



POLYTECH[®]
NICE-SOPHIA

Projet 'MultiTrack'

Modules : Programmable Web Server Side - Client Side

Etabli par :

**GUIZANI Mounir
JOMAA Farouk
DIALLO Mahmoud
MAGROUN Hamza**

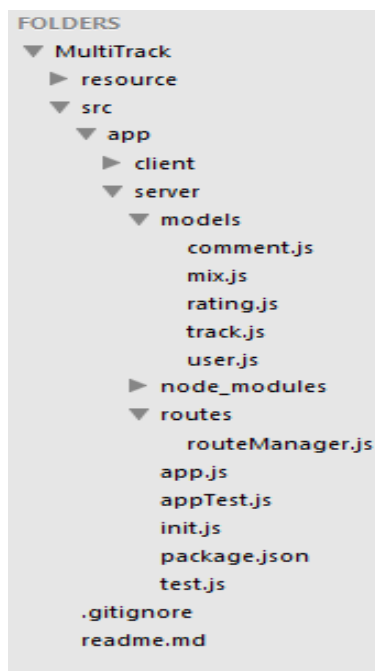
Année universitaire : 2015-2016

I. Introduction

Le présent document a pour objectif de clarifier en premier lieu la structure de notre projet côté serveur. Dans une deuxième étape, nous allons expliquer les étapes à suivre pour installer notre application.

II. Structure du projet

Cette figure illustre la structure générale de notre projet :



L'implémentation côté serveur de notre application a suivi cette approche qui permet la :

- Création des fichiers contenant les différents modèles qui sont en relation avec notre cas d'utilisation, qui sont : **user.js**, **comment.js**, **track.js**, **rating.js**, **mix.js**.
- Création du fichier **routeManager.js** qui contient toutes les méthodes de gestion qu'on aura besoin pour établir des interactions avec la base des données.
- Création du fichier **app.js** qui permet à travers la commande "**node app.js**" d'exécuter notre application dans sa base des données principale.
- Création du fichier **init.js** qui permet l'insertion des données initiales nécessaires pour garantir le bon fonctionnement de notre application. À titre d'exemple, ce fichier permet

l'insertion des données relatives au premier utilisateur de l'application, qui est un administrateur (le seul administrateur).

- Création du fichier **appTest.js** qui permet à travers la commande "**node appTest.js**" d'exécuter notre application dans la base des données de test.

- Création du fichier **test.js** qui contient les différents tests unitaires relatifs à chaque méthode. Dans ces tests, on a essayé de créer des méthodes qui se déroulent dans les mêmes conditions et sur les mêmes données, c'est à dire que les blocs des tests s'exécutent avec la manière suivante :

1. Vider la base des données (qui est la base de test).
2. Insérer un ensemble des données (qui est le même à chaque fois).
3. Implémenter les méthodes des tests unitaires de ce bloc.

III. Exécution du projet

Pour exécuter notre projet, il suffit de bien suivre ces étapes :

- 1/ Lancer un premier terminal, taper "**mongod**" : qui permet de lancer **MongoDB**.
- 2/ Si vous voulez consulter les différentes bases des données, lancer un deuxième terminal, taper "**mongo**" ensuite "**show dbs**" dans le même terminal.
- 3/ Lancer un troisième terminal, accéder à l'**emplacement** du fichier **app.js** (existant dans le répertoire "**src/app/server**" de notre projet), ceci en tapant la commande **cd CHEMIN**.
- 4/ Dans ce troisième terminal, taper "**npm install**" (pour installer les différents plugins (les dépendances) qui sont mentionnées dans le fichier "**package.json**", il suffit d'installer ces plugins juste pour la première exécution.
- 5/ Dans ce troisième terminal, taper "**npm install -g mocha**" : afin d'installer tous les dépendances qui sont en relation avec le Framework des tests unitaires "**mocha**", il suffit d'installer ces plugins juste pour la première exécution.

6/ Dans ce troisième terminal, taper "**npm install mongoose-auto-increment**" : afin d'avoir des id générés avec le principe d'auto-increment, il suffit d'installer ces plugins juste pour la première exécution.

7/ Dans ce troisième terminal, taper "**node appTest.js**" afin de démarrer la base de données de test intitulé "**MultiTrackDB_Test**", cette dernière est indépendant du déroulement de notre application **principale**.

8/ Dans un quatrième terminal, accéder à l'emplacement du fichier **test.js** (existant dans le répertoire "**src/app/server**" de notre projet), ceci en utilisant la commande **cd CHEMIN** et taper "**mocha test.js**" pour consulter les résultats des divers tests unitaires. Avant chaque bloc de test, on écrase les données qui existent dans la base de données et on insère des données de test qui sont les mêmes à chaque fois afin de garantir que les différents tests peuvent avoir les mêmes conditions d'exécution.

9/ Taper maintenant (CTRL + C) dans le troisième terminal puis saisir la commande "**node app.js**".

10/ Taper maintenant (CTRL + C) dans le quatrième terminal puis saisir la commande "**mocha init.js**" afin d'insérer **le premier utilisateur** qui représente **l'administrateur** de notre application et également la liste des "**tracks**" initiales dans la base de données principale de l'application intitulé **MultiTrackDB**. L'utilisateur inséré est un administrateur donc il peut consulter les interfaces d'administration, son **username** est '**admin**' et son **password** est '**admin**'.

11/ Tester avec les URL mentionnés en bas du fichier "**routeManager.js**" qui existe dans le dossier "**src/app/server/routes**". Taper par exemple cet URL "**http://localhost:8080/users**" qui permet de récupérer la liste des utilisateurs insérées dans la base **MultiTrackDB**.

12/ Visiter l'application sur sa page d'accueil "**http://localhost:8080**".

Si vous rencontrez des problèmes durant l'installation ou pendant l'utilisation de l'application, veuillez contacter :

- GUIZANI Mounir pour la partie serveur : mounir_guizani@hotmail.fr
- DIALLO Mahmoud pour les interfaces d'administration, sign-up et sign-in : mahmoud.diallo@etu.unice.fr
- JOMAA Farouk pour la page principale de l'application (écoute de musique, mixes, commentaires, upload, etc) : jomaa.farouk@gmail.com