Joshua Catoe

CSCI 350

Reflection Assignment Tutorial 3 - Intro to Asynchronous Programming with REST

Async Part 1. Intro to Asynchronous Programming with REST Example

Complete the tutorial at

<https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/console-webapiclient>

As you’re working through the tutorial, answer questions 1-11.

1. Which namespace defines the type HttpClient?

**The** System.Net.Http **namespace defines the** HttpClient **type.**

When a program needs to retrieve data or call a service over the Internet, the most common way to do this (in any language), is to create a send an HTTP Request object. An HTTP Request object is made up of a body and a set of headers, that describe the request. Review <https://docs.microsoft.com/en-us/dotnet/api/system.net.http.headers.httprequestheaders?view=netframework-4.8> and answer the following questions 2-5:

1. Which header describes the type of application making the request?

**The** User-Agent **header describes the type of application making the request.**

1. Which header indicates whether a cached copy of the requested resource is acceptable?

**The** Cache-Control **header indicates if a cached copy of a requested resource is acceptable.**

1. Which header contains authorization credentials for protected resources?

**The** Authorization **header contains authorization credentials for protected resources.**

1. Which header describes what language is expected for the response?

**The** Accept**-**Language **header describes the language expected for the response.**

Read the remarks section on the HttpClient api documentation page <https://docs.microsoft.com/en-us/dotnet/api/system.net.http.httpclient?view=netframework-4.8>

and answer the following questions 6-8.

1. If an app using the HttpClient intends to download large amounts of data, what will happen if it uses the default buffering method? What is the recommended threshold amount of data for changing from the default buffering method?

**The recommended threshold for streaming data instead of buffering is 50MB. If the app uses buffering for an amount of data larger than that, its memory usage will become quite large and performance will be reduced.**

1. Why are HttpClient variables typically declared as static? What might happen if they are not?

**If** HttpClient **variables were declared for every request, the number of sockets available may be used up, thus resulting in** SocketException **errors.**

1. How many constructors does HttpClient provide?

HtttpClient **provides three constructors:**

HttpClient() – **initializes instance of** HttpClient **with default handler**

HttpClient(Http MessageHandler) **– initializes instance of** HttpClient **with specified handler**

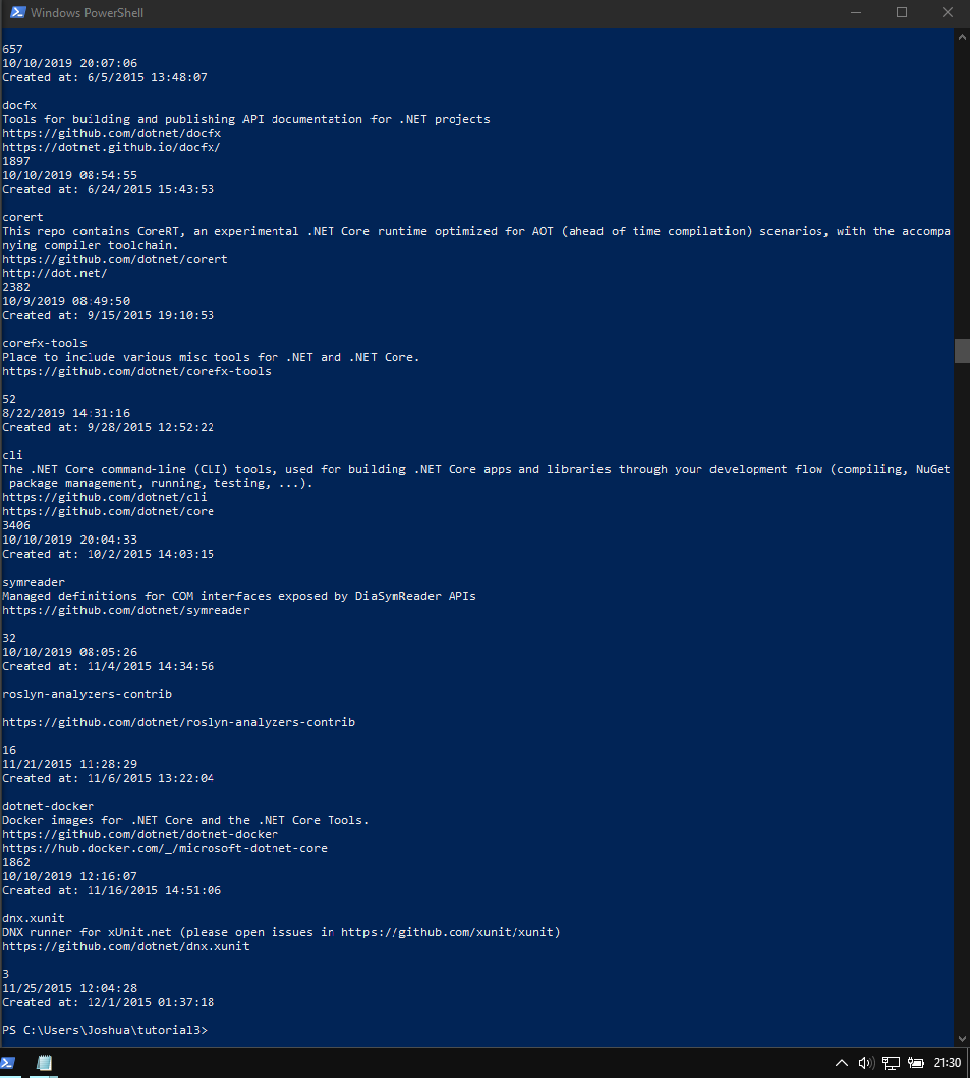
HttpClient(Http MessageHandler, Boolean) – **initializes instance of** HttpClient **with specified handler and determines if handler should be disposed**

9. How do you define a read only property in C#?

**Read-only properties are defined by only using the** get **accessor and not the** set **accessor.**

10. Modify repo.cs and Program.cs to also display the created\_at date (in addition to the pushed\_at date). Describe your modifications here. Include a screenshot of the new output. (This same answer will be the submission for your Tutorial assignment).

**In** repo.cs**, I created a new private string data member called** CreatedDate **that uses the** get/set **auto-implemented property to return the** created\_at **date and a new method,** Created**,** **that formats the created date and converts it to a C# compatible format. In** Program.cs**, I simply added the line** Console.WriteLine(repo.created) **to the** foreach **loop in** Main**, which prints the created date formatted by the** Create **method in** repo.cs**.**



11. Write a paragraph about JSON. Give an example that shows the syntax.

**JSON (JavaScript Object Notation), created by Douglas Crawford, is a human-readable, language-independent format used to store and retrieve data. It consists of two types of data structures: name/value pairs (objects) and/or an ordered list of values (arrays). JSON uses six different data types, those being** objects**,** arrays**,** numbers**,** strings**,** Booleans**, and** null**. The syntax of JSON is rather simple:** objects **are held within curly braces, name/value pairs are in quotes, bound together with a colon and separated by a comma, and** arrays **are held within square brackets with their contents separated by commas. The following is an example of JSON that contains a** car **object consisting of two cars.**

{

“car”: [

{

“make”: “Dodge”,

“model”: “Charger”,

“year”: “2009”

“stolen”: “no”

},

{

“make”: “Honda”

“model”: “Civic”

“year”: “2001”

“stolen”: “yes”

}

]

}