

Projet final – Création d'un blog

Énoncé

L'objectif de ce projet est de créer un blog où les utilisateurs auront accès aux fonctionnalités suivantes:

- Créer un article
- Ajouter des commentaires
- Liker/dislike un article
- Follow/Unfollow un blogueur
- Afficher le profile
- Mettre un article dans les favoris

Vous devrez dans un premier temps créer les fonctions qui effectuent les requêtes SQL pour interagir avec la base de données puis créer le server http.

Vous trouverez ci-dessous la liste des endpoints à implémenter.

Endpoints

Endpoints publiques

GET /home -> consulter les 5 derniers articles postés par les utilisateurs

POST /login (email dans le body) -> se connecter à son compte (enregistre l'email dans le server)

POST /user -> créer un utilisateur

Endpoints
nécessitants une
connexion



Users

GET /user/:userid/profile (email dans le body, l'utilisateur doit être log) -> récupérer le profil utilisateur (infos, et articles)

Articles

GET /articles/ (email dans le body, l'utilisateur doit être log) -> récupérer tous les articles d'un blogueur

POST /articles (contenu dans le body) -> ajouter un article sur le blog de l'utilisateur

POST /articles/:articleid/comment (commentaire dans le body) -> ajouter un commentaire à un article

POST /articles/:articleid/like -> liker un article (incrémente les likes de l'article)

POST /articles/:articleid/dislike -> disliker un article

(décrémente les likes de l'article)

DELETE /articles/:article/ -> supprimer un article

Bonus (0.5 par endpoint)

Il es possible qu'il faille modifier la base de donnée afin d'implémneter ces fonctionnalité ^^

GET /followers/:userid -> lister les follower du blogger

POST /follow/:userId -> follow à un blogueur

POST /unfollow/:userId -> unfollow un blogueur

GET /favorites/ -> consulter ses favoris

POST /favorites/:articleid ->ajouter un aticle dans ses favoris
DELETE /favorites/:articleid -> supprimer un aticle dans ses favoris

Structure des tables

```
CREATE TABLE IF NOT EXISTS users (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    email VARCHAR(50),  
    firstname VARCHAR(50),  
    lastname VARCHAR(50)  
);
```

```
CREATE TABLE IF NOT EXISTS articles (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    user_id INT,  
    content VARCHAR(100),  
    likes INT DEFAULT 0,  
    FOREIGN KEY (user_id) REFERENCES users(id)  
);
```

```
CREATE TABLE IF NOT EXISTS comments (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    article_id INT,  
    content VARCHAR(100),  
    FOREIGN KEY (article_id) REFERENCES articles(id)  
);
```

Ressources



db:

- <https://go.dev/doc/tutorial/database-access>

json:

- <https://go.dev/blog/json>
- https://www.alexedwards.net/blog/how-to-properly-parse-a-json-request-body_

http:

- <https://pkg.go.dev/net/http#hdr-Servers>
- <https://gobyexample.com/http-servers>