

DESENVOLVIMENTO PARA WEB II

Primeira Aplicação: Tutorial Blog



Professor: Alexandre Strapação Guedes Vianna – IFPE Igarassu
alexandrevianna.net

Roteiro da Aula

Relembrando...

Instalação da Infraestrutura

PHP 7 , MySQL 5.6, *Apache 2* e Composer

Instalação do CakePHP

- Instalação do intl extension e do mbstring
- Uso do composer para instalar o CakePHP
- Executando o CakePHP no servidor de desenvolvimento embutido

Roteiro da Aula

1. Terminar a instalação da infraestrutura
 - Executar o self-update como admin (sudo)
 - Colocar o servidor para rodar.
2. Construção da primeira aplicação
 - Configurar a senha de acesso ao banco
 - Criar um esquema no banco
 - Tutorial: construção da aplicação BookMark



Tutorial Blog

Tutorial de construção de um blog com o CakePHP

Este tutorial irá orientá-lo através da criação de um simples blog. Faremos a instalação do CakePHP, criaremos um banco de dados e implementaremos a lógica capaz de listar, adicionar, editar e apagar postagens do blog.

Este tutorial esta disponível no site do CakePHP em:

<http://book.cakephp.org/3.0/pt/tutorials-and-examples/blog/blog.html>

Tutorial Blog

1º Etapa: Criação de um novo projeto do CakePHP

Bom antes de tudo realizaremos a construção de um novo projeto do Cake utilizando o composer. (escolham uma pasta adequada para tal)

```
$ sudo composer self-update
```

```
$ composer create-project --prefer-dist  
cakephp/app cake_blog
```

Tutorial Blog

1º Etapa: Criação de um novo projeto do CakePHP

A instalação perguntará se deve configurar as permissões de acesso a pastas automaticamente, marque Y para Yes:

```
set Folder Permissions ? (Default to Y) [Y,n]? y
```

Tutorial Blog

2º Etapa: Configurando o Banco de Dados

Com o MySQL já instalado precisamos realizar algumas configurações da DataSource para que nossa aplicação consiga se conectar a base de dados.

Acessar o MySQL por linha de comando:

```
$ mysql -u root -p
```

Uma senha deve ser solicitada, se você deixou a senha em branco de um enter e prossiga

Tutorial Blog

2º Etapa: Configurando o Banco de Dados

Agora vamos criar um esquema:

```
mysql> CREATE DATABASE cake_blog_db;
```

Para criar tabelas em um esquema devemos primeiro selecioná-lo com o comando USE

```
mysql> USE cake_blog_db;
```

Use o comando SHOW para exibir os esquemas existentes, teu novo esquema deve estar na lista:

```
mysql> SHOW databases;
```


Tutorial Blog

3º Etapa: Criação do Banco de Dados

Vamos criar a tabela articles para armazenar os artigos do blog:

```
mysql> CREATE TABLE articles (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    title VARCHAR(50),  
    body TEXT,  
    created DATETIME DEFAULT NULL,  
    modified DATETIME DEFAULT NULL  
);
```



Atenção a padronização de nomes no Cake é importante, nomes de esquemas são no plural, verifique mais detalhes em <http://book.cakephp.org/3.0/pt/intro/conventions.html>

Tutorial Blog

3º Etapa: Criação do Banco de Dados

Utilize o comando SHOW e DESCRIBE para verificar a criação da sua tabela:

```
mysql> SHOW tables;
```

O DESCRIBE exibe mais detalhes:

```
mysql> DESCRIBE articles;
```

Tutorial Blog

3º Etapa: Criação do Banco de Dados

Vamos inserir diretamente no banco alguns artigos para o nosso blog :

```
mysql> INSERT INTO articles (title,body,created)
VALUES ('The title', 'This is the article body.', NOW());
INSERT INTO articles (title,body,created)
VALUES ('A title once again', 'And the article body follows.', NOW());
INSERT INTO articles (title,body,created)
VALUES ('Title strikes back', 'This is really exciting! Not.', NOW());
```

Tutorial Blog

3º Etapa: Criação do Banco de Dados

Utilize o comando SELECT para exibir todos os artigos cadastrados

```
mysql> SELECT * FROM articles;
```

Para sair do MySQL utilize o comando EXIT

```
mysql> EXIT;
```

Tutorial Blog

4º Etapa: Configurar a DataSource no seu projeto do CakePHP

Para que a sua aplicação possa se conectar a base dados ela precisa saber as credenciais de acesso e o nome do esquema. Estas configurações são realizadas no arquivo **config/app.php** na propriedade DataSource.

Configure: username,
password e database.

```
'Datasources' => [  
    'default' => [  
        'className' => 'Cake\Database\Connection',  
        'driver' => 'Cake\Database\Driver\Mysql',  
        'persistent' => false,  
        'host' => 'localhost',  
        //'port' => 'non standard port number',  
        'username' => 'root',  
        'password' => '',  
        'database' => 'cake_blog_db',  
        'encoding' => 'utf8',  
        'timezone' => 'UTC',  
        'flags' => [],  
        'cacheMetadata' => true,  
        'log' => false,
```

Tutorial Blog

5º Etapa: Rodar o servidor e verificar instalação

Bom, após configurar o ambiente vamos verificar se esta tudo funcionando. Entre dentro da pasta do projeto :

```
$ cd cake_blog
```

Execute o servidor:

```
$ bin/cake server
```

Acesse o endereço <http://localhost:8765/> no seu browser e verifique se todas as configurações estão funcionando

Environment

- ✓ Your version of PHP is 5.5.9 or higher (detected 7.0.6-1+donate.sury.org~xenial+1).
 - ✓ Your version of PHP has the mbstring extension loaded.
 - ✓ Your version of PHP has the openssl extension loaded.
 - ✓ Your version of PHP has the intl extension loaded.
-

Filesystem

- ✓ Your tmp directory is writable.
 - ✓ Your logs directory is writable.
 - ✓ The *FileEngine* is being used for core caching. To change the config edit config/app.php
-

Database

- ✓ CakePHP is able to connect to the database.
-

DebugKit

- ✓ DebugKit is loaded.

Tutorial Blog

Uso do Bake para gerar classes...

Bom, após criar o modelo no banco de dados nosso trabalho de criar as classes de modelo, controller e view será automatizado com a ferramenta bake. Esta ferramenta é executada em linha de comando e gera classes de modelo, controle e visão por inferência da base de dados.

Acesse o endereço:

<http://book.cakephp.org/3.0/en/bake/usage.html> e saiba mais sobre o bake.

Tutorial Blog

Classes do Modelo

Os arquivos de classes, correspondentes aos models, no CakePHP estão divididos entre os objetos Table e Entity. Objetos Table provêm acesso à coleção de entidades armazenada em uma tabela e são alocados em `src/Model/Table`.

Execute o comando a seguir para gerar o modelo com bake:

```
$ bin/cake bake model Articles
```



Convenções de nomenclatura são muito importantes no CakePHP. Ao nomear nosso objeto como `ArticlesTable`, o CakePHP automaticamente deduz que o mesmo utilize o `ArticlesController` e seja relacionado à tabela `articles`.

Tutorial Blog

Classes do Modelo

O bake gerou uma serie de arquivos, mas por enquanto não iremos entra em detalhes.

```
// src/Model/Table/ArticlesTable.php

namespace App\Model\Table;

use Cake\ORM\Table;

class ArticlesTable extends Table
{
    public function initialize(array $config)
    {
        $this->addBehavior('Timestamp');
    }
}
```

Tutorial Blog

Classes de Controle

A seguir, criaremos um controller (controlador) para nossos artigos. O controller é responsável pela lógica de interação da aplicação. É o lugar onde você utilizará as regras contidas nos models e executará tarefas relacionadas aos artigos. Criaremos um arquivo chamado ArticlesController.php no diretório src/Controller.

```
$ bin/cake bake controller Articles
```

Tutorial Blog

Classes de Controle

```
// src/Controller/ArticlesController.php

namespace App\Controller;

class ArticlesController extends ApplicationController
{
}
```

Tutorial Blog

Classes de Controle

Agora, vamos verificar o controller gerado, podemos notar que o cake adicionou diversas **actions** (ação) ao nosso controller. Actions frequentemente, representam uma função ou interface em uma aplicação, por exemplo: delete, index, view, add e edit são as mais comuns.

O cake gera estas actions automaticamente.

Quando os usuários requisitarem www.example.com/articles/index (sendo o mesmo que www.example.com/articles/), eles esperam ver uma lista de artigos.

```
// src/Controller/ArticlesController.php
namespace App\Controller;
class ArticlesController extends ApplicationController
{
    public function index()
    {
        $articles = $this->Articles->find('all');
        $this->set(compact('articles'));
    }
    public function view($id = null)
    {
        $article = $this->Articles->get($id, [
            'contain' => []
        ]);

        $this->set('article', $article);
        $this->set('_serialize', ['article']);
    }
}
```

Tutorial Blog

Criando a View

Agora que nós temos os dados fluindo pelo nosso model, e nossa lógica da aplicação definida em nosso controller, vamos criar uma view (visualização) para a action `index()`.

As views do CakePHP são camadas de apresentação que se encaixam nos layouts da aplicação. Para a maioria das aplicações, elas são uma mescla entre HTML e PHP.

Um layout é um conjunto de códigos encontrado ao redor das views. Múltiplos layouts podem ser definidos, e você pode alterar entre eles, mas agora, vamos usar o default, localizado em `src/Template/Layout/default.ctp`.

Tutorial Blog

Criando a View

Utilize o bake para gerar os templates:

```
$ bin/cake bake template Articles
```


Tutorial Blog

Criando a View

Utilize o bake para gerar os templates:

```
$ bin/cake bake template Articles
```

Acesse a pasta: src/Template/Articles e verifique que o bake gerou 4 arquivos: view.ctp, index.ctp, edit.ctp e add.ctp

Tutorial Blog

Criando a View

No controller atribuímos à última sessão a variável '**articles**' à view usando o método `set()`? Isso levará a coleção de objetos gerada pela query a ser invocada numa iteração `foreach`.

Arquivos de template do CakePHP são armazenados em `src/Template` dentro de uma pasta com o nome do controller correspondente (nós teremos que criar a pasta '`Articles`' nesse caso).

Comandos do MySQL

Comandos do MySQL

Acessar o MySQL por linha de comando:

```
$ mysql -u root -p
```

Listar esquemas cadastrados:

```
mysql> SHOW databases;
```

Criar um esquema:

```
mysql> CREATE DATABASE teste;
```

Antes de criar uma tabela ou realizar qualquer operação, é necessário selecionar o esquema que vai ser usado:

```
mysql> USE teste;
```

Instalação do CakePHP

Outros problemas na instalação:

Se estiver com problemas de permissões de leitura e escrita dos arquivos do projeto provavelmente você criou o seu projeto com o comando `sudo`, apague a pasta e refaça a criação do projeto sem o comando `sudo`. Outra possibilidade é setar as permissões com o comando **`chmod`**

```
$ sudo composer self-update
```

```
$ composer create-project --prefer-dist  
cakephp/app cake_blog
```