
Flint Documentation

Release 1.2.2

Henrik Bjrnskov

Sep 27, 2017

Contents

1	Features	3
2	Install	5
3	Getting Started	7
3.1	Inheriting	8
3.2	Replacing Values	8
4	Tests	9

“A worker who fastens things by tacking them (as with tacks or by spotwelding)”

Tacker is a config component that binds Symfony Config together into a reuseable config loader for Pimple.

CHAPTER 1

Features

Works with `yml`, `ini`, `json`, and `php`.

- Inherit configurations files by using a special `@import`
- Mix different configuration formats while using `@import`
- Read configuration files are cached if a `cache_dir` is set and `debug` is set to `false`.
- Simple for all users of Pimple such as `Silex` and `Flint`.
- Only depends on `Symfony Config`.

CHAPTER 2

Install

Use composer to install this beautiful library.

```
{  
  "require" : {  
    "flint/tacker" : "~1.0"  
  }  
}
```


CHAPTER 3

Getting Started

```
<?php

use Pimple\Container;
use Symfony\Component\Config\FileLocator;
use Tacker\Configurator;
use Tacker\Loader\CacheLoader;
use Tacker\Loader\JsonFileLoader;
use Tacker\Loader\NormalizerLoader;
use Tacker\Normalizer\ChainNormalizer;

$resources = new ResourceCollection;
$loader = new JsonFileLoader(new FileLocator, $resources);

// add CacheLoader and NormalizerLoader
$loader = new CacheLoader(new NormalizerLoader($loader, new ChainNormalizer),
    ↪$resources);

// load just the parameters
$parameters = $loader->load('config.{ini,json,yml}');

// or use cofigurator with pimple
$configurator = new Configurator($loader);
$configurator->configure(new Container, 'config.json');
```

Fortunately there exists a better way of doing all of this.

```
<?php

use Tacker\LoaderBuilder;

$loader = LoaderBuilder::create($paths)
    ->build()
;
```

```
$configurator = new Configurator($loader);
$configurator->configure($container, 'config.json');
```

The LoaderBuilder will try and setup sane defaults. Which means all the loaders for php, ini, json and yml if `Symfony\Component\Yaml\Yaml` is available.

The builder contains methods for configuring the different parts, but configuring normalizers or loaders will remove the default unless you call `$builder->addDefaultNormalizers()` and `$builder->addDefaultLoaders()`.

Inheriting

All of the loaders supports inherited configuration files. These are supported through a special `@import` key. `@import` can contain a single string like `config.json` or an array of configuration files to load. In yml it would look something like:

```
@import:
- config.json
- another.yaml
- third.ini
```

Replacing Values

Tacker comes with support for replacing values from environment value and/or Pimple services.

For environment variables it matches `#CAPITALIZED_NAME#` and for Pimple it matches `%service.name%`.

```
<?php

use Pimple\Container;
use Tacker\Normalizer\EnvironmentNormalizer;
use Tacker\Normalizer\PimpleNormalizer;
use Tacker\Normalizer\ChainNormalizer;

$normalizer = new ChainNormalizer(array(
    new EnvironmentNormalizer,
    new PimpleNormalizer(new Container),
));
```

CHAPTER 4

Tests

Tacker is spec tested with PhpSpec. You run the tests with:

```
$ ./vendor/bin/phpspec run
```