

فریم ورک Artemis

فریم ورک آرتمیس برپایه زبان PHP و بر اساس معماری MVC نوشته شده است.

کار با فریم ورک آرتمیس :

Controller: کنترلرها در واقع مغز متفکر معماری MVC هستند. کنترلرها در مسیر `app/controllers/` قرار میگیرند. نام فایل کنترلر باید به صورت حروف کوچک نوشته شده باشد.

مثلا میخواهیم یک کنترلر با نام `user` ایجاد کنیم . برای این کار در مسیر `app/controllers/` یک فایل با نام `user.php` میسازیم. نام کلاس کنترلرها باید با حرف بزرگ شروع بشه و کلمه کنترلر به انتهای آن اضافه شود :

```
class UserController
{
    //action
}
```

کلاسهای کنترلر باید از کلاس `Controller` ارث بری کنند.

```
class UserController extends Controller
{
    //action
}
```

کلاس **AppController** : گاهی نیاز داریم که یک تابع رو بتونیم در همه کنترلرها استفاده کنیم. برای این کار در مسیر `app/controllers` یک کلاس با نام `AppController` میسازیم که از کلاس کنترلر اصلی ارث بری میکند.

```
<?php
```

```
/**
 * app/controllers/AppController.php
 **/
class AppController extends Controller
{

}
```

بعد از اینکه این کلاس رو ایجاد کردیم کنترلرهایمون از این به بعد باید از کلاس `AppController` ارث بری کنند.

```
Class UserController extends AppController
{

    function index()
    {

        echo 'Hello Artemis Framework!';

    }

}
```

حالا هر متودی رو که در کلاس `AppController` ایجاد کنیم توی همه کنترلرها همیشه ازش استفاده کرد.

Models : مدلها در معماری MVC لایه کار با دیتابیس هستند. برای هر جدول از دیتابیس باید یک Model مجزا ساخت. برای ساخت Model در مسیر `app/model` یک فایل میسازیم. این فایل باید همانم با جدول مورد نظر در دیتابیس باشه.

مثال : جدولی داریم در دیتابیس با نام `post`. ساخت `model` برای این جدول به صورت زیره :

```
/**
 * app/model/posts.php
 */
class Posts extends AppModel
{
    protected $pk = 'pid';
}
```

به صورت بالا کلاس `Posts` رو میسازیم. نام کلاس باید همانم با جدولمون در دیتابیس باشه. و باید با حرف بزرگ شروع بشه.

این کلاس باید از کلاس `AppModel` ارث بری کنه.

```
protected $pk = 'pid';
```

در خط بالا ما مشخص میکنیم که فیلد اصلی جدول کدوم فیلده. در صورتی که نام فیلد اصلی `id` بود نیازی به تعیین فیلد اصلی نیست.

Views : `View` ها صفحه ای است که کاربر آنرا مشاهده میکند. `view` ها در مسیر `app/views` قرار میگیرند.

`View` ها باید داخل پوشه ای همانم با نام کنترلر قرار بگیرند. نام فایل هر `view` باید همانم `action` باشه.

به مثال زیر دقت کنید :

```
//app/controllers/user.php

class UserController extends ApplicationController
{
    public function index()
    {
        $this->view->hello = 'Hello World';
        $this->view->view();
    }
}
```

حالا در مسیر `app/views/User/` یک فایل با نام `index.php` ایجاد کنید به صورت زیر :

```
//app/views/User/index.php

<h1><?php echo $this->hello?></h1>
```

راه اندازی و کار با فریم ورک آرتمیس :

برای راه اندازی و کار ابتدا باید یه سری تنظیمات رو انجام بدید.اول فریم ورک آرتمیس رو از اینجا دانلود کنید.محتوای فایل فشره بعد از `extract` به شکل زیر خواهد بود :

```
artemis/
  app/
    controllers/
    layout/
    models/
```

views/
Elements/
Artemis/
components/
config/
helpers/
public/
css/
images/
js/
.htaccess
Index.php

پوشه app و محتویات اون رو قبلا توضیح دادم براتون.

در پوشه Artemis هسته فریم ورک آرتمیس قرار داره.

Components : کامپوننتها فقط در داخل controllerها قابل دسترسی هستند.

Config : حاوی فایل config.php که تنظیمات فریم ورک رو نگهداری میکنه.

Helpers : Helperها فقط در view ها قابل دسترسی هستند.

Public : فایلهای جاوااسکریپت و CSS و عکسها دراین پوشه قرار میگیره.

برای دسترسی به هر کدام از این پوشه ها یه سری ثابت تعریف شده :

IMG_FOLDER : مسیر پوشه images

JS_FOLDER : مسیر پوشه js

CSS_FOLDER : مسیر پوشه CSS

برای راه اندازی فایل config/config.php رو باز کنید :

```
<?php
/**
 * config/config.php
 */
$config['base'] = '/artemis/';
$config['base_url'] = 'http://127.0.0.1'.$config['base'];
//server db
$config['server'] = 'localhost';
//username db
$config['username'] = "Database Username";
//pass db
$config['password'] = 'Database Password';
$config['database'] = 'Database Name';
```

در قسمت `$config['base']` باید نام پوشه ای ذکر بشه که فریم ورک داخلش قرار داره. اگر فریم ورک در root اصلی باشه به صورت زیر تغییرش بدید :

```
$config['base'] = '/';
```

تنظیمات مربوط به کانکشن دیتابیس رو هم از همینجا میتونید انجام بدید.

حالا در مرور آدرس زیر رو وارد کنید :

<http://127.0.0.1/artemis/>

میبینید که صفحه اصلی این فریم ورک براتون میاد.

در مسیر `app/controllers/` یک فایل با نام `index.php` وجود داره. بازش کنید.

```
<?php

class IndexController extends ApplicationController
{
    function index()
    {
        $this->view->fr = 'Hello World!';
        $this->view->view();
    }
}
```

اگر شما در url نام کنترلر رو ذکر نکنید کنترلر پیش فرض `IndexController` و `action` پیش فرض `index()` در نظر گرفته میشود.

`$this->view->fr` متغیری است که در فایل `view` که در مسیر `app/views/Index/index.php` قرار گرفته به صورت زیر فراخوانی میشه :

```
echo $this->fr;
```

کار با دیتابیس :

ابتدا یک دیتابیس با یک جدول به نام `posts` با سه فیلد به نامهای `pid` , `title` , `body` ایجاد کنید. بعد تنظیمات مربوط در فایل `config.php` رو اعمال کنید.

برای دسترسی به جداول دیتابیس ، برای هر جدول باید یک کلاس همانام با نام جدول در پوشه `app/models` بسازید. نام فایل این کلاس باید با حروف کوچک نوشته بشه تا خود کلاس باید با حرف بزرگ شروع بشه و از کلاس `AppModel` ارث بری کنه.

```
<?php
// app/models/posts.php
class Posts extends AppModel
{
    //set primary key.
    protected $pk = 'pid';
}
```

`protected $pk = 'pid';` مشخص میکنه که کلید اصلی جدول کدوم فیلده.

برای دسترسی به جداول دیتابیس در کنترلرها باید اونا رو به شکل زیر در کنترلر مورد نظر بار گذاری کنیم :

```
class IndexController extends ApplicationController
{
    //load model
    protected $model = 'Posts';
    function index()
    {
        $this->view->fr = 'Hello World!';
        $this->view->view();
    }
}
```

در بالا `Posts` نام کلاس `model` میباشد.

اگر نیاز بود تعداد مدلهای بیشتر رو در کنترلر لود کنیم باید به صورت زیر عمل کنیم :

```
protected $model = array('Posts','User');
```

حالا متد `index` رو به شکل زیر تغییر بدید :

```
function index()
{
    $this->view->posts = $this->Posts->findAll();
    $this->view->view();
}
```

در اینجا ما تمام رکوردهای جدول `posts` رو در متغیر `posts` قرار دادیم.

حالا برای نمایش رکوردهای جدول به فایل `views/Index/index.php` میریم و فایل رو به صورت زیر تغییر میدیم :

```
<?php
foreach($this->posts as $post)
{
    echo '<a href="'.BASE_URL.'post/'.$post['id'].'">'. $post['title'].'</a><br>';}
?>
```

برای نمایش پستها در کنترلر `IndexController` یک اکشن با نام `post` میسازیم :

```
function post($id = 0)
{
    $this->view->post = $this->Posts->find()->where(array('pid'=>$id))->fetchOne();
    $this->view->view();
}
```

حالا برای نمایش این پست باید یک فایل با نام `post.php` در مسیر `app/controllers/Index` بسازیم.

```
<h1><?php echo $this->post['title'];?></h1>
<?php echo $this->post['body'];
```

کار با قالب

با طراحی یک قالب خواص میتونیم تمام صفحات سایتمون رو در قالب ایجاد شده نمایش بدیم.

کار با قالبها خیلی راحت و برای این کار فقط یک قالب طراحی کنید و در مسیر `app/layout/` قرار بدید.

ابتدا قالب مورد نظر خودتون رو طراحی کنید. سپس در مسیر `app/layout` یک پوشه خالی با نام دلخواه انتخاب کنید. این نام قالب مورد نظره که با استفاده از این نام میتونید بهش دسترسی داشته باشید. حالا قالب طراحی شده رو داخل این پوشه با نام `index.php` قرار بدید.

برای استفاده از این قالب در کنترلر مورد نظرتون به صورت زیر عمل کنید :

```
$this->view->theme = 'layout name';
```

در قسمت `layout name` نامی رو که برای قالبتون مشخص کردید بذارید.

اما در طراحی قالب باید قسمت اصلی که قرار محتویات سایت نمایش داده بشه رو با دستور

```
echo $this->content
```

مشخص کنید.

برای ویرایش قالب جاری که فریم ورک ازش استفاده میکنه به پوشه `app/layout/default/` برید و فایل `index.php` رو ویرایش کنید.

این یه آموزش کوچیک برای آشنایی با این فریم ورکه. این فریم ورک رو بنده در حدود ۲ ماه وقت گذاشتم و تقریبا آمادش کردم. برای ساخت این فریم ورک از بقیه فریم ورکهای معروف هم ایده گرفتم. به عنوان مثال بخش کار با دیتابیس ها رو از دوتا فریم ورک `cakephp` و `codeigniter` ایده گرفتم. به هر حال احساس میکنم که در مدتی که برای ساخت این فریم ورک وقت گذاشتم خیلی چیزا یاد گرفتم . امیدوارم شما دوست عزیز هم اگر مایل به گسترش این پروژه هستید اعلام کنید تا هماهنگی های لازم انجام بشه.

برای داشتن یک فریم ورک تقریبا کامل باید خیلی از مسایل رو رعایت کرد. که به دلیل نبود وقت زیاد نتونستم گسترشش بدم. مطمئنن ایرادها و کمبودهای زیادی داره که امیدوارم در طول زمان این مشکلات با یاری دوستانی که به php مسلط و یا دوستانی که علاقه مند هستن رفع بشه.

همین سایت phpro.ir رو بنده با همین فریم ورک آماده کردم.

منتظر نظرات و انتقادهای خوب و سازنده شما هستم .

PHPro.ir@gamil.com

با تشکر ۱۳۹۰/۸/۲۰

سعید مقدم زاده.