

PROJET LP

Contexte général

Nous allons nous intéresser ici au d'enveloppement d'un site Web de type DIGG/REDDIT. Un tel site Web est basé sur le principe suivant :

- Des utilisateurs peuvent s'inscrire (en utilisant le nom + prénom) et avoir des amis (identifiés par leur login)
- Un utilisateur peut **ajouter une news** au site si elle n'existe pas déjà dans le site, c'est à dire référencer une page Web (une URL) pour signaler son existence aux autres utilisateurs. Chaque URL est identifiée par un identifiant (un nombre hexadécimal de 32 bits unique pour chaque news)
- Un utilisateur peut **aimer** ou **détester** une news ajoutée précédemment par un utilisateur
- Un utilisateur peut **commenter** une news ajoutée précédemment par un utilisateur

Spécifications des Web Services

Pour chaque service, vous spécifierez les entrées du service, et donnerait un exemple de sortie au format JSON. Par exemple :

== Service Login == Entrée: login : le login de l'utilisateur password : le mot de passe (en clair...) de

l'utilisateur == Description:

Le service permet de s'authentifier sur le site et renvoie, en cas de login réussi, une clef de session. En cas d'erreur, le service renvoie un message d'erreur == Erreurs possibles :

* Login inconnu * Mot de passe

incorrect == Exemple de sortie:

{key = "0BAE569F3AC"} - si login OK

{error = "Login inconnu"} - en cas d'erreur

Bases de Données

On s'intéresse au stockage des différentes données du site. Nous utiliserons **uniquement** une base de données Cassandra pour stocker les informations de ce site.

Rappelez l'intérêt d'utiliser une base Cassandra/NoSQL plutôt qu'une base de type SQL.

On considère qu'une news décrite par : son identifiant (32 caractères), son URL, l'utilisateur qui l'a ajoutée, la date d'ajout et le titre de la news. On souhaiterait stocker **dans le même enregistrement** les j'aime/je déteste ainsi que les commentaires. Soit la news suivante :

Titre: COVID-19 2020 – Le Maroc est félicité pour sa gestion de la crise

Auteur de l'ajout: Rachid

Date de l'ajout: Lun, 16 1vril 23:23:00 2020

URL: <http://fr.yahoo.com/29052012>

J'aime par Mohamed, Abdelhaq

Je déteste par Khalid

- * Date: mar, 17 Avril 12:32:00 2020
- * Auteur : Khalid
- * Commentaire: Avec quoi nous allons vivre si nous ne pouvons pas sortir de chez nous.

Proposez un format de stockage en JSON de cette news.

Si l'on considère que les news sont stockées dans une collection **examen** sur une BD **News123** du serveur cassandra, écrivez la fonction JAVA permettant de stocker une news en fonction de son URL, son titre, la date d'ajout (sous forme de chaîne de caractères), son auteur. Ajoutez une méthode à votre Service permettant d'afficher le titre des news que *Abdelhaq* aime.

Ecrivez une servlet qui prend deux nombres en paramètres, et retourne la multiplication de ces deux nombres (format JSON).

Réécrivez le service de Login afin de permettre une sécurité des mots de passe

Map/Reduce

Si l'on considère maintenant que les j'aime/je déteste sont stockés dans une collection différente de celle des news sous la forme :

```
{ id_news = "A09B8E8CFF...", login
  = "Rachid", score = "1"
} avec score = -1 si l'utilisateur déteste.
```

Ecrivez un service qui utilise les fonctions map et reduce pour calculer le score final (la somme des scores) de toutes les news. Par exemple :

```
{news= "A09B8E8CFF...", score_total = 38} {news=
"0BAE569F3AC...", score_total = -25} ...
```

Ecrivez les fonctions map et reduce permettant de calculer le score moyen donné par chaque utilisateur sur toutes les news - on considérera pour un utilisateur donné, uniquement les news scorées. Par exemple :

```
{login= "Rachid", score_moyen = 2.3} {login= "Abdelhaq",
score_total = -5.3} ...
```

Développement d'un client

On s'intéresse ici au développement d'un client simple, permettant aux utilisateurs d'ajouter et de noter des news.

Proposez une interface en HTML/CSS permettant de répondre à ce besoin.