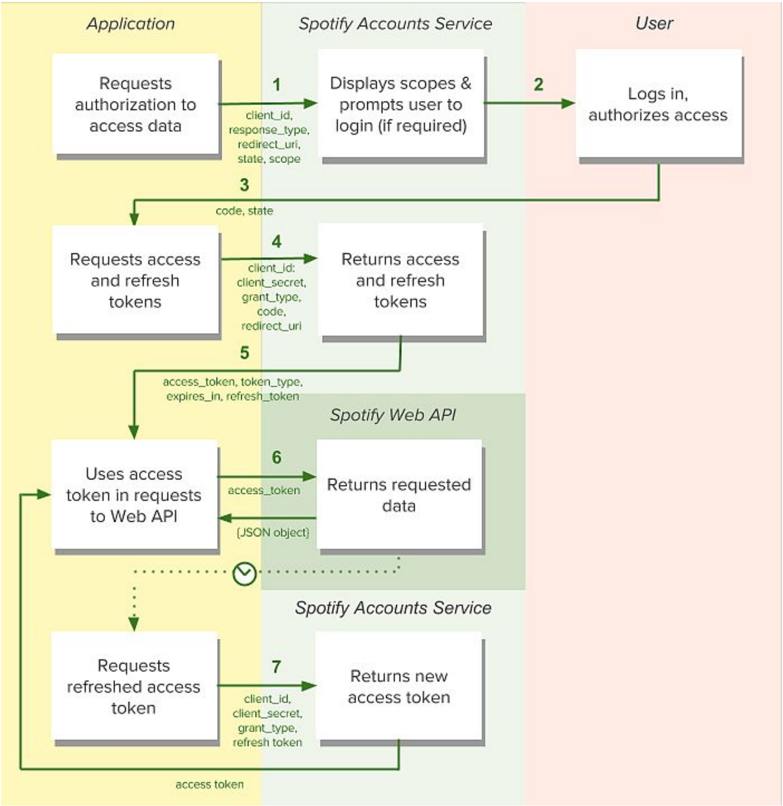
HW3 & FP Preparation

Demo of Homework 3

It’s fun!

Spotify Web API OAuth Guide - Overview



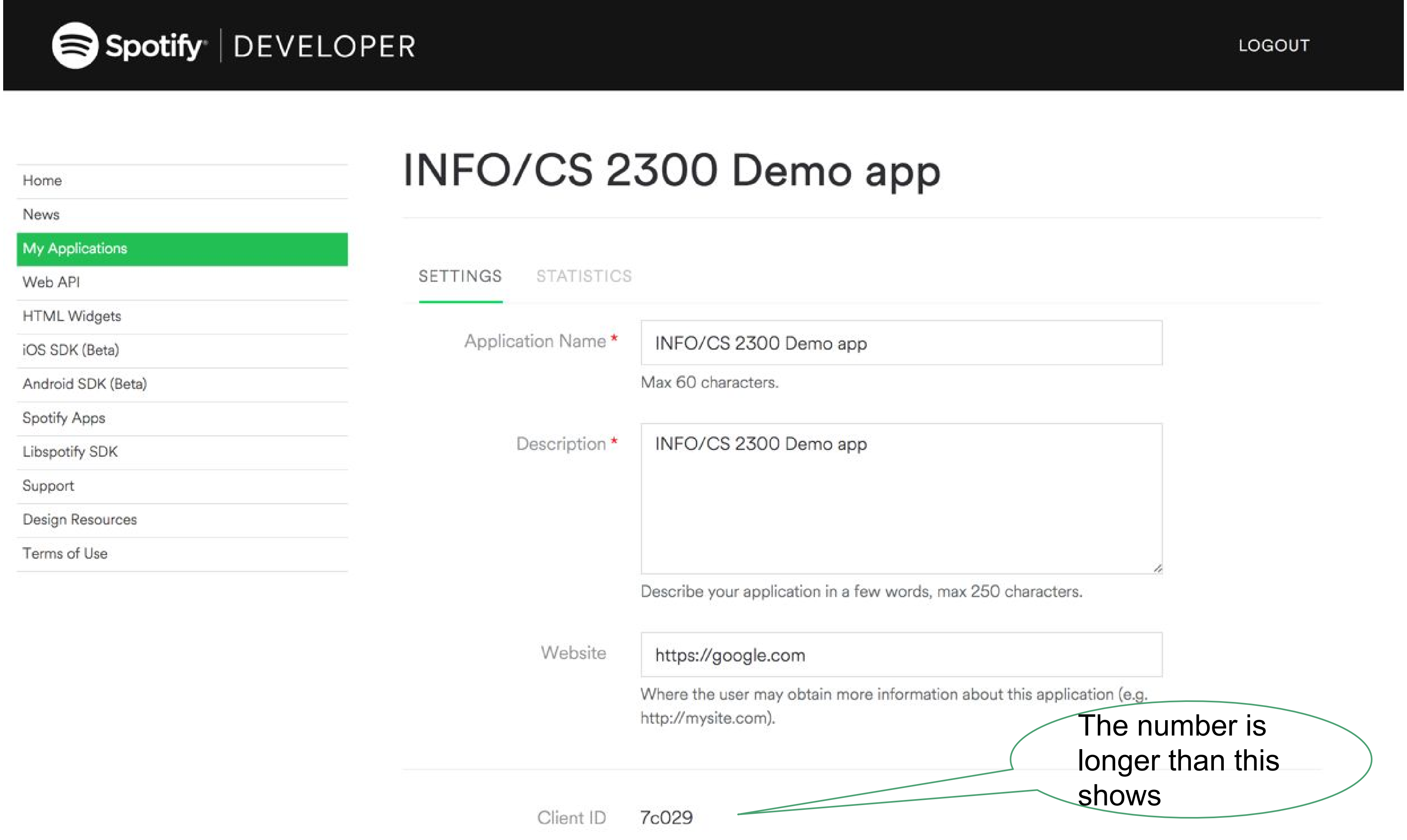
Application
Spotify Accounts Service
User
Requests authorization to access data
1
client_id, response_type, redirect_uri, state, scope
Displays scopes & prompts user to login (if required)
2
Logs in, authorizes access
3
code, state
Requests access and refresh tokens
4
client_id: client_secret, grant_type, code, redirect_uri
Returns access and refresh tokens
5
access_token, token_type, expires_in, refresh_token
Spotify Web API
Uses access token in requests to Web API
6
access_token
{JSON object}
Returns requested data
Spotify Accounts Service
Requests refreshed access token
7
client_id, client_secret, grant_type, refresh token
Returns new access token
access token

https://developer.spotify.com/web-api/authorization-guide/

Application Registration

https://developer.spotify.com/my-applications/

- Register a new application by filling out the form
- Must get a unique client ID from the registration
- You will need this to make the initial authentication call



INFO/CS 2300 Demo app

Application Name * INFO/CS 2300 Demo app

Max 60 characters.

Description * INFO/CS 2300 Demo app

Describe your application in a few words, max 250 characters.

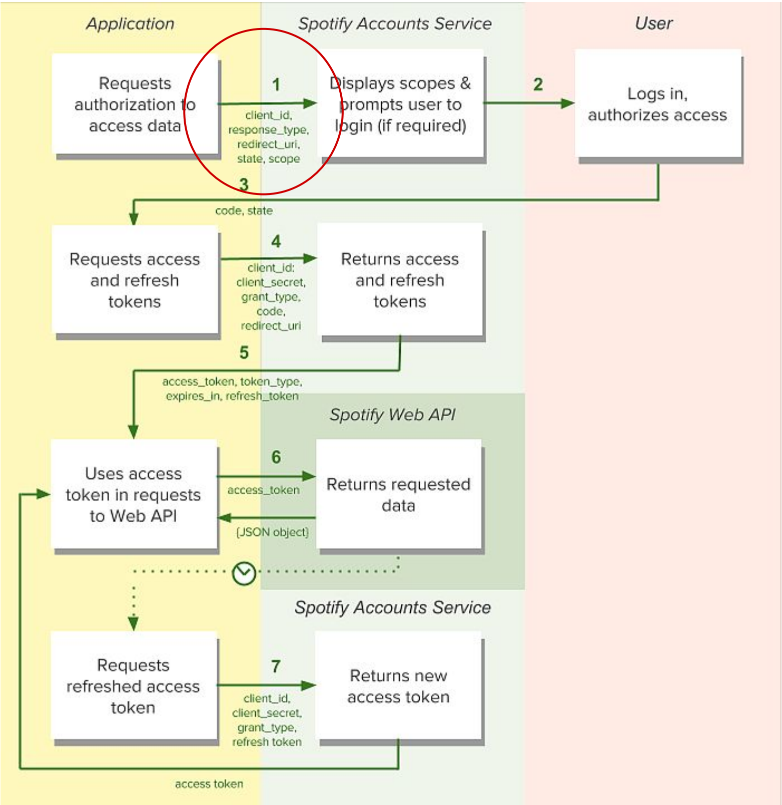
Website https://google.com

Where the user may obtain more information about this application (e.g. http://mysite.com).

Client ID 7c029

The number is longer than this shows

Spotify Web API OAuth Guide



Application

Spotify Accounts Service

User

Requests authorization to access data

1 client_id, response_type, redirect_uri, state, scope

Displays scopes & prompts user to login (if required)

2

Logs in, authorizes access

3 code, state

Requests access and refresh tokens

4 client_id: client_secret, grant_type, code, redirect_uri

Returns access and refresh tokens

5 access_token, token_type, expires_in, refresh_token

Spotify Web API

Uses access token in requests to Web API

6 access_token

{JSON object}

Returns requested data

Spotify Accounts Service

Requests refreshed access token

7 client_id, client_secret, grant_type, refresh token

Returns new access token

access token

To login the following are needed to get an authorization token

- client_id - from your application
- response_type = token
- redirect_url - one for each server on which you run
- state - not used
- scope - levels of permission your login allows

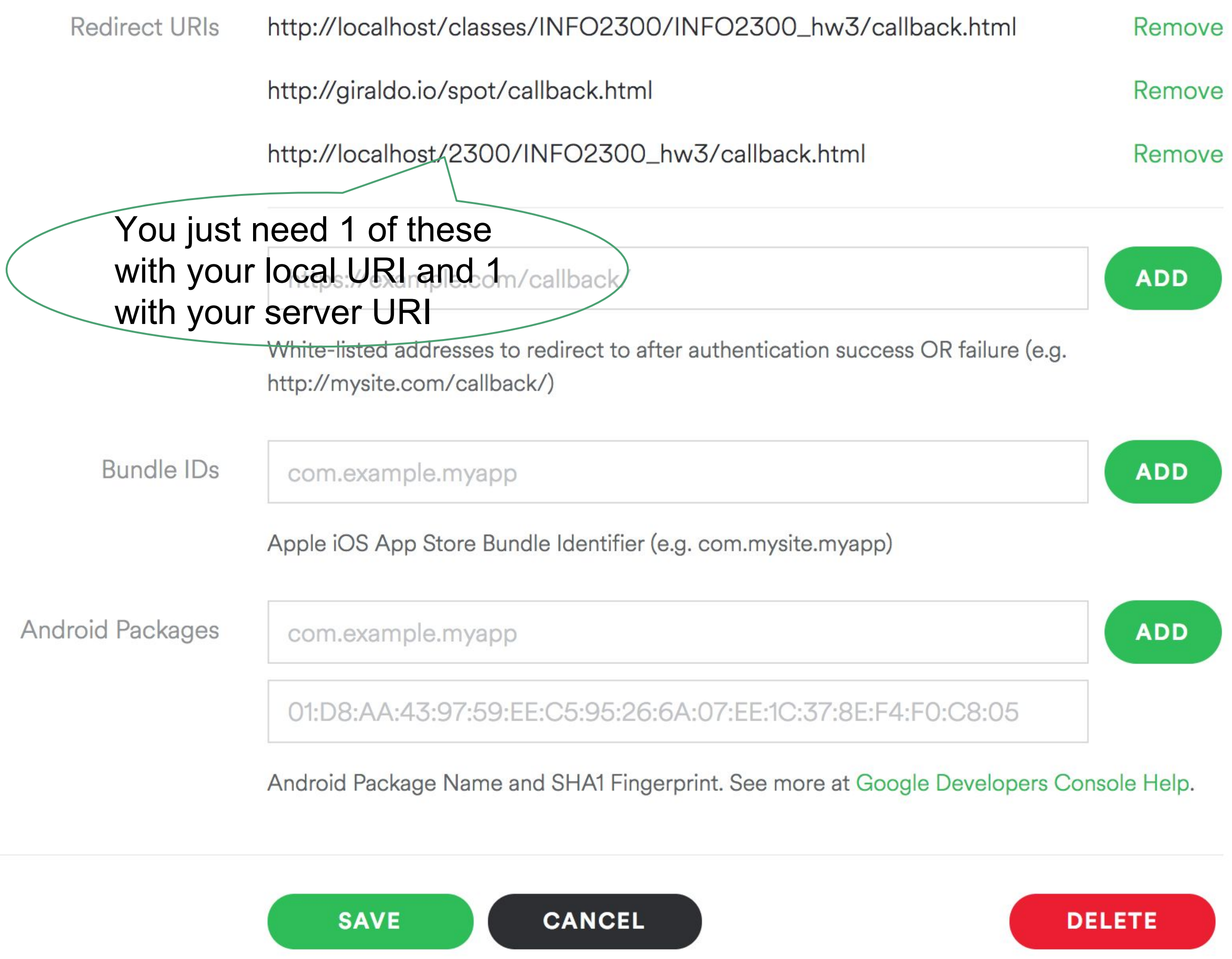
Redirect URIs

- After a request is sent to the spotify API to authenticate a user, if the authentication was successful, it sends back a hashed token.
- This redirect URIs tells the spotify API where to send that token in your application.
- Ideally, it should be a dedicated piece of your application that simple decodes the hashed access token, and sends that back to your main part of the application.

Click In!

What would happen if you forgot to put a redirect URI in the application console, and user tried to log in?

A. Spotify would send the access token back to the base application
B. The login window would fail to load due to an invalid URI
C. The login window would prompt that the user input a URI
D. The login window would fail to load due to an invalid access token
E. The users spotify account is terminated



Redirect URIs

http://localhost/classes/INFO2300/INFO2300_hw3/callback.html

http://giraldo.io/spot/callback.html

http://localhost/2300/INFO2300_hw3/callback.html

https://example.com/callback

White-listed addresses to redirect to after authentication success OR failure (e.g. http://mysite.com/callback/)

Bundle IDs

com.example.myapp

Apple iOS App Store Bundle Identifier (e.g. com.mysite.myapp)

Android Packages

com.example.myapp

01:D8:AA:43:97:59:EE:C5:95:26:6A:07:EE:1C:37:8E:F4:F0:C8:05

Android Package Name and SHA1 Fingerprint. See more at Google Developers Console Help.

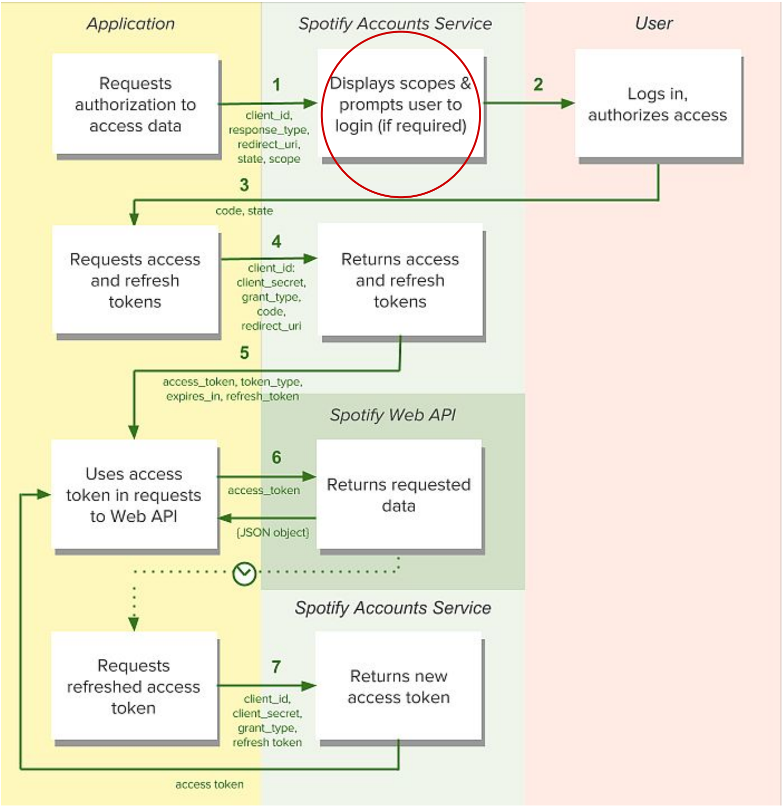
You just need 1 of these with your local URI and 1 with your server URI

Scopes

https://developer.spotify.com/web-api/using-scopes/

- Levels of access you want to request of the user logging in
- Different scopes (if authenticated) let you have access to different API endpoints
- Example: “user-read-email” is the email scope for a user.
- You can request many scopes, in hw3 we store each one in an array to use later in the final login url

Spotify Web API OAuth Guide


Application
Spotify Accounts Service
User
Requests authorization to access data
1
client_id, response_type, redirect_uri, state, scope
Displays scopes & prompts user to login (if required)
2
Logs in, authorizes access
3
code, state
Requests access and refresh tokens
4
client_id: client_secret, grant_type, code, redirect_uri
Returns access and refresh tokens
5
access_token, token_type, expires_in, refresh_token
Spotify Web API
Uses access token in requests to Web API
6
access_token
Returns requested data
(JSON object)
Spotify Accounts Service
Requests refreshed access token
7
client_id, client_secret, grant_type, refresh token
Returns new access token
access token

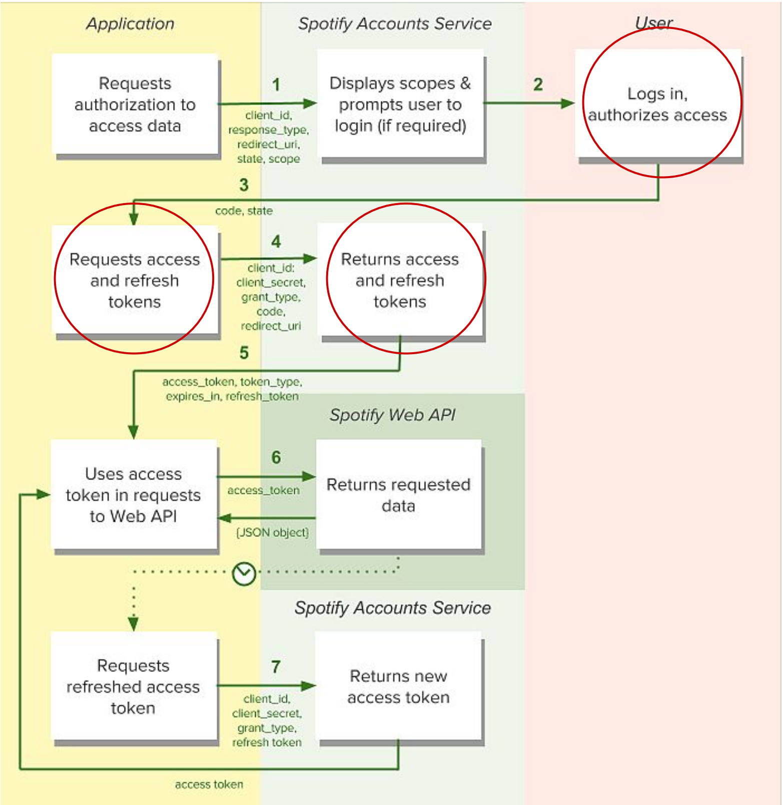
When the user clicks on the login button:

- Build the custom login url with scopes, client ID, and redirect URI
- It will ask the user if you would like the application to have access to these scopes

Logging in

- If the user successfully logs in (and authorizes the scopes you requested), the API kicks back to the redirect URI.
- In the redirect URI, you can decode the hashed access token, and then send that back to your main application.
- The generation of the unique access token lets you hit API endpoints, based on the default scopes from Spotify and scopes you asked the user for

Spotify Web API OAuth Guide



Application

Spotify Accounts Service

User

Requests authorization to access data

1

client_id, response_type, redirect_uri, state, scope

Displays scopes & prompts user to login (if required)

2

Logs in, authorizes access

3

code, state

Requests access and refresh tokens

4

client_id: client_secret, grant_type, code, redirect_uri

Returns access and refresh tokens

5

access_token, token_type, expires_in, refresh_token

Spotify Web API

Uses access token in requests to Web API

6

access_token

{JSON object}

Returns requested data

Spotify Accounts Service

Requests refreshed access token

7

client_id, client_secret, grant_type, refresh token

Returns new access token

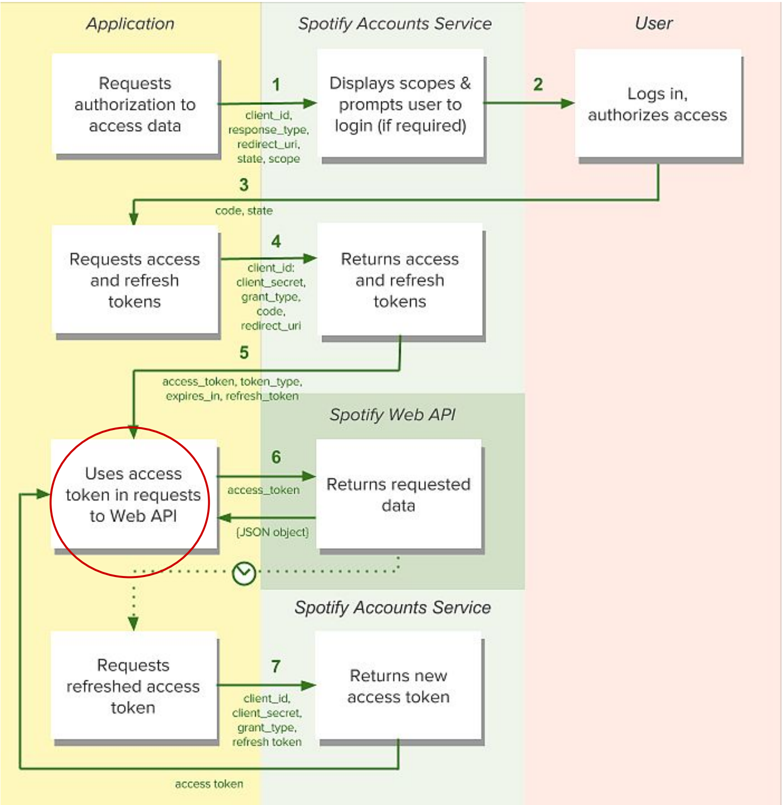
access token

Getting the token needed for authorized API calls

Access Token / API requests

- With the access token, we can make ajax calls to various API endpoints and get data back about the currently logged in user
- The access token only lasts for a limited amount of time, your browser might cache it after you login, which is fine
- If the token expires, you simply need to log in again

Spotify Web API OAuth Guide


Application
Spotify Accounts Service
User
Requests authorization to access data
1
client_id, response_type, redirect_uri, state, scope
Displays scopes & prompts user to login (if required)
2
Logs in, authorizes access
3
code, state
Requests access and refresh tokens
4
client_id: client_secret, grant_type, code, redirect_uri
Returns access and refresh tokens
5
access_token, token_type, expires_in, refresh_token
Spotify Web API
Uses access token in requests to Web API
6
access_token
Returns requested data
{JSON object}
Spotify Accounts Service
Requests refreshed access token
7
client_id, client_secret, grant_type, refresh token
Returns new access token
access token

Now we have a token and can make calls to the API endpoints

- Endpoints are URLs which you can send an AJAX request to and receive a response

API endpoints

https://developer.spotify.com/web-api/endpoint-reference/

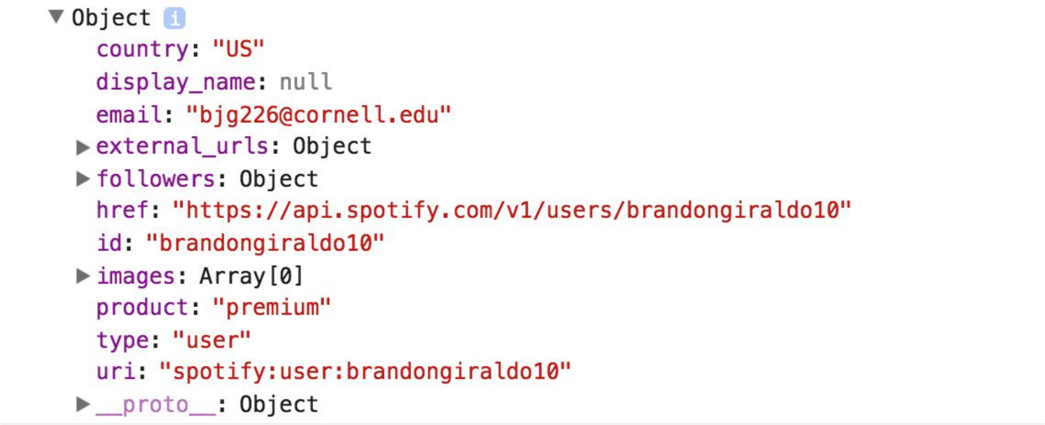
- Many endpoints to hit, each getting back unique data about the logged in user
- Base URL for the API - https://api.spotify.com
- Append your endpoint to this url, for example, getting the user’s saved albums

https://api.spotify.com/v1/me/albums
Note: this won’t return anything if a token wasn’t included

Response from API call

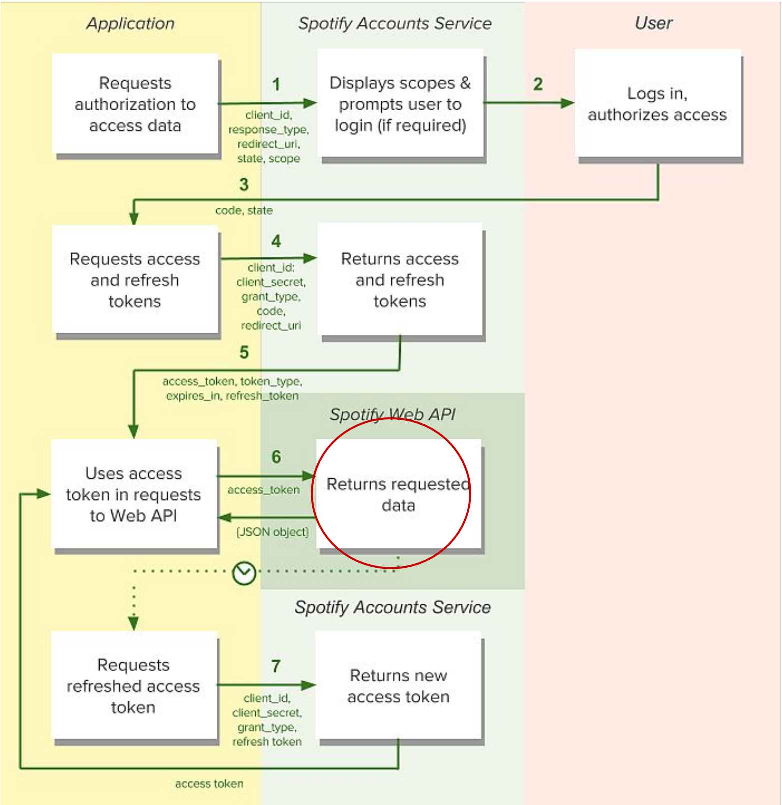
Making a successful call to the API, with the access token, will return a JSON object based on the url hit. This endpoint is for https://api.spotify.com/v1/me

> console.log(response);



```
▼ Object ℹ
    country: "US"
    display_name: null
    email: "bjg226@cornell.edu"
  ▶ external_urls: Object
  ▶ followers: Object
    href: "https://api.spotify.com/v1/users/brandongiraldo10"
    id: "brandongiraldo10"
  ▶ images: Array[0]
    product: "premium"
    type: "user"
    uri: "spotify:user:brandongiraldo10"
  ▶ __proto__: Object
```

Spotify Web API OAuth Guide


Application
Spotify Accounts Service
User
Requests authorization to access data
1
client_id, response_type, redirect_uri, state, scope
Displays scopes & prompts user to login (if required)
2
Logs in, authorizes access
3
code, state
Requests access and refresh tokens
4
client_id: client_secret, grant_type, code, redirect_uri
Returns access and refresh tokens
5
access_token, token_type, expires_in, refresh_token
Spotify Web API
Uses access token in requests to Web API
6
access_token
Returns requested data
{JSON object}
Spotify Accounts Service
Requests refreshed access token
7
client_id, client_secret, grant_type, refresh token
Returns new access token
access token

Getting back the response from the API call returns a JSON object

From here you can do anything with that object. In the case of homework 3, we would like to display 50 songs from a user’s library onto the DOM

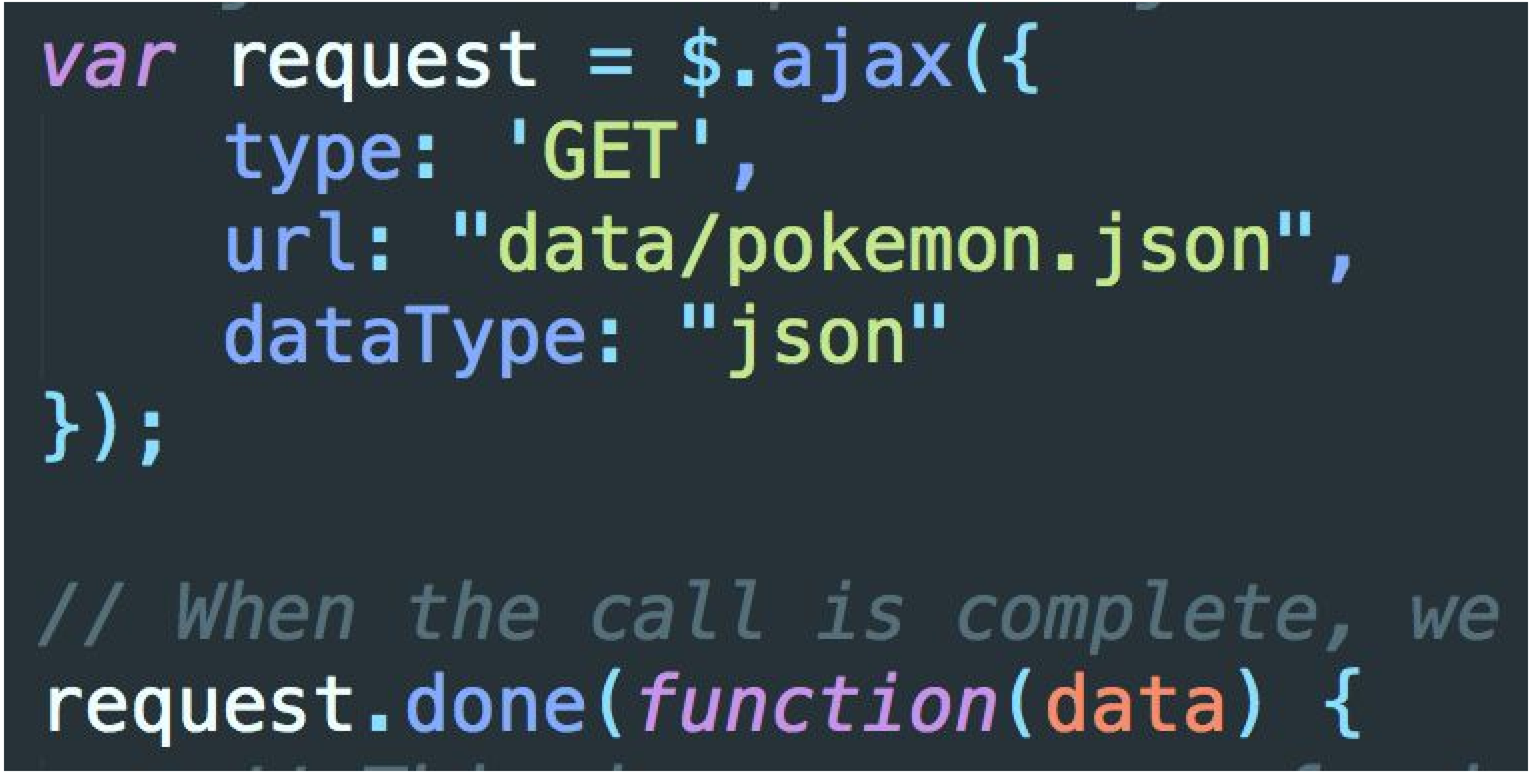
Application details

- Login function flow is important, will be outlined in more detail on hw3 rubric
- Remember access to config variables in spotify-config.js for other JS files
- api-call.js contain functions that **return a reference to an ajax call**, when you call these functions, you can bind an ajax function to them

Example: getUserData() returns an ajax function which then runs. You can wait for the ajax call to finish by attaching an event listener (such as .error) to the ajax call.

```
getUserData(accessToken).error(function(error) { ... });
```

Remember this?



```
var request = $.ajax({
    type: 'GET',
    url: "data/pokemon.json",
    dataType: "json"
});

// When the call is complete, we
request.done(function(data) {
```

Click In!

Click In!

What is your experience with the Myers Briggs personality type indicator? (INSP, ESFJ etc)

A.I've never heard of it

B.I've heard of it but not used it

C.I've used it but don't remember my type

D.I know my 4-letter type

Final Project Team Forming Survey:

`https://goo.gl/aI02Bk`

Personality: (2 points on P3_M3 if done by Tuesday 5pm)

`https://goo.gl/DJ854l`