

Test task report

23 липня 2022 р. 21:00

Introduction.

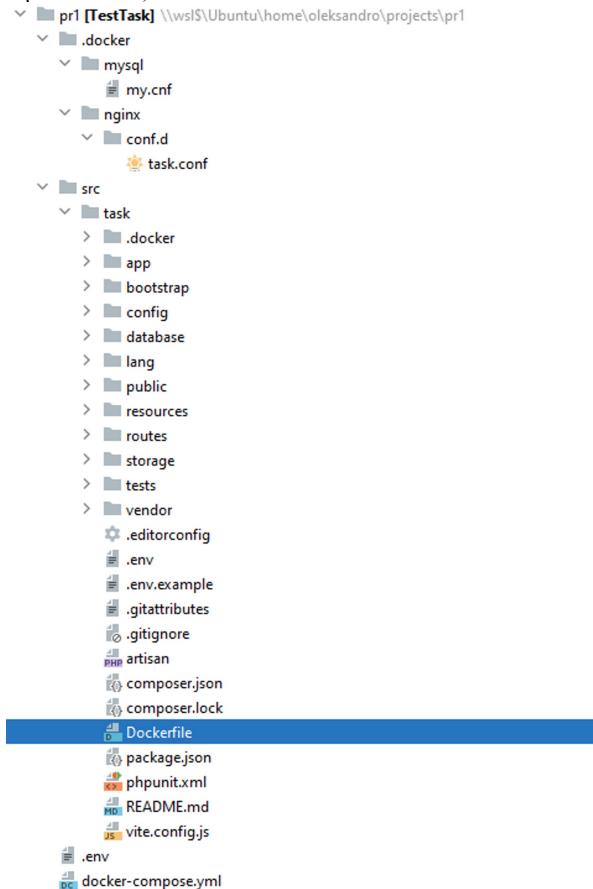
So that it would be clearer how I did the task, I decided to create a report, which clearly and understandably describe each of my steps, so that the course of my thoughts was as clear as possible.

1. Knowledge preparation

I haven't really gotten into it yet, but I figured out that I need to use Docker and Laravel. I decided to start development based on the Laravel framework because I had heard of it before and knew that it could be used to write an API. I knew nothing about it before, so I had to learn it from scratch. It took me about a week.

1. Environment preparation

I did not quite understand the point of the task, whether it was necessary to use the Docker. But since I already learned how to work with it, so I used it. I created containers for each service and saved their configuration files in the folder .docker . I put the project files in the service Task and saved them in a separate folder, where Dockerfile is also located. Here is how the structure of my project looks like



2. Once everything necessary for the work was prepared, I began to study the terms of reference. To make it easier, I imagined what the routes should look like and what follows them. It may not have been very clear, but it made it clearer to me what to do and I proceeded to the tasks

[] return {"hello": "MacPaw Internship 2022!"}

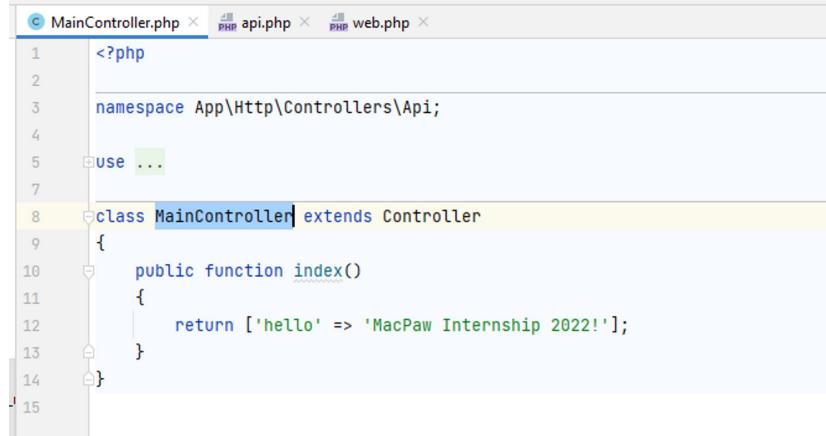
[] /neo/hazardous
SELECT * FROM `NearEarthObject` WHERE `hazardous` = 1

[] /neo/fastest?hazardous=(true|false)
SELECT * FROM `NearEarthObject` WHERE `hazardous` = {1|0} ORDER BY `speed` DESC

3. Task completion

3.1. Specify a default controller

The easiest task that took me 5 minutes of time. We create a controller by command `docker compose run --rm task php artisan make:controller Api/MainController`, and since it is the main one, we call it MainController, and in it we define the method where and return the required response



```
<?php

namespace App\Http\Controllers\Api;

use ...

class MainController extends Controller
{
    public function index()
    {
        return ['hello' => 'MacPaw Internship 2022!'];
    }
}
```



```
<?php

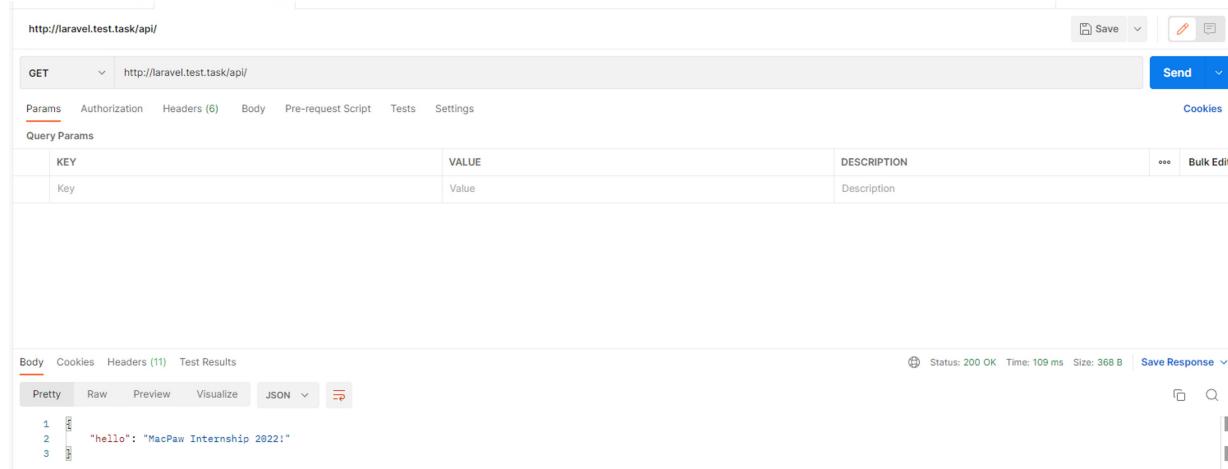
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});

Route::get('/', [\App\Http\Controllers\Api\MainController::class, 'index']);
```

Check and see that everything is working



http://laravel.test.task/api/

GET http://laravel.test.task/api/

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
1:   "hello": "MacPaw Internship 2022!"
```

Status: 200 OK Time: 109 ms Size: 368 B Save Response

3.2. Create Table in DB NearEarthObject

It's also a fairly easy task. For this, I created a model, migration and a factory at the same time using the command `docker compose run --rm task php artisan make:model doNearEarthObject -mf`

And then I defined the necessary columns. There were no instructions about field attributes in the task, so I left them by default

But here I ran into a problem that in the task it says that the columns should be called `createdAt` and `updatedAt`, but by default in Laravel they are called `created_at` and `updated_at`. So I decided to change the name of the corresponding constants. And I also had to rename the columns to make it work properly

```
PHP 2022_07_24_071230_create_near_earth_objects_table.php ×
1 <?php
2
3     use Illuminate\Database\Migrations\Migration;
4     use Illuminate\Database\Schema\Blueprint;
5     use Illuminate\Support\Facades\Schema;
6
7     return new class extends Migration
8     {
9         /**
10          * Run the migrations.
11          *
12          * @return void
13          */
14         public function up()
15         {
16             Schema::create('NearEarthObject', function (Blueprint $table) {
17                 $table->unsignedBigInteger('referenced');
18                 $table->string('name');
19                 $table->unsignedDouble('speed');
20                 $table->boolean('is_hazardous');
21                 $table->date('Date');
22                 $table->timestamps();
23
24                 $table->primary('referenced');
25                 $table->index('referenced');
26             });
27         }
28
29         /**
30          * Reverse the migrations.
31          *
32          * @return void
33          */
34         public function down()
35         {
36             Schema::dropIfExists('NearEarthObject');
37         }
38     };
39
```

Database > Eloquent > Model.php > CREATED_AT

```
Model.php x
175     protected static $lazyLoadingViolationTracker;
176
177     /**
178      * Indicates if broadcasting is currently enabled.
179      *
180      * @var bool
181      */
182     protected static $isBroadcasting = true;
183
184     /**
185      * The name of the "created at" column.
186      *
187      * @var string|null
188      */
189     const CREATED_AT = 'createdAt';
190
191     /**
192      * The name of the "updated at" column.
193      *
194      * @var string|null
195      */
196     const UPDATED_AT = 'updatedAt';
197
198     /**
199      * Create a new Eloquent model instance.
200      *
201      * @param array $attributes
202      * @return void
203      */
204     public function __construct(array $attributes = [])
205     {
206         $this->bootIfNotBooted();
207
208         $this->initializeTraits();
209
210         $this->syncOriginal();
211
212         $this->fill($attributes);
213     }
214
215     /**
```

```
Model.php × 2022_07_24_181532_edit_column_created_at_and_updated_at_to_near_earth_object_table.php ×
```

```
1 <?php
2
3 use ...
4
5 return new class extends Migration
6 {
7     /**
8      * Run the migrations.
9      *
10     * @return void
11     */
12
13     public function up()
14     {
15         Schema::table('NearEarthObject', function (Blueprint $table) {
16             $table->renameColumn('created_at', 'createdAt');
17             $table->renameColumn('updated_at', 'updatedAt');
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      *
24      * @return void
25      */
26
27     public function down()
28     {
29         Schema::table('NearEarthObject', function (Blueprint $table) {
30             $table->renameColumn('createdAt', 'created_at');
31             $table->renameColumn('updatedAt', 'updated_at');
32         });
33     }
34 };
35 }
```

And as a result, I got the necessary table

The screenshot shows a database management interface with a toolbar at the top and a table definition below. The table is named 'NearEarthObject'. The columns listed are 'referenced', 'name', 'speed', 'is_hazardous', 'Date', 'created_at', and 'updated_at'. The 'created_at' and 'updated_at' columns are highlighted in yellow, indicating they have been renamed from 'created_at' and 'updated_at' respectively.

3.3. Create a route /neo/hazardous

First of all I create a controller NearEarthObjectController and edit the Factory, created earlier, to fill the database with test data, to check if everything works correctly

```
④ NearEarthObjectFactory.php ×
1 <?php
2
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6
7 /**
8 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\NearEarthObject>
9 */
10 class NearEarthObjectFactory extends Factory
11 {
12     /**
13      * Define the model's default state.
14      *
15      * @return array<string, mixed>
16      */
17     public function definition()
18     {
19         return [
20             'referenced' => random_int(1000000, 9999999),
21             'name' => $this->faker->name,
22             'speed' => $this->faker->randomFloat( nbMaxDecimals: 1000, min: 100000),
23             'is_hazardous' => $this->faker->boolean( chanceOfGettingTrue: 50),
24             'Date' => $this->faker->date,
25             'createdAt' => now()
26         ];
27     }
28 }
```

And created 50 objects

```
④ Model.php × ④ NearEarthObjectFactory.php ×
1 <?php
2
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6
7 /**
8 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\NearEarthObject>
9 */
10 class NearEarthObjectFactory extends Factory
11 {
12     /**
13      * Define the model's default state.
14      *
15      * @return array<string, mixed>
16      */
17     public function definition()
18     {
19         return [
20             'referenced' => random_int(1000000, 9999999),
21             'name' => $this->faker->name,
22             'speed' => $this->faker->randomFloat( nbMaxDecimals: 1000, min: 100000),
23             'is_hazardous' => $this->faker->boolean( chanceOfGettingTrue: 50),
24             'Date' => $this->faker->date
25         ];
26     }
27 }
```

File Git Window Help DataGripProjects - NearEarthObject

t

console NearEarthObject

50 rows ORDER BY WHERE

	referenced	name	speed	is_hazardous	Date	createdAt	updatedAt
1	1213074	Liana Lueilwitz	66609122.934011	0	2015-05-26	2022-07-24 18:16:03	2022-07-24 18:16:03
2	1237092	Sterling Trantow Sr.	100000	0	2009-11-11	2022-07-24 18:16:03	2022-07-24 18:16:03
3	1458727	Prof. Ryleigh Prosacco II	160515.27923916	1	2006-04-28	2022-07-24 18:16:03	2022-07-24 18:16:03
4	1858331	Elva Hauck	114145.92482358	1	1992-05-04	2022-07-24 18:16:03	2022-07-24 18:16:03
5	2063112	Dr. Emmie Harris DVM	100000	1	2007-05-31	2022-07-24 18:16:03	2022-07-24 18:16:03
6	3022011	Ms. Lily Dietrich DDS	100000	0	2000-03-18	2022-07-24 18:16:03	2022-07-24 18:16:03
7	3045592	Nelda Stracke	100000	1	2010-10-10	2022-07-24 18:16:03	2022-07-24 18:16:03
8	3158920	Jean Kerluke	100000	0	2004-01-27	2022-07-24 18:16:02	2022-07-24 18:16:02
9	3289435	Candido Schiller	214875160.72832	1	2003-01-13	2022-07-24 18:16:03	2022-07-24 18:16:03
10	3342972	Prof. Shaniya Borer II	5500928.3087496	1	1970-07-20	2022-07-24 18:16:03	2022-07-24 18:16:03
11	3614531	Nicolas Harvey	100000	1	1988-02-11	2022-07-24 18:16:03	2022-07-24 18:16:03
12	3864588	Dr. Laury McKenzie	100000	0	2009-07-25	2022-07-24 18:16:03	2022-07-24 18:16:03
13	4281962	Holly Becker I	100000	0	1988-05-06	2022-07-24 18:16:03	2022-07-24 18:16:03
14	4506478	Eden Mann V	100000	0	2007-04-06	2022-07-24 18:16:03	2022-07-24 18:16:03
15	4570564	Mr. Francis Cruickshank	100000	1	2017-09-11	2022-07-24 18:16:02	2022-07-24 18:16:02
16	4587169	Prof. Maynard Stamm DDS	100000	1	1991-05-14	2022-07-24 18:16:03	2022-07-24 18:16:03
17	4709180	Janick Jacobi	10275172.39471	0	2013-10-13	2022-07-24 18:16:03	2022-07-24 18:16:03
18	4757204	Camila Lemke	100000	0	2017-09-02	2022-07-24 18:16:03	2022-07-24 18:16:03
19	5320888	Lera Runte	100000	1	2008-09-16	2022-07-24 18:16:03	2022-07-24 18:16:03
20	5738590	Thora Spencer	100000	1	1980-04-14	2022-07-24 18:16:03	2022-07-24 18:16:03
21	5842307	Wilburn Blick	4234841.0298988	1	1972-03-08	2022-07-24 18:16:03	2022-07-24 18:16:03
22	6170602	Citlalli Haag	100000	1	2004-10-26	2022-07-24 18:16:03	2022-07-24 18:16:03
23	6232932	Myah Bernhard	100000	1	2010-10-11	2022-07-24 18:16:03	2022-07-24 18:16:03
24	6290554	Coleman Quigley	100000	1	1975-06-02	2022-07-24 18:16:02	2022-07-24 18:16:02
25	6461072	Josie Kovacek	100000	0	1993-03-01	2022-07-24 18:16:03	2022-07-24 18:16:03
26	7429700	Ella Hills	497385.95318857	0	1987-04-30	2022-07-24 18:16:03	2022-07-24 18:16:03
27	7499821	Lenna Powlowski	100000	1	1993-04-02	2022-07-24 18:16:03	2022-07-24 18:16:03
28	7523138	Ciara Koelpin	6940550.3517189	1	1979-07-17	2022-07-24 18:16:03	2022-07-24 18:16:03
29	7747784	Colin Jacobson V	262641128.60397	0	1982-10-29	2022-07-24 18:16:03	2022-07-24 18:16:03
30	7776050	Bernadette Howell	100000	1	2007-12-22	2022-07-24 18:16:03	2022-07-24 18:16:03
31	7879058	Aimee Pouros	68046034.353069	0	1984-02-21	2022-07-24 18:16:02	2022-07-24 18:16:02
32	7959618	Miss Vernie Stokes	100000	1	2005-11-22	2022-07-24 18:16:03	2022-07-24 18:16:03
33	7966883	Olaf Emmerich	100000	1	2017-11-26	2022-07-24 18:16:03	2022-07-24 18:16:03
34	8081609	Alec Zboncak	100000	0	1986-01-04	2022-07-24 18:16:03	2022-07-24 18:16:03
35	8124176	Taryn Beier II	399128.44175848	0	1972-07-14	2022-07-24 18:16:03	2022-07-24 18:16:03
36	8214231	Prof. Elwin Stiedemann DDS	58697481.562064	1	2011-12-07	2022-07-24 18:16:03	2022-07-24 18:16:03

es Database Changes

Execution 240 ms

After the base was complete, I started editing the controller, created a method and created a route. Since the specified route did not match any action from the Actions Handled By Resource Controller table, I created the method "hazardous" that handled action. And it is not specified in the task, but a situation may arise that we do not need all the data, but only certain fields. Therefore, we need to create a Resource

GET https://api.nasa.gov/ne

GET http://laravel.test.task/

No Environment

http://laravel.test.task/api/neo/hazardous/

Send

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (12) Test Results

Status: 200 OK Time: 90 ms Size: 5.31 KB Save Response

Pretty Raw Preview Visualize

```
[{"referenced":3967687,"name":"Dr. Jennings Bailey PhD","speed":46707213.8486,"is_hazardous":1,"Date":"1992-04-10","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":9586248,"name":"Nayeli Steuber","speed":13434844.238867,"is_hazardous":1,"Date":"1999-10-12","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":6816892,"name":"Marge Hackett","speed":74515891.45487,"is_hazardous":1,"Date":"1989-03-28","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":6962891,"name":"Ms. Rice I","speed":100000,"is_hazardous":1,"Date":"1988-04-28","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":69456193,"name":"Alicia Mante","speed":160456193.124520,"is_hazardous":1,"Date":"1993-04-23","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":3569281,"name":"Burdette Will","speed":100000,"is_hazardous":1,"Date":"1995-12-23","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":3786124,"name":"Kaleb Mills","speed":100000,"is_hazardous":1,"Date":"2006-11-09","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":6411342,"name":"Jezel Ratke","speed":1329975.842868,"is_hazardous":1,"Date":"1978-06-24","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":5853758,"name":"Arch Walter MD","speed":100000,"is_hazardous":1,"Date":"2003-11-30","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":8836661,"name":"Prof. Johnny McClure","speed":29399532.451567,"is_hazardous":1,"Date":"2000-09-07","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":4718702,"name":"Jesse Carroll","speed":100000,"is_hazardous":1,"Date":"1984-03-27","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":5536262,"name":"Damian Block DDS","speed":100000,"is_hazardous":1,"Date":"2013-03-12","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":9734124,"name":"Eudox Lakin Jr.","speed":100000,"is_hazardous":1,"Date":"1987-11-03","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":7669433,"name":"Jeff Mayert DDS","speed":100000,"is_hazardous":1,"Date":"2009-06-11","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":7738456,"name":"Alivia Yundt","speed":100000,"is_hazardous":1,"Date":"1977-08-10","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":1492688,"name":"Mohammed Mante","speed":100000,"is_hazardous":1,"Date":"1991-02-06","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":6659461,"name":"Joana Jacobson II","speed":1621974.618043,"is_hazardous":1,"Date":"1992-04-06","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":3048513,"name":"Shaun Lesch","speed":100000,"is_hazardous":1,"Date":"2009-03-03","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":4042341,"name":"Wendy Lesch","speed":1334641.93376362,"is_hazardous":1,"Date":"2008-12-05","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":6373910,"name":"Miss Fan Beahan","speed":100000,"is_hazardous":1,"Date":"1974-05-07","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":294612,"name":"Clovis Corkery","speed":100000,"is_hazardous":1,"Date":"1974-11-10","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:43.000000Z"}, {"referenced":7819189,"name":"Ms. Isabella Wisoky","speed":1641266.82226721,"is_hazardous":1,"Date":"2014-04-03","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:44.000000Z"}, {"referenced":4238929,"name":"Axvid Heller","speed":32231494.8823803,"is_hazardous":1,"Date":"2002-09-26","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:44.000000Z"}, {"referenced":8288264,"name":"Cameron Schamberger","speed":1929783.5461156,"is_hazardous":1,"Date":"2002-04-24","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:44.000000Z"}, {"referenced":4631956,"name":"Prof. Mariano Bayer Sr.","speed":100000,"is_hazardous":1,"Date":"1998-09-13","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:44.000000Z"}, {"referenced":3642039,"name":"Mz. Nicolas O'Keefe","speed":100000,"is_hazardous":1,"Date":"2002-09-01","createdAt":"2022-07-24T08:32:43.000000Z","updatedAt":"2022-07-24T08:32:44.000000Z"}]
```

3.4. Create a route /neo/fastest?hazardous=(true|false)

For this I added the corresponding method to the NearEarthObjectController and the corresponding route in api.php

```
1 <?php
2
3 namespace App\Http\Controllers\Api;
4
5 use App\Http\Controllers\Controller;
6 use App\Http\Resources\NearEarthObjectResources;
7 use App\Models\NearEarthObject;
8 use Illuminate\Http\Request;
9 use PHPUnit\Util\Filter;
10
11 class NearEarthObjectController extends Controller
12 {
13     /**
14      * Display all DB entries which contain potentially hazardous asteroids.
15      *
16      * @return \Illuminate\Http\Resources\Json\AnonymousResourceCollection
17      */
18     public function hazardous()
19     {
20         $objects = NearEarthObject::where('is_hazardous', 1)->get();
21         return NearEarthObjectResources::collection($objects);
22     }
23
24     public function fastest(Request $request)
25     {
26         $hazardous = false;
27         if($request->has( key: 'hazardous') && $request->input( key: 'hazardous')) {
28             $hazardous = $request->boolean( key: 'hazardous');
29         }
30         $objects = NearEarthObject::where('is_hazardous', $hazardous)->get();
31         return $objects->sortByDesc('speed');
32     }
33
34 }
35
```

I'm not sure if I handled the query parameters correctly, but I found no other way and decided to do it this way. And it worked correctly in the end

GET http://laravel.test.task/api/neo/fastest?hazardous=false

Params

KEY	VALUE	DESCRIPTION	Bulk Ed
hazardous	false		
Key	Value	Description	

Body

```
[{"id": 8, "referenced": 6421194, "name": "Emie Volkman", "speed": 298393379.44248, "is_hazardous": 0, "Date": "1975-07-22", "createdAt": "2022-07-24T08:32:43.000000Z", "updatedAt": "2022-07-24T08:32:43.000000Z"}, {"id": 22, "referenced": 9410213, "name": "Meile Schowalter", "speed": 65698699.598663, "is_hazardous": 0, "Date": "1999-10-05", "createdAt": "2022-07-24T08:32:43.000000Z", "updatedAt": "2022-07-24T08:32:44.000000Z"}, {"id": 16, "referenced": 2548279, "name": "Asha Brown", "speed": 65664632.991166, "is_hazardous": 0, "Date": "2020-10-17", "createdAt": "2022-07-24T08:32:43.000000Z", "updatedAt": "2022-07-24T08:32:44.000000Z"}]
```

GET http://laravel.test/task/api/neo/fastest?hazardous=true

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> hazardous	true	
Key	Value	Description

Body Cookies (2) Headers (12) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 103 ms Size: 5.44 KB Save Response

```
1 "4": {
2     "referenced": 3289435,
3     "name": "Candido Schiller",
4     "speed": 214875160.72832,
5     "is_hazardous": 1,
6     "Date": "2003-01-13",
7     "createdAt": "2022-07-24T18:16:03.000000Z",
8     "updatedAt": "2022-07-24T18:16:03.000000Z"
9 },
10 "20": {
11     "referenced": 8214231,
12     "name": "Prof. Elwin Stiedemann DDS",
13     "speed": 56697481.562664,
14     "is_hazardous": 1,
15     "Date": "2011-12-07",
16     "createdAt": "2022-07-24T18:16:03.000000Z",
17     "updatedAt": "2022-07-24T18:16:03.000000Z"
18 },
19 "16": {
20     "referenced": 7523138,
21     "name": "Cizara Koelpin",
22     "speed": 6948650.3517189,
23     "is_hazardous": 1,
24     "Date": "1979-07-17",
25     "createdAt": "2022-07-24T18:16:03.000000Z",
26     "updatedAt": "2022-07-24T18:16:03.000000Z"
27 },
28 }
```

Additional Task.

1. to request the data from the last 3 days from nasa api (<https://api.nasa.gov/>) -> API Name Asteroids - NeoWs

In order to work with this site, we need to get the API Key. The instructions are obtained from the site, this key I save in a .env file

```
 .env × Model.php × NearEarthObjectFactory.php × NearEarthObject.php × NearEarthObjectResources.php × NearEarthObjectController.php ×
 25 NERACHED_HOST=127.0.0.1
 26
 27 REDIS_HOST=127.0.0.1
 28 REDIS_PASSWORD=null
 29 REDIS_PORT=6379
 30
 31 MAIL_MAILER=smtp
 32 MAIL_HOST=mailhog
 33 MAIL_PORT=1025
 34 MAIL_USERNAME=null
 35 MAIL_PASSWORD=null
 36 MAIL_ENCRYPTION=null
 37 MAIL_FROM_ADDRESS="hello@example.com"
 38 MAIL_FROM_NAME="${APP_NAME}"
 39
 40 AWS_ACCESS_KEY_ID=
 41 AWS_SECRET_ACCESS_KEY=
 42 AWS_DEFAULT_REGION=us-east-1
 43 AWS_BUCKET=
 44 AWS_USE_PATH_STYLE_ENDPOINT=false
 45
 46 PUSHER_APP_ID=
 47 PUSHER_APP_KEY=
 48 PUSHER_APP_SECRET=
 49 PUSHER_HOST=
 50 PUSHER_PORT=443
 51 PUSHER_SCHEME=https
 52 PUSHER_APP_CLUSTER=mt1
 53
 54 VITE_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
 55 VITE_PUSHER_HOST="${PUSHER_HOST}"
 56 VITE_PUSHER_PORT="${PUSHER_PORT}"
 57 VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
 58 VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
 59
 60 NASA_API_KEY="0R5qBEo1cF68WqcxN6PQ7pnsPig2w75Ctf09je0"
 61
```

I will use Guzzle for my work. I create an `ImportDataClient` class in which I use the documentation to create the client.

```
1 .env x ImportDataClient.php x ImportDataFromNasaCommand.php x
2 <?php
3
4 namespace App\Components;
5
6 use GuzzleHttp\Client;
7
8 class ImportDataClient
9 {
10     public $client;
11
12     /**
13      * @param Client $client
14      */
15     public function __construct()
16     {
17         $this->client = new Client([
18             // Base URI is used with relative requests
19             'base_uri' => 'https://api.nasa.gov/neo/rest/v1/',
20             // You can set any number of default request options.
21             'timeout' => 2.0,
22             'verify' => false
23         ]);
24     }
25
26 }
27
```

After that, I created a command `import:nasadata` that runs the query and gets the data

```
7 .env x ImportDataClient.php x ImportDataFromNasaCommand.php x
8
9 class ImportDataFromNasaCommand extends Command
10 {
11     /**
12      * The name and signature of the console command.
13      *
14      * @var string
15     protected $signature = 'import:nasadata';
16
17     /**
18      * The console command description.
19      *
20      * @var string
21     protected $description = 'Get data from NASA';
22
23     /**
24      * Execute the console command.
25      *
26      * @return int
27     */
28     public function handle()
29     {
30         $import = new ImportDataClient();
31         $response = $import->client->request( method: 'GET', uri: 'feed', [
32             'query' => [
33                 'start_date' => '2015-09-07',
34                 'end_date' => '2015-09-08',
35                 'api_key' => '0R5q8Eo1cF68WqcxN6PQ7pnsPig2w75Ctf09je0'
36             ]
37         ]);
38         $data = json_decode($response->getBody()->getContents());
39         dd($data);
40     }
41 }
42
```

As you can see, the date is successfully obtained

I can make sure it is, because if you run the same query in another program, there will be the same data

https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-05-07&end_date=2015-09-08&api_key=0R5qBE0tCF68WqcxN6PQ/pnsPIg2w/5CtI09...

GET
https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-07&end_date=2015-09-08&api_key=OR5qBEo1cF68WqcxN6PQ7pnsPig2w75CCtf09je0
Send

Params	Authorization	Headers (8)	Body	Pre-request Script	Tests	Settings	Cookies	
							...	Bulk Edit
KEY			VALUE		DESCRIPTION		...	Bulk Edit
<input checked="" type="checkbox"/> start_date			2015-09-07					
<input checked="" type="checkbox"/> end_date			2015-09-08					
<input checked="" type="checkbox"/> api_key			OR5qBEo1cF68WqcxN6PQ7pnsPig2w75CCtf09je0					
Body	Cookies	Headers (20)	Test Results					Save Response

After that I create a migration to change the database accordingly to the requirements. To do this we need to make some small changes: rename Date to date and referenced to reference

```

<?php
use ...;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('NearEarthObject', function (Blueprint $table) {
            $table->renameColumn('referenced', 'reference');
            $table->renameColumn('Date', 'date');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('NearEarthObject', function (Blueprint $table) {
            $table->renameColumn('reference', 'referenced');
            $table->renameColumn('date', 'Date');
        });
    }
};

```

The result is a table as in the model

	reference	name	speed	is_hazardous	date
1	1213074	Liana Lueilwitz	66609122.934011	0	2015-05-26
2	1237092	Sterling Trantow Sr.	100000	0	2009-11-11
3	1458727	Prof. Ryleigh Prosacco II	160515.27923916	1	2006-04-28
4	1858331	Elva Hauck	114145.92482358	1	1992-05-04
5	2063112	Dr. Emmie Harris DVM	100000	1	2007-05-31
6	3022011	Ms. Lily Dietrich DDS	100000	0	2000-03-18
7	3045592	Nelda Stracke	100000	1	2010-10-10
8	3158920	Jean Kerluke	100000	0	2004-01-27
9	3289435	Candido Schiller	214875160.72832	1	2003-01-13
10	3342972	Prof. Shaniya Borer II	5500928.3087496	1	1970-07-20
11	3614531	Nicolas Harvey	100000	1	1988-02-11
12	3864588	Dr. Laury McKenzie	100000	0	2009-07-25
13	4281962	Holly Becker I	100000	0	1988-05-06
14	4506478	Eden Mann V	100000	0	2007-04-06
15	4570564	Mr. Francis Cruickshank	100000	1	2017-09-11
16	4587169	Prof. Maynard Stamm DDS	100000	1	1991-05-14
17	4709180	Janick Jacobi	10275172.39471	0	2013-10-13
18	4757204	Camila Lemke	100000	0	2017-09-02
19	5320888	Lera Runte	100000	1	2008-09-16
20	5738590	Thora Spencer	100000	1	1980-04-14
21	5842307	Wilburn Blick	4234841.0298988	1	1972-03-08
22	6170602	Citlalli Haag	100000	1	2004-10-26
23	6232932	Myah Bernhard	100000	1	2010-10-11
24	6290554	Coleman Quigley	100000	1	1975-06-02
25	6461072	Josie Kovacek	100000	0	1993-03-01
26	7429700	Ella Hills	497385.95318857	0	1987-04-30
27	7499821	Lenna Powlowski	100000	1	1993-04-02
28	7523138	Ciara Koelpin	6940550.3517189	1	1979-07-17
29	7747784	Colin Jacobson V	262641128.60397	0	1982-10-29
30	7776050	Bernadette Howell	100000	1	2007-12-22
31	7879058	Aimee Pouros	68046034.353069	0	1984-02-21
32	7959618	Miss Vernie Stokes	100000	1	2005-11-22
33	7966883	Olaf Emmerich	100000	1	2017-11-26
34	8081609	Alec Zboncak	100000	0	1986-01-04
35	8124176	Taryn Beier II	399128.44175848	0	1972-07-14
36	8214231	Prof. Elwin Stiedemann DDS	58697481.562064	1	2011-12-07

Now we need to edit the command to get the data and add data to the table. It doesn't look good, but I didn't know how else to refine it

```
18     /**
19      * The console command description.
20      *
21      * @var string
22      */
23     protected $description = 'Get data from NASA';
24
25     /**
26      * Execute the console command.
27      *
28      * @return int
29      */
30     public function handle()
31     {
32         $import = new ImportDataClient();
33         $response = $import->client->request( method: 'GET', uri: 'feed', [
34             'query' => [
35                 'start_date' => date( format: 'Y-m-d H:i:s', strtotime( datetime: "-3 days" )),
36                 'end_date' => date( format: 'Y-m-d H:i:s'),
37                 'api_key' => env( key: 'NASA_API_KEY')
38             ]
39         ]);
40         $data = json_decode($response->getBody()->getContents());
41         foreach ($data->near_earth_objects as $objects) {
42             foreach ($objects as $object) {
43                 NearEarthObject::firstOrCreate([
44                     'name' => $object->name
45                 ], [
46                     'date' => $object->close_approach_data[0]->close_approach_date,
47                     'reference' => $object->neo_reference_id,
48                     'name' => $object->name,
49                     'speed' => $object->close_approach_data[0]->relative_velocity->kilometers_per_hour,
50                     'is_hazardous' => $object->is_potentially_hazardous_asteroid,
51                 ]);
52             }
53         }
54     }
55 }
```

console < NearEarthObject > ...

WHERE ORDER BY

	reference	name	speed	is_hazardous	date	createdAt	updatedAt
1	2022099	22099 (2000 EX106)	34722.4520770863	0	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
2	2141765	141765 (2002 MP3)	59917.3270302622	0	2022-07-23	2022-07-24 20:04:40	2022-07-24 20:04:40
3	2205378	205378 (2001 BJ16)	36323.0120372551	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
4	2215188	215188 (2000 NM)	49936.5971292589	0	2022-07-23	2022-07-24 20:04:40	2022-07-24 20:04:40
5	2303450	303450 (2005 BY2)	80547.0702880363	1	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
6	2398188	398188 Agni (2010 LE15)	27702.2938589686	1	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
7	2523796	523796 (2016 LE51)	77842.5975242852	0	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
8	3092506	(2001 WJ4)	25113.6536367423	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
9	3430304	(2008 TX3)	20886.5698831625	0	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
10	3444552	(2009 BG11)	62036.4339295236	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
11	3548887	(2010 TE55)	8590.7465565463	0	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
12	3561029	(2011 GE3)	17348.8994298438	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
13	3568386	(2011 M04)	38200.7845773187	0	2022-07-23	2022-07-24 20:04:40	2022-07-24 20:04:40
14	3596030	(2012 BV13)	26878.4116162448	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
15	3602837	(2012 HN13)	19029.3824928529	0	2022-07-23	2022-07-24 20:04:40	2022-07-24 20:04:40
16	3644180	(2013 MR6)	69722.5501710252	0	2022-07-24	2022-07-24 20:04:39	2022-07-24 20:04:39
17	3653969	(2013 WR45)	61484.3712346968	0	2022-07-24	2022-07-24 20:04:40	2022-07-24 20:04:40
18	3678630	(2014 OL339)	29089.4714264708	0	2022-07-24	2022-07-24 20:04:40	2022-07-24 20:04:40
19	3740342	(2016 AU192)	65643.5791951834	0	2022-07-23	2022-07-24 20:04:40	2022-07-24 20:04:40
20	3753595	(2016 KO)	53557.2712764875	0	2022-07-24	2022-07-24 20:04:40	2022-07-24 20:04:40
21	3756949	(2016 PF8)	54163.7377371731	1	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
22	3767310	(2017 BU32)	56950.6325126119	0	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
23	3781414	(2017 RX2)	51169.1967062136	0	2022-07-24	2022-07-24 20:04:40	2022-07-24 20:04:40
24	3802409	(2018 GY)	36774.9689859403	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
25	3802553	(2018 EA10)	29062.5587831128	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
26	3826626	(2018 PY7)	31520.5959398694	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
27	3837653	(2019 AY4)	95197.5725528203	0	2022-07-24	2022-07-24 20:04:40	2022-07-24 20:04:40
28	3840866	(2019 GJ4)	12718.5881987385	0	2022-07-23	2022-07-24 20:04:40	2022-07-24 20:04:40
29	3843342	(2019 ON)	42202.6834633556	0	2022-07-24	2022-07-24 20:04:40	2022-07-24 20:04:40
30	3989284	(2020 BW12)	36466.3160467229	0	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
31	3991581	(2020 CL2)	54293.8582554775	0	2022-07-24	2022-07-24 20:04:40	2022-07-24 20:04:40
32	3991729	(2020 DC2)	58553.3753177602	0	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03
33	54016781	(2020 HL7)	60977.0431301263	0	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
34	54049906	(2020 PP)	29430.8771407112	1	2022-07-21	2022-07-24 20:06:04	2022-07-24 20:06:04
35	54054748	(2020 TG)	20705.1766249273	0	2022-07-23	2022-07-24 20:04:40	2022-07-24 20:04:40
36	54139093	(2021 HB2)	49687.1387376311	0	2022-07-22	2022-07-24 20:06:03	2022-07-24 20:06:03

Database Changes

I also changed the name of the columns in the tasks earlier, so that everything would work as it should be

Conclusions

During the assignment, I tried to emphasize not the simple execution of tasks, because I am sure that it was possible to write everything much easier. I tried to match the PSR, but I'm not sure I always succeeded. I also couldn't test my code because I didn't work with them at all and I wasn't able to learn them that fast. Also, I had difficulties with the fact that there were a lot of questions before the tests, and on the one hand this opened up opportunities for action, but on the other hand it restricted me because the requirements were not clearly understood.