

利用单元测试对PHP代码进行检查 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/167/2021_2022__E5_88_A9_E7_94_A8_E5_8D_95_E5_c103_167252.htm 测试驱动的开发和单元测试是确保代码在经过修改和重大调整之后依然能如我们期望的一样工作的最新方法。在本文中，您将学习到如何在模块、数据库和用户界面（UI）层对自己的PHP代码进行单元测试。现在是凌晨3点。我们怎样才能知道自己的代码依然在工作呢？Web应用程序是24x7不间断运行的，因此我的程序是否还在运行这个问题会在晚上一直困扰我。单元测试已经帮我对自己的代码建立了足够的信心 这样我就可以安稳地睡个好觉了。单元测试是一个为代码编写测试用例并自动运行这些测试的框架。测试驱动的开发是一种单元测试方法，其思想是应该首先编写测试程序，并验证这些测试可以发现错误，然后才开始编写需要通过这些测试的代码。当所有测试都通过时，我们开发的特性也就完成了。这些单元测试的价值是我们可以随时运行它们 在签入代码之前，重大修改之后，或者部署到正在运行的系统之后都可以。PHP单元测试对于PHP来说，单元测试框架是PHPUnit2。可以使用PEAR命令行作为一个PEAR模块来安装这个系统：`% pear install PHPUnit2`。在安装这个框架之后，可以通过创建派生于PHPUnit2_Framework_TestCase的测试类来编写单元测试。模块单元测试 我发现开始单元测试最好的地方是在应用程序的业务逻辑模块中。我使用了一个简单的例子：这是一个对两个数字进行求和的函数。为了开始测试，我们首先编写测试用例，如下所示。清单 1. TestAdd.php

```
assertTrue( add( 1, 2 )
```

```
== 3 ). } function test2() { $this->assertTrue( add( 1, 1 ) == 2 ). }?>
```

这个 TestAdd 类有两个方法，都使用了 test 前缀。每个方法都定义了一个测试，这个测试可以与清单 1 一样简单，也可以十分复杂。在本例中，我们在第一个测试中只是简单地断定 1 加 2 等于 3，在第二个测试中是 1 加 1 等于 2。PHPUnit2 系统定义了 assertTrue() 方法，它用来测试参数中包含的条件值是否为真。然后，我们又编写了 Add.php 模块，最初让它产生错误的结果。清单 2. Add.php 现在运行单元测试时，这两个测试都会失败。清单 3. 测试失败% phpunit

```
TestAdd.phpPHPUnit 2.2.1 by Sebastian Bergmann.FFTime:
```

```
0.0031270980834961There were 2 failures:1) test1(TestAdd)2)
```

```
test2(TestAdd)FAILURES!!!Tests run: 2, Failures: 2, Errors: 0,
```

```
Incomplete Tests: 0.现在我知道这两个测试都可以正常工作了
```

。因此，可以修改 add() 函数来真正地做实际的事情了。现在这两个测试都可以通过了。清单 4. 测试通过% phpunit

```
TestAdd.phpPHPUnit 2.2.1 by Sebastian Bergmann...Time:
```

```
0.0023679733276367OK (2 tests)% 尽管这个测试驱动开发的例子非常简单，但是我们可以从中体会到它的思想。我们首先
```

创建了测试用例，并且有足够多的代码让这个测试运行起来，不过结果是错误的。然后我们验证测试的确是失败的，接着实现了实际的代码使这个测试能够通过。我发现在实现代码时我会一直不断地添加代码，直到拥有一个覆盖所有代码路径的完整测试为止。在本文的最后，您会看到有关编写什么测试和如何编写这些测试的一些建议。数据库测试 在进行模块测试之后，就可以进行数据库访问测试了。数据库访问测试 带来了两个有趣的问题。首先，我们必须在每次测试之

前将数据库恢复到某个已知点。其次，要注意这种恢复可能会对现有数据库造成破坏，因此我们必须对非生产数据库进行测试，或者在编写测试用例时注意不能影响现有数据库的内容。数据库的单元测试是从数据库开始的。为了阐述这个问题，我们需要使用下面的简单模式。清单 5.

```
Schema.sqlDROP TABLE IF EXISTS authors.CREATE TABLE
authors ( id MEDIUMINT NOT NULL AUTO_INCREMENT,
name TEXT NOT NULL, PRIMARY KEY ( id )).清单 5 是一个
authors 表，每条记录都有一个相关的 ID。接下来，就可以编写测试用例了。清单 6.
```

```
TestAuthors.phpassertTrue(
Authors::0delete_all() ). } function test_insert() { $this->assertTrue(
Authors::0delete_all() ). $this->assertTrue( Authors::insert( Jack ) ).
} function test_insert_and_get() { $this->assertTrue(
Authors::0delete_all() ). $this->assertTrue( Authors::insert( Jack ) ).
$this->assertTrue( Authors::insert( Joe ) ). $found =
Authors::get_all(). $this->assertTrue( $found != null ).
```

```
$this->assertTrue( count( $found ) == 2 ). }?> 这组测试覆盖了
从表中删除作者、向表中插入作者以及在验证作者是否存在的同时插入作者等功能。这是一个累加的测试，我发现对于寻找错误来说这非常有用。观察一下哪些测试可以正常工作，而哪些测试不能正常工作，就可以快速地找出哪些地方出错了，然后就可以进一步理解它们之间的区别。最初产生失败的 dblib.php PHP 数据库访问代码版本如下所示。清单 7.
```

```
dblib.phpgetMessage()). } return $db. } public static function
0delete_all() { return false. } public static function insert( $name ) {
return false. } public static function get_all() { return null. }?>对清
```

单 8 中的代码执行单元测试会显示这 3 个测试全部失败了：

```
清单 8. dblib.php% phpunit TestAuthors.phpPHPUnit 2.2.1 by
Sebastian Bergmann.FFTime: 0.007500171661377There were 3
failures:1) test_0delete_all(TestAuthors)2)
```

```
test_insert(TestAuthors)3)
```

```
test_insert_and_get(TestAuthors)FAILURES!!!Tests run: 3, Failures:
```

```
3, Errors: 0, Incomplete Tests: 0.% 现在我们可以开始添加正确访
```

问数据库的代码 一个方法一个方法地添加 直到所有这 3 个测试都可以通过。最终版本的 dblib.php 代码如下所示。清单

```
9. 完整的 dblib.phpgetMessage()). } return $db. } public static
```

```
function 0delete_all() { $db = Authors::get_db(). $sth =
```

```
$db->prepare( DELETE FROM authors ). $db->execute( $sth ).
```

```
return true. } public static function insert( $name ) { $db =
```

```
Authors::get_db(). $sth = $db->prepare( INSERT INTO authors
```

```
VALUES (null,?) ). $db->execute( $sth, array( $name ) ). return
```

```
true. } public static function get_all() { $db = Authors::get_db(). $res
```

```
= $db->query( "SELECT * FROM authors" ). $rows = array().
```

```
while( $res->fetchInto( $row ) ) { $rows []= $row. } return $rows.
```

```
}}?> 在对这段代码运行测试时，所有的测试都可以没有问题
```

地运行，这样我们就可以知道自己的代码可以正确工作了。

100Test 下载频道开通，各类考试题目直接下载。详细请访问

www.100test.com