

TP1 gestion des sessions

Objectif du TP

1. Comprendre le fonctionnement des sessions HTTP.
2. Apprendre à enregistrer, lire et supprimer des attributs de session dans des servlets.
3. Mettre en pratique l'interaction entre plusieurs servlets à l'aide de Tomc

Vous allez créer trois servlet une qui créer la session pour un utilisateur et y stock son nom, une deuxième pour consulte le nom stocké si il existe et la troisième détruit la session

Structure du TP

Étape 1 : Configuration de l'environnement

1. Assurez-vous que Tomcat est configuré dans Eclipse.
 2. Créez un nouveau projet web dynamique nommé TP_Servlet_Sessions.
 3. Configurez le fichier `web.xml` (ou utilisez des annotations).
-

Étape 2 : Implémentation des Servlets

Servlet Connect

Cette servlet permet de créer une session et d'y enregistrer un paramètre (`nom`).

```
@WebServlet("/connect")
public class Connect extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String name = request.getParameter("nom");
        HttpSession session = request.getSession();
        session.setAttribute("username", name);

        response.setContentType("text/html");
        response.getWriter().println("<h1>Bonjour, " + name + "!</h1>");
        response.getWriter().println("<a href='user'>Accéder à User</a>");
    }
}
```

Servlet User

Cette servlet affiche le nom stocké dans la session.

```
@WebServlet("/user")
public class User extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        HttpSession session = request.getSession(false); // Ne crée pas de
session si elle n'existe pas
        response.setContentType("text/html");

        if (session != null && session.getAttribute("username") != null) {
            String name = (String) session.getAttribute("username");
            response.getWriter().println("<h1>Bienvenue, " + name +
"!</h1>");
            response.getWriter().println("<a
href='disconnect'>Déconnexion</a>");
        } else {
            response.getWriter().println("<h1>Session non trouvée.
Connectez-vous d'abord.</h1>");
            response.getWriter().println("<a href='index.html'>Se
connecter</a>");
        }
    }
}
```

Servlet Disconnect

Cette servlet détruit la session et redirige l'utilisateur vers la page d'accueil.

```
import java.io.IOException;

@WebServlet("/disconnect")
public class Disconnect extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        HttpSession session = request.getSession(false); // Récupérer la
session existante
        if (session != null) {
            session.invalidate(); // Détruire la session
        }

        response.setContentType("text/html");
        response.getWriter().println("<h1>Vous avez été déconnecté.</h1>");
        response.getWriter().println("<a href='index.html'>Se connecter à
nouveau</a>");
    }
}
```

Étape 3 : Fichier HTML pour le formulaire

Créez un fichier `index.html` dans le dossier `webapp` ou `src/main/webapp`.

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <title>Connexion</title>
</head>
<body>
  <h1>Connexion</h1>
  <form action="connect" method="post">
    <label for="nom">Nom :</label>
    <input type="text" id="nom" name="nom" required>
    <button type="submit">Se connecter</button>
  </form>
</body>
</html>

```

Étape 4 : Configuration du fichier `web.xml`

Si vous ne voulez pas utiliser les annotations, configurez les servlets dans le fichier `web.xml`.

```

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <servlet>
    <servlet-name>Connect</servlet-name>
    <servlet-class>Connect</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Connect</servlet-name>
    <url-pattern>/connect</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>User</servlet-name>
    <servlet-class>User</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>User</servlet-name>
    <url-pattern>/user</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>Disconnect</servlet-name>
    <servlet-class>Disconnect</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Disconnect</servlet-name>
    <url-pattern>/disconnect</url-pattern>
  </servlet-mapping>

</web-app>

```

Étape 5 : Test

1. Lancez Tomcat à partir d'Eclipse ou manuellement.
2. Accédez à `http://localhost:8080/TP_Servlet_Sessions/index.html`.
3. Entrez un nom dans le formulaire pour vous connecter.
4. Naviguez vers `/user` pour afficher le nom stocké dans la session.
5. Accédez à `/disconnect` pour détruire la session.