



PostgreSQL

CakePHP

How to CakePHP Version 3.x










Create By Pakgon R&D Team



CakePHP3.x Content Outline



Day 1








-  Introduction of Framework
-  OOP Concept
-  MVC Framework
-  Introduction of CakePHP3.x
-  CakePHP3.x Convention
-  Team Convention
-  Setup and Virtual Host Configure
-  CakePHP3.x Folder Structure
-  CakePHP3.x Lifecycle



CakePHP3.x Content Outline



Day 2






-  CakePHP3.x Routing
-  CakePHP3.x Path and URL
-  CakePHP3.x Controller
-  CakePHP3.x Model (Entity, Table, Relation)
-  CakePHP3.x Validation and Rule
-  CakePHP3.x View
-  CakePHP3.x Resource (CSS, JS, IMG, File)



CakePHP3.x Content Outline



Day 3

-  CakePHP3.x Helper
-  CakePHP3.x Component
-  CakePHP3.x File upload
-  CakePHP3.x Authentication
-  CakePHP3.x Pagination

What is CakePHP3.x MVC Framework ?

- CakePHP คือ Framework ที่พัฒนาด้วยภาษา PHP ซึ่งเป็น Open-Source โดยมีการออกแบบโปรแกรมแบบ MVC (Model-view-controller) และ ORM (Object-Relational Mapping) เข้ามาเป็นมาตรฐาน และข้อตกลงในการพัฒนาโปรแกรมด้วย Pattern ที่เป็น OOP (Object Oriented Programing) และ MVC

What is CakePHP3.x MVC Framework ?

- ความหมายของ Framework ก็คือ กรอบของการทำงาน แปลตรงตัวเลย Frame + work แต่ถ้ามองในแง่โปรแกรม ก็จะเป็น ข้อกำหนดหรือขอบเขตในการเขียนโปรแกรมให้ เป็นไปในรูปแบบเดียวกันและสามารถนำไปเขียนเพิ่มเพื่อตอบสนองความต้องการอย่างใดอย่างหนึ่งได้ ดังนั้น Cake PHP framework ก็จะมี class และ function ต่างๆ ให้เราได้ เรียกใช้ได้ โดยที่เราไม่ต้องไปเสียเวลาเขียนขึ้นมาเอง ซึ่งจะทำให้เกิดความรวดเร็วในการพัฒนา Web Application ได้อย่างมาก

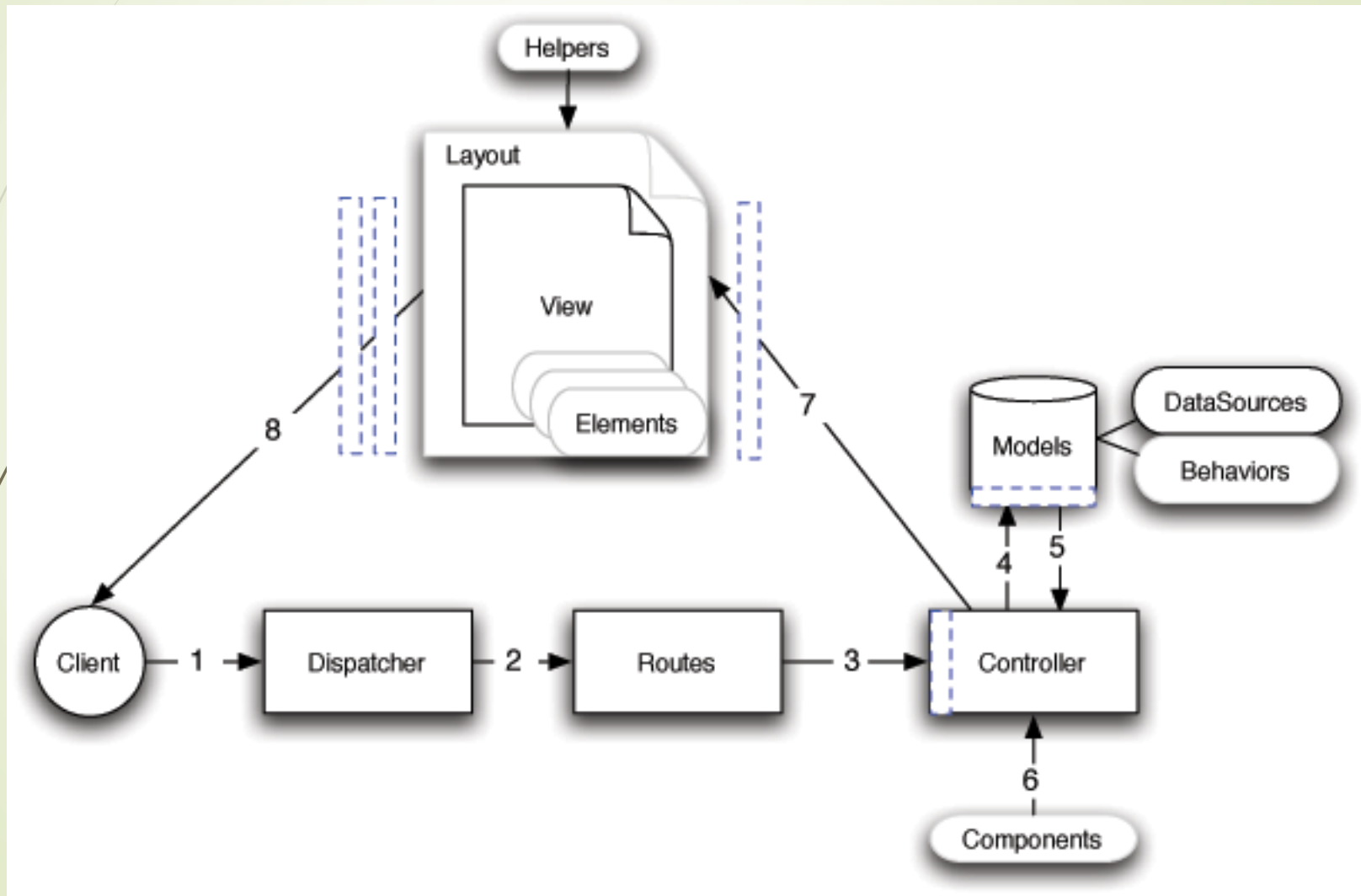
สิ่งที่ควรรู้ก่อนใช้งาน CakePHP3.x

- PHP
- Composer PHP Dependency management
- HTML & HTML5
- CSS & CSS3 Bootstrap
- SQL, MySQL, PostgreSQL etc.
- OOP Concept (Object Oriented Programming)
- Java script, JSON, XML and Ajax Concept
- HTTP and HTTP request

What are benefit of CakePHP3.x

- มีลิขสิทธิ์ที่ยืดหยุ่น
- ทำงานร่วมกับ PHP version 4 และ 5
- รวมคำสั่ง CRUD (Create, Read, Update and Delete) สำหรับ Database ที่ทำให้ง่ายต่อการคิวรี
- ใช้สถาปัตยกรรม MVC (Model-View-Controller)
- มี Helper ช่วยพัฒนาได้ง่ายและรวดเร็ว เช่น AJAX, JavaScript, HTML Forms and อื่นๆ
- มีความปลอดภัยในเรื่อง Email, Cookie, Security, Session, และรองรับการทำงานในรูปแบบ Components
- มีการเรียก URL ที่ส่งค่าแบบสวยงาม เหมาะกับการทำ SEO (Search Engine Optimization)

MVC Framework



MVC Framework (Continuous)

MVC ย่อมาจาก Model-View-Controller คือ สถาปัตยกรรมซอฟต์แวร์ (software architecture) ที่มีการแบ่งแยกระบบออกเป็น 3 ส่วนหลักๆ ได้แก่

- 1) Data Model => Model (M)
- 2) UX, UI => View (V)
- 3) Logic and Process => Controller (C)

MVC Framework (Continuous)

- **M = Models** จะทำหน้าที่เป็นตัวแทนของ ตารางข้อมูล มีการเชื่อมต่อกับตารางของฐานข้อมูล และกำหนดความสัมพันธ์ของแต่ละตารางไว้ที่นี่ นามสกุลไฟล์จะเป็น .php เช่น
src/Model/Entity/User.php คือทำหน้าที่เชื่อมต่อกับ table users จาก Class
src/Model/Table/UserTable.php
- **C = Controller** จะทำหน้าที่ในการประมวลผล คำสั่ง, ฟังก์ชัน และ เงื่อนไขต่างๆ เปรียบเหมือน Boss สั่งให้ส่วนอื่นๆ ทำงาน และตัวเองมาประมวลผลต่อ นามสกุลไฟล์จะเป็น .php เช่น src/Controller/UsersController.php
- **V = View** จะทำหน้าที่เป็นส่วนติดต่อกับผู้ใช้งาน ซึ่งจะถูกเก็บไว้ใน folder src/Template นามสกุลไฟล์จะเป็น .ctp เช่น src/Template/Users/index.ctp

CakePHP3.x Change Feature

Better performance

CakePHP3.x ประกอบด้วยการปรับปรุงประสิทธิภาพการทำงานของกระบวนการ bootstrap, กระบวนการกำหนดเส้นทางหรือ Routing และกระบวนการต่างๆ สำหรับการสร้าง Class ต้นแบบ และ ส่วนของฟังก์ชันช่วยเหลือต่างๆ

Enhanced components and helpers

CakePHP3.x ให้การสนับสนุนที่เพิ่มขึ้นสำหรับ Flash message ด้วย FlashHelper และ FlashComponent แบบใหม่ นอกจากนี้แล้วส่วนของ CookieComponent ยังได้รับการปรับปรุงแล้วทำให้เราสามารถแยกการกำหนดค่าของพื้นที่ Cookie และการจัดการข้อมูลของ Cookie ได้ง่ายขึ้น

Improved session management

การจัดการเซสชันเป็น static class ซึ่งใน CakePHP ได้มีการพบปัญหาการใช้งาน Session ใน Version2.x สำหรับ Version3.x เราสามารถเข้าถึงข้อมูลที่เก็บอยู่ใน Session ได้ง่ายและปลอดภัยขึ้นด้วยคำสั่ง `$this->request->session()` การเปลี่ยนแปลงนี้ทำให้ การเข้าถึงข้อมูลที่เก็บอยู่ใน Session สามารถทำได้ง่าย และยังง่ายต่อการทดสอบและทำให้ CakePHP สามารถใช้ PHPUnit 4.x ได้

CakePHP3.x Change Feature (Continuous)

➤ Improved consistency of conventions

มีการปรับโครงสร้างของ Application ให้สอดคล้องกับโครงสร้าง Directory ของ Plugin เดียวกัน เพื่อความสอดคล้องของระบบหลัก และ Dependency ต่างๆ

➤ Themes and plugins merged

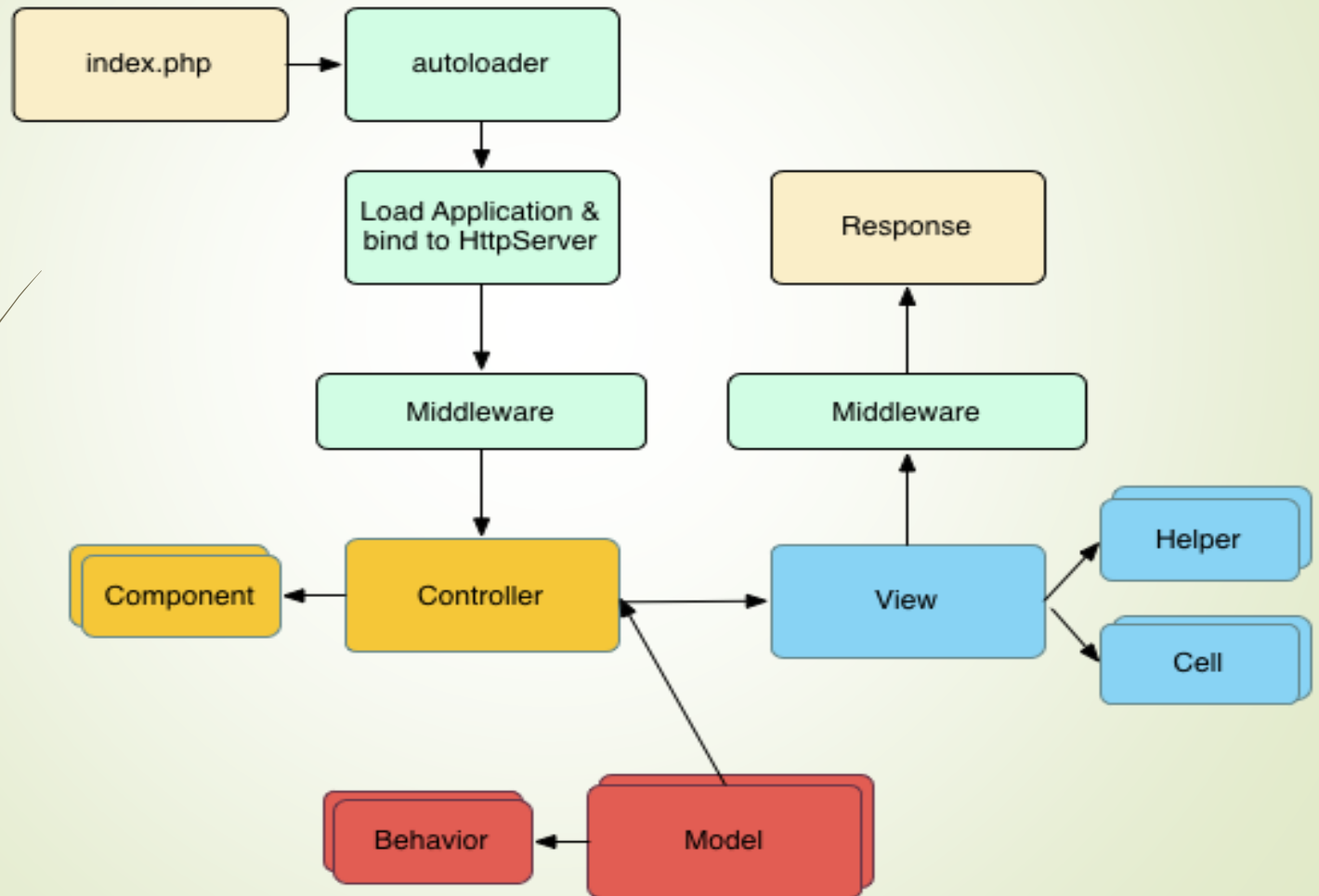
เป้าหมายหลักของ CakePHP3.x คือการทำให้ธีม มีความฉลาดมากขึ้น ซึ่งการทำงานไปสู่เป้าหมายดังกล่าวได้นั้น ได้กลายมาเป็นที่ชัดเจนแล้วว่าสิ่งที่ต้องการจริงๆ คือเพื่อให้ธีม มีความสามารถ เช่นเดียวกับปลั๊กอิน ดังนั้นปลั๊กอินใด ๆ จึงสามารถใช้เป็นธีม ซึ่งจะช่วยลดความยุ่งยาก และ ซับซ้อนในการบรรจุและแจกจ่ายซ้ำได้

CakePHP3.x Change Feature (Continuous)

ORM Improvements

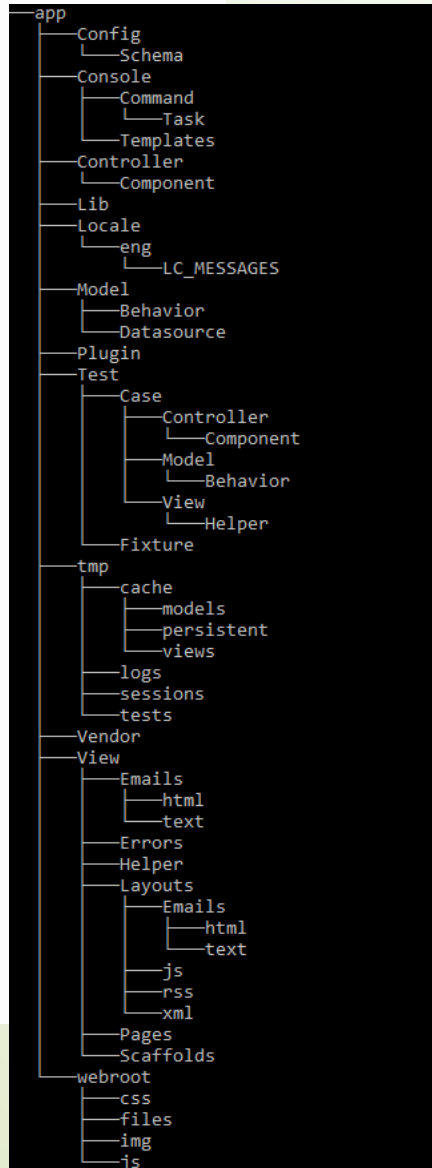
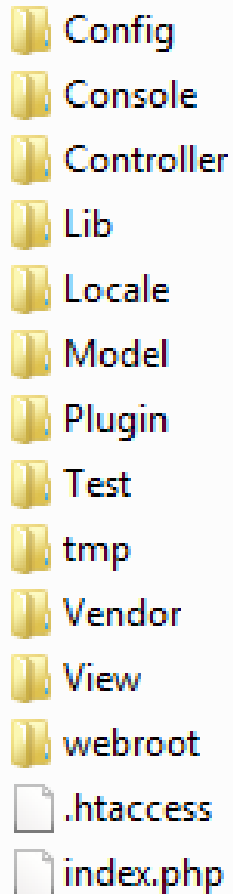
ORM (Object Relational Mapping) มีการเปลี่ยนแปลง API หลายรายการใน ORM (การเฝ้าวัตถุเชิงสัมพันธ์) ทำให้ง่ายต่อการใช้งาน และลดเวลาในการเรียนรู้ให้น้อยลง เพิ่มเวลาในการพัฒนาให้มากขึ้น และยังประมวลผลการมุลได้เร็วขึ้นด้วย

CakePHP3.x Lift Cycle

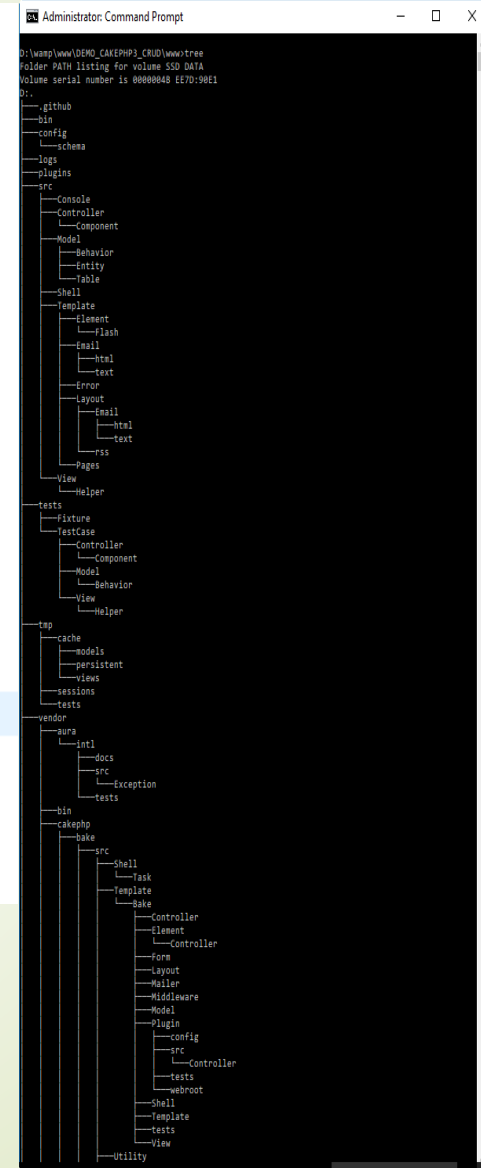
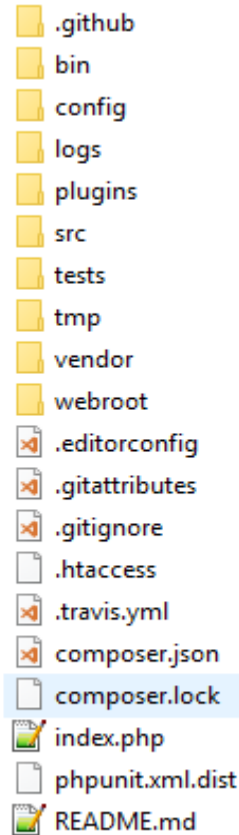


CakePHP2.x VS CakePHP3.x Folder Structure

CakePHP v2.x



CakePHP v3.x



CakePHP3.x folder structure (Continuous)

- **bin** folder holds the Cake console executables.
- **config** folder holds the (few) Configuration files CakePHP uses.
- **plugins** folder is where the Plugins your application uses are stored.
- **logs** folder normally contains your log files.
- **src** folder will be where your application's files will be placed.
- **tmp** folder is where CakePHP stores temporary data.
- **vendor** folder is where CakePHP and other application dependencies will be installed by composer.
- **webroot** directory is the public document root of your application.

CakePHP3.x “src” folder structure

- CakePHP’s src folder is where you will do most of your application development. Let’s look a little closer at the folders inside src.
- **Controller** Contains your application’s controllers and their components.
- **Locale** Stores string files for internationalization.
- **Model** Contains your application’s tables, entities and behaviors.
- **Shell** Contains the console commands and console tasks for your application. For more information see Shells, Tasks & Console Tools.
- **View** Presentational classes are placed here: views, cells, helpers.
- **Template** Presentational files are placed here: elements, error pages, layouts, and view template files.

CakePHP3.x URL & Paths + parameters

- Cake uses url to pass parameters
- Apache mod_rewrite converts url into script name and parameters
- <http://www.example.com/ControllerName/action/param1/param2/.../paramsN>
- <http://demo.cakephp3.clud.local/Users/index>
- <http://demo.cakephp3.clud.local/Users/add>
- <http://demo.cakephp3.clud.local/Users/edit/2>
- <http://demo.cakephp3.clud.local/Users/view/2>
- <http://demo.cakephp3.clud.local/Users/delete/2>

URL to controller action mapping using default routes:

➤ URL: /Monkeys/jump

Mapping: MonkeysController->jump();

➤ URL: /Products

Mapping: ProductsController->index();

➤ URL: /Tasks/view/45

Mapping: TasksController->view(45);

➤ URL: /Donations/view/recent/2001

Mapping: DonationsController->view('recent', '2001');

URL to controller action mapping using default routes:

`$this->passArgs, $this->params`

➤ URL: /contents/view/chapter:**models**/section:**associations**

Mapping: ContentsController->view();

➤ `$this->passedArgs['chapter'] = 'models';`

➤ `$this->passedArgs['section'] = 'associations';`

➤ `$this->params['named']['chapter'] = 'models';`

➤ `$this->params['named']['section'] = 'associations';`

OOP Key, Main Core OOP

1. Extends ?
2. Polymorphic ?
3. Encapsulate (Accessor) ?
4. Overload ?
5. Overwrite ?
6. Abstract ?
7. Interface ?

➡ Homework

Explain for the main Core of the OOP and OOP Concept ?

OOP in PHP

- Limited support in PHP > 5.3
- Much better support in PHP >= 5.4.x
- Simpler than Java OOP

```
class SomeClass {  
    function func() {  
        ....  
    }  
}
```

```
SomeClass s = new someClass();
```

```
s->func();
```


Camel case, Capital camel case and Snake case

- **Camel** = อูฐ นึกถึงตัวอูฐ จะมีลักษณะสูงๆ ต่ำๆ โดยตัวอักษรตัวแรกของประโยคเป็นตัวอักษรภาษาอังกฤษพิมพ์เล็ก โดยทุกคำจะอยู่ติดกันหมดขึ้นคำใหม่ด้วย ตัวอักษรพิมพ์ใหญ่
- **Capital Camel** = อูฐ นึกถึงตัวอูฐ จะมีลักษณะสูงๆ ต่ำๆ โดยตัวอักษรตัวแรกของประโยคเป็นตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ โดยทุกคำจะอยู่ติดกันหมดขึ้นคำใหม่ด้วย ตัวอักษรพิมพ์ใหญ่
- **Snake case** = งู นึกถึงงู ตัวยาว หัวถึงหาง เท่ากันหมด โดยเป็นตัวอักษรภาษาอังกฤษพิมพ์เล็กทั้งหมด เชื่อมแต่ละคำเข้าด้วยกันด้วยเครื่องหมาย “_”

➤ Example (User + Comment)

Camel case = userComment

Capital Camel case = UserComment

Snake case = user_comment

Camel case, Capital camel case and Snake case (Continuous)

Function and Class name in Camel Case

Example:

Class: User(Model), Users (Controller), Comment(Model), Comments (Controller)

Function: index, view, add, edit, delete, viewUserComment

File: User.php, Users.php, add.ctp, index.cth, edit.ctpview.ctp, view_user_comment.ctp

Camel case, Capital Camel case and Snake case (Continuous)

- users (Table)
- User.php (Model Entity) => Class name = file name with out extension
- UsersTable.php (Model Table) => Class name = file name with out extension
- UsersController.php => Class name = file name without suffix Controller and extension
- Function (index(), view(), add(), edit(), delete(), viewUserComment())
- View: Template/Users/add.ctp, Template/Users/view.ctp , Template/Users/add.ctp, Template/Users/edit.ctp, delete function without view if success flash and redirect to the list page, Template/Users/view_user_comment.ctp

CakePHP3.x Convention

- กฎข้อที่ 1 : **การตั้งชื่อตาราง** การตั้งชื่อตารางในฐานข้อมูล จะต้องเป็นชื่อภาษาอังกฤษ ตัวพิมพ์เล็กทั้งหมด และต้องเป็นพหูพจน์ ซึ่งพหูพจน์โดยทั่วไปให้เติม s หรือ es ตามหลักไวยากรณ์ ของภาษาอังกฤษ เช่น ตารางข้อมูลผู้ใช้งาน (user) จะได้เป็น users และ ตารางเก็บคอมเมนต์ผู้ใช้งาน (comment) จะได้เป็น comments เป็นต้น
- กฎข้อที่ 2 : **การตั้งชื่อฟิลด์ที่เป็นคีย์หลัก (Primary Key)** การตั้งชื่อฟิลด์ที่เป็นคีย์หลัก (PK) ของแต่ละตารางให้ตั้งชื่อฟิลด์นั้นว่า id เสมอ โดย Datatype เป็นอะไรก็ได้ แต่ต้อง Unique ตามหลัก database integrity
- กฎข้อที่ 3 : **การตั้งชื่อฟิลด์ที่เป็นคีย์รอง (Foreign Key)** การตั้งชื่อฟิลด์ที่เป็นคีย์รอง (FK) ของแต่ละตารางให้ตั้งชื่อฟิลด์โดยใช้ชื่อ ตารางที่เป็นคีย์หลัก_id แต่ชื่อตารางที่เป็นคีย์หลักให้เปลี่ยนจากพหูพจน์เป็นเอกพจน์ด้วย เช่น user_id และ comment_id

CakePHP3.x Convention (Continuous)

- กฎข้อที่ 4: **การตั้งชื่อไฟล์ของ Controller** การตั้งชื่อไฟล์ของ controller นั้นจะใช้กฎเดียวกันกับการตั้งชื่อตาราง แต่ต่างกันตรงที่ชื่อไฟล์ของ Controller จะต้องเป็นไปตามรูปแบบนี้คือ **TableNameController.php** เช่น UsersController.php, CommentsController.php เป็นต้น และชื่อ Class ต้องเป็นชื่อเดียวกับชื่อไฟล์ด้วยเช่นกัน
- กฎข้อที่ 5: **การตั้งชื่อไฟล์ของ Model** การตั้งชื่อไฟล์ของ Model นั้นจะใช้ชื่อเดียวกับชื่อของตาราง ที่ Model ที่สร้างขึ้นเพื่อเป็นตัวแทนของ ตารางข้อมูลนั้น (แต่อยู่ในรูปแบบเอกพจน์) ในการตั้งชื่อไฟล์ เช่น User.php, Comment.php เป็นต้น และชื่อ Class ต้องเป็นชื่อเดียวกับชื่อไฟล์ด้วยเช่นกัน
- กฎข้อที่ 6: **การตั้งชื่อไฟล์ของ View** การตั้งชื่อไฟล์ของ View หรือส่วนแสดงผลนั้นจะใช้ชื่อของตาราง(พหูพจน์) ในการตั้งชื่อไฟล์ ซึ่งก็คือชื่อเดียวกับชื่อ Controller แต่ตัดส่วน suffix Controller ออก เช่น Users เป็นต้น



Team Convention

Server Environment Requirements

- Web server like Apache with mod_rewrite enabled
- PHP version 5.6.3 or greater I recommend 7.x in reason full support in OOP
- XAMPP (Apache, MySQL, PHP) I recommended XAMPP v 7.1.0 (pure PHP7.x)
- WAMP (Apache, MySQL, PHP5.x, PHP7.x) for many MA support I recommended WAMP v 3.1.0 (Bundle PHP5.x and PHP7.x)
- Database engine (Right now, there is support for MySQL 5.1+, SQLite 3, PostgreSQL 8+, SQLServer 2008+, Oracle (through a community plugin)).

Install CakePHP3.5.x

MINGW64; d:/wamp/www

```
dk_antiKorNBKANTTKORNB_MINGW64 /d/wamp/www
$ composer create-project --prefer-dist cakephp/app:"3.5.*" DEMO_CAKEPHP3_CRUD/www
Installing cakephp/app (3.5.0)
- Installing cakephp/app (3.5.0): Loading from cache
Created project in DEMO_CAKEPHP3_CRUD/www
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 39 installs, 0 updates, 0 removals
- Installing cakephp/plugin-installer (1.0.0): Loading from cache
- Installing aura/intl (3.0.0): Loading from cache
- Installing mobiledetect/mobiledetectlib (2.8.26): Loading from cache
- Installing psr/http-message (1.0.1): Loading from cache
- Installing zendframework/zend-diactoros (1.6.1): Loading from cache
- Installing psr/log (1.0.2): Loading from cache
- Installing cakephp/chronos (1.1.2): Loading from cache
- Installing cakephp/cakephp (3.5.4): Loading from cache
- Installing symfony/polyfill-mbstring (v1.6.0): Loading from cache
- Installing symfony/yaml (v3.3.10): Loading from cache
- Installing symfony/debug (v3.3.10): Loading from cache
- Installing symfony/console (v3.3.10): Loading from cache
- Installing symfony/filesystem (v3.3.10): Loading from cache
- Installing symfony/config (v3.3.10): Loading from cache
- Installing robmorgan/phinx (v0.8.1): Loading from cache
- Installing cakephp/migrations (1.7.1): Loading from cache
- Installing ml/env (2.1.0): Loading from cache
- Installing josegonzalez/dotenv (2.1.0): Loading from cache
- Installing jakub-ondarka/php-console-color (0.1): Loading from cache
- Installing jakub-ondarka/php-console-highlighter (v0.3.2): Loading from cache
- Installing dnoegel/php-xdg-base-dir (0.1): Loading from cache
- Installing nikic/php-parser (v3.1.1): Loading from cache
- Installing symfony/var-dumper (v3.3.10): Loading from cache
- Installing psy/psysh (v0.8.13): Loading from cache
- Installing jdorn/sql-formatter (v1.2.17): Loading from cache
- Installing symfony/process (v3.3.10): Loading from cache
- Installing symfony/finder (v3.3.10): Loading from cache
- Installing seld/phar-utils (1.0.1): Loading from cache
- Installing seld/jsonlint (1.6.1): Loading from cache
- Installing seld/cli-prompt (1.0.3): Loading from cache
- Installing justinrainbow/json-schema (5.2.6): Loading from cache
- Installing composer/spdx-licenses (1.1.6): Loading from cache
- Installing composer/semver (1.4.2): Loading from cache
- Installing composer/ca-bundle (1.0.8): Loading from cache
- Installing composer/composer (1.5.2): Loading from cache
- Installing cakephp/debug-kit (3.11.1): Loading from cache
- Installing cakephp/bake (1.4.4): Loading from cache
- Installing squizlabs/php_codesniffer (3.1.1): Loading from cache
- Installing cakephp/cakephp-codesniffer (3.0.1): Loading from cache
cakephp/app suggests installing markstory/asset_compress (An asset compression plugin which provides file concatenation and a flexible filter system for preprocessing and minification.)
cakephp/app suggests installing dereuromark/cakephp-ide-helper (After baking your code, this keeps your annotations in sync with the code evolving from there on for maximum IDE and PHPStan compatibility.)
cakephp/app suggests installing phpunit/phpunit (Allows automated tests to be run without system-wide install.)
cakephp/cakephp suggests installing lib-ICU (The intl PHP library, to use Text::transliterate() or Text::slug())
symfony/console suggests installing symfony/event-dispatcher ()
ml/env suggests installing ml/vars (For loading of configs)
symfony/var-dumper suggests installing ext-symfony_debug ()
psy/psysh suggests installing ext-pcntl (Enabling the PCNTL extension makes PsySH a lot happier :))
psy/psysh suggests installing ext-posix (If you have PCNTL, you'll want the POSIX extension as well.)
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)
cakephp/debug-kit suggests installing ext-sqlite (DebugKit needs to store panel data in a database. SQLite is simple and easy to use.)
Writing lock file
Generating autoload files
> Cake\Composer\Installer\PluginInstaller::postAutoLoadDump
> App\Console\Installer::postInstall
Created 'config/app.php' file
Set Folder Permissions ? (Default to Y) [Y,n]? Y
Updated Security.salt value in config/app.php
```

Setup for Visual Host (VHost)

➡ Adding for two line at below to => C:\Windows\System32\drivers\etc\hosts

127.0.0.1 demo.cakephp3.clud.local

::1 demo.cakephp3.clud.local

VHost for Apache

Adding for each line at below to => D:\wamp\bin\apache\apache2.4.27\conf\extra\httpd-vhosts.conf

```
<VirtualHost *:80>
```

```
    ServerName demo.cakephp3.clud.local
```

```
    ServerAlias demo.cakephp3.clud.local
```

```
    DocumentRoot "${INSTALL_DIR}/www/DEMO_CAKEPHP3_CRUD/www/"
```

```
    ErrorLog "logs/demo.cakephp3.clud.local-error.log"
```

```
    CustomLog "logs/demo.cakephp3.clud.local-access.log" common
```

```
    <Directory "${INSTALL_DIR}/www/DEMO_CAKEPHP3_CRUD/www/">
```

```
        Options +Indexes +Includes +FollowSymLinks +MultiViews
```

```
        AllowOverride All
```

```
        Require local
```

```
    </Directory>
```

```
</VirtualHost>
```



CakePHP3.x Model

- Typically Represent Database Table.
- Define Relationships Between Tables.
- CRUD (Create, Retrieve, Update, Delete).

CakePHP3.x Model (Continuous)

Database support :

MySQL 5.1+

SQLite 3

PostgreSQL 8+

SQLServer 2008+

Oracle (through a community plugin)

CakePHP3.x Database Connection.

Config/app.php

```
'Datasources' => [  
    'default' => [  
        'className' => 'Cake\Database\Connection',  
        'driver' => 'Cake\Database\Driver\Mysql',  
        'persistent' => false,  
        'host' => 'localhost',  
        'username' => 'my_app',  
        'password' => 'sekret',  
        'database' => 'my_app',  
        'encoding' => 'utf8',  
        'timezone' => 'UTC',  
        'cacheMetadata' => true,  
    ]  
]
```



CakePHP3.x ConnectionManager.

```
use Cake\Datasource\ConnectionManager;  
  
$conn = ConnectionManager::get('default');
```



CakePHP3.x Create Connection on Run time

```
ConnectionManager::config('my_connection', $config);
```

```
$conn = ConnectionManager::get('my_connection');
```

CakePHP3.x Running select statements

```
use Cake\Datasource\ConnectionManager;
```

```
$connection = ConnectionManager::get('default');
```

```
$results = $connection->execute('SELECT * FROM articles')->fetchAll('assoc');
```

```
$results = $connection
```

```
->execute('SELECT * FROM articles WHERE id = :id', ['id' => 1])
```

```
->fetchAll('assoc');
```

CakePHP3.x Running select statements (Continuous)

```
$results = $connection
```

```
->newQuery()
```

```
->select('*')
```

```
->from('articles')
```

```
->where(['created >' => new DateTime('1 day ago'), ['created' => 'datetime']])
```

```
->order(['title' => 'DESC'])
```

```
->execute()
```

```
->fetchAll('assoc');
```


CakePHP3.x Running insert statements

```
use Cake\Datasource\ConnectionManager;

$connection = ConnectionManager::get('default');
$connection->insert('articles', [
    'title' => 'A New Article',
    'created' => new DateTime('now')
], ['created' => 'datetime']);
```



CakePHP3.x Running update statements

```
use Cake\Datasource\ConnectionManager;
```

```
$connection = ConnectionManager::get('default');
```

```
$connection->update('articles', ['title' => 'New title'], ['id' => 10]);
```

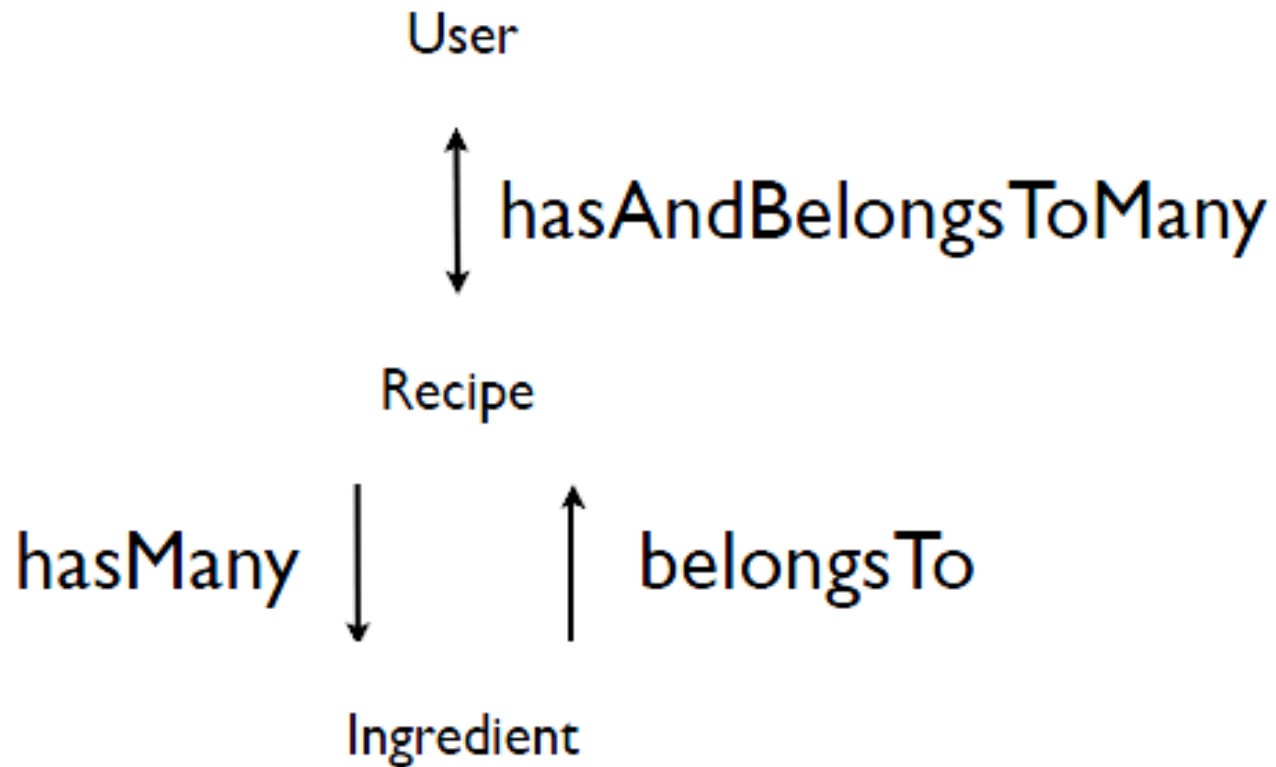
CakePHP3.x Running delete statements

```
use Cake\Datasource\ConnectionManager;
```

```
$connection = ConnectionManager::get('default');
```

```
$connection->delete('articles', ['id' => 10]);
```

CakePHP3.x Model Associations



CakePHP3.x Table Conventions

User



Recipe



Ingredient

users

id
name

recipes

id
name

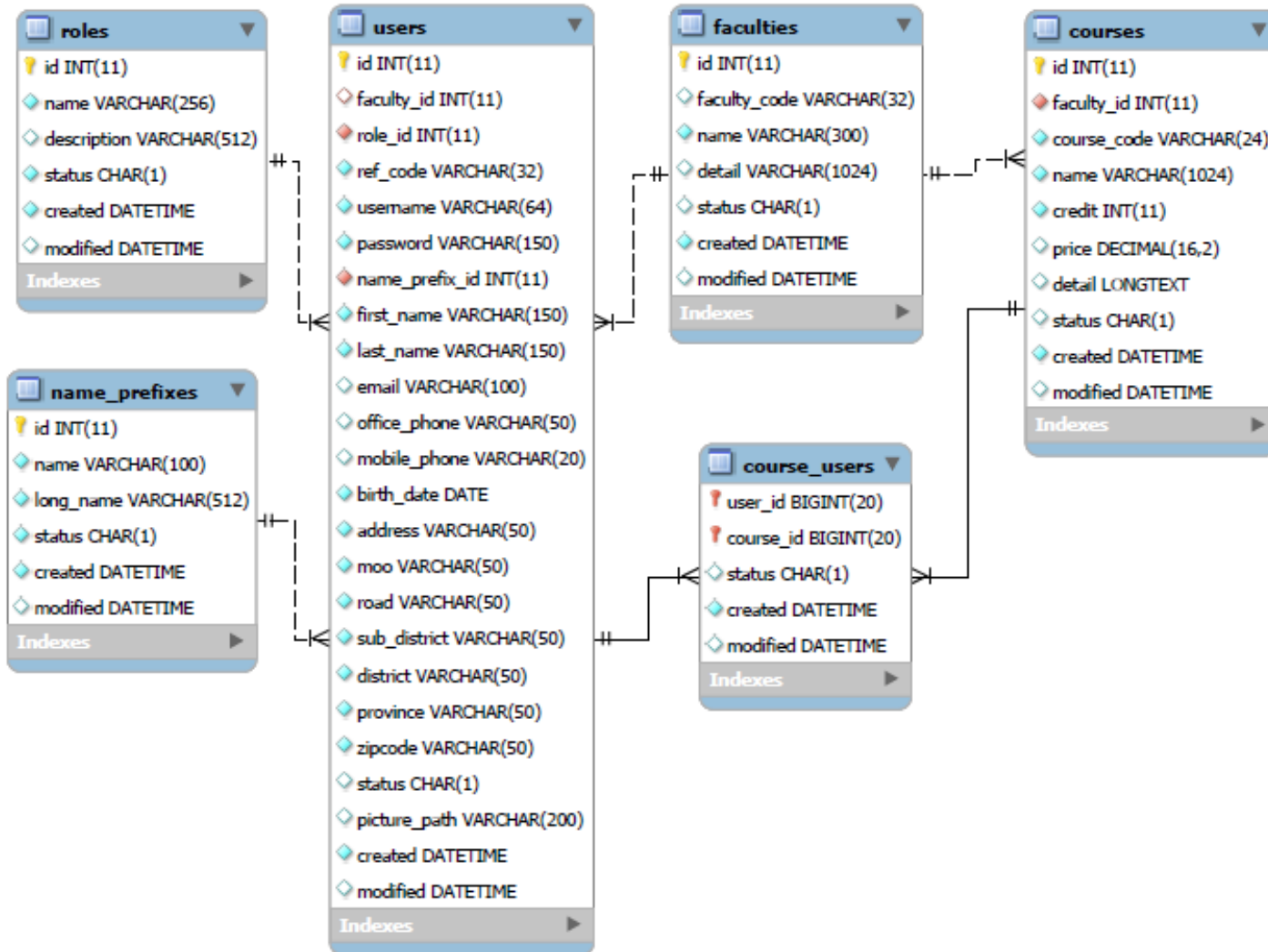
ingredients

id
recipe_id
name

recipes_users

recipe_id
user_id

CakePHP3.x Table Conventions



CakePHP3.x Relationship Types

Association		
Relationship	Type	Example
one to one	hasOne	A user has one profile.
one to many	hasMany	A user can have multiple articles.
many to one	belongsTo	Many articles belong to a user.
many to many	belongsToMany	Tags belong to many articles.

CakePHP3.x HasOne Relations

hasOne: the *other* model contains the foreign key.

Relation	Schema
Users hasOne Addresses	addresses.user_id
Doctors hasOne Mentors	mentors.doctor_id

CakePHP3.x HasOne Relations (Continuous)

```
class UsersTable extends Table
{
    public function initialize(array $config)
    {
        $this->hasOne('Addresses');
    }
}
```

CakePHP3.x HasOne Relations (Continuous)

```
class UsersTable extends Table
{
    public function initialize(array $config)
    {
        $this->hasOne('Addresses')
            ->setName('Addresses')
            ->setConditions(['Addresses.primary' => '1'])
            ->setDependent(true);
    }
}
```

CakePHP3.x HasMany Relations

hasMany: the *other* model contains the foreign key.

Relation	Schema
Article hasMany Comment	Comment.article_id
Product hasMany Option	Option.product_id
Doctor hasMany Patient	Patient.doctor_id

CakePHP3.x HasMany Relations (Continuous)



```
class ArticlesTable extends Table
```

```
{
```

```
    public function initialize(array $config)
```

```
    {
```

```
        $this->hasMany('Comments')
```

```
        ->setConditions(['approved' => true]);
```

```
        $this->hasMany('UnapprovedComments', [
```

```
            'className' => 'Comments'
```

```
        ])
```

```
        ->setConditions(['approved' => false])
```

```
        ->setProperty('unapproved_comments');
```

```
    }
```



CakePHP3.x BelongsTo Relations

belongsTo: the *current* model contains the foreign key.

Relation	Schema
Addresses belongsTo Users	addresses.user_id
Mentors belongsTo Doctors	mentors.doctor_id



If a Table contains a foreign key, it belongs to the other Table.

CakePHP3.x BelongTo Relations

```
class ArticlesTable extends Table
{
    public function initialize(array $config)
    {
        $this->belongsTo('Authors', [
            'className' => 'Publishing.Authors'
        ])
        ->setForeignKey('authorid')
        ->setProperty('person');
    }
}
```

CakePHP3.x BelongTo Relations (Continuous)

```
class ArticlesTable extends Table
{
    public function initialize(array $config)
    {
        $this->belongsTo('Authors', [
            'className' => 'Publishing.Authors',
            'foreignKey' => 'authorid',
            'propertyName' => 'person'
        ]);
    }
}
```

CakePHP3.x Relations (Many to Many) HABTM

belongsToMany requires a separate join table that includes both *model* names.

Relationship	Join Table Fields
Article belongsToMany Tag	articles_tags.id, articles_tags.tag_id, articles_tags.article_id
Patient belongsToMany Doctor	doctors_patients.id, doctors_patients.doctor_id, doctors_patients.patient_id.

CakePHP3.x for HABTM Relations (Continuous)

```
class TagsTable extends Table
{

    public function initialize(array $config)
    {

        $this->belongsToMany('Articles', [

            'joinTable' => 'articles_tags',

        ]);

    }

}
```

Join type

- LEFT Join
- RIGHT Join
- INNER Join
- **Homework**

How to difference between all SQL join type explain ?

CakePHP3.x Table Relation

Homework

How many database relation and may some relation has problem and explain for how you result them Explain ?

CakePHP3.x Model

➤ The following are core find types:

➤ `$this->SomeModel->find('type', []);`

1. first

2. all

3. count

4. list

CakePHP3.x Model

- Magic Find Type

- **findAllBy<fieldName>**

- **findAllBy<fieldName>**(string \$value, array \$fields, array \$order, int \$limit, int \$page, int \$recursive)

- **findBy**

- **findBy<fieldName>**(string \$value[, mixed \$fields[, mixed \$order]]);

CakePHP3.x Insert & Update

- Insert records: adding new records.

```
$this->SomeModel->create();
```

```
$this->SomeModel-
```

```
>save(['SomeModel'=>['someField1'=>'SomeValue1','someField2'=>'someValue2']]);
```

- Update a record where id equals 3: no new records.

```
➤ $this->SomeModel->id = 3;
```

```
➤ $this->SomeModel-
```

```
>save(['SomeModel'=>['someField1'=>'SomeValue1','someField2'=>'someValue2']]);
```

CakePHP3.x Delete

➤ `$this->SomeModel->delete($id);`

➤ **Same same :**

`$this->SomeModel->id = $id;`

`$this->SomeModel->delete();`

CakePHP3.x Controller (src/Controller/UsersController.php)

```
1 <?php
2
3 App::uses('AppController', 'Controller');
4 class UsersController extends AppController {
5     public $name = 'Users';
6     public $components = array('Paginator', 'Flash', 'Session');
7     public $uses = array('User', 'Role');
8
9     public function index() {
10         $this->User->recursive = 0;
11         $this->set('users', $this->Paginator->paginate('User'));
12     }
13
14     public function add() {
15         //Insert your code logic here
16     }
17
18     public function view($id = null) {
19         //Insert your code logic here
20     }
21
22     public function edit($id = null) {
23         //Insert your code logic here
24     }
25
26     public function delete($id = null) {
27         //Insert your code logic here
28     }
29 }
```


CakePHP3.x View (src/Template/Users/index.ctp)

```
1 <div class="box box-active">
2   <div class="box-header with-border">
3     <h3 class="box-title"><?php echo __('User'); ?></h3>
4   </div>
5   <div class="box-body">
6     <table cellpadding="0" cellspacing="0" class="table-short-information">
7       <thead>
8         <tr>
9           <th class="nindex"><?php echo __('#'); ?></th>
10          <th><?php echo $this->Paginator->sort('username'); ?></th>
11          <th><?php echo $this->Paginator->sort('first_name'); ?></th>
12          <th><?php echo $this->Paginator->sort('last_name'); ?></th>
13          <th><?php echo $this->Paginator->sort('email'); ?></th>
14          <th><?php echo $this->Paginator->sort('phone_no'); ?></th>
15          <th><?php echo $this->Paginator->sort('department_id'); ?></th>
16          <th><?php echo $this->Paginator->sort(__('loginfail')); ?></th>
17          <th><?php echo $this->Paginator->sort('role_id'); ?></th>
18          <th><?php echo $this->Paginator->sort('status'); ?></th>
19          <th class="actions"></th>
20        </tr>
21      </thead>
22      <tbody>
23        <?php if (!empty($users)): ?>
24          <?php foreach ($users as $k => $user): ?>
25            <tr>
26              <td class="nindex"><?php echo $this->Paginator->counter("[:start]") + $k; ?></td>
27              <td><?php echo h($user['User']['username']); ?></td>
28              <td><?php echo h($user['User']['first_name']); ?></td>
29              <td><?php echo h($user['User']['last_name']); ?></td>
30              <td><?php echo h($user['User']['email']); ?></td>
31              <td><?php echo h($user['User']['phone_no']); ?></td>
32              <td><?php echo h($department[$user['User']['department_id']]); ?></td>
33              <?php
34              if ($user['User']['loginfail'] < '4') {
35                <?>
36                <td><span class="label label-success"><?php echo __('User') ?></span></td>
37                <?php
38              } else if ($user['User']['loginfail'] >= '4') {
39                <?>
40                <td><span class="label label-danger"><?php echo __('locked') ?></span></td>
41                <?php
42              }
43              <?>
44              <td><?php echo h($role[$user['User']['role_id']]); ?></td>
45              <td><?php echo h($user['User']['status']); ?></td>
46              <td class="actions">
47                <?php echo $this->Html->link(__('View'), "/Users/view/{$user['User']['id']}", array('class' => 'btn btn-warning')); ?>
48              </td>
49            </tr>
50          <?php endforeach; ?>
51        <?php else: ?>
52          <?php echo ' <tr class="text-center notfound"><td colspan="11"> . __('Not found data') . ' </td></tr>'; ?>
53        <?php endif; ?>
54      </tbody>
55    </table>
56  </div>
57 </div>
```

Reference

- Download Cake PHP Cook book
- https://book.cakephp.org/3.0/_downloads/en/CakePHPCookbook.pdf
- <http://book.cakephp.org/3.0/en/controllers.html>
- <http://book.cakephp.org/3.0/en/models.html>
- <http://book.cakephp.org/3.0/en/views.html>

➤ WAMP Version 3.1.0

➤ <https://sourceforge.net/projects/wampserver/files/>

➤ Java JDK Version 1.8 x86-64:

➤ <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

➤ NetBean Editor Version 8.1:

➤ <https://netbeans.org/downloads/> (Your Java JDK needed to install first)

➤ PostgreSQL Version 9.4.9

➤ <http://www.enterprisedb.com/postgresql-949-installers-win64?ls=Crossover&type=Crossover>

➤ Composer

➤ <https://getcomposer.org/Composer-Setup.exe>



Editor and tool

➤ Netbean Plugin

➤ <https://github.com/junichi11/cakephp3-netbeans/releases>



Thank You