

OpenID Connect Flow

In this exercise you'll learn how to request an OpenID Connect ID token and extract the user's information from it.

The goal of this exercise is to get an ID token and to extract the user's profile information from the ID token. We will be building on the previous exercise where you used the authorization code flow to get an access token. Rather than repeat all the setup steps here, we'll assume you have already created an application and have gone through the authorization code flow at least once.

To get an ID token, you need to add the **openid** scope to the authorization request. You can also add the **profile** and **email** scopes to get more information about the user. Build the authorization URL including those three scopes.

Generate Random String

Code Verifier plaintext random string

Save the Code Verifier and keep it secret, you won't need that until the end.

Calculate Hash

Code Challenge base64-url-encoded SHA256 hash of the code ver

Fill in the placeholder values with your own values. (Make sure to replace the curly brackets, those are just to indicate placeholder values.)

Authorization Request

Note that we are still using the authorization code flow with PKCE when getting the ID token so that we get it over the back channel, simplifying the process by avoiding the need to verify the ID token signature.

```
https://xxxxxx.us.auth0.com/authorize?  
  response_type=code&  
  client_id={YOUR_CLIENT_ID}&  
  state={RANDOM_STRING}&  
  scope={SCOPE}&  
  redirect_uri=https://example-  
app.com/redirect&  
  code_challenge={YOUR_CODE_CHALLENGE}&
```

Create the initial URL for the authorization request and paste it above. Once it's correct, a "Log In" button with that URL will appear below

Check Your URL

Token Response

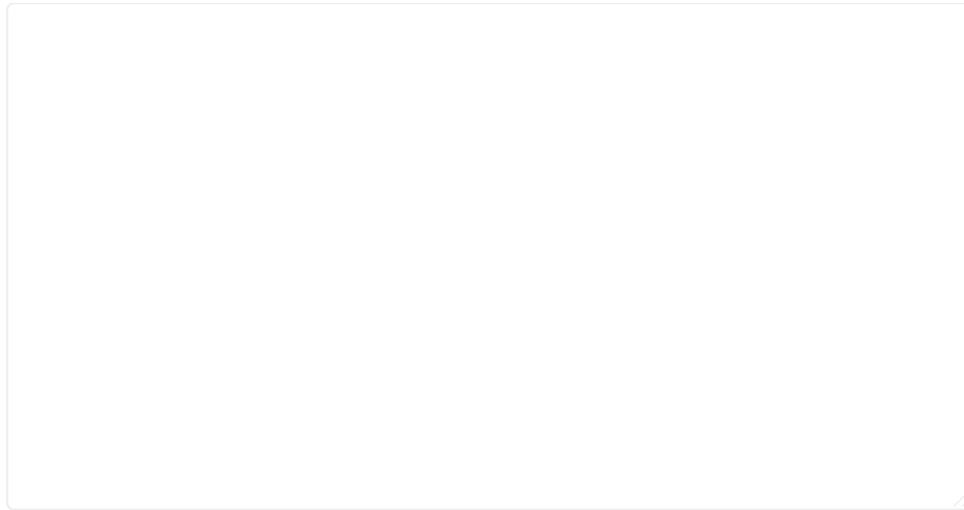
```
{  
  "token_type": "Bearer",  
  ...  
}
```

Use the authorization code flow to get an ID token, then paste the entire token response JSON here to check your work

Check Your Response

ID Token Claims

The ID token returned from the token endpoint is below.



Parse the claims from the JWT using a [Base64 decoder](#) and paste the user's subject, name and email address into the form below. Remember that because you got this ID token over the back channel, you don't need to worry about verifying the JWT signature.

Subject (sub)

Email address

Name