

## Java EE – Fiche TP N°2

### Enoncé : Sessions

On souhaite créer une application Web JEE qui utilise la notion de session. L'application comporte deux vues :

- Une page de connexion : elle permet de s'authentifier via nom de compte et mot de passe.
- Une page profile d'utilisateur : elle permet d'afficher les détails du profil de l'utilisateur connecté et éventuellement de se déconnecter

L'application Web doit être conforme au patron de conception MVC.

### Travail à faire :

Créez un nouveau projet de type **Web Application** en lui associant l'URL (Context Path) **/tp\_session\_...** (les points représentent votre prenom, puis \_ , puis votre nom de famille).

#### 1. Création des vues :

Créer la vue login.jsp qui doit contenir :

- un « expression language » permettant d'afficher un éventuel message d'erreur (au cas où les paramètres de connexion sont faux) qui porte le nom « erreur »
- un formulaire avec un champ pour le nom de compte (login), un champ pour le mot de passe (psw) et un bouton d'envoi. Le formulaire envoie les données via la méthode POST au contrôleur (dont l'URL est : myBook).



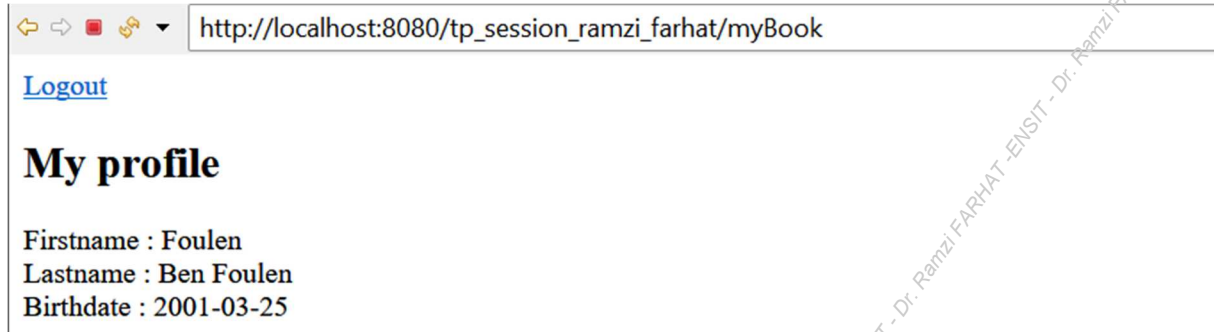
login :  password :

Login or password not correct !

login :  password :

Créer la vue profile.jsp qui doit contenir :

- Un lien hypertexte vers myBook ?logout=true permettant la déconnexion du compte
- Les détails de l'utilisateur (user) via des « expressions language » pour afficher les paramètres : firstname, lastname et birthdate.



## 2. Création du modèle :

Créer une classe Java User qui contient les attributs login, psw, firstname, lastname et birthday. La classe doit contenir un constructeur qui permet d'initialiser tous ces paramètres et les accesseurs et mutateurs associés.

## 3. Création d'une couche services :

Créer une classe UserService qui comporte une liste statique dont le contenu est initialisé dans un bloc statique et qui va simuler une base de données. La classe comporte une méthode statique getUser(String login, String psw) qui retourne un Optional de User.

## 4. Création de la couche contrôleur

### 4.1. Cas d'appel via la méthode GET

#### Algorithme :

---

Récupérer la session

**Si** (session qui ne contient pas l'attribut « user » ) **Alors**

Afficher la vue login.jsp

**Sinon**

Récupérer le paramètre de requête « logout »

**Si** (logout est différent de null et égale à true) **Alors**

Invalidiser la session

Afficher la vue login.jsp

**Sinon**

Afficher la vue profile.jsp

**FinSi**

**FinSi**

---

## 4. 2. Cas d'appel via la méthode POST

### Algorithme :

---

```
Récupérer le paramètre de requête « login »
Récupérer le paramètre de requête « psw »
Utiliser le UserService pour récupérer l'utilisateur sous forme d'un Optional
Si (Une valeur est présente dans Optional) Alors
    Mettre la valeur dans un attribut de session nommé « user »
    Afficher la vue profile.jsp
Sinon
    Ajouter un attribut de requête « error » contenant un message d'erreur
    Afficher la vue login.jsp
FinSi
```

---

### 5. Améliorations possibles :

- Ajouter la possibilité de stocker une liste TODO dans le profil de l'utilisateur afin d'ajouter de pouvoir ajouter des tâches, modifier des tâches ou supprimer des tâches une fois connecté à son compte
- Créer une page d'inscription permettant à de nouveaux utilisateurs de s'inscrire
- Ajouter une couche DAO afin de stocker les utilisateurs dans une base de données
- Ajouter des feuilles de style
- etc.