How to Create WEB API in ASP.NET CORE?

➢ DataBase & SP

- **Database Name = APIDemo**

❖ **Select All User Store Procedure**

```
CREATE OR ALTER PROCEDURE PR_SELECT_ALL_USER
AS
BEGIN
	SELECT
		[UserID]
		,[Name]
		,[Contact]
		,[Email]
	FROM [dbo].[UserTbl]
END
```

❖ **Select By PK User Store Procedure**

```
CREATE OR ALTER PROCEDURE PR_SELECT_BY_PK_USER 2
@UserID INT
AS
BEGIN
	SELECT
		[UserID]
		,[Name]
		,[Contact]
		,[Email]
	FROM [dbo].[UserTbl]
	WHERE
		[UserID] = @UserID
END
```

❖ **Insert User Data Store Procedure**

```
CREATE OR ALTER PROCEDURE PR_INSERT_USER
@Name VARCHAR(50)
,@Contact VARCHAR(50)
,@Email VARCHAR(50)
AS
BEGIN
	INSERT INTO [dbo].[UserTbl]
	VALUES
	(
		@Name
		,@Contact
		,@Email
	)
END
```

❖ **Update User Data Store Procedure**

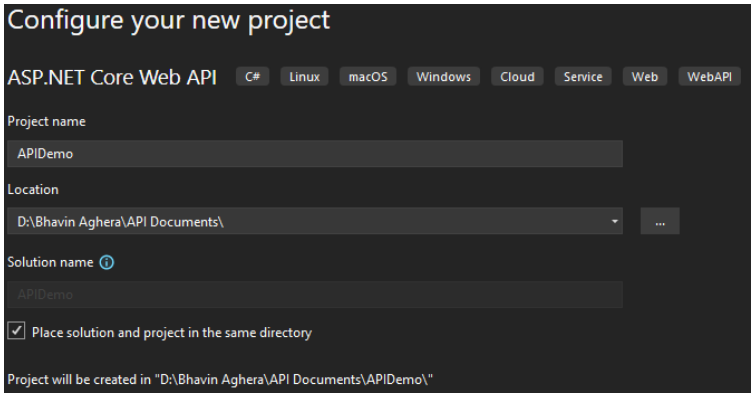
```
CREATE OR ALTER PROCEDURE PR_UPDATE_USER
@UserID INT
,@Name VARCHAR(50)
,@Contact VARCHAR(50)
,@Email VARCHAR(50)
AS
BEGIN
	UPDATE [dbo].[UserTbl]
	SET
		[Name] = @Name
		,[Contact] = @Contact
		,[Email] = @Email
	WHERE
		[UserID] = @UserID
END
```

❖ **Delete User Data Store Procedure**

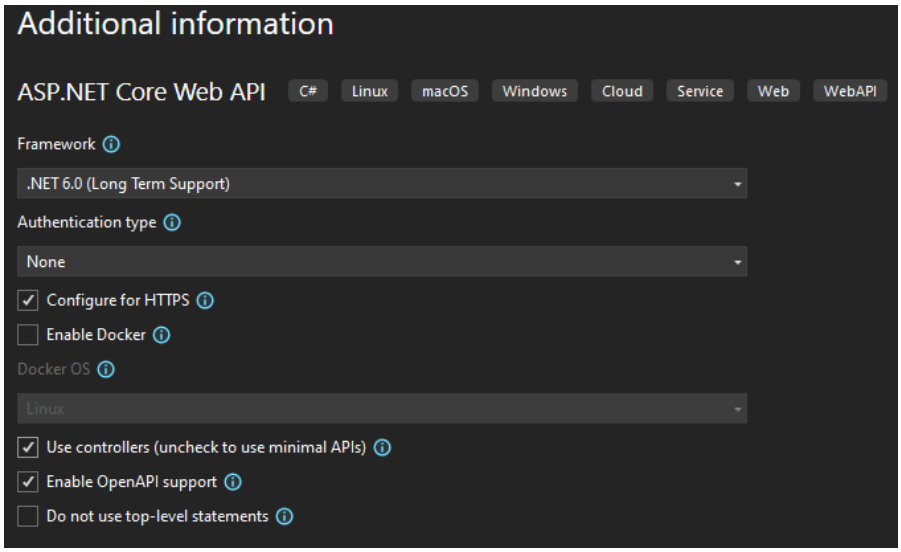
```
CREATE OR ALTER PROCEDURE PR_DELETE_USER
@UserID INT
AS
BEGIN
	DELETE FROM [dbo].[UserTbl]
	WHERE
		[UserID] = @UserID
END
```

➢ Creating the ASP .NET Core Web API Project

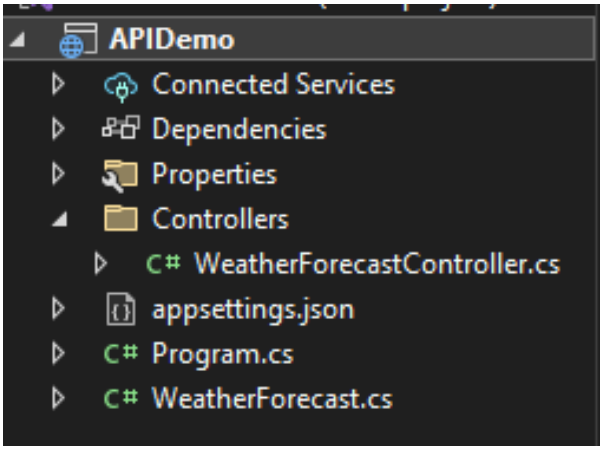
- Steps to Create the ASP.NET Core Web API Project
  - Open Visual Studio
  - Select **ASP .NET Core Web API** Project And Press **Next** Button
  - After that you show the following window
    - Give your **Project Name**
    - Select The Location Where you want to save the project
    - Press **Next** Button



  - In next window select the framework authentication Type as shown below
  - Press **Create** Button



- After Creating the Project you See the Following Folder Structure



- Delete the **WeatherForcastController.cs** & **WeatherForcast.cs** File Which is default file
- Now Create the Following Folder & Files
  - **Controllers**
    - **UserController.cs**
  - **BAL**
    - **User_BALBase.cs**
  - **DAL**
    - **DAL_Helpers.cs**
    - **User_DALBase.cs**
  - **Models**
    - **UserModel.cs**
- After Creating that put the following **Connection String** in **appsettings.json** file.

```
"ConnectionStrings": {
        "MyConnectionString": "Data Source= NAIMISH\SQLEXPRESS;Initial Catalog=APIDemo;
Integrated Security=true"
}
```

- Where,
  - **Data Source = Server Name**
  - **Initial Catelog = Database Name**
  - If Windows Authentication
    - Trusted Connection = true (if Windows Authentication)

    Data Source=NAIMISH\\SQLEXPRESS;
    Initial Catalog=APIDemo;
    Trusted_Connection=true
- Now Go to Nuget Package Manager and install following library
  - **System.Data.SqlClient**
  - **Microsoft.Practices.EnterpriseLibrary.Data**

➢ Coding Part

- Make a **UserModel** in UserModel.cs File

```
namespace APIDemo.Models
{
        public class UserModel
```

```
        {
            public int UserId { get; set; }
            public string Name { get; set; }
            public string Contact { get; set; }
            public string Email { get; set; }
        }
}
```

➢ DAL Folder

- **Get the Data From Database and bind it in UserModel**
- **User_DALBase.cs inherits DAL_Helpers.cs**

1. DAL_Helpers.cs

- Hold the Connection String of Database

```
namespace APIDemo.DAL
{
    1 reference
    public class DAL_Helpers
    {
        public static string ConnString = new ConfigurationBuilder().AddJsonFile("appsettings.json").Build
            ().GetConnectionString("MyConnectionString");
    }
}
```

2. User_DALBase.cs

- It Perform SelectAll, SelectByPk, Insert, Update, Delete Operation

❖ Select ALL User Data Method

```
#region PR_SELECT_ALL_USER
0 references
public List<UserModel> PR_SELECT_ALL_USER()
{
    try
    {
        SqlDatabase sqlDatabase = new SqlDatabase(ConnString);
        DbCommand dbCommand = sqlDatabase.GetStoredProcCommand("PR_SELECT_ALL_USER");
        List<UserModel> userModels = new List<UserModel>();
        using (IDataReader dr = sqlDatabase.ExecuteReader(dbCommand))
        {
            while (dr.Read())
            {
                UserModel userModel = new UserModel();
                userModel.UserId = Convert.ToInt32(dr["UserID"].ToString());
                userModel.Name = dr["Name"].ToString();
                userModel.Contact = dr["Contact"].ToString();
                userModel.Email = dr["Email"].ToString();
                userModels.Add(userModel);
            }
        }
        return userModels;
    }
    catch (Exception ex)
    {
        return null;
    }
}
#endregion
```

❖ Select By PK User Data Method

```
#region PR_SELECT_BY_PK_USER
1 reference
public UserModel PR_SELECT_BY_PK_USER(int UserID)
{
    try
    {
        SqlDatabase sqlDatabase = new SqlDatabase(ConnString);
        DbCommand dbCommand = sqlDatabase.GetStoredProcCommand("PR_SELECT_BY_PK_USER");
        sqlDatabase.AddInParameter(dbCommand, "@UserID", SqlDbType.Int, UserID);
        UserModel userModel = new UserModel();
        using (IDataReader dr = sqlDatabase.ExecuteReader(dbCommand))
        {
            userModel.UserId = Convert.ToInt32(dr["UserID"].ToString());
            userModel.Name = dr["Name"].ToString();
            userModel.Contact = dr["Contact"].ToString();
            userModel.Email = dr["Email"].ToString();
        }
        return userModel;
    }
    catch (Exception ex)
    {
        return null;
    }
}
#endregion
```

➢ BAL Folder

- This is the collection of class in which you can add your **business logic** there.
- It interects with **DAL classes** and get the data according to the business logic

❖ Select All User Data Method

```
#region PR_SELECT_ALL_USER
0 references
public List<UserModel> PR_SELECT_ALL_USER()
{
    try
    {
        User_DALBase user_DALBase = new User_DALBase();
        List<UserModel> userModels = user_DALBase.PR_SELECT_ALL_USER();
        return userModels;
    }
    catch (Exception ex)
    {
        return null;
    }
}
#endregion
```

❖ Select By PK User Data Method

```
#region PR_SELECT_BY_PK_USER
0 references
public UserModel PR_SELECT_BY_PK_USER(int UserID)
{
    try
    {
        User_DALBase user_DALBase = new User_DALBase();
        UserModel userModel = user_DALBase.PR_SELECT_BY_PK_USER(UserID);
        return userModel;
    }
    catch (Exception ex)
    {
        return null;
    }
}
#endregion
```

➢ Create the following Methods in Controller

❖ Get All User Data Method

- This method Make this type of URL
- **https://localhost:7241/api/User**

```
#region Get All Users
[HttpGet]
0 references
public IActionResult Get()
{
    User_BALBase bal = new User_BALBase();
    List<UserModel> users = bal.PR_SELECT_ALL_USER();
    // Make the Response in Key Value Pair
    Dictionary<string, dynamic> response = new Dictionary<string, dynamic>();
    if (users.Count > 0 && users != null)
    {
        response.Add("status", true);
        response.Add("message", "Data Found.");
        response.Add("data", users);
        return Ok(response);
    }
    else
    {
        response.Add("status", false);
        response.Add("message", "Data Not Found.");
        response.Add("data", null);
        return NotFound(response);
    }
}
#endregion
```

❖ Get UserData By ID Method

- This method Make this type of URL
- **https://localhost:7241/api/User/2**

```
#region Get User By UD
[HttpGet("{UserID}")]
0 references
public IActionResult Get(int UserID)
{
    User_BALBase bal = new User_BALBase();
    UserModel user = bal.PR_SELECT_BY_PK_USER(UserID);
    // Make the Response in Key Value Pair
    Dictionary<string, dynamic> response = new Dictionary<string, dynamic>();
    if (user.UserId != 0)
    {
        response.Add("status", true);
        response.Add("message", "Data Found.");
        response.Add("data", user);
        return Ok(response);
    }
    else
    {
        response.Add("status", false);
        response.Add("message", "Data Not Found.");
        response.Add("data", null);
        return NotFound(response);
    }
}
#endregion
```