

Real World Attributes



pixiv Inc.
USAMI Kenta

pixiv

お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE / にゃんだーすわん
- ピクシブ株式会社 pixiv事業本部 エンジニア
 - 最近はピクシブ百科事典(dic.pixiv.net)を開発しています
- Emacs Lisper, PHPer
 - Emacs PHP Modeを開発しています(2017年-)
- プログラミング言語にちょっとこだわりのある素人

PHPerKaigiで話した

Attributeを極める

Mastering Attributes



pixiv Inc.
USAMI Kenta

pixiv

今北産業

今北産業

- なんか #[Attribute] みたいなへんてこ記法がPHP 8.0で追加された
- いままで /** DocComment */ に書かれてたやつのだうとして提案
- クラスとかメソッドに書けるコメントみたいなもんだけどデータを取り出せる

こんなの何に使うの
って？

キミだけの最強の活用
法を見付けだせ！！！！

ご静聴ありがとうございました！

…というひどい
発表をしてきた

#[Attr()]と書く
謎記法が出てきた
...ということだけ、
お見知りおきください

誰が使うんだ？
という話

PHPUnitの実行制御

```
/**  
 * @dataProvider idsProvider  
 */  
public function test($id, $expected):  
void {
```

@dataProvider

```
/**  
 * [$id, $expected]  
 */  
public function idsProvider() {  
    return [  
        [1, 'value'],  
        [2, 'value'],  
    ];  
}
```

PHPUnit 10 vs Attribute

- PHPUnit 10でAttributeに移行、11で非推奨化、12で削除予定
- 超大量の移行リスト… [sebastianbergmann/phpunit#4502](https://github.com/sebastianbergmann/phpunit/issues/4502)
- その他の変更は02氏の発表を参照のこと



PHPUnit 10 概論

2023/03/23 PHPerKaigi 2023

@02

Support PHP 8 attributes for adding metadata to test classes and test methods as well as tested code units #4502

New issue

Closed

sebastianbergmann opened this issue on Oct 30, 2020 · 0 comments



sebastianbergmann commented on Oct 30, 2020 · edited

Owner

PHPUnit currently supports the following annotations in special PHP comments ("DocBlocks", "doc-comments"):

- @after
- @afterClass
- @backupGlobals enabled and @backupGlobals disabled
- @backupStaticAttributes enabled and @backupStaticAttributes disabled
- @before
- @beforeClass
- @codeCoverageIgnore
- @covers ::functionName
- @covers ClassName
- @covers ClassName::methodName (not recommended because too fine-grained)
- @covers ClassName<extended> (deprecated; will be removed in PHPUnit 10)
- @covers ClassName::<public> (deprecated; will be removed in PHPUnit 10)
- @covers ClassName::<protected> (deprecated; will be removed in PHPUnit 10)
- @covers ClassName::<private> (deprecated; will be removed in PHPUnit 10)
- @covers ClassName::<!public> (deprecated; will be removed in PHPUnit 10)
- @covers ClassName::<!protected> (deprecated; will be removed in PHPUnit 10)
- @covers ClassName::<!private> (deprecated; will be removed in PHPUnit 10)

Assignees

sebastianbergmann

Labels

type/enhancement

Projects

None yet

Milestone

PHPUnit 10.0

Development

No branches or pull requests

Notifications

Customize

Subscribe

You're not receiving notifications from this thread.

テスト好きかい？

「うん、大好きさ」

むかし書いた

pixiv inside [archive]



pixiv insideは移転しました！ » <https://inside.pixiv.blog/>

2016-10-25

DocCommentでPHPのユニットテストの書きやすさを劇的に改善する手法

45 B!ブックマーク 0 シェアする ツイート

こんにちは、pixivでPHPをやってる@tadsanです。好きなテストングフレームワークはPHPUnitです！好きな某CALOIDはテトさんです！

みなさまはユニットテストを書いていらっしゃいますか？今回はDocCommentとPHPUnitのデータプロバイダーをうまく利用してテストの記述を省力化する手法を紹介いたします べ(// > <)バ

提案手法

実装のDocCommentに「期待値」と「入力パラメータ」を記述することで、テストケースメソッドをいちいち追加せずともユニットテストできるようになります。また、入力（パラメータ）と出力（期待値）を明記することで、実行せずともコメントとしてわかりやすくする効果があります。

pixiv insideとは

<https://inside.pixiv.blog/> に移転しました。こちらは2016年末までのアーカイブです。

✓ 読者です 234

[このブログについて](#)

月別アーカイブ

▼ 2016 (52)
2016 / 12 (28)
2016 / 11 (6)
2016 / 10 (1)
2016 / 9 (4)
2016 / 8 (1)
2016 / 7 (2)
2016 / 6 (1)
2016 / 5 (4)
2016 / 4 (1)
2016 / 2 (2)
2016 / 1 (2)

▶ 2015 (60)

▶ 2014 (63)

実装の上にコメントでテストを書ける

```
<?php

/**
 * @route\example http://www.pixiv.net/i/12345 {illust_id: 12345}
 */
public static function fullTouchMemberIllustMedium(array $params)
{
    Util_Assert::num($params['illust_id']);
    $params = array_merge(['mode' => 'medium'], $params);
    return self::buildUrl(SYSTEM_URL_TOUCH, '/member_illust.php', ['mode', 'il
}
```

DataProviderを

これはAttributeに
できる

ということでやったよ

Attributeの定義はこれだけ

```
#[Attribute(Attribute::TARGET_METHOD | Attribute::IS_REPEATABLE)]  
#[Spiral\Attributes\NamedArgumentConstructor]  
final class RouteExample  
{  
    /**  
     * @param non-empty-string $expected URL または 例外のクラス名  
     * @param array<mixed> $params  
     */  
    public function __construct(  
        private string $expected,  
        private array $params  
    )  
    {}  
}
```


実装の上にAttributeでテスト

```
/**
 * イラストの短縮URL
 *
 * @route\example https://www.pixiv.net/i/12345 {id: 12345}
 */
#[RouteExample('https://www.pixiv.net/i/12345', ['id' => 12345])]
public static function fullWwwShortI(array $params)
{
    Util_Assert::int($params['id']);

    return self::_buildUrl( host: SYSTEM_URL_WWW, route: '/i/:id', [], $params);
}
```