Réseau social

1. Description

L'objectif du projet est de créer une API web permettant d'interroger la base de données d'un réseau social. L'intérêt de l'API étant que plusieurs projets peuvent s'appuyer dessus (Siteweb, application mobile, application bureau).

Le but de cette API est de gérer un réseau social, vous devrez donc avoir :

- Des utilisateurs (User)
- Des relations entre utilisateurs :
    - Demande d'amitié,
    - Amitié confirmée,
    - Annuler ou refuser une amitié
- Des publications (Post)
- Des commentaires (Comment) associés à une publication
- La possibilité de « tagguer » des utilisateurs dans une publication ou un commentaire
- La possibilité de « liker » des publications ou commentaires
- Des notifications devront être gérées à chaque événement notable (nouvelle amitié, nouveau commentaire sur un de mes posts, nouveau commentaire sur un post où je suis « taggué », nouveau « tag » me concernant)

2. Consignes

Vous devrez vous appuyer sur toutes les notions vues précédemment :

- **POO** : l'API doit être réalisé en programmation orientée objet
- **PDO** : l'API doit interroger la base de données en utilisant PDO.
- **HTTP** : l'API doit être interrogée via des requêtes http (GET, POST, PUT, DELETE). Vous devrez choisir la bonne méthode en fonction du service.

Le format de sortie des services est exclusivement en JSON.

Bonus : Proposer un paramètre output sur chacun des services permettant de choisir le format de sortie.
Exemple : output=json ou output=xml

3. Livrables

- **Modélisation de la base de données :** Vous devrez proposer une modélisation de la base de données, et en livrer une représentation schématique.
- **Documentation de l'API :** Vous devrez proposer une description pour chaque webservice permettant de décrire les entrées et sorties
- **Code de l'API :** Vous devrez fournir le code source de chaque service
- **Requêtes de tests sur PostMan :** Vous devrez fournir un export de vos requêtes.

Exemple de modélisation

Voici ce que le schéma de la base de données peut donner ☺

NB : Pensez à justifier les choix qui semblent le nécessiter

Description des services

Un token devra être généré lors de l'inscription ou de la connexion. Celui-ci devra être transmis dans chacun des appels de service pour identifier l'utilisateur en cours

Utilisateur

1. Créer un utilisateur (Subscribe)

Création d'un utilisateur à partir des informations fournies. Cette méthode renverra également l'utilisateur qui aura été créée ainsi que son identifiant. Cette méthode devra également connecter directement l'utilisateur.
- création d'une session
- renvoi d'un token, il devra être transmis lors de l'appel des autres services pour reconnaître l'utilisateur connecté.

1. Se connecter

Permet à un utilisateur de se connecter à partir de son email et de son mot de passe.
- création d'une session
- renvoi d'un token, il devra être transmis lors de l'appel des autres services pour reconnaître l'utilisateur connecté.

2. Modifier un utilisateur

Modifie un utilisateur existant à partir des informations fournies.

3. Supprimer un utilisateur

Supprime un utilisateur à partir de son identifiant. Il faudra penser à supprimer tous les éléments associés à cet utilisateur.

Amis

1. Créer une demande d'amitié
Permet à un utilisateur de faire une demande d'ami.

2. Répondre à une demande d'amitié
Permet à l'utilisateur connecté d'accepter ou refuser une demande d'amitié

3. Supprimer une amitié
Permet de supprimer une amitié

4. Refuser une demande d'amitié
Permet de refuser une demande d'amitié

5. Annuler une demande d'amitié
Permet à l'utilisateur ayant initié une demande d'amitié de se rétracter

6. Liste de mes amis
Retourne la liste des amis.

NB : il faudra utiliser un système de pagination sur les résultats.
Ce service prendra en paramètre :
offset : Position du premier élément à renvoyer
limit : Nombre d'éléments à renvoyer

7. Rechercher un utilisateur
Permet de faire une recherche sur l'ensemble des utilisateurs. Il faut que l'on puisse distinguer parmi ces utilisateurs ceux qui sont déjà amis avec l'utilisateur connecté.
Attention il ne faut renvoyer l'utilisateur connecté !

NB : il faudra utiliser un système de pagination sur les résultats.
Ce service prendra en paramètre :
offset : Position du premier élément à renvoyer
limit : Nombre d'éléments à renvoyer

8. Liste des utilisateurs ayant le plus de publications sur le réseau

Retourne la liste des utilisateurs comptabilisant le plus de publications dans l'ordre décroissant.

NB : il faudra utiliser un système de pagination sur les résultats.
Ce service prendra en paramètre :
offset : Position du premier élément à renvoyer
limit : Nombre d'éléments à renvoyer

Publication

1. Créer une publication

Création d'une publication à partir des informations fournies.
Un post contient une description qui est en fait un commentaire.
Donc lors de la création du post il doit être possible de créer un commentaire

- Il est possible d'associer (tagguer) des amis à une publication

2. Mettre à jour une publication

Modifie une publication existante à partir des informations fournies.

3. Supprimer un publication

Supprime une publication à partir de son identifiant. Il faudra penser à supprimer tous les éléments associés à cette publication.

4. Liste des publications concernant l'utilisateur connecté et de ses amis (Timeline)

Retourne la liste des publications des amis de l'utilisateur connecté ainsi que les siennes. Le tout doit être trié par date.

NB : il faudra utiliser un système de pagination sur les résultats.
Ce service prendra en paramètre :
offset : Position du premier élément à renvoyer
limit : Nombre d'éléments à renvoyer

5. Liste de mes publications comptabilisant le plus de « like »

Retourne la liste de mes publications ayant reçues le plus de « like »

NB : il faudra utiliser un système de pagination sur les résultats.
Ce service prendra en paramètre :
offset : Position du premier élément à renvoyer
limit : Nombre d'éléments à renvoyer

6. Liste des publications de mon réseau associées à un hashtag donné

Retourne la liste de mes publications et de mes amis ayant un commentaire contenant le hashtag fourni en paramètre

NB : il faudra utiliser un système de pagination sur les résultats.
Ce service prendra en paramètre :
offset : Position du premier élément à renvoyer
limit : Nombre d'éléments à renvoyer

Commentaire

1. Créer un commentaire

Créer un commentaire sur un post.

Il doit être possible :

- d'identifier des amis,
- d'intégrer des « hashtags » dans le contenu du commentaire.

  Il faudra ensuite prévoir d'extraire les « hashtags » pour les associer au commentaire et ainsi permettre une recherche par « hashtag »

2. Modifier un commentaire

Modifie un commentaire existant à partir des informations fournies.

3. Supprimer un commentaire

Supprime un commentaire à partir de son identifiant. Il faudra penser à supprimer tous les éléments associés à ce commentaire.

J'aime

1. « Liker » une publication

Doit permettre d'aimer ou de ne plus aimer une publication

2. « Liker » un commentaire

Doit permettre d'aimer ou de ne plus aimer un commentaire