DESENVOLVIMENTO PARA WEB II
Atualizando o CakePHP, Associations

Professor: Alexandre Strapação Guedes Vianna – IFPE Igarassu

Agenda

- Como atualizar o CakePHP para um nova versão ?

- Associations
  - HasOne
  - HasMany
  - BelongsTo
  - BelongsToMany

Atualizando o CakePHP

- A atualização do CakePHP e componentes de terceiros é realizada no arquivo de configuração do composer:

```
composer.json
```

Atualizando o CakePHP

```
{
    "name": "cakephp/app",
    "description": "CakePHP skeleton app",
    "homepage": "http://cakephp.org",
    "type": "project",
    "license": "MIT",
    "require": {
        "php": ">=5.5.9",
        "cakephp/cakephp": "3.4.*",
        "mobiledetect/mobiledetectlib": "2.*",
        "cakephp/migrations": "~1.0",
        "cakephp/plugin-installer": "*"
    },
    "require-dev": {
        "psy/psysh": "@stable",
        "cakephp/debug_kit": "~3.2",
        "cakephp/bake": "~1.1"
    },
```

Atualizando o CakePHP

- Edite a configuração da versão do CakePHP
- Por exemplo se o seu cake esta na versão

  `cakephp/cakephp": "3.3.*`

  você pode colocar para a versão

  `cakephp/cakephp": "3.4.*`

- Depois disso execute o comando:

```
$ composer update
```

Ao baixar um projeto do Git

- A pasta vendor contém arquivos binários do cake de componentes de terceiros, esta pasta não deve ser inclusa no repositório. O composer já coloca a pasta vendor na ignore list no arquivo oculto .gitignore
- Ao baixar um projeto cake do repositório você deve gerar a pasta vendor com o composer

```
$ composer install
```

Associations

- Associations descrevem como as entidades do meu sistema se relacionam.
- Ao modelar um sistema você percerá que existem várias formas de relacionamento entre entidades
- Estes relacionamentos podem ser traduzidos das associações existentes no banco de dados (no caso dos bancos relacionais)
- Porém a camada de software pode abstrair de moldar estes relacionamento em um nível maior de abstração

Associations

- O cakePHP tem nomes específicos para as associações que equivalem aos relacionamentos construídos no banco de dados.

Relacionamento	Tipo de Associação	Exemplo
one to one	hasOne	Um usuário tem um endereço
one to many	hasMany	Uma disciplina pode ter várias turmas associadas a ela.
many to one	belongsTo	Várias turmas pertencem a um professor. Uma turma não pode pertencer a mais de um professor
many to many	belongsToMany	Turma tem vários alunos matriculados e alunos podem estar matriculados em diversar turmas. Necessário uma tabela intermediária.

Associations: hasOne

- Os relacionamentos devem ser expressos no arquivo table do modelo, dentro da função **initialize.**
- Por exemplo, um relacionamento 1 para 1, em que o usuário tem um endereço. A chave estrangeira fica no lado endereço, e o relacionamento **hasOne** no CakePHP do lado User.

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

Associations: hasMany

- Por exemplo, um relacionamento 1 para N, em que uma disciplina pode ter várias turmas associadas a ela. A chave estrangeira fica no lado das turmas, e o relacionamento **hasMany** no CakePHP do lado Disiciplinas.

```php
class DisciplinasTable extends Table
{
    public function initialize(array $config)
    {
        $this->hasMany('Turmas');
    }
}
```

Associations: belongsTo

- Este relacionamento é uma visão na direção inversa ao hasMany, e abstrair a associação de uma forma um pouco diferentes. Por exemplo, um relacionamento N para a, em que um professor pode estar associado (leciona) a várias turmas. A chave estrangeira fica no lado das Turmas, e o relacionamento **belongsTo** no CakePHP do lado de Turmas.

```php
class TurmasTable extends Table
{
    public function initialize(array $config)
    {
        $this->belongsTo('Professores');
    }
}
```

Associations: belongsToMany

- Por exemplo, um relacionamento N para N, em que um aluno pode associar-se(matricular) à várias turmas, e as turmas pode associar-se a vários alunos. Neste caso criamos uma tabela intermediária chamada turmas_alunos, há qual poderá ter a chave primária composta das duas chaves estrangeiras de turma e aluno. O relacionamento **belongsToMany** no CakePHP vai nos dois lados do relacionamento.

Associations: belongsToMany

```php
class TurmasTable extends Table
{
    public function initialize(array $config)
    {
        $this->belongsToMany('Alunos');
    }
}

class AlunosTable extends Table
{
    public function initialize(array $config)
    {
        $this->belongsTo('Turmas');
    }
}
```

Associations: belongsToMany

- Este relacionamento N para N, pode possuir muitas variações de acordo com o contexto.

- Por exemplo o relacionamento aluno <-> turma precisa incluir em qual semestre o aluno cursou a disciplina pois se o aluno reprovar ele precisa realizar outra matricula e gerar outra associação, o que causaria um inconsistência no sistema se a chave primária for a combinação de aluno e turma.

- A tabela intermediária também conter informações que são geradas do relacionamento: situaçõe do aluno, semestre, nota ...

Upload de Arquivo

- Não é desejável que o upload de arquivos seja realizado nos controllers, pois se você tem várias entidades salvando arquivos esta função precisará se replicada. Isto vai contra a prática de reuso de código.

- Neste caso o upload do arquivo deve ser realizado no model de Files, o qual qualquer entidade com um relacionamento pode acessar.

Upload de Arquivo

- FilesTable.php

```
public function uploadAndSaveFile($tmp_name, $path, $fileName){
        $uploadFile = $path.$fileName;
        if(move_uploaded_file($tmp_name,WWW_ROOT.$uploadFile)){
            $uploadData = $this->newEntity();
            $uploadData->name = $fileName;
            $uploadData->path = $path;
            $uploadData->created = date("Y-m-d H:i:s");
            $uploadData->modified = date("Y-m-d H:i:s");
            if ($this->save($uploadData)) {
                return $uploadData;
            }
        }
        return false;
    }
```

Upload de Arquivo

- Como pegar a extensão do arquivo.

```
$extension = pathinfo($this->request->data['arquivo']['name'],
PATHINFO_EXTENSION);
```