

TP Sessions

Travail à faire :

Créez un nouveau projet Web JEE. Puis créez une servlet qui teste la présence d'un Cookie avec le nom "visites".

S'il existe, on va : récupérer sa valeur, la transformer en entier, l'incrémenter, afficher un message à l'utilisateur qui comporte le numéro de cette visite, mettre la nouvelle valeur dans le Cookie, mettre l'âge de l'expiration du Cookie à une année et ajouter le cookie à la réponse http. S'il n'existe pas, on va : afficher le message "C'est votre première visite depuis une année !", créer un Cookie avec le nom "visites" et la valeur "1", mettre l'âge de l'expiration à une année et ajouter le Cookie à la réponse.

Enoncé : Sessions

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

- Une page de connexion
- Une page de déconnexion
- Une page d'accès au profile de l'utilisateur (s'il est connecté)

Chaque page est gérée par une Servlet.

Travail à faire :

Créez un nouveau projet de type **Web Application** en lui associant l'URL (Context Path) **/tp2_sessions**.

1. Les trois pages disposent d'un même entête qui comporte le nom de l'application et un menu permettant d'aller d'une page à une autre. Ainsi, nous allons créer une page HTML appelée **header.html** dans un répertoire **include** sous le répertoire **WEB-INF** du projet. Elle contient le code suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <title>TP2 bis</title>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/index.css">
  </head>
  <body>
    <header><h1>Profile d'utilisateur</h1></header>
```

```

<nav>
<a href="LoginServlet">Login</a> |
<a href="LogoutServlet">Logout</a> |
<a href="ProfileServlet">Profile</a>
</nav>
<br><br>

```

2. Dans le même répertoire on va créer également un code HTML qui contient un formulaire de connexion dans un fichier **login_form.html** dont voici le rendu :

Nom de compte	Mot de passe	Connexion
---------------	--------------	-----------

3. Dans le même répertoire on va créer également un code HTML **footer.html** pour compléter le codage de la page html :

```

<footer> M2I - TP JEE </footer>
</body>
</html>

```

4. Vous pouvez créer un répertoire au même niveau que WEB-INF pour créer un répertoire **css** pour contenir un fichier de style CSS de votre choix. Voici un exemple de rendu :

5. Créer une classe « **Utilisateur** » qui représente la session d'un utilisateur sous forme d'un JavaBean qui contient deux variables d'instances : **login** et **psw**. Générer les accesseurs et les mutateurs de ces variables.

6. Développement de la servlet responsable de la connexion : **LoginServlet**

6.1. Cas d'appel via la méthode **POST** (utilisateur qui a cliqué sur le bouton Connexion)

Algorithme :

Mettre le Type de contenu "**text/html; charset=UTF-8**"

Récupérer un **PrintWriter** de la réponse http

Inclure le **header.html**

Si (Valeurs des paramètres de connexion sont valides : contiennent la chaîne « user ») **Alors**

- Créer une instance du JavaBean « Utilisateur » et l'initialiser avec les valeurs du nom de compte et du mot de passe.

- Récupérer l'objet HttpSession et ajouter l'instance du JavaBean comme attribut qui porte le nom « **utilisateur** ».
- Afficher le message "Vous êtes connecté avec le nom de compte : ... "

Sinon

- Afficher le message d'erreur : "Mauvais paramètres de connexion"
- Inclure le fichier **login_form.html**

FinSi

Inclure le fichier **footer.html**

6. 2. Cas d'appel via la méthode GET (appel via menu ou barre d'adresse ou lien hypertexte)

Algorithme :

Mettre le Type de contenu "**text/html; charset=UTF-8**"

Récupérer un **PrintWriter** de la réponse http

Inclure le **header.html**

Si (la session contient un attribut qui porte le nom « **utilisateur** ») **Alors**

- Afficher le message "**vous êtes déjà connecté avec le login : ... !**"

Sinon

- Inclure le fichier login_form.html

FinSi

Inclure le fichier **footer.html**

7. On va créer une servlet pour gérer la déconnexion (dont l'URL est LogoutServlet). Celle-ci, permet de rendre la session invalide et afficher un message à l'utilisateur pour l'informer qu'il n'est pas connecté.

8. Finalement, on va créer une servlet (dont l'URL est ProfileServlet) pour gérer l'affichage de la page contenant le profil de l'utilisateur. Si l'utilisateur est connecté alors on affiche les paramètres de connexion, sinon on le redirige vers la servlet LoginServlet

Remarque : toutes les pages contiennent les fichiers **header.html** et **footer.html**