SAE61 - projet 1
Page web d'inscription à un service

1 Présentation

L'objectif de ce projet est de développer la partie "inscription" d'un service web.

En partant de ce que vous avez fait dans l'exercice 4 du TP12 (Flask/regex), il s'agit ici de développer le projet pour inclure l'inscription d'un nouvel utilisateur dans une base de donnée.

1.1 Cahier des charges UI

On demande :

- une page d'accueil, indiquant simplement un lien sur une page "/newuser".
- Cette dernière page proposera trois champs de formulaire web permettant de saisir identifiant, password et adresse mail, avec les contraintes décrites ci-dessous (qui seront validées avec une regex).
- Cette page aura également deux boutons, l'un pour valider la saisie, l'autre pour annuler, ce dernier devant renvoyer l'utilisateur vers la page d'acceuil.
  Si la saisie est acceptée, un message *ad hoc* devra apparaitre sur la page.

1.2 Cahier des charges techniques

On demande une conteneurisation du projet. Il faut prévoir un script qui va construire (le cas échéant) et lancer deux conteneurs, l'un faisant tourner un serveur Web Flask, l'autre faisant tourner un serveur MySql[1]. Un script devra prévoir la création de la base de données dans le serveur ainsi que la création de la table avec ses colonnes (le "schéma" de la BDD, voir ci-dessous).

Les pages web devront être générées via des "templates" (Jinja ou autre, mais l'utilisation de Jinja est probablement la solution la plus simple).

Pour l'affichage du résultat ("ok" ou erreur dans la saisie, ou identifiant ou mail déjà utilisés), on peut prévoir soit une "re-génération" de la page à partir du template, soit une action directe sur le DOM[2]. On peut en effet le manipuler depuis Flask.

En cas de succès, ce nouvel utilisateur devra être stocké dans la BDD, qui devra donc disposer d'une table stockant ces informations. Attention, le password ne **doit pas être stocké en clair**, il faut impérativement stocker le "hash" correspondant. Voir par exemple :
`https://www.geeksforgeeks.org/how-to-hash-passwords-in-python/`

La table aura donc 3 colonnes : identifiant et adresse mail en clair, et hash du password.

Tous vos scripts devront être versionnés dans votre dépot `TP-R504` dans un sous-dossier `sae61-1`.

2 Annexes

2.1 Modalités pratiques

Ce projet doit être mené en **equipe de deux**.

2.2 Contraintes sur les identifiants

1. **Contraintes sur le *username***

- entre 6 et 10 caractères
- uniquement en minuscules
- aucun autre caractère que alphanumérique ASCII (pas d'Unicode !)

1. Ou autre SGBD relationnel
2. *Document Object Model*, voir `https://fr.wikipedia.org/wiki/Document_Object_Model`

De plus, l'identifiant ne doit pas être déjà utilisé. Il faut donc, outre la validation par regex, aller faire une requête SQL dans la BDD et vérifier que l'identifiant n'existe pas déjà. De même pour l'adresse email, si l'adresse est déjà utilisé, il faudra invalider la demande.

2. **Contraintes sur le *password***

— Entre 6 et 15 caractères
— Au moins 1 chiffre
— Au moins 1 majuscule et 1 minuscule
— Au moins 1 caractère parmi les 5 suivants : `#%{}@`

3. **Adresse mail** : vous trouverez facilement des regex adaptées, plus ou moins fidèles à la RFC spécifiant ce qui est permis, mais ne cherchez pas à faire quelque chose de trop compliqué.
(Pensez à lire cette page à ce sujet :
`https://www.bortzmeyer.org/arreter-d-interdire-des-adresses-legales.html`)

En cas d'erreur (non respect des contraintes ou identifiant déjà utilisé), un message d'erreur clair devra apparaitre sur la page.

3 Extensions

Ci-dessous, quelques possibilités d'extensions, si vous arriver à remplir le cahier des charges de départ.

1. Ajouter un "endpoint" `/liste` qui va afficher l'ensemble des identifiants enregistré dans la BDD.
2. Ajouter un endpoint `/connect` qui permet de "se connecter" : un formulaire à deux champs (id/-passwd) et un bouton envoi, qui va provoquer une requete SQL pour vérifier :
- que l'identifiant existe
- que le hash du password donné est le même que le hash du passwd stocké dans la BDD.
En cas de succès, faire un sorte qu'un cookie de connection soit placé dans le navigateur.