

LAB-1

```
Program.cs
1 var builder = WebApplication.CreateBuilder(args);
2
3
4 builder.Services.AddOpenApi();
5
6 var app = builder.Build();
7
8 if (app.Environment.IsDevelopment())
9 {
10     app.MapOpenApi();
11 }
12
13 app.UseHttpsRedirection();
14
15 var summaries = new[]
16 {
17     "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering", "Scorching"
18 };
19
20 app.MapGet("/weatherforecast", () =>
21 {
22     var forecast = Enumerable.Range(1, 5).Select(index =>
23         new WeatherForecast
24         {
25             DateOnly.FromDateTime(DateTime.Now.AddDays(index)),
26             Random.Shared.Next(-20, 55),
27             summaries[Random.Shared.Next(summaries.Length)]
28         })
29         .ToArray();
30     return forecast;
31 })
32 .WithName("GetWeatherForecast");
33
34 app.Run();
35
36 1 reference
37 record WeatherForecast(DateOnly Date, int TemperatureC, string? Summary)
38 {
39     0 references
40     public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);
41 }
```

```
Controllers > ValuesController.cs
1 using Microsoft.AspNetCore.Mvc;
2
3 namespace MyFirstWebAPI.Controllers
4 {
5     [ApiController]
6     [Route("[controller]")]
7     0 references
8     public class ValuesController : ControllerBase
9     {
10         [HttpGet]
11         0 references
12         public IActionResult Get()
13         {
14             return Ok(new string[] { "value1", "value2" });
15         }
16
17         [HttpGet("{id}")]
18         0 references
19         public IActionResult Get(int id)
20         {
21             return Ok("value " + id);
22         }
23
24         [HttpPost]
25         0 references
26         public IActionResult Post([FromBody] string value)
27         {
28             return Ok("Posted: " + value);
29         }
30
31         [HttpPut("{id}")]
32         0 references
33         public IActionResult Put(int id, [FromBody] string value)
34         {
35             return Ok($"Updated id {id} with value '{value}'");
36         }
37
38         [HttpDelete("{id}")]
39         0 references
40         public IActionResult Delete(int id)
41         {
42             return Ok($"Deleted value with id {id}");
43         }
44     }
45 }
```

```
MyFirstWebAPI.csproj
1 <Project Sdk="Microsoft.NET.Sdk.Web">
2
3     <PropertyGroup>
4         <TargetFramework>net9.0</TargetFramework>
5         <Nullable>enable</Nullable>
6         <ImplicitUsings>enable</ImplicitUsings>
7     </PropertyGroup>
8
9     <ItemGroup>
10         <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="9.0.6" />
11     </ItemGroup>
12
13 </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
1  using Microsoft.AspNetCore.Mvc;
2  using System.Collections.Generic;
3
4  namespace MyFirstWebAPI.Controllers
5  {
6      [ApiController]
7      [Route("[controller]")] // maps to /Employee
8      public class EmployeeController : ControllerBase
9      {
10         [HttpGet]
11         public IEnumerable<string> Get()
12         {
13             return new string[] { "John", "Doe", "Smith" };
14         }
15
16         [HttpGet("{id}")]
17         public string Get(int id)
18         {
19             var employees = new string[] { "John", "Doe", "Smith" };
20             if (id >= 0 && id < employees.Length)
21                 return employees[id];
22             return "Employee not found";
23         }
24     }
25 }
26
```

```
Program.cs
1  using Microsoft.OpenApi.Models;
2
3  var builder = WebApplication.CreateBuilder(args);
4
5  builder.Services.AddControllers();
6  builder.Services.AddEndpointsApiExplorer();
7  builder.Services.AddSwaggerGen(c =>
8  {
9      c.SwaggerDoc("v1", new OpenApiInfo
10         {
11             Title = "Swagger Demo",
12             Version = "v1",
13             Description = "TBD",
14             TermsOfService = new Uri("https://example.com/terms"),
15             Contact = new OpenApiContact
16             {
17                 Name = "John Doe",
18                 Email = "john@xyzmail.com",
19                 Url = new Uri("https://www.example.com")
20             },
21             License = new OpenApiLicense
22             {
23                 Name = "License Terms",
24                 Url = new Uri("https://www.example.com")
25             }
26         });
27 });
28
29 var app = builder.Build();
30
31 if (app.Environment.IsDevelopment())
32 {
33     app.UseSwagger();
34     app.UseSwaggerUI();
35 }
36
37 app.UseHttpsRedirection();
38 app.UseAuthorization();
39 app.MapControllers();
40
41 app.Run();
```

```


MyFirstWebAPI.esproj
1 <Project Sdk="Microsoft.NET.Sdk.Web">
2
3   <PropertyGroup>
4     <TargetFramework>net9.0</TargetFramework>
5     <Nullable>enable</Nullable>
6     <ImplicitUsings>enable</ImplicitUsings>
7   </PropertyGroup>
8
9   <ItemGroup>
10    <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="9.0.6" />
11    <PackageReference Include="Swashbuckle.AspNetCore" Version="9.0.3" />
12  </ItemGroup>
13
14 </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.

```


Swagger.
Supported by SMARTBEAR

Select a definition
MyFirstWebAPI v1

Swagger Demo v1 OAS 3.0

http://localhost:5224/swagger/v1/swagger.json

TBD

[Terms of service](#)
[John Doe - Website](#)
[Send email to John Doe](#)
[License Terms](#)

Employee

GET
/Employee

Parameters

Cancel

No parameters

Execute

Clear

GET

/Employee

Parameters

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

New Section 1 Page 3

LAB-3

```
Controllers > EmployeeController.cs > EmployeeController
1  using Microsoft.AspNetCore.Authorization;
2  using Microsoft.AspNetCore.Mvc;
3  using System.Collections.Generic;
4  using System;
5  using EmployeeApi.Models;
6  using EmployeeApi.Filters;
7
8  namespace EmployeeApi.Controllers
9  {
10     [ApiController]
11     [Route("[controller]")]
12     [ServiceFilter(typeof(CustomAuthFilter))]
13
14     0 references
15     public class EmployeeController : ControllerBase
16     {
17         1 reference
18         private List<Employee> GetStandardEmployeeList()
19         {
20             return new List<Employee>
21             {
22                 new Employee
23                 {
24                     Id = 1,
25                     Name = "John",
26                     Salary = 50000,
27                     Permanent = true,
28                     DateOfBirth = new DateTime(1990, 5, 1),
29                     Department = new Department { Id = 1, Name = "HR" },
30                     Skills = new List<Skill>
31                     {
32                         new Skill { Id = 1, SkillName = "C#" },
33                         new Skill { Id = 2, SkillName = "ASP.NET" }
34                     }
35                 }
36             };
37         }
38
39         [HttpGet]
40         [ProducesResponseType(200)]
41         [ProducesResponseType(500)]
42         0 references
43         public ActionResult<List<Employee>> GetStandard()
44         {
45             return Ok(GetStandardEmployeeList());
46         }
47
48         [HttpPost]
49         [AllowAnonymous]
50         0 references
51         public ActionResult<Employee> PostEmployee([FromBody] Employee emp)
52         {
53             return Ok(emp);
54         }
55     }
56 }
```

```

Program.cs > Program > <top-level-statements-entry-point>
1 using EmployeeApi.Filters;
2 using Microsoft.OpenApi.Models;
3
4 var builder = WebApplication.CreateBuilder(args);
5
6 builder.Services.AddControllers(options =>
7 {
8     options.Filters.Add<CustomExceptionFilter>();
9 });
10 builder.Services.AddScoped<CustomAuthFilter>();
11
12
13 builder.Services.AddEndpointsApiExplorer();
14 builder.Services.AddSwaggerGen(c =>
15 {
16     c.SwaggerDoc("v1", new OpenApiInfo { Title = "EmployeeApi", Version = "v1" });
17
18     c.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme
19     {
20         In = ParameterLocation.Header,
21         Description = "Enter 'Bearer' followed by your token",
22         Name = "Authorization",
23         Type = SecuritySchemeType.ApiKey
24     });
25
26     c.AddSecurityRequirement(new OpenApiSecurityRequirement
27     {
28         {
29             new OpenApiSecurityScheme
30             {
31                 Reference = new OpenApiReference
32                 {
33                     Type = ReferenceType.SecurityScheme,
34                     Id = "Bearer"
35                 }
36             },
37             new string[] { }
38         }
39     });
40 });
41
42 var app = builder.Build();
43
44 if (app.Environment.IsDevelopment())
45 {
46     app.UseSwagger();
47     app.UseSwaggerUI();
48 }
49
50 app.UseHttpsRedirection();
51 app.UseAuthorization();
52 app.MapControllers();
53 app.Run();
54

```

```

Models > Employee.cs > ...
1 using System;
2 using System.Collections.Generic;
3
4 namespace EmployeeApi.Models
5 {
6     6 references
7     public class Employee
8     {
9         1 reference
10         public int Id { get; set; }
11         1 reference
12         public string Name { get; set; }
13         1 reference
14         public int Salary { get; set; }
15         1 reference
16         public bool Permanent { get; set; }
17         1 reference
18         public Department Department { get; set; }
19         1 reference
20         public List<Skill> Skills { get; set; }
21         1 reference
22         public DateTime DateOfBirth { get; set; }
23     }
24
25     2 references
26     public class Department
27     {
28         1 reference
29         public int Id { get; set; }
30         1 reference
31         public string Name { get; set; }
32     }
33
34     4 references
35     public class Skill
36     {
37         2 references
38         public int Id { get; set; }
39         2 references
40         public string SkillName { get; set; }
41     }
42 }

```

```

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

```

GET /Employee

Parameters

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  http://localhost:5206/Employee/ \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer testtoken'
```

Request URL

```
http://localhost:5206/Employee
```

Server response

Code

Details

200

Response body

```
{
  "id": 1,
  "name": "John",
  "salary": 50000,
  "permanent": true,
  "department": {
    "id": 1,
    "name": "HR"
  },
  "skills": [
    {
      "id": 1,
      "skillname": "C#"
    },
    {
      "id": 2,
      "skillname": "ASP.NET"
    }
  ],
  "dateOfBirth": "1998-05-01T00:00:00"
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 12 Jul 2025 09:03:28 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code

Description

Links

200

OK

No links

Media type

text/plain

Content Accept header

Example Value | Schema

```
{
  "id": 0,
  "name": "string",
  "salary": 0,
  "permanent": true,
  "department": {
    "id": 0,
    "name": "string"
  },
  "skills": [
    {
      "id": 0,
      "skillname": "string"
    }
  ],
  "dateOfBirth": "2025-07-12T00:00:00.000Z"
}
```

500

Internal Server Error

No links

POST /Employee

New Section 1 Page 3

LAB-4

```
Program.cs > Program > <top-level-statements-entry-point>
1 using Microsoft.AspNetCore.Builder;
2 using Microsoft.Extensions.DependencyInjection;
3 using Microsoft.Extensions.Hosting;
4 using Microsoft.OpenApi.Models;
5
6
7 var builder = WebApplication.CreateBuilder(args);
8
9 builder.Services.AddControllers();
10 builder.Services.AddEndpointsApiExplorer();
11 builder.Services.AddSwaggerGen();
12
13 var app = builder.Build();
14
15 if (app.Environment.IsDevelopment())
16 {
17     app.UseSwagger();
18     app.UseSwaggerUI();
19 }
20
21 app.UseHttpsRedirection();
22
23 app.UseAuthorization();
24
25 app.MapControllers();
26
27 app.Run();
28
```

```
Controllers > EmployeeController.cs > EmployeeController
1 using Microsoft.AspNetCore.Mvc;
2 using EmployeeApi.Models;
3 using System.Collections.Generic;
4 using System.Linq;
5
6 namespace EmployeeApi.Controllers
7 {
8     [ApiController]
9     [Route("api/[controller]")]
10     public class EmployeeController : ControllerBase
11     {
12         private static List<Employee> employees = new List<Employee>
13         {
14             new Employee { Id = 1, Name = "John", Department = "HR" },
15             new Employee { Id = 2, Name = "Alice", Department = "IT" },
16             new Employee { Id = 3, Name = "Bob", Department = "Finance" }
17         };
18
19         [HttpPut("{id}")]
20         public ActionResult<Employee> UpdateEmployee(int id, [FromBody] Employee updatedEmp)
21         {
22             if (id <= 0)
23             {
24                 return BadRequest("Invalid employee id");
25             }
26
27             var existingEmployee = employees.FirstOrDefault(e => e.Id == id);
28
29             if (existingEmployee == null)
30             {
31                 return BadRequest("Invalid employee id");
32             }
33
34             existingEmployee.Name = updatedEmp.Name;
35             existingEmployee.Department = updatedEmp.Department;
36
37             return Ok(existingEmployee);
38         }
39     }
40 }
41
```

```
Models > Employee.cs > ...
1 namespace EmployeeApi.Models
2 {
3     7 references
4     public class Employee
5     {
6         4 references
7         public int Id { get; set; }
8         5 references
9         public string? Name { get; set; }
10        5 references
11        public string? Department { get; set; }
12    }
13 }
14
```

```
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 | <div>OK</div> <div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": 0, "name": "string", "department": "string" }</pre></div> | No links |

LAB-5

```
Controllers > AuthController.cs > AuthController > GenerateJSONWebToken
1  using Microsoft.AspNetCore.Mvc;
2  using Microsoft.AspNetCore.Authorization;
3  using Microsoft.IdentityModel.Tokens;
4  using System.IdentityModel.Tokens.Jwt;
5  using System.Security.Claims;
6  using System.Text;
7
8  namespace JwtAuthDemo.Controllers
9  {
10     [ApiController]
11     [Route("api/[controller]")]
12     public class AuthController : ControllerBase
13     {
14         [HttpGet("token")]
15         [AllowAnonymous]
16         public IActionResult GetToken()
17         {
18             var token = GenerateJSONWebToken(1, "Admin");
19             return Ok(new { token });
20         }
21
22         private string GenerateJSONWebToken(int userId, string userRole)
23         {
24             var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("mysuperdupersecurekeythatishard2guess!"));
25             var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);
26
27             var claims = new[]
28             {
29                 new Claim(ClaimTypes.Role, userRole),
30                 new Claim("UserId", userId.ToString())
31             };
32
33             var token = new JwtSecurityToken(
34                 issuer: "mySystem",
35                 audience: "myUsers",
36                 claims: claims,
37                 expires: DateTime.Now.AddMinutes(2),
38                 signingCredentials: credentials);
39
40             return new JwtSecurityTokenHandler().WriteToken(token);
41         }
42     }
43 }
```

```
Controllers > EmployeeController.cs > EmployeeController
1  using Microsoft.AspNetCore.Authorization;
2  using Microsoft.AspNetCore.Mvc;
3
4  namespace JwtAuthDemo.Controllers
5  {
6     [ApiController]
7     [Route("api/[controller]")]
8     [Authorize(Roles = "Admin,POC")]
9     public class EmployeeController : ControllerBase
10     {
11         [HttpGet]
12         public IActionResult Get()
13         {
14             return Ok(" You are authorized to access employee data!");
15         }
16     }
17 }
```

```

Program.cs > Program > <top-level-statements-entry-point>
1 using Microsoft.AspNetCore.Authentication.JwtBearer;
2 using Microsoft.IdentityModel.Tokens;
3 using System.Text;
4
5 var builder = WebApplication.CreateBuilder(args);
6
7 var securityKey = "mysuperdupersecurekeythatishard2guess!";
8 var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(securityKey));
9
10 builder.Services.AddControllers();
11 builder.Services.AddCors(options =>
12 {
13     options.AddPolicy("AllowAll", policy =>
14     {
15         policy.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader();
16     });
17 });
18 builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
19     .AddJwtBearer(options =>
20     {
21         options.TokenValidationParameters = new TokenValidationParameters
22         {
23             ValidateIssuer = true,
24             ValidateAudience = true,
25             ValidateLifetime = true,
26             ValidateIssuerSigningKey = true,
27             ValidIssuer = "mySystem",
28             ValidAudience = "myUsers",
29             IssuerSigningKey = key
30         };
31     });
32
33 builder.Services.AddAuthorization();
34
35 var app = builder.Build();
36 app.UseDeveloperExceptionPage();
37
38
39 app.UseCors("AllowAll");
40 app.UseAuthentication();
41 app.UseAuthorization();
42
43 app.MapControllers();
44 app.Run();
45

```

```

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

```

The screenshot shows the Postman interface with a GET request to `http://localhost:5154/api/Auth/token`. The response is a 200 OK status with a response time of 6 ms and a body size of 421 B. The response body is a JSON object with a single key-value pair: `"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy5taWV3bnZnZm1lZ3dLZlIwMDgvMDYvaWR1bnRpdHkvY2xhaW1zL3JvbGUiOiJBZG1pbilzIlVzZXJJZCI6IjEiLCJleHAiOjE3NTIzODE3MzgsIm1zcyI6Im15U3lzdGVtIiwiaXVkiOiB1VjVycyJ9.HqADhWfQTFwZDBgGKmo0M8UmJEzIQxW2_j0Iovx05a4"`.