2022

CA3 ASP.Net Core Blazor Project

BLAZOR WEB PAGE USING WEBAPI

X00160711 CIANAN NICOLAI

Contents

Introduction

> *An ASP.Net Core Client Side Blazor application is required which provides simple interactive functionality to display data from a public API and provide other simple features. A Single-Page Application is required implemented using Blazor (client side). 1/2 pages will suffice. The application should interact with a public API of your choice (GET only). The API should be RESTful (JSON), a public open API (with no authentication) is fine.*

GitHub Repo URL

https://github.com/CiananNicolai/ASP.Net-Core-Blazor-Project-API

URI for Deployed App

https://blazorchucknorris.azurewebsites.net/

URI for Video

https://youtu.be/z2Jij6-6Ev4

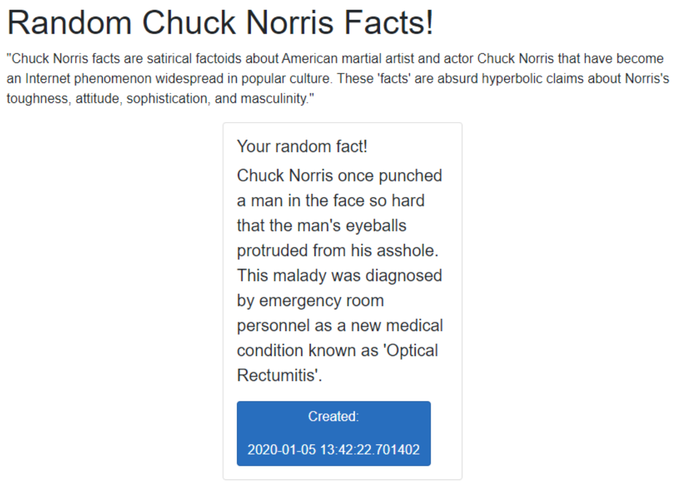
Description of System

This project uses the free API chucknorris.io to generate facts also known as jokes about Chuck Norris. The features that were intended to deliver upon where random jokes which is done on the first page, random joke by category selection and a search functionality to look through the jokes that contain that word.

Bootstrap for blazor otherwise known as blazorstrap.io was used to help with some quick basic formatting.

.Net packages that were used were the following, Microsoft.AspNetCore.Http and newtonsoft.

Screenshots



Random Chuck Norris Facts!

"Chuck Norris facts are satirical factoids about American martial artist and actor Chuck Norris that have become an Internet phenomenon widespread in popular culture. These 'facts' are absurd hyperbolic claims about Norris's toughness, attitude, sophistication, and masculinity."

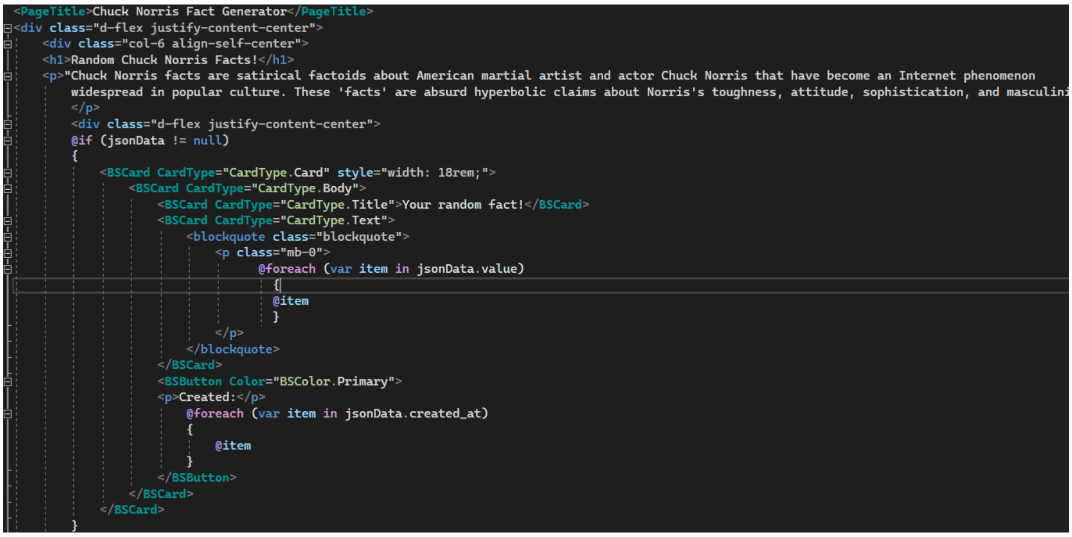
Your random fact!

Chuck Norris once punched a man in the face so hard that the man's eyeballs protruded from his asshole. This malady was diagnosed by emergency room personnel as a new medical condition known as 'Optical Rectumitis'.

Created:

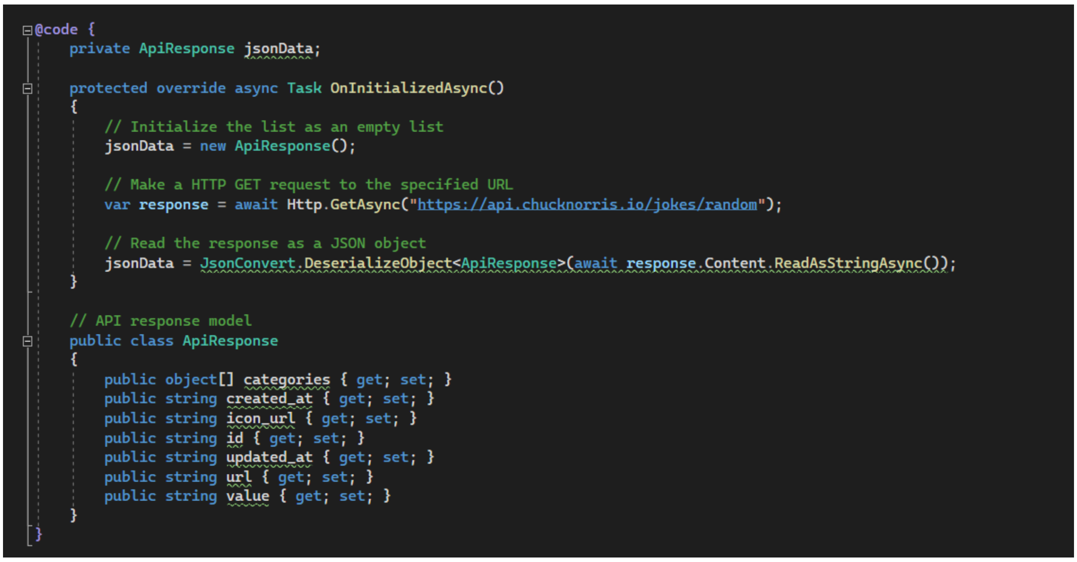
2020-01-05 13:42:22.701402

Figure Random Chuck Norris Facts



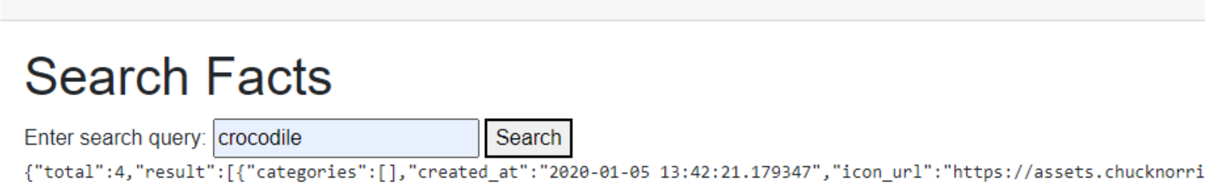
```
<PageTitle>Chuck Norris Fact Generator</PageTitle>
<div class="d-flex justify-content-center">
    <div class="col-6 align-self-center">
    <h1>Random Chuck Norris Facts!</h1>
    <p>"Chuck Norris facts are satirical factoids about American martial artist and actor Chuck Norris that have become an Internet phenomenon
        widespread in popular culture. These 'facts' are absurd hyperbolic claims about Norris's toughness, attitude, sophistication, and masculini
        </p>
        <div class="d-flex justify-content-center">
        @if (jsonData != null)
        {
            <BSCard CardType="CardType.Card" style="width: 18rem;">
                <BSCard CardType="CardType.Body">
                    <BSCard CardType="CardType.Title">Your random fact!</BSCard>
                    <BSCard CardType="CardType.Text">
                        <blockquote class="blockquote">
                            <p class="mb-0">
                                  @foreach (var item in jsonData.value)
                                    {
                                    @item
                                    }
                            </p>
                        </blockquote>
                    </BSCard>
                    <BSButton Color="BSColor.Primary">
                    <p>Created:</p>
                        @foreach (var item in jsonData.created_at)
                        {
                            @item
                        }
                    </BSButton>
                </BSCard>
            </BSCard>
        }
```

Here you can see me using the Blazor for Bootstrap to create the cards, mixed with a little bit of html to centre then display the jsonData.



```
@code {
    private ApiResponse jsonData;

    protected override async Task OnInitializedAsync()
    {
        // Initialize the list as an empty list
        jsonData = new ApiResponse();

        // Make a HTTP GET request to the specified URL
        var response = await Http.GetAsync("https://api.chucknorris.io/jokes/random");

        // Read the response as a JSON object
        jsonData = JsonConvert.DeserializeObject<ApiResponse>(await response.Content.ReadAsStringAsync());
    }

    // API response model
    public class ApiResponse
    {
        public object[] categories { get; set; }
        public string created_at { get; set; }
        public string icon_url { get; set; }
        public string id { get; set; }
        public string updated_at { get; set; }
        public string url { get; set; }
        public string value { get; set; }
    }
}
```

OnInitialisedAysnc() initializes the list, then uses http to GET from the json asynchronously. The jsondata uses newtonsoft to deserialize the json it pulled then populates the Apiresponse model.

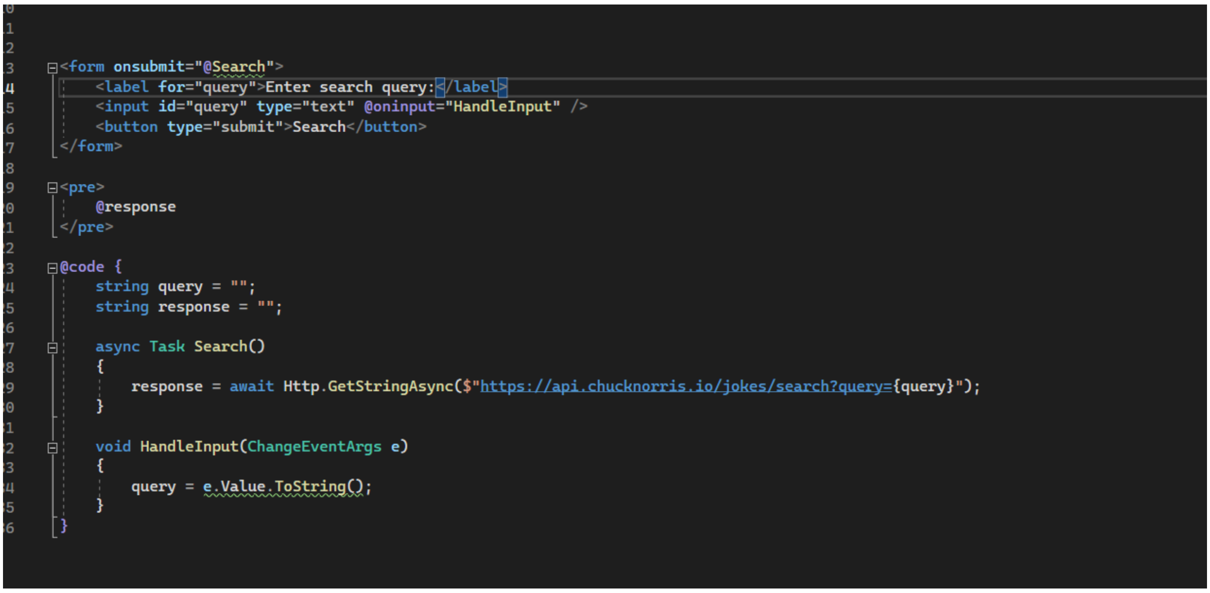


Search Facts

Enter search query: crocodile

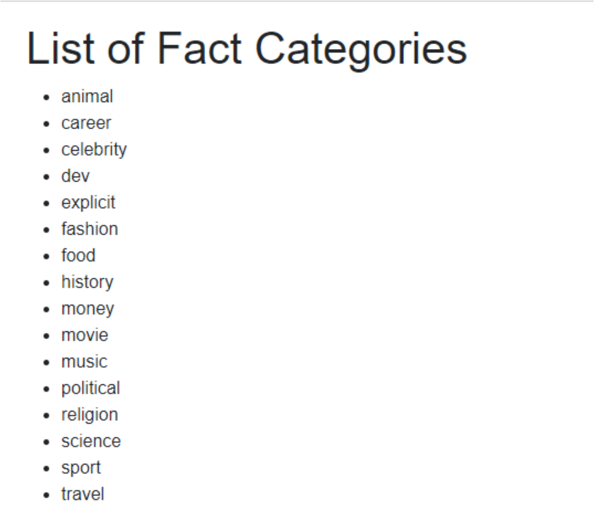
```
{"total":4,"result":[{"categories":[],"created_at":"2020-01-05 13:42:21.179347","icon_url":"https://assets.chucknorri
```

Figure 1 Search Chuck Norris Facts, as of writing this I struggled to display it correctly.



```
<form onsubmit="@Search">
    <label for="query">Enter search query:</label>
    <input id="query" type="text" @oninput="HandleInput" />
    <button type="submit">Search</button>
</form>

<pre>
    @response
</pre>

@code {
    string query = "";
    string response = "";

    async Task Search()
    {
        response = await Http.GetStringAsync($"https://api.chucknorris.io/jokes/search?query={query}");
    }

    void HandleInput(ChangeEventArgs e)
    {
        query = e.Value.ToString();
    }
}
```

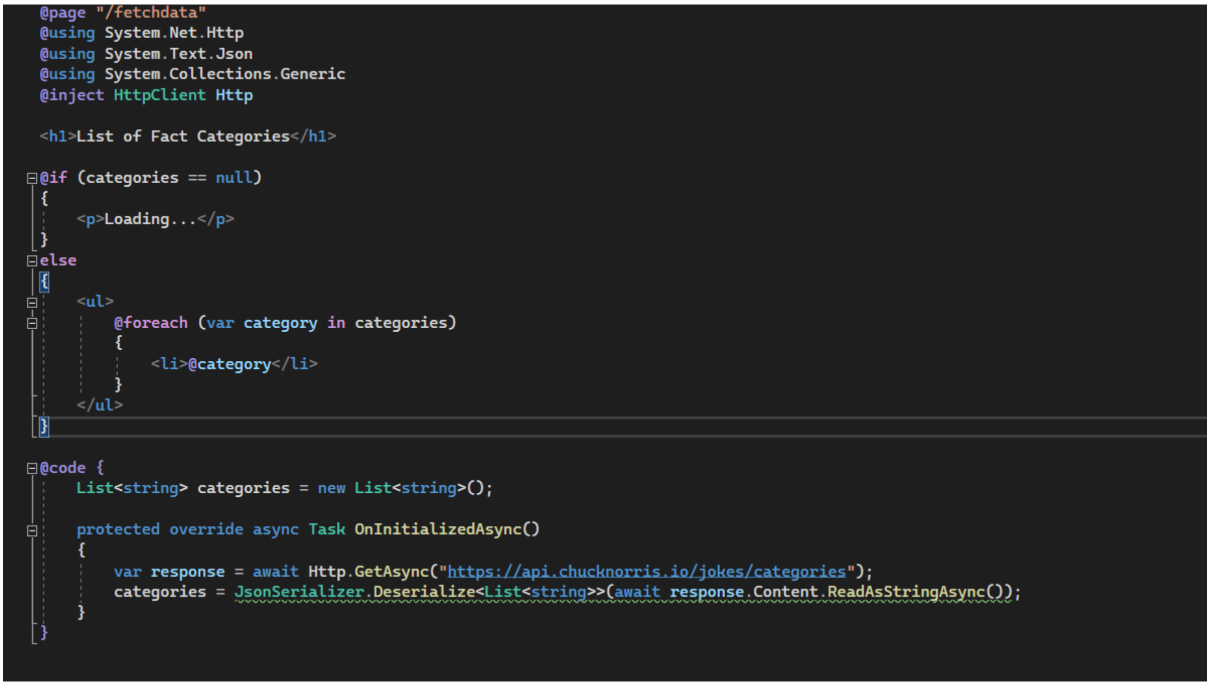
The handle input sets the query to string, then puts the string into the query. The input is taken from the Search.



List of Fact Categories

- animal
- career
- celebrity
- dev
- explicit
- fashion
- food
- history
- money
- movie
- music
- political
- religion
- science
- sport
- travel

Figure 2 A list of all the categories that the website has,



```
@page "/fetchdata"
@using System.Net.Http
@using System.Text.Json
@using System.Collections.Generic
@inject HttpClient Http

<h1>List of Fact Categories</h1>

@if (categories == null)
{
    <p>Loading...</p>
}
else
{
    <ul>
        @foreach (var category in categories)
        {
            <li>@category</li>
        }
    </ul>
}

@code {
    List<string> categories = new List<string>();

    protected override async Task OnInitializedAsync()
    {
        var response = await Http.GetAsync("https://api.chucknorris.io/jokes/categories");
        categories = JsonSerializer.Deserialize<List<string>>(await response.Content.ReadAsStringAsync());
    }
}
```

This pulls from the categories api then displays them.

URI for Deployed App

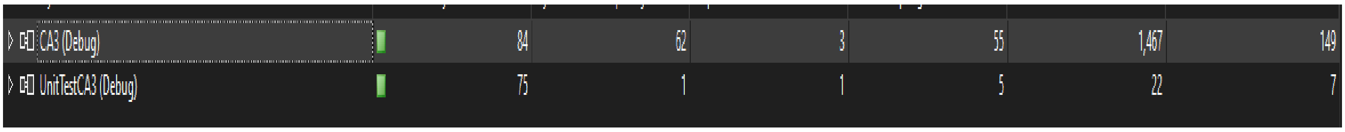
https://blazorchucknorris.azurewebsites.net/

Description of Testing

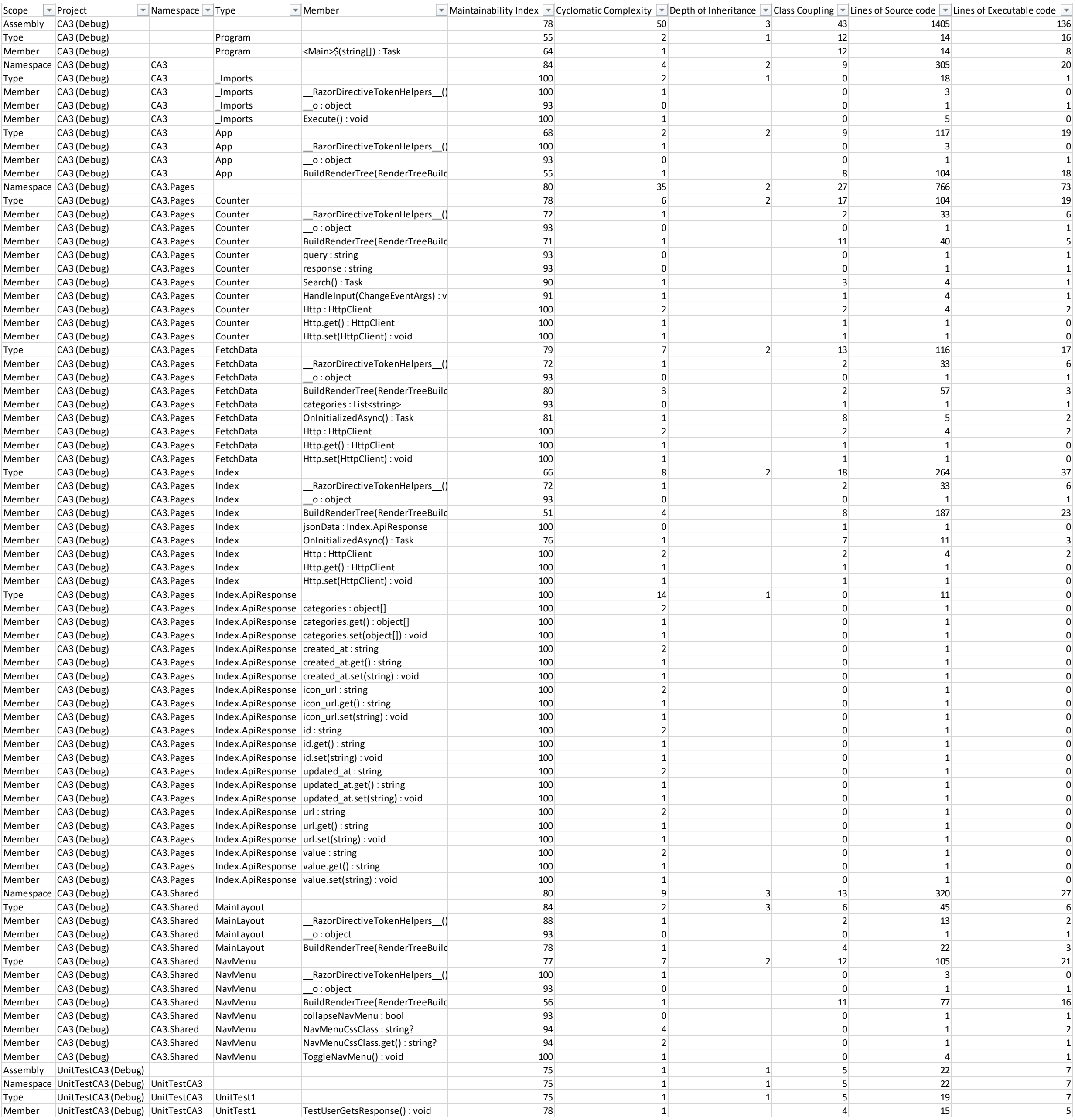
I set up unit testing however I failed to figure out how to appropriately do so, the theoretical tests I would attempt would be testing that the search function outputs search results, testing that there was a response from the API and testing that the categories correctly showed up as they should.

Code Quality & Metrics

The visual studio code metric tells me these are of decent quality.



CA3 (Debug)	🟩	84	62	3	55	1,467	149
UnitTestCA3 (Debug)	🟩	75	1	1	5	22	7



Scope	Project	Namespace	Type	Member	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Source code	Lines of Executable code
Assembly	CA3 (Debug)				78	50	3	43	1405	136
Type	CA3 (Debug)		Program		55	2	1	12	14	16
Member	CA3 (Debug)		Program	<Main>$(string[]) : Task	64	1		12	14	8
Namespace	CA3 (Debug)	CA3			84	4	2	9	305	20
Type	CA3 (Debug)	CA3	_Imports		100	2	1	0	18	1
Member	CA3 (Debug)	CA3	_Imports	__RazorDirectiveTokenHelpers__()	100	1		0	3	0
Member	CA3 (Debug)	CA3	_Imports	__o : object	93	0		0	1	1
Member	CA3 (Debug)	CA3	_Imports	Execute() : void	100	1		0	5	0
Type	CA3 (Debug)	CA3	App		68	2	2	9	117	19
Member	CA3 (Debug)	CA3	App	__RazorDirectiveTokenHelpers__()	100	1		0	3	0
Member	CA3 (Debug)	CA3	App	__o : object	93	0		0	1	1
Member	CA3 (Debug)	CA3	App	BuildRenderTree(RenderTreeBuild	55	1		8	104	18
Namespace	CA3 (Debug)	CA3.Pages			80	35	2	27	766	73
Type	CA3 (Debug)	CA3.Pages	Counter		78	6	2	17	104	19
Member	CA3 (Debug)	CA3.Pages	Counter	__RazorDirectiveTokenHelpers__()	72	1		2	33	6
Member	CA3 (Debug)	CA3.Pages	Counter	__o : object	93	0		0	1	1
Member	CA3 (Debug)	CA3.Pages	Counter	BuildRenderTree(RenderTreeBuild	71	1		11	40	5
Member	CA3 (Debug)	CA3.Pages	Counter	query : string	93	0		0	1	1
Member	CA3 (Debug)	CA3.Pages	Counter	response : string	93	0		0	1	1
Member	CA3 (Debug)	CA3.Pages	Counter	Search() : Task	90	1		3	4	1
Member	CA3 (Debug)	CA3.Pages	Counter	HandleInput(ChangeEventArgs) : v	91	1		1	4	1
Member	CA3 (Debug)	CA3.Pages	Counter	Http : HttpClient	100	2		2	4	2
Member	CA3 (Debug)	CA3.Pages	Counter	Http.get() : HttpClient	100	1		1	1	0
Member	CA3 (Debug)	CA3.Pages	Counter	Http.set(HttpClient) : void	100	1		1	1	0
Type	CA3 (Debug)	CA3.Pages	FetchData		79	7	2	13	116	17
Member	CA3 (Debug)	CA3.Pages	FetchData	__RazorDirectiveTokenHelpers__()	72	1		2	33	6
Member	CA3 (Debug)	CA3.Pages	FetchData	__o : object	93	0		0	1	1
Member	CA3 (Debug)	CA3.Pages	FetchData	BuildRenderTree(RenderTreeBuild	80	3		2	57	3
Member	CA3 (Debug)	CA3.Pages	FetchData	categories : List<string>	93	0		1	1	1
Member	CA3 (Debug)	CA3.Pages	FetchData	OnInitializedAsync() : Task	81	1		8	5	2
Member	CA3 (Debug)	CA3.Pages	FetchData	Http : HttpClient	100	2		2	4	2
Member	CA3 (Debug)	CA3.Pages	FetchData	Http.get() : HttpClient	100	1		1	1	0
Member	CA3 (Debug)	CA3.Pages	FetchData	Http.set(HttpClient) : void	100	1		1	1	0
Type	CA3 (Debug)	CA3.Pages	Index		66	8	2	18	264	37
Member	CA3 (Debug)	CA3.Pages	Index	__RazorDirectiveTokenHelpers__()	72	1		2	33	6
Member	CA3 (Debug)	CA3.Pages	Index	__o : object	93	0		0	1	1
Member	CA3 (Debug)	CA3.Pages	Index	BuildRenderTree(RenderTreeBuild	51	4		8	187	23
Member	CA3 (Debug)	CA3.Pages	Index	jsonData : Index.ApiResponse	100	0		1	1	0
Member	CA3 (Debug)	CA3.Pages	Index	OnInitializedAsync() : Task	76	1		7	11	3
Member	CA3 (Debug)	CA3.Pages	Index	Http : HttpClient	100	2		2	4	2
Member	CA3 (Debug)	CA3.Pages	Index	Http.get() : HttpClient	100	1		1	1	0
Member	CA3 (Debug)	CA3.Pages	Index	Http.set(HttpClient) : void	100	1		1	1	0
Type	CA3 (Debug)	CA3.Pages	Index.ApiResponse		100	14	1	0	11	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	categories : object[]	100	2		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	categories.get() : object[]	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	categories.set(object[]) : void	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	created_at : string	100	2		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	created_at.get() : string	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	created_at.set(string) : void	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	icon_url : string	100	2		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	icon_url.get() : string	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	icon_url.set(string) : void	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	id : string	100	2		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	id.get() : string	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	id.set(string) : void	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	updated_at : string	100	2		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	updated_at.get() : string	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	updated_at.set(string) : void	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	url : string	100	2		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	url.get() : string	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	url.set(string) : void	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	value : string	100	2		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	value.get() : string	100	1		0	1	0
Member	CA3 (Debug)	CA3.Pages	Index.ApiResponse	value.set(string) : void	100	1		0	1	0
Namespace	CA3 (Debug)	CA3.Shared			80	9	3	13	320	27
Type	CA3 (Debug)	CA3.Shared	MainLayout		84	2	3	6	45	6
Member	CA3 (Debug)	CA3.Shared	MainLayout	__RazorDirectiveTokenHelpers__()	88	1		2	13	2
Member	CA3 (Debug)	CA3.Shared	MainLayout	__o : object	93	0		0	1	1
Member	CA3 (Debug)	CA3.Shared	MainLayout	BuildRenderTree(RenderTreeBuild	78	1		4	22	3
Type	CA3 (Debug)	CA3.Shared	NavMenu		77	7	2	12	105	21
Member	CA3 (Debug)	CA3.Shared	NavMenu	__RazorDirectiveTokenHelpers__()	100	1		0	3	0
Member	CA3 (Debug)	CA3.Shared	NavMenu	__o : object	93	0		0	1	1
Member	CA3 (Debug)	CA3.Shared	NavMenu	BuildRenderTree(RenderTreeBuild	56	1		11	77	16
Member	CA3 (Debug)	CA3.Shared	NavMenu	collapseNavMenu : bool	93	0		0	1	1
Member	CA3 (Debug)	CA3.Shared	NavMenu	NavMenuCssClass : string?	94	4		0	1	2
Member	CA3 (Debug)	CA3.Shared	NavMenu	NavMenuCssClass.get() : string?	94	2		0	1	1
Member	CA3 (Debug)	CA3.Shared	NavMenu	ToggleNavMenu() : void	100	1		0	4	1
Assembly	UnitTestCA3 (Debug)				75	1	1	5	22	7
Namespace	UnitTestCA3 (Debug)	UnitTestCA3			75	1	1	5	22	7
Type	UnitTestCA3 (Debug)	UnitTestCA3	UnitTest1		75	1	1	5	19	7
Member	UnitTestCA3 (Debug)	UnitTestCA3	UnitTest1	TestUserGetsResponse() : void	78	1		4	15	5

Common Issues Encountered

The webapi out putted it’s results as json files however these were not in arrays, I didn’t realise this at the start and spent a significant amount of time struggling with formatting the json’s output correctly and still do.

I switched API’s twice during this project, the first one was switched out as RapidAPI limited the number of requests you could make from an API which proved problematic and the second one was switched out as I couldn’t consistently pull from the API.

My original code was I would said a lot more advanced and you can see it under the folder ‘Cocktails’ until ‘spring cleaning’ git push to main, I would have preferred to do my code this way but because of the many issues and the incredible amount of stress I basically stripped down my code to bare bones.

API URL

https://api.chucknorris.io/

GitHub Repo URL

https://github.com/CiananNicolai/ASP.Net-Core-Blazor-Project-API