

Contents

Middleware	2
1. Requirements	2
2. Data persistence	2
Database Setup	2
3. Email Setting	3
4. Start Application	3
Front-end	4
5. Requirement	4
6. Environment REST API url	4
7. Start Application	5

Middleware

REST API is created using ASP.NET core Web Api using .net 6 framework and using MS SQL Server database.

1. Requirements

- .Net 6
Latest version of .NET 6 SDK and ASP.NET Core Runtime can be downloaded from <https://dotnet.microsoft.com/en-us/download/dotnet/6.0>
- Microsoft SQL Server
SQL Server database is required for web api be functional.

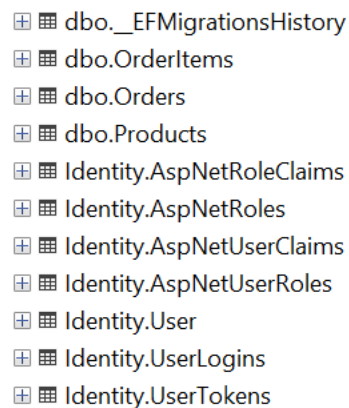
2. Data persistence

Application use MS SQL relational database.

Database Setup

REST API is using EF Core 6 code first which will create a database and its Tables.

Tables

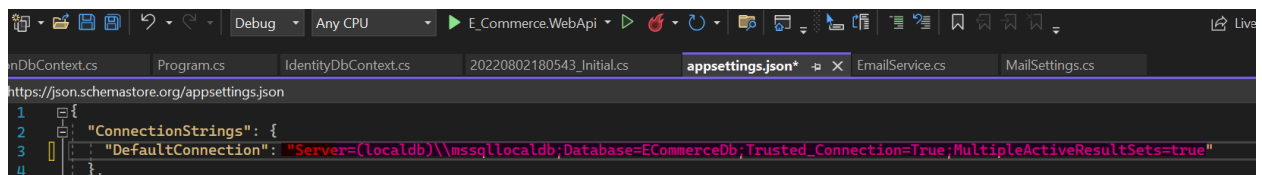


- dbo._EFMigrationsHistory
- dbo.OrderItems
- dbo.Orders
- dbo.Products
- Identity.AspNetRoleClaims
- Identity.AspNetRoles
- Identity.AspNetUserClaims
- Identity.AspNetUserRoles
- Identity.User
- Identity.UserLogins
- Identity.UserTokens

Product Data.

Please run **product data.sql** file to populate product data

Please update connection string in **appsettings.json** file located in **E_Commerce.WebApi** project with correct values of your MS SQL Server



```
1 {
2   "ConnectionStrings": {
3     "DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=ECommerceDb;Trusted_Connection=True;MultipleActiveResultSets=true"
4   }
5 }
```

3. Email Setting

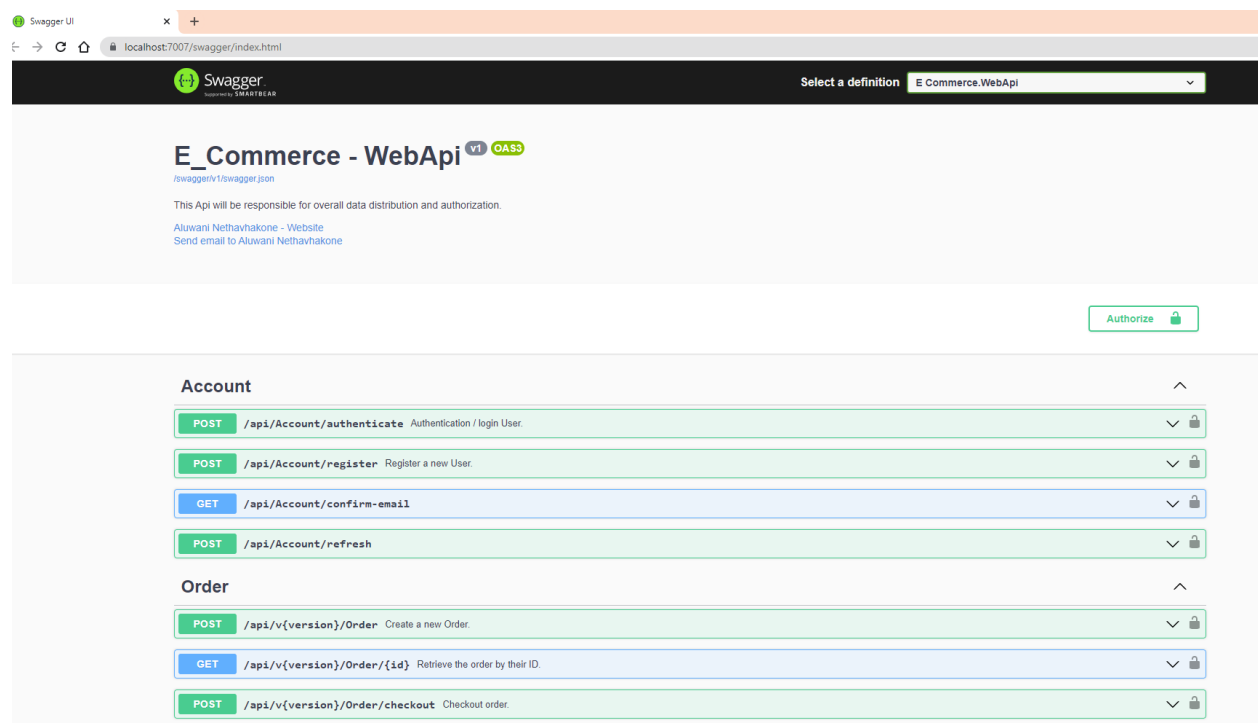
REST API require email setting so it can able to send email to user, when user registered email confirmation with be send to active an account. When user do a checkout, an email will be sent to current login email address with order details.

Please update email settings in **appsettings.json** json file located in **E_Commerce.WebApi** project as below screen

```
{
  "MailSettings": {
    "EmailFrom": "contactus@kodeca.co.za",
    "SmtpHost": "mail.kodeca.co.za",
    "SmtpPort": 587,
    "SmtpUsername": "contactus@kodeca.co.za",
    "SmtpPassword": "*****",
    "DisplayName": "Aluwani Nethavhakone"
  }
}
```

4. Start Application

Make sure in **E_Commerce.WebApi** project is set as a startup project, you can run the application if run successfully you will see a swagger page with all endpoints



Swagger UI

localhost:7007/swagger/index.html

Select a definition: E_Commerce.WebApi

E_Commerce - WebApi v1 OAS3

/swagger/v1/swagger.json

This Api will be responsible for overall data distribution and authorization.

Aluwani Nethavhakone - Website
Send email to Aluwani Nethavhakone

Authorize

Account

- POST** /api/Account/authenticate Authentication / login User
- POST** /api/Account/register Register a new User.
- GET** /api/Account/confirm-email
- POST** /api/Account/refresh

Order

- POST** /api/v{version}/Order Create a new Order.
- GET** /api/v{version}/Order/{id} Retrieve the order by their ID.
- POST** /api/v{version}/Order/checkout Checkout order.

Front-end

Frontend is build using Angular 13.

5. Requirement

- Node.js
Angular requires an active LTS or maintenance LTS version of Node.js
- Angular CLI
Before building the app, you need to have Angular CLI installed. you can get it through **npm install -g @angular/cli**
- npm package manager
Once Node and Angular cli are install you can open ui project and run npm i to install all required packages for the project.

6. Environment REST API url

Front-end require api-url to communicate with REST API please update apiUrl in environment.ts file located in app/Environment

The screenshot shows an IDE with three main panels. The left panel is the file explorer, showing a project structure with 'src' as the root. Under 'src', there is an 'app' folder containing subfolders like 'core', 'data', 'features', 'guards', 'layouts', 'pages', and 'shared'. It also lists several TypeScript files, including 'app-navigation.ts', 'app-routing.module.ts', 'app.component.html', 'app.component.scss', 'app.component.ts', 'app.module.ts', 'not-authorized-container.ts', and 'unauthenticated-content.ts'. Below these are 'assets' and 'environments' folders, with 'environment.prod.ts' and 'environment.ts' listed. The 'environment.ts' file is selected. The middle panel displays the code for 'environment.ts'. It starts with comments explaining that the file can be replaced during build and that the list of file replacements can be found in the Angular CLI documentation. The code defines an 'environment' object with 'production' set to false and 'apiUrl' set to 'https://localhost:7007'. There are also comments about debugging in development mode and ignoring zone-related error stack frames. The bottom panel is the terminal, which shows the output of the build process. It has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active, showing the build output for 'main.js' (309.69 kB) and 'runtime.js' (6.52 kB).

```
1 // This file can be replaced during build by using the Angular CLI
2 // `ng build` replaces `environment.ts` with `environment.prod.ts`.
3 // The list of file replacements can be found in `angular.json`.
4
5 export const environment = {
6   production: false,
7   apiUrl: 'https://localhost:7007'
8 };
9
10 /*
11  * For easier debugging in development mode, you can set the zone.js
12  * to ignore zone related error stack frames such as `zone.js` or
13  * `zone-patch/browser-zone-patch.js`.
14  * This import should be commented out in production to avoid
15  * on performance if an error is thrown.
16  */
17 // import 'zone.js/plugins/zone-error'; // Included with Angular CLI
18
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
	main.js	main	309.69 kB
	runtime.js	runtime	6.52 kB

7. Start Application

Run “npm run start” command to start the application, once compiled successfully you can open your favorite browser and go to <http://localhost:4200> which will open frontend UI.

