LAB-1

Program.cs > Program > <top-level-statements-entry-point>

```
var builder = WebApplication.CreateBuilder(args);


builder.Services.AddOpenApi();

var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.MapOpenApi();
}

app.UseHttpsRedirection();

var summaries = new[]
{
    "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering", "Scorching"
};

app.MapGet("/weatherforecast", () =>
{
    var forecast =  Enumerable.Range(1, 5).Select(index =>
        new WeatherForecast
        (
            DateOnly.FromDateTime(DateTime.Now.AddDays(index)),
            Random.Shared.Next(-20, 55),
            summaries[Random.Shared.Next(summaries.Length)]
        ))
        .ToArray();
    return forecast;
})
.WithName("GetWeatherForecast");

app.Run();

1 reference
record WeatherForecast(DateOnly Date, int TemperatureC, string? Summary)
{
    0 references
    public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);
}
```

Controllers > ValuesController.cs > ...

```
using Microsoft.AspNetCore.Mvc;

namespace MyFirstWebAPI.Controllers
{
    [ApiController]
    [Route("[controller]")]
    0 references
    public class ValuesController : ControllerBase
    {
        [HttpGet]
        0 references
        public IActionResult Get()
        {
            return Ok(new string[] { "value1", "value2" });
        }

        [HttpGet("{id}")]
        0 references
        public IActionResult Get(int id)
        {
            return Ok("value " + id);
        }

        [HttpPost]
        0 references
        public IActionResult Post([FromBody] string value)
        {
            return Ok("Posted: " + value);
        }

        [HttpPut("{id}")]
        0 references
        public IActionResult Put(int id, [FromBody] string value)
        {
            return Ok($"Updated id {id} with value '{value}'");
        }

        [HttpDelete("{id}")]
        0 references
        public IActionResult Delete(int id)
        {
            return Ok($"Deleted value with id {id}");
        }
    }
}
```

MyFirstWebAPI.csproj

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

  <PropertyGroup>
    <TargetFramework>net9.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="9.0.6" />
  </ItemGroup>

</Project>
```

OUTPUT:

```
PS C:\Users\Lenovo\MyFirstWebAPI> dotnet run
>>
Using launch settings from C:\Users\Lenovo\MyFirstWebAPI\Properties\launchSettings.json...
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5224
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Lenovo\MyFirstWebAPI
```

[{"date":"2025-07-12","temperatureC":15,"summary":"Bracing","temperatureF":58},{"date":"2025-07-13","temperatureC":-14,"summary":"Scorching","temperatureF":7},{"date":"2025-07-14","temperatureC":18,"summary":"Sweltering","temperatureF":64},{"date":"2025-07-15","temperatureC":-17,"summary":"Bracing","temperatureF":2},{"date":"2025-07-16","temperatureC":48,"summary":"Scorching","temperatureF":118}]

LAB-2

Controllers > EmployeeController.cs > EmployeeController > Get

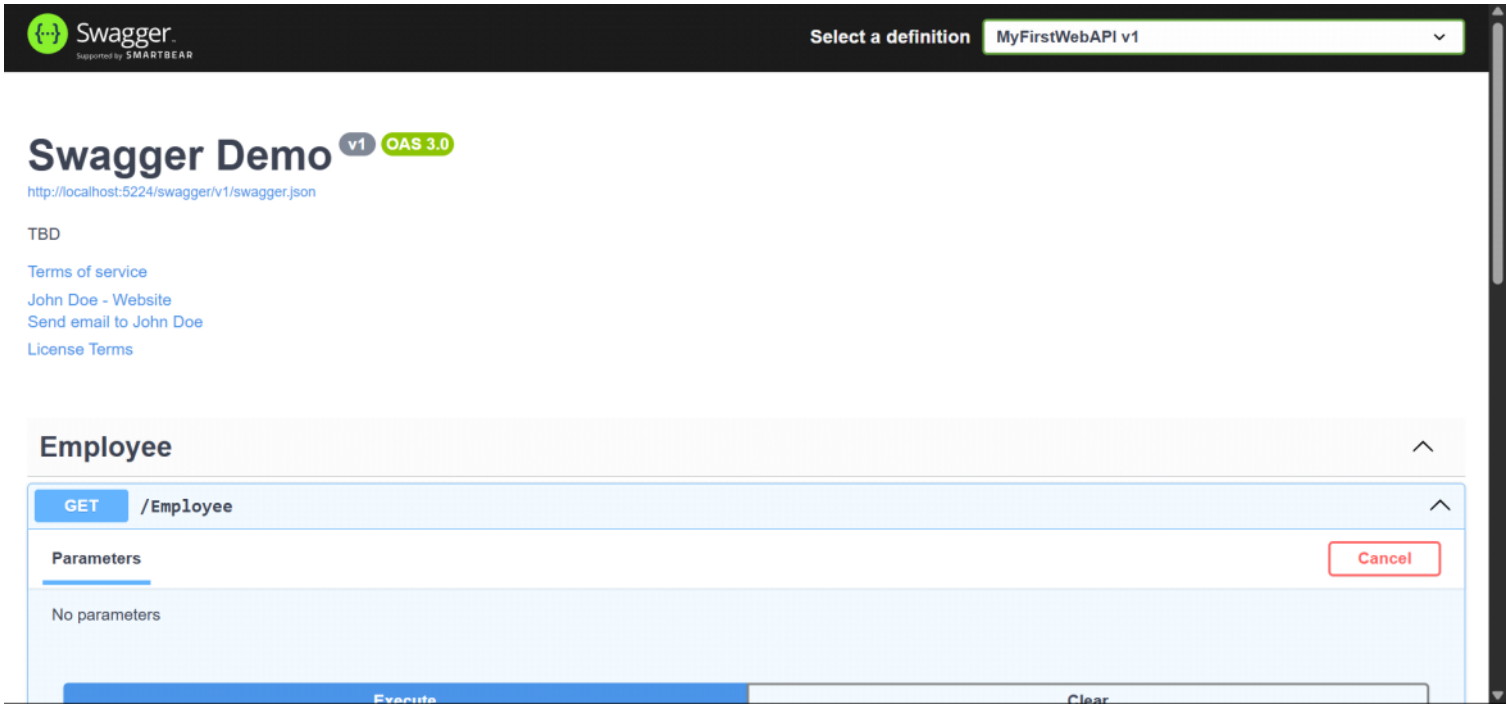
```
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;

namespace MyFirstWebAPI.Controllers
{
    [ApiController]
    [Route("[controller]")] // maps to /Employee
    0 references
    public class EmployeeController : ControllerBase
    {
        [HttpGet]
        0 references
        public IEnumerable<string> Get()
        {
            return new string[] { "John", "Doe", "Smith" };
        }

        [HttpGet("{id}")]
        0 references
        public string Get(int id)
        {
            var employees = new string[] { "John", "Doe", "Smith" };
            if (id >= 0 && id < employees.Length)
                return employees[id];
            return "Employee not found";
        }
    }
}
```

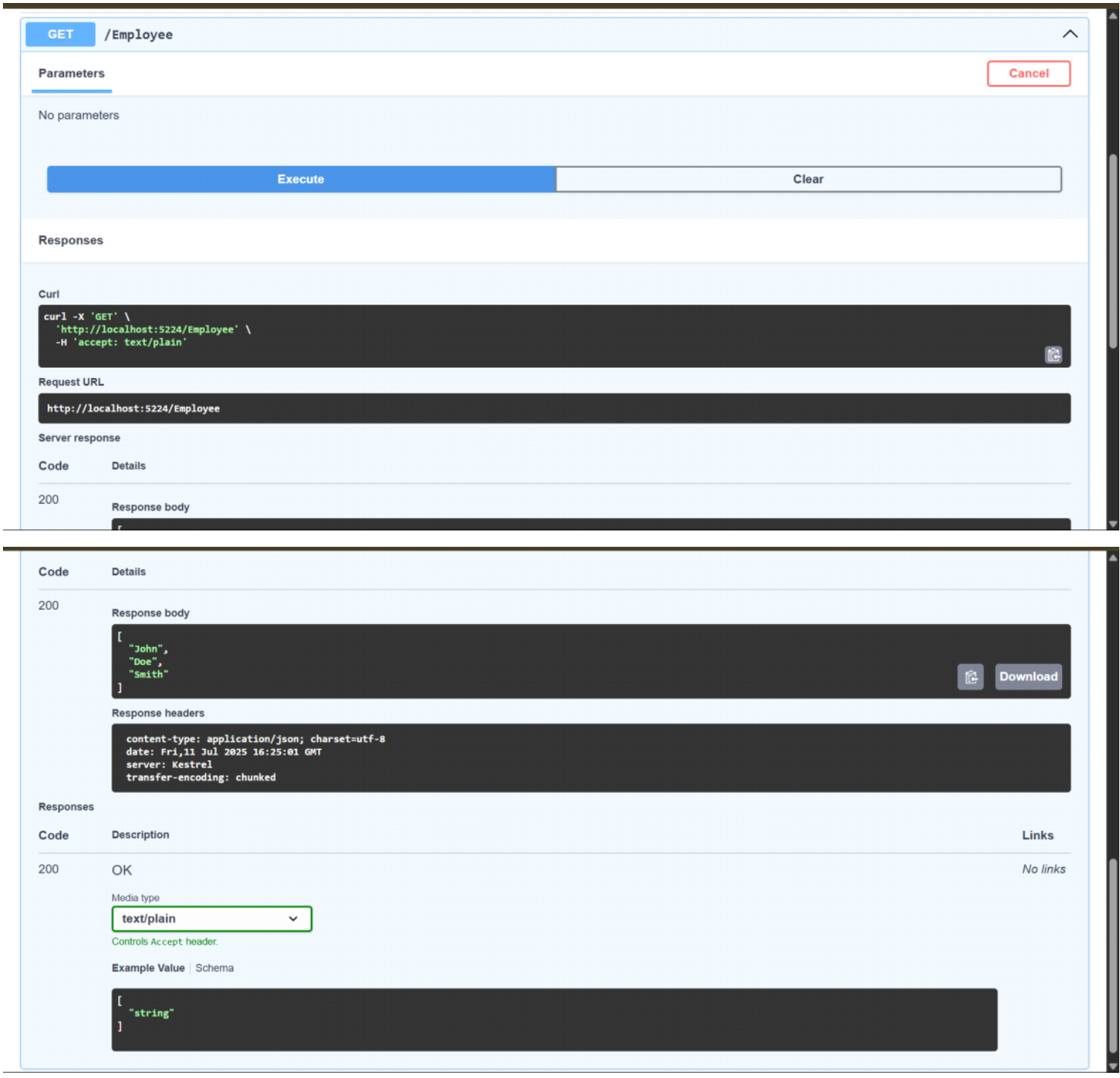
```
using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
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();
}

app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();

app.Run();
```

MyFirstWebAPI.csproj

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

  <PropertyGroup>
    <TargetFramework>net9.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="9.0.6" />
    <PackageReference Include="Swashbuckle.AspNetCore" Version="9.0.3" />
  </ItemGroup>

</Project>
```

```
PS C:\Users\Lenovo\MyFirstWebAPI> dotnet run
Using launch settings from C:\Users\Lenovo\MyFirstWebAPI\Properties\launchSettings.json...
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5224
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Lenovo\MyFirstWebAPI
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```




GET
/Employee
Parameters
Cancel
No parameters
Execute
Clear
Responses
Curl
curl -X 'GET' \
'http://localhost:5224/Employee' \
-H 'accept: text/plain'
Request URL
http://localhost:5224/Employee
Server response
Code
Details
200
Response body

Code
Details
200
Response body
[
"John",
"Doe",
"Smith"
]
Download
Response headers
content-type: application/json; charset=utf-8
date: Fri,11 Jul 2025 16:25:01 GMT
server: Kestrel
transfer-encoding: chunked
Responses
Code
Description
Links
200
OK
No links
Media type
text/plain
Controls Accept header.
Example Value
Schema
[
"string"
]

LAB-3

```
Controllers > EmployeeController.cs > EmployeeController
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;
using System;
using EmployeeApi.Models;
using EmployeeApi.Filters;

namespace EmployeeApi.Controllers
{
    [ApiController]
    [Route("[controller]")]
    [ServiceFilter(typeof(CustomAuthFilter))]
    0 references
    public class EmployeeController : ControllerBase
    {
        1 reference
        private List<Employee> GetStandardEmployeeList()
        {
            return new List<Employee>
            {
                new Employee
                {
                    Id = 1,
                    Name = "John",
                    Salary = 50000,
                    Permanent = true,
                    DateOfBirth = new DateTime(1990, 5, 1),
                    Department = new Department { Id = 1, Name = "HR" },
                    Skills = new List<Skill>
                    {
                        new Skill { Id = 1, SkillName = "C#" },
                        new Skill { Id = 2, SkillName = "ASP.NET" }
                    }
                }
            };
        }

        [HttpGet]
        [ProducesResponseType(200)]
        [ProducesResponseType(500)]
        0 references
        public ActionResult<List<Employee>> GetStandard()
        {
            return Ok(GetStandardEmployeeList());
        }

        [HttpPost]
        [AllowAnonymous]
        0 references
        public ActionResult<Employee> PostEmployee([FromBody] Employee emp)
        {
            return Ok(emp);
        }
    }
}
```

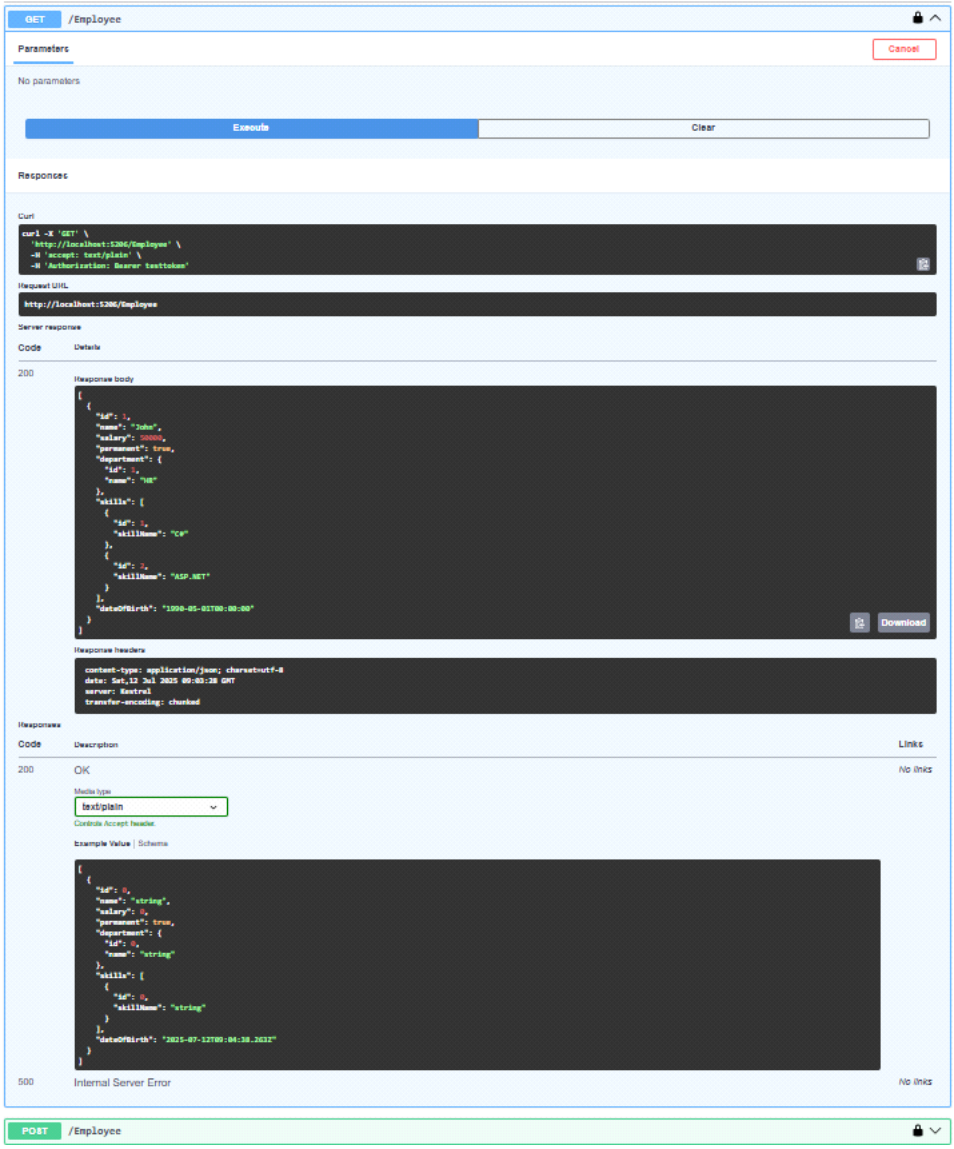
Program.cs > Program > <top-level-statements-entry-point>

```
using EmployeeApi.Filters;
using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers(options =>
{
    options.Filters.Add<CustomExceptionFilter>();
});
builder.Services.AddScoped<CustomAuthFilter>();


builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new OpenApiInfo { Title = "EmployeeApi", Version = "v1" });

    c.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme
    {
        In = ParameterLocation.Header,
        Description = "Enter 'Bearer' followed by your token",
        Name = "Authorization",
        Type = SecuritySchemeType.ApiKey
    });

    c.AddSecurityRequirement(new OpenApiSecurityRequirement
    {
        {
            new OpenApiSecurityScheme
            {
                Reference = new OpenApiReference
                {
                    Type = ReferenceType.SecurityScheme,
                    Id = "Bearer"
                }
            },
            new string[] {}
        }
    });
});

var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();

```

Models > Employee.cs > ...

```
using System;
using System.Collections.Generic;

namespace EmployeeApi.Models
{
    6 references
    public class Employee
    {
        1 reference
        public int Id { get; set; }
        1 reference
        public string Name { get; set; }
        1 reference
        public int Salary { get; set; }
        1 reference
        public bool Permanent { get; set; }
        1 reference
        public Department Department { get; set; }
        1 reference
        public List<Skill> Skills { get; set; }
        1 reference
        public DateTime DateOfBirth { get; set; }
    }

    2 references
    public class Department
    {
        1 reference
        public int Id { get; set; }
        1 reference
        public string Name { get; set; }
    }

    4 references
    public class Skill
    {
        2 references
        public int Id { get; set; }
        2 references
        public string SkillName { get; set; }
    }
}

```

```
Build succeeded with 5 warning(s) in 2.1s
PS C:\Users\Lenovo\EmployeeApi> dotnet run
Using launch settings from C:\Users\Lenovo\EmployeeApi\Properties\launchSettings.json...
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5206
info: Microsoft.Hosting.Lifetime[0]
info: Microsoft.Hosting.Lifetime[0]
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Lenovo\EmployeeApi
```



LAB-4

Program.cs > Program > <top-level-statements-entry-point>

```
using Microsoft.AspNetCore.Builder;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.OpenApi.Models;


var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

```

Controllers > EmployeeController.cs > EmployeeController

```
using Microsoft.AspNetCore.Mvc;
using EmployeeApi.Models;
using System.Collections.Generic;
using System.Linq;

namespace EmployeeApi.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    0 references
    public class EmployeeController : ControllerBase
    {
        1 reference
        private static List<Employee> employees = new List<Employee>
        {
            new Employee { Id = 1, Name = "John", Department = "HR" },
            new Employee { Id = 2, Name = "Alice", Department = "IT" },
            new Employee { Id = 3, Name = "Bob", Department = "Finance" }
        };

        [HttpPut("{id}")]
        0 references
        public ActionResult<Employee> UpdateEmployee(int id, [FromBody] Employee updatedEmp)
        {
            if (id <= 0)
            {
                return BadRequest("Invalid employee id");
            }

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

            if (existingEmployee == null)
            {
                return BadRequest("Invalid employee id");
            }

            existingEmployee.Name = updatedEmp.Name;
            existingEmployee.Department = updatedEmp.Department;

            return Ok(existingEmployee);
        }
    }
}

```

Models > Employee.cs > ...

```
namespace EmployeeApi.Models
{
    7 references
    public class Employee
    {
        4 references
        public int Id { get; set; }
        5 references
        public string? Name { get; set; }
        5 references
        public string? Department { get; set; }
    }
}

```

```
PS C:\Users\Lenovo\LAB> dotnet run
Using launch settings from C:\Users\Lenovo\LAB\Properties\launchSettings.json...
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5170
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Lenovo\LAB
```

Example Value | Schema


```
{
  "id": 0,
  "name": "string",
  "department": "string"
}
```


Responses

Code	Description	Links
200	OK Media type text/plain Controls Accept header. Example Value | Schema	*No links*


```
{
  "id": 0,
  "name": "string",
  "department": "string"
}
```

LAB-5

Controllers > AuthController.cs > AuthController > GenerateJSONWebToken

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Authorization;
using Microsoft.IdentityModel.Tokens;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;

namespace JwtAuthDemo.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    0 references
    public class AuthController : ControllerBase
    {
        [HttpGet("token")]
        [AllowAnonymous]
        0 references
        public IActionResult GetToken()
        {
            var token = GenerateJSONWebToken(1, "Admin");
            return Ok(new { token });
        }

        1 reference
        private string GenerateJSONWebToken(int userId, string userRole)
        {
var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("mysuperdupersecurekeythatishard2guess!"));
            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(2),
                signingCredentials: credentials);

            return new JwtSecurityTokenHandler().WriteToken(token);
        }
    }
}
```

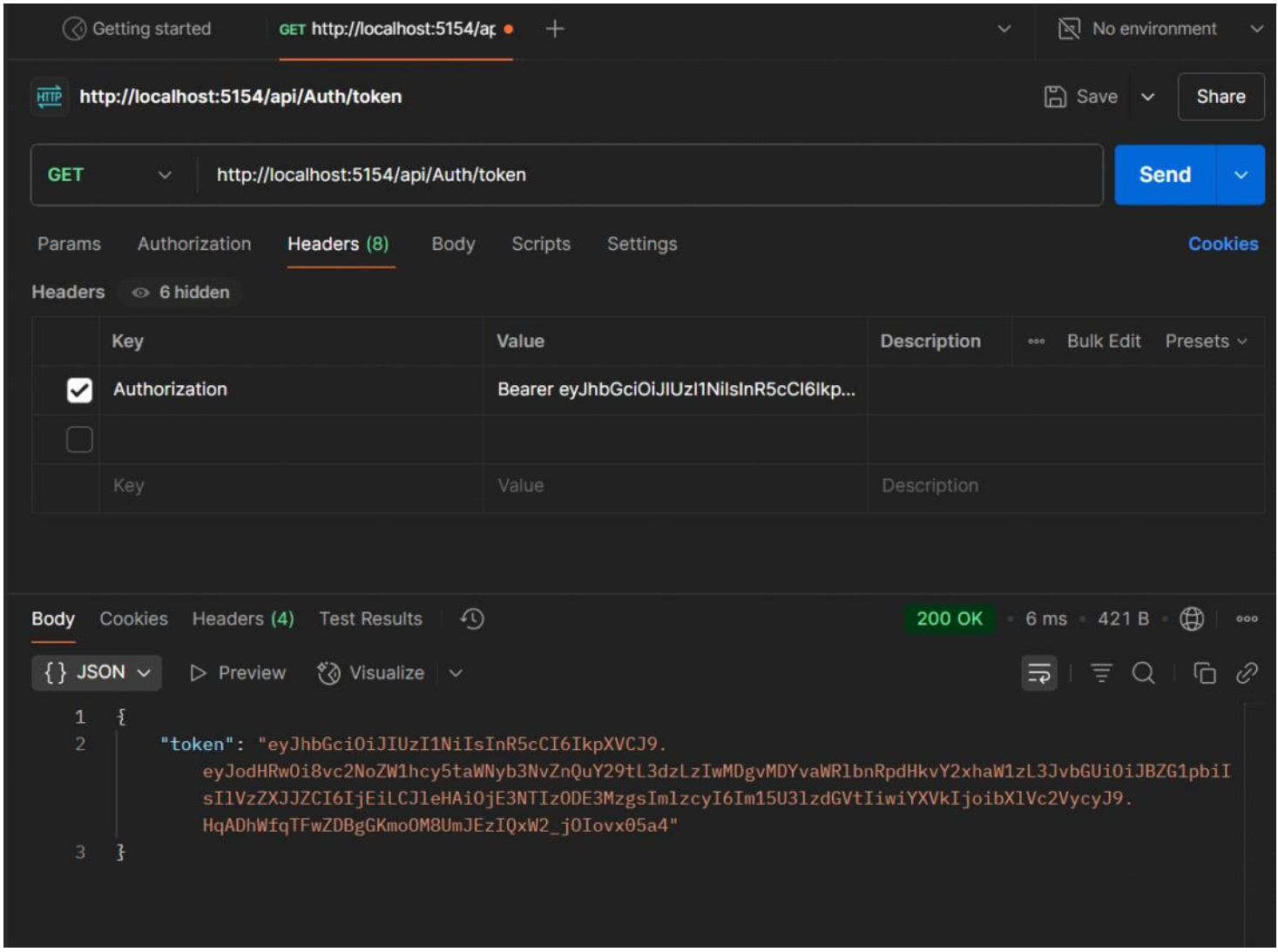
Controllers > EmployeeController.cs > EmployeeController

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;

namespace JwtAuthDemo.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    [Authorize(Roles = "Admin,POC")]
    0 references
    public class EmployeeController : ControllerBase
    {
        [HttpGet]
        0 references
        public IActionResult Get()
        {
            return Ok(" You are authorized to access employee data!");
        }
    }
}
```

Program.cs > Program > <top-level-statements-entry-point>

```
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.IdentityModel.Tokens;
using System.Text;

var builder = WebApplication.CreateBuilder(args);

var securityKey = "mysuperdupersecurekeythatishard2guess!";
var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(securityKey));

builder.Services.AddControllers();
builder.Services.AddCors(options =>
{
    options.AddPolicy("AllowAll", policy =>
    {
        policy.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader();
    });
});
builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
    .AddJwtBearer(options =>
    {
        options.TokenValidationParameters = new TokenValidationParameters
        {
            ValidateIssuer = true,
            ValidateAudience = true,
            ValidateLifetime = true,
            ValidateIssuerSigningKey = true,
            ValidIssuer = "mySystem",
            ValidAudience = "myUsers",
            IssuerSigningKey = key
        };
    });

builder.Services.AddAuthorization();

var app = builder.Build();
app.UseDeveloperExceptionPage();


app.UseCors("AllowAll");
app.UseAuthentication();
app.UseAuthorization();

app.MapControllers();
app.Run();
```

```
PS C:\Users\Lenovo\JwtAuthDemo> dotnet clean
  >> dotnet build
  >> dotnet run
  >>

  Build succeeded in 0.5s
  Restore complete (0.7s)
    JwtAuthDemo succeeded (0.9s) → bin\Debug\net9.0\JwtAuthDemo.dll
                                   file:///C:/Users/Lenovo/JwtAuthDemo/bin/Debug/net9.0 (ctrl + click)

  Build succeeded in 1.8s
  Using launch settings from C:\Users\Lenovo\JwtAuthDemo\Properties\launchSettings.json...
  Building...
  info: Microsoft.Hosting.Lifetime[14]
        Now listening on: http://localhost:5154
  info: Microsoft.Hosting.Lifetime[0]
        Application started. Press Ctrl+C to shut down.
  info: Microsoft.Hosting.Lifetime[0]
        Hosting environment: Development
  info: Microsoft.Hosting.Lifetime[0]
        Content root path: C:\Users\Lenovo\JwtAuthDemo
```


Getting started
GET http://localhost:5154/ap
No environment
http://localhost:5154/api/Auth/token
Save
Share
GET
http://localhost:5154/api/Auth/token
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
Headers
6 hidden
Key
Value
Description
Bulk Edit
Presets
Authorization
Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp...
Key
Value
Description
Body
Cookies
Headers (4)
Test Results
200 OK
6 ms
421 B
JSON
Preview
Visualize
{
"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJodHRwOi8vc2NoZW1hcy5taWNyb3NvZnQuY29tL3dzLzIwMDgvMDYvaWRlbnRpdHkvY2xhaW1zL3JvbGUiOiJBZG1pbiI
sIlVzZXJJZCI6IjEiLCJleHAiOjE3NTIzODE3MzgsImlzcyI6Im15U3lzdGVtIiwiYXVkIjoibXlVc2VycyJ9.
HqADhWfqTFwZDBgGKmoOM8UmJEzIQxW2_jOIovx05a4"
}