

OAuth tokens

Did you know you can easily exchange your OAuth tokens to access many different apps from your Hogwarts access token? This guide will cover the essentials on how to go about it and the possibilities behind it.

- [What are OAuth tokens?](#)
- [Where are my OAuth tokens?](#)
- [How can I exchange my Hogwarts token for an app X token?](#)
- [Which apps can I exchange my Hogwarts token for?](#)
- [An easier approach to OAuth and on-behalf-of \(Token credentials\)](#)

What are OAuth tokens?

OAuth tokens are simply tokens that enable users to access resources (e.g. REST APIs) on their behalf. These access tokens are generally emitted by services or apps after a user has logged in and will expire after X amount of time. During the login process, the users or the app may receive a *refresh token* which can then be used to generate new access tokens once theirs expires.

OAuth tokens contain several important bits of information. In our case, the *audience* and *scope* are the most important bits as they specify which app and which API permission the user can access with this specific token. We will use these bits of information later on during our on-behalf-of exchanges.

Where are my OAuth tokens?

In Hogwarts, users can access their current session's OAuth access and refresh tokens in the form of an environment variable.


```
[1]: import os

print(os.environ.get("NOTEBOOK_ACCESS_TOKEN"))

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5PbzNa

[2]: print(os.environ.get("NOTEBOOK_REFRESH_TOKEN"))

0.ASwAQL6c2h7s10mvxfYdXRXGmmQDcp4WINKiluPdZNdwMEsAK
```

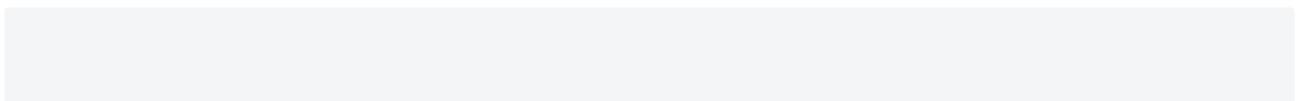
 Hogwarts will be standardizing its token environment variables to `VAULT_ACCESS_TOKEN` and `VAULT_REFRESH_TOKEN` in the near future as these are ready to use with the Hogwarts Auth Python library.

How can I exchange my Hogwarts token for an app X token?

Performing the on-behalf-of process is trivial when using the [Hogwarts Auth python library](#). The only requirement is for you to run your code on a Hogwarts execution platform (i.e. JupyterHub, Airflow)

As a start, follow the example below on your own JupyterHub server:

▼ 1. Obtain your refresh token



```
1 import os
2 jupyterhub_refresh_token = os.environ.get("VAULT_REFRESH_TOKEN")
```


2. Use the token and create a ContextualRefreshToken

```
1 from hogwarts.auth.vault.contextual_refresh_token import ContextualRefreshToken
2 jupyterhub_ctx_refresh_token = ContextualRefreshToken(jupyterhub_refresh_token)
```

Print the token to make sure it follows the correct structure

```
print(jupyterhub_ctx_refresh_token)

jupyterhub/jupyterhub:0.ASwAQL6c2h7s10mvsxfYdXRXGmmQDcp
```

 A ContextualRefreshToken is structured as such : <refresh_client>/<obo_client>:<refresh_token> The client names are generated for you and are necessary for the on-behalf-of process.

3. Create a VaultClient

```
1 from hogwarts.auth.vault.vault_client import VaultClient
2
3 # Vault client uses the Hogwarts Vault by default
4 vault_client = VaultClient()
```

4. Perform an on-behalf-of exchange for the Azure Storage app

Audience + scope = https://storage.azure.com/user_impersonation

```
1 storage_access_token, storage_ctx_refresh_token = vault_client.on_behalf_of(
2     'https://storage.azure.com/user_impersonation',
3     jupyterhub_ctx_refresh_token
4 )
```

5. Make Requests to the Azure Storage API

Replace the URL to any file you wish to print its contents.

```
1 import requests
2 r = requests.get(
3     "https://stpilotdatalakedev.blob.core.windows.net/data/sm-test/hello.txt",
4     headers={
5         'Authorization': f'Bearer {storage_access_token}',
6         'x-ms-version': '2017-11-09'
7     }
8 )
```

```
print(r)
print(r.text)
```

```
<Response [200]>
hello
```

6. If your access token ever expires (~1 hour), refreshing is a breeze

```
1 refreshed_storage_access_token = vault_client.refresh(storage_ctx_refresh_token)
```

 The Notebook code for this example can be found in the [Hogwarts example repository](#).

Which apps can I exchange my Hogwarts token for?

With an Airflow or JupyterHub token, you can obtain the following app tokens using the on-behalf-of flow:

Azure apps

- Azure Storage (Aud=<https://storage.azure.com/>) - Supports storage, DataLake, etc.
- Microsoft Graph (Aud=<https://graph.microsoft.com/>) - User info

Hogwarts apps

- Spark

Other apps

- Alfred (api://alfred.pb - PB only)

 Don't see an app that you would like? [Send us your suggestions](#).

An easier approach to OAuth and on-behalf-of (Token credentials)

Managing app audiences and scopes, as well as keeping track of your token's expiry can be overwhelming. This is why the Hogwarts team offers a series of Token credentials to make your life easier. These Token credentials encapsulate the refresh and on-behalf-of steps to keep your code clean and easy to read, while simultaneously managing the required audience and scope for a specific app.

For more information about these token credentials, visit the [Hogwarts token credentials guide](#).