# *Web Programming V (420-C50-HR)*

# *Lab 08 – JSON, jQuery & .NET*

Date assigned: Wednesday, November 8, 2017

Date due: **Wednesday, November 8, 2017, 5:50 p.m.**

**Lab is marked during the lab session and is due by the end of the lab session. Exceptions must be cleared by me in advance of the deadline.**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will have:

* Work with AJAX
* Work with jQuery
* Work with json files using jQuery
* Convert from XML to JSON
* Work with JSON and .NET
* Serializing and Deserializing JSON in .NET

Lab Setup

1. Attached is also a zip file which contains the source files you will need for the lab.
2. Create a folder to use for all parts of the lab called *username\_C50\_L08*. Add all the files from the C50\_L08\_Files zip file (on Moodle) to this folder.

To do:

Part A: JSON and .NET

NOTE: There are two sample solutions that will provide a starting point for you. See:

1. JSONNET Example for a simple use of reading/writing JSON files and enum handling
2. JSONDataStorageExample for dealing with lists/arrays

Put your solution in the PartA folder provided.

1. Create a JSON file called people.JSON with an array of people with the following data in it

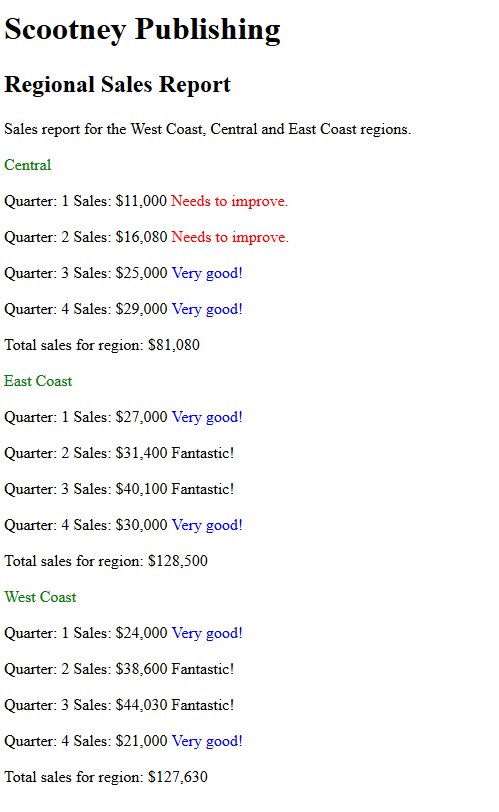
|  |  |  |  |
| --- | --- | --- | --- |
| name | age | Job | gender |
| Mike | 20 | teacher | Male |
| Suzanne | 19 | chef | Female |
| Robin | 18 | ninja | Pangender |
| Harambe | 17 | gorilla | Male |
| Thor | 3166 | Norse God | Male |

Implementation notes:

1. For the purposes of this simple exercise, you may assume that *name* is unique. Optionally, you can assign a unique Id to each person that is not used in the JSON file but is used in the in-memory model.
2. In your model, Gender is an enum with 3 possible values as shown in the sample table above. ( For those of you who are curious about gender identity, Facebook no longer has a limited list, but here’s [some](https://alphabetizer.flap.tv/lists/list-of-facebook-gender-options.php). We’ll just deal with the 3.)
3. Create an MVC .NET project that reads in this file at startup and keep an in memory list of people. This is the only time the people.JSON file is ever read into your code.
   1. Provide views which List the people and provide Create, Edit and Delete methods. (All provided by the MVC templating!)
   2. Do not update the datafile unless a “Commit Changes” is triggered by the user. Put a “Commit Changes” button somewhere in your Index view. **Only** when this button is selected is the people.JSON datafile written out. (You will do a controller action for this). Provide feedback to the user that the Commit was pressed such as an alert or a noticeable animation/transition.
   3. Shutdown and startup your Application again, make sure that your commit changes really works.
   4. If any of the “Detail” links are clicked in your list view, you will have a pop-up window showing the Person detail for the person selected. This will be implemented as:
      1. A new action on your controller that returns a JSON object (the person object selected)
      2. Client-side code which gets the JSON object and presents the pop up. Style Name, Age, Job and Gender as separate classes and make it look nice.

Part B: jQuery and JSON vs XML and XSLT

Let’s look at functional equivalency and compare an XML/XSLT solution with a JSON/jQuery solution. You will put your solution in the PartB folder along with the input files provided.

* 1. Take a look at the PartB folder. You will find the sales.xml file and a transformation that will produce a report. Load up the sales.xml file into a browser to checkout the output format and functionality.
  2. Create sales.JSON file that has the equivalent data as the sales.xml. The JSON file must be equivalent to the XML file, with the same hierarchy, data, in the same order.
  3. Create a web page that does an asynchronous retrieve of the sales.JSON and build up the javascript/jQuery to make the same report.
  4. Your report should look like: 
  5. Compare the code of the original XML/XSLT version to the JSON/jQuery model. Which approach would you choose (assume you had a choice of receiving data in either XML or JSON) and why? What is easier to do in XSLT vs javascript/jQuery?

(**answer below**).

**To submit**

When you have completed the lab exercise, create a zip file of the folder (including this file with your answer to questions, sub-folders PartA, PartB) and load it to the Moodle page.