

# データベースの処理 - テストデータとページネーション

## ファクトリー

データベース駆動型アプリケーションのテストを容易にするため、Laravelはさまざまな便利なツールを提供しています。

それに加えて、Laravelのファクトリー（テスト用インスタンスの生成）とシーダー（初期データ設定）により、Eloquentモデルとリレーションを使用し、テストデータのレコードを簡単に作成できます。

テストを実行する前に、データベースにいくつかのレコードを挿入する必要があります。このテストデータを作成するときに各カラムの値を自分でいちいち指定する代わりに、Laravelではファクトリーを使用して、デフォルト属性を定義できます。

### ファクトリーの作成と設定

ファクトリーの作成は、artisanコマンドの「**make:factory**」を使っておこないます。

ファクトリーファイルは、「**database/factories**」の中に作成されます。

```
php artisan make:factory ArticleFactory
```

```
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

class ArticleFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array
     */
    public function definition()
    {
        return [
            //
        ];
    }
}
```

作成したファクトリークラスに対応したモデルを「**\$model**」プロパティで定義します。

```
use App\Models\Article;

class ArticleFactory extends Factory
{
    protected $model = Article::class;
    <<<<< 略 >>>>>
}
```

作成したファクトリークラスには、「**definition**」メソッドが用意されます。

definitionメソッドは、ファクトリーを使用してモデルを作成するときに適用する必要がある属性値のデフォルトセットを返します。

```
class ArticleFactory extends Factory
{
    protected $model = Article::class;

    /**
     * Define the model's default state.
     *
     * @return array
     */
    public function definition()
    {
        return [
            "title" => "サンプルのタイトル",
            "body"  => "サンプルの投稿内容。",
        ];
    }
}
```

## シーダーでファクトリーの利用

作成したファクトリーをシーダーで利用することで、シーディング時にファクトリーで定義したテストデータを使用することができます。

```
class ArticlesTableSeeder extends Seeder
{
    /**
```

```

* 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：
        //     ],
        //     [
        //         '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、ECC国際外語専門学校・ECCアーティスト美容専門学校と共に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）ECCコンピュータ専門学校では、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ます。',
        //     ]
        // ]);

        // モデルファクトリーを利用したテストデータの挿入（100件分）
        Article::factory()->count( 100 )->create();
    });
}

```

## Fakerライブラリの活用

Fakerはテスト用のダミーデータを生成するPHPライブラリになります。  
Laravelには、テストデータの生成用にFakerライブラリが最初から組み込まれています。

FakerPHP / Faker

🔍 Search

FakerPHP/Faker  
v1.19.0 ⭐ 2.4k 🗨️ 183

FakerPHP / Faker

Faker

Available Formatters

Formatters >

Locales >

ORM Integration

Third-Party Libraries

downloads 4M/month build passing

Faker is a PHP library that generates fake data for you. Whether you need to bootstrap your database, create good-looking XML documents, fill-in your persistence to stress test it, or anonymize data taken from a production service, Faker is for you.

It's heavily inspired by Perl's Data::Faker, and by Ruby's Faker.

Installation

Faker requires PHP >= 7.1.

Table of contents

Installation

Basic Usage

Autoloading

Create fake data

Modifiers

Localization

Seeding the Generator

Faker Internals: Understanding Providers

Language specific formatters

Misnamed locales

License

FakerPHP / Faker

Faker is a PHP library that generates fake data for you. Whether you need to bootstrap your database, create good-looking XML documents, fill-in your persistence to stress test it, or anonymize data taken from a production service, Faker is for you. It's heavily inspired by Perl's Data::Faker, and by Ruby's Faker.

<https://fakerphp.github.io/>

FAKERPHP

## configの編集

config/app.php 内で設定されている「**faker\_locale**」オプションで、アプリケーションのFakerロケールを変更することができます。デフォルトでは、「en\_US」になっているため「**ja\_JP**」へ修正して、ダミーデータを日本語で生成されるようにしましょう。

```
/*
|-----|
| Faker Locale
|-----|
|
| This locale will be used by the Faker PHP library when generating fake
| data for your database seeds. For example, this will be used to get
| localized telephone numbers, street address information and more.
|
*/

// 'faker_locale' => 'en_US',
'faker_locale' => 'ja_JP',
```

## ファクトリーのテストデータをフェイカーで生成する

ファクトリークラス内のdefinitionメソッドで定義しているテストデータをフェイカーのダミーデータを使用するように修正しましょう。

```
public function definition()
{
    return [
        "title" => $this->faker->realText( 20 ),
        "body"  => $this->faker->realText( 200 ),
        "created_at" => $this->faker->dateTime( "now" ),
    ];
}
```

Laravel 8ドキュメント - テストデータ  
<https://readouble.com/laravel/8.x/ja/database-testing.html>

## ページネーションの実装

Webアプリケーションにペジネーションを実装するのは、非常に苦勞することがあります。  
Laravelは、デフォルトのページネーションを利用する場合は、簡単に実装できるようになっています。  
デフォルトでは、ペジネーターによって生成されたHTMLはTailwind CSSフレームワークと互換性があります。  
※ Bootstrapペジネーションのサポートも利用できます。

## コントローラーの修正

```
// $articles = Article::get();
Article::paginate( 10 );
```

Laravel 8ドキュメント - ページネーション  
<https://readouble.com/laravel/8.x/ja/pagination.html>

## 課題

- [kadai06\\_2](#)