



Sviluppo web con CakePHP

Andrea Chiarottino e Francesco Ronchi

Linux Day Torino
24-10-2009

Presentazione

- Chi siamo...
- Chi siete?

Perchè un framework?

- Velocità di sviluppo
- Riutilizzo del codice
- Struttura del codice
- Uso di modelli standard (MVC, ORM...)
- Codice di terze parti (community...)
- Documentazione

CakePHP (1)

- Basato su PHP, ereditando idee da altri linguaggi (vedi Ruby on Rails)
- Oggetti: Modello MVC, ORM
- Coding by convention
- Scaffolding & Bake
- Codice semplice e “pulito”

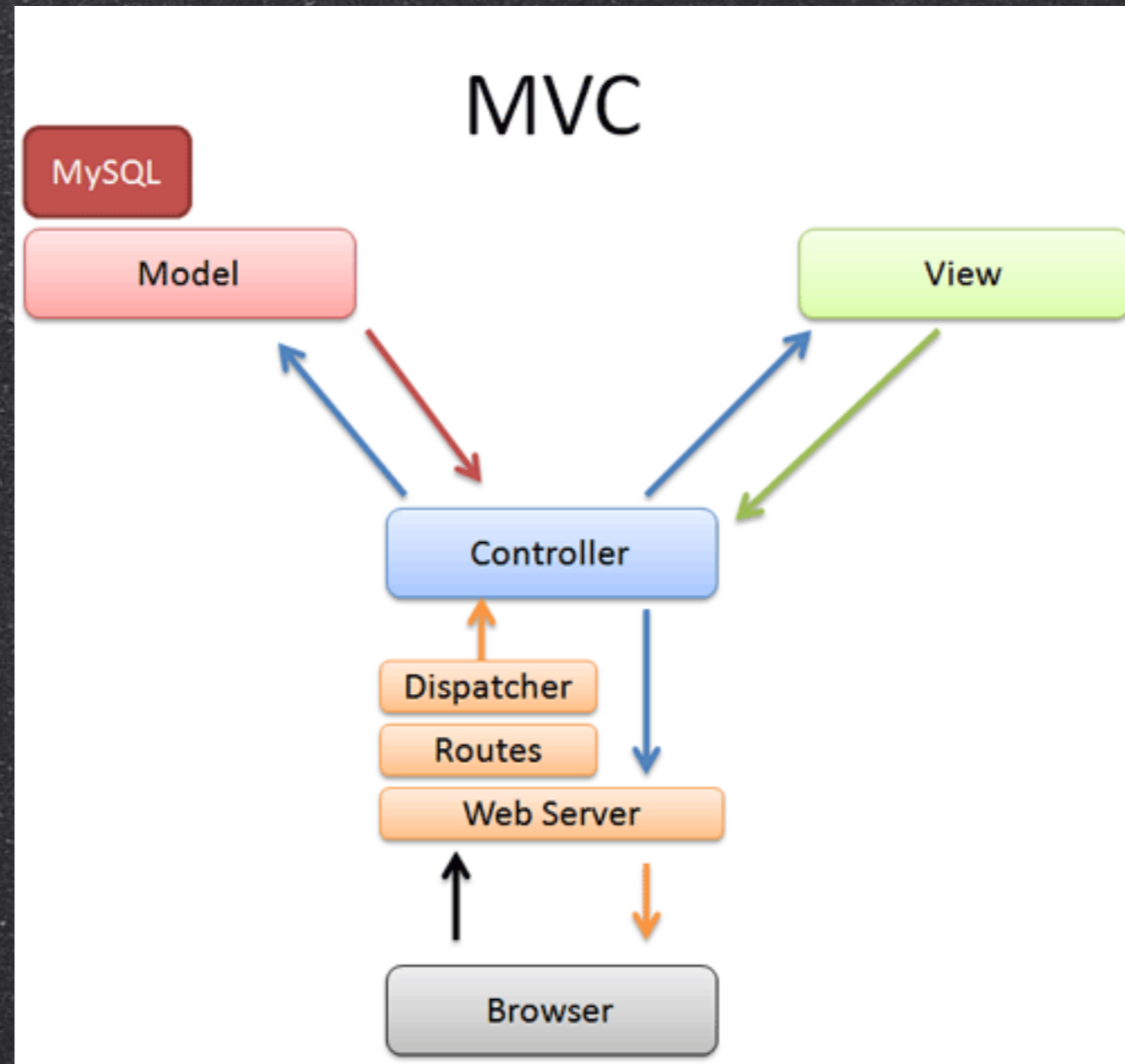
CakePHP (2)

- Facile da estendere con componenti, helper, behaviours e plugin.
- Gestione della validazione
- Supporto AJAX
- Autenticazione e ACL
- Integrazione con SimpleTest

Modello MVC

- MVC = Model, View, Controller
- Model: i dati dell'applicazione, tipicamente il database.
- View: la presentazione delle informazioni, tipicamente pagine web
- Controller: la logica dell'applicazione

MVC visto da CakePHP



Struttura Cartelle

- app/
- cake/
- vendors/
- .htaccess
- index.php

Struttura Cartelle:

app/



app/

- config/
- controllers/
- locale/
- models/
- plugins/
- tmp/
- vendors/
- views/
- webroot/

App Extensions

- `app_model.php`, `app_controller.php`, `app_helpers.php` presenti nella cartella `app`
- Sono le superclassi di `model`, `view` e `controller` dove trovano posto i metodi comuni a tutte le sottoclassi

Components

- Classi integrate nel FW per svolgere compiti comuni che possono essere utilizzate dai controller estendone le funzionalità
- Core Components: ACL, Auth, Cookie, Email, RequestHandler, Security, Session

Behaviors

- Classi che estendono le funzionalità dei model
- Core behaviors: ACL, Containable, Translate, Tree

Helpers

- Classi che includono logica per il livello presentazione che possono essere riutilizzate nelle view
- Core helpers: AJAX, Cache, Form, Javascript, Number, Paginator, RSS, Session, Text, Time, XML

Common Tasks

- Validazione e sanitization dei dati
- Gestione errori e debugging
- Caching e Logging
- Testing
- I18n & L10n
- REST

Utility Libs

- Classi con funzioni di utilizzo generale richiamabili all'interno dell'applicazione
- Core Libs: Inflector, String, Xml, Set, Security, Cache, HttpSocket

Console Apps

- BAKE: script per la generazione di codice: models, views, controllers
- Possibilità di creare template da utilizzare per generare elementi dell'applicazione personalizzati

Scaffolding & Bake

- Creazione dinamica del CRUD delle tabelle del database (Scaffolding)
- Creazione statica delle classi HTML e dei controller (Bake)

Ambiente di sviluppo

- Eclipse
 - PDT per sviluppare in PHP
 - Subclipse per collegamenti a repository SVN
- Firefox
 - Firebug
 - Web developer bar
- Apache (con Mod Rewrite)
- MySQL

Let's go code!

- Pratica con Cake: esempio

Struttura DB

posts	
id	INT(10)
title	VARCHAR(255)
body	TEXT
user_id	INT(11)
created	DATETIME
modified	DATETIME
status	TINYINT(1)
Indexes	

posts_tags	
id	INT(10)
post_id	INT(10)
tag_id	INT(10)
created	DATETIME
modified	DATETIME
Indexes	

tags	
id	INT(10)
tag	VARCHAR(100)
created	DATETIME
modified	DATETIME
Indexes	

users	
id	INT(10)
username	VARCHAR(50)
password	VARCHAR(32)
first_name	VARCHAR(255)
last_name	VARCHAR(255)
created	DATETIME
modified	DATETIME
Indexes	

Code by Convention

- Definire una convenzione per fare le cose, anziché configurarle
- Codice più semplice/leggibile, più leggero e più standard
- Configurazione sempre disponibile per esigenze particolari

Convenzioni per il DB

- Regole principali:
 - Nomi delle tabelle al plurale (in inglese)
 - Chiave primaria campo 'id'
 - Chiavi esterne con nome della tabella al singolare + '_id'

Convenzioni per model

- Regole principali:
 - Nomi delle classi singolari, CamelCased: Person, Apple, BigPerson
 - Nomi dei file separati da _ se composti di più parole: person.php, apple.php, big_person.php

Convenzioni per controller

- Regole principali:
 - Nomi delle classi plurali, CamelCased e terminanti con Controller: PersonController, BigPersonController
 - Nomi dei file separati da _ se composti di più parole: person_controller.php, big_person_controller.php

Convenzioni per view

- Regole principali:
 - Le view sono contenute in una cartella all'interno di una tabella che ha il nome del controller
 - Ciascuna view è un file che ha il nome di un metodo (action) del controller separato da _ se contiene più parole

CakePHP URL

- `...siteurl/controllername/actionname/par1/par2`
- Si possono comunque personalizzare gli URL utilizzando le route

- # Risorse
- Website: <http://www.cakephp.org>
 - Cookbook: <http://book.cakephp.org>
 - Bakery: <http://bakery.cakephp.org>
 - API: <http://api.cakephp.org>
 - CakeForge: <http://www.cakeforge.org>
 - Test Cases:
<http://api.cakephp.org/tests>
 - Google Group:
<http://groups.google.com/group/cakephp>

Libri

Apress – Beginning CakePHP from Novice to Professional (Pag. 341)

Apress – CAKEPHP – Practical CakePHP Projects (Pag. 389)

Packt – Publishing CakePHP Application Development (Pag. 328)

Tutorial redatti da IBM visibili anche su scribd.



Grazie!!!

Andrea Chiarottino e Francesco Ronchi

contacts@crop.it

<http://www.crop.it>

