

Compte Rendu TP7

PWS

POO et MVC

Introduction :

L'objectif de ce TP est d'apprendre comment mettre en oeuvre un code respectant les règles du MVC.

Pour ce faire, nous allons utiliser un exemple tout fait d'étudiant et leur groupe et l'adapter au emplacement et leur type.

Les pages modele

Connect.inc.php

Code commenté

```
<?php
try {
    $conn = new PDO('mysql:host=localhost;dbname=basephp029;charset=UTF8', 'umyphp029', 'max01598', array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
}
catch (PDOException $e) {
    echo "Erreur: ".$e->getMessage()."<br/>" ;
    die() ;
}
?>
```

Ici, on adapte seulement l'objet PDO pour qu'il puisse se connecter à notre base de données comme dans les Tps précédents.

Emplacement

Code commenté

```
<?php
// Cette classe servira à transférer, en mode objet, des données entre le modèle, le contrôleur et la vue
class Emplacement {
    public $idEmpl;
    public $adresseEmpl;
    public $anneeConstruction;
    public $idType;

    public function __construct($idEmpl, $adresseEmpl, $anneeConstruction, $idType) {
        $this->idEmpl = $idEmpl;
        $this->adresseEmpl = $adresseEmpl;
        $this->anneeConstruction = $anneeConstruction;
        $this->idType = $idType;
    }
}
?>
```

Ce code permet d'initialiser un emplacement, un emplacement est composé d'un id, d'une adresse, d'une année de construction et de l'id de son type. On passe donc ces valeurs en public et on les initialisera lorsque l'on crée un emplacement.

Type

Code commenté

```

<?php
// Type.php
// Cette classe servira à transférer, en mode objet, des données entre le modèle, le contrôleur et la vue
class Type {
    public $idType;
    public $nomType;

    public function __construct($idType, $nomType) {
        $this->idType = $idType;
        $this->nomType = $nomType;
    }
}
?>

```

Ce code permettra d'initialiser un type, il fonctionne de la même manière qu'un emplacement sauf qu'un type est composé d'un id et de son nom que l'on passera en paramètre à chaque fois que l'on crée un nouveau type.

ModeleEmplacement

Code commenté

```

$res = $conn->prepare("Select * from Emplacement");
$res->execute();
foreach($res as $emp) {
    $listeEmp[] = new Emplacement($emp["idEmpl"], $emp["adresseEmpl"], $emp["anneeConstruction"], $emp[
"idType"]);
}

```

Dans la fonction getListeEmplacements(), on renvoie l'ensemble des emplacements. On adapte la requête préparée pour qu'elle sélectionne tous les emplacements. Puis on adapte la création des emplacements dans le tableau avec les bons attributs.

```

$res = $conn->prepare("Select * from Emplacement where idEmpl = :pidEmplacement");
$res->execute(array('pidEmplacement' => $idEmpl));
$emp = $res->fetch();
$unEmplacement = new Emplacement($emp["idEmpl"], $emp["adresseEmpl"], $emp["anneeConstruction"], $emp[
"idType"]);

```

Dans la fonction getEmplacement(\$idEmpl), on renvoie l'emplacement dont le l'id est passé en paramètre. On modifie la requête préparée pour qu'elle sélectionne l'emplacement où idEmpl=\$idEmpl.

```

$res = $conn->prepare("Select * from Emplacement where idType = :pidType");
$res->execute( array ('pidType' => $idType) );
$listeEmplType = NULL;
foreach($res as $emp) {
    $listeEmplType[] = new Emplacement($emp["idEmpl"], $emp["adresseEmpl"], $emp["anneeConstruction"], $emp[
"idType"]);
}

```

Dans la fonction getListeEmplacementsbyType(\$idType), on renvoie l'ensemble des emplacements correspondant au type passé en paramètre. On adapte la requête préparée pour qu'elle sélectionne tous les emplacements où idType= \$idType . Puis on crée le tableau comme dans la première fonction.

ModeleType

Code commenté

```
$res = $conn->prepare("Select * from Type");  
$res->execute();  
foreach($res as $typ) {  
    $ListeTyp[] = new Type($typ["idType"], $typ["nomType"]);  
}
```

Cette fonction getListeType() permet de récupérer tous les Types, on modifie juste la requête préparée pour sélectionner tous les types de la table Type.

Contrôleur

Pour les contrôleurs, c'est simple, on garde le même cheminement que l'exemple fourni, mais on l'adapte aux nouvelles méthodes créées.

ContrôleurType

Code commenté

```
public function getListeTypes() {
    $vListeTypes = $this->modeleTyp->getListeTypes();
    include 'Vue/VueListeTypes.php';
}

public function getListeEmplacementsByType($idType) {
    $vListeEmplacements = $this->modeleEmpl->getListeEmplacementsByType($idType);
    include 'Vue/VueListeEmplacements.php';
}
```

ContrôleurEmplacement

Code commenté

```
public function getListeEmplacements() {
    $vListeEmplacements = $this->modeleEmpl->getListeEmplacements();
    include 'Vue/VueListeEmplacements.php';
}

public function getEmplacement($idEmpl) {
    $vEmpl = $this->modeleEmpl->getEmplacement($idEmpl);
    include 'Vue/VueEmplacement.php';
}
```

Routeur

Code commenté

```
case 'emplacementRead' :    if (isset($_GET['id'])) {
    $ctrlEmpl = new ControleurEmplacement();
    $ctrlEmpl->getEmplacement($_GET['id']);
    }
    else { // s'il manque le paramètre id alors on affiche la liste des Emplacements
    $ctrlEmpl = new ControleurEmplacement();
    $ctrlEmpl->getListeEmplacements();
    }
    break;

case 'typeRead' :    if (isset($_GET['id'])) {
    $ctrlTyp = new ControleurType();
    $ctrlTyp->getListeEmplacementsByType($_GET['id']);
    }
    else { // s'il manque le paramètre id alors on affiche la liste des types
    $ctrlTyp = new ControleurType();
    $ctrlTyp->getListeTypes();
    }
    break;

default:
    // pour toutes les autres valeurs, on affiche la liste des Types
    $ctrlTyp = new ControleurType();
    $ctrlTyp->getListeTypes();
    break;
}

// aucune paramètre 'route' : on affiche la liste des Types
else {
    $ctrlTyp = new ControleurType();
    $ctrlTyp->getListeTypes();
}
```

De même pour le routeur, le code suit le même enchainement que le code fournit en exemple mais adapté aux fonctions créées précédemment dans les classes modèles.

Vue

VueEmplacement

Code commenté

```
<?php
echo 'id Emplacement : ' . $vEmpl->idEmpl. '<br/>';
echo 'adresse Emplacement : ' . $vEmpl->adresseEmpl. '<br/>';
echo 'annee Construction : ' . $vEmpl->anneeConstruction. '<br/>';
echo 'id Type : ' . $vEmpl->idType. '<br/>';
?>
```

Pour le bon affichage d'un emplacement, on adapte les attributs à ceux de l'emplacement de la fonction `getEmplacement($idEmpl)` de la classe `ControleurEmplacement`.

Rendu

← → ↻	Sécurisé https://webdev.iut-blagnac.fr/~mdeliot/PWS/TP/TP7/index.php?route=emplacementRead&id=231
id Emplacement : 231 adresse Emplacement : 5 rue des Fleurs annee Construction : 2003 id Type : 200	

VueListeEmplacements

Code commenté

```
foreach ($vListeEmplacements as $vEmpl) {
    // on génère une URL dynamique afin de pouvoir visualiser le
    // détail d'un emplacement
    echo '<tr><td><a href="index.php?route=emplacementRead&id='.$vEmpl->idEmpl.'">'.$vEmpl->idEmpl.'</a><
/td>';
    echo '<td>'.$vEmpl->adresseEmpl.'</td><td>'.$vEmpl->anneeConstruction.'</td><td>'.$vEmpl->idType.'</td>
</tr>';
}
```

De même que précédemment, on adapte les attributs à afficher par ceux qui correspondent à chaque emplacement et on modifie le lien qui permettra lorsque l'on clique sur l'id de l'emplacement d'en afficher seulement ses caractéristiques.

Rendu

← → ↻ Sécurisé | https://webdev.iut-blagnac.fr/~mdeliot/PWS/TP/TP7/index.php?route=typeRead&id=100

id Emplacement	adresse Emplacement	annee Construction	id Type
115	4 rue du Soleil	2012	100
198	7 rue des Fleurs	2008	100

VueListeTypes

Code commenté

```
<?php
    foreach ($vListeTypes as $vTyp) {
        echo '<tr><td><a href="index.php?route=typeRead&id='
            . $vTyp->idType . '">'. $vTyp->idType . '</a></td>';
        echo '<td>'. $vTyp->nomType . '</td></tr>';
    }
?>
```

On adapte juste les attributs à afficher à ceux des types et on adapte le lien pour que, lorsque l'on clique sur un id de type, on obtienne tous les emplacements correspondants.

Rendu



The screenshot shows a web browser window with the address bar displaying "Sécurisé | https://webdev.iut-blagnac.fr/~mdeliot/PWS/TP/TP7/". Below the address bar is a table with two columns: "Id Type" and "Nom Type". The table contains five rows of data, each with a blue underlined link in the "Id Type" column and a text label in the "Nom Type" column.

Id Type	Nom Type
100	Bungalow
200	Mobil-Home
300	Emplacement
400	Caravane
500	Villa

Conclusion :

A l'issus de ce Tp, nous avons acquis les bases de la mise en place d'un code qui respecte une structure en MVC.