

Librairies requises (pour l'upload), à ajouter au \$CLASSPATH :

- Fileupload : <http://commons.apache.org/proper/commons-fileupload/>
- IO : <http://commons.apache.org/proper/commons-io/>
- SQLibrary : joint avec le contexte 'projet8'

### **Page de login ('Login') :**

Si le couple utilisateur/mot de passe n'existe pas dans la BDD, affichage d'un message d'erreur et ré-affichage du formulaire de login.

Si l'utilisateur a laissé des champs vides dans le formulaire, affichage d'un message d'erreur et ré-affichage du formulaire de login..

Si le login et le mot de passe sont corrects, on envoie une redirection au client vers la page « Files ».

### **Page d'upload ('Files') :**

Si l'utilisateur essaye d'accéder à la page sans s'être connecté avant, affichage d'un message d'erreur.

Si le répertoire correspondant à l'utilisateur n'existe pas, on le crée.

Si la requête contient un fichier uploadé, on traite la récupération et la sauvegarde du fichier dans le répertoire de l'utilisateur.

Affichage de l'arborescence actuelle de l'utilisateur, et affichage du formulaire d'upload.

### **Librairie FileUpload :**

La librairie FileUpload d'Apache permet de simplifier grandement l'envoi de fichier depuis une page Web.

Elle permet de récupérer les fichiers envoyés, et de les sauvegarder sur le disque, ou de les garder en mémoire (pour les fichiers légers). La taille max. de sauvegarde en mémoire est configurable, ainsi que l'emplacement du dossier stockant les fichiers temporaires.

La librairie permet également de pleinement configurer les caractéristiques des fichiers que l'on veut accepter ou refuser : type, taille, extension, ... via l'interface FileItem :

```
List<FileItem> items = upload.parseRequest(req);

// Process the uploaded items
Iterator<FileItem> iter = items.iterator();
while (iter.hasNext())
{
    FileItem item = iter.next();

    if (! item.isFormField())
    {
        String fieldName = item.getFieldName();
        String fileName = item.getName();
        String contentType = item.getContentType();
```

```
    boolean isInMemory = item.isInMemory();  
    long sizeInBytes = item.getSize();  
    // [...]  
}  
}
```

On peut ainsi filtrer les fichiers à accepter/refuser selon leurs caractéristiques, simplement en testant les valeurs ci-dessus.