

Fixing sign out

(Demo prep)

-

Fixing the sign out experience

As we saw in the previous lesson, the sign out experience is broken and this is because of a few changes we are missing in our Identity microservice. So let's fix it.

In Play.Identity

1. Add PostLogoutRedirectUri to appsettings.Development.json:

```
"IdentityServerSettings": {
  "Clients": [
    {
      "ClientId": "frontend",
      ...
      "RedirectUri": [
        "http://localhost:3000/authentication/login-callback"
      ],
      "PostLogoutRedirectUri": [
        "http://localhost:3000/authentication/logout-callback"
      ],
      "AllowedScopes": [
        ...
      ]
    }
  ]
}
```

2. Install code generation NuGet packages:

```
dotnet add package Microsoft.EntityFrameworkCore.Design
dotnet add package Microsoft.EntityFrameworkCore.SqlServer
```

3. Generate the logout page

```
dotnet aspnet-codegenerator identity --files "Account.Logout"
```

4. Delete Areas\Identity\Data directory

5. Delete Areas\Identity\IdentityHostingStartup.cs

6. Remove EntityFramework package references from csproj

7. Remove ConnectionStrings from appsettings.json
8. Update Logout.cshtml.cs:

```
public class LogoutModel : PageModel
{
    private readonly SignInManager<ApplicationUser> _signInManager;
    private readonly ILogger<LogoutModel> _logger;
    private readonly IdentityServerInteractionService interaction;

    public LogoutModel(
        SignInManager<ApplicationUser> signInManager,
        ILogger<LogoutModel> logger,
        IdentityServerInteractionService interaction)
    {
        _signInManager = signInManager;
        _logger = logger;
        this.interaction = interaction;
    }

    public async Task<ActionResult> OnGet(string logoutId)
    {
        var context = await interaction.GetLogoutContextAsync(logoutId);
        if (context?.ShowSignoutPrompt == false)
        {
            return await this.OnPost(context.PostLogoutRedirectUri);
        }

        return Page();
    }

    public async Task<ActionResult> OnPost(string returnUrl = null)
    {
        await _signInManager.SignOutAsync();
        _logger.LogInformation("User logged out.");
        if (returnUrl != null)
        {
            return Redirect(returnUrl);
        }
        else
        {
            return RedirectToPage();
        }
    }
}
```

```
}
```

9. Place breakpoint in OnGet method
10. F5 Identity microservice to start debugging
11. Try Log out again in front-end
12. Notice no prompt and we are back in front-end

In the next lesson we will explore how the updated front-end portal is taking advantage of OpenID Connect ID Tokens to secure and dynamically render UI elements based on the claims of the signed in user.