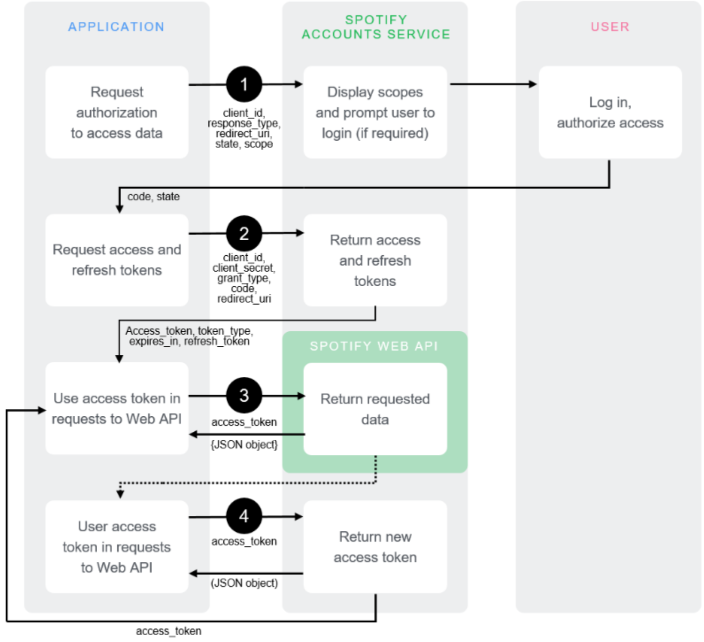
Dev Prep for Spotify API Token Authorisation

Using the authorisation code with PKCE OAuth flow, we get access to user resources and token refresh but are not required to store a secret key server-side.

The flow works like this (full guide on https://developer.spotify.com/documentation/general/guides/authorization/code-flow/):


APPLICATION
SPOTIFY ACCOUNTS SERVICE
USER
Request authorization to access data
1
client_id, response_type, redirect_uri, state, scope
Display scopes and prompt user to login (if required)
Log in, authorize access
code, state
Request access and refresh tokens
2
client_id, client_secret, grant_type, code, redirect_uri
Return access and refresh tokens
Access_token, token_type, expires_in, refresh_token
SPOTIFY WEB API
Use access token in requests to Web API
3
access_token
{JSON object}
Return requested data
User access token in requests to Web API
4
access_token
(JSON object)
Return new access token
access_token

authorization code flow

Figure 1: From https://developer.spotify.com/documentation/general/guides/authorization/code-flow/

1. Get authorisation from the user. The app must build and send a GET request to the /authorise endpoint with these parameters: client_id (set to the one our app is assigned); response_type set to 'code'; redirect_uri set to the main page; state (to protect against cross-site forgery; show_dialog set to false; code_challenge_method set to S256 and code_challenge set to the hash of the code verifier using SHA256.
2. If the user accepts then the app can exchange the token by making a POST request to the /api/token endpoint. It needs these parameters encoded in application/x-www-form-urlencoded: grant_type set to 'authorization_code'; code set to the return of the previous request; redirect_uri which must match the one before; client_id as before and code_verifier must match the code_verifyer in the previous step. It must also include HTTP headers: Authorisation, in the format 'Authorization: Basic <base64 encoded client_id:client_secret> and Content-Type set to 'application/x-www-form-urlencoded'.
3. To refresh a token a POST request is sent encoded in application/x-www-form-urlencoded with the parameters: grant_type as 'refresh_token'; refresh_token set to the token we already have; client_id as before; and a header of Content-Type set to application/x-www-form-urlencoded.