

TP 1 : Java EE

Création du projet web avec Eclipse

- Sélectionnez **Dynamic Web Project**. J'appelle ici mon projet *test*.
- Cliquez sur le bouton New Runtime...et sélectionnez alors Apache Tomcat 9.0
- Remplissez correctement les informations relatives à votre installation de Tomcat en allant chercher le répertoire d'installation de Tomcat sur votre poste.
- Validez alors en cliquant sur Finish et le projet est créé.

Création d'une page web

- Sélectionnez HTML File, puis donnez un nom à votre page et enfin validez. Je nomme ici ma page *index.html*.
- Ajoutez en un paragraphe le texte " Ceci est une page HTML " dans votre page HTML.
- Pour faire l'exécution, lancez le serveur Tomcat (run on server).

Création de Servlet

- Choisissez New > Servlet
- Nommez le package par "com.servlets" et la servlet par "Test".
- Enlevez les commentaires et le contenu de doGet () et doPost ().
- Créez un fichier XML sous le répertoire /WEB-INF et nommez le web.xml.

Ajoutez la balise web-app

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<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">`

`</web-app>`

- Déclarez la servlet :

`<servlet>`

`<servlet-name>.....</servlet-name>`

`<servlet-class>.....</servlet-class>`

`</servlet>`

- Il faut ensuite faire correspondre notre servlet déclarée à une URL, afin qu'elle soit joignable par les clients :

`<servlet-mapping>`

`<servlet-name>.....</servlet-name>`

`<url-pattern>.....</url-pattern>`

`</servlet-mapping>`

- Lancez le serveur web par le bouton vert. Que remarquez-vous ?
- Exécutez en utilisant url `http://localhost:8085/test/test1`

- Dans la servlet, ajoutez ce code pour faire un affichage :

```
public void doGet( HttpServletRequest request, HttpServletResponse response )  
throws ServletException, IOException{
```

```
    response.setContentType("text/html");
```

```
    response.setCharacterEncoding( "UTF-8" );
```

```
    PrintWriter out = response.getWriter();
```

```
    out.println("ma Servlet");
```

```
}
```

- Mettez en place une simple page HTML pour afficher " Ceci est une page générée depuis une servlet " dans le navigateur du client

Mise en place d'une JSP

Création de la vue

- Depuis votre projet Eclipse faites un clic droit sur le dossier WebContent de votre projet, puis choisissez New > JSP File, nommez la page JSP par *bonjour.jsp*.
- Videz la méthode `doGet()` du contenu et migrez le dans la JSP
- Pour associer la servlet à notre vue, appelez la JSP à partir de `doGet()`, La méthode `forward()` de l'objet `RequestDispatcher` permet depuis une servlet de rediriger la paire requête/réponse HTTP vers une page JSP.

```
public void doGet( HttpServletRequest request, HttpServletResponse response )
    throws ServletException, IOException {

    this.getServletContext().getRequestDispatcher( "page.jsp" ).forward( request,
response );}
```

Transmission de données

Transmettre des variables de la servlet à la JSP

- Créez une chaîne de caractères depuis la servlet, et transmettez-la à la JSP pour affichage en appelant la méthode `setAttribute()` de l'objet requête pour y enregistrer un attribut.

Cette méthode prend en paramètre le nom que l'on souhaite donner à l'attribut suivi de l'objet lui-même.

- Récupérer et afficher l'objet côté vue en appelant la méthode `getAttribute()`
- Ajouter une boucle for dans la JSP pour afficher " une boucle Java"
- Exécuter une condition, en affichant bonjour, si c'est le jour sinon bonsoir, sachant que la servlet définit si c'est le jour ou non.

Pour éviter la répétition du même code dans plusieurs pages JSP, il faut centraliser le code dans une page spécifique (inclusions de JSP).

- Inclure cette partie de code dans votre page JSP en utilisant `<%@ include file=" "%>`

menu.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

```
<ul>
```

```
    <li>page 1</a></li>
```

```
    <li>page 2</a></li>
```

```
    <li>page 3</a></li>
```

```
    <li>page 4</a></li>
```

```
</ul>
```

- Créer une nouvelle servlet Accueil.java, la définir dans web.xml ensuite créer une nouvelle JSP Accueil.jsp qui contient un menu de 4 pages.

Avec la page Accueil forme la racine de notre site (url /)

- Modifier le menu dans les deux pages JSP et ajouter un lien vers chaque page JSP.