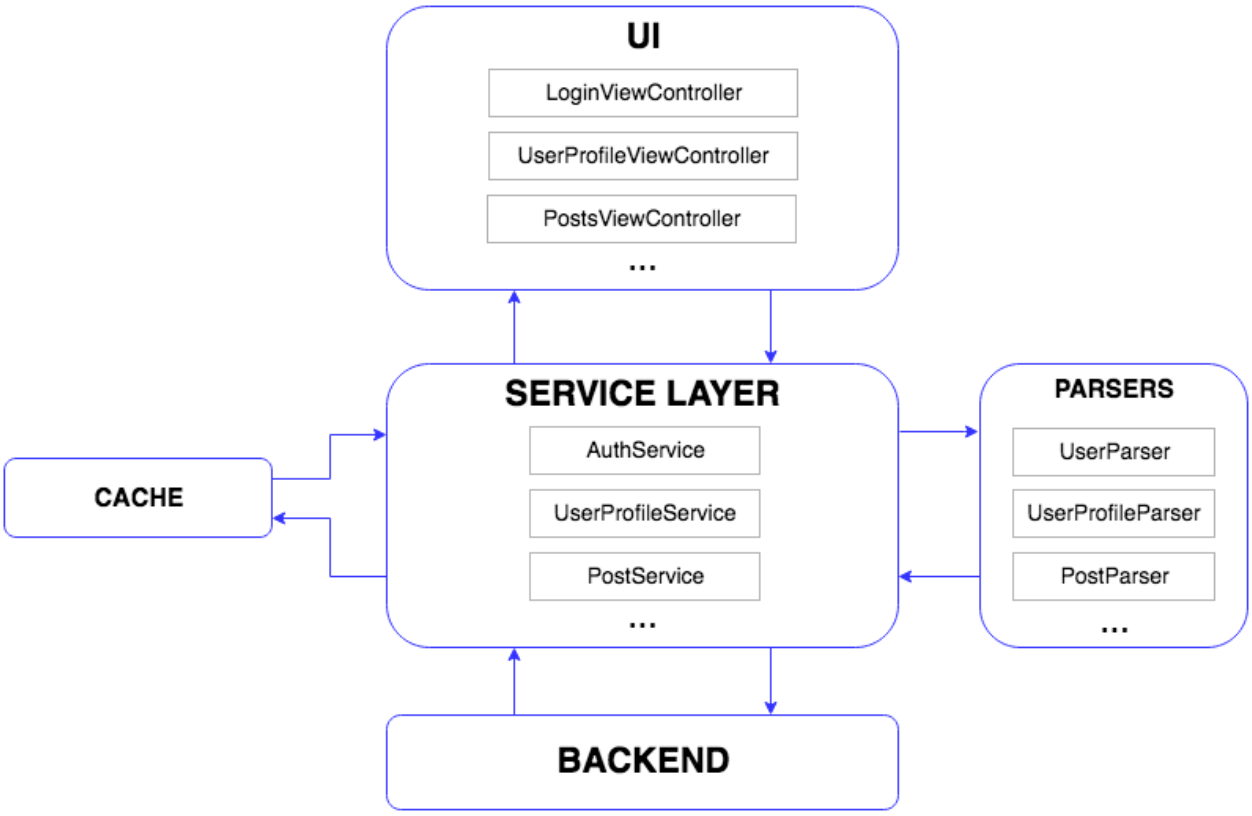
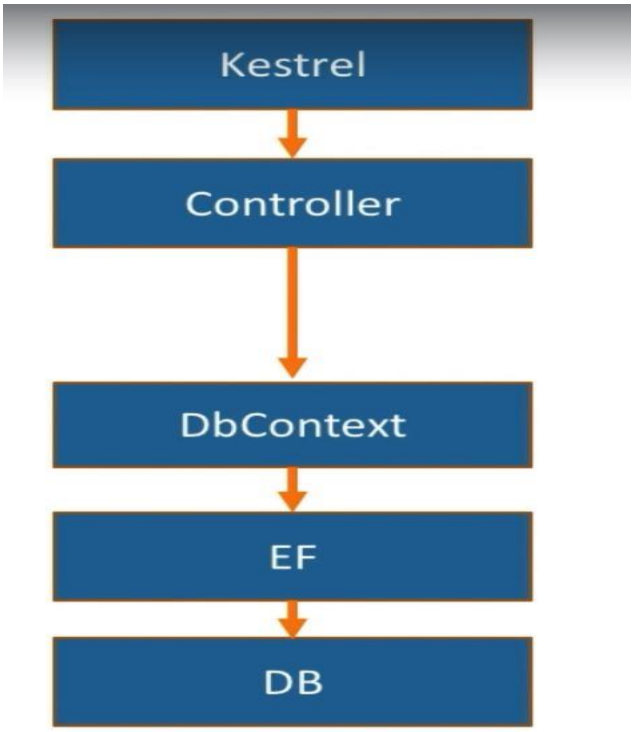
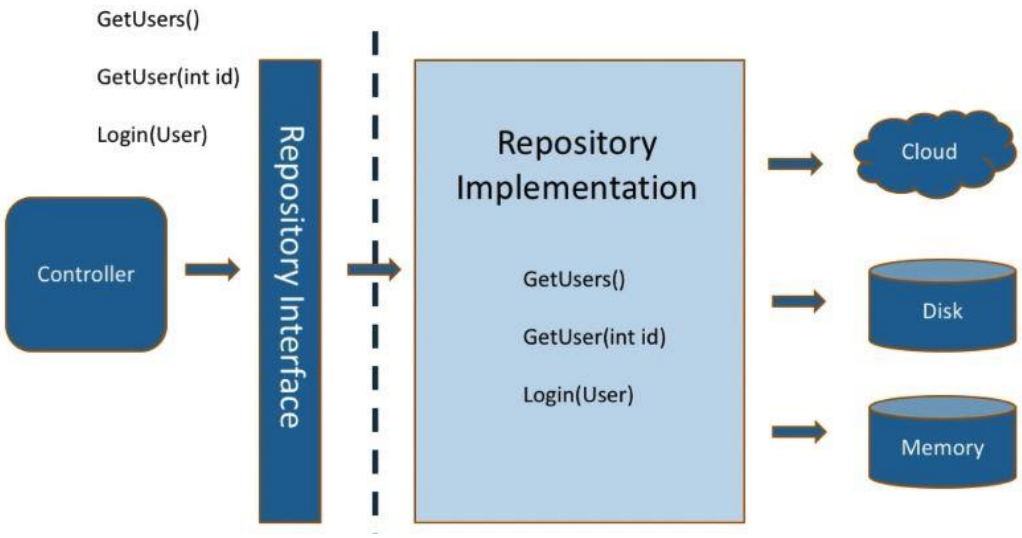
Below is the repository between the data source layer and the business layers of the application. It queries the data source and the data, maps the data from the data source to a business entity, and persists changes in the business entity to the data source. This repository will separate the business logic from the interaction with the web service.

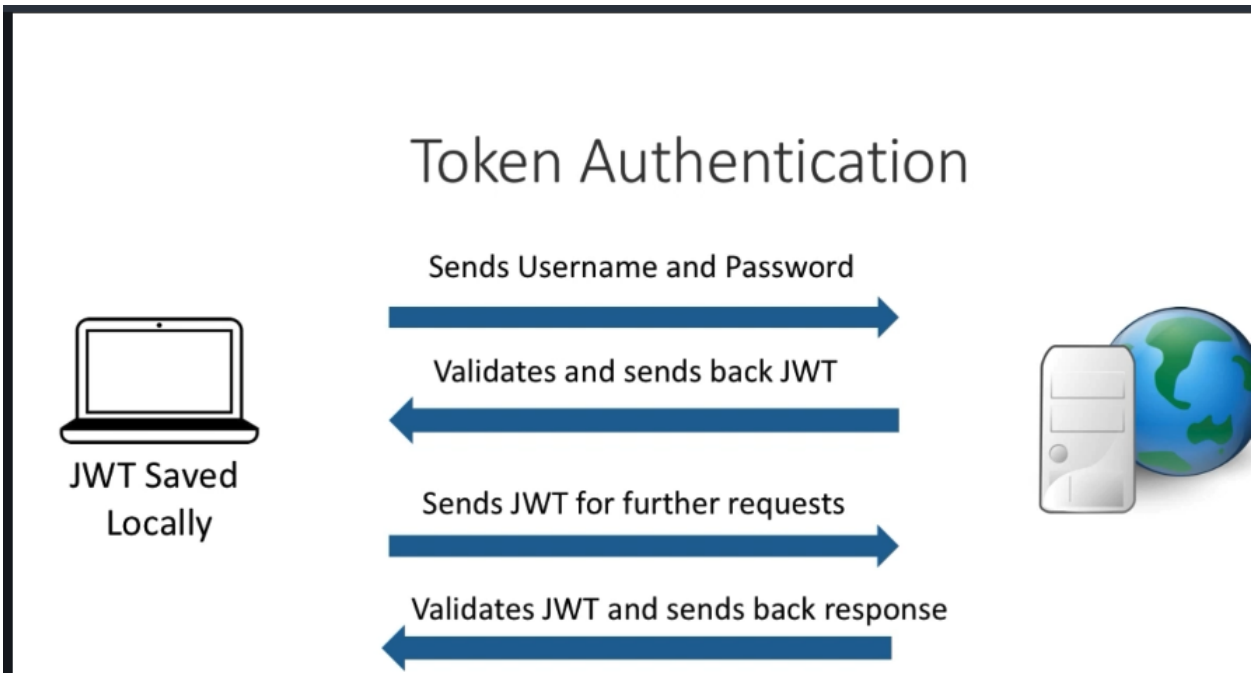




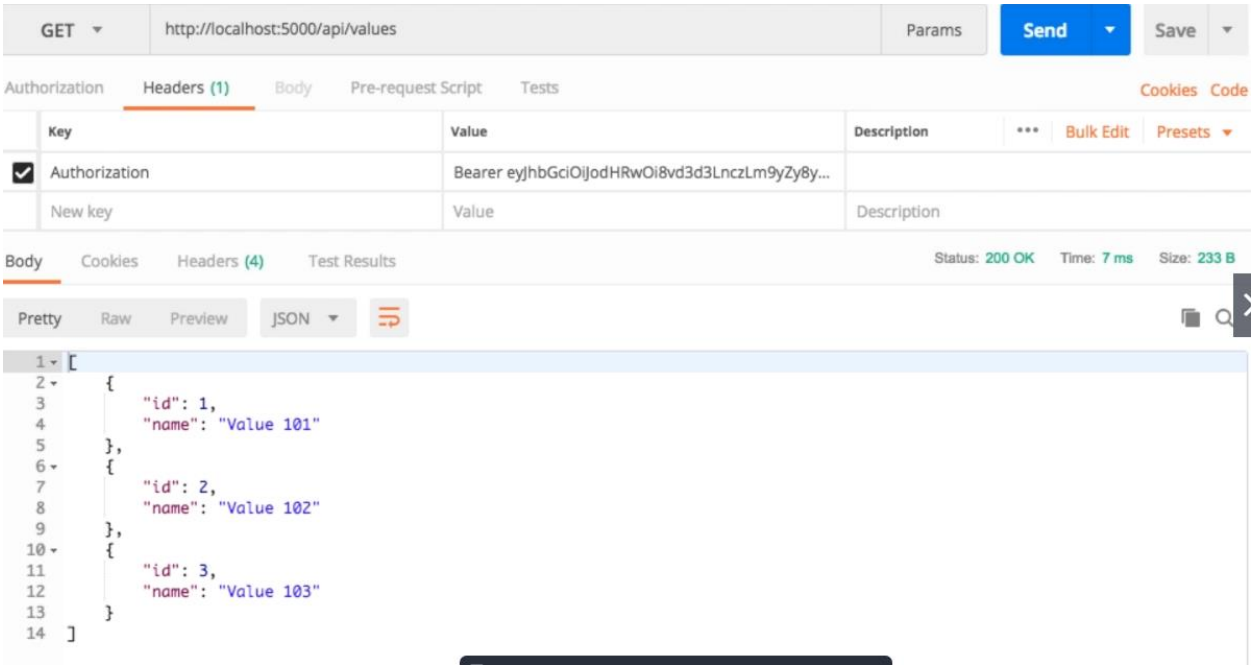
The interface itself exposes methods that the controller can utilize. In the case our authentication coming up, we are going to use free method to get userid and login to particular users. We write the instruction to entity framework, so it can go out and query whichever the data sources are.



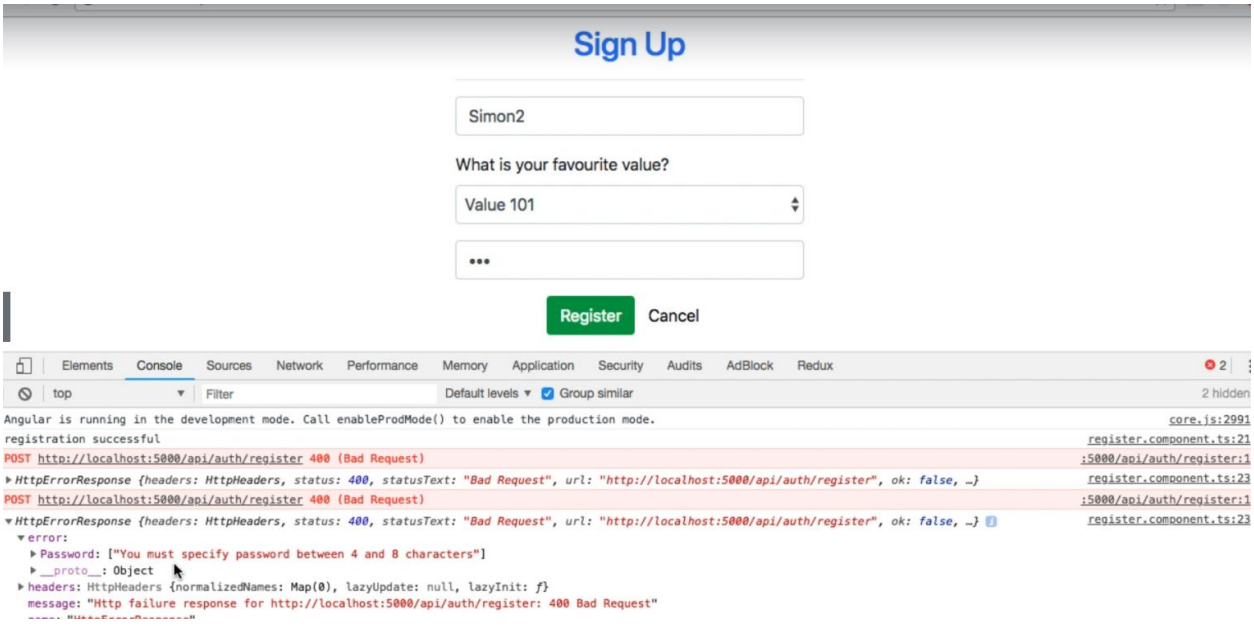
Rest API and Token Authentication: In the project, I will be utilizing Rest API standard methods to get request from users and send to the server for validations. Once the server can validate the credentials, it will send back to the users a JWT. JWT is a Json web tokens which contain secret information such as credentials, claims, and other information. Once the user got the JWT saved, for every single API request to the server, the users only need to send JWT to server to authenticate without having the server to go check the user and password over again.



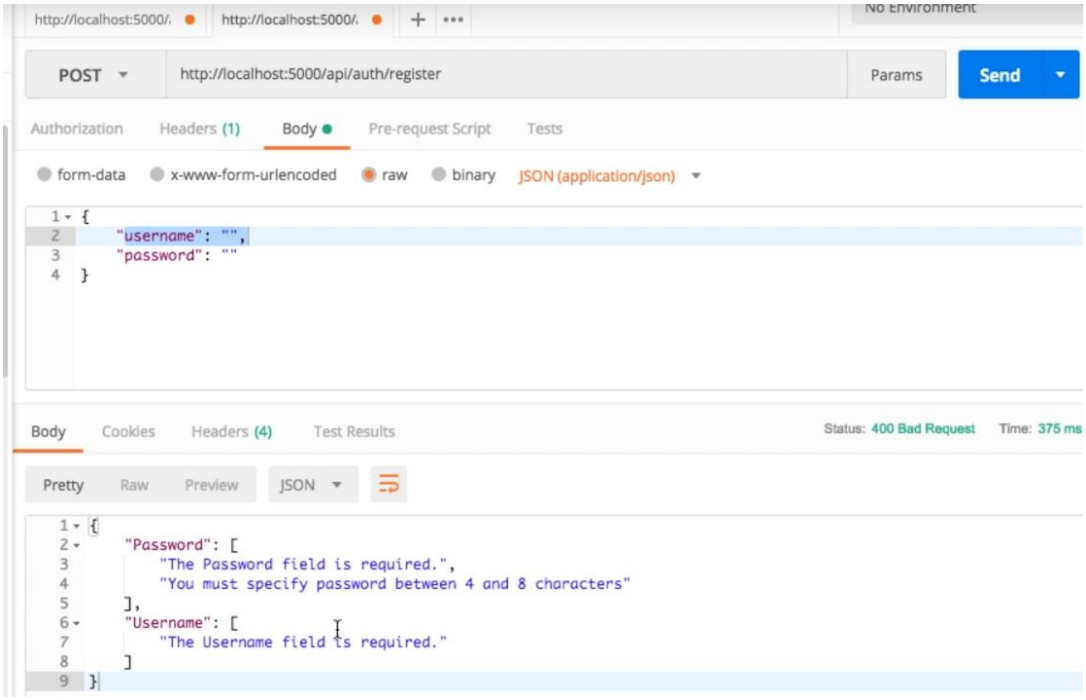
Success Request and Response with token validation



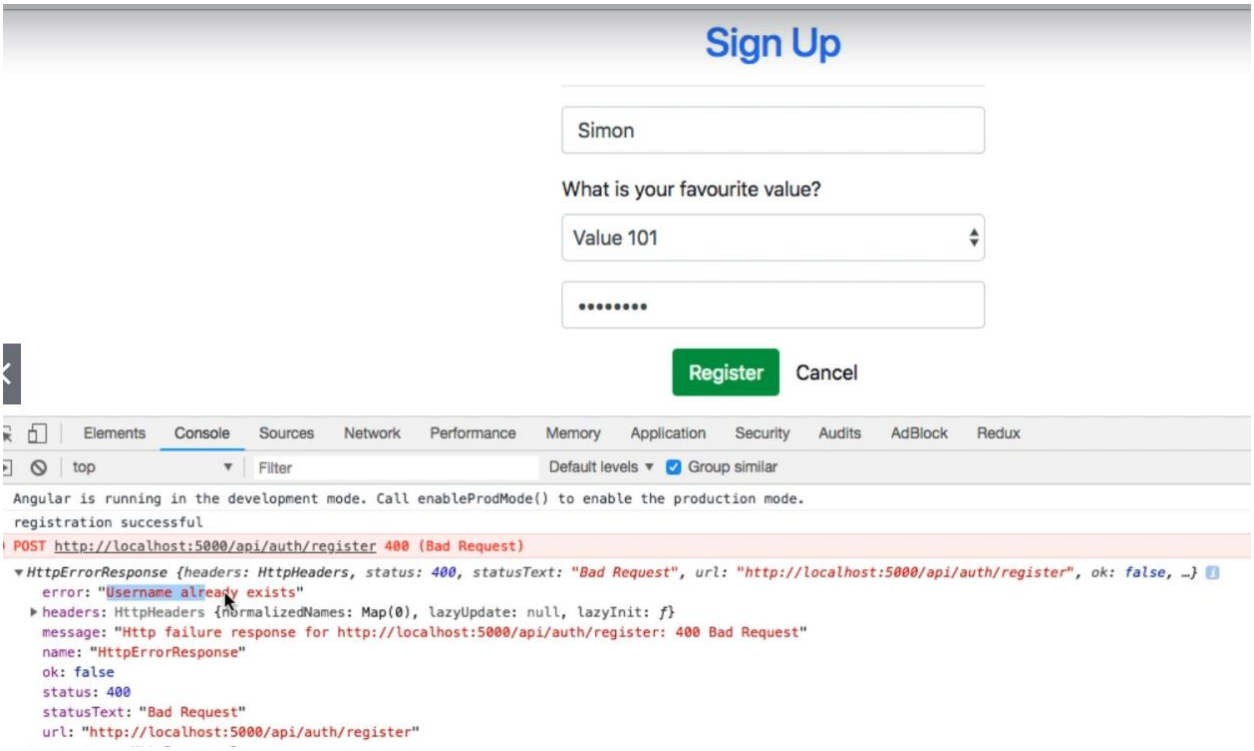
Error 400: Bad Request



Error 400: Password is required



Error 400: User already exist



New feature: Adding the “Messaging Service System” – This will be a private message that provides one to one chat system that users can utilize to exchange messages

- The Message Entity
- CRUD
- Inbox, Outbox, Unread Messages
- Delete message
- Mark as Read