L3 MIASHS – Année 2018-2019
Programmation web
TD n° 5-6 : PHP, mise à jour de base de données, gabarits et architecture MVC
J. Darmont (http://eric.univ-lyon2.fr/~jdarmont/), 24/09/18

L'objectif de ce TD est de créer une interface de consultation et de mise à jour (saisie / modification / suppression) des données de la table MATIERE créée au TD n° 4, qui respecte de plus l'architecture MVC.

Page web modèle : http://eric.univ-lyon2.fr/~jdarmont/docs/web/td56.php

1. Recopier les fichiers td4.php, td4.class.php et td4.tpl.html sous les noms td56.php, td56modele.class.php et td56tab.tpl.html, respectivement.

2. Dans td56modele.class.php, remplacer la référence à td4.tpl.html par td56tab.tpl.html.

3. Dans td56.php :
    a. couper/coller les portions de code HTML situées avant et après le code PHP dans les fichiers td56header.inc.html et td56footer.inc.html, respectivement ;
    b. inclure ces deux fichiers en début et en fin de td56.php, respectivement ;
    c. remplacer la référence à td4.class.php par td56modele.class.php. Dans la partie résultats de requêtes, ne conserver que l'affichage de la table MATIERE.

4. Dans td5tab.tpl.html, ajouter un lien « Nouvelle matière » sous le tableau d'affichage. L'URL cible est un champ nommé {cible} assorti du paramètre « suivant=form_ajout ». Dans td56modele.class.php, ajouter à la méthode afficherTab() l'assignation à la variable cible de l'URL de la page courante (variable d'environnement $_SERVER["PHP_SELF"]).

5. Dans td56modele.class.php, créer une classe AccesMatiere contenant deux attributs privés $pdo (identifiant de connexion) et $qmat (objet de classe Requete), un constructeur qui initialise $pdo à un paramètre $param_pdo et $qmat à la requête qui affiche la liste des matières ; ainsi qu'une méthode publique nommée liste(). Couper/coller (et adapter si nécessaire) le code de création, d'exécution et d'affichage de la requête depuis td56.php. Vérifier le bon fonctionnement de la méthode liste() en créant dans td56.php un objet $accmat de classe AccesMatiere et en appelant la méthode liste() pour $accmat. Ensuite, supprimer l'appel à liste().

6. Dans un nouveau fichier td56form.tpl.html, écrire un formulaire dont la description suit.
    a. URL cible (attribut action) : champ {cible}, méthode : get
    b. Champ caché de nom « suivant » et de valeur le champ {suivant}
    c. Champ texte de nom « codemat » de taille 3 et de taille maximum 3
    d. Champ texte de nom « libelle » de taille 20 et de taille maximum 20
    e. Champ texte de nom « coef » de taille 3 et de taille maximum 3
    f. Boutons « reset » et « submit »

7. Dans un nouveau fichier td56controleur.class.php, créer une classe Appli contenant une méthode publique nommée formulaire() prenant en paramètre une page suivante. Cette méthode doit afficher le gabarit de formulaire td56form.tpl.html en associant au champ « cible » $_SERVER["PHP_SELF"] et au champ « suivant » le paramètre page suivante de la méthode. Inclure td56controleur.class.php dans td56.php.

8. Dans td56modele.class.php, ajouter à la classe AccesMatiere une méthode publique nommée ajouterMatiere() prenant en paramètres un code matière, un libellé et un coefficient. Cette méthode doit insérer les valeurs transmises par les paramètres dans la table MATIERE (utiliser une requête paramétrée).

9. Dans td56controleur.class.php, ajouter à la classe Appli une méthode publique nommée moteur() prenant en paramètre un objet $accmat de classe AccesMatiere. Dans la méthode moteur(), définir une variable $action égale à la page suivante demandée par l'utilisateur ($_GET["suivant"]) si cette dernière est définie, ou à une chaîne vide sinon. Pour cela, utiliser la fonction PHP isset(), qui retourne true si la variable passée en paramètre est définie.

10. Toujours dans la méthode moteur(), définir un « switch » sur la variable $action.
    a. Si l'action est « form_ajout », appeler la méthode formulaire() avec le paramètre « ajout » (qui représente la page suivante).
    b. Si l'action est « ajout », appeler la méthode ajouterMatiere() pour $accmat en lui passant en paramètres les données transmises depuis le formulaire à l'aide de la méthode « get ». Appeler ensuite la méthode liste() pour $accmat.
    c. L'action par défaut est l'affichage de la liste des matières, effectué en appelant la méthode liste() pour $accmat.

11. Dans td56.php, appeler la méthode moteur() pour $appli avec comme paramètre l'objet $accmat préalablement défini. Tester l'ajout de quelques matières.

12. Dans td56controleur.class.php, rendre la méthode formulaire() privée. Tester.

13. Dans td56tab.tpl.html, ajouter à chaque ligne du tableau une cellule contenant un lien « Modification ». L'URL cible est un champ {cible} assorti des paramètres « suivant=form_modif », « codemat={codemat} » (code de la matière courante), « libelle={libelle} » (libellé de la matière courante) et « coef={coef} » (coefficient de la matière courante).

14. Dans td56modele.class.php, modifier la méthode afficherTab() de la classe Requete comme suit : remplacer l'assignation de champ « rien » par l'assignation des champs « codemat », « libelle » et « coef » à $ligne["codemat"], $ligne["libelle"] et $ligne["coef"], respectivement. Des remarques ? Vérifier que le lien « Modification » s'affiche avec les bons paramètres d'URL.

15. Dans td56form.tpl.html, ajouter un champ caché de nom « codemat_actuel » et de valeur {codemat}. Dans la définition de chacun des champs « codemat », « libelle » et « coef », ajouter une valeur par défaut (value) égale à {codemat}, {libelle} et {coef}, respectivement.

16. Dans td56controleur.class.php, modifier la méthode formulaire() de la classe Appli en lui ajoutant trois paramètres ($codemat, $libelle et $coef), tous avec une valeur par défaut vide. Dans l'assignation des champs, associer aux champs « codemat », « libelle » et « coef » les paramètres $codemat, $libelle et $coef, respectivement.

17. Dans td56modele.class.php, ajouter à la classe AccesMatiere une méthode publique nommée modifierMatiere() prenant en paramètres un nouveau code matière, un nouveau libellé, un nouveau coefficient et le code matière actuel. Cette méthode doit modifier dans la table MATIERE les valeurs transmises par les paramètres pour le n-uplet correspondant au code matière actuel (utiliser une requête paramétrée).

18. Dans td56controleur.class.php, ajouter les deux options suivantes au switch de la méthode moteur().
    a. Si la page suivante est « form_modif », appeler la méthode formulaire() pour $accmat avec les paramètres « modif » et les données transmises par l'URL (code de matière, libellé et coefficient, dans cet ordre).
    b. Si la page suivante est « modif », appeler la méthode modifierMatiere() pour $accmat en lui passant en paramètres les données transmises depuis le formulaire à l'aide de la méthode « get ». Appeler ensuite la méthode liste() pour $accmat.

19. Tester la modification de quelques matières.

20. Dans td56tab.tpl.html, ajouter à chaque ligne du tableau une cellule contenant un lien « Suppression ». L'URL cible est un champ {cible} assorti des paramètres « suivant=suppr » et « codemat={codemat} » (code de la matière courante).

21. Dans td56modele.class.php, ajouter à la classe AccesMatiere une méthode publique nommée supprimerMatiere() prenant en paramètre un code matière. Cette méthode doit supprimer dans la table MATIERE le n-uplet correspondant au code matière (utiliser une requête paramétrée).

22. Dans td56controleur.class.php, ajouter l'option suivante au switch de la méthode moteur(). Si la page suivante est « suppr », appeler la méthode supprimerMatiere() pour $accmat en lui passant en paramètre le code de matière transmis par l'URL. Appeler ensuite la méthode liste() pour $accmat.

23. Tester la suppression de quelques matières.

24. Vérifier la validité du code HTML et CSS de tous les éléments de l'application (affichage de la table MATIERE et formulaire de saisie/modification, notamment).

25. L'utilisation de la méthode « get » est-elle la plus sûre ? Comment remédier à ses inconvénients ?

26. Récapituler en commentaire les fichiers qui appartiennent au modèle, à la vue et au contrôleur, respectivement.

Correction : Modèle

```
<?php // td56modele.class.php

    class Requete {
        private $pdo;   // Identifiant de connexion
        private $nom;   // Nom de la requête
        private $req;   // Requête à exécuter
        private $res;   // Résultat de la requête
        private $atts;  // Attributs à afficher

        function __construct($param_pdo, $param_nom, $param_req, $param_atts) {
            $this->pdo = $param_pdo;
            $this->nom = $param_nom;
            $this->req = $param_req;
            $this->atts = $param_atts;
        }

        public function executer() {
            $this->res = $this->pdo->prepare($this->req);
            $this->res->execute();
        }

        public function afficherTab() {
            // Définition du gabarit
            $gab = new Template("./");
            $gab->set_filenames(array("body" => "td56tab.tpl.html"));
            // Légende
            $gab->assign_vars(array("nom" => $this->nom));
            // Entête
            foreach($this->atts as $att)
                $gab->assign_block_vars("tableau", array("entete" => $att));
            // Données
            foreach ($this->res as $ligne) {
                $gab->assign_block_vars("ligne",
                                        array("codemat" => $ligne["codemat"],
                                              "libelle" => $ligne["libelle"],
                                              "coef" => $ligne["coef"]) );
                /* Question 14 : Il est dommage d'insérer des valeurs
                   spécifiques dans une méthode générique. Toutefois, cela
                   supprime la bidouille du champ vide qui permettait de boucler
                   sur les lignes dans le gabarit. */
                foreach($this->atts as $att)
                    $gab->assign_block_vars("ligne.attribut",
                                            array("valeur" => $ligne[$att]) );
            }
            // Lien nouvelle matière
            $gab->assign_vars(array("cible" => $_SERVER["PHP_SELF"]));
            // Affichage du gabarit
            $gab->pparse('body');
        }
    }

    class AccesMatiere {
        private $pdo;   // Identifiant de connexion
        private $qmat;  // Requête fixe liste des matières
```

```
        function __construct($param_pdo) {
            $this->pdo = $param_pdo;
            $this->qmat = new Requete($this->pdo,
                                      "Liste des matières",
                                      "SELECT * FROM MATIERE",
                                      array("codemat", "libelle", "coef")
            );
        }

        public function liste() {
            $this->qmat->executer();
            $this->qmat->afficherTab();
        }

        public function ajouterMatiere($codemat, $libelle, $coef) {
            $res = $this->pdo->prepare("INSERT INTO MATIERE VALUES(?, ?, ?)");
            $res->execute([$codemat, $libelle, $coef]);
        }
        public function modifierMatiere($codemat,
                                        $libelle,
                                        $coef,
                                        $codemat_actuel) {
            $res = $this->pdo->prepare("UPDATE MATIERE SET codemat = ?,
                                                           libelle = ?,
                                                           coef = ?
                                        WHERE codemat = ?");
            $res->execute([$codemat, $libelle, $coef, $codemat_actuel]);
        }

        public function supprimerMatiere($codemat) {
            $res = $this->pdo->prepare("DELETE FROM MATIERE WHERE codemat = ?");
            $res->execute([$codemat]);
        }
    }
?>
```

Correction : Vue

```
/* tableau.css */

body {
    font-family: arial;
}

table, th, td {
    border: 1px solid black;
}

table {
    border-collapse: collapse;
}

caption {
    font-style: italic;
}
```

```
<!DOCTYPE html> <!-- td56header.inc.html -->

<html lang="fr">

    <head>
        <meta charset="utf-8" />
        <meta name="Author" content="Jérôme Darmont" />
        <meta name="Keywords" content="Programmation,Web,PHP,MariaDB,MVC" />
        <meta name="Description" content="PHP objet, BD, MVC et gabarits" />
        <title>Mise à jour de base de données</title>
        <link rel="stylesheet" type="text/css" href="tableau.css" />
    </head>

    <body>


        <!-- td56footer.inc.html -->
        <p>Validation <a href="http://validator.w3.org/check/referer">HTML</a> |
          <a href="http://jigsaw.w3.org/css-validator/check/referer">CSS</a></p>
        </body>
</html>


<!-- td56tab.tpl.html -->
<table>
    <caption>{nom}</caption>
    <tr>
        <!-- BEGIN tableau -->
        <th>{tableau.entete}</th>
        <!-- END tableau -->
        <th>action</th>
        <th>action</th>
    </tr>
    <!-- BEGIN ligne -->
    <tr>
        <!-- BEGIN attribut -->
        <td>{ligne.attribut.valeur}</td>
        <!-- END attribut -->
        <td><a
href="{cible}?suivant=form_modif&amp;codemat={ligne.codemat}&amp;libelle={ligne.
libelle}&amp;coef={ligne.coef}">Modification</a></td>
        <td><a
href="{cible}?suivant=suppr&amp;codemat={ligne.codemat}">Suppression</a></td>
    </tr>
    <!-- END ligne -->
</table>
<p><a href="{cible}?suivant=form_ajout">Nouvelle matière</a></p> <hr />


<!-- td56form.tpl.html -->
<form action="{cible}" method="get">
    <fieldset>
        <input type="hidden" name="suivant" value="{suivant}" />
        <input type="hidden" name="codemat_actuel" value="{codemat}" />
        <p>
            <label for="codemat">Code matière</label>
            <input type="text" id="codemat" name="codemat" size="3"
                   maxlength="3" value="{codemat}" />
        </p>
        <p>
            <label for="libelle">Libellé</label>
            <input type="text" id="libelle" name="libelle" size="20"
                   maxlength="20" value="{libelle}" />
        </p>
```

```
        <p>
            <label for="coef">Coefficient</label>
            <input type="text" id="coef" name="coef" size="3" maxlength="3"
                   value="{coef}" />
        </p>
        <p>
            <input type="reset" value="Annuler" />
            <input type="submit" value="Valider" />
        </p>
    </fieldset>
</form>
```

Correction : Contrôleur

```
<?php
    // connect.inc.php
    $host = "localhost";
    $login = "darmont";
    $password = "XXX";
    $dbname = $login;
?>
```

```
<?php // td56controleur.class.php

    class Appli {

        private function formulaire($suivant,
                                    $codemat = "",
                                    $libelle = "",
                                    $coef = "") {
            // Définition du gabarit
            $gab = new Template("./");
            $gab->set_filenames(array("body" => "td56form.tpl.html"));
            // Assignation des valeurs des champs
            $gab->assign_vars(array("cible" => $_SERVER["PHP_SELF"],
                                    "suivant" => $suivant,
                                    "codemat" => $codemat,
                                    "libelle" => $libelle,
                                    "coef" => $coef) );
            // Affichage du gabarit
            $gab->pparse('body');
        }

        public function moteur($accmat) {
            if (isset($_GET["suivant"])) $action = $_GET["suivant"];
            else $action = "";
            switch ($action) {
                case "form_ajout": // Formulaire d'ajout
                    $this->formulaire("ajout");
                break;
                case "ajout"; // Ajout
                    $accmat->ajouterMatiere($_GET["codemat"],
                                            $_GET["libelle"],
                                            $_GET["coef"]);
                    $accmat->liste();
                break;
                case "form_modif": // Formulaire de modification
                    $this->formulaire("modif",
                                      $_GET["codemat"],
                                      $_GET["libelle"],
                                      $_GET["coef"]);
                break;
```

```
                case "modif": // Modification
                    $accmat->modifierMatiere($_GET["codemat"],
                                             $_GET["libelle"],
                                             $_GET["coef"],
                                             $_GET["codemat_actuel"]);
                    $accmat->liste();
                break;
                case "suppr": // Suppression
                    $accmat->supprimerMatiere($_GET["codemat"]);
                    $accmat->liste();
                    break;
                default: // Liste des matières
                    $accmat->liste();
                break;
            }
            /* Question 25 : L'utilisation de la méthode "get" fait apparaître
toutes les données en URL, ce qui la surcharge et peut être gênant lorsqu'on
transfère des données délicates comme des mots de passe. De plus, un utilisateur
mal intentionné peut s'en servir pour insérer des valeurs à sa convenance, voire
tenter de pirater le système par injection de code SQL malicieux. Il est donc
préférable d'utiliser la méthode "post", et donc de remplacer les liens
"Nouvelle matière", "Modification" et "Suppression" par des formulaires ne
contenant qu'un bouton. */
        }
    }
?>


<?php // td56.php
    include("td56header.inc.html");
    require("connect.inc.php");
    require("template.class.php");
    require("td56modele.class.php");
    require("td56controleur.class.php");
    try {
        // Connexion
        $c = new PDO("mysql:host=$host;dbname=$dbname", $login, $password);
        $c->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $c->exec("set names utf8");
        // Moteur de l'application
        $accmat = new AccesMatiere($c);
        $appli = new Appli($c);
        $appli->moteur($accmat);
s    } catch(PDOException $erreur) {
        echo "<p>Erreur : " . $erreur->getMessage() . "</p>\n";
    }
    include("td56footer.inc.html");
?>
```