

# APIの作成

LaravelでAPIを作成する

## APIリソースルートの定義

APIに使用するリソースルートを宣言する場合、**create**や**edit**のようなHTMLテンプレートを提供するルートを除外したいことがよく起こります。  
そのため、これらの 2 ルートを自動的に除外する、**apiResource**メソッドが使用できます。

```
use App\Http\Controllers\Kadai11_1Controller;

Route::apiResource( 'v1/articles', Kadai11_1Controller::class );
```

**apiResources**メソッドに配列として渡すことで、一度に複数のAPIリソースコントローラを登録することもできます。

```
// Route::apiResource( 'v1/articles', Kadai11_1Controller::class, );
Route::apiResources(
    [
        'v1/articles' => Kadai11_1Controller::class,
    ]
);
```

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

| Domain | Method    | URI                       | Name             | Action   | Middleware |
|--------|-----------|---------------------------|------------------|--|------------|
|        | GET HEAD  | api/v1/articles           | articles.index   | App\Http\Controllers\Kadai11_1Controller@index   | api        |
|        | POST      | api/v1/articles           | articles.store   | App\Http\Controllers\Kadai11_1Controller@store   | api        |
|        | GET HEAD  | api/v1/articles/{article} | articles.show    | App\Http\Controllers\Kadai11_1Controller@show    | api        |
|        | PUT PATCH | api/v1/articles/{article} | articles.update  | App\Http\Controllers\Kadai11_1Controller@update  | api        |
|        | DELETE    | api/v1/articles/{article} | articles.destroy | App\Http\Controllers\Kadai11_1Controller@destroy | api        |

## APIリソースコントローラーの作成

**createメソッド**や**editメソッド**を含まないAPIリソースコントローラーを素早く作成するには、artisanコマンドの「**make:controller**」を実行する際に、  
**--api**を使用します。

```
php artisan make:controller { コントローラー名 } --api
```

```
php artisan make:controller Kadai11_1Controller --api
```

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class Kadai11_1Controller extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        //
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        //
    }
}
```

## APIリソースの作成

php artisan make:resource { リソース名 }

```
php artisan make:resource ArticleResource
```

app/Http/Resources内にAPIリソースファイルが作成されます。

```
<?php

namespace App\Http\Resources;

use Illuminate\Http\Resources\Json\JsonResource;

class ArticleResource extends JsonResource
{
    /**
     * Transform the resource into an array.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return array|\Illuminate\Contracts\Support\Arrayable|\JsonSerializable
     */
    public function toArray($request)
    {
        return parent::toArray($request);
    }
}
```

toArrayメソッドの中に、APIリソースのデータ構造を定義します。

既存のreturnは、コメントアウトしておきます。

```
<<<<< 省略 >>>>>
public function toArray($request)
{
    // return parent::toArray($request);
    return [
        'id'          => $this->id,
        'title'       => $this->title,
        'body'        => $this->body,
        'thumbnails' => null,
        'created_at' => $this->created_at,
        'updated_at' => $this->updated_at,
    ];
}
```

## コントローラーで、リソースクラスを使ったJSONデータを指定する

作成したResourceクラスを使ったJSONデータを出力するようにします。

```
public function show($id)
{
    $article = Article::find( $id );
    return new ArticleResource( $article );
}
```

データが複数件の場合は、collectionメソッドを利用する

```
public function index()
{
    $articles = Article::orderBy( "created_at", "desc" )->paginate( 10 );
    return ArticleResource::collection( $articles );
}
```

サムネイルのリソースクラスを作成して、投稿記事のAPIリソースにサムネイルを表示できるようにサムネイルのパスを含むようにしましょう。

```
php artisan make:resource ThumbnailResource
```