

Plan

1 Introduction

- Buts du projet
- API utilisées

2 Serveur

- Généralités
- Route events
- Route infos
- Route tracks
- MongoDB
- Déploiement sur AWS

3 Client

4 Conclusion

Introduction

Buts du projet

Créer un moteur de recherche d'événements musicaux, permettant à l'utilisateur de :

- Afficher des événements sur une carte interactive
- Afficher des informations d'un événement en particulier
- Afficher des informations à propos des artistes
- Jouer (en arrière plan) un extrait d'un son d'un artiste de l'événement

Introduction

API utilisées

- Spotify : recherche d'artistes et top tracks (route events, infos et tracks)
- Eventful : liste des événements (lieux, dates et artistes) (route events)
- Wikipédia : principale source d'informations sur un artiste (route infos)
- MusicBrainz : informations complémentaires sur les artistes (route infos)
- BandsInTown : informations complémentaires sur les artistes (route events et infos)
- Google Maps : pour la carte interactive côté client

Serveur

Node.js et Promesses

Serveur Node.js avec structure basée sur les promesses (Promise) :

```
1  let doSomething = function(data) {  
2      return new Promise((resolve, reject) => {  
3          if (test) resolve(data);  
4          else reject("error");  
5      });  
6  }  
7  doSomething()  
8  .then(result => { return doSomethingElse(result); })  
9  .then(finalResult => { console.log("OK :" + finalResult); })  
10 .catch(error => { console.log(error); });
```

Serveur

Route events

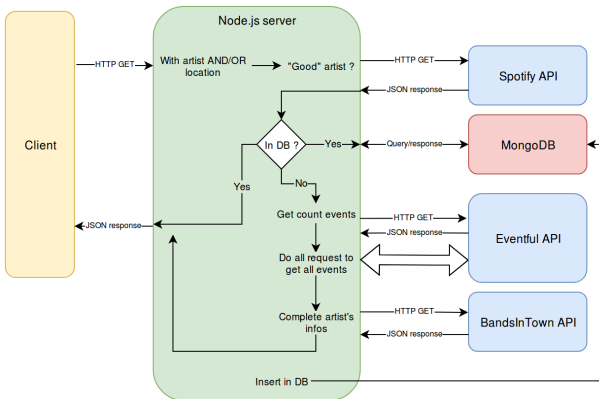


Figure – Route events

Serveur

Route infos

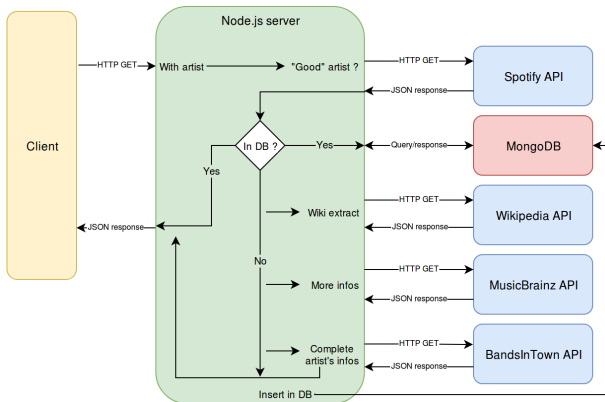


Figure – Route infos

Serveur

Route tracks

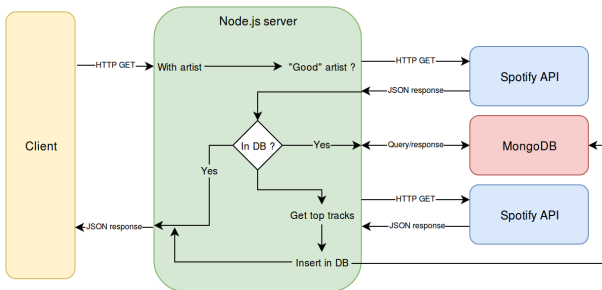


Figure – Route tracks

Serveur

MongoDB

Utilisation de MongoDB comme base de données :

- Simplicité/compatibilité avec Javascript et Node.js (JSON)
- Données "temporaires" avec la feature MongoDB Time To Live
- Découverte d'un SGBD alternatif (NoSQL)

Serveur

Déploiement sur AWS

Utilisation du cloud AWS d'Amazon (avec scripts shell pour le déploiement) :

- Pour héberger le serveur Node.js
- Pour héberger la base de données MongoDB
- Pour héberger la documentation de l'API
- Pour héberger le serveur Apache pour servir les fichiers client (HTML)
- Mettre en pratique le cours de Cloud

Client



texte important

