Assignment 3 – Web Requests in Javascript  
COS318 – Web Programming

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 “https://webrequestsserverbethel.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. **HTML:** 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 Pull,” “Force Push,” and “Force Read.”
   2. **Force Pull:** 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.
   3. **Force Push**: 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 the data you just posted. The JSON you send to the server must have the fields FirstName, LastName, and FavoriteCharacter.
   4. **Force Read:** The button “Force Read” will GET the data from the server endpoint /favoriteCharacters/{index}. The {index} will be a random index from the list on the server.
      1. In addition to the data that came back from the server, also display on the page the random index that was chosen.
      2. Hint: To get a random index, first make a normal GET request like you did for Force Pull, then use the length of the result to get a random index. Then immediately make a second request to /favoriteCharacters/{index}. (Don’t display any results from the Force Pull used to get the length of the data).
2. **(40 Points) Views**
   1. **HTML:** Create a second form on the page with a text box labeled “View Date” and two buttons labeled “Force Insight” and “Watch Movies.”
   2. **Force Insight:** 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. **Watch Movies:** The button “Watch Movies” will POST the data from the views form to the endpoint /favoriteCharacters/{index}/views. The JSON you send to the server must have the field ViewDate.
3. **(20 Points)** Code style, formatting, completeness, and quality.
   1. Keep in mind that other classmates of yours are 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 30 entries at a time. It will automatically empty the list if the number of entries 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 Read” is used directly before it, it will delete the index that “Force Read” returned. If “Force Read” was not just used, then just delete a random index as normal.

The Rules

1. No inline styles or inline 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.