



Université de Cergy-Pontoise

RAPPORT

pour le projet de Développement Web

Licence d'Informatique deuxième année

allSoundS

rédigé par

Ahmed BOUZIDIA \* Dihia BELARBIA



07-05-2022

## **Table de Matiere :**

1. Introduction
  - 1.1 Objectif du Projet
  - 1.2 Contexte du Projet
2. Spécification du Projet
  - 1.1 A propos de l'API
  - 1.2 Réalisation
    - 1.2.1 Détails et architecture du site
  - 1.3 Notions de base
3. Plan du site
4. Répartition de tâches
5. Conclusion
  - 4.1 Amélioration Possible

# 1. Introduction:

## 1.1 Objectif du Projet:

Dans le cadre du projet de développement web, nous avons créé un site web en utilisant les langages HTML 5 / CSS 3 / PHP 7 et en utilisant les ressources de l'API deezer.

## 1.2 Contexte du Projet:

Le but était essentiellement de créer un site web qui propose de la musique à l'utilisateur selon les dernières tendances ainsi que plusieurs genres musicaux.

# 2. Spécification du Projet:

## 1.1 A propos de l'API:

Nous avons utilisé l'api deezer (deezer for développeurs), elle permet de récupérer les informations relatives aux artistes, musiques, albums, playlists ainsi que les podcasts et la radio, les informations sont transférées avec le format de transfert de données JSON.

Les données d'une musique sont: le titre, l'artiste, le cover de l'album ainsi qu'un extrait du son.

Pour les playlists, les albums, et les artistes, nous avons affiché les image de chacun qui mène vers une autre page qui fait une requête pour afficher la liste des titres (tracklist).

- <https://developers.deezer.com/api>

La requête json vers l'API Deezer est un élément fondamental pour le fonctionnement du site web, donc chaque flux de données JSON est le résultat

d'une requête vers L'API.

Les requêtes sont spécifiques à l'élément recherché et au type d'informations qu'on a choisis.

chaque playliste, album ou artiste est identifié dans l'API grâce à un identifiant numérique, c'est une valeur entière qui est unique pour chaque élément, en utilisant cette identifiant on a pu faire principalement les requête suivante:

Top 10 des artistes, albums, et playlists ainsi que les podcasts, se sont les plus populaires donc les plus consultés d'après l'api deezer.

Les requêtes qui sont faites lors d'une recherche, elles renvoient 4 différents flux pour chaque type d'information.

Les filtres qui sont appliqués après les résultats de recherche, permettent de détailler la recherche de l'utilisateur selon 4 types différents qui sont précédemment identifiés.

Le flux qui permet d'afficher la liste de son de chaque type d'information car chaque identifiant à une liste de musique (tracklist) qui permet par exemple d'accéder au contenu de l'album ou le contenu de la playliste.

## 1.2 Réalisation

Dans la partie réalisation, nous allons aborder la manière dont nous avons travaillé tout au long du projet et de différente partie du travail.

La première partie de ce projet nous a permis de nous familiariser avec les deux formats d'échange auxquels on a eu recours pendant ce projet, notamment le format d'échange JSON.

Pour appliquer le format json, on a réalisé un affichage de l'image ou la vidéo du jour de la Nasa, on récupère les informations qui nous intéressent à partir de l'API Nasa sous le format JSON puis on manipule ce résultat avec le langage php pour afficher le résultat final sur un navigateur web, ainsi que l'affichage de la position de l'utilisateur en utilisant l'API Geoplugin.

Dans un second temps nous nous somme essentiellement servie de l'Api deezer pour extraire les données qui sont représentées sous le format JSON et qui seront manipuler en php dans le code source de la page web (côté serveur) puis afficher en langage HTML5 et css 3 du coté client(serveur web).

Chaque fonction est implémentée en php pour faire une requête JSON précise et renvoie le résultat de cette requête sous le format HTML, nous avons donc procédé de cette manière:

- Faire la requête
- Récupérer le contenu JSON résultant de la requête
- Transformer ce résultat en un Tableau en utilisant la fonction native `jsondecode()`.
- Récupérer les informations qui nous intéressent
- Créer un élément HTML (list, tableau, h1, h2..) à partir de ce résultat ou créer un élément php
- Renvoyer le résultat (php ou/et html).

### 1.2.1 Détails et architecture du site:

Notre site est constitué de 6 fichiers php essentiels, ainsi que le fichier header, footer et le fichier contenant les fonctions utilisés dans l'intégralité du projet.

- `Index.php`: la page d'accueil où on affiche premièrement les albums, artistes et les playlists récemment consultés si ce n'est pas la première fois que le site est visité, ensuite on propose quelques tendances de chaque type d'information avec un bouton permettant d'afficher plus.

Une liste sur le côté permettant de consulter les top 10 de chaque type de flux, qui donc envoie vers la page `top10.php`.

- `search.php`: un formulaire de recherche qui fait une requête et renvoie à la même page des flux html, selon les 4 résultats de recherche.  
Une liste sur le côté qui permet de filtrer les informations renvoyées sur une autre page, `filtre.php`.
- `filtre.php`: fait le filtre de la recherche.
- `top10.php`: dans ce fichier le code php permet de renvoyer un flux avec les nouveautés de chacun des 4 types d'information.
- `tracks.php`: la fonction appelée dans ce fichier est "`rechercherMusic()`" qui renvoie dans la page web du navigateur les musiques que contient un album ou une playliste ou les chansons d'un chanteur sélectionné à partir d'un id.
- `genre.php`: renvoie un flux d'artistes selon le genre sélectionné dans un formulaire html, selon une liste (pop, jazz, dance, r&b, rock, folk, reggae).

## 1.3 Notions de base:

Nous avons utilisé plusieurs notions durant la réalisation du projet:

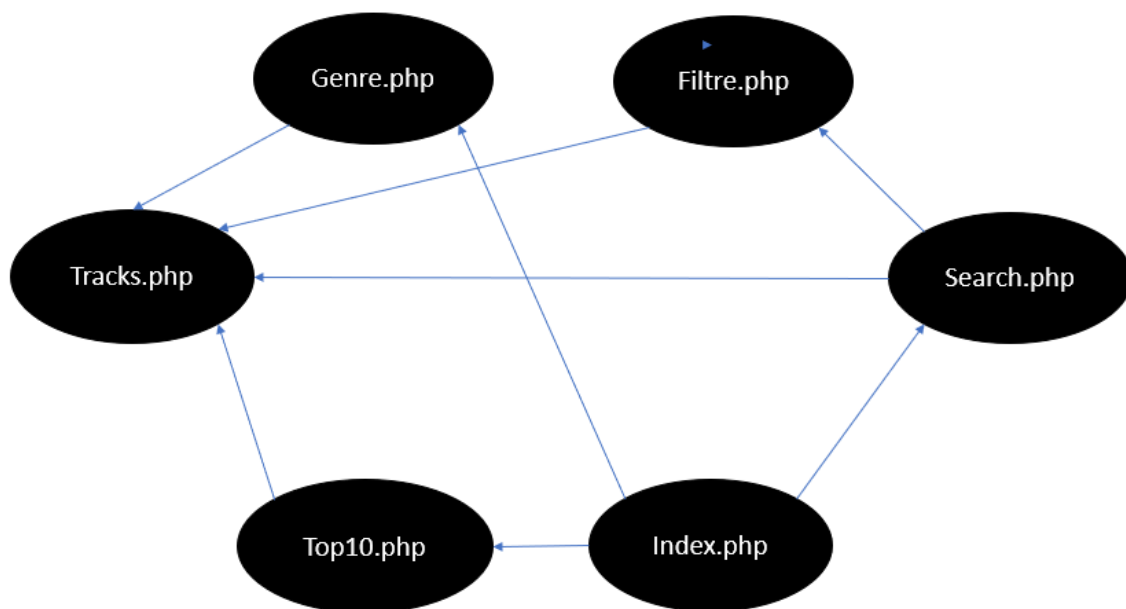
- Les cookies: On a mis en place un cookie qui stocke les dernières informations consultées, il a une durée de vie de 1 an et il est implémenté à chaque visite du site.

Pour les films et les séries, on a utilisé le même principe pour stocker le dernier

album consulté, en effet on a créé un cookie qui stocke l'id du dernier album consulté puis grâce à cet id et à l'aide d'une fonction on a réussi à afficher les données qu'on voulait du film en question, le traitement est similaire pour le dernier artiste et dernière playlist consulté.

- Les session: on a utilisé les session pour stocker dedans le mot recherché (récupéré depuis la barre de recherche) pour pouvoir l'utiliser dans les autres pages, donc ne pas le perdre.

### 3. Plan du site:



### 4. Répartition de tâches:

Pour la répartition des différentes tâches nous sommes coordonnés de sorte à se voir régulièrement pour avancer ensemble sur le projet.

Planning de Réalisation :

Tâche réalisée	Semaine
html ,Css , api Nasa	semaine 12
affichage de différentes type d'information (album, artistes, playliste et podcaste)	semaine 14
Réalisation complète	semaine 18

## 5. Conclusion:

A l'issue de ce projet nous avons pu mettre en œuvre les solutions citées précédemment, en effet , nous avons pu comprendre et appliquer le format JSON, Ainsi, nous avons réussi à créer le site qui permet de proposer les musiques qui sont en tendances et les albums qui sont récemment consultés et autre.

### 4.1 Amélioration Possible:

Nous n'avons pas pu gérer l'erreur qui est affichée lorsqu'on fait une recherche contenant un espace, principalement par manque de temps.