

我们查看Post类和Comment类，它都是简单的PHP对象，内部也只有getters和setters方法，并不会依赖于其他的类，所以我们在进行单元测试时，可以直接使用new关键字来创建对象，然后进行单元测试。

但是PostFactory类它是一个服务类，服务类他提供了一些功能方法。有时它会依赖于其他的服务类。在进行单元测试时，如果我们使用了new关键字来创建了服务类，如果它有依赖的服务类，那么我们也需要把依赖的服务类使用new关键字创建出来。这样单元测试就失去了意义。

单元测试只针对一个单元功能块进行测试，比如我们测试PostFactory类，我们就单纯的测试PostFactory类。phpunit提供了一个Mock的概念，我这里叫替身，但是文档叫做仿件对象。打开浏览器搜索phpunit，查看文档，第八节测试替身，仿件对象，将对象替换为能验证预期行为的测试替身的实践方法称为模仿mocking。

在单元测试中，如果我们测试的是服务类，我们来创建一个替身，这个替身叫做仿件对象，我们可以对仿件对象验证它的预期的行为来进行测试。

修改代码，phpunit提供了一个createMock()方法，参数是服务类的类名，它的返回值就是一个mock object，一个仿件对象。现在我们创建了一个factory仿件对象。来进行一下测试，测试出错了。

它提示\$post->getStatus()，它的返回值并不是draft，现在我们的factory对象，它是一个仿件对象。仿件对象它内部的方法返回的类型也是一个仿件对象，我们在第16行输入dd()查看一下\$post对象。

再次执行单元测试，\$post类型也是一个仿建对象，它的status属性是null，所以最终测试出错了。

在下一节。我们将修改测试代码，让测试代码可以顺利通过。