

گزارش تست نویسی

برای تست نویسی از phpunit استفاده شده است. یکی از ابزار های تست در پروژه های مبتنی بر PHP استفاده از PHPUnit می باشد.

برای نصب phpunit روی پروژه php از composer استفاده می شود.

چه زمانی ما تست نیاز داریم ؟ هر زمان که کدی برای اجرا باشد تست هم باید باشد. اما این بسته به بزرگ یا کوچک بودن پروژه لزوم خود را نشان می دهد. در پروژه های بزرگ داشتن تست هزینه پروژه را پایین می آورد.

Unit testing با مفهوم Software Flow متفاوت است. این که یک کاربر چگونه از اپلیکیشن شما استفاده می کند Software flow می باشد. اما Unit testing یعنی تست واحد یا method test که با اجرای phpunit نتیجه حاصل می گردد.

unit به معنی واحد یا یک است . به معنی این که هر متد یک عمل را تست می کند. به طور مثال شما یک متد برای چک کردن مقدار یک عدد می نویسید. تست بر اساس نحوه عملکرد یک متد می باشد ، تست را هم برای آن می نویسید. منطق متد خود را عوض می کنید. با اجرای تستی که از قبل نوشته اید می توانید نحوه عملکرد متد را نیز بررسی کنید.

پس از نصب phpunit باید تنظیمات آن را پیاده سازی کنیم. تنظیمات یک فایل xml است که از ساختار زیر پیروی می کند.

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <phpunit
3     bootstrap="./api/index.php"
4     colors="true"
5     verbose="true"
6     stopOnFailure="true"
7 >
8
9 <testsuites>
10 <testsuite name="unit">
11     <directory>./tests/unit</directory>
12     <file>./tests/unit/oneTest.php</file>
13 </testsuite>
14
15 </testsuites>
16 </phpunit>
```

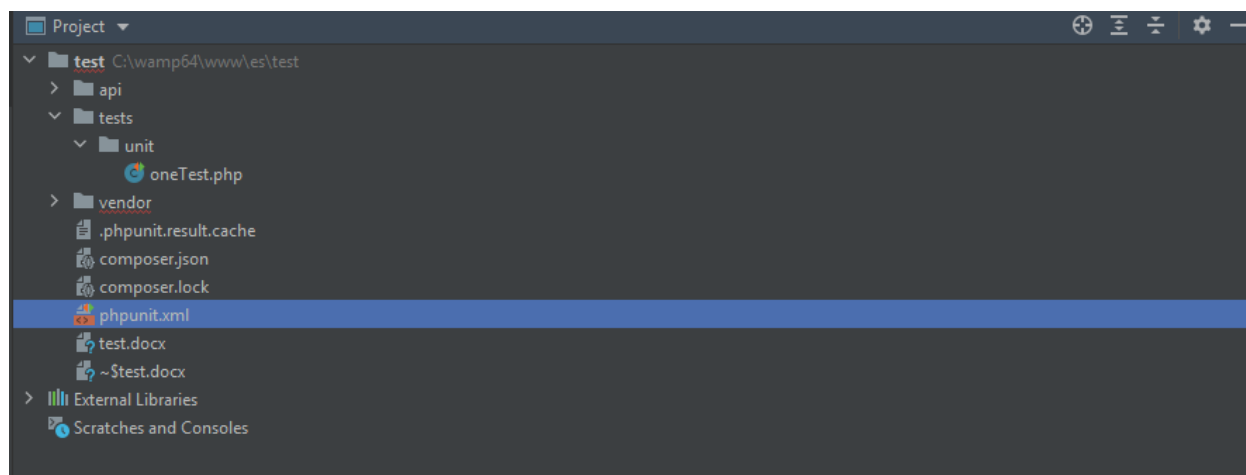
تگ phpunit ویژگی هایی دارد که همان تنظیمات اجرای تست ها می باشد. مثلا stopOnFailure یعنی اگر تستی fail شد برنامه تست را متوقف کند.

bootstrap باید مسیر شروع تست را مشخص کند یعنی ایجاد دسترسی به نرم افزار یا همان bootstrap نرم افزار شما. در بسیاری از موارد مسیر autoload می باشد. مثلا می توان مسیر شروع را خودمان با توجه به فایل های مورد نیاز برای تست خودمان تعیین کنیم.

Run PHPUnit : فرمان زیر تست های دایرکتوری test را بر اساس تنظیمات phpunit.xml در روت پروژه اجرا می کند.

vendor/bin/phpunit

Directory mapping PHPUnit : ساختار دایرکتوری های مهم در پروژه پس از پیاده سازی unit testing در آن به شکل زیر است :



Writing Tests for PHPUnit

کلاس های تست باید از کلاس testCase ارث بری کند. این کلاس در مسیر PHPUnit\Framework\TestCase می باشد.

```
use PHPUnit\Framework\TestCase;

class Example extends TestCase
{
    public function testOne(){
        $this->assertTrue(true);
    }

    /** @test */
    public function tow(){
        $this->assertTrue(true);
    }
}
```

هر کلاس test می تواند تعدادی تست در خود داشته باشد. اگر ابتدای نام آن test باشد یا annotation @test را در ابتدای متد بیاوریم به عنوان تست آن را می شناسد.

پس از اجرای تست در محیط ترمینال به ازای هر تست یک علامت نمایش داده می شود که به شرح زیر است:

. : که به معنی موفقیت آمیز بودن تست می باشد.

F : به معنی خطا در تست می باشد.

.... چهار نقطه در کنار هم یعنی هر 4 تست successfull بوده اند. F.. یعنی دو تست success و یک تست fail شده و سپس یک تست دیگر success شده است.

تست 1:

تابع array_msort(\$array, \$cols) : این تابع برای مرتب سازی یک آرایه دو سطحی بر اساس یک کلید خاص نوشته شده است.

```

8 >> class oneTest extends TestCase
9 {
10 > public function test_array_msort()
11 {
12     $arr1 = array(
13         array('id'=>1, 'name'=>'aA', 'cat'=>'cc'),
14         array('id'=>2, 'name'=>'aa', 'cat'=>'dd'),
15         array('id'=>3, 'name'=>'bb', 'cat'=>'cc'),
16         array('id'=>4, 'name'=>'bb', 'cat'=>'dd')
17     );
18     $expected = array(
19         array('id'=>3, 'name'=>'bb', 'cat'=>'cc'),
20         array('id'=>4, 'name'=>'bb', 'cat'=>'dd'),
21         array('id'=>1, 'name'=>'aA', 'cat'=>'cc'),
22         array('id'=>2, 'name'=>'aa', 'cat'=>'dd'),
23     );
24
25     $res = array_msort($arr1, array('name'=>SORT_DESC, 'cat'=>SORT_ASC));
26     $i = 0;
27     foreach ($res as $d)
28     {
29         $this->assertEquals($expected[$i], $d);
30         $i++;
31     }
32 }
33 }

```

نتیجه:

```

C:\wamp64\www\es\test>vendor\bin\phpunit
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

Runtime:       PHP 7.4.9
Configuration: C:\wamp64\www\es\test\phpunit.xml

.
1 / 1 (100%)

Time: 00:00.035, Memory: 6.00 MB

OK (1 test, 4 assertions)
C:\wamp64\www\es\test>

```

تست 2:

تابع (`array_getKeyValue($array, $key, $value)`): این تابع یک آرایه دو سطحی دریافت می کند و چک میکند در هر آرایه اگر کلید برابر مقدار مورد نظر باشد در این صورت آن آرایه را برمیگرداند.

```

35 public function test_array_getKeyValue()
36 {
37     $arr1 = array(
38         array('id'=>1,'name'=>'ali','cat'=>'cm'),
39         array('id'=>2,'name'=>'aa','cat'=>'dd'),
40         array('id'=>3,'name'=>'ali','cat'=>'cc'),
41         array('id'=>4,'name'=>'bfb','cat'=>'dd'),
42         array('id'=>5,'name'=>'buub','cat'=>'dd'),
43         array('id'=>6,'name'=>'bpkob','cat'=>'dd')
44     );
45
46     $expected = array(
47         array('id'=>1,'name'=>'ali','cat'=>'cm'),
48         array('id'=>3,'name'=>'ali','cat'=>'cc'),
49     );
50
51
52     $res = array_getKeyValue($arr1, key: 'name', value: 'ali');
53     $i = 0;
54     foreach ($res as $d)
55     {
56         $this->assertEquals($expected[$i],$d);
57         $i++;
58     }
59
60
61 }

```

نتیجه:

```

C:\wamp64\www\es\test>vendor\bin\phpunit
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

Runtime:       PHP 7.4.9
Configuration: C:\wamp64\www\es\test\phpunit.xml

.
1 / 1 (100%)

Time: 00:00.016, Memory: 6.00 MB

OK (1 test, 2 assertions)

C:\wamp64\www\es\test>

```

تست 3:

تابع (`searchByAge($array, array $key=[15,45])`: این تابع یک آرایه دو سطحی دریافت می کند و چک میکند در هر آرایه اگر سن در بازه مورد نظر باشد در این صورت آن آرایه را برمیگرداند.

```

63 public function test_searchByAge()
64 {
65     $arr1 = array(
66         array('id'=>1,'name'=>'aa1','Age'=>20),
67         array('id'=>2,'name'=>'aa2','Age'=>31),
68         array('id'=>3,'name'=>'aa3','Age'=>24),
69         array('id'=>4,'name'=>'aa4','Age'=>19),
70         array('id'=>5,'name'=>'aa5','Age'=>29),
71         array('id'=>6,'name'=>'aa6','Age'=>31),
72         array('id'=>7,'name'=>'aa6','Age'=>22),
73         array('id'=>8,'name'=>'aa7','Age'=>27),
74         array('id'=>9,'name'=>'aa8','Age'=>37),
75         array('id'=>10,'name'=>'aa9','Age'=>17),
76         array('id'=>11,'name'=>'aa7','Age'=>32),
77     );
78
79     $expected = array(
80         array('id'=>3,'name'=>'aa3','Age'=>24),
81         array('id'=>5,'name'=>'aa5','Age'=>29),
82         array('id'=>8,'name'=>'aa7','Age'=>27),
83     );
84
85
86     $res = searchByAge($arr1,[23,30]);
87     $i = 2;
88     $expected1 = array_msort($expected,array('id'=>SORT_DESC));
89     $res1 = array_msort($res,array('id'=>SORT_DESC));
90
91     foreach ($res1 as $d)
92     {
93         $this->assertEquals($expected1[$i],$d);
94
95         $i--;
96     }
97
98 }

```

نتیجه:

```

C:\wamp64\www\es\test>vendor\bin\phpunit
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

Runtime:       PHP 7.4.9
Configuration: C:\wamp64\www\es\test\phpunit.xml

.                                                     1 / 1 (100%)

Time: 00:00.016, Memory: 6.00 MB

OK (1 test, 3 assertions)
C:\wamp64\www\es\test>

```

نتیجه اجرا همزمان:

```
C:\wamp64\www\es\test>vendor\bin\phpunit
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.
```

```
Runtime:      PHP 7.4.9
```

```
Configuration: C:\wamp64\www\es\test\phpunit.xml
```

```
... 3 / 3 (100%)
```

```
Time: 00:00.146, Memory: 6.00 MB
```

```
OK (3 tests, 9 assertions)
```

```
C:\wamp64\www\es\test>
```