



5. コントローラー

artisanでファイル作成

`artisan` の `make` オプションでファイル作成をします。作成できるファイルタイプはあらかじめ決まっています。

```
% php artisan make:タイプ ファイル名
```

Controller の作成

Controller ファイルの作成は、`make:controller` オプションを指定します。

```
% php artisan make:controller コントローラ名
```

作成場所

コントローラファイルは、`app/Http/Controllers/` をベースに作成されます。フォルダを作成するときは、コントローラ名の前に `/` (スラッシュ) でフォルダパスを指定します。

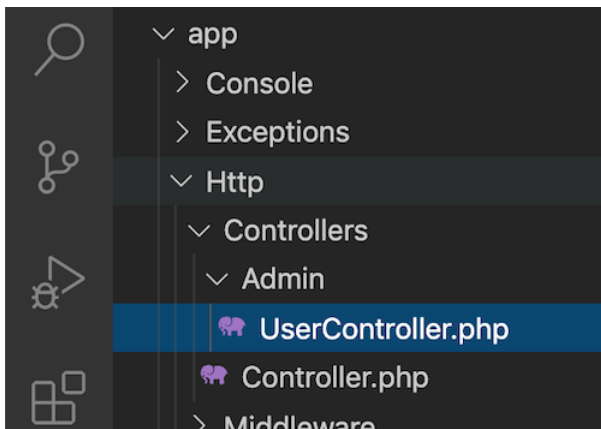
```
% php artisan make:controller フォルダ名/コントローラ名
```

コントローラ作成

`Admin/` に `UserController.php` を作成してみましょう。

```
% php artisan make:controller Admin/UserController
```

`app/Http/Controllers/Admin/` に `UserController.php` が作成されました。



コントローラファイルの確認

作成したコントローラファイルには、最低限のコードが記載されています。

```
<?php
namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class UserController extends Controller
{
    //
}
```

ネームスペース（namespace）とは？

PHPでは同じファイルや関数名が存在しても衝突しないように動作させる、**ネームスペース（namespace）** という仕組みがあります。ネームスペースは、**namespace** で定義して、**use** で有効にします。

ネームスペースの定義

```
namespace ネームスペース;
```

ネームスペースの利用

```
use ネームスペース;
```

ネームスペースのパス

ネームスペースのパスは、`\` (バックスラッシュ) で区切るのが特徴です。

`Admin\UserController.php` は作成時に、ネームスペースが自動定義されています。

```
namespace App\Http\Controllers\Admin;
```

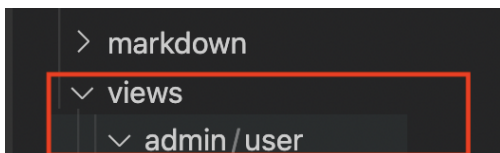
Action と View の追加

コントローラーに `index` アクションを追加し、`index.blade.php` をレンダリングするようにします。

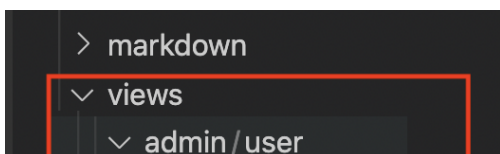
```
class UserController extends Controller
{
    //Action: index
    public function index()
    {
        //View: admin/user/index.blade.php
        return view('admin.user.index');
    }
}
```

Viewファイル作成

`resources/views/` に `admin` フォルダを作成します。



`resources/views/admin/` に `user` フォルダを作成します。



`resources/views/admin/user/` に `index.blade.php` ファイルを作成します。

```
views
  admin/user
    index.blade.php
```

`index.blade.php` に HTML を記述します。

```
<h2>ユーザー一覧</h2>
```

コントローラーとルーティング

静的ルーティングと動的ルーティング

Laravel では、コントローラーにアクションを追加しただけでは、URL アクセスできません。ルーティングファイルに手動で設定が必要で、これを「静的ルーティング」といいます。他のフレームワークでは、コントローラーとアクションを自動認識する「動的ルーティング」の機能もあります。

コントローラとアクション

Route ファサードで、URI に対するコントローラとアクションをルーティングします。`get()` の第 2 引数は Laravel 8 から、Class と Action の配列になりました。

```
Route::get(URI, [クラス名::class, アクション]);
```

ルーティングの追加

コントローラーのルーティングを追加してみましょう。`web.php` に `Admin\UserController.php` を利用できるよう、`use` で指定します。

```
use App\Http\Controllers\Admin\UserController;
```

`admin/user` に GET リクエストでルーティングします。

```
Route::get('admin/user/', [UserController::class, 'index']);
```

当サイトの教材をはじめとするコンテンツ（テキスト、画像等）の無断転載・無断使用を固く禁じます。これらのコンテンツについて権利者の許可なく複製、転用等する事は法律で禁止されています。尚、当ウェブサイトの内容をWeb、雑誌、書籍等へ転載、掲載する場合は「ロジコヤ」までご連絡ください。