

サーバーサイドスクリプトⅡ 課題10 補足資料

今回の課題では、サムネイル用のテーブルを追加します。

以前の課題で行ったことを復習しつつ、新たなテーブル作成および処理の修正を行ってください。

1-1. 課題作成手順(概要のみ)

必要な手順は以下の通りです。

1. Migration ファイルの作成
 2. Model クラスの作成
 3. マイグレーションの実行
 4. 新規登録画面のデザイン(kadai08_1.blade)修正
 5. Article モデル編集
 6. 新規登録処理の修正
 7. 一覧画面のデザイン(kadai06_1.blade)の修正
- ここまでで新規登録が可能であり、一覧画面に画像が表示されることを確認すること！！
8. 詳細画面のデザイン(kadai07_1.blade)の修正
 9. 詳細表示処理の修正
 10. 更新画面のデザイン(kadai09_1.blade)の修正
 11. 更新処理の修正
 12. 削除処理の修正

1-2. 課題作成手順(ヒント)

サムネイルテーブルの仕様は下記の通りです。

サムネイルテーブル

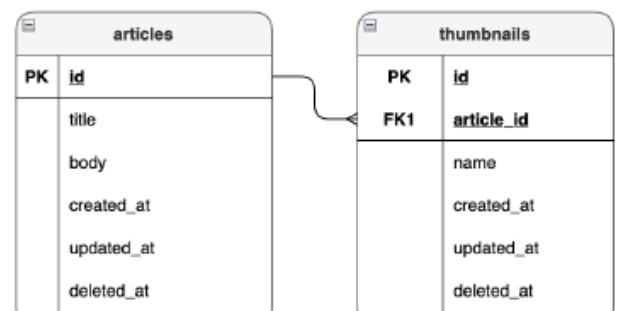
投稿記事のサムネイル画像の情報を保存するテーブル

テーブル名	thumbnails
-------	------------

サムネイルのテーブル情報

column	type	size	option	option
id	BIGINT		AUTO_INCREMENT	PRIMARY KEY
article_id	BIGINT		NO NULL	FOREIGN KEY
name	VARCHAR	255	NO NULL	
created_at	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP	
updated_at	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP
deleted_at	TIMESTAMP			

投稿記事とサムネイルは「1対多」の関係となります。



1. Migration ファイルの作成

①ファイル作成はターミナルからコマンドで作成する

■migration の作成コマンド **{}**の箇所は{}も含め、適宜置き換える。

```
php artisan make:migration create_{table_name}_table
```

②up メソッドと down メソッドに必要事項を記載する。

※今回、Thumbnails テーブルにはデモデータを挿入する必要はないので、Seeder クラスは不要です。

2. Model クラスの作成

①ファイル作成はターミナルからコマンドで作成する

■Model クラス作成コマンド **{}**の箇所は**{}**も含め、適宜置き換える。

```
php artisan make:model {モデル名}
```

原則、テーブル(複数形)の単数形で命名する。

②バリデーションチェック用のルールとエラーメッセージの定義を行う

3. マイグレーションの実行

■実行コマンド

```
php artisan migrate
```

4. 新規登録画面のデザイン(kadai08_1.blade)修正

画像ファイルのアップロードができるようにする。

5. Article モデル編集

Article モデルを呼び出すと同時に、サムネイル用のクラスと紐づけを行う。

■Article モデルに追加 **{}**の箇所は**{}**も含め、適宜置き換える。1対多のため、hasMany を使用。

```
public function thumbnails() {  
    return $this->hasMany({サムネイルクラス名}::class );  
}
```

6. 新規登録処理の修正

Kadai06_1Controller クラスで新規登録処理を担っている箇所を書き替える。

確認方法: 新規登録画面で登録ボタン押下時に、どのメソッドを呼ぶように指定しているかで確認。

- ① 画像ファイルの保存先はローカルの Storage とすること
- ② サムネイルテーブルに保存するのは①で保存したファイル名(元のファイル名ではなく、保存時にランダムに命名された名前)であること

7. 一覧画面のデザイン(kadai06_1.blade)の修正

画像を表示できるように追加する。

Article モデルでサムネイルと紐づけているため、Select を個別に追加する必要はない。

※但し、1対多なので、サムネイルは配列形式で取得となる。

ここまでで新規登録が可能であり、一覧画面に画像が表示されることを確認すること！！
登録が完了しなければ、以下を実装しても確認できない。(更新や削除なので)
また、ここまででエラーがないことを確認しなければ、この後の処理を実装して動かない場合、
どこが違っているのか特定することが困難となる。

8. 詳細画面のデザイン(kadai07_1.blade)の修正

条件分岐で画像があれば表示。

9. 詳細表示処理の修正

10. 更新画面のデザイン(kadai09_1.blade)の修正

11. 更新処理の修正

12. 削除処理の修正