PoSobota 107

Spameri/Elastic

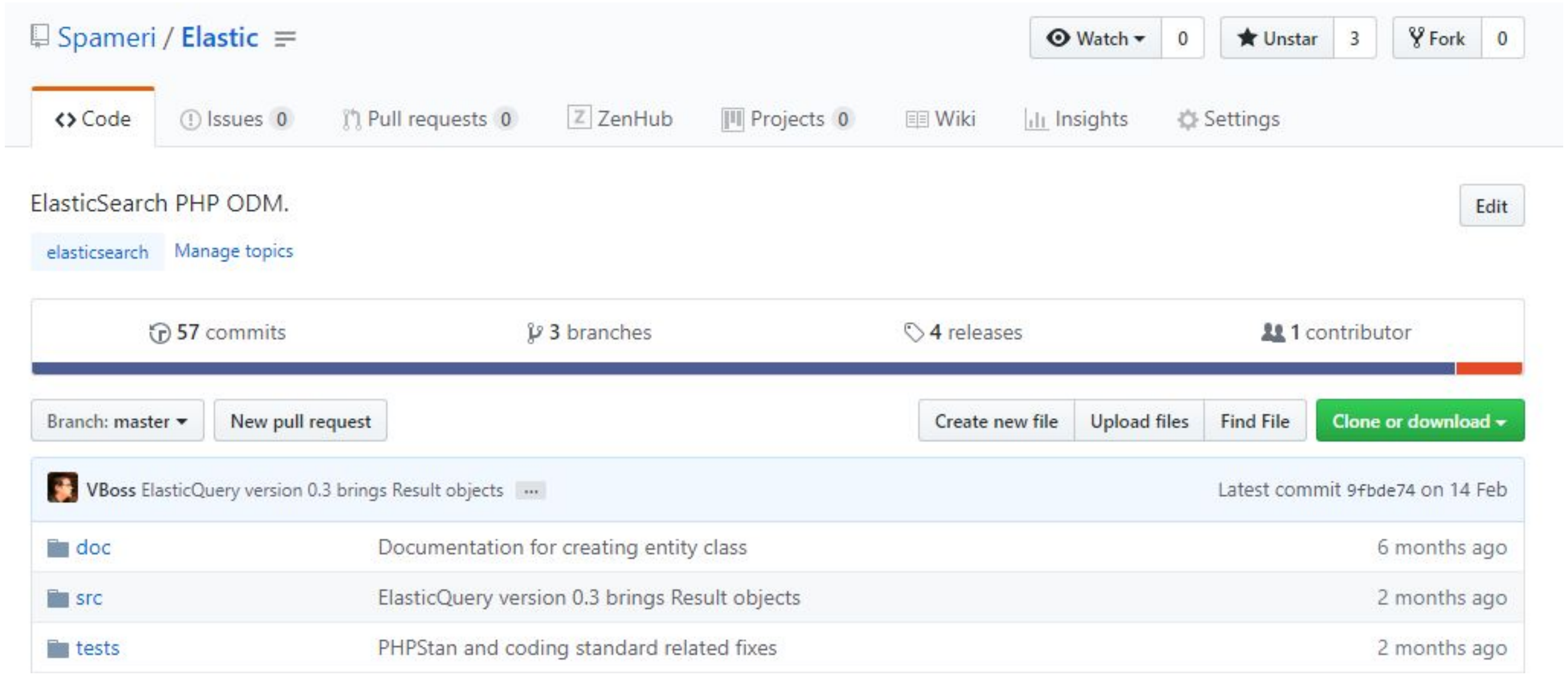
Václav Čevela / Developer

Máte hledání u vás na webu?

- Databázové hledání je limitující

- Ponořit se do ElasticSearch znamená spoustu polního programování 🚜

- Alternativy jsou: drahé, komplikované, chybí funkčnost (vyber 2 vlastnosti)



ElasticSearch PHP ODM.

57 commits | 3 branches | 4 releases | 1 contributor

VBoss ElasticQuery version 0.3 brings Result objects ... Latest commit 9fbde74 on 14 Feb

doc	Documentation for creating entity class	6 months ago
src	ElasticQuery version 0.3 brings Result objects	2 months ago
tests	PHPStan and coding standard related fixes	2 months ago

10 Steps to happiness

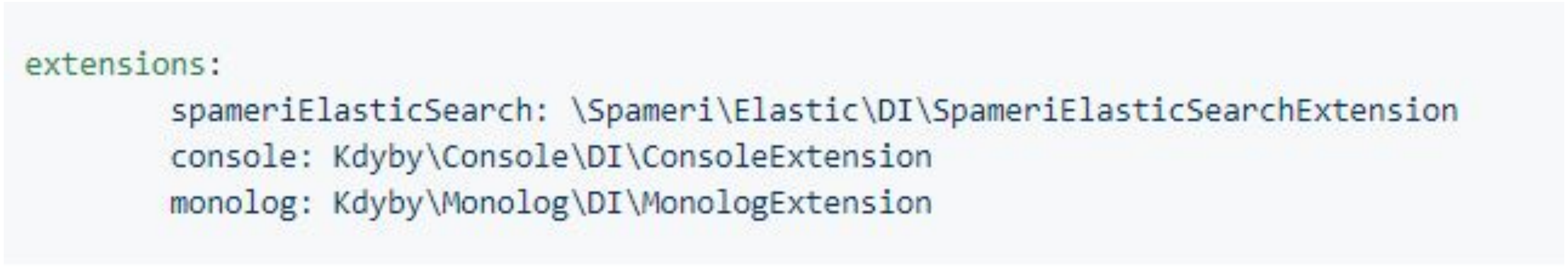
1. Composer
2. Configure
3. Entity
4. Mapping
5. Export
6. Frontend
7. List
8. Fine tuning
9. ???
10. Profit

10 Steps to Happiness
1. Hate less, love more
2. Worry less, dance more
3. Take less, give more
4. Consume less, create more
5. Frown less, smile more
6. Talk less, listen more
7. Fear less, try more
8. Judge less, accept more
9. Watch less, do more
10. Complain less, appreciate more
@sylvia duckworth

1. Composer

composer require **spameri/elastic**

2. Configure



```
extensions:
	spameriElasticSearch: \Spameri\Elastic\DI\SpameriElasticSearchExtension
	console: Kdyby\Console\DI\ConsoleExtension
	monolog: Kdyby\Monolog\DI\MonologExtension
```

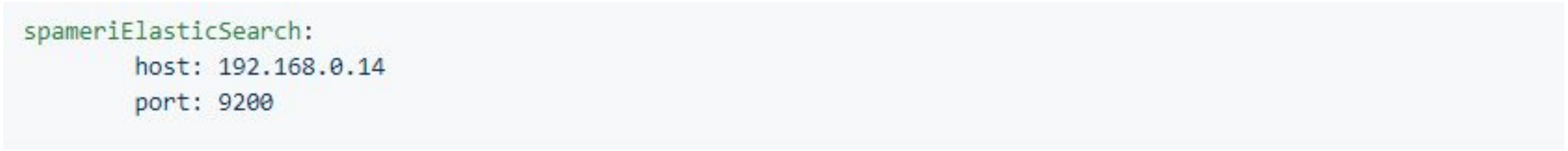
PeckaDesign

Václav Spamer Čevela
Spameri/Elastic

FB facebook.com/peckadesign

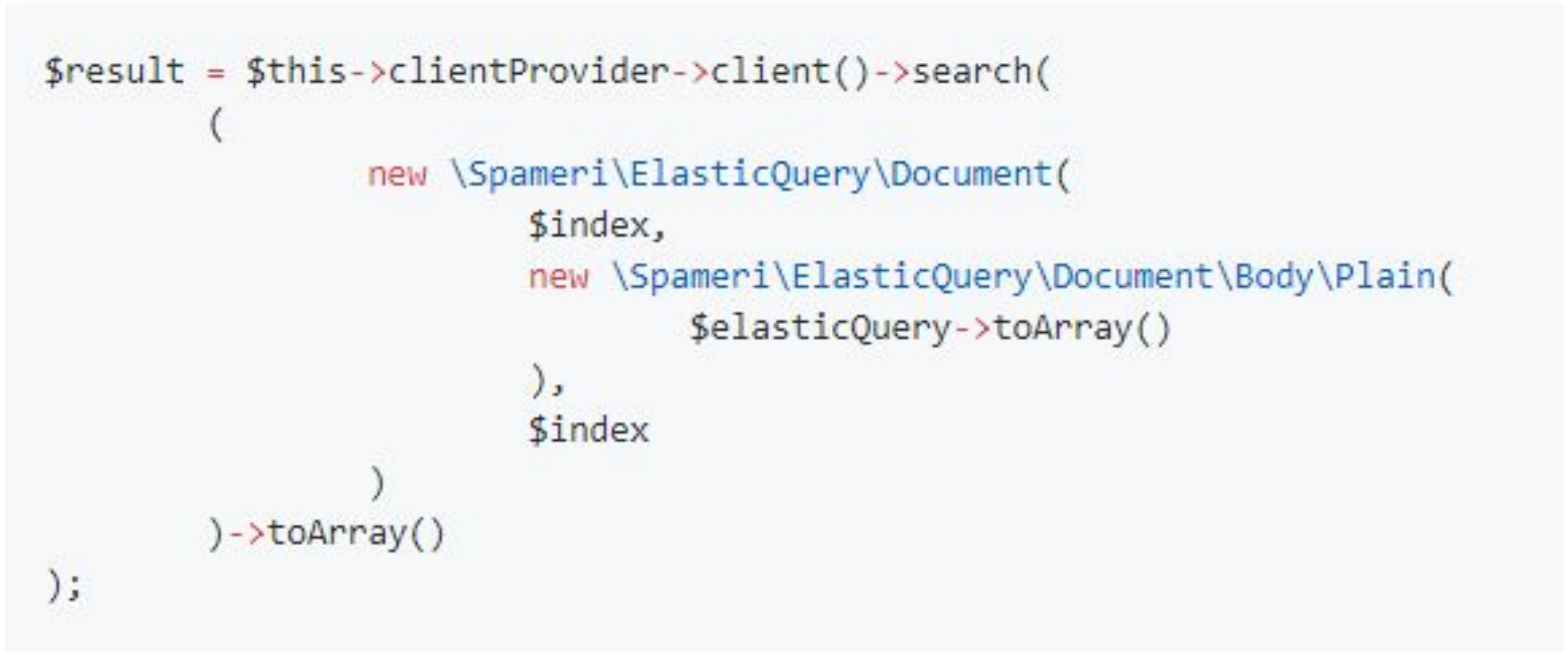
TW @Spamercz

2. Configure



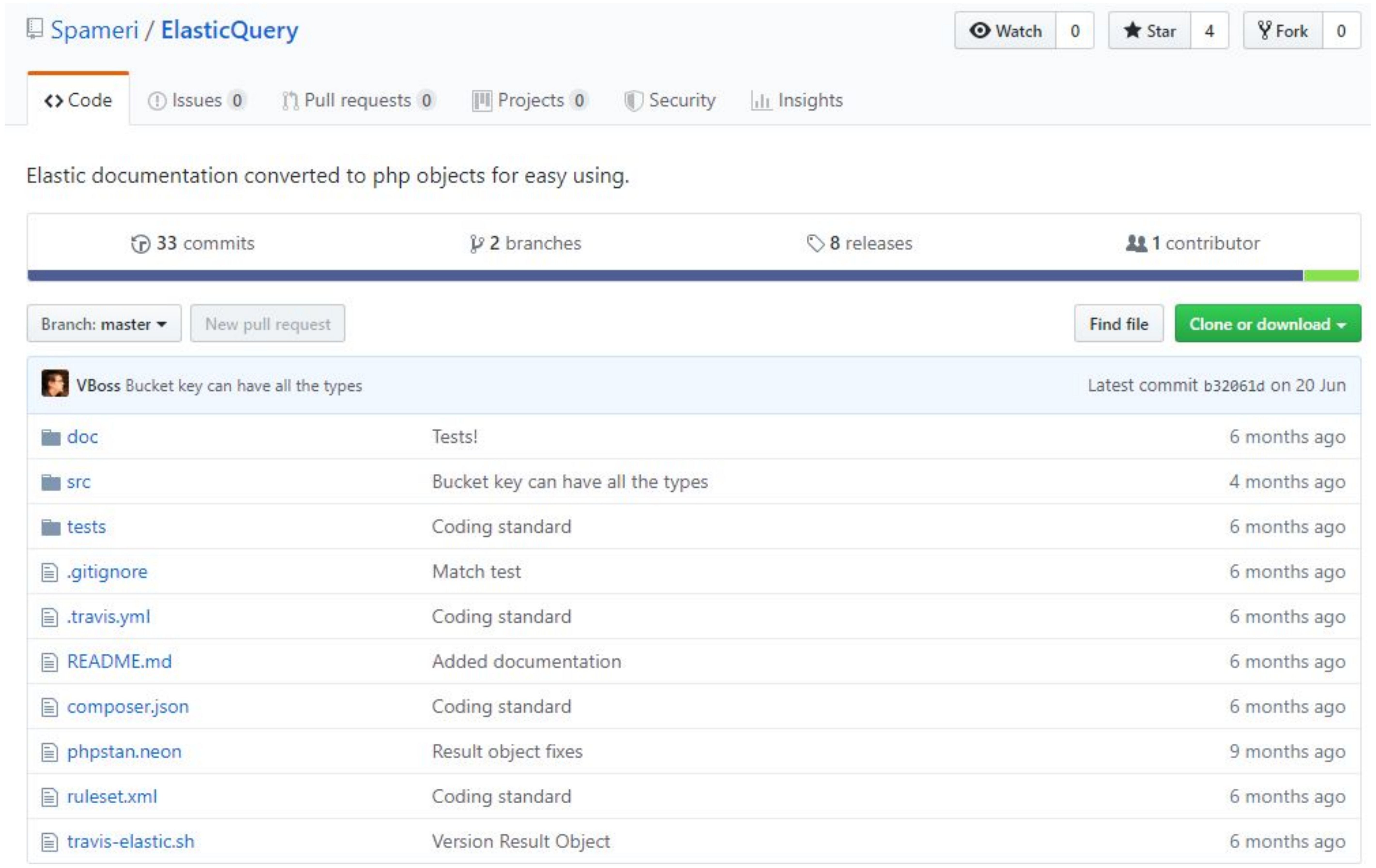
```
spameriElasticSearch:
        host: 192.168.0.14
        port: 9200
```

Basic usage



```php
$result = $this->clientProvider->client()->search(
        (
                new \Spameri\ElasticQuery\Document(
                        $index,
                        new \Spameri\ElasticQuery\Document\Body\Plain(
                                $elasticQuery->toArray()
                        ),
                        $index
                )
        )->toArray()
);
```

Spameri/ElasticQuery



Spameri / ElasticQuery

Elastic documentation converted to php objects for easy using.

33 commits | 2 branches | 8 releases | 1 contributor

VBoss Bucket key can have all the types — Latest commit b32061d on 20 Jun

doc	Tests!	6 months ago
src	Bucket key can have all the types	4 months ago
tests	Coding standard	6 months ago
.gitignore	Match test	6 months ago
.travis.yml	Coding standard	6 months ago
README.md	Added documentation	6 months ago
composer.json	Coding standard	6 months ago
phpstan.neon	Result object fixes	9 months ago
ruleset.xml	Coding standard	6 months ago
travis-elastic.sh	Version Result Object	6 months ago

ElasticQuery

- Query
- Filter
- Sort
- Aggregation
- Options

QueryCollection

- MustCollection
- ShouldCollection
- MustNotCollection

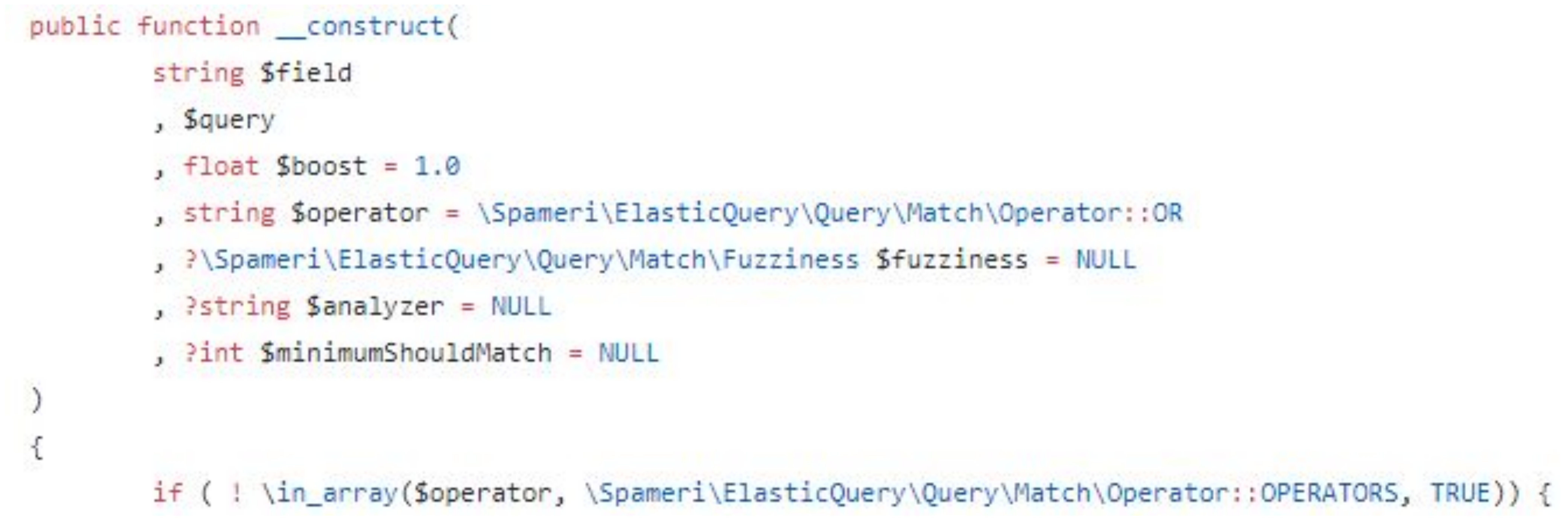
FilterCollection

- MustCollection

LeafQueryInterface

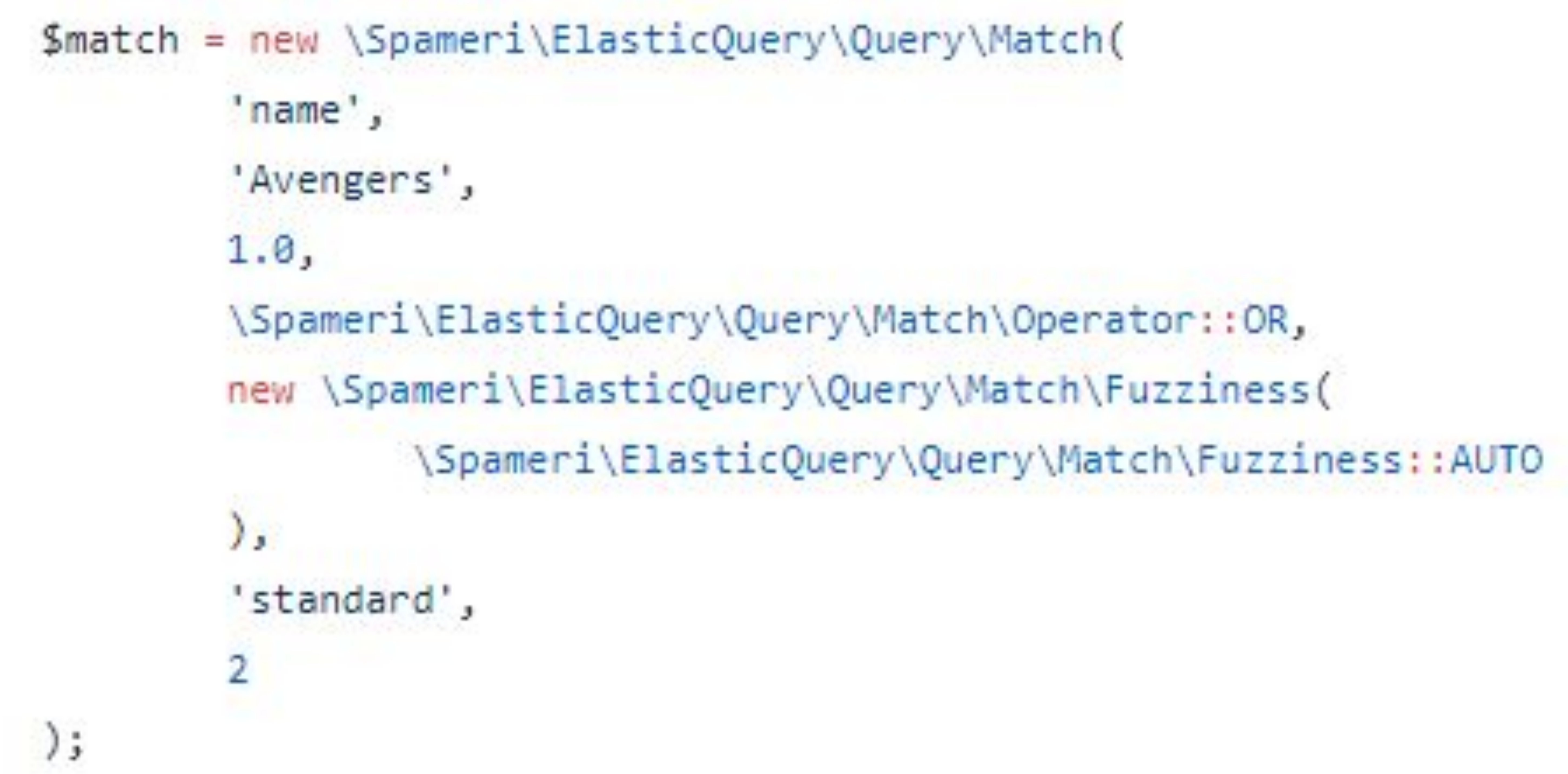
- Match
- Fuzzy
- Term
- Terms
- MatchPhrase
- Range
- WildCard

Match



```
public function __construct(
		string $field
		, $query
		, float $boost = 1.0
		, string $operator = \Spameri\ElasticQuery\Query\Match\Operator::OR
		, ?\Spameri\ElasticQuery\Query\Match\Fuzziness $fuzziness = NULL
		, ?string $analyzer = NULL
		, ?int $minimumShouldMatch = NULL
)
{
		if ( ! \in_array($operator, \Spameri\ElasticQuery\Query\Match\Operator::OPERATORS, TRUE)) {
```

Match

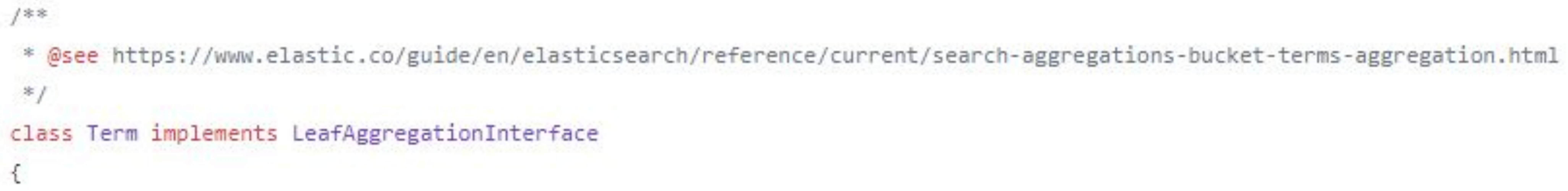


```
$match = new \Spameri\ElasticQuery\Query\Match(
        'name',
        'Avengers',
        1.0,
        \Spameri\ElasticQuery\Query\Match\Operator::OR,
        new \Spameri\ElasticQuery\Query\Match\Fuzziness(
                \Spameri\ElasticQuery\Query\Match\Fuzziness::AUTO
        ),
        'standard',
        2
);
```

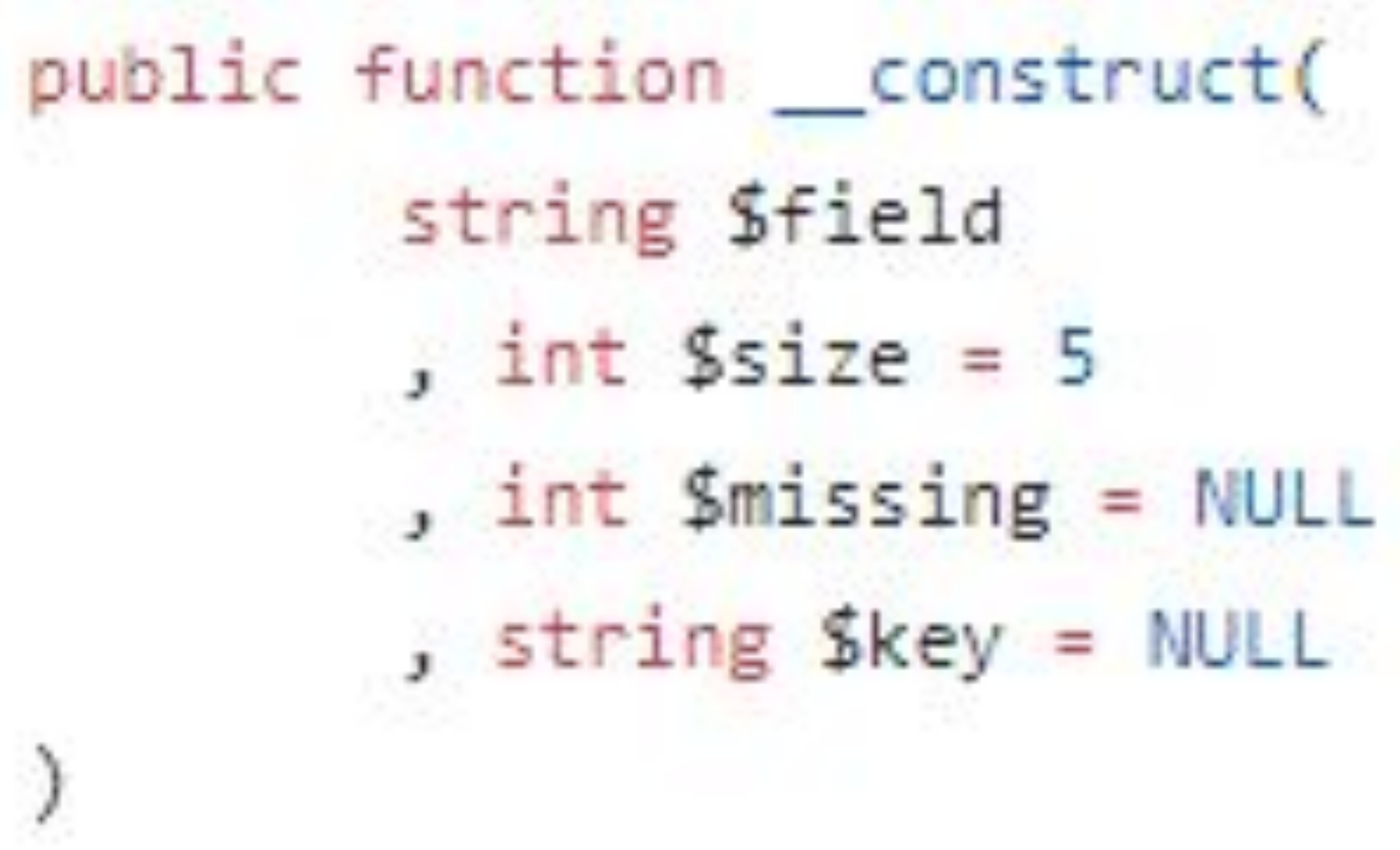
AggregationCollection

- Filter
- Nested
- Histogram
- Range
- Term

Term



```
/**
 * @see https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket-terms-aggregation.html
 */
class Term implements LeafAggregationInterface
{
```



```
public function __construct(
        string $field
        , int $size = 5
        , int $missing = NULL
        , string $key = NULL
)
```

PeckaDesign

Václav Spamer Čevela
Spameri/Elastic

FB facebook.com/peckadesign

TW @Spamercz

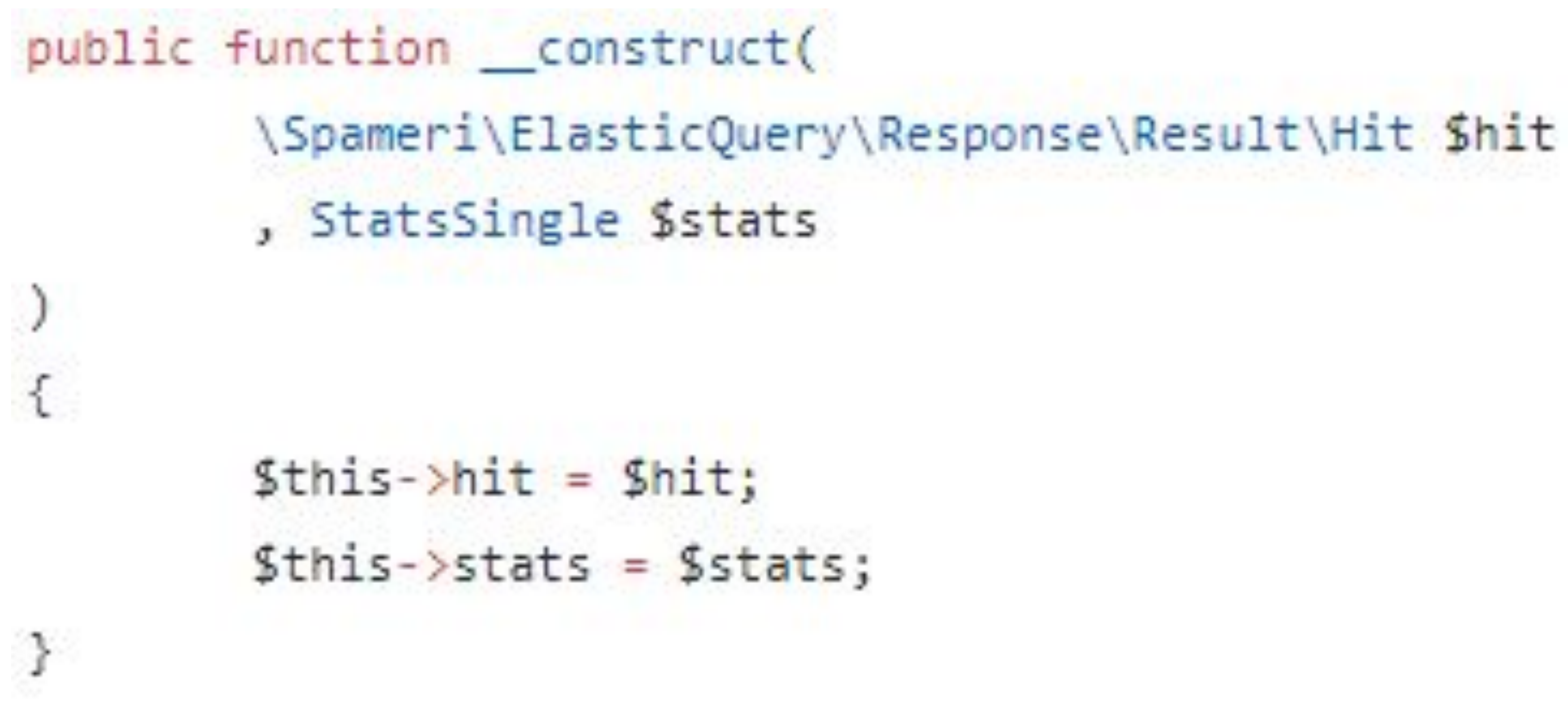
Document + Body

- Base transport object
- Settings
- Bulk
- Plain

ResultMapper

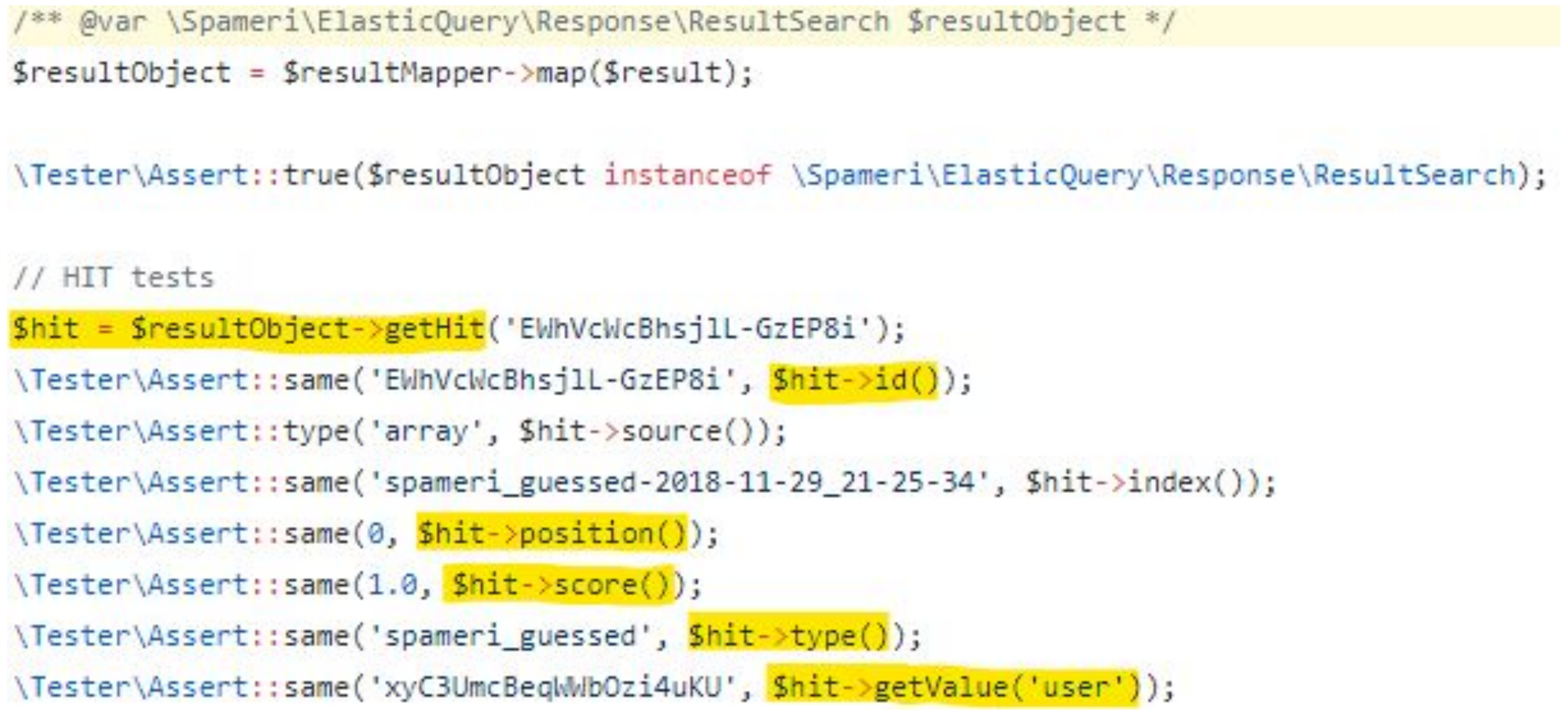
- Version
- Bulk
- Search
- Single

ResultSingle

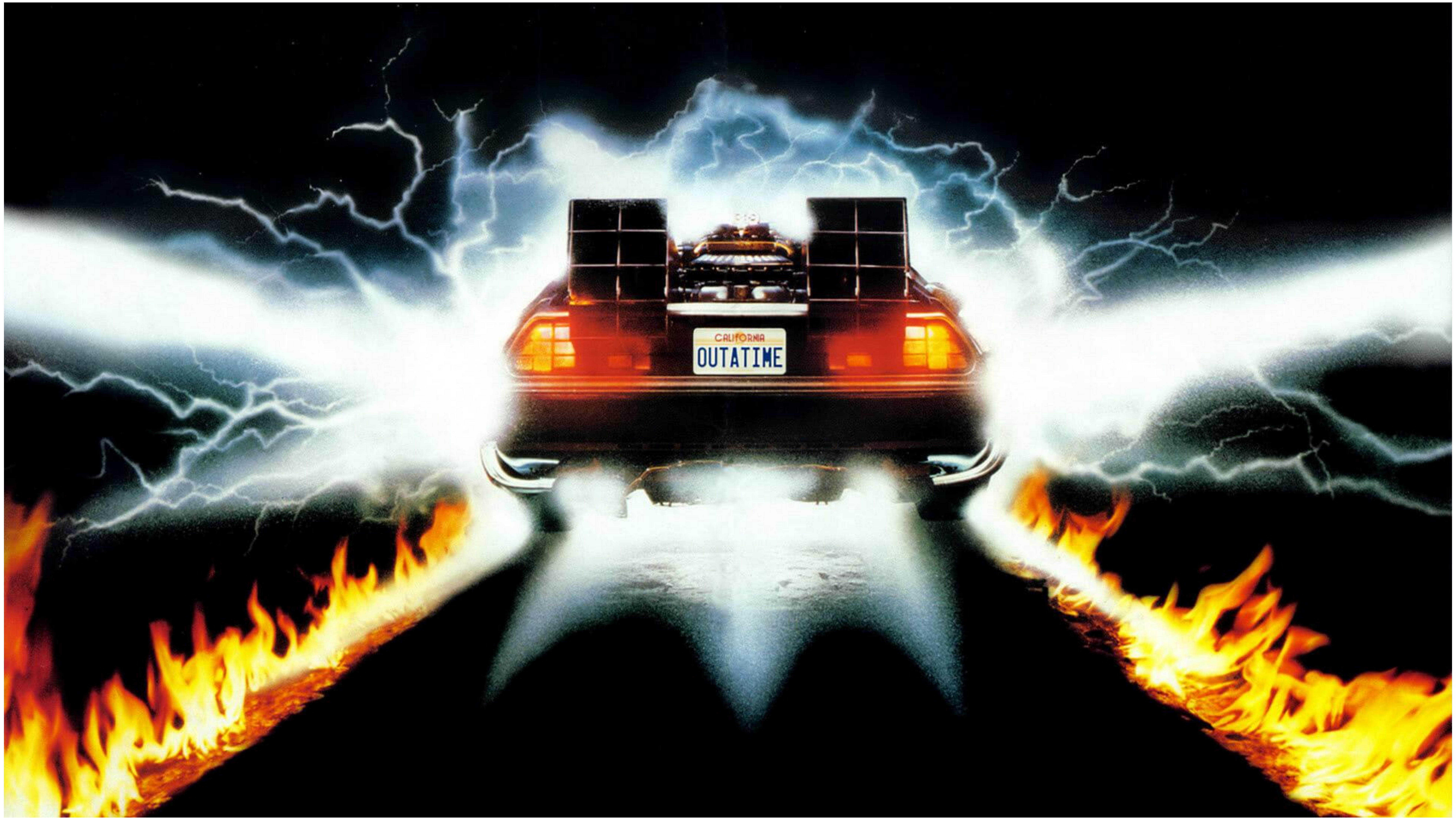


```
public function __construct(
	\Spameri\ElasticQuery\Response\Result\Hit $hit
	, StatsSingle $stats
)
{
	$this->hit = $hit;
	$this->stats = $stats;
}
```

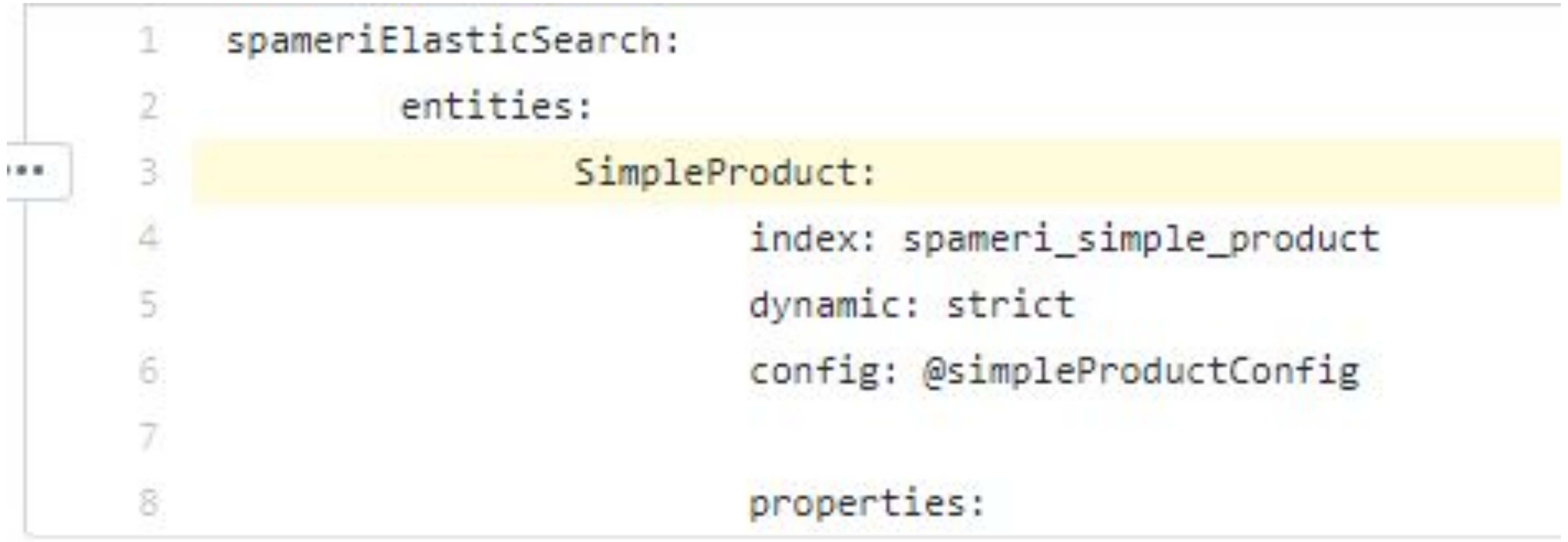
ResultSearch



```
/** @var \Spameri\ElasticQuery\Response\ResultSearch $resultObject */
$resultObject = $resultMapper->map($result);

\Tester\Assert::true($resultObject instanceof \Spameri\ElasticQuery\Response\ResultSearch);

// HIT tests
$hit = $resultObject->getHit('EWhVcWcBhsjlL-GzEP8i');
\Tester\Assert::same('EWhVcWcBhsjlL-GzEP8i', $hit->id());
\Tester\Assert::type('array', $hit->source());
\Tester\Assert::same('spameri_guessed-2018-11-29_21-25-34', $hit->index());
\Tester\Assert::same(0, $hit->position());
\Tester\Assert::same(1.0, $hit->score());
\Tester\Assert::same('spameri_guessed', $hit->type());
\Tester\Assert::same('xyC3UmcBeqWWbOzi4uKU', $hit->getValue('user'));
```

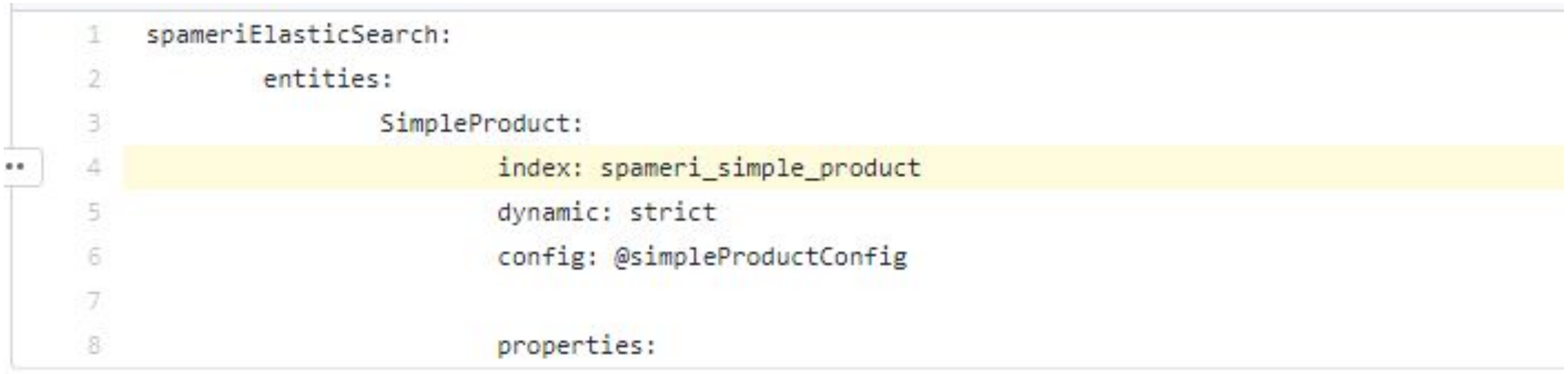

CALIFORNIA
OUTATIME

3. Entity - Config



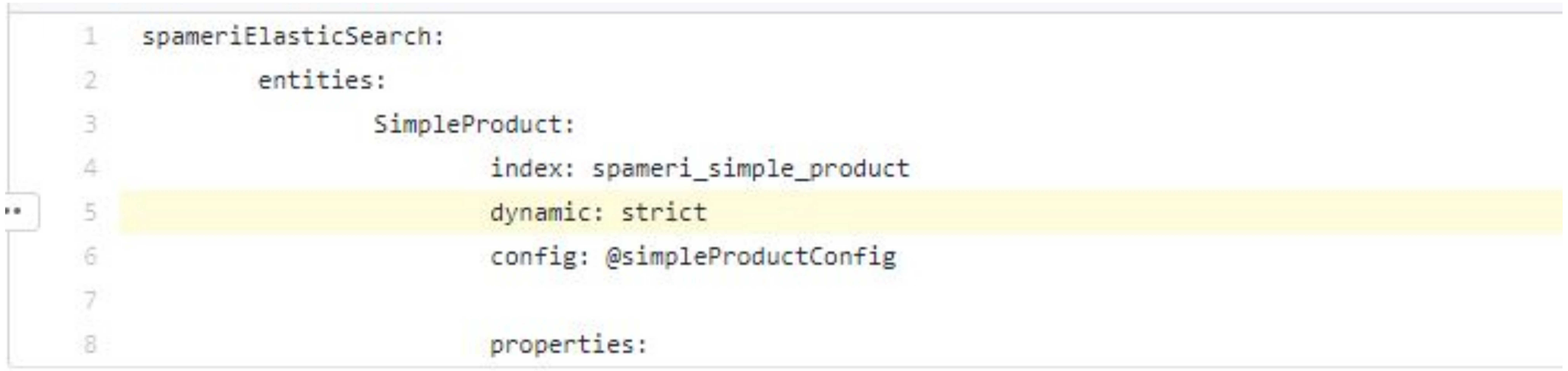
```
spameriElasticSearch:
	entities:
		SimpleProduct:
			index: spameri_simple_product
			dynamic: strict
			config: @simpleProductConfig

			properties:
```

3. Entity - Config



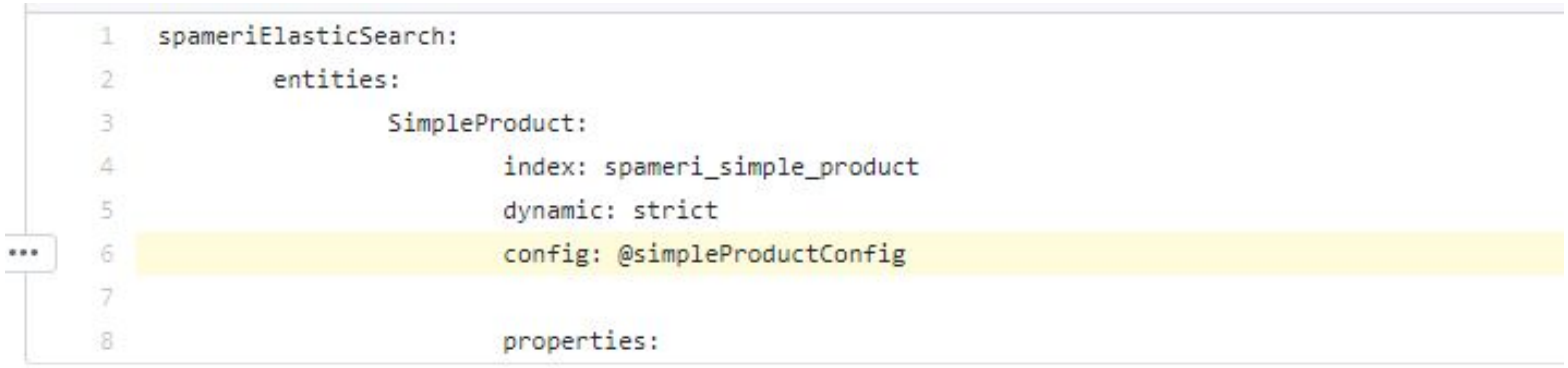
```
spameriElasticSearch:
	entities:
		SimpleProduct:
			index: spameri_simple_product
			dynamic: strict
			config: @simpleProductConfig

			properties:
```

3. Entity - Config



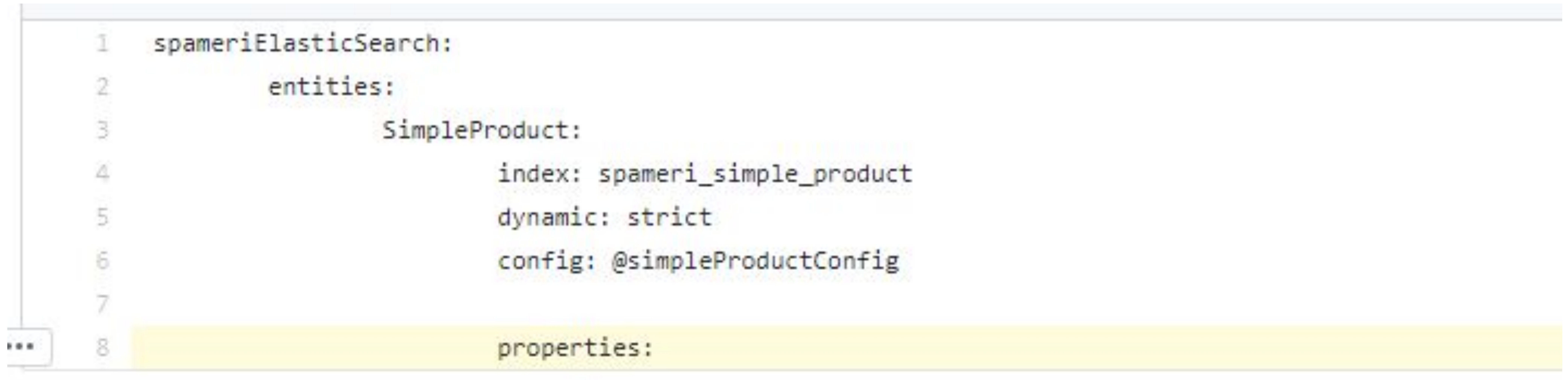
```
spameriElasticSearch:
        entities:
                SimpleProduct:
                        index: spameri_simple_product
                        dynamic: strict
                        config: @simpleProductConfig

                        properties:
```

3. Entity - Config



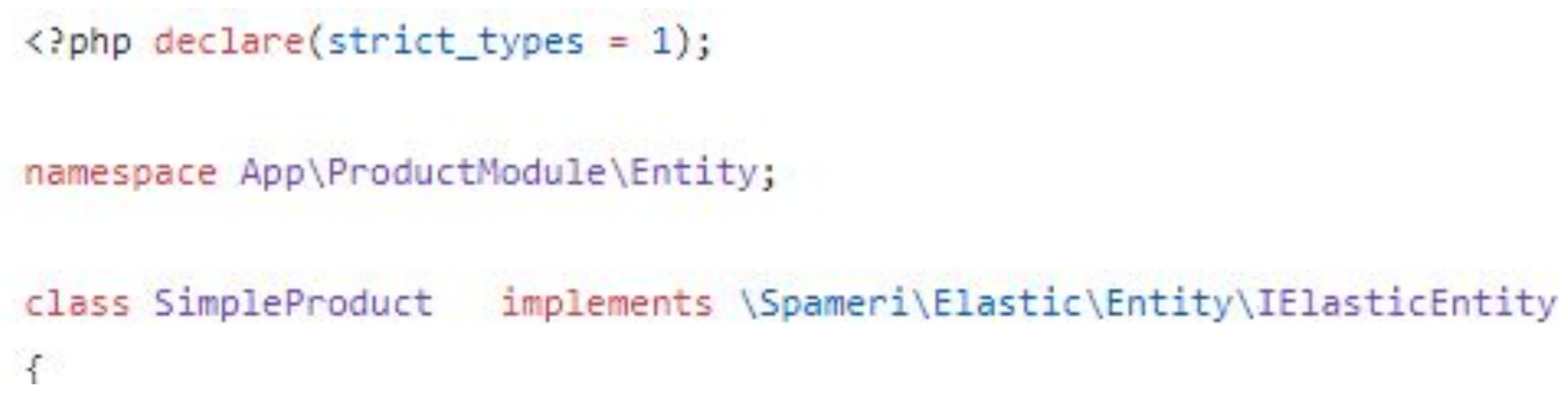
```
spameriElasticSearch:
        entities:
                SimpleProduct:
                        index: spameri_simple_product
                        dynamic: strict
                        config: @simpleProductConfig

                        properties:
```

3. Entity - Config

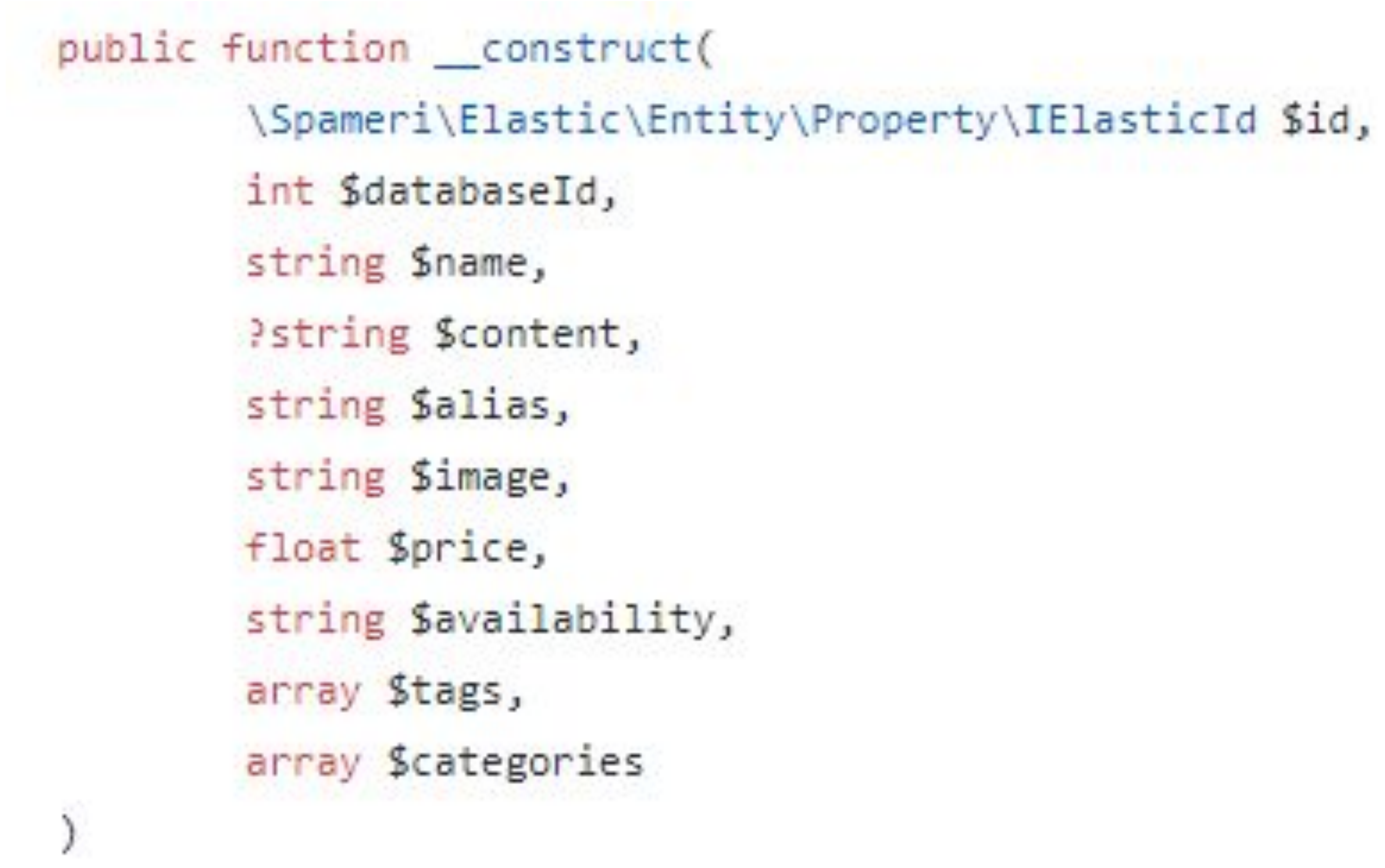


```
spameriElasticSearch:
	entities:
		SimpleProduct:
			index: spameri_simple_product
			dynamic: strict
			config: @simpleProductConfig

			properties:
```

3. Entity - Class


```
<?php declare(strict_types = 1);

namespace App\ProductModule\Entity;

class SimpleProduct   implements \Spameri\Elastic\Entity\IElasticEntity
{
```

3. Entity - Class



```
public function __construct(
        \Spameri\Elastic\Entity\Property\IElasticId $id,
        int $databaseId,
        string $name,
        ?string $content,
        string $alias,
        string $image,
        float $price,
        string $availability,
        array $tags,
        array $categories
)
```

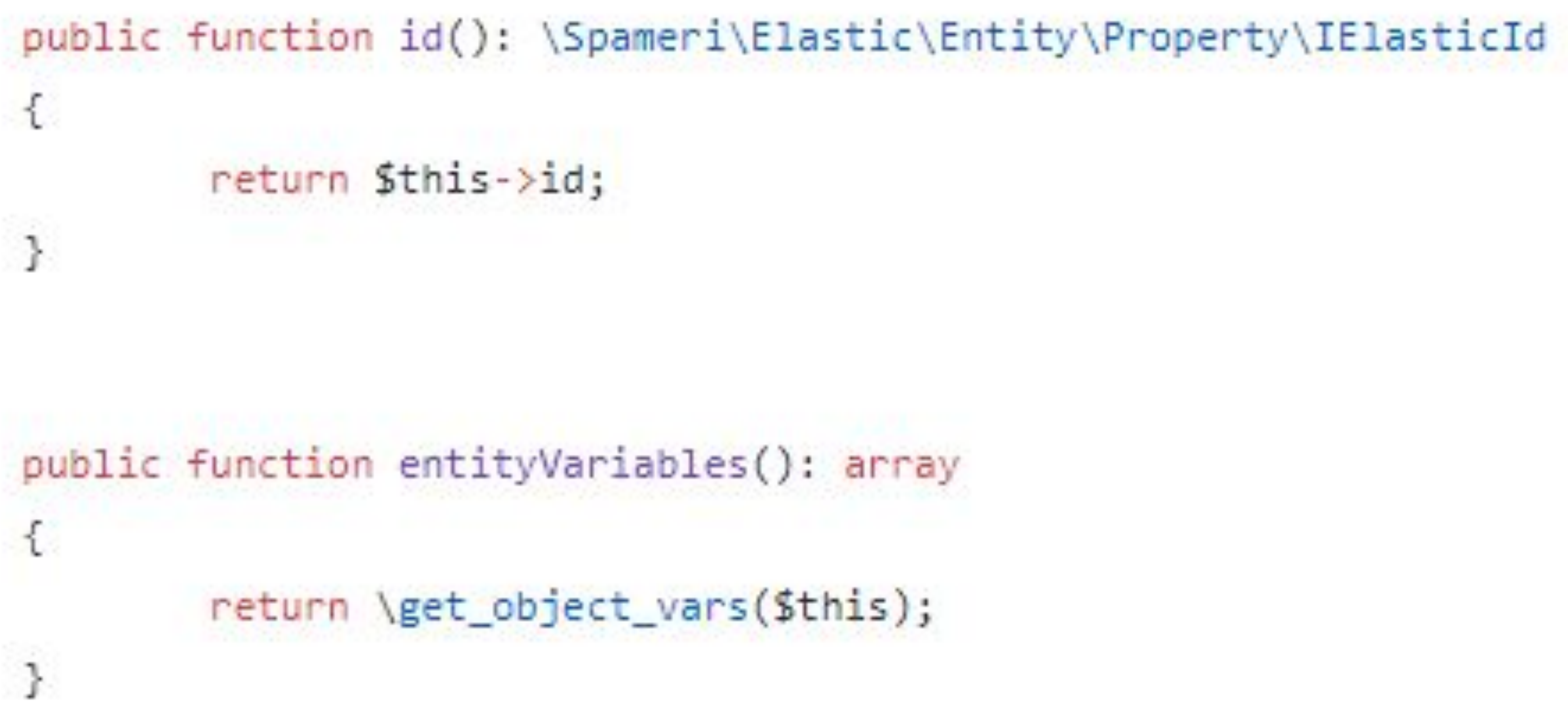
PeckaDesign

Václav Spamer Čevela
Spameri/Elastic

FB facebook.com/peckadesign

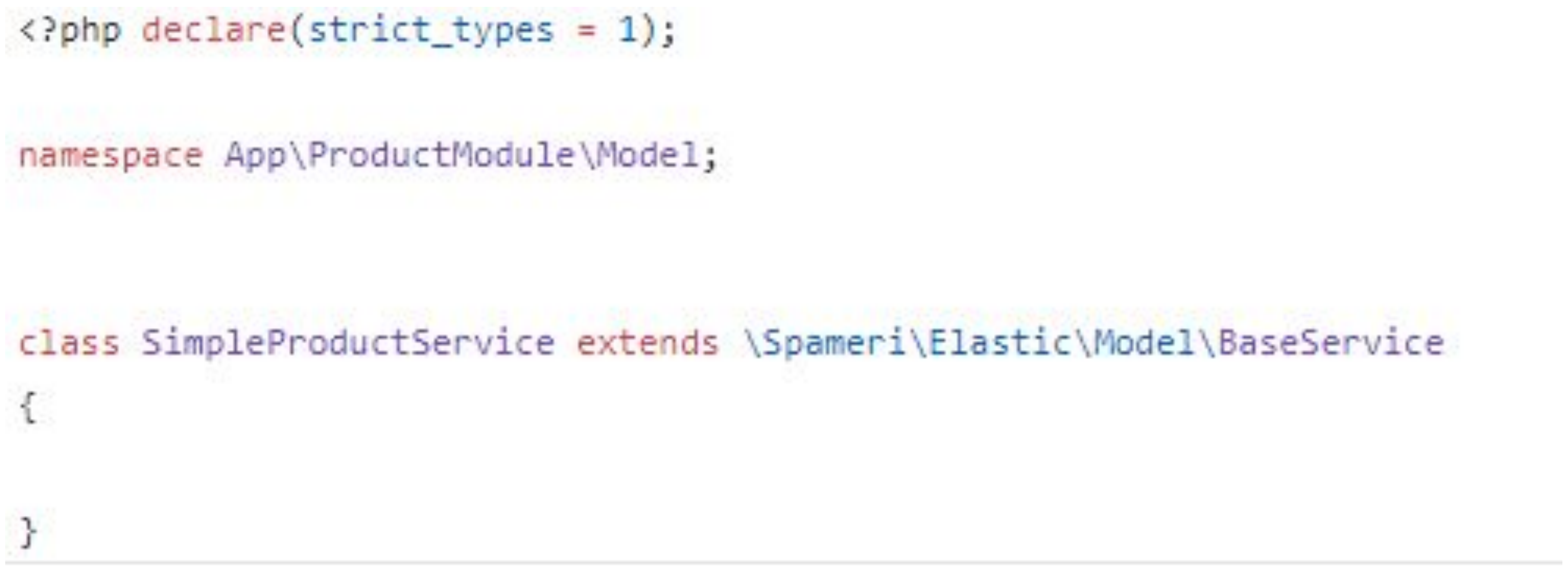
TW @Spamercz

3. Entity - Class



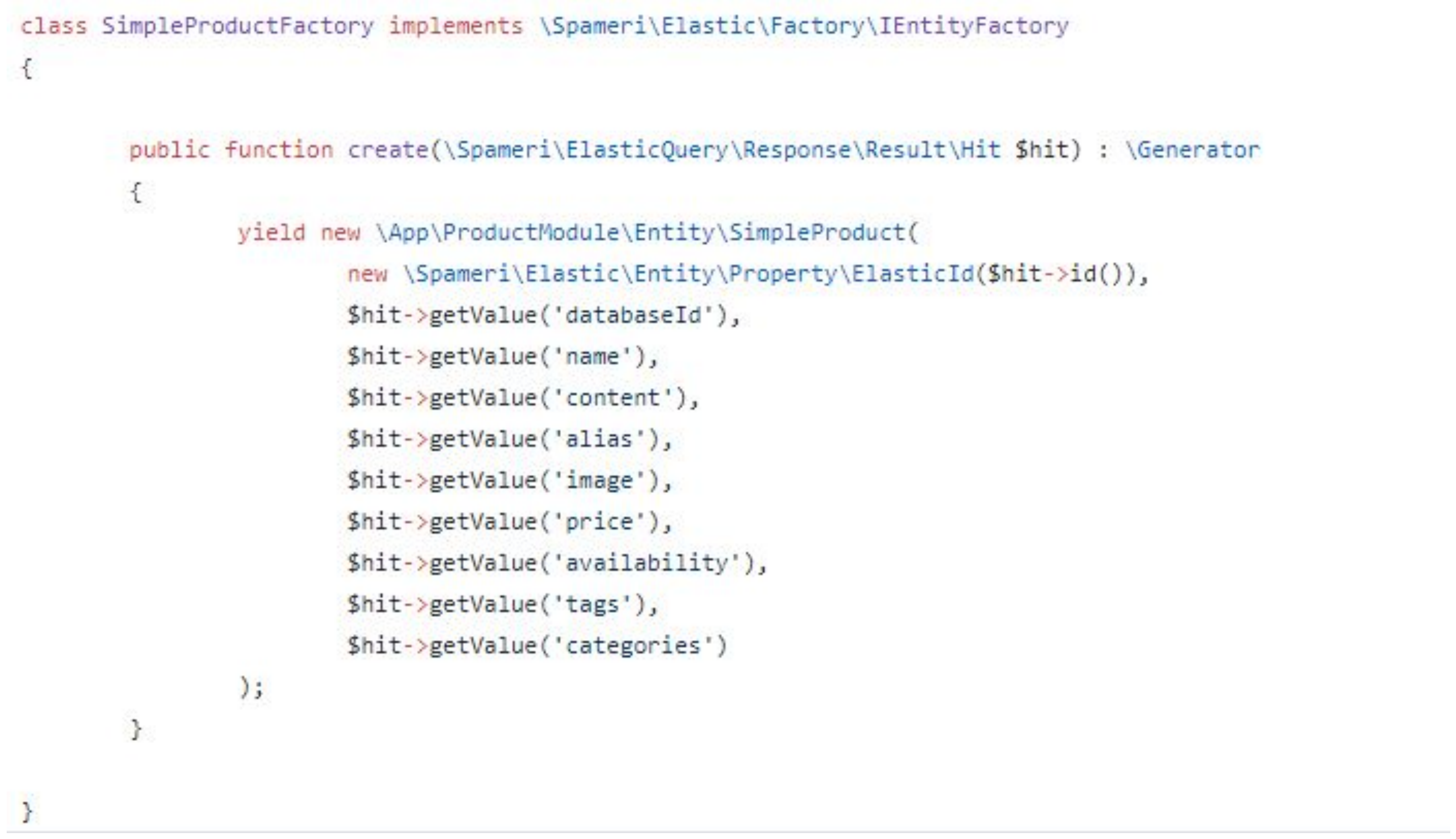
```php
public function id(): \Spameri\Elastic\Entity\Property\IElasticId
{
        return $this->id;
}


public function entityVariables(): array
{
        return \get_object_vars($this);
}
```

3. Entity - Service



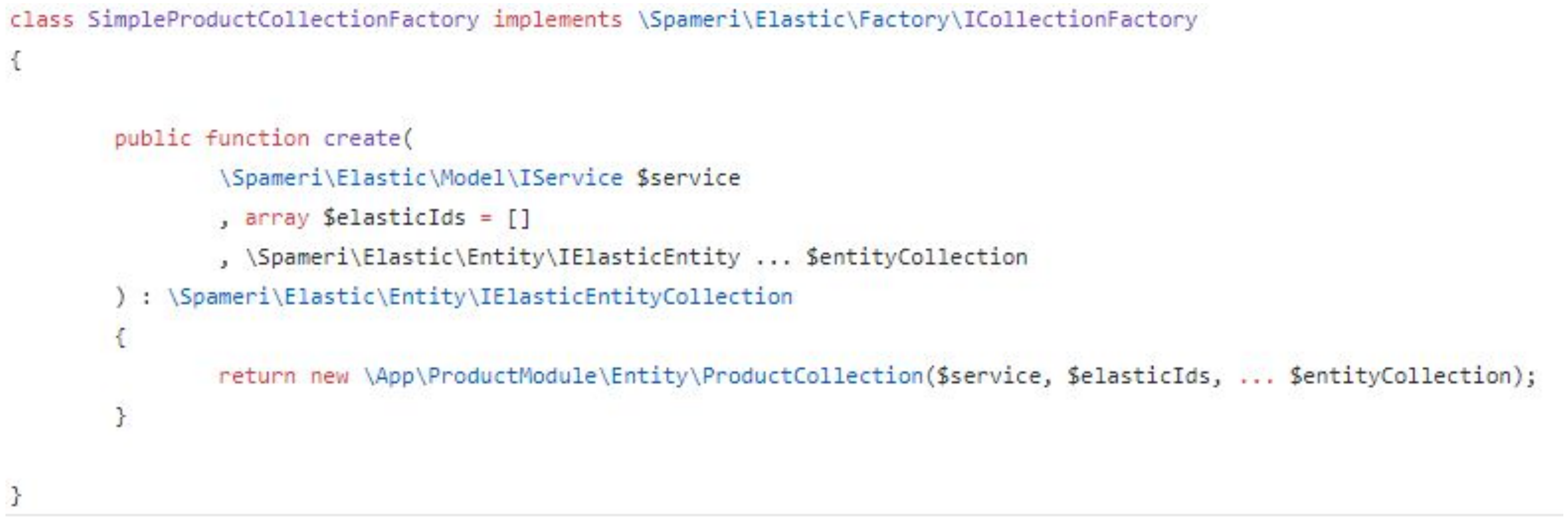
```
<?php declare(strict_types = 1);

namespace App\ProductModule\Model;


class SimpleProductService extends \Spameri\Elastic\Model\BaseService
{

}
```

3. Entity - Factory



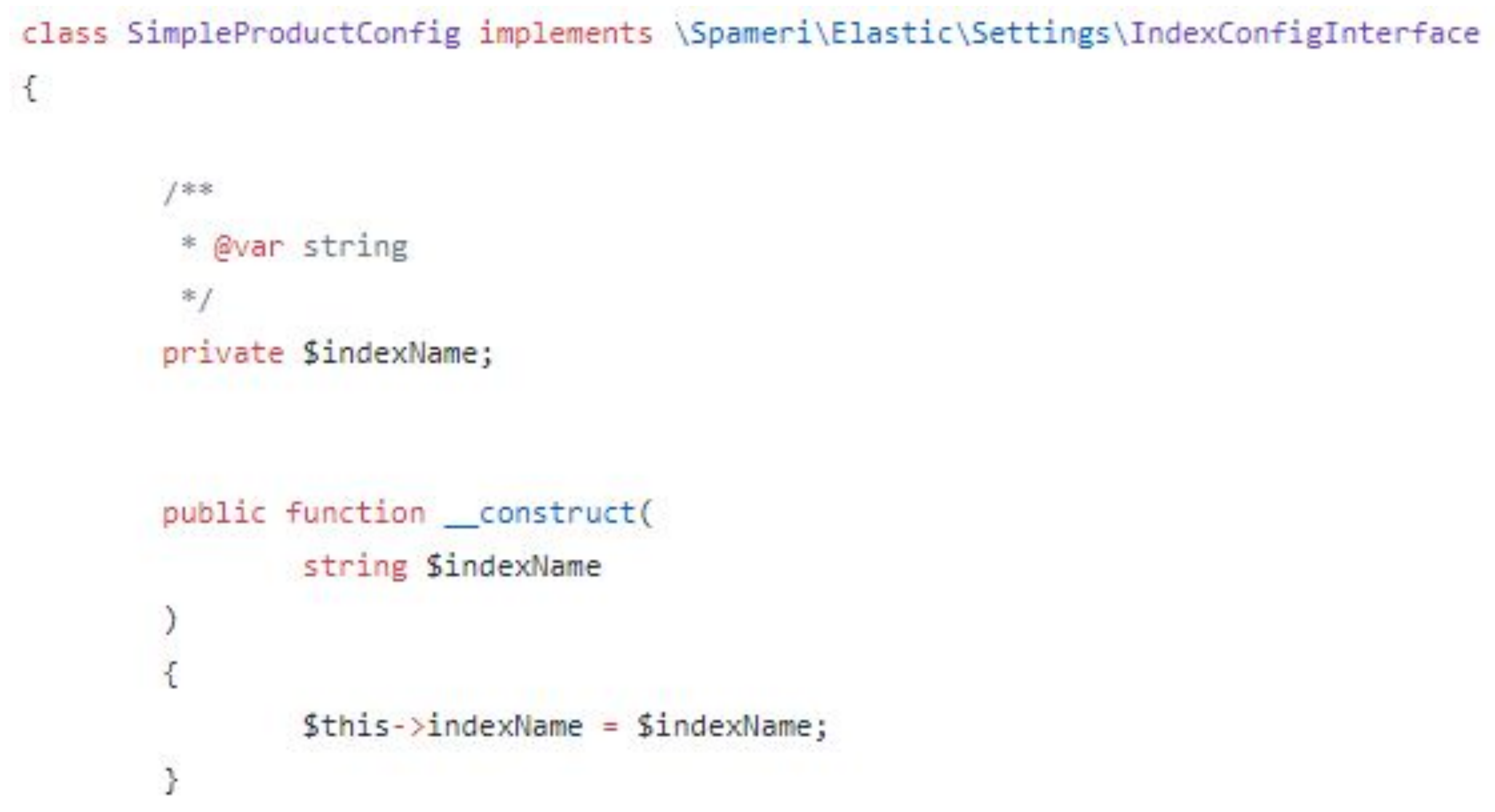
```
class SimpleProductFactory implements \Spameri\Elastic\Factory\IEntityFactory
{

	public function create(\Spameri\ElasticQuery\Response\Result\Hit $hit) : \Generator
	{
		yield new \App\ProductModule\Entity\SimpleProduct(
			new \Spameri\Elastic\Entity\Property\ElasticId($hit->id()),
			$hit->getValue('databaseId'),
			$hit->getValue('name'),
			$hit->getValue('content'),
			$hit->getValue('alias'),
			$hit->getValue('image'),
			$hit->getValue('price'),
			$hit->getValue('availability'),
			$hit->getValue('tags'),
			$hit->getValue('categories')
		);
	}

}
```

3. Entity - CollectionFactory



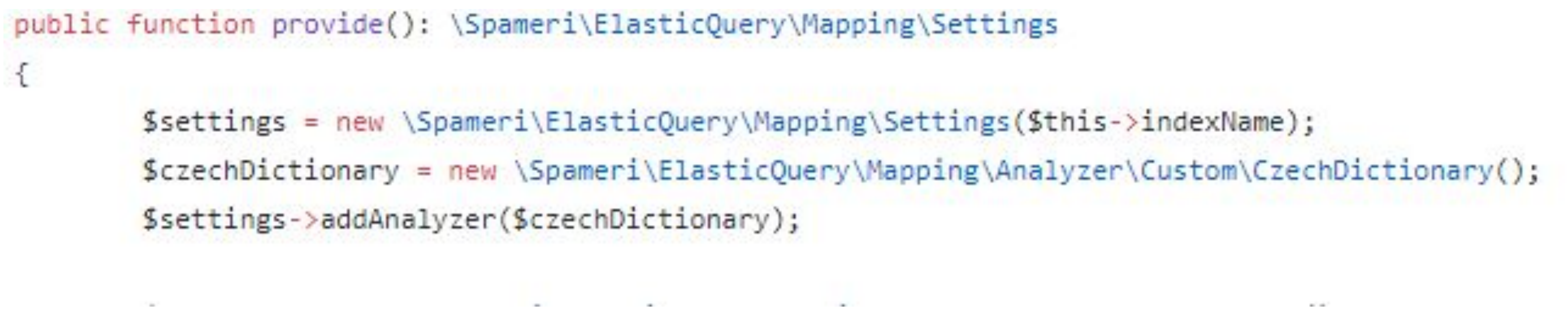
```php
class SimpleProductCollectionFactory implements \Spameri\Elastic\Factory\ICollectionFactory
{

	public function create(
		\Spameri\Elastic\Model\IService $service
		, array $elasticIds = []
		, \Spameri\Elastic\Entity\IElasticEntity ... $entityCollection
	) : \Spameri\Elastic\Entity\IElasticEntityCollection
	{
		return new \App\ProductModule\Entity\ProductCollection($service, $elasticIds, ... $entityCollection);
	}

}
```

4. Mapping



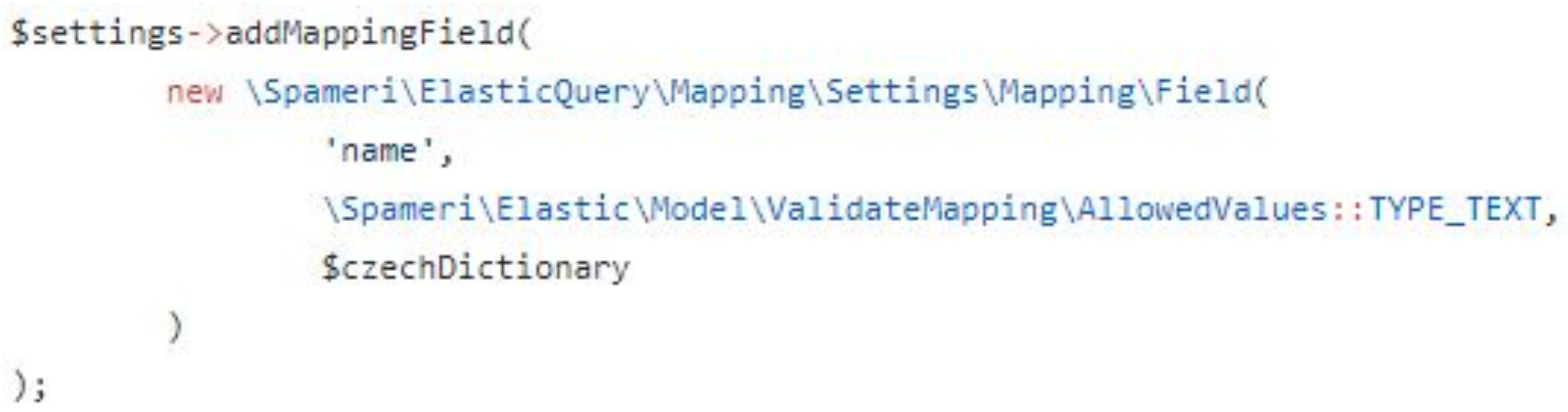
```php
class SimpleProductConfig implements \Spameri\Elastic\Settings\IndexConfigInterface
{

	/**
	 * @var string
	 */
	private $indexName;


	public function __construct(
		string $indexName
	)
	{
		$this->indexName = $indexName;
	}
```

4. Mapping



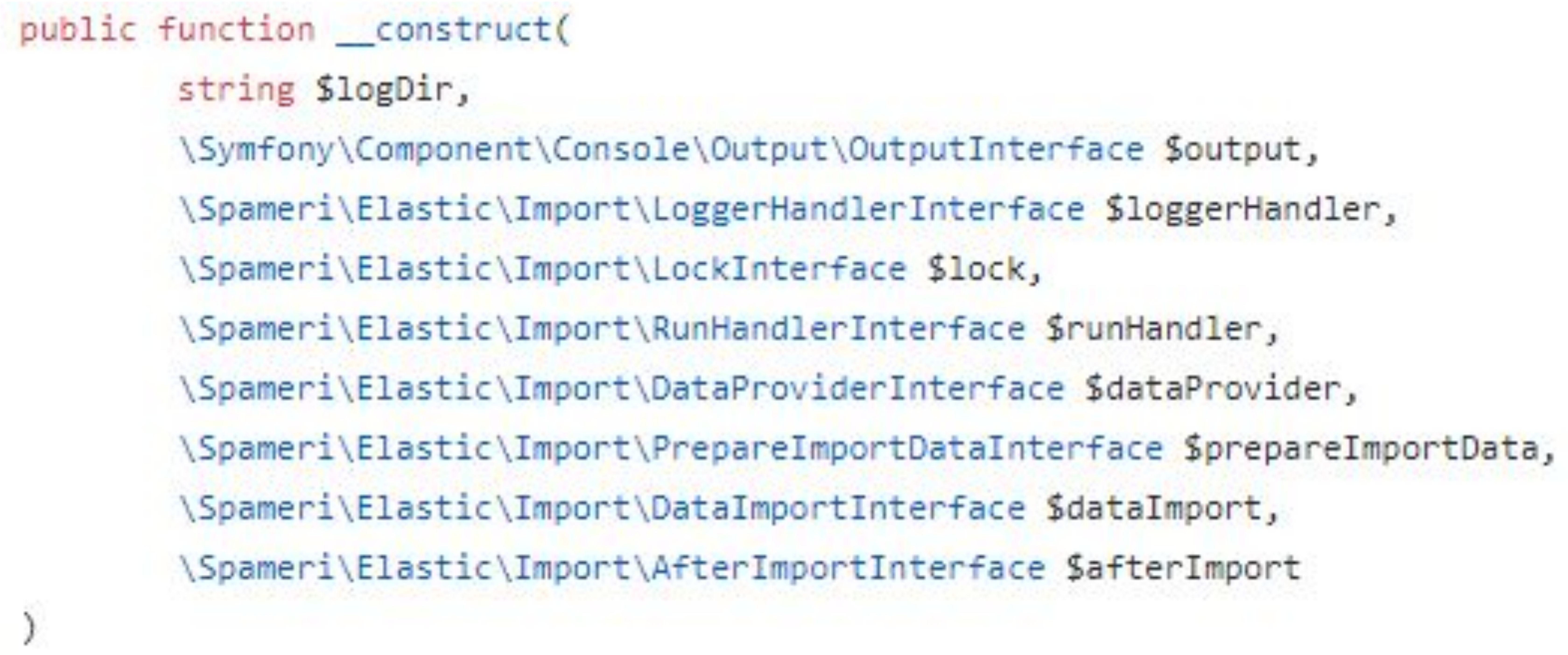
```
public function provide(): \Spameri\ElasticQuery\Mapping\Settings
{
        $settings = new \Spameri\ElasticQuery\Mapping\Settings($this->indexName);
        $czechDictionary = new \Spameri\ElasticQuery\Mapping\Analyzer\Custom\CzechDictionary();
        $settings->addAnalyzer($czechDictionary);
```

4. Mapping



```
$settings->addMappingField(
        new \Spameri\ElasticQuery\Mapping\Settings\Mapping\Field(
                'name',
                \Spameri\Elastic\Model\ValidateMapping\AllowedValues::TYPE_TEXT,
                $czechDictionary
        )
);
```

5. Export - \Spameri\Elastic\Import\Run

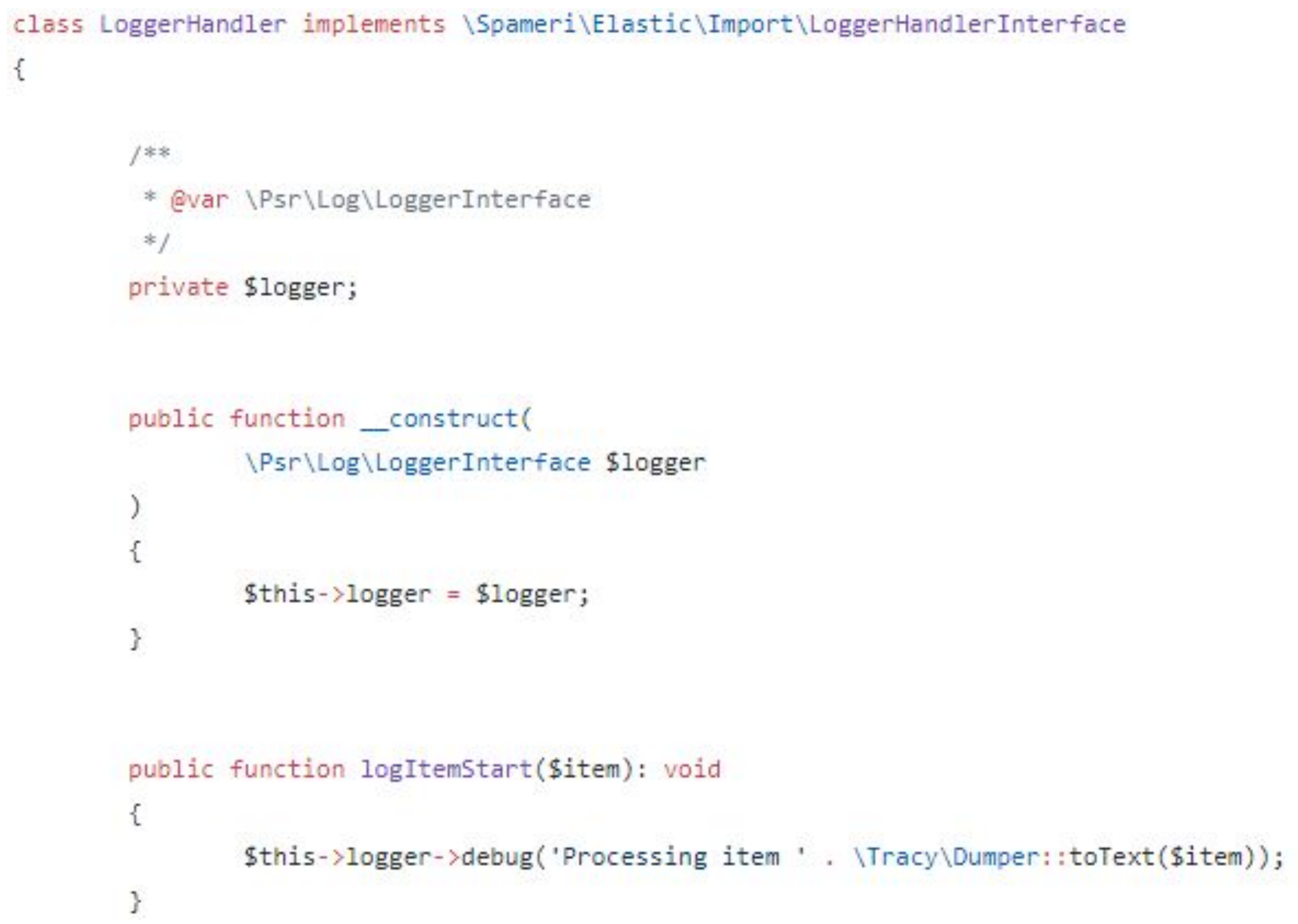


```
public function __construct(
        string $logDir,
        \Symfony\Component\Console\Output\OutputInterface $output,
        \Spameri\Elastic\Import\LoggerHandlerInterface $loggerHandler,
        \Spameri\Elastic\Import\LockInterface $lock,
        \Spameri\Elastic\Import\RunHandlerInterface $runHandler,
        \Spameri\Elastic\Import\DataProviderInterface $dataProvider,
        \Spameri\Elastic\Import\PrepareImportDataInterface $prepareImportData,
        \Spameri\Elastic\Import\DataImportInterface $dataImport,
        \Spameri\Elastic\Import\AfterImportInterface $afterImport
)
```

5. Export - Output

- Symfony console output
- NullOutput
- ConsoleOutput

5. Export - LoggerHandler



```php
class LoggerHandler implements \Spameri\Elastic\Import\LoggerHandlerInterface
{

    /**
     * @var \Psr\Log\LoggerInterface
     */
    private $logger;


    public function __construct(
        \Psr\Log\LoggerInterface $logger
    )
    {
        $this->logger = $logger;
    }


    public function logItemStart($item): void
    {
        $this->logger->debug('Processing item ' . \Tracy\Dumper::toText($item));
    }
```

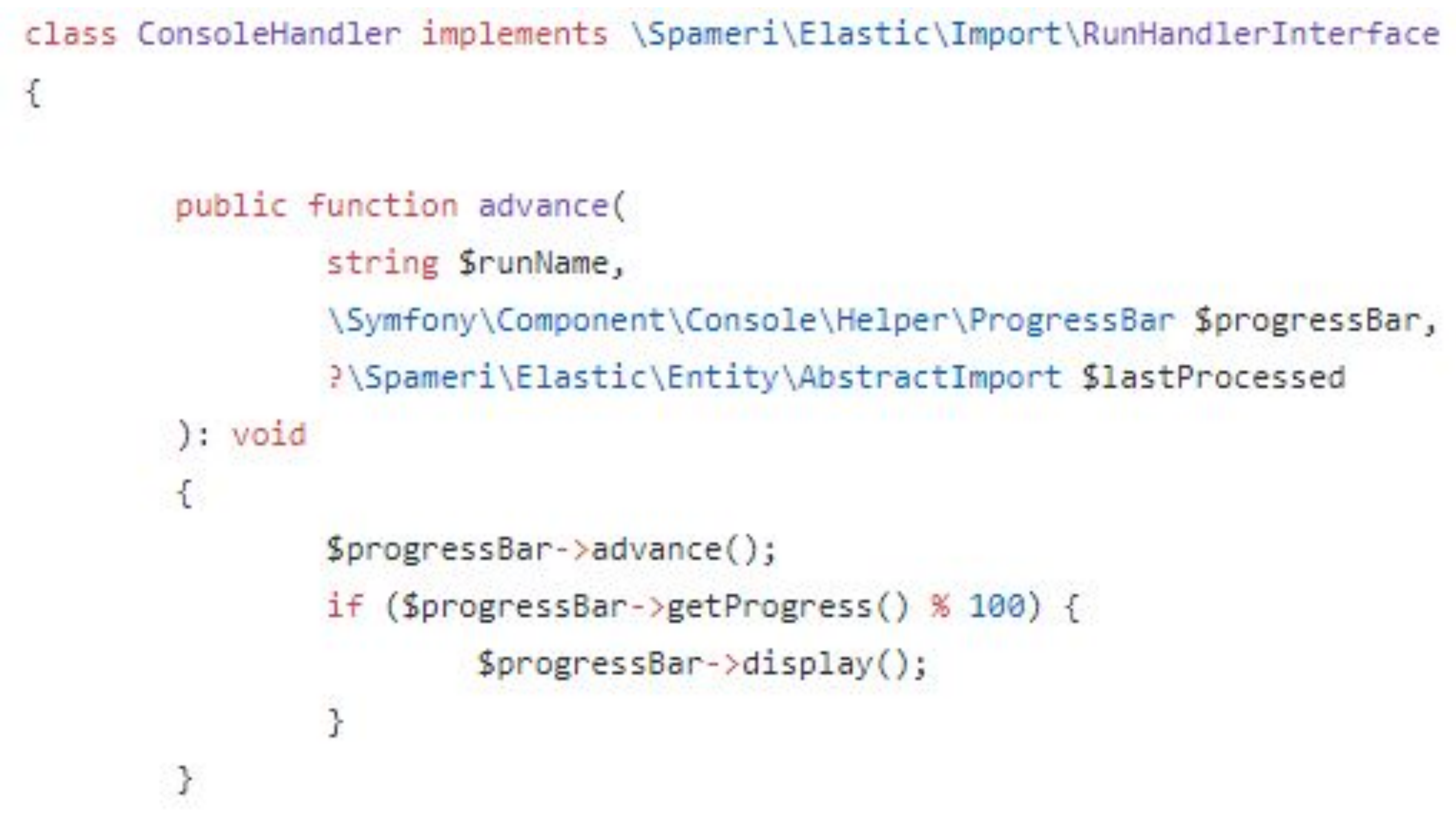
5. Export - LoggerHandler

- Item start
- Item prepared
- Response
- Omit Exception
- Error Exception
- Fatal Exception
- Finish

5. Export - Lock

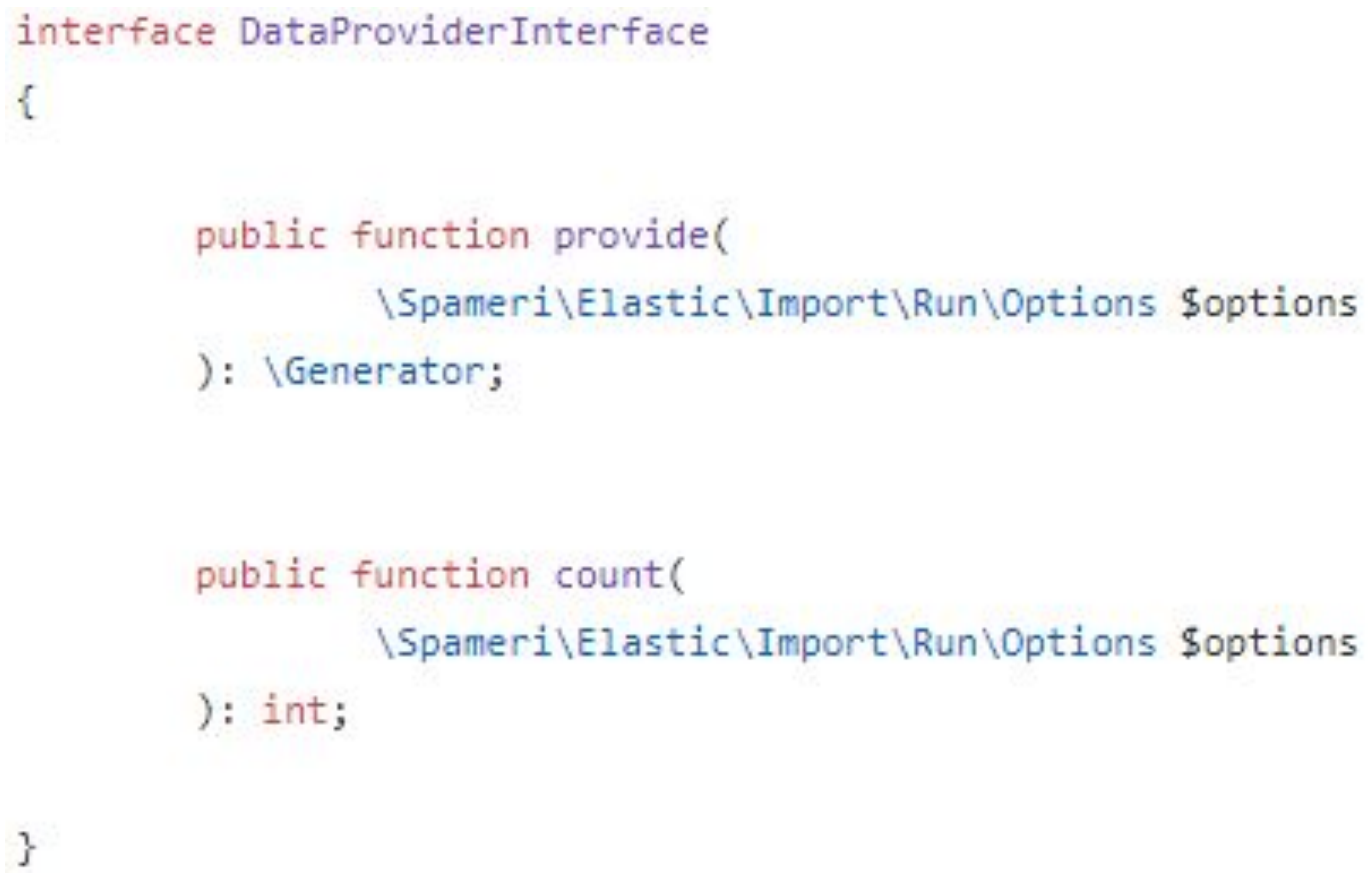
- Ensures one run at time
- If export takes too long it can extend itself
- If crashed can recover after time element expires
- NullLock variant

5. Export - RunHandler



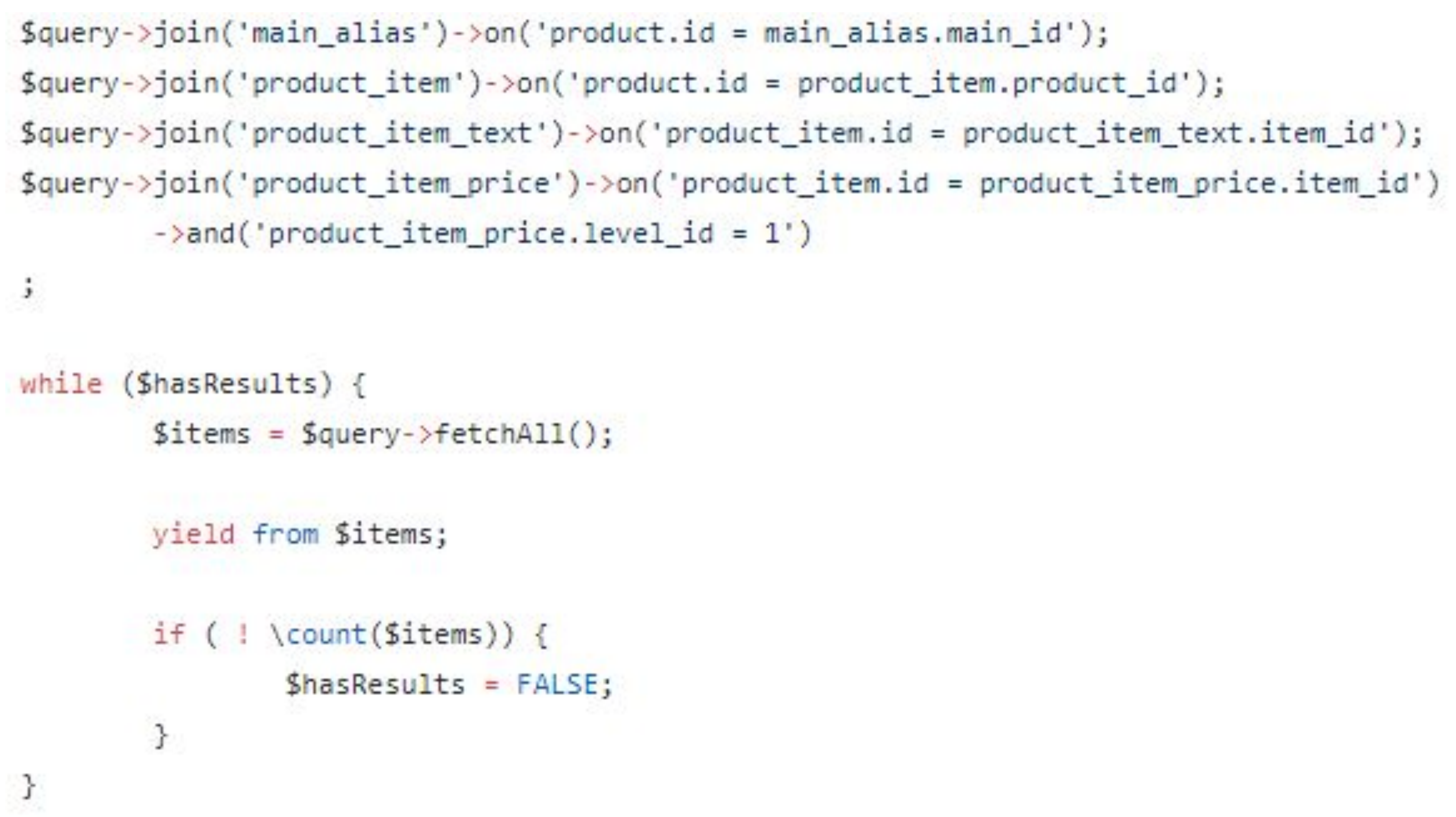
```php
class ConsoleHandler implements \Spameri\Elastic\Import\RunHandlerInterface
{

    public function advance(
        string $runName,
        \Symfony\Component\Console\Helper\ProgressBar $progressBar,
        ?\Spameri\Elastic\Entity\AbstractImport $lastProcessed
    ): void
    {
        $progressBar->advance();
        if ($progressBar->getProgress() % 100) {
            $progressBar->display();
        }
    }
```

5. Export - DataProvider



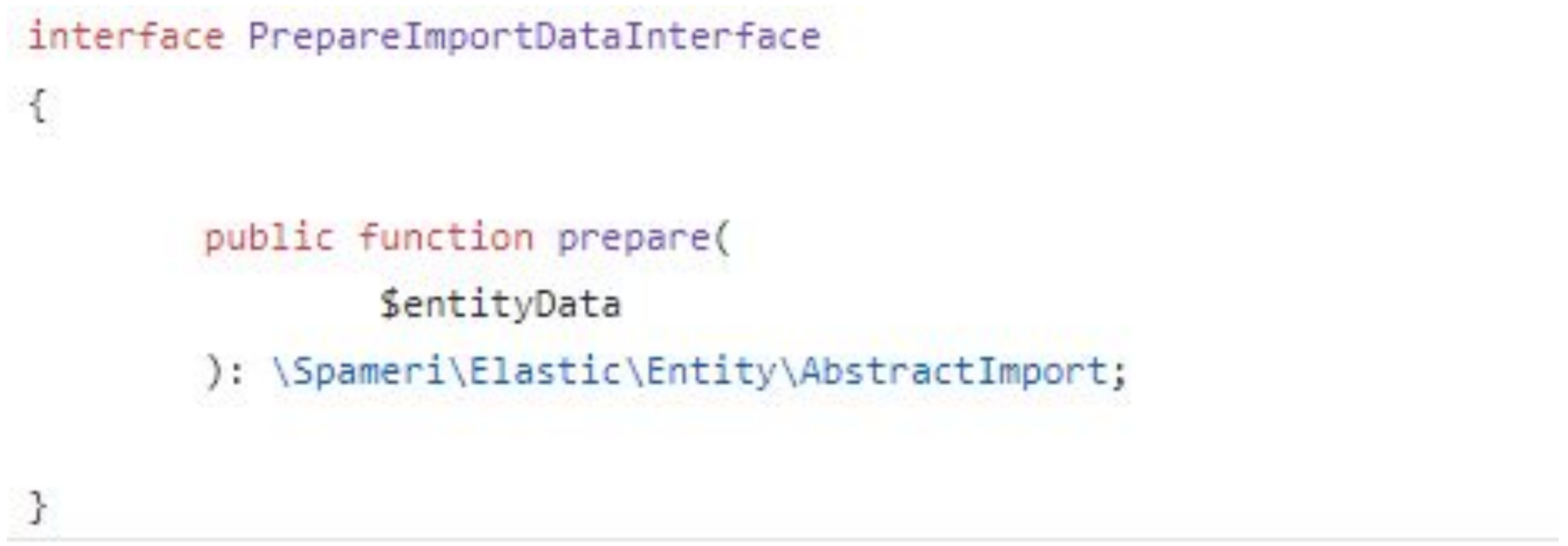
```
interface DataProviderInterface
{

	public function provide(
		\Spameri\Elastic\Import\Run\Options $options
	): \Generator;


	public function count(
		\Spameri\Elastic\Import\Run\Options $options
	): int;

}
```

5. Export - DataProvider



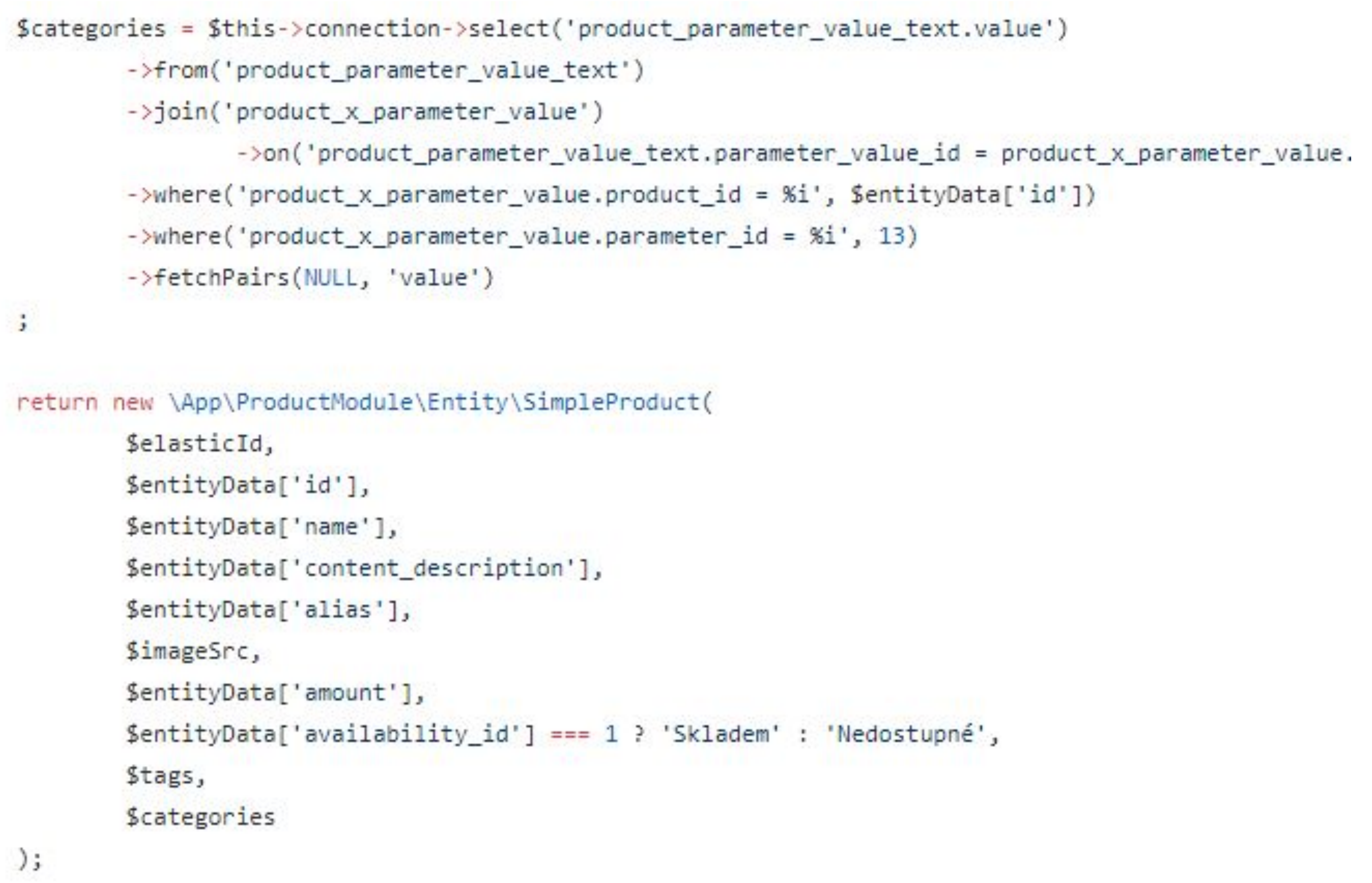
```php
$query->join('main_alias')->on('product.id = main_alias.main_id');
$query->join('product_item')->on('product.id = product_item.product_id');
$query->join('product_item_text')->on('product_item.id = product_item_text.item_id');
$query->join('product_item_price')->on('product_item.id = product_item_price.item_id')
        ->and('product_item_price.level_id = 1')
;

while ($hasResults) {
        $items = $query->fetchAll();

        yield from $items;

        if ( ! \count($items)) {
                $hasResults = FALSE;
        }
}
```

5. Export - PrepareData



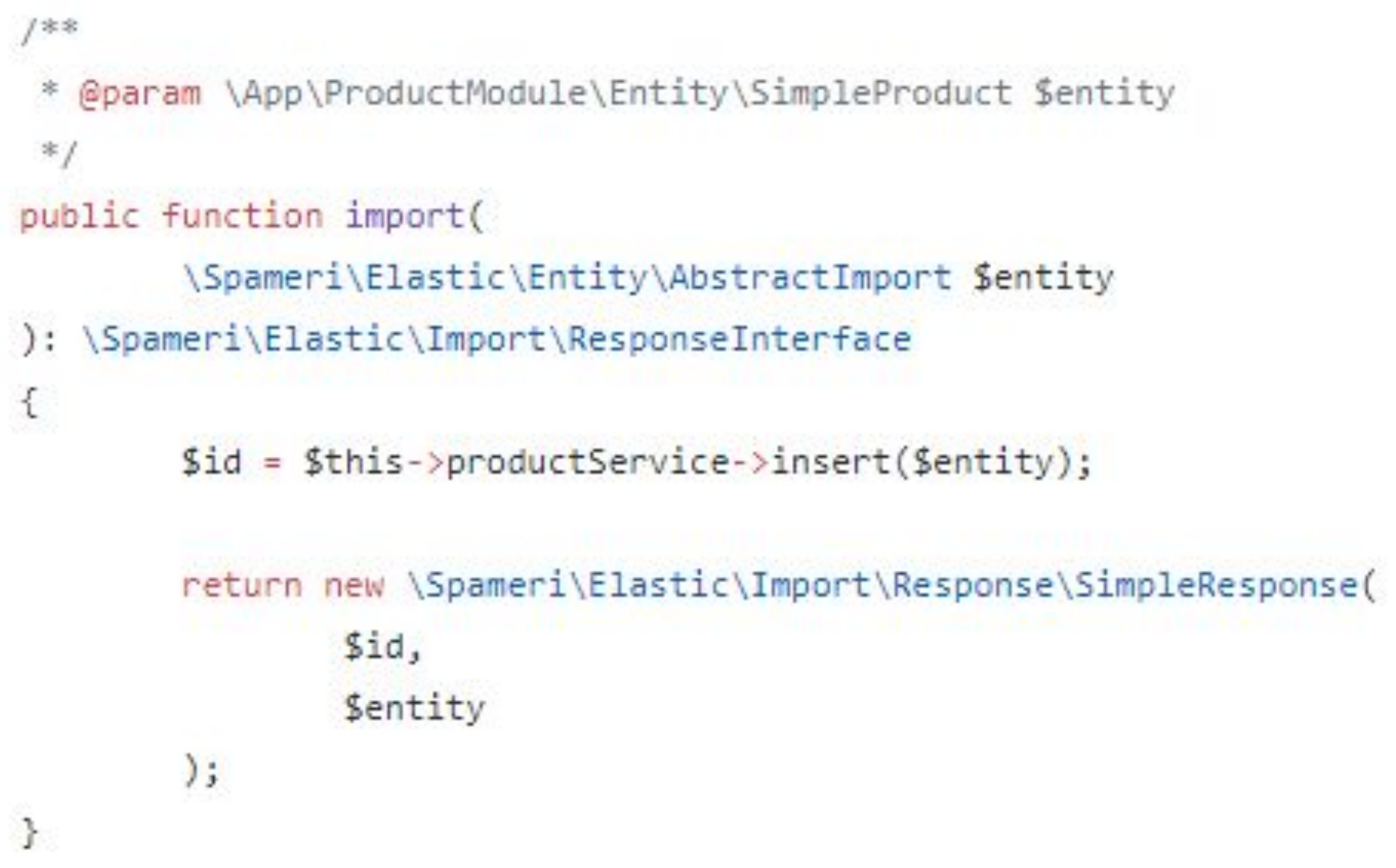
```
interface PrepareImportDataInterface
{

	public function prepare(
		$entityData
	): \Spameri\Elastic\Entity\AbstractImport;

}
```

5. Export - PrepareData



```
$categories = $this->connection->select('product_parameter_value_text.value')
        ->from('product_parameter_value_text')
        ->join('product_x_parameter_value')
                ->on('product_parameter_value_text.parameter_value_id = product_x_parameter_value.
        ->where('product_x_parameter_value.product_id = %i', $entityData['id'])
        ->where('product_x_parameter_value.parameter_id = %i', 13)
        ->fetchPairs(NULL, 'value')
;

return new \App\ProductModule\Entity\SimpleProduct(
        $elasticId,
        $entityData['id'],
        $entityData['name'],
        $entityData['content_description'],
        $entityData['alias'],
        $imageSrc,
        $entityData['amount'],
        $entityData['availability_id'] === 1 ? 'Skladem' : 'Nedostupné',
        $tags,
        $categories
);
```

5. Export - Import

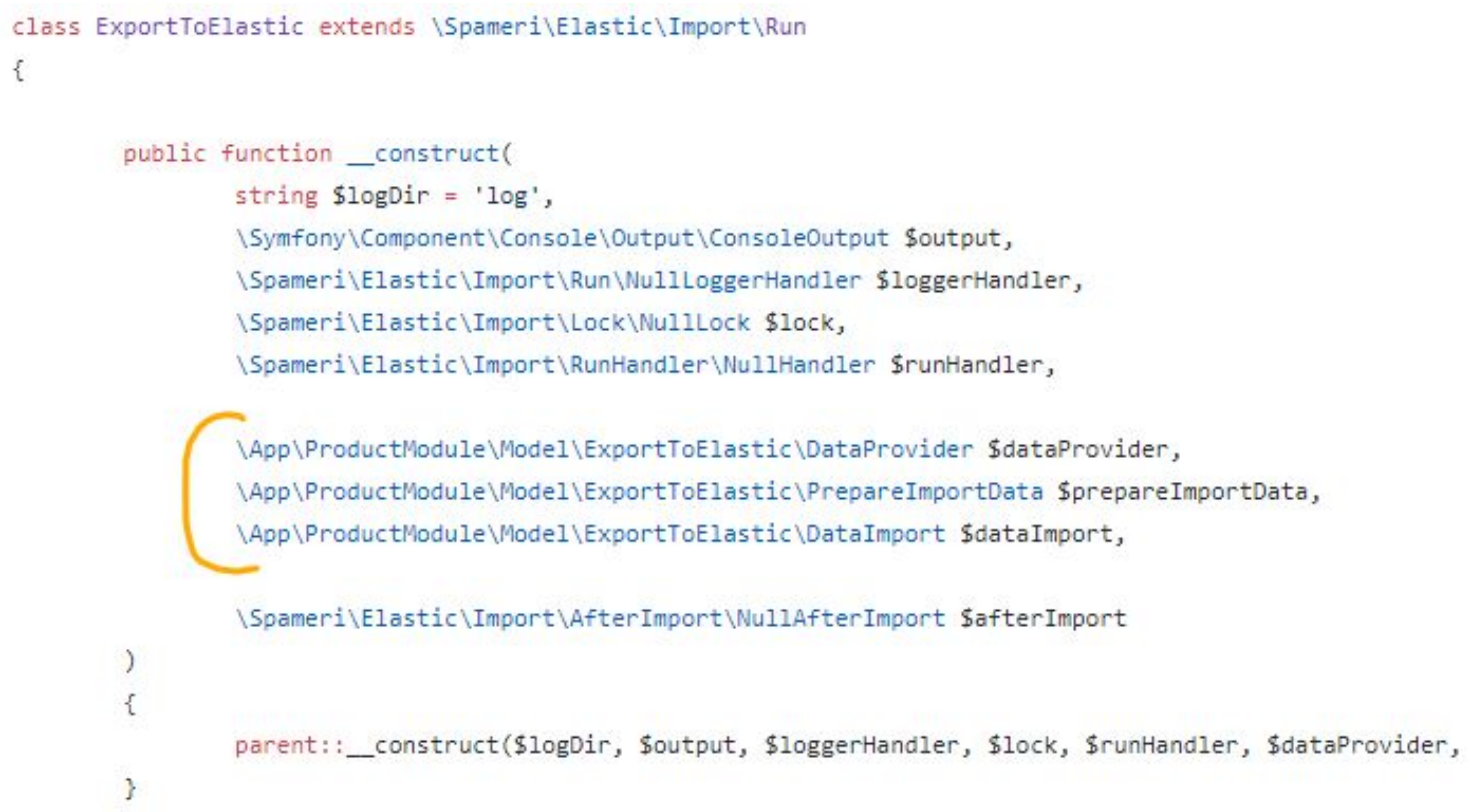


```php
/**
 * @param \App\ProductModule\Entity\SimpleProduct $entity
 */
public function import(
	\Spameri\Elastic\Entity\AbstractImport $entity
): \Spameri\Elastic\Import\ResponseInterface
{
	$id = $this->productService->insert($entity);

	return new \Spameri\Elastic\Import\Response\SimpleResponse(
		$id,
		$entity
	);
}
```

5. Export - AfterImport

- When you need something done after every item
- Cache cleanup
- Identity map purge
- unlink files
- etc

5. Export - ExportToElasticRun

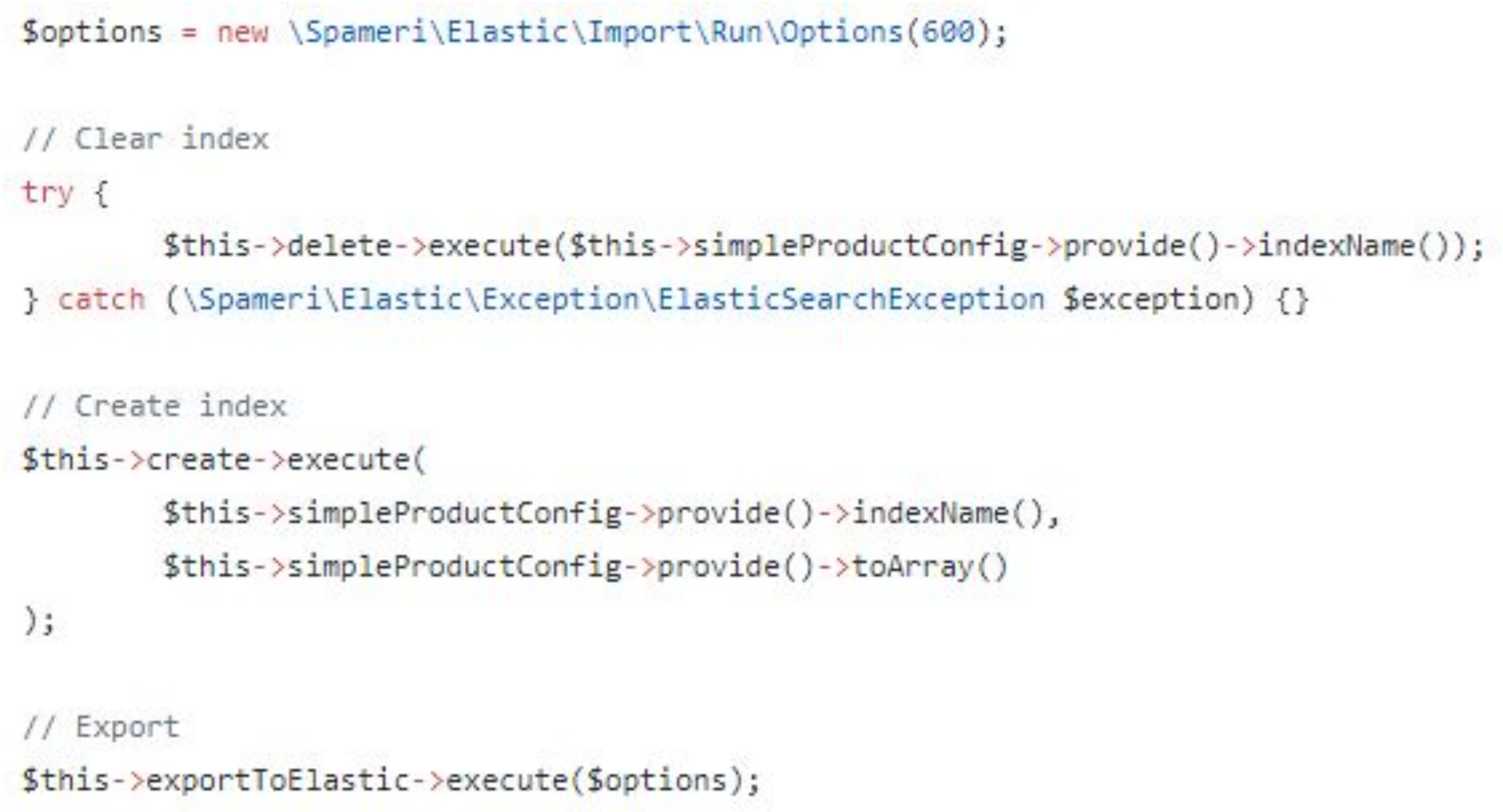


```
class ExportToElastic extends \Spameri\Elastic\Import\Run
{

	public function __construct(
		string $logDir = 'log',
		\Symfony\Component\Console\Output\ConsoleOutput $output,
		\Spameri\Elastic\Import\Run\NullLoggerHandler $loggerHandler,
		\Spameri\Elastic\Import\Lock\NullLock $lock,
		\Spameri\Elastic\Import\RunHandler\NullHandler $runHandler,

		\App\ProductModule\Model\ExportToElastic\DataProvider $dataProvider,
		\App\ProductModule\Model\ExportToElastic\PrepareImportData $prepareImportData,
		\App\ProductModule\Model\ExportToElastic\DataImport $dataImport,

		\Spameri\Elastic\Import\AfterImport\NullAfterImport $afterImport
	)
	{
		parent::__construct($logDir, $output, $loggerHandler, $lock, $runHandler, $dataProvider,
	}
```

5. Export - Command

- Delete old index
- Create new index
- Put settings
- Put mappings
- Put data

5. Export - Command



```
$options = new \Spameri\Elastic\Import\Run\Options(600);

// Clear index
try {
	$this->delete->execute($this->simpleProductConfig->provide()->indexName());
} catch (\Spameri\Elastic\Exception\ElasticSearchException $exception) {}

// Create index
$this->create->execute(
	$this->simpleProductConfig->provide()->indexName(),
	$this->simpleProductConfig->provide()->toArray()
);

// Export
$this->exportToElastic->execute($options);
```

PeckaDesign

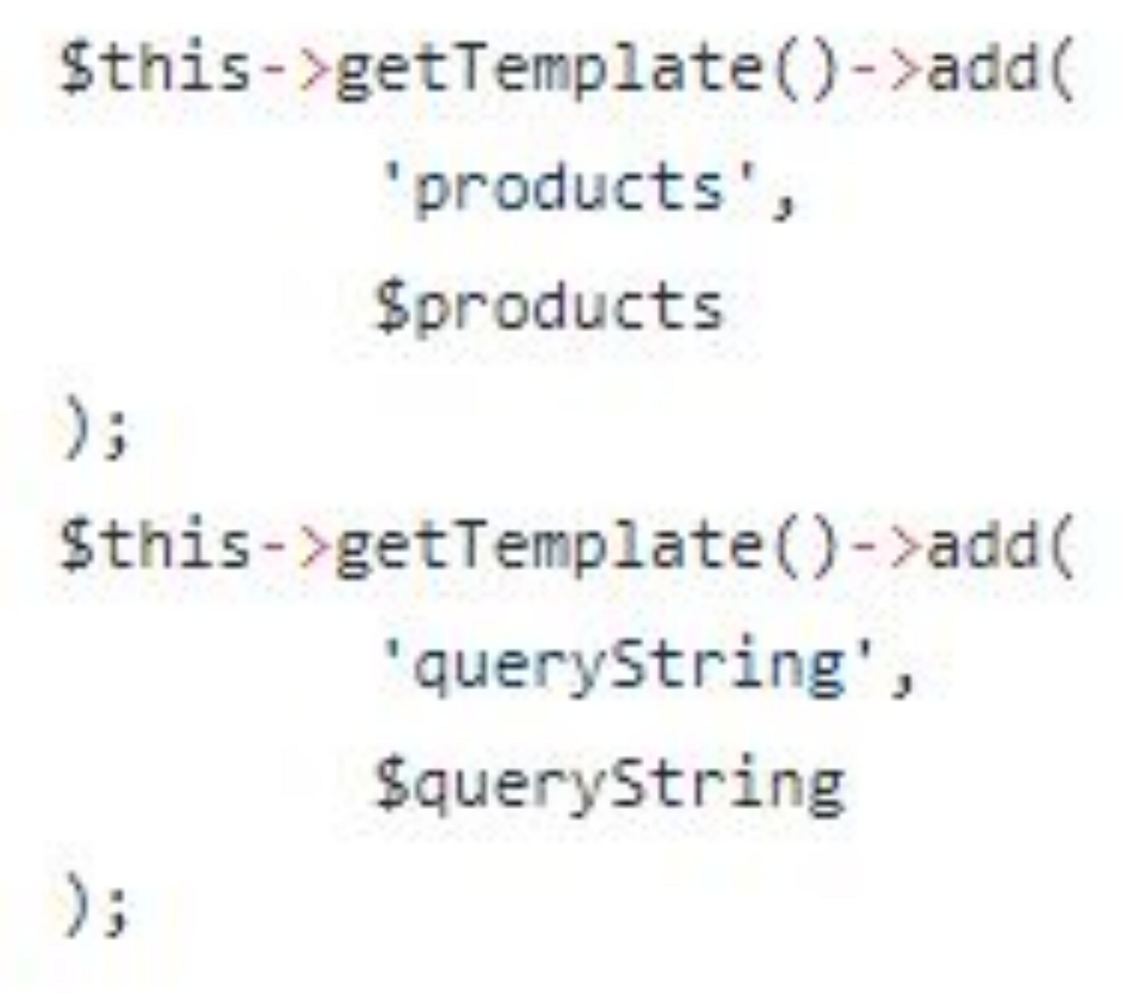
Václav Spamer Čevela
Spameri/Elastic

FB facebook.com/peckadesign

TW @Spamercz

6. Frontend - Presenter

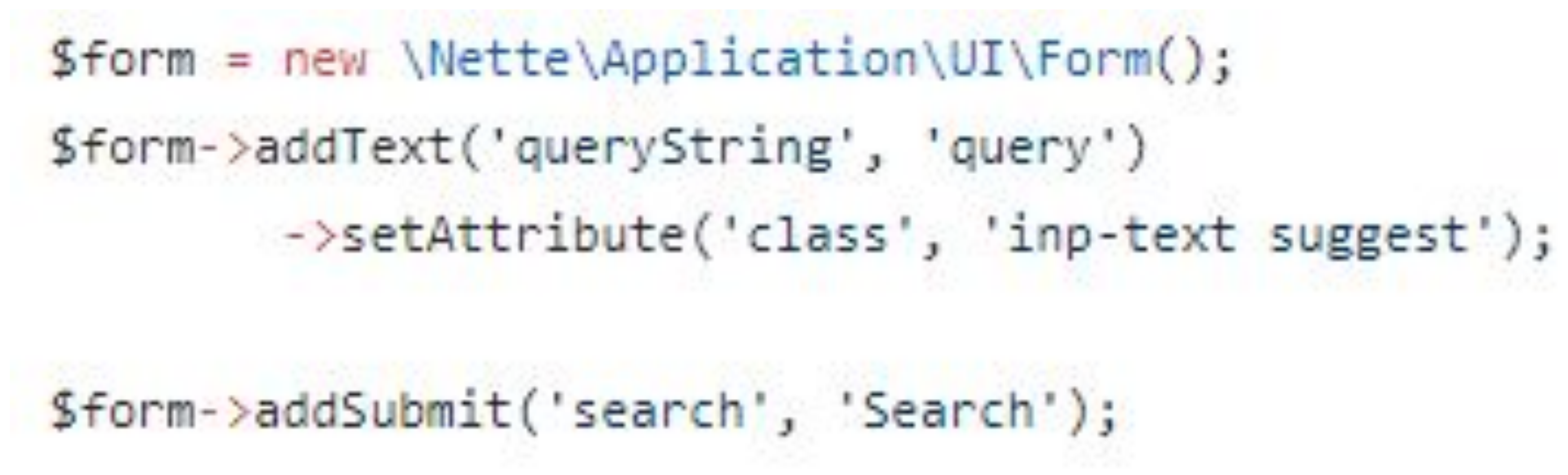
SimpleProductListPresenter::renderDefault($queryString)



```
$this->getTemplate()->add(
        'products',
        $products
);
$this->getTemplate()->add(
        'queryString',
        $queryString
);
```

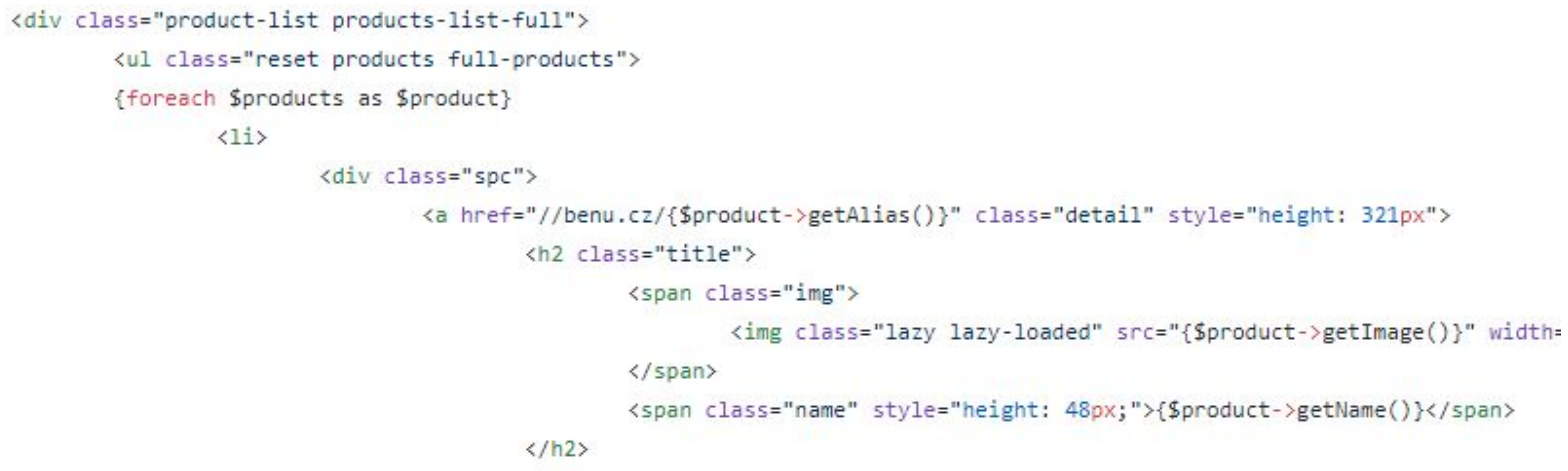
6. Frontend - Form

CreateComponentSearchForm



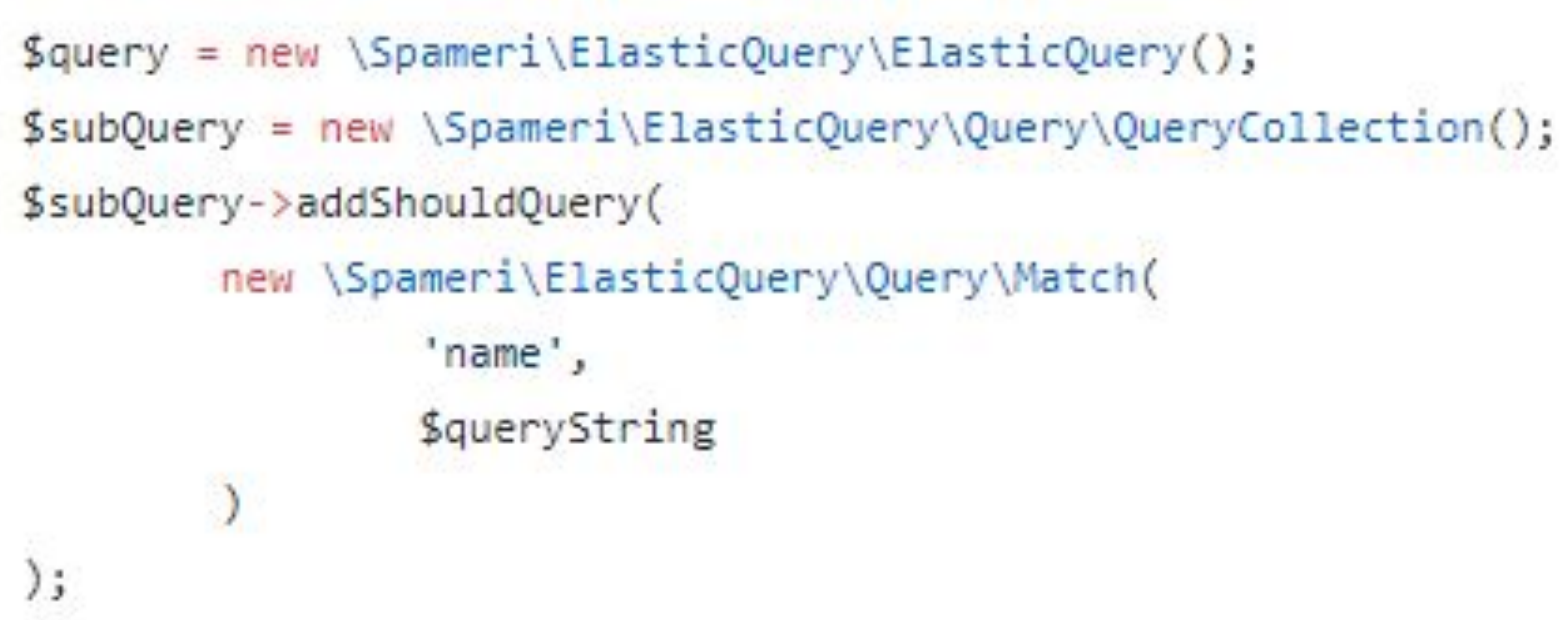
```
$form = new \Nette\Application\UI\Form();
$form->addText('queryString', 'query')
        ->setAttribute('class', 'inp-text suggest');

$form->addSubmit('search', 'Search');
```

6. Frontend - Latte



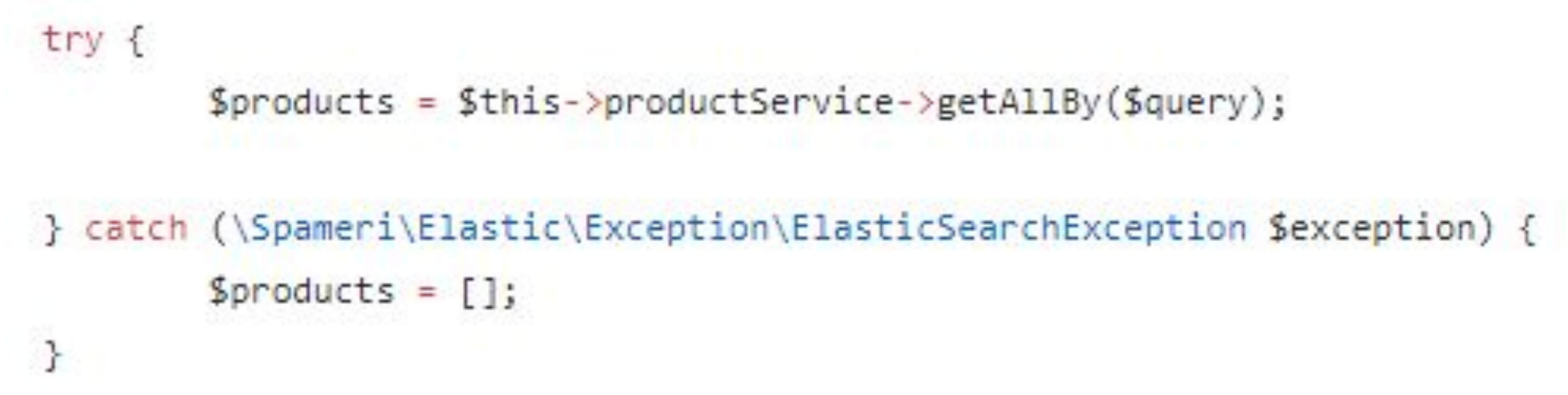
```
<div class="product-list products-list-full">
        <ul class="reset products full-products">
        {foreach $products as $product}
                <li>
                        <div class="spc">
                                <a href="//benu.cz/{$product->getAlias()}" class="detail" style="height: 321px">
                                        <h2 class="title">
                                                <span class="img">
                                                        <img class="lazy lazy-loaded" src="{$product->getImage()}" width=
                                                </span>
                                                <span class="name" style="height: 48px;">{$product->getName()}</span>
                                        </h2>
```

7. Search - Name

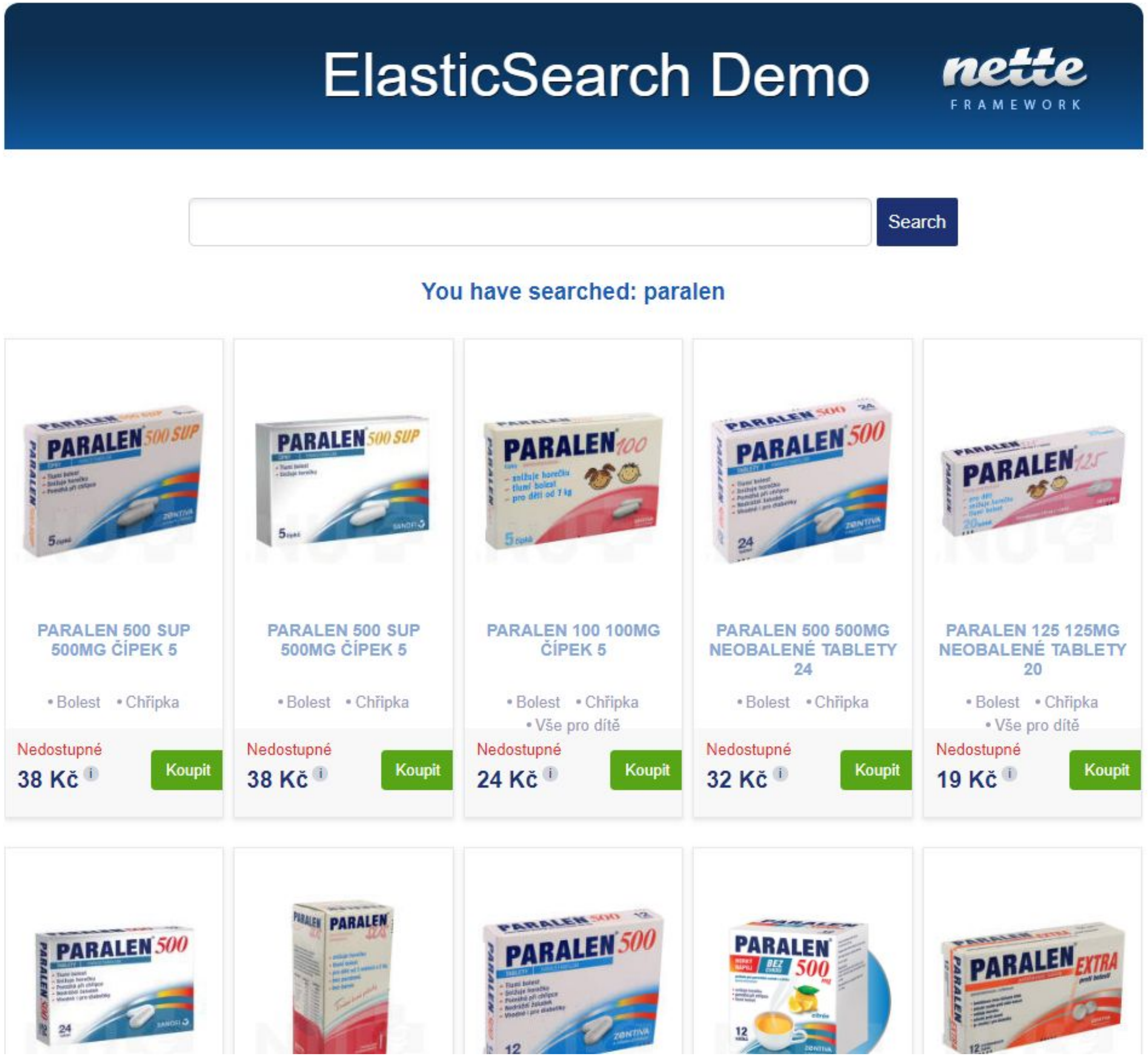


```
$query = new \Spameri\ElasticQuery\ElasticQuery();
$subQuery = new \Spameri\ElasticQuery\Query\QueryCollection();
$subQuery->addShouldQuery(
        new \Spameri\ElasticQuery\Query\Match(
                'name',
                $queryString
        )
);
```

7. Search - Entities



```
try {
        $products = $this->productService->getAllBy($query);

} catch (\Spameri\Elastic\Exception\ElasticSearchException $exception) {
        $products = [];
}
```



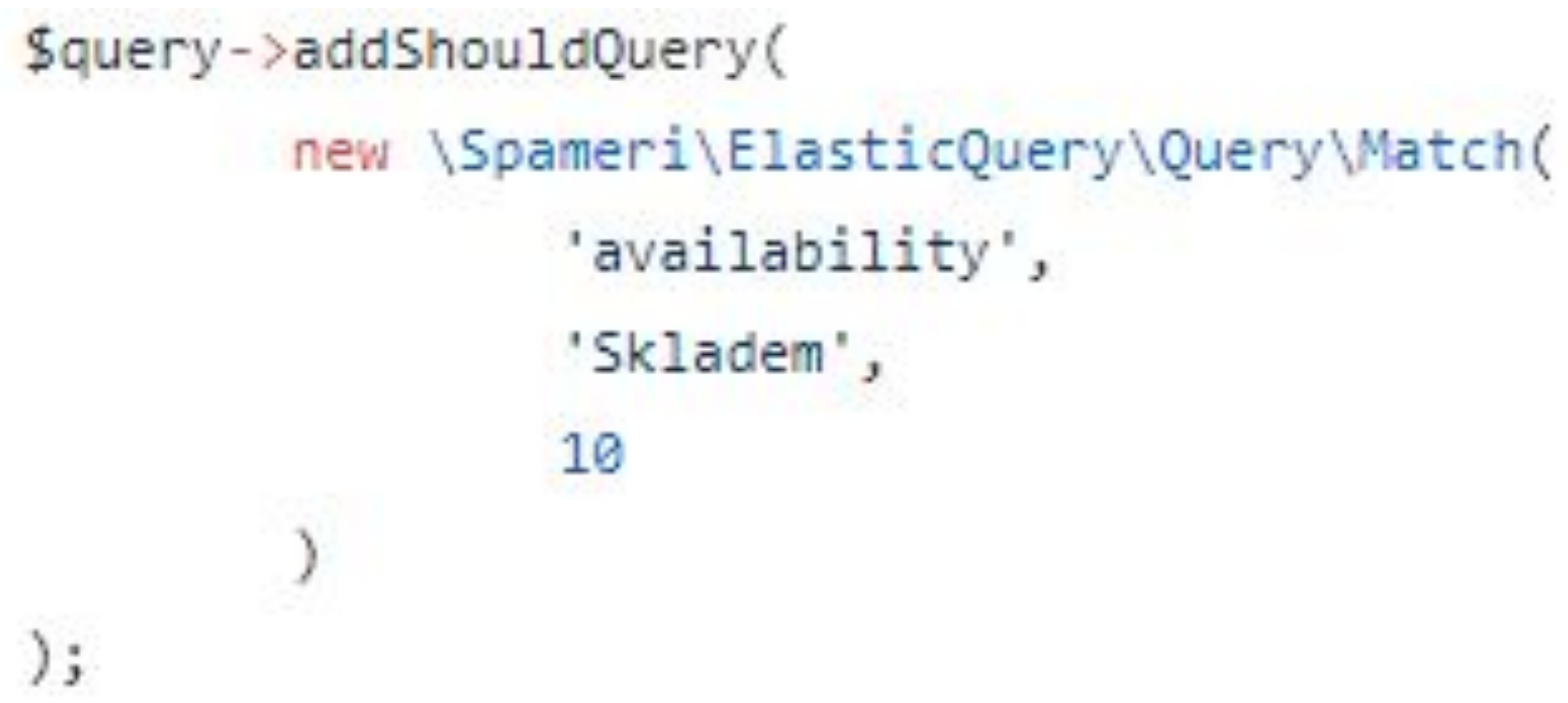
ElasticSearch Demo

You have searched: paralen

PARALEN 500 SUP 500MG ČÍPEK 5	PARALEN 500 SUP 500MG ČÍPEK 5	PARALEN 100 100MG ČÍPEK 5	PARALEN 500 500MG NEOBALENÉ TABLETY 24	PARALEN 125 125MG NEOBALENÉ TABLETY 20
• Bolest • Chřipka	• Bolest • Chřipka	• Bolest • Chřipka • Vše pro dítě	• Bolest • Chřipka	• Bolest • Chřipka • Vše pro dítě
Nedostupné	Nedostupné	Nedostupné	Nedostupné	Nedostupné
38 Kč	38 Kč	24 Kč	32 Kč	19 Kč
Koupit	Koupit	Koupit	Koupit	Koupit

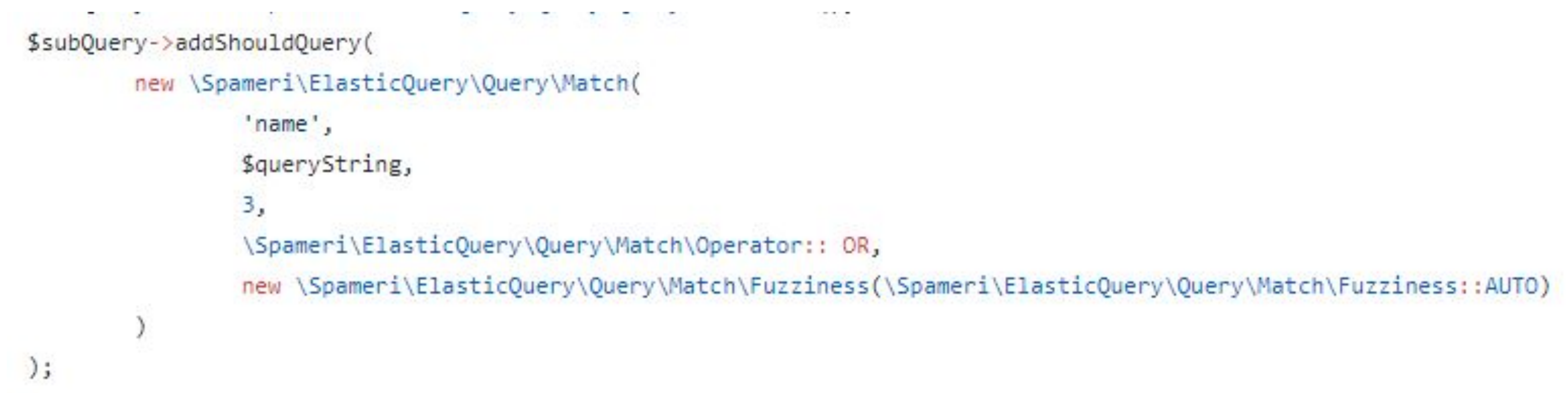
PeckaDesign

8. Fine tuning - Availability



```
$query->addShouldQuery(
        new \Spameri\ElasticQuery\Query\Match(
                'availability',
                'Skladem',
                10
        )
);
```

8. Fine tuning - Match



```
$subQuery->addShouldQuery(
        new \Spameri\ElasticQuery\Query\Match(
                'name',
                $queryString,
                3,
                \Spameri\ElasticQuery\Query\Match\Operator:: OR,
                new \Spameri\ElasticQuery\Query\Match\Fuzziness(\Spameri\ElasticQuery\Query\Match\Fuzziness::AUTO)
        )
);
```

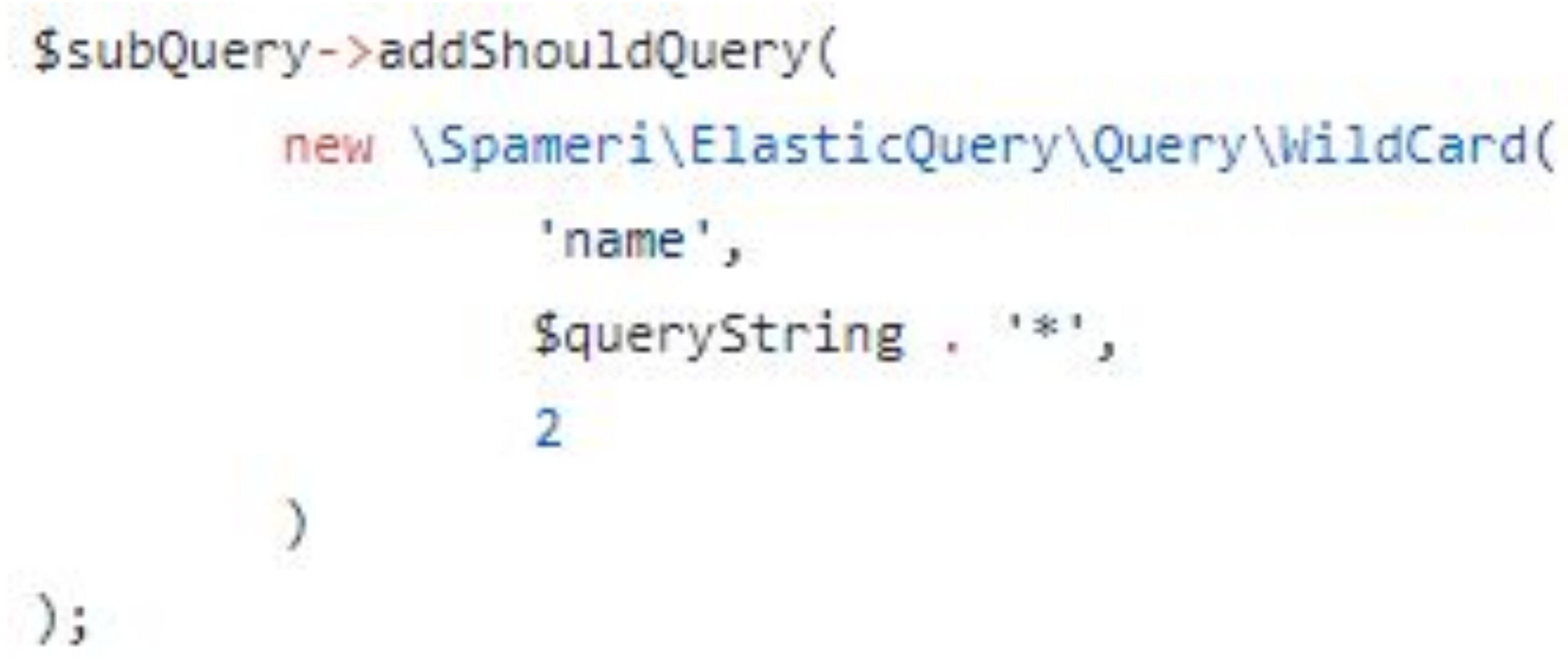
PeckaDesign

Václav Spamer Čevela
Spameri/Elastic

FB facebook.com/peckadesign

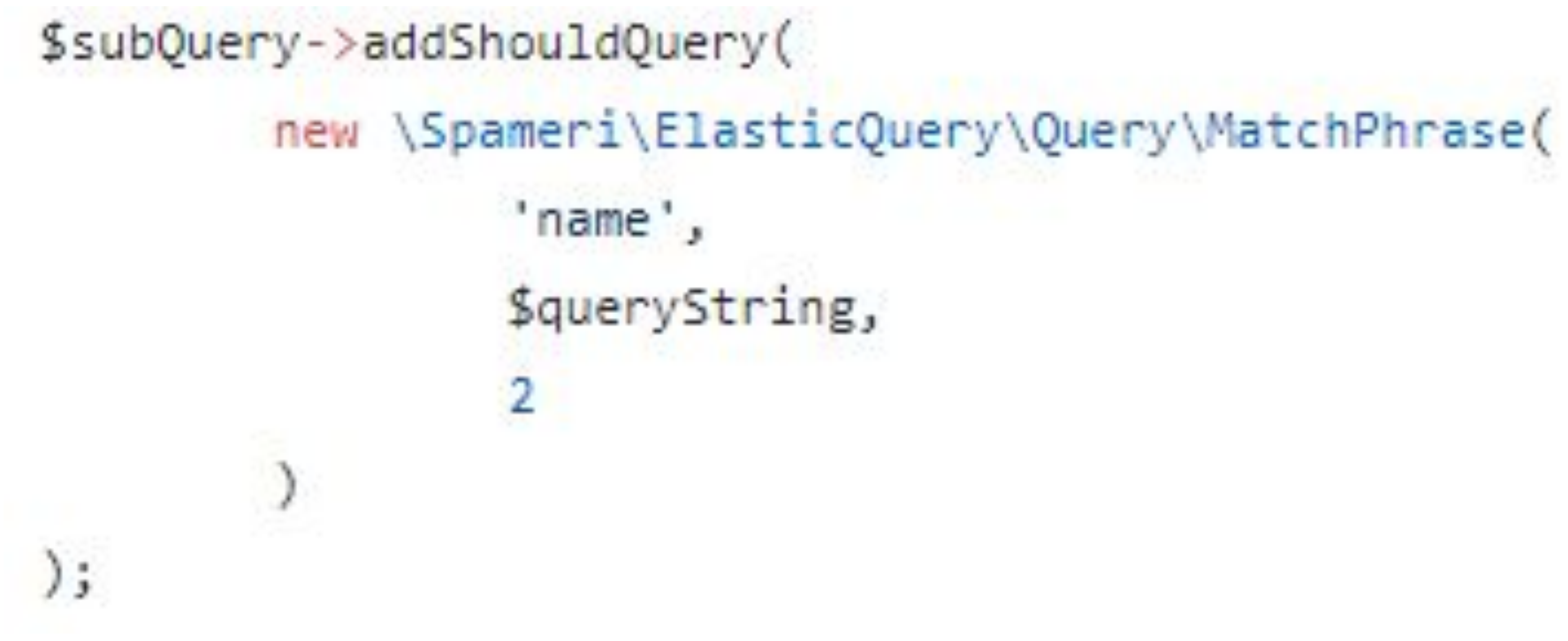
TW @Spamercz

8. Fine tuning - WildCard



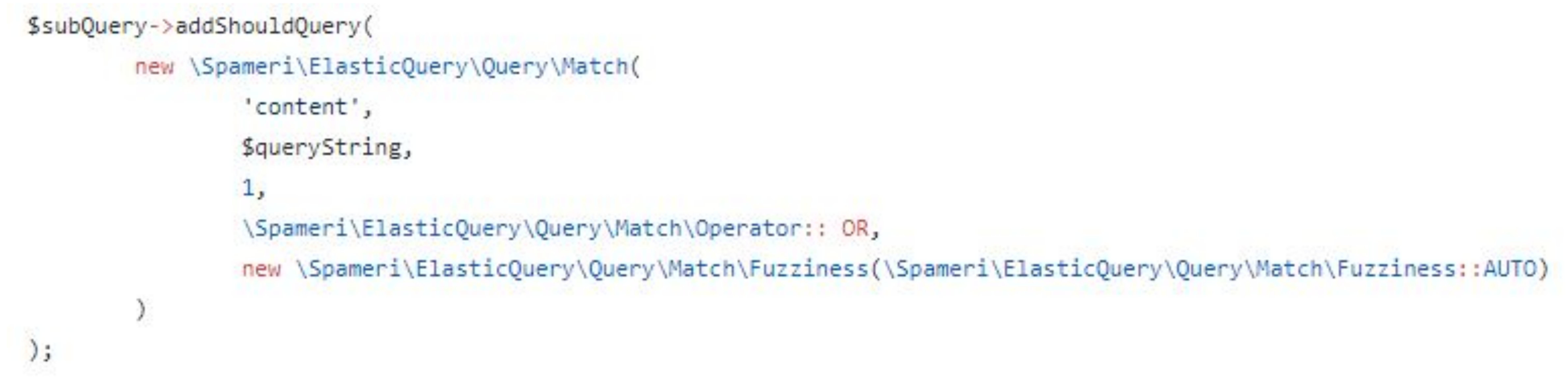
```
$subQuery->addShouldQuery(
        new \Spameri\ElasticQuery\Query\WildCard(
                'name',
                $queryString . '*',
                2
        )
);
```

8. Fine tuning - Phrase

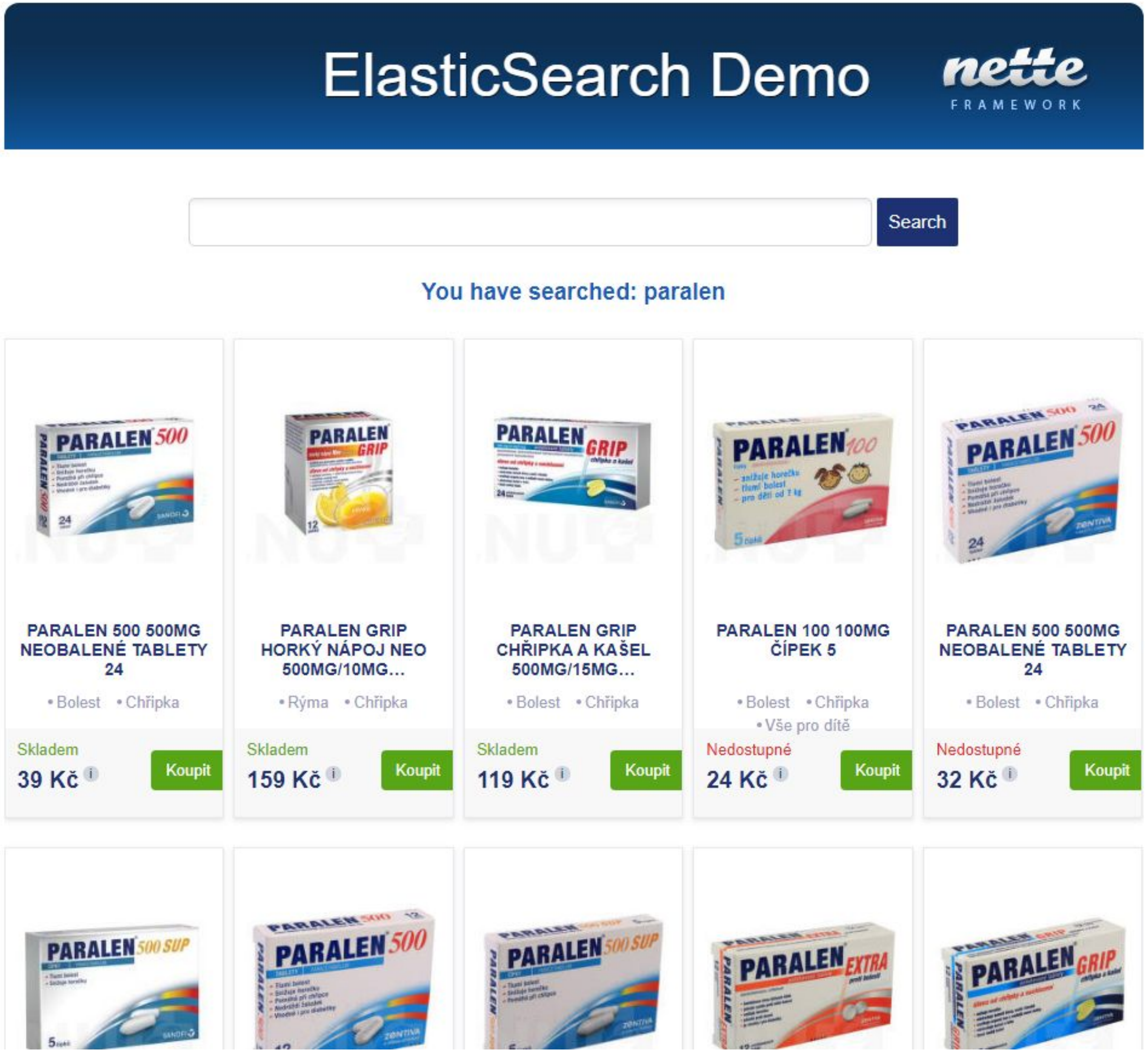


```
$subQuery->addShouldQuery(
        new \Spameri\ElasticQuery\Query\MatchPhrase(
                'name',
                $queryString,
                2
        )
);
```

8. Fine tuning - Content



```
$subQuery->addShouldQuery(
        new \Spameri\ElasticQuery\Query\Match(
                'content',
                $queryString,
                1,
                \Spameri\ElasticQuery\Query\Match\Operator:: OR,
                new \Spameri\ElasticQuery\Query\Match\Fuzziness(\Spameri\ElasticQuery\Query\Match\Fuzziness::AUTO)
        )
);
```



You have searched: paralen

PARALEN 500 500MG NEOBALENÉ TABLETY 24	PARALEN GRIP HORKÝ NÁPOJ NEO 500MG/10MG...	PARALEN GRIP CHŘIPKA A KAŠEL 500MG/15MG...	PARALEN 100 100MG ČÍPEK 5	PARALEN 500 500MG NEOBALENÉ TABLETY 24
• Bolest • Chřipka	• Rýma • Chřipka	• Bolest • Chřipka	• Bolest • Chřipka • Vše pro dítě	• Bolest • Chřipka
Skladem	Skladem	Skladem	Nedostupné	Nedostupné
39 Kč	159 Kč	119 Kč	24 Kč	32 Kč
Koupit	Koupit	Koupit	Koupit	Koupit

PeckaDesign

8. Fine tuning - Custom searches

Fuzzy

- GS Condro

CommonGrams

- Neo-angin

WordDelimiter

- Donna Hair

9. ???

10. Profit

Elastic objekty https://github.com/Spameri/**ElasticQuery**

Elastic obsluha https://github.com/Spameri/**Elastic**

QuickStart - https://github.com/Spameri/Elastic/blob/master/doc/**00_quick_start**.md

Demo - https://github.com/Spameri/**Sandbox**

Co nás čeká

Automatická EntityFactory
Konfigurace pomocí anotací, neonu
Vylepšená podpora Nested Query a Aggregation
Rozšiřování Query builderu
Další předem definované analyzery
Školení a workshopy

Děkuji za pozornost

Máte dotazy?

Já na vás dotaz mám tady:
http://bit.ly/skoleni-na-morave

Github/Spameri | Twitter @Spamercz | Blog blog.spameri.cz