目前，在 Java 阵营中主要的 Mock 测试工具有 Mockito，JMock，EasyMock 等。

# 前言

网上有许多关于单元测试的好处，这里我就不去说了。我写单元测试的理由很简单粗暴，就是图一个方便。试想一下这个场景：我们在写一个新功能，每写一部分，我们就安装到手机上查看一下，这个过程中你要点击到对应的页面，做对应的操作，最后才能反馈给你结果。如果达到了预期效果，那么恭喜你。可是一旦这次失败了，是不是又要重复这一过程？是不是感到很麻烦？很费时间？如果你想早点写完下班，那么你就需要掌握单元测试。因为它能大大的缩短你自我验证的时间。

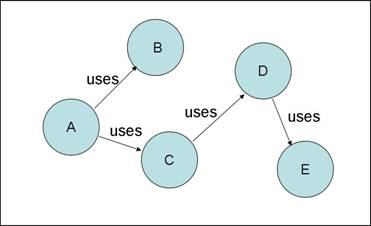
# Mockito

什么是 Mock 测试

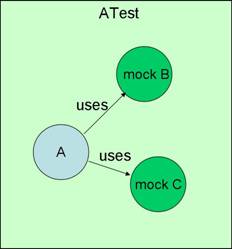
Mock 测试就是在测试过程中，对于某些不容易构造（如 HttpServletRequest 必须在Servlet 容器中才能构造出来）或者不容易获取比较复杂的对象（如 JDBC 中的ResultSet 对象），用一个虚拟的对象（Mock 对象）来创建以便测试的测试方法。

Mock 最大的功能是帮你把单元测试的耦合分解开，如果你的代码对另一个类或者接口有依赖，它能够帮你模拟这些依赖，并帮你验证所调用的依赖的行为。

比如一段代码有这样的依赖：



当我们需要测试A类的时候，如果没有 Mock，则我们需要把整个依赖树都构建出来，而使用 Mock 的话就可以将结构分解开，像下面这样：



# JUNIt4

在Android Studio中进行单元测试和UI测试

<https://www.jianshu.com/p/03118c11c199>

<https://blog.csdn.net/qq_17766199/article/details/78243176>

<https://tech.meituan.com/Android_unit_test.html>

Android单元测试——初探

<https://smallsoho.com/android/2017/04/19/Android单元测试实践/>

<https://www.jianshu.com/p/bc99678b1d6e>

Android单元测试 - 如何开始？

# androidTest

# TASK

风格识别，自动化测试

# REF

<https://waylau.com/mockito-quick-start/>

[Android单元测试(一)：JUnit框架的使用](https://blog.csdn.net/qq_17766199/article/details/78243176)