

## Consolidando o seu conhecimento

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

1) No PhpStorm, abra a classe `BaseController` e altere o método `index` para usar paginação:

```
public function index(Request $request)
{
    return $this->classe::paginate($request->per_page);
}
```

2) Com o servidor rodando, acesse, por exemplo, `/api/series?page1=per_page=2`. Repare que dados da paginação estão aparecendo.

Observação: Teste também o atributo `$perPage` na classe `Serie`.

3) No arquivo `routes/web.php`, adicione uma nova rota dentro do grupo com o prefixo `series`:

```
$router->get('{serieId}/episodios', 'EpisodiosController@buscaPorSerie');
```

4) Abra a classe `EpisodiosController` e adicione um novo método `buscaPorSerie`:

```
public function buscaPorSerie(int $serieId)
{
    $episodios = Episodio::query()
        ->where('serie_id', $serieId)
        ->paginate();

    return $episodios;
}
```

Com o servidor rodando, insira alguns episódios de uma série. Depois, teste o relacionamento, usando a URL `/api/series/1/episodios`.

5) Para adicionar as informações sobre o relacionamento nos recursos, abra a classe `Serie`, adicione um novo atributo `links` e um novo método `getLinksAttribute`.

Segue a classe `Serie` completa:

```
<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class Serie extends Model
{
    public $timestamps = false;
    protected $fillable = ['nome'];
}
```

```

        protected $appends = ['links'];

        public function episodios()
        {
            return $this->hasMany(Episodio::class);
        }

        public function getLinksAttribute($links): array
        {
            return [
                'self' => '/api/series/' . $this->id,
                'episodios' => '/api/series/' . $this->id . '/episodios'
            ];
        }
    }
}

```

Igualmente, devemos alterar a classe `Episodio`:

```

<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class Episodio extends Model
{
    public $timestamps = false;
    protected $fillable = ['temporada', 'numero', 'assistido', 'serie_id'];
    protected $appends = ['links'];

    public function serie()
    {
        return $this->belongsTo(Serie::class);
    }

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

    public function getLinksAttribute($links): array
    {
        return [
            'self' => '/api/episodios/' . $this->id,
            'serie' => '/api/series/' . $this->serie_id
        ];
    }
}

```

6) Acesse todas as séries e todos os episódios para ver os links gerados no JSON.

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.

