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

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

Posted on Dec 11, 2020 • Updated on Dec 15, 2020

**Setting up an Authorization Server with OpenIddict - Part I - Introduction**

[#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**
* [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](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-vi-refresh-tokens-5669)

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

Authorization Server implemented with OpenIddict.

Setting up an authorization server allows you to support token-based authentication and authorization. It also allows you to authenticate users for all your applications in one central place, Single Sign-On(SSO).

An authorization server can offer one or multiple authentication methods simultaneously, like local username-password but also external identity providers such as Google, Facebook or ADFS.

By implementing OAuth2 and OpenID Connect you are able to facilitate multiple authorization scenario's: Web applications, desktop applications, machine-to-machine communication and even authorization for living room devices. Both Oauth2 and OpenID Connect are industry standards.

**IdentityServer**

For over a decade the go-to project in .NET for implementing a secure token service and later OAuth2 + OpenID Connect was [IdentityServer](https://identityserver.io/). Lately, the creators/maintainers of IdentityServer [decided to dual license future versions of IdentityServer](https://leastprivilege.com/2020/10/01/the-future-of-identityserver/). Now unless you are working on an open-source project you will have to pay for a commercial license.

Although, paying for good software shouldn't be an issue for a lot of projects or companies, I went out to look for an alternative anyway.

**OpenIddict**

An alternative for IdentityServer is [OpenIddict](https://github.com/openiddict/openiddict-core). OpenIddict provides a solution to implement an OpenID Connect server in any ASP.NET Core 2.1, 3.1 and 5.0 application, and starting with OpenIddict 3.0, any ASP.NET 4.x or OWIN application too.

OpenIddict is little bit more low-level than IdentityServer. Identity Server gives you a running solution out-of-the-box, where for OpenIddict to work you need to implement some details yourself, like creating claim identities and setting up the correct endpoints.

In this article we will use OpenIddict to implement our Authorization Server. We will implement the [Client Credentials Flow](https://oauth.net/2/grant-types/client-credentials), the [Authorization Code Flow](https://oauth.net/2/grant-types/authorization-code) and setup [refresh tokens](https://oauth.net/2/grant-types/refresh-token/) as an example. We will also demonstrate how to leverage OpenID Connect to retrieve user information.

**OAuth2 and OpenID Connect**

OAuth2 and OpenID Connect can be confusing at times, at least to me it is.

It helped me to view an Authorization Server as just another web application. It's an application where you can login with a username-password combination or with some other external provider. Up to this point, nothing is different from any other web applications we have build.

After setting up authentication we can implement OAuth2 and OpenID Connect flows (this is where OpenIddict comes into play). These flows are leveraged by other applications (clients), basically asking 'can I do some stuff on behalf of this user?'.

To answer this question, users are first redirected to the Authorization Server by the clients. The user will authenticate himself if not already authenticated (SSO). After that, the user is asked if the client is allowed to make requests on behalf of the user. If yes, the external application receives an access token to make requests in the user's behalf.

Besides the access token (OAuth2), a separate identity token (OpenID Connect) can be issued. The identity token can be used by clients to extract user information.

**Next**

In the [next article](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-ii-create-aspnet-project-4949) we will start by creating a new ASPNET Core project and implement authentication as the first step toward a full-blown Authorization Server.

**Discussion (5)**

Subscribe

Top of Form





Bottom of Form

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

Mahmoud Alaskalany

•[Apr 3 '21](https://dev.to/mhmoudalaskalany/comment/1d3l9)

is it free for .net core 5

3 likes[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid/comments/new/1d3l9)

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

Robin van der Knaap

•[Apr 3 '21](https://dev.to/robinvanderknaap/comment/1d3li)

Yes

4 likes[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid/comments/new/1d3li)

[keatkeat87 profile image](https://dev.to/keatkeat87)

keatkeat87

•[Jun 25 '21](https://dev.to/keatkeat87/comment/1fknh)

Really good article and examples. It saves me so much time.Thank for sharing.

2 likes[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid/comments/new/1fknh)

[kev4ever profile image](https://dev.to/kev4ever)

Kevin van Schaijk

•[Apr 7 '21](https://dev.to/kev4ever/comment/1d72g)

Great article, very helpful!

2 likes[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid/comments/new/1d72g)

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

Benzara Tahar Benlahcene

•[May 10 '21](https://dev.to/benzaratahar/comment/1ebef)

Great serie of articles,just in time thanks

1 like[Reply](https://dev.to/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid#/robinvanderknaap/setting-up-an-authorization-server-with-openiddict-part-i-introduction-4jid/comments/new/1ebef)

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