Feature Request: Accurate Auth Token Expiry

**August 11th 2021**

**OBJECTIVE**

To ensure the authorisation token expiry time in the application logic matches the expiry time from Spotify, rather than starting the timing interval from the user’s next request.

**BACKGROUND**

Currently the Spotify token may expire before the application logic expects it to, leading to refused API requests. This detracts from user experience, and also leads to unnecessary calls to the Spotify server and increased traffic.

The current state of the app is that the clearing of the *Spotify.userAccessToken* is scheduled by calling a *setTimeout()* when *Spotify.getAccessToken()* is called, so long as the current URL contains *access\_token* and *expires\_in* parameters. This may be some time after the request for the access token was made.

The feature will:

* Keep track of the expiry time sent from Spotify in response to a request for an authorisation token as soon as it is returned.
* Log the user in on page load, meaning the remaining user experience is not interrupted.

**TECHNICAL DESIGN**

To implement the required behaviour, the URL should be checked immediately on loading, and a timeout set if the required parameters are present. Directing the browser to the Spotify endpoint in *getAccessToken()* is an essential part of requesting an access token, and since setting *window.location* loads a new page no logic placed after this would run and so would not provide a solution.

After investigating the behaviour of the app, it is clear that the *App* component is only mounted on page load, and could therefore be used to provide an initial check of the URL, conditionally setting the timeout for the access token if appropriate. Any time a new access token is sent, the page will be reloaded, so this would be the only place this check is required.

A **componentDidMount()** method should be created in the *App* component with a single line body being a call to *Spotify.getAccessToken()*. This will call the existing code, which then populates the *Spotify.userAccessToken* variable and sets the timeout immediately.

**CAVEATS**

This solution does entail the user logging in to their Spotify account as soon as the application loads, which may not be desirable if the user only wishes to use the search function as this does not require a login. As Jamming currently does not offer playback, it is believed that this ‘search only’ use-case is sufficiently unlikely, that the proposed solution is still optimal. It also has the benefit of making efficient use of existing code.

If it becomes necessary to implement a different approach, the following is suggested:

Refactor the functionality of checking the URL and setting a timeout which currently makes up part of the *Spotify.getAccessToken()* method into a separate method, **setAccessTimeout()**, which only carries out the parsing of the URL and the conditional setting of the timeout for *userAccessToken*. This could then be called from within *getAccessToken()* as a condition eg *if (setAccessTimeout()) {…}.* This new *setAccessTimeout()* method could then be called in *App.componentDidMount()* instead of *getAccessToken().*