



EBook Gratuito

APPENDIMENTO cakephp-3.0

Free unaffiliated eBook created from
Stack Overflow contributors.

#cakephp-

3.0

Sommario

Di.....	1
Capitolo 1: Iniziare con cakephp-3.0.....	2
Osservazioni.....	2
Examples.....	2
Installazione Cakephp 3.X.....	2
Impostazione del progetto.....	3
Costruisci prima "Hello World!" applicazione con CakePHP 3.x (Introduzione Parte 1).....	3
Costruisci prima "Hello World!" applicazione con CakePHP 3.x (Migrazioni delle tabelle di	4
Costruisci prima "Hello World!" applicazione con CakePHP 3.x (Controller, Risposta, Vista.....	5
Installazione di CakePHP 3.4 su CentOS 7 con PHP 7 e SELinux abilitato.....	6
Capitolo 2: Impaginazione personalizzata Ajax in cakephp 3.2.....	9
Examples.....	9
Impaginazione personalizzata Ajax in cakephp 3.2.....	9
Titoli di coda.....	12

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

Capitolo 1: Iniziare con cakephp-3.0

Osservazioni

Questa sezione fornisce una panoramica di cosa sia CakePHP-3.0 e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di cakephp-3.0 e collegarsi agli argomenti correlati. Poiché la documentazione di cakephp-3.0 è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione Cakephp 3.X

Requisiti:

- PHP 5.6.0 o versione successiva dell'estensione PHP mbstring (l'impostazione predefinita funziona in WAMP / XAMPP, in caso contrario abilitarlo)
- estensione PHP intl (disponibile in WAMP / XAMPP, devi abilitarlo da php.ini)
- CakePHP verrà eseguito su una varietà di server Web come nginx,
- LightHTTPD o Microsoft IIS.

Prima di iniziare dovresti assicurarti che la tua versione di PHP sia aggiornata:

```
php -v
```

Usa Composer per installare Cakephp 3 Framework,

Il Composer è ufficialmente un metodo supportato per l'installazione, quindi scarica compositore su, [Composer \(Windows / Linux / Mac\)](#)

Esegui questa operazione per installare cakephp,

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

Una volta che Composer ha terminato il download dello scheletro dell'applicazione e della libreria core CakePHP, è necessario installare un'applicazione CakePHP funzionante tramite Composer. Assicurati di mantenere i file composer.json e composer.lock con il resto del codice sorgente.

Oppure segui questo modo più semplice per installare cakephp

Segui i passaggi qui sotto,

1. vai a questo [repository Git](#)

2. Scarica la tortaDC / forno per una facile installazione
3. Estrai il file zip all'interno di LOCALHOST
4. Dare 777 autorizzazioni a quella cartella
5. Esegui il file oven.php (disponibile all'interno della cartella)
6. Si aprirà una pagina con molte opzioni, scegliere l'opzione secondo le proprie preferenze
7. Clicca sull'immagine e siediti. Il tuo progetto verrà installato tra un paio di minuti

eseguire:

```
bin/cake server
```

Per impostazione predefinita, senza argomenti forniti, questo servirà la tua applicazione su [http://localhost: 8765 /](http://localhost:8765/) .

spara al tuo browser, <http://example.com/> o [http:// localhost: 8765 /](http://localhost:8765/) . A questo punto, ti verrà presentata la casa predefinita di CakePHP e un messaggio che ti informa sullo stato della tua connessione al database corrente e sei pronto per la tua prima applicazione.

Per ulteriori dettagli su Installazione e configurazione, seguire l'installazione di [Cakephp 3.X](#).

Impostazione del progetto

All'inizio, dovresti creare un database con `mysql` , `phpMyAdmin` , `HeidiSQL` o un altro strumento per lavorare con il Database e consentire all'utente di crearne uno nuovo.

Dopo tale procedura, è necessario fornire l'accesso al database per il progetto.

Devi andare in `/path/to/your/project/config/app.php` , quindi cerca il `default Datasources` . In quell'array, è necessario cambiare `localhost` (su richiesta), `user` , `password` e `database` .

o sul browser e sulla pagina di aggiornamento. Il problema del DB dovrebbe sparire e mostrare il `Green Tick` sul lato sinistro.

Fatto! Il tuo primo progetto è stato creato!

Costruisci prima "Hello World!" applicazione con CakePHP 3.x (Introduzione Parte 1)

CakePHP 3.x ha la capacità di `bake controllers` , i `models` , le `views` e altri oggetti definiti dal framework.

Nota : se hai avuto esperienza con il framework di `Laravel` , il componente `artisan` è simile a `bake` .

L'applicazione di `bake` si trova nella cartella `bin` ; i seguenti sono alcuni dei comandi disponibili:

- `bin/cake bake shell %shellName%` - per cuocere `ShellClass`
- `bin/cake bake controller %controllerName%` - per cuocere `Controller Class`
- `bin/cake bake model %modelName%` - per cuocere `Model + Entity Class`

- `bin/cake bake view %viewName%` - to bake Visualizza modello
- `bin/cake bake all %className%` - per cuocere Controller, Modello + Entità, Visualizza per sviluppatore.

Nota : non sarà possibile `bake model` se non ci sono tabelle nel database

Nota : se si `bake all` componenti, si ottengono i `Controllers` con azioni `CRUD` predefinite.

Costruisci prima "Hello World!" applicazione con CakePHP 3.x (Migrazioni delle tabelle di database. Parte 2)

Puoi facilmente `create` tabelle per il tuo database o `drop` se vuoi. Se desideri farlo, dovresti imparare come scrivere le `Migrations` per il database desiderato.

I file di migrazione devono trovarsi nella cartella `config/Migrations` . I nomi dei file possono essere nei seguenti formati:

- `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");
}
}

```

Se si desidera eseguire la migrazione, è necessario eseguire il comando successivo:

`bin/cake migrations migrate` per creare table (-s).

Se si desidera eseguire il rollback:

`bin/cake migrations rollback` - ripristinerà l'ultima migrazione, dove esiste la funzione `drop()`

`bin/cake migrations (-t|--target) all` - ripristinerà tutte le migrazioni, dove esiste la funzione `drop()`

Costruisci prima "Hello World!" applicazione con CakePHP 3.x (Controller,

Risposta, Vista. Parte 3)

Vuoi creare un controller? Ci sono 2 modi per crearlo:

- Manualmente (sarai costretto a creare manualmente il file Controller in `src/Controller`)
- Baked (Comando `bin/cake bake controller %controllerName%` da CLI)

Se vuoi crearlo manualmente, vai alla cartella `src/Controller` e crea il file che segue il seguente modello:

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

In quel controller, dovresti definire `namespace` , che verrà usato:

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

Quindi dovresti nominarlo come nome file, es. AdminController:

```
use App\Controller\AppController;

class AdminController extends AppController{}
```

All'interno di questa classe, dovresti creare il tuo primo metodo, es. `login` :

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

Se si digiterà nel browser: `http://{{project-name}}/admin/login` verrà generato un errore del modello mancante. Come risolvere questo problema?

È necessario creare sotto il file `src/Template/Admin/ login.ctp` .

Nota : * .ctp carattere jolly - è un file modello di torta, che viene utilizzato per passare / eseguire il rendering dei dati che si imposta tramite il controller.

In quel file digita semplicemente "Hello World!" dove vuoi, aggiorna la pagina con errore del modello e otterrai il tuo `World` , che ti saluta!

Nota : per impostazione predefinita, `src/Template/Layout/default.ctp` viene visualizzato come layout, se non ne è stato definito uno

Installazione di CakePHP 3.4 su CentOS 7 con PHP 7 e SELinux abilitato

Questo è quello che ho fatto per installare CakePHP su un CentOS 7 appena installato

- Installato un CentOS-7-x86_64-Minimal-1611.iso in VirtualBox, due interfacce di rete: primo NAT, secondo solo host
- imposta ONBOOT = sì in `/etc/sysconfig/network-scripts/ifcfg-enp0s3`
- riavvio

- aggiornamento yum
- yum install net-tools (per ottenere ifconfig e netstat)
- yum installa 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 installa php
- systemctl abilita httpd
- systemctl avvia httpd
- firewall-cmd --permanent --zone = public --add-service = http
- firewall-cmd --permanent --zone = public --add-service = https
- firewall-cmd --reload
- yum install httpd mariadb-server mariadb php phpmyadmin
- systemctl start mariadb
- systemctl abilita mariadb
- systemctl restart httpd
- Installa 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
- chop apache: -R MyApp /
- Crea database:

```
# 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
```

- Crea il file /etc/httpd/conf.d/my_app.conf con il contenuto:

```
<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
- tocca /.autorelabel; riavvio
- Sul mio host modifico / etc / hosts e inserisco la riga (192.168.56.101 è l'indirizzo IP host-

only della mia VM) 192.168.56.101 cakephp.myapp.net

- Naviga su <http://cakephp.myapp.net/>
- ToDo: modifica il file di connessione al database.

Leggi Iniziare con cakephp-3.0 online: <https://riptutorial.com/it/cakephp-3-0/topic/5785/iniziare-con-cakephp-3-0>

Capitolo 2: Impaginazione personalizzata

Ajax in cakephp 3.2

Examples

Impaginazione personalizzata Ajax in cakephp 3.2

Qui ho intenzione di spiegare l'impaginazione personalizzata ajax in cakephp 3.2. Questo è più facile da usare e comprensibile.

Esegui questo codice all'interno di qualsiasi funzione e qualsiasi controller di cui hai bisogno.

Fai una chiamata ajax per l'impaginazione

```
<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) {

            });

        });

    });

});
```

Ottieni l'attuale pagina richiesta no (1,2,3 ...) su un post di posta elettronica

```
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.

```

Quindi imposta i tuoi dati su ajax ctp.

```

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

```

Metti la parte di impaginazione solo nella pagina ctp ajax, in modo che si aggiorni ogni volta e carichi tutti i dati.

Ora recupera tutti i codici nella pagina ctp (ajax_load_sales_listing.ctp) e inserisci questo codice al suo interno dopo il ciclo

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

Quindi chiama questa funzione paginate dall'assistente personalizzato.

```

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' . $i.'">' . $i.'"</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="'. $i.'" title="Page
' . $i.'">' . $i.'"</a></li>';
        }
    }
    if($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

```

Quindi impostare l'impaginazione.

Leggi Impaginazione personalizzata Ajax in cakephp 3.2 online: <https://riptutorial.com/it/cakephp-3-0/topic/7112/impaginazione-personalizzata-ajax-in-cakephp-3-2>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con cakephp-3.0	bikash.bilz , clemens.tewinkel , Community , Ken Y-N , Mr. J , Roman Kozin
2	Impaginazione personalizzata Ajax in cakephp 3.2	sradha