

Programmation Web sur le Serveur

Mini projet Tweeter (partie 0 - Une vue globale de l'application Web à réaliser)

Amine boumaza

amine.boumaza@univ-lorraine.fr

Contexte

Le but de l'application est de réaliser une mini application [Tweeter](#). Cet outil doit permettre aux utilisateur de créer des Tweet, de parcourir les Tweets existants.

Les acteurs de l'application sont de deux types :

- Tout le monde (invités) : des internautes curieux qui n'ont pas de compte sur l'application. Ces derniers ne peuvent qu'exécuter les fonctionnalités de base : voir les Tweets et voir les Tweets d'un utilisateur donné.
- Les utilisateur authentifiés : possédant un compte sur l'application, et qui peuvent publier des Tweets, suivre d'autres utilisateurs et noter les Tweets (*like*).

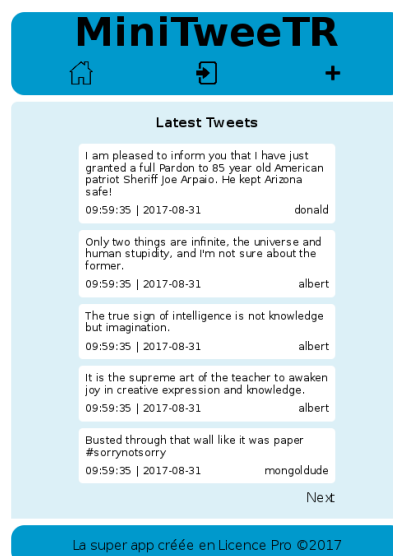
I. Fonctionnalités pour les utilisateurs invités

Afficher la liste des Tweets (1)

Cette Fonctionnalité consiste à afficher une liste des Tweets publiés dans l'ordre inverse de leur date de publication, les plus récents d'abord. Les informations affichées sont :

- le texte du Tweet ;
- la date de publication ;
- le nom de son auteur.

- URL : **/home/** ou juste **/** (la route par défaut)
- Requête : **GET** sans paramètres
- Données modèle : récupère une liste d'objet Tweet
- Autres données : néant



Afficher un Tweet (version invité) (2)

Cette Fonctionnalité consiste a afficher un Tweet. Les informations affichées sont :

- le texte du Tweet ;
- la date de publication ;
- le nom de son auteur;
- le score du Tweet.

- URL : **/view/**
- Requête : **GET** avec un paramètre **id**
- Données modèle : récupère un objet Tweet
- Autres données : l'identifiant du Tweet **id**

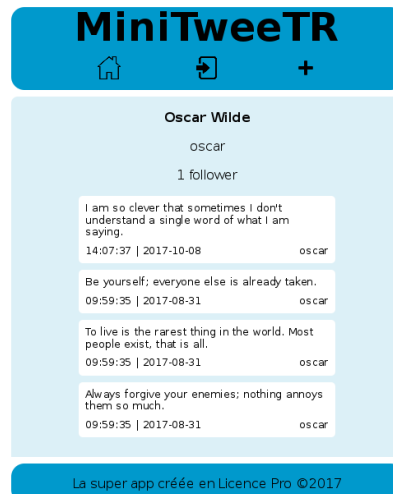


Afficher les Tweets d'un utilisateur (3)

Cette fonctionnalité consiste a afficher les Tweets publiés par un utilisateur donné. Les informations affichées sont :

- le nom complet de l'utilisateur ;
- son non d'utilisateur ;
- le nombre de personnes qui le suivent ;
- la liste de ses Tweets (même format que pour l'affichage de la liste des Tweets).

- URL : **/user/**
- Requête : **GET** avec un paramètre **id**
- Données modèle : récupère un objet User
- Autres données : l'identifiant de l'utilisateur **id**



Créer un compte (4)

Tout les utilisateurs peuvent créer un compte.

Les utilisateurs qui ne disposent pas de comptes peuvent s'inscrire en fournissant les informations suivantes :

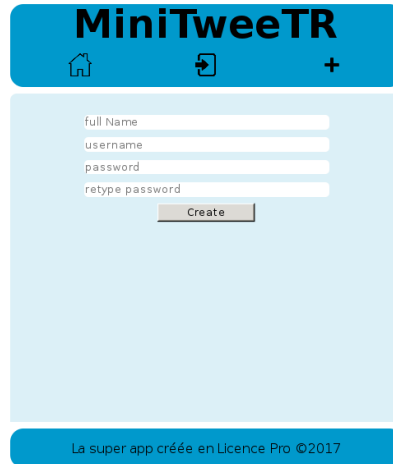
- le nom complet de l'utilisateur ;
- un nom d'utilisateur ;
- un mot de passe (entrer deux fois).

Affichage du formulaire :

- URL : **/signup/**
- Requête : **GET** sans paramètres
- Données modèle : néant
- Autres données : néant

Enregistrement de l'utilisateur

- URL : **/check_signup/**
- Requête : **POST** avec données de formulaire
- Données modèle : crée un nouveau objet User
- Autres données : les champs du formulaire



The image shows a mobile app interface for MiniTweeTR. At the top is a blue header with the app name 'MiniTweeTR' and three icons: a home icon, a share icon, and a plus icon. Below the header is a light blue background containing a signup form. The form has four input fields: 'full Name', 'username', 'password', and 'retype password'. A 'Create' button is positioned below the 'retype password' field. At the bottom of the screen is a blue footer bar with the text 'La super app créée en Licence Pro ©2017'.

Se connecter (5)

Les utilisateurs disposant d'un compte peuvent se connecter à l'application en fournissant :

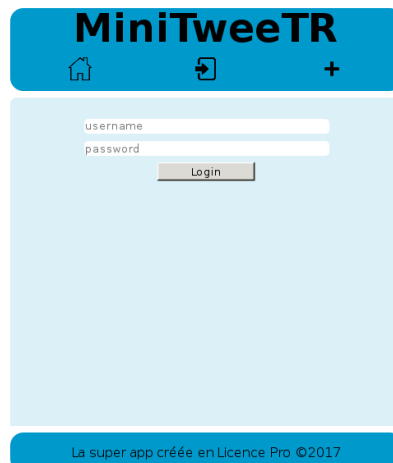
- leur nom d'utilisateur ;
- leur mot de passe.

Affichage du formulaire :

- URL : **/login/**
- Requête : **GET** sans paramètres
- Données modèle : néant
- Autres données : néant

Connexion de l'utilisateur :

- URL : **/checkLogin/**
- Requête : **POST** avec données de formulaire
- Données modèle : récupère un objet User
- Autres données :
 - les champs du formulaire
 - crée une variable de session



The image shows the same mobile app interface for MiniTweeTR, but with the login form displayed. The header and footer are identical to the signup form. The login form has two input fields: 'username' and 'password'. A 'Login' button is positioned below the 'password' field. The background is light blue.

II. Fonctionnalités pour les utilisateurs authentifiés

Rédiger et publier un Tweet (6)

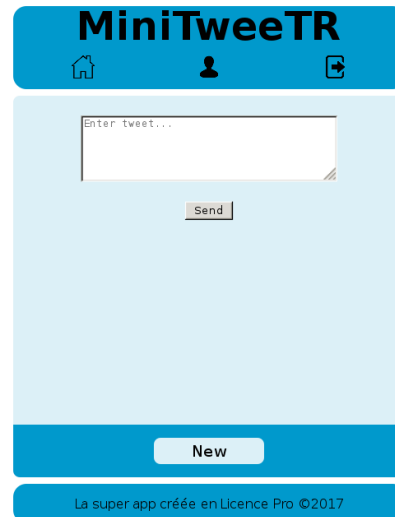
Un utilisateur authentifié peut publier de nouveaux Tweets. Le texte, qui ne doit pas dépasser 140 caractères, est entré dans un formulaire et envoyé avec un bouton valider.

Affichage du formulaire :

- URL : **/post/**
- Requête : **GET** sans paramètres
- Données modèle : néant
- Autres données : néant

Enregistrement du tweet :

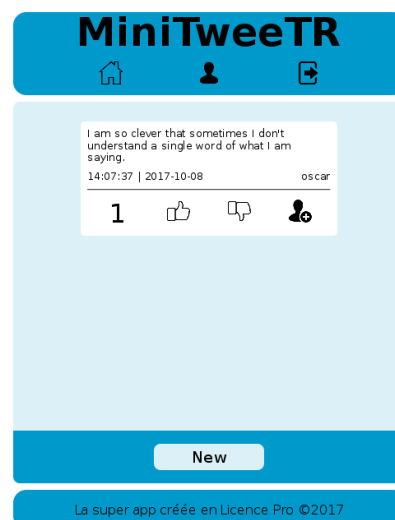
- URL : **/send/**
- Requête : **POST** avec données de formulaire
- Données modèle : récupère un objet User + crée un objet Tweet
- Autres données : les champs du formulaire



Afficher un Tweet (version authentifié) (2 bis)

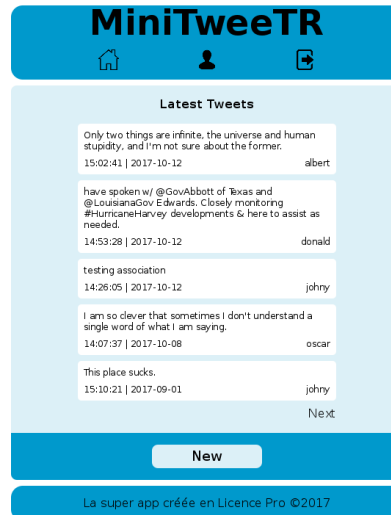
Un utilisateur authentifié peut donner son avis sur les Tweets des autres et suivre d'autres utilisateurs. Ces dernières fonctionnalités sont accessibles depuis l'affichage d'un Tweet qui sera modifié quand un utilisateur est authentifié comme suit :

- des boutons pour évaluer le Tweet ;
- un bouton pour ajouter l'auteur à sa liste des utilisateurs suivis.



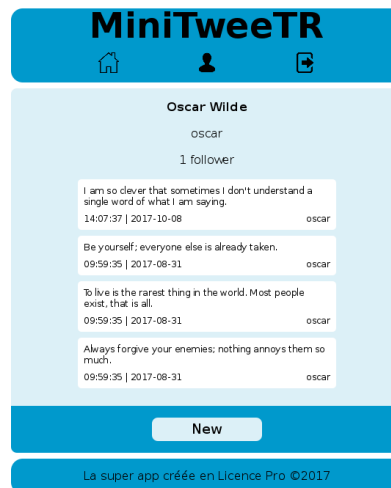
Afficher la liste des Tweets (version authentifié) (1 bis)

Identique à la version invité, excepté pour le menu du haut et le bouton "new" en bas.



Afficher les Tweets d'un utilisateur (version authentifié) (3bis)

Identique à la version invité, excepté pour le menu du haut et le bouton "new" en bas.



Évaluer les Tweets des autres (7)

Les utilisateurs peuvent évaluer les Tweets dont ils ne sont pas les auteurs. En utilisant un système de "j'aime" ou "j'aime pas", ils peuvent ajouter un point respectivement réduire d'un point le score du Tweet.

- Initialement un Tweet a un score de zéro ;

- A chaque fois qu'un utilisateur "aime" un Tweet, son score est incrémenté de 1 et réduit de 1 si un utilisateur n'aime pas ;
 - en fonction de préférences des utilisateurs un Tweet peut avoir un score positif, négatif ou zéro ;

Notes

- Un utilisateur ne peut pas aimer un Tweet qu'il a déjà aimé i.e. un utilisateur ne peut pas évaluer un Tweet plusieurs fois ;
- Un utilisateur ne peut pas évaluer les Tweets qu'il a publiés.

Note : Le modèle **Like** doit être mis à jour.

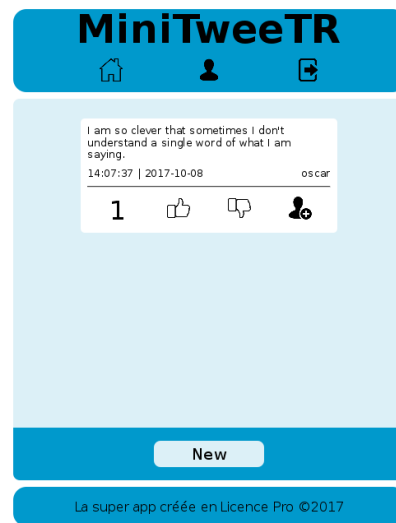
Les boutons  et  permettent d'évaluer un Tweet ils exécutent chacun les routes suivantes

Ajouter un au score du Tweet :

- URL : **/like/**
- Requête : **GET** avec un paramètre **id**
- Données modèle : le Tweet à évaluer et Like
- Autres données : le paramètre **GET**

Diminuer de un le score d'un Tweet :

- URL : **/dislike/**
- Requête : **Get** avec un paramètre **id**
- Données modèle : le Tweet à évaluer et Like
- Autres données : le paramètre **GET**




Suivre les autres utilisateurs (8)

Les utilisateurs authentifiés peuvent suivre les publications d'autres utilisateurs. Si un utilisateur décide de suivre un autre, ce dernier est ajouté à la liste des utilisateurs suivis.

Note : Le modèle **Follow** doit être mis à jour.

Pour suivre un utilisateur, l'utilisateur connecté utilise le bouton , l'utilisateur est ajoutée la liste des suiveur.

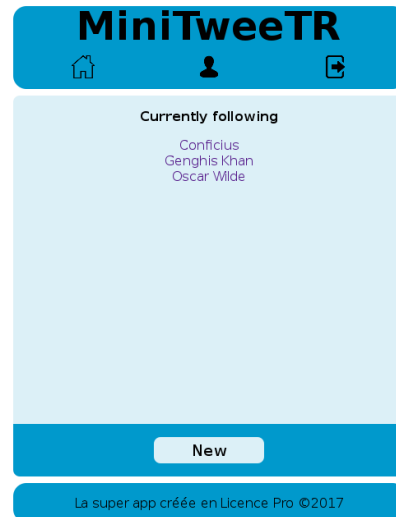
Pour afficher les utilisateur qu'on suit l'utilisateur connectée utilise le bouton .

Suivre un utilisateur :

- URL : **/follow/**
- Requête : **GET** avec un paramètre **id**
- Données modèle : le modèle du User connecté et Follow
- Autres données : le paramètre **GET**

Afficher la liste des utilisateurs qu'on suit :

- URL : **/following/**
- Requête : **GET** sans paramètre
- Données modèle : le modèle du User connecté et Follow
- Autres données : la variable de session de l'utilisateur connecté



Transitions

Voici le schémas de toutes les transitions entre les vues existantes pour les fonctionnalités des parties I et II.

Télécharger le [fichier](#).

