

EISTI 2008-2009 – TD/TP de Java EE – Séance 2

Introduction aux Servlets

0 : Finir l'installation de l'environnement (TP1) !

ATTENTION ! Pour réaliser la suite des TD/TP, il est impératif d'avoir un environnement qui fonctionne correctement !

1 : Réalisation d'un Servlet ECHO

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/>.

1.1 : Traitement des requêtes GET

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 ``).

1.1 : Traitement des paramètres des requêtes GET

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.

1.1 : Traitement des requêtes POST

Complétez votre servlet pour traiter également les requêtes POST sans et avec paramètres.

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 : Compression de fichiers via un Servlet

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 :

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,
2. testez votre premier prototype avec un fichier de texte que vous afficherez dans la réponse du servlet (`fileItem.getString()`),
3. retravaillez votre prototype pour sauvegarder maintenant le fichier transmis dans l'arborescence Tomcat de votre application Web (utilisez pour cela la propriété « `wtp.deploy` ») afin qu'il soit accessible via votre serveur Web Tomcat,
4. étudiez la javadoc de la classe `ZipOutputStream` :

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

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.

3 : 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.

De même, analysez la requête effectuée pour rechercher sur ces moteurs, afin de sélectionner la langue correspondant à celle utilisée par le client (analyse de l'entête des requêtes ou champ du formulaire).