

# Premier projet Phalcon

## -- Création

A partir de votre IDE, créer un nouveau projet PHP, en prenant soin de spécifier un dossier appartenant à la racine du serveur web (htdocs ou www).

## -- Structure des fichiers

Créer l'arborescence suivante :

```
firstPhalcon/  
  app/  
    controllers/  
    models/  
    views/  
  public/  
    css/  
    img/  
    js/
```

## -- URLs et sécurité

L'accès direct aux dossiers différents de public sera interdit. On utilise le mod-rewrite de apache (qu'il faut éventuellement activer dans http.conf), et on établit la réécriture d'url via les fichiers htaccess suivants :

Toutes les URLs sont redirigées vers le dossier public (les autres dossiers sont inaccessibles, donc public est le root de l'application)

```
<IfModule mod_rewrite.c>  
  RewriteEngine on  
  RewriteRule ^$ public/ [L]  
  RewriteRule (.*?) public/$1 [L]  
</IfModule>
```

Le second fichier htaccess indique que pour les fichiers existants de public, il n'y a pas de réécriture. Pour tous les autres, c'est le fichier index.php qui gère est qui est le point d'entrée de l'application.

```
AddDefaultCharset UTF-8  
  
<IfModule mod_rewrite.c>  
  RewriteEngine On  
  RewriteCond %{REQUEST_FILENAME} !-d  
  RewriteCond %{REQUEST_FILENAME} !-f  
  RewriteRule ^(.*)$ index.php?url=/$1 [QSA,L]  
</IfModule>
```

## -- Mise en place de MVC

Créer le fichier index.php dans public :

```
<?php

try {

    //Enregistrement d'un autoloader pour permettre l'inclusion auto des fichiers
    de classe
    $loader = new \Phalcon\Loader();
    $loader->registerDirs(array(
        '../app/controllers/',
        '../app/models/'
    ))->register();

    //DI est le service responsable de l'injection de dépendance des services
    Phalcon utilisés
    $di = new Phalcon\DI\FactoryDefault();

    //Configuration des vues
    $di->set('view', function(){
        $view = new \Phalcon\Mvc\View();
        $view->setViewsDir('../app/views/');
        return $view;
    });

    //Configuration de l'URL de base
    $di->set('url', function(){
        $url = new \Phalcon\Mvc\Url();
        $url->setBaseUri('/firstPhalcon/');
        return $url;
    });

    //Interception de la requête pour routage et dispatching
    $application = new \Phalcon\Mvc\Application($di);

    echo $application->handle()->getContent();

} catch(\Phalcon\Exception $e) {
    echo "PhalconException: ", $e->getMessage();
}
```

L'architecture MVC est déjà mise en place et prête à fonctionner (en quelques 10 lignes).

## -- Premier contrôleur

Dans app/controllers, créer le fichier IndexController.php :

```
<?php

class IndexController extends \Phalcon\Mvc\Controller{

    public function indexAction()
    {
        echo "<h1>Hello world !</h1>";
    }

}
```

Ce contrôleur devient le contrôleur par défaut pour les requêtes vers les URLs:

- /
- /index



# Hello world !

## -- Première vue

Créer la vue app/views/index/index.phtml, qui sera appelée automatiquement par le contrôleur **IndexController** :

```
<?php
echo "<h1>Hello world !</h1>";
echo Phalcon\Tag::linkTo("inscription","Inscrivez-vous...");
```

L'appel de la méthode **linkTo** sur la classe **Tag** insère un lien hypertexte pointant vers le contrôleur **InscriptionController** qui n'existe pas encore.

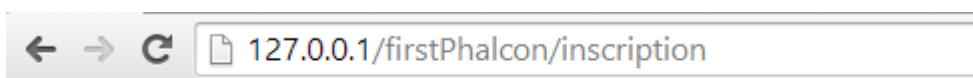
Tester la page, et cliquer sur le lien :



# Hello world !

[Inscrivez-vous...](#)

Résultat du click sur **Inscrivez-vous** :



PhalconException: InscriptionController handler class cannot be loaded

Ajouter le code pour éviter cette erreur

## -- Premier formulaire

Créer le contrôleur manquant : app/controllers/InscriptionController.php :

```
<?php
class InscriptionController extends \Phalcon\Mvc\Controller
{

    public function indexAction()
    {

    }

}
```

Créer le formulaire app/views/inscription/index.phtml :

```
<?php use Phalcon\Tag; ?>

<h2>Inscription</h2>

<?php echo Tag::form("inscription/enregistrement"); ?>

<p>
    <label for="nom">Nom : </label>
    <?php echo Tag::textField(array("nom","required")) ?>
</p>

<p>
    <label for="email">Email : </label>
    <?php echo Tag::emailField("email") ?>
</p>

<p>
    <?php echo Tag::submitButton("Enregistrement") ?>
</p>

</form>
```

La validation HTML par défaut est bien présente :

## Inscription

Nom :

Email :

! Veuillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "myaddressma".

Mais la validation du formulaire correctement renseigné signale l'absence de l'action **enregistrement** du contrôleur **InscriptionController**.

PhalconException: Action 'enregistrement' was not found on handler 'inscription'

### -- Premier modèle

#### -- Création de la base

Créer la base de données Mysql **firstPhalconDb**.

Créer la table users :

```
CREATE TABLE `users` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `nom` varchar(70) NOT NULL,  
  `email` varchar(70) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

#### -- Création d'un modèle

```
<?php  
class Users extends \Phalcon\Mvc\Model{  
  
}
```

#### -- Connexion à la base

```
...  
$di = new Phalcon\DI\FactoryDefault();
```

```
//Configuration du service database
$di->set('db', function(){
    return new \Phalcon\Db\Adapter\Pdo\Mysql(array(
        "host" => "localhost",
        "username" => "root",
        "password" => "",
        "dbname" => "firstPhalconDb"
    ));
});
...

```

## -- Validation du formulaire et enregistrement

Ajouter la méthode suivante dans **InscriptionController**, correspondant à l'url : **inscription/enregistrement**

```
<?php
class InscriptionController extends \Phalcon\Mvc\Controller
{

    public function indexAction(){

    }

    public function enregistrementAction(){
        $user = new Users();

        //Enregistrement et vérification des erreurs
        $success = $user->save($this->request->getPost(), array('nom', 'email'));

        if ($success) {
            echo "Merci de votre enregistrement !";
        } else {
            echo "Erreurs lors de l'enregistrement : ";
            foreach ($user->getMessages() as $message) {
                echo $message->getMessage(), "<br/>";
            }
        }
        $this->view->disable();
    }
}

```

Tester le formulaire d'ajout, et vérifier l'insertion du nouvel utilisateur dans la base de données.

## Liste des utilisateurs

En utilisant la méthode find() de la classe Model :

Afficher la liste des utilisateurs à partir de l'action index

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/slam4/php/phalcon/first>

Last update: **2019/08/31 14:21**

