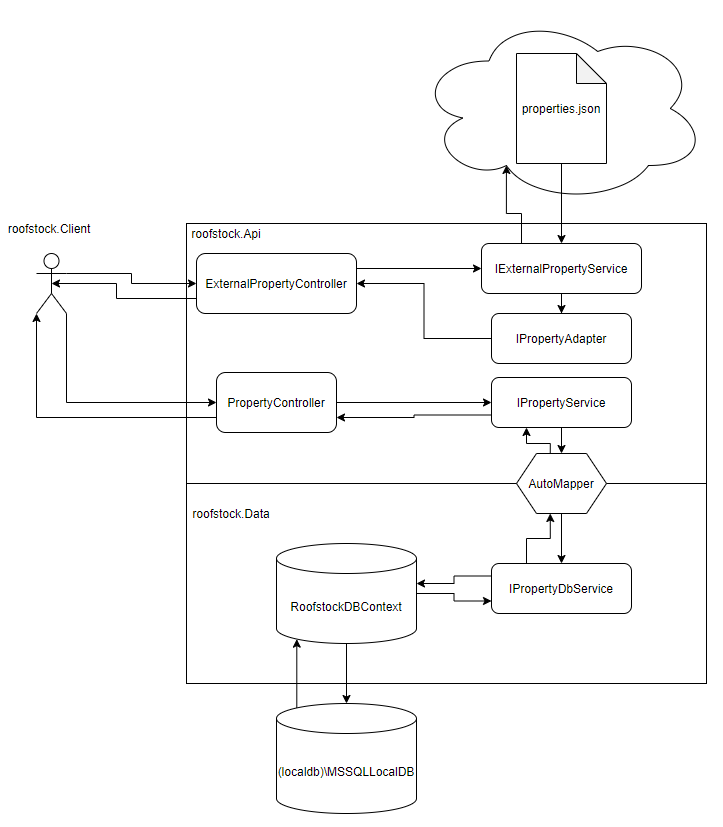
# Coding Exercise – Full stack

The exercise was built in two simple projects as WebAPI (NetCore 3.1) and UI-Client (Angular 11). Regarding to the WebApi was split into two layers/projects: roofstock.Api and roofstock.Data. In the roofstock.Api I register the services using DependencyInjection and provided a couple of extension methods to register the DbContext from roofstock.Data and the AutoMapper custom configuration which is how I communicate between layers. The DbContext uses EFCore with LINQ and CodeFirst strategy for the database creation. For consuming the external file from the API provided in the requirement I used the Adapter patter to convert the data.

# Architecture Diagram



# How to run the Application

## Requirements:

1- Git Client: <https://git-scm.com/book/en/v2/Getting-Started-The-Command-Line>

2- NetCore CLI: <https://docs.microsoft.com/en-us/dotnet/core/tools/>

3- EFCore CLI: <https://docs.microsoft.com/en-us/ef/core/cli/dotnet>

4- Angular CLI: <https://angular.io/cli>

5- SQLExpress: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-express-localdb?view=sql-server-ver15>

## Environment setup:

1- Open a new console and clone the repository with the Api

> git clone <https://github.com/crissanchez92/properties-api.git>

2- Open a new console and clone the repository with the Client

> git clone <https://github.com/crissanchez92/properties-client.git>

3- In the Api console navigate to the Api folder

> cd properties-api

4- Generate the Database (**Review notes at the bottom**)[1]:

> cd roofstock.Data

> dotnet ef database update

5- Build the Api

> cd ../roofstock.Api

> dotnet restore

> dotnet build

6- Run the Api

> dotnet run

7- In the Client console navigate to the Client folder and use develop branch

> cd properties-client

> git checkout develop

8- Install dependencies for Client

> yarn install

9- Run the Client

> ng serve

10- Open the Application

> <http://localhost:4200/>

## Notes:

[1]: make sure the connection string is pointing to valid local SQL Server, appsettings.json has the default connection string so use the following pattern in case of changing it:

Data Source=[YOUR\_SQLSERVER];Initial Catalog=Roofstock;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False

Both database creation factory and dbcontext are using the connection string from appsettings.json in roofstock.Api project.

The appsettings file is located in the folder: **\properties-api\roofstock.Api\appsettings.json**.