**Practical No: 9**

**Date: 19/04/2023**

**Aim: Building an ASP.NET Core Web Application & Invoking REST APIs from JavaScript**

**Description:**

**Explain Invoking REST APIs from JavaScript**

**Code & Output:**

To make a request to a REST API using JavaScript, you can use the fetch() method or a third-party library like Axios or jQuery. The basic steps involved are as follows:

1. Determine the endpoint URL for the API you want to call.
2. Choose the appropriate HTTP method, such as GET or POST.
3. Include any required request headers, such as Authorization or Content-Type.
4. If needed, include a request body with any data that should be sent to the server.
5. Use the fetch() method or a library function to send the HTTP request to the API endpoint.
6. Handle the response received from the API, which may contain either data or an error message.
7. It is crucial to handle errors and response data properly when calling a REST API from JavaScript. This may involve parsing the data into a specific format, such as JSON, or displaying it in the application's user interface.

**Code & Output:**

**Controller Folder**

ApiController.cs

using Microsoft.AspNetCore.Mvc;

using StatlerWaldorfCorp.WebApp.Models;

namespace StatlerWaldorfCorp.WebApp.Controllers

{

[Route("api/test")]

public class ApiController : Controller

{

[HttpGet]

public IActionResult GetTest()

{

return this.Ok(new StockQuote { Symbol = "API", Price = 9999 });

}

}

}

HomeController.cs

using Microsoft.AspNetCore.Mvc;

using System.Threading.Tasks;

using StatlerWaldorfCorp.WebApp.Models;

namespace StatlerWaldorfCorp.WebApp.Controllers

{

public class HomeController : Controller

{

public IActionResult Index()

{

var model = new StockQuote { Symbol = "HLLO", Price = 3200 };

return View(model);

}

}

}

**Models Folder**

StockQuote.cs

namespace StatlerWaldorfCorp.WebApp.Models

{

public class StockQuote

{

public string Symbol { get; set; }

public int Price { get; set; }

}

}

Views Folder

Index.cshtml

<html>

<head>

<title>Hello world</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

<script src="/Scripts/hello.js"></script>

</head>

<body>

<h1>Hello World</h1>

<div>

<h2>Stock Quote</h2>

<div>

Symbol: @Model.Symbol<br/>

Price: [$@Model.Price<br/](mailto:$@Model.Price%3cbr/)>

</div>

</div>

<br/>

<div>

<p class="quote-symbol">The Symbol is </p>

<p class="quote-price">The price is $</p>

</div>

</body>

</html>

**Folder wwwroot\Scripts**

hello.js

$(document).ready(function () {

$.ajax({

url: "/api/test"

}).then(function (data) {

$('.quote-symbol').append(data.symbol);

$('.quote-price').append(data.price);

});

});

**Program.cs**

using System;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Builder;

using Microsoft.Extensions.Configuration;

using System.IO;

namespace StatlerWaldorfCorp.WebApp

{

public class Program

{

public static void Main(string[] args)

{

var config = new ConfigurationBuilder()

.AddCommandLine(args)

.Build();

var host = new WebHostBuilder()

.UseContentRoot(Directory.GetCurrentDirectory())

.UseKestrel()

.UseStartup<Startup>()

.UseConfiguration(config)

.Build();

host.Run();

}

}

}

**Startup.cs**

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.Extensions.Logging;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Configuration;

namespace StatlerWaldorfCorp.WebApp

{

public class Startup

{

public Startup(IHostingEnvironment env)

{

var builder = new ConfigurationBuilder()

.SetBasePath(env.ContentRootPath)

.AddEnvironmentVariables();

Configuration = builder.Build();

}

public IConfiguration Configuration { get; set; }

public void ConfigureServices(IServiceCollection services) {

services.AddMvc();

}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env, ILoggerFactory loggerFactory)

{

//loggerFactory.AddConsole();

//loggerFactory.AddDebug();

app.UseDeveloperExceptionPage();

app.UseRouting();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

});

app.UseStaticFiles();

}

}

}

**StatlerWaldorfCorp.WebApp.csproj**

<Project Sdk="Microsoft.NET.Sdk.Web">

<PropertyGroup>

<TargetFramework>**netcoreapp5.0**</TargetFramework>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.AspNetCore" Version="1.1.1" />

<PackageReference Include="Microsoft.AspNetCore.Mvc" Version="1.1.2" />

<PackageReference Include="Microsoft.AspNetCore.StaticFiles" Version="1.1.1" />

<PackageReference Include="Microsoft.Extensions.Logging.Debug" Version="1.1.1" />

<PackageReference Include="Microsoft.VisualStudio.Web.BrowserLink" Version="1.1.0" />

<PackageReference Include="Microsoft.Extensions.Configuration" Version="1.1.1"/>

<PackageReference Include="Microsoft.Extensions.Options.ConfigurationExtensions" Version="1.1.1"/>

<PackageReference Include="Microsoft.Extensions.Configuration.Json" Version="1.1.1"/>

<PackageReference Include="Microsoft.Extensions.Configuration.CommandLine" Version="1.1.1"/>

</ItemGroup>

</Project>

**Text, letter

Description automatically generated**

**Graphical user interface, text

Description automatically generated with medium confidence**