

Objectifs

Réaliser un site web simple permettant à chacun de s'identifier puis de déposer ou récupérer un fichier

1 Description

- Ce site est constitué uniquement de deux pages, l'une permettant de s'identifier et l'autre affichant d'une part la liste des fichiers disponibles ainsi qu'un formulaire pour en placer un nouveau.
- A priori la base de données ne contient qu'une seule table : `personne(nom, login, mdp)` permettant d'authentifier un utilisateur.
- Afin de faciliter le rangement et d'éviter les conflits de noms, on choisit de créer dans le contexte web, un répertoire particulier pour chaque utilisateur, dans lequel on rangera les fichiers déposés

```
webapps
|-- projet
|   |-- index.html
|   `-- users
|       |-- identifiant1
|       |   |-- file1
|       |   |-- file2
|       |-- identifiant2
|       |   |-- file1
|       |   |-- file2
|       `-- identifiant3
|           |-- file1
|           |-- file2
```

- Dans un premier temps les personnes sont insérées dans la base à la main et les répertoires correspondants sont aussi créés à la main.
- Pour faciliter la gestion de l'upload, on pourra utiliser l'API [FileUpload](#) fournie par la fondation Apache. Néanmoins depuis Servlet 3.0 il est possible d'uploader un fichier avec les méthodes natives (`ServletFileUpload`)
- Pour récupérer la liste des fichiers disponibles, on utilisera la classe `File` et sa méthode `list()` qui retourne l'ensemble des fichiers d'un répertoire. On rappelle que le chemin d'accès système au contexte est obtenu par `getServletContext().getRealPath()`

2 Pour aller plus loin

Améliorez le projet en permettant à un nouvel utilisateur de se créer un compte. Si `id` est le nom de la variable contenant l'identifiant utilisateur, le code suivant permet de créer le répertoire système nécessaire lors de la création d'un nouvel utilisateur.

```
Runtime.getRuntime().exec("mkdir "+id, null, new
File(getServletContext().getRealPath("/"))).waitFor();
```