

Ce TP doit permettre gérer le processus d'authentification pour l'administrateur du backoffice du site

Il n'y a qu'un seul administrateur reconnu dans cette application.

Les informations sur le compte admin ont été mises en place à la création du site. Elles figurent dans la table **fande_parameters** dans les clés **admin_name** et **admin_pwd**.

Activité 1

I:/_xampp/htdocs/dev.fande.fr/apps/models/parameter.class.php

Dans un premier temps, pour pouvoir travailler avec la table **fande_parameters**, vous allez mettre en place une **classe de gestion** de cette table.

Le concept est identique à celui mis en place dans le fichier /apps/models/movie.class.php.

1. Créez le fichier /apps/models/parameter.class.php. Complétez la trame de code suivante :

```
class Parameter .... {
    ...
}
```

Dans cette classe, **deux méthodes** spécifiques doivent être écrites pour pouvoir **lire** un paramètre et **modifier** la valeur d'un paramètre.

Voici le code de la première méthode à ajouter :

```
/**
 * Retourne les informations sur un paramètre.
 * @param string $param le paramètre
 * @return string
 */
public function getParam( $param ) {

    return $this->fpdo->from( $this->table )->where( 'name', $param )->fetch('value');
}
```

Donnez le code de la seconde méthode :

```
/**
 * Retourne les informations sur un paramètre.
 * @param string $param le paramètre à modifier
 * @param string $value la valeur du paramètre
 * @return string
 */
public function setParam( $param, $value ) {

    ... à compléter
}
```

EVENTUELLEMENT, mettez en place un petit protocole de test sur ces 2 fonctions.

Activité 2

I:/_xampp/htdocs/dev.fande.fr/apps/templates/login.tpl

Pour pouvoir s'identifier, nous allons utiliser la commande **admin-login**
localhost/dev.fande.fr/admin-login

1. Intégrez, dans le routeur, la commande **admin-login**. Ajoutez, dans le contrôleur, la fonction de traitement associée (**login_action**).

2. Ajoutez le template (**login.tpl**) pour l'identification qui doit ressembler à :

Utilisateur:

Mot de passe:



La validation de ce formulaire doit lancer la commande `login-check` et envoyer, en mode **POST**, les 2 paramètres suivants: `admin_name`, `admin_pwd`.

3. Intégrez, dans le routeur, la commande `login-check`. Ajoutez le traitement PHP qui vérifie que si des données sont transmises en **POST** et si les conditions d'authentification sont correctes, alors une **SESSION** d'administrateur peut être ouverte et l'utilisateur est redirigé vers la page de filmographie. Ce traitement sera intégré directement dans le routeur.

```
//== vérification de l'identification =====
elseif ( 'login-check' == $command
        && isset( $_POST['admin_name'] )
        && ... à compléter )
{
    // vérifier le login
    $oParameter = new Parameter( $fpdo );
    if ( ... à compléter ) {
        ... à compléter
    } else {
        redirect('admin-login');
    }
}
```

Remarque 1: pour récupérer la valeur du nom d'admin et le mot de passe d'admin de la base, il faut utiliser la méthode `getParam` de l'objet `$oParameter` :

```
$oParameter->get( 'admin_name' );
```

Remarque 2: pour gérer les variables de **SESSION**, il faut écrire `session_start()`; en toute première instruction de votre script (avant même les appels de librairie). Dans notre contexte de travail, cette écriture doit être portée dans le fichier `setting.inc.php`.

Remarque 3: la session sera matérialisée par la variable `$_SESSION['login'] = true`;

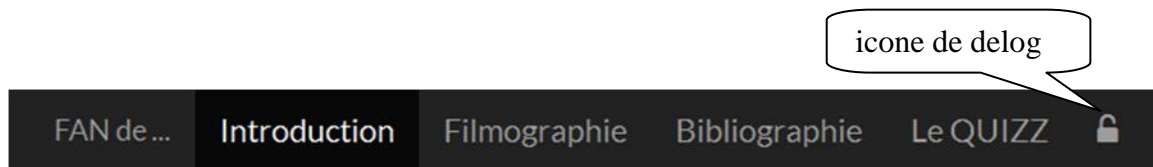
Activité 3

En fait, quand une session d'administrateur est ouverte, le contenu des pages va être enrichi.

Par exemple, la page de filmographie doit permettre à l'administrateur **d'éditer**, **d'effacer** ou **d'ajouter** un film.

Plus simplement, pour le moment l'objectif est d'ajouter une icône (dans le menu) qui permette à l'administrateur de clore sa session s'il le souhaite.

1. Ajoutez, dans le gabarit `layout.tpl`, le HTML qui permette d'obtenir le visuel suivant :



Cette icône ne doit apparaître que si une session est ouverte. Vous devez, autour du HTML précédent, ajouter un test

```
{if isset( $smarty.session.admin ) && $smarty.session.admin}
    ... ici HTML pour l'icône ...
{/if}
```

Consultez la page smarty.net/docsv2/fr/language.variables.smarty.tpl pour comprendre la forme d'écriture appropriée qui permet de manipuler les variables **GET**, **POST** ou **SESSION** avec **SMARTY**.

2. Un click sur ce pictogramme doit exécuter la commande `logout`. Le code associé à cette commande doit :

- effacer la variable `$_SESSION['admin']`,
- supprimer la session

En utilisant la documentation du PHP, donnez ce code.

Bon travail et ... courage !!!

«Rome ne s'est pas faite en un jour»