Web Infrastructure

Assignment #4

REST Web Service, Client and User Interfaces

Introduction:

In this assignment, you will create APIs and a website that uses JavaScript to obtain information from your API and display it to user.

Remember the course goals: to know how things work under the hood not to just get something working. So, you are restricted to bare-basics. No frameworks to do the REST routing for you, you must use Apache & htaccess and route/parse your REST APIs manually using native methods of the scripting language of your choice.

For example: Understanding how REST parameters are actually parsed and how web server actually sees it is what we are after not just getting it working using a library or a framework that does it for you.

Part #1: 10 points

In this part, you will create and add a subdomain named "BART" to your domain that you created in Assignment 1. (For example: BART.mydomain.com). This subdomain will point to a new web directory tree on your Amazon WS's webserver.

i) Create a sub directory to host the new web site

ii) Make sure you change the ownership and permissions on the subdirectory using the following commands:

```
sudo chown ubuntu bartdir (Ubuntu)
sudo chmod -R o+r bartdir
```

iii) To create a subdomain, login to your domain manager and create a subdomain

iv) Configure your apache virtual hosts to link to the subdomain's directory

v) Edit /etc/httpd/conf/httpd.conf

vi) Enable name-based virtual hosting by uncommenting the line: NameVirtualHost *:80

vii) Link your subdomain by adding in the following lines:

```
<VirtualHost *:80>
        DocumentRoot /var/www/html/bartdir
        ServerName bart.yourdomain.com
</VirtualHost>
```

viii) After you are done with above, add support for port 443, and change above to auto-redirect to port443 (https) directly.

ix) Save your changes and restart Apache for your changes to take effect.

Note: You can use NodeJS if you prefer. Configure Virtual Host accordingly.

Part#2: Obtaining and parsing XML & JSON, 20 points

NOTE: Although BART API now supports JSON calls, we will only use the BART-XML APIs. It is one of this assignment's goals to get you to use both XML and JSON.

NOTE#2: For this assignment you will follow my "Religion" ☺ … REST APIs must be 100% in the form of a URI. No parameters, etc.

NOTE#3: As mentioned before, you are restricted to do-it-yourself tools. Read the note at the beginning

In this task, you should use the BART API (http://api.bart.gov/docs/overview/index.aspx). Use BART general-use API key: **MW9S-E7SL-26DU-VV8V** (Or sign up for your own key). Using its API documentation (link), find the resource that enables you to access BART's real-time departure estimates. Using this resource, **create the following APIs (that deliver JSON responses):**

i) /stations : Gets a list of BART stations using BART API as JSON

```
[ GET /stations ]
```

ii) /trips : Gives a detailed list of information for trains between supplied Source and Destination

```
[GET /trips/source/<STN_ABBR>/dest/<STN_ABBR>]
```

iii) /station : Gives Information about supplied Station

```
[GET /station/source/<STN_ABBR> ]
```

Note: You may need to add more APIs if you feel you need more to solve the requirements in Part #2

Part#2: User Interface, 20 points

Using your own REST APIs from Part #1 do the following functionalities in front end (you may use Front-End frameworks like Bootstrap or jQuery)

i) Uses the modern HTML Symantec tags. The following HTML5 elements must be included in the page:
Header Section Aside Headings (h1, h2, etc.) Footer
ii) Call **`/stations`** to populate your station list.
iii) Have the user select a departure station and arrival station.
iv) Call your **`/trips`** API and Display a list that includes
v) Real Time Departure time for trains to the destination station
vi) Fare (for each train route listed)
vii) Time of arrival
viii) Call **`/station`** to display the source station information in a sidebar (use <aside> HTML tag)
ix) Display a countdown (in steps of seconds) until the next train leaves toward arrival station.
x) Your website should update itself by re-loading the latest information from the BART API every 30 seconds.

Note: Be creative! Make your website look good. Use any online resource and don't forget to quote it. Your website will be validated with W3's HTML checker.

Part#3: I know you, 5 points

When loaded, your web page will display a message welcoming a user back to your site if he or she has visited your page on the browser before. In that case the welcome message will include the number of times this user visited your site.

- Hint: use localStorage.setItem / localStorage.getItem()

Part#4: Display Google route map, 5 points

1) Register for Google Maps API key (you can choose the standard plan for free)
2) Use Google maps JavaScript API and research how to do routing maps using public transit
`https://developers.google.com/maps/documentation/javascript/directions`
3) On the page in part#2, display a Google map of the route taken by the next departing train from departure station. You will need:
   a) the departure time of next train that you got in previous step
   b) The longitude & latitude of both stations (included in the **`/stations`** API result)

Hints:

- The BART API provides its data in XML format. Use an appropriate library to parse the responses and convert it to JSON. For PHP, you may refer http://php.net/manual/en/book.simplexml.php , http://outlandish.com/blog/xml-to-json/ and tons of other resources available online.
- For building REST APIs use appropriate library/framework for your server. For PHP, you may use refer https://www.leaseweb.com/labs/2015/10/creating-a-simple-rest-api-in-php/ or any popular framework.
- You can make use of the jQuery library for simplifying HTTP requests and parsing of the JSON.
- Your application should run in Firefox and Chrome.

Part#5: The Same-Origin Policy, 5 points

The JavaScript same-origin policy (Links to an external site) aims to prevent cross-site request forgery by limiting how resources on different hosts can interact with each other. Knowing this, why is it possible for your website to load and display data from the BART API? Include your answer as a comment within the HTML markup of your created website.

Optional: A jQuery Counter (Bonus)

Visualize the remaining time until the next train using the Countdown widget from the jQuery library.

Deliverables:

1) Upload all your code in <lastname>.zip to Camino for Assignment#4. Your code should to be documented and entries that need changing for testing is clearly shown at the beginning of the code.
2) As in Assignment#1: all login information needed to assess your setup, installation, etc.