## **OpenID Connect**

OIDC在OAuth的基础上扩展了一些新的概念，避免了OAuth中的很多误区：

****ID Tokens****

OpenID Connect Id Token是一个签名的JSON Web Token（JWT：RFC7519），它包含一组关于用户身份的声明（claim），如：用户的标识（sub）、颁发令牌的提供程序的标识符（iss）、创建此标识的Client标识（aud），还包含token的有效期以及其他相关的上下文信息。

由于ID Token使用的是JWT签名，客户端可以直接解析出Token中的内容而无需依赖外部服务，因此我们可以使用ID Token来做身份认证，而不需要使用access\_token。不过，OIDC为了保持于OAuth的兼容，会同时提供Id token和access\_token。

****UserInfo Endpoint****

OIDC还提供了一个包含当前用户信息的标准的受保护的资源。UserInfo Endpoint不是身份认证的一部分，而是提供附加的标识信息，它提供了一组标准化的属性：比如profile、email、phone和address。

OIDC中定义了一个特殊的openid scope，并且是必须的，它包含对Id token和UserInfo Endpoint的访问权限。

****Hybrid Flow****

混合流模式是授权码模式与隐式模式的组合，在一次请求中可以同时获取Code和ID Token，ResponseType可以是：code id\_token, code id\_token token 和 code token。

简单的介绍了一下OIDC的基本概念（网上对于OIDC的介绍很多，本文就不多叙述），下面就来介绍一下 ASP.NET Core 中的OIDC认证。

### **示例**

首先添OpenIdConnect的Nuget包引用：

dotnet add package Microsoft.AspNetCore.Authentication.OpenIdConnect --version 2.0.0

OIDC的注册其实要比OAuth还简单些：

services.AddAuthentication(options =>

{

options.DefaultAuthenticateScheme = CookieAuthenticationDefaults.AuthenticationScheme;

options.DefaultSignInScheme = CookieAuthenticationDefaults.AuthenticationScheme;

options.DefaultChallengeScheme = OpenIdConnectDefaults.AuthenticationScheme;

})

.AddCookie()

.AddOpenIdConnect(o =>

{

o.ClientId = "oidc.hybrid";

o.ClientSecret = "secret";

// 若不设置Authority，就必须指定MetadataAddress

o.Authority = "https://oidc.faasx.com/";

// 默认为Authority+".well-known/openid-configuration"

//o.MetadataAddress = "https://oidc.faasx.com/.well-known/openid-configuration";

o.RequireHttpsMetadata = false;

// 使用混合流

o.ResponseType = OpenIdConnectResponseType.CodeIdToken;

// 是否将Tokens保存到AuthenticationProperties中

o.SaveTokens = true;

// 是否从UserInfoEndpoint获取Claims

o.GetClaimsFromUserInfoEndpoint = true;

// 在本示例中，使用的是IdentityServer，而它的ClaimType使用的是JwtClaimTypes。

o.TokenValidationParameters.NameClaimType = "name"; //JwtClaimTypes.Name;

// 以下参数均有对应的默认值，通常无需设置。

//o.CallbackPath = new PathString("/signin-oidc");

//o.SignedOutCallbackPath = new PathString("/signout-callback-oidc");

//o.RemoteSignOutPath = new PathString("/signout-oidc");

//o.Scope.Add("openid");

//o.Scope.Add("profile");

//o.ResponseMode = OpenIdConnectResponseMode.FormPost;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*相关事件\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 未授权时，重定向到OIDC服务器时触发

//o.Events.OnRedirectToIdentityProvider = context => Task.CompletedTask;

// 获取到授权码时触发

//o.Events.OnAuthorizationCodeReceived = context => Task.CompletedTask;

// 接收到OIDC服务器返回的认证信息（包含Code, ID Token等）时触发

//o.Events.OnMessageReceived = context => Task.CompletedTask;

// 接收到TokenEndpoint返回的信息时触发

//o.Events.OnTokenResponseReceived = context => Task.CompletedTask;

// 验证Token时触发

//o.Events.OnTokenValidated = context => Task.CompletedTask;

// 接收到UserInfoEndpoint返回的信息时触发

//o.Events.OnUserInformationReceived = context => Task.CompletedTask;

// 出现异常时触发

//o.Events.OnAuthenticationFailed = context => Task.CompletedTask;

// 退出时，重定向到OIDC服务器时触发

//o.Events.OnRedirectToIdentityProviderForSignOut = context => Task.CompletedTask;

// OIDC服务器退出后，服务端回调时触发

//o.Events.OnRemoteSignOut = context => Task.CompletedTask;

// OIDC服务器退出后，客户端重定向时触发

//o.Events.OnSignedOutCallbackRedirect = context => Task.CompletedTask;

});

### **运行流程**

#### **登录**

来探索一下OIDC的运行流程，首先运行，点击一下我的信息：

请求：

GET http://localhost:5002/profile HTTP/1.1

响应：

HTTP/1.1 302 Found

Location: https://oidc.faasx.com/connect/authorize?client\_id=oidc.hybrid&redirect\_uri=http://localhost:5002/signin-oidc&response\_type=code id\_token&scope=openid profile&response\_mode=form\_post&nonce=636428853279287956.N2IxZmFlZDgtNDNmZC00OTRmLTljMWItNTVjMmQwOTVjNTQ3MWUyMDcxNTctZDg4Yy00MDRiLThmNmQtYjE1YTdjOGE4MThm&state=CfDJ8B4XRZETkRhMt3mT9VduB8LSACJO9seruKlM3kYPxaRyWcUSt0BvPMd6RUGAiay8qraTWLdMh9B3ClRJDE-BtMRYTmzGJSHegueIW-fyq2G9TpUtSQCd23BxAYrdB4SeGQte2IXaQ82cKMz-aSHQ7TTzhPO\_fgDtIVlwDJBtwgKQzEkEyyLsfH2DHxwr\_Ojn3M-uRHId2bi9RF2gR\_1hqoTdYlv-CZodFKuUGSMCqJO4cZLsuuAb-PrSnamz7h7MOpPixIOgQq5gd25sxF8avpSTsoT5HbU2fCiqX7g3rbCLzMG-rTnDftN8uZRiqc-JcyGkLPGIoj-FLNoW\_yfZbGk&x-client-SKU=ID\_NET&x-client-ver=2.1.4.0

Set-Cookie: .AspNetCore.OpenIdConnect.Nonce.CfDJ8B4XRZETkRhMt3mT9VduB8IHI9Q\_BeT6uPrlI72UUqej78UfqAdiczZsPOxF3Gy7bSm7Swh9Jh0\_haVi-UnQxUTGq-9xQLcaMX-PuXMjf6-6sLqc15NUgoLQ1w3KWqjkt7-3NlYW4qka6LqDPRtWJxT7vICtPhjx8ecNWtW\_ijqBg\_W8osmZLvFGS3PzPipP1UF14AkaIp48dFV1qNqt67Yta8ebXH7SHGkUhfcA5R-O0B8t-Q7sWL4NTN8AdKQF02HMqs\_duf-4FLf2p7Rbpsc=N; expires=Fri, 06 Oct 2017 11:30:27 GMT; path=/signin-oidc; httponly

Set-Cookie: .AspNetCore.Correlation.OpenIdConnect.BqW1filmAVCStL92ghXFQZBoLQjG5-Gl\_m60zxCW9BA=N; expires=Fri, 06 Oct 2017 11:30:27 GMT; path=/signin-oidc; httponly

重定向地址中包含如下几个参数：

****client\_id**** 客户端标识，对应于OpenIdConnectOptions.ClientId。

****redirect\_uri**** 回调地址，对应于OpenIdConnectOptions.CallbackPath。

****response\_type**** 授权类型，对应于OpenIdConnectOptions.ResponseType。

****scope**** 权限范围，对应于OpenIdConnectOptions.Scope。

****response\_mode**** 响应模式，对应于OpenIdConnectOptions.ResponseMode，表示OIDC服务器来跳转到我们的应用时传递参数的方式。

****nonce**** OIDC服务器会在identity token中包含此参数，在认证时与Cookie中的.AspNetCore.OpenIdConnect.Nonce.xxx对比验证。

****state**** 用于保存状态，会原封不动地返回，在 ASP.NET Core 中，用AuthenticationProperties对象来表示。

****x-client-SKU/x-client-ver**** IdentityServer附加信息。

在OIDC服务器登录成功后，connect/authorize输出的是一个自动提交的表单：

<form method='post' action='http://localhost:5002/signin-oidc'>

<input type='hidden' name='code' value='eb53860906da276a1bb5318c5d539db085c6ca1fd3467da9d918aa7524f20f63' />

<input type='hidden' name='id\_token' value='eyJhbGciOiJSUzI1NiIsImtpZCI6IjdlYzk5MjVlMmUzMTA2NmY2ZmU2ODgzMDRhZjU1ZmM0IiwidHlwIjoiSldUIn0.eyJuYmYiOjE1MDcyODg1MjcsImV4cCI6MTUwNzI4ODgyNywiaXNzIjoiaHR0cDovL29pZGMuZmFhc3guY29tIiwiYXVkIjoib2lkYy5oeWJyaWQiLCJub25jZSI6IjYzNjQyODg1MzI3OTI4Nzk1Ni5OMkl4Wm1GbFpEZ3RORE5tWkMwME9UUm1MVGxqTVdJdE5UVmpNbVF3T1RWak5UUTNNV1V5TURjeE5UY3RaRGc0WXkwME1EUmlMVGhtTm1RdFlqRTFZVGRqT0dFNE1UaG0iLCJpYXQiOjE1MDcyODg1MjcsImNfaGFzaCI6ImJNT1FNVGdDV3VMX25EbHo5MDU4M3ciLCJzaWQiOiJjZTVjODg2ZmFhZDFiNTc4MDVkNzExYjkzZTliYWQ0ZiIsInN1YiI6IjAwMSIsImF1dGhfdGltZSI6MTUwNzI4ODM5NSwiaWRwIjoibG9jYWwiLCJhbXIiOlsicHdkIl19.BBn0SigvOW9USk-Mi1WP\_lJPWI9I06gsuomhqp69Ip5y3kqFyiBCVanzULsR4fBa0tOOBOtcPJzAfivzLqsMRwW1QfRamfVIlXuuzcRsR8WP1pFxvFPekwKi-D6-RLmmQzUT-\_78WvboiAu\_dZtwe0cm4ZLDCJH6LLCPs2xXTHYuNI7YoyAgeGKDAhWle0VrsbdlrcubPPgQFfFXPdLDInnLr8eEMpUZ7nru0FJgxm3Ah4hGXPKMud8jhLUMXDcSaKseL8tDgxIowmhpXOknU-y9x5FlrZUFOReDxaBZe7DG5V0xsPrdhMxMkZQbHHz8cJoaYrqcwHClm8rScEPxVA' />

<input type='hidden' name='scope' value='openid profile' />

<input type='hidden' name='state' value='CfDJ8B4XRZETkRhMt3mT9VduB8LSACJO9seruKlM3kYPxaRyWcUSt0BvPMd6RUGAiay8qraTWLdMh9B3ClRJDE-BtMRYTmzGJSHegueIW-fyq2G9TpUtSQCd23BxAYrdB4SeGQte2IXaQ82cKMz-aSHQ7TTzhPO\_fgDtIVlwDJBtwgKQzEkEyyLsfH2DHxwr\_Ojn3M-uRHId2bi9RF2gR\_1hqoTdYlv-CZodFKuUGSMCqJO4cZLsuuAb-PrSnamz7h7MOpPixIOgQq5gd25sxF8avpSTsoT5HbU2fCiqX7g3rbCLzMG-rTnDftN8uZRiqc-JcyGkLPGIoj-FLNoW\_yfZbGk' />

<input type='hidden' name='session\_state' value='g0m\_0W2scFKHxVQP\_8jdNt48r2XV8wDlF\_\_-ST9aYtk.28e1d34baad6d122a92c667329084600' />

</form>

<script>(function(){document.forms[0].submit();})();</script>

如上，可以看到，表单中包含有id\_token，因为我们使用的是code id\_token类型，然后便进入到了我们应用程序的OIDC认证逻辑中：

当我们做身份验证时，可能会需要更详细的用户Claims，可以将GetClaimsFromUserInfoEndpoint设置为True，使用UserInfoEndpoint返回的信息来重新创建ClaimsPrincipal对象：

#### **退出**

退出分为两种，一种是本地的退出，并不会使OIDC服务器退出，只需要简单的调用CookieHandler的SignOutAsync即可，无需多说。

而远程退出则会同时退出本地应用和OIDC服务器，大致逻辑是先跳转到OIDC服务器，退出后，OIDC服务器会回调本地应用，完成本地的退出，该回调地址是通过OpenIdConnectOptions.RemoteSignOutPath指定的。

当我们点击退出时，首先执行OpenIdConnectHandler的SignOutAsync方法，Http报文如下：

请求报文：

GET http://localhost:5002/signout-remote HTTP/1.1

Cookie: .AspNetCore.Cookies=xxx

响应报文：

HTTP/1.1 302 Found

Location: http://oidc.faasx.com/connect/endsession?post\_logout\_redirect\_uri=http://localhost:5002/signout-callback-oidc&id\_token\_hint=eyJhbGciOiJSUzI1NiIsImtpZCI6IjdlYzk5MjVlMmUzMTA2NmY2ZmU2ODgzMDRhZjU1ZmM0IiwidHlwIjoiSldUIn0.eyJuYmYiOjE1MDc0MjgzMTAsImV4cCI6MTUwNzQyODYxMCwiaXNzIjoiaHR0cDovL29pZGMuZmFhc3guY29tIiwiYXVkIjoib2lkYy5oeWJyaWQiLCJub25jZSI6IjYzNjQzMDI1MTExOTUyNjYxNS5NVFU0WXpNMFl6RXRZemcwWmkwME5UWTFMVGcyTW1ZdE5XUmtZbUZtTUdabU5UWXlaalF6TkRReE9HWXROalEwWXkwMFpHTmtMVGhqTlRFdE1XWTROV0V3TW1NM09UZG0iLCJpYXQiOjE1MDc0MjgzMTAsImF0X2hhc2giOiJQSHRvd1JrUHhYLWNsclBzRnYxMkpnIiwic2lkIjoiMzFmMWJmMTA1MTU4MmZjMGE3ZTZhZjFjY2I0Y2RlMzUiLCJzdWIiOiIwMDEiLCJhdXRoX3RpbWUiOjE1MDc0MjgzMTAsImlkcCI6ImxvY2FsIiwiYW1yIjpbInB3ZCJdfQ.hIxkDhsx\_WE6IxM68O7uqkqdQquXXnOtxhlnrYiBJuU7Ex\_aApVXoKUdHS8HMx1nLswntr6SRsrygyMJnGMdzP5JutGsmfO\_i1WYGqk3BlTD7ry0wfBd\_U9OaVFcJhcVZq4q5u3SA47Wxqex9vifiHrTBQFT\_l6JqpevRLn-y91IxTl9rnXKfrHowhPsHJjdLzda3Lyj0wWWtb2N\_ng19mRChmDd4RXucP9mBHdQDyLZtvIJ5iIzV4pqtL7VCylFzV4RBLbCzUeuRnHI3E\_MqTaGWvVyFbUAKpr55TmVoc6lAOS4ie4CPzilR52KLWZ4l9eQh-WeIOA3NBgzHlJigg&state=CfDJ8B4XRZETkRhMt3mT9VduB8IbG5c9um6S\_maiHLsMKlIFRVtKMyLuXfqgB6e1OhWWVVJDYbgedt6hmyi1ny2aMbKW-SgHTru73YezUAZpre2ELXM3trlnX3YW\_FTkcGE\_RUyaR3hQ3eEYFmMdgdZdf0M&x-client-SKU=ID\_NET&x-client-ver=2.1.4.0

其中id\_token\_hint为id\_token