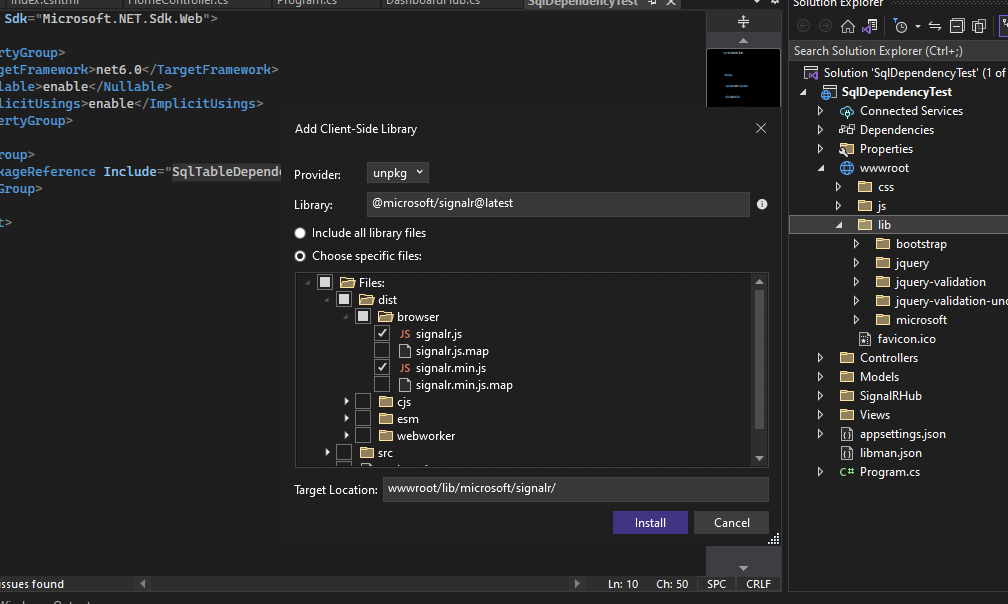
1. Create new project
2. Install Package SqlTableDependency
3. Add client side library called signalr
4. 

Create class DashboardHub as

public class DashboardHub : Hub

{

public async Task SendProducts()

{

try

{

var products = "test sql dependency";

await Clients.All.SendAsync("ReceivedProducts", products);

}

catch (Exception ex)

{

var msg=ex.Message;

}

}

}

Create ApplicationBuilderExtension class as

public static class ApplicationBuilderExtension

{

public static void UseSqlTableDependency<T>(this IApplicationBuilder applicationBuilder, string connectionString)

where T : ISubscribeTableDependency

{

var serviceProvider = applicationBuilder.ApplicationServices;

var service = serviceProvider.GetService<T>();

service.SubscribeTableDependency(connectionString);

}

}

Create ISubscribeTableDependency as

public interface ISubscribeTableDependency

{

void SubscribeTableDependency(string connectionString);

}

Create SubscribeProductTableDependency as

public class SubscribeProductTableDependency : ISubscribeTableDependency

{

SqlTableDependency<Product> tableDependency;

DashboardHub dashboardHub;

public SubscribeProductTableDependency(DashboardHub dashboardHub)

{

this.dashboardHub = dashboardHub;

}

public void SubscribeTableDependency(string connectionString)

{

tableDependency = new SqlTableDependency<Product>(connectionString);

tableDependency.OnChanged += TableDependency\_OnChanged;

tableDependency.OnError += TableDependency\_OnError;

tableDependency.Start();

}

private void TableDependency\_OnChanged(object sender, TableDependency.SqlClient.Base.EventArgs.RecordChangedEventArgs<Product> e)

{

if (e.ChangeType != TableDependency.SqlClient.Base.Enums.ChangeType.None)

{

dashboardHub.SendProducts();

}

}

private void TableDependency\_OnError(object sender, TableDependency.SqlClient.Base.EventArgs.ErrorEventArgs e)

{

Console.WriteLine($"{nameof(Product)} SqlTableDependency error: {e.Error.Message}");

}

}

Create class Product as

public class Product

{

[Key]

public int Id { get; set; }

[Required]

public string? Name { get; set; }

[Required]

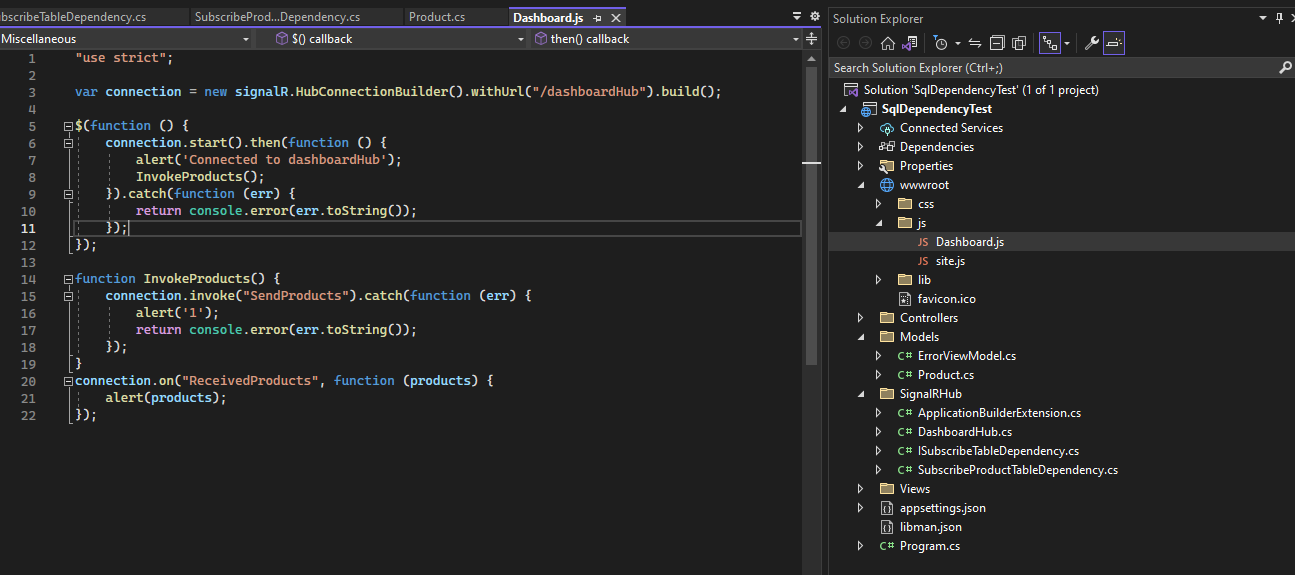
public string? Category { get; set; }

[Required]

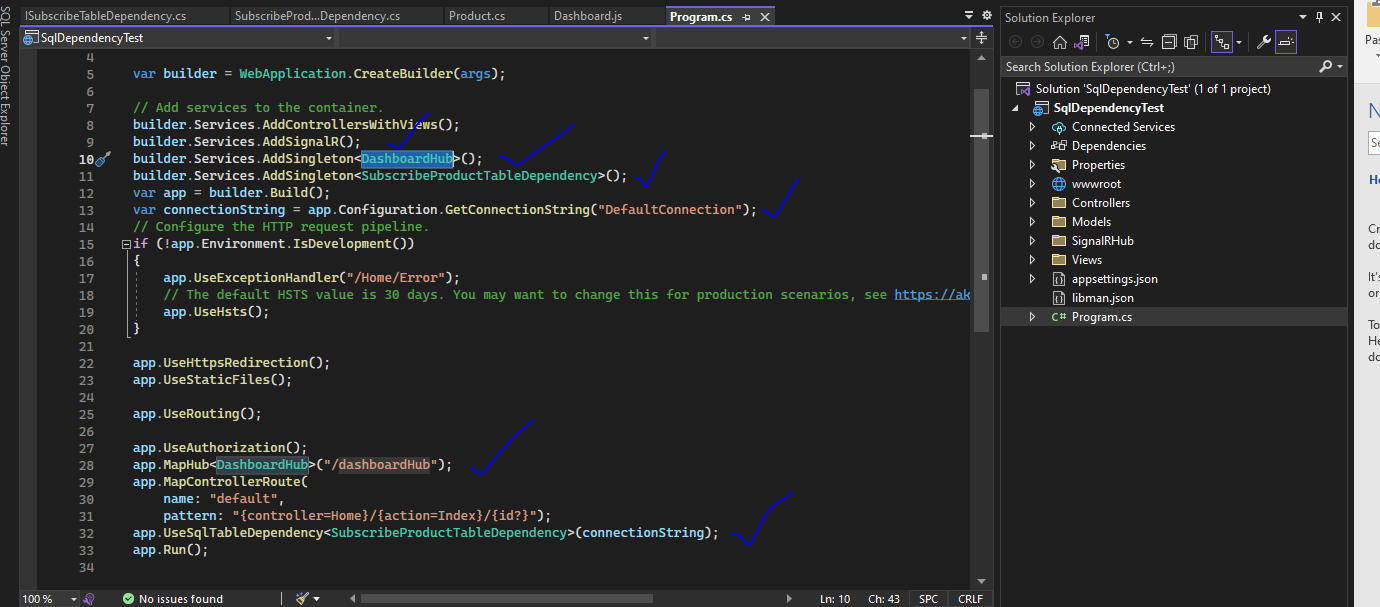
public decimal? Price { get; set; }

}

Add Dashboard.js file as follow



Update program.cs file as



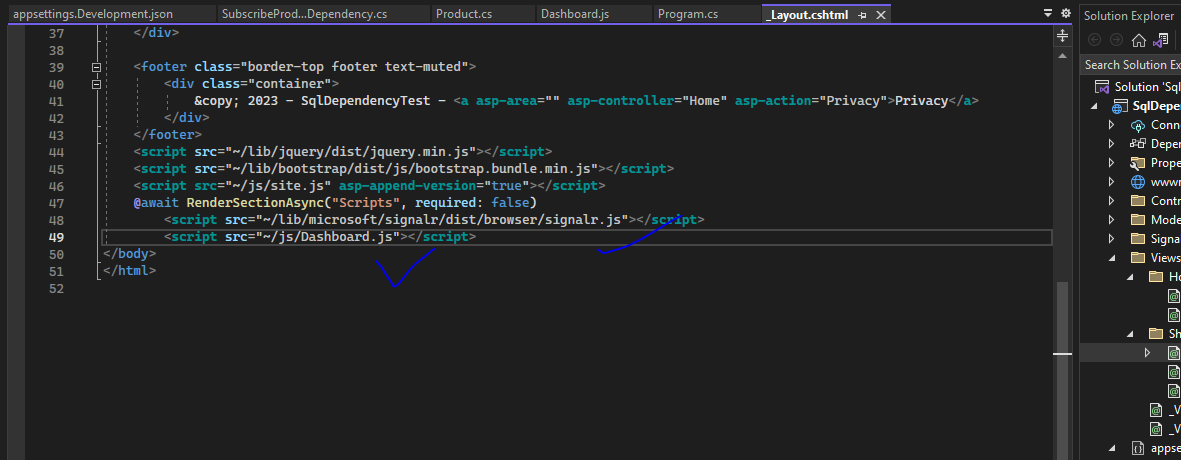
Add connection string

"ConnectionStrings": {

"DefaultConnection": "Data Source=(localdb)\\ProjectsV13;Initial Catalog=ProductsDB;Integrated Security=True"

}

In layout.cstml add the js file



Create table

CREATE TABLE [dbo].[Product](

[Id] [int] IDENTITY(1,1) primary key ,

[Name] [varchar](50) NOT NULL,

[Category] [varchar](50) NOT NULL,

[Price] [decimal](10, 2) NOT NULL,

)

If you insert data by script it will pop up automatically . ie notification will be displayed automatically.

INSERT [dbo].[Product] [Name], [Category], [Price]) VALUES (N'Potato', N'Groceries', CAST(100.00 AS Decimal(10, 2)))