

Convenções no CakePHP 3

A equipe do CakePHP é grande admiradora de convenção sobre configuração. Seguindo convenções você recebe funcionalidades gratuitamente e libera a si mesmo do pesadelo de manter arquivos de configuração.

Aviso Importante: Portanto, antes de começar a trabalhar com CakePHP é muito importante conhecer suas convenções para tirar delas enormes vantagens. Caso não as use o CakePHP não será de muita ajuda.

Controllers

Nomes de classes tipo Controller devem estar no plural, ser CamelCase e terminarem com o sufixo Controller.

Exemplos: ClientesController, PeopleController e LatestArticlesController

Actions - Métodos public nos controllers são chamados de actions e se comunicam com views com mesmo nome que eles e extensão .ctp.

Um exemplo: o action Controller/ClientesController/index() é mapeado automaticamente para a view src/Template/Clientes/index.ctp.

Métodos protected ou private não podem ser acessados via Routing.

Considerações sobre URL para nomes de controllers

O ClientesController que está no arquivo ClientesController.php é chamado pelo navegador com:

`http://localhost/aplicacao/clientes`

Nomes de controllers com palavras compostas podem ficar assim:

- /redApples
- /RedApples
- /Red_apples
- /red_apples

Todos resolverão com o controller RedApples.

/red-apples/go-pick resolve para o action RedApplesController::goPick()

Criar links:

```
$this->Html->link('link-title', [  
    'prefix' => 'MyPrefix' // CamelCased  
    'plugin' => 'MyPlugin', // CamelCased  
    'controller' => 'ControllerName', // CamelCased  
    'action' => 'actionName' // camelCased  
])
```

Namespaces

Todas as classes do core do CakePHP agora (3.x) usam namespaces e seguem as especificações de autoload (auto-carregamento) do **PSR-4**.

Por exemplo
src/Cache/Cache.php

tem o namespace
Cake\Cache\Cache

Constantes globais e métodos de helpers como `__()` e `debug()` não usam namespaces por questões de conveniência.

Nomes de arquivos e Classes:

Classe `KissesAndHugsController` está no arquivo `KissesAndHugsController.php`
Classe `ClientesController` está no arquivo `ClientesController.php`.

Alguns exemplos de classes e seus arquivos:

- O Controller class `KissesAndHugsController` deve estar no arquivo com nome `KissesAndHugsController.php`
- O Component class `MyHandyComponent` deve estar no arquivo com nome `MyHandyComponent.php`
- A Table class `OptionValuesTable` deve estar no arquivo com nome `OptionValuesTable.php`
- A Entity class `OptionValue` deve estar no arquivo com nome `OptionValue.php`
- O Behavior class `EspecialyFunkableBehavior` deve estar no arquivo com nome `EspecialyFunkableBehavior.php`
- A View class `SuperSimpleView` would deve estar no arquivo com nome `SuperSimpleView.php`
- O Helper class `BestEverHelper` deve estar no arquivo com nome `BestEverHelper.php`

Model e Bancos de Dados

Nomes de classes Table - são no plural e CamelCase

Chave Primária

Toda tabela, obrigatoriamente deve ter uma chave primária e o nome da chave deve ser `id`.

O Cake tem um recurso online importante para mostrar o plural de nomes:

<http://inflector.cakephp.org/>

Nomes válidos: Clientes, Pelople, BigPeople e ReallyBigPeople

Nomes de tabelas são em minúsculas, no plural e palavras compostas separadas por sublinhado. Nomes de tabelas para os acima:

clientes, people, big_people e really_big_people

A convenção é para usar tabelas e campos com nomes na língua inglesa.
Se por alguma razão precisar usar nomes de tabelas em outro idioma, então você deve usar a classe utility:

Cake\Utility\Inflector

Nomes de campos compostos são separados por sublinhado: first_name.

Nomes de campos são em minúsculas e quando compostos por palavras compostas são separados por sublinhado.

Se usarmos os **campos username e password** (com estes nomes) na tabela users, o Cake deve estar apto para auto-configurar muitas coisas para nós, quando implementando o user login.

Obs.: quando usar autenticação use o tamanho do campo password com varchar(255). Também ajuda adicionar um campo chamado role na tabela users.

Relacionamentos

Chave estrangeira nos relacionamentos hasMany, belongsTo ou hasOne são reconhecidas por default com o nome (singular) da tabela relacionada seguida de "_id".

Exemplos: groups e users. Em users o campo group_id para relacionar.
Relacionamento entre articles e users. Em articles adicionar o campo user_id.

Relacionamentos

Um - Muitos - hasMany

Se relacionamos Articles com User, inserimos um campo user_id em articles.

No model UsersTable

Um Users pode conter muitos (hasMany) Articles

Muitos Articles belongsTo Users

Ver documentação para detalhes.

Relacionamento Muitos para Muitos

Exemplo: Para relacionamento muitos para muitos de platillos com cocineros, criaremos a tabela

```
create table cocineros_platillos(  
    id int unsigned auto_increment primary key,  
    cocinero_id int(11) not null,  
    platillo_id int(11) not null  
);
```

Tipos de Relacionamentos

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.

Para uma classe Bakers teremos uma chave estrangeira assim: baker_id.

Para uma tabela como category_types a chave estrangeira será category_type_id.

Nomes de campos especiais, que levam o Cake a tomar iniciativas importantes para nós:

title

created

modified

Convenções para as Views

As views tem nomes de arquivos em minúsculas com extensão .ctp.

O método getReady() do controller PeopleController está associado ao template/view src/Template/People/get_ready.ctp.

Exemplo geral:

- Database table: "people"
- Table class: "PeopleTable", found at src/Model/Table/PeopleTable.php
- Entity class: "Person", found at src/Model/Entity/Person.php
- Controller class: "PeopleController", found at src/Controller/PeopleController.php
- View template, found at src/Template/People/index.ctp

Usando essas convenções o CakePHP sabe que uma requisição para <http://localhost/aplicativo/people> mapeia para uma chamada da função index() do PeopleController, onde o model Person está automaticamente disponível (e automaticamente vinculado à tabela people do banco) e renderizada para um arquivo src/Template/People/index.ctp. Nenhum desses relacionamentos precisa ser configurado por qualquer meio mas apenas pela criação de arquivos e classes que precisamos criar sempre seguindo as convenções.

Arquivos da Aplicação

Todos os arquivos da aplicação que criamos ficam na pasta src.

Criptografia

Por padrão o CakePHP 3.x usa a criptografia bcrypt para proteger as senhas.

Documentação oficial

<http://book.cakephp.org/3.0/pt/intro/conventions.html>