PhpCollection

Release latest

Steevan BARBOYON

Nov 19, 2024

CONTENTS

steevanb/php-collection is a PHP library to create collections.

A collection is a list of typed data with methods to work on it.

The goal of this library is to remove *array* as much as possible to have typed data.

CHAPTER

ONE

CONTENTS

1.1 Installation

1.1.1 Requirements

PhpCollection only needs `PHP ^8.1`. That's all!

1.1.2 Installation

Install PhpCollection with Composer:

```
composer require steevanb/php-collection:6.*
```

1.2 Symfony bridge

PhpCollection comes with a bridge for `Symfony >6.1` to be integrated with symfony/serializer.

By default, the bridge is not autoloaded, to not add useless files to Composer autoload when you want to use PhpCollection as a simple PHP library, outside of Symfony.

You will need to do some manuals steps to install the bridge.

1.2.1 Autoload

Add the bridge to your Composer autoload in `composer.json`:

```
{
    "autoload": {
        "psr-4": {
            "Steevanb\\PhpCollection\\Bridge\\Symfony\\": "vendor/steevanb/php-
↪collection/bridge/Symfony"
        }
    }
}
```

1.2.2 Register the bundle

Add `PhpCollectionBundle` in `config/bundles.php`:

```
<?php
return [
```

(continues on next page)

(continued from previous page)

```
    Steevanb\PhpCollection\Bridge\Symfony\PhpCollectionBundle::class => ['all' => true]
];
?>
```

1.2.3 Integration with symfony/serializer

PhpCollectionBundle add 2 denormalizers: `ObjectCollectionDenormalizer` and `ScalarCollectionDenormalizer`.

This two denormalizers integrate PhpCollection in symfony/serializer, to denormalize array in PhpCollection.

Example

```
<?php
use Steevanb\PhpCollection\ScalarCollection\StringCollection;

// $collection is an instance of StringCollection, with values foo and bar
$collection = $denormalizer->denormalize(['foo', 'bar'], StringCollection::class);
?>
```

1.3 Methods

Methods on Collection can come from:

- CollectionInterface
- AbstractCollection
- AbstractObjectCollection
- AbstractObjectNullableCollection

1.3.1 CollectionInterface

This interface should be implemented by all Collection.

It contains all basics methods to interact with your Collection.

Why all methods are not in CollectionInterface? Because it's a huge work to implement all PHP array functions, so the interface contains only basics methods to not have a BC break when we want to implement a PHP function in the Collection.

set(string|int $key, mixed $value): static

Define the value for the key $key. If the key does not exists, it will be created.

```
<?php
$collection = new FooCollection();
$collection->set(42, 'foo');
$collection->set('key', 'bar');
?>
```

1.3.2 AbstractCollection

This abstract class implement CollectionInterface and add some methods.

You can extends this class for any Collection, but if you create an object Collection, you should extends `AbstractObjectCollection` or `AbstractObjectNullableCollection`.

1.3.3 AbstractObjectCollection

getValueFqcn(): string

Return the fully qualified class name (FQCN) of the class, enum or interface each value should be an instance of.

```php
<?php
use Steevanb\PhpCollection\ObjectCollection\AbstractObjectCollection;

class FooCollection extends AbstractObjectCollection
{
    public static function getValueFqcn(): string
    {
        return \DateTimeInterface::class;
    }
}
?>
```

1.4 Create a Collection

Your class should implement `Steevanb\PhpCollection\CollectionInterface`.

1.4.1 AbstractObjectCollection

Most of the time, you can extend `Steevanb\PhpCollection\ObjectCollection\AbstractObjectCollection` instead of implementing CollectionInterface, it will be easier to create your Collection with it.

AbstractObjectCollection use phpstan *Generics* and validate instance of each value.

Generics are used for methods like AbstractObjectCollection::toArray(), but that's not enough: maybe you do not use phpstan, so nothing will validate values added in your Collection. So AbstractObjectCollection validate that each value should be an instance of the return of `getValueFqcn()`.

Example

```php
<?php
use Steevanb\PhpCollection\ObjectCollection\AbstractObjectCollection;

/** @extends AbstractObjectCollection<\DateTimeImmutable> */
class DateTimeImmutableCollection extends AbstractObjectCollection
{
    public static function getValueFqcn(): string
    {
        return \DateTimeImmutable::class;
    }
```

(continues on next page)

(continued from previous page)

```
}
?>
```

1.4.2 AbstractObjectNullableCollection

Exact same behavior of AbstractObjectCollection, but allow `null`.

Example

```
<?php
use Steevanb\PhpCollection\ObjectCollection\AbstractObjectNullableCollection;

/** @extends AbstractObjectNullableCollection<\DateTimeImmutable> */
class DateTimeImmutableCollection extends AbstractObjectNullableCollection
{
    public static function getValueFqcn(): string
    {
        return \DateTimeImmutable::class;
    }
}
?>
```

1.5 Null value

You can see if a Collection allow `null` in it's name: if it contains `Nullable`, `null` is allowed, instead null is not allowed.

Example: `StringCollection` and `StringNullableCollection`.

1.6 Read only

By default, you can add, update or remove a value in the Collection.

You can define the Collection as read only when you want to be sure all values will not be modified, with `setReadOnly()`.

1.6.1 Example

```
<?php
$collection = new StringCollection(['foo', 'bar']);
$collection->setReadOnly();

// Steevanb\PhpCollection\Exception\ReadOnlyException will be throwned
$collection->add('baz');
?>
```

1.7 Generics

1.7.1 Generics in PHP

Generics in PHP are discuted in Generic types and functions RFC and in Generic arrays RFC, but they are not adopted and generics will never be integrated in PHP.

1.7.2 Generics with phpstan

phpstan can solve this problem with it's generics implementation.

1.7.3 Generics in PhpCollection

When you *Create a Collection*, you configure generics with `/** @extends AbstractObjectCollection<Foo> */`.

1.8 About

PhpCollection has been developed by Steevan BARBOYON.

Some links about me:

- Online CV
- Contact me on LinkedIn
- My GitHub