IUT de Montpellier - Semaine Web - TD3
Introduction à PHP - Connection à MySQL

Semaine du 4 Novembre 2013

1 Connexion à la base de données : insertion des tuples

Connectez vous à votre base de données MySQL, à l'aide de l'interface PhpMyAdmin `http://infolimon/my` et changez votre mot de passe puis créez une table "contacts" possédant 3 champs texte : nom, prenom, email. (Attention, comme pour les variables : ne pas utiliser d'accent pour le nom des champs).

Dupliquez votre formulaire en `form-insert.html` et modifiez le pour qu'il appelle le script `insert.php`.

Ce script, dupliqué à partir de `echo.php` devra récupérer les valeurs du formulaire puis se connecter à la base, et insérer un tuple dans la table contacts.

```
<?php
$hostname='serveur_base_donne';  //infolimon
$login='root';                  //login iut
$passwd='root';           // votre passwd
$base='test';                //login iut
$connect = mysql_connect($hostname,$login,$passwd) or die ("erreur de connexion");
mysql_select_db($base,$connect) or die ("erreur de connexion base");


$req = "INSERT INTO ma_table (nom,prenom,email) VALUES('$nom','$prenom','$mail')";
mysql_query($req) or die("erreur insertion");
?>
```

Insérez quelques tuples à la table contacts.

La partie connexion (jusqu'à l'instruction mysql_select_db (comprise)) sera commune à toutes les pages qui accèdent à la base. Plutôt que de la copier/coller de multiples fois, il convient de l' écrire à part dans un fichier à part `connection.php`, puis de l'inclure en première ligne du script `insert.php`.

```
<?php
include("connection.php");
?>
```

Insérez quelques contacts dans la base de données à l'aide du formulaire contacts. Que se passe t'il si vous insérez une chaine de caractères sans "@" dans le champ email ?

2 Consultation des contacts présents dans la base

Les fonctions php *mysql_query* et *mysql_fetch_array* permettent respectivement d'envoyer une requête et de traiter le résultat :

```
<?php
$req="SELECT * from contacts";                  // requete
$res =mysql_query($req);       // envoi de la requete

while ($tuple = mysql_fetch_array($res)) {  //on récupere une ligne du resultat
   echo $tuple['nom'];  //on récupère la valeur du champs nom de cette ligne
}
?>
```

Ecrire une page `list.php` qui affiche tous les contacts de la base, mis en page dans un tableau.

2.1 Clé primaire de la table contact

Avez vous pensé à définir une clé primaire dans la table contact ? Si oui, que se passe t'il en cas de violation de cette contrainte d'intégrité ?

Utilisez la fonction *mysql_errno* (`http://www.php.net/manual/en/function.mysql-errno.php` pour détecter cette violation. Plutôt qu'un message d'erreur qui nécessite de revenir en arrière à l'aide de la touche "back" de votre navigateur (peu ergonomique), revenez sur le formulaire de départ, avec les autres champs (nom et prenom) pré-remplis.

3 Votre site web :

Votre site web doit disposer une page contient des produits, dont la description est stockée dans une base de donnée.

1. Créez une table SQL pour stocker vos produits
2. Ecrire la page de consultation des produits.
3. Ecrire la formulaire et le script php d'insertion d'un produit.

4 Injection de code

1. Il existe 3 APIs permettant d'accéder à MySQL en PHP : ext/mysql (depuis PHP 2.0), ext/mysqli (depuis PHP 5.0) et PDO_MySQL (depuis PHP 5.1). ext/mysql étant la plus utilisée (car la plus ancienne) mais la moins sécurisée. PDO permet, par exemple, d'utiliser des prepared statement côté client ce qui vérifie les requêtes pour retirer les caractères d'échappement notamment (et donc empêche tout type d'injection SQL).

   Pour plus d'information : `http://www.php.net/manual/en/mysqlinfo.api.choosing.php`

   Un lien pas mal sur PDO : `http://studio.jacksay.com/tutoriaux/php/connection-mysql-avec-pdo`

2. Qu'est ce qu'une injection SQL : " Une injection SQL est un type d' exploitation d'une faille de sécurité d'une application interagissant avec une base de données, en injectant une requête SQL non prévue par le système et pouvant compromettre sa sécurité ."
   En pratique, il s'agit de rajouter une requête ou une partie de requête dans un champ de saisie afin que cela soit exécuté sur le serveur.
   Exemple : imaginons une page de login avec champs login/password, à la validation, la requête exécutée est :

```
SELECT id FROM users WHERE login = '(nom)' AND password = '(mot de passe hashé)' ;
```

Si aucune vérification n'est faite sur les champs de saisie et que l'utilisateur saisie `' or 1='1' -` , la requête exécutée sur le serveur sera :

```
SELECT id FROM users WHERE name = '(nom)' AND password = '' OR 1 = 1 --';
```

Cette requête permettra donc de se logger sans connaître le mot de passe de l'utilisateur ! Solution : utiliser mysql_real_escape_string (voir doc : `http://www.php.net/manual/en/function.mysql-real-escape-string.php`)

Historiquement mysql_query permettait d'exécuter des requêtes multiples, mais les dangers de cette fonctionnalité (un utilisateur pouvait par exemple rajouter une requête DROP ou INSERT !) ont obligé une mise à jour. Ce type d'injection n'est donc plus possible.