



Kostenloses eBook

LERNEN cakephp-3.0

Free unaffiliated eBook created from
Stack Overflow contributors.

#cakephp-

3.0

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit cakephp-3.0	2
Bemerkungen.....	2
Examples.....	2
Installation von Cakephp 3.X.....	2
Projekt einrichten.....	3
Baue zuerst "Hallo Welt!" Anwendung mit CakePHP 3.x (Einführung. Teil 1).....	3
Baue zuerst "Hallo Welt!" Anwendung mit CakePHP 3.x (Datenbank-Tabellenmigrationen. Teil 2).....	4
Baue zuerst "Hallo Welt!" Anwendung mit CakePHP 3.x (Controller, Response, View. Teil 3).....	6
Installation von CakePHP 3.4 auf CentOS 7 mit aktiviertem PHP 7 und SELinux.....	7
Kapitel 2: Ajax benutzerdefinierte Paginierung in cakephp 3.2	9
Examples.....	9
Ajax benutzerdefinierte Paginierung in cakephp 3.2.....	9
Credits	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

Kapitel 1: Erste Schritte mit cakephp-3.0

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was cakephp-3.0 ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in cakephp-3.0 erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für cakephp-3.0 neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation von Cakephp 3.X

Bedarf:

- PHP 5.6.0 oder höher mbstring PHP-Erweiterung (Standardeinstellung in WAMP / XAMPP, wenn nicht, dann aktivieren Sie es)
- intl PHP extension (Verfügbar in WAMP / XAMPP, müssen Sie es aktivieren aus php.ini)
- CakePHP kann auf verschiedenen Webservern wie Nginx,
- LightHTTPD oder Microsoft IIS.

Bevor Sie beginnen, sollten Sie sicherstellen, dass Ihre PHP-Version auf dem neuesten Stand ist:

```
php -v
```

Verwenden Sie Composer, um Cakephp 3 Framework zu installieren.

Composer ist offiziell unterstützte Methode für die Installation. Laden Sie Composer daher unter [Composer \(Windows / Linux / Mac\)](#) herunter.

Führen Sie dies aus, um cakephp zu installieren.

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

Nachdem Composer das Anwendungsskelett und die Kern-CakePHP-Bibliothek heruntergeladen hat, sollte eine funktionierende CakePHP-Anwendung über Composer installiert sein. Stellen Sie sicher, dass die Dateien composer.json und composer.lock zusammen mit dem restlichen Quellcode aufbewahrt werden.

Oder folgen Sie dieser einfachsten Methode, um cakephp zu installieren

Folgen Sie den unten aufgeführten Schritten.

1. Gehe zu diesem [Git Repository](#)
2. Laden Sie den cakeDC / Ofen für die einfache Installation herunter
3. Extrahieren Sie die ZIP-Datei im LOCALHOST
4. Erteilen Sie diesem Ordner die Berechtigung 777
5. Führen Sie die Datei oven.php aus (im Ordner verfügbar).
6. Eine Seite wird mit vielen Optionen geöffnet. Wählen Sie die gewünschte Option aus
7. Klicken Sie auf das Bild und lehnen Sie sich zurück. Ihr Projekt wird in wenigen Minuten installiert

ausführen:

```
bin/cake server
```

Standardmäßig wird dies ohne Angabe von Argumenten für Ihre Anwendung unter [http: // localhost: 8765 /](http://localhost:8765/) bereitgestellt.

feuern Sie dies an Ihrem Browser ab, <http://example.com/> oder [http: // localhost: 8765 /](http://localhost:8765/) . An dieser Stelle werden Sie mit der Standard-Startseite von CakePHP und einer Nachricht angezeigt, die Sie über den Status Ihrer aktuellen Datenbankverbindung informiert und Sie sind bereit für Ihre erste Anwendung.

Weitere Informationen zu Installation und Setup finden Sie unter [Cakephp 3.X Installation](#)

Projekt einrichten

Zunächst sollten Sie eine Datenbank mit `mysql` , `phpMyAdmin` , `HeidiSQL` oder einem anderen Instrument erstellen, um mit Database arbeiten zu können, und dem Benutzer das Erstellen neuer ermöglichen.

Nach dieser Prozedur sollten Sie Zugriff auf die Datenbank für das Projekt gewähren.

Sie müssen in die Datei `/path/to/your/project/config/app.php` und dann nach `Datasources default` suchen. In diesem Array müssen Sie `localhost` (bei Bedarf), `user` , `password` und `database` ändern.

o Zu Ihrem Browser und zur Aktualisierungsseite. Die DB-Ausgabe sollte verschwunden sein und auf der linken Seite das `Green Tick` zeigen.

Erledigt! Ihr erstes Projekt wurde eingerichtet!

Baue zuerst "Hallo Welt!" Anwendung mit CakePHP 3.x (Einführung. Teil 1)

CakePHP 3.x kann `controllers` , `models` , `views` und andere im Framework definierte Objekte `bake` .

Hinweis : Wenn Sie Erfahrung mit dem `Laravel` Framework haben, ähnelt die `artisan` dem `bake` .

Die `bake` Anwendung befindet sich im Ordner `bin` . Im Folgenden sind einige der verfügbaren Befehle aufgeführt:

- `bin/cake bake shell %shellName%` - um ShellClass zu backen
- `bin/cake bake controller %controllerName%` - um Controller-Klasse zu backen
- `bin/cake bake model %modelName%` - zum Backen von Model + Entity Class
- `bin/cake bake view %viewName%` - backen Sie die View-Vorlage
- `bin/cake bake all %className%` - um Controller, Model + Entity, View für Entwickler zu backen.

Hinweis : Wenn Sie keine Tabellen in Ihrer Datenbank haben, können Sie das `bake model` nicht `bake model`

Hinweis : Wenn Sie `bake all` Komponenten `bake all` , erhalten Sie Controllers mit vordefinierten `CRUD` Aktionen.

Baue zuerst "Hallo Welt!" Anwendung mit CakePHP 3.x (Datenbank-Tabellenmigrationen. Teil 2)

Sie können auf einfache Weise Tabellen für Ihre Datenbank `create` oder sie `drop` , wenn Sie möchten. Wenn Sie das möchten, sollten Sie lernen, `Migrations` für die gewünschte Datenbank zu schreiben.

Migrationsdateien müssen sich im Ordner `config/Migrations` . Dateinamen können in folgenden Formaten vorliegen:

- `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");
}
}

```

Wenn Sie die Migration ausführen möchten, müssen Sie den nächsten Befehl ausführen:

`bin/cake migrations migrate` , um eine Tabelle (-s) zu erstellen.

Wenn Sie einen Rollback ausführen möchten:

`bin/cake migrations rollback` - Die letzte Migration wird rückgängig gemacht, wenn die `drop()`

Funktion vorhanden ist

`bin/cake migrations (-t|--target) all` - setzt alle Migrationen zurück, bei denen die `drop()` Funktion vorhanden ist

Baue zuerst "Hallo Welt!" Anwendung mit CakePHP 3.x (Controller, Response, View. Teil 3)

Möchten Sie einen Controller erstellen? Es gibt zwei Möglichkeiten, es zu erstellen:

- Manuell (Sie müssen die Controller-Datei manuell in `src/Controller` erstellen.)
- Baked (Ausführen des Befehls `bin/cake bake controller %controllerName%` Befehlszeilenschnittstelle)

Wenn Sie es manuell erstellen möchten, gehen Sie in den Ordner `src/Controller` und erstellen Sie eine Datei mit folgendem nächsten Muster:

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

In diesem Controller müssen Sie den `namespace` definieren, der verwendet wird:

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

Dann sollten Sie es als Dateiname angeben, z. AdminiController:

```
use App\Controller\AppController;

class AdminController extends AppController{}
```

Innerhalb dieser Klasse sollten Sie Ihre erste Methode erstellen, z. `login` :

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

Wenn Sie in Ihren Browser `http://{{project-name}}/admin/login` eingeben: `http://{{project-name}}/admin/login` , wird eine Fehlermeldung wegen fehlender Vorlage ausgegeben. Wie löse ich dieses Problem?

Sie müssen unter `src/Template/Admin/ login.ctp` Datei erstellen.

Hinweis : * .ctp-Platzhalter - ist eine Cake-Vorlagendatei, die zum Weitergeben / Rendern von Daten verwendet wird, die Sie über den Controller einstellen.

Geben Sie in dieser Datei einfach "Hallo Welt!" wo Sie wollen, aktualisieren Sie die Seite mit Template-Fehler und Sie erhalten Ihre `World` , die Sie begrüßt!

Hinweis : Standardmäßig wird `src/Template/Layout/default.ctp` als Layout gerendert, wenn Sie keines definiert haben

Installation von CakePHP 3.4 auf CentOS 7 mit aktiviertem PHP 7 und SELinux

So habe ich CakePHP auf einem frisch installierten minimalen CentOS 7 installiert

- Installierte ein CentOS-7-x86_64-Minimal-1611.iso in VirtualBox, zwei Netzwerkschnittstellen: erstes NAT, zweites Host-Only
- Setzen Sie ONBOOT = yes in / etc / sysconfig / network-scripts / ifcfg-enp0s3
- Neustart
- yum Update
- yum install net-tools (um ifconfig und netstat zu erhalten)
- yum installiere wget
- yum installiere 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 - aktiviert den remi-php71
- yum install php
- systemctl httpd aktivieren
- systemctl start 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 aktivieren mariadb
- Starten Sie httpd neu
- yum install 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
- chown apache: -R MyApp /
- Datenbank erstellen:

```
# 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
```

- Erstellen Sie die Datei /etc/httpd/conf.d/my_app.conf mit Inhalt:

```
<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`
- `touch /.autorelabel`; Neustart
- Auf meinem Host bearbeite ich / etc / hosts und gebe die Zeile ein (192.168.56.101 ist die Host-Only-IP-Adresse meiner VM) `192.168.56.101 cakephp.myapp.net`
- Surfen Sie auf <http://cakephp.myapp.net/>
- Aufgabe: Datenbank-Verbindungsdatei bearbeiten.

Erste Schritte mit cakephp-3.0 online lesen: <https://riptutorial.com/de/cakephp-3-0/topic/5785/erste-schritte-mit-cakephp-3-0>

Kapitel 2: Ajax benutzerdefinierte Paginierung in cakephp 3.2

Examples

Ajax benutzerdefinierte Paginierung in cakephp 3.2

Hier erkläre ich die Ajax Custom Paginierung in Cakephp 3.2. Dieser ist benutzerfreundlicher und verständlicher.

Führen Sie diesen Code in jeder Funktion und jedem Controller aus, den Sie benötigen.

Machen Sie einen Ajax-Anruf für die Paginierung

```
<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) {

            });

        });

    });

});
```

Holen Sie sich die aktuell angeforderte Seite Nr. (1,2,3 ...) auf 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.

```

Setzen Sie dann Ihre Daten auf ajax ctp.

```

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

```

Setzen Sie den Seitenteil nur in eine ajax ctp-Seite, damit er jedes Mal aktualisiert wird und alle Daten geladen werden.

Rufen Sie nun alle Codes auf der ctp-Seite (ajax_load_sales_listing.ctp) ab und fügen Sie diesen Code nach der Schleife ein

```

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

```

Rufen Sie dann diese paginate_function vom benutzerdefinierten Helfer auf.

```

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

```

Dann Paginierung eingestellt.

Ajax benutzerdefinierte Paginierung in cakephp 3.2 online lesen:

<https://riptutorial.com/de/cakephp-3-0/topic/7112/ajax-benutzerdefinierte-paginierung-in-cakephp-3-2>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit cakephp-3.0	bikash.bilz , clemens.tewinkel , Community , Ken Y-N , Mr. J , Roman Kozin
2	Ajax benutzerdefinierte Paginierung in cakephp 3.2	sradha