

EISTI – TP de Java EE

Séance 2 : Introduction aux Servlets

1 : Import/Export WAR sous Eclipse

- Reprenez le WAR créé dans le TP précédent et l'importer dans Eclipse.
- Modifier la Servlet pour qu'elle renvoie un tableau (balise <table>) de 10 cases par 10 selon le modèle suivant :

M(1,1)	M(1,2)	M(1,3)	M(1,4)	M(1,5)	M(1,6)	M(1,7)	M(1,8)	M(1,9)	M(1,10)
M(2,1)	M(2,2)	M(2,3)	M(2,4)	M(2,5)	M(2,6)	M(2,7)	M(2,8)	M(2,9)	M(2,10)
M(3,1)	M(3,2)	M(3,3)	M(3,4)	M(3,5)	M(3,6)	M(3,7)	M(3,8)	M(3,9)	M(3,10)
M(4,1)	M(4,2)	M(4,3)	M(4,4)	M(4,5)	M(4,6)	M(4,7)	M(4,8)	M(4,9)	M(4,10)
M(5,1)	M(5,2)	M(5,3)	M(5,4)	M(5,5)	M(5,6)	M(5,7)	M(5,8)	M(5,9)	M(5,10)
M(6,1)	M(6,2)	M(6,3)	M(6,4)	M(6,5)	M(6,6)	M(6,7)	M(6,8)	M(6,9)	M(6,10)
M(7,1)	M(7,2)	M(7,3)	M(7,4)	M(7,5)	M(7,6)	M(7,7)	M(7,8)	M(7,9)	M(7,10)
M(8,1)	M(8,2)	M(8,3)	M(8,4)	M(8,5)	M(8,6)	M(8,7)	M(8,8)	M(8,9)	M(8,10)
M(9,1)	M(9,2)	M(9,3)	M(9,4)	M(9,5)	M(9,6)	M(9,7)	M(9,8)	M(9,9)	M(9,10)
M(10,1)	M(10,2)	M(10,3)	M(10,4)	M(10,5)	M(10,6)	M(10,7)	M(10,8)	M(10,9)	M(10,10)

- Exportez ensuite votre projet en WAR et le lancer sous Tomcat

2 : Réalisation d'un Servlet ECHO

Créez un nouveau **Dynamic Web Project**. N'oubliez pas de lier votre serveur Tomcat à votre projet Eclipse pour pouvoir l'exécuter.

Attention : Le serveur tomcat ne peut pas exécuter directement le projet depuis le workspace d'Eclipse car il n'est pas sous la forme d'un projet JEE correct. Eclipse crée donc un répertoire de déploiement (visible par la propriété `wtp.deploy`), et copie le contenu du workspace dans ce répertoire suivant les règles d'un projet JEE définies précédemment. Puis le serveur Tomcat lance le projet depuis le répertoire de déploiement.

Vous pouvez vérifier l'emplacement de ce dossier de déploiement en affichant la chaîne renvoyée par

```
System.getProperty("wtp.deploy")
```

Le but de cet exercice est de réaliser un Servlet ECHO qui construit une page HTML présentant les informations des requêtes qu'il reçoit. Le développement de ce Servlet se fera de façon incrémentale : il faut donc tester son fonctionnement étape après étape.

Servez-vous du code présenté en cours et surtout de l'API java :

<http://java.sun.com/javaee/5/docs/api/>

Note : la version 6 de JEE existe depuis peu. Nous ne l'utiliserons pas aujourd'hui.

2.1 : Traitement des requêtes GET

Toutes les requêtes que reçoit le conteneur web contiennent des informations sur le client (version de l'OS, du navigateur, des langues supportées, ...).

Après avoir analysé la javadoc de la classe `HttpServletRequest`, reprenez le code de la classe `ShowRequestHeader` présentée en cours, pour construire une réponse affichant toutes les informations que vous pouvez extraire d'une requête GET simple (sans paramètre).

Pour tester votre servlet, créez une page HTML `index.html` à la racine de votre application Web, qui comporte un lien vers votre servlet (balise ``).

2.2 : Traitement des paramètres des requêtes GET

Cette fois-ci, nous allons créer un formulaire, afin que le client envoie des données à notre serveur JEE. C'est notre servlet qui sera chargée de traiter ces données.

Toujours après avoir analysé la javadoc de la classe `HttpServletRequest`, reprenez le code de la classe `ShowParameters` présentée en cours, pour compléter la réponse de votre servlet afin d'afficher en plus les paramètres reçus via une requête **GET**.

Pour tester votre servlet, complétez votre page `index.html` à la racine de votre application Web, en ajoutant un formulaire qui utilise la méthode GET pour soumettre ses données à votre servlet.

2.3 : Traitement des requêtes POST

Complétez votre servlet pour traiter également les requêtes POST.

Pour tester votre servlet, complétez votre page `index.html` à la racine de votre application Web, en ajoutant un formulaire qui utilise la méthode POST pour soumettre ses données à votre servlet.

2.4 : Un tableau dynamique

On va reprendre ici l'exercice 1 en rendant la taille du tableau dynamique.

Créez un formulaire récupérant 2 entiers : **nbLig** et **nbCol** et les envoyant à notre Servlet.

La Servlet doit ensuite récupérer ces paramètres puis afficher un tableau de taille `nbLig x nbCol`.

3 : Compression de fichiers via un Servlet

Nous avons pour l'instant créé des formulaires envoyant des données textuelles. Qu'en est-il si l'on veut envoyer des données binaires (ex : un fichier)?

Dans cet exercice, on se propose de créer via un servlet l'archive Zip d'un fichier transmis par un formulaire. Le développement se fera par étapes successives :

3.1. Après avoir installé les librairies de « Common File Upload » dans votre projet (cf. cours), reprenez le code présenté en cours pour transmettre un fichier à un servlet en utilisant un formulaire.

3.2. Testez votre premier prototype avec un fichier de texte que vous afficherez dans la réponse du servlet (`fileItem.getString()`)

3.3. Retravaillez votre prototype pour sauvegarder maintenant le fichier transmis sur votre serveur. Par défaut, si l'on crée un fichier sur l'emplacement . , le fichier sera généré dans le workspace de l'environnement d'Eclipse. Pour récupérer l'URL de l'emplacement où votre serveur est déployé, vous pourrez utiliser `System.getProperty("wtp.deploy")`.

Attention : Le fichier est sauvegardé dans le répertoire de déploiement. Vous ne pourrez donc pas le voir dans le workspace d'Eclipse.

Attention : La propriété `wtp.deploy` n'existe que sous Eclipse. Il faut utiliser la méthode `request.getSession().getServletContext().getRealPath()` pour accéder au répertoire de déploiement depuis n'importe quelle application JEE (marche également pour un WAR déployé sur un serveur d'application).

3.4. Etudiez la javadoc de la classe `ZipOutputStream` :

<http://java.sun.com/javase/6/docs/api/java/util/zip/ZipOutputStream.html>

3.5. Puis complétez votre servlet pour créer une archive Zip du fichier reçu et renvoyer une page html comprenant un lien vers l'url de ce nouveau fichier. Testez.

4 : Redirection vers Google ou Yahoo

Analysez les paramètres utilisés par Google et Yahoo lors d'une recherche pour réaliser un servlet qui à partir d'un formulaire renvoie alternativement (ou aléatoirement) vers la réponse de l'un de ces deux moteurs de recherche.

Pour aller plus loin : Analysez l'entête de la requête du client afin d'effectuer la recherche dans la langue correspondante.