Laravel REST API With Model Relationships

Software as a Service - Back-End Development
Session 03

Developed by Adrian Gould

Contents

Region

Let's make sure that Region is correct.

Tests: Region Unit Test for the Model

Create a test for the `Region` model (`tests/Unit/RegionTest.php`):

```
use App\Models\Region;
use Illuminate\Foundation\Testing\RefreshDatabase;
```

```
uses(RefreshDatabase::class);

it('can create a region', function () {
    $region = Region::factory()->create();

    expect($region)->toBeInstanceOf(Region::class);
    expect($region->name)->not()->toBeNull();
});
```

Tests: Region Feature Test

Create a test for the `RegionController` (`tests/Feature/RegionControllerTest.php`):

```
use App\Models\Region;
use Illuminate\Foundation\Testing\RefreshDatabase;

uses(RefreshDatabase::class);
```

Fetch all regions test

```
it('can fetch a list of regions', function () {
    $regions = Region::factory(3)->create();

    $response = $this->getJson('/api/regions');

    $response->assertStatus(200)
        ->assertJsonStructure([
            'status',
            'message',
            'data' => [
                '*' => [
                    'id',
                    'name',
                    'subregions',
                    'countries',
                ],
            ],
        ])
        ->assertJsonCount(3, 'data');
});
```

Fetch single region test

```
it('can fetch a single region', function () {
    $region = Region::factory()->create();

    $response = $this->getJson("/api/regions/{$region->id}");

    $response->assertStatus(200)
        ->assertJsonStructure([
            'status',
            'message',
            'data' => [
                'id',
                'name',
                'subregions',
                'countries',
            ],
        ]);
});
```

Invalid Region Requested test

```
it('returns an error for a non-existent region', function () {
    $response = $this->getJson('/api/regions/99999');

    $response->assertStatus(404)
        ->assertJsonStructure([
            'status',
            'message',
        ]);
});
```

Run the Tests

Run the tests using PEST:

```
./vendor/bin/pest
```

> *Note:*
> *Remember that you could create an alias for this in your `.aliases` file.*
> *Check the Help Desk FAQs for examples.*

Migration

The Region Migration's `up` method should read:

```
public function up()
{
    Schema::create('regions', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->timestamps();
    });
}
```

Model

The Region Model should read:

```
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Region extends Model
{
    use HasFactory;

    protected $fillable = ['name'];

    public function countries()
    {
        # Comment out relationship until Subregion model is created
        # return $this->hasManyThrough(Country::class, Subregion::class);
    }

    public function subregions()
    {
        # Comment out relationship until Subregion model is created
        # return $this->hasMany(Subregion::class);
    }
}
```

Factory

Create a factory for the `Region` model:

```
php artisan make:factory RegionFactory --model=Region
```

Update the factory (`database/factories/RegionFactory.php`):

```
namespace Database\Factories;

use App\Models\Region;
use Illuminate\Database\Eloquent\Factories\Factory;

class RegionFactory extends Factory
{
    protected $model = Region::class;

    public function definition()
    {
        return [
            'name' => $this->faker->word,
        ];
    }
}
```

Run the Tests

Run the tests using PEST:

```
./vendor/bin/pest
```

Routes

Ensure the region routes are correct.

Controller

If you do not have a Region controller then we need to create one...

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

Check that your current country controller looks similar to the update shown below `RegionController` (`app/Http/Controllers/RegionController.php`):

```
namespace App\Http\Controllers;

use App\Models\Region;
use Illuminate\Http\Request;
```

```
class RegionController extends Controller
{
    public function index()
    {
        return Region::all();
        # We will add the subregions and countries later
        # return Region::with('subregions', 'countries')->get();
    }

    public function show($id)
    {
        return Region::findOrFail($id);
        # We will include the subregions and countries later
        # return Region::with('subregions', 'countries')-
>findOrFail($id);
    }
}
```

Run the Tests

Run the tests using PEST:

```
./vendor/bin/pest
```

Exercises

The S03-Exercises-and-Journal-Entry contains further development exercises for you to complete.