

Les Servlets

Atelier1 :

Ecrire un servlet qui affiche la date courante et l'heure de l'exécution de la servlet. Effectuer cette tâche à l'aide de la classe Date et du package java.util.*, sans oublier de modifier le fichier web.xml.

Atelier2 :

Ecrire un servlet qui compte le nombre de fois qu'elle a été invoquée.

Atelier3 :

Nous allons utiliser un servlet pour traiter les données saisies via une page html :

- Ecrire une page html pour saisir le nom et le mot de passe de l'internaute.
- Ecrire un servlet qui affiche le nom et le mot de passe saisie auparavant (cette servlet est l'action à exécuter une fois le formulaire de saisie rempli).

Atelier4 :

Nous considérons dans cet exercice le traitement d'un paramètre (saisie) qui peut avoir de multiples valeurs. Cela est important par exemple pour le traitement des questions à choix multiples. Pour cela, vous devez :

- Ecrire une page html qui affiche une question avec plusieurs réponses possible. Par exemple 'Quel joueur préférez-vous ?' avec comme réponse possible 'Zidane', 'Maradona', 'Pelé', 'Platini'. L'utilisateur sélectionne une ou plusieurs réponses parmi les réponses possibles. Le servlet traite ensuite ses données.
- Ecrire un servlet qui affiche les réponses, éventuellement multiples, données à la question posée auparavant.

Atelier5 :

Maintenant, nous allons traiter le problème de traitement de la réponse par un tiers en considérant la redirection entre servlets. Nous considérons le cas où un utilisateur saisi son nom et son mot de passe via un formulaire de saisie. Ensuite, un servlet vérifie si le nom est 'mohamed' et le mot de passe est 'moi' et qui affiche un message d'erreur en cas d'échec ou redirige le traitement vers un autre servlet qui effectue les traitements nécessaires en cas de succès. Ecrire :

- Un formulaire pour saisir le nom et le mot de passe de l'internaute

- Un Servlet qui effectue la redirection en cas de succès vers un autre servlet et envoie le message d'erreur 'vous n'êtes pas un utilisateur valide' dans le cas contraire.
- Ecrire un servlet qui effectue le traitement en cas de succès en affichant le message Suivant : 'Bienvenue au cours Applications Web' et 'Bon travail'.

Atelier6 :

Cet atelier illustre le cycle de vie d'un servlet. Pour cela nous allons utiliser les méthodes init(), et doGet() et destroy(). Ecrire un servlet qui :

- Déclare une variable i de type entier.
- Ecrire la méthode init() pour initialiser la variable i à zéro.
- Ecrire la méthode doGet qui retourne un document de type 'text/html' et qui distingue les trois cas suivants :
 - Cas $i = 0$: dans ce cas on affiche le message 'la valeur de i est initialisée par la méthode init' en donnant la valeur de i. Ensuite, on incrémente i de 1.
 - Cas $i = 5$: afficher alors le message 'lorsque la valeur de i atteint 5 alors on appelle la méthode destroy pour l'initialiser à zéro'. afficher aussi la valeur de i, puis, appeler ensuite la méthode destroy().
 - Cas $i < 5$: afficher simplement le message 'la valeur de i est incrémentée dans doGet' suivi de la valeur courante de la variable i.
- Ecrire la méthode destroy() qui affecte la valeur zéro à la variable i.
- Modifier le document web.xml pour tenir compte de ce nouveau servlet, puis invoquer cet servlet via votre navigateur. Que se passe-t-il ?

Atelier7 :

Ecrire un servlet qui permet de vérifier si une session est nouvelle ou non ? Pour cela vous devez utiliser l'objet 'request' et sa méthode getSession(). Ensuite, vous utilisez la méthode isNew() de l'objet session pour savoir si la session est nouvelle ou non. Vous afficherez alors le message 'Vous venez de créer une nouvelle session' si la session est nouvelle et 'La session existe déjà' dans le cas contraire.