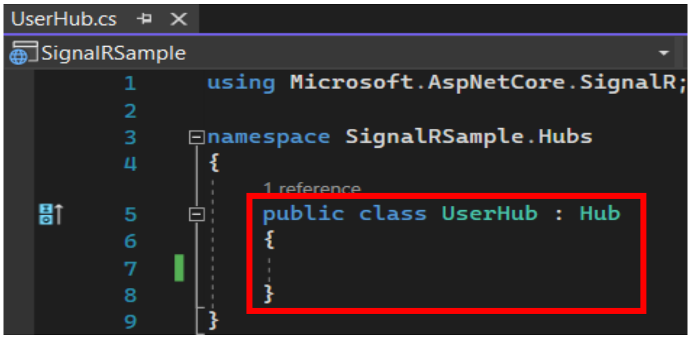
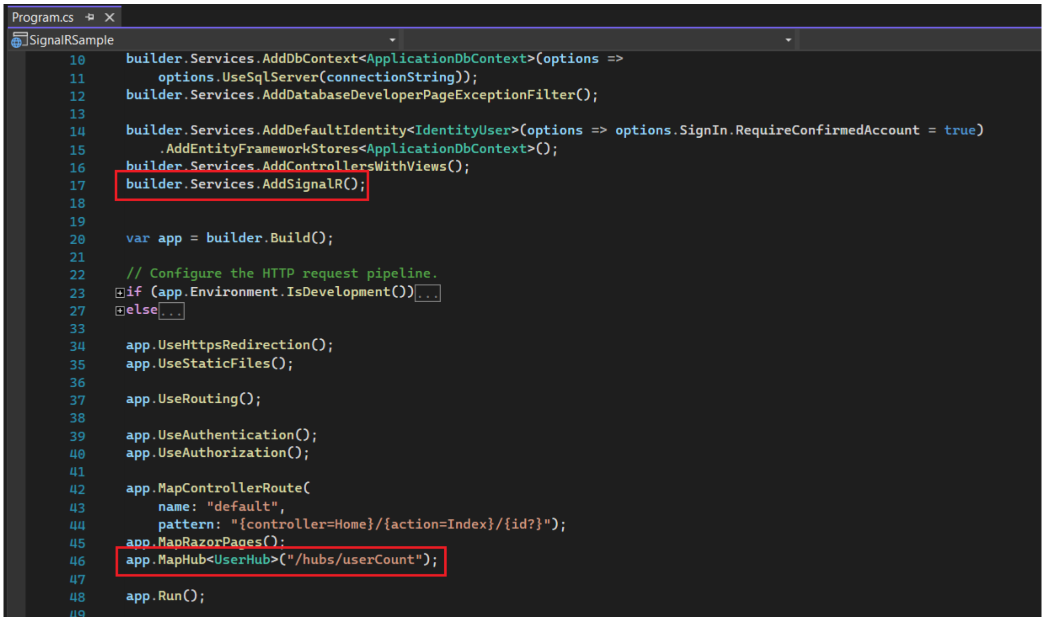
SignalR Basic Flow

1. Create SignalR Hub

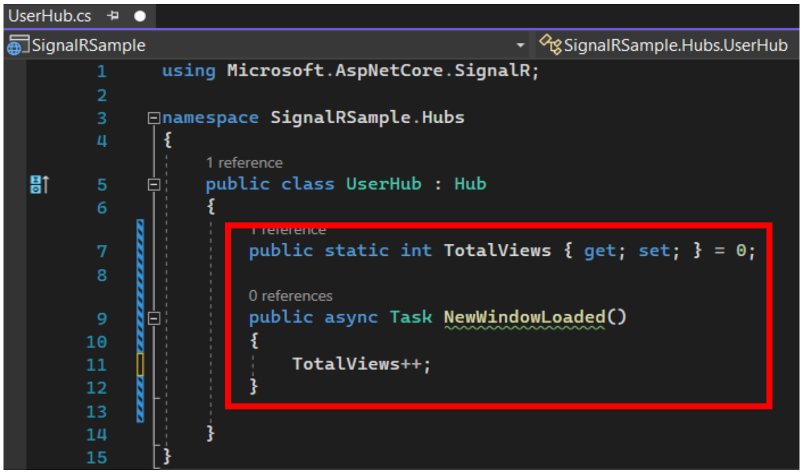


```
using Microsoft.AspNetCore.SignalR;

namespace SignalRSample.Hubs
{
    public class UserHub : Hub
    {

    }
}
```



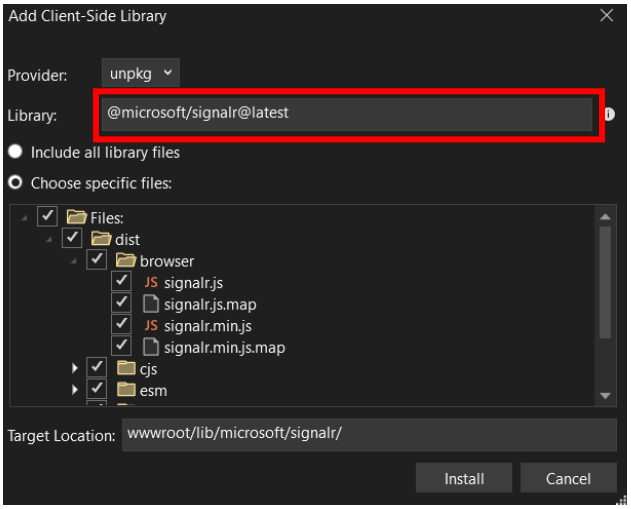
```
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(connectionString));
builder.Services.AddDatabaseDeveloperPageExceptionFilter();

builder.Services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
    .AddEntityFrameworkStores<ApplicationDbContext>();
builder.Services.AddControllersWithViews();
builder.Services.AddSignalR();


var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())...
else...

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();
app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
app.MapRazorPages();
app.MapHub<UserHub>("/hubs/userCount");

app.Run();
```

2. Add Methods to Hub



```
using Microsoft.AspNetCore.SignalR;

namespace SignalRSample.Hubs
{
    public class UserHub : Hub
    {
        public static int TotalViews { get; set; } = 0;

        public async Task NewWindowLoaded()
        {
            TotalViews++;
        }

    }
}
```

3. Add Client side SignalR



Provider: unpkg

Library: @microsoft/signalr@latest

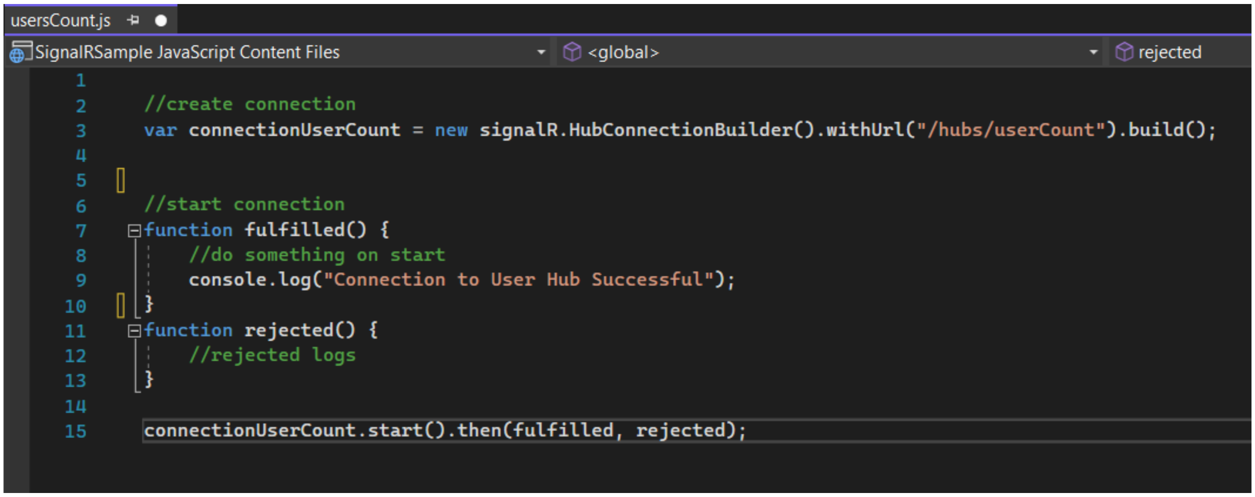
○ Include all library files

● Choose specific files:

- ☑ Files:
  - ☑ dist
    - ☑ browser
      - ☑ signalr.js
      - ☑ signalr.js.map
      - ☑ signalr.min.js
      - ☑ signalr.min.js.map
    - ☑ cjs
    - ☑ esm

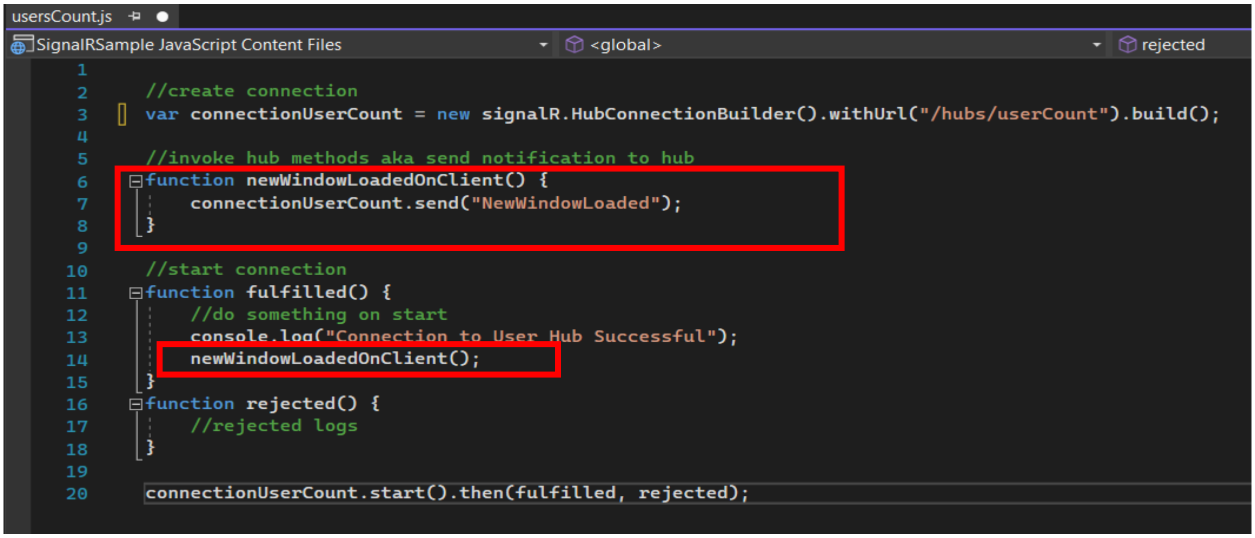
Target Location: wwwroot/lib/microsoft/signalr/

4. Connect to SignalR Hub from Client JS



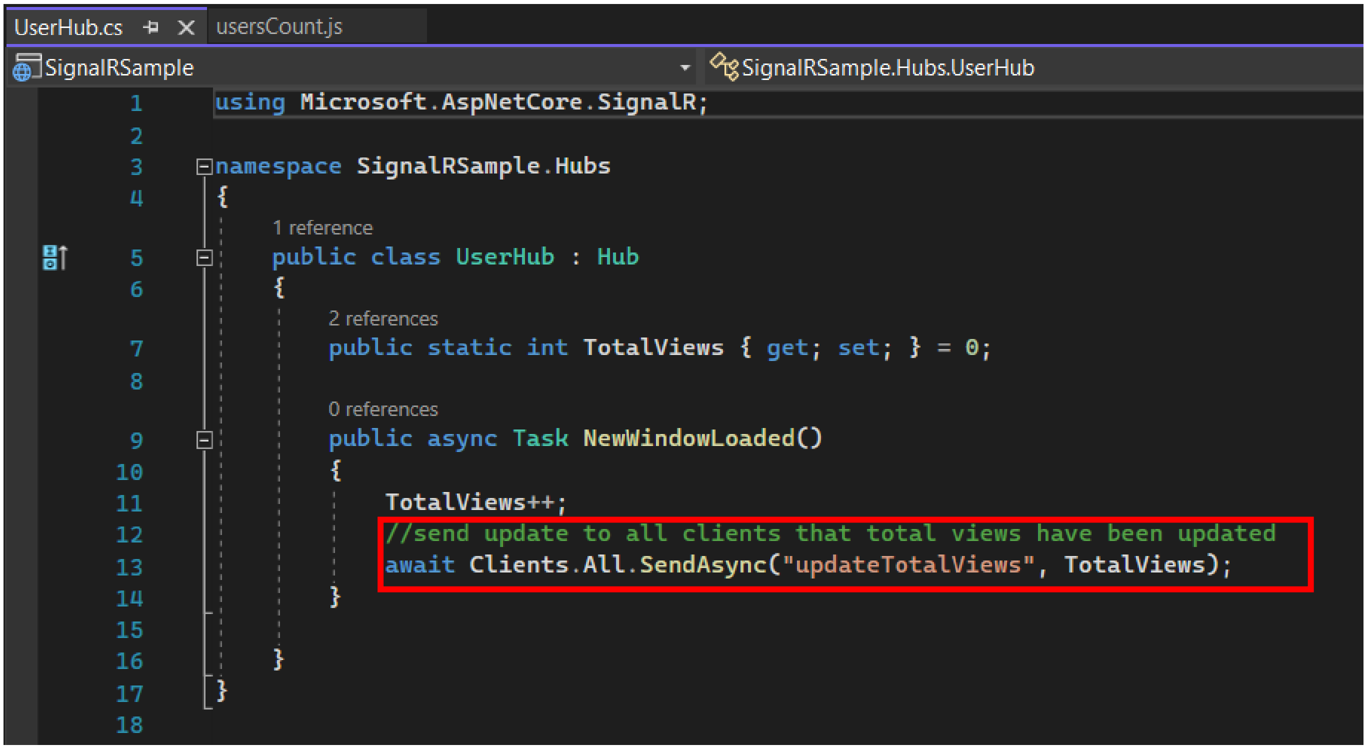
```

//create connection
var connectionUserCount = new signalR.HubConnectionBuilder().withUrl("/hubs/userCount").build();


//start connection
function fulfilled() {
    //do something on start
    console.log("Connection to User Hub Successful");
}
function rejected() {
    //rejected logs
}

connectionUserCount.start().then(fulfilled, rejected);
```

5. Call SignalR Hub method



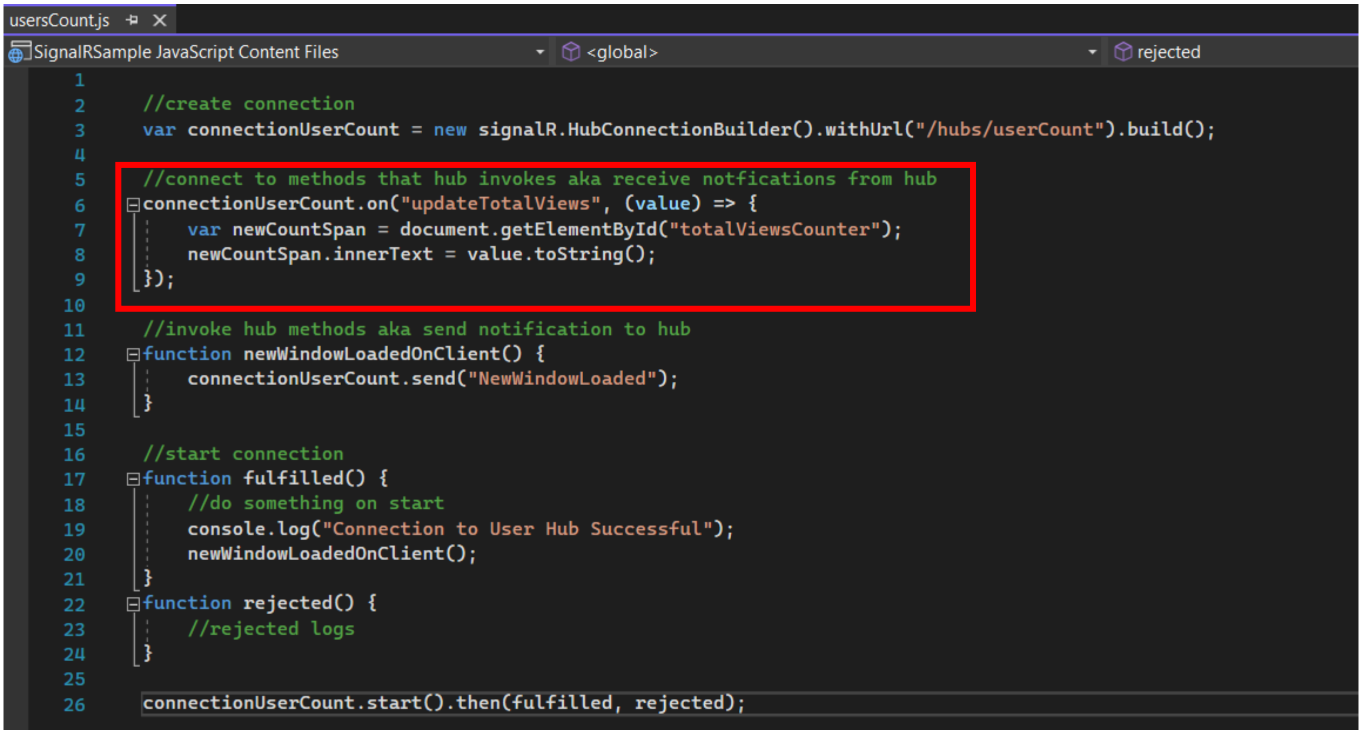
```

//create connection
var connectionUserCount = new signalR.HubConnectionBuilder().withUrl("/hubs/userCount").build();

//invoke hub methods aka send notification to hub
function newWindowLoadedOnClient() {
    connectionUserCount.send("NewWindowLoaded");
}

//start connection
function fulfilled() {
    //do something on start
    console.log("Connection to User Hub Successful");
    newWindowLoadedOnClient();
}
function rejected() {
    //rejected logs
}

connectionUserCount.start().then(fulfilled, rejected);
```

6. SignalR Hub invokes method in Client JS to notify clients.



```csharp
using Microsoft.AspNetCore.SignalR;

namespace SignalRSample.Hubs
{
    public class UserHub : Hub
    {
        public static int TotalViews { get; set; } = 0;

        public async Task NewWindowLoaded()
        {
            TotalViews++;
            //send update to all clients that total views have been updated
            await Clients.All.SendAsync("updateTotalViews", TotalViews);
        }

    }
}
```

7. Client receives update from SignalR hub and performs action



```javascript

//create connection
var connectionUserCount = new signalR.HubConnectionBuilder().withUrl("/hubs/userCount").build();

//connect to methods that hub invokes aka receive notfications from hub
connectionUserCount.on("updateTotalViews", (value) => {
    var newCountSpan = document.getElementById("totalViewsCounter");
    newCountSpan.innerText = value.toString();
});

//invoke hub methods aka send notification to hub
function newWindowLoadedOnClient() {
    connectionUserCount.send("NewWindowLoaded");
}

//start connection
function fulfilled() {
    //do something on start
    console.log("Connection to User Hub Successful");
    newWindowLoadedOnClient();
}
function rejected() {
    //rejected logs
}

connectionUserCount.start().then(fulfilled, rejected);
```