Protocole HTTP

Exercice 1 - Un simple serveur de fichier HTTP

Écrire un petit programme Java qui permet d'émuler un serveur HTTP très simple. Voici les fonctionnalités qui doivent être implantées par ce petit serveur:

- répondre aux requêtes `GET` des clients sur des noms de fichiers réguliers (pas les répertoires);
- attribuer le type MIME `application/octet-stream` aux ressources retournées. On pourra également faire des essais, plus visuels, avec les types `text/plain` ou `text/html`. Ensuite, on pourra tenter de déterminer le `Content-Type` à partir de l'extension du fichier à renvoyer;
- pour simplifier la gestion de la connexion avec le client, on pourra fermer la connexion après l'envoi de la ressource. En revanche, il est nécessaire de lire l'intégralité de la requête du client avant de fermer la connexion.

Exercice 2 - Les messages d'erreur

Ajouter au serveur précédent la fonctionalité suivante: lorsqu'une ressource recherchée par un client n'est pas un fichier régulier, et que le serveur n'est pas en mesure de la retourner, renvoyer un message d'erreur `404 Not Found` accompagné d'une page HTML expliquant poliment que la ressource n'est pas accessible.

Exercice 3 - Échange d'information grâce aux formulaires

En plus des services précédents, on veut maintenant que le serveur permette d'accéder à des ressources qui ne sont pas statiques, mais créées dynamiquement à partir des informations des clients.
Pour mettre en oeuvre ces fonctionnalités, on pourra écrire un fichier `suscribe.html`, accessible par les clients et leur offrant un formulaire comme celui-ci (adaptez le à votre goût si vous le souhaitez, le but étant d'utiliser quelques balises comme `form`, `input`, `select`, etc.) :
Les données recueillies par ce formulaire seront, dans un premier temps, envoyées avec la méthode `GET` à une URL choisie, par exemple `/suscribe`, relative au serveur. Lorsque ces données sont reçues par le serveur, elles doivent être stockées.
Les manières de stocker ces informations peuvent êtres diverses et variées, mais on pourra par exemple créer une instance d'une classe `Membre`, définie de sorte à pouvoir stocker les informations relatives à une inscription, et stocker cette instance dans un vecteur.
En revanche, il est important de rendre la gestion de ce service le plus modulaire possible (éviter de le coder en dur "`if then else`" dans le serveur), afin de pouvoir fournir d'autre services différents en utilisant le même serveur.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final //EN">
<html>
<head>
  <title>Formulaire d'inscription</title>
</head>
```

```
       <body bgcolor="white">
        <form enctype="application/x-www-form-urlencoded"
              action="suscribe" method=GET>
          Nom :
          <input type=text size=20 name=username>
          Prénom :
          <input type=text size=20 name=firstname>
         <br>
          Niveau de sécurité :
          faible
          <input type=radio name=level value="low">
          normal
                <input type=radio name=level value="normal"
checked>
                élevé
                <input type=radio name=level value="high">
               <br>
                Fréquentation du service :
                <select name=usage>
                  <option value="day">quotidienne
                  <option value="week">hebdomadaire
                  <option value="month">mensuelle
                  <option value="year">annuelle
                </select>
               <hr>
                <input type=submit value="Envoi">
                <input type=reset value="Réinitialiser">
              </form>
            </body>
          </html>
```

Exercice 4 - Extensions de services

À partir de ce serveur et des informations relatives à ce formulaire, on peut imaginer différentes extensions telles que.

- Renvoyer une confirmation d'inscription.
- Proposer la génération dynamique, correspondant à un URL offert aux clients, de l'ensemble des inscrits.
- Rendre la configuration du serveur et/ou les informations recueillies persistantes.
- Affecter un mot de passe aux clients, et l'exiger pour la consultation de certaines informations.