

30/04/2021

Ruby on Rails

Oktotweet



Gimenez, Tim – VOLAT, Louis
LP CASIR

Contents

Objet du document	2
Organisation	2
Fonctionnel implémenté	2
Choix d'implémentation	3
Difficultés rencontrées	3
Améliorations possibles	3
Ressenties	4

Objet du document

Ce document a pour objectif de présenter différentes parties de gestion et choix réalisé sur le projet Ruby on rails.

Organisation

Présentons l'organisation de ce projet.

Avant toute chose, nous devons d'avoir un environnement de développement opérationnel. Nous avons donc pris le temps d'installer ruby et rails.

Tout d'abord, nous avons commencé par implémenter la gestion d'utilisateur en générant, dans un premier temps le model.

Une fois créé, nous avons développé des tests sur celui-ci :

- Test de création d'un utilisateur et d'unicité de login.
- Le login ne peut posséder que des lettres entre a et z, en majuscule ou en minuscule, peut avoir uniquement le Under score « _ » et des chiffres entre 0 et 9.
- Tests sur la longueur du login ;
- Tests sur la longueur du nom ;
- Test nominal.

Une fois ces tests réalisés, nous avons créé nos contrôleurs puis nos vues.

À la suite de cela, nous avons répété la même manipulation pour toutes les autres fonctionnalités implémentés...

- Tweets ;
- Système de suivi des utilisateurs ;
- Hashtag ;

Enfin, pour chaque module implémenté, les modèles ainsi que les tests ont été géré par Tim GIMENEZ, quant aux contrôleurs et aux vues, celles-ci ont été programmé par Louis VOLAT et Tim GIMENEZ en observation, conseil et vérification du travail.

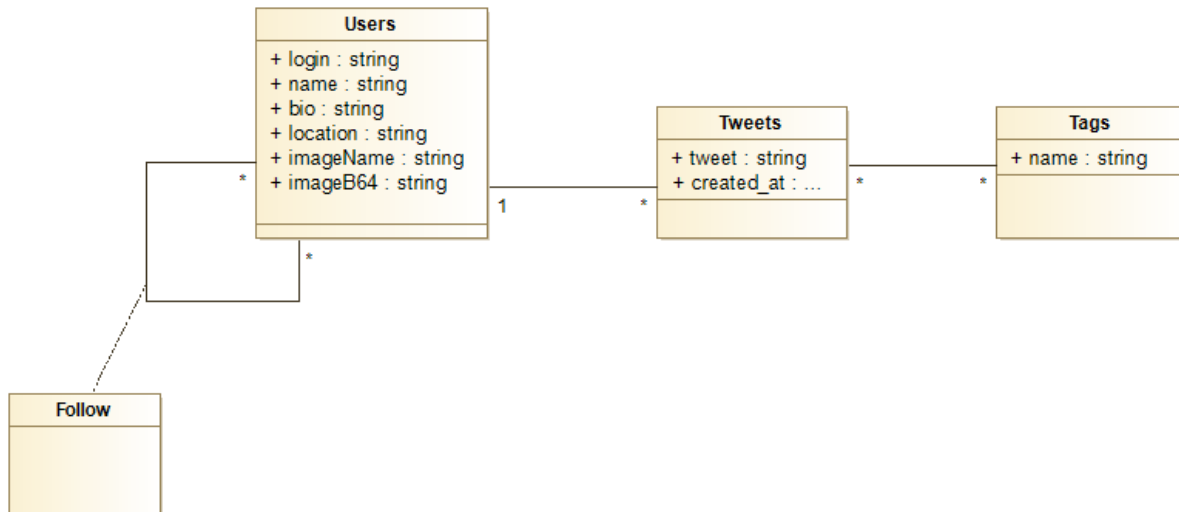
Fonctionnel implémenté

Comme dit précédemment, nous avons implémenté les éléments suivants :

- Gestion des utilisateurs (ajouter / modifier / supprimer un utilisateur) ;
- Possibilité d'ajout de tweets ;
- Visualisation de l'ensemble des tweets de l'utilisateur ;
- Possibilité de suppression de ses propres tweets ;
- Un système de suivi permettant à un utilisateur connecté de suivre ou d'arrêter de suivre un autre utilisateur ;
- Des hashtags peuvent être ajouté aux tweets et la visualisation de ceux-ci se font directement dans le tweet.

Choix d'implémentation

Pour l'implémentation, nous avons choisi de tout implémenté en utilisant le model suivant.



Dans celui-ci vous pouvez observer :

- Une table utilisateurs (gérant les utilisateur) ;
- Une table tweets (offrant la possibilité de gérer les tweets) ;
- Une table de liaison follows (permettant de gérer le suivi des utilisateurs);
- Et une tables tags (stockant les tags de tweets) ;

Difficultés rencontrées

Des difficultés se sont observé à plusieurs moments.

Tout d'abord, l'installation de Ruby a été relativement compliqué. Nous avons obtenu pas mal de message d'erreur avant d'enfin pouvoir commencer à travailler.

Ensuite, nous avons eu beaucoup de mal à installer tailwindCSS. Certes cela n'était pas demandé mais notre volonté de vouloir faire les choses bien et d'en apprendre toujours plus a fait de cela une nécessité.

Ensuite, nous avons présenté des problèmes lors de la génération de la table de liaison des utilisateurs. Un utilisateur devait pouvoir suivre plusieurs utilisateurs et cela signifiait que l'on devait posséder 2 clés étrangères dans notre table.

Enfin, nous avons également rencontré des difficultés lors de différents lancement de tests qui ne se faisaient pas à cause de certaines fixtures incomplètes.

Améliorations possibles

Parmi les améliorations possibles, on aurait bien aimé :

- Ajouter une recherche de tweets / hashtags.
- Créer un système d'authentification avec mot de passe ;
- Offrir la possibilité à l'utilisateur d'aimer un tweet et de voir le nombre de « love » des tweets affichés ;

- Pouvoir ajouter des photos (gif, jpg) ainsi que des liens (urls) aux tweets ;
- Gérer des utilisateurs d'administrateurs ;
- Signaler des tweets ;

Ressenties

Parmi nos ressenti sur Ruby le langage est relativement simple à comprendre et utiliser. Cependant, nous avons trouvé perturbant de ne pas posséder d'accolades comme dans le JS, Java ou encore php.

Quant à Rails, nous trouvons que le projet est bien structuré, on s'y repère bien. En revanche, il exécute beaucoup de chose en arrière-plan ralentissant éventuellement le programme. Il faudrait surement beaucoup plus de connaissances sur rails de façon à bien l'optimiser à 100%.