



Projet Festival – Application du MVC

## Sommaire

Introduction.....	2
Présentation du projet Festival.....	2
MVC.....	2
Qu'est-ce que le MVC ?.....	2
Application sur Festival.....	2
Compétences acquises.....	8

# Introduction

Ce projet nous a été proposé dans le cadre de notre formation de deuxième année en BTS SIO option SLAM. Les enseignants nous réparti par groupe de 2-3 pour le mener à bien.

Ainsi ce projet a été réalisé par Linguet Joshua, le chef de groupe, Gavriliuc Carin et Té Cédric.

## Présentation du projet Festival

Le contexte proposé est le suivant : la maison de Ligue de Lorraine souhaite organiser un festival de sports nommée le festival Sp'Or mais ne dispose d'aucune application pour gérer l'hébergement des participants à ce festival. Chaque ligue pourra y inscrire des participants et/ou équipes et tenir un stand de présentation de la ligue pendant 15 jours. Les établissements scolaires de la région vont mettre à disposition de la M2L des chambres pour l'hébergement des personnes participant au festival.

Afin de réaliser ladite application, les enseignants ont mis à disposition un code que les étudiants devront modifier et adapter afin de répondre aux besoins de la M2L.

Ce projet a été réalisé en première année de BTS SIO mais en deuxième année, il nous a été demandé de le reprendre et d'appliquer l'architecture MVC.

## MVC

Qu'est-ce que le MVC ?

MVC veut dire Modèle-Vue-Contrôleur. C'est un motif d'architecture très utilisé dans la conception et programmation d'application web. En effet, il permet de découper le code en 3 parties, apportant une meilleure visibilité et lecture du code :

- Modèle : va contenir toutes les fonctions en lien avec la base de données
- Vue : va retourner/afficher les résultats fournis par le contrôleur à l'utilisateur
- Contrôleur : sert d'intermédiaire entre l'utilisateur et le modèle : en fonction de la demande de l'utilisateur, le contrôleur va alors interroger le modèle, ce dernier va lui retourner un résultat que le contrôleur va transmettre à la vue qui va l'afficher

### Application pour Festival

L'application de cette nouvelle architecture a été faite en plusieurs temps. Tout d'abord, il a fallu rassembler tous les fichiers contenant les fonctions puis faire le tri entre celles permettant de récupérer des informations et celles les modifiant. Un modèle pour la connexion a la base de données a également été créé afin d'éviter la redondance d'information pour les fichiers devant se connecter à la base de données.

Ces 3 nouveaux fichiers ont été stockés dans un dossier appelé « models ».

```

<?php

// fichier de connexion à la bdd, à inclure sur les autres pages

//identifiant et mdp mysql
$user = 'root';
$pswd = 'root';

//hote de l'application
$host = 'localhost';

//nom de la base de donnée
$bdd = 'festival';

//data source name, soit la definition de l'hote et du nom de la base
$dsn = "mysql:$host=localhost;dbname=$bdd";

//connexion

function getBDD()
{
    try{
        return new PDO('mysql:localhost=localhost;dbname=festival', 'root', 'root'); // $dbh l'objet de notre bdd connecté à l'application
        //echo "connexion pdo reussie (à retirer) <br/>";
    } catch(PDOException $e) {
        return "erreur de connexion : ".$e->getMessage()."<br/>";
    }
}
}
>>

```

Modèle pour la connexion à la base de données

Quelques exemples de fonctions que l'on retrouve dans le fichier permettant la récupération d'information

```

function getEtabs(){
    //tableau associatif de TOUS les etablissements
    $bdd = getBDD();
    $reqEtab = $bdd->query("select id, nom, nombreChambresOffertes from Etablissement order by id");
    $Etabs = $reqEtab->fetchAll(PDO::FETCH_ASSOC);
    return $Etabs;
}

```

Fonction permettant de récupérer les établissements classés par ID

```

function getGroupeDetails($id){
    $bdd = getBDD();
    $reqGroupe = $bdd->query("select * from Groupe where id = '$id'");
    $Groupe = $reqGroupe->fetch(PDO::FETCH_ASSOC);
    return $Groupe;
}

```

Fonction permettant d'afficher le détail d'un groupe en particulier

```
function getNbChambresRestantes($id){ //etablisements qui ont déjà des attributions (pas vides)
    $bdd = getBDD();
    $req = $bdd->query("select sum(nombreChambresOffertes - (select sum( nombreChambres )
    | from attribution where idEtab='$id' )) as nb from etablisement where id='$id'");
    $nb = $req->fetch(PDO::FETCH_ASSOC);
    return $nb['nb'];
}
```

Fonction retournant le nombre de chambres restantes d'un établissement

Quelques fonctions se trouvant dans le fichier permettant la modification d'éléments

```
function updateEtab($id, $nom, $adresseRue, $codePostal,
    $ville, $tel, $adresseElectronique, $type,
    $civiliteResponsable, $nomResponsable,
    $prenomResponsable, $nombreChambresOffertes)
{
    $bdd = getBDD();

    $req="update Etablissement set nom='$nom',adresseRue='$adresseRue',
    codePostal='$codePostal',ville='$ville',tel='$tel',
    adresseElectronique='$adresseElectronique',type='$type',
    civiliteResponsable='$civiliteResponsable',nomResponsable=
    '$nomResponsable',prenomResponsable='$prenomResponsable',
    nombreChambresOffertes='$nombreChambresOffertes' where id='$id'";

    $bdd->exec($req);
}
```

Fonction permettant la modification d'un établissement

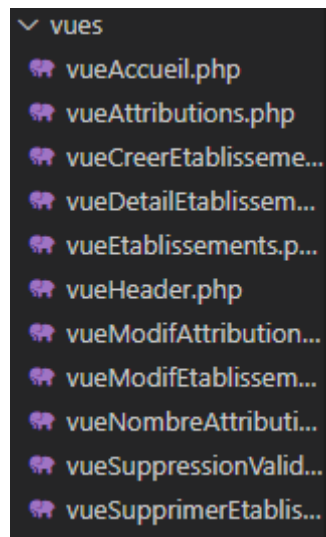
```
function deleteEtab($id){
    $bdd = getBDD();

    $req="delete from Etablissement where id='$id'";

    $bdd->exec($req);
}
```

Fonction permettant la suppression d'un établissement

Par la suite nous avons créé une vue pour chacun des fonctionnements implémentés, que cela soit la consultation des ligues ou la page des établissements d'hébergement avec le nombre de chambres disponible.



Liste des vues créées

FESTIVAL SP'OR  
HÉBERGEMENT DES ÉQUIPES

ACCUEIL

GESTION ÉTABLISSEMENTS

ATTRIBUTIONS CHAMBRES

Etablissements

College Lamartine-0350123A

Voir détail

Modifier

Supprimer

College de Moka-0350785N

Voir détail

Modifier

Supprimer

College Leonard de Vinci-0351234W

Voir détail

Modifier

Supprimer

Centre de rencontres internationales-05DF7D8Z

Voir détail

Modifier

Supprimer

Créer

Vue des établissements

FESTIVAL SP'OR  
HÉBERGEMENT DES ÉQUIPES

ACCUEIL

GESTION ÉTABLISSEMENTS

ATTRIBUTIONS CHAMBRES

[Effectuer ou modifier les attributions](#)

College Lamartine - 0350123A

Nom groupe

(France) Equipe de Pole Dance de Paris (38 membres)

(France) Equipe feminine de Football Marseillaise (22 membres)

Chambres attribuées

13

8

College de Moka - 0350785N

Nom groupe

(France) Equipe de Football Limoges (40 membres)

(France) Equipe de Rugby Rouen (25 membres)

Chambres attribuées

11

9

College Leonard de Vinci - 0351234W

Nom groupe

(France) Equipe de Football Limoges (40 membres)

(France) Equipe feminine de badminton de Meaux (29 membres)

(France) Equipe de Handball de Grenoble (19 membres)

Chambres attribuées

3

10

7

Vue des attributions de chambres avec les établissements hébergeurs

Enfin la création du contrôleur qui va contenir les fonctions qui vont interroger le modèle.

```

<?php

require 'models/modelGet.php';
require 'models/modelUpdate.php';

function accueil(){ //index sans action
    require 'vues/vueAccueil.php';
}

////////// ETABLISSEMENTS //////////

function etab(){ // action etablisement

    if($_REQUEST['action'] == "validerCreEtab"){
        createEtab($_REQUEST['id'], $_REQUEST['nom'], $_REQUEST['adresseRue'], $_REQUEST['codePostal'], $_REQUEST['ville'], $_REQUEST['tel'],
            $_REQUEST['adresseElectronique'], $_REQUEST['type'], $_REQUEST['civilliteResponsable'],
            $_REQUEST['nomResponsable'], $_REQUEST['prenomResponsable'], $_REQUEST['nombreChambresOffertes']);
    }

    $etabs = getEtabs();

    require 'vues/vueEtablissements.php';
}

function etabDetail($id){ // action details
    $etab = getEtabsDetails($id);
    require 'vues/vueDetailEtablissement.php';
}

function creerEtab(){ // action créer
    $create = updateCreerEtab();
    require 'vues/vueCreerEtablissement.php';
}

```

```

function modifEtab($id){

    $tabCivillite=array("M.", "Mme", "Melle");

    if($_REQUEST['action'] == "validerModifEtab"){
        updateEtab($id, $_REQUEST['nom'], $_REQUEST['adresseRue'], $_REQUEST['codePostal'], $_REQUEST['ville'], $_REQUEST['tel'],
            $_REQUEST['adresseElectronique'], $_REQUEST['type'], $_REQUEST['civilliteResponsable'],
            $_REQUEST['nomResponsable'], $_REQUEST['prenomResponsable'], $_REQUEST['nombreChambresOffertes']);
    }

    $etab = getEtabsDetails($id);

    require "vues/vueModifEtablissement.php";
}

function suppEtab($id){

    $etab = getEtabsDetails($id);

    if(isset($_REQUEST['valide'])){
        deleteEtab($id);
        require 'vues/vueSuppressionValide.php';
    }
    else{
        require "vues/vueSupprimerEtablissement.php";
    }
}

```

```
function attrib(){ // action attributions

    if(count(getEtabsHebergeur()) == 0){
        echo "il n'y a pas d'etablissements hebergeur :/";} //erreur a mettre dans la vue plus tard
    else
        $etabs = getEtabsAyantAttrib();

        $i = 0;
        foreach($etabs as $etab):

            $groupes[$i] = getAttribsFromEtab($etab['id']);
            $i++;

        endforeach;

        require 'vues/vueAttributions.php';
    }

function modifAttrib(){ //
    $etabs = getEtabsHebergeur();
    $groupes = getGroupes();

    $i = 0;
    foreach($groupes as $groupe):

        foreach($etabs as $etab):

            $attribs[$i] = getNbAttrib($etab['id'], $groupe['id']);
            $i++;

        endforeach;

    endforeach;

    require 'vues/vueModifAttributions.php';
}
```

```
function nbAttrib($idEtab, $idGroupe){  
  
    $etab = getEtabsDetails($idEtab);  
    $groupe = getGroupeDetails($idGroupe);  
    $nbAttrib = getNbAttrib($idEtab, $idGroupe);  
  
    //APRES AVOIR CLIQUER SUR VALIDER  
    if(isset($_REQUEST['nbChambres']))  
    {  
        $nbChambres = $_REQUEST['nbChambres'];  
  
        if($nbAttrib == $nbChambres){  
            //rien ici  
        }  
        elseif($nbAttrib == 0)  
            updateAttrib($idEtab, $idGroupe, $nbChambres, 'C');  
  
        elseif($nbChambres == 0)  
            updateAttrib($idEtab, $idGroupe, $nbChambres, 'D');  
  
        else  
            updateAttrib($idEtab, $idGroupe, $nbChambres, 'U');  
    }  
  
    $nbSelect = getNbChambresRestantes($idEtab);  
    if($nbSelect == 0)  
        $nbSelect = getNbAttrib($idEtab, $idGroupe)-1;  
  
    require 'vues/vueNombreAttributions.php';  
}  
?>
```

## Compétences acquises

À travers ce projet, nous avons pu découvrir ce qu'était le MVC et nous avons appris à le mettre en place.