آموزش کاربردی

CakePHP

مهندس محمد مرادی

انتشارات پندار پارس

سرشناسه	:	مرادی، محمد، ۱۳۶۶ -
عنوان و نام پدیدآور	:	آموزش کاربردی CakePHP/ محمد مرادی.
مشخصات نشر	:	تهران : پندار پارس، ۱۳۹۱.
مشخصات ظاهری	:	۲۴۸ ص.: مصور، جدول.
شابک	:	978-600-6529-25-7: ۱۲۰۰۰۰ ریال
وضعیت فهرست نویسی	:	فیپا
یادداشت	:	کتابنامه.
موضوع	:	پی.اچ.پی. (زبان برنامه‌نویسی کامپیوتر) -- دستنامه‌ها
موضوع	:	اینترنت -- برنامه‌نویسی
موضوع	:	وب-- سایت‌ها-- طراحی-- دستنامه‌ها
رده بندی کنگره	:	QA۷۶/۷۳ ۱۳۹۱ پ۸۷م۴/
رده بندی دیویی	:	۲۷۶۲/۰۰۵
شماره کتابشناسی ملی	:	۲۹۶۳۱۰۴

انتشارات پندارپارس

دفتر فروش: انقلاب، ابتدای کارگرجنوبی، کوی رشتچی، شماره ۱٤، واحد ۱٦ **www.pendarepars.com**

تلفن: ٦٦٥٧٢٣٣٥ – تلفکس: ٦٦٩٢٦٥٧٨ همراه: ۰۹۱۲۲٤٥٢٣٤٨ **info@pendarepars.com**

نام کتاب	:	**آموزش کاربردی CakePHP**
ناشر	:	**انتشارات پندار پارس**
ترجمه و تالیف	:	**محمد مرادی**
چاپ نخست	:	**پاییز ۹۱**
شمارگان	:	**۱۰۰۰ نسخه**
طرح جلد	:	**فرزانه آقابیگی روزبهانی**
لیتوگرافی، چاپ، صحافی	:	**ترام‌سنج، صالحان، خیام**
قیمت	:	**۱۲۰۰۰ تومان به همراه CD**

شابک: ۷-۲۵-۶۵۲۹-۶۰۰-۹۷۸

سخن مؤلف

سخن به گزافه نرانده‌ایم اگر بگوییم که در هزاره‌ی جدید، تکنولوژی لحظه به لحظه در حال ارتقاء و توسعه است و این فرایند، در همه‌ی عرصه‌های زندگی، جاری و ساری می‌باشد. اگر بخواهیم این تحولات را در دنیای برنامه‌نویسی و تولید برنامه‌های کاربردی بررسی کنیم، به روشنی، شاهد به‌وجود آمدن تغییرهایی عمده در روش‌های مرسوم خواهیم بود.

در زمانی نه چندان دور، و حتی هم‌اکنون، بسیاری از برنامه‌های کاربردی به صورتی غیرساخت‌مند و به بیانی دیگر، به صورت جزیره‌وار ایجاد می‌شدند که رنج و مشقت زیادی برای برنامه‌نویسان در پی داشت و خروجی آن نیز برنامه‌ای ناکارآمد با نتیجه‌بخشی ضعیف بود.

چنین برنامه‌هایی افزون بر اینکه در زمان ترمیم، اصلاح و ارتقاء، از خود مقاومت نشان می‌دهند، امکان افزوده شدن اعضای جدید به تیم توسعه‌دهندگان را نیز به پایین‌ترین حد ممکن می‌رساند. علت این امر نیز چیزی نیست جز عادت‌های منحصربه‌فرد برنامه‌نویسان مختلف که برنامه‌های خود را بر پایه‌ی آن ایجاد می‌کنند. به بیانی ساده‌تر، هر برنامه‌نویس تنها خود می‌داند که برنامه را چگونه ایجاد کرده است. اگرچه برای حل چنین مشکلاتی راه حل‌هایی ارائه و به‌کار گرفته شد، ولی توفیق چندانی به‌دست نیامد.

موج توسعه‌ی تکنولوژی در صنعت برنامه‌نویسی و نرم‌افزار، رویکردی جدید و بسیار سودمند را ارائه نمود و آن چیزی جز برنامه‌نویسی بر مبنای فریم‌ورک‌ها در مقابل برنامه‌نویسی سنتی نبود. در این شیوه، برنامه‌نویسان باید خود را با شرایط و ضوابط فریم‌ورک مورد استفاده سازگار نمایند و به بیانی، همسو با دستورکارهای آن عمل نمایند و در ازای آن، از امکانات و کتابخانه‌های کد فراهم شده توسط آن بهره ببرند.

از عواملی که سبب محبوبیت روزافزون فریم‌ورک‌ها شده است می‌توان به ساخت‌مند کردن فرایند تولید برنامه‌های کاربردی در کنار سرعت‌بخشیدن به مراحل توسعه و ترمیم اشاره کرد.

اکنون فریم‌ورک‌ها برای زبان‌های برنامه‌نویسی مختلف ارائه شده‌اند؛ به گونه‌ای که شرکت مایکروسافت که معمولا در مقابل چنین پدیده‌هایی از خود مقاومت نشان می‌دهد نیز دست به‌کار شده و با پذیرش این رویکرد، فریم‌ورک ASP.NET MVC را به جامعه‌ی برنامه‌نویسی عرضه نموده است.

اگرچه برای زبان محبوب PHP، ده‌ها فریم‌ورک موجود است، اما در این کتاب به معرفی و شرح چگونگی استفاده از CakePHP به‌عنوان یکی از چهار فریم‌ورک برتر جهان پرداختیم.

قطعا هیچگاه این امکان وجود ندارد که بتوان همه‌ی جنبه‌های یک تکنولوژی را بررسی نمود، با این حال در این کتاب تلاش کردیم که با استفاده از منابع مختلف، مهم‌ترین مسائل مطرح درباره‌ی این فریم‌ورک را شرح دهیم و شرایطی را فراهم کنیم تا خواننده پس از خواندن کتاب، به توانایی با سطح بالاتر از متوسط دست یابد. امیدواریم که به یاری خداوند منان در این امر موفق شده باشیم. همچون همیشه،آماده‌ی دریافت نظرات، پیشنهادات و انتقادات شما خوانندگان گرامی هستیم.

moradi.c85@gmail.com

محمد مرادی   تابستان 91

فهرست

فصل 1

معرفی CakePHP

بیشتر کتاب‌هایی که درباره‌ی تکنولوژی‌های مختلف نگارش می‌شوند، معمولا با شرح ویژگی‌های تکنولوژی مورد بحث آغاز می‌شوند و پس از آن، کاربران و خوانندگان را قانع می‌کنند که از این تکنولوژی و یا نرم‌افزار جدید استفاده نمایند. ولی برخلاف این شیوه‌ی رایج، می‌خواهیم کار را با یک هشدار آغاز کنیم.

"وقتی شروع به خواندن این کتاب نمایید، دیگر هیچ راه برگشتی ندارید! دیگر توسعه‌ی نرم‌افزارهای تحت وب به‌صورتی‌که پیش‌ترآن را می‌شناختید، نخواهد بود و ایجاد چنین برنامه‌هایی آن‌قدر ساده خواهد بود که ممکن است خوانندگان را تنبل کند! به‌راستی باید نحوه‌ی نگارش و قواعد دستورات PHP را فراموش کنید، در پایان نیز واژه‌ی Cake معنی دیگری برای شما پیدا خواهد کرد!"

اگر همچنان مصمم به خواندن این کتاب هستید، پس خوش آمدید، ولی بعدا ما را سرزنش نکنید! هشدارهای لازم داده شد.

نخستین گام در سفر ما به دنیای CakePHP، در این فصل از کتاب برداشته می‌شود و در آن به این خواهیم پرداخت که CakePHP چیست؟ چه مزایایی برای ما دارد؟ و چرا باید از آن استفاده کنیم؟

CakePHP چیست؟

طبق اعلان وب‌سایت رسمی CakePHP (در آدرس cakephp.org)، Cake یک فریم‌وُرک برای توسعه‌ی سریع PHP است که از الگوهای طراحی رایج مانند MVC، Front Controller، Association mapping و Active Record استفاده می‌کند. "هدف اصلی ما (تیم ایجادکنندگان) فراهم‌کردن یک فریم‌وُرک ساخت‌یافته است که بتواند کاربران PHP با هر سطح توانایی و دانش را قادر به توسعه‌ی برنامه‌های کاربردی قدرتمند تحت وب نماید. البته بدون از دست رفتن هرگونه انعطاف‌پذیری."

ممکن است افرادی که در رابطه با Frame workها آشنایی زیادی ندارند، بخش‌هایی از این تعریف را به خوبی متوجه نشوند، به همین دلیل در ادامه به بیان توضیحاتی درباره‌ی آنها می‌پردازیم.

از آنجا که برخی از اصطلاحات، در واقع به طور کامل بیان کننده‌ی ویژگی خاصی می‌باشند و در میان برنامه‌نویسان در سراسر جهان شناخته شده‌اند، بهتر است که با همان نام ذکر شوند، چرا که ترجمه‌ی آنها افزون بر آنکه به‌طور کامل انعکاس دهنده‌ی مطلب نخواهد بود، ممکن است باعث ایجاد مشکل در فرایند آموزش نیز بشود. از این رو ما هم در ادامه‌ی کتاب، این چنین کلمات و اصطلاحات را به صورت زبان اصلی مورد استفاده قرار می‌دهیم.

PHP Framework

یک فریم‌وُرک PHP مجموعه‌ای از کدها، کتابخانه (library)، کلاس‌ها و محیط‌های زمان اجرا (Run-time Environment) می‌باشد که به برنامه‌نویسان اجازه می‌دهد برنامه‌های خود را با سرعت بیشتری تولید نمایند. ایده‌ی اصلی استفاده از فریم‌وُرک‌ها این است که کارایی و قابلیت‌هایی که به صورت رایج مورد استفاده قرار می‌گیرند و نیز ساختار اصلی مورد استفاده در بیشتر برنامه‌ها، برای برنامه‌نویسان، از پیش فراهم شود.

همان‌گونه که ممکن است برای خود شما هم رخ داده باشد، بسیاری از برنامه‌نویسان PHP، توابع و کلاس‌های مربوط به خود را دارند که در بیشتر پروژه‌ها از آنها استفاده می‌کنند. حال ایده‌ی فریم‌وُرک‌ها و به ویژه CakePHP نیز توسعه یافته‌ی همین نظریه است، یعنی برنامه‌نویسان با استفاده از یک ساختار مشخص و البته قدرتمند و با مجموعه‌ای از توابع و کتابخانه‌های کد که توسط تعداد زیادی از برنامه‌نویسان حرفه‌ای و مجرب تولید شده و نیز به مراتب توسط تعداد بیشتری از برنامه‌نویسان مورد استفاده قرار گرفته است، قادر خواهند بود در ایجاد برنامه‌های خود، از تجارب افراد دیگر استفاده کنند. همچنین انجام این کار، قابلیت اعتماد برنامه‌ها را نیز افزایش می‌دهد.

به بیان دیگر، برای استفاده از یک PHP Framework، باید قواعد استفاده از آن فریم‌وُرک را بیاموزید (که البته کار بسیار ساده‌ای است)، سپس با استفاده از ویژگی‌ها و قدرت آن فریم‌وُرک، به سرعت خواهید توانست برنامه‌های خود را ایجاد کنید.

نکته‌ی جالب دیگر، استفاده از قواعد مشخص برای کار با یک فریم‌وُرک می‌باشد که به شما اجازه می‌دهد زمانی که حجم کدها بسیار زیاد شد و یک برنامه‌ی پیچیده به‌دست آمد، به راحتی بتوانید به مدیریت و اصلاح آن بپردازید. چراکه عکس حالت عادی که ممکن است در هر زمان به‌گونه‌ای به کدنویسی بپردازیم، در طول یک پروژه‌ی بر پایه CakePHP مجبور هستیم از قواعد آن برای کدنویسی استفاده کنیم که این ویژگی بی‌نهایت مفید خواهد بود. همچنین این امر سبب می‌شود که در هر

مرحله‌ای از پروژه، هر برنامه‌نویس دیگری که با این قواعد آشنایی دارد به‌راحتی بتواند به پروژه ملحق شود و کار را پی بگیرد.

Design Patternهای رایج

یک Design Pattern (یا الگوی طراحی) یک راه‌حل عمومی برای یک مسئله‌ی رایج در توسعه‌ی نرم‌افزارها (تحت وب) می‌باشد. هر Design Pattern مجموعه کاملی از کد نیست، بلکه شرح چگونگی حل یک مسئله در موقعیت‌های گوناگون است.

> اگر با قواعد مربوط به شیءگرایی و طراحی شیءگرا آشنایی داشته باشید، مطمئنا Design Patternها را می‌شناسید، ولی اگر چنین نیست نگران نباشید، در طول این کتاب می‌آموزید که چگونه از Design Pattern استفاده کنید و به صورت شیءگرا (Object Oriented) کدنویسی کنید، البته بدون اینکه خودتان متوجه شوید که چگونه این کار را آموخته‌اید. این است قدرت و سادگی CakePHP!

در دنیای توسعه‌ی نرم‌افزارهای تحت وب، Design Patternهای مختلفی وجود دارند که همان‌گونه که پیش‌تر بیان شد، Cake از چند نمونه‌ی معروف آنها استفاده می‌کند. در میان آنها MVC[1] در واقع هسته‌ی اصلی CakePHP می‌باشد. در ادامه، در رابطه با آن بیشتر صحبت خواهد شد.

✓ توسعه‌ی سریع

ادغام Design Patternها در CakePHP به این معنی است که توسعه‌دهندگان برای انجام پروژه‌های مختلف نیازی به شروع از نقطه‌ی صفر ندارند. به عبارت دیگر مسائلی که در دنیای توسعه‌ی وب مطرح است و راه‌حل‌های مناسب و عملی برای آنها وجود دارد، دیگر نیاز به حل شدن دوباره ندارند و راه‌حل‌های آنها در CakePHP گنجانده شده است. اگر پیش از این درگیر پروژه‌های تا حدی بزرگ بوده‌اید و با مشکلات آنها آشنا هستید، قطعا پس از فراگرفتن کار با CakePHP لذت بیشتری از انجام چنین پروژه‌هایی خواهید برد، چراکه خواهید توانست بسیار سریع‌تر و البته دقیق‌تر از گذشته کارهای مربوط را انجام دهید.

---

[1] Model View Control

✓ سازگاری با PHP4 , PHP5

یکی دیگر از ویژگی‌های برتر این فریم‌وُرک، سازگاری آن با نسخه‌های 4 و 5 از زبان برنامه‌نویسی PHP می‌باشد. اگرچه در صورت نبود محدودیت، پیشنهاد می‌شود از نسخه 5 به بالا استفاده کنید ولی این سازگاری اجازه می‌دهد، در مواقعی که به دلایل مختلف امکان استفاده از نسخه‌های بالاتر وجود ندارد، بتوانید برنامه‌ی خود را بر روی سروری که از نسخه PHP4 استفاده می‌کند، اجرا نمایید.

نتیجه این می‌شود که برای استفاده از CakePHP هیچ مانعی وجود ندارد؛ زیرا به نظر نمی‌رسد بتوان وب‌سروری یافت که از نسخه‌یPHP پایین‌تر از 4 استفاده کند.

افزون بر این، Cake یک پروژه‌ی کد منبع باز (open source) و رایگان می‌باشد؛ به این معنی که هر شخص، افزون بر اینکه به صورت رایگان می‌تواند از آن به هر منظوری استفاده کند، اجازه‌ی مشاهده‌ی کد منبع آن را نیز دارد.

✓ کدهای ساخت‌یافته

PHP یک زبان برنامه‌نویسی فوق‌العاده است. افزون بر توانمندی‌های زیاد و نرمش‌پذیری بالا برای ایجاد برنامه‌های مختلف و کاربردی، سادگی استفاده از آن، به یکی از مهم‌ترین مزایای آن تبدیل شده است. هر کاربر، حتی بدون داشتن دانش پیشین برنامه‌نویسی می‌تواند با صرف اندک زمانی، صفحات وب داینامیک ایجاد کند. در واقع هیچ محیط برنامه‌نویسی ویژه‌ای برای نوشتن کدهای PHP توسط توسعه‌دهندگان آن ارائه نشده است. این مساله از یک دیدگاه به‌عنوان نقطه قوت آن در نظر گرافته می‌شود که کاربران را محدود به استفاده از یک محیط خاص نمی‌کند و برنامه‌نویسان می توانند از هر ویرایشگر ساده‌ای برای برنامه‌نویسی استفاده کنند. در سوی دیگر، این موضوع سبب می‌شود که هر یک از برنامه‌نویسان PHP به شیوه‌ی خود به انجام این کار مبادرت ورزند و در این راه، مشکلاتی از جمله نبود استانداردهای کدنویسی رخ می‌دهد. شاید یکی از دلایل اقبال تعداد زیادی از برنامه‌نویسان به محیط ویژوال استودیو و تکنولوژی ASP.NET نیز وجود IDE منحصربه‌فرد آن باشد.. البته مسئله‌ی اصلی این نیست، بلکه موضوع مهم در رابطه با اعمال محدودیت این است که سبب می‌شود هر برنامه‌نویس در هر پروژه، از ساختار و قواعد مختص به خود پیروی کند و این موضوع برای برنامه‌نویسان کم تجربه، زمانی که برنامه بزرگ‌تر شود، بسیار مشکل‌ساز خواهد بود. رفع اشکال و ترمیم چنین برنامه‌ای حتی می‌تواند از برنامه‌نویسان حرفه‌ای نیز مدت زمان زیادی را تلف نماید. همچنین، وجود نداشتن قواعد و عادات کدنویسی استاندارد، امکان افزوده شدن افراد دیگر به پروژه و همکاری آنها با تیم توسعه‌دهنده را به نوعی ناممکن می‌سازد. هرچند چنین مشکلاتی در محیط‌هایی مانند ویژوال استودیو نیز به‌وفور رخ می‌دهد.

خوشبختانه CakePHP با محدودکردن برنامه‌نویسان به استفاده از ساختار سفت و سخت خود، این مشکل و نظایر آن را برطرف کرده و سبب می‌شود که کدهای هر پروژه به صورت کلی دارای ساختاری شوند که به راحتی قابل مدیریت و نگهداری باشد.

الگوی MVC

مهم‌ترین Design Pattern مورد استفاده توسط CakePHP، الگوی MVC می‌باشد. این الگوی رایج در توسعه‌ی نرم‌افزارها، کدهای مربوط به پروژه را به سه قسمت اصلی تقسیم می‌کند: controllers، views و models.

البته هدف و نحوه‌ی کار هر یک از این سه بخش، بستگی مستقیم به نوع پیاده‌سازی دارد و از این رو ممکن است این تقسیم‌بندی در فریم‌وُرک‌های مختلف به صورت‌های گوناگونی انجام شود. به هر حال در ادامه، با چگونگی پیاده‌سازی این الگو توسط CakePHP آشنا خواهیم شد.

✓ Models

هر Model در CakePHP، یک جدول مشخص از بانک اطلاعاتی را ارائه می‌دهد. هر جدول از بانک اطلاعاتی باید دارای چنین Modelی باشد. بنابراین در CakePHP هر جدول دارای مدل منحصر به فرد خویش است که داده‌های مربوط در آن ذخیره و یا از آن بازیابی می‌شود. همه‌ی کدهای PHP مربوط به دستیابی، افزودن، اصلاح و یا حذف رکوردها از جدول‌ها، توسط مدل‌های آنها فراهم می‌شود. همچنین، هر Model دارای کدهایی است که ارتباط آن را با دیگر Modelها تعریف می‌کند. مسئولیت تعریف قواعد اعتبارسنجی (Validation Rules) برای اضافه‌کردن و به‌هنگام‌سازی داده‌ها نیز بر عهده‌ی Model می‌باشد. در معماری 3 لایه‌ای می‌توان Model را به عنوان لایه‌ی داده‌ای درنظر گرفت.

و اینکه Model مکانی است که منطق اصلی پروژه در آنجا قرار می‌گیرد؛ برای نمونه، اگر یک Model داشته باشیم که موجودیت ماشین‌ها را ارائه می‌کند، همه‌ی عملیات مربوط به خرید و فروش ماشین و مانند آن باید در این Model تعریف گردد.

✓ Controllers

کنترلرها (Controllers) در CakePHP، جریان یا منطق برنامه را کنترل می‌کنند. هر درخواست وب به سمت یک کنترلر مشخص هدایت می‌شود و این عمل در مکانی انجام می‌شود که ورودی کاربر پذیرفته شده است (درخواست‌ها در وب به دو صورت POST و یا GET می‌باشند). پس از آن منطق

موجود (تعریف شده) در کنترلر تصمیم می‌گیرد که پاسخ به چه صورت تولید شود. منطق کنترلر معمولا شامل فراخوانی Modelها برای دسترسی به داده‌ها و نیز سایر وظایف مانند بررسی وضعیت دسترسی‌ها می‌باشد. کنترلر در پایان، پاسخ تولید شده را به View ارسال می‌کند تا برای کاربران به نمایش درآید.

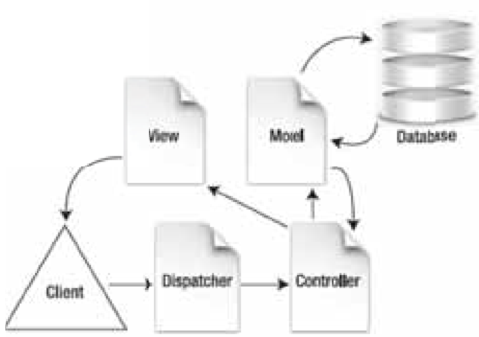
هرچه‌قدر که Model باید قدرتمند بوده و منطق برنامه را به‌صورت کامل فراهم کند، کنترلر باید عملیات مختلف را به‌سمت آن هدایت کند و پاسخ‌های مناسب را تولید نماید؛ از این رو (به نسبت model) باید هر چه بیشتر سبک‌تر (light) باشد.

✓ Views

Viewها خروجی‌ها و یا پاسخ‌هایی هستند که پس از پردازش یک درخواست، به کاربران برگشت داده خواهند شد (برای آنها نمایش داده می‌شوند). به‌صورت استاندارد آنها شامل کدهای HTML به‌همراه کدهای جاسازی شده‌ی[1] PHP می‌باشند، ولی با این حال می‌توانند به صورت‌های دیگری مانند PDF، XML و مانند آن نیز نمایش داده شوند.

✓ چگونگی کارکرد

برای درک بهتر چگونگی کارکرد این سه جزء و ارتباط آنها با یکدیگر، شکل زیر را مشاهده نمایید.



بر طبق این شکل:

1. درخواست کاربر به کنترلر ارسال می‌شود (به همراه داده‌های کاربر و با متدهای POST یا GET)

[1]embedded

2. کنترلر، درخواست را پردازش می‌کند و برای دسترسی به داده‌ها، Model را فراخوانی می‌کند.
3. Model با فرستادن و یا ذخیره‌کردن داده‌ها، پاسخ فراخوانی کنترلر را می‌دهد.
4. کنترلر، داده‌های خروجی را به View ارسال می‌کند.
5. View داده‌های خروجی را در فرمت و ساختار مناسبی برای کاربران به نمایش می‌گذارد.

همان‌گونه که پیش‌تر گفته شد، با استفاده از چنین معماری و الگوی طراحی، افزون بر ساخت‌مند کردن کدها و چارچوب برنامه، مدیریت و نگه‌داری آن نیز ساده‌تر می‌شود.

به‌عنوان نمونه، اگر نیاز به تغییری در بانک اطلاعاتی باشد و یا بانک اطلاعاتی دچار مشکل شود، می‌دانید که باید چه قسمتی از کد را بررسی نمایید و در صورت لزوم تغییرات را بر روی آن اعمال کنید، هرچند این تغییرات هیچ تأثیری بر روی دیگر بخش‌های برنامه نخواهند گذاشت.

دیگر ویژگی‌ها

افزون بر موارد ذکر شده، CakePHP بسیاری از ویژگی‌ها و امکانات دیگر را برای برنامه‌نویسان فراهم کرده است که با آنها در طول کتاب آشنا خواهیم شد؛ ولی برای نمونه، چند ویژگی مهم را معرفی می‌کنیم.

✓ پیکربندی کمتر، قراردادهای بیشتر

منطق CakePHP بر این اصل استوار است که تا حد امکان، کمترین مقدار تنظیمات، پیش از شروع کار انجام شود و از این رو از دردسرهای مربوط به آن کاسته و البته سرعت کار بیشتر شود. در واقع تنها کاری که به‌عنوان تنظیمات باید انجام شود، مشخص کردن نام بانک اطلاعاتی مورد استفاده است و هیچ چیز دیگری مورد نیاز نیست، به همین سادگی!

از سویی دیگر،CakePHP به گونه‌ای طراحی شده است که بیشتر تکیه بر قواعد و قراردادهای تعیین شده، به‌ویژه قواعد نام‌گذاری دارد.

در واقع هر یک از بخش‌های اصلی (مانند نام جدول‌ها، نام فیلدها، مکان قرارگیری کنترلرها و Viewها و ...) در CakePHP از قاعده‌ی خاصی پیروی می‌کنند و با پیروی از این قاعده، فریم‌وُرک به صورت خودکار، کارهای دیگر را انجام می‌دهد. هنگامی که این قواعد را فرا گرفتید دیگر نیاز نیست نگران پیکربندی آن باشید.

✓ تولید خودکار کد

یکی از ویژگی‌های فوق‌العاده جالب CakePHP، ویژگی تولید کد (code generation) به صورت خودکار می‌باشد. این ویژگی به صورت درون ساخت در فریم‌وُرک وجود دارد و به baking script معروف است. گرچه در ادامه با این ویژگی بی‌نظیر آشنا خواهید شد، ولی در این لحظه فقط در این حد بدانید که از طریق این ویژگی تنها لازم است که جدول‌های بانک اطلاعاتی مورد استفاده در پروژه را تعیین کنید و پس از آن، بخش اصلی و چارچوب کدهای موردنیازتان به صورت خودکار ایجاد می‌شود و می‌توانید دیگر مراحل را پی بگیرید.

✓ دسترسی سریع به داده‌ها

با استفاده از الگوی طراحی مورد استفاده در CakePHP، دسترسی به داده‌های بانک اطلاعاتی (چه به صورت ترتیبی و چه به صورت منفرد از یک جدول) بسیار ساده شده است. افزون بر این، با توجه به *لایه‌ی انتزاع داده‌ای*[1] فراهم شده در Cake، برای دسترسی به داده‌ها نیازی به نوشتن چندباره‌ی پرس‌وجوهای SQL نیست و می‌توان این کار را از طریق توابع مناسب ایجاد شده برای مدل‌ها انجام داد.

✓ نسخه‌ی مورد استفاده

هم‌اکنون، نسخه‌ی CakePHP 2.2 موجود است ولی در این کتاب و برای انجام تمرین‌ها و پروژه‌ها، از نسخه‌ی 1.2 استفاده می‌کنیم. البته این به این معنا نیست که استفاده از نسخه‌ی جدیدتر، در انجام تمرین‌های این کتاب ایجاد مشکل می‌کند، بلکه نسخه‌ی جدید تنها دارای ویژگی‌های افزوده شده‌ی دیگری می‌باشد.

✓ منابع آموزشی

همواره یکی از دغدغه‌های کاربران فارسی زبان برای استفاده از تکنولوژی‌های جدید، نبود پشتیبانی مناسب و نیز مطالب آموزشی و انجمن‌های پرسش و پاسخ می‌باشد. اگرچه این کتاب می‌تواند به عنوان یک نقطه‌ی شروع خوب تلقی شود (و مطالب آن به گونه‌ای تنظیم شده‌اند که پس از پایان، به

---

[1] data abstraction layer

یک کاربر Cake با توانایی بالاتر از متوسط تبدیل شوید)، ولی همواره ممکن است پرسش‌ها و مشکلاتی پیش آید که نیاز به منابع تعاملی برای حل آنها وجود داشته باشد. خوشبختانه در این رابطه جای هیچ گونه نگرانی نیست و افزون بر این کتاب، دو منبع بسیار مناسب دیگر در اختیار دارید. نخستین بخش، مربوط به مستندات CakePHP در وب سایت رسمی آن می‌باشد که شامل نسخه‌ی ترجمه شده به فارسی نیز می‌باشد (راهنمای-فارسی/http://book.cakephp.org/fa/view/3)

و مکان دیگر که قطعا بسیار مفید خواهد بود و در آنجا خواهید توانست تجربیات خود را با افرادی که در این زمینه فعالیت می‌کنند در میان گذارید، وب سایت پشتیبانی فارسی از Cake در آدرس cakephp.ir است.

در این فصل، تنها مروری کوتاه به منظور آشنایی آغازین با Cake انجام دادیم و مفاهیم اصلی آن را مورد بررسی قرار دادیم. ولی این تنها نخستین گام در دنیای CakePHP است و در ادامه، با قدرت و البته طعم این توع Cake بیشتر آشنا خواهید شد! با ما همراه باشید.

فصل دوم

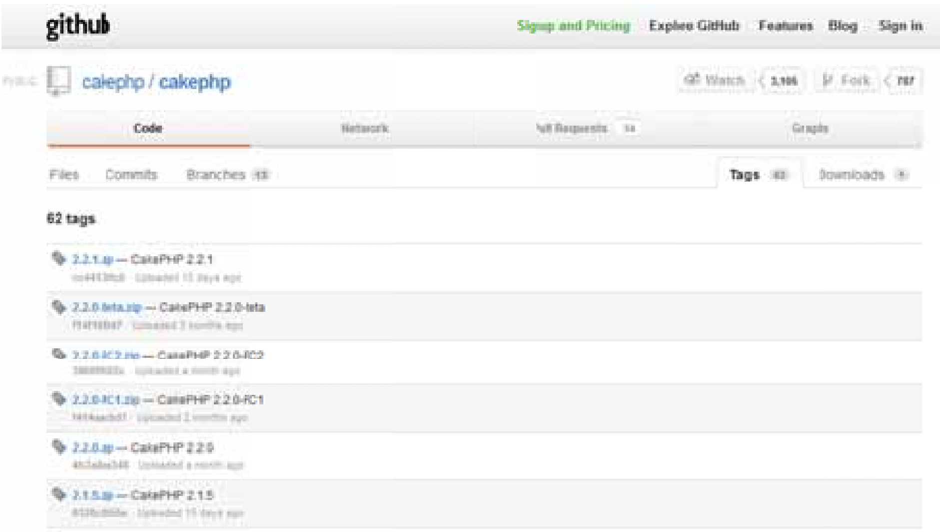
نصب سریع

در فصل نخست آموختیم که چگونه CakePHP می‌تواند به ما کمک کند تا برنامه‌های کاربردی تحت وب را به صورت هر چه بیشتر ساخت‌یافته و البته سریع ایجاد کنیم. گام منطقی بعدی برای ما این است که Cake را نصب کرده و پس از آن شروع به ایجاد برنامه‌های موردنظر خود نماییم.

در این فصل نگاهی گذرا به مراحل نصب و آماده‌سازی CakePHP می‌اندازیم. همان‌گونه که خواهید دید، اگر با کار در محیط‌های توسعه‌ی وب با استفاده از وب سرور Apache، بانک اطلاعاتی MySQL و زبان برنامه‌نویسی PHP آشنا باشید، نصب Cake کار سختی نخواهد بود.

دریافت CakePHP

نخستین کاری که باید انجام شود دریافت نسخه‌ی 1.2CakePHP از وب سایت رسمی آن می‌باشد. نسخه‌ی جاری Cake را در صفحه‌ی نخست سایت می‌بینید، ولی از آنجا که ما در این کتاب از نسخه‌ی 1.2 استفاده می‌کنیم می‌توانید آن را از مسیر زیر دریافت کنید. البته برای راحتی کار خوانندگان گرامی، نسخه‌های مختلف CakePHP در لوح فشرده همراه کتاب قرار گرفته است.



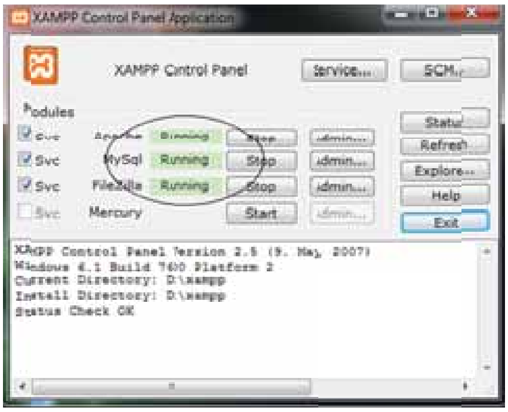
امکانات مورد نیاز سیستم

CakePHP برای اجرا شدن، نیاز به نصب بر روی یک وب سرور دارد. اما از آنجا که ممکن است دسترسی به اینترنت همیشه ممکن نباشد و اصلا شیوه‌ی منطقی همه‌ی برنامه‌نویسان برای آزمایش چنین پروژه‌هایی، نصب و اجرای آن بر روی سیستم محلی (local) است ساده‌ترین راه، استفاده از محیط‌های مجتمع مانند XAMPP، LAMP و مانند آن می‌باشد.

اگر با چنین بسته‌های نرم‌افزاری آشنایی ندارید، جالب است که بدانید پس از نصب، آنها برای شما افزون بر نصب PHP و وب سرور Apache، امکان دسترسی و مدیریت بانک اطلاعاتی را نیز فراهم می‌کنند.

به همین منظور بسته‌ی نرم‌افزاری XAMPP در لوح فشرده‌ی همراه کتاب قرار داده شده است. نصب این نرم‌افزار به سادگی انجام می‌پذیرد و پس از آن می‌توانید از طریق تایپ آدرس Localhost یا 127.0.0.1 در نوار آدرس مرورگر، وارد صفحه مدیریت آن شوید. البته دقت کنید که نخست باید نرم‌افزار را فعال کرده باشید.

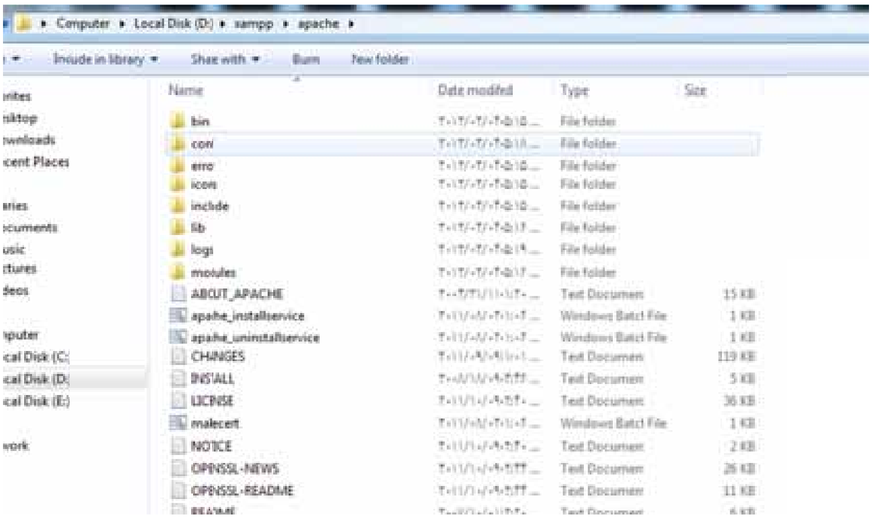

XAMPP 1.6.3a win32 (Basic Package)
XAMPP Options
Install options on NT/2000/XP Professional systems.
XAMPP DESKTOP
XAMPP START MENU
Create an Apache Friends XAMPP folder in the start menu
SERVICE SECTION
Install Apache as service
Install MySQL as service
Install Filezilla as service
See also the XAMPP for Windows FAQ Page
< Back
Install
Cancel


XAMPP Control Panel Application
XAMPP Control Panel
Service...
SCM...
Modules
Svc
MySql
Running
Stop
Admin...
FileZilla
Mercury
Start
Status
Refresh
Explore...
Help
Exit
XAMPP Control Panel Version 2.5 (9. May, 2007)
Windows 6.1 Build 7600 Platform 2
Current Directory: D:\xampp
Install Directory: D:\xampp
Status Check OK


XAMPP for Windows
Welcome to XAMPP for Windows Version 1.6.3a !

پیکربندی Apache

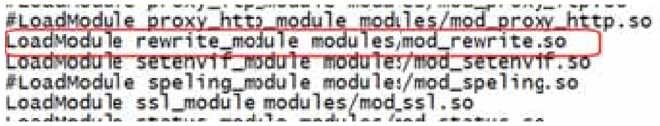
برای اینکه Cake بتواند به صورت صحیح بر روی وب سرور اجرا شود، نیاز به یک دست‌کاری کوچک در فایل httpd.conf داریم. این فایل در دایرکتوری Conf در محل نصب وب سرور Apache قرار دارد.



در این فایل باید بخشی به نام <Directory> وجود داشته باشد. مقدار تعیین شده برای این ویژگی، تعیین کننده‌ی محل قرارگیری پوشه‌ی اصلی (محل قراردادن فایل CakePHP) می‌باشد.تنها کاری که در این بخش باید انجام دهیم تعیین مقدار all برای ویژگی AllowOverride است:

```
<Directory "D:/xampp/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride all
Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1
</Directory>
```

البته باید بررسی کنیم که ماژول mod-rewrite به طور صحیح بارگذاری شده باشد. به همین منظور در فایل httpd.conf در بخش SupportDynamic Shared Object (DSO) که محل قرارگیری ماژول‌های مختلفی است که توسط Apache بارگذاری شده است، باید ماژول mod-rewrite را فعال کنیم. یعنی علامت # که به معنی فعال نبودن آن می‌باشد را از ابتدای خط نمایش داده شده در شکل زیر حذف نماییم.

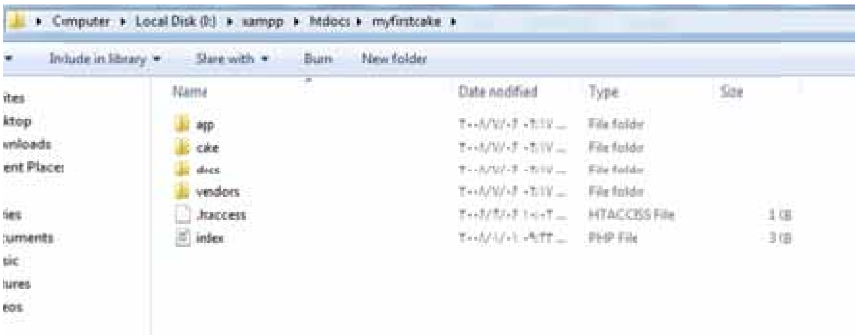


در آخر و پس از ذخیره‌ی فایل، وب سرور را راه‌اندازی دوباره نمایید تا تغییرات اعمال شوند.

اگر این تغییرات به درستی انجام شده باشند، برای راه‌اندازی CakePHP نباید مشکلی داشته باشید.

نصب CakePHP

برای نصب Cake لازم است که فایل فشرده شده‌ی آن را از حالت فشرده خارج نمایید و آن را در داخل یک دایرکتوری جدید در دایرکتوری htdocs قرار دهید. نام این دایرکتوری جدید را می‌توانید به هر نام اختیاری تغییر دهید. در اینجا ما آن را myfirstcake می‌نامیم. اگر به این دایرکتوری وارد شوید، نمایی همانند تصویر زیر را خواهید دید.



نصب Cake به همین سادگی انجام شد. ولی یک گام دیگر باقی مانده است و آن اطمینان یافتن از درستی نصب آن می‌باشد. به این منظور کافی است (پس از فعال کردن Apache) در نوار آدرس مرورگر وب خود، آدرس Local host/myfirstcake را وارد نمایید.

البته ممکن است در برخی موارد، این صفحه به این زیبایی و آراستگی نمایش داده نشود و متون داخل صفحه به صورت ساده به نمایش در آیند ولی این به معنی نصب نشدن Cake نیست و مسئله خاصی ایجاد نمی‌کند، پس نگران نشوید!

اگر پس از انجام این کار، صفحه‌ای همانند تصویر بالا و البته با پیغام‌های فوق نمایش یافت، یعنی CakePHP به صورت درست و کامل بر روی سیستم شما (وب سرور) نصب شده است. پس از انجام همه‌ی این گام‌ها و دیدن نتیجه‌ی مورد نظر، آماده‌ی استفاده از Cake برای ایجاد برنامه‌های کاربردی وب، آن هم با کیفیتی که ذکر شد، می‌باشید. این دقیقا همان کاری است که در فصل بعد به آن خواهیم پرداخت.

فصل سوم

ایجاد یک برنامه‌ی سریع

در فصل پیش، همه‌ی کارهای مربوط به آماده‌سازی CakePHP انجام شد و به بیانی، اجاق پختن Cake روشن و گرم شد، و تنها کاری که باید انجام دهیم، دست به‌کارشدن برای پختن برنامه‌ی کاربردی مورد نظرمان است!

در این فصل برای آشنایی آغازین با موارد اساسی CakePHP و نیز نحوه‌ی کار آن، با ایجاد یک برنامه‌ی کاربردی کوچک و سبک وارد عمل می‌شویم.

برنامه‌ای که قصد ایجاد آن را داریم، یک برنامه‌ی to-do-list می‌باشد. همان‌گونه که شاید بدانید (و در گوشی‌های تلفن همراه خود از این ویژگی استفاده کرده‌اید) این برنامه و موارد مشابه آن به‌عنوان یک یادآور برای انجام کارهای روزمره مورد استفاده قرار می‌گیرد. حتما تاکنون مواردی پیش آمده که از این برنامه با استفاده از قلم و کاغذ استفاده کرده باشید! مانند لیست کالاهایی که باید خریداری شوند، کتاب‌هایی که باید خوانده شوند و مانند آن. در آغاز، نام این برنامه را CakeTooDoo انتخاب می‌کنیم.

CakeTooDoo: یک برنامه‌ی ساده

پیش از آغاز، نخستین کاری که باید انجام شود، تعیین نام دایرکتوری اصلی محل قرارگیری فایل‌های CakePHP به CakeTooDoo است. به بیان دیگر، نام دایرکتوری که در فصل پیش myfirstcake تعیین شد را به نام جدید تغییر دهید. این برنامه به این صورت خواهد بود که هر کاری در لیست ثبت شده است، دارای یک عنوان و یک وضعیت می‌باشد. عنوان (Title)، مشخص کننده‌ی نوع عملی است که باید انجام شود و وضعیت (Status)، نشان دهنده‌ی این است که کار به صورت کامل انجام شده است یا نه. همچنین افزون بر این موارد، هر کار (Task)، زمان ایجادشدن و اعمال آخرین تغییرات را نیز ذخیره می‌کند. با استفاده از این برنامه قادر خواهیم بود کارهای جدید را اضافه کنیم، وضعیت یک کار را تغییر دهیم، یک کار را حذف کنیم و همه کارها را مشاهده کنیم.

شاید در نگاه نخست، انجام این کارها کمی سخت به نظر بیاید ولی در ادامه خواهید دید که انجام این عملیات به سادگی و البته خوشمزگی خوردن یک برش کیک می‌باشد!

پیکربندی Cake برای کار با بانک اطلاعاتی

نخستین کاری که باید انجام دهیم ایجاد یک بانک اطلاعاتی برای برنامه‌ی مورد نظرمان است. ایجاد یک بانک اطلاعاتی برای برنامه‌های ایجاد شده توسط CakePHP تفاوتی با ایجاد بانک برای دیگر برنامه‌ها و کارهایی که پیش‌تر انجام داده‌اید ندارد. اما در اینجا باید برای نام‌گذاری جدول‌های بانک اطلاعاتی، از یک سری قواعد ساده پیروی کنید. نخست، یک بانک اطلاعاتی جدید به نام caketoodoo ایجاد می‌نماییم.

> برای ایجاد بانک اطلاعاتی ممکن است به جای نرم‌افزار phpMyAdmin از محیط‌های دیگر و حتی از Command-line و اجرای دستورات MySQL استفاده کنید. اگرچه هیچ تفاوتی میان روش‌های مختلف انجام این کار وجود ندارد، اما برای رعایت یکنواختی کار در این کتاب، تنها کدهای لازم را بیان می‌کنیم و می‌توانید آنها را به هر روشی که برای ایجاد بانک اطلاعاتی به کار می‌گیرید، مورد استفاده قرار دهید. ضمن اینکه کدهای کامل همه‌ی تمرین‌ها، در لوح فشرده همراه کتاب وجود دارد.

در بانک اطلاعاتی ایجاد شده، جدول tasks را با ساختار زیر ایجاد نمایید:

```
USE caketoodoo;
CREATE TABLE tasks (
id int(10) unsigned NOT NULL auto_increment,
title varchar(255) NOT NULL,
done tinyint(1) default NULL,
created datetime default NULL,
modified datetime default NULL,
PRIMARY KEY (id)
);
```

در درون دایرکتوری CakeTooDoo/app/Config فایلی به نام database.php.default وجود دارد. آن را یافته و نامش را به database.php تغییر دهید. سپس این فایل را با استفاده از ویرایشگر مورد نظرتان باز نمایید و به خط 73، جایی که آرایه‌ای به نام $default قرار دارد، بروید. این آرایه، دربردارنده‌ی اطلاعات مربوط به اتصال بانک اطلاعاتی می‌باشد.

اطلاعات آن (login برای نام کاربری، Password برای رمز عبور و database برای نام بانک اطلاعاتی) را با استفاده از اطلاعات بانک اطلاعاتی مورد استفاده‌ی خود ویرایش نمایید و پس از، آن فایل را ذخیره کنید.

```
var $default = array(
'driver' => 'mysql',
'persistent' => false,
'host' => 'localhost',
```

```
	'port' => '',
	'login' => 'ahsan',
	'password' => 'sims',
	'database' => 'caketoodoo',
	'schema' => '',
	'prefix' => '',
	'encoding' => ''
);
```

پس از انجام این موارد، اگر آدرس Localhost/CakeTooDoo را در مرورگر خود وارد نمایید، در صفحه‌ای پیش رو، شاهد این دو خط خواهید بود: نخست Your database configuration file is present و دیگر Cake is able to connect to the database.

اینها به این معنا هستند که Cake به بانک اطلاعاتی شما متصل شده است.

نکته‌ای که در اینجا لازم به توضیح است، توجه به نوع نام‌گذاری جدول‌های بانک اطلاعاتی است.

همانطور که مشاهده کردید، نام جدول را tasks انتخاب کردیم. این یک قاعده در Cake می‌باشد که جدول‌های بانک اطلاعاتی باید نام‌هایی به صورت اسم جمع، مانند Comments، Users و مانند آن داشته باشند. همچنین کلید اصلی (Primary key) جدول باید فیلدی به نام id باشد.

نوشتن نخستین Model

آن‌گونه که در فصل نخست بیان شد، هر جدول در بانک اطلاعاتی، نیاز به یک Model منطبق دارد. این Model به منظور دستیابی به اطلاعات جدول و نیز ایجاد تغییرات در آن استفاده می‌گردد. برای

ایجاد این Model، در مسیر CakeTooDoo/app/ models فایلی به نام task.php ایجاد و کدهای زیر را در درون آن وارد نمایید:

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

پیش از ذخیره‌ی فایل، مطمئن شوید که هیچ فضای خالی (Space یا Tab) پیش از تگ php?> و پس از تگ <? قرار نگرفته باشد.

همه‌ی Modelها در CakePHP باید در دایرکتوری models (در مسیری که بیان شد) قرار گیرند. در این بخش نیز دو قاعده برای نام‌گذاری باید مورد توجه قرار گیرد. نخست اینکه برای نام فایل‌های مربوط به هر Model باید نامی همانند نام جدول بانک اطلاعاتی ولی به صورت اسم مفرد (در اینجا task.php) تعیین شود.

قاعده‌ی دیگر، نوعی نام‌گذاری کلاس مربوط به model می‌باشد. این نام نیز باید همانند نام جدول مربوط و به صورت اسم مفرد انتخاب شود و البته به صورت Camel Cased یعنی حرف اول هر واژه به صورت بزرگ (uppercase) و دیگر واژگان آن به صورت کوچک (lowercase). دقت نمایید که نام‌های چند بخشی مانند CarName به این صورت نوشته می‌شوند.

اگر با مفاهیم شیءگرایی آشنایی داشته باشید (البته آگاهی از آنها الزامی نیست)، متوجه شده‌اید که کلاس (class) تعریف شده توسط شما از کلاس AppModel ارث‌بری می‌کند. در Cake، همه‌یModelها باید از این کلاس ارث‌بری نمایند و این کلاس نیز به نوبه‌ی خود از کلاس Model ارث می‌برد و از این طریق اجازه‌ی انجام اَعمال اضافه‌کردن، حذف، ویرایش و دسترسی به داده‌های بانک اطلاعاتی را به Modelها می‌دهد.

نکته‌ی آخر اینکه ما متغیری به نام $name داشتیم که نام Model مورد نظر را به‌عنوان مقدار به آن منتسب کردیم. اگرچه این کار ضروری نیست و خود Cake این عمل را به صورت خودکار انجام می‌-دهد، ولی بهتر است که این کار را خودمان به صورت دستی انجام دهیم.

نوشتن نخستین کنترلر

پس از ایجاد Model، اکنون نوبت نوشتن نخستین کنترلر می‌باشد. وقتی که یک درخواست (یک عملیات) به یک برنامه‌ی تحت وب ارسال می‌شود، کنترلرها تصمیم می‌گیرند که چه چیزی باید انجام شود. به عبارت دیگر آنها جریان کار برنامه‌ها را کنترل می‌کنند. به‌عنوان مثال اگر داده‌ای بخواهد

مورد دستیابی قرار بگیرد، کنترلر، Modelی را که وظیفه‌ی واکشی[1] اطلاعات را بر عهده دارد فراخوانی می‌کند.

برای ایجاد یک کنترلر، نخست باید در مسیر CakeTooDoo/app/Controllers فایلی به نام tasks_controller.php را ایجاد و سپس، قطعه کد زیر را در آن وارد نمایید:

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

در توضیح عملیات انجام شده باید گفت که همه‌ی کنترلرها در Cake باید در این دایرکتوری بیان شده، قرار بگیرند. در Cake هر Model ایجاد شده نیاز به یک کنترلر مختص به خود دارد. پس برای Model ما به نام Task، یک کنترلر به نام TasksController وجود دارد. البته در برنامه‌های پیچیده‌تر، این امکان وجود دارد که یک کنترلر، از بیش از یک Model استفاده کند (به آن احتیاج داشته باشد).

سبک نام‌گذاری کنترل کننده‌ها نیز به این صورت است که فایل آنها شامل نام Model به صورت اسم جمع به‌علاوه‌ی یک علامت Under Score (_) به همراه واژه‌یControllerمی‌باشد. همان‌گونه که در قطعه کد بالا می‌بینید، الگوی نام‌گذاری کلاس مربوط به کنترلر نیز به صورت CamelCased و اسم جمع می‌باشد.

همانند Modelها، همه‌ی کلاس‌های کنترلر نیز از کلاس AppController و این کلاس هم از کلاس دیگری به نام Controller ارثبری می‌کند و به این روش، قابلیت‌های لازم برای انجام عملیات مختلف را به‌دست می‌آورد. دوباره همانند Modelها، ما متغیری تعریف کردیم و مقدار آن را برابر Tasks قرار دادیم. اگرچه این کار ضروری نیست ولی بهتر است که همیشه انجام شود.

به یاد داشته باشید که نام‌های Modelها همواره به صورت مفرد می‌باشند، در حالی‌که نام‌های کنترلرها همیشه باید به صورت جمع تعیین شوند.

[1]Fetch

مشاهده‌ی همه‌یTaskها در برنامه

اکنون که Model و کنترلر مربوط به آن ایجاد شده‌اند، نوبت به این رسیده است که قابلیت‌هایی چند را به برنامه‌ی خود بیافزاییم. نخستین کاری که می‌خواهیم انجام دهیم، مشاهده‌ی لیستی از کارها (Taskها) می‌باشد. برای انجام این کار، دوباره فایل tasks_controller.php را گشوده و همانند قطعه کد زیر، متدی به نام index را به کلاس TasksController بیافزایید. توجه داشته باشید که همه‌ی توابع public در داخل کلاس‌های کنترلرها، action نامیده می‌شوند.

```
<?php
class TasksController extends AppController {
var $name = 'Tasks';
function index() {
$this->set('tasks', $this->Task->find('all'));
}
}
?>
```

سپس به دایرکتوری CakeTooDoo/app/views رفته و پوشه‌ای به نام tasks در داخل آن ایجاد نمایید. در داخل این پوشه فایلی به نام index.ctp ایجاد و کدهای زیر را در آن وارد نمایید:

```
<h2>Tasks</h2>
<?php if(empty($tasks)): ?>
There are no tasks in this list
<?php else: ?>
<table>
<tr>
<th>Title</th>
<th>Status</th>
<th>Created</th>
<th>Modified</th>
<th>Actions</th>
</tr>
<?php foreach ($tasks as $task): ?>
<tr>
<td>
<?php echo $task['Task']['title'] ?>
</td>
<td>
<?php
if($task['Task']['done']) echo "Done";
else echo "Pending";
?>
</td>
<td>
<?php echo $task['Task']['created'] ?>
</td>
<td>
<?php echo $task['Task']['modified'] ?>
</td>
<td>
<!-- actions on tasks will be added later -->
</td>
</tr>
<?php endforeach; ?>
```