

IUT de Montpellier - Semaine Web - TD5

Introduction à PHP - Modèle Vue Contrôleur

Semaine du 4 Novembre 2013

Si vous programmez de manière monolithique, vos pages actuelles mélangent traitement (couche d'accès aux données) et présentation (balises HTML) : Or, tout comme le copier/coller : *c'est le mal !*.

Découpez votre code en 3 parties :

produit.php est le modèle. Il contient la classe `Produit`, avec un constructeur (`__construct`) des getters, une méthode `save()` qui enregistre un article dans la base et une méthode statique `getTrucById($id)` qui charge un objet `Truc` depuis la base à partir de son identifiant.

Une fonction statique est une fonction dont le comportement ne dépend pas d'une instance de la classe (un objet) mais de la classe elle-même. On appelle une méthode statique de la façon suivante : `Truc::getTrucById(..)`

```
<?php
include 'connection.php';

class Truc {
    //les attributs:
    private $id;
    private $nom;
    private $prix;

    //les methodes:
    public function getPrix() { //un getter
        return $this->prix;
    }

    //Un constructeur
    public function __construct ($id,$nom, $prix) {
        $this->id = $id;
        $this->nom = $nom;
        $this->prix = $prix;
    }

    public function save() {
        $req = "INSERT INTO produit VALUES ('$id','$nom','$prix')";
        mysql_query($req) or die ("erreur insertion");
    }

    public static function getProduitById($id) { //une fonction statique
        $req = $SELECT * FROM produit WHERE id='$id'
        $res = mysql_query($req) or die ("erreur insertion");
```

```

        if (mysql_num_rows() == 0) //le produit n'existe pas
            return null;
        $tuple = mysql_fetch_array($res);
        $nom = $tuple['nom'];
        $prix = $tuple['prix'];
        return new Produit($id, $nom, $prix);
    }
}
?>

```

getProduit.php est le contrôleur.

```

<?php
    include ('produit.php'); //connait le modèle
    $id = $_GET['id']; //recupère le produit passé dans l'url
    $prod = Produit::getProduitById($id);
    if ($prod == null)
        require ('viewErrorProduit.php'); //redirige vers une vue d'erreur
    else
        require ('viewProduit.php'); //redirige vers la vue
    }
?>

```

viewProduit.php et **viewErrorProduit.php** sont des fichiers contenant quasiment exclusivement du code HTML, à l'exception de quelques *echo* permettant d'afficher les variables pré-remplies par le contrôleur.

Remarque : Il existe des variantes du MVC avec 4 ou même 5 couches. Mais le modèle à 3 couches permet déjà de dissocier le code métier de la présentation. Ce qui est l'essentiel.