

Aharouite Ahmed

Groupe 4

PWS / TP07

Table des matières

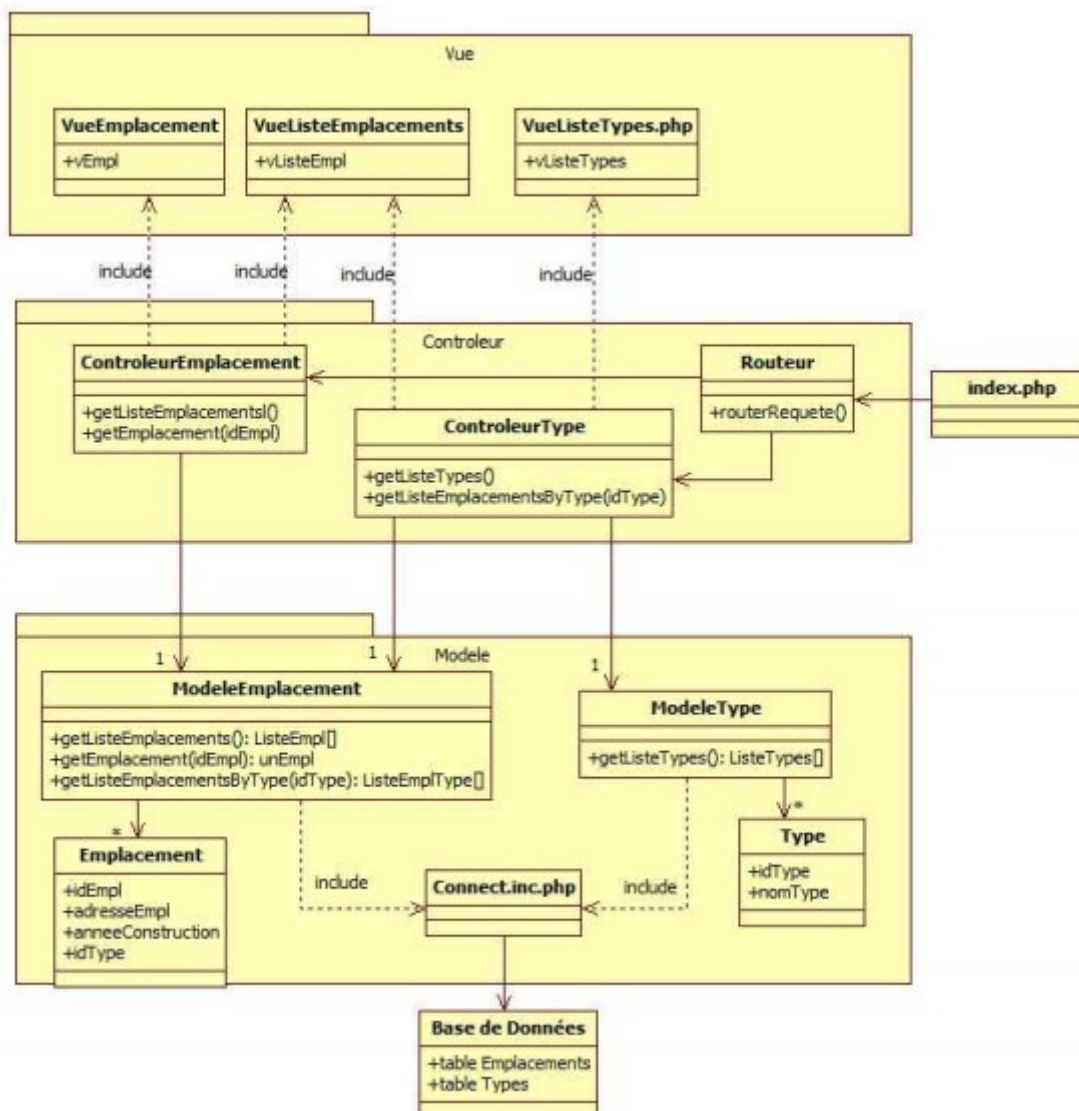
1. Introduction.....	2
2. Adaptation d'une application MVC.....	3
3. Exemple de scénario.....	6
4. Conclusion.....	7

1. Introduction

On veut adapter notre TD8 afin de pouvoir faire le même traitement pour les types (au lieu des groupes) et les emplacements (au lieu des étudiants). Nous procéderons de manière méthodique en changeant les différents noms d'abord dans la partie Modèle, puis dans la partie Contrôleur et enfin dans la partie Vue. Nous n'utiliserons pas la charte graphique, ni l'identification/session/cookie pour ce TP.

Lien du site : <http://193.54.227.74/~pws2001/PWS/TP/TP7/index.php>

Diagramme des classes :



2. Adaptation d'une application MVC

A) Exemple de vues

```
<!-- VueEmplacement.php -->
<html>
  <head></head>
  <body>
    <?php
      echo "id Emplacement : ".$vEmpl->idEmpl . "<br>";
      echo "id Type : ".$vEmpl->idType . "<br>";
      echo "adresse Emplacement : ".$vEmpl->adresseEmpl . "<br>";
      echo "année de Construction : ".$vEmpl->anneeConstruction . "<br>";
    ?>
  </body>
</html>
```

Résultat :

```
id Emplacement : 115
id Type : 100
adresse Emplacement : 4 rue du Soleil
année de Construction : 2012
```

Résultat :

```
<!-- VueListeEmplacements.php -->
<html>
  <head></head>
  <body>
    <?php
      // s'il y a des emplacements à afficher
      if ($vListeEmplType != null) {
        echo "<table border=2>";
        echo "<tbody><tr>
          <th>Id Empl</th>
          <th>Type de l'emplacement</th>
          <th>Adresse de l'emplacement</th>
          <th>Année de construction</th>
        </tbody></tr>";
        foreach ($vListeEmplType as $vEmpl) {
          // on génère une URL dynamique afin de pouvoir visualiser le
          // détail d'un emplacement
          echo "<tr>";
          echo "<td><a href='index.php?route=emplacementRead&id=".$vEmpl->idEmpl."'>".$vEmpl->idEmpl."</a></td>";
          echo "<td>".$vEmpl->idType."</td>";
          echo "<td>".$vEmpl->adresseEmpl."</td>";
          echo "<td>".$vEmpl->anneeConstruction."</td>";
          echo "</tr>";
        }
        echo "</table>";
      }
      else {
        echo "Pas d'emplacements...<BR>";
      }
    ?>
  </body>
</html>
```

Id Empl	Type de l'emplacement	Adresse de l'emplacement	Année de construction
115	100	4 rue du Soleil	2012
198	100	7 rue des Fleurs Jaunes	2008

B) Exemples de contrôleurs

```
<?php
// ControleurType.php
// Cette classe est créée et utilisée par le routeur.
// Elle utilise ModeleEmplacement.php pour récupérer les données et inclut ensuite
// VueListeEmplacements.php ou VueEmplacement.php pour afficher ces données.
include_once("Modele/ModeleType.php");
include_once("Modele/ModeleEmplacement.php");

class ControleurType {
    private $modeleType;
    private $modeleEmpl;

    public function __construct() {
        $this->modeleType = new ModeleType();
        $this->modeleEmpl = new ModeleEmplacement();
    }

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

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

```
class Routeur {
    // Route une requête entrante : exécution la bonne méthode de contrôleur en fonction de l'URL
    public function routerRequete() {
        // s'il y a un paramètre 'route'
        if (!empty($_GET['route'])) {
            // on détermine avec quelle méthode de quel contrôleur on veut travailler
            switch($_GET['route']) {
                case 'emplacementRead' : if (!empty($_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->getListeEmplacement();
                }
                break;

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

                default:
                    // pour toutes les autres valeurs, on affiche la liste des Types
                    $ctrlType = new ControleurType();
                    $ctrlType->getListeTypes();
                    break;
            }
        }
        // aucun paramètre 'route' : on affiche la liste des Types
        else {
            $ctrlType = new ControleurType();
            $ctrlType->getListeTypes();
        }
    }
}
```

C) Exemples de modèles

```
<?php
// ModeleType.php
// Cette classe offre des méthodes pour accéder aux données.
// Elle utilise et renvoie des objets de type Type
include_once("Type.php");
include_once("connect.inc.php");

class ModeleType {
    // methode qui renvoie un tableau d'objets Type
    public function getListeTypes() {
        // cette instruction permet d'utiliser dans cette fonction la variable
        // $conn qui a été initialisée dans le script connect.inc.php
        global $conn;
        $res = $conn->prepare("Select * from Type");
        $res->execute();
        foreach($res as $type) {
            $Listetype[] = new Type($type["idType"], $type["nomType"]);
        }
        return $Listetype;
    }

    public function getType($idType) {
        global $conn;
        $res = $conn->prepare("Select * from Type
        where idType = :pIdtypee");
        $res->execute(array('pIdtypee' => $idType));
        $typ = $res->fetch();
        $type = new Type($typ["idType"], $typ["nomtype"]);
        return $type;
    }
}
?>
```

```
<?php
// Type.php
// Cette classe servira à transferer, 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;
    }
}
?>
```

3. Exemple de scénario

Scénario : On veut afficher un tableau d'emplacements par défaut (index.php)

Premièrement, le routeur va « se mettre par défaut ». Il va appeler le contrôleur des types « ControleurType » qui lui par la suite va afficher la liste des types d'emplacements en forme de tableau grâce à une fonction et une vue.

➔ Routeur.php : Appelle getListeTypes() de ControleurType.php

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

➔ ControleurType.php : Appelle getListeTypes() de ModeleType.php et inclus une vue

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

➔ ModeleType.php (fonction) : renvoie un tableau de types à ControleurType.php

```
// methode qui renvoie un tableau d'objets Type
public function getListeTypes() {
    // cette instruction permet d'utiliser dans cette fonction la variable
    // $conn qui a été initialisée dans le script connect.inc.php
    global $conn;
    $res = $conn->prepare("Select * from Type");
    $res->execute();
    foreach($res as $type) {
        $Listetype[] = new Type($type["idType"], $type["nomType"]);
    }
    return $Listetype;
}
```

➔ Type.php : Le constructeur va permettre d'initialiser les types ajoutés dans \$Listetype.

```
<?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;
    }
}
```

➔ VueListe : (vue) Affichage du tableau en HTML.

```
<table border="2">
  <tbody>
    <tr>
      <th>Id Type</th>
      <th>Nom Type</th>
    </tr>
  </tbody>
</table>
<?php
  foreach ($vListeTypes as $vType) {
    echo "<tr>";
    echo "<td><a href='\"index.php?route=typeRead&id=\".$vType->idType.\"'>\".$vType->idType.</a></td>";
    echo "<td>\".$vType->nomType.</td>";
    echo "</tr>";
  }
?>
</table>
```

➔ Résultat : <http://193.54.227.74/~pws2001/PWS/TP/TP7/index.php>

Id Type	Nom Type
<u>100</u>	Bungalow
<u>200</u>	Mobil-Home
<u>300</u>	Emplacement
<u>400</u>	Caravane

4. Conclusion

Ce TP07 a pour but de mettre en application ce que nous avons déjà vu au cours des TD/TP et cours précédents , c'est à dire la manipulation du langage PHP et des applications MVC.

Durant ce TP, on a pu adapter notre site à une application MVC.

Ce TP07 nous a permis d'approfondir nos connaissances sur les applications MVC et sur leur exploitation grâce au langage PHP.