**EntityMetadataWebAPI Doc**

This assignment is developed using **ASP.Net Core Web API**. It has got 3 layer

1. API Endpoint layer (**EntityMetadataWebAPI**)

This layer basically exposes the API endpoint i.e. GET (for **Reading configuration requirement**) and POST (for **Saving configuration requirement**). This layer also wraps the response data into **ResponseData<T>** object,whichis our generic response object.

1. Business Layer (**EntityMetadataBL**)

This layer holds the business logic which needs to be performed as per the requirement for both these end points i.e.

I. GET (for **Reading configuration requirement**)

**“System shall be able to get the data from two different sources and merge them. Once it's merged then join with the configuration that's available in the database (structure of the sample data given below). And finally return the object”**

II. POST (for **Saving configuration requirement**)

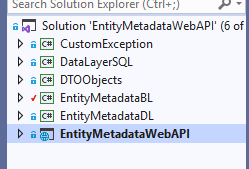
**“System shall be able to perform bulk insert/update operations. If a field specific entry is available then update that entry otherwise insert.”**

1. Data Layer (**EntityMetadataDL**)

This layer is responsible for communicating with database. Performing Read, Update and Insert operation as per the request.

**Project Structure**

Below is the project structure, we have total 6 projects. And they are described below:



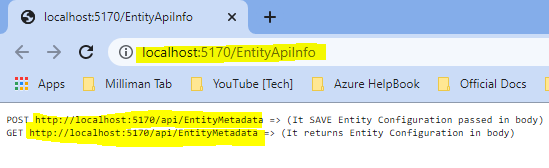
1. **CustomException:-** This library is developed for handling the exception with custom message as we don’t want the end user to get the internal programming error.
2. **DataLayerSQL:-** This library is responsible for communicating with database i.e. establishing connection, executing various queries, mapping those to given model/entity class.
3. **DTOObjects:-** This library is for storing data transfer objects with which we will bind the database object(tables and queries result set).
4. **EntityMetadataBL:-** This library is for handling the business logic of our Entity Metadata API as per the request.

It also contains all the helper methods which are needed to implement the business logic.

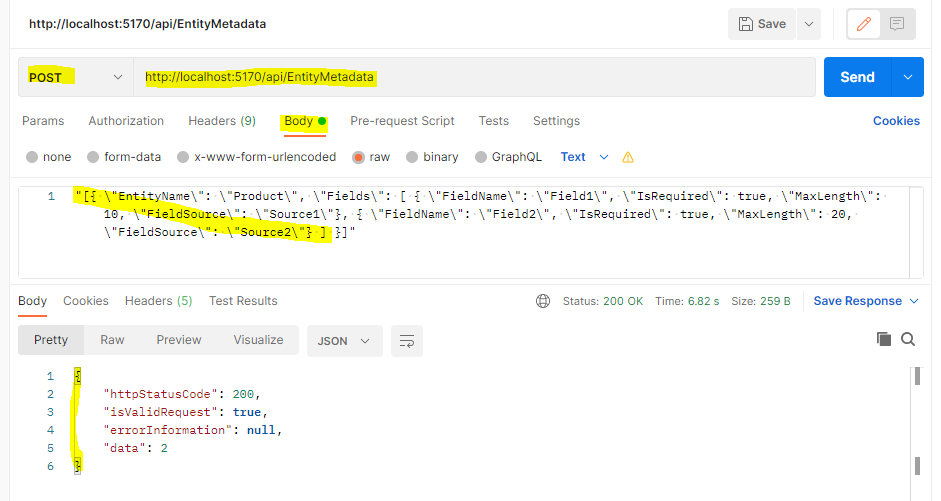
1. **EntityMetadataDL:-** This library contains the database oriented method which is responsible of preparing sql queries and parameters for execution on database engine.
2. **EntityMetadataWebAPI:-** This is the main Web API project which exposes the API endpoint i.e. GET (for **Reading configuration requirement**) and POST (for **Saving configuration requirement**). And also provides the response in generic response object.

**Project Setup** In order to setup and execute the project follow the below steps:

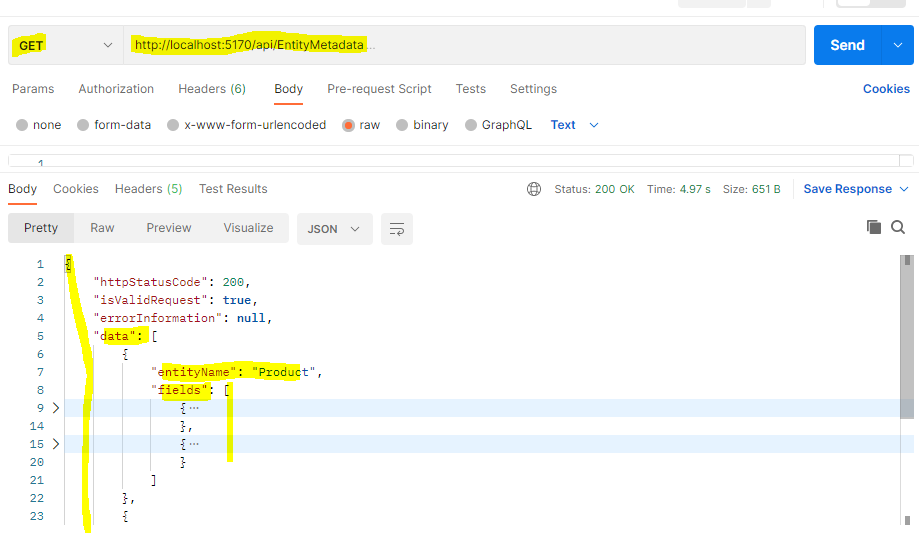
1. First of all, execute the **CreateSqlScript.sql** script into the sql server.
2. Open EntityMetadataWebAPI project in visual studio and build it and then run it. You will get to see the below output:



1. Now open the postman, and try to hit the above 2 highlighted url with correct http verb shown below:
   1. For Saving the configuration hit POST url:



* 1. For getting the configuration hit GET url:



**3rd Task:-** Describe how you can clean up the configured data that's already saved in the database if any fields are removed from source2 later time**.** System shall not need to maintain configuration if fields are not available in either of the sources.

**Approach:-** Considering we have only 2 API end point (GET & POST), so in case of GET request, where we are retrieving the fields configuration of different source fields from database, we can perform configuration cleanup for those fields which are not present in field list.