



## TP 4 : Session

Dans ce TP, nous aborderons le suivi de session dans JSP. HTTP est un protocole "sans état" qui signifie que chaque fois qu'un client récupère une page Web, le client ouvre une connexion distincte au serveur Web et le serveur ne conserve automatiquement aucun enregistrement de la demande du client précédent.

### Cookie et Session

Un serveur Web peut attribuer un identifiant de session unique en tant que cookie à chaque client Web. Pour les demandes ultérieures du client, ils peuvent être reconnus à l'aide du cookie reçu. Cela peut ne pas être un moyen efficace, car le navigateur ne prend parfois pas en charge les cookies. Il n'est pas recommandé d'utiliser cette procédure pour gérer les sessions.

### L'objet de session

Outre les options mentionnées, JSP utilise l'interface `HttpSession` fournie par le servlet. Cette interface permet d'identifier un utilisateur.

- Une demande d'une page
- Visiter un site web
- Stocker des informations sur cet utilisateur

Par défaut, le suivi de session est activé dans les JSP et un nouvel objet `HttpSession` est automatiquement instancié pour chaque nouveau client. La désactivation du suivi de session nécessite explicitement de le désactiver en définissant l'attribut `session` de la directive de page sur `false` comme suit :

```
<%@ page session = "false" %>
```

## 1. Création Session

Cet exemple explique comment utiliser l'objet `HttpSession` pour connaître l'heure de création et l'heure de la dernière utilisation d'une session. Nous associerions une nouvelle session à la demande s'il n'en existe pas déjà une.

```
<%@ page import = "java.io.*,java.util.*" %>
<%
    // Get session creation time.
    Date createTime = new Date(session.getCreationTime());

    // Get last access time of this Webpage.
    Date lastAccessTime = new Date(session.getLastAccessedTime());

    String title = "Welcome Back to my website";
    Integer visitCount = new Integer(0);
    String visitCountKey = new String("visitCount");
    String userIDKey = new String("userID");
    String userID = new String("ABCD");
%>
```



```
// Check if this is new comer on your Webpage.  
if (session.isNew() ){  
    title = "Welcome to my website";  
    session.setAttribute(userIDKey, userID);  
    session.setAttribute(visitCountKey, visitCount);  
}  
visitCount = (Integer)session.getAttribute(visitCountKey);  
visitCount = visitCount + 1;  
userID = (String)session.getAttribute(userIDKey);  
session.setAttribute(visitCountKey, visitCount);  
%>
```

```
<html>  
  <head>  
    <title>Session Tracking</title>  
  </head>  
  
  <body>  
    <center>  
      <h1>Session Tracking</h1>  
    </center>  
  
    <table border = "1" align = "center">  
      <tr bgcolor = "#949494">  
        <th>Session info</th>  
        <th>Value</th>  
      </tr>  
      <tr>  
        <td>id</td>  
        <td><% out.print( session.getId()); %></td>  
      </tr>  
      <tr>  
        <td>Creation Time</td>  
        <td><% out.print(createTime); %></td>  
      </tr>  
      <tr>  
        <td>Time of Last Access</td>  
        <td><% out.print(lastAccessTime); %></td>  
      </tr>  
      <tr>  
        <td>User ID</td>  
        <td><% out.print(userID); %></td>  
      </tr>  
      <tr>  
        <td>Number of visits</td>  
        <td><% out.print(visitCount); %></td>  
      </tr>  
    </table>  
  
  </body>  
</html>
```



## 2. Suppression Session

Lorsque vous avez terminé avec les données de session d'un utilisateur, vous avez plusieurs options -

- **Supprimer un attribut particulier** - Vous pouvez appeler la méthode **public void removeAttribute(String name)** pour supprimer la valeur associée à la clé particulière.
- **Supprimer toute la session** - Vous pouvez appeler la méthode **public void invalidate()** pour supprimer une session entière.
- **Définition du délai d'expiration de session** - Vous pouvez appeler la méthode **public void setMaxInactiveInterval(int interval)** pour définir le délai d'expiration d'une session individuellement.
- **Déconnecter l'utilisateur** - Les serveurs prenant en charge les servlets 2.4, vous pouvez appeler la déconnexion pour déconnecter le client du serveur Web et invalider toutes les sessions appartenant à tous les utilisateurs.
- **Configuration web.xml** - Si vous utilisez Tomcat, outre les méthodes mentionnées ci-dessus, vous pouvez configurer le délai d'expiration de la session dans le fichier web.xml comme suit :

```
<session-config>  
  <session-timeout>15</session-timeout>  
</session-config>
```