

Low level test

2018/05/01 19:19 - Posted By [起衣](#) - Tags: [Ilt](#) - Category: [Code](#)

1 条评论

开发人员最为熟悉的测试手段是单元测试，单元测试通常只关注单个类、单个方法的功能，(java)在写作单元测试用例时一般基于 Junit 框架使用 EasyMock/PowerMock 等为被测试类的依赖打桩，可以说是非常方便了，不过单元测试无法验证系统或者模块在一个业务流程上的处理逻辑是否符合预期，为了验证当前系统或者模块对整个业务流程的处理，我们就需要做集成测试了。集成测试时，我们不再只初始化单个的类，而是按照系统运行要求启动整个模块或者整个系统，然后模拟其他系统来和当前系统进行交互，并对运行结果进行检查。很明显，让一个系统完整地运行起来，需要满足非常多的外部条件，比如Web容器、数据库、缓存、消息队列、其他外部依赖服务等，想要满足这么多的条件非常不易，更何况自动化的测试用例还需要尽可能地确保用例之间的隔离，用例不同执行者之间的隔离（如果共享公共的外部环境依赖，很容易不同人在执行用例时会互相干扰）。因为集成测试外部条件的准备不易，很多项目只能放弃这一测试手段，而期待测试团队的验证。

足够多的自动化测试用例是提升产品质量的有效保证，是减少问题、减少加班性价比非常高的方式，而让开发人员自主就能完成用例编写又能极大降低问题发现、处理的成本，所以在条件允许的情况下，开发人员来编写的单元测试、集成测试用例是非常有必要的。那么集成测试是否可以参考单元测试的EasyMock等框架，为很多通用的外部环境依赖提供简单、好用的Mock能力呢？我尝试为这个事情开个坑，目标是在集成测试里成为类似于单元测试里EasyMock的存在，为大部分应用提供可以通过API进行控制的Web运行环境、数据库依赖、etcd依赖、redis依赖、cassandra依赖、kafka依赖等，让开发人员可以很低成本地开发集成测试用例，坑在这里：<https://github.com/qiyi/Ilt>，后续再慢慢介绍不同外部环境依赖的使用方式。

本文链接: <http://isouth.org/archives/367.html>，转载请注明出处，此外还可以[订阅我](#)。

相关日志 Relate Posts

- 无相关日志

收藏与分享 : [Twitter](#) | [Facebook](#) | [微博](#) | [人人](#) | [Google+](#) | [PDF](#)

“Low level test”1条留言



路易大叔: May 6, 2018 at 2:06 am
这个就比较专业了

Reply

发表留言 (Ctrl+Enter提交)

昵称 (必填)

E-Mail (不公开, 必填)

网站 (选填)



发表

themed by [white](#) 1.18 hosted by [衡天主机](#) copyright © 2017 [isouth.org](#) 20 queries in 0.415 seconds.

[回到顶端](#) ↑

