**<https://auth0.com/docs/tokens/management-api-access-tokens/get-management-api-access-tokens-for-production>**

**https://auth0.com/docs/flows/authorization-code-flow**

[**https://auth0.com/docs/tokens/json-web-tokens**](https://auth0.com/docs/tokens/json-web-tokens)

[**https://auth0.com/docs/tokens/refresh-tokens/configure-refresh-token-expiration**](https://auth0.com/docs/tokens/refresh-tokens/configure-refresh-token-expiration)

[**https://auth0.com/docs/tokens/management-api-access-tokens/get-management-api-access-tokens-for-production#use-access-tokens**](https://auth0.com/docs/tokens/management-api-access-tokens/get-management-api-access-tokens-for-production#use-access-tokens)

[**https://auth0.com/docs/get-started**](https://auth0.com/docs/get-started)

[**https://auth0.com/docs/universal-login/universal-vs-embedded-login**](https://auth0.com/docs/universal-login/universal-vs-embedded-login)

[**https://auth0.com/docs/tokens/management-api-access-tokens/get-management-api-access-tokens-for-testing#get-access-tokens-manually**](https://auth0.com/docs/tokens/management-api-access-tokens/get-management-api-access-tokens-for-testing#get-access-tokens-manually)

[**https://www.oauth.com/**](https://www.oauth.com/)

[**https://smartbear.com/ppc/ready-api/soapui/?utm\_source=aw&utm\_medium=ppcg&utm\_term=%2Btesting%20%2Bapis&utm\_content=469733882335&utm\_campaign=342047661&gclid=Cj0KCQiAlsv\_BRDtARIsAHMGVSYO29HVmexPeBdgO8MIOJhwspR7IUwzy0jQ8Px-ZGll5A9un\_Y3vg0aAijLEALw\_wcB&gclsrc=aw.ds**](https://smartbear.com/ppc/ready-api/soapui/?utm_source=aw&utm_medium=ppcg&utm_term=%2Btesting%20%2Bapis&utm_content=469733882335&utm_campaign=342047661&gclid=Cj0KCQiAlsv_BRDtARIsAHMGVSYO29HVmexPeBdgO8MIOJhwspR7IUwzy0jQ8Px-ZGll5A9un_Y3vg0aAijLEALw_wcB&gclsrc=aw.ds)

Get Management API Access Tokens for Production

To make scheduled frequent calls for a production environment, you have **to build a process at your backend that will provide you with a token automatically** (and thus simulate a non-expiring token).

Prerequisites

* [Create and Authorize a Machine-to-Machine Application](https://auth0.com/docs/tokens/management-api-access-tokens/create-and-authorize-a-machine-to-machine-application).

**Get access tokens**

To ask Auth0 for a Management API v2 token, perform a POST operation to the https://YOUR\_DOMAIN/oauth/token endpoint, using the credentials of the Machine-to-Machine Application you created in the prerequisite step.

The payload should be in the following format:

curl --request POST \

--url 'https://YOUR\_DOMAIN/oauth/token' \

--header 'content-type: application/x-www-form-urlencoded' \

--data grant\_type=client\_credentials \

--data 'client\_id=YOUR\_CLIENT\_ID' \

--data **client\_secret**=YOUR\_CLIENT\_SECRET \

--data 'audience=https://YOUR\_DOMAIN/api/v2/'

# **python to make HTTPS requests**

import http.client

conn = http.client.**HTTPSConnection**("")

**payload** = "**grant\_type**=client\_credentials&client\_id=%24%7Baccount.clientId%7D&**client\_secret**=YOUR\_CLIENT\_SECRET&audience=https%3A%2F%2F%24%7Baccount.namespace%7D%2Fapi%2Fv2%2F"

**headers** = { 'content-type': "application/x-www-form-urlencoded" }

conn.request("POST", "/YOUR\_DOMAIN/oauth/token", payload, headers)

res = conn.getresponse()

data = res.read()

print(data.decode("utf-8"))

#-----------------------

The **response** will contain a [**signed JWT**](https://auth0.com/docs/tokens/json-web-tokens), an expiration time, the scopes granted, and the token type.

{

"**access\_token**": "eyJ...Ggg",

"**expires\_in**": 86400,

"scope": "read:clients create:clients read:client\_keys",

"**token\_type**": "Bearer"

}

From the above, we can see that our Access Token will expire in 24 hours (86400 seconds), it has been authorized to read and create applications, and it is **a**[**Bearer Access Token**](https://tools.ietf.org/html/rfc6750).

# Revoke Tokens

Once issued, access tokens and ID tokens cannot be revoked in the same way as cookies with session IDs for server-side sessions.

As a result, tokens should be issued for relatively short periods, and then [refreshed](https://auth0.com/docs/tokens/refresh-tokens) periodically if the user remains active.