The Authorization Code Flow

0:00:00.5: Welcome to this lesson out of Learn Spring Security OAuth, where we're going to analyze what is arguably the most commonly used OAuth flow, the Authorization Code Flow. Let's begin with the high level actors here, the actors that will interact in this flow, starting from the end user, the resource owner. Its role is to explicitly grant access to the client for its protected resources. Now, there are many types of clients, but for the purposes of our discussion here, we're going to consider a traditional MVC application with a server-side component as well as a client-side component or a user agent. In our case, of course, that's going to be the browser. The authorization code flow involves redirects, as well as interactions with a user, as we'll see in a second.

0:00:52.6: All right. Let's jump right in and start discussing the flow. The very first step in the flow is when the client requests authorization to a protected resource, the browser will be redirected to the authorization endpoint of the authorization server. That request will include a redirection URI as well as the client ID, but notice no client credentials, as their confidentiality cannot be guaranteed on the client side on a non-confidential client.

0:01:23.5: Of course, the second step is the authorization server now starting the authentication process with the resource owner. The authorization server will also commonly ask the resource owner to grant or to deny the scopes that were requested by the client. Remember, OAuth is all about partial authorization. One of the core reasons why we're doing OAuth is exactly for this reason, is to give the client partial, limited access, not full access to the entire account.

0:01:57.3: All right. The next step is the redirection, and this is where things get interesting because this is what makes this particular flow different and more secure than the other OAuth flows, and that is that we're not getting an access token back from the authorization server. The access token is not issued yet at this point, because as we've talked about, this is all passing through the browser, through the user agent on the client side. Instead, the authorization server returns an intermediary code, the authorization code, hence the name of the flow, and basically the browser is redirect URI here in the client application. Remember, we supplied this URI at the very beginning in step one. And as you can see, the response here includes the authorization code. That's simply a short-lived generated code that is only meant to be used once and it's only meant to be exchanged for the access token.

0:02:57.2: All right. Step number four here, we're finally getting the actual access token. So the client application now needs to exchange the authorization code for the access token, and that is done by sending a POST request to the authorization server's token end point. Note that, unlike the previous step where everything was passing through the user agent, through the browser, through the client side, here the request is made from the server-side component of the client application directly to the authorization server, and that is why we can actually include the client secret here. The client needs to be authenticated by the authorization server, so this request will be including the client secret, and again, not touching the browser.

0:03:44.8: All right. And we're now finally to the point where we get back the access token. The authorization server will issue the access token back to the client as a response to the post that we've just sent. And finally the client can start requesting access to the protected resources, of course, using the access token that it just got back.

0:04:07.3: Now, before we wrap up here, it's also worth mentioning that, in the authorization code flow, the usage of refresh tokens is the standard mechanism that we're going to use to deal with short-lived access tokens. But remember, there is a dedicated lesson focused on this, which is why we're not exploring that here. And finally on this one, make sure you have a look at the lesson notes, as I've included more details about each of these steps.

0:04:35.8: All right. Hope you're excited. See you in the next lesson.