FREE eBook

LEARNING cakephp-3.0

Free unaffiliated eBook created from
Stack Overflow contributors.

#cakephp-3.0

Table of Contents

About

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.

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

Chapter 1: Getting started with cakephp-3.0

Remarks

This section provides an overview of what cakephp-3.0 is, and why a developer might want to use it.

It should also mention any large subjects within cakephp-3.0, and link out to the related topics. Since the Documentation for cakephp-3.0 is new, you may need to create initial versions of those related topics.

Examples

Installation Cakephp 3.X

Requirements:

- PHP 5.6.0 or greater mbstring PHP extension (default works in WAMP/XAMPP, if not then enable it)
- intl PHP extension (Available in WAMP/XAMPP, you have to enable it from php.ini)
- CakePHP will run on a variety of web servers such as nginx,
- LightHTTPD, or Microsoft IIS.

Before starting you should make sure that your PHP version is up to date:

```
php -v
```

Use Composer to install Cakephp 3 Framework,

Composer is officially supported method for Installation, so download composer at, Composer (Windows/Linux/Mac)

Run this to Install cakephp,

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

Once Composer finishes downloading the application skeleton and the core CakePHP library, you should have a functioning CakePHP application installed via Composer. Be sure to keep the composer.json and composer.lock files with the rest of your source code.

Or follow this easyest way to install cakephp

Follow the bellow steps,

1. go to this Git Repository

2. Download the cakeDC/oven for easy installation
3. Extract the zip file inside the LOCALHOST
4. Give 777 permission to that folder
5. Run file oven.php (available inside the folder)
6. A page will open with lot of option, choose option as per your preference
7. Click on the image and sit back. Your project will be installed in couple of minutes

execute:

```
bin/cake server
```

By default, without any arguments provided, this will serve your application at http://localhost:8765/ .

fire this at your browser, http://example.com/ or http://localhost:8765/. At this point, you'll be presented with CakePHP's default home, and a message that tells you the status of your current database connection and you are ready for your first application.

For more details on Installation and Setup follow, Cakephp 3.X Installation

Setting up Project

At first, You should create database with `mysql`, `phpMyAdmin`, `HeidiSQL` or another instruments to work with Database and let user create new one.

After that procedure, You should provide access to database for project.

You need to go into file `/path/to/your/project/config/app.php`,then look up for `Datasources default`. In that array, You need to change `localhost` (on demand), `user`, `password` and `database`.

o to Your browser and refresh page. DB issue should gone and show `Green Tick` at left side.

Done! Your first project has been set up!

Build up first 'Hello World!' application with CakePHP 3.x (Introduction. Part 1)

CakePHP 3.x has the ability to `bake controllers`, `models`, `views` and other framework defined objects.

> **Note** : If you have had some experience with the `Laravel` framework, the `artisan` component is similar to `bake`.

The `bake` application is located in `bin` folder; the following are some of the available commands:

- `bin/cake bake shell %shellName%` - to bake ShellClass
- `bin/cake bake controller %controllerName%` - to bake Controller Class
- `bin/cake bake model %modelName%` - to bake Model + Entity Class
- `bin/cake bake view %viewName%` - to bake View template
- `bin/cake bake all %className%` - to bake Controller, Model + Entity, View for developer.

Note : You will not be able to `bake model` if you have no tables in your database

Note : If you `bake all` components, you will get `Controllers` with pre-defined `CRUD` actions.

Build up first 'Hello World!' application with CakePHP 3.x (Database tables migrations. Part 2)

You can easily `create` tables for Your database or `drop` them if You want. If You wish to do that, You should learn how to write `Migrations` for desired database.

Migrations files must be located in `config/Migrations` folder. File names can be in next formats:

- `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");
    }

}
```

If You want to run migration, You need to run next command:

`bin/cake migrations migrate` to create table(-s).

If You want to rollback:

`bin/cake migrations rollback` - will revert last migration, where `drop()` function exists

`bin/cake migrations (-t|--target) all` - will revert all migrations, where `drop()` function exists

Build up first 'Hello World!' application with CakePHP 3.x (Controllers , Response, View. Part 3)

Want to create a controller? There is 2 ways of creating it:

- Manually (You will be forced to manually create Controller file in `src/Controller`)
- Baked (Running `bin/cake bake controller %controllerName%` command from CLI)

If You want to create it manually, go to `src/Controller` folder and create file that following next pattern:

```
([A-Z]{1}[a-z]{1,})Controller.php
```

In that controller, You should define `namspace`, that will be used:

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

Then You should name it as filename, ex. AdminiController:

```
use App\Controller\AppController;

class AdminController extends AppController{}
```

Inside of this class, You should create Your first method, ex. `login`:

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

If You will type in Your browser : `http://{{project-name}}/admin/login` it will throw an error of missing template. How to solve this problem?

You need to create under `src/Template/Admin/ login.ctp` file.

> **Note** : *.ctp wildcard - is Cake Template file, that is using to pass/render data You setting through controller.

In that file just type 'Hello World!' where You want, refresh page with template error and You will get Your `World`, that greets You!

> **Note** : By default, `src/Template/Layout/default.ctp` is rendering as layout, if You didn't defines one

Installing CakePHP 3.4 on CentOS 7 with PHP 7 and SELinux enabled

This is what I did to install CakePHP on a fresh installed minimal CentOS 7

- Installed a CentOS-7-x86_64-Minimal-1611.iso in VirtualBox, two network interfaces: first NAT, second Host-Only
- set ONBOOT=yes in /etc/sysconfig/network-scripts/ifcfg-enp0s3
- reboot
- yum update
- yum install net-tools (to get ifconfig and netstat)
- yum install 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 install php
- systemctl enable httpd
- 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 enable mariadb
- systemctl restart httpd
- 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/
- Create 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
```

- Create file /etc/httpd/conf.d/my_app.conf with content:

```
<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; reboot
- On my host I edit /etc/hosts and enter the line (192.168.56.101 is the host-only-ip-address of my VM) `192.168.56.101 cakephp.myapp.net`
- Surf to http://cakephp.myapp.net/

- ToDo: Edit database-connection file.

Read Getting started with cakephp-3.0 online: https://riptutorial.com/cakephp-3-0/topic/5785/getting-started-with-cakephp-3-0

Chapter 2: Ajax custom pagination in cakephp 3.2

Examples

Ajax custom pagination in cakephp 3.2

Here i am going to explain the ajax custom pagination in cakephp 3.2. This one is easier to use and understandable.

Do this code inside any function and any controller you need.

Make an ajax call for the 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) {


        });

    });


});
```

Get the current requested page no (1,2,3...)on 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.
```

Then set your data to ajax ctp.

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

Put pagination part in ajax ctp page only ,so that it will refresh every time and load all the datas.

Now fetch all the codes in the ctp page(ajax_load_sales_listing.ctp) and put this code inside it after the loop

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

Then call this paginate_function from custom helper.

```
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="javscript:;" data-page="1"
title="First">&laquo;</a></li>'; //first link
        $pagination .= '<li><a href="javscript:;" data-page="'.($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){
            $pagination .= '<li><a href="javscript:;" data-page="'.$i.'" title="Page
'.$i.'">'.$i.'</a></li>';
        }
    }
    if($current_page < $total_pages){
            $next_link = ($i > $total_pages) ? $total_pages : $i;
            $pagination .= '<li><a href="javscript:;" data-page="'.($current_page+1).'"
title="Next">&gt;</a></li>'; //next link
            $pagination .= '<li class="last"><a href="javscript:;" data-
page="'.$total_pages.'" title="Last">&raquo;</a></li>'; //last link
    }

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

Then pagination set .

Read Ajax custom pagination in cakephp 3.2 online: https://riptutorial.com/cakephp-3-0/topic/7112/ajax-custom-pagination-in-cakephp-3-2

Credits

S. No	Chapters	Contributors
1	Getting started with cakephp-3.0	bikash.bilz, clemens.tewinkel, Community, Ken Y-N, Mr. J, Roman Kozin
2	Ajax custom pagination in cakephp 3.2	sradha