**Hands-On: Stage 4 - Web API – REST Concepts - Day 73 – Hands on 1**

1. **First Web Api using .Net core**

**ValuesController.cs**

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

// For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

namespace SecondWebapi.Controllers

{

    [Route("api/[controller]")]

    [ApiController]

    public class ValuesController : ControllerBase

    {

        // GET: api/<ValuesController>

        [HttpGet]

        public IEnumerable<string> Get()

        {

            return new string[] { "value1", "value2" };

        }

        // GET api/<ValuesController>/5

        [HttpGet("{id}")]

        public string Get(int id)

        {

            return "value";

        }

        // POST api/<ValuesController>

        [HttpPost]

        public void Post([FromBody] string value)

        {

        }

        // PUT api/<ValuesController>/5

        [HttpPut("{id}")]

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

        {

        }

        // DELETE api/<ValuesController>/5

        [HttpDelete("{id}")]

        public void Delete(int id)

        {

        }

    }

}

**Startup.cs**

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.HttpsPolicy;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using Microsoft.Extensions.Logging;

using Microsoft.OpenApi.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace SecondWebapi

{

    public class Startup

    {

        public Startup(IConfiguration configuration)

        {

            Configuration = configuration;

        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.

        public void ConfigureServices(IServiceCollection services)

        {

            services.AddControllers();

            services.AddSwaggerGen(c =>

            {

                c.SwaggerDoc("v1", new OpenApiInfo { Title = "SecondWebapi", Version = "v1" });

            });

        }

        // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

        {

            if (env.IsDevelopment())

            {

                app.UseDeveloperExceptionPage();

                app.UseSwagger();

                app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json", "SecondWebapi v1"));

            }

            app.UseHttpsRedirection();

            app.UseRouting();

            app.UseAuthorization();

            app.UseEndpoints(endpoints =>

            {

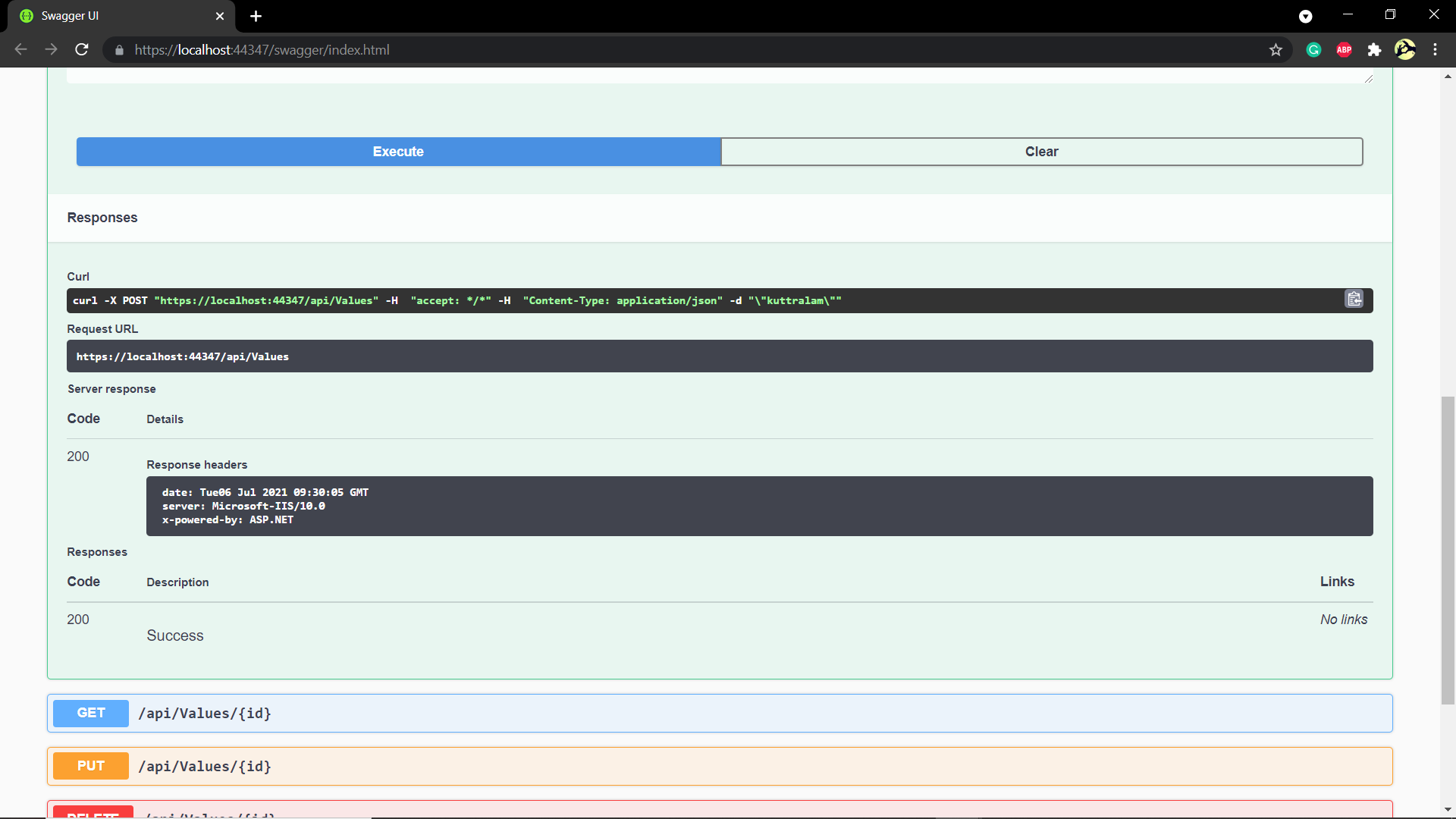
                endpoints.MapControllers();

            });

        }

    }

}

**OUTPUT:**