

Total Economic Impact of Auth0 ▶

- 🔍
- Articles
- Auth0 APIs
- QuickStarts
- Libraries
- PSaaS Appliance

Articles

Refresh Token



In this article

Version [current](#)

A **Refresh Token** is a special kind of token that contains the information required to obtain a new [Access Token](#) or [ID Token](#).

Usually, a user will need a new Access Token only after the previous one expires, or when gaining access to a new resource for the first time.

Refresh Tokens are subject to strict storage requirements to ensure that they are not leaked. Also, [Refresh Tokens can be revoked](#) by the Authorization Server.

OIDC-conformant applications

The behaviour in this document is applicable to [OIDC-conformant applications](#). An application can be configured as OIDC-conformant in two ways:

1. By enabling the **OIDC Conformant** flag for an Application
2. By passing an `audience` to the `/authorize` endpoint

For more information on our authentication pipeline, refer to [Introducing OIDC Conformant Authentication](#).

Overview

The response of an [authentication request](#) can result in an Access Token and/or an ID Token being issued by Auth0. The Access Token is used to make authenticated calls to a secured API, while the ID Token contains user profile attributes represented in the form of *claims*. Both JWTs have an expiration date indicated by the `exp` claim (among other security measures, like signing).

A Refresh Token allows the app to get a new Access Token or ID Token directly, without having to re-authenticate as long as the Refresh Token has not been revoked.

✓
ie a new Access Token or ID Token
ork as long as the Refresh Token has not

Restrictions

You can only get a Refresh Token if you are implementing: [Authorization Code Grant](#), [Authorization Code Grant \(PKCE\)](#) or [Resource Owner Password Grant](#).

A Single Page Application (normally implementing [Implicit Grant](#)) should not under any circumstances get a Refresh Token. The reason for that is the sensitivity of this piece of information. You can think of it as user credentials, since a Refresh Token allows a user to remain authenticated essentially forever. Therefore you cannot have this information in a browser, it must be stored securely.

If you are implementing an SPA using [Implicit Grant](#) and you need to renew a token, the only secure option is to use [Silent Authentication](#).

Another safeguard is that the API should allow offline access. This is configured via the **Allow Offline Access** switch on the [API Settings](#). If the switch is disabled, Auth0 will not return a Refresh Token for this API, even if you included the `offline_access` scope.

Get a Refresh Token

To get a Refresh Token, you must include the `offline_access` scope when you initiate an authentication request through the [authorize](#) endpoint.

For example, if you are using [Authorization Code Grant](#), the authentication request would look like the following:

```
https://YOUR_AUTH0_DOMAIN/authorize?  
audience={API_AUDIENCE}&
```

```
scope=offline_access&
response_type=code&
client_id=YOUR_CLIENT_ID&
redirect_uri=https://YOUR_APP/callback&
state={OPAQUE_VALUE}
```

Did it help? [Yes](#) / [No](#)

Once the user authenticates successfully, the application will be redirected to the `redirect_uri`, with a `code` as part of the URL: `https://YOUR_APP/callback?code=BPPLN3Z4qCTvSN0y`. You can exchange this code with an Access Token using the `/oauth/token` endpoint.

[cURL](#) [C#](#) [Go](#) [Java](#) [jQuery](#) [Node.JS](#) [Obj-C](#) [PHP](#) [Pyt](#)

```
curl --request POST \
--url 'https://YOUR_AUTH0_DOMAIN/oauth/token' \
--header 'content-type: application/json' \
--data '{"grant_type":"authorization_code","client_id": "YOUR_CLIENT_ID","client_secret
```

Did it help? [Yes](#) / [No](#)

The response should contain an Access Token and a Refresh Token.

```
{
  "access_token": "eyJz93a...k4laUWw",
  "refresh_token": "GEbRxBN...edjnXbL",
  "token_type": "Bearer"
}
```

Did it help? [Yes](#) / [No](#)

If you are requesting a `refresh_token` for a mobile app using the corresponding Native Client (which is public) then you don't need to send the `client_secret` in the request since it's only needed for [confidential applications](#).

⚠ Refresh Tokens must be ✓ since they allow a user to remain authenticated essentially.

For more information on how to implement this using Authorization Code Grant refer to [Execute an Authorization Code Grant Flow](#). For other grants refer to [API Authorization](#).

📖 If the response did not include a Refresh Token, check that you comply with the Restrictions listed in this document.

Use a Refresh Token

To refresh your token, using the `refresh_token` you already got during authorization, make a POST request to the `/oauth/token` endpoint in the Authentication API, using `grant_type=refresh_token`.


[cURL](#)[C#](#)[Go](#)[Java](#)[jQuery](#)[Node.JS](#)[Obj-C](#)[PHP](#)[Pyt](#)

```
curl --request POST \
  --url 'https://YOUR_AUTH0_DOMAIN/oauth/token' \
  --header 'content-type: application/json' \
  --data '{ "grant_type": "refresh_token", "client_id": "YOUR_CLIENT_ID", "client_secret"
```

Did it help? [Yes](#) / [No](#)

Where:

- `grant_type` : The type of grant to execute (the `/token` endpoint is used for various grants, for more information refer to the [Authentication API](#)). To refresh a token use `refresh_token`.
- `client_id` : Your application's Client ID.

- `client_secret` (optional) . Only required for [confidential applications](#).
- `refresh_token` : The Refresh Token to use.

The response will include a new Access Token, its type, its lifetime (in seconds), and the granted scopes. If the scope of the initial token included `openid` , then a new ID Token will be in the response as well.

```
{
  "access_token": "eyJ...MoQ",
  "expires_in": 86400,
  "scope": "openid offline_access",
  "id_token": "eyJ...0NE",
  "token_type": "Bearer"
}
```

Did it help? [Yes](#) / [No](#)

Rate limits

You should only ask for a new token if the Access Token has expired or you want to refresh the claims contained in the ID Token. For example, it's a bad practice to call the endpoint to get a new Access Token every time you call an API. There are rate limits in Auth0 that will throttle the amount of requests to this endpoint that can be executed using the same token from the same IP.

Revoke a Refresh Token

Since Refresh Tokens never expire it is important to be able to revoke them in case they get compromised.

Auth0 handles token revocation by potentially exposing the token to malicious adversaries. Hence each refresh token is only valid for the specific token, but all other tokens based on the same authorization grant. This means that all Refresh Tokens that have been issued for the same user, application, and audience will be revoked.

You can revoke a Refresh Token either by posting a request to [the Authentication API /oauth/revoke endpoint](#) or using the [dashboard](#).

Use the API

To revoke a Refresh Token you can send a `POST` request to `https://YOUR_AUTH0_DOMAIN/oauth/revoke`.

The API first validates the application credentials and then verifies whether the token was issued to the application making the revocation request. If this validation fails, the request is refused and the application is informed of the error. Next, the API invalidates the token. The invalidation takes place immediately, and the token cannot be used again after the revocation. Note that each revocation request invalidates all the tokens that have been issued for the same authorization grant.

[cURL](#)[C#](#)[Go](#)[Java](#)[jQuery](#)[Node.JS](#)[Obj-C](#)[PHP](#)[Pyl](#)

```
curl --request POST \
  --url 'https://YOUR_AUTH0_DOMAIN/oauth/revoke' \
  --header 'content-type: application/json' \
  --data '{ "client_id": "YOUR_CLIENT_ID", "client_secret": "YOUR_CLIENT_SECRET", "token"
```

Did it help? [Yes](#) / [No](#)

Where:

Parameter	Description
-----------	-------------

Parameter	Description	✓
client_id <div>REQUIRED</div>	Your application's Client ID. The application should match the one the Refresh Token was issued for.	
client_secret	Your application's Client Secret. Required for confidential applications .	
token <div>REQUIRED</div>	The Refresh Token you want to revoke.	

The application should match the one the Refresh Token was issued for.

Revoke a token without the Client Secret

For applications that cannot keep the Client Secret safe (for example, native apps), the [Revoke endpoint](#) supports access without the Client Secret but the application itself must have the property `tokenEndpointAuthMethod` set to `none`. You can change the `tokenEndpointAuthMethod` value, either from the UI ([Dashboard > Clients > Application Settings](#)), or using the [Management API](#).

If the request is valid, the Refresh Token is revoked and the response is `HTTP 200`, with an empty response body. Otherwise, the response body contains the error code and description.

```
{
  "error": "invalid_request|invalid_client",
  "error_description": "Description of the error"
}
```

Did it help? [Yes](#) / [No](#)

The possible responses are:

HTTP

Status	Description
--------	-------------

HTTP



Status	Description
--------	-------------

200	The Refresh Token is revoked, does not exist, or was not issued to the application making the revocation request. The response body is empty.
-----	---

400	The required parameters were not sent in the request (<code>"error": "invalid_request"</code>).
-----	---

401	The request is not authorized (<code>"error": "invalid_client"</code>). Check that the application credentials (<code>client_id</code> and <code>client_secret</code>) are present in the request and hold valid values.
-----	--

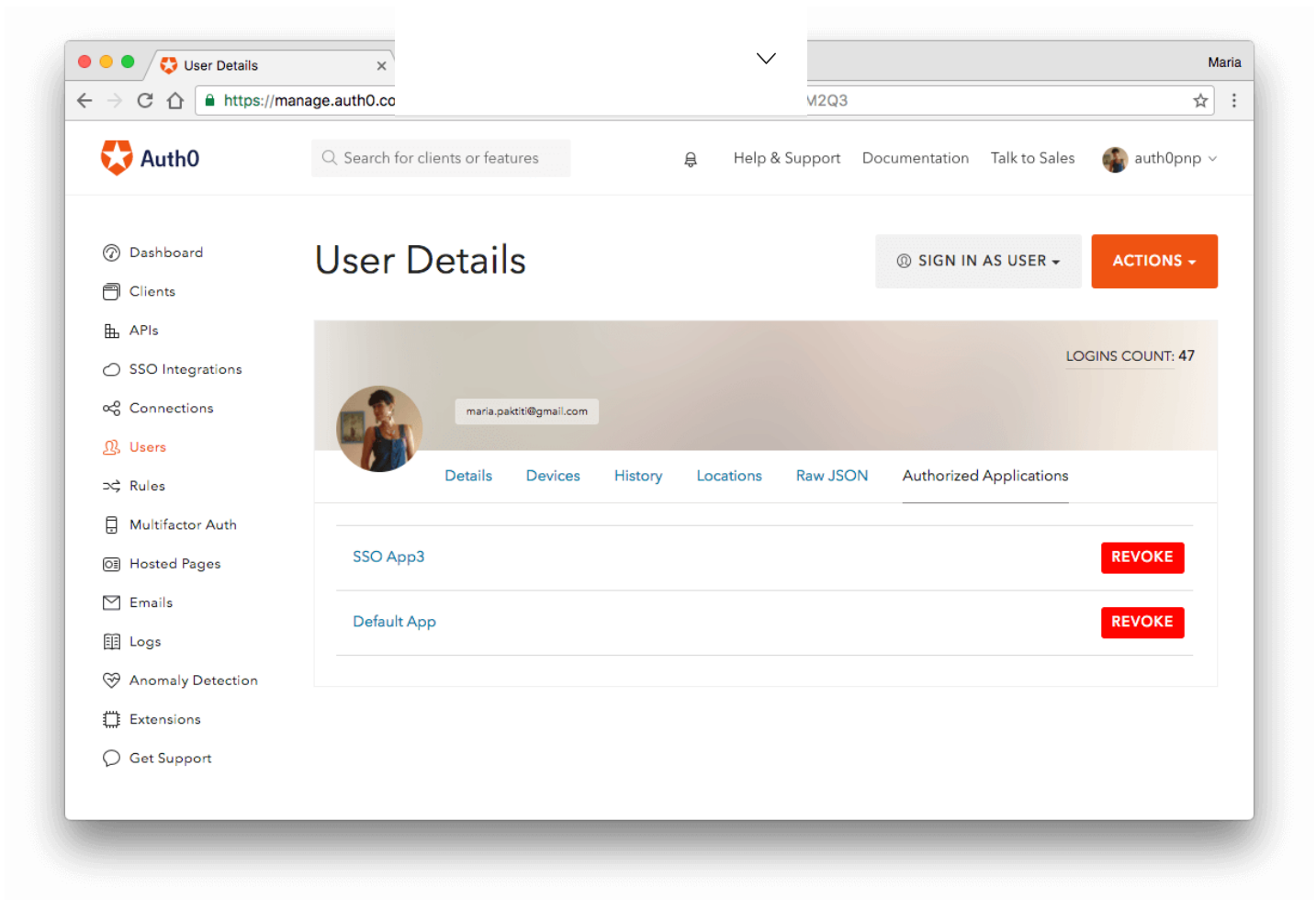
Use the Dashboard

When you revoke a Refresh Token using the dashboard, you have to revoke the user's authorized access to the application that issued the token. This renders the Refresh Token useless.

To do so, go to the [Users section](#) of the [dashboard](#). Click the name of the user to view their *Details* page.

Select the *Authorized Applications* tab. This page lists all the applications to which the user has authorized access. Revoking an authorized application revokes also its associated Refresh Tokens.

To revoke the user's access to an authorized application, and hence invalidate the Refresh Token, click **Revoke**.



Rules

Rules will run for the [Refresh Token Exchange](#). To execute special logic, you can look at the `context.protocol` property in your rule. If the value is `oauth2-refresh-token`, then this is the indication that the rule is running during the [Refresh Token Exchange](#).

⚠ If you try to do a redirect with `context.redirect`, the authentication flow will return an error.

SDK Support

Web Apps

All our main SDKs support Refresh Tokens. For a complete list of supported SDKs, see [Refresh Tokens](#), [Node.js](#), [ASP.NET Core](#), [PHP](#), [Java](#), and many more. For a complete list of supported SDKs, see [Refresh Tokens](#).



ie are [Node.js](#), [ASP.NET Core](#), [PHP](#), [Java](#), and many more. For a complete list of supported SDKs, see [Refresh Tokens](#).

Single Page Apps

For web apps that execute on the browser, the way to refresh a token is using [Silent Authentication](#). [Auth0.js](#), our client-side library, provides methods for this out of the box.

- The `authorize` method, redirects the user to the `/authorize` endpoint, in order to login and provide consent.
- The `parseHash` method, parses a URL hash fragment to extract the result of an Auth0 authentication response.
- The `checkSession` method, attempts to get a new token from Auth0, using [silent authentication](#). For more details refer to [Using checkSession to Acquire New Tokens](#).

More information on the library:

- [Auth0.js Reference](#)
- [Auth0.js GitHub repo](#)

Mobile / Native Apps

For more information on using Refresh Tokens with our mobile SDKs refer to:

- [Mobile / Native Quickstarts](#)
- [Lock Android: Refreshing JWT Tokens](#)
- [Lock iOS: Saving and Refreshing JWT Tokens](#)

Keep reading