**Learn 100**

[[Client]] → Create File : libman.json

[wwwroot] → Update File : index.html

**Learn 200**

[Properties] → Update File : launchSettings.json

[Shared] → Delete File : SurveyPrompt.razor

[Pages] → Delete File : Counter.razor

[Pages] → Delete File : FetchData.razor

[wwwroot] → Delete Folder : sample-data

[Pages] → Update File : Index.razor

[Pages] → Create File : Page1.razor

[Pages] → Create File : Page2.razor

[Pages] → Create File : Page3.razor

[Pages] → Create File : Page4.razor

[Pages] → Create File : Page5.razor

[Shared] → Update File : NavMenu.razor

Link to Pages…

[Pages] → Create Folder : Authentication

[Authentication] → Create File : Login.razor

[Authentication] → Create File : Logout.razor

[Shared] → Update File : MainLayout.razor

Link to Login and Logout

**Learn 300**

[Properties] → Update File : launchSettings.json

**Resources:**

[ASP.NET Core Blazor authentication and authorization](https://learn.microsoft.com/en-us/aspnet/core/blazor/security/webassembly/?view=aspnetcore-7.0)

<https://learn.microsoft.com/en-us/aspnet/core/blazor/security>

Secure ASP.NET Core Blazor WebAssembly

https://learn.microsoft.com/en-us/aspnet/core/blazor/security/WebAssembly

ASP.NET Core Blazor state management

https://learn.microsoft.com/en-us/aspnet/core/blazor/state-management

ASP.NET Core Blazor Hybrid authentication and authorization

https://learn.microsoft.com/en-us/aspnet/core/blazor/hybrid/security

ASP.NET Core Blazor WebAssembly additional security scenarios

https://learn.microsoft.com/en-us/aspnet/core/blazor/security/webassembly/additional-scenarios

**NuGets:**

https://www.nuget.org/packages/Microsoft.AspNetCore.Components.Authorization

[[Client]] → Update File : Client.csproj

Add NuGets…

[[Client]] → Create Folder : Infrastructure

[Infrastructure] → Create File : CustomAuthenticationStateProvider.cs

First: Anonymous

[[Client]] → Update File : Program.cs

builder.Services.AddOptions();

// If we do not have Policy!

//builder.Services.AddAuthorizationCore();

// If we have Policy!

builder.Services.AddAuthorizationCore(options =>

{

options.AddPolicy("CanBuy", policy =>

policy.RequireClaim("Over21"));

options.AddPolicy("CanDelete", policy =>

policy.RequireRole("Administrator"));

});

builder.Services.AddScoped

<AuthenticationStateProvider,

CustomAuthenticationStateProvider>();

[Shared] → Create File : RedirectToLogin.razor

[[Client]] → Update File : App.razor

[[Client]] → Update File : \_Imports.razor

@using Microsoft.AspNetCore.Authorization; → For [Authorize]

@using Microsoft.AspNetCore.Components.Authorization → For App.razor

Breakpoints:

[Infrastructure] → Create File: CustomAuthenticationStateProvider.cs

[Shared] → Create File: RedirectToLogin.razor

[Pages] → Update File : Page2.razor

@attribute [Authorize]

[Pages] → Update File : Page3.razor

@attribute [Authorize(Roles = "Administrator")]

[Pages] → Update File : Page4.razor

<AuthorizeView></AuthorizeView>