phpstorm-metadata-export Documentation

Release 1.7.0

Jan 18, 2021

CONTENTS

1 Installation **3**

2 Containers **5**
2.1 Pimple 5
2.2 PHP-DI 5
2.3 Laminas (Zend) ServiceManager 5

3 Usage **7**
3.1 Using Generator directly 7
3.2 PSR-15 7
3.3 Slim 3 8
3.4 Silex 8

4 License **11**

Export PhpStorm Advanced Metadata from DI containers to enable code completion.

The library is inspired by Pimple Container Dumper for Silex but it doesn't require IDE plugin because it uses native PhpStorm export format. It also supports Slim and is extensible to support more DI containers and frameworks in future.

CHAPTER

ONE

INSTALLATION

Install by composer

```
composer require sandfoxme/phpstorm-metadata-export --dev
```

CHAPTER

TWO

CONTAINERS

2.1 Pimple

The library has full support for the Pimple DI version 3.*

2.2 PHP-DI

The library has full support for the PHP-DI version 6.*

2.3 Laminas (Zend) ServiceManager

The library has partial support for the Laminas ServiceManager and Zend ServiceManager. Please note that there's no open and public way of iterating over SM entries so the implementation is tied to the internal structure not covered by the semantic version compatibility promise. Feel free to open an issue if it is broken for some scenario.

Obviously the implementation also cannot create hints for abstract factories

The library is tested against Zend SM version 3.3 and should work with any 3.* version

CHAPTER

THREE

USAGE

The library has integration classes available for Slim and Silex. By default exporter will create directory `.phpstorm.meta.php` in the root path of your project (where `vendor` directory is) and put metadata file to it on every request. It is a good idea to enable integration components only in development environment. Don't forget to gitignore your `.phpstorm.meta.php` directory.

3.1 Using Generator directly

```
<?php

use SandFox\PhpStorm\Metadata\Generator;

$container = new Container();

file_put_contents(
    '/path/to/project/.phpstorm.meta.php/my_export_file.meta.php',
    Generator::get([$container])
);

// OR

Generator::store(
    '/path/to/project/.phpstorm.meta.php/my_export_file.meta.php',
    [$container]
);
```

3.2 PSR-15

Version 1.1 adds Psr15 compliant middleware implementation like Slim 4 or Mezzio. It generally uses the same approach as the earlier Slim Middleware. The classname is `SandFox\PhpStorm\Metadata\Integration\Psr15\ContainerExportMiddleware`.

```
<?php

use SandFox\PhpStorm\Metadata\Integration\Psr15\ContainerExportMiddleware;

$middleware = new ContainerExportMiddleware($container);

// You can also override metadata filename like this
```

(continues on next page)

(continued from previous page)

```
$middleware = new ContainerExportMiddleware($container, [
    // it is a good idea to use full path here
    'filename' => '/path/to/project/.phpstorm.meta.php/my_export_file.meta.php',
]);

// Register middleware the way your compliant framework allows it
$myPsr15CompliantApp->registerMiddleware($middleware);
```

3.3 Slim 3

Note: Slim 4 can also use this middleware but PSR-15 is preferable

Add middleware class `SandFox\PhpStorm\Metadata\Integration\Slim\ContainerExportMiddleware` to your Slim app.

```
<?php

use Slim\App;
use SandFox\PhpStorm\Metadata\Integration\Slim\ContainerExportMiddleware;

$app = new App();

$app->add(new ContainerExportMiddleware($app->getContainer()));

// You can also override metadata filename like this
$app->add(new ContainerExportMiddleware($app->getContainer(), [
    // it is a good idea to use full path here
    'filename' => '/path/to/project/.phpstorm.meta.php/my_export_file.meta.php',
]));
```

3.4 Silex

Warning: This class is deprecated since 1.7.0 and will be removed in 2.0. You should use neither this middleware nor Silex itself.

Add service provider class `SandFox\PhpStorm\Metadata\Integration\Silex\ContainerExportProvider` to your Silex app.

```
<?php

use Silex\Application;
use SandFox\PhpStorm\Metadata\Integration\Silex\ContainerExportProvider;

$app = new Application();

$app->register(new ContainerExportProvider());

// You can also override metadata filename like this
```

(continues on next page)

(continued from previous page)

```
$app->register(new ContainerExportProvider(), [
    // it is a good idea to use full path here
    'phpstorm.metadata.filename' => '/path/to/project/.phpstorm.meta.php/my_export_
↪file.meta.php',
]);
```

CHAPTER

FOUR

LICENSE

The library is available as open source under the terms of the MIT License. See LICENSE.md