

# TP 08 (noté) — Espionnage

fabien.givors@unice.fr

2014-2015

- Le présent TP a pour objet de vous faire réaliser une mini base de données pistant les utilisateurs de votre site.
- Ce faisant, vous allez utiliser le langage SQL et la classe PDO pour la création et l'utilisation de la BDD.
- Le barème est donné à titre indicatif.

## 1 Structure de données (12 pts)

**Exercice 1.** Création de la classe et instantiation (facile — 2 pts) :

1. Dans un fichier `Espion.php`, créez une classe `Espion`. Cette classe devra contenir :
  - Un champ privé `$bdd` qui stockera la base de données espion.
  - Un constructeur, affichant « Coucou ».
2. Testez votre code en instanciant votre classe `Espion` depuis un fichier `test.php`.

**Exercice 2.** Création et initialisation de la base de données (facile — 3 pts) :

*Commencez par créer un dossier `bdd` dans lequel le serveur pourra écrire sa base de données. Ce dossier doit être dans le même répertoire que vos fichiers `.php`.*

```
cd ~/www/php/votre-répertoire-de-travail
mkdir bdd
chmod 1777 bdd
```

Ajoutez dans la classe une méthode privée `initBdd()`.

1. Cette méthode initialise un objet PDO en ouvrant la base de données `bdd/espion.sqlite`.
2. La méthode crée ensuite deux tables SQL. Les énoncés SQL pour la création<sup>1</sup> sont les suivants :

```
CREATE TABLE IF NOT EXISTS connexions (
    "id" INTEGER PRIMARY KEY,
    "ip" TEXT,
    "navigateur" TEXT)

CREATE TABLE IF NOT EXISTS pages_vues (
    "id" INTEGER PRIMARY KEY,
    "connexion" INTEGER,
    "page" TEXT,
    "date" INTEGER)
```

**Exercice 3.** Insertion de données (assez facile — 7 pts) :

1. Définissez une méthode privée `nouvelle_connexion()` qui insère dans la table `connexions` l'ip, et l'utilisateur du navigateur (vous trouverez toutes ces données dans la super-globale `$_SERVER`<sup>2</sup>).
2. Complétez la méthode `ajoute_connexion()` pour qu'elle renvoie l'id de la connexion qu'elle vient de rajouter, en utilisant la méthode `lastInsertId` de la classe PDO<sup>3</sup>.
3. Définissez une méthode privée `visite_page($id_connexion)` qui insère dans la table `pages_vues` l'id de la connexion (reçu en paramètre), d'adresse de la page courante (récupérée depuis `$_SERVER`, et l'heure de visite (`NOW()`)).

<sup>1</sup>On notera les mots clés « IF NOT EXISTS », qui permettent de ne pas renvoyer d'erreur si la table existe déjà.

<sup>2</sup>Voir <https://php.net/manual/fr/reserved.variables.server.php>

<sup>3</sup>Pour la méthode `bdd->lastInsertId()`, voir <http://php.net/manual/fr/pdo.lastinsertid.php>.

- Définissez une méthode publique `espionner()` qui ajoute la connexion et la page visitée.
- Créez trois pages `un.php`, `deux.php` et `trois.php`<sup>4</sup> qui affichent leur nom et sont espionnées par Espion, ainsi qu'une page `admin.php`, non-espionnée, qui affiche simplement son nom.
- Testez votre code via `test.php` en rajoutant dans Espion la méthode publique `_dump()` ci-dessous, affichant tout le contenu de toutes les tables.

```

1 public function _dump()
2 {
3     $tables = array("connexions", "pages_vues");
4     foreach($tables as $t)
5     {
6         $req = "SELECT * FROM $t;";
7         $req = $bdd->prepare($req);
8         $req->execute();
9         while($ligne=$req->fetch(PDO::FETCH_ASSOC))
10            print_r($ligne);
11     }
12 }

```

## 2 Affichage des données (7 pts)

On veut pouvoir afficher les données des tables sous la forme suivante :

ID	IP	Navigateur	Pages visitées		
			ID	Page	Date
1	127.0.0.1	Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586).	1	/un.php	04/12/2014 17:15
2	192.168.0.1	Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586).	2	/un.php	04/12/2014 17:16
			3	/deux.php	04/12/2014 17:17
			4	/un.php	04/12/2014 17:18
3	127.0.0.1	Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586).	6	/trois.php	04/12/2014 17:20
			5	/un.php	04/12/2014 17:19

**Exercice 4.** Les trois premières colonnes (assez facile — 3 pts)

En vous inspirant de la méthode `_dump()`, ajoutez à la classe Espion une méthode publique `afficheRapport()` affichant un tableau correspondant aux trois première colonnes du rendu souhaité.

**Exercice 5.** Les trois dernières colonnes (plutôt difficile, et long — 4 pts)

Les trois dernières colonnes correspondent aux données renvoyées par la requête SQL suivante (dans laquelle `n` est à remplacer par l'id de la connexion en cours) :

```
SELECT * FROM pages_vues WHERE connexion=n
```

Modifiez `afficheRapport()` pour qu'elle affiche en plus des trois premières colonnes, les trois dernières.

## 3 Sessions (facile et super rapide — 4 pts)

**Exercice 6.** Maintenant, vous voulez que l'espion ne crée qu'une seule entrée dans la table `connexions` par session du navigateur.

Pour arriver à cela,

- Modifiez vos pages pour qu'elles utilisent les sessions.
- Modifiez la méthode `espionner()` pour qu'elle enregistre l'id de connexion dans la session.
- Modifiez la méthode `espionner()` pour qu'elle ne rajoute la nouvelle connexion dans la table qu'une fois par session.
- Ajoutez une page de suppression de session.
- Testez le résultat.

<sup>4</sup>Le choix des noms est libre...