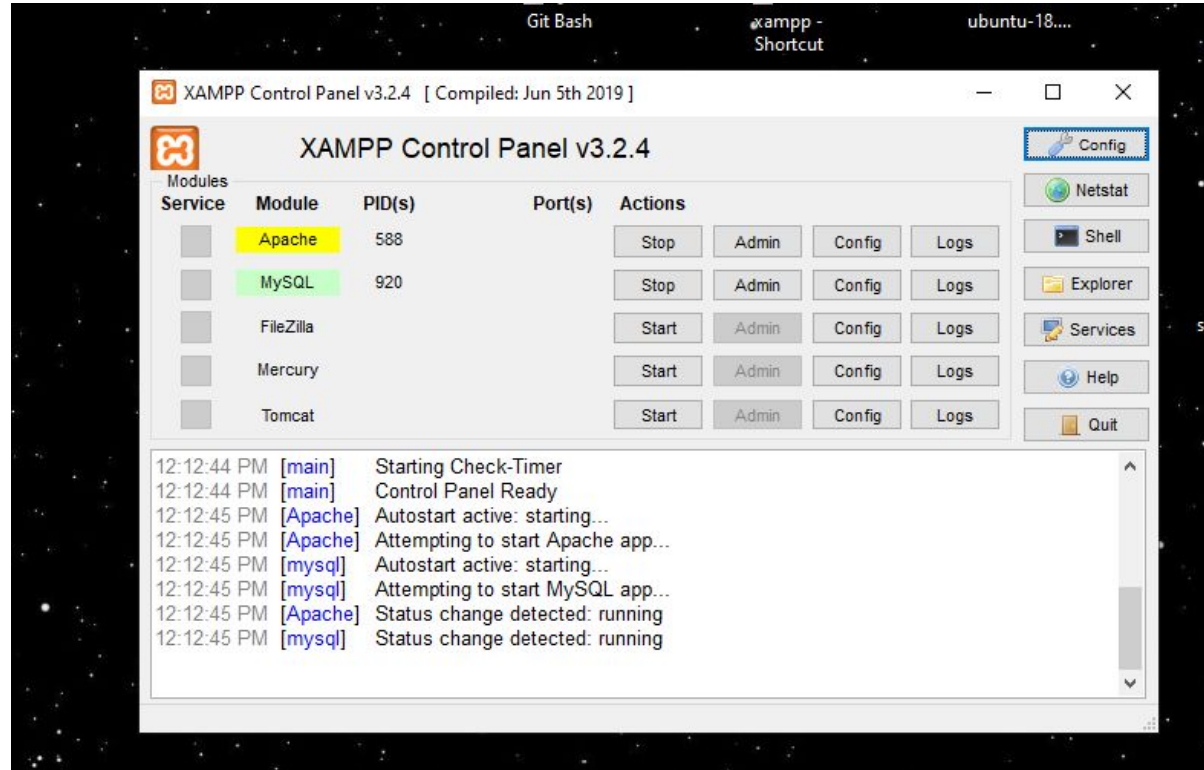


# Languages Used

- Since we were using XAMPP, we had to try and abide by its main languages, MySQL and PHP.
- MariaDB is the standard of MySQL that was being used, thus documentation was read from their site.
- For front end was HTML, CSS, Bootstrap.

# Working with XAMPP

- Allows us to turn our computer into server (localhost), and run server side operations.
- Same for database events.
- Open Control Panel ShortCut
- Click start on Apache and MySQL parts



## Open Web browser

- Our team used Google Chrome
- Type localhost in the url bar, and hit enter
- Should see screen similar to right (apache and mysql need to be on)



### Welcome to XAMPP for Windows 7.1.33

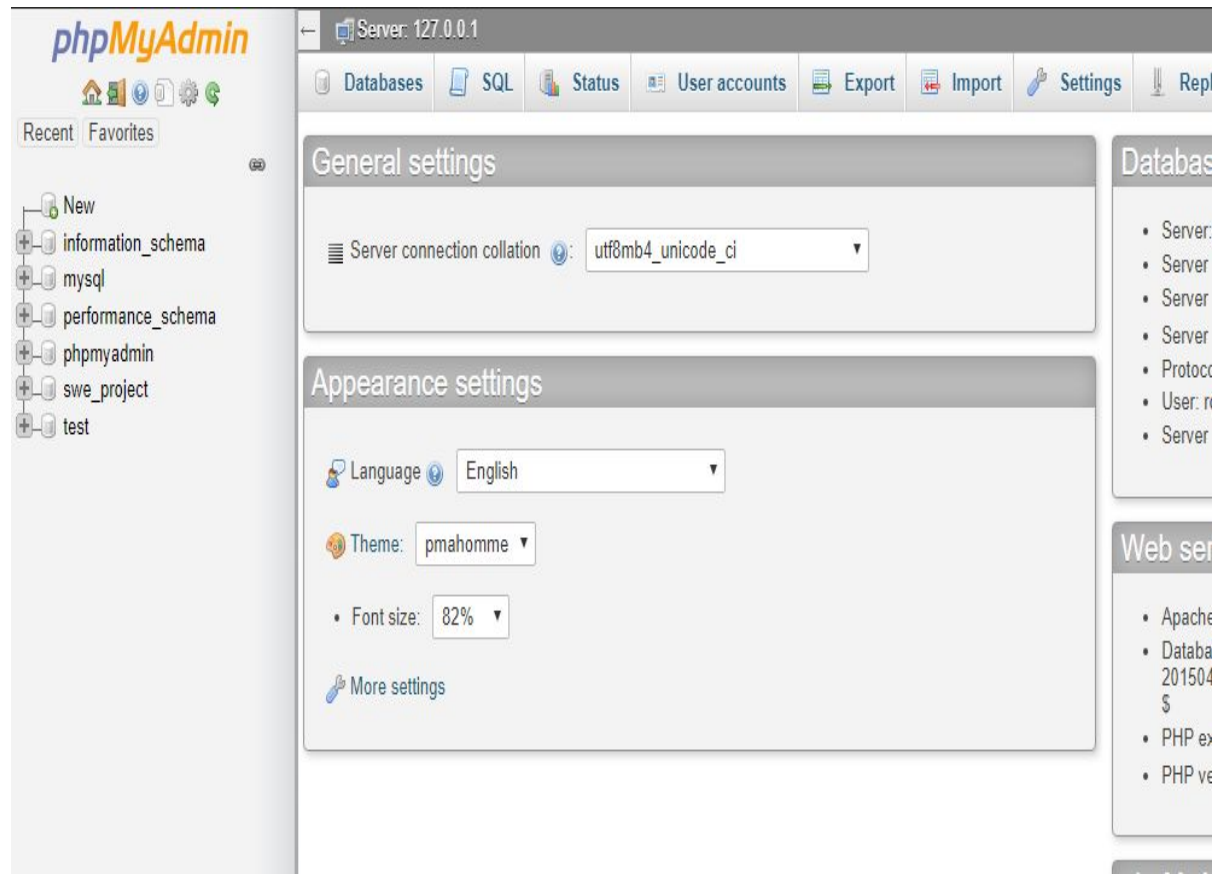
You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the [FAQs](#) to learn how to protect your site. Alternatively you can use [WAMP](#), [MAMP](#) or [LAMP](#) which are similar packages which are more suitable for production.



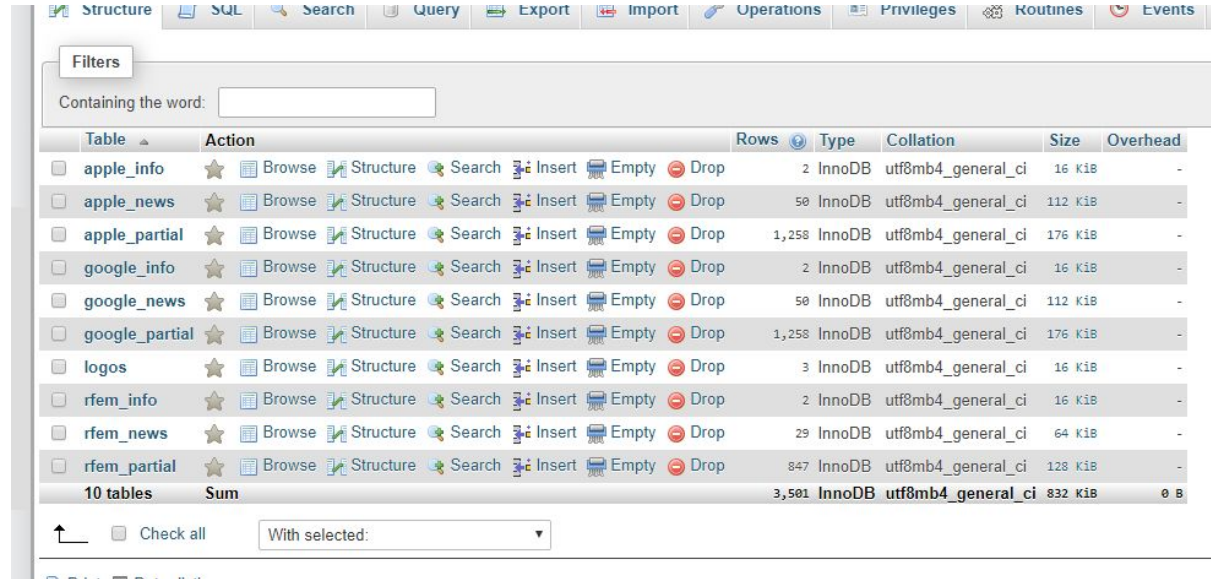
# myPHPadmin

- Click on top right where it says myPHPadmin
- Once there, you should see screen similar to right
- This is a GUI that comes with XAMPP.
- Allows for table manipulation, database creation, setting events, managing permissions, etc.
- Simpler and helpful to go with MySQL shell,



# myPHPadmin

- Relevant tables
- Database is MariaDB
- Comes installed with XAMPP

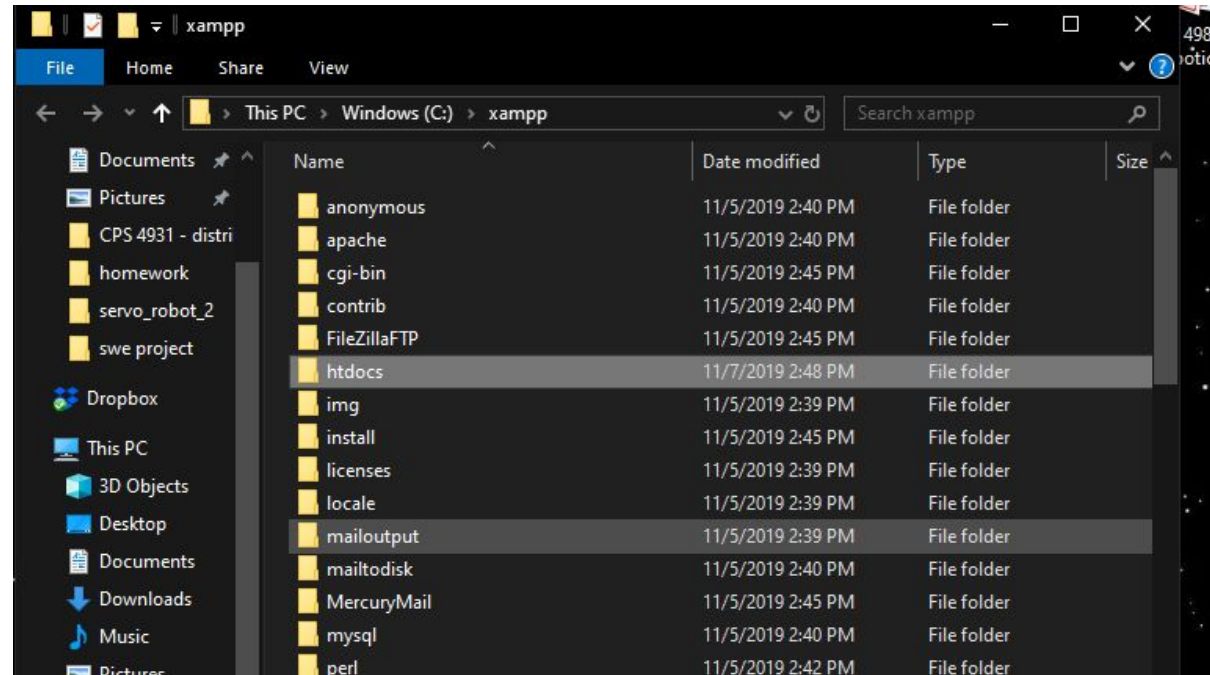


The screenshot shows the myPHPadmin interface with a menu bar at the top containing: Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, and Events. Below the menu is a 'Filters' section with a text input labeled 'Containing the word:'. The main content area displays a table of database tables. The table has columns for 'Table', 'Action', 'Rows', 'Type', 'Collation', 'Size', and 'Overhead'. The 'Table' column lists 10 tables: apple\_info, apple\_news, apple\_partial, google\_info, google\_news, google\_partial, logos, rfem\_info, rfem\_news, and rfem\_partial. The 'Action' column provides icons for Browse, Structure, Search, Insert, Empty, and Drop for each table. The 'Rows' column shows the number of rows for each table. The 'Type' column shows 'InnoDB' for all tables. The 'Collation' column shows 'utf8mb4\_general\_ci' for all tables. The 'Size' column shows the size of each table in KiB. The 'Overhead' column shows the overhead for each table. At the bottom of the table, there is a 'Sum' row showing the total for all 10 tables. Below the table, there is a 'Check all' checkbox and a 'With selected:' dropdown menu.

| Table          | Action                                    | Rows  | Type   | Collation          | Size    | Overhead |
|----------------|---|-------|--------|--------------------|---------|----------|
| apple_info     | Browse Structure Search Insert Empty Drop | 2     | InnoDB | utf8mb4_general_ci | 16 KiB  | -        |
| apple_news     | Browse Structure Search Insert Empty Drop | 50    | InnoDB | utf8mb4_general_ci | 112 KiB | -        |
| apple_partial  | Browse Structure Search Insert Empty Drop | 1,258 | InnoDB | utf8mb4_general_ci | 176 KiB | -        |
| google_info    | Browse Structure Search Insert Empty Drop | 2     | InnoDB | utf8mb4_general_ci | 16 KiB  | -        |
| google_news    | Browse Structure Search Insert Empty Drop | 50    | InnoDB | utf8mb4_general_ci | 112 KiB | -        |
| google_partial | Browse Structure Search Insert Empty Drop | 1,258 | InnoDB | utf8mb4_general_ci | 176 KiB | -        |
| logos          | Browse Structure Search Insert Empty Drop | 3     | InnoDB | utf8mb4_general_ci | 16 KiB  | -        |
| rfem_info      | Browse Structure Search Insert Empty Drop | 2     | InnoDB | utf8mb4_general_ci | 16 KiB  | -        |
| rfem_news      | Browse Structure Search Insert Empty Drop | 29    | InnoDB | utf8mb4_general_ci | 64 KiB  | -        |
| rfem_partial   | Browse Structure Search Insert Empty Drop | 847   | InnoDB | utf8mb4_general_ci | 128 KiB | -        |
| 10 tables      | Sum                                       | 3,501 | InnoDB | utf8mb4_general_ci | 832 KiB | 0 B      |

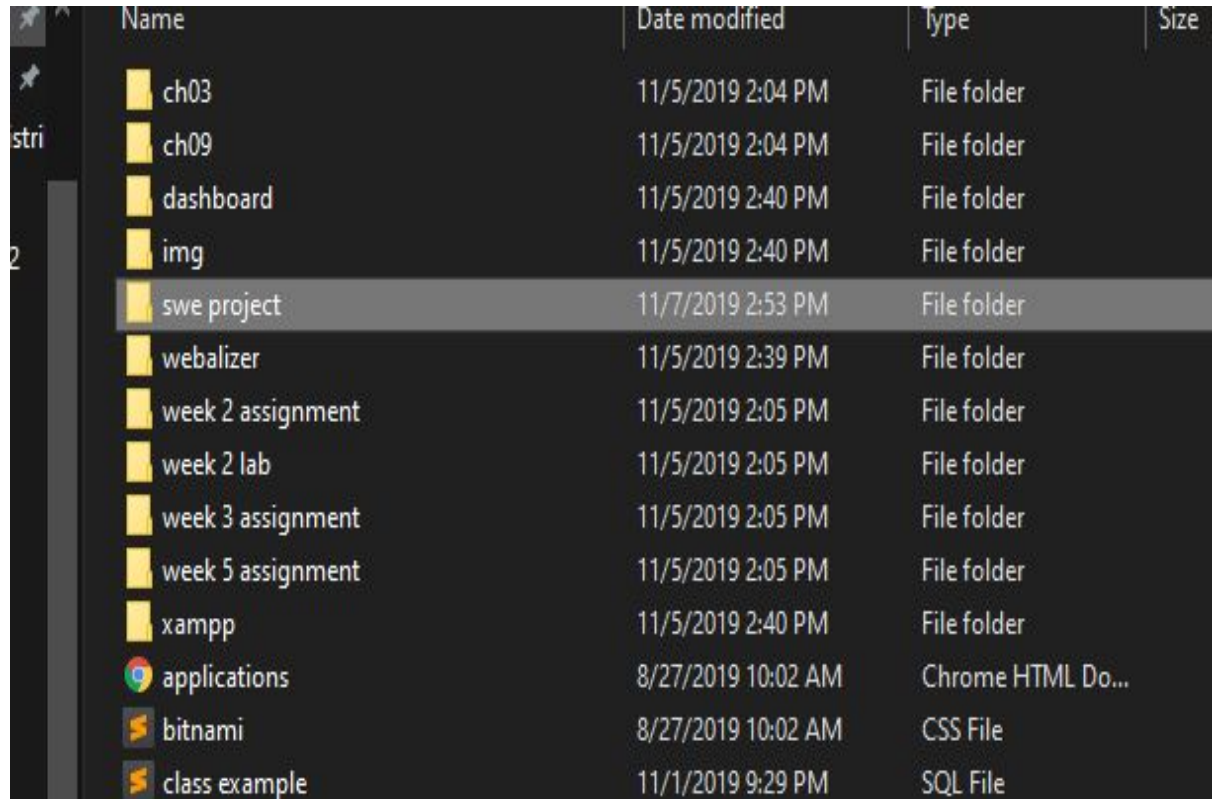
# Coding

- To create and run files, open up XAMPP folder
- Go to htdocs
- Create relevant folder
- From there, place relevant source PHP, html, etc.



# Coding

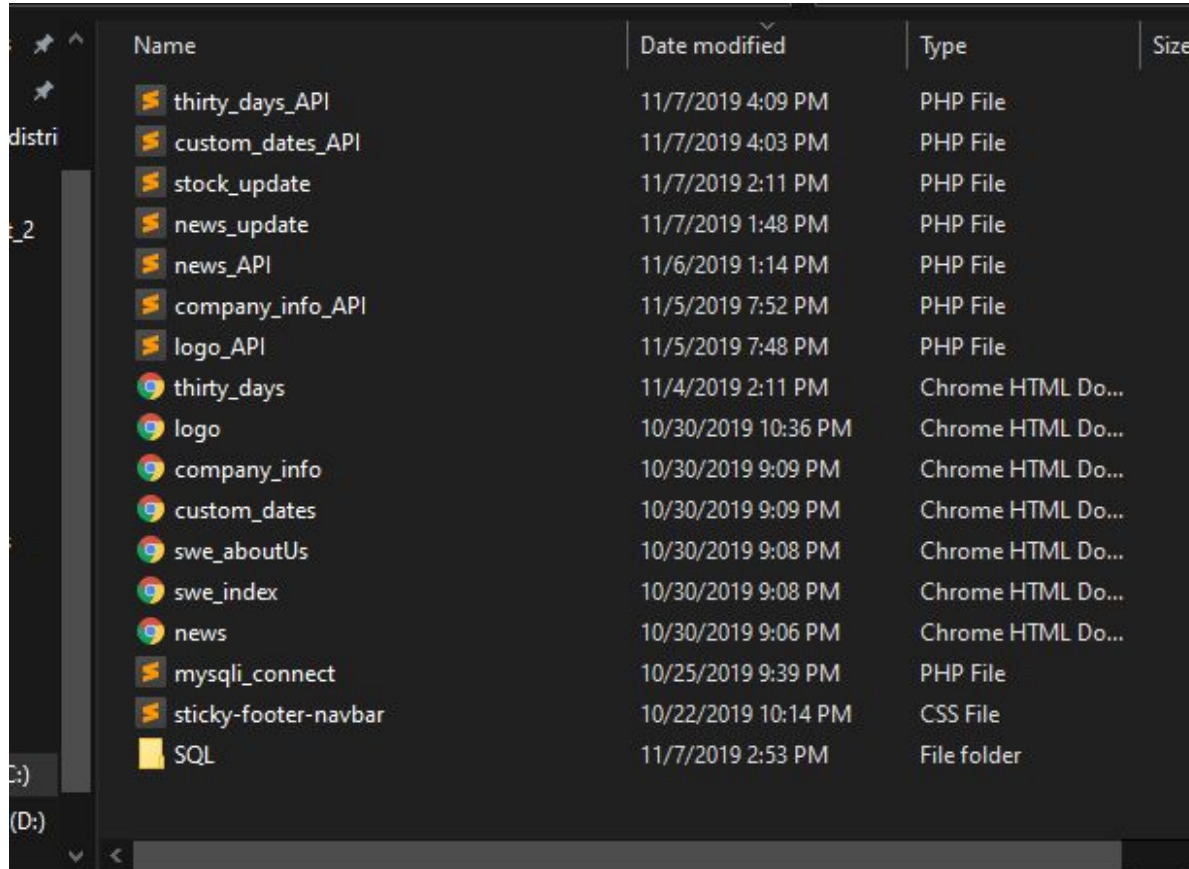
- To create and run files, open up XAMPP folder
- Go to htdocs
- Create relevant folder
- From there, place relevant source PHP, html, etc.
- Our project folder would be called 'swe project'



| Name              | Date modified      | Type              | Size |
|-------------------|--------------------|-------------------|------|
| ch03              | 11/5/2019 2:04 PM  | File folder       |      |
| ch09              | 11/5/2019 2:04 PM  | File folder       |      |
| dashboard         | 11/5/2019 2:40 PM  | File folder       |      |
| img               | 11/5/2019 2:40 PM  | File folder       |      |
| swe project       | 11/7/2019 2:53 PM  | File folder       |      |
| webalizer         | 11/5/2019 2:39 PM  | File folder       |      |
| week 2 assignment | 11/5/2019 2:05 PM  | File folder       |      |
| week 2 lab        | 11/5/2019 2:05 PM  | File folder       |      |
| week 3 assignment | 11/5/2019 2:05 PM  | File folder       |      |
| week 5 assignment | 11/5/2019 2:05 PM  | File folder       |      |
| xampp             | 11/5/2019 2:40 PM  | File folder       |      |
| applications      | 8/27/2019 10:02 AM | Chrome HTML Do... |      |
| bitnami           | 8/27/2019 10:02 AM | CSS File          |      |
| class example     | 11/1/2019 9:29 PM  | SQL File          |      |

# Coding

- Once files are created, you are ready to test on browser.
- myPHPadmin will also help, with setting up database and table configuration, and just being an overall strong GUI
- Make sure apache and mySQL are still on control panel



A screenshot of a Windows File Explorer window displaying a directory listing. The window has a dark theme. The left sidebar shows a navigation pane with a search bar and a list of locations. The main area shows a table of files and folders. The table has four columns: Name, Date modified, Type, and Size. The files listed include PHP files, HTML documents, a CSS file, and a folder.

| Name                 | Date modified       | Type              | Size |
|----------------------|---------------------|-------------------|------|
| thirty_days_API      | 11/7/2019 4:09 PM   | PHP File          |      |
| custom_dates_API     | 11/7/2019 4:03 PM   | PHP File          |      |
| stock_update         | 11/7/2019 2:11 PM   | PHP File          |      |
| news_update          | 11/7/2019 1:48 PM   | PHP File          |      |
| news_API             | 11/6/2019 1:14 PM   | PHP File          |      |
| company_info_API     | 11/5/2019 7:52 PM   | PHP File          |      |
| logo_API             | 11/5/2019 7:48 PM   | PHP File          |      |
| thirty_days          | 11/4/2019 2:11 PM   | Chrome HTML Do... |      |
| logo                 | 10/30/2019 10:36 PM | Chrome HTML Do... |      |
| company_info         | 10/30/2019 9:09 PM  | Chrome HTML Do... |      |
| custom_dates         | 10/30/2019 9:09 PM  | Chrome HTML Do... |      |
| swe_aboutUs          | 10/30/2019 9:08 PM  | Chrome HTML Do... |      |
| swe_index            | 10/30/2019 9:08 PM  | Chrome HTML Do... |      |
| news                 | 10/30/2019 9:06 PM  | Chrome HTML Do... |      |
| mysqli_connect       | 10/25/2019 9:39 PM  | PHP File          |      |
| sticky-footer-navbar | 10/22/2019 10:14 PM | CSS File          |      |
| SQL                  | 11/7/2019 2:53 PM   | File folder       |      |

















- Type localhost and folder name. You don't need to type full path, and include htdocs.
- Hit enter
- You should see directory of files or just home index, depending on how files are titled.



# Index of /swe project

## File directory

- From here, you can see your files
- You can then click and start troubleshooting.

| <u>Name</u>   | <u>Last modified</u> | <u>Size</u> | <u>Descripti</u> |
|---|----------------------|-------------|------------------|
|  <a href="#">Parent Directory</a>          |                      | -           |                  |
|  <a href="#">SQL/</a>                      | 2019-11-07 14:53     | -           |                  |
|  <a href="#">company_info.html</a>         | 2019-10-30 21:09     | 1.9K        |                  |
|  <a href="#">company_info_API.php</a>      | 2019-11-05 19:52     | 1.0K        |                  |
|  <a href="#">custom_dates.html</a>         | 2019-10-30 21:09     | 2.2K        |                  |
|  <a href="#">custom_dates_API.php</a>      | 2019-11-07 16:03     | 5.9K        |                  |
|  <a href="#">logo.html</a>                 | 2019-10-30 22:36     | 1.9K        |                  |
|  <a href="#">logo_API.php</a>              | 2019-11-05 19:48     | 1.1K        |                  |
|  <a href="#">mysqli_connect.php</a>        | 2019-10-25 21:39     | 348         |                  |
|  <a href="#">news.html</a>                 | 2019-10-30 21:06     | 2.1K        |                  |
|  <a href="#">news_API.php</a>              | 2019-11-06 13:14     | 1.3K        |                  |
|  <a href="#">news_update.php</a>           | 2019-11-07 13:48     | 2.4K        |                  |
|  <a href="#">sticky-footer-navbar.&gt;</a> | 2019-10-22 22:14     | 469         |                  |
|  <a href="#">stock_update.php</a>          | 2019-11-07 14:11     | 2.2K        |                  |

## After troubleshooting and testing

- We have to make sure the database is updated daily with appropriate values
- We, again, used PHP and mySQL to facilitate that.

```
?>
3 <?php
4
5 function fetch_json($url)
6 {
7     $curl = curl_init();
8     curl_setopt_array($curl,
9     [
10         CURLOPT_RETURNTRANSFER => 1,
11         CURLOPT_URL => $url
12     ]);
13
14     $resp = curl_exec($curl);
15     curl_close($curl);
16
17     if($resp == "You have exceeded your allotted message quota. Please enable pay-as-you-go to r
18     {
19         return "expired_token";
20     }
21
22     return json_decode($resp);
23 }
24
25 function build_url($base,$token,$symbol)
26 {
27     $replace_symbol = str_replace("xsymbolx",$symbol,$base);
28     $replace_token = str_replace("xtokenx",$token,$replace_symbol);
29
30     return $replace_token;
31 }
32
33 ?>
```



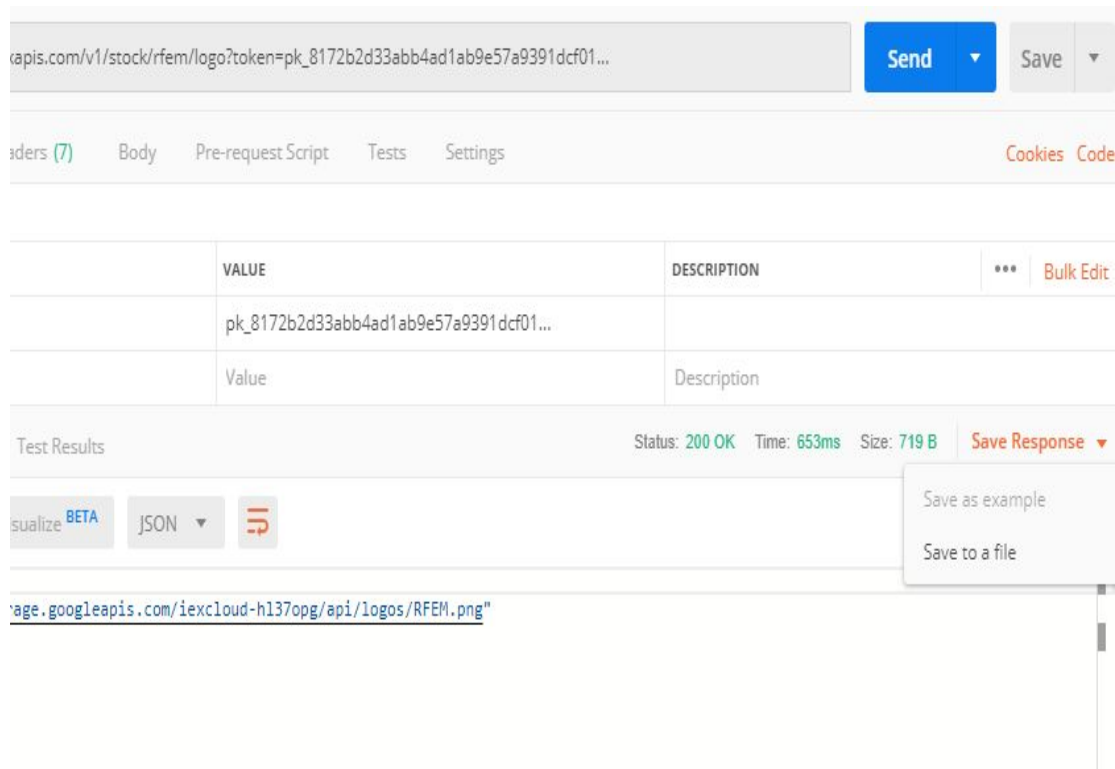
## After troubleshooting and testing

- Ensure that daily IEX API calls are made
- Take the JSON from the calls, process it, and store it in our MariaDB

```
?>
3 <?php
4
5 function fetch_json($url)
6 {
7     $curl = curl_init();
8     curl_setopt_array($curl,
9     [
10         CURLOPT_RETURNTRANSFER => 1,
11         CURLOPT_URL => $url
12     ]);
13
14     $resp = curl_exec($curl);
15     curl_close($curl);
16
17     if($resp == "You have exceeded your allotted message quota. Please enable pay-as-you-go to r
18     {
19         return "expired_token";
20     }
21
22     return json_decode($resp);
23 }
24
25 function build_url($base,$token,$symbol)
26 {
27     $replace_symbol = str_replace("xsymbolx",$symbol,$base);
28     $replace_token = str_replace("xtokenx",$token,$replace_symbol);
29
30     return $replace_token;
31 }
32
33 ?>
```

# Initial database setup

- We used Postman to visualize and perfect our API calls
- Once we got our proper calls, we downloaded the output into JSON files.



# IEX API calls used

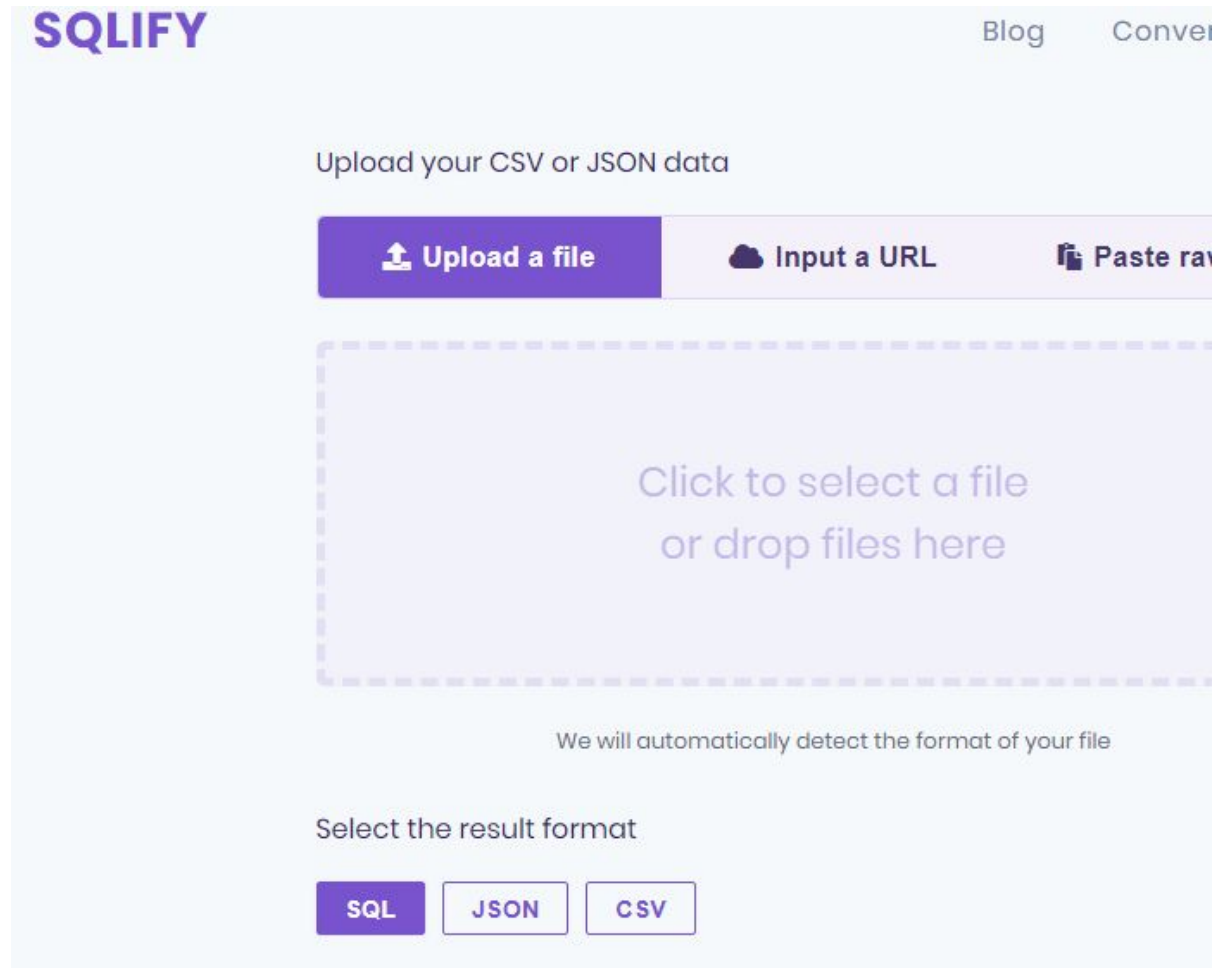
- We used these GET API calls from IEX to read and store our data into our databases
- <https://iexcloud.io/docs/api/#historical-prices>
- This was the going to be used for 5 year data pulls for Apple, Google, and RFEM. This contained stock info, such as open,close, etc.
- <https://iexcloud.io/docs/api/#logo>
- The above link shows and discusses the API call for getting company logos
- <https://iexcloud.io/docs/api/#news>
- This link, above, discusses getting news on respective companies.
- <https://iexcloud.io/docs/api/#company>
- The above link discusses using the IEX API to get basic info on a company.

# Getting the JSON to SQL form

- Several ways were available to convert our JSON to SQL.
- While our update script (for day to day updates) would have no issue making the conversion.....
- We ended up using a cool third party tool/website to assist with it.
- Once, the conversion was done, we took the SQL files, and made adjustments to them.
- Once the conversions were done, we imported them to our database using myPHPAdmin.
- Database creation, as well as user privilege, was also handled on myPHPAdmin.

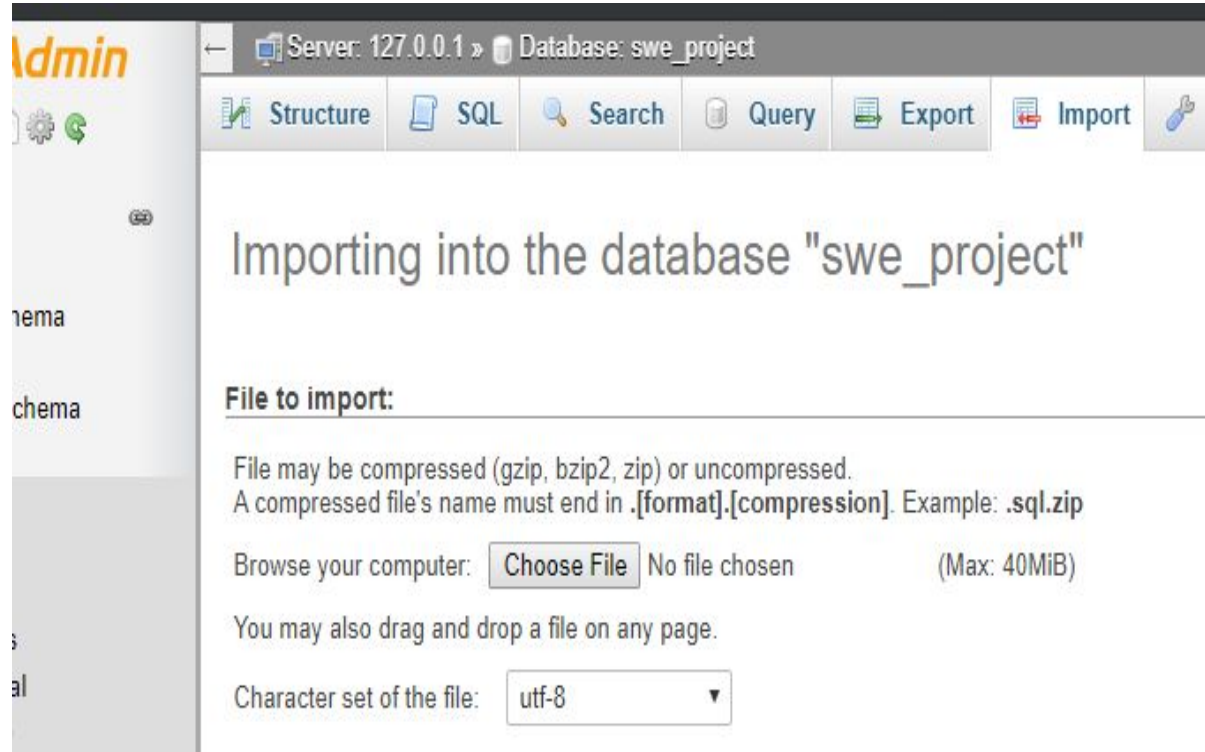
# The conversion tool

- <https://sqlify.io/converter>
- Strong tool
- Upload JSON
- Drop and add columns.
- Still needed to make some more adjustments in sublime text editor.



# Importing the SQL

- Once done, we imported to myPHPadmin.
- The conversion tool was for initial use.
- Any updates would be done being read, converted, and inserted with PHP and MySQL.



# SQL files on GitHub

- The SQL files here, have 5 years of data for google and apple
- For RFEM, it goes back to June 15, 2016.
- The SQL files also feature multiple table creation.
- The files go up October 24, 2019. Because that is when I made the 5 year calls for each of the 3 companies.
- After October 24, 2019, The sql files were updated on MariaDB & myPHPadmin.
- This was done using two PHP files, acting as scripts.

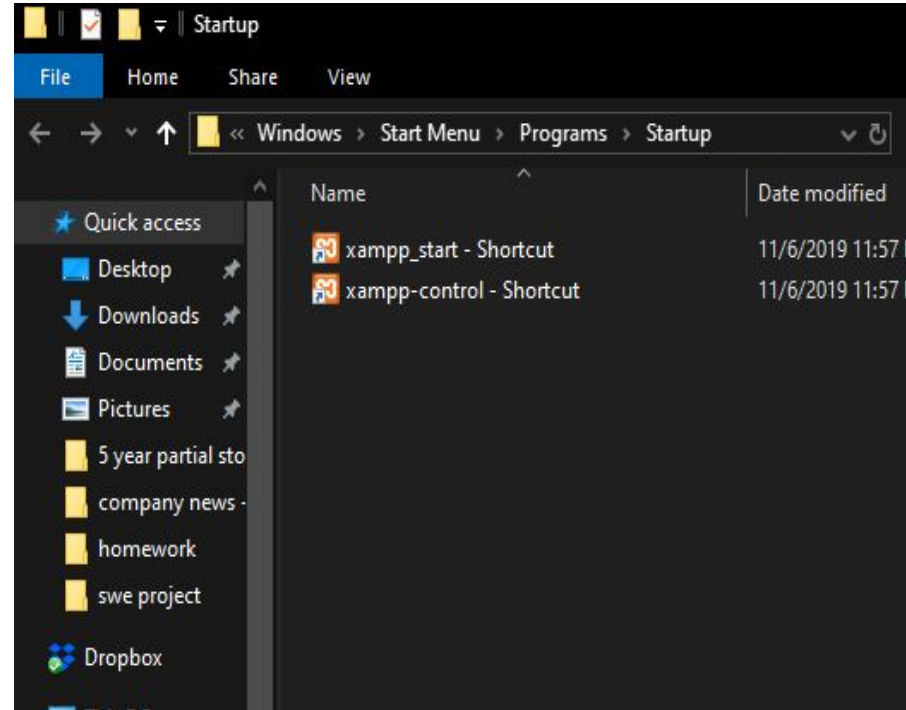
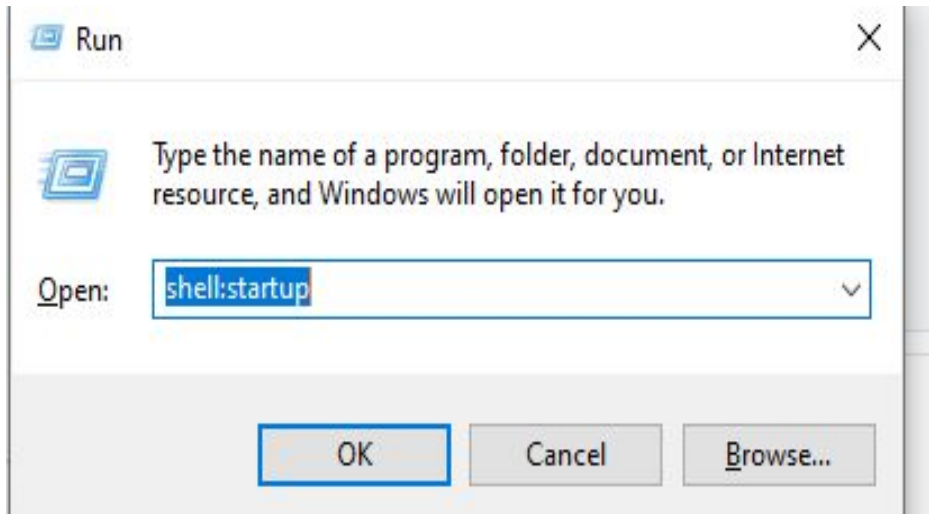
# Automation and daily updates

- We supported our daily update using Automation.
- We are developing in Windows Environment, so we used Windows Task Scheduler.
- This would require a few things...
- Some useful links for all....
- <https://www.youtube.com/watch?v=C4PdPqEOo6A>
- <https://code-examples.net/en/q/f81336>
- <https://www.computerhope.com/issues/ch001666.htm>



# First things first

- Any update done to the database required having our Apache and MySQL on.
- Thus when we automated our computer to turn on, when off, we had to include those in the start procedure.
- When turned on, we used our task scheduler to run our created batch file.
- The batch file would have php commands to files located in the htdocs folder.
- The php update scripts would be needed in that folder, in order for it to work.
- We had this all run daily, and automatically. If any issues, it would rerun until success
- Inside the PHP files, it had the IEX API urls and token. It would create a connection, read the JSON, convert it and send to the SQL database.



Getting XAMPP to start when the PC starts, so database and server connection is alive.



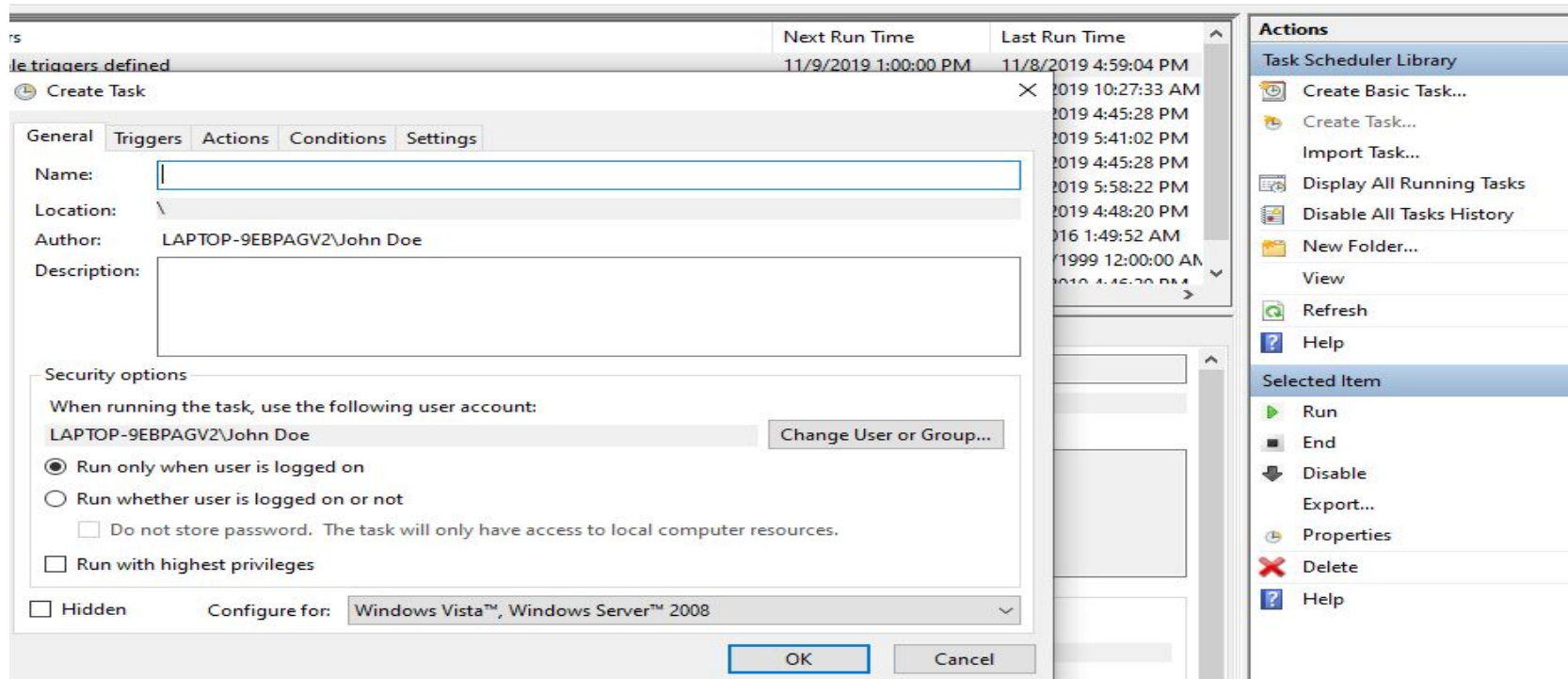
daily\_update - Notepad

File Edit Format View Help

```
C:\xampp\php\php.exe -f "C:\xampp\htdocs\swe project\stock_update.php"
```

```
C:\xampp\php\php.exe -f "C:\xampp\htdocs\swe project\news_update.php"
```

Simple batch file made to make calls to PHP compiler and PHP files. These files hold instructions to read and update.



Windows Task Scheduler....where we set up the automated task of reading the batch daily, and starting up the PC, if necessary










## Working with web and Postman

- Once we tested, and implemented our design, we had a working product.
- The product, as displayed in class, can work with postman, or work with our web front end.
- Our web front end had forms that connected to our PHP API.
- Our PHP API would read and calculate the SQL query and convert, finally to JSON output.

# Info about how to test it on both Web and Postman

- Next few images will be self explanatory.
- It will show how we have true working APIs.
- We make our requests on the web, and get JSON from there.
- We can then take the same URL, and paste it into Postman, and get same result.
- We will be working with 'most recent 30 days API'. But the same principles apply to all our API php files, on our github.

# Index of /swe project

| <u>Name</u>  | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|--|----------------------|-------------|--------------------|
|  <a href="#">Parent Directory</a>     |                      | -           |                    |
|  <a href="#">SQL/</a>                 | 2019-11-08 23:16     | -           |                    |
|  <a href="#">company_info.html</a>    | 2019-10-30 21:09     | 1.9K        |                    |
|  <a href="#">company_info_API.php</a> | 2019-11-05 19:52     | 1.0K        |                    |
|  <a href="#">custom_dates.html</a>    | 2019-10-30 21:09     | 2.2K        |                    |
|  <a href="#">custom_dates_API.php</a> | 2019-11-07 16:03     | 5.9K        |                    |
|  <a href="#">logo.html</a>            | 2019-10-30 22:36     | 1.9K        |                    |
|  <a href="#">logo_API.php</a>         | 2019-11-05 19:48     | 1.1K        |                    |
|  <a href="#">mysqli_connect.php</a>   | 2019-10-25 21:39     | 348         |                    |

## Company's most recent 30 trading days

---

Information, such as date, open, volume change percentage, etc are displayed.

Enter either: aapl (for Apple), googl (for Google), or rfem (for Riverfront)

company:



localhost/swe%20project/thirty\_days\_API.php?\_company=aapl&submit=submit+your+information

```
{ "row_ID": "1268", "date": "2019-11-07", "close": "258.74", "open": "259.43", "high": "260.35", "low": "258.11", "volume": "23735083", "change":  
18.24", "volumeChange": "4768959", "volumeChangePercent": "25.14", "volumeChangeOverTime": "-6.41" }, { "row_ID": "1267", "date": "2019-11-  
257.49", "low": "255.37", "volume": "18966124", "change": "-0.28", "change_percent": "-0.11", "ChangeOverTime": "17.34", "volumeChange": "-10  
volumeChangeOverTime": "-25.22" }, { "row_ID": "1266", "date": "2019-11-05", "close": "257.05", "open": "257.13", "high": "258.19", "low": "256.  
change_percent": "-0.11", "ChangeOverTime": "17.47", "volumeChange": "-5843525", "volumeChangePercent": "-22.63", "volumeChangeOverTime  
close": "257.33", "open": "257.5", "high": "257.85", "low": "255.38", "volume": "25817952", "change": "7.79", "change_percent": "3.12", "ChangeOv  
volumeChangePercent": "-31.66", "volumeChangeOverTime": "1.80" }, { "row_ID": "1264", "date": "2019-11-01", "close": "249.54", "open": "255.8  
37781334", "change": "2.30", "change_percent": "0.93", "ChangeOverTime": "14.04", "volumeChange": "2990814", "volumeChangePercent": "8.60"  
1263", "date": "2019-10-31", "close": "247.24", "open": "248.76", "high": "249.17", "low": "237.26", "volume": "34790520", "change": "2.48", "chan  
volumeChange": "3659998", "volumeChangePercent": "11.76", "volumeChangeOverTime": "37.18" }, { "row_ID": "1262", "date": "2019-10-30", "cl  
ow": "241.21", "volume": "31130522", "change": "-4.21", "change_percent": "-1.69", "ChangeOverTime": "11.85", "volumeChange": "-4579345", "v  
volumeChangeOverTime": "22.75" }, { "row_ID": "1261", "date": "2019-10-29", "close": "248.97", "open": "243.29", "high": "249.75", "low": "242.5  
change_percent": "0.63", "ChangeOverTime": "13.78", "volumeChange": "11566626", "volumeChangePercent": "47.91", "volumeChangeOverTime":  
close": "247.42", "open": "249.05", "high": "249.25", "low": "246.72", "volume": "24143241", "change": "4.26", "change_percent": "1.75", "ChangeC  
volumeChangePercent": "31.43", "volumeChangeOverTime": "-4.80" }, { "row_ID": "1259", "date": "2019-10-25", "close": "243.16", "open": "246.5  
18369296", "change": "-0.42", "change_percent": "-0.17", "ChangeOverTime": "11.12", "volumeChange": "453041", "volumeChangePercent": "2.53"  
1258", "date": "2019-10-24", "close": "243.58", "open": "244.51", "high": "244.8", "low": "241.81", "volume": "17916255", "change": "0.40", "chang  
volumeChange": "-2016290", "volumeChangePercent": "-10.12", "volumeChangeOverTime": "-29.36" }, { "row_ID": "1257", "date": "2019-10-23",  
ow": "241.22", "volume": "19932545", "change": "3.22", "change_percent": "1.34", "ChangeOverTime": "11.13", "volumeChange": "-2751456", "vo  
volumeChangeOverTime": "-21.41" }, { "row_ID": "1256", "date": "2019-10-22", "close": "239.96", "open": "241.16", "high": "242.2", "low": "239.6
```

Pay attention to URL. We will copy and paste same URL into POSTMAN, and use GET option.

The output right now is JSON.

← → ↻ ⓘ localhost/swe%20project/thirty\_days\_API.php?\_company=aapl&submit=submit+your+information

```
[ { "row_ID": "1268", "date": "2019-11-07", "close": "258.74", "open": "259.43", "high": "260.35", "low": "258.11", "volume": "23735083",  
"18.24", "volumeChange": "4768959", "volumeChangePercent": "25.14", "volumeChangeOverTime": "-6.41" }, { "row_ID": "1267", "date":  
"257.49", "low": "255.37", "volume": "18966124", "change": "-0.28", "change_percent": "-0.11", "ChangeOverTime": "17.34", "volumeCha  
"volumeChangeOverTime": "-25.22" }, { "row_ID": "1266", "date": "2019-11-05", "close": "257.05", "open": "257.13", "high": "258.19", "1  
"change_percent": "-0.11", "ChangeOverTime": "17.47", "volumeChange": "-5843525", "volumeChangePercent": "-22.63", "volumeChange
```

GET

Untitled Request

×

+

...

No Environment

👁

⚙

Untitled Request

Comments (0)

GET

Enter request URL

Send

Save

Params

Authorization

Headers

Body

Pre-request Script

Tests

Settings


Cookies

Code

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description |     |           |

Response



GET http://localhost/swe%20project...

No Environment

Untitled Request

Comments (0)

GET

http://localhost/swe%20project/thirty\_days\_API.php?\_company=aapl&submit=submit+your+information

Send

Save

Params

Authorization

Headers

Body

Pre-request Script

Tests

Settings

Cookies

Code

Query Params

|                                     | KEY      | VALUE                   | DESCRIPTION | ... | Bulk Edit |
|-------------------------------------|----------|-------------------------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | _company | aapl                    |             |     |           |
| <input checked="" type="checkbox"/> | submit   | submit+your+information |             |     |           |
|                                     | Key      | Value                   | Description |     |           |

Response

Pasting the URL generates the key fields

- Click send
- Wait
- Switch the option to JSON (drop down is near visualize beta button)
- And there you have it, a working RESTful API that supports the GET method.
- The output is in pretty JSON
- Any questions or comments, feel free to ask me in class or email
- --Andrew Nashid
- [nashidan@kean.edu](mailto:nashidan@kean.edu)

GET http://localhost/swe%20project...



No Environment



## Untitled Request

Comments (0)

GET

http://localhost/swe%20project/thirty\_days\_API.php?\_company=aapl&submit=submit+your+information

Send

Save

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies Co

### Query Params

|                                     | KEY      | VALUE                   | DESCRIPTION | ... | Bulk Ec |
|-------------------------------------|----------|-------------------------|-------------|-----|---------|
| <input checked="" type="checkbox"/> | _company | aapl                    |             |     |         |
| <input checked="" type="checkbox"/> | submit   | submit+your+information |             |     |         |
|                                     | Key      | Value                   | Description |     |         |

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 278ms

Size: 12.38 KB

Save Response

Pretty

Raw

Preview

Visualize BETA

JSON



```
1 [
2   {
3     "row_ID": "1268",
4     "date": "2019-11-07",
5     "close": "258.74",
6     "open": "259.43",
7     "high": "260.35",
8     "low": "258.11",
9     "volume": "23735083",
10    "change": "1.97",
```

# Fin

Questions? Comments?