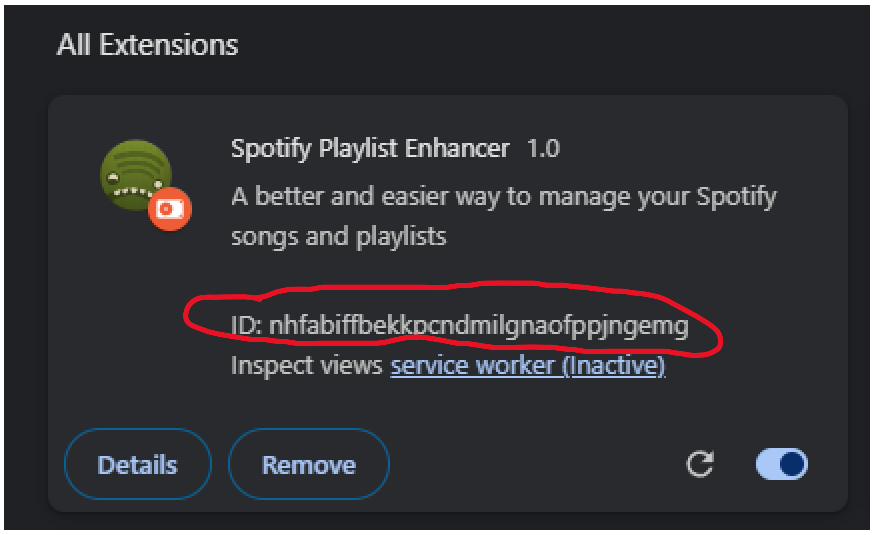
CS322 Music Programming Project
Spotify Account Authentication

Since the project scope was for a university module, the application has not been published to the public through Chrome Web Store, therefore people who clone the GitHub repository will not have access to authentication of their Spotify account for the following reasons:



All Extensions

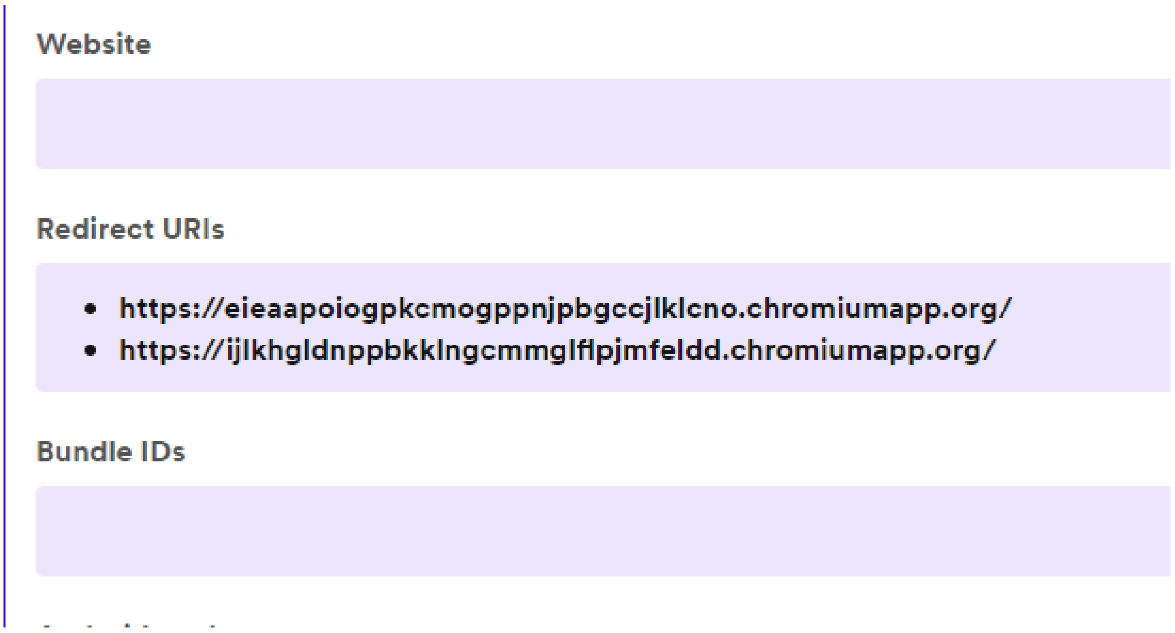
Spotify Playlist Enhancer 1.0

A better and easier way to manage your Spotify songs and playlists

ID: nhfabiffbekkpcndmilgnaofppjngemg

Inspect views service worker (Inactive)

When the extension is installed, Google Chrome give you a TEMPORARY ID for the build folder that you created as a developer. This ID must be added to Spotify Developer Dashboard like this in order to be able to connect to the actual Spotify API and perform calls:



Website

Redirect URIs

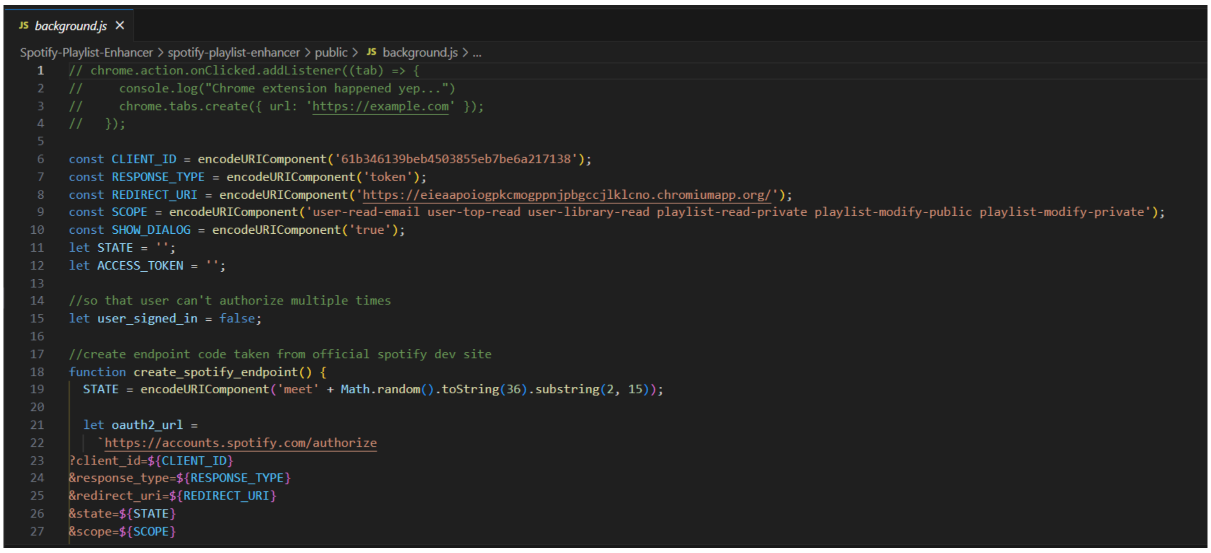
- https://eieaapoiogpkcmogppnjpbgccjlklcno.chromiumapp.org/
- https://ijlkhgldnppbkklngcmmglflpjmfeldd.chromiumapp.org/

Bundle IDs

So every time a new machine downloads the extension locally and creates the build folder, Chrome gives you a new ID, which has to be added again to the above place in a screenshot.

This issue is solved when the application is actually published onto Chrome Web Store, because Google Chrome gives you a PERMANENT ID that you put in the mentioned space above.

If you really want to run the application and authenticate your Spotify account to test all of the features of the app, there can be a slight work around for you. You can try and hardcore the Extension URI inside the background.js file like shown in the screenshot below:


background.js
Spotify-Playlist-Enhancer > spotify-playlist-enhancer > public > background.js > ...

```javascript
// chrome.action.onClicked.addListener((tab) => {
//     console.log("Chrome extension happened yep...")
//     chrome.tabs.create({ url: 'https://example.com' });
//   });

const CLIENT_ID = encodeURIComponent('61b346139beb4503855eb7be6a217138');
const RESPONSE_TYPE = encodeURIComponent('token');
const REDIRECT_URI = encodeURIComponent('https://eieaapoiogpkcmogppnjpbgccjlklcno.chromiumapp.org/');
const SCOPE = encodeURIComponent('user-read-email user-top-read user-library-read playlist-read-private playlist-modify-public playlist-modify-private');
const SHOW_DIALOG = encodeURIComponent('true');
let STATE = '';
let ACCESS_TOKEN = '';

//so that user can't authorize multiple times
let user_signed_in = false;

//create endpoint code taken from official spotify dev site
function create_spotify_endpoint() {
  STATE = encodeURIComponent('meet' + Math.random().toString(36).substring(2, 15));

  let oauth2_url =
    `https://accounts.spotify.com/authorize
?client_id=${CLIENT_ID}
&response_type=${RESPONSE_TYPE}
&redirect_uri=${REDIRECT_URI}
&state=${STATE}
&scope=${SCOPE}
```

The application works without issues on the developers machine because he has the redirect URI's set up already with the temporary ID's hardcoded into the background file and the Spotify Developer Dashboard. You can see the full demonstration of this in the Project Demonstration Video.