Assignment 3 – Web Requests in Javascript and C#
COS318 – FA2015

Due Date: September 24th, 2015
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 and C#. The server endpoint you will be using is "http://webprogrammingassignment3.azurewebsites.net/api/starWars" 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 requirements are only the client side javascript and C#. The force will be with you, always.

Here are your requirements.

1. **(40 Points)** Create an html web page that contains a form. The form **should contain two fields**; first name and favorite Star Wars character. There must also be **three buttons at the bottom** with the values "Force Push," "Force Pull," and "Force Sight."
   a. 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.
   b. 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.
   c. The button "Force Sight" will **GET the data from the server endpoint** /starWars/{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.
   d. JSON key/value format: {FirstName: *, Character: *}
2. **(40 Points)** Create a C# console program that prompts the user for their first name and how many times they have seen any of the Star Wars movies. Then ask the user if they want to PUT or PATCH the data.
   a. If the user chooses PUT, **PUT the data to the server endpoint** and display the resulting response text and status code. The data you put will replace the data on the server, and the response will contain only the data that you just put.
   b. If the user chooses PATCH, **PATCH the data to the server endpoint** and display the resulting response text and status code. The data you patch will be added (patched) to the database. The response will contain all of the data sent to the server so far.
   c. JSON key/value format: {FirstName: *, NumberOfTimes: *}
3. **(20 Points)** Code style, formatting, completeness, and quality.

The Rules

1. You may use jQuery to perform your javascript web requests if you prefer. If you do this, use a CDN for jQuery reference. Do not include jQuery code in your assignment submission.

2. 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.
3. 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.
4. You should go see Star Wars Episode 7: The Force Awakens when it comes out.

Stretch Levels

If you already have a lot of experience with Javascript and C# requests or you feel that the force is with you, try to complete these stretch levels for extra credit. The levels are cumulative, so for example, don't try for silver if you haven't finished bronze. If you try for the stretch levels, make sure to type it in the comments on Moodle so I don't miss it.

Bronze Level (Qui-gon Jinn)

In the C# console program, display the status code that is being sent back from the server in addition to the response text.

Silver Level (Obi-wan Kenobi)

Add a fourth button to your html page, "Force Delete." Send a DELETE to the endpoint /starWars/{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.

Gold Level (Luke Skywalker)

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.