

WEEK 3 - ASP.NET Core 8.0 Web API

NAME : SHALU KUMARI

SUPERSET ID : 6363523

1.WebApi_Handson

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
var app = builder.Build();
app.UseSwagger();
app.UseSwaggerUI();
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
```

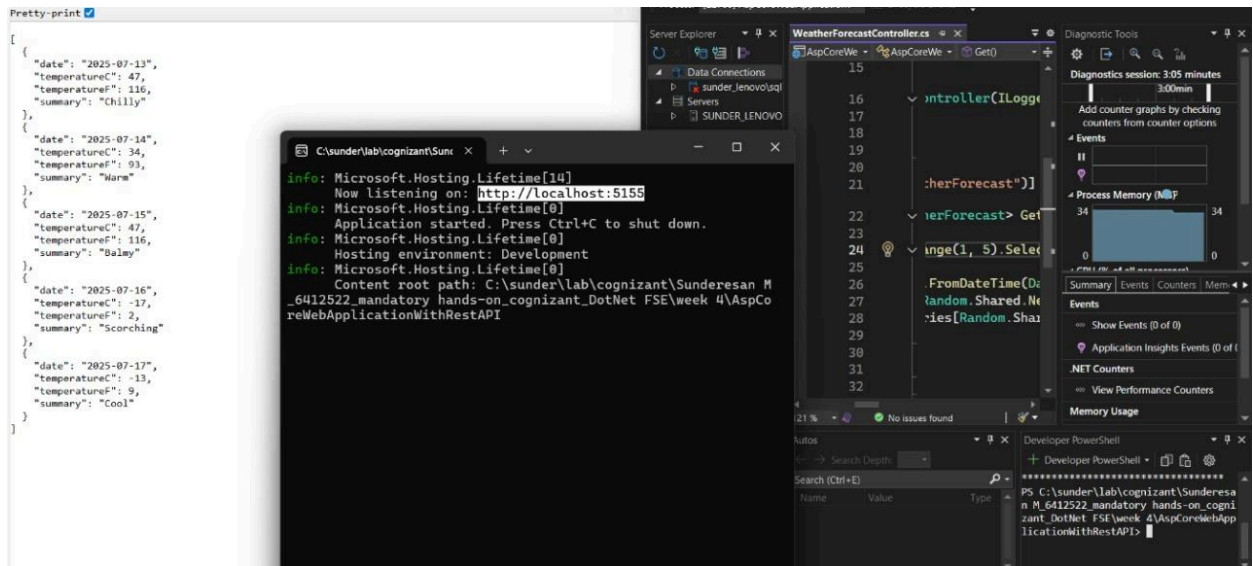
```
using Microsoft.AspNetCore.Mvc;
namespace MyFirstWebAPI.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class ValuesController : ControllerBase
    {
        private static List<string> values = new List<string> { "value1", "value2" };

        [HttpGet]
        public IActionResult Get()
        {
            return Ok(values);
        }

        [HttpPost]
        public IActionResult Post([FromBody] string value)
        {
            values.Add(value);
            return Ok(value);
        }
    }
}
```

OUTPUT

```
[
  "value1",
  "value2",
  "hello"
]
```



2.WebApi_Handson

```
using Microsoft.OpenApi.Models;
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllers();
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();
app.UseSwagger();
app.UseSwaggerUI(c =>
{
    c.SwaggerEndpoint("/swagger/v1/swagger.json", "Swagger Demo");
});
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();

using Microsoft.AspNetCore.Mvc;
namespace SwaggerDemoAPI.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class EmployeeController : ControllerBase
    {
        private static List<string> employees = new List<string> { "John", "Jane" };
    }
}
```

```

[HttpGet]
[ProducesResponseType(StatusCodes.Status200OK)]
public IActionResult Get()
{
    return Ok(employees);
}

[HttpPost]
[ProducesResponseType(StatusCodes.Status200OK)]
public IActionResult Post([FromBody] string name)
{
    employees.Add(name);
    return Ok(name);
}
}
}

```

OUTPUT

Responses

Curly

```
curl -X 'GET' \
  'http://localhost:5155/api/Employee' \
  -H 'accept: text/json' \
  -H 'Authorization: Bearer xyz123'
```

Request URL

```
http://localhost:5155/api/Employee
```

Server response

Code Details

200

Response body

```
[
  {
    "id": 1,
    "name": "Sunderesan",
    "salary": 7000000,
    "permanent": true,
    "dateOfBirth": "2005-02-14T00:00:00"
  },
  {
    "id": 2,
    "name": "Suraj",
    "salary": 6000000,
    "permanent": false,
    "dateOfBirth": "2005-06-03T00:00:00"
  }
]
```

Response headers

```
content-type: text/json; charset=utf-8
date: Sat, 12 Jul 2025 19:09:21 GMT
server: Kestrel
transfer-encoding: chunked
```

Collections

+ Search collections

> AUTH

Environments

Flows

History

http://localhost:5155/api/Employee

Save Share

GET http://localhost:5155/api/Employee Send

Params Authorization Headers (0) Body Scripts Tests Settings Cookies

Query Params

| Key | Value | Description | Bulk Edit |
|-----|-------|-------------|-----------|
| Key | Value | Description | |

Body Cookies Headers (4) Test Results

200 OK 105 ms 343 B

JSON Preview Visualize

```

1 [
2   {
3     "id": 1,
4     "name": "Sunderesan",
5     "salary": 7000000,
6     "permanent": true,
7     "dateOfBirth": "2005-02-14T00:00:00"
8   },
9   {
10    "id": 2,
11    "name": "Suraj",
12    "salary": 6000000,
13    "permanent": false,
14    "dateOfBirth": "2005-06-03T00:00:00"
15  }
16 ]

```

3.WebApi_HandsOn

```
namespace SwaggerDemoAPI.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; }
    }

    public class Department
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }

    public class Skill
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }
}
```

```
using Microsoft.AspNetCore.Mvc;
using SwaggerDemoAPI.Models;
using SwaggerDemoAPI.Filters;
```

```
namespace SwaggerDemoAPI.Controllers
```

```
{
    [ApiController]
    [Route("api/[controller]")]
    [ServiceFilter(typeof(CustomAuthFilter))]
    public class EmployeeController : ControllerBase
    {
        private readonly List<Employee> _employees;

        public EmployeeController()
        {
            _employees = GetStandardEmployeeList();
        }

        [AllowAnonymous]
        [HttpGet]
        [ProducesResponseType(StatusCodes.Status200OK)]
        [ProducesResponseType(StatusCodes.Status500InternalServerError)]
        public ActionResult<List<Employee>> Get()
        {
            throw new Exception("Test exception logging"); // for testing
            return Ok(_employees);
        }

        [HttpPost]
        public IActionResult Post([FromBody] Employee employee)
        {
            return Ok(employee);
        }

        private List<Employee> GetStandardEmployeeList()
        {
            return new List<Employee>
            {
                new Employee
                {
                    Id = 1,
                    Name = "Alice",
                    Salary = 50000,
                    Permanent = true,
                    DateOfBirth = new DateTime(1990, 5, 1),
                }
            }
        }
    }
}
```

```
[HttpPost]
public IActionResult Post([FromBody] Employee employee)
{
    return Ok(employee);
}

using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Filters;
```

```
{
  "id": 2,
  "name": "Bob",
  "salary": 40000,
  "permanent": false,
}
```

```

"department": {
  "id": 2,
  "name": "HR"
},
"skills": [
  { "id": 3, "name": "Excel" },
  { "id": 4, "name": "Communication" }
],
"dateOfBirth": "1995-08-12T00:00:00"
}

```

OUTPUT

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5155/api/Employee/standard' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer xyz123'
```

Request URL

```
http://localhost:5155/api/Employee/standard
```

Server response

Code Details

200

Response body

```
{
  {
    "id": 3,
    "name": "Alice",
    "salary": 80000,
    "permanent": true,
    "dateOfBirth": "1985-04-12T00:00:00"
  },
  {
    "id": 2,
    "name": "Bob",
    "salary": 75000,
    "permanent": false,
    "dateOfBirth": "1990-09-23T00:00:00"
  }
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 12 Jul 2025 19:18:10 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5155/api/Employee' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5155/api/Employee
```

Server response

Code Details

400

Error: Bad Request

Response body

```
Invalid request - No Auth token
```

Response headers

```
content-type: text/plain; charset=utf-8
date: Sun, 13 Jul 2025 07:11:34 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code Description Links

Available authorizations

Bearer (apiKey)

Authorized

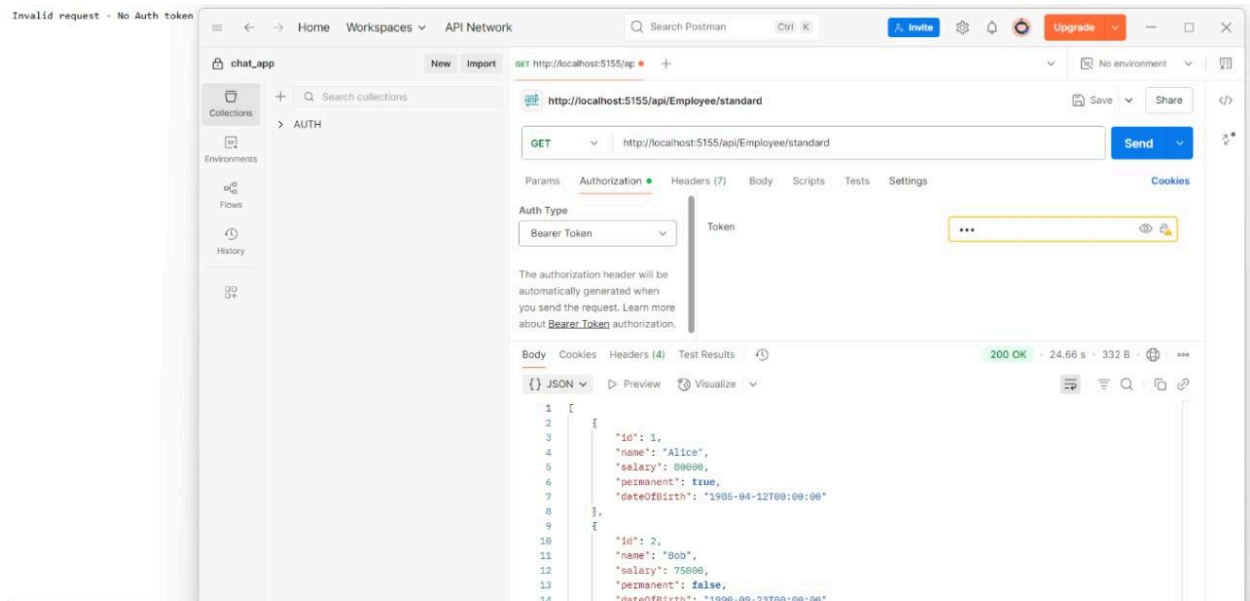
Enter 'Bearer' [space] and then your token.
Example: "Bearer abcdef12345"

Name: Authorization

In header

Value: *****

Logout Close



4.WebApi_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; }
}

```

```

public class Department
{
    public int Id { get; set; }
    public string Name { get; set; }
}

```

```

public class Skill
{
    public int Id { get; set; }
    public string Name { get; set; }
}

```

```

using Microsoft.AspNetCore.Mvc;
using SwaggerDemoAPI.Models;
using System.Linq;

```

```

namespace SwaggerDemoAPI.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class EmployeeController : ControllerBase
    {
        private static List<Employee> _employees = new List<Employee>
        {
            new Employee
            {
                Id = 1,
                Name = "Alice",
                Salary = 50000,
                Permanent = true,
                DateOfBirth = new DateTime(1990, 5, 1),
                Department = new Department { Id = 1, Name = "IT" },
                Skills = new List<Skill>
                {

```

```

        new Skill { Id = 1, Name = "C#" },
        new Skill { Id = 2, Name = "SQL" }
    }
},
new Employee
{
    Id = 2,
    Name = "Bob",
    Salary = 60000,
    Permanent = false,
    DateOfBirth = new DateTime(1985, 3, 10),
    Department = new Department { Id = 2, Name = "HR" },
    Skills = new List<Skill>
    {
        new Skill { Id = 3, Name = "Excel" },
        new Skill { Id = 4, Name = "Communication" }
    }
}
};

[HttpGet]
public ActionResult<List<Employee>> Get()
{
    return Ok(_employees);
}

[HttpPost]
public ActionResult<Employee> Create([FromBody] Employee employee)
{
    employee.Id = _employees.Max(e => e.Id) + 1;
    _employees.Add(employee);
    return Ok(employee);
}

[HttpPut]
public ActionResult<Employee> Update([FromBody] Employee updatedEmployee)
{
    if (updatedEmployee.Id <= 0)
        return BadRequest("Invalid employee id");

    var existingEmployee = _employees.FirstOrDefault(e => e.Id == updatedEmployee.Id);

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

    existingEmployee.Name = updatedEmployee.Name;
    existingEmployee.Salary = updatedEmployee.Salary;
    existingEmployee.Permanent = updatedEmployee.Permanent;
    existingEmployee.DateOfBirth = updatedEmployee.DateOfBirth;
    existingEmployee.Department = updatedEmployee.Department;
    existingEmployee.Skills = updatedEmployee.Skills;

    return Ok(existingEmployee);
}

[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    var employee = _employees.FirstOrDefault(e => e.Id == id);

    if (employee == null)
        return NotFound("Employee not found");

    _employees.Remove(employee);
    return Ok("Employee deleted");
}
}
}

```

OUTPUT

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:5155/api/Employee/1' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer xyz123' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 1,
    "name": "Sunderesan M",
    "salary": 7100000,
    "permanent": true,
    "dateOfBirth": "2005-02-14T19:19:46.488Z"
  }'
```

Request URL

http://localhost:5155/api/Employee/1

Server response

| Code | Details |
|------|--|
| 200 | <p>Response body</p> <pre>{ "id": 1, "name": "Sunderesan M", "salary": 7100000, "permanent": true, "dateOfBirth": "2005-02-14T19:19:46.488Z" }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Sat, 12 Jul 2025 19:20:44 GMT server: Kestrel transfer-encoding: chunked</pre> |

Responses

5.WebApi_Handson

```
string securityKey = "mysuperdupersecret";
var symmetricSecurityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(securityKey));
```

```
builder.Services.AddAuthentication(x =>
{
    x.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
    x.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
    x.DefaultSignInScheme = JwtBearerDefaults.AuthenticationScheme;
})
.AddJwtBearer(x =>
{
    x.TokenValidationParameters = new TokenValidationParameters
    {
        ValidateIssuer = true,
        ValidateAudience = true,
        ValidateLifetime = true,
        ValidateIssuerSigningKey = true,
        ValidIssuer = "mySystem",
        ValidAudience = "myUsers",
        IssuerSigningKey = symmetricSecurityKey
    };
});
```

```
builder.Services.AddAuthorization();
builder.Services.AddCors(options =>
{
    options.AddPolicy("AllowAll",
        builder =>
        {
            builder.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod();
        });
});
app.UseCors("AllowAll");
app.UseAuthentication();
app.UseAuthorization();
using Microsoft.AspNetCore.Mvc;
using Microsoft.IdentityModel.Tokens;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;
```

```
[Route("api/[controller]")]
[ApiController]
[AllowAnonymous]
```

```

public class AuthController : ControllerBase
{
    private string GenerateJSONWebToken(int userId, string userRole)
    {
        var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("mysuperdupersecret"));
        var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);

        var claims = new List<Claim>
        {
            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);
    }

    [HttpGet]
    public IActionResult GetToken()
    {
        var token = GenerateJSONWebToken(101, "Admin");
        return Ok(token);
    }
}

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using SwaggerDemoAPI.Models;

[ApiController]
[Route("api/[controller]")]
[Authorize(Roles = "Admin,POC")]
public class EmployeeController : ControllerBase
{
    private static List<Employee> _employees = new List<Employee>
    {
        new Employee { Id = 1, Name = "Alice", Salary = 50000, Permanent = true, DateOfBirth = DateTime.Now }
    };

    [HttpGet]
    public ActionResult<List<Employee>> Get()
    {
        return Ok(_employees);
    }
}

Key: Authorization
Value: Bearer <paste_token_here>
services.AddSwaggerGen(c =>
{
    c.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme
    {
        In = ParameterLocation.Header,
        Description = "Please enter token",
        Name = "Authorization",
        Type = SecuritySchemeType.Http,
        BearerFormat = "JWT",
        Scheme = "bearer"
    });

    c.AddSecurityRequirement(new OpenApiSecurityRequirement
    {
        {
            new OpenApiSecurityScheme
            {
                Reference = new OpenApiReference { Type = ReferenceType.SecurityScheme, Id = "Bearer" }
            },
            Array.Empty<string>()
        }
    });
});

```

});

OUTPUT

GET

http://localhost:5155/api/auth/token?userId=123&role=Admin

Send

Params Authorization Headers (6) Body Scripts Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

1 Ctrl+Alt+P for Postbot

Body Cookies Headers (4) Test Results

200 OK 1.07 s 406 B

Raw Preview Visualize

1 eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJodHRwOi8vc2NoZW1hcy5taWY3NzZnQyY29tL3d2LzIwMDgvMDYvaWR1bnRpdHkvY2xhaw1zL3JvbGU1OjBZG1pb1IsIlVzZXJJZCI6IjEyMyIsImV4cCI6MTc1MjM5NjY
3NiWiaXNzIjoibXlTeXN0ZW0iLCJhdwQ1O1JteVZzZXJzIn0.30Z_A0o9zuh4II3jj3fD6F-SSUSn0smTbK0D17d5Vc

GET

http://localhost:5155/api/Employee/standard

Send

Params Authorization Headers (7) Body Scripts Tests Settings

Auth Type

Bearer Token

Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body Cookies Headers (4) Test Results

200 OK 286 ms 332 B

{ } JSON Preview Visualize

1 [
2
3 {
4 "id": 1,
5 "name": "Alice",
6 "salary": 80000,
7 "permanent": true,
8 "dateOfBirth": "1985-04-12T00:00:00"
9 },
10 {
11 "id": 2,
12 "name": "Bob",
13 "salary": 75000,
14 "permanent": false,
15 "dateOfBirth": "1990-09-23T00:00:00"
16 }
17]

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body Cookies Headers (4) Test Results

200 OK 34 ms 403 B

Raw Preview Visualize

1 eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJodHRwOi8vc2NoZW1hcy5taWY3NzZnQyY29tL3d2LzIwMDgvMDYvaWR1bnRpdHkvY2xhaw1zL3JvbGU1OjBZG1pb1IsIlVzZXJJZCI6IjEyMyIsImV4cCI6MTc1MjM5NjY3NiWiaXNzIjoibXlTeXN0ZW0iLCJhdwQ1O1JteVZzZXJzIn0.30Z_A0o9zuh4II3jj3fD6F-SSUSn0smTbK0D17d5Vc

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Auth Type

JWT Bearer

The authorization header will be automatically generated when you send the request. Learn more about [JWT Bearer](#) authorization.

Algorithm ⓘ

HS256

Secret ⓘ

.....

☐ Secret Base64 encoded

Payload ⓘ

{}

401 Unauthorized

48 ms

203 B

Body

Cookies

Headers (4)

Test Results

Raw

Preview

Visualize

1

Cookies