

Modèle-Vue-Contrôleur (introduction)

Un exemple web avec PHP

monnerat@u-pec.fr

23 mai 2019

IUT de Fontainebleau

Introduction

Un exemple fait "main"

Codeigniter

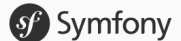
Introduction

Modèle MVC

Principe de conception d'applications (WEB ou pas) qui sépare les fonctions nécessaires en trois composants distincts :

Modèle		Gère les données
Vue		Gère la présentation (UI)
Contrôleur		Agit

Il existe de nombreux framework PHP basés sur cette architecture.



Gestion des données

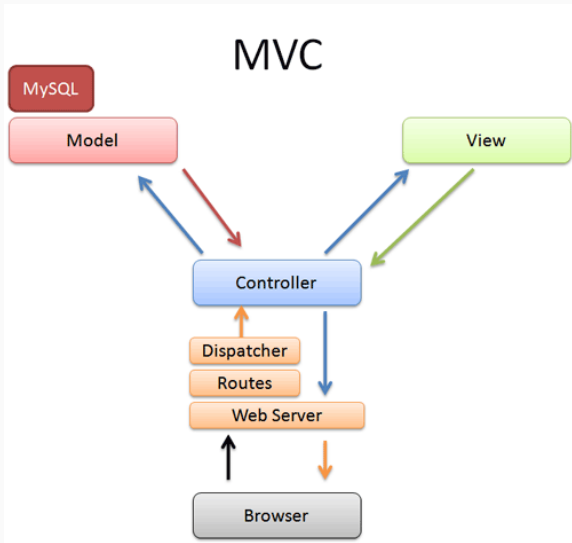
- Gère les échanges avec la base de données.
- Abstrait sous forme de classes les entités (les tables) de la bd.
- On peut utiliser un ORM (object-relational mapping) : doctrine, pdoMap, Propel, etc ...

Gestion de la présentation des données

- Affichage Html essentiellement (mais pas que).
- Avec des parties utilisant des variables php, remplis par le contrôleur à l'aide du Modèle (détail d'un film, etc ...).
- On peut utiliser des templates.

Gère les requêtes des utilisateurs, retourne une réponse avec l'aide mutuelle des couches Modèle et Vue.

- Le "cerveau" de l'application
- En fonction des requêtes du client, il fournit la bonne réponse en piochant dans le modèle et la vue pour afficher le bon résultat à l'utilisateur.
- Autant de classes que nécessaires, regroupés en entités logiques (Films, Utilisateurs, etc ...)
- Chaque contrôleur possède des actions : lister, insérer, supprimer, etc ...
- Chaque contrôleur a ses vues associées.



Modèle et Vue sont séparés

- L'application est plus flexible, plus claire, facilement évolutive.
- On peut les changer, remplacer, tester séparément.
- Le "design" peut-être confié à un designer.
- Abstraction de l'application par rapport aux données.

Un exemple fait "main"

Gestionnaire de contacts mail

Le but de concevoir une toute petite application web permettant de gérer (insertion, modification, édition, suppression) sa liste de contacts mails constitués d'un nom, prénom et d'une adresse mail.

Les plus visités ▾ Getting Started Latest Headlines ▾ PostBac 🇫🇷 Gestion Admission P... ADE test phpadmin

Listes des contacts

Prénom	Nom	Email	
Denis	Monnerat	monnerat@u-pec.fr	
Pierre	Valarcher	valarcher@u-pec.fr	

[page d'accueil](#)

Les routes de notre site

On choisit un système d'url segmenté de la forme

URL

`monsite.com/class/function/ID`

comme par exemple

Exemple

`monsite.com/contact/afficher/2`



Cela impose de configurer le serveur WEB (apache) et d'utiliser l'URL rewriting, qui transformera l'url précédente en

`monsite.com/index.php?query=contact/afficher/2`

.htaccess pour apache

Mise en place de la ré-écriture

```
RewriteEngine On
```

Adresse de base de réécriture

```
RewriteBase /~denis/contact/
```

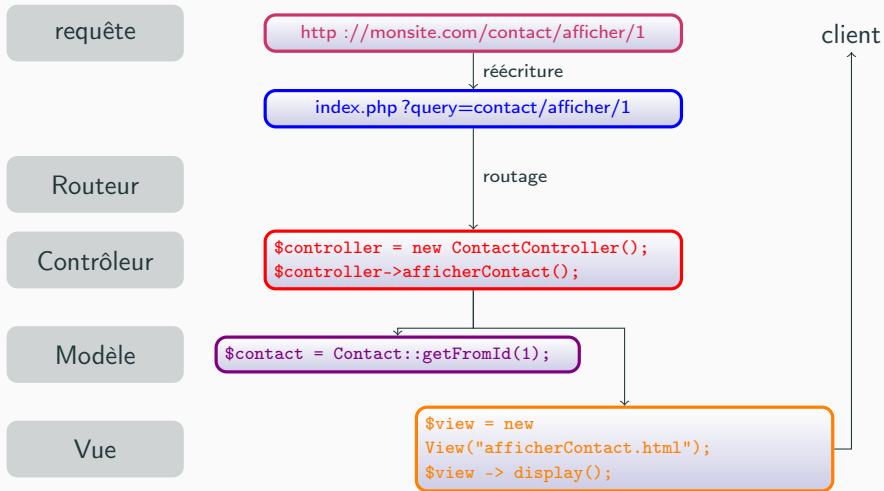
Règles

```
RewriteCond %{SCRIPT_FILENAME} !-f
```

```
RewriteCond %{SCRIPT_FILENAME} !-d
```

```
RewriteRule (.*?) index.php?query=$1 [QSA,L]
```

Une requête et son cheminement.



Point d'entrée : `index.php`.

Execution de `Kernel::run()`

- Analyse l'url (réécrite par apache).
- Initialise le bon contrôleur.
- Exécute la bonne méthode.

```
class Kernel {
    public static function run() {
        // Autoload des classes
        spl_autoload_register('Kernel::autoload');
        // Analyser la requete
        $query = isset($_GET["query"]) ? $_GET["query"] : "";
        $route = Router::analyze( $query );
        // Instancier le controleur et
        // executer l'action
        $class = $route["controller"]."Controller";
        if(class_exists($class)) {
            $controller = new $class ($route);
            $method = array($controller, $route["action"]);
            if( is_callable( $method ) )
                call_user_func($method);
        }
        // Gestion des erreurs
    }
}
```


Avec la méthode autoload qui permet d'inclure les fichiers en fonctions des besoins.

```
class Kernel {  
  
    public static function autoload($class) {  
        if(file_exists(ROOT."/app/kernel/$class.php"))  
            require_once(ROOT."/app/kernel/$class.php");  
        else if(file_exists(ROOT."/app/controller/$class.php"))  
            require_once(ROOT."/app/controller/$class.php");  
        else if(file_exists(ROOT."/app/model/$class.php"))  
            require_once(ROOT."/app/model/$class.php");  
    }  
}
```

Analyse de la route

Assuré par Router::analyze

Un extrait pour /afficher/contact/id

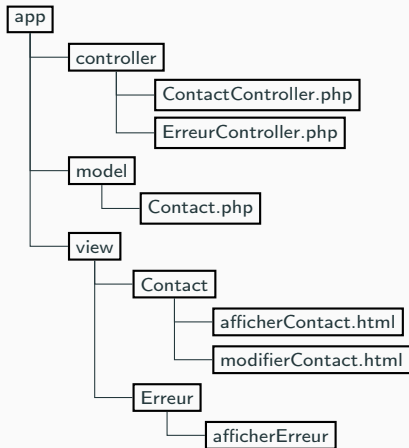
```
class Router {
    public static function analyze( $query ) {
        // un extrait
        $parts = explode("/", $query);

        if ((count($parts) == 3)
            && ($parts[1] == "afficher")
            && ($parts[0] == "contact")){

            $result["controller"] = "Contact";
            $result["action"] = "afficherContact";
            $result["params"]["id"] = $parts[2];
        }
        return $result;
    }
}
```

Strucutre des répertoires

En dehors de la racine du serveur (plus sûr).



Classe (patron : singleton) pour les accès à la base de données.

```
class Database {
    static protected $_instance = null;
    protected $_db;

    static public function getInstance() {
        if( is_null(self::$_instance) )
            self::$_instance = new Database();
        return self::$_instance;
    }
    public function query($sql){
        return $this->_db->query($sql);
    }
    public function prepare($sql){
        return $this->_db->prepare($sql);
    }

    protected function __construct() {
        $this->_db = new PDO(
            "mysql:host=localhost;dbname=mvc;charset=utf8",
            "root",
            "toto"
        );
    }
}
```

et les classes correspondantes aux entités.

```
class Contact extends Model {
    public $id, $nom, $prenom, $email;

    public static function getFromId( $id ) {
        $db = Database::getInstance();
        $sql = "SELECT * FROM contacts WHERE id = :id";
        $stmt = $db->prepare($sql);
        $stmt->setFetchMode(PDO::FETCH_CLASS, "Contact");
        $stmt->execute(array(":id" => $id));
        return $stmt->fetch();
    }

    public static function setFromId( $id,$data ) {
        $db = Database::getInstance();
        $sql = "UPDATE contacts set nom=:nom,prenom=:prenom,email=:email WHERE id = :id";
        $stmt = $db->prepare($sql);
        return $stmt->execute(array(
            ":id" => $id,
            ":nom"=>$data['nom'],
            ":prenom"=>$data['prenom'],
            ":email"=>$data['email']));
    }
}
```

```
class ContactController extends Controller {
  public function afficherContact() {
    $id = $this->route["params"]["id"];
    $this->view->contact = Contact::getFromId($id);
    $this->view->display();
  }
  public function modifierContact()
  {
    $id = $this->route["params"]["id"];
    $data = $this->route["params"]["post"];
    if (count($data) != 0) {
      $this->view->success = Contact::setFromId($id,$data) ;
    }
    $this->view->contact = Contact::getFromId($id);
    $this->view->display();
  }
}
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>Contacts</title>
</head>
<body>
  <header>
    <h1>Contact</h1>
  </header>
  <h2><?php echo ($this->contact->prenom)." ".($this->contact->
  <h3>Email : <?php echo $this->contact->email; ?></p>
  <footer>
    <p><a href="">retour à la page d'accueil</a></p>
  </footer>
</body>
</html>
```

Codeigniter

Codeigniter est un framework PHP MVC, très simple de prise en main et d'utilisation.

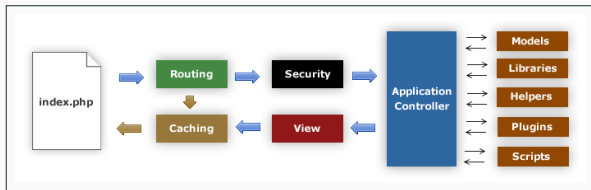
Site de CodeIgniter

Dernière version stable : version 3.1.10

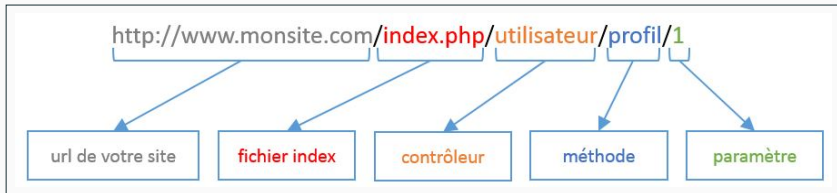
<http://www.codeigniter.com> 

- Instanciation dynamique des ressources.
- Indépendance des composants

Le fonctionnement général



1. index.php est le seul point d'entrée, initialise les ressources de bases utilisées par CI.
2. Le Routeur examine la requête HTTP pour savoir quoi faire.
3. Si un cache existe, il est renvoyé sans passer le système d'exécution normal.
4. Sécurité. Avant l'instanciation du bon contrôleur, la requête HTTP et toutes les informations soumises par l'utilisateur sont filtrés.
5. le contrôleur charge le modèle, les bibliothèques, les helpers, et toutes les ressources nécessaires pour la réponse.
6. La vue finale est calculée et envoyée. Si le cache est activé, la vue est mise en cache pour des requêtes futures.



On peut supprimer `index.php` en utilisant un mécanisme de réécriture d'url du serveur `http`.

Installation

Téléchargez CI (version 3.x), et désarchivez dans un répertoire servi par un serveur http. Vous obtenez l'arborescence suivante :

- **application**
Dossier où est codé l'application elle-même.
- **system**
C'est le dossier qui contient le cœur de CodeIgniter. Ne pas modifier son contenu !
- **user_guide**
la même doc que celle disponible en ligne. Vous pouvez supprimer ce dossier.
- **index.php**
c'est le fichier que l'internaute appellera lorsqu'il se connectera à votre site. Il est très important, à ne pas toucher.

Dossier application

C'est le dossier où est codé l'application. Voici son découpage avec les plus importants (pour le moment) :

- `config`
ce dossier contient la configuration du site.
- `controllers`
les contrôleurs.
- `helpers`
les helpers (fonction d'aides)
- `librairies`
bibliothèques utilisables par votre application.
- `models`
les modèles.
- `views`
les vues.

Le répertoire `config` contient :

- `autoload.php`
helpers et librairies chargés par défaut
- `config.php`
configuration de bases du serveur (url, protocoles, etc.)
- `database.php`
Accès bases de données.

L'exemple du gestionnaire de contacts

Gestion de contacts x

← → ↻ local host/~denis/CI314/index.php/ ☆ 🔔 👤 ⚙

Gestion des contacts

Ajouter

Nom	Prénom	Email	
Sloann	Julie	a@orciUT.ca	✕ 🔍
Griffin	Kalia	eros.nec.tellus@mollis.com	✕ 🔍
Osbornee	Dacey	tristique.pharetra@aliquamadipiscing.net	✕ 🔍
Sanders	Bradley	semper.et.lacinia@Donec.edu	✕ 🔍
Morton	May	mi.enim@necmetus.net	✕ 🔍
Singleton	Candice	ornare.lectus.justo@velitPellentesque.ca	✕ 🔍
Moss	Hamilton	Phasellus.at.augue@libero.co.uk	✕ 🔍
Rodriguez	Pearl	vulputate.posuere.vulputate@sem.co.uk	✕ 🔍
Solis	Charde	parturient.montes.nascetur@loremfringillaornare.ed	✕ 🔍
Lloyd	Hayes	augue@erosnectellus.co.uk	✕ 🔍
Compton	Signe	magna@tellusNunc.org	✕ 🔍
Caldwell	Denise	justo.faucibus@magnaNamligula.edu	✕ 🔍
Ramirez	Keefe	Vivamus.rhonus@Integerurna.org	✕ 🔍
Vasquez	Zachery	risus@enim.net	✕ 🔍

1 2 3 4 5 6 > »

🏠 🔔

Configuration

url de base dans config.php

```
/*
 * |-----
 * | Base Site URL
 * |-----
 * |
 * | URL to your CodeIgniter root. Typically this will be your base URL,
 * | WITH a trailing slash:
 * |
 * | slashhttp://example.com/
 * |
 * | WARNING: You MUST set this value!
 * |
 * | If it is not set, then CodeIgniter will try guess the protocol and path
 * | your installation, but due to security concerns the hostname will be set
 * | to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
 * | The auto-detection mechanism exists only for convenience during
 * | development and MUST NOT be used in production!
 * |
 * | If you need to allow multiple domains, remember that this file is still
 * | a PHP script and you can easily do that on your own.
 * |
 * */
$config['base_url'] = '~/denis/CI/';
```


Configuration

Accès bases de données dans database.php

```
$db['default'] = array(  
    'dsn'=> '',  
    'hostname' => 'localhost',  
    'username' => 'user',  
    'password' => 'user',  
    'database' => 'contacts',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,  
    'cachedir' => '',  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => '',  
    'encrypt' => FALSE,  
    'compress' => FALSE,  
    'stricton' => FALSE,  
    'failover' => array(),  
    'save_queries' => TRUE  
);
```

Chargement automatique dans autoload.php

```
/*  
| -----  
|  Auto-load Helper Files  
| -----  
| Prototype:  
|  
| $autoload['helper'] = array('url', 'file');  
*/  
$autoload['helper'] = array('url','html','download');
```

Routes dans routes.php

```
$route['default_controller'] = 'contacts';  
$route['contacts/(:any)'] = 'contacts/view/$1';
```

Le routage est implicite (mapping url-contrôleurs), mais on peut ajouter des routes manuellement ici.

Le code pour obtenir cette vue ?

Gestion de contacts x

localhost/~denis/CI314/index.php

Gestion des contacts

Ajouter

Nom	Prénom	Email	
Sloann	Julie	a@orciUt.ca	✕ 🔍
Griffin	Kalia	eros.nec.tellus@mollis.com	✕ 🔍
Osbornee	Dacey	tristique.pharetra@aliquamadipiscing.net	✕ 🔍
Sanders	Bradley	semper.et.lacinia@Donec.edu	✕ 🔍
Morton	May	mi.enim@necmetus.net	✕ 🔍
Singleton	Candice	ornare.lectus.justo@velitPellentesque.ca	✕ 🔍
Moss	Hamilton	Phasellus.at.augue@libero.co.uk	✕ 🔍
Rodriguez	Pearl	vulputate.posuere.vulputate@sem.co.uk	✕ 🔍
Solis	Charde	parturient.montes.nascetur@loremfringillaornare.ed	✕ 🔍
Lloyd	Hayes	augue@erosnectellus.co.uk	✕ 🔍
Compton	Signe	magna@tellusNunc.org	✕ 🔍
Caldwell	Denise	justo.faucibus@magnaNamligula.edu	✕ 🔍
Ramirez	Keefe	Vivamus.rhonus@Integerurna.org	✕ 🔍
Vasquez	Zachery	risus@enim.net	✕ 🔍

1 2 3 4 5 6 >>

Le modèle

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Model_contact extends CI_Model {
    public function __construct(){
        $this->load->database();
    }
    public function get_contact($id=null){
        if (isset($id)){
            $this->db->select('*')
                ->from('contacts')
                ->where('id', $id)
                ->limit(1);
            $query = $this->db->get();

            if($query->num_rows() == 1)
                return $query->row();
            else
                return false;
        }else{
            $this->db->select('*')
                ->from('contacts');
            $query = $this->db->get();
            return $query->result();
        }
    }
}
```

Le contrôleur

- Chargement du modèle.
- Récupération du tableau de tous les contacts.
- Chargement des vues, en passant le tableau précédent.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Contacts extends CI_Controller {

    public function view($id = null)
    {
        if ($id == null){
            $this->load->model('model_contact');
            $this->load->library('table');
            $contacts = $this->model_contact->get_contact();
            $data=array('contacts' => $contacts);
            $this->load->view('templates/header');
            $this->load->view('liste_contact', $data);
            $this->load->view('templates/footer');
        }
    }
}
```

L'entête

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title>Gestion de contacts avec Code igniter</title>
    <link rel="stylesheet" href="http://www.iut-fbleau.fr/css/concise.min.css">
    <link rel="stylesheet" href="http://www.iut-fbleau.fr/css/concise-utils/concise-utils">
    <link rel="stylesheet" href="http://www.iut-fbleau.fr/css/concise-ui/concise-ui.min.c">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/f">
    <?link_tag('assets/css/style.css')?>
  </head>
  <body container>
    <h1 class="_bb1 _mbxs">Gestion des contacts</h1>
```

Le tableau des contacts est passé à la vue par le contrôleur.

```
<p class="text-right">
<?=anchor('contacts/create', 'Ajouter')?>
</p>
<?php
$this->table->set_heading(array('Nom', 'Prénom', 'Email',''));
$template = array(
    'table_open' => '<table class="table table-striped  table-condensed">'
);
$this->table->set_template($template);
foreach ($contacts as $contact){
    $this->table->add_row(
        $contact->nom,
        $contact->prenom,
        $contact->email,
        array(
            'data'=>anchor('contacts/delete/'.$contact->id,
            '<i class="icon-remove-sign"></i>',
            array('class'=>'btn btn-danger'))." ".
            anchor('contacts/edit/'.$contact->id,
            '<i class="icon-search"></i>',
            array('class'=>'btn btn-info')),
            'style'=>"text-align:right")
    );
}
```


Chargement

```
$this->load->helper('url');
```

ou dans le fichier de config `application/config/autoload`

- **URL helper** : construction d'urls relatives au site.
- **HTML helper** : génération de code html.
- **SECURITY helper** : fonctions liées à la sécurité.
- etc.

```
<?=anchor('news/local/123', 'My News', 'title="News title"');?>  
// Prints:  
// <a href="http://example.com/index.php/news/local/123" title="News title">My News</a>
```

CI vient avec des librairies stockées dans `systemlibraries`.

```
$this->load->library('class_name');
```

- **Form Validation** : validation de formulaires.
- **Input Class** : entrées (requêtes, cookies, etc.) et filtrage automatique.
- **Session Library** : gestion des sessions.
- **File Uploading Class** : upload de fichiers.
- **Security Class** : sanitization des données, CSRF protection.
- etc.