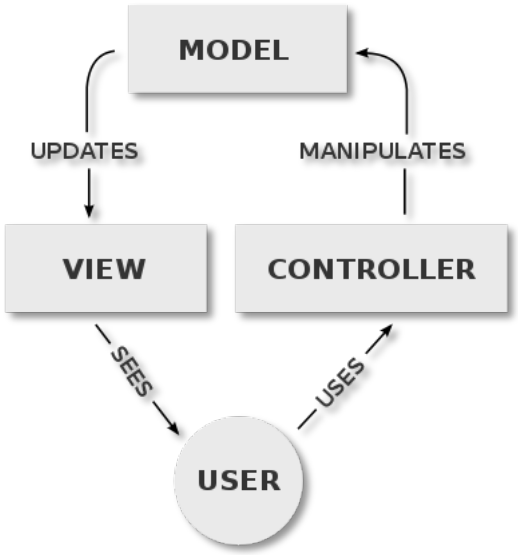
Cakephp 2 Simple Tutorial

Author : Ricky Lam
Last Modified : 2016-12-08



CakePHP

- Open source web framework written in PHP
- Follows Model – View – Controller (**MVC**) approach
- Latest version : For CakePHP 2, v2.9
- Minimum requirements
    - PHP 5.3 or greater
    - HTTP server, e.g. Apache with mod_rewrite (preferred for URL rewriting)

MVC

- Software design pattern
- Divides a given software application into 3 interconnected parts

Model in MVC

- The core component in MVC
- Captures the behavior of the application in terms of problem domain, independent of the user interface
- Directly **manages the data, logic and the rules** of the application
- **Behavior**
  - Separate and reuse logic
  - Single tasks without requiring inheritance
- **Callback** methods
  - Implement some **logic before or after model operation**
  - beforeFind → afterFind
  - beforeValidate → afterValidate
  - beforeSave → afterSave
  - beforeDelete → afterDelete

Relationship	Association Type	Example
one to one	hasOne	A user has one profile.
one to many	hasMany	A user can have multiple recipes.
many to one	belongsTo	Many recipes belong to a user.
many to many	hasAndBelongsToMany	Recipes have, and belong to, many ingredients.

Model associations

- Links between models
- **Aliases** for each model **MUST be unique** across the application
- Usually use belongsTo with hasMany, instead of using hasAndBelongsToMany (HABTM)

Object Relational Mapping (**ORM**)

- Map with Relational database management system (RDBMS) and Object-oriented Objects
- No need to write SQL directly

View in MVC

- Can be any output **representation of information**
- **Layout**
  - Different action can apply different layouts
    - E.g.
      - For admin dashboard → “admin” layout
      - For end-users → default layout
- **Element**
  - **Reusable views** with the same information / contents
    - E.g.
      - Header / footer / sidebar / menu
      - Content blocks / widget
- **Helper**
  - **Functional classes** for presentation layer
    - Share between views, layouts and elements
    - E.g.
      - FormHelper, HtmlHelper, SessionHelper, TimeHelper

Controller in MVC

- **Accepts input** and **converts it to commands for the model or view**
- Handle interpreting the request data, making sure the correct models are called, and correct response or view is rendered
- **Middle man between Model and View**
- ***Fat Model, Thin Controller***
- Controller actions → URL
    - Format
        - /{:admin}/{:plugin}/:controller/:action/:params
    - E.g.
        - /admin/inventory/stocktakes/edit/1 → Inventory plugin, stocktakes controller, admin_edit action, prarms id = 1
- **Component**
    - **Packages of logic**, shared between controllers
    - E.g.
        - Pagination, Flash, Authentication, Request Handling, Cookie
- **Callback** methods
    - **Request life cycle**
        - beforeFilter → afterFilter → beforeRender

Conventions

- Controller
  - Plural, CamelCased
    - E.g. PeopleController, LatestArticlesController
- Model
  - Singular, CamelCased
    - E.g. Person, LatestArticle
- DB Table
  - Corresponding to its Model
  - Plural, underscored
    - E.g. people, latest_articles
- View
  - getReady() → get_ready.ctp
  - URL : /:controller/get_ready

DB Table structure – Ricky's practice

- Usually used 'id' (int 11) as **Primary key, Auto Increment**
- Usually used 'slug' / 'token' as the mask to protect the 'id'
  - Try **not to disclose the 'id' directly** in Web / URL / API
- Usually added 'status' (tinyint 1) / 'enabled' (tinyint 1) field to control different status of that record
- Usually added 'updated_by' (int 11) and/or 'created_by' (int 11) to log who create / update that record
  - Refer to TrackableBehavior
- Usually added 'updated' (datetime) and/or 'created' (datetime) to log the time when create / update the record
  - Simple function in beforeSave callback

Useful tools

- Generate code with **bake**
  - Bake is a console scripts that help to create any basic ingredients in CakePHP
    - Project, plugin, model, view, controller
  - http://book.cakephp.org/2.0/en/console-and-shells/code-generation-with-bake.html
  - Procedure
    1. Bake project
    2. Bake plugin (if any)
    3. Bake models (in plugin, if any)
       a. Bake will help to do all relationships between models for you
       b. Need to manually check all models are well enough for your use
          i. E.g. you don't want to use HABTM relationship, you have to change it manually before proceed
          ii. E.g. if there is any fields ended with "id" that is NOT related to any model within your application, you have to remove this baked relationship from model
    4. Bake controllers (in plugin, if any)
       a. You can provide parameters to help generate actions in a faster way
          i. Routing prefix, e.g. 'admin' will help to bake admin_index, admin_add etc.
          ii. Public e.g. simply bake index, add etc.
          iii. Theme / template
       b. Check once before proceed
          i. All actions you want are baked correctly
    5. Bake views (in plugin, if any)
       a. It will automatically find all actions in the controllers, generate the corresponding views

References

1. CakePHP request life cycle
   - http://stackoverflow.com/a/20534204
2. CakePHP cheat sheet
   - http://demos.developerhints.com/cakephp-cheat-sheet/
3. Controller Request life cycle callbacks
   - http://programming-tips.in/cakephp-request-life-cycle-callbacks/