

# RAPPORT DU PROJET INTÉGRÉ

## SEMESTRE 5

**Application Microservice pour le  
scrapping et l'analyse des sentiments  
des tweets**

Réalisé par Groupe :  
BETTAOUI Saif Elislam  
LAAMOUMRI Souhail  
MALKI Otman

Encadré par:  
Prof. BAH Slimane

# TABLE DE MATIERE

<b>I. Introduction.....</b>	<b>4</b>
A. Contexte du projet.....	4
1. Motivation :.....	5
2. But du projet:.....	5
3. Enjeux et Opportunités:.....	5
B. Objectifs du projet.....	6
C. Importance des microservices dans le développement.....	7
<b>II. Architecture du Projet.....</b>	<b>9</b>
A. Présentation générale de l'architecture.....	9
B. Description des microservice.....	10
1. Microservice 1 : Tweet Scraper.....	10
2. Microservice 2 : Modèle ML pour l'analyse de sentiment.....	10
3. Microservice 3 : Frontend.....	10
4. Microservice 4 : Backend.....	11
C. Communication entre les microservices.....	11
Utilisation des API dans la Communication entre Microservices :.....	11
Open Feign dans le Backend pour Simplifier la Communication :.....	12
<b>III. Microservice 1 : Tweet Scraper.....</b>	<b>12</b>
A. Objectifs du microservice.....	12
B. Technologies utilisées.....	12
1. Flask:.....	13
2. Selenium.....	13
3. Beautiful Soup 4.....	13
C. Fonctionnalités principales :.....	14
D. Intégration avec les autres microservices.....	14
<b>IV. Microservice 2 : Modèle ML pour l'analyse de sentiment.....</b>	<b>15</b>
B. Objectif du microservice.....	15
B. Technologies utilisées.....	16
1. Flask :.....	16
2. Jupyter Notebook :.....	16
3. Bibliothèques Python :.....	16
4. Json :.....	16
C. Fonctionnement.....	16
<b>V. Microservice 3 : Frontend.....</b>	<b>17</b>
A. Objectif du microservice.....	17
B. Technologies Utilisées.....	17
1. React.js :.....	17
2. Axios :.....	18
C. Interface Utilisateur.....	18
D. Interaction avec le Backend.....	18

<b>VII. Microservice 4 : Backend.....</b>	<b>19</b>
A. Objectif du Microservice.....	19
B. Technologies Utilisées.....	19
1. Spring Boot :.....	19
2. H2 Database :.....	19
3. Open Feign :.....	20
4. Spring Web :.....	20
5. Spring Data JPA :.....	20
C. Logique Metier.....	21
D. Interaction avec la base de données.....	21
<b>VIII. Résultats et Performances.....</b>	<b>22</b>
A. Page d'accueil.....	22
B. Interface d'envoi de requêtes.....	22
C. Affichage de résultats.....	23
D. Exemple de filtrage.....	23
<b>VIII. Conclusion.....</b>	<b>26</b>
A. Récapitulation des Points Clés du Projet :.....	26
B. Réussites et Leçons Apprises :.....	26
C. Perspectives d'Avenir et Améliorations Possibles :.....	27

# I. Introduction

## A. Contexte du projet

Le projet que nous avons initié a pour objectif le développement d'une application microservices complète dédiée à l'analyse des sentiments exprimés sur Twitter. Dans le contexte actuel de la prolifération des médias sociaux, la compréhension des émotions des utilisateurs revêt une importance cruciale, impactant divers domaines tels que la veille médiatique, l'analyse des tendances, et la compréhension de l'opinion publique.

### 1. Motivation :

La motivation sous-jacente à ce projet découle de l'explosion des données générées sur les réseaux sociaux, notamment sur Twitter, offrant ainsi une opportunité inestimable pour extraire des informations significatives. L'analyse de sentiment, au cœur de notre démarche, permet de discerner les tendances émotionnelles, apportant ainsi une perspective précieuse pour les entreprises, les chercheurs, et les décideurs.

### 2. But du projet:

Les objectifs du projet sont multiples. Nous cherchons à mettre en place un "tweet scraper" permettant la collecte en temps réel de données provenant de Twitter. Nous envisageons également d'implémenter un modèle de machine learning spécifiquement conçu pour l'analyse de sentiment des tweets. Parallèlement, nous concevrons un frontend interactif destiné aux utilisateurs finaux, et érigerons un backend robuste pour la gestion des données, la communication entre les microservices, et la prise de décision.

### 3. Enjeux et Opportunités:

Quant au public cible de cette application, il englobe une variété de profils tels que les analystes de données, les chercheurs, les professionnels du marketing, et toute personne cherchant à comprendre les tendances et les opinions exprimées sur Twitter.

Les enjeux associés à ce projet incluent le traitement en temps réel de données massives, la nécessité d'une précision accrue dans l'analyse de sentiment, et la création d'une interaction fluide avec l'interface utilisateur.

Toutefois, ces défis s'accompagnent d'opportunités significatives telles que l'exploration de technologies émergentes, l'amélioration de la compréhension des émotions sur les réseaux sociaux, et la fourniture d'une solution pratique aux parties intéressées.

Notre contribution à l'état de l'art réside dans l'intégration de microservices spécialisés, formant ainsi une solution novatrice pour l'analyse des sentiments. Cette application offre une approche distinctive en combinant un "scraper", un modèle ML, et une interface utilisateur conviviale.

En somme, notre projet aspire à répondre à la demande croissante d'analyses de sentiments sur les réseaux sociaux, en fournissant une application à la fois robuste et évolutive.

## B. Objectifs du projet

Les objectifs de notre projet sont divers et s'inscrivent dans une approche holistique visant à développer une application microservices complète dédiée à l'analyse des sentiments sur Twitter. Chacun de ces objectifs contribue à l'accomplissement global de notre vision, mettant en lumière la pertinence et l'utilité de chaque composant du projet.

Tout d'abord, nous visons à développer un "tweet scraper" capable de collecter des données en temps réel à partir de Twitter. Ce microservice sera conçu pour extraire efficacement les tweets pertinents. La mise en œuvre de cette composante est cruciale, car elle constitue la première étape dans la chaîne d'approvisionnement de données, fournissant des informations fraîches et pertinentes pour l'analyse ultérieure.

Un autre objectif central de notre projet est l'implémentation d'un modèle de machine learning spécifiquement conçu pour l'analyse de sentiment des tweets. Ce microservice joue un rôle essentiel dans la transformation des données brutes collectées en informations significatives. Le modèle doit être formé de manière à comprendre les nuances du langage utilisé sur Twitter, tenant compte des expressions informelles, des hashtags, et des émojis. La précision de ce modèle impacte directement la qualité des résultats fournis par l'application, et donc sa valeur.

Parallèlement, nous nous attelons à concevoir un frontend interactif pour l'utilisateur final. Ce composant est l'interface visible de notre application, et il doit être conçu de manière ergonomique et conviviale. L'objectif est de fournir une expérience utilisateur intuitive tout en permettant aux utilisateurs d'interagir efficacement avec les résultats de l'analyse de sentiment. Cela inclut la mise en place de fonctionnalités telles que la recherche, la visualisation des résultats, et la personnalisation de l'interface pour répondre aux besoins spécifiques des utilisateurs.

Enfin, un backend robuste est prévu pour gérer l'ensemble du système. Ce backend est responsable de la gestion des données, de la communication efficace entre les différents microservices, et de la prise de décision basée sur les résultats de l'analyse de sentiment. Il devra être capable de traiter les données massives générées par le "tweet scraper", de coordonner les requêtes provenant du frontend, et de garantir la cohérence globale du système.

L'ensemble de ces objectifs converge vers la création d'une application complète, répondant aux besoins croissants d'analyse de sentiments sur les réseaux sociaux. Ils soulignent la nécessité d'une approche intégrée, où chaque microservice joue un rôle spécifique mais essentiel dans la fourniture d'une solution efficace et performante.

### C. Importance des microservices dans le développement

L'importance des microservices dans le développement d'applications modernes réside dans leur capacité à répondre aux défis croissants et aux exigences complexes auxquels font face les développeurs et les entreprises dans un environnement technologique en constante évolution. Les microservices, en tant qu'architecture logicielle, offrent des avantages significatifs qui améliorent l'agilité, la scalabilité, la maintenabilité et la flexibilité des applications modernes.

Tout d'abord, les microservices permettent une agilité accrue dans le processus de développement. En adoptant une approche modulaire, les équipes de développement peuvent travailler de manière indépendante sur des microservices spécifiques sans affecter le reste de l'application. Cela

favorise le déploiement continu, accélère les cycles de développement et facilite la gestion des mises à jour, ce qui est essentiel dans un paysage technologique en rapide mutation.

La scalabilité constitue un autre aspect crucial. Les microservices permettent de faire évoluer les différentes parties d'une application de manière indépendante en fonction des besoins. Cette approche modulaire offre une flexibilité permettant d'ajuster les ressources allouées à chaque microservice en fonction de la charge de travail spécifique, garantissant ainsi une utilisation efficace des ressources et une réactivité optimale de l'application.

La maintenabilité des applications est grandement simplifiée grâce aux microservices. Chaque microservice peut être développé, testé, déployé et mis à jour indépendamment des autres. Cela signifie que les équipes de développement peuvent se concentrer sur des domaines spécifiques de l'application sans être entravées par des contraintes liées à d'autres parties du système. En outre, cela facilite le débogage, la maintenance et la résolution des problèmes, accélérant ainsi le processus de développement global.

La flexibilité offerte par les microservices permet aux entreprises de choisir les technologies les mieux adaptées à chaque composant de l'application. Chaque microservice peut être développé dans le langage de programmation ou le framework le plus approprié pour sa tâche spécifique. Cela permet une diversité technologique au sein d'une application, favorisant l'innovation et la possibilité d'adopter de nouvelles technologies sans avoir à réécrire l'ensemble de l'application.

En outre, les microservices facilitent la résilience de l'application. Si l'un des microservices échoue, cela n'affecte pas nécessairement l'ensemble de l'application. La conception distribuée des microservices permet de mettre en place des mécanismes de secours, de gestion des erreurs et de reprise après sinistre de manière plus ciblée, garantissant une meilleure disponibilité globale du système.

En résumé, les microservices sont devenus une approche fondamentale dans le développement d'applications modernes en raison de leur capacité à relever les défis de l'agilité, de la scalabilité, de la maintenabilité, de la flexibilité et de la résilience. Cette architecture offre une réponse efficace aux exigences

changeantes des entreprises dans un paysage technologique en constante évolution, permettant aux applications de rester compétitives, réactives et évolutives.



## *II. Architecture du Projet*

### **A. Présentation générale de l'architecture**

Le projet repose sur une architecture de microservices, une approche moderne de développement logiciel qui favorise la décomposition d'une application en services autonomes, indépendants et spécialisés. Chaque microservice est conçu pour accomplir une tâche spécifique, favorisant ainsi la modularité, la scalabilité et la facilité de maintenance.

L'architecture du projet se compose de quatre principaux microservices, chacun jouant un rôle crucial dans la réalisation des objectifs définis. Ces microservices sont interconnectés pour assurer une communication fluide et efficace entre les différentes parties de l'application.

### **B. Description des microservice**

#### **1. Microservice 1 : Tweet Scraper**

- **Objectif du microservice** : Le microservice de scrapping de tweets vise à collecter des données à partir de la plateforme Twitter. Son rôle est de récupérer des tweets en fonction de certains critères définis, fournissant ainsi une source de données pour l'analyse ultérieure.

- **Technologies utilisés** :

\* Flask : Utilisé pour le développement du backend du microservice.

\* Selenium : Employé pour l'automatisation des interactions avec le navigateur Twitter.

\* BeautifulSoup 4 : Utilisé pour l'analyse et l'extraction des données HTML récupérées.

#### **2. Microservice 2 : Modèle ML pour l'analyse de sentiment**

- **Objectif du microservice** : Ce microservice est dédié à l'analyse de sentiment des tweets à l'aide d'un modèle de Machine Learning (ML). Son objectif est d'évaluer le sentiment associé à chaque tweet, classifiant ceux-ci comme positifs ou négatifs. Les résultats de cette analyse de sentiment seront ensuite utilisés pour fournir des insights significatifs aux utilisateurs de l'application.

- **Technologies utilisés** :

- \* Bibliothèques Python : Le microservice utilise différentes bibliothèques Python telles que scikit-learn, pandas, et nltk pour le prétraitement des données et l'entraînement du modèle de Régression Logistique (LR) utilisé dans l'analyse de sentiment.

- \* Flask : Utilisé pour développer le backend du microservice, facilitant ainsi l'exposition d'API pour la communication avec les autres parties de l'application.

### 3. Microservice 3 : Frontend

- **Objectif du microservice** : Le frontend est la partie visible de l'application destinée aux utilisateurs. Son rôle est de présenter les résultats de l'analyse de sentiment de manière conviviale.

- **Technologies utilisés** :

- \* React.js : Utilisé pour la création d'interfaces utilisateur réactives.

- \* Axios : Employé pour les requêtes HTTP vers le backend.

- **Interface utilisateur** : Développée de manière intuitive pour permettre aux utilisateurs d'interagir avec les résultats de l'analyse.

### 4. Microservice 4 : Backend

- **Objectif du microservice** : Le backend agit comme le cœur de l'application, gérant la logique métier, l'interaction avec la base de données, et facilitant la communication entre les différents microservices.

- **Technologies utilisés** :

- \* Spring Boot : Utilisé pour le développement rapide d'applications basées sur Java.

- \* H2 Database : Base de données relationnelle légère utilisée pour le stockage des données.

- \* Open Feign : Employé pour simplifier la communication entre les microservices.

- \* Spring Web et Spring Data JPA : Utilisés pour la gestion des requêtes HTTP et l'accès aux données de la base de données.

## C. Communication entre les microservices

### Utilisation des API dans la Communication entre Microservices :

La communication entre les microservices repose sur l'utilisation d'API, des interfaces standardisées qui définissent les points d'entrée pour chaque service. Chaque microservice expose des API spécifiques, détaillant les méthodes, les paramètres et les formats de données acceptés et retournés. Cette approche standardisée facilite l'intégration des différents services et permet une interaction transparente entre eux. Les API servent de ponts pour l'échange d'informations, contribuant ainsi à la modularité et à l'indépendance des microservices.

### Open Feign dans le Backend pour Simplifier la Communication :

Dans le backend du projet, Open Feign, une bibliothèque Java, est employé pour simplifier la communication entre microservices. Il permet la déclaration d'interfaces d'API décrivant les interactions avec d'autres services. Ces interfaces sont ensuite automatiquement implémentées par Open Feign, simplifiant ainsi la gestion des requêtes HTTP. L'utilisation de l'interface déclarée dans le code facilite les appels aux microservices externes, réduisant la complexité et la duplication du code. Open Feign offre également des fonctionnalités intégrées telles que la gestion des erreurs et la définition des délais, contribuant à une communication plus robuste entre les microservices.

### III. Microservice 1 : Tweet Scraper

#### A. Objectifs du microservice

Le microservice "Tweet Scraper" est spécifiquement conçu pour atteindre des objectifs clés dans l'extrapolation efficace de données à partir de Twitter, en utilisant des technologies telles que Flask, Selenium et BeautifulSoup 4. Ces outils sont soigneusement sélectionnés pour optimiser la collecte des tweets, assurer la fiabilité du processus, et maintenir la souplesse nécessaire pour s'adapter aux différentes évolutions

#### B. Technologies utilisées

Le microservice "Tweet Scraper" repose sur un ensemble de technologies soigneusement sélectionnées pour assurer une extraction efficace, fiable et en temps réel des données provenant de Twitter. Les outils choisis pour ce microservice comprennent Flask, Selenium et BeautifulSoup 4, chacun apportant des fonctionnalités spécifiques

##### 1. Flask:

Flask, un framework web en Python, est au cœur du microservice pour plusieurs raisons. Sa légèreté, sa simplicité d'utilisation et son extensibilité en font un choix idéal pour créer une API robuste. L'utilisation de Flask permet de créer des points d'entrée clairs et d'établir des routes pour gérer les requêtes provenant d'autres composants de l'application. Il offre également la possibilité d'étendre facilement les fonctionnalités de l'API, facilitant ainsi l'intégration avec d'autres microservices.



##### 2. Selenium

Sélénium, quant à lui, est un outil essentiel pour l'automatisation du navigateur. Il simule le comportement d'un utilisateur réel en interagissant dynamiquement avec les pages web. Cette approche est particulièrement utile pour contourner les limitations de l'API Twitter officielle. Selenium permet d'automatiser la navigation sur le site Twitter, d'effectuer des



recherches en temps réel et d'extraire les tweets pertinents de manière rapide et fiable.

### 3. Beautiful Soup 4

Beautiful Soup 4, une bibliothèque de parsing HTML en Python, complète le processus en facilitant l'analyse des pages

web extraites par Selenium. Elle offre une interface conviviale pour extraire des informations spécifiques du HTML, simplifiant ainsi le processus d'extraction des données. Beautiful Soup 4 joue un rôle crucial dans la gestion des données extraites, permettant au microservice de cibler les éléments nécessaires tels que le texte des tweets, les métadonnées, et d'autres informations pertinentes.



#### C. Fonctionnalités principales :

Les fonctionnalités principales du microservice "Tweet Scraper" sont spécifiquement conçues pour permettre une extraction efficace et ciblée des données provenant de Twitter. Ces fonctionnalités sont cruciales pour garantir la pertinence, la fraîcheur et la fiabilité des informations collectées en temps réel. Voici un aperçu détaillé des fonctionnalités clés :

1. **Extraction de Tweets Pertinents** : La fonction principale du "Tweet Scraper" est d'extraire de manière ciblée les tweets pertinents en fonction de critères spécifiques tels que des hashtags, des mots-clés ou des comptes d'utilisateurs. Cette fonctionnalité garantit que seuls les tweets ayant une importance pour l'analyse de sentiments sont collectés.
2. **Automatisation de la Navigation** : Grâce à l'utilisation de Selenium, le "Tweet Scraper" automatise la navigation sur le site Twitter de manière dynamique. Cette automatisation permet d'interagir avec la plateforme comme un utilisateur réel, surmontant ainsi certaines limitations imposées par l'API officielle.

## D. Intégration avec les autres microservices

L'intégration du microservice "Tweet Scraper" avec les autres microservices est cruciale pour le bon fonctionnement et la cohérence de l'ensemble du système. Dans ce scénario, où les microservices communiquent via des requêtes HTTP, l'intégration est facilitée par la définition claire des points d'entrée API et l'utilisation de mécanismes de requêtes.

1. **Communication avec le Microservice d'Analyse des Sentiments :** Une fois que le "Tweet Scraper" a extrait les tweets pertinents, il communique avec le microservice d'analyse des sentiments en effectuant des requêtes HTTP. Ces requêtes incluent les données des tweets collectés, et le microservice d'analyse des sentiments les utilise comme entrée pour évaluer les émotions associées à chaque tweet.
2. **Transmission aux Autres Microservices :** Après l'analyse des sentiments, le microservice d'analyse transmet les résultats aux autres microservices via des requêtes HTTP. Ces résultats peuvent inclure des données telles que le sentiment global sur une période donnée, les tendances émotionnelles, ou d'autres métriques pertinentes. Chaque microservice destinataire, qu'il s'agisse du frontend, du backend, ou d'autres composants, utilise ces données pour ses propres fonctionnalités.
3. **Utilisation de Méthodes HTTP Appropriées :** L'intégration est facilitée par l'utilisation de méthodes HTTP appropriées pour les requêtes. Par exemple, l'analyse des sentiments peut répondre avec les résultats à l'aide de la méthode GET ou POST, selon les besoins.
4. **Définition Claire des Points d'Entrée :** Une définition claire des points d'entrée API pour chaque microservice est essentielle. Cela garantit une compréhension mutuelle des données échangées et des méthodes d'accès aux fonctionnalités de chaque service.

## *IV. Microservice 2 : Modèle ML pour l'analyse de sentiment*

### **B. Objectif du microservice**

Le Microservice 2, dédié à l'analyse de sentiment des tweets via un modèle de Machine Learning (ML), a pour objectifs principaux d'appliquer un modèle de Régression Logistique préalablement entraîné pour classifier les tweets en positifs ou négatifs. Il réalise un pré traitement avancé du texte, assurant la qualité des données, et maintient une adaptabilité du modèle aux évolutions linguistiques. L'intégration transparente avec d'autres composants de l'architecture, notamment le frontend et le backend, vise à présenter des insights significatifs aux utilisateurs. En mettant l'accent sur la gestion des erreurs, le microservice assure la robustesse et la fiabilité de l'analyse de sentiment dans divers scénarios. En somme, le Microservice 2 joue un rôle central en fournissant des évaluations automatisées et exploitables du ton émotionnel des tweets au sein de l'application.

### **B. Technologies utilisées**

#### **1. Flask :**

Framework web utilisé pour créer le serveur web et exposer l'API permettant de recevoir les termes de recherche et de fournir les résultats d'analyse de sentiment.

#### **2. Jupyter Notebook :**

Utilisé pour la construction, l'entraînement et l'évaluation du modèle d'analyse de sentiment. Le notebook (model.ipynb) implique le prétraitement des données textuelles, la construction du modèle (probablement une régression logistique), et l'évaluation de sa performance.

#### **3. Bibliothèques Python :**

Diverses bibliothèques Python sont utilisées dans le Jupyter Notebook et le script Flask, notamment :

- **Pandas** : Pour la manipulation et l'analyse des données.
- **NumPy** : Pour le support des structures de données et les opérations mathématiques.
- **Scikit-learn** : Utilisé probablement pour la mise en œuvre du modèle de régression logistique et d'autres utilitaires d'apprentissage automatique.

- **NLTK (Natural Language Toolkit)** : Employé pour le traitement du langage naturel, y compris la lemmatisation.

#### 4. Json :

Format de données utilisé pour représenter et transmettre les résultats d'analyse de sentiment de l'API.

### C. Fonctionnement

Le microservice décrit dans le script et le Jupyter Notebook est un outil d'analyse de sentiment conçu pour traiter les données Twitter. Ce service utilise Flask pour créer un serveur web avec un point d'accès qui reçoit un terme de recherche, récupère les tweets pertinents d'une URL spécifiée, puis traite ces tweets pour une analyse de sentiment.

Le Jupyter Notebook (model.ipynb) est utilisé pour construire et entraîner le modèle d'analyse de sentiment. Il implique le prétraitement des données textuelles, qui comprend le nettoyage, la normalisation et la vectorisation des tweets. Le modèle, probablement un classificateur de régression logistique, est entraîné sur ces données prétraitées pour classer les tweets en sentiments comme positif ou négatif.

L'application Flask charge le modèle pré-entraîné et le vectoriseur. Elle expose ensuite un point d'API qui, lorsqu'il est invoqué avec un terme de recherche, récupère les tweets, les prétraite en utilisant la même méthode que la phase d'entraînement, et utilise le modèle pour prédire leurs sentiments. Les résultats, qui incluent les données originales des tweets ainsi que les sentiments prédits, sont ensuite retournés au format JSON.

En résumé, ce microservice sert d'intermédiaire entre une source de données Twitter et l'utilisateur final, ajoutant des informations d'analyse de sentiment aux tweets.



## V. Microservice 3 : Frontend

### A. Objectif du microservice

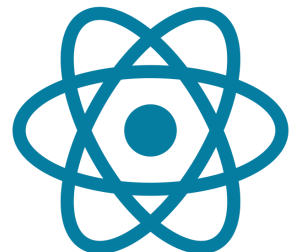
Le microservice Frontend a pour objectif de fournir une interface utilisateur conviviale permettant aux utilisateurs d'interagir avec le système de manière intuitive. Ses principales missions incluent :

- Offrir une expérience utilisateur fluide et agréable lors de l'utilisation de l'application.
- Permettre à l'utilisateur d'initier le processus de scrapping en envoyant des requêtes au microservice de scrapping.
- Afficher les résultats du scraping de manière claire et compréhensible à travers une interface graphique.

### B. Technologies Utilisées

#### 1. React.js :

React.js est une bibliothèque JavaScript développée par Facebook, principalement utilisée pour construire des interfaces utilisateur interactives et réactives. Elle permet de créer des composants réutilisables qui rendent le développement d'applications web plus efficace.



#### 2. Axios :

Axios est une bibliothèque JavaScript populaire pour effectuer des requêtes HTTP depuis le navigateur ou Node.js. Elle simplifie les requêtes AJAX et les manipulations de données.

### C. Interface Utilisateur

L'interface utilisateur du microservice Frontend est conçue de manière à être ergonomique et en plus esthétiquement agréable. Elle comprend les éléments suivants :

- Formulaire d'envoi de requête : Permet à l'utilisateur d'initier le processus de scrapping en fournissant le mot clé de recherche avec laquelle une requête au microservice scraper sera envoyé.
- Tableau de résultats : Affiche les tweets scrappés avec leurs analyses de sentiments (positif/négatif) tout en fournissant une multitude de filtres et barres de recherche pour faciliter le parcours des données et le tri.

#### D. Interaction avec le Backend

Le microservice Frontend interagit étroitement avec le backend. Les points clés de cette interaction sont les suivants :

- Envoi de requêtes de scrapping : L'utilisateur utilise l'interface pour lancer le processus de scrapping en envoyant des requêtes au microservice de scrapping via les endpoints fournis par le backend.
- Réception des résultats : Le Frontend reçoit les résultats du scrapping et de l'analyse de sentiments du backend, les présentant ensuite de manière claire à l'utilisateur.

## VII. Microservice 4 : Backend

#### A. Objectif du Microservice

Le microservice Backend a pour objectif de gérer le flux de données entre le Frontend, le microservice de scrapping, et le microservice d'analyse des sentiments. Ses principales missions incluent :

- Recevoir les résultats du microservice de scrapping et du microservice d'analyse des sentiments.
- Effectuer des traitements et vérifications sur les données reçues avant de les enregistrer dans la base de données.

- Exposer des endpoint permettant au Frontend de récupérer les données scrappées et analysées.

## B. Technologies Utilisées

Le microservice Backend repose sur plusieurs technologies afin d'atteindre ses objectifs :

### 1. Spring Boot :

Utilisé comme framework principal pour le développement du backend c'est un framework de développement d'applications Java basé sur le principe de l'inversion de contrôle (IoC) et la programmation orientée aspect (AOP). Il fournit une infrastructure complète pour le développement d'applications Java d'entreprise, couvrant des aspects tels que la gestion des transactions, la gestion des données, la sécurité, et bien plus encore..



### 2. H2 Database :

H2 Database est un système de gestion de base de données relationnelles (SGBDR) écrit en Java. Il s'agit d'une base de données open-source, légère, rapide et intégrée qui peut être utilisée dans des applications Java. Il est employé comme base de données embarquée pour stocker les résultats du scrapping et de l'analyse des sentiments pendant le développement et les tests.



### 3. Open Feign :

OpenFeign est une bibliothèque déclarative basée sur Java, développée par Netflix, qui simplifie la création de clients HTTP pour interagir avec des services REST. Elle fait partie de l'écosystème Spring Cloud et est souvent utilisée dans les applications microservices pour faciliter la communication entre différents services. Il est intégré pour la communication avec le microservice d'analyse des sentiments.



#### 4. Spring Web :

Utilisé pour créer des points d'extrémité (endpoints) pour la communication avec le Frontend. Il fait référence à la partie du framework Spring dédiée au développement d'applications web. Cette partie inclut plusieurs modules et composants qui simplifient la création d'applications web robustes et évolutives.



#### 5. Spring Data JPA :

Employé pour faciliter l'interaction avec la base de données. Spring Data JPA est un projet au sein de l'écosystème Spring qui simplifie le développement d'accès aux données dans les applications Java en utilisant la technologie JPA (Java Persistence API). JPA est une API de persistance Java standard qui permet de gérer des données relationnelles dans des applications Java. Spring Data JPA s'appuie sur JPA et fournit des fonctionnalités supplémentaires pour rendre l'accès aux données plus facile et plus efficace.



### C. Logique Metier

La logique métier du microservice Backend comprend plusieurs étapes importantes :

- Réception des résultats : Les résultats du scrapping et de l'analyse des sentiments sont reçus en utilisant des points d'extrémité dédiés.
- Traitements et vérifications : Les données reçues sont soumises à des traitements métier spécifiques, tels que la validation des formats, la vérification de l'intégrité des données, etc.
- Enregistrement en base de données : Les données traitées sont ensuite enregistrées dans la base de données H2.

- Exposition d'endpoint : Un endpoint est exposé pour permettre au Frontend de récupérer les résultats enregistrés dans la base de données.

#### D. Interaction avec la base de données

L'interaction avec la base de données H2 se fait de manière transparente grâce à l'utilisation de Spring Data JPA. Les opérations suivantes sont effectuées :

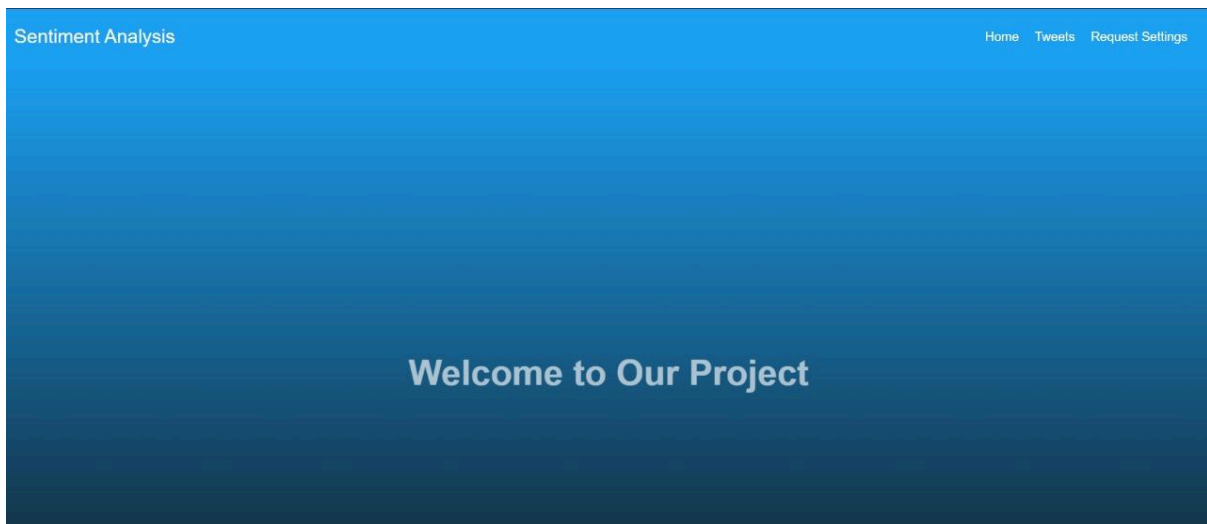
- Création des entités : Des entités JPA sont créées pour représenter les données à stocker dans la base de données.
- Opérations CRUD : Les opérations de création, lecture, mise à jour et suppression sont effectuées sur les entités pour gérer les résultats du scrapping et de l'analyse des sentiments.

## *VIII. Résultats et Performances*

Voici un exemple d'utilisation de notre application :

#### A. Page d'accueil

La première page que l'utilisateur voit dans notre site web est cette interface d'accueil :



## B. Interface d'envoi de requêtes

Dans cette interface, dans l'onglet Request Settings, l'utilisateur peut saisir le mot clé de la recherche qui souhaite faire, afin de lancer le service de scrapping en recherchant le mot clé sur twitter.

A titre d'exemple voici un exemple de recherche qui va nous permettre d'avoir des tweets avec des émotions mixés.

L'utilisateur saisie le mot de recherche et ensuite clique sur le bouton pour lancer le fonctionnement de notre application :

The image displays the "Request Settings" tab of the application. In the center, there is a white box with a blue border titled "Twitter Search Form". Inside this box, the text "Search Query:" is followed by a text input field containing the word "Palestine". Below the input field is a blue button with the text "Search on Twitter" in white. The background of the page is a light blue gradient, and the header is the same blue bar as in the previous screenshot.

## C. Affichage de résultats

Après que le service ait fini son travail de scrapping et d'analyse de sentiment le resultat est enregistré automatiquement dans la base de données et est affiché dans notre interface frontend de la manière suivante :

Tweets Sauvegardés				
ID	Time Stamp	Username	Content	Sentiment
1	2024-01-18 11:54:24	ISKANDER.@ISKANDER_TM_ :59m	Crazy, it took me literally three minutes to google about Israel and Palestin to understand that Israel colonised Palestine and now commuting genocide on Palestinians, but nooooooo, it's "complicated" and they all wrong and we just want peaceeee	Negative
2	2024-01-18 09:10:12	حسن نصر الله@SH_NasrallahEng :3h	Kill us wherever you may find us And on every front And on the door of every Hussainiya and Mosque (But know this) We are the followers of Ali (as) Ibne Abi-Talib (as) and WE WILL NOT ABANDON PALESTINE.	Positive
3	2024-01-18 00:07:25	Jackson Hinkle @jacksonhinkle :12h	ISRAEL just blew up the University of Palestine in GAZA — even though it has LITERALLY NOTHING to do with Hamas.	Negative
4	2024-01-18 06:21:50	Robert Martin @Robert_Martin72 :6h	Israel's defending itself??? Israeli forces have destroyed 140 mosques and 3 churches in Palestine's Gaza since October 7, according to a statement by the Palestinian government media office in the enclave. 240 other mosques have also been partially damaged as a result of...	Negative
5	2024-01-18 10:00:02	Khalissee@Khalissee :2h	In 2001 a hidden camera recorded Netanyahu speak about his genocidal plans for Palestine and how easy it is to influence America. 'We must hit the Palestinians hard...take them to a state of panic and fear...and the world will not say anything.'	Positive
6	2024-01-18 00:07:25	Jackson Hinkle @jacksonhinkle :12h	ISRAEL just blew up the University of Palestine in GAZA — even though it has LITERALLY NOTHING to do with Hamas.	Negative
7	2024-01-18 06:21:50	Robert Martin @Robert_Martin72 :6h	Israel's defending itself??? Israeli forces have destroyed 140 mosques and 3 churches in Palestine's Gaza since October 7, according to a statement by the Palestinian government media office in the enclave. 240 other mosques have also been partially damaged as a result of...	Negative

On peut visualiser les différents champs comme date, nom d'utilisateur, contenu de tweet, etc.

D. Exemple de filtrage

Notre interface offre plusieurs moyen de filtrage et de recherche qui vont servir à l'utilisateur pour trier et filtrer les données selon son besoin voici les différents filtres offerts par notre application :

Sentiment Analysis

HomeTweetsRequest Settings

Search by Username:

Enter username

Search by Content:

Enter content

Filter by Sentiment:

All Sentiments

Filter by Date Range:

Start Date:

dd / mm / yyyy

End Date:

dd / mm / yyyy

Refresh Page

Clear Database

Tweets Sauvegardés

Par exemple pour notre cas on va utiliser le filtre de sentiment, d'abord pour visualiser les sentiments négatifs seulement :

Filter by Sentiment:

Negative

Filter by Date Range:

Start Date:

dd/mm/yyyy

End Date:

dd/mm/yyyy

Refresh Page

Clear Database

Tweets Sauvegardés

ID	Time Stamp	Username	Content	Sentiment
1	2024-01-18 11:54:24	ISKANDER.@ISKANDER_TM_-59m	Crazy, it took me literally three minutes to google about Israel and Palestin to understand that Israel colonised Palestine and now commuting genocide on Palestinians, but noooooo, it's "complicated" and they all wrong and we just want peaceeee	Negative
3	2024-01-18 00:07:25	Jackson Hinkle @jacksonhinkle-12h	ISRAEL just blew up the University of Palestine in GAZA — even though it has LITERALLY NOTHING to do with Hamas.	Negative
4	2024-01-18 06:21:50	Robert Martin @Robert_Martin72-6h	Israel's defending itself??? Israeli forces have destroyed 140 mosques and 3 churches in Palestine's Gaza since October 7, according to a statement by the Palestinian government media office in the enclave. 240 other mosques have also been partially damaged as a result of...	Negative
6	2024-01-18 00:07:25	Jackson Hinkle @jacksonhinkle-12h	ISRAEL just blew up the University of Palestine in GAZA — even though it has LITERALLY NOTHING to do with Hamas.	Negative

Exemple pour les sentiments positifs :

Filter by Sentiment:

Positive

Filter by Date Range:

Start Date:

dd/mm/yyyy

End Date:

dd/mm/yyyy

Refresh Page

Clear Database

Tweets Sauvegardés

ID	Time Stamp	Username	Content	Sentiment
2	2024-01-18 09:10:12	حسن نصرالله @SH_NasrallahEng-3h	Kill us wherever you may find us And on every front And on the door of every Hussainiya and Mosque (But know this) We are the followers of Ali (as) Ibné Abi-Talib (as) and WE WILL NOT ABANDON PALESTINE.	Positive
5	2024-01-18 10:00:02	Khalissee@Khalissee-2h	In 2001 a hidden camera recorded Netanyahu speak about his genocidal plans for Palestine and how easy it is to influence America. 'We must hit the Palestinians hard...take them to a state of panic and fear...and the world will not say anything.'	Positive
8	2024-01-18 10:00:02	Khalissee@Khalissee-2h	In 2001 a hidden camera recorded Netanyahu speak about his genocidal plans for Palestine and how easy it is to influence America. 'We must hit the Palestinians hard...take them to a state of panic and fear...and the world will not say anything.'	Positive
14	2024-01-18 12:06:24	Zynab @ZynabAlAziz-47m	Palestine is fighting for it's freedom	Positive



## VIII. Conclusion

Voilà un bilan exhaustif de l'ensemble du processus, mettant en lumière les accomplissements, les enseignements tirés, et les perspectives d'avenir:

### A. Récapitulation des Points Clés du Projet :

Le projet s'est distingué par la conception et la réalisation réussies de quatre microservices spécialisés, chacun jouant un rôle déterminant dans l'accomplissement de l'objectif global : fournir une analyse de sentiments en temps réel sur la plateforme Twitter. Le "Tweet Scraper" a assuré la collecte efficace des données, le modèle de machine learning a enrichi l'analyse de sentiments, le frontend a offert une interface utilisateur interactive, et le backend a soutenu l'ensemble du système de manière robuste.

En complément des technologies mentionnées précédemment, l'utilisation de React pour le frontend, scikit-learn pour le modèle de machine learning, et Spring Boot pour le backend a consolidé la base technologique du projet. React a facilité la création d'une interface utilisateur réactive et intuitive, tandis que scikit-learn a apporté des fonctionnalités avancées d'apprentissage automatique pour améliorer la précision de l'analyse de sentiments. Spring Boot a permis de développer un backend fiable, assurant la gestion efficace des données et la communication entre les microservices.

L'intégration réussie de ces technologies a été cruciale pour créer un écosystème cohérent et interactif. Les choix technologiques ont non seulement permis d'atteindre des performances rapides et précises dans l'extraction des données, mais ont également contribué à offrir une expérience utilisateur fluide et complète.

### B. Réussites et Leçons Apprises :

Les réussites marquantes du projet résident dans la collecte en temps réel des tweets pertinents, l'intégration régulière du modèle de machine learning, et la conception réussie d'un frontend réactif. Ces accomplissements ont permis de fournir une expérience utilisateur dynamique et actuelle, tout en garantissant la pertinence des analyses de sentiments sur Twitter.

Cependant, des enseignements essentiels ont émergé tout au long du processus. Ils ont mis en lumière l'importance cruciale de la gestion proactive des erreurs pour maintenir la stabilité du système, l'optimisation continue des performances pour assurer une efficacité opérationnelle optimale, et la nécessité d'une adaptabilité constante face aux évolutions technologiques rapides. Ces leçons constituent une base solide pour des améliorations futures, guidant le projet vers une évolution continue et une excellence dans le domaine de l'analyse de sentiments sur les médias sociaux.

### C. Perspectives d'Avenir et Améliorations Possibles :

Pour l'avenir, le projet envisage d'explorer de nouvelles fonctionnalités, notamment l'intégration de méthodes d'apprentissage automatique plus avancées afin de renforcer la précision de l'analyse de sentiments. L'optimisation continue des performances demeure une priorité, visant à maintenir une efficacité opérationnelle optimale. La prise en charge d'autres réseaux sociaux représente une perspective intrigante, élargissant ainsi la portée de l'application au-delà de Twitter.

Parallèlement, l'implémentation de la personnalisation des paramètres d'analyse pour les utilisateurs individuels constitue une avenue prometteuse, permettant une expérience utilisateur plus adaptée et personnalisée. En résumé, le projet a réussi à établir une application microservices robuste pour l'analyse de sentiments sur Twitter. Les accomplissements et les enseignements tirés fourniront une base solide pour des améliorations futures, positionnant l'application à l'avant-garde de l'innovation dans le domaine dynamique des médias sociaux.