COMPASS PROJECT BACKEND SOLUTION DETAILS:

The Backend solution follows the encapsulated architect, Not exactly Domain Driven Design pattern, However the idea behind the design follows the same rule.

I have implemented the minimum code for the back end, so you can open the solution in either Visual Studio Code or Visual Studio to look at the structure of the development.

The Solution consist of 5 projects:

1) **Domain** : Which basically implements the Entities definitions like Survey, Question and Option

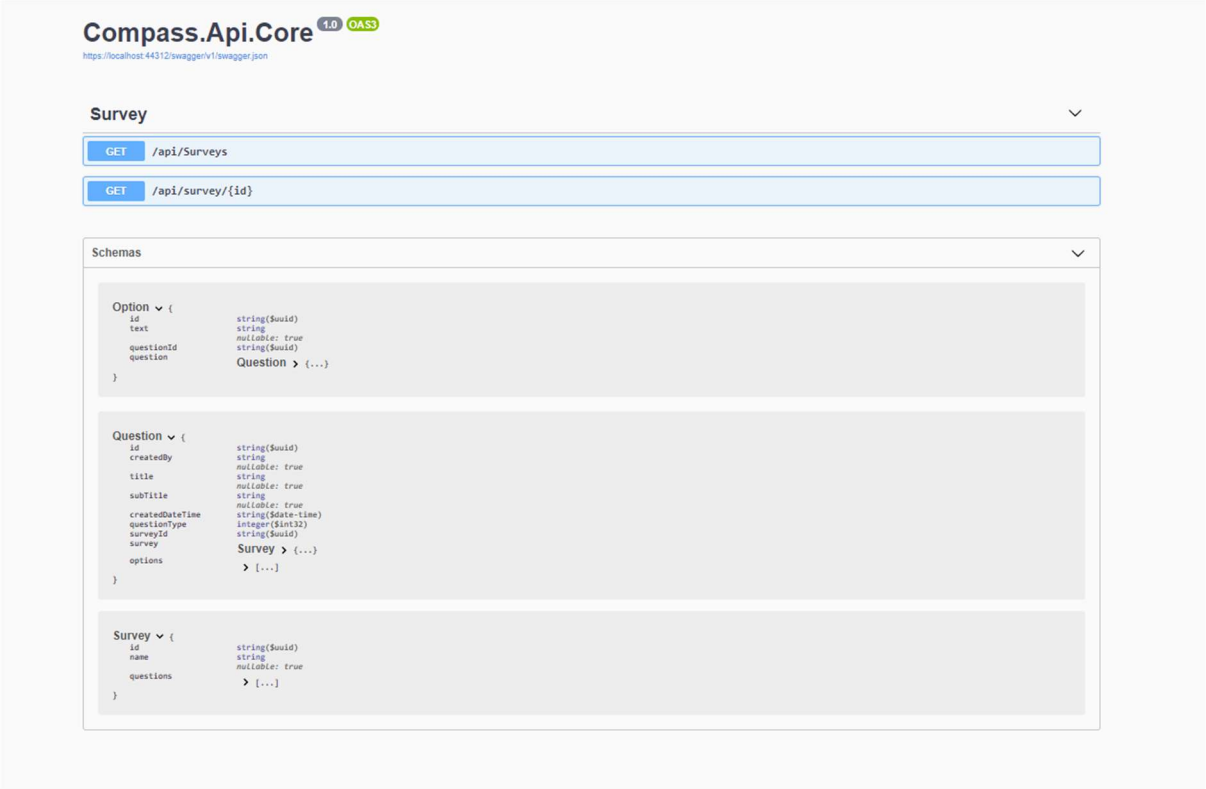
2) **Repository**: Which is handling all the DB connection handling code, I used Entity Framework for Initial Adding and proposed migrations for later changes which will keep the domain and the DB in Sync, I have chosen the **Sql Server** database since I am more familiar and comfortable with it right now, it can be changed to nay different relational DB if desired.

3) **Services**: Which is responsible to all complex functionality in the system and will be in contact with Repository to manipulate data in the database.

4) **RESTful API**: This the main project in the Solution which will be the point of the contact with outside world to provide data and update the Data base through using our services project. Each Controller would have all CRUD Methods, however for the test purposes I just implemented to Get methods to feed the UI.

   Note: The Service will be passed to the **API** Controller in order to have the ability of dependency injections as I already implemented that in the project to test the controller with the Mock object of service .

Here is a snapshot of what I have implemented as the API interface:



5) **API.Test :**Is basically responsible for API Controller Methods Automation testing.

Note: Noe that I am wiring the description, as we Add more functionalities to the Services Project I probably would add the **Services.Unit.Test** project as well to automate all the complex functionalities that service is responsible for. And move towards mor TDD environment.