

# TD9 : structuration d'un site web

UCBL - Département Informatique de Lyon 1 – BDW - automne 2023

Objectif du TD : structurer un site web avec l'architecture MVC

## Rappel du schéma relationnel :

ACTIVITÉS (idA, titre, type, #idU)  
SEMAINES (année, numéro, ouverteAuVote)  
UTILISATRICES (idU, nom, prénom, email)  
VOTER (#idA, date, #idU, #année, #numéro, aAimé)

```
<?php
require('inc/routes.php');
require('inc/includes.php');
require('inc/config-bd.php');
require_once('modele/modele.php');
$connexion = getConnexionBD();
?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title><?= $nomSite ?></title>
  <link href="css/style.css" type="text/css">
</head>
<body>
  <?php include('static/header.php'); ?>
  <div id="divCentral">
    <?php include('static/menu.php'); ?>
    <main>
      <?php
      $controlleur = 'controlleurAccueil';
      $vue = 'vueAccueil';
      if(isset($_GET['page'])) {
        $nomPage = $_GET['page'];
        if(isset($routes[$nomPage])) {
          $controlleur = $routes[$nomPage]['controlleur'];
          $vue = $routes[$nomPage]['vue'];
        }
      }
      include('controlleurs/' . $controlleur . '.php');
      include('vues/' . $vue . '.php');
      ?>
    </main>
  </div>
  <?php include('static/footer.php'); ?>
</body>
</html>
```

Fichier 1

```
<?php
$nomSite = "Activotes";
$baseline = "Votez pour vos activités préférées !";
?>
```

Fichier 2

```
body {
  background-color: #FFFFFF;
  box-sizing: border-box;
  margin: 0;
}
```

Fichier 3 (extrait)

```
<h1>Bienvenue !</h1>
...
<section>
<h2>Rechercher une activité</h2>
<form method="post" action="index.php?page=afficher">
  <label for="champRequete">Activité : </label>
  <input type="text" name="champRequete" id="champRequete"
  ↪ placeholder="cinéma"><br>
  <label for="champCategorie">Catégorie : </label>
  <select name="champCategorie" id="champCategorie"><?= @@@@
  ↪ ?></select><br/>
  <input type="submit" name="boutonRechercher"
  ↪ value="Rechercher">
</form>
</section>
```

Fichier 4 (extrait)

```
<?php
$categoriesActivite = @@@@($connexion, 'ACTIVITÉS', 'type');
$optionsCategorie = "";
@@@@($categoriesActivite as @@@@) {
  $optionsCategorie .= '<option value="' . $cat['type'] . '>'
  ↪ . $cat['type'] . '</option>';
}
?>
```

Fichier 5

```
<?php
$routes = array(
  'afficher' => array('controlleur' => 'controlleurAfficher',
  ↪ 'vue' => 'vueAfficher'),
  'rechercher' => array('controlleur' => 'controlleurRechercher',
  ↪ 'vue' => 'vueRechercher'),
  'voter' => array('controlleur' => 'controlleurVoter', 'vue' =>
  ↪ 'vueVoter')
);
?>
```

Fichier 6

```
<?php
function getDistinctValuesFrom($connexion, $nomTable,
  ↪ $attribut) {
  $requete = "SELECT DISTINCT(@@@@) FROM @@@@";
  $res = @@@@($connexion, $requete);
  $instances = @@@@($res, MYSQLI_ASSOC);
  return $instances;
}
?>
```

Fichier 7 (extrait)

## Exercice 1 Compréhension de l'architecture MVC

Le site est structuré selon l'architecture MVC et utilise le principe de routage. Les fichiers précédents correspondent au code de la page d'accueil, qui affiche un formulaire de recherche. Ignorez les '####' pour l'instant.

1. En analysant les fichiers numérotés de 1 à 7, déduisez-en le nom de chacun de ces fichiers.
2. Décrivez le cheminement dans les fichiers quand on demande la page (`index.php`).
3. On souhaite développer la page de vote pour une activité. Quelles modifications devez-vous apporter aux fichiers existants ? Créez ensuite la page, dont le contenu se compose d'un titre et du paragraphe "Page disponible prochainement".

## Exercice 2 Compréhension de code

1. Le code des fichiers 4, 5 et 7 est incomplet (au niveau des huit '####'), à vous de le compléter !

## Exercice 3 Écriture de code

1. Complétez la fonctionnalité permettant de voter pour une activité : le contrôleur et la vue sont fournis, mais pas les fonctions du modèle appelées par le contrôleur (`insertVote`, `getActivites` et `getSemaines`).

```
1 <?php
2 if(isset($_POST['boutonValider'])) { // formulaire soumis
3     $idU = 1; // stocké en session
4     $idA = $_POST['activite'];
5     $exp = explode('-', $_POST['semaine']);
6     $annee = $exp[0]; // extrait l'année
7     $semaine = $exp[1]; // extrait la semaine
8
9     $insertion = insertVote($connexion, $idU, $idA, $annee, $semaine);
10    if($insertion == TRUE)
11        $message = "Le vote est enregistré !";
12    else
13        $message = "Erreur lors de l'enregistrement du vote.";
14 }
15
16 $activites = getActivites($connexion);
17 $optionsActivite = "";
18 foreach($activites as $a)
19     $optionsActivite .= '<option value="' . $a['idA'] . '>' .
20     ↪ $a['titre'] . '</option>';
21
22 $semaines = getSemaines($connexion, true);
23 $optionsSemaine = "";
24 foreach($semaines as $s)
25     $optionsSemaine .= '<option>' . $s['sem'] . '</option>';
26 ?>
```

Fichier controleurVoter.php

```
<h2>Votez pour une activité</h2>
<?php
if(isset($message)) {
    echo "<p
    ↪ class=\"fond-jaune\">$message</p>";
}
?>
<form method="post" action="#">
<label for="activite">Activité : </label>
<select name="activite" id="activite"><?
↪ $optionsActivite ?></select>
<br/>
<label for="semaine">Semaine : </label>
<select name="semaine" id="semaine"><?
↪ $optionsSemaine ?></select>
<br/>
<input type="submit" name="boutonValider"
↪ value="Voter"/>
</form>
```

Fichier vueVoter.php

## Exercice 4 Bonus

1. Quels inconvénients pouvez-vous détecter par rapport à l'architecture utilisée ?