Ingénierie du Web (INF347), TELECOM ParisTech

TP PHP

Pierre Senellart (pierre.senellart@telecom-paristech.fr)

29 mai 2012

Le but de ce TP est de se familiariser avec la réalisation de sites Web simples en PHP (de base tout d'abord, puis avec le framework CodeIgniter). On va réaliser avec les fonctions de base de PHP un système très élémentaire de forum (avec un seul thread), ajouter une gestion d'authentification, puis réécrire ces mêmes fonctionnalités en utilisant le framework CodeIgniter.

Rendu

Le TP doit être rendu individuellement dès la fin du TP (aucune extension accordée, le TP doit être réalisée dans les trois heures de la séance de TP). Envoyer par email à `pierre.senellart@telecom-paristech.fr` en indiquant `[INF347-PHP]` dans le sujet :

- une archive ZIP contenant l'ensemble des fichiers (PHP, CSS, etc.) du site
- l'URL de la page principale de votre(s) site(s)
- l'identifiant MySQL attribué
- votre nom

Environnement

Vous placerez le site Web produit dans le sous-répertoire `public_html` de votre répertoire Unix principal. Si votre login Unix est `pierre`, vous pourrez accéder à une page `index.php` via un serveur Web à l'URL : `http://www.infres.enst.fr/~pierre/index.php`. S'assurer que le serveur Web a bien un droit de lecture sur votre répertoire principale et le sous-répertoire `public_html` à l'aide de la commande :

```
chmod a+rx ~ ~/public_html
```

Il peut être nécessaire de réutiliser cette commande sur chaque nouveau fichier, en fonction de la configuration.

On utilisera pour ce TP le serveur de bases de données MySQL/MariaDB accessible avec les paramètres suivants :

Serveur : mysql.tp.enst.fr

Port : 3307 (parfois donné après le serveur, séparé d'un ":")

Identifant : de la forme toto*n* (toto1, toto2, ..., toto60), tel qu'attribué pendant le TP

Mot de passe : identique à l'identifiant

Base : identique à l'identifiant

Ces paramètres peuvent être utilisés pour se connecter au serveur de bases de données :

en ligne de commande avec le programme de ligne de commande `mysql` qui s'utilise comme suit :

```
mysql -h serveur -P port -u identifiant -p base
```

avec l'outil phpMyAdmin depuis `http://www.infres.enst.fr/packages/phpMyAdminTp/`

depuis PHP à l'aide de la fonction `mysql_connect`

depuis le framework CodeIgniter en éditant le fichier `application/config/database.php`

On pourra (et devra) se referer tout au long de ce TP à la documentation en ligne de MySQL (`http://dev.mysql.com/doc/`), de PHP (`http://www.php.net/`) et de CodeIgniter (`http://codeigniter.com/user_guide/`).

MySQL et PHP

L'accès à une base de données MySQL depuis PHP utilise les fonctions suivantes :

`mysql_connect(`*`$serveur,$login,$mdp`*`)`	établit une connexion au serveur
`mysql_select_db(`*`$nom_base`*`)`	sélectionne une base de données
`mysql_query(`*`$ordre_sql`*`)`	exécute un ordre SQL
`mysql_fetch_array($resultat)`	retourne une liste de résultat, sous forme de tableau indicé
`mysql_fetch_assoc($resultat)`	retourne une liste de résultat, sous forme de tableau associatif
`mysql_error()`	renvoie le dernier message d'erreur

Une utilisation typique est ainsi :

```
if(!mysql_connect("mysql.infres.enst.fr:3307","login","password")) {
  echo "<p>Desolé, connexion impossible</p>"; exit;
}
if(!mysql_select_db("database")) {
  echo "<p>Desolé, accès à la base impossible</p>"; exit;
}
mysql_query("SET NAMES utf8"); /* Pour l'encodage */
$resultat= mysql_query("SELECT * FROM Personne");
if($resultat) {
  while($ligne=mysql_fetch_assoc($resultat)) {
    echo "<p>".$ligne["prenom"]." a ". $ligne["age"]." ans</p>";
  }
} else {
  echo "<p>Erreur dans l'exécution de la requête.</p>";
  echo "<p>Message de MySQL: ".mysql_error()."</p>";
}
```

1 Forum élémentaire

On veut créer un forum élémentaire : il y a une seule page contenant les messages du forum (un seul *thread*) et chaque message comporte un texte, une référence à un utilisateur, et une estampille temporelle (*timestamp*) du moment où ce message a été posté. Un utilisateur a un identifiant, un mot de passe, une adresse e-mail et, de manière optionnelle, une URL pointant vers une page Web personnelle.

1. Imaginer un schéma de base de données relationnelle adapté à cette application et créer les tables correspondantes dans l'outil en ligne de commande ou l'interface phpMyAdmin.
2. Peupler cette bases de données avec quelques utilisateurs et messages d'exemple.
3. Créer une page PHP `forum.php` listant l'ensemble des messages, triés par date (le plus récent d'abord), avec utilisateur et date. Ne pas se préoccuper de la mise en forme pour le moment.

4. Faire en sorte que les noms des utilisateur associés à un message soient des liens pointant vers la page Web de cet utilisateur, si celle-ci existe.
5. Ajouter au début de la page PHP un formulaire permettant d'ajouter un nouveau message (pour l'instant, fixer l'utilisateur postant ce message de manière arbitraire). Ce formulaire a comme action `forum.php` qui doit donc se comporter différemment selon qu'il a reçu ou non des paramètres HTTP demandant l'ajout d'un message.
6. Ajouter une feuille de style CSS permettant de mettre en forme la page d'affichage des commentaires : mettre en valeur l'utilisateur postant ce commentaire, séparer clairement chaque message du suivant, etc.
7. Valider la page HTML produites par le script PHP et la feuille CSS, à l'aide des validateurs du W3C. Corriger les erreurs éventuelles.

2 Authentification

1. Ajouter une page `form_login.html` avec un formulaire demandant login et mot de passe. L'action de ce formulaire pointe vers un script `login.php`.
2. Écrire le script `login.php`. Si les identifiants et mots de passe sont correcte, on stocke le nom de l'utilisateur dans une variable de session et on redirige (avec `header("Location: forum.php");`) vers la page du forum sinon, on redirige vers le formulaire d'identification.
3. Modifier `forum.php` pour contrôler qu'on est bien connecté ; sinon, on redirige vers le formulaire d'identification. Faire en sorte que l'utilisateur postant le message soit bien l'utilisateur connecté.
4. Ajouter une fonctionnalité de déconnexion.

3 CodeIgniter, un framework PHP

1. Télécharger le framework PHP CodeIgniter dans sa dernière version depuis `http://codeigniter.com/downloads/`
2. Suivre les instructions de `http://codeigniter.com/user_guide/installation/index.html` pour installer ce framework dans un sous-répertoire de votre répertoire `public_html` (uniquement les 4 points numérotés, inutile de suivre les instructions sur les dossiers `application` et `system`).
3. Suivre le tutoriel du framework CodeIgniter pour apprendre à se servir de ce framework : `http://codeigniter.com/user_guide/tutorial/index.html`
4. Réimplémenter le site de forum développé dans la première partie de ce TP avec le framework CodeIgniter (se concentrer sur les fonctionnalités de base d'affichage des messages et de création d'un nouveau message, inutile de gérer l'authentification).

4 S'il vous reste du temps...

1. Modifier l'application pour ajouter un système de threads : un utilisateur peut créer des threads, et les commentaires sont associés à un thread donné, chaque thread étant affiché sur une page séparée.
2. Ajouter la possibilité pour un utilisateur de s'inscrire, avec un système de vérification de l'email par envoi d'un email.
3. Ajouter la possibilité de joindre un fichier à un message.
4. Ajouter une interface d'administration, accessible à un utilisateur unique, pour supprimer des messages.