

# TD développement Web et déploiement de site internet

Patrick Nourrissier – 2016

Ce TD complète les premiers td en javascript et algorithmique. Il est vous proposé d'exploiter les ressources offertes par PHP et Mysql.

## 1. Conception de la base de données

Depuis votre environnement Linux, vous devez créer une base de données appelée **td**. Si le serveur de base de données n'est pas installé, vous devez faire :

**apt-get update**

**apt-get install mysql-server** (un mot de passe sera demandé deux fois, si déjà installé mot de passe : **fc** ou **pat** selon la version de l'image iso)

Ensuite, connectez vous en ligne de commande : **Mysql -uroot -p**

Tapez votre mot de passe.

Pour créer une base de données : **create database td ;**

Pour supprimer : **drop database td ;**

Pour se mettre sur une base de données : **use td ;**

Pour créer un compte utilisateur, avec login et mot de passe uniquement sur la base de données td :

**grant all privileges on td.\* to USER@localhost identified by 'VOTREMOTDEPASSE' with grant option ;**

La syntaxe localhost restreint uniquement la connexion au local, si vous souhaitez l'étendre à tout le réseau, il faut mettre USER@%"

Testez la connexion avec le login et mot de passe créés, il faut rajouter le nom de la bd à la commande mysql : **mysql -uVOTREUSER -p td**

## 2. Installation de phpmyadmin

L'installation de phpmyadmin s'effectue depuis la ligne de commande suivante en tant qu'administrateur :

**apt-get install phpmyadmin**

*Info* : Suite au passage à Jessie, une anomalie de configuration s'est glissée, veuillez éditer le fichier **/etc/apache2/conf.d/phpmyadmin.conf** puis ajouter en fin de ligne 18 le contenu suivant :

php\_admin\_value open\_basedir

**/usr/share/phpmyadmin/:/etc/phpmyadmin/:/var/lib/phpmyadmin/:/usr/share/phpmyadmin/:/usr/share/phpmyadmin/:/usr/share/phpmyadmin/:/usr/share/phpmyadmin/**

## 3. Gestion de l'alias

Par défaut l'alias à utiliser est géré à la syntaxe ligne 3 du même fichier édité précédemment.

**Alias /phpmyadmin /usr/share/phpmyadmin**

Ou si déjà modifié :

**Alias /phpmyadmin54 /usr/share/phpmyadmin**

Si vous avez l'alias /phpmyadmin54, veuillez aussi mettre en commentaire les dernières lignes <Directory... jusqu'à la fin en rajoutant un # à chaque ligne. Vous pourrez ensuite au moment du test des droits les enlever. (Voir point suivant).

```
#<Directory /usr/share/phpmyadmin>
# <IfModule mod_authn_file.c>
# AuthType Basic
# AuthName "phpMyAdmin Access"
# AuthUserFile /etc/phpmyadmin/htpasswd
# </IfModule>
# Require valid-user
#</Directory>
```

Vous pouvez faire un essai en saisissant :

<http://td/phpmyadmin>  
ou <http://td/phpmyadmin54>

Je vous demande de modifier l'alias depuis ce même fichier :  
**/etc/apache2/conf.d/phpmyadmin.conf**

en utilisant un autre alias moins parlant (afin d'éviter les robots). Puis faites un rechargement de configuration : **service apache2 reload**

#### **4. Protection par mot de passe .htaccess**

Idéalement, il faudrait protéger ce dossier en y mettant un login et mot de passe. Nous allons utiliser la commande **htpasswd** pour cela.

Editer le fichier **/etc/apache2/conf.d/phpmyadmin.conf**

Puis ajouter le contenu suivant en fin de fichier :

```
<Directory /usr/share/phpmyadmin>
  <IfModule mod_authn_file.c>
    AuthType Basic
    AuthName "phpMyAdmin Access"
    AuthUserFile /etc/phpmyadmin/htpasswd
  </IfModule>
  Require valid-user
</Directory>
```

Enregistrez et quittez. Nous allons maintenant générer le login et mot de passe.

Faites

**cd /etc/phpmyadmin/**

Puis générez un compte :

**htpasswd -c htpasswd USERNAME**

Un mot de passe est demandé, ne l'oubliez pas !

N'oubliez pas le rechargement de service à chaque modification. Faites de nouveau la tentative de connexion sur votre phpmyadmin avec l'alias modifié, vous devriez avoir le message suivant :

?

**Authentication Required**

A username and password are being requested by http://192.168.56.100. The site says: "phpMyAdmin Access"

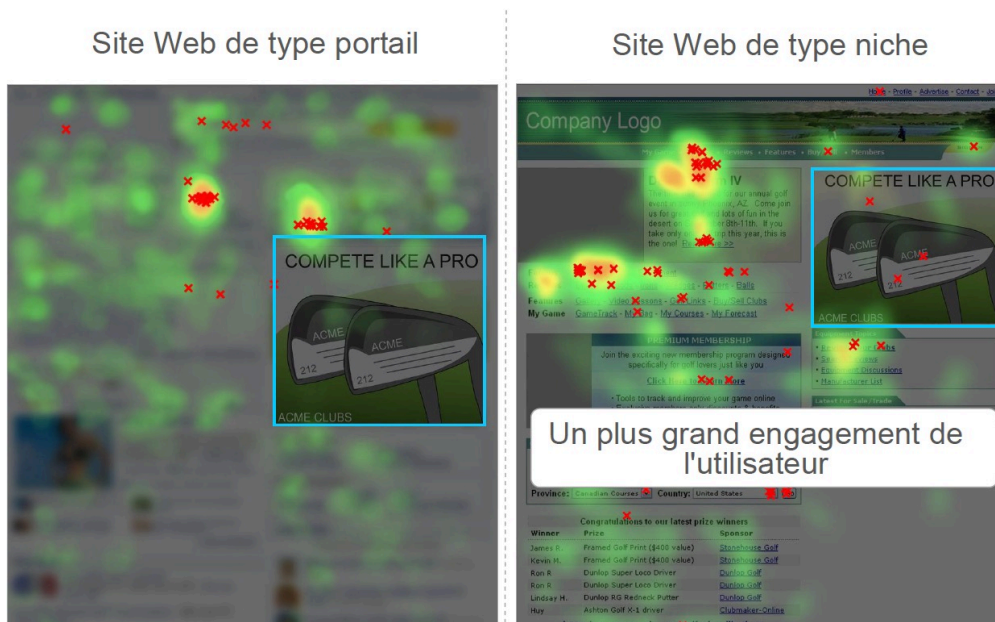
User Name:

Password:

Se rappeler de ce mot de passe avec LastPass.com

## 5. Modélisation de la base de données

L'objectif est de pouvoir stocker l'ensemble des générations de points à chaque appel Ajax. Cette génération aléatoire de points peut s'assimiler à des clics d'utilisateur. Ainsi la grille de damier que vous avez commencé à faire peut finalement être vue comme un découpage de zones sur une page internet. Voici un exemple de ce que produit Google lors du travail d'analyse du comportement d'un utilisateur sur un site :



L'idée est qu'ensuite chaque case prendra une couleur qui sera dépendante du nombre de fois que celle-ci a été sélectionnée de manière aléatoire.

Il est demandé dans un premier temps de modéliser ce qui vous semble nécessaire, notamment :

- les appels
- le résultat de la génération

Pensez aux optimisations possibles. Une fois la base conçue, faites une nouvelle fonction sur base de CallSampleBuildAjax afin qu'elle puisse côté PHP alimenter vos tables nouvellement créées. Vous pouvez définir un nouveau paramètre lu en POST ou GET pour tester plus facilement), il faudra alors modifier un peu le début du script PHP.

```
if (isset($_GET["op"])) $op=$_GET["op"];
```

Pensez à bien gérer les différents cas de gestion :

- Initialisation de la base
- Traitement d'un appel comprenant :
  - o Gestion de la session : initialisation avec session\_start(), lecture avec session\_id()
  - o Insertion des nouvelles coordonnées
  - o Actualisation de la base (en cas de nécessité d'actualiser)
  - o Génération des données de retour

### Exemple de connexion à la base de données (version PDO) :

```
$user='MONCOMPTE';
$password='MONPASSWORD';
$pdo = new PDO('mysql:host=localhost;dbname=MADB, $user,
$password,array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET names utf8') );
```

### Exemple de requête de sélection :

```
$strRq = "SELECT count(nb) as nb1 FROM matrice WHERE nb=20";
$st = $pdo->prepare($strRq);
$st->execute();
$fields=$st->fetch();
//lecture
$nb1=$fields['nb1'];
```

### Exemple de requête de sélection préparée (protection des données insérées) :

```
$strRq6="SELECT count(nbpions) as nbtot FROM appel WHERE nbpions=:nbpions";
$st6 = $pdo->prepare($strRq6);
$st6->bindValue(':nbpions', $nbvalue, PDO::PARAM_INT);
$st6->execute();
```

### Exemple d'insertion à la base de données par paquet de record :

```
$st5 = $pdo->query("insert into matrice(xy, tot, coef, nb) values
(\"1_1\",0,0,20) ");
```

### Exemple d'insertion avec les méthodes bind :

```
$strRq1='INSERT INTO appel(nbpions, dateappel) VALUES (:nbpions,
:dateappel)';
$st1 = $pdo->prepare($strRq1);
$st1->bindValue(':nbpions', $nbvalue, PDO::PARAM_STR);
$st1->bindValue(':dateappel', $date, PDO::PARAM_STR);

$res=$st1->execute();

// récupération du dernier élément inséré
$idappel = $pdo->lastInsertId($res);
```