**916217-Gannarapupavankumar**

**Stage-3 Day 74 Web API**

**Hands on-1**

**Startup.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

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.Logging;

using Microsoft.Extensions.Options;

using Microsoft.OpenApi.Models;

namespace Handson\_2

{

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.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version\_2\_1);

services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1.0", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "1.0",

Description = "TBD",

//TermsOfService = "None",

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") }

}); ;

});

}

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

public void Configure(IApplicationBuilder app, IHostingEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseMvc();

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1.0/swagger.json", "My Demo API (V 1.0)");

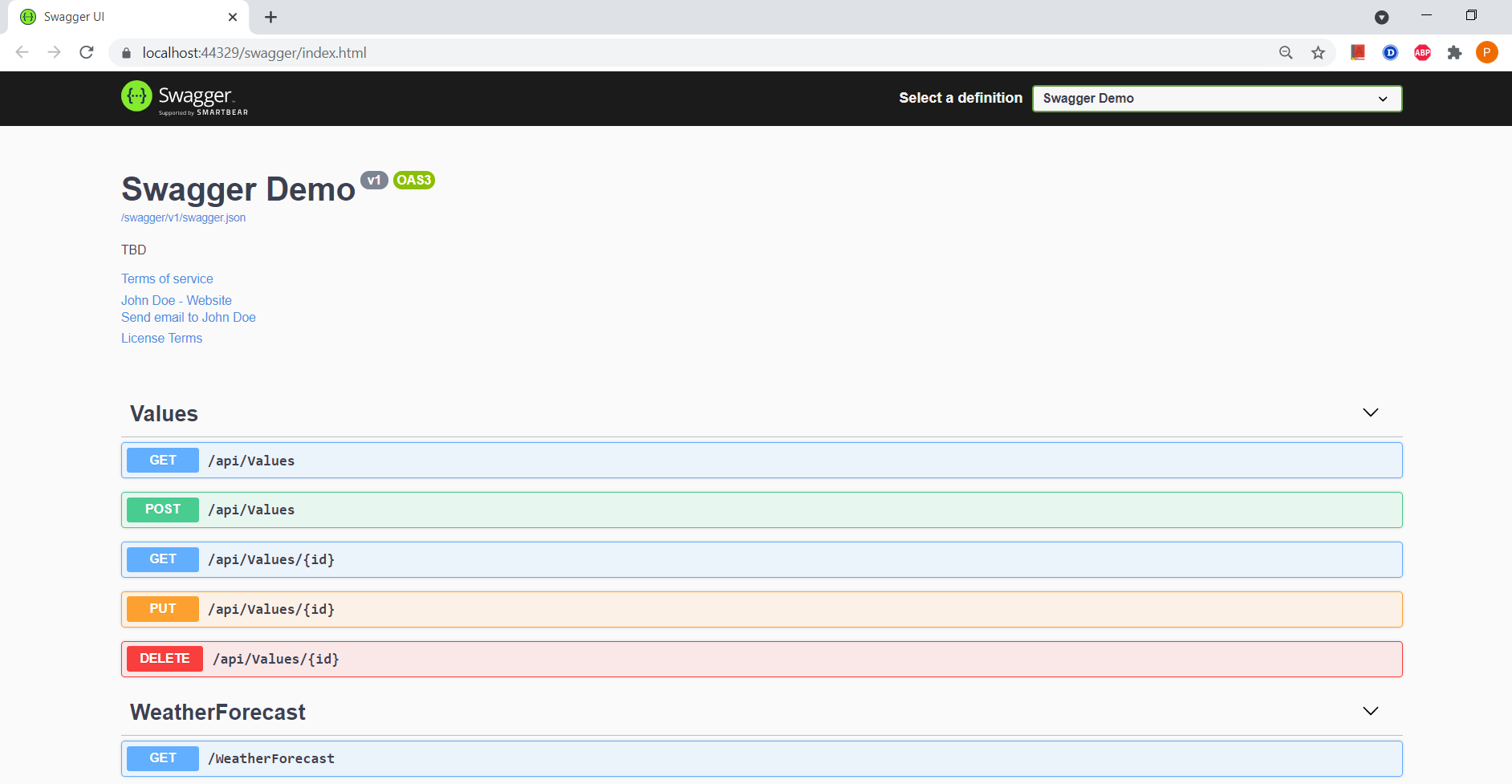
});

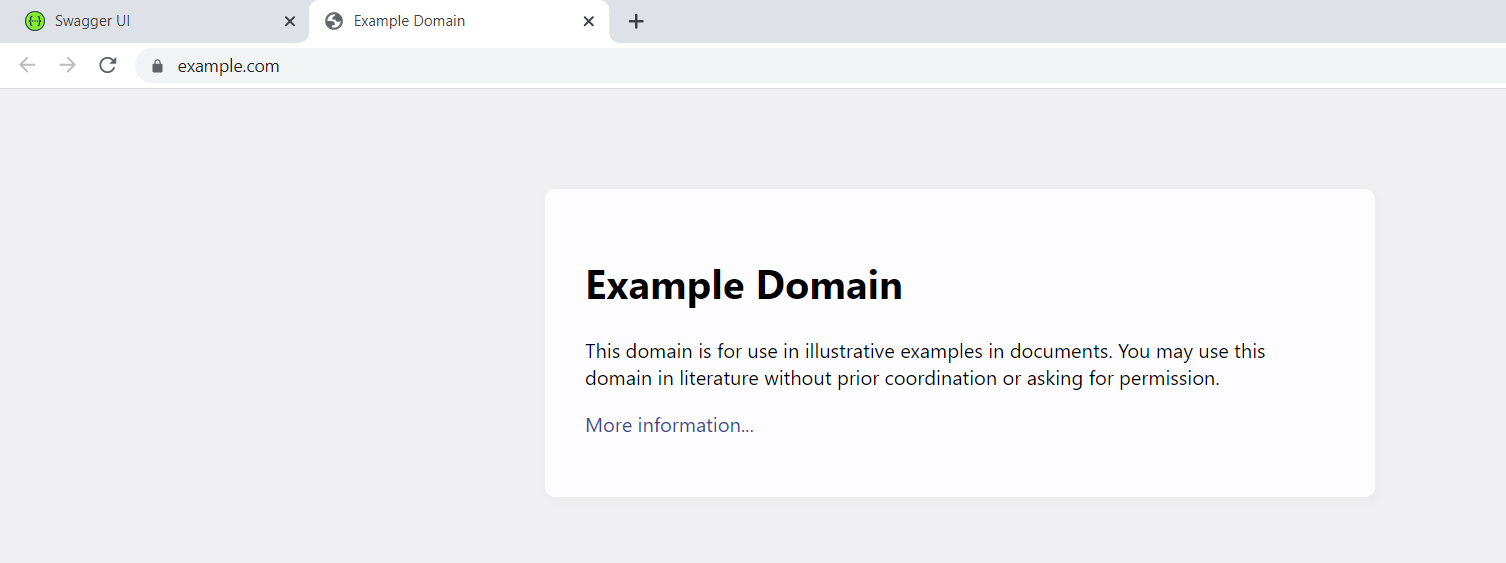
}

}

}

**Output:**





**-----------------------------------------------------------------------------------------------------------------**

**Hands 0n-2**

**EmployeeController.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using Handson\_3.Models;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Authorization;

using HandsOn3.Filters;

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

namespace Handson\_3.Controllers

{

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

[ApiController]

public class EmployeesController : ControllerBase

{

private Employee[] emp = new Employee[]

{

new Employee { Id=1 , Name="Pavan" , Salary=90000 , Permanent=true, Department="CDE",Skills="Dotnet" ,DateOfBirth=new DateTime(2000,02,04) },

new Employee { Id=2 , Name="uday" , Salary=90000 , Permanent=true, Department="AI",Skills="java" ,DateOfBirth=new DateTime(1999,01,09) } ,

new Employee { Id=3 , Name="manoj" , Salary=90000 , Permanent=false, Department="ML",Skills="python" ,DateOfBirth=new DateTime(2000,04,08) } ,

};

private IEnumerable<Employee> GetStandardEmployeeList()

{

return emp;

}

// GET: api/<ValuesController1>

//[CustomAuthFilter]

[HttpGet]

public IEnumerable<Employee> Get()

{

return GetStandardEmployeeList();

}

// GET api/<ValuesController1>/5

[HttpGet("{id}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status404NotFound)]

public IActionResult GetById(int id)

{

var prod = emp.FirstOrDefault((p) => p.Id == id);

if (prod == null)

{

return NotFound();

}

return Ok(prod);

}

// POST api/<ValuesController1>

[HttpPost]

public void Post([FromBody] string value)

{

}

// PUT api/<ValuesController1>/5

[HttpPut("{id}")]

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

{

}

// DELETE api/<ValuesController1>/5

[HttpDelete("{id}")]

public void Delete(int id)

{

}

}

}

**Employee.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Handson\_3.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public bool Permanent { get; set; }

public string Department { get; set; }

public string Skills { get; set; }

public DateTime DateOfBirth { get; set; }

}

}

**Startup.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.HttpsPolicy;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Logging;

using Microsoft.Extensions.Options;

using Microsoft.OpenApi.Models;

using Handson\_3.Models;

namespace Handson\_3

{

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.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version\_2\_1);

services.AddScoped<Employee>();

services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1.0", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "1.0",

Description = "TBD",

TermsOfService = new Uri("https://google.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") }

}); ;

});

}

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

public void Configure(IApplicationBuilder app, IHostingEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseMvc();

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1.0/swagger.json", "My Demo API (V 1.0)");

});

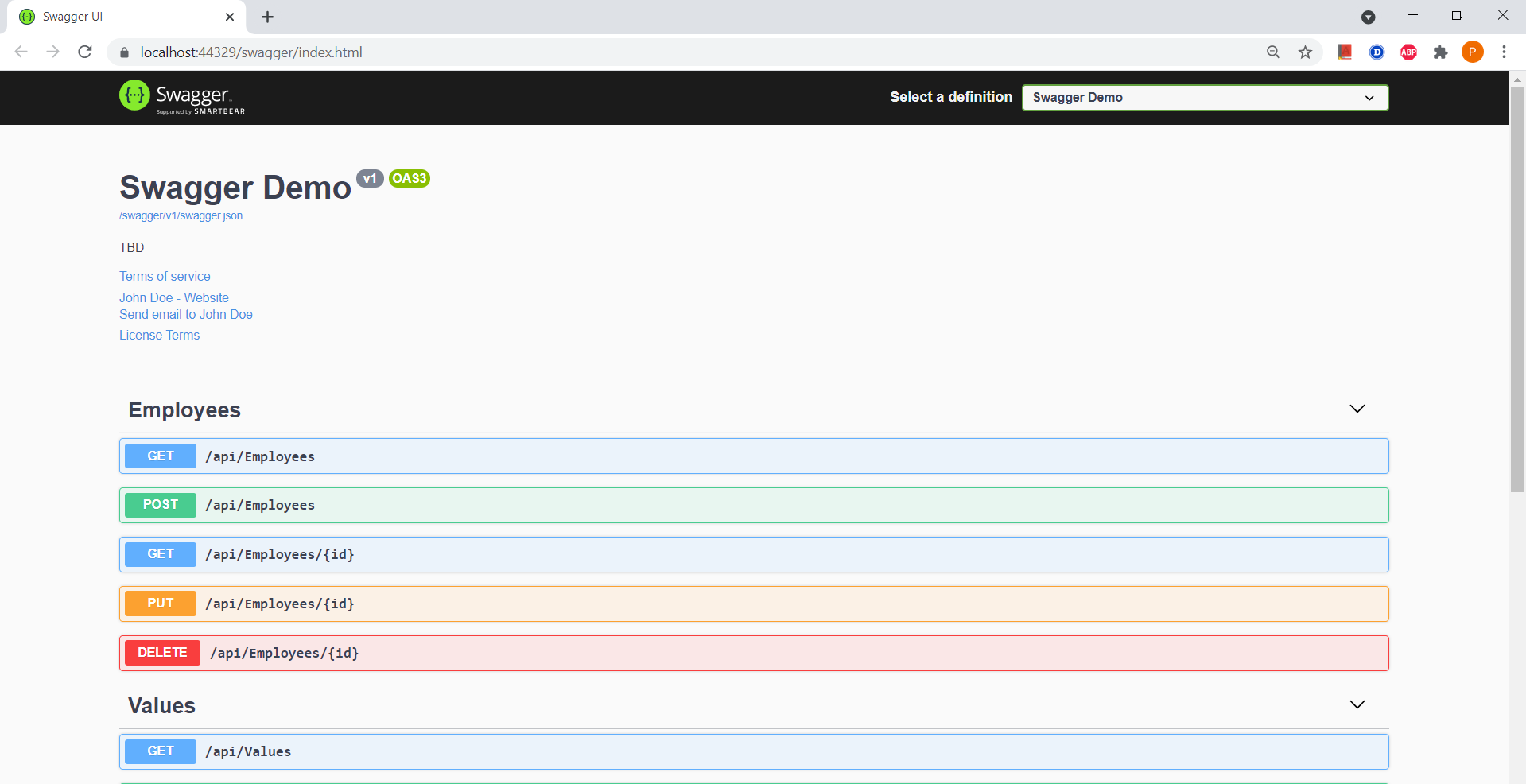
}

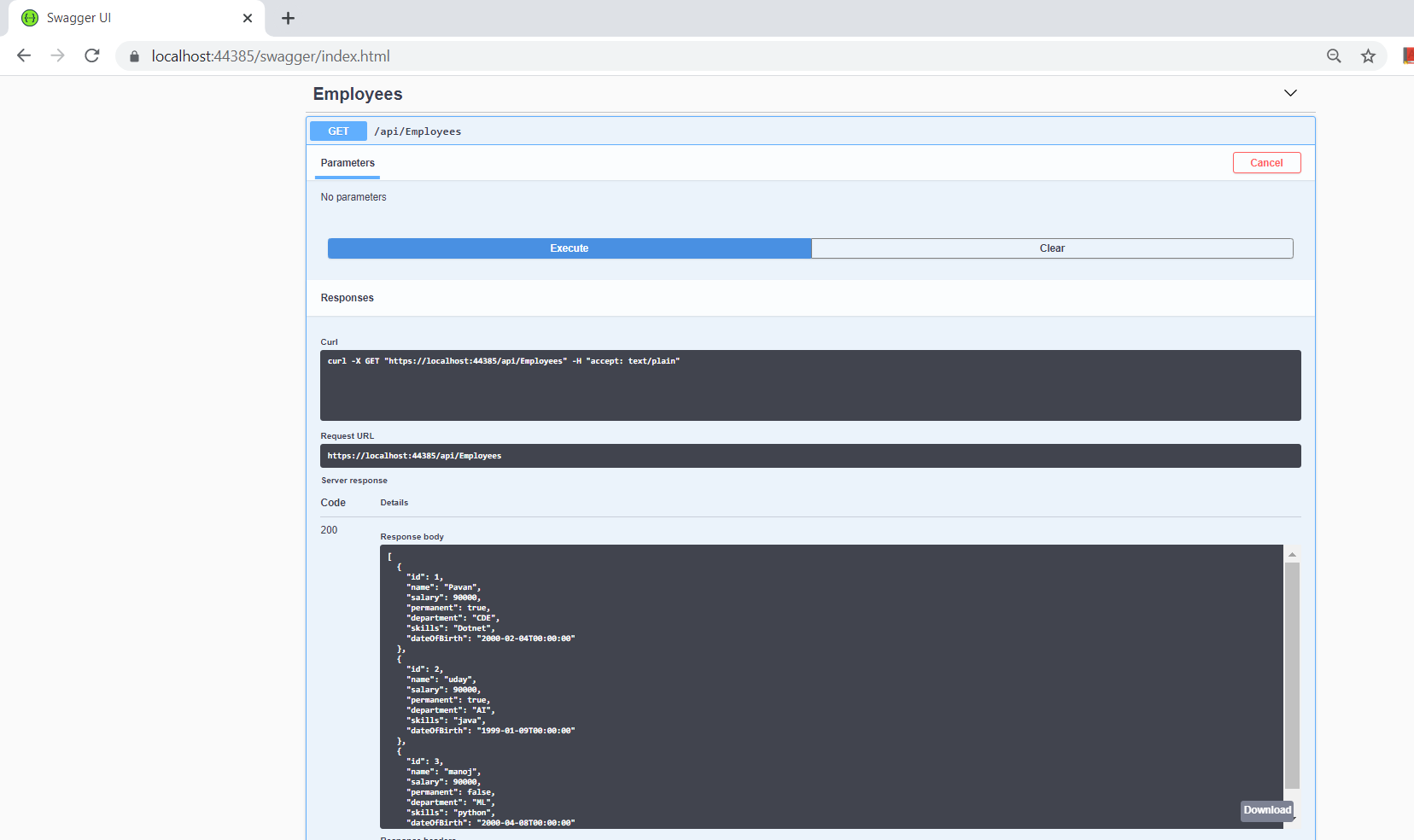
}

}

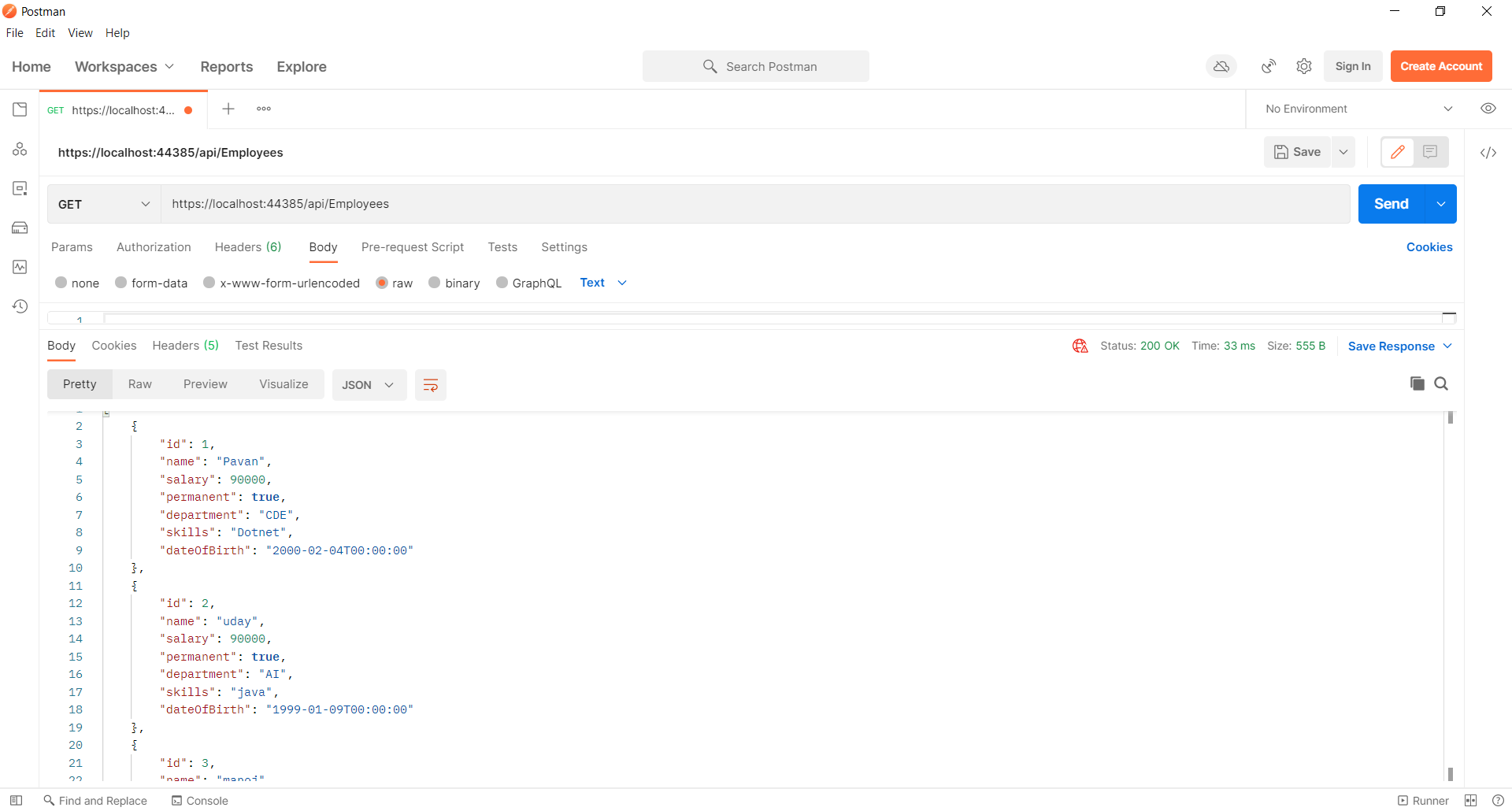
**Output:**

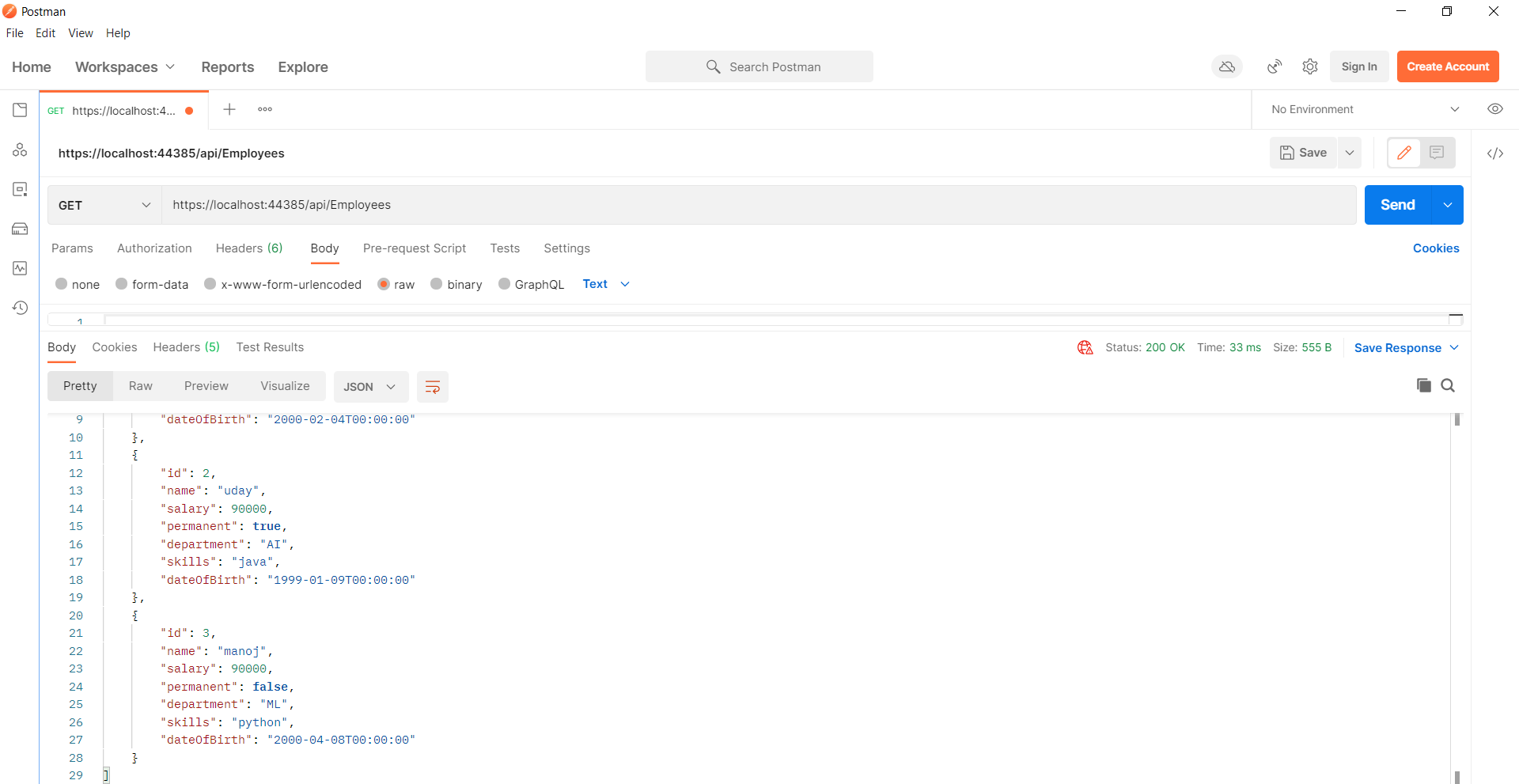
**Employee details in swagger In swagger**

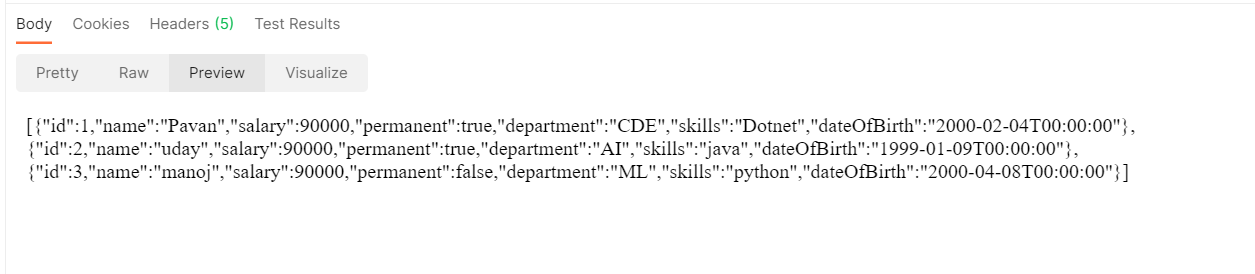




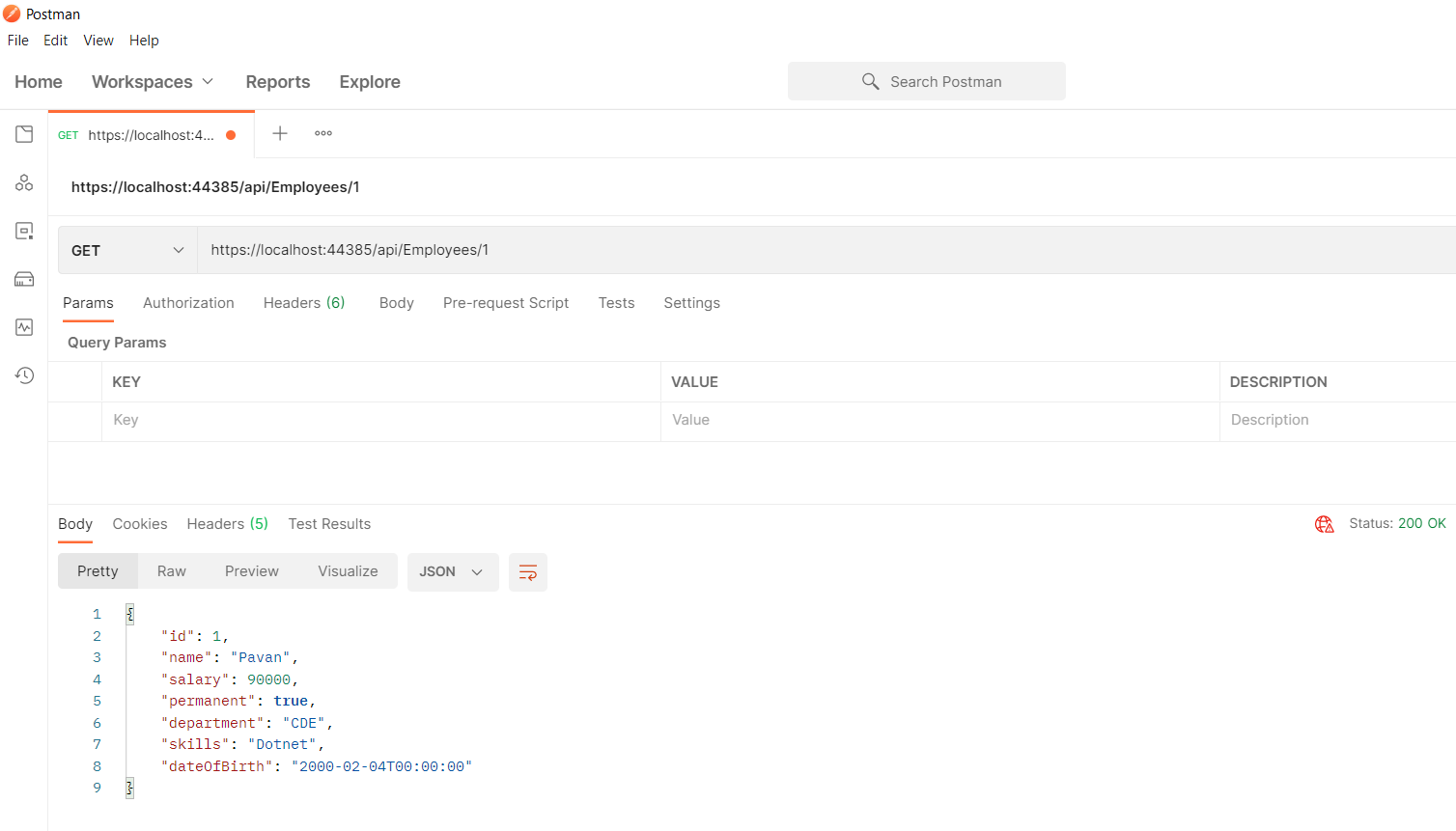
**Employee details in postman**







**Employee search by id in postman**



**Hands-on 2**

**--------------------------------------------------------------------------------------------------------**

**1. Web Api using custom model class**

Create a Custom class ‘Employee’ of the below defined structure

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public bool Permanent { get; set; }

public Department Department { get; set; }

public List<Skill> Skills { get; set; }

public DateTime DateOfBirth { get; set; }

}

Create a new controller - EmployeeController with Read Write actions

Constructor: Create few records, HTTPGet, HTTPPost/HTTPPut

Create a Private method GetStandardEmployeeList that returns a List of Employee class. Invoke this method in the Get action method of the EmployeeController that was created in the previous step.

Public ActionResult<Employee> GetStandrad()

Modify the return type of the Get action method(without parameter) to return List of Employee class object

Add ProducesResponseType to the GET action method for Status code 200

Check the Swagger description for the GET method for success status code

**EmployeesController.cs**

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Day2\_Handson.Model;

using Day2\_Handson.Filters;

using Microsoft.AspNetCore.Http;

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

namespace Day2\_Handson.Controllers

{

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

[ApiController]

public class EmployeesController : ControllerBase

{

private List<Employee> Employees;

public EmployeesController()

{

Employees = new List<Employee>()

{

new Employee() { Id= 1, Name="Pavan",Salary=90000,Permanent=true, Department= new Department( 2,"CDE") ,Skills=new List<Skill>(){ new Skill ( 1,"Python"), new Skill(3,"c#" ) }, DateOfBirth=new DateTime(2000,04,25) },

new Employee() { Id= 2, Name="Uday",Salary=20000,Permanent=true, Department= new Department( 2,"AI") ,Skills=new List<Skill>(){ new Skill ( 2,"Java"), new Skill(1,"Python" ) }, DateOfBirth=new DateTime(2000,04,02) }

};

}

private List<Employee> GetStandardEmployeeList()

{

return Employees;

}

[HttpGet]

[ProducesResponseType(StatusCodes.Status200OK)]

public IEnumerable<Employee> Get()

{

return GetStandardEmployeeList();

}

[HttpGet("Employee/getstandard")]

public ActionResult<Employee> GetStandard()

{

return Employees.Where(i => i.Permanent == true).FirstOrDefault();

}

// GET api/<EmployeeController>/5

[HttpGet("{id}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status404NotFound)]

public IActionResult GetById(int id)

{

var prod = Employees.FirstOrDefault((p) => p.Id == id);

if (prod == null)

{

return NotFound();

}

return Ok(prod);

}

// POST api/<EmployeeController>

[HttpPost]

public void Post([FromBody] string value)

{

}

// PUT api/<EmployeeController>/5

[HttpPut("{id}")]

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

{

}

// DELETE api/<EmployeeController>/5

[HttpDelete("{id}")]

public void Delete(int id)

{

}

}

}

**Employee.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Day2\_Handson

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public bool Permanent { get; set; }

public Department Department { get; set; }

public List<Skill> Skills { get; set; }

public DateTime DateOfBirth { get; set; }

}

}

**Department.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Day2\_Handson.Model

{

public class Department

{

public Department()

{ }

public Department(int DepID, string DepName)

{

this.DepID = DepID;

this.DepName = DepName;

}

public int DepID { get; set; }

public string DepName { get; set; }

}

}

**Skill.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Day2\_Handson.Model

{

public class Skill

{

public Skill() { }

public Skill(int skillID, string skillName)

{

this.SkillID = skillID;

this.SkillName = skillName;

}

public int SkillID { get; set; }

public string SkillName { get; set; }

}

}

**CustomAuthFilter.cs**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Day2\_Handson.Filters

{

public class CustomAuthFilter : ActionFilterAttribute

{

public override void OnActionExecuting(ActionExecutingContext context)

{

if (context.HttpContext.Request.Query.ContainsKey("Authorization") && context.HttpContext.Request.Query["Authorization"] == "true")

{

context.Result = new UnauthorizedResult();

}

else

{

base.OnActionExecuting(context);

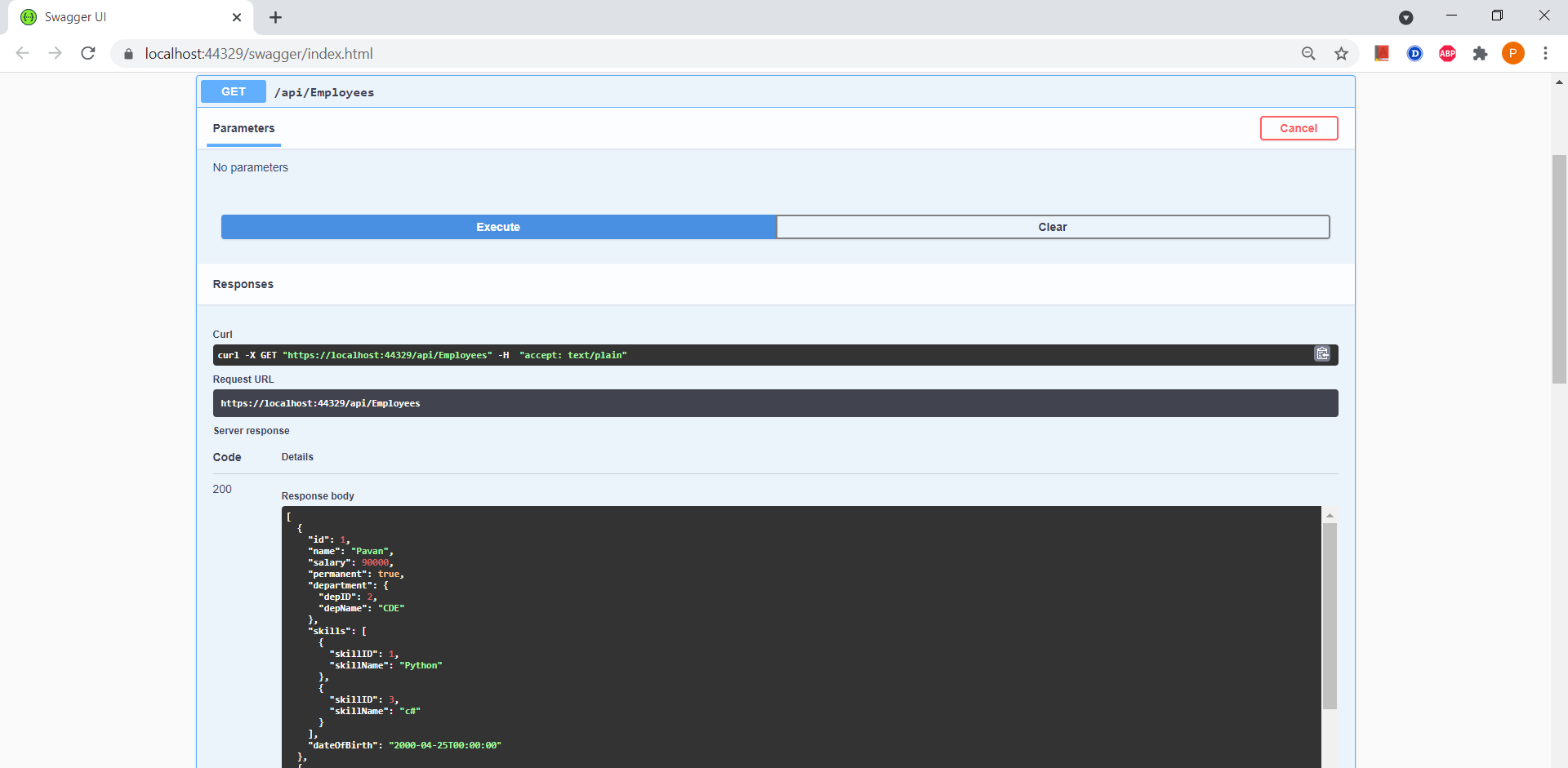
}

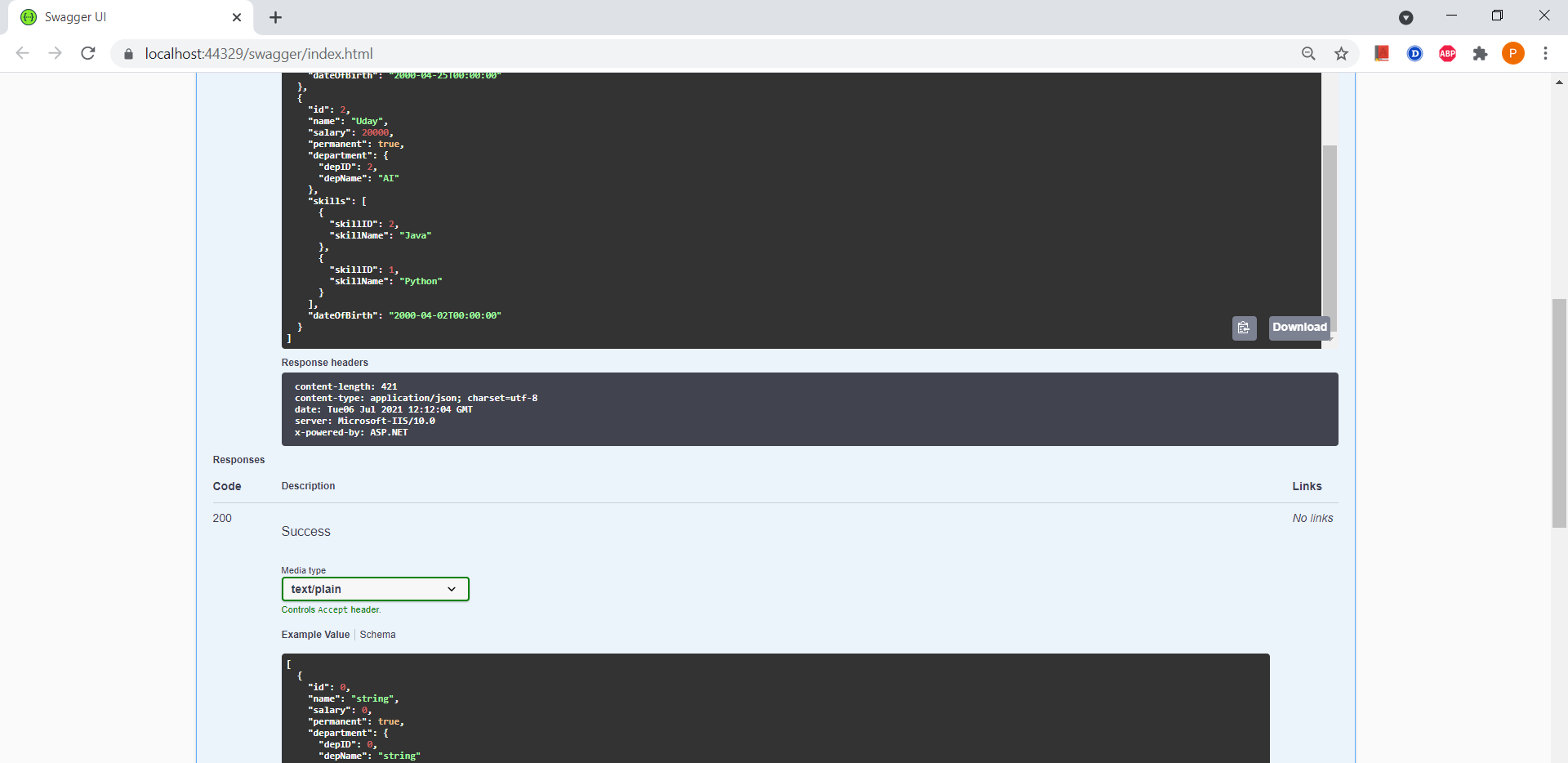
}

}

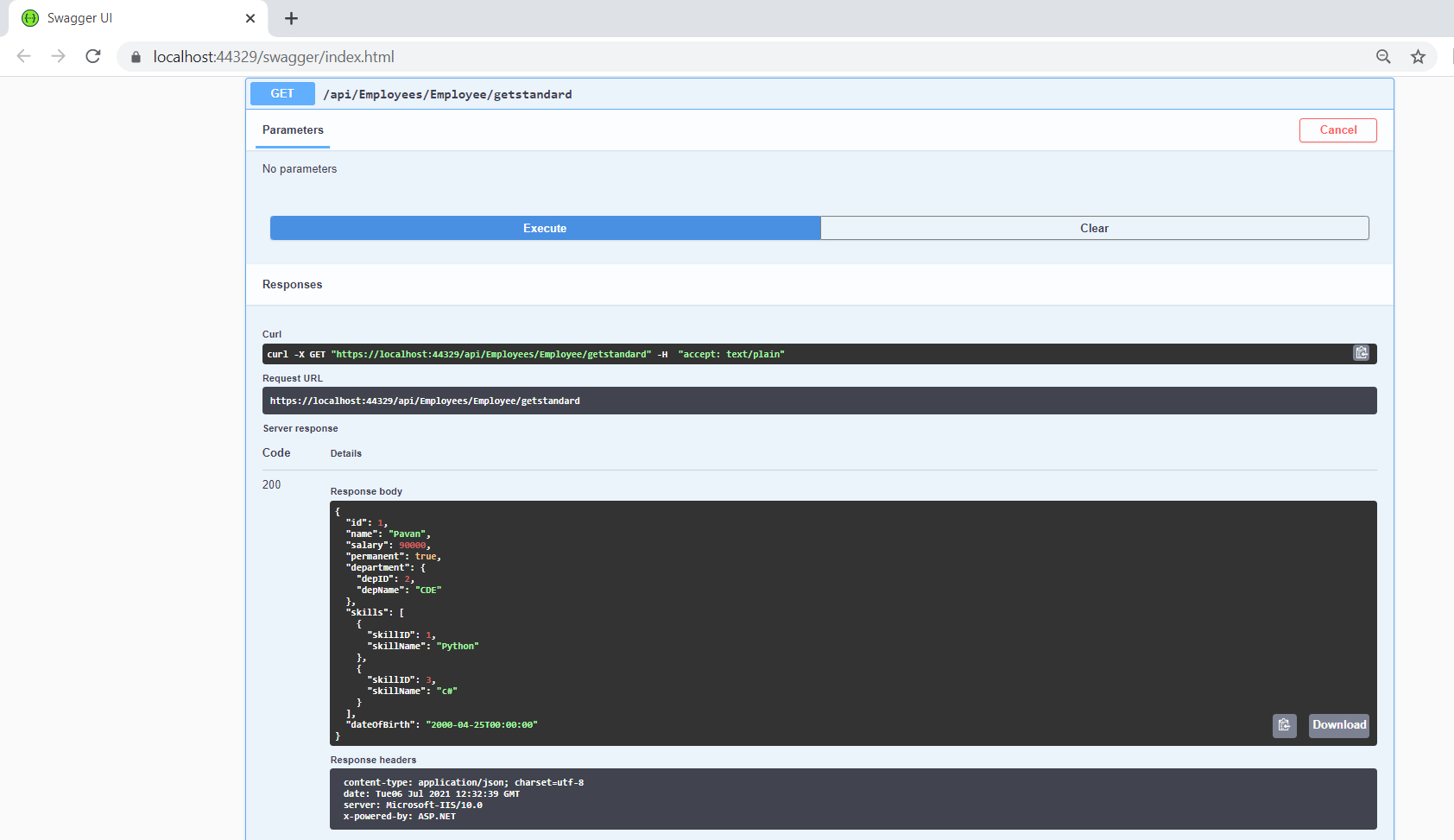
}

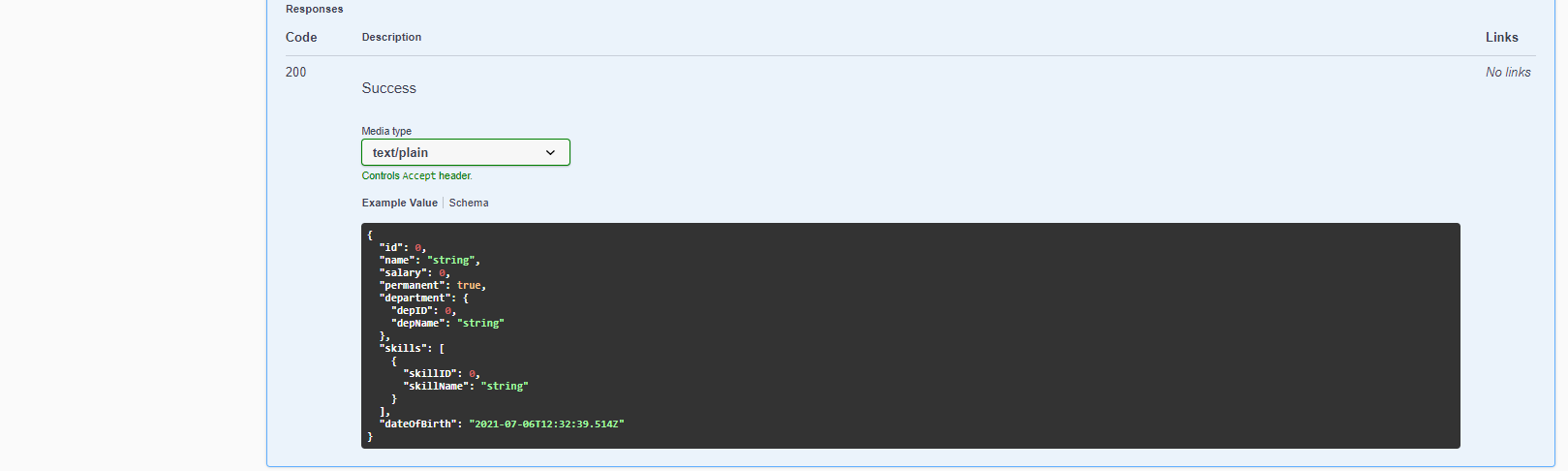
**Output:**

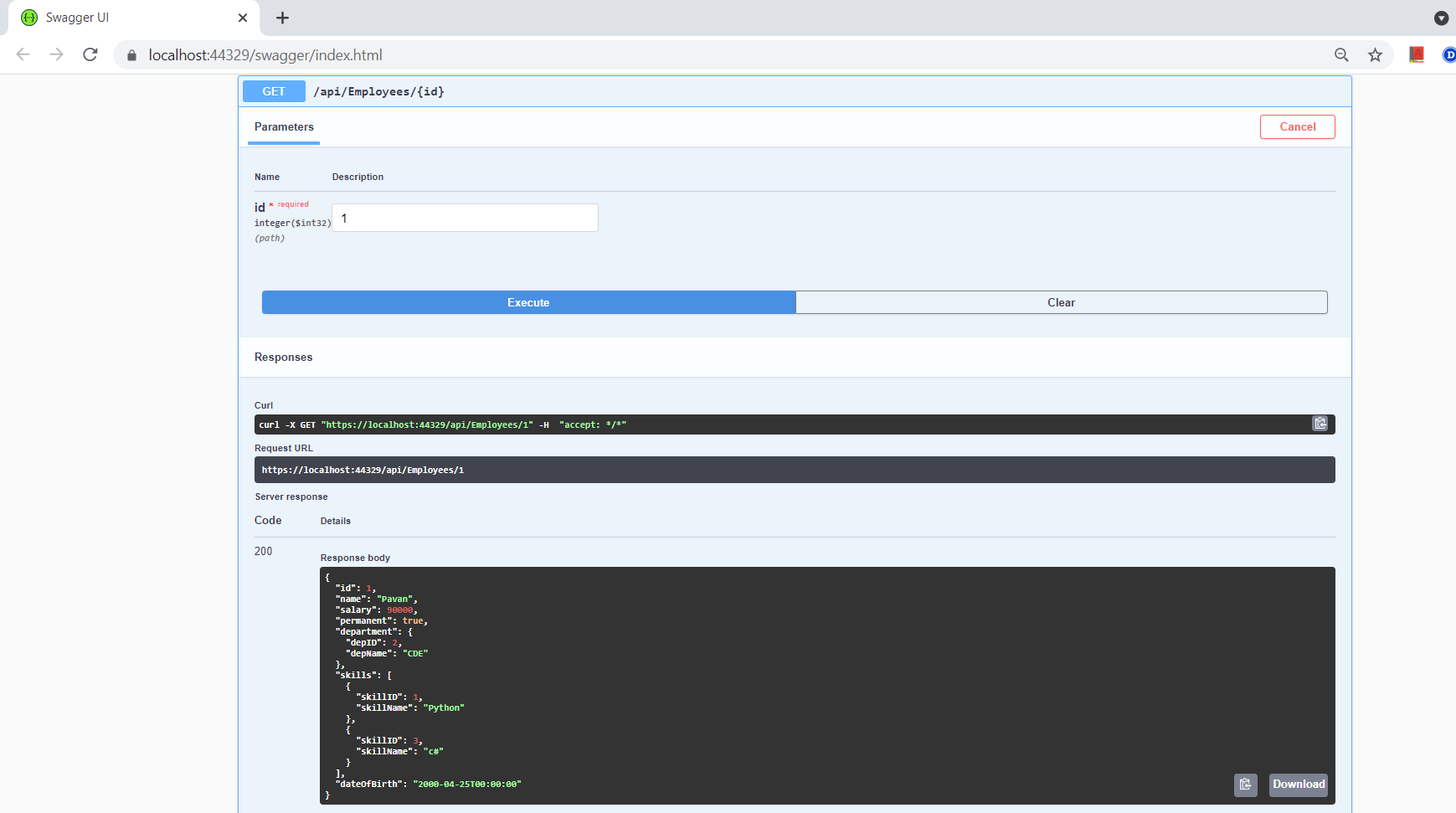


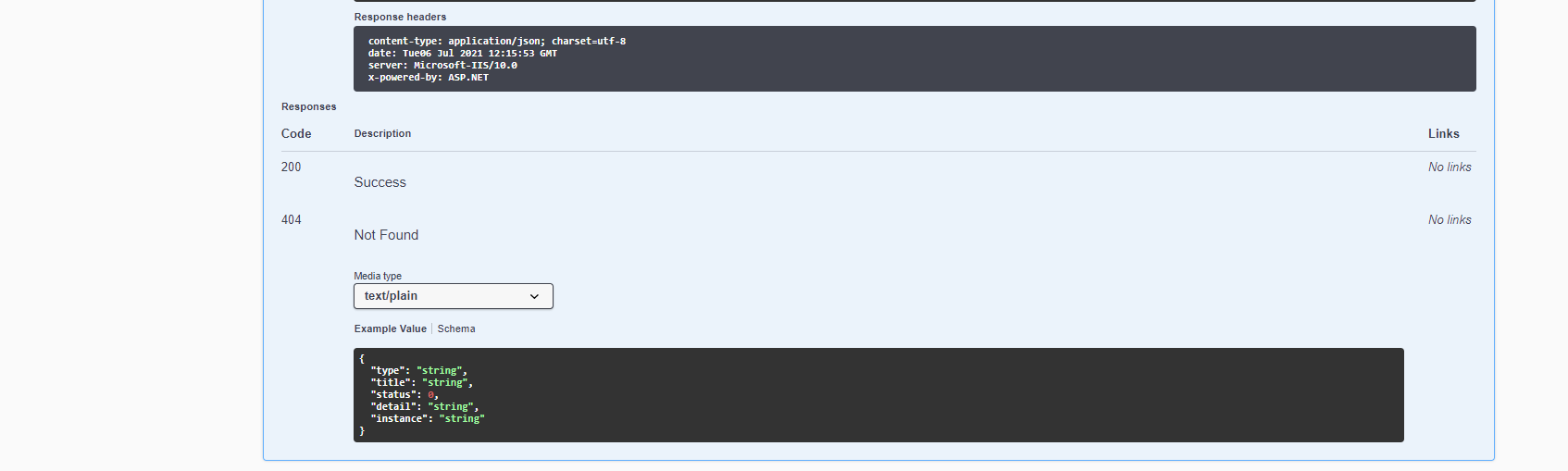


**Getstandard methods**









**--------------------------------------------------------------------------------------------------------------**

**Hands-on3**

**Employee.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Day2\_Handson.Model;

namespace Day2\_Handson

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**EmployeesController.cs**

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Day2\_Handson.Model;

using Day2\_Handson.Filters;

using Microsoft.AspNetCore.Http;

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

namespace Day2\_Handson.Controllers

{

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

[ApiController]

public class EmployeesController : ControllerBase

{

private static List<Employee> \_emp = new List<Employee>();

// GET: api/<ValuesController1>

[HttpGet(Name = "GetAllStudent")]

public IActionResult Get()

{

return new ObjectResult(\_emp);

}

// GET api/<ValuesController1>/5

[HttpGet("{id}", Name = "GetStudent")]

public IActionResult Get(int id)

{

return new ObjectResult(\_emp.FirstOrDefault(p => p.Id == id));

}

// POST api/<ValuesController1>

[HttpPost(Name = "CreateStudent")]

public IActionResult Post([FromBody] Employee emps)

{

\_emp.Add(emps);

return CreatedAtRoute("GetStudent", new { id = emps.Id }, emps);

}

// PUT api/<ValuesController1>/5

[HttpPut("{id}", Name = "UpdateStudent")]

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

{

\_emp.FirstOrDefault(p => p.Id == id).Name = emps.Name;

return CreatedAtRoute("GetStudent", new { id = emps.Id }, emps);

}

// DELETE api/<ValuesController1>/5

[HttpDelete("{id}", Name = "DeleteStudent")]

public IActionResult Delete(int id)

{

var \_emps = \_emp.FirstOrDefault(p => p.Id == id);

\_emp.Remove(\_emps);

return new NoContentResult();

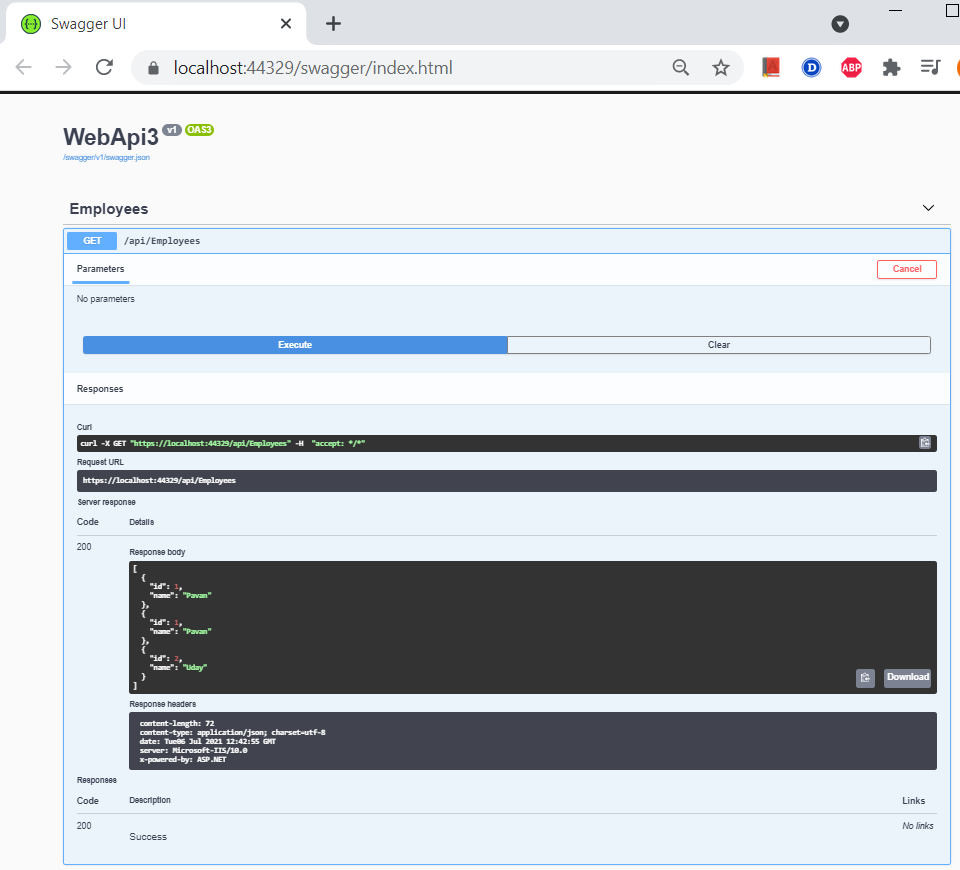
}

}

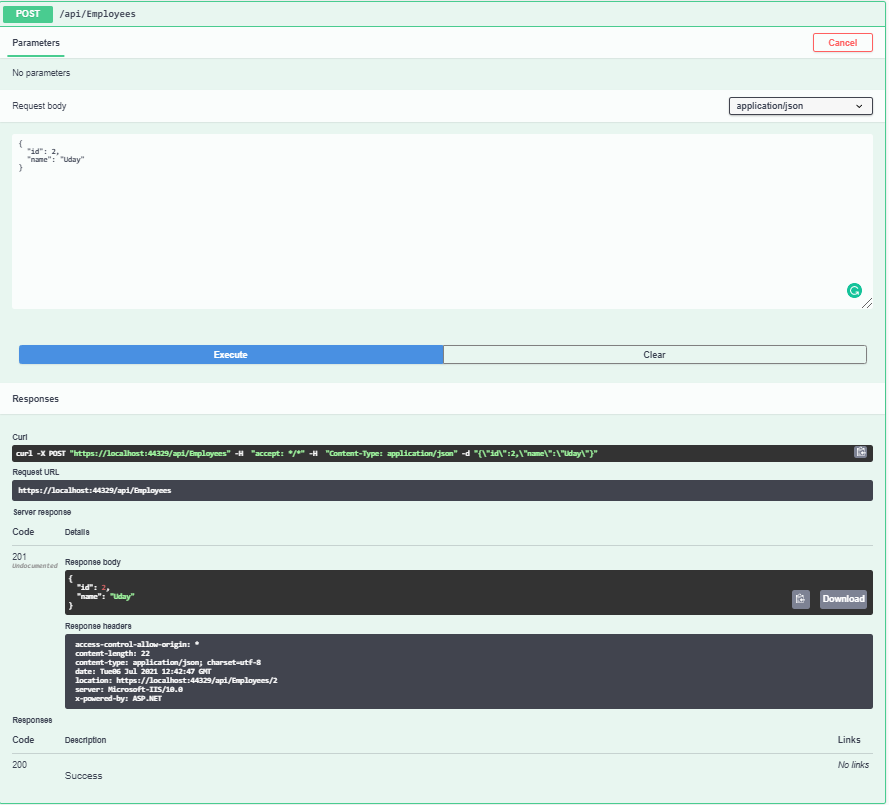
}

**Output:**

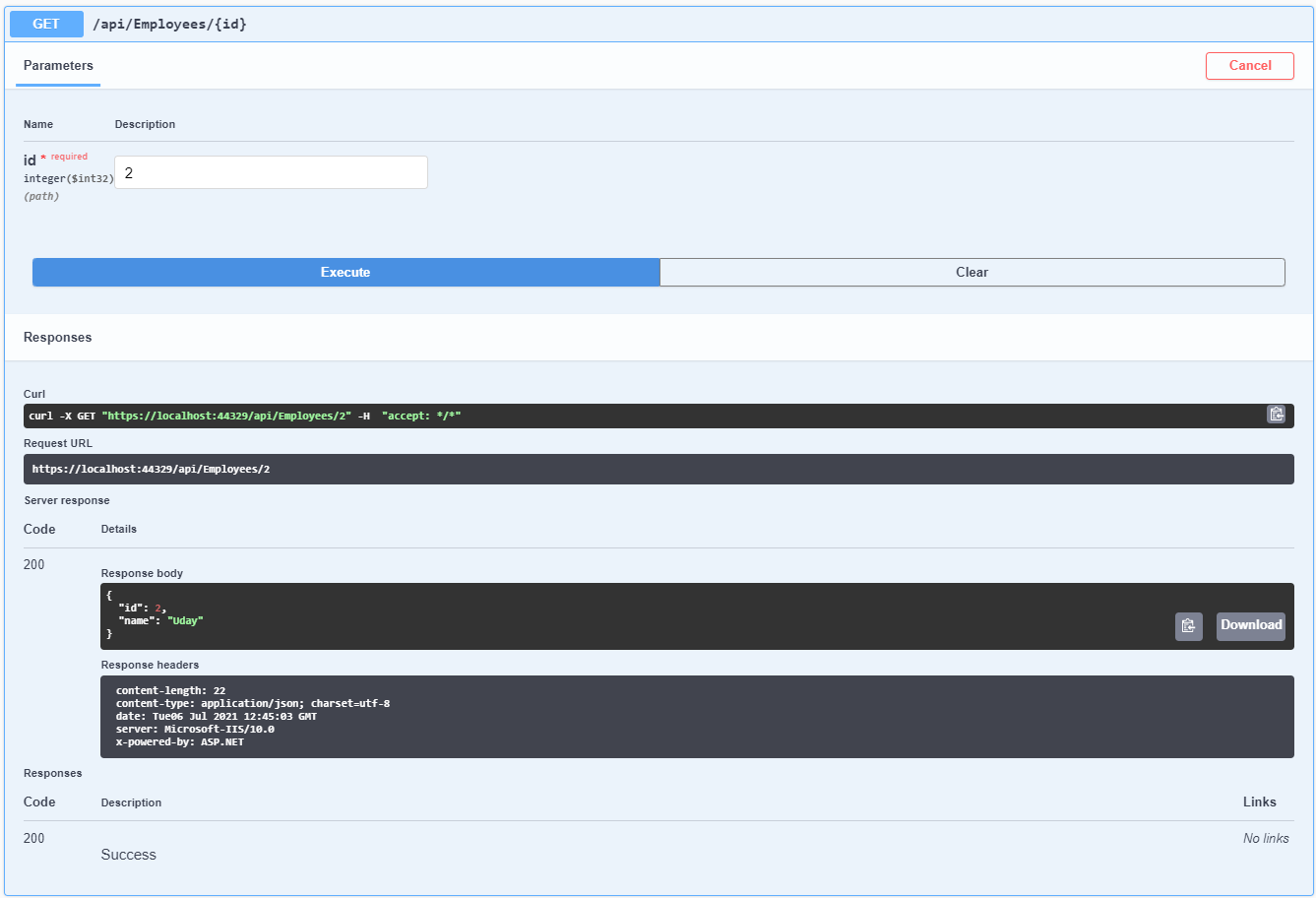
**After adding values**



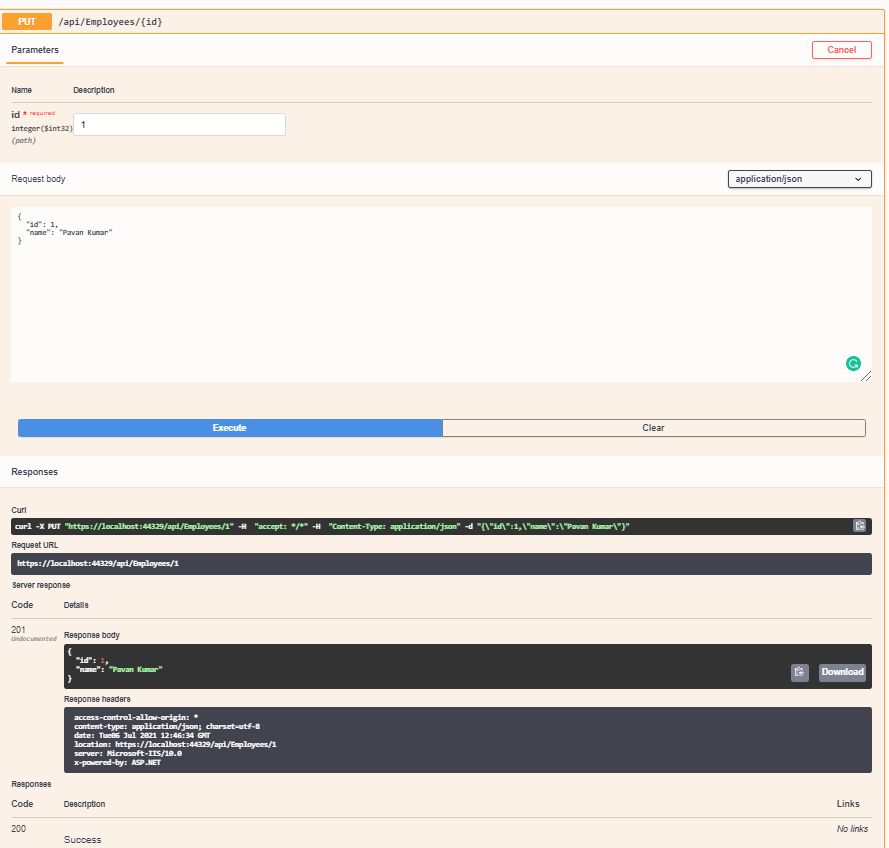
**Posting values**



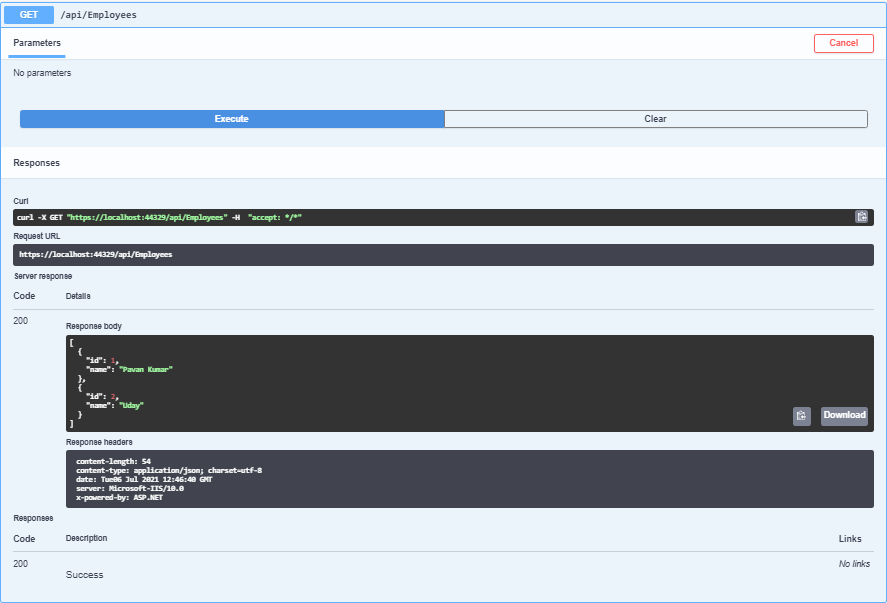
**Getting values by ID**



**Updating values by ID**



**After updated values**



**Deleting values by ID**

