Using APIs breakout

in list Breakouts

Description

Add a more detailed description…

API

Delete.

0%

- [ ] 1) Go to your API's website
- [ ] 2) Find a "Getting Started"
- [ ] 3) It should give you a URL to use
- [ ] 4) Note that URL
- [ ] 5) Make note of any indexes in the data and how they correlate with each other. IE in the weather API Temp label indexes match the rest of the lists.
- [ ] NOTE: Every API is going to be a little different: the tags will be organized different, you might be required to sign up or apply for credentials. When using APIs be flexible.

Visual Studio

Delete.

0%

- [ ] 0) Can't find a URL? Use this one as a test: "http://forecast.weather.gov/MapClick.php?lat=42.335722&lon=-83.049944&FcstType=json"
- [ ] 1) Create an MVC App, using the MVC template
- [ ] 2) go into your homecontroller
- [ ] 3) Create a getData method with the following signature
- [ ] 4) public ActionResult GetData()
- [ ] 5) Create a data() method in your controller and a data view
- [ ] 6) In the data method, set your return to be Getdata()
- [ ] 7) HttpWebRequest request = WebRequest.CreateHttp("URL"); Paste this into your get data method
- [ ] 8) Add in you APIs URL, this is calling the API on the remote server.

- [ ] 9) next paste in this line: request.UserAgent = @"User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36";
- [ ] 10) the user agent specifies information about your client, namely their browser and OS
- [ ] 11) NOTE: Some APIs require keys, those keys will go around here, every APIs keys will function a little different, many won't have a key, but it needs to go before we pull out info from our request. It would look a little like: request.Headers.Add("X-Mashape-Key",value);
- [ ] 12) Next we need to get the response from the server, this is the reply from the server and it gives us back allllll it's info.
- [ ] 13) Paste in this line: HttpWebResponse response = (HttpWebResponse)request.GetResponse(); This is where we're asking the remote server for data.
- [ ] 14) Next we need to read in the raw data into a streamreader. Remember from file I/O a read allows us to look at an external file.
- [ ] 15) paste in: StreamReader rd = new StreamReader(response.GetResponseStream());
- [ ] 16) Now we need to store that raw data into a string, which will be easier to parse.
- [ ] 17) string ApiText = rd.ReadToEnd();
- [ ] 18) It's time to parse our string and turn it into a JSON object or XML object NOTE: This is where things change based upon whether you use JSON/XML. Most APIs come back in JSON, many do both, and very few are XML only.
- [ ] 19) We're going to do JSON to test if it worked, paste in this: ViewBag.ApiText = "The Current Temperature is "+ weatherData["data"]["temperature"][0];
- [ ] 20) Note that temperature is a collection and we can use the .toList() method to dump all it's data into a list. This gives access to all the methods and advantages present in a list.
- [ ] 21) either using viewbags or passing the list along to a view, print it all out or search up what you're looking for. Once you call .ToList() from then on out it will be a list of JTokens and you can use it as you'd use any other list!

Activity

Show Detai