# SCALA - Tweet Analyser

Professeur : Nastaran Fatemi

Assistants :
Miguel Santamaria
Maxime Lovito

Auteurs :
Dejvid Muaremi
Mentor Reka
Xavier Vaz Afonso

Département TIC

7 juin 2019

# Table des matières

# Chapitre 1

# Introduction

Tweet Analyser est une application web qui permet d ?analyser les sentiments provenant des tweets sur Twitter. Il permettrait par exemple à une entreprise d ?analyser les réponses du tweet le plus récent et d ?avoir un sentiment général vis-à-vis de celui-ci. Il est également possible de visualiser ces résultats avec l'aide d'un graphe. Réalisé lors du cours à option de SCALA, qui a lieu lors du dernier semèstre pour les étudiants en TIC. La durée de développement du projet fût de 5 semaines avec un délai de remise au 10.06.2019. Dans le cadre de la phase de projet, l'équipe de développement est composée par Dejvid Muaremi, Mentor Reka, Xavier Vaz Afonso.

# Chapitre 2

# Objectifs

Créé le 21 mars 2006 et lancé en juillet de la même année, Twitter est une application permettant à ses utilisateurs de publier gratuitement des messages de 280 caractères, également appelé tweets, sur internet dont la durée de vie est, en moyenne, d'une heure. En 2018, Twitter avait plus de 300 millions d'utilisateurs actif par mois et plus de 500 millions de tweets envoyés par jour.

Actuellement, ce sont les personnalités publique qui envoient le plus de tweets. On retrouve principalement des chefs d'états, des artiste, mais aussi des chefs d'entreprise, des entreprises, et des chaînes de télévision. Twitter n'est désormais plus utilisé uniquement pour envoyé de simples messages mais aussi en tant que plateforme d'information et d'échanges lors de certains événements comme par exemple l'Eurovision qui a fait couler beaucoup de bits. Aussi, il existe déjà de nombreuses entreprises qui proposent un service de support passant par ce médium.

Une entreprise qui publie énormément aura forcément besoin de connaître l'avis général et la réaction de ses abonnés et autres utilisateurs par rapport à leurs tweets. Il lui serait également intéressant de connaître la manière dont sont accueilli leurs nouveaux produits mais aussi les réactions aux tweets des membres de l'entreprises.

# Chapitre 3

# Conception et architecture du projet

## 3.1 Téchnologies et outils utilisés

### 3.1.1 React

La bibliothèque JavaScript libre développée par Facebook et une communauté de développeur indépendant. Le but principal de React est de faciliter la création de composants pour les interfaces utilisateur. Ces composants dépendent d'un état et lorsque celui-ci change, une nouvelle page HTML est générée. En plus d'avoir eu le cours de TWEB lors du premier semèstre de troisième année, certains de nos membres avaient déjà de l'expérience et de la facilité avec cette bibliothèque.

### 3.1.2 Scala

Conçu à l'École polytechnique fédérale de Lausanne (EPFL), Scala est un langage de programmation multi-paradigme, il combine la programmation orientée objet et la programmation fonctionnelle. Son but est d'exprimer les modèles de programmation courants dans une forme concise et élégante.

### 3.1.3 Scala-Play

Basé sur Akka, Play est un framework web open source qui permet de rapidement et facilement créer des applications web basée sur Java. La particularité de Play est de ne pas être basé sur le moteur Java, ce qui en fait un moteur plus simple et plus puissant lors d'une application web. Depuis la version 2.0, le framework a subit une refonte et a été écrit en Scala, le build et le déploiement ont été migré sur du SBT.

### 3.1.4 Slick (database)

Scala Language-Integrated Connection Kit est une librairie de type Functional Relational Mapping (FRM) qui permet de facilement accéder à une base de données avec du Scala. Cette librairie permet de traiter les données comme s'il s'agissait d'une collections Scala tout en laissant aux développeurs, un contrôle sur les accès à la base de données. Les données sont traitées de manière asynchrone ce qui permet de facilement l'intégrer dans un projet Scala-Play. L'un des avantages principal de Slick est qu'il est Typesafe et permet donc un traitement correcte des données selon le modèle.

### 3.1.5 MySQL

Le système de gestion de bases de données relationnelles (SGDBR), faisant partie des logiciels de gestion des bases de données les plus utilisés au monde et que l ?entièreté de notre équipe connaît avec suffisamment de détails pour être facilement utilisé dans un projet de cette envergure. De plus, il est recommandé de l'utiliser dans un projet utilisant Slick.

### 3.1.6 Docker

Le logiciel libre qui automatise le déploiement d ?application dans des conteneurs logiciels. Ces conteneurs sont isolés et peuvent être exécutés sur n ?importe quel système qui prend en charges Docker. Principalement utilisé lors de la phase de développement, il permet de créer un conteneur MySQL qui sera le même pour chaque membre du groupe et ce facilement.

### 3.1.7 Git

Le gestionnaire de version décentralisé libre, que l'on a choisi d'utiliser afin de gérer la totalité du projet ainsi que ses différentes versions.

Nous avons créé utilisés des branches afin d'implémenter les nouvelles fonctionnalités une fois celle-ci prête elles sont envoyée dans une copie temporaire de la branche Master afin de s'assurer du bon fonctionnement du projet. Une fois les modifications acceptées, celle-ci sont envoyée sur la branche Master.

### 3.1.8 GitHub

Le service web permettant de parcourir graphiquement l'historique Git du client et du serveur et qui nous a également permis d'héberger nos sources en ligne. GitHub offre également de nombreux outils de gestion de projet qui sont intéressantes lorsque l'on doit travailler en équipe.

## 3.2 Architecture du client

Nous avons choisi d'utiliser React pour le client, de par nos expériences personnelle avec la librairie. React, l'oblige, le projet client est structuré en composant.

**Un répertoire page** qui contient les composants des pages de **Login**, **Register**, **Home**, **Analyse**, et le **Header**.

**Un répertoire Components** qui contient les **Canvas** utilisant la librairie permettant de créer des graphes, le composant **Graph**, qui contiendra le Canvas. et pour finir le composant **Post** qui lui contient les rapport d'analyse.

**Un répertoire utils** qui contient un composant qui utilise le contexte de React pour partager des informations entre les composant comme nous l'avons appris lors du cours de TWEB. Pour ce qui est des requêtes Axios,elles sont contenues dans le fichier **user.service**.

**Le fichier App.js** le programme principal, s'occupe de rediriger l'utilisateur selon ses autorisations de la page courantes vers les pages désirées tout en faisant le lien parmi les différents composants vu ci-dessus.

## 3.3 Architecture du serveur

Pour ce qui est du serveur, nous avons gardé la structure de base du projet tel qu'il a été fournis par Miguel Santamaria en y intégrant nos fonctionnalités, à savoir, pour l'analyse de sentiments. Plus précisément, nous retrouvons les éléments suivants :

- **controllers** : Comme son nom l'indique, il contient les différents contrôleurs du projet.
- **filters** : Contient les différents filtres qui peuvent être appliqué au données.
- **models** : Tout simplement les modèles de données qui seront stocké dans la base de données que nous traiterons un peu plus loin.
- **repositories** : Permet de faire le lient entre les données et la base de données.
- **services** : Les différents services offerts par notre serveur, dont l'analyse de sentiments.
- **views** : Les vues HTML, offertes par notre serveur.

### 3.3.1 Définition de la base de données

#### Schéma de la base de données

En nous basant sur les données fournies par l'API de Twitter, nous avons construits nos modèles en ciblant la simplicité et la clarté.

lancer le projet dans MySQL workbench et générer des Schéma et les commenter un peu plus.

#### User

Cette table contient la liste des utilisateurs pouvant utiliser notre application. Chaque utilisateur est identifié par un email unique et possède un mot de passe ainsi qu'une date de création.

#### Tweet

Cette table contient la liste des Tweets qu'un utilisateur souhaite analyser, un utilisateur peut analyser un à plusieurs Tweets et chaque Tweets est analysé par un seul et unique utilisateur. Chaque Tweets est défini par un id unique et possède un auteur, le texte du Tweets, une date de création, une date d'analyse, et un ressentiment moyen.

#### TweetResponse

Cette table contient la liste des réponses à un Tweets, leur texte sera utilisé dans l'analyse de sentiment. Un Tweets peut posséder entre une et plusieurs réponses et chaque réponses est lié à un et un seul Tweets. Tout comme les Tweets, les réponses possèdent un id, un auteur, un texte qui sera analysé, une date de création, une date d'analyse, et un ressentiment moyen. De la même manière qu'un Tweet peut intéresser une entreprise, les réponses aux Tweet d'autrui peuvent avoir un intérêt.

# Chapitre 4

# Implémentations détaillées

Compléter selon les indications ci-dessous. Ajouter des bouts de code intéressant. Ajouter un schema d'execution Ajouter plus de détails sur l'implémentation

Indications :

Dans le fichier TwitterClientService, peut être un truc interessant c'est la fonction getTweets qui utilise un accumulateur en recursif comme en a vu en cours. Mais en gros cette fonction récupère les 100 premières réponses, prend la valeur next pour chopper les 100 d'après anisi de suite jusqu'à que le compteur tombre à 0 ou qu'il y ai plus de tweet. On a utilisé une promess qui va attendre d'avoir tous résolu pour passé à la suite, je sais honnetement pas si c'est une bonne pratique mais je voyais d'autre alternatif

# Chapitre 5

# Difficultés rencontrées

Api Twitter gratuit / enterprise
Programmation concurrente
On peut pas garantir de pouvoir lister toutes les réponses d'un tweets, d'ou le fait que j'ai rajouter les différents modes qui permet de parcourir moins ou plus de tweets pour maximier le nombre de réponse souhaité mais ça résout pas completement le problème.

# Chapitre 6

# Problèmes connu dans l'application

Api twitter

Juste pour que vous le sachiez, on peut supprimer une analyse easy depuis postman sans ou avec le mauvaus token, je trouve aucun moyen de contourner ça , un if avant semble pas être pris en compte mais voila ça me soule de chercher plus C'est peut-être la logique qui est pété ou j'oublie de faire un check dieu sait (voilà pk des libs toutes faite c'est mieux hehe) dans le jwtUtility si jamais j'ai utilisé cette lib

# Chapitre 7

# Possibles améliorations

Api Twitter toujours

La concurrence, lire la doc avant de dire nimp

Meilleur système d'authentification -> voir les bonnes pratiques, bouncy castle fait le café en java, un équivalent scala ou inclusion de lib java ?

Optimisations, c'est toujours possibles. L'outil d'apache dont j'ai oublié le nom ?

Est-ce qu'on viole la vie privée des employé ?

# Chapitre 8

# conclusion

## 8.1 Projet

Ce projet, l'un des derniers de notre vie de bachelier fût très intéressant. Il nous a non seulement permis de mettre en pratique tout ce que l'on a appris lors du cours à option de Scala,mais également de l'expérience acquise lors des nombreux projets réalisé à l'HEIG-VD.

La possibilité de choisir un projet qui nous plaît a également été une motivation supplémentaire pour notre groupe.

Une gestion rigoureuse du projet sous git ainsi que la liberté de choisir la partie qui nous intéresse et que l'on maîtrise au mieux à permis à chaque membre du groupe de pleinement s'épanouir dans la tâche qui lui a été confiée.

## 8.2 Groupe

Chaque membre était pleinement impliqué dans ce projet. Nous avons su faire preuve d'une réelle volonté, d'un intérêt certain, et d'une remarquable résistance au stress lors de cette période trouble qu'est la fin du semestre.

La régularité et la justesse a été la clé dans le succès de ce travail. La communication entre les différents membres du groupe, les décision prise par chacun d'entre nous et le bon sens a permis de résoudre de nombreux problèmes dans les plus bréf délai ainsi que le succès de notre projet.

## 8.3 Avis des membres

**Dejvid Muaremi** Ce projet fut très intéressant à réaliser. J ?ai pu découvrir certaines technologies pour lesquelles je n'aurai pas forcément investi le temps nécessaire à leur compréhension par moi-même et j'ai pu mettre en pratique tout ce que j ?ai appris jusqu ?à maintenant. Il arrivait parfois que je prenne du retard sur mes tâches, mais, au final, j ?ai pu le rattraper.

**Mentor Reka**

**Xavier Vaz Afonso**