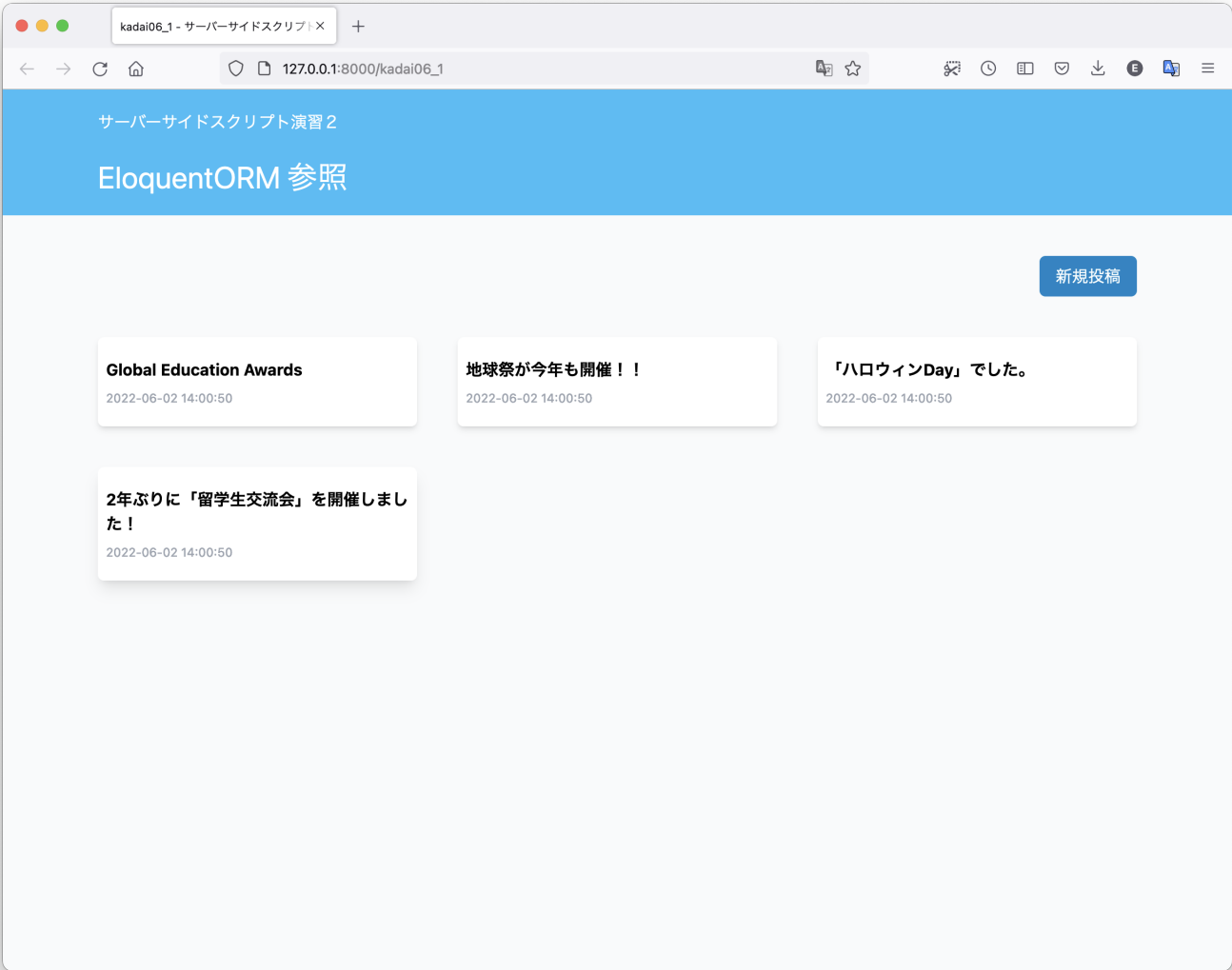


# kadai06\_1

LaravelのEloquentORMを使って、MySQLデータベースからレコードを抽出します。



## 作成するファイル

- Kadai06\_1Controller.php
- Article.php （モデル）
- {タイムスタンプ}\_create\_articles\_table.php (マイグレーション)
- ArticlesTableSeeder.php （シーダー）

## データベースの作成

データベース名	laravel8
---------	----------

データベースがない場合は、ターミナルでMySQLへ「CREATE DATABASE」命令で作成してください。

```
-- データベース：laravel8
CREATE DATABASE `laravel8` DEFAULT CHARACTER SET utf8mb4
```

## 投稿テーブルの作成

テーブル名	articles
-------	----------

### テーブル構造

column	type	SIZE	option	option
id	BIGINT		AUTO_INCREMENT	PRIMARY KEY
title	VARCHAR	255	NOT NULL	
body	TEXT			
created_at	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP	
updated_at	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP
deleted_at	TIMESTAMP			

## マイグレーションの作成

artisanコマンドを使って、投稿テーブルのマイグレーションファイルを作成しましょう。

```
php artisan make:migration create_articles_table
```

## upメソッドの処理

投稿テーブルに必要なカラムを定義してください。

```
/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('articles', function (Blueprint $table) {
        $table->id()->unsigned();
        $table->string( 'title', 255 );
        $table->text( 'body' )->nullable();
        $table->timestamp( 'created_at' )->useCurrent();
        $table->timestamp( 'updated_at' )->useCurrent()->useCurrentOnUpdate();
        $table->softDeletes();
    });
}
```

## Articleの作成

artisanコマンドで投稿テーブルのモデル「**Article**」を作成してください。

```
php artisan make:model Article
```

## Articleの設定

レコードの削除は、論理削除でおこなうため、「**SoftDeletes**」をuseで指定してください。

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Article extends Model
{
    use HasFactory;
    use SoftDeletes;           // 論理削除の場合

    protected $table = 'articles'; // テーブル名
}
```

## 投稿テーブル用シーダーの作成

投稿テーブルにテストデータを準備するためのシーダーファイルを作成してください。

```
php artisan make:seeder ArticlesTableSeeder
```

## runメソッドの処理

```
/**
 * Run the database seeds.
 *
 * @return void
 */
public function run()
{
    DB::transaction(function(){
        Article::insert([
            [
                'title' => 'Global Education Awards',
            ]
        ]);
    });
}
```

```
        'body' => 'ECCコンピュータ専門学校をはじめ、姉妹校であるECC国際外語専門学校、ECCアーティスト美容専門学校の学生が集結し、「山口学園の“Global Education”を通じて得た考えや自己成長などについて語る・伝える・共有する」をテーマに創  
    ],  
    [  
        'title' => '地球祭が今年も開催!!',  
        'body' => '12月17日に地球祭（学園祭）が行われました。毎年3校合同で開催される年末の一大行事ですが、感染対策のため昨年と今年は各校別々で行っています。'  
    ],  
    [  
        'title' => '「ハロウィンDay」でした。',  
        'body' => '10/29（金）恒例の「ハロウィン」が行われました。今年も学生会による校舎の飾りつけや仮装で在校生を楽しませてくれました。'  
    ],  
    [  
        'title' => '2年ぶりに「留学生交流会」を開催しました! ',  
        'body' => 'コロナの影響で中止していた「留学生交流会」を2年ぶりに実施しましたその様子をお伝えします。',  
    ],  
    ]  
});  
}
```

シーダーファイルで定義したテストデータを、artisanコマンドの「db:seed」を使って挿入してください。

## DatabaseSeederへの登録

準備した投稿テーブル用のシーダーをDatabaseSeederのrunメソッドへ追加してください。

```
/**  
 * Seed the application's database.  
 *  
 * @return void  
 */  
public function run()  
{  
    // \App\Models\User::factory(10)->create();  
    $this->call([  
        ArticlesTableSeeder::class,  
    ]);  
}
```

## シーダーの実行

```
php artisan db:seed
```

## Kadai06\_1Controllerの作成

artisanコマンドを使って、**Kadai06\_1Controller.php**を作成してください。

artisanコマンドでコントローラを作成する際に「--resource」を付けて、CRUD用のメソッドを自動的に定義された状態で作成してください。

## indexメソッドの処理

articlesテーブルからすべてのレコードを抽出してください。

viewヘルパーで、表示するページ「kadai06\_1.blade.php」を指定して、articlesテーブルから抽出したレコードデータを渡してください。

## ビューの作成

### kadai06\_1ビュー

投稿記事一覧を表示するページになります。

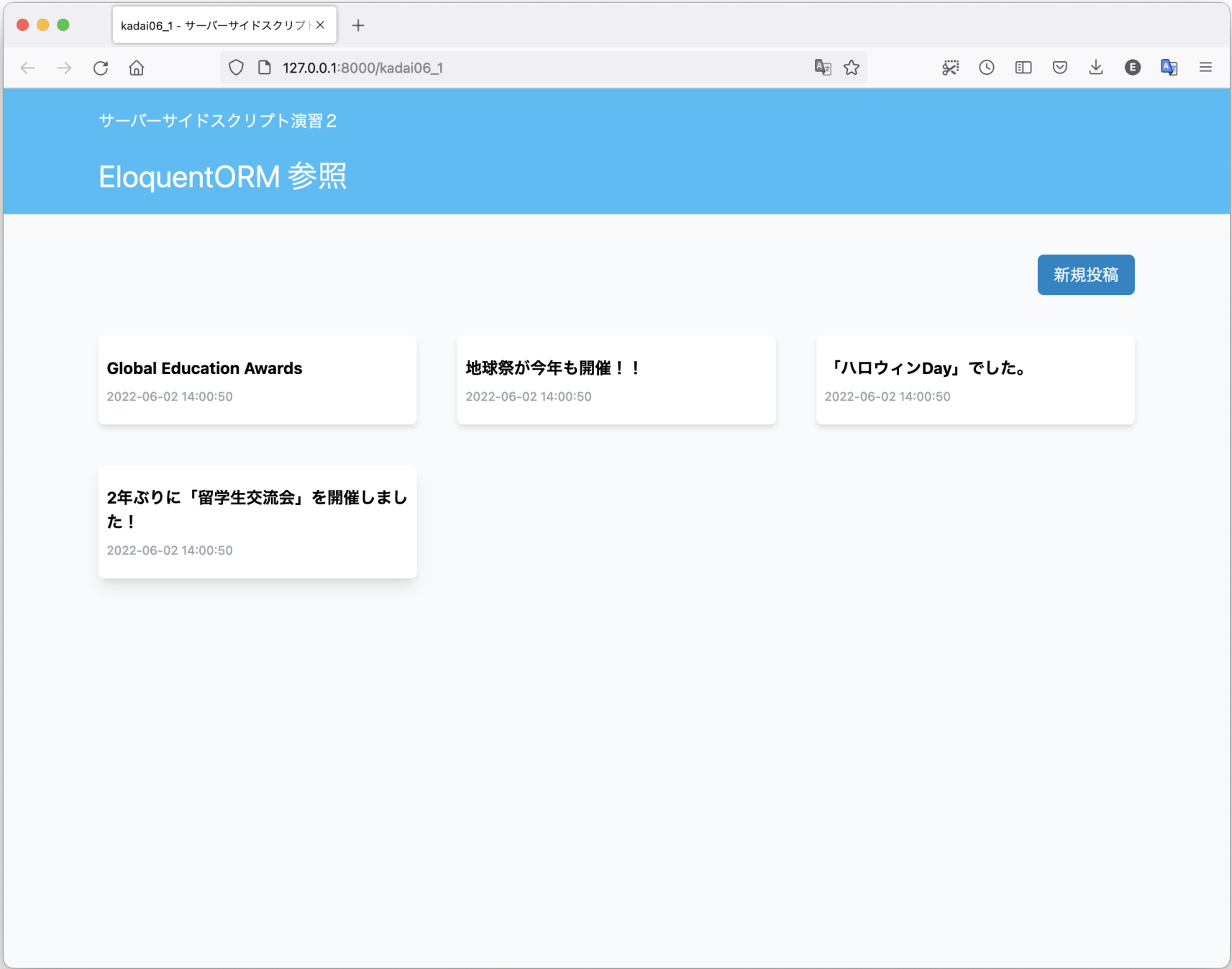
タイトル	kadai06_1
課題の見出し	EloquentORM 参照

投稿テーブルに登録されているすべての投稿記事を表示してください。

### コンテンツ部分のHTML構造

```
<div class="flex justify-end mb-10">  
    <a href="#" class="block w-24 text-white text-center bg-sky-600 hover:bg-sky-500 px-3 py-2 rounded-md">新規投稿</a>  
</div>  
  
<section class="grid grid-cols-1 lg:grid-cols-3 gap-10 mb-10">  
  
    <article class="row-span-2 bg-white hover:bg-white rounded-md shadow-md hover:shadow-lg transition-shadow overflow-hidden">  
        <a href="#" class="block w-full h-full">  
            <h3>投稿記事のタイトル</h3>  
            <p><time datetime="投稿記事の日時">投稿記事の日時</time></p>  
        </a>  
    </article>  
  
</section>
```

デザインはTailwind CSSを使って自由に変更してください。



## ルーティングの設定

「**Route::resource**」でkadai06\_1へ各メソッドごとのルーティング設定を一括で指定してください。

```
php artisan route:list --path=kadai06

+-----+-----+-----+-----+-----+-----+
| Domain | Method | URI | Name | Action | Middleware |
+-----+-----+-----+-----+-----+-----+
| | GET|HEAD | kadai06_1 | kadai06_1.index | App\Http\Controllers\kadai06_1Controller@index | web |
| | POST | kadai06_1 | kadai06_1.store | App\Http\Controllers\kadai06_1Controller@store | web |
| | GET|HEAD | kadai06_1/create | kadai06_1.create | App\Http\Controllers\kadai06_1Controller@create | web |
| | GET|HEAD | kadai06_1/{kadai06_1} | kadai06_1.show | App\Http\Controllers\kadai06_1Controller@show | web |
| | PUT|PATCH | kadai06_1/{kadai06_1} | kadai06_1.update | App\Http\Controllers\kadai06_1Controller@update | web |
| | DELETE | kadai06_1/{kadai06_1} | kadai06_1.destroy | App\Http\Controllers\kadai06_1Controller@destroy | web |
| | GET|HEAD | kadai06_1/{kadai06_1}/edit | kadai06_1.edit | App\Http\Controllers\kadai06_1Controller@edit | web |
+-----+-----+-----+-----+-----+-----+
```

## チェック

PHPサーバーを起動して、正常に表示されるかを確認しましょう。

[http://127.0.0.1:8000/kadai06\\_1](http://127.0.0.1:8000/kadai06_1)