

TP : Servlet 1

Exercice 1. Première servlet

Les servlets sont des programmes qui fonctionnent sur un serveur et qui construisent des pages web à la volée (comme pour PHP). Les servlets sont également des objets Java qui permettent la programmation simple de pages web dynamiques. Les servlets, qui nous concernent, sont des HttpServlet qui permettent de répondre à des requêtes GET :

```
import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

public class HWorld extends HttpServlet {

    public void doGet(HttpServletRequest request,

        HttpServletResponse response)

        throws ServletException, IOException {

        // Utilisez "request" pour lire les entêtes HTTP entrante

        // (Comme les cookies...) et le contenu des formulaires

        // Utilisez "response" pour creer la reponse HTTP

        PrintWriter out = response.getWriter();

        // Utilisez "out" pour envoyer du contenu au navigateur

    }

}
```

Dans un fichier HWorld.java. Une servlet est donc une classe Java fille de la classe

HttpServlet et qui permet, si on surcharge la fonction doGet, de générer du contenu à toute requête GET du serveur. La classe HttpServletRequest permet d'obtenir les entêtes de la requête HTTP, les valeurs d'un formulaire, etc. La classe HttpServletResponse possède des méthodes permettant de spécifier la réponse HTTP, son entête et surtout de pouvoir obtenir un PrintWriter utilisable pour envoyer du contenu au client.

Par exemple, notre servlet pourrait facilement envoyer du texte simple au client en rajoutant la ligne :

```
out.println("Hi guys");
```

Mais le plus intéressant est de construire une page web. Pour ce faire, nous devons dire au navigateur (client) que l'on va envoyer du HTML et renvoyer non plus du texte pur mais bel

Et bien une page HTML. Pour dire au navigateur qu'il reçoit une page HTML il convient de fixer le content-type à text/html grâce à la méthode `setContentType` (de quel objet java à votre avis ?) puis (attention l'ordre est important) il reste à écrire la page web à l'aide du `PrintWriter`.

Modifiez le fichier `HWorld` précédent afin d'écrire "Hi Guys" dans des balises `<h1>`. Ne vérifiez pas tout de suite le résultat.

Exercice 2. Déploiement de servlet sur le serveur local

Réalisez une page web `index.html` (un simple fichier html donc) qui affiche un texte de bienvenue et propose un lien vers la page servlet/`HWorld`. Placez le fichier dans votre répertoire `bonjour` et rechargez la page qui doit s'afficher.

Ajouter des servlets est à peine plus compliqué. Vous devez enregistrer votre servlet auprès du fichier `web.xml` du répertoire `WEB-INF`. Ajoutez à ce fichier les lignes suivantes :

```
<servlet>
<servlet-name>welcome</servlet-name>
<servlet-class>HWorld</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>welcome</servlet-name>
<url-pattern>/servlet/*</url-pattern>
</servlet-mapping>
```

La balise `<servlet>` permet de définir une nouvelle servlet c'est à dire son nom et sa classe de base. Il peut également contenir une balise `<init-param>` permettant de fixer une variable globale à une certaine valeur en utilisant deux sous balise `<init-param>` : `<param-name>` Et `<param-value>`. Il est nécessaire de rajouter une information de "mapping" permettant au serveur de savoir où chercher la servlet.

Déplacez le fichier HWorld.java précédent dans le répertoire bonjour/WEB-INF/classes. Compilez-le. Puis rechargez la page précédente et cliquez sur le lien. La page doit s'afficher.

Exercice 3. Date

Générer une page HTML qui contient l'heure et la date, à chaque fois qu'on actualise.

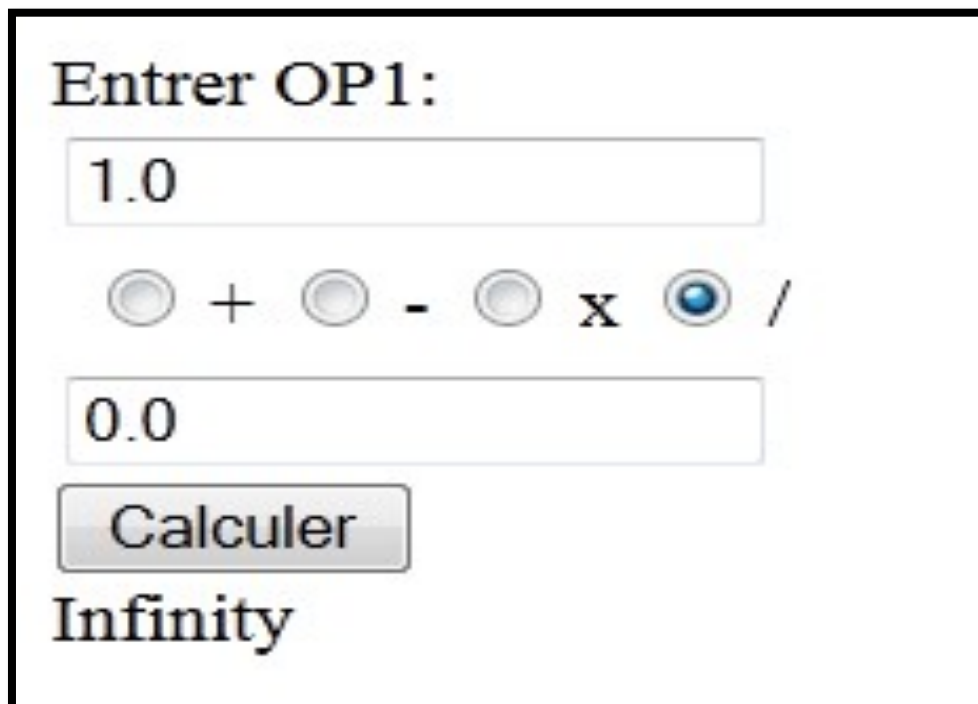
Exercice 4. La force des servlets

Les servlets bénéficient de toute la puissance et la portabilité de Java. Elles proposent également des moyens simples d'accès aux différentes données contenues sur le serveur et envoyées par le client.

Ainsi, récupérer des valeurs issues d'un formulaire HTML se fait simplement en appelant la méthode `getParameter` de l'objet `HttpServletRequest`.

Faire un formulaire HTML qui demande le nom de l'utilisateur et affiche une page d'accueil (servlet) avec ce nom, prénom, âge et la date courante.

Exercice 5. Calculatrice, qui garde le résultat d'avant.



The image shows a web-based calculator interface. It features two input fields for numbers. The first field is labeled "Entrer OP1:" and contains the value "1.0". The second field contains the value "0.0". Between the two fields is a row of radio buttons for selecting an operator: "+", "-", "x", and "/". The "/" radio button is currently selected. Below the input fields is a button labeled "Calculer". At the bottom of the interface, the word "Infinity" is displayed, indicating the result of the calculation.

Exercice 6. Sauvegarde des éléments.

Créez la classe « ServletListe » En se basant sur le fichier Html suivant

```
<!--fichier html pour l'exemple de la figure Q6.1-->
<html>
<body>
<head>
<title>Servlet Liste</title>
</head>
<body bgcolor="white">
<h3>Add</h3>
<P>
<form action="../servlet/ServletListe" method=POST>
Titre:
<input type=text size=20 name="titre">
<br>
Description:
<input type=text size=20 name="description">
<br>
<input type="submit" name="add" value="Add">
</form>
</body>
</html>
```



vert	grenouille, gazon
blanche	poudre, barbe, boule
rouge	drapeau, pomme, rosette

Remarque : la Servlet utilisera la classe ItemBean contenant deux propriétés (String title; String description;)