TP pour le cours “Javascript - Web API - JSON”

2024 - 2025

Cours de RIBEZZI Valentin

La problématique

Dans ce projet, vous allez créer une API permettant à des utilisateurs de gérer leurs activités physiques (Randonnée, Course à pied, Cyclisme) et d'interagir avec les activités des autres utilisateurs à travers un système de suivi, de likes, et de statistiques. L'objectif est de construire une plateforme qui permette aux utilisateurs de suivre leurs performances, de se connecter à d'autres utilisateurs, et d'analyser leur progression dans le temps.

L'API devra permettre aux utilisateurs de s'inscrire, de gérer leur profil, d'ajouter des activités, de suivre des abonnés, de liker les activités des autres utilisateurs et d'avoir accès à des statistiques détaillées sur leurs performances. Vous devrez également implémenter une sécurité basique avec l'authentification JWT.

Exercice 1 : Architecture de base et gestion des utilisateurs

- **Objectifs** : Créer l'architecture de l'API, la base de données et gérer les utilisateurs.

- **Instructions** :

    1. Créez l'API avec Express.js et connectez-la à une base de données MySQL.

    2. Implémentez une route `POST /signup` permettant l'inscription d'un utilisateur avec `email`, `password`, `name`.

    3. Implémentez une route `POST /login` pour l'authentification et la génération d'un token JWT.

4. Créez une route protégée `GET /me` permettant de récupérer les informations de l'utilisateur connecté à partir du token JWT.

Exercice 2 : Ajout et gestion des activités physiques

- **Objectifs** : Permettre aux utilisateurs d'ajouter des activités physiques avec des parcours GPS.

- **Instructions** :

    1. Implémentez une route `POST /activities` permettant d'ajouter une activité avec les informations suivantes : `type` (Randonnée, Course à pied, Cyclisme), `distance`, `duration`, `date`, `user_id`, et un `path` (tableau de coordonnées GPS).

    2. Créez une route `GET /activities` qui permet de lister toutes les activités de l'utilisateur connecté.

Exercice 3 : Abonnés et gestion des relations

- **Objectifs** : Ajouter un système de suivi des autres utilisateurs (follow/unfollow) et gérer les relations.

- **Instructions** :

    1. Créez une route `POST /follow/:user_id` permettant à un utilisateur de suivre un autre utilisateur.

    2. Créez une route `POST /unfollow/:user_id` permettant à un utilisateur de se désabonner d'un autre utilisateur.

    3. Créez une route `GET /followers` permettant de lister tous les utilisateurs suivis par l'utilisateur connecté.

    4. Créez une route `GET /activities/following` qui permet d'afficher les activités des abonnés de l'utilisateur connecté.

Exercice 4 : Ajout d'un système de likes sur les activités

- **Objectifs** : Implémenter un système de "like" sur les activités des autres utilisateurs.

- **Instructions** :

    1. Créez une route `POST /activities/:activity_id/like` permettant à un utilisateur de liker une activité.

    2. Créez une route `POST /activities/:activity_id/unlike` permettant de retirer un like d'une activité.

    3. Modifiez la route `GET /activities` pour inclure un champ `likes` indiquant le nombre de likes d'une activité.

Exercice 5 : Statistiques par sport

- **Objectifs** : Calculer et afficher des statistiques par sport pour différentes périodes (semaine, mois, 3 derniers mois, 6 derniers mois).

- **Instructions** :

    1. Créez une route `GET /statistics` permettant de récupérer les statistiques pour chaque sport, avec les données suivantes : `total_distance`, `total_duration`, `average_speed`, pour chaque période.

    2. Implémentez des filtres pour permettre à l'utilisateur de choisir la période : semaine en cours, mois en cours, 3 derniers mois, 6 derniers mois.

Vous soumettez votre code source, via un repo GitHub ou GitLab ou via une archive compressée (ZIP, RAR, …), y compris les fichiers nécessaires à l'exécution du projet, ainsi qu'une brève documentation expliquant comment exécuter et utiliser votre application.

Évaluation

Votre projet sera évalué sur les fonctionnalités réussies et la qualité du code, son organisation et sa lisibilité