420-4W6

TP2

Super Musique Infinie

Date de remise : Semaine 7

Pondération : 20%

# Consigne

Le projet sera fait de façon individuelle.

Le projet sera fait avec le Framework Angular.

# Mise en contexte

Vous devez faire une gestion de contenu musique utilisant différents APIs.

Vous devrez ajouter des artistes, puis aller chercher leurs albums à l’aide de l’API de Spotify.

À partir des artistes, vous devrez aussi aller chercher leurs spectacles à l’aide de l’API de Bandsintown.

À partir d’un album vous devrez aussi lister des chansons aussi à l’aide de l’API de Spotify.

À partir des spectacles d’un artiste, vous devrez afficher sur une carte l’emplacement du spectacle à l’aide de Google Maps API.

À partir d’une chanson, vous devrez être en mesure d’utiliser Youtube pour faire jouer la chanson.

# Consignes générales

L’application devra être disponible en Anglais et en Français.

# Page d’accueil (Liste d’artistes)

À partir de la page d’accueil, l’utilisateur pourra ajouter des artistes à une liste d’artistes. Cette liste sera persistée (sauvegardée) dans le localStorage du navigateur.

Une requête à Spotify permettra de récupérer des informations sur l’artiste. Vous devrez afficher le nom et une image de chaque artiste.

À partir de l’artiste nous pourrons aller voir les spectacles prévus de l’artiste ou encore aller voir les albums de l’artiste.

# Page des concerts

Une page permettra d’afficher les concerts d’un artiste. Nous récupérerons les concerts à partir d’une requête à l’API de BandsInTown.

Le nom de l’artiste devra être fourni en paramètre dans le routage du composant.

Un carte Google Maps permettra de voir l’emplacement des concerts.

# Page des albums

Une page permettra d’afficher les albums d’un artiste. Nous récupérerons les albums à partir d’une requête à l’API de Spotify. Vous devrez afficher le nom et une image de chaque album.

Le nom de l’artiste (ou son ID) devra être fourni en paramètre dans le routage du composant.

# Page des chansons

Une page permettra d’afficher les chansons d’un album. Nous récupérerons les chansons à partir d’une requête à l’API de Spotify. Vous devrez seulement afficher le nom de chaque chanson.

Lors du clic sur une chanson, une requête à YouTube sera effectuée pour avoir accès à la chanson. Pour la requête, il est suggéré de placer le nom de la chanson et l’artiste dans la recherche.

Une fois la requête effectuée, il sera possible d’afficher une vidéo YouTube de la chanson. Pour ce faire, nous utiliserons le Id de la vidéo et un iFrame.

Au minimum, le id de l’album devra être fourni en paramètre dans le routage du composant.

# Correction

|  |  |  |  |
| --- | --- | --- | --- |
| /110 | Super Musique Infinie | | |
| Pour chaque item (Artiste, Album et Chanson) | | | |
|  | 2 pts | Classe présente | |
|  |  |  | Utilisation de la classe |
|  | 5 pts | Requête HTTP pour aller chercher les informations | |
|  |  |  | Lecture de la réponse |
|  | 3 pts | Afficher une liste d'item | |
|  |  |  | Afficher l'image de l'item lorsque disponible |
|  | 2 pts | Component pour afficher l'item | |
|  | 3 pts | Routage fonctionnel pour afficher l'item | |
|  |  |  | Il est possible de taper direct l'url dans la barre d'adresse |
| Total | 45 pts | (15 pts x 3) | |
| Concerts | |  |  |
|  | 5 pts | Requête HTTP pour aller chercher les informations | |
|  |  |  | Lecture de la réponse |
|  | 2 pts | Component pour afficher l'item | |
|  | 3 pts | Routage fonctionnel pour afficher l'item | |
|  |  |  | Il est possible de taper direct l'url dans la barre d'adresse |
|  | 10 pts | Afficher les concerts dans une carte Google (ou une carte pour chaque concert) | |
| Total | 20 pts |  |  |
| Autres consignes | | | |
|  | 15 pts | Utiliser au moins un service pour les requêtes HTTP | |
|  |  |  | Le service devrait retourner des Observable ou un Promise |
|  |  |  | Le .subsbribe (ou l’équivalent avec async-await) devrait être dans le component |
|  | 5 pts | La liste d'artistes doit être sauvegardée dans le localStorage | |
|  |  |  | La liste devrait être chargée lorsque l'on revient sur l'application |
|  | 10 pts | L'application doit être disponible en anglais et en français | |
|  | 15 pts | À partir d'une chanson, on doit être capable de récupérer sont ID sur Youtube | |
|  |  |  | Il est possible de faire jouer une chanson à partir d'un iFrame et d’une variable ID |
|  |  |  | Un Pipe est utilisé pour autoriser l'URL |
| Total | 45 pts |  |  |

# URLS et doc

YouTube Search API

<https://developers.google.com/youtube/v3/docs/search/list>

Example d’utilisation

[https://www.googleapis.com/youtube/v3/search?part=id&maxResults=1&key={API\_KEY}&q={MA\_RECHERCHE}](https://www.googleapis.com/youtube/v3/search?part=id&maxResults=1&key=%7bAPI_KEY%7d&q=%7bMA_RECHERCHE%7d)

Vous devrez vous-même créer une clé d’API pour utiliser les API de Google

BandsInTown

<https://app.swaggerhub.com/apis/Bandsintown/PublicAPI/3.0.0#/artist%20events/artistEvents>

Clé pour utiliser l’API de BandsInTown

**2b32475766802ac01eefda45e9e42ea0**

<https://rest.bandsintown.com/artists/MON_ARTISTE/events?app_id=API_KEY>

Spotify API

<https://developer.spotify.com/documentation/web-api/reference/#/>

Exemple d’utilisation de l’API

https://github.com/CEM-420-4W6/Spotify