Module S2301- Semaine bloquée Programmation Web - Projet "Site de veille d'information personnalisé"

Partie 1 : Collecte dynamique sur le web et stockage des informations dans une base de données

1. Tâche

L'objet de cette partie est la construction de la base de données et son remplissage à partir des informations collectées sur différents sites web via leurs flux RSS. En vue de simplifier la gestion de la base de données nous utiliserons le SGBD ***SQLite3*** qui est intégré à PHP et qui ne nécessite pas l'installation d'un serveur de base de données. Avant de vous lancer dans la création des bases et leur remplissage, nous vous invitons dans la partie suivante, à comprendre ce qu'est un flux RSS (encore appelé fil RSS).

2. Flux RSS

De nombreux sites d'actualité (mais pas seulement) mettent à disposition des internautes un fichier XML appelé flux RSS. Ce fichier correspond à une suite d'informations comprenant chacune un titre, une description brève, un lien sur une page web et éventuellement des images. Ce fichier est régulièrement mis à jour en fonction de l'actualité. Pour comprendre la structure du fichier xml, commencez par charger dans votre navigateur l'url suivante :
http://www.lemonde.fr/rss/une.xml
Faites afficher le code source pour comprendre comment le fichier est structuré.
Pour en savoir plus, lisez la page wikipedia *http://fr.wikipedia.org/wiki/RSS*.
Des vidéos explicatives peuvent aussi être consultées sur :
http://www.journaldunet.com/solutions/0410/041029_faq_rss.shtml
http://www.koreus.com/video/rss-explication.html

3. Création de la base de données avec SQLite3

3.1 SQLite3

Le site officiel de SQLite est : http://www.sqlite.org/. SQLite3 est intégré à PHP, il n'y a aucune installation à faire. Une base de donnée SQLite est un simple fichier, car il n'y a aucun serveur, contrairement à Postgres. Pour construire vos requêtes à la base de donnée à partir de vos scripts PHP, vous n'aurez pas à connaître les fonctions spécifiques d'accès aux bases SQLite3 puisque vous utiliserez PDO.

3.2 Création des tables

Le schéma relationnel de la base de donnée utilisée est le suivant :

```
CREATE TABLE flux (
url varchar(255) primary key
);
CREATE TABLE nouvelles (
id integer primary key autoincrement,
date varchar(80),
titre varchar(255),
description varchar(1024),
lien varchar(255),
image varchar(80),
flux varchar(255)
);
CREATE TABLE utilisateurs (
login varchar(80) primary key,
mp varchar(80)
);

CREATE TABLE flux_utilisateur (
login varchar(80) primary key,
flux varchar(255),
```

```
nom varchar(80),
categorie varchar(80)
);
```

Il y a donc 4 tables :

1. Une table ***flux*** stockant l'url de tous les flux rss à partir desquels nous récupérerons des nouvelles.
2. Une table ***nouvelles*** contenant toutes les nouvelles extraites des différents flux. Une nouvelle a un identifiant ***id*** (numéro qui s'incrémente automatiquement à chaque fois qu'une nouvelle est ajoutée dans la base), une ***date*** (date de récupération de la nouvelle) et les champs ***titre***, ***description*** et ***lien*** du flux rss. Une nouvelle est aussi associée à une ***image*** récupérée en local dans le répertoire *images*. Le champ image devra contenir le nom du fichier correspondant. Une nouvelle est aussi associée au ***flux*** rss dont elle provient.
3. Une table ***utilisateurs*** stockant le ***login*** et le mot de passe ***mp*** des différents utilisateurs (uniquement utile dans la partie 3 du projet).
4. Une table ***flux_utilisateur*** permettant de stocker les ***nom***s et ***catégorie***s définis par l'utilisateur et associé à chaque ***flux***. (uniquement utile dans la partie 3 du projet).

Pour créer la base, tapez la commande ***sqlite3*** (et pas *sqlite*) dans un terminal. La commande *.help* permet de visualiser toutes les commandes disponibles. En ce qui nous concerne, les instructions suivantes suffisent à créer la base :
`sqlite>.read create.sql` (création des tables à partir de create.sql)
`sqlite>.backup main newsDB` (sauvegarde dans le fichier newsDB)
`sqlite>.exit`
On peut aussi taper la commande ***sqlite3 newsDB*** dans un terminal puis, les instructions suivantes:
`sqlite>.read create.sql`
`sqlite>.exit`

Pour interroger la base, tapez la commande ***sqlite3*** dans un terminal. Puis
`sqlite>.restore main newsDB` (chargement de la base newsDB)
`sqlite>select * from nouvelles;` (requête sql)
On peut aussi taper la commande ***sqlite3 newsDB*** dans un terminal puis directement :
`sqlite>select * from nouvelles;`

Pour ceux qui travailleraient sous windows, un interpréteur de commandes sqlite3 est disponible ici : https://www.sqlite.org/download.html, (choisir sqlite-tools-win32-x86-3350300.zip) dont les commandes sont résumées ici : https://www.sqlite.org/cli.html.

4. Remplissage de la base de données

4.1 Script de collecte des informations

Le script ***collecte.php*** vous montre comment il est possible de collecter le contenu de différents flux rss en s'appuyant sur la fonction `simplexml_load_file` de PHP. Il choisit au hasard un flux dans une liste puis télécharge et affiche les nouvelles du flux choisi. En décommentant les dernières lignes du script, en plus d'afficher le titre des nouvelles, il télécharge les images associées (s'il y en a) et les range dans le répertoire images (qui doit donc exister et être ouvert en écriture). Lancez-le en ligne de commande (***php collecte.php***) ou via le serveur web (http://www-etu-info.iut2.upmf-grenoble.fr/~votrelogin/projet-s2301/collecte.php) et vérifiez le résultat.

4.3 Script de collecte des informations et stockage dans la base de données

Le fichier XML correspondant au flux RSS change régulièrement. Il est donc nécessaire de sauver l'information dans une base de données, si l'on souhaite pouvoir continuer à accéder à des informations quand celles-ci auront disparu du flux. En vous appuyant sur le code de ***collecte.php***, créer le script ***collecteBD.php*** de façon à remplir les tables *flux* et *nouvelles* de la base de données ***newsDB***. Les flux seront ajoutés à la table des flux s'ils ne sont pas déjà présents. Les nouvelles seront ajoutées à la table des nouvelles si une nouvelle ayant le même titre et la même description n'existe pas déjà. Pour la connexion à la base de données on utilisera PDO. Pas besoin de s'authentifier auprès du serveur, il n'y en a pas. La connexion à une base sqlite3 s'écrit simplement :

```
$dsn = 'sqlite:newsDB'; // Data source name
try {$dbh = new PDO($dsn);
} catch (PDOException $e) { die ("Erreur : ".$e->getMessage());
}
```

Par ailleurs, attention, quand vous rangez les titres ou descriptions RSS dans la base de données, ceux-ci peuvent contenir des apostrophes. Utilisez la procédure *quote* de PDO pour échapper les chaînes de caractères. On écrira, par exemple : `$titre=dbh->quote($titre);`

Enfin pour générer un numéro d'image vous vous appuierez sur le nombre de nouvelles dans la base au moment de la récupération de la nouvelle. Pour récupérer ce nombre il vous suffit de compter le nombre d'éléments (fonction `count` en PHP) du tableau contenant les nouvelles renvoyées par la commande `select * from nouvelles;`