

 eBook Gratuit

APPRENEZ

cakephp-3.0

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#cakephp-

3.0

Table des matières

À propos	1
Chapitre 1: Démarrer avec cakephp-3.0	2
Remarques.....	2
Exemples.....	2
Installation Cakephp 3.X.....	2
Mise en place du projet.....	3
Construis d'abord 'Hello World!' application avec CakePHP 3.x (Introduction. Partie 1).....	3
Construis d'abord 'Hello World!' application avec CakePHP 3.x (migrations de tables de bas.....	4
Construis d'abord 'Hello World!' application avec CakePHP 3.x (Contrôleurs, Réponse, Affic.....	6
Installer CakePHP 3.4 sur CentOS 7 avec PHP 7 et SELinux activés.....	6
Chapitre 2: Pagination personnalisée Ajax dans cakephp 3.2	9
Exemples.....	9
Pagination personnalisée Ajax dans cakephp 3.2.....	9
Crédits	12

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [cakephp-3-0](#)

It is an unofficial and free cakephp-3.0 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official cakephp-3.0.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec cakephp-3.0

Remarques

Cette section fournit une vue d'ensemble de ce qu'est cakephp-3.0 et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans cakephp-3.0, et établir un lien avec les sujets connexes. La documentation de cakephp-3.0 étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Installation Cakephp 3.X

Exigences:

- PHP PHP 5.6.0 ou supérieur extension PHP (fonctionne par défaut dans WAMP / XAMPP, sinon activez-le)
- Intl extension PHP (Disponible dans WAMP / XAMPP, vous devez l'activer de php.ini)
- CakePHP fonctionnera sur une variété de serveurs Web tels que nginx,
- LightHTTPD ou Microsoft IIS.

Avant de commencer, vous devez vous assurer que votre version de PHP est à jour:

```
php -v
```

Utilisez Composer pour installer Cakephp 3 Framework,

Composer est une méthode officiellement prise en charge pour l'installation, alors téléchargez le compositeur sur, [Composer \(Windows / Linux / Mac\)](#)

Exécutez ceci pour installer Cakephp,

```
php composer.phar create-project --prefer-dist cakephp/app my_app_name
```

Une fois que Composer a fini de télécharger le squelette de l'application et la bibliothèque principale de CakePHP, vous devriez avoir une application CakePHP opérationnelle installée via Composer. Veillez à conserver les fichiers composer.json et composer.lock avec le reste de votre code source.

Ou suivez ce moyen le plus simple pour installer cakephp

Suivez les étapes ci-dessous,

1. allez dans ce [dépôt Git](#)
2. Téléchargez le gâteau / four pour une installation facile
3. Extraire le fichier zip dans le LOCALHOST
4. Donne l'autorisation 777 à ce dossier
5. Exécutez le fichier oven.php (disponible dans le dossier)
6. Une page s'ouvrira avec beaucoup d'options, choisissez l'option selon vos préférences
7. Cliquez sur l'image et asseyez-vous. Votre projet sera installé dans quelques minutes

exécuter:

```
bin/cake server
```

Par défaut, sans aucun argument fourni, cela servira votre application sur <http://localhost:8765/>

lancez ceci sur votre navigateur, <http://example.com/> ou <http://localhost:8765/> . À ce stade, vous recevrez la page d'accueil par défaut de CakePHP et un message vous indiquant l'état de votre connexion de base de données actuelle et vous êtes prêt pour votre première application.

Pour plus de détails sur l'installation et la configuration, suivez [Cakephp 3.X Installation](#)

Mise en place du projet

Au début, vous devez créer une base de données avec `mysql` , `phpMyAdmin` , `HeidiSQL` ou d'autres instruments pour travailler avec Database et permettre à l'utilisateur d'en créer un nouveau.

Après cette procédure, vous devez fournir un accès à la base de données pour le projet.

Vous devez aller dans le fichier `/path/to/your/project/config/app.php` , puis rechercher `Datasources default` . Dans ce tableau, vous devez modifier `localhost` (à la demande), `user` , `password` et `database` .

o sur votre navigateur et actualiser la page. La question de la base de données devrait disparaître et afficher `Green Tick` à gauche.

Terminé! Votre premier projet a été mis en place!

Construis d'abord 'Hello World!' application avec CakePHP 3.x (Introduction. Partie 1)

CakePHP 3.x a la capacité de faire `bake controllers` , des `models` , des `views` et d'autres objets définis par le framework.

Remarque : Si vous avez déjà `Laravel framework` `Laravel` , le composant `artisan` est similaire à celui de la `bake` .

L'application de `bake` est située dans le dossier `bin` ; Voici quelques-unes des commandes disponibles:

- `bin/cake bake shell %shellName%` - pour cuire ShellClass
- `bin/cake bake controller %controllerName%` - pour cuire la classe Controller
- `bin/cake bake model %modelName%` - pour cuire la classe Model + Entity
- `bin/cake bake view %viewName%` - pour cuire le modèle de vue
- `bin/cake bake all %className%` - pour cuire Controller, Model + Entity, View for developer.

Note : Vous ne pourrez pas faire de `bake model` si vous n'avez pas de tables dans votre base de données

Remarque : Si vous `bake all` composants, vous obtiendrez des `Controllers` avec des actions `CRUD` prédéfinies.

Construis d'abord 'Hello World!' application avec CakePHP 3.x (migrations de tables de base de données. Partie 2)

Vous pouvez facilement `create` tables pour votre base de données ou les `drop` si vous le souhaitez. Si vous souhaitez le faire, vous devriez apprendre à écrire des `Migrations` pour la base de données souhaitée.

Les fichiers de migration doivent être situés dans le dossier `config/Migrations` . Les noms de fichiers peuvent être dans les formats suivants:

- `YYYYMMDDHHIISS_(Create|Alter|Delete)AdministratorsTable.php`
- `(1-9){1,}__(Create|Alter|Delete)AdministratorsTable.php`

```
<?php
use Migrations\AbstractMigration;
use Cake\Log\Log;

/**
 * Class AdministratorsTableMigration
 */
class AdministratorsTableMigration extends AbstractMigration
{

    /**
     * @var string
     */
    private $_tableName;

    /**
     * @var string
     */
    private $_tablePrefix;

    public function init()
    {
        $this->_tableName = 'Administrators';
        $this->_tablePrefix = 'administrators';
    }

    public function up()
    {
        Log::info("Trying to create {$this->_tableName} table");
    }
}
```

```

$administratorsTable = $this->table($this->_tablePrefix);

if ($administratorsTable->exists()) {
    return Log::warning("Table {$this->_tableName} already exists");
}

$administratorsTable
->addPrimaryKey('id')
->addColumn('username', 'char', [
    'length' => 25,
    'null' => false
])
->addColumn('password', 'char', [
    'length' => 255,
    'null' => false
])
->addColumn('email', 'char', [
    'length' => 50,
    'null' => false
])
->addColumn('first_name', 'char', [
    'length' => 50,
    'null' => false
])
->addColumn('last_name', 'char', [
    'length' => 50,
    'null' => false
])
->addColumn('avatar', 'char', [
    'length' => 255,
    'default' => '/img/no-avatar.png'
])
->addColumn('active', 'boolean', [
    'default' => 0
])
->addTimestamps()
->create();

return Log::notice("Table {$this->_tableName} has been created");
}

public function down()
{
    if ($this->table($this->_tablePrefix)->exists()) {
        $this->table($this->_tablePrefix)->drop();
        return Log::info("Table {$this->_tableName} has been dropped");
    }

    return Log::warning("Table {$this->_tableName} does not exists");
}
}

```

Si vous souhaitez exécuter la migration, vous devez exécuter la commande suivante:

`bin/cake migrations migrate` pour créer une table (-s).

Si vous voulez annuler:

`bin/cake migrations rollback` - annule la dernière migration, où la fonction `drop()` existe

`bin/cake migrations (-t|--target) all - bin/cake migrations (-t|--target) all` les migrations, où la fonction `drop()` existe

Construis d'abord 'Hello World!' application avec CakePHP 3.x (Contrôleurs, Réponse, Affichage. Partie 3)

Vous voulez créer un contrôleur? Il y a 2 façons de le créer:

- Manuellement (vous serez obligé de créer manuellement un fichier de contrôleur dans `src/Controller`)
- Baked (Exécution du `bin/cake bake controller %controllerName%` depuis l'interface de ligne de commande)

Si vous voulez le créer manuellement, allez dans le dossier `src/Controller` et créez le fichier suivant:

```
([AZ]{1}[az]{1,})Controller.php
```

Dans ce contrôleur, vous devez définir `namespace`, qui sera utilisé:

```
<?php
namespace App\Controller;
```

Ensuite, vous devriez le nommer comme nom de fichier, ex. `AdminController`:

```
use App\Controller\AppController;

class AdminController extends AppController{}
```

Dans cette classe, vous devriez créer votre première méthode, ex. `login` :

```
public function login(){}
```

Si vous tapez votre navigateur: `http://{{project-name}}/admin/login`, une erreur de modèle manquant apparaîtra. Comment résoudre ce problème?

Vous devez créer sous le fichier `src/Template/Admin/login.ctp`.

Remarque : le caractère générique `*.ctp` est un fichier de gabarit utilisé pour transmettre / rendre les données que vous définissez via le contrôleur.

Dans ce fichier, tapez simplement 'Hello World!' où vous voulez, rafraîchir la page avec une erreur de modèle et vous obtiendrez votre `World`, qui vous salue!

Note : Par défaut, `src/Template/Layout/default.ctp` est rendu en tant que présentation, si vous n'en avez pas défini

Installer CakePHP 3.4 sur CentOS 7 avec PHP 7 et SELinux activés

C'est ce que j'ai fait pour installer CakePHP sur un nouveau CentOS 7 minimal installé

- Installation d'un CentOS-7-x86_64-Minimal-1611.iso dans VirtualBox, deux interfaces réseau: premier NAT, deuxième hôte uniquement
- définir ONBOOT = yes dans / etc / sysconfig / network-scripts / ifcfg-enp0s3
- redémarrer
- mise à jour de miam
- yum install net-tools (pour obtenir ifconfig et netstat)
- yum installer wget
- yum install yum-utils
- wget -q <http://rpms.remirepo.net/enterprise/remi-release-7.rpm>
- wget -q <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>
- rpm -Uvh remi-release-7.rpm epel-release-latest-7.noarch.rpm
- yum-config-manager --enable remi-php71
- yum installer php
- systemctl activer httpd
- systemctl start httpd
- firewall-cmd --permanent --zone = public --add-service = http
- firewall-cmd --permanent --zone = public --add-service = https
- firewall-cmd --reload
- yum installer httpd mariadb-server mariadb php phpmyadmin
- systemctl démarrer mariadb
- systemctl activer mariadb
- systemctl restart httpd
- yum installe php-mbstring php-intl
- mysql_secure_installation
- curl -s <https://getcomposer.org/installer> | php
- cd / var / www / html /
- php composer.phar create-project --prefer-dist cakephp / app MyApp
- apache chown: -R MyApp /
- Créer une base de données:

```
# mysql -u root -p
Enter password:

mysql> CREATE DATABASE mydb;
mysql> GRANT ALL ON mydb.* to 'myuser'@'localhost' IDENTIFIED BY '_password_';
mysql> FLUSH PRIVILEGES;
mysql> quit
```

- Créez le fichier /etc/httpd/conf.d/my_app.conf avec le contenu:

```
<VirtualHost *:80>
    ServerAdmin root@localhost
    ServerName cakephp.myapp.net
    DocumentRoot /var/www/html/MyApp
    <Directory /var/www/html/MyApp>
        Allowoverride All
    </Directory>
</VirtualHost>
```

- `cd / var / www / html / secure_logging; chcon -Rv --type = httpd_user_content_rw_t tmp`
- touchez `./autorelabel`; redémarrer
- Sur mon hôte, je modifie `/ etc / hosts` et j'entre la ligne (192.168.56.101 est l'adresse IP de l'hôte uniquement de ma VM) `192.168.56.101 cakephp.myapp.net`
- Surfez sur <http://cakephp.myapp.net/>
- À faire: Modifier le fichier de connexion à la base de données.

Lire Démarrer avec cakephp-3.0 en ligne: <https://riptutorial.com/fr/cakephp-3-0/topic/5785/demarrer-avec-cakephp-3-0>

Chapitre 2: Pagination personnalisée Ajax dans cakephp 3.2

Exemples

Pagination personnalisée Ajax dans cakephp 3.2

Ici, je vais expliquer la pagination personnalisée ajax dans cakephp 3.2. Celui-ci est plus facile à utiliser et compréhensible.

Faites ce code dans n'importe quelle fonction et tout contrôleur dont vous avez besoin.

Faire un appel ajax pour la pagination

```
<script type="text/javascript">

$(document).ready(function () {

    $("#count-order-tr").load(strUrl + "http://test.com/Appadmins/ajaxLoadSalesListing");
    //load initial records

    //executes code below when user click on pagination links
    $("#count-order-tr").on("click", ".pagination a", function (e) {

        e.preventDefault();

        var page = $(this).attr("data-page");

        $("#count-order-tr").load(strUrl + "Appadmins/ajaxLoadSalesListing", {"page":
page}, function (data) {

            });

        });

    });

});
```

Récupère la page actuelle demandée no (1,2,3 ...) sur ajax post

```
public function ajaxLoadSalesListing() {
    $this->viewBuilder()->layout('ajax');

    if(isset($_POST["page"])){

        $page_number = filter_var($_POST["page"], FILTER_SANITIZE_NUMBER_INT,
FILTER_FLAG_STRIP_HIGH); //filter number
```

```

    if(!is_numeric($page_number)){die('Invalid page number!');} //incase of invalid page
number
}else{
    $page_number = 1; //if there's no page number, set it to 1
}

$item_per_page=5;//total number of records in a page.I have assumed it here 5

$get_total_rows = $this->Models->find('all')->count(); //hold total records in variable

$total_pages = ceil($get_total_rows/$item_per_page);//count total number of pages

$this->paginate['limit'=>$item_per_page
,'page'=>$page_number,'order'=>['Models.id'=>'DESC']];//Here we can give page ,limit ,order
,conditions and contain also.

$lists=$this->paginate($this->Models);//this is important ,this will do all the stuffs.This
one is used for the paginations.

```

Ensuite, définissez vos données sur ajax ctp.

```

$this->set(compact('Lists','item_per_page','page_number','get_total_rows','total_pages'));}

```

Ne placez la partie de pagination que dans la page ajax ctp uniquement, de sorte qu'elle soit actualisée à chaque fois et charge toutes les données.

Maintenant, récupérez tous les codes dans la page ctp (ajax_load_sales_listing.ctp) et mettez ce code à l'intérieur après la boucle

Custom-> paginate_function (\$ item_per_page, \$ page_number, \$ get_total_rows, \$ total_pages);

Appelez ensuite cette fonction paginate_function à partir de l'assistant personnalisé.

```

function paginate_function($item_per_page, $current_page, $total_records, $total_pages){
    $pagination = '';
    if($total_pages > 0 && $total_pages != 1 && $current_page <= $total_pages){ //verify total
pages and current page number
        $pagination .= '<ul class="pagination">';

        $right_links    = $current_page + 3;
        $previous       = $current_page - 3; //previous link
        $next           = $current_page + 1; //next link
        $first_link     = true; //boolean var to decide our first link

        if($current_page > 1){
            $previous_link = ($previous==0)? 1: $previous;
            $pagination .= '<li class="first"><a href="javascript:;" data-page="1"
title="First">&laquo;</a></li>'; //first link
            $pagination .= '<li><a href="javascript:;" data-page="'.($current_page-1).' "
title="Previous">&lt;</a></li>'; //previous link
            for($i = ($current_page-2); $i < $current_page; $i++){ //Create left-hand side
links
                if($i > 0){
                    $pagination .= '<li><a href="#" data-page="'. $i.'"

```

```

title="Page'.'.si.'">'.'.si.'</a></li>';
    }
    }
    $first_link = false; //set first link to false
}

if($first_link){ //if current active page is first link
    $pagination .= '<li class="first active">'.'.current_page.'</li>';
}elseif($current_page == $total_pages){ //if it's the last active link
    $pagination .= '<li class="last active">'.'.current_page.'</li>';
}else{ //regular current link
    $pagination .= '<li class="active">'.'.current_page.'</li>';
}

for($i = $current_page+1; $i < $right_links ; $i++){ //create right-hand side links
    if($i<=$total_pages){
        $pagination .= '<li><a href="javascript:;" data-page="'.'.si.'" title="Page
'.'.si.'">'.'.si.'</a></li>';
    }
}
if($current_page < $total_pages){
    $next_link = ($i > $total_pages) ? $total_pages : $i;
    $pagination .= '<li><a href="javascript:;" data-page="'.'.($current_page+1).' "
title="Next">&gt;</a></li>'; //next link
    $pagination .= '<li class="last"><a href="javascript:;" data-
page="'.'.total_pages.'" title="Last">&raquo;</a></li>'; //last link
}

$pagination .= '</ul>';
}
return $pagination; }}//return pagination links

```

Puis pagination ensemble.

Lire Pagination personnalisée Ajax dans cakephp 3.2 en ligne: <https://riptutorial.com/fr/cakephp-3-0/topic/7112/pagination-personnalisee-ajax-dans-cakephp-3-2>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec cakephp-3.0	bikash.bilz , clemens.tewinkel , Community , Ken Y-N , Mr. J , Roman Kozin
2	Pagination personnalisée Ajax dans cakephp 3.2	sradha