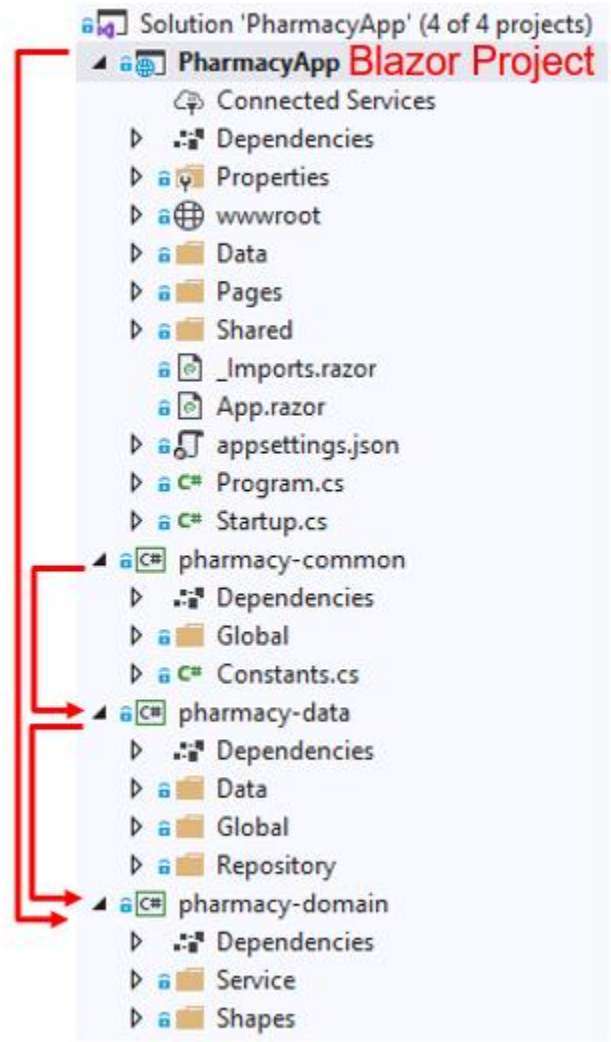
Table of Contents

1 – Create Blazor App Template :

The Blazor template is made up of four projects. The projects are dependent on each other and we show these dependencies using red arrows.

1. PharmacyApp project which contains the razor web pages, html, css, etc.
2. Pharmacy Common is a library project for storing commonly used Model classes like security and login models etc.
3. Pharmacy Data is a library project for storing all database calls. This is the application data layer or middle tier.
4. Pharmacy Domain contains all service layer components used in the razor pages.



The Template works like this.

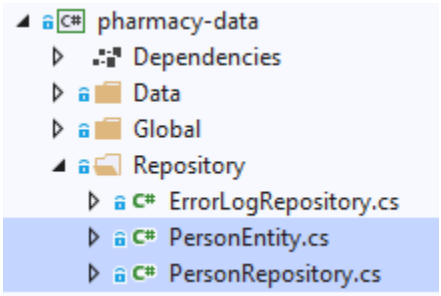
Step 1: We go into pharmacy-data\Data\SQLUtility.cs class the sql query. Add any stored procedures for inserting and updating data in the stored procedure section.

SQLUtility.cs

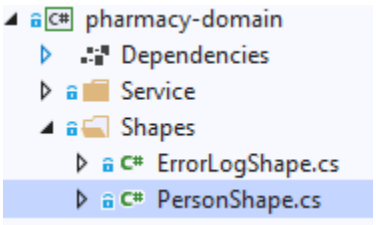
pharmacy-data — Pharmacy.Data.SQLUtility

```
1   using System;
2   using System.Collections.Generic;
3   using System.Text;
4   using System.Data;
5   using System.Data.SqlClient;
6
7   namespace Pharmacy.Data
8   {
        6 references | Jason Azim, 23 hours ago | 1 author, 1 change
9       public static class SQLUtility
10      {
11          public static readonly string VIEW_PREFIX = "v";
12          public static readonly string PERSON_TABLE_NAME = "HR";
13
14          public static readonly string WHERE = " where ";
15          public static readonly string ORDER_BY = " order by ";
16
17          public static readonly string PERSONS_VIEW_SELECT = "select top 20 MRN,FIRST_NAME,LAST_NAME from HR";
18
19          public static readonly string ERRORLOG_VIEW_SELECT = "SELECT ErrorLogID,ErrorClass,ErrorType,Code,Object,
20          public static readonly string ERRORLOG_FILTER_BY_ID = "ErrorLogID = {0}";
21
22          // utility class does not make database connections. It only works on database objects
23          Data Operations
300
301         Add Parameters
406
407         Stored procedure Calls
441
442     }
443
444 }
445
```

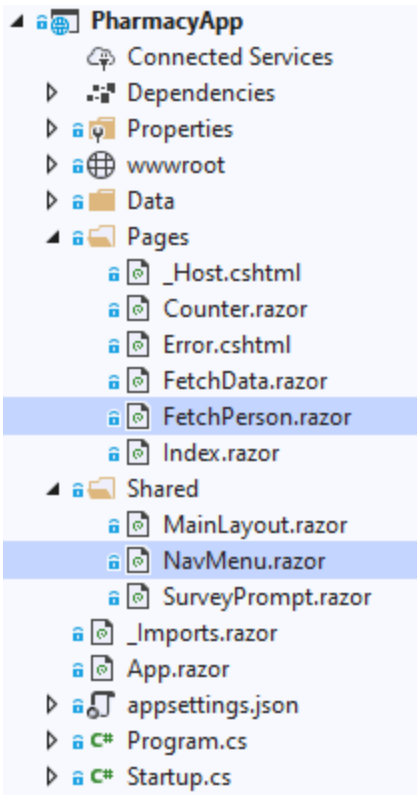
Step 2: Add a repository class which will call this query and create the corresponding database entity class called PersonEntity to hold this data.



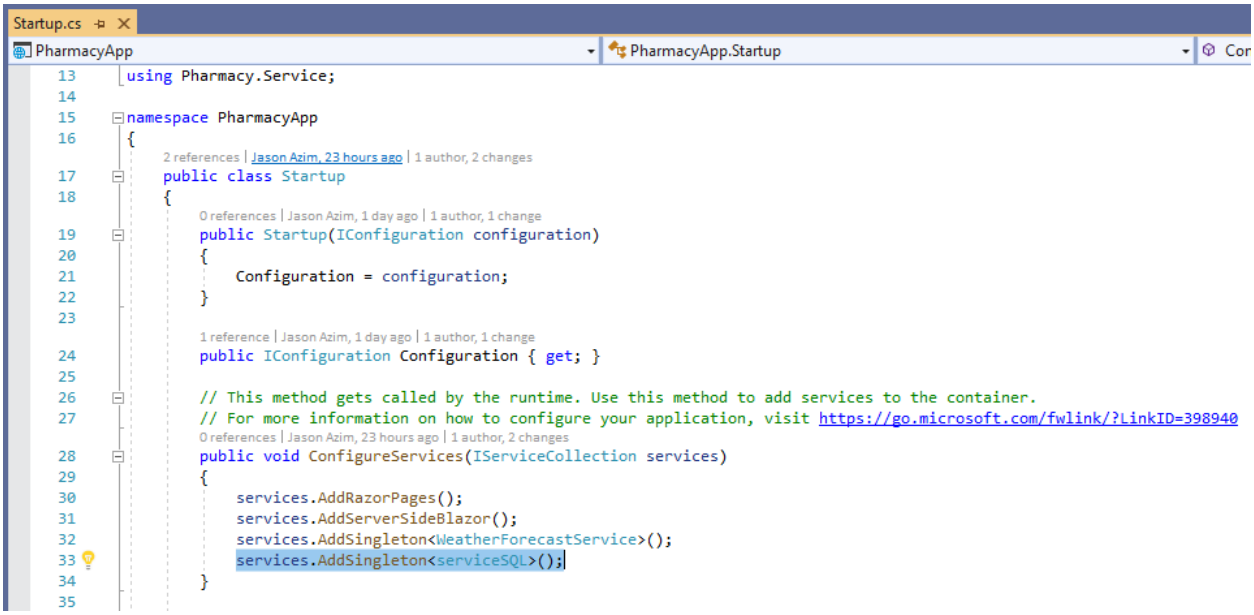
Step 3: Create the PersonShape class. This class is used in the Razor web components.



Step 4: Add a razor screen (Form) component which will be used to display the PersonShape class. The screen will need to be added to the navigation menu for the app so it shows up.



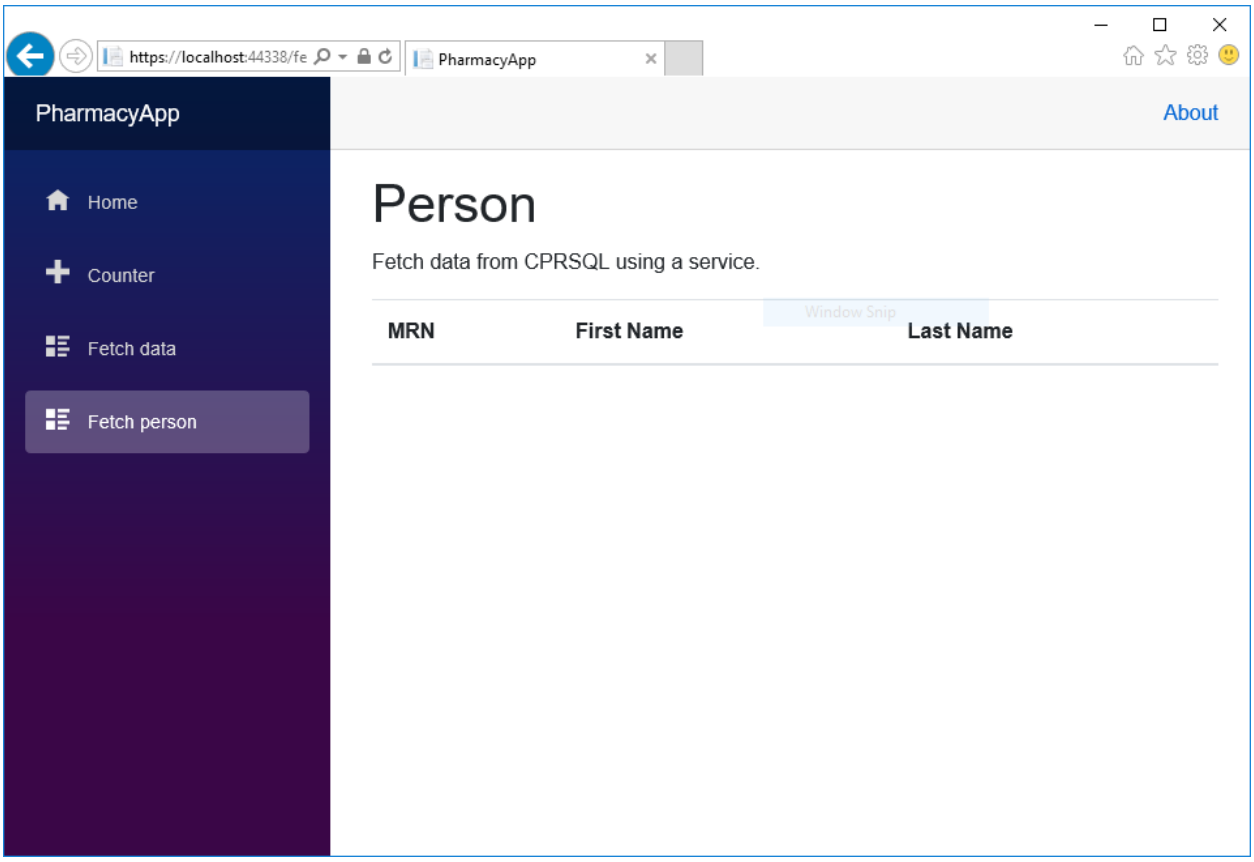
Step 5: Wire up the web service singleton for the Person Data Services.



Step 6: Run the Razor person fetch data screen.

2 – Use Telerik UI for Blazor to build Grid :

We can create our own custom grid which looks like this show below. We have a left navigation menu and we can see all the patient data in the panel on the right.



Appendix :

Build your first Blazor hello word app
https://dotnet.microsoft.com/en-us/learn/aspnet/blazor-tutorial/intro

Telerik UI for Blazor
https://demos.telerik.com/blazor-ui