**Web API Assignments ( Week – 4 )**

**Assignment - 1**

ValueController.cs :-

using Microsoft.AspNetCore.Mvc;

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

namespace MyFirstWebAPI.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)

{

}

}

}

**OUTPUT :-**

*Get method on Swagger.*

**

**Assignment – 2**

*Program.cs :-*

using Microsoft.AspNetCore.Builder;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer(); // Optional but helps Swagger

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

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

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();

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI(c =>

{

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

});

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

*EmployeeController.cs :-*

using Microsoft.AspNetCore.Builder;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer(); // Optional but helps Swagger

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

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

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();

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI(c =>

{

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

});

}

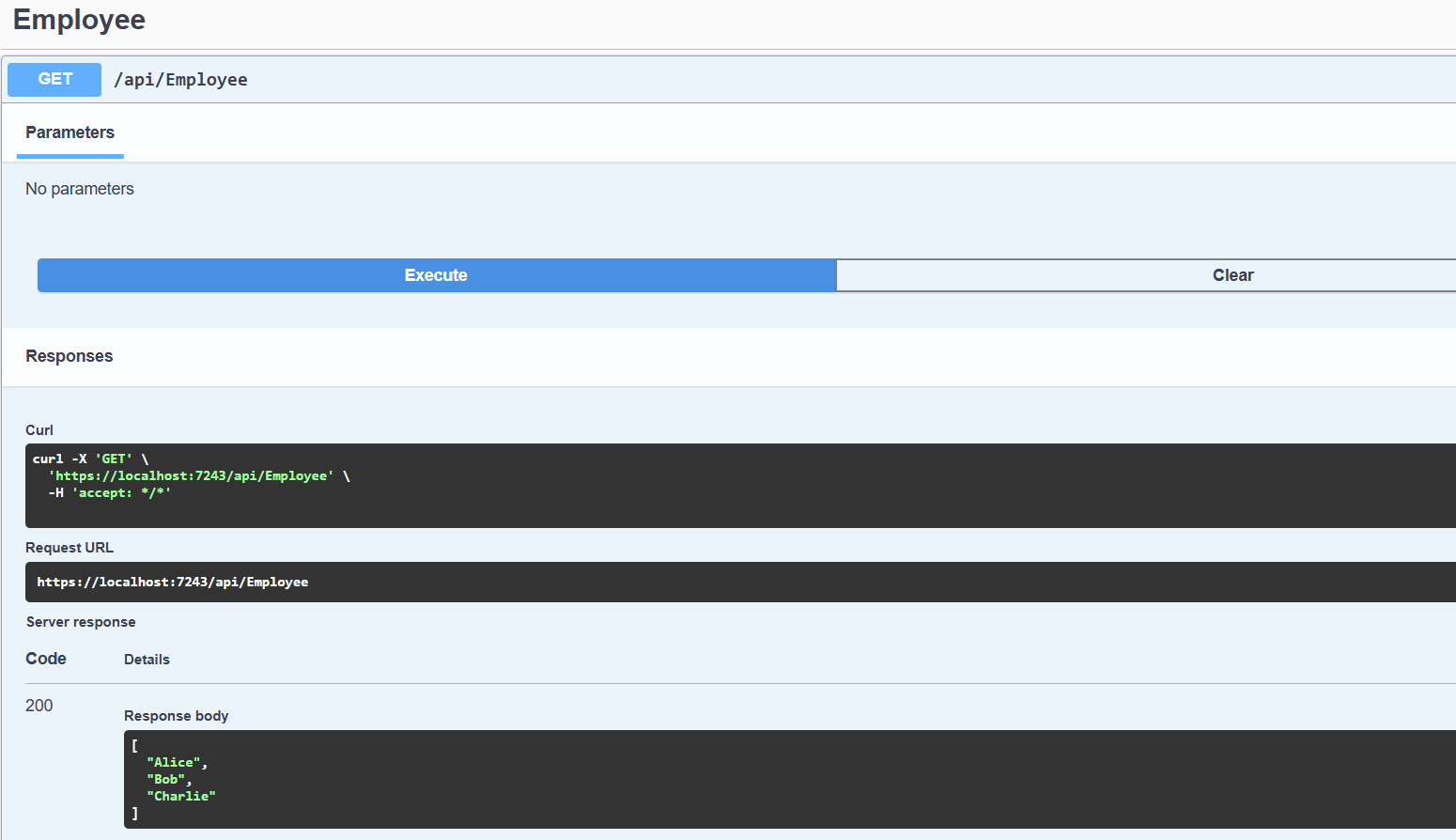
app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

**OUTPUT :-**



**Assignment – 3**

New folder called ‘**Models**’ is created.

It consists of 3 classes namely:-

Department.cs

Skill.cs

Employee.cs

*Department.cs :-*

namespace MyFirstWebAPI.Models

{

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

}

}

*Skill.cs :-*

namespace MyFirstWebAPI.Models

{

public class Skill

{

public int Id { get; set; }

public string Name { get; set; }

}

}

*Employee.cs :-*

using System;

using System.Collections.Generic;

namespace MyFirstWebAPI.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 Department Department { get; set; }

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

public DateTime DateOfBirth { get; set; }

}

}

*Updated EmployeeController.cs file :-*

using Microsoft.AspNetCore.Mvc;

using MyFirstWebAPI.Models;

namespace MyFirstWebAPI.Controllers

{

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

[ApiController]

public class EmployeeController : ControllerBase

{

private static List<Employee> \_employees = new List<Employee>

{

new Employee

{

Id = 1,

Name = "Shreyas",

Salary = 50000,

Permanent = true,

Department = new Department

{

Id = 101,

Name = "CSE"

},

Skills = new List<Skill>

{

new Skill { Id = 1, Name = "C#" },

new Skill { Id = 2, Name = "SQL" }

},

DateOfBirth = new DateTime(2004, 8, 21)

},

new Employee

{

Id = 2,

Name = "Alice",

Salary = 45000,

Permanent = false,

Department = new Department

{

Id = 102,

Name = "ECE"

},

Skills = new List<Skill>

{

new Skill { Id = 3, Name = "Java" },

new Skill { Id = 4, Name = "Python" }

},

DateOfBirth = new DateTime(1998, 4, 10)

}

};

// GET: api/employee

[HttpGet]

[ProducesResponseType(typeof(List<Employee>), 200)]

[ProducesResponseType(500)]

public IActionResult Get()

{

return Ok(\_employees);

}

// GET: api/employee/1

[HttpGet("{id}")]

public IActionResult Get(int id)

{

var employee = \_employees.FirstOrDefault(e => e.Id == id);

if (employee == null)

return NotFound();

return Ok(employee);

}

// POST: api/employee

[HttpPost]

public IActionResult Post([FromBody] Employee employee)

{

\_employees.Add(employee);

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

}

// PUT: api/employee/1

[HttpPut("{id}")]

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

{

var employee = \_employees.FirstOrDefault(e => e.Id == id);

if (employee == null)

return NotFound();

employee.Name = updatedEmployee.Name;

employee.Salary = updatedEmployee.Salary;

employee.Permanent = updatedEmployee.Permanent;

employee.Department = updatedEmployee.Department;

employee.Skills = updatedEmployee.Skills;

employee.DateOfBirth = updatedEmployee.DateOfBirth;

return NoContent();

}

// DELETE: api/employee/1

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

var employee = \_employees.FirstOrDefault(e => e.Id == id);

if (employee == null)

return NotFound();

\_employees.Remove(employee);

return NoContent();

}

}

}

*Updated .csproj file :-*

<Project Sdk="Microsoft.NET.Sdk.Web">

<PropertyGroup>

<TargetFramework>net8.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

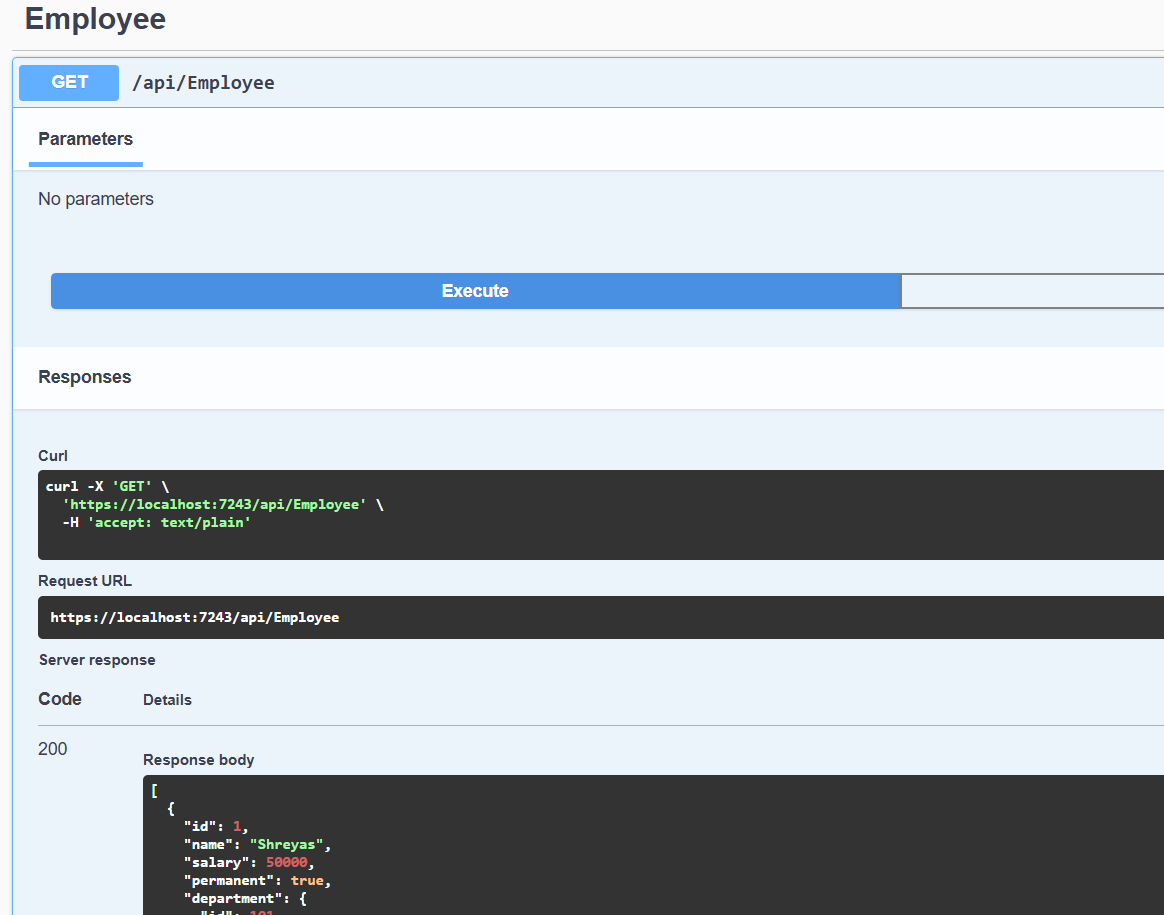
<ItemGroup>

<PackageReference Include="Swashbuckle.AspNetCore" Version="9.0.3" />

</ItemGroup>

</Project>

**OUTPUT :-**

****

**

**Assignment – 4**

*Updated EmployeeController.cs :-*

using Microsoft.AspNetCore.Mvc;

using MyFirstWebAPI.Models;

using System;

using System.Collections.Generic;

using System.Linq;

namespace MyFirstWebAPI.Controllers

{

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

[ApiController]

public class EmployeeController : ControllerBase

{

// 🔹 Hardcoded employee list

private static List<Employee> \_employees = new List<Employee>

{

new Employee

{

Id = 1,

Name = "Shreyas",

Salary = 50000,

Permanent = true,

Department = new Department

{

Id = 101,

Name = "CSE"

},

Skills = new List<Skill>

{

new Skill { Id = 1, Name = "C#" },

new Skill { Id = 2, Name = "SQL" }

},

DateOfBirth = new DateTime(2004, 8, 21)

},

new Employee

{

Id = 2,

Name = "Alice",

Salary = 45000,

Permanent = false,

Department = new Department

{

Id = 102,

Name = "ECE"

},

Skills = new List<Skill>

{

new Skill { Id = 3, Name = "Java" },

new Skill { Id = 4, Name = "Python" }

},

DateOfBirth = new DateTime(1998, 4, 10)

}

};

[HttpGet]

[ProducesResponseType(typeof(List<Employee>), 200)]

[ProducesResponseType(500)]

public IActionResult Get()

{

return Ok(\_employees);

}

[HttpGet("{id}")]

public IActionResult Get(int id)

{

var employee = \_employees.FirstOrDefault(e => e.Id == id);

if (employee == null)

return NotFound();

return Ok(employee);

}

// POST: api/employee

[HttpPost]

public IActionResult Post([FromBody] Employee employee)

{

\_employees.Add(employee);

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

}

// ✅ PUT: api/employee/{id}

[HttpPut("{id}")]

[ProducesResponseType(typeof(Employee), 200)]

[ProducesResponseType(400)]

public ActionResult<Employee> Put(int id, [FromBody] Employee updatedEmployee)

{

if (id <= 0)

{

return BadRequest("Invalid employee id");

}

var employee = \_employees.FirstOrDefault(e => e.Id == id);

if (employee == null)

{

return BadRequest("Invalid employee id");

}

employee.Name = updatedEmployee.Name;

employee.Salary = updatedEmployee.Salary;

employee.Permanent = updatedEmployee.Permanent;

employee.Department = updatedEmployee.Department;

employee.Skills = updatedEmployee.Skills;

employee.DateOfBirth = updatedEmployee.DateOfBirth;

return Ok(employee);

}

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

var employee = \_employees.FirstOrDefault(e => e.Id == id);

if (employee == null)

return NotFound();

\_employees.Remove(employee);

return NoContent();

}

}

}

*Input at Swagger :-*

*{*

*"id": 1,*

*"name": "Shreyas the Dev",*

*"salary": 99999,*

*"permanent": true,*

*"department": {*

*"id": 201,*

*"name": "Engineering"*

*},*

*"skills": [*

*{ "id": 1, "name": "C#" },*

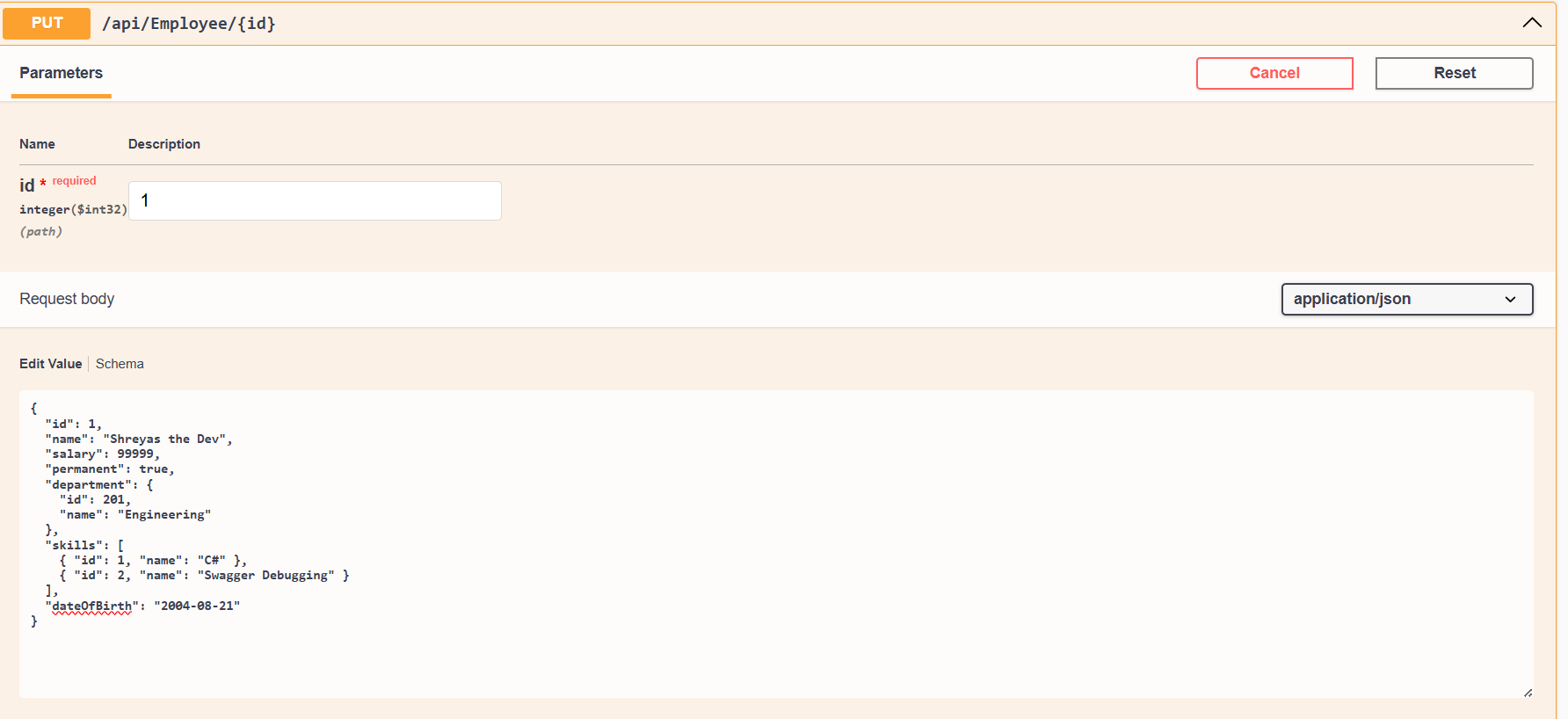
*{ "id": 2, "name": "Swagger Debugging" }*

*],*

*"dateOfBirth": "2004-08-21"*

*}*

**OUTPUT:-**

****

****

**Assignment – 5**

*Updated Program.cs :-*

using Microsoft.AspNetCore.Authentication.JwtBearer;

using Microsoft.IdentityModel.Tokens;

using System.Text;

var builder = WebApplication.CreateBuilder(args);

var key = "mysuperdupersecurekeythatishardtoguess123!";

var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(key));

builder.Services.AddAuthentication(x =>

{

x.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

x.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;

})

.AddJwtBearer(x =>

{

x.TokenValidationParameters = new TokenValidationParameters

{

ValidateIssuer = true,

ValidateAudience = true,

ValidateLifetime = true,

ValidateIssuerSigningKey = true,

ValidIssuer = "mySystem",

ValidAudience = "myUsers",

IssuerSigningKey = securityKey

};

});

builder.Services.AddAuthorization();

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

app.UseHttpsRedirection();

app.UseAuthentication();

app.UseAuthorization();

app.MapControllers();

app.Run();

*AuthController.cs :-*

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using Microsoft.IdentityModel.Tokens;

using System.IdentityModel.Tokens.Jwt;

using System.Security.Claims;

using System.Text;

namespace MyFirstWebAPI.Controllers

{

[ApiController]

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

[AllowAnonymous]

public class AuthController : ControllerBase

{

[HttpGet]

public IActionResult GetToken()

{

var token = GenerateJSONWebToken(1, "Admin");

return Ok(new { token });

}

private string GenerateJSONWebToken(int userId, string userRole)

{

var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("mysuperdupersecurekeythatishardtoguess123!"));

var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);

var claims = new[]

{

new Claim(ClaimTypes.Role, userRole),

new Claim("UserId", userId.ToString())

};

var token = new JwtSecurityToken(

issuer: "mySystem",

audience: "myUsers",

claims: claims,

expires: DateTime.Now.AddMinutes(10),

signingCredentials: credentials);

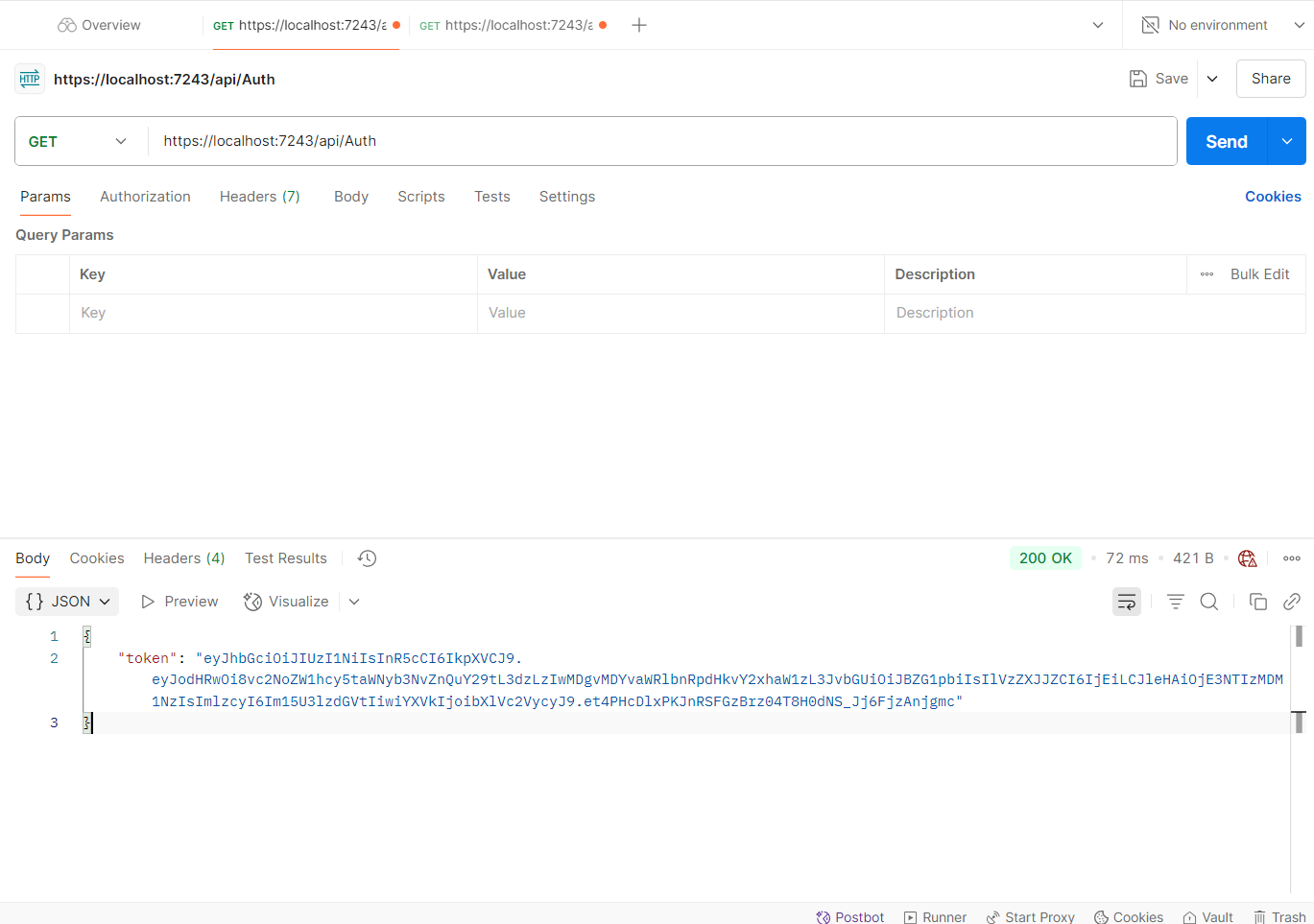
return new JwtSecurityTokenHandler().WriteToken(token);

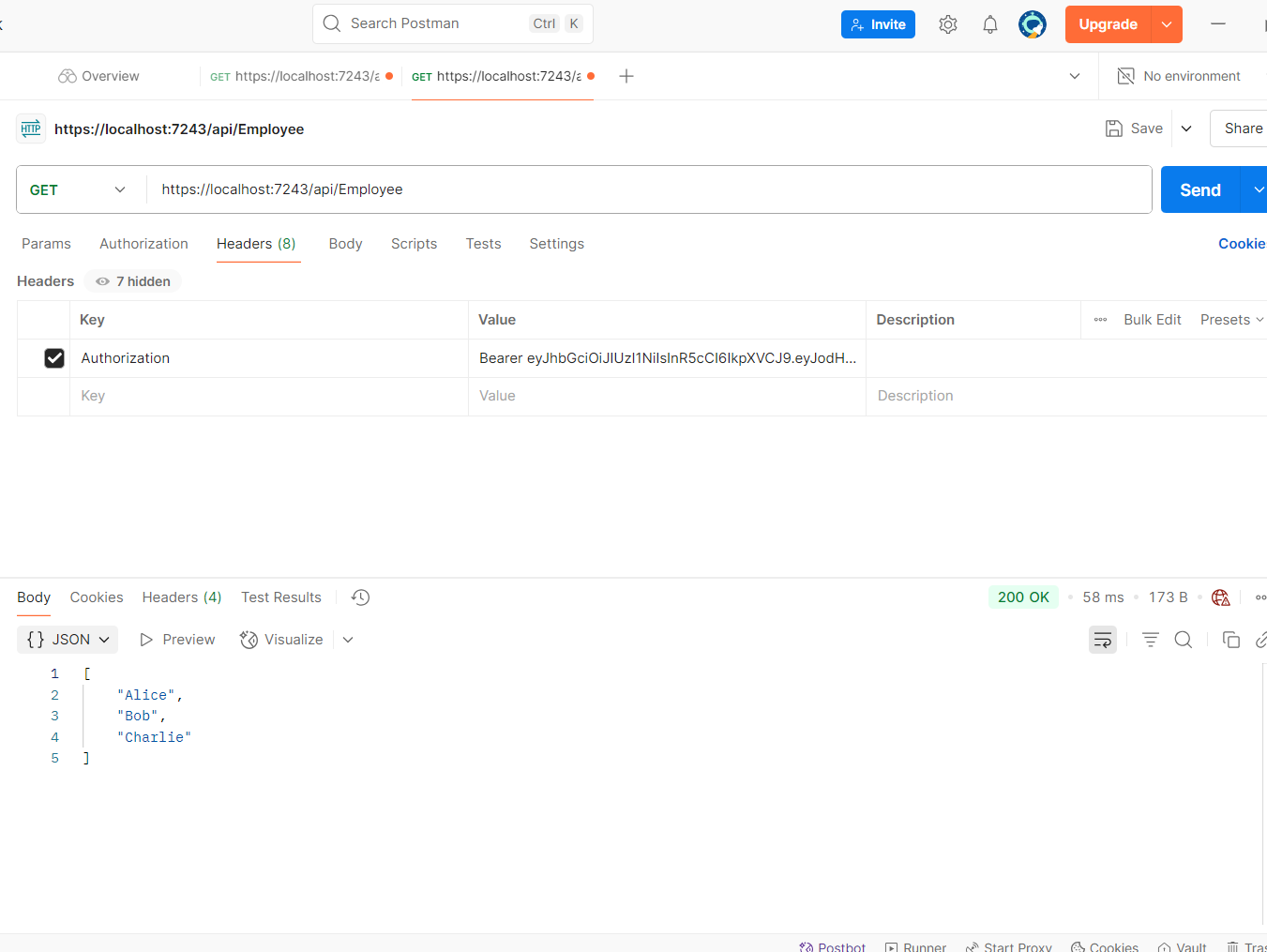
}

}

}

**OUTPUT :-**

****

****