using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Text;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.Logging;

using Newtonsoft.Json;

using WebClient.Models;

namespace WebClient.Controllers

{

public class EmployeeController : Controller

{

private readonly ILogger<EmployeeController> \_logger;

public EmployeeController(ILogger<EmployeeController> logger)

{

\_logger = logger;

}

public async Task<IActionResult> Index()

{

if (TempData["msg"] != null)

ViewBag.msg = TempData["msg"];

List<Employee> students = new List<Employee>();

using (var client = new HttpClient())

{

//Send HTTP requests from here.

string endpoint = "https://localhost:44371/api/";

client.BaseAddress = new Uri(endpoint);

//GET Method

HttpResponseMessage response = await client.GetAsync("EMployees");

if (response.IsSuccessStatusCode)

{

var jsonString = response.Content.ReadAsStringAsync().Result;

students = JsonConvert.DeserializeObject<List<Employee>>(jsonString);

}

return View(students);

}

}

public async Task<IActionResult> Details(int Id)

{

Employee student = new Employee();

using (var client = new HttpClient())

{

//Send HTTP requests from here.

string endpoint = "https://localhost:44371/api/";

client.BaseAddress = new Uri(endpoint);

//GET Method

HttpResponseMessage response = await client.GetAsync($"EMployees/{Id}");

if (response.IsSuccessStatusCode)

{

var jsonString = response.Content.ReadAsStringAsync().Result;

student = JsonConvert.DeserializeObject<Employee>(jsonString);

return View(student);

}

else

{

ViewBag.msg = response.StatusCode;

return View();

}

}

}

public IActionResult Create()

{

Employee employee = new Employee();

return View(employee);

}

[HttpPost]

public IActionResult Create(Employee employee)

{

HttpClient client = new HttpClient();

{

StringContent content = new StringContent(JsonConvert.SerializeObject(employee), Encoding.UTF8, "application/json");

string endpoint = "https://localhost:44371/api/employees";

client.BaseAddress = new Uri(endpoint);

var Response = client.PostAsync(endpoint, content);

Response.Wait();

var result = Response.Result;

if (result.IsSuccessStatusCode)

{

TempData["msg"] = "Record Inserted";

return RedirectToAction("Index");

}

else if (Response.Result.StatusCode == System.Net.HttpStatusCode.Conflict)

{

ModelState.Clear();

ModelState.AddModelError("Id", "Id Already Exist");

return View();

}

else

{

TempData["msg"] = Response.Result.StatusCode;

return RedirectToAction("Index");

}

}

}

public async Task<IActionResult> Delete(int id)

{

Employee student = new Employee();

using (var client = new HttpClient())

{

//Send HTTP requests from here.

string endpoint = "https://localhost:44371/api/";

client.BaseAddress = new Uri(endpoint);

//GET Method

HttpResponseMessage response = await client.GetAsync($"EMployees/{id}");

if (response.IsSuccessStatusCode)

{

var jsonString = response.Content.ReadAsStringAsync().Result;

student = JsonConvert.DeserializeObject<Employee>(jsonString);

return View(student);

}

else

{

ViewBag.msg = response.StatusCode;

return View();

}

}

}

[HttpPost]

public async Task<ActionResult> Delete(int id, Employee employee)

{

using (var client = new HttpClient())

{

//Send HTTP requests from here.

string endpoint = "https://localhost:44371/api/";

client.BaseAddress = new Uri(endpoint);

HttpResponseMessage response = await client.DeleteAsync($"employees/{id}");

TempData["msg"] = "Record Deleted";

return RedirectToAction("Index");

}

}

public async Task<IActionResult> Edit(int id)

{

Employee student = new Employee();

using (var client = new HttpClient())

{

//Send HTTP requests from here.

string endpoint = "https://localhost:44371/api/";

client.BaseAddress = new Uri(endpoint);

//GET Method

HttpResponseMessage response = await client.GetAsync($"EMployees/{id}");

if (response.IsSuccessStatusCode)

{

var jsonString = response.Content.ReadAsStringAsync().Result;

student = JsonConvert.DeserializeObject<Employee>(jsonString);

return View(student);

}

else

{

ViewBag.msg = response.StatusCode;

return View();

}

}

}

[HttpPost]

public async Task<ActionResult> Edit(int id, Employee employee)

{

StringContent content = new StringContent(JsonConvert.SerializeObject(employee), Encoding.UTF8, "application/json");

using (var client = new HttpClient())

{

//Send HTTP requests from here.

string endpoint = "https://localhost:44371/api/";

client.BaseAddress = new Uri(endpoint);

HttpResponseMessage response = await client.PutAsync($"employees/{id}", content);

TempData["msg"] = "<script> alert('Record Edited')</script>";

return RedirectToAction("Index");

}

}

}

}

----------------------------------------------------

2nd way

**appsettings.json file**

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*",

**"WebAPIBaseUrl": "https://localhost:** **44371/api/"**

}

-----------------------------------------------

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Text;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.Logging;

using Newtonsoft.Json;

using WebClient.Models;

namespace WebClient.Controllers

{

public class EmployeeController : Controller

{

private readonly ILogger<EmployeeController> \_logger;

**private IConfiguration \_Configure;**

**string apiBaseUrl;**

public EmployeeController(ILogger<EmployeeController> logger, **IConfiguration configuration**)

{

\_logger = logger;

**\_Configure = configuration;**

**apiBaseUrl = \_Configure.GetValue<string>("WebAPIBaseUrl");**

}

public async Task<IActionResult> Index()

{

if (TempData["msg"] != null)

ViewBag.msg = TempData["msg"];

List<Employee> students = new List<Employee>();

using (var client = new HttpClient())

{

//Send HTTP requests from here.

// string endpoint = "https://localhost:44371/api/";

client.BaseAddress = new Uri(**apiBaseUrl**);

//GET Method

HttpResponseMessage response = await client.GetAsync("EMployees");

if (response.IsSuccessStatusCode)

{

var jsonString = response.Content.ReadAsStringAsync().Result;

students = JsonConvert.DeserializeObject<List<Employee>>(jsonString);

}

return View(students);

}

}

-----------------------------------------

@{if (@ViewBag.msg != null)

{

@Html.Raw(ViewBag.msg);

}

}

<https://www.youtube.com/watch?v=bgsROO8kDh0>