

Mini Projet PHP

Système de Folksonomies

Avec le Web 2.0, la participation de chaque internaute est sollicitée. Un exemple de cette participation peut être identifiée au travers des outils de Folksonomies (classement de documents par association de mots-clés ou de tags) tels que *Delicious* (<https://delicious.com>).

Cet outil permet de sauvegarder des pages Web et de les associer à des mots-clés (tags) qui les caractérisent. Ces mots-clés permettent de retrouver plus facilement les documents sauvegardés en naviguant. On peut considérer cette classification comme étant une évolution des signets (bookmarks) des navigateurs. Le grand intérêt de ces outils est qu'ils sont multi-utilisateurs.

Le but de ce mini-projet est de construire un mini-outil de Folksonomies. Ce projet sera implanté en PHP avec un support MySQL. Les feuilles de style CSS doivent également être utilisées à bon escient.

Vous devez lors de l'implantation de ce projet (et bien au-delà) respecter les règles suivantes :

- Tout comme le projet collectif de SIAWeb prévoir de pouvoir exporter votre site Web (chemins relatifs, répertoire complet...). Il est important que tous les éléments de votre site soient dans un même répertoire
- Un document de présentation en PDF (contenu identique au projet collectif)

***La date limite de remise des projets vous sera donnée par les encadrants de TP.
Aucun retard ne sera admis et l'évaluation se fera sur la base des documents rendus.***

L'évaluation concernera le document PDF ainsi que le site rendu (qui sera testé sur une autre machine). *Nous vous conseillons donc de tester "l'exportation" de votre site avant la date limite.*

0) Préparation (idem TP SIAWeb)

Créer un fichier contenant les lignes suivantes dans lesquelles vous remplacerez les <valeur> par les valeurs réelles.

```
<?
$serveur="<valeur>";      // Nom du serveur mysql
$user="<valeur>";        // Utilisateur ayant les droits
$pass="<valeur>";        // Mot de passe
$base="<valeur>";        // Nom de la base (existante) où seront créées
les tables
?>
```

Ce fichier sera (inclus) dans les différents scripts PHP nécessitant un accès à la base de données.

Pour inclure un fichier dans un autre utiliser l'instruction :

```
require("<fichierAInclure">);           // ex : require ("infos.php");
```

1) Création des relations

Créer un fichier "**creabase.php**" contenant les instructions permettant de construire les relations (tables) nécessaires au projet. Celles-ci seront déduites notamment du dictionnaire des données et du MCD fournis en *annexe 1*.

2) Création de l'index

Créer un fichier "**index.html**" contenant un lien vers la page de création des relations. Chaque lien vers les pages à réaliser (voir sections suivantes) sera ajouté à cette page.

3) Saisie des documents

Créer un fichier "**saisie.html**" contenant un formulaire permettant de saisir un **document** ainsi que toutes les informations qui le caractérisent.

==> Piste de résolution : la saisie des mots-clés peut être faite au travers d'un champs texte unique. Ces derniers seront séparés par des ';' (ex de 2 mots-clés : *IRIT ; Laboratoire de Recherche*). Le découpage de la chaîne se fera au niveau du serveur (cf. instruction **explode()** du langage PHP). Veuillez également à supprimer les espaces superflus éventuels présents autour de chaque mot-clé (cf. instruction **trim()** du langage PHP).

Ajouter un document "**ajouterDocument.php**" permettant de traiter le résultat du formulaire en sauvegardant les informations dans la base de données. **Attention aux doublons dans les URL et les mots-clés.** Veuillez donc à fournir des algorithmes corrects pour que le script PHP vérifie correctement ces contraintes avant d'insérer les informations dans la base, c'est-à-dire :

- ❶ Il ne doit pas y avoir de doublons dans les URL des différents tuples de la relation **Document**.
- ❶ Il ne doit pas y avoir de doublons dans le champ **motCle** des différents tuples de la relation **Terme**.

==> Pour simplifier, on utilisera les informations déjà existantes dans la base pour les compléter par exemple. Ainsi par exemple, si l'url existe déjà on se limitera à ajouter les nouveaux mots-clés au document (c'est-à-dire ceux qui n'ont pas encore été associés au document).

Par ailleurs, plutôt que d'utiliser un entier **auto-incremental (AUTO_INCREMENT)** comme identifiant (notamment pour les documents et pour les termes), le script PHP peut générer automatiquement cet identifiant. Pour cela vous pouvez utiliser une instruction du style :

```
$id = uniqid(rand(), true);
```

4) Listing des documents

Proposer un fichier PHP permettant de lister les documents par **ordre croissant de titre**. Chaque document sera illustré par ses informations ainsi que par les termes/tags qui lui sont associés.

Un clic sur le titre du document permet d'ouvrir ce document dans le navigateur (via son URL).

5) Nuage de mots-clés

Proposer un programme PHP permettant de lister les mots-clés par ordre croissant. Chaque terme sera écrit avec une taille correspondant à son usage dans l'outil. Plus un terme sera associé à des documents, plus la taille du texte sera importante (voir l'exemple ci-dessous).

La taille du texte correspondant à un mot-clé peut être calculé selon la méthode suivante :

$$\text{taille}_t = 20 + (\text{nbDocs}_t / \text{nbDocs}) * 100 ;$$

Où nbDocs_t correspond au nombre de documents de la base associé au mot-clé t ,
 nbDocs correspond au nombre de documents de la base

IRIT Laboratoire de Recherche.PHP

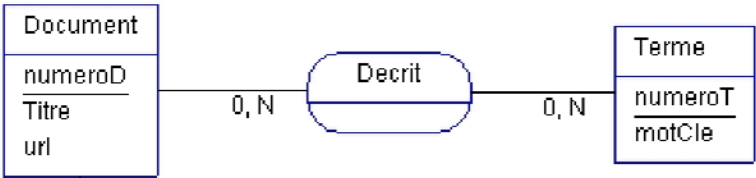
Les fonctions précédentes correspondent au minimum attendu dans ce projet. Les éléments indiqués comme piste de résolution ne sont que des propositions et nous vous encourageons vivement à proposer VOTRE solution. Il vous est possible également d'ajouter de nouvelles fonctionnalités si cela vous plait (cf ci-dessous). Ces nouvelles fonctionnalités vous rapporteront des points bonus si elles sont pertinentes et bien implantées. Vous n'obtiendrez aucun point bonus si les 5 étapes présentées ci-dessus n'ont pas été totalement réalisées.

6) Evolutions possibles (5 pts points bonus maximum pour les évolutions)

Faire évoluer le système pour prendre en compte les éléments suivants :

- Ajouter lors du listing, les mots-clés associés à chaque document. Un clic sur un mot-clé permet d'obtenir (au travers d'un nouveau script PHP) le détail de tous les documents ayant été associés à ce mot-clé.
- Insérer un champ de recherche de documents basée sur les mots-clés dans la page d'accueil.
- Transformer l'application pour que désormais elle gère plusieurs utilisateurs. Chaque utilisateur pourra s'authentifier avec son nom uniquement. Chaque usager pourra ajouter une description textuelle personnelle à chaque document qu'il souhaite sauvegarder. Les mots-clés seront associés aux documents par chaque usager.
- Ajouter lors du listing des documents les utilisateurs ayant enregistré un lien vers ce document (même URL). Un clic sur l'utilisateur permet de voir tous les documents de cet utilisateur.
- ...

Nom	ID	Type	Taille
numeroD	numeroD	varchar	150
Titre	titre	varchar	255
url	url	varchar	255
numeroT	numeroT	varchar	150
motCle	motcle	varchar	255



ANNEXE 1