

Project Structure

```
PS C:\Users\sanja\source\CSHarp\OrderMgmtAPI\OrderMgmtAPI> tree /f
Folder PATH listing for volume Windows-SSD
Volume serial number is 3A71-698F
C:.
├── appsettings.Development.json
├── appsettings.json
├── OrderMgmtAPI.csproj
├── OrderMgmtAPI.csproj.Backup.tmp
├── OrderMgmtAPI.csproj.user
├── OrderMgmtAPI.http
└── Program.cs

bin
└── Debug
    └── net8.0
        ├── appsettings.Development.json
        ├── appsettings.json
        ├── Azure.Core.dll
        ├── Azure.Identity.dll
        ├── Humanizer.dll
        ├── Microsoft.AspNetCore.Authentication.JwtBearer.dll
        ├── Microsoft.AspNetCore.Cryptography.Internal.dll
        ├── Microsoft.AspNetCore.Cryptography.KeyDerivation.dll
        ├── Microsoft.AspNetCore.Identity.EntityFrameworkCore.dll
        ├── Microsoft.Bcl.AsyncInterfaces.dll
        ├── Microsoft.CodeAnalysis.CSharp.dll
        ├── Microsoft.CodeAnalysis.CSharp.Workspaces.dll
        ├── Microsoft.CodeAnalysis.dll
        ├── Microsoft.CodeAnalysis.Workspaces.dll
        ├── Microsoft.Data.SqlClient.dll
        ├── Microsoft.EntityFrameworkCore.Abstractions.dll
        ├── Microsoft.EntityFrameworkCore.Design.dll
        ├── Microsoft.EntityFrameworkCore.dll
        ├── Microsoft.EntityFrameworkCore.Relational.dll
        ├── Microsoft.EntityFrameworkCore.SqlServer.dll
        ├── Microsoft.Extensions.Caching.Memory.dll
        ├── Microsoft.Extensions.DependencyInjection.Abstractions.dll
        ├── Microsoft.Extensions.DependencyInjection.dll
        ├── Microsoft.Extensions.DependencyModel.dll
        ├── Microsoft.Extensions.Identity.Core.dll
        ├── Microsoft.Extensions.Identity.Stores.dll
        ├── Microsoft.Extensions.Logging.Abstractions.dll
        ├── Microsoft.Extensions.Logging.dll
        └── Microsoft.Extensions.Options.dll
```



```
.\Debug  
└── net8.0  
    ├── .NETCoreApp, Version=v8.0.AssemblyAttributes.cs  
    ├── ApiEndpoints.json  
    ├── apphost.exe  
    ├── OrderMgmt.96586139.Up2Date  
    ├── OrderMgmtAPI.AssemblyInfo.cs  
    ├── OrderMgmtAPI.AssemblyInfoInputs.cache  
    ├── OrderMgmtAPI.assets.cache  
    ├── OrderMgmtAPI.csproj.AssemblyReference.cache  
    ├── OrderMgmtAPI.csproj.BuildWithSkipAnalyzers  
    ├── OrderMgmtAPI.csproj.CoreCompileInputs.cache  
    ├── OrderMgmtAPI.csproj.FileListAbsolute.txt  
    ├── OrderMgmtAPI.dll  
    ├── OrderMgmtAPI.GeneratedMSBuildEditorConfig.editorconfig  
    ├── OrderMgmtAPI.genruntimeconfig.cache  
    ├── OrderMgmtAPI.GlobalUsings.g.cs  
    ├── OrderMgmtAPI.MvcApplicationPartsAssemblyInfo.cache  
    ├── OrderMgmtAPI.MvcApplicationPartsAssemblyInfo.cs  
    ├── OrderMgmtAPI.pdb  
    ├── rjsmcshmtl.dsوا.cache.json  
    ├── rjsmraor.dsوا.cache.json  
    ├── rspwa.dsوا.cache.json  
    ├── staticwebassets.build.endpoints.json  
    ├── staticwebassets.build.json  
    ├── staticwebassets.build.json.cache  
    ├── staticwebassets.references.upToDateCheck.txt  
    └── staticwebassets.removed.txt  
  
    └── EndpointInfo  
        ├── OrderMgmtAPI.json  
        └── OrderMgmtAPI.OpenApiFiles.cache  
  
    └── ref  
        └── OrderMgmtAPI.dll  
  
    └── refint  
        └── OrderMgmtAPI.dll  
  
    └── staticwebassets  
  
Properties  
launchSettings.json
```

Postman Results

Register :

The screenshot shows the Postman interface for the 'Register' API. The request method is 'POST' to 'https://localhost:7030/api/Auth/register'. The 'Body' tab is selected, showing a JSON payload:

```
1 {  
2   "email": "test@example.com",  
3   "password": "Test@123"  
4 }
```

The response status is '200 OK' with a duration of 4.20 s and a size of 170 B. The response body is:

```
1 User Registered Successfully
```

Login :

The screenshot shows the Postman interface for the 'Login' API. The request method is 'POST' to 'https://localhost:7030/api/Auth/login'. The 'Body' tab is selected, showing a JSON payload:

```
1 {  
2   "email": "test@example.com",  
3   "password": "Test@123"  
4 }
```

The response status is '200 OK' with a duration of 626 ms and a size of 484 B. The response body is a JSON object containing a token:

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
3     eyJodHRwOi8vczNoZWlhcy54bWxz2FwLn9yZy93cy8yMDA1LzA1L21kZW50aXR5L2NsYw  
4     ltcy9uYW1lIjoiGVzdEB1eGftcGx1LmNvbSISImp0aSI6IjZ1NWQ5ZmYzLTA1ZDQtNDRj  
Mi04MGE3LWQwZnh0GEwZdg5YSIsImVaccC16MTc2NDU0MDY1NSwiZXIjoiT3JkZXJBcG  
kiLCJhdWQiOiJPcmRlcFwaVvzZXJzIn0.  
hf0FB00gAJ8U_st6fetteXZqqTJY0IngmR12D7nng"
```

Unauthorized Access of Protected Endpoint :

The screenshot shows a Postman request configuration. The method is set to POST, and the URL is https://localhost:7030/api/Orders. The Headers tab shows 9 items, and the Body tab is selected, containing the following JSON:

```
1 {  
2   "productName": "Tablet",  
3   "quantity": 10,  
4   "unitPrice": 12000  
5 }
```

After sending the request, the response status is 401 Unauthorized, with a response body of "1".

Setting up the bearer token :

The screenshot shows a Postman request configuration for the same endpoint. The Headers tab is selected and contains the following items:

Key	Value
Authorization	Bearer eyJhbGciOiJIUzI1NiIs...
Cache-Control	no-cache
Postman-Token	<calculated when request is s...
Content-Type	application/json
Content-Length	<calculated when request is s...
Host	<calculated when request is s...
User-Agent	PostmanRuntime/7.49.1
Accept	*/*

The response status is 200 OK.

Order Placed :

The screenshot shows a Postman request for a POST method to the URL `https://localhost:7030/api/Orders`. The request body is a JSON object:

```
1 {  
2   "productName": "tablet",  
3   "quantity": 10,  
4   "unitPrice": 12000  
5 }
```

The response status is 200 OK, with a response time of 437 ms and a response size of 199 B. The response body is:

```
1 {  
2   "message": "Order placed successfully",  
3   "orderId": 1  
4 }
```

Get Order by ID :

The screenshot shows a Postman request for a GET method to the URL `https://localhost:7030/api/Orders/1`. The response status is 200 OK, with a response time of 85 ms and a response size of 310 B. The response body is:

```
1 {  
2   "id": 1,  
3   "productName": "tablet",  
4   "quantity": 10,  
5   "unitPrice": 12000.00,  
6   "totalAmount": 120000.00,  
7   "userName": "test@example.com",  
8   "createdAt": "2025-11-30T20:17:34.700Z"  
9 }
```

Edit Order :

The screenshot shows the Postman interface for editing an order. The URL is `https://localhost:7030/api/Orders/1`. The request method is `PUT`. The body contains the following JSON:

```
1 {  
2   "productName": "laptop",  
3   "quantity": 10,  
4   "unitPrice": 32000  
5 }
```

The response status is `200 OK` with a response time of 166 ms and a response size of 340 B. The response body is:

```
1 {  
2   "message": "Order updated",  
3   "order": {  
4     "id": 1,  
5     "productName": "laptop",  
6     "quantity": 10,  
7     "unitPrice": 32000,  
8     "totalAmount": 320000,  
9     "userName": "test@example.com",  
10    "createdAt": "2025-11-30T20:17:34.700Z"  
11  }  
12 }
```

Delete Order :

The screenshot shows the Postman interface for deleting an order. The URL is `https://localhost:7030/api/Orders/1`. The request method is `DELETE`. There are no query parameters.

The response status is `200 OK` with a response time of 62 ms and a response size of 175 B. The response body is:

```
1 {  
2   "message": "Order deleted"  
3 }
```

Get All Order :

The screenshot shows the Postman application interface. At the top, it displays the URL `https://localhost:7030/api/Orders`. Below the URL, there are tabs for `Body`, `Params`, `Auth`, `Headers (10)`, `Scripts`, `Tests`, and `Settings`. The `Body` tab is selected. Under the `Body` tab, there are two dropdown menus: `raw` (selected) and `Text`. The response body is displayed as JSON:

```
1 {  
2   "productName": "tablet",  
3   "quantity": 10,  
4   "unitPrice": 12000  
5 }  
6  
7 [  
8   {  
9     "id": 2,  
10    "productName": "tablet",  
11    "quantity": 10,  
12    "unitPrice": 12000.00,  
13    "totalAmount": 120000.00,  
14    "userName": "test@example.com",  
15    "createdAt": "2025-11-30T21:09:48.6188075"  
16  },  
17  {  
18    "id": 3,  
19    "productName": "Telivisions",  
20    "quantity": 20,  
21    "unitPrice": 62000.00,  
22    "totalAmount": 1240000.00,  
23    "userName": "test@example.com",  
24    "createdAt": "2025-11-30T21:10:32.0289081"  
25  },  
26  {  
27    "id": 4,  
28    "productName": "Phones",  
29  }  
30 ]
```

The response status is `200 OK`, with a response time of `33 ms` and a size of `644 B`. There is also a `Save Response` button.