Лабораторная работа №4

Laravel и Elasticsearch

Введение

Перед началом работы над ЛР №4 выполните следующие действия:

1. Откройте ваш репозиторий из ЛР №3.
2. Откройте Merge Request / Pull Request (MR / PR), который вы создавали (результат выполнения ЛР №3).
3. Смержите (если не делали этого ранее) ветку в master / main.
4. Создайте (от ветки master / main) новую ветку под названием lab4.
5. В заданиях №1 - 3 продолжите работать в данном репозитории в ветке lab4.
6. Выполните задания №1 - 3 в ветке lab4 и создайте MR / PR.
7. Для задания №4 создайте новый репозиторий (для сервиса-потребителя).

7.1. Разверните фреймворк Laravel в master / main (как вы делали это ранее).

7.2. Создайте новую ветку lab4.

7.3. Выполните задание №4 в данной ветке и создайте MR / PR.

В ходе выполнения задания №4 у вас появится **два** новых репозитория:

1. Репозиторий сервиса-потребителя (на базе фреймворка Laravel)
2. Репозиторий с автогенерируемым клиентом.

   В данном репозитории не требуется создавать никаких дополнительных веток, необходимо сразу отправлять результат генерации в master / main.

Итого **три** репозитория:

1. Основной сервис (репозиторий из ЛР №3) **[R1]**
2. Репозиторий сервиса-потребитель **[R2]**
3. Репозиторий с автогенерируемым клиентом **[R3]**

Однако по результатам выполнения ЛР №4 необходимо отправить в ОРИОКС (Домашние задания) только **две** ссылки на MR / PR:

1. Ссылка на MR основного сервиса **[R1]**
2. Ссылка на MR сервиса-потребителя **[R2]**

Практическая часть

Задание 1. Создание индекса

Продумайте структуру и создайте индекс в Elasticsearch для вашей модели данных.

Индекс должен создаваться консольной командой.

Для взаимодействия с Elasticsearch из php используйте пакет elasticsearch/elasticsearch

Задание 2. Заполнение индекса

Реализуйте заполнение индекса данными.

Используйте обработчики событий моделей, чтобы при создании / изменении / удалении сущности, изменения применялись и к индексу.

Также создайте консольную команду, которая заполняет пустой индекс первичными данными.

Задание 3. Чтение из индекса

Реализуйте новый метод, который должен отдавать данные из индекса, в соответствии с переданными фильтром, сортировкой, пагинацией.

Опишите метод с помощью OAS и реализуйте по аналогии с ЛР №3.

Задание 4. Сервис-потребитель

Разверните новый сервис **[R2]**, который будет обращаться к основному **[R1]** для чтения данных из эластика.

Создайте клиент для основного сервиса с помощью пакета ensi/laravel-openapi-client-generator и запушьте в новый репозиторий **[R3]**.

Разверните новый сервис и подключите в него созданный пакет.

В новом сервисе **[R2]** создайте один метод, по аналогии с ЛР №3. Этот метод должен принимать фильтр, сортировку и пагинацию, и отдавать данные из Elasticsearch. Отличие с методом из задания 3 в том, что необходимо брать данные не напрямую из Elasticsearch, а из первого сервиса **[R1]**, используя созданный пакет **[R2]**.

Примеры

Подключение пакета

Для подключения и корректной работы пакета, созданного с помощью ensi/laravel-openapi-client-generator, необходимо произвести регистрацию через свой сервис-провайдер.

Пример:

```php
<?php

namespace App\Providers;

use Ivanov\MyClient\MyClientProvider;
use GuzzleHttp\Client;
use GuzzleHttp\ClientInterface;
use Illuminate\Support\ServiceProvider;

class OpenApiClientsServiceProvider extends ServiceProvider
{
	public function register(): void
	{
		$baseUri = config('openapi-clients.my-service-url');
		$configurationClass = MyClientProvider::$configuration;
		$this->app->bind($this->trimFQCN($configurationClass), fn () =>
(new $configurationClass())->setHost($baseUri));
		foreach (MyClientProvider::$apis as $api) {
			$this->app->when($this->trimFQCN($api))
				->needs(ClientInterface::class)
				->give(fn () => new Client([
					'base_uri' => $baseUri,
				]));
		}
	}

	private function trimFQCN(string $name): string
	{
		return ltrim($name, '\\');
	}
}
```

Файл /config/openapi-clients.php:

```php
<?php

return [
	// ...
	'my-service-url' => env('MY_SERVICE_URL'),
	// ...
];
```

Файл /.env:

```
// ...
MY_SERVICE_URL=http://localhost
// ...
```

Внедрение Api в Action

```php
<?php

namespace App\Domain\My\Actions;

use Ivanov\MyClient\Api\MyApi;

class SearchElasticAction
{
	public function __construct(protected MyApi $myApi)
	{
	}

	public function execute(): void
	{
		$this->myApi->searchElastic();
	}
}
```