UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Licence 3 INFO
INFO0503

TP n°4

Communications HTTP

Le but de ce TP est de mettre en place des communications HTTP entre applications *Java* et/ou PHP. Pour la programmation *Java*, nous utiliserons les classes `HttpServer` et `URLConnection`. En PHP, nous utiliserons la fonction `get_file_contents`.

1 Premier serveur HTTP en Java

Dans un premier temps, il est conseillé de reprendre les codes des fiches *Serveur HTTP en Java* et *Communications HTTP en Java*, de les compiler et les exécuter. Étudiez ensuite le comportement des programmes *Java*.

Nous allons réaliser un serveur HTTP constitué de deux pages (*ie* deux *handlers*) : `index.html` affiche un formulaire demandant à l'utilisateur de saisir son login et son mot de passe, `login.html` vérifie que le login et le mot de passe sont corrects. La liste des logins et des mots de passe est contenue dans un fichier JSON qui sera géré par le gestionnaire d'utilisateurs.

Questions

1. Écrivez la classe `GestionnaireUtilisateurs` qui permet de gérer les utilisateurs. Le constructeur prend en paramètre un nom de fichier dans lequel sera lu (et sauvegardé) les login/mot de passe.
2. Écrivez la classe `ServeurHttp` qui crée le serveur HTTP et qui associe l'URL `index.html` à un *handler* `IndexHandler`.
3. Écrivez la classe `IndexHandler`. Le but de ce *handler* est d'afficher le formulaire de connexion qui envoie vers la page `login.html` (qui correspondra à un autre *handler*). Vérifiez qu'il est fonctionnel à l'aide d'un navigateur.
4. Quand le gestionnaire d'utilisateurs doit-il être instancié ? Où sa référence doit-elle être placée ?
5. Écrivez la classe `LoginHandler`. Il doit récupérer les paramètres envoyés par le formulaire (le login et le mot de passe). Un message d'erreur doit être affiché si les login et mot de passe sont incorrects, ainsi qu'un lien permettant de revenir au formulaire. En cas de réussite, un message de bienvenue est affiché.
6. Si on souhaite compléter le site, nous allons créer d'autres *handlers*. Comment s'assurer que seul un utilisateur connecté peut y accéder ?

2 Échanges HTTP entre des programmes en Java

Le traitement des données envoyées en POST, ainsi que l'affichage du HTML directement au sein des classes des *handlers* peut rendre le code *Java* très vite illisible. En INFO0503, nous allons principalement utiliser le serveur HTTP *Java* pour échanger du contenu JSON. Les données envoyées au serveur seront au format JSON, tout comme les réponses de celui-ci.

Nous allons maintenant réaliser un client *Java* qui communiquera avec le serveur HTTP. Le client *Java* envoie le login et le mot de passe au format JSON au serveur HTTP.

Questions

1. Le *handler* `IndexHandler` est-il encore nécessaire ?
2. Modifiez la classe `LoginHandler` pour traiter en entrée du JSON et pour envoyer du JSON en lieu et place du HTML.
3. Écrivez le client *Java* : le login et le mot de passe seront saisis au clavier par l'utilisateur.

3 Échanges HTTP entre des programmes en PHP et en *Java*

Dans un premier temps, il est conseillé de reprendre les codes de la fiche *Requête HTTP depuis un script PHP* et les exécuter. Étudiez ensuite le comportement des scripts *PHP*.

Nous allons maintenant réaliser un script PHP qui communiquera avec le serveur HTTP en *Java* (c'est-à-dire que le script PHP sera le client HTTP). Un utilisateur saisit son login et son mot de passe dans un formulaire HTML. La validité du login et du mot de passe est vérifiée sur le serveur *Java*.

Questions

1. Qu'est-ce que cela change sur le serveur *Java* ?
2. Écrivez le script PHP : le login et le mot de passe seront saisis dans un formulaire.