[Robin van der Knaap](https://dev.to/robinvanderknaap)

[Robin van der Knaap](https://dev.to/robinvanderknaap)

Posted on Dec 11, 2020

**Setting up an Authorization Server with OpenIddict - Part VI - Refresh tokens**

[#dotnet](https://dev.to/t/dotnet)[#oauth2](https://dev.to/t/oauth2)[#openiddict](https://dev.to/t/openiddict)[#aspnet](https://dev.to/t/aspnet)

This article is part of a series called **Setting up an Authorization Server with OpenIddict**. The articles in this series will guide you through the process of setting up an OAuth2 + OpenID Connect authorization server on the the [ASPNET Core](https://dotnet.microsoft.com/apps/aspnet) platform using [OpenIddict](https://github.com/openiddict/openiddict-core).

* [Part I: Introduction](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid)
* [Part II: Create ASPNET project](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-ii-create-aspnet-project-4949)
* [Part III: Client Credentials Flow](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-iii-client-credentials-flow-55lp)
* [Part IV: Authorization Code Flow](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-iv-authorization-code-flow-3eh8)
* [Part V: OpenID Connect](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-v-openid-connect-a8j)
* **Part VI: Refresh tokens**

GitHub logo [robinvanderknaap](https://github.com/robinvanderknaap)/ [**authorization-server-openiddict**](https://github.com/robinvanderknaap/authorization-server-openiddict)

Authorization Server implemented with OpenIddict.

Access tokens typically have a short lifetime for security reasons. In this part we will enable the usage of refresh tokens. A refresh token allows an application to obtain a new access token without prompting the user.

**Enable refresh tokens**

First we need to enable the refresh token flow in Startup.cs:

options

.AllowAuthorizationCodeFlow()

.RequireProofKeyForCodeExchange()

.AllowClientCredentialsFlow()

.AllowRefreshTokenFlow();

In the token endpoint in the AuthorizationController we need to handle refresh token requests, just like we handled client credentials requests, and requests to exchange authorization codes for access tokens:

[HttpPost("~/connect/token")]

public async Task<IActionResult> Exchange()

{

...

else if (request.IsAuthorizationCodeGrantType())

{

...

}

else if (request.IsRefreshTokenGrantType())

{

// Retrieve the claims principal stored in the refresh token.

claimsPrincipal = (await HttpContext.AuthenticateAsync(OpenIddictServerAspNetCoreDefaults.AuthenticationScheme)).Principal;

}

...

}

Basically, we only need to retrieve the ClaimsPrincipal from the refresh token, and sign in again, which will return a new access token.

Finally, we need to allow the client to use refresh tokens:

Permissions =

{

OpenIddictConstants.Permissions.Endpoints.Authorization,

OpenIddictConstants.Permissions.Endpoints.Token,

OpenIddictConstants.Permissions.GrantTypes.AuthorizationCode,

OpenIddictConstants.Permissions.GrantTypes.ClientCredentials,

OpenIddictConstants.Permissions.GrantTypes.RefreshToken,

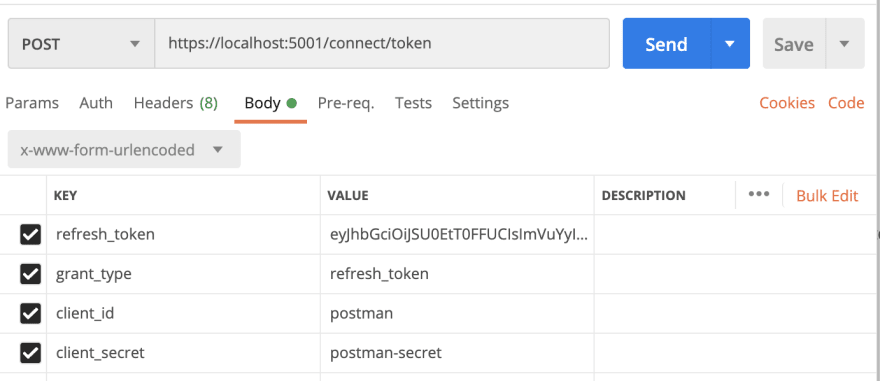
OpenIddictConstants.Permissions.Prefixes.Scope + "api",

OpenIddictConstants.Permissions.ResponseTypes.Code

}

If a client wants to use refresh tokens, it needs to request the offline\_access scope. Just like we requested the openid scope for identity tokens. Now, if we request a new access token with Postman, we will receive a refresh token together with the access token.

After that, we can use the refresh token to request a new access token with Postman. Send a POST request to /connect/token with a body containing the client credentials, grant type and of course the refresh token.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--Aeb59xMF--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/xoftb5an9yb50rwszwp8.png)

In the response you will find a new access token and also a new refresh token. So, refresh tokens are automatically rotated, which is more secure than reusing the same refresh token.

**Discussion (5)**

Subscribe

Top of Form





Bottom of Form

[capesean profile image](https://dev.to/capesean)

Sean Walsh

•[Jan 19 '21](https://dev.to/capesean/comment/1aiab)

Thanks Robin. I needed to add the following lines, to include the refresh token in the jwt, not sure if it's necessary in your codebase though:

[github.com/openiddict/openiddict-s...](https://github.com/openiddict/openiddict-samples/blob/ca2741f9019e0a9266d914cfb1755221a1bc4be1/samples/Imynusoph/Imynusoph.Server/Controllers/AuthorizationController.cs#L76-L83)

1 like[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669/comments/new/1aiab)

[capesean profile image](https://dev.to/capesean)

Sean Walsh

•[Jan 19 '21](https://dev.to/capesean/comment/1aiak)

Ah, never mind. I see you have it there:  
[github.com/robinvanderknaap/author...](https://github.com/robinvanderknaap/authorization-server-openiddict/blob/f29e8a87eeda8e5c1425171fca618c956df83286/AuthorizationServer/Controllers/AuthorizationController.cs#L56)

2 likes[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669/comments/new/1aiak)

[timothydowd profile image](https://dev.to/timothydowd)

Tim Dowd

•[Aug 24 '21](https://dev.to/timothydowd/comment/1heb2)

embassador, with these rocher your spoil us

2 likes[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669/comments/new/1heb2)

[hypervtechnics profile image](https://dev.to/hypervtechnics)

hypervtechnics

•[Jun 23 '21](https://dev.to/hypervtechnics/comment/1fjf7)

Great series! Thanks!

Would you make another article on how to leverage custom grants?

1 like[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669/comments/new/1fjf7)

[thiagofferreira profile image](https://dev.to/thiagofferreira)

Thiago Ferreira

•[Aug 19 '21](https://dev.to/thiagofferreira/comment/1hb7o)

How to implement the logout flow, where I need to return a redirect\_uri to the client, and also invalidate the id\_token?

1 like[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669/comments/new/1hb7o)

[Code of Conduct](https://dev.to/code-of-conduct) • [Report abuse](https://dev.to/report-abuse)