Assignment 3 – Web Requests in Javascript  
COS318 – FA2018

Due Date: September 20th, 2018  
Turn in all files using Moodle

We have arrived at the third assignment. In this assignment, you will be making several different web requests to a server using javascript. The server endpoint you will be using is “http://webprogrammingassignment3.azurewebsites.net/api/favoriteCharacters” You do not need to write any server side code for this assignment. All references to the server in this assignment are already written for you and deployed to Azure at the above endpoint URL. Your requirement is only the client side javascript. The force will be with you, always.

1. **(40 Points) Favorite Characters**
   1. Create an html web page that contains a form. The form should contain three fields; first name, last name, and favorite Star Wars character. There must also be three buttons at the bottom with the values “Force Push,” “Force Pull,” and “Force Read.”
   2. The button “Force Push” will POST the data in the form to the server endpoint and then display the resulting response text. The data you post will be added to the server’s database. The response will contain only the data you just posted.
   3. The button “Force Pull” will GET the data from the server endpoint and then display the resulting response text. The response will contain the entire list of data stored on the server.
   4. The button “Force Read” will GET the data from the server endpoint /favoriteCharacters/{index}, passing in a random number for the index. You must check if the specified index exists on the server. The user should not notice that you are making this check to the server first. Then display the resulting text and the index that you requested from the server during the second GET. The response for the second request will contain only the data for the name and favorite character of the list index you requested.
   5. JSON key/value format: {FirstName: \*, LastName: \*, Character: \*}
2. **(40 Points) Views**
   1. Create a second form on the page with a text box labeled “View Date” and two buttons labeled “Force Insight” and “Watch Movies.”
   2. The button “Force Insight” will load the views data for the last character that was “Force Read.” The endpoint used is /favoriteCharacters/{index}/views
   3. The button “Watch Movies” will POST the data from the “View Date” form to the endpoint /favoriteCharacters/{index}/views.
   4. JSON key/value format: {ViewDate: \* }
3. **(20 Points)** Code style, formatting, completeness, and quality.
   1. Keep in mind that other classmates of yours are potentially using the server endpoints at the same time you are. Because of this, you might see some of their data returned to you too.
   2. The server’s database will only store thirty elements at a time. It will automatically reset the list to the base value if the number of elements gets too high.

Stretch Levels

If you already have a lot of experience with Javascript requests or you feel that the force is with you, try to complete these stretch levels for a reputation bonus. If you try for the stretch levels, make sure to type it in the comments on Moodle so I don’t miss it.

**Qui-gon Jinn Level**

Add some CSS to your page to make it look nicer. Background colors, font colors, or anything that looks good.

**Obi-wan Kenobi Level**

Add a fourth button to your html page, “Force Delete.” Send a DELETE to the endpoint /favoriteCharacters/{index} with a random index. Again, it should check first to make sure the index exists. Since DELETE requests return no content, after you perform the delete, query the entire list again, and display it. Also display the index that you deleted.

**Luke Skywalker Level**

You will need to complete Obi-wan Kenobi level first. Change “Force Delete” so that if “Force Sight” is used directly before it, it will delete the index that “Force Sight” returned. If “Force Sight” wasn’t just used, then just delete a random index as normal.

The Rules

1. No inline styles or javascript.
2. Error messages must be “in-page” i.e. no pop-ups or alerts.
3. Any resources not created by you (images, javascript libraries, etc.) must be referenced using a CDN or URL, not directly included in your assignment submission.