



TD Applications Réparties

Programmation Web

1. Programmation Web

Dans ce Td nous allons utiliser les servlets qui constituent un moyen d'offrir des services sur le Web par l'intermédiaire d'un serveur http.

Un(e) Servlet fournit les services par le biais de requêtes qui lui parviennent via le serveur http. Elle réagit à ces requêtes via une réponse qui fait partie intégrante du modèle.

La méthode dont nous ferons usage ici est doGet qui correspond à une requête dont les paramètres (éventuels) font partie de la requête http. Il existe d'autres façons de communiquer (doPost, ...).

Pour construire notre application, définissez un projet de type "Dynamic Web Project" que vous nommerez Jus.Aor.Jsp.

Dans l'onglet "Java Resources" dans la partie src construisez le package Jus.Aor.Jsp.

La servlet HelloWorld

Dans l'onglet WebContent construisez le fichier web.xml qui contient :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_ID"
  version="3.0">
  <display-name>Jus.Aor.Jsp</display-name>
  <servlet>
    <servlet-name>HelloWorld</servlet-name>
    <servlet-class>jus.aor.jsp.HelloWorld</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloWorld</servlet-name>
    <url-pattern>/HelloWorld</url-pattern>
  </servlet-mapping>
```



```
</web-app>
```

Dans ce fichier on décrit la servlet avec un nom local et la classe y correspondant. Dans une seconde partie on fixe le nom (uri) de celle-ci dans l'espace http.

Développez la classe HelloWorld qui héritera de `HttpServlet` et qui dans sa méthode `doGet` renverra le texte `"Hello World from HelloWorld Class"`.

Pour fonctionner vous devez utiliser un serveur http (soit local, ...). Celui-ci contient un directory nommé "webapp". Depuis votre projet exportez (export) le war associé à vos servlets avec le nom "Jus.Aor.Jsp.war".

Sur l'onglet servers dans la liste de vos projets, vous devez voir votre serveur tomcat et votre Jsp (Jus.Aor.Jsp)., démarrez le serveur tomcat.

Depuis un navigateur tapez l'adresse de votre serveur http complétée du nom de la servlet `"/Jus.Aor.Jsp/HelloWorld"`, vous devez en retour voir afficher le message renvoyé.

La Jsp HelloWorld

Dans l'onglet WEB-INF créez le fichier HelloWorld.jsp dont le contenu est :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <% out.println("Hello World JSP"); %>
</body>
</html>
```

Depuis un navigateur tapez l'adresse de votre serveur http complétée du nom du Jsp `"/Jus.Aor.Jsp/HelloWorld.jsp"`, vous devez en retour voir afficher le message renvoyé (Hello World JSP).

La servlet Calculatrice

On souhaite réaliser une servlet plus sophistiquée capable de réaliser des opérations arithmétiques élémentaires (+, -, *, /) et de renvoyer les résultats de celles-ci. Vous utiliserez aussi la méthode `doGet` qui cette fois aura en paramètres les valeurs des opérandes ainsi que l'opération à réaliser.

Au niveau de votre projet définissez le fichier Calculatrice.html suivant qui servira d'interface utilisateur :



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title></title>
</head>
<body>
    <div style="text-align: center">
        <strong>
            Résultat de l'opération: "Nombre1" "Opération" "Nombre2"
= "Résultat"
        </strong>
        <form method="get" action="">
            <p>
                <input type="submit" name="nouvelleOperation"
value="Nouvelle opération" />
            </p>
        </form>
        <strong>Effectuer votre opération</strong> <br/> <br/> <br/>
        <form method="get"
action="http://localhost:8080/Jus.Aor.Jsp/Calculatrice">
            <p>
                <input type="text" name="nombre1" />
                <input type="text" name="nombre2" />
                <select name="operation">
                    <option value="+"
selected='selected'>Addition</option>
                    <option value="-">Soustraction</option>
                    <option value="*">Multiplication</option>
                    <option value="/">Division</option>
                </select>
                <input type="submit" name="calculer"
value="Calculer" />
            </p>
        </form>
    </div>
</body>
</html>
```

Pour intégrer cette futur servlet, vous devez ajouter dans votre fichier web.xml, la description de celle-ci. Pour éditer ce fichier vous pouvez désormais utiliser l'onglet "Deployment Descriptor: Jus.Aor.Jsp". Vous appellerez celle-ci "Calculatrice".

Réalisez dans le package Jus.Aor.Jsp la classe Calculatrice.java dont voici le squelette :

```
package jus.aor.jsp;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
```



```
* Servlet implementation class Calculatrice
*/
public class Calculatrice extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Calculatrice() {super();}

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
    HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        response.setContentType("text/html");
        response.getWriter().print(String.format(reponse, title, title, n1, opera
        tion, n2, result));
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
    HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        doGet(request, response);
    }

    /**
     * le squelette de la réponse html qui attend :
     * l'opérande n°1
     * l'opération
     * l'opérande n°2
     * le résultat
     */
    private static String reponse =
        "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
    Transitional//EN">\n<HTML>\n<HEAD><TITLE>Résultat de
    l'opération</TITLE></HEAD>\n"+
        "<BODY BGCOLOR=\"#FDF5E6\">\n" +
        "<H2 ALIGN=CENTER>Résultat de l'opération</H2>\n" +
        "%s %s %s = %s"+
        "<form>"+
        "<input type=\"button\" value=\"Retour\"
    onclick=\"history.go(-1)\">"+
        "</form>;"
}
```

Fabriquez à nouveau votre fichier Jus.Aor.Jsp.war, dès lors vous pouvez utiliser votre Calculatrice via le fichier html.

La servlet Caddy

Il s'agit d'avoir une servlet qui comporte une session permettant de sauvegarder des informations d'une requête à une autre.

Créez le sous-package jus.aor.jsp.caddy dans lequel vous allez définir les classes Client et Caddy. La classe Client comporte 2 attributs et les setter/getter associés dénombrant la quantité réservée par le client du produit1 et produit2 en vente.



La classe Caddy disposera d'une Map associant la HttpSession avec le client correspondant. A chaque requête (doGet), on récupérera cette HttpSession ce qui permettra ainsi d'accéder aux caractéristiques du client. Lors de la requête, elle "dispatchera" à la Jsp Caddy le traitement, elle pourra aussi imprimer la situation du client.

Dans l'onglet WebContent construisez le fichier Caddy.jsp suivant où vous complèterez le calcul des quantités (q1, q2) à l'endroit opportun :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <%@ page session="true" import="jus.aor.jsp.caddy.Client"%>
    <jsp:useBean id="Client" class="jus.aor.jsp.caddy.Client"/>
    <%!
        int q1=0, q2=0;
    %>
    <div style="text-align: center">
        <strong>
            Caddy
        </strong>
        <form method="get"
action="http://localhost:8080/Jus.Aor.Jsp/Caddy">
            <p>
                <input type="submit" name="Produit1" value="One
more Product 1" />
                <input type="submit" name="Produit2" value="One
more Product 2" />
            </p>
        </form>
    </div>
    <div style="text-align: center">
        <strong>
            Produit1 = <%= q1 %><br>
            Produit2 = <%= q2 %><br>
            Total = <%= q1+q2 %>
        </strong>
    </div>
</body>
</html>
```

Mettez à jour votre web.xml et testez.