

## TP1 : App utilisant le SDK Spotify

Objectifs :

- Créer une app diffusant de la musique à l'aide du SDK Spotify
- Afficher des informations relatives à la chanson jouée
- Séparer le modèle de la vue du modèle
- Utiliser des techniques vues en classe ( `startActivityForResult`, intents externes )
- Apprendre le bon fonctionnement du SDK à l'aide des fichiers d'aide

Nous développerons une app qui utilisera particulièrement la librairie App Remote du SDK de Spotify, c'est-à-dire qu'on utilise Spotify pour diffuser la musique et pour utiliser les contrôles de base ( `PLAY`, `PAUSE`, `NEXT` ... )

Avec cette librairie, on ne peut pas faire de recherche avancée.

Quel sera l'avantage de notre app ? Elle devra cibler une clientèle particulière.

EX : Un Spotify belge → me permet d'écouter des chansons d'artistes belges

Un Spotify « Années '70 » → me permet d'écouter des grands succès de cette décennie

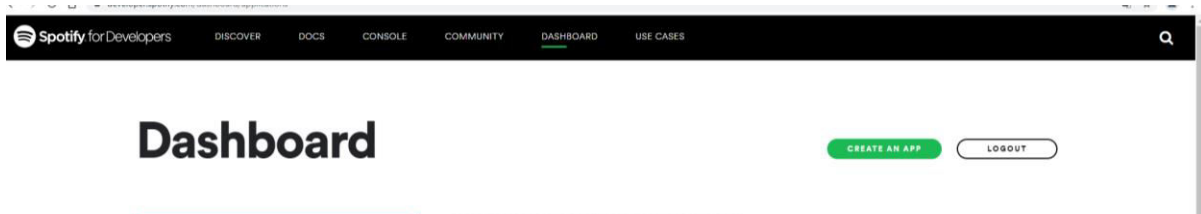
Vous devez donc trouver votre « thème » qui devra se voir dans :

- Les choix de couleurs / polices / images de votre app
- Les choix que vos utilisateurs devront faire : ( par exemple, sur le Spotify belge, l'utilisateur devra choisir s'il veut écouter du Stromae, du Hooverphonic ou du Brel )

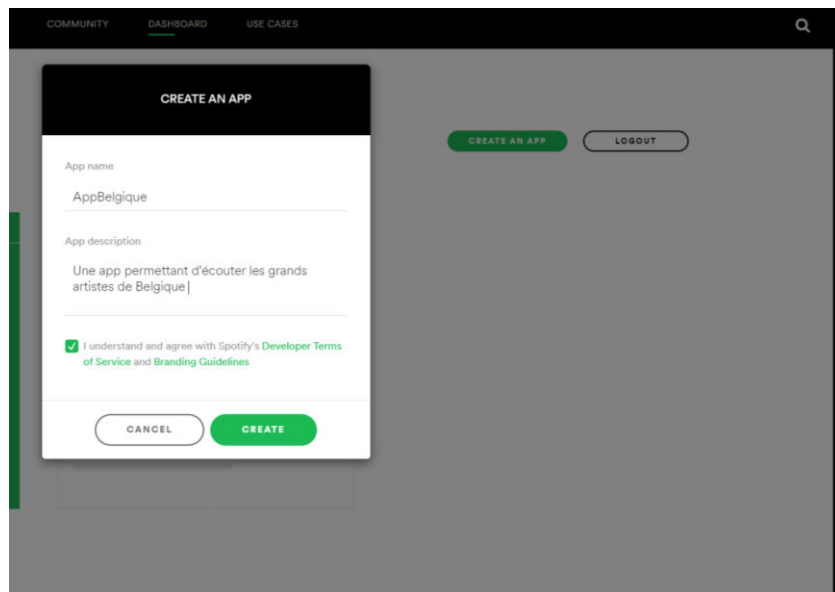
### 1. Préalables

- Se créer un compte gratuit sur Spotify ( ou utiliser son compte payant )  
→ <https://www.spotify.com/ca-fr/>
- ( À la maison ) Installer l'app Spotify sur votre émulateur / votre téléphone
  - Aller sur Google Play ( vous aurez besoin d'un compte Google )
  - Installer Spotify
  - Se loguer à son compte sur Spotify
- ( à l'école )
  - Démarrez Spotify sur l'émulateur et se loguer à son compte

- Créer son squelette de projet sur AndroidStudio ( pour avoir son nom de package, entre autres )
- Devenir « développeur Spotify » en allant sur <https://developer.spotify.com/dashboard/> et se loguer sur son compte.



- Cliquer sur Create an App :



- Notez le client ID qui vous est produit dans la page web
- Cliquez sur Edit Settings :

The screenshot shows the 'App Information' page in the Google Play Console for an application named 'AppBelgique'. The page is divided into several sections, each with a title and a description, followed by a list of values and an 'Add' button. The sections are: 'Application name' (AppBelgique), 'Application description' (Une app permettant d'écouter les grands artistes de Belgique), 'Website' (Add a website), 'Redirect URIs' (https://example.com/callback/), 'Bundle IDs' (com.example.myapplication), and 'Android Packages' (01D8AA439759EEC595266A07EE3C).

Field	Value
Application name	AppBelgique
Application description	Une app permettant d'écouter les grands artistes de Belgique
Website	Add a website
Redirect URIs	https://example.com/callback/
Bundle IDs	com.example.myapplication
Android Packages	01D8AA439759EEC595266A07EE3C

Remplir les champs suivants :

### RedirectURI

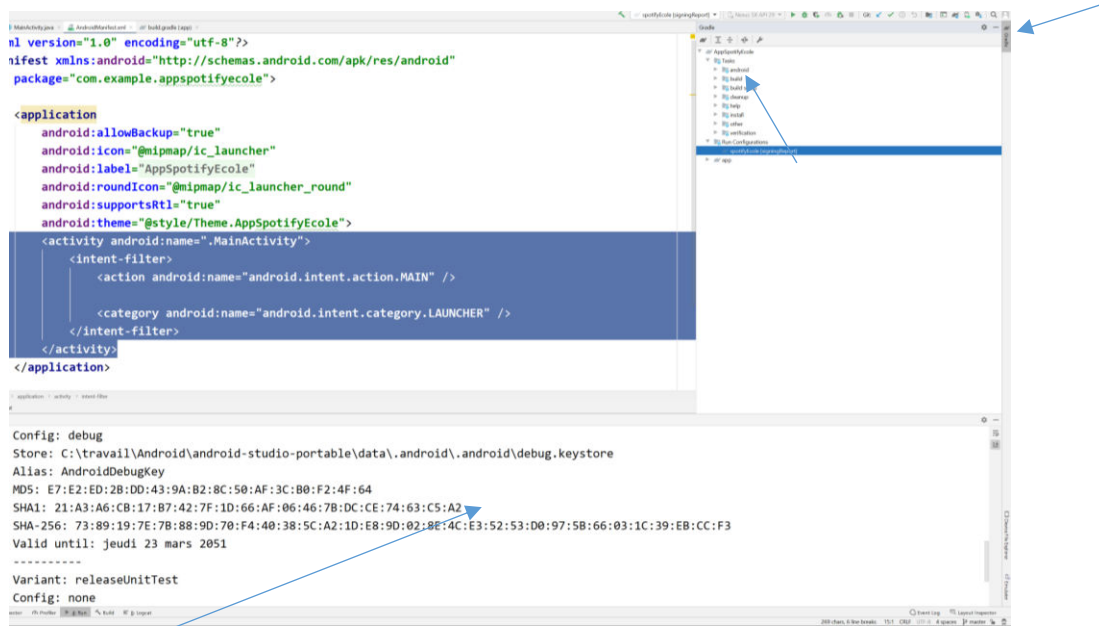
Écrire : **nom de votre package://callback**

### Android Packages

Écrire le nom du package de votre projet

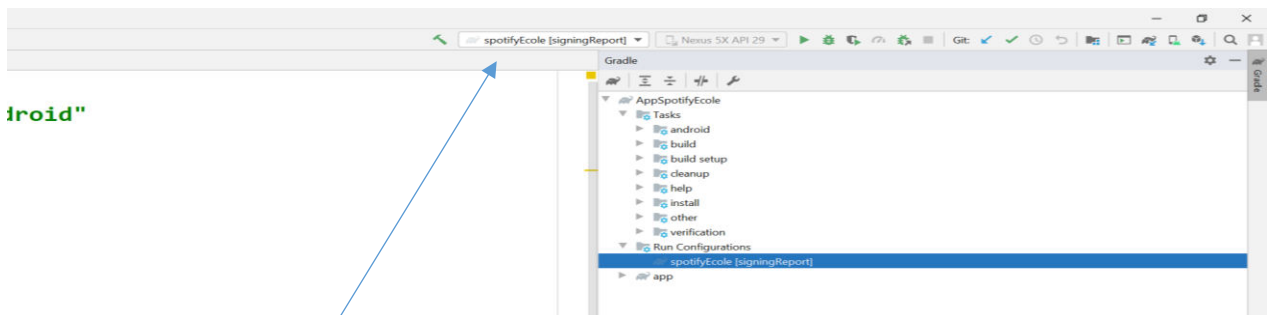
Écrire également le code SHA1 ( clé encryptée pour votre app )

Vous pouvez la générer ( J'ai cherché longtemps, va falloir zoomer ) :



Ouvrir la section android → faire signingReport

\*\*\* important : remplacer le « run » par le nom de votre projet pour faire rouler votre app par la suite \*\*\*



Changer pour app

## 2. Télécharger le sdk et l'intégrer à notre projet :

Spotify SDK : <https://github.com/spotify/android-sdk/releases>

Placer ce fichier dans un dossier présent dans le dossier libs de votre app :

Par exemple : app\libs\spotify-app-remote\spotify-app-remote-release....aar

L'intégrer dans notre projet ( fichier build.gradle – module ) :

```
dependencies {

    implementation files('libs/spotify-app-remote/spotify-app-remote-release-0.7.2.aar')
    implementation "com.google.code.gson:gson:2.8.5"
    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

### 3. Le travail comme tel

En classe : explication du fichier d'exemple

<https://developer.spotify.com/documentation/android/quick-start/#source-code>

Votre travail :

- A) Trouver le thème de votre app et faire les recherches sur Spotify pour trouver les liens correspondants aux playlists , artistes ou chansons que vous désirez utiliser

Exemples :

spotify:playlist:37i9dQZF1DX2sUQwD7tbmL

spotify:artist:2MLEz0a0JzhNpbqXn2UxgG

- B) Développer un modèle de données crédible : classe Chanson, classe Artiste, etc...

Obligatoire : classe SpotifyDiffuseur qui contiendra tous les variables d'instances et les actions relatives à la diffusion de la musique :

Par exemple :

- Variables : objet PlayerApi, objet SpotifyAppRemote, objet Context...
- Méthodes : play/pause, suivant, précédent, avancer d'un certain nombre de secondes, afficher des infos sur la chanson ou sur l'artiste, etc

C) En ce qui a trait à votre / vos activités, elles devront bien sûr utiliser les variables / méthodes de votre modèle de données. Elles devront inclure :

- Un intent de type Boomerang ( Annexe 2 )
- Un lien hypertexte vers un site ayant un lien avec votre thème ( Annexe 5 )
- Une liste complexe ( Annexe 4 )
- Un thread permettant le déplacement d'une seekBar suivant la progression de la chanson ( Annexe 6 ) ainsi qu'un décompte min :sec
- La diffusion de différentes chansons, à l'aide de touches PLAY / PAUSE / SUIVANT / PRÉCÉDENT
- Une représentation visuelle de la chanson ( pochette, nom de l'artiste, nom de la chanson...et c 0

**3. Grille de correction sommaire :**

Élaboration du thème	8 pts
Diffusion des chansons, affichage des infos relatives, originalité et compréhension du code	42 pts
Séparation du modèle et de la Vue	15 pts
Commentaires pertinents	7 pts
Utilisation du « boomerang »	10 pts
L'app continue de diffuser malgré autres événements sur le téléphone / robustesse / cycle de vie	8 pts
Lien vers artiste ou autres	5 pts
Design de la ou des Activités vs taille du téléphone	5 pts

**4. Date de remise : Mercredi 26 Octobre**