



Projet MultiTrack

Matières : Programmable Web Server Side - Client Side

Etabli par :

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

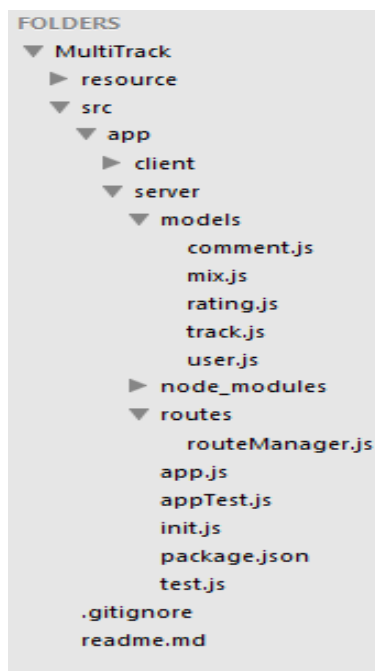
Année universitaire : 2015-2016

I. Introduction

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

II. Structure du projet

Cette figure illustre la structure général de notre dossier d'application :



L'implémentation du 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 premières données nécessaires pour garantir le bon fonctionnement de notre application. À titre d'exemple, ce fichier permet

l'insertion des données relatives premier utilisateur de l'application, qui est l'administrateur dans notre cas.

- Création du fichier **appTest.js** qui permet à travers la commande "**node appTest.js**" d'exécuter notre application dans sa 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 le même environnement, c'est à dire que les blocs des tests s'exécute 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 test unitaires.

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/ Lancer un deuxième terminal, taper "**mongo**" si vous voulez consulter les différentes bases des données, ceci en tapant après "**show dbs**" dans le même terminal.
- 3/ Lancer un troisième terminal, accéder à l'**emplacement** du fichier **app.js** (existant dans le répertoire "**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 tout 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 démarer la base de données de test intitulé "**MultiTrackDB_Test**", cette dernière qui 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 "**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 efface la base de données et on insère les mêmes données afin que les différents tests peuvent avoir les mêmes conditions d'exécution à chaque fois.

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**.

11/ Tester avec les URL existants dans notre projet mentionnés en bas du fichier "**routeManager.js**" qui est existant dans le dossier "**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 "**http://localhost:8080/#**".