

Assignment 4 – WebApi  
COS318 – FA2015

Due Date: October 8<sup>th</sup>, 2015  
Turn in all files using Moodle

In the fourth assignment, we will be returning to Star Wars. “I think my eyes are getting better. Instead of a big dark blur, I’m seeing a big bright blur.” In the third assignment, you wrote javascript and C# code that communicated with a server that was hosted on Azure. In this assignment, you’ll be recreating that server. You’ll be able to use your assignment three code to test your new server. Also, the server from assignment three is still deployed to Azure at <http://webprogrammingassignment3.azurewebsites.net/api/starWars> so you can verify if your server is acting in the same way as the reference. Your server will be tracking two different lists, Characters and NumberOfTimes. Your server does not need to persist any data. It is okay if any data is lost when your server is stopped.

Your StarWars controller must support the following verbs:

1. **(10 Points) GET**
  - a. Return a JSON array of all of the favorite characters currently stored on the server.
2. **(10 Points) GET/{index}**
  - a. Return a JSON object containing the favorite character and name data for the specified index.
  - b. Validate that the index sent in is valid.
3. **(20 Points) POST**
  - a. Accept JSON data for FirstName and Name and save it on the server.
  - b. Validate that the data being sent into the server is valid and not empty.
  - c. Return the appropriate JSON data for a POST request.
4. **(20 Points) PUT**
  - a. Accept JSON data for FirstName and NumberOfTimes. Replace all the data on the server for FirstName and NumberOfTimes.
  - b. Validate that FirstName is not empty and NumberOfTimes is an int.
  - c. Return the appropriate JSON data for a PUT request.
5. **(20 Points) PATCH**
  - a. Accept JSON data for FirstName and NumberOfTimes. Append that data to the list of FirstName and NumberOfTimes already on the server.
  - b. Validate that FirstName is not empty and NumberOfTimes is an int.
  - c. Return a JSON array of all of the items in the NumberOfTimes list.
6. **(20 Points) Code style, formatting, completeness, and quality.**

The Rules

1. Both lists (Character or NumberOfTimes) should begin with one entry.
2. If either list of names (Character or NumberOfTimes) gets larger than 30, empty it, and then reset it back to its single starting value.
3. Visual Studio projects contain a lot of files. Zip up your entire project directory and submit only a single file to Moodle.

4. Seriously, you really 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 WebApi, or if you aren't frozen in carbonite, 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 (Purple Lightsaber)**

Add middleware or a filter to your server that will reject the request before the controller code is executed if the verb is POST, PUT, or PATCH and the content type isn't application/json.

##### **Silver Level (Green Lightsaber)**

Modify your controller so that if extra fields are sent along in the JSON payloads, your server will reject the request. Here's a hint: `JsonInputFormatter`

##### **Gold Level (Blue Lightsaber)**

The in-memory list for `NumberOfTimes` that you are using isn't thread safe. Change it so that it is thread safe. Feel free to skip changing the `Characters` list however; that one would be much harder anyway.