**Creating a First Web Api using .NET Core**

**Objective:**

To Create a .NET Core web API using the Api template, and validate basic crud functionality using default ValuesController.

**ValuesController.cs:**

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

namespace MyFirstWebAPI.Controllers

{

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

[ApiController]

public class ValuesController : ControllerBase

{

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

[HttpGet]

public ActionResult<IEnumerable<string>> Get()

{

return values;

}

[HttpGet("{id}")]

public ActionResult<string> Get(int id)

{

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

return NotFound();

return values[id];

}

[HttpPost]

public IActionResult Post([FromBody] string value)

{

values.Add(value);

return Ok();

}

[HttpPut("{id}")]

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

{

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

return NotFound();

values[id] = value;

return Ok();

}

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

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

return NotFound();

values.RemoveAt(id);

return Ok();

}

}

}

**Program.cs:**

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

// Add services to the container.

// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

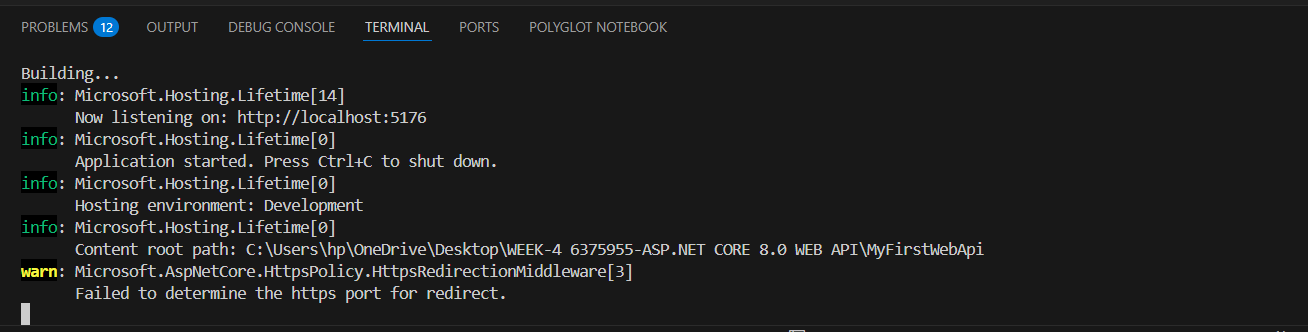
app.UseHttpsRedirection();

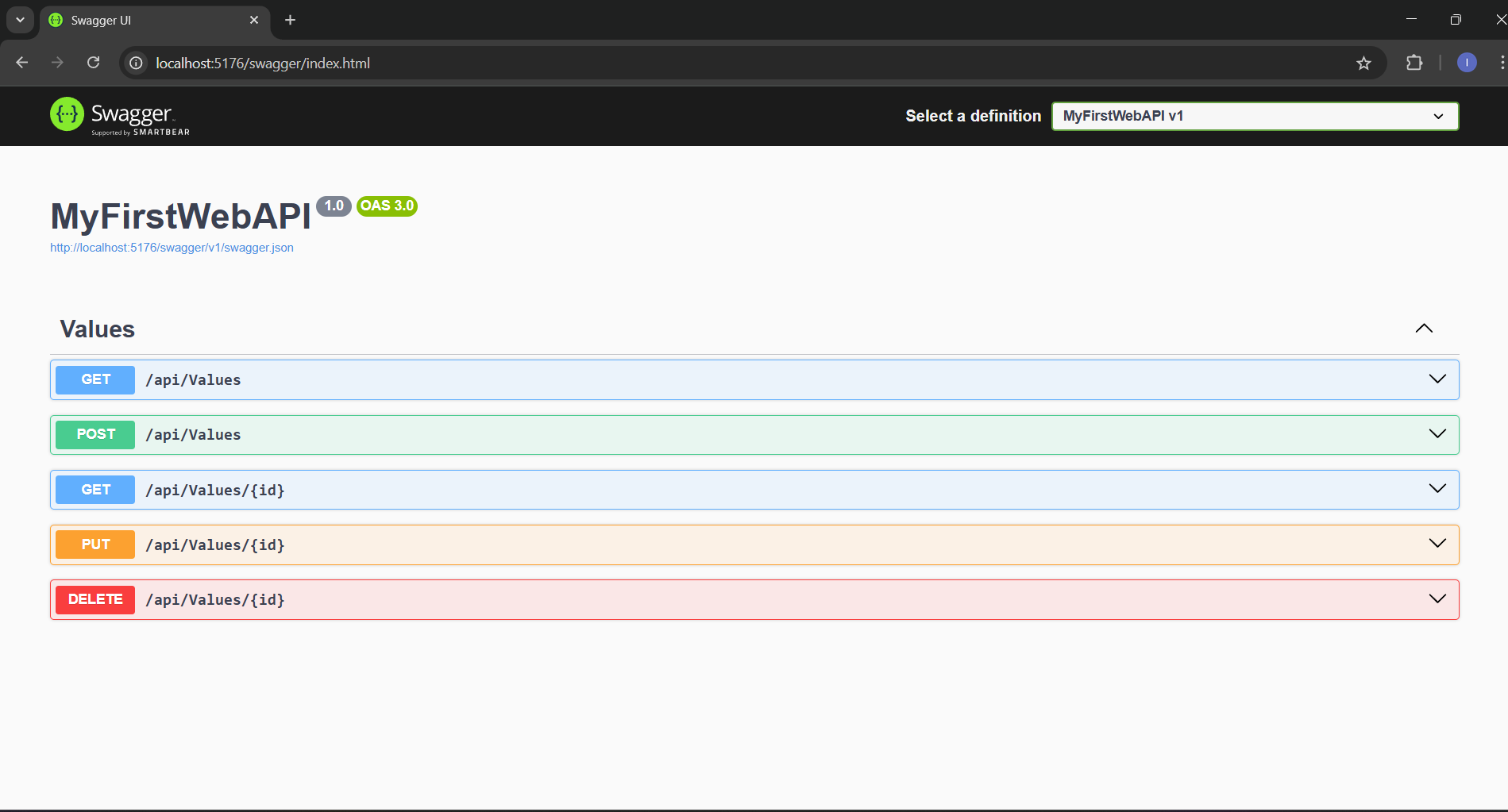
app.UseAuthorization();

app.MapControllers();

app.Run();

**Output:**

****

****

**Conclusion:**

You have successfully created and tested your first web API using .NET Core.this project demonstrates how the default controller responds to http get requests and sets the foundation for developing Restful API’S.