以.net core为例

认证服务器配置

// 客户端1

new Client {

    ClientId = "mvc1",

        ClientName = "客户端1",

        AllowedGrantTypes = GrantTypes.HybridAndClientCredentials,

        ClientSecrets = {

            new Secret ("secret".Sha256 ())

        },

        // 登录地址

        RedirectUris = { "http://localhost:5002/signin-oidc" },

        // 登出回调地址

        PostLogoutRedirectUris = { "http://localhost:5002/signout-callback-oidc" },

        // 联合登出地址

        FrontChannelLogoutUri = "http://localhost:5002/Account/UnionLogout",

        AllowedScopes = {

            IdentityServerConstants.StandardScopes.OpenId,

            IdentityServerConstants.StandardScopes.Profile,

            "api1"

        },

        AllowOfflineAccess = true

},

// 客户端2

new Client {

    ClientId = "mvc2",

        ClientName = "客户端2",

        AllowedGrantTypes = GrantTypes.HybridAndClientCredentials,

        ClientSecrets = {

            new Secret ("secret".Sha256 ())

        },

        // 登录地址

        RedirectUris = { "http://localhost:6000/signin-oidc" },

        // 登出地址

        PostLogoutRedirectUris = { "http://localhost:6000/signout-callback-oidc" },

        // 联合登出地址

        FrontChannelLogoutUri = "http://localhost:6000/Account/UnionLogout",

        AllowedScopes = {

            IdentityServerConstants.StandardScopes.OpenId,

            IdentityServerConstants.StandardScopes.Profile,

            "api1"

        },

},

客户端1配置（端口5002）

services.AddAuthentication (options => {

        options.DefaultScheme = "Cookies"; // 默认认证使用cookie

        options.DefaultChallengeScheme = "oidc"; // 登录跳转使用oidc

    })

    .AddCookie ("Cookies") // 添加cookie认证方式

    .AddOpenIdConnect ("oidc", options => // 添加oidc认证方式

        {

            options.SignInScheme = "Cookies";

            options.Authority = "http://localhost:5000";

            options.RequireHttpsMetadata = false;

            options.ClientId = "mvc";

            options.ClientSecret = "secret";

            options.ResponseType = "code id\_token";

            options.SaveTokens = true;

            options.GetClaimsFromUserInfoEndpoint = true;

            options.Scope.Add ("api1");

            options.Scope.Add ("offline\_access");

        });

客户端2配置（端口6000）

services.AddAuthentication (options => {

        options.DefaultScheme = "Cookies"; // 默认认证使用cookie

        options.DefaultChallengeScheme = "oidc"; // 默认登录跳转使用oidc

    })

    .AddCookie ("Cookies") // 添加cookie认证方式

    .AddOpenIdConnect ("oidc", options => // 添加oidc认证方式

        {

            options.SignInScheme = "Cookies";

            options.Authority = "http://localhost:5000";

            options.RequireHttpsMetadata = false;

            options.ClientId = "mvc";

            options.ClientSecret = "secret";

            options.ResponseType = "code id\_token";

            options.SaveTokens = true;

            options.GetClaimsFromUserInfoEndpoint = true;

            options.Scope.Add ("api1");

            options.Scope.Add ("offline\_access");

        });

客户端（1和2使用相同的原理）登录方法

配置[options.DefaultChallengeScheme = "oidc";]指明了未授权的处理方式，所以我们没有登录方法

客户端（1和2使用相同的原理）登出方法

public async Task Logout () {

    await HttpContext.SignOutAsync ("Cookies"); // 执行cookie认证的登出流程

    await HttpContext.SignOutAsync ("oidc"); // 执行oidc认证的登出流程

}

客户端（1和2使用相同的原理）联合登出方法，认证服务器登出会回调该方法

public class AccountController : Controller {

    public async Task UnionLogout () {

        if (User?.Identity.IsAuthenticated == true) {

            await HttpContext.SignOutAsync ("Cookies");

        }

    }

}

关于联合登出，IdentityServer给出的解决方案解析

LogoutRequest logout = await \_interaction.GetLogoutContextAsync(logoutId);

LogoutRequest的属性：

SignOutIFrameUrl：联合登出iframe url

前端解析

前端通过iframe载入SignOutIFrameUrl

<iframe width="0" height="0" class="signout" src="http://localhost:5000/connect/endsession/callback?endSessionId=CfDJ8MjbRQK-dj1KuWho4RCkRWGAelvl6lAvN\_7HoDHOSbvTKZ0wpqzBsemo4K\_bX8LTYWeHpa6lnQOkUsARMI5d39XG6UsrZto6dbh\_GwYAWmZn3ztfQz6F\_GEd2Vhv0gn7CWLrIPgc4E-mZCNZd9jc2S-16EGQWiTcbS5TQt6bn-pIxJq5h4\_v15W95Hl7hfAZpXxby1dkHikdlp2ht7KkJX2gh6xemHLRjmv6PJ-DGsTojCZFjeq3x8eaXb8qF89NRS76Pq6vShm6A3alup3FW2rLWc-2xLHvGibbi6TraBha"></iframe>

SignOutIFrameUrl指向的iframe的内容如下

// 客户端2的联合登出地址

<iframe src="http://localhost:6000/Account/UnionLogout?sid=95e59530f0f942965dc77e940e3e48d8&amp;iss=http%3A%2F%2Flocalhost%3A5000"></iframe>

// 客户端1的联合登出地址

<iframe src="http://localhost:5002/Account/UnionLogout?sid=95e59530f0f942965dc77e940e3e48d8&amp;iss=http%3A%2F%2Flocalhost%3A5000"></iframe>