ساخت لینکدونی با CakePHP (بخش دوم)

پیش نیاز: ساخت لینکدونی با CakePHP (بخش اول)

در پست قبل تا ساخت پایگاه داده و مقداردهی فایل database.php بمنظور ارتباط با MySQL پیش رفتیم. حال یک جدول بنام links با شش فیلد زیر در دیتابیس linkdump که قبلاً ساخته‌ایم ایجاد می‌کنیم. همچنین جهت تست دو لینک اضافه می‌کنیم:

```
1  /* First, create our links table: */
2  CREATE TABLE links (
3    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
4    url VARCHAR(255),
5    title VARCHAR(255),
6    body TEXT,
7    created DATETIME DEFAULT NULL,
8    modified DATETIME DEFAULT NULL
9  );
10
11 /* Then insert some links for testing: */
12 INSERT INTO links (url, title, body, created) VALUES
13 ('http://alvanweb.com', 'Alvanweb', 'All about web design and progrmming', NOW()),
14 ('http://cakephp.org', 'CakePHP', 'The rapid development php framework', NOW());
15
```

نام جدول و ستون‌ها اختیاری نیستند! اگر از قواعد نامگذاری جداول و کلاس‌های کیک پیروی کنید خواهید توانست از مزایای توابع توکار کیک بدون پیکربندی براحتی استفاده کنید. کیک تا انجا انعطاف پذیر است که حتی با وجود عدم تطابق با طرح‌های پیشفرض می‌توانید آن را سفارشی کنید اما استفاده از قواعد در صرفه‌جویی زمان موثر است.

برای اطلاعات بیشتر در مورد این قواعد نامگذاری اینجا را ببینید. اما همین اندازه کافی است که بدانید جدول linksبطور خودکار به مدلی بنام link اشاره دارد و همچنین فیلدهای created و modified بطور منطقی و خودکار توسط کیک مدیریت می‌شوند.

ایجاد یک مدل

به سراغ کدنویسی کیک می‌رویم. اولین فایلی که باید ایجاد کنیم یک مدل برای لینک‌ها است. به زبان ساده بگویم مدل در واقع نان و کره‌ی یک برنامه نوشته شده با کیک است. با ایجاد یک مدل با دیتابیس ارتباط برقرار کرده‌ایم. در ابتدا پایه کد را می‌نویسیم سپس اعمالی نظیر نمایش، افزودن، ویرایش و حذف لینک را اضافه خواهیم کرد.

کلاسی که به عنوان مدل تعریف می‌شود در **app/models** قرار می‌گیرد و محتویات فایلی که در **app/models/link.php** ذخیره خواهد شد بصورت زیر است:

```
<?php
class Link extends AppModel {
   var $name = 'Link';
}
?>
```

یادتان نرود قواعد نامگذاری در کیک بسیار مهم است. با نامیدن مدل بنام **Link** کیک بطور خودکار می‌تواند تشخیص دهد که این مدل توسط کنترلری بنام **LinksController** بکار گرفته خواهد شد و با جدولی بنام **links** در دیتابیس در ارتباط است.

اگر کیک نتواند فایلی مطابق با مدل در **app/models** بیابد، بطور داینامیک یک آبجکت مدل ایجاد می‌کند. این همچنین به این معناست که اگر بطور تصادفی نام فایل مدل را اشتباه تایپ کنید (مثلا بجای **link.php** نام فایل **links.php** شود) کیک هیچ کدام از تنظیمات شما را لحاظ نمی کند و در عوض مقادیر پیشفرض خودش را جانشین خواهد کرد.

ایجاد یک کنترل‌کننده

در این مرحله برای لینک هایمان یک کنترلر ایجاد می‌کنیم. کنترلر جایی است که قصد داریم جزئیات لینک ها را استخراج کنیم. فایلی بنام **links_controller.php** را با محتویات پایه‌ی زیر در **app/controllers** ایجاد می‌کنیم:

```
<?php
class LinksController extends AppController {
  var $name = 'Links';
}
?>
```

حال اجازه دهید یک کنش (**Action**) به کنترلر بیفزایم. کنش اغلب به یک تابع در برنامه اشاره دارد. برای مثال وقتی کاربران درخواست **example.com/cake/links/index** را می‌دهند (که با **example.com/cake/links** مشابه است) انتظار دارند با لیستی از لینک‌ها مواجه شوند. کدی که این کنش را تعریف می‌کند بصورت زیر است:

```
<?php
class LinksController extends AppController {
  var $name = 'Links';
  function index() {
    $this->set('links', $this->Link->find('all'));
  }
}
?>
```

با تعریف تابع **index** در **LinksController**، کاربران می‌توانند به درخواستی مانند این **example.com/cake/links/index** دسترسی داشته باشند. بطور مشابه اگر تابعی بنام **foobar** تعریف کنیم کاربران قادر خواهند بود به درخواست **example.com/links/foobar** دسترسی داشته باشند.

ممکن است کنجکاو شوید تا به طریقی نام‌های کنترلر و کنش را تغییر دهید. فعلاً دست نگهدارید و از قواعد کیک بمنظور ایجاد کنش‌های قابل فهم استفاده کنید. در آینده بدین منظور با **routes** در کیک آشنا خواهید شد.

در این کنش از تابع **set** برای گذر داده از کنترل‌کننده به نمایش (**view**) استفاده می‌کنیم. این خط، متغیر **links** را برابر مقدار بازگشتی از find('all') تنظیم می‌کند. مدل لینک بطور خودکار در $this->Link موجود خواهد بود چراکه از قواعد نامگذاری کیک استفاده کردیم.

ایجاد یک نما

تا هم اکنون با جریان داده به مدل، منطق برنامه و روند تعریف شده توسط کنترلر آشنا شدید حال اجازه دهید یک نما برای کنش **index** که در بالا تعریف شد ایجاد کنیم. نماهای کیک در واقع نمایش اجزایی هستند که در طرح‌بندی (**layout**) برنامه کنار هم قرار میگیرند. برای بیشتر برنامه‌ها **HTML** مخلوط با **PHP** استفاده می‌شود اما می تواند بصورت **XML**، **CSV** یا حتی داده باینری باشد.

آخرین قسمت عنوان قبل را بیاد آورید، که چگونه متغیر **links** را با متد **set** به نما ارجاع می‌دادیم؟ خروجی آرایه را می‌توانیم به صورت زیر نشان دهیم:

```
// print_r($links) output:
Array
(
 [0] => Array
  (
          [Link] => Array
           (
           [id] => 1
           [url] => http://alvanweb.com
           [title] => Alvanweb
 [body] => All about web design and progrmming
           [created] => 2008-10-22 20:29:21
           [modified] =>
           )
 )
 [1] => Array
  (
          Link] => Array
           (
           [id] => 2
           [url] => http://cakephp.org
           [title] => CakePHP
 [body] => The rapid development php framework
           [created] => 2008-10-22 20:29:21
           [modified] =>
           )
  )
)
```

فایل‌های مربوط به نما در **app/views** در فولدری همنام با واژه‌ای که در کنترلر تعریف می‌شود، ذخیره می‌شوند. (در اینجا می‌بایستی نام فولدر ما **links** باشد) برای فرم‌دادن به داده‌ی لینک‌هایمان در قالب یک جدول زیبا(ایجاد صفحه‌ای برای کنش **index**)، از کد زیر استفاده می‌کنیم:

```
/app/views/links/index.ctp
<h1>Links:</h1>
<table>
<tr>
<th>Id</th>
<th>Title</th>
<th>Created</th>
</tr>
< ?php foreach ($links as $link): ?>
<tr>
<td>< ?php echo $link['Link']['id']; ?></td>
<td>< ?php echo $html->link($link['Link']['title'],"/links/view/".$link['Link']['id']); ?></td>
<td>< ?php echo $link['Link']['created']; ?></td>
</tr>
< ?php endforeach; ?>
</table>
```

احتمالاً متوجه استفاده از آبجکتی بنام **$html** شده‌اید. این یک نمونه از کلاس **HtmlHelper** کیک است. کیک پی‌اچ‌پی مجموعه‌ای از این آبجکت‌ها را تحت نام **view helpers** در خود گنجانده است مواردی از قبیل لینک‌گذاری، خروجی فرم، جاوااسکریپت و آژاکس از این دسته‌اند. اطلاعات تکمیلی درمورد کار با آنها را می‌توانید اینجا بیابید. اما انچه در اینجا مهم است که بدان اشاره شود متد **link** است که بوسیله‌ی پارامترهایی که گرفته است، یک لینک **HTML** را ایجاد خواهد کرد. در این مرحله می بایستی بتوانید در مرورگرتان مسیر **example.com/cake/links/index**را ببینید.

اگر بر روی هر یک از لینک‌ها کلیک کنید تا توضیحاتش را ببینید با صفحه‌ی خطایی مواجه خواهید شد مبنی بر این که کنش مربوطه در فایل کنترل‌کننده هنوز تعریف نشده است. بنابراین مجدداً به فایل **links_controller.php** بر می‌گردیم تا یک کنش بنام **view** بمنظور نمایش اطلاعات هر لینک ایجاد کنیم:

```
<?php
class LinksController extends AppController {
  var $name = 'Links';
  function index() {
    $this->set('links', $this->Link->find('all'));
  }
  function view($id = null) {
    $this->Link->id = $id;
    $this->set('link', $this->Link->read());
  }
}
?>
```

خوب با **set** که از قبل آشنایی دارید اما نکته‌ای که در اینجا لازم است به آن اشاره کنیم این است که بجای **find('all')** از **read** استفاده کردیم چراکه قصد داریم اطلاعات تنها یک لینک را استخراج کنیم.

دقت کنید کنش **view** یک پارامتر می‌گیرد که برابر با **ID** هر لینک است. این پارامتر از طریق درخواست از طریق **URL** مقداردهی می‌شود بطوریکه اگر کاربری **links/view/2** را درخواست دهد، مقدار **2** برابر **ID** خواهد بود. حال اجازه دهید نمای **view** را برای این کنش در **app/views/links** بصورت زیر ایجاد کنیم:

```
/app/views/links/view.ctp
<h1>< ?php echo $html->link($link['Link']['title'],
$link['Link']['url']); ?></h1>
<p><small>Created: < ?php echo $link['Link']['created']?></small></p>
<p>< ?php echo $link['Link']['body']?></p>
```

حال با کلیک بر روی هر لینک اطلاعات مربوط به آن لینک در صفحه‌ای دیگر نمایش داده خواهد شد. در پست بعدی سایر مراحل را دنبال خواهیم کرد.