Office_Hours02.22

Wednesday, February 15, 2023 8:09 PM

When using the code to change header to allow for CORS we ran into an Error: cannot edit HTTP header

- Said header was readonly

Is our solution correct?

- We use use dotnet addcors
  - Although this works we shouldn't do it since it obfuscates what's happening
- We should do it manually at the pipeline level

Do it at the web server level so the new person adding features knows whether or not to handle CORS

This is fine for development but in actual production we leave it to the **webserver** to handle

When trying to follow SOLID principles, we added IAnalyticsService we ran into an error with cache data

- Singleton so we don't have to constantly request the database
- In all lifetime objects we need to specify the concrete type we want and the last parameter is the implementation function
  - DI must know concrete type to implement as

If we want to use 1 service logger object for everything, we create it once and then call it when it is requested

Register it first so that if anything needs it, it will be known and requested
Builder.services.addsingleton
If many things need service factory it must be passed in, but this is not favorable
We want DI to handle instantiation logic

The first parameter is IServiceProvider, which is mentioned in DI service container, register logging service as a singleton so it isn't required to be instantiated

How to securely pass data from client to backend

- Webworker does ajax call to service in separate call
- Pass field value into webworker, gets encrypted, and returns the data to the client which does a post
- Only the thing which created the webworker has access
  If you want to avoid a webworker, all work must be done on the client

For adding scoped objects, if the request is from the same person will it be the same

object?

- Every method which needs the dependency will get the same instance throughout the whole request
- If a new request comes into the controller, a new instance of the dependency will be created, a new scope essentially

When will a scoped object be needed?

- Whenever an object needs a new instance since it is expensive to create
- If we want a clean transaction then multiple instances will be created

If you are using HTML input fields, then data will be sent in body as key-value pair

- In the post request, you can specify what that data looks like in the parameters
- The object structure correlates to C# Poco (plain old class object)
  Model Binding
  - Takes input from JSON and maps it to an object with the same property names
  - Not necessarily the same casing

AJAX DEMO: need to know what the data structure looks like

- Define client side what the data structure should be
- If the structure changes system will crash and we don't know why, knowing the data structure can help that
- Folders which contain the data type, structure, and what it contains
  - What if we have types shared across multiple features? Create a higher level folder called models

When handling User Authentication for the site (if they don't have access to a page) how do we handle that on client-side?

- Usually a client-side framework will have a router that has a guard/interceptor to check for access to page
- Homepage has logic built in since it must call the same method to check token (is user authenticated first? does it have permissions to a specific page?)
  - If they don't have access, there should be an error or a message saying "no access"
  - Don't even show the page so no operation can be performed at all
- Check server-side if token has permission, and client-side for whether or not it can be viewed at all
  - Client notices navigation event, client checks if access is allowed
  - Every page has an onload event which can check for perms
- 3 checks in total
  - On load
  - On navigation
  - On backend