Gérer des sessions avec les servlets

[sommaire]

- ## Objectifs

  - Le but de ce code est de permettre le passage d'un formulaire d'authentification ou de connexion par exemple, à une autre page, dite d'accueil, spécifique à chaque utilisateur tout en stockant le nom ou le login dans notre variable de sessions.

1. ## Présentation

- Le protocole **HTTP** est un protocole sans état ce qui signifie que le serveur ignore qu'une séquence de requêtes provient d'un même client
  - En mode **HTTP**, pour le serveur, deux requêtes successives d'un même client sont indépendantes
  - Le serveur **HTTP** voit les requêtes, pas les clients.
  - En effet une adresse IP n'est pas suffisante pour identifier un utilisateur
- La classe **Servlet HttpSession** vous montre un moyen d'identifier un utilisateur à travers plusieurs requêtes de page ou visites sur un site Web et de stocker des informations sur cet utilisateur.
- Le conteneur de **servlets** utilise cette interface pour créer une session entre un client HTTP et un serveur HTTP.
- La session persiste pendant une période spécifiée, sur plus d'une connexion ou demande de page de l'utilisateur. Une session correspond généralement à un utilisateur, qui peut visiter un site plusieurs fois.
- La session représente un espace mémoire alloué pour chaque utilisateur, permettant de sauvegarder des informations tout le long de leur visite ;
- Le contenu d'une session est conservé jusqu'à ce que l'utilisateur ferme son navigateur, reste inactif trop longtemps, ou encore lorsqu'il se déconnecte du site ;

- ## Qu'est-ce qu'une session en Java EE ?

  - La session représente un espace mémoire alloué pour chaque utilisateur, permettant de sauvegarder des informations tout le long de leur visite ;
  - Le contenu d'une session est conservé jusqu'à ce que l'utilisateur ferme son navigateur, reste inactif trop longtemps, ou encore lorsqu'il se déconnecte du site ;

apcpedagogie.com

- L'objet Java sur lequel se base une session est l'objet **HttpSession** ;
- Il existe un objet implicite **sessionScope** permettant d'accéder directement au contenu de la session depuis une expression EL dans une page JSP.

- **L'objet HttpSession**

  - L'objet **HttpSession** permet de mémoriser les données de l'utilisateur, grâce à une structure similaire à une table de hachage, permettant de relier chaque id de session à l'ensemble des informations relatives à l'utilisateur.
  - L'objet **HttpSession** s'obtient grâce à la méthode **getSession()** de l'objet **HttpServletRequest**.
  - L'objet HTTPSession permet :
    - La gestion des objets dans la session : ajout, modification, retrait, recherche, etc.
    - La gestion des méta-information à propos de la session elle-même : date de création, id de la session, etc.

Exemple

- Considérons l'objet de type "Personne" suivant : **`Personne personne = new Personne (100, "Nour", "MILI");`**
- Pour créer une variable session : **`HttpSession session = request.getSession();`**
- Pour ajouter l'objet "personne" dans la session : **`session.setAttribute("perso", personne);`**
- Pour récupérer une donnée de session (dans une Servlet) : **`session.getAttribute("perso");`**
- Pour récupérer une donnée de session (dans une JSP)

```
<%
Personne p = (Personne) session.getAttribute("
perso");
out.print(p);
%>
```

Application

- **Exercice 01**

1. **Énoncé**

apcpedagogie.com

- Créer un nouveau projet Java EE.
- Créer trois Servlets **PremierServlet**, **DeuxiemeServlet** et **TroisiemeServlet** accessibles respectivement via les chemins premier, deuxieme et troisieme.
- Créer une vue "index.jsp" contenant un formulaire d'authentification (nom d'utilisateur et mot de passe) et placer la directement dans WebContent.
- A la soumission du formulaire, les données seront envoyées au `doPost()` de PremierServlet.
- Pour accéder à n'importe quelle page de l'application (hormis *index.jsp*), l'utilisateur doit s'authentifier.
- S'il réussit à s'authentifier, il sera enregistré dans la session.
- Créer trois vues **premier.jsp**, **deuxieme.jsp** et **troisieme.jsp** dans WEB-INF appelées respectivement par **PremierServlet**, **DeuxiemeServlet** et **TroisiemeServlet**.
- Dans chaque vue (**premier.jsp**, **deuxieme.jsp** et **troisieme.jsp**), on affiche le nom de la vue, le nom de l'utilisateur connecte, deux liens pour naviguer vers les autres vues et un lien de **deconnexion**.

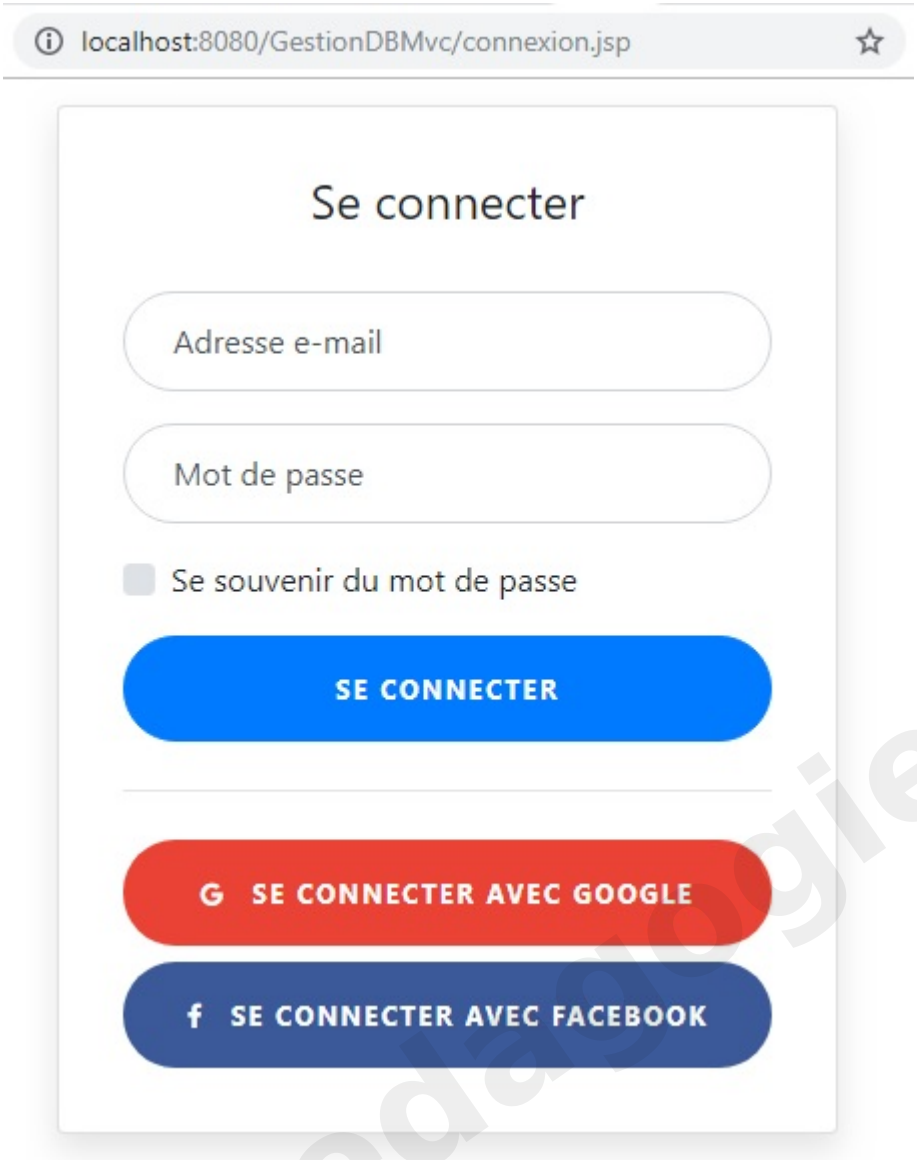
- **Solution**

- **Exercice 02**

1. **Énoncé**

- Le but de cet exemple est d'illustrer seulement le processus du passage des informations d'une page JSP à une autre.
- Pour ce faire nous avons besoin de deux pages JSP et une Servlet
- Le passage d'une page JSP à une autre peut se faire sans passer par une Servlet, mais dans ce cas le passage par une Servlet.
- La première JSP sera nommée: *connexion.jsp* elle contiendra un formulaire qui envoie les données saisies à la Servlet .

apcpedagogie.com



- La Servlet *LoginServlet.java* servira à récupérer les données saisies et les stocker dans la variable de sessions.
- La deuxième page *index.jsp* chargera les données à partir de la Servlet et les affichera avec un message de Bienvenue.

- **Solution**

Sommaire du cours JSP

Source : https://apcpedagogie.com/gerer-des-sessions-avec-les-servlets · téléchargé le 04/11/2026 · © APC Pédagogie · apcpedagogie.com