Sprinklr Client Library - Generating an Access Token

The Sprinklr Client Library and test app are available at https://github.com/DzRepo/SprinklrClient

It is built using Python 3.x and requires two external libraries: Requests and EasySettings (EasySettings is only used by the SprinklrClientTest.py, Requests is used by the SprinklrClient)

The first step is to create an account and application at https://developer.sprinklr.com This account and data live on Mashery and exist independent of Sprinklr's production instances. You should not use the same password on both sites for security. Many customers will create a generic mashery account (development@company.com) so the API key is not tied to an individual user's email address.

The next step is to generate a temporary code that is used to request an access token and refresh token. This process also creates a link between the application API Key and a specific Sprinklr client environment.

> *Note: A single API key can be used to generate multiple codes, and thus access tokens, to separate customer environments (for example, a multi-tenant 3rd party reporting application could be used by multiple sprinklr customers, but would only use a single API key.*

Before proceeding, if the environment in use is not Production, create or edit Sprinklr.conf to specify path, for example:

```
path=prod2
```

If you are not sure which environment your Sprinklr instance is running on, please ask your Success Manager.

In the following command, Redirect URL is where the 'code' parameter will be sent, as a URL parameter.

Using the API key created at Developer.Sprinklr.com, execute:

```
python SprinklrClientTest.py authorize {APIKey} {Redirect URL}
```

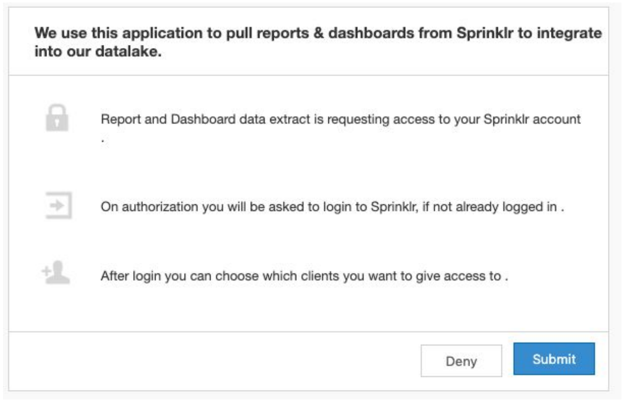
For example:

```
python SprinklrClientTest.py authorize fs9kxxxxxxzap4 https://www.sprinklr.com
```

This will load a web page in the default browser, redirected to:

```
https://app.sprinklr.com/ui/oauth/authorize?client_id={APIKey}&response_type=co
de&redirect_uri={Redirect URL}
```

That looks like:



We use this application to pull reports & dashboards from Sprinklr to integrate into our datalake.

Report and Dashboard data extract is requesting access to your Sprinklr account .

On authorization you will be asked to login to Sprinklr, if not already logged in .

After login you can choose which clients you want to give access to .

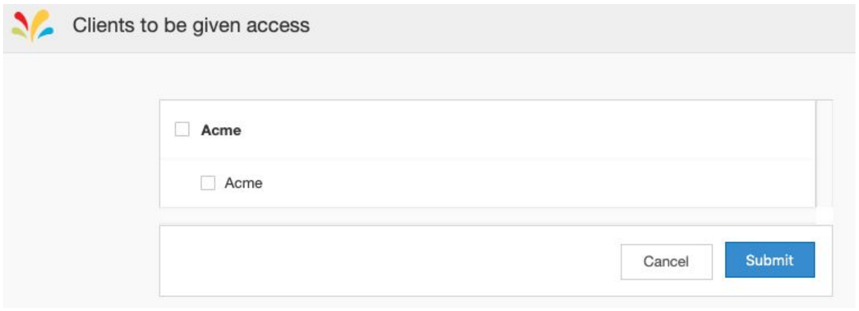
(Note the form shows the description that was entered when the application was created on the developer portal)

Clicking submit takes you to:

```
https://app.sprinklr.com/ui/login?returnTo=%2Fui%2Foauth%2Fpre_authorized%3Fsta
te%3Dnull%26client_id%3{APIKey}%26redirect_uri%3D{Redirect URL}
```

And show the login screen (unless you are already logged into the Sprinklr app in another tab)

Once logged in, you will be presented with a selection of workspaces the API could have access to, for example:



Clients to be given access

- [ ] **Acme**
  - [ ] Acme

On the right workspaces are selected and the form submitted, the page redirects to {Redirect URL} with Code as a parameter.

```
{Redirect URL}?code=5e1xxxxxxxxxxx713479e265&state=null
```

The next step is to use this code, along with the Secret key of the Application to generate Access and Renewal tokens. This allows the application to make API requests to access customer account:

```
python SprinklrClientTest.py GetAccessToken {apikey} {secret} {code} {redirect
uri}
```

The only output of this is command is “Success” if the process was finished without error. The Access Token, Renewal token, API key and secret will all be stored in Sprinklr.conf. Now, the rest of the SprinklrClientTest functions can be used. The SprinklrClientTest.py app will use the tokens stored in Sprinklr.conf for authentication.