

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula:

1) Na pasta `app/Http/Controllers`, crie a classe `BaseController`, para centralizar o código em comum entre os *controllers*. A classe terá o seguinte conteúdo:

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

use Illuminate\Http\Request;

abstract class BaseController
{
    protected $classe;

    public function index()
    {
        return $this->classe::all();
    }

    public function store(Request $request)
    {
        return response()
            ->json(
                $this->classe::create($request->all()),
                201
            );
    }

    public function show(int $id)
    {
        $recurso = $this->classe::find($id);
        if (is_null($recurso)) {
            return response()->json('', 204);
        }

        return response()->json($recurso);
    }

    public function update(int $id, Request $request)
    {
        $recurso = $this->classe::find($id);
        if (is_null($recurso)) {
            return response()->json([
                'erro' => 'Recurso não encontrado'
            ], 404);
        }
        $recurso->fill($request->all());
        $recurso->save();

        return $recurso;
    }
}
```

```

public function destroy(int $id)
{
    $qtdRecursosRemovidos = $this->classe::destroy($id);
    if ($qtdRecursosRemovidos === 0) {
        return response()->json([
            'erro' => 'Recurso não encontrado'
        ], 404);
    }

    return response()->json('', 204);
}
}

```

2) Continuando na pasta `app/Http/Controllers`, crie a nova classe `EpisodiosController`, que estende o `BaseController`:

```

<?php
namespace App\Http\Controllers;

use App\Episodio;

class EpisodiosController extends BaseController
{
    public function __construct()
    {
        $this->classe = Episodio::class;
    }
}

```

3) Ainda na pasta `app/Http/Controllers`, na classe `SeriesController`, estenda o `BaseController`, remova os métodos e adicione o construtor:

```

<?php
namespace App\Http\Controllers;

use App\Serie;

class SeriesController extends BaseController
{
    public function __construct()
    {
        $this->classe = Serie::class;
    }
}

```

4) Abra a classe `Episodio`, que está na pasta `app`. Adicione o método *acessor*:

```

public function getAssistidoAttribute($assistido): bool
{
    return $assistido;
}

```

5) Abra o arquivo `routes/web.php` . Crie um novo sub-grupo para o recurso `series` :

```
$router->group(['prefix' => 'api'], function () use ($router) {  
    $router->group(['prefix' => 'series'], function () use ($router) {  
        $router->post('', 'SeriesController@store');  
        $router->get('', 'SeriesController@index');  
        $router->get('{id}', 'SeriesController@show');  
        $router->put('{id}', 'SeriesController@update');  
        $router->delete('{id}', 'SeriesController@destroy');  
    });  
  
    //aqui vem as rotas do episodios  
});
```

Além disso, adicione, dentro do grupo "api", as rotas relacionadas com o recurso `episodios` :

```
$router->group(['prefix' => 'episodios'], function () use ($router) {  
    $router->post('', 'EpisodiosController@store');  
    $router->get('', 'EpisodiosController@index');  
    $router->get('{id}', 'EpisodiosController@show');  
    $router->put('{id}', 'EpisodiosController@update');  
    $router->delete('{id}', 'EpisodiosController@destroy');  
});
```

6) Na linha de comando, dentro do projeto, rode o servidor:

```
php -S localhost:8000 -t public
```

7) Teste se as rotas das séries continuam funcionando. Teste também o CRUD do recurso `/episodios` .

Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.