

HandsOn - Project Setup

This project uses **.NET 9** for the back-end and **Angular with Nx** for the front-end. Below are the detailed instructions to set up and run the development environment.

📋 Requirements

Before starting, install the following dependencies on your system:

🖥 Front-End

- [Node.js 20+](#) (Tested with: **v22.14.0**)
- [Nx CLI](#) (Install globally):

```
npm add --global nx@latest
```

🖥 Back-End

- [.NET 9](#)
- [Entity Framework Core](#) - Install globally after .NET:

```
dotnet tool install --global dotnet-ef
```

- [MySQL](#)
- A SQL database management tool, such as:
 - [DBeaver](#)
 - [HeidiSQL](#)
 - [MySQL Workbench](#)

🔧 Auxiliary Tools

- [Git](#)
- [Visual Studio Code \(VSCode\)](#)
 - Recommended extensions:
 - [.NET Extension Pack](#)
 - [Nx Console](#)
 - [Prettier - Code Formatter](#)
 - [ESLint](#)

⚙ How to Run the Project

After installing all the requirements, follow the instructions to run the back-end and front-end.

🖥 Back-End

1 - Configure Environment Variables

- Navigate to the back-end root folder:

```
cd HandsOn-Back
```

- Open the configuration file:

```
HandsOn-Back/src/API/appsettings.json
```

- Edit the database connection details to match your MySQL credentials:

```
"ConnectionStrings": {  
  "DefaultConnection": "server=YOUR_SERVER;port=YOUR_SERVER_PORT;database=YOUR_DATABASE;user=YOUR_USER;password=YOUR_PASSWORD"  
}
```

- **Tip:** You can create a `appsettings.Development.json` file to avoid modifying the original file.

2 - Restore Dependencies

Run the following command to download the required packages:

```
dotnet restore
```

3 - Create the Database and Apply Migrations

- If the database does not exist, create it in MySQL.
- Apply Entity Framework migrations:

```
dotnet ef database update --verbose --project "src/Infrastructure" --startup-project "src/API" --context UsersDbContext
```

- To create new migrations, use:

```
dotnet ef migrations add "migration_name" --verbose --project "src/Infrastructure" --startup-project "src/API" --context User
```

4 - Run the API

- **Via Terminal:**

```
dotnet run --project "src/API/API.csproj"
```

- **Or using VS Code:**

- Open the project in VS Code.
- Ensure the **startup project** is set to:

```
./src/API/API.csproj
```

- Use the **debugger** to start the application.

🔗 The API will be available at: `http://localhost:5143`

🔗 Accessing Swagger

After starting the API, you can access the Swagger documentation in your browser at:

🔗 <http://localhost:5143>

Swagger allows you to test endpoints and view API documentation interactively.

🔗 Front-End

1 - Configure Environment Variables

- Navigate to the front-end root folder:

```
cd HandsOn
```

- Locate the environment file:

```
HandsOn/src/app/environments/environment.prod.ts
```

- Copy and rename it to `environment.ts`:

```
cp HandsOn/src/app/environments/environment.prod.ts HandsOn/src/app/environments/environment.ts
```

- Edit the `environment.ts` file and update the API URL to point to the back-end:

```
export const environment = {  
  production: false,  
  
  clientId: "",  
  redirectUri: "",  
  
  jwtToken: "jwt_token",  
  allowedDomains: ["http://localhost:5143"],  
  
  authApiUrl: "http://localhost:5143/api/users",  
  usersApiUrl: "http://localhost:5143/api/users",  
};
```

2 - Install Dependencies

Run the following command in the front-end root directory to install the required packages:

```
npm install
```

3 - Run the Front-End

Start the Angular server:

```
npm run start
```

🔗 The front-end will be available at: <http://localhost:4200/>.