Security Intro

There are many ways to handle security, authentication and authorization.

And it normally is a complex and "difficult" topic.

In many frameworks and systems just handling security and authentication takes a big amount of effort and code (in many cases it can be 50% or more of all the code written).

FastAPI provides several tools to help you deal with **Security** easily, rapidly, in a standard way, without having to study and learn all the security specifications.

But first, let's check some small concepts.

In a hurry?

If you don't care about any of these terms and you just need to add security with authentication based on username and password *right now*, skip to the next chapters.

OAuth2

OAuth2 is a specification that defines several ways to handle authentication and authorization.

It is quite an extensive specification and covers several complex use cases.

It includes ways to authenticate using a "third party".

That's what all the systems with "login with Facebook, Google, Twitter, GitHub" use underneath.

OAuth 1

There was an OAuth 1, which is very different from OAuth2, and more complex, as it included directly specifications on how to encrypt the communication.

It is not very popular or used nowadays.

OAuth2 doesn't specify how to encrypt the communication, it expects you to have your application served with HTTPS.

> **Tip**
>
> In the section about **deployment** you will see how to set up HTTPS for free, using Traefik and Let's Encrypt.

OpenID Connect

OpenID Connect is another specification, based on **OAuth2**.

It just extends OAuth2 specifying some things that are relatively ambiguous in OAuth2, to try to make it more interoperable.

For example, Google login uses OpenID Connect (which underneath uses OAuth2).

But Facebook login doesn't support OpenID Connect. It has its own flavor of OAuth2.

OpenID (not "OpenID Connect")

There was also an "OpenID" specification. That tried to solve the same thing as **OpenID Connect**, but was not based on OAuth2.

So, it was a complete additional system.

It is not very popular or used nowadays.

OpenAPI

OpenAPI (previously known as Swagger) is the open specification for building APIs (now part of the Linux Foundation).

FastAPI is based on **OpenAPI**.

That's what makes it possible to have multiple automatic interactive documentation interfaces, code generation, etc.

OpenAPI has a way to define multiple security "schemes".

By using them, you can take advantage of all these standard-based tools, including these interactive documentation systems.

OpenAPI defines the following security schemes:

- `apiKey`: an application specific key that can come from:
  - A query parameter.
  - A header.
  - A cookie.

- `http`: standard HTTP authentication systems, including:
  - `bearer`: a header `Authorization` with a value of `Bearer` plus a token. This is inherited from OAuth2.
  - HTTP Basic authentication.
  - HTTP Digest, etc.
- `oauth2`: all the OAuth2 ways to handle security (called "flows").
  - Several of these flows are appropriate for building an OAuth 2.0 authentication provider (like Google, Facebook, Twitter, GitHub, etc):
    - `implicit`
    - `clientCredentials`
    - `authorizationCode`
  - But there is one specific "flow" that can be perfectly used for handling authentication in the same application directly:
    - `password`: some next chapters will cover examples of this.
- `openIdConnect`: has a way to define how to discover OAuth2 authentication data automatically.
  - This automatic discovery is what is defined in the OpenID Connect specification.

> **Tip**
>
> Integrating other authentication/authorization providers like Google, Facebook, Twitter, GitHub, etc. is also possible and relatively easy.
>
> The most complex problem is building an authentication/authorization provider like those, but **FastAPI** gives you the tools to do it easily, while doing the heavy lifting for you.

FastAPI utilities

FastAPI provides several tools for each of these security schemes in the `fastapi.security` module that simplify using these security mechanisms.

In the next chapters you will see how to add security to your API using those tools provided by **FastAPI**.

And you will also see how it gets automatically integrated into the interactive documentation system.

Last update: September 11, 2022

Created: September 11, 2022