

09

Consolidando o seu conhecimento

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

- 1) Na linha de comando, use o `composer` para criar o projeto **Lumen**, na versão **5.8.0**:

```
composer create-project --prefer-dist laravel/lumen=5.8.0 api-series
```

- 2) Ainda na linha de comando, entre na pasta do projeto `api-series` e gere duas *migrations*:

```
php artisan make:migration criar_tabela_series --create-series
```

E:

```
php artisan make:migration criar_tabela_episodios --create-episodios
```

- 3) Importe o projeto no PhpStorm. Após a importação, abra a classe da *migration* `criar_tabela_series` e altere o método `up` para ficar da seguinte forma:

```
public function up()
{
    Schema::create('series', function (Blueprint $table) {
        $table->tinyIncrements('id');
        $table->string('nome')->unique();
    });
}
```

Altere também o método `up` da *migration* para criar a tabela `episodios`:

```
public function up()
{
    Schema::create('episodios', function (Blueprint $table) {
        $table->tinyIncrements('id');
        $table->integer('temporada');
        $table->integer('numero');
        $table->boolean('assistido')->default(false);
        $table->integer('serie_id');

        $table->foreign('serie_id')
            ->references('series')
            ->on('id');
    });
}
```

- 4) Vamos preparar as configurações referentes ao comando de persistência. Para tal:

- Crie, na raiz da pasta `database`, o arquivo vazio, com o nome `database.sqlite`
- Abra o arquivo `.env`, que está na raiz do projeto, e altera a linha com a configuração `DB_CONNECTION` para:

```
DB_CONNECTION=sqlite
```

- Abra o arquivo `bootstrap/app.php` e descomente a linha `$app->withEloquent();`

5) Agora, novamente na linha de comando e dentro do projeto, execute as *migrations*:

```
php artisan migrate
```

6) De volta ao PhpStorm, crie duas classes na pasta `app` para representar o nosso modelo, `Serie` e `Episodio`:

```
<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

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

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

E a classe `Episodio`:

```
<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

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

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

7) Abra o arquivo `routes/web.php` para definir a primeira rota, usando um prefixo:

```
$router->group(['prefix' => '/api'], function () use ($router) {
    $router->get('/config', 'SerieController@index');
```

```
    public function index()
    {
        return response()->json([
            'series' => $series,
            'seriesController@index' => []
        ]);
    }
}
```

8) Na pasta `app\Http\Controllers` , crie uma nova classe `SeriesController` , para atender a rota `/series` :

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

use App\Serie;
use Illuminate\Http\Request;

class SeriesController
{
    public function index()
    {
        return Serie::all();
    }
}
```

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

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

10) Use o **Postman** para executar a URL:

```
http://localhost:8000/api/series
```

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.