TD Servlets – Variables sessions

Gestion d'une authentification

Objectifs :

Créer une variable session.

Modifier le schéma relationnel pour étendre un SI à un accès intranet/extranet.

Enoncé :

Le but de ce td est de créer une variable session et de s'en servir dans le contexte du problème d'authentification lors d'un accès internet au SI d'une entreprise.

Tout client, lors de sa connexion initiale à une application web de tomcat, ouvre une session http enregistrée sur tomcat. Cette session est valide jusqu'à ce que le client soit déconnecté par le serveur (suite à dépassement de temps d'inactivité ou d'autres raisons) ou se déconnecte lui-même (en général via un lien que les pages web lui proposent).

Une variable session est un objet Java que l'on nomme et que l'on associe (« binds » sur la doc) à la session du client. C'est donc un couple (nom, valeur). La durée de vie d'une variable session est celle de la session. Une session peut gérer autant de variables sessions que vous le souhaitez.

L'interface javax.servlet.http.HttpSession représente la session (http://docs.oracle.com/javaee/6/api/).

La syntaxe

HttpSession session = request.getSession(true) ;

dans un doGet/doPost vous donne accès à l'objet session du client pour le compte duquel s'exécute la servlet (le paramètre *true* permet d'en créer une la première fois)

Les méthodes de HttpSession

Object getAttribute(String)

void setAttribute(String, Object)

permettent respectivement d'obtenir ou de créer une variable session.

Exercice 1 : Créer une variable session user

Vous allez écrire une servlet AuthentificationServlet recevant en paramètre de la requête un login et un password, créant un objet ***user*** de la classe *data.User* (une simple encapsulation de ces 2 données avec un constructeur et un affichage via toString())., et envoyant en réponse une page html confirmant l'enregistrement de la variable session et proposant de vérifier via un HREF. Ce HREF devra renvoyer, si l'on clique dessus, à une VerificationServlet se contentant de récupérer la variable session et de l'afficher en guise de réponse.

La classe User se trouve dans un package à part (data) et l'accès à la BD n'est pas nécessaire pour ce premier exercice

Exercice 2 : faire une véritable authentification

On veut maintenant rendre opérationnelle l'authentification. Il faut donc ajouter au schéma relationnel de la base Oracle une table dans laquelle on enregistrera via un client sql quelques login/passwd connus.

Cette table, aura pour clé primaire login (on ne veut pas avoir 2 logins identiques), un champ simple passwd, et 2 clés étrangères matricule et numpassager dont l'une des 2 sera null. On pourra ainsi facilement et sans ambiguité savoir, à partir d'un couple login/passwd, qui s'est authentifié.

Ensuite, vous rajouterez la requête SQL et le code jdbc dans AuthentificationServlet afin qu'elle vérifie l'existence du couple login/passwd saisi en plus de créer la variable session.

Exercice 3 : un message de bienvenu personnalisé

Vous ferez en sorte que la réponse à l'authentification réussie soit personnalisée avec le nom et le type d'utilisateur (« Ouverture de session de Amandier, pilote. » par exemple)