**Web API using .NET Core with Swagger and Postman**

**Objective:** Demonstrate Swagger installation to WebAPI and WebAPI listing on browser.

- Nuget package to download Swashbuckle.AspNetCore, Usage of ProducesResponseType to Web API method, AddSwaggerGen, UseSwaggerUI.

- Demonstrate the usage of Postman tool to hit WebAPI methods.

- Structure in Postman tool, Headers with Authorization, Body as JSON, Option to choose the type of request, Request collection and how to add a new request in the collection, Tabs in the center pane that corresponds to the request.

- Demonstrate the usage of Route and Explain Name attribute in Http requests.

- Importance of user friendly name to action method, Explain the usage of ActionName to have more than 1 method with the same Action verb.

- Demonstrate creation of a simple WebAPI - With Read, Write actions.

**Controllers:**

**EmployeeController.cs:**

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

using System.Linq;

namespace MySwaggerDemo.Controllers

{

[ApiController]

[Route("api/Emp")] // Custom route name

public class EmployeeController : ControllerBase

{

// In-memory employee list

private static readonly List<Employee> employees = new()

{

new Employee { Id = 1, Name = "Alice", Email = "alice@example.com" },

new Employee { Id = 2, Name = "Bob", Email = "bob@example.com" }

};

// GET: api/Emp

[HttpGet]

public IActionResult Get()

{

return Ok(employees);

}

// GET: api/Emp/1

[HttpGet("{id}")]

public IActionResult Get(int id)

{

var emp = employees.FirstOrDefault(e => e.Id == id);

if (emp == null) return NotFound();

return Ok(emp);

}

// POST: api/Emp

[HttpPost]

public IActionResult Post([FromBody] Employee newEmp)

{

if (employees.Any(e => e.Id == newEmp.Id))

return Conflict("Employee with this ID already exists.");

employees.Add(newEmp);

return CreatedAtAction(nameof(Get), new { id = newEmp.Id }, newEmp);

}

// PUT: api/Emp/1

[HttpPut("{id}")]

public IActionResult Put(int id, [FromBody] Employee updatedEmp)

{

var emp = employees.FirstOrDefault(e => e.Id == id);

if (emp == null) return NotFound();

emp.Name = updatedEmp.Name;

emp.Email = updatedEmp.Email;

return NoContent();

}

// DELETE: api/Emp/1

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

var emp = employees.FirstOrDefault(e => e.Id == id);

if (emp == null) return NotFound();

employees.Remove(emp);

return NoContent();

}

}

// Model class

public class Employee

{

public int Id { get; set; } // Unique ID

public string Name { get; set; } = ""; // Employee name

public string Email { get; set; } = ""; // Employee email

}

}

**ValuesController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace MySwaggerDemo.Controllers

{

[ApiController]

[Route("api/[controller]")]

public class ValuesController : ControllerBase

{

private static readonly List<string> values = new() { "value1", "value2" };

// GET: api/values

[HttpGet]

public IActionResult Get()

{

return Ok(values);

}

// GET: api/values/1

[HttpGet("{id}")]

public IActionResult Get(int id)

{

if (id < 0 || id >= values.Count)

return NotFound();

return Ok(values[id]);

}

// POST: api/values

[HttpPost]

public IActionResult Post([FromBody] string value)

{

values.Add(value);

return CreatedAtAction(nameof(Get), new { id = values.Count - 1 }, value);

}

// PUT: api/values/1

[HttpPut("{id}")]

public IActionResult Put(int id, [FromBody] string value)

{

if (id < 0 || id >= values.Count)

return NotFound();

values[id] = value;

return NoContent();

}

// DELETE: api/values/1

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

if (id < 0 || id >= values.Count)

return NotFound();

values.RemoveAt(id);

return NoContent();

}

}

}

**Program.cs:**

var builder = WebApplication.CreateBuilder(args);

// Add services to the container

builder.Services.AddControllers();

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

TermsOfService = new Uri("https://example.com"),

Contact = new OpenApiContact

{

Name = "John Doe",

Email = "john@xyzmail.com",

Url = new Uri("https://www.example.com")

},

License = new OpenApiLicense

{

Name = "License Terms",

Url = new Uri("https://www.example.com")

}

});

});

var app = builder.Build();

// Configure the HTTP request pipeline

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI(c =>

{

// Specify the Swagger JSON endpoint

c.SwaggerEndpoint("/swagger/v1/swagger.json", "Swagger Demo");

});

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

**Output:**







