Module 05 Lab 01 Worksheet

Accessing Data with JSON

# Overview

This lab will give you an introduction to parsing JSON-formatted data with C# and .NET. Since we'll be using a new software package for this project, let's install a new VS Code extension to make that easier.

* Open VS Code.
* Click on the Extensions icon in the Side Bar.
* In the search field, type NuGet.
* Select NuGet Package Manager from the listed results and click on Install.
* Restart VS Code when prompted before continuing.

Now you'll be able to search for and automatically install .NET and ASP.NET packages in your VS Code projects.

|  |  |
| --- | --- |
| * Using the aspnet generator, create a console application and name it Mod05Lab01. * Open the project folder in VS Code. * Extract the people.json file from people.zip to your project folder. * Press F1 to get the command palette. * Type NuGet and select Add New Package * Type the package name Newtonsoft.Json and press <Enter>. * Press <Enter> to select the latest (non-beta) version. Restore dependencies when prompted before continuing. | 0 |
| Open people.json  How many objects are listed? (5 pts.)  10  How many properties does each JSON object have? (5 pts.)  List them. (5 pts.) | 15 |
| Create a Person class file with the same properties. (Use your previous lab work for reference).  Copy and paste your Person class definition here. (10 pts.) | 10 |
| Now we're going to read in the JSON file and *deserialize* the objects listed. That is, we're going to convert the contents into a List of Person objects.  In order to do that, we're going to need some extra packages. Add the following namespaces to Program.cs:  System (for stuff like Console.WriteLine)  System.IO (for reading/writing files)  Newtonsoft.Json (for working with JSON data)  System.Collections.Generic (provides our generic List container type)  First let's convert the contents of our JSON file into a string. In Main(), add the line:  string personStr = File.ReadAllText("people.json");  (This is a System.IO method that reads all of the lines of a given file and returns a string consisting of the file contents.)  Use Console.WriteLine() to output the contents of personStr. What do you see? (5 pts.) | 5 |
| Remove the Console.Writeline() call and replace it with a line that creates a new List variable of type Person called PersonList.  Copy and paste the code here. (5 pts.) | 5 |
| Next we're going to use our Json.NET framework to *deserialize* PersonStr into a set of Person objects and store them in PersonList.  PersonList = JsonConvert.DeserializeObject<List<Person>>(personStr);  (this should be all on one line.)  **NOTE**: You may wish to surround this with a try/catch block, just in case.  Make sure this runs without errors, then try to print the contents of PersonList with Console.Writeline.  What output do you see? ( 5 pts.) | 5 |
| PersonList is a Collection data structure, so we can use the foreach() method to iterate through each member with the Console.WriteLine method.  What is the result? (5 pts.) | 5 |
| We can also treat a List like an array. Use this idea to print out the LastName of the first Person element on PersonList. (Remember Person is an object data type.)  What code would you use? (5 pts.)  What was the output? (5 pts.) | 10 |
| Given what you've learned so far, modify your code to output the JSON data to the console as an ASCII table. Remember to put appropriate headers on each column and format the output to be easily readable. (40 pts.) | 40 |
| **Total** |  |

Complete this worksheet and submit it to your instructor along with your project folder in a ZIP file..