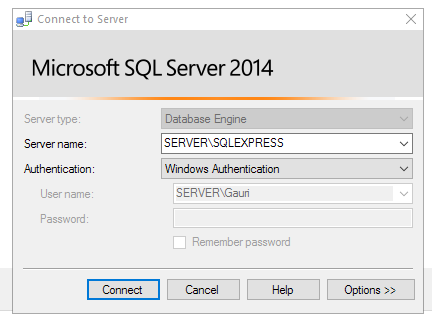
**1. Data Store Design**

1.1 To Store Data, Sql server is being used.

1.1.1 Open Sql Sever Management Studio, Select the server as shown below.



1.1.2 Open new Query window by Pressing ctrl+N, Copy following Query in it and press F5 to execute it.

USE MASTER

GO--SQL--

IF NOT EXISTS(SELECT \* FROM SYS.DATABASES WHERE NAME='ShowBridge')

BEGIN

CREATE DATABASE SHOWBRIDGE

END

GO--SQL--

Above Query will create a Database Named **SHOWBRIDGE**

1.1.3 Again Open new Query window by Pressing ctrl+N, Copy following Query in it and press F5 to execute it.

USE SHOWBRIDGE

GO--SQL--

IF NOT EXISTS(SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME='PRODUCT')

BEGIN

CREATE TABLE PRODUCT

(

ID BIGINT IDENTITY PRIMARY KEY,

NAME VARCHAR(50) NOT NULL CONSTRAINT UNIQUE\_PRODUCT UNIQUE ,

PRICE NUMERIC(15,2) NOT NULL,

DESCRIPTION VARCHAR(100) NOT NULL,

ACTIVE CHAR(1) NOT NULL

)

END

Above Query will create a Table Named Product**.**

**END OF DATA Store Design**

**2. Api Logic, Visual Studio 2019 is being Used.**

2.1 get the copy of project from github repository and download and load it in visual studio 2019.

<https://github.com/gau-910/ShowBridge/tree/master>

2.2 In project, Open WebConfig File and Change Server Name

(Replace SERVER\SQLEXPRESS with your server name in following line of config File)

<add name="ShowBridgeEntities" connectionString="metadata=res://\*/InventoryModel.csdl|res://\*/InventoryModel.ssdl|res://\*/InventoryModel.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=SERVER\SQLEXPRESS;initial catalog=SHOWBRIDGE;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

2.3 Click on F5 to run the project.

**3. How To Consume API**

**3.1 To Get List Of Products**

**REQUEST :**

  "url": "https://localhost:XXXX/api/product",

  "method": "GET"

  "headers":

 {

    "Content-Type": "application/json"

  }

**RESPONSE :**

Status Code : 200 ok

[application/json]

[{"ID":1,"NAME":"Product 1","PRICE":3.00,"DESCRIPTION":"Product is Scalable","ACTIVE":"1"}]

**3.2 To Add Product**

**REQUEST :**

  "url": "https://localhost:XXXX/api/product",

  "method": "POST"

  "headers":

 {

    "Content-Type": "application/json"

  }

Body :

{"ID":"1","NAME":"Product 2","PRICE":3,"DESCRIPTION":"Product is Scalable","ACTIVE":**true**}

**RESPONSE :**

Status Code : 200 ok

[application/json]

{"ID":2,"NAME":"Product 2", "PRICE":3.0, "DESCRIPTION":"Product is Scalable", "ACTIVE":"1"}

**3.3 To Update Product**

**REQUEST :**

  "url": "https://localhost:XXXX/api/product/1",

  "method": "Put"

  "headers":

 {

    "Content-Type": "application/json"

  }

Body :

{"ID":"1","NAME":"Product 3","PRICE":3,"DESCRIPTION":"Product is Scalable","ACTIVE":**true**}

**RESPONSE :**

Status Code : 204 no Content

**3.4 To Delete Product**

**REQUEST :**

  "url": "https://localhost:XXXX/api/product/1",

  "method": "Delete"

  "headers":

 {

    "Content-Type": "application/json"

  }

**RESPONSE :**

Status Code : 200 ok

{"ID":"1","NAME":"Product 3","PRICE":3,"DESCRIPTION":"Product is Scalable","ACTIVE":**true**}