

# TP 9 : Jukebox en MVC (avec solutions)

## Objectifs du TP:

- Structurer un site WEB selon le modèle MVC.

## 1. Mise en place de la structure MVC

Vous allez reprendre une dernière fois votre site WEB Jukebox de musique pour le transformer au format MVC. On vous propose de réaliser une structure MVC de type 3 (voir le cours), c'est à dire avec un ensemble de contrôleurs séparés des vues, et avec une URL unique : le fichier `index.php`.

## 2. Le jukebox en MVC

### 2.1 Répertoires et répartition/renommage des fichiers

Cette partie concerne la mise en forme du site du Jukebox dans une structure MVC. Cela consiste donc à répartir/découper les fichiers du TP précédent selon le paradigme modèle, vue et contrôleur. Il faut commencer par une simple recopie des fichiers. Ces fichiers seront modifiés dans la suite.

#### Travail préparatoire de recopie :

- Dupliquer tout votre code du Jukebox du TP précédent dans un nouveau répertoire.
- Créer les répertoires supplémentaires suivants :
  - `view` : va contenir la partie HTML de `jukebox.php`, et `playId.php`.
  - `contrôleur` : va contenir la partie PHP de `jukebox.php`, et `playId.php`.
  - `framework` : placez-y le fichier `view.fw.php` que vous prenez dans l'exemple du jeu de recherche du nombre.
  - `index.php` : le routeur (ou méta-contrôleur) à mettre à jour à partir des exemples de TP précédents.
  - `model` : déjà existant, il contient le code de la classe `Music` et le DAO pour l'accès à la BD.
- Les fichiers `jukebox.php` et `playId.php` sont composés d'une partie contrôle en PHP et d'une partie vue en HTML avec un peu de PHP. En MVC, il faut découper le contenu de ces fichiers en deux parties. La partie PHP de contrôle va naturellement dans le répertoire `contrôleur` et la partie HTML dans `view`. Recopiez et renommez ces fichiers selon les conventions de renommage proposées. Par exemple : `jukebox.php` devient `view/jukebox.view.php` et `contrôleur/jukebox.ctr1.php`. Attention : les fichiers à la racine `jukebox.php` et `playId.php` doivent être au final supprimés.

### 2.2 Mise en place des vues

La vue contient majoritairement de l'HTML avec quelques balises PHP pour afficher le contenu de variables dans la vue.

#### À faire :

- Supprimer la partie PHP du début.
- Changer le chemin relatif vers le style : la vue va être déclenchée à partir de la racine du répertoire, car seule l'URL du fichier `index.php` est autorisée. Le chemin sur le style est alors à partir du répertoire contenant le fichier `index.php`, comme par exemple :

```
<link rel="stylesheet" type="text/css" href="view/design/style.css">
```

- Modifier les URL des liens et actions des formulaires pour désigner un contrôleur avec le nom `ctr1` dans la query string. Par exemple, remplacer :

```
<a href="jukebox.php?page=1&pageSize=?=$pageSize?">
```

par :

```
<a href="index.php?ctr1=jukebox&page=1&pageSize=?=$pageSize?">
```

ou même simplement :

```
<a href="?ctr1=jukebox&page=1&pageSize=?=$pageSize?">
```

- Repérer la liste de toutes les variables PHP utilisées pour afficher du contenu dans la page HTML. Cette liste sera utilisée pour passer les valeurs à la vue dans le contrôleur.

#### ▼ Solution pour `playId.view.php` :

Liste des variables : `music`, `page`, `pageSize`.

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="view/design/style.css">
    <title></title>
  </head>
  <body>
    <header>
      <h1>Playing : <?=$music->getTitle()?> from <?=$music->getAuthor()?></h1>
    </header>
    <nav>
      <a href="?ctr1=jukebox&page=<?=$page?>&pageSize=<?=$pageSize?>">
        back
      </a>
    </nav>
    <section>
      <figure>
        <img src=<?=$music->getCover()?> >?>
        <audio src=<?=$music->getMp3()?> controls autoplay ></audio>
      </figure>
    </section>
  </body>
</html>
```

#### ▼ Solution pour `jukebox.view.php` :

Liste des variables : `page`, `pageSize`, `prev`, `next`, `nbButtonPage`, `lastPage`, `list`. La variable `$i` est interne à la boucle `for` et donc ne fait pas partie de la liste. Idem pour la variable `$music`.

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>&#x1f399; Mon jukebox avec BD</title>
    <link rel="stylesheet" type="text/css" href="view/design/style.css">
    <title></title>
  </head>
  <body>
    <header>
      <h1>Ma musique dans mon Jukebox</h1>
    </header>
    <!-- Navigation -->
    <nav>
      <p>
        <a href="?ctr1=jukebox&page=1&pageSize=<?=$pageSize ?>">
          <span class="arrow left"></span><span class="arrow left"></span>
        </a>
        <a href="?ctr1=jukebox&page=<?=$prev ?>&pageSize=<?=$pageSize ?>">
          <span class="arrow left"></span>
        </a>
        <a class="selected" href="?ctr1=jukebox&page=<?=$page ?>&pageSize=<?=$pageSize ?>"><?=$page ?>
        </a>
        <?php for ($i = $page + 1; $i < $page + $nbButtonPage && $i <= $lastPage; $i++) : ?>
          <a href="?ctr1=jukebox&page=<?=$i ?>&pageSize=<?=$pageSize ?>"><?=$i ?>
        </a>
        <?php endfor; ?>
        <a href="?ctr1=jukebox&page=<?=$next ?>&pageSize=<?=$pageSize ?>">
          <span class="arrow right"></span>
        </a>
        <a href="?ctr1=jukebox&page=<?=$lastPage ?>&pageSize=<?=$pageSize ?>">
          <span class="arrow right"></span><span class="arrow right"></span>
        </a>
      </p>
      <form action="?ctr1=jukebox&page=<?=$page ?>" method="get">
        <p>Musiques/page</p>
        <input type="text" name="pageSize" value=<?=$pageSize ?>" maxlength="4" size="2">
        <input type="hidden" name="page" value=<?=$page ?>">
      </form>
    </nav>

    <!-- Contenu principal -->
    <main>
      <section>
        <?php foreach ($list as $music) : ?>
          <figure>
            <a href="?ctr1=playId&id=<?=$music->getId()?>&page=<?=$page ?>&pageSize=<?=$pageSize ?>">
              <img src=<?=$music->getCover()?> ?> />
            </a>
            <figcaption>
              <music-song><?=$music->getTitle()?></music-song>
              <music-group><?=$music->getAuthor()?></music-group>
              <music-category><?=$music->getCategory()?></music-category>
            </figcaption>
          </figure>
        <?php endforeach; ?>
      </section>
    </main>
    <footer>
      Jukebox IUT
    </footer>
  </body>
</html>
```

### 2.3 Mise en place des contrôleurs

Un contrôleur est composé de 3 parties qui sont dans l'ordre :

- La récupération des données, par exemple celle de la query string, les valeurs de configuration, etc.
- La réalisation des calculs, avec l'utilisation du modèle.
- La sélection et le déclenchement de la vue.

#### À faire :

- Supprimer la partie HTML.
- Changer les chemins des inclusions. Comme les fichiers sont maintenant dans le sous-répertoire `contrôleur`, pour accéder au modèle il faut simplement désigner ce répertoire. Par exemple :

```
require_once('model/music.class.php');
```

- Ajouter l'inclusion de l'unique classe du framework :

```
require_once('framework/view.fw.php');
```

- Créer l'objet instance de `View` et passer en paramètre chaque variable de la liste de l'étape précédente grâce à `assign()` puis appeler la vue avec `display()`.

#### ▼ Solution pour `playId.ctr1.php` :

Liste des variables de la vue : `music`, `page`, `pageSize`.

```
<?php
require_once('model/music.class.php');
require_once('framework/view.fw.php');

// récupération des valeurs de la query string
if (isset($_GET['id'])) {
    $id = intval($_GET['id']);
} else {
    exit("Erreur : id non définit");
}

// Si la page n'est pas indiquée, on repart à 1
if (isset($_GET['page'])) {
    $page = intval($_GET['page']);
} else {
    $page = 1;
}

// Si le nombre de musique par page n'est pas indiqué, on choisit 8
if (isset($_GET['pageSize'])) {
    $pageSize = intval($_GET['pageSize']);
} else {
    $pageSize = 8;
}

// Récupération de l'objet musique correspondant à l'id
$music = Music::read($id);

////////// AJOUTE POUR MVC
$view = new View();

// Passage des paramètres à la vue
$view->assign("music",$music);
$view->assign("page",$page);
$view->assign("pageSize",$pageSize);

// Appel de la vue
$view->display("playId");
?>
```

#### ▼ Solution pour `jukebox.ctr1.php` :

Liste des variables de la vue : `page`, `pageSize`, `prev`, `next`, `nbButtonPage`, `lastPage`, `list`.

```
<?php
require_once('model/music.class.php');
require_once('model/dao.class.php');
require_once('framework/view.fw.php');

// Nombre de boutons de choix de page
$nbButtonPage = 8;

// Partie analyse des valeurs de la query string
$page = $_GET['page'] ?? 1;

// Nombre de musiques par page
if (isset($_GET['pageSize'])) {
    $pageSize = intval($_GET['pageSize']);
} else {
    // Valeur par défaut
    $pageSize = 8;
}

// Calcul de la dernière page : division entière
$lastPage = intdiv(Music::maxId(),$pageSize);

// Test si la page prévue est correcte
if ($page < 1 || $page > $lastPage) {
    $page = 1;
}

// Calcul de premier id de la page
$firstId = ($page - 1)*$pageSize + 1;

$list = array();
// Récupération des données à placer dans la vue à partir du modèle
for($i=$firstId; $i<$firstId+$pageSize && $i <= Music::maxId();$i++){
    // Récupération de l'objet Music
    $m = Music::read($i);
    // Ajout à la liste des images à afficher
    $list[] = $m;
}

////////// AJOUTE POUR MVC

$view = new View();
// Passage des paramètres à la vue
$view->assign("page",$page);
$view->assign("pageSize",$pageSize);
$view->assign("prev",$prev);
$view->assign("next",$next);
$view->assign("nbButtonPage",$nbButtonPage);
$view->assign("list",$list);
$view->assign("lastPage",$lastPage);
// Appel de la vue
$view->display("jukebox");
?>
```

### 2.4 Le point d'entrée du site

Il ne reste plus qu'à mettre à jour dans le fichier `index.php`, la liste des contrôleurs acceptés.

#### ▼ Solution complète :

À récupérer dans le fichier zip [data/jukebox\\_MVC.zip](#)

### 2.5 Sécurité

Pour sécuriser votre site, ajoutez à la racine du site un fichier `.htaccess`. Il indique au serveur Apache qu'il ne faut autoriser que le fichier `index.php` et bien sûr tous les fichiers de style, c'est à dire ceux qui se terminent par `css`.

```
# Les règles qui interdisent d'abord, puis autorise ensuite
Order deny,allow

# Par défaut aucune autorisation à rien sur ce site
Deny from all

# On lève l'interdiction sur le fichier index.php
# Autorise aussi l'accès au site sans index.php
<FilesMatch "^.(index\.php)?$">
    Allow from all
</FilesMatch>

# On autorise les fichiers de style
<Files "*.css">
    Allow from all
</Files>
```