**我是如何一步一步做接口测试的一些感悟**

　　不知不觉在公司做接口测试已经接近一个月了。由于之前没做过接口测试，所以上手时走了不少弯路，而且老实说现在还在走弯路中，所以只能说是感悟吧。  
**接口测试的目的**  
　　这个算是老生常谈了，但我觉得只要聊到接口这个还是绕不过的，没有目标就没有评判标准，所以测试的目的还是很重要的。  
　　先搬运一下维基百科上的英文解释（中文没找到，百度的就算了。。。）：  
　　API testing is a type of software testing that involves testing application programming interfaces (APIs) directly and as part of integration testing to determine if they meet expectations for functionality， reliability， performance， and security.[1] Since APIs lack a GUI， API testing is performed at the message layer.[2] API testing is now considered critical for automating testing because APIs now serve as the primary interface to application logic and because GUI tests are difficult to maintain with the short release cycles and frequent changes commonly used with Agile software development and DevOps).[3][4]  
　　翻译过来就是：  
　　API 测试是一种作为集成测试的一部分，通过直接控制被测应用的接口（API）来确定是否在功能、可靠性、性能和安全方面达到预期的软件测试活动。[1]由于 API 都没有 GUI 界面，API 测试都是在通讯层进行的。[2]现在 API 测试在自动化测试中有着很重要的地位，因为 API 一般是应用逻辑的主要接口，同时 GUI 测试在敏捷开发和 DevOps 的快速迭代和频繁变更中很难维护。[3][4]  
　　[1]:Testing APIs protects applications and reputations， by Amy Reichert， SearchSoftwareQuality March 2015  
　　[2]:All About API Testing: An Interview with Jonathan Cooper， by Cameron Philipp-Edmonds， Stickyminds August 19， 2014  
　　[3]:The Forrester Wave? Evaluation Of Functional Test Automation (FTA) Is Out And It's All About Going Beyond GUI Testing， by Diego Lo Giudice， Forrester April 23， 2015  
　　[4]:Produce Better Software by Using a Layered Testing Strategy， by Sean Kenefick， Gartner January 7， 2014  
　　然后我目前在工作中 Leader 对我的预期：  
　　发现尽可能多的问题。先说下背景，他是 app 端的主程，他对我做接口测试的期待是尽早发现接口的问题，减少他们 app 和服务端联调是发现的问题。他举了个例子，例如服务器返回一个 {"width":30， "url":"<http://xxx/a.jpg>"} ，那么有可能存在这个图片的实际宽度和 width 字段不一致。  
　　和大东等曾经做过 API 测试的人的交流：  
　　覆盖业务，包括业务正常/异常的情况  
　　我一开始的理解：  
　　发现问题，恩。那就是把文档里说的必须字段缺了试试，只有必须字段试试，必须字段用非法值试试。可选字段缺了试试，可选字段非法值试试。  
　　结果就是：一个接口至少6个用例，甚至更多。而且很多时候不一定测得到业务（如接口只返回成功失败，但有可能实际上失败了但返回的是成功，需要调用其他接口验证）  
　　目前的理解（估计还是有点走偏）：  
　　结合业务设计用例。如有翻页参数，那就试试默认值、有效值、最后一页、最后一页+1、无效值。针对各个参数和业务场景去设计用例。  
**接口测试的实现**  
　　在这一点上我走了两条路：Jmeter 和纯 Java 代码。现在看来，两条都不是好路，所以就不分享了。这里主要说下 Testerhome 上最近出现的一些接口测试工具方面的帖子，简单汇总一下他们的实现方式：





**到目前为止的做过的接口测试的总结**  
　　怎么说呢，我觉得我现在做的虽然也是接口测试，但我设计的用例更多的是具体功能。例如发表朋友圈，我会调用上传接口（顺带检查上传的文件 md5 对不对，顺序对不对，相关属性字段对不对），发圈子接口（基本就检查返回值就够了，但要有不同类型的文字，包括特殊符号、长度等），查看圈子接口（图片 md5、顺序、相关属性对不对、发布人对不对、回复和点赞方面的数据对不对）。因为是三个独立的接口，所以每个接口都需要有一定的封装方法（发信息的、获取返回值里指定字段的，等）。每个封装方法平均10行代码左右，一个用例用3个接口基本就需要 3x10+10（一些说明和不好封装的部分）行代码。一个功能大概要覆盖10个用例，所以就需要10x40行代码，就是400行代码。除去一些可以共用的代码，基本需要300行代码左右（某些复杂功能会更多）。  
　　PS：通过 git 统计了一下我每天的代码量，大概在1000左右，但用例数还是每天15个左右。  
　　可以看出，我的速度主要是慢在了代码量太大，时间都用在打代码上了，而且代码调试起来又会花一定的时间，所以效率自然低。所以如果优化写用例的方式，就算吐血也写不快了。接下来得总结一下那些地方可以抽取出来，用录制或者自动生成的方法来完成，把写用例精简成填表格或者纯填数据。  
　　Updated 2015.11.30  
　　今天和 Monkey 以及公司另一位有做过 api 测试的同事交流了一下，发现了一个最根本的问题： 我的用例设计有问题  
　　这个讲概念比较难说清楚，还是以上面提到的发表朋友圈作为例子。  
　　假设我要验证发表朋友圈的 接口，它的工作流程如下：  
　　上传图片，服务器返回这些图片的 url  
　　发表朋友圈，包含朋友圈文字内容和各个图片 url 两个主要字段。服务器只会返回是否成功以及这条朋友圈的 id  
　　我设计的正常发朋友圈的用例如下：  
　　用上传图片接口上传图片，验证图片是否上传成功，各 url 对应文件的 md5 是否和我本地上传的图片的 md5 吻合。  
　　用发本地圈接口发本地圈，其中图片 url 来自第一步的返回值  
　　用查看本地圈接口查看发出的本地圈，检查它的内容、图片是否正确。  
　　咋看之下好像没啥问题（相信做过 api 测试的童鞋已经看出问题了），但其实这个用例本身存在一个严重的问题： 接口成为了手段，验证点其实是功能。  
　　我交流后得到的 api 测试用例主要应该有两类：  
　　单一接口测试。调用一个接口就是一个用例  
　　多接口的业务测试。会调用多个接口，且接口之间可能存在数据传递。  
　　api 测试最简单，并且每个接口都应该至少有一个的用例应该就是使用 默认参数调用 单个接口。重点在这里：单个。像我上面这样的用例已经不是验证发朋友圈的接口了，而是验证 发朋友圈的功能。  
　　那么发朋友圈的接口应该有什么用例呢？我按照现在的测试接口的思路重新想了一下，主要有这几个：  
　　有图片、有文字，预期返回发布成功  
　　无图片，有文字，预期返回发布成功  
　　有图片，无文字，预期返回发布成功  
　　无图片，无文字，预期返回发布失败  
　　有图片、有文子，预期返回发布成功，同时数据库中的记录符合预期  
　　每一个用例测试一个点，发布成功这个返回值是否正确由一个单独的用例来覆盖就足够了。  
　　用例确定了，那么可以看出接口测试其实有很多可以重用的点。接口测试的本质是通过测试参数的排列组合验证返回值/数据库变更是否符合预期，从而确定接口相关代码是否正确。从业务角度考虑的意思是这些排列组合不是随便想出来，而是业务中有可能出现的排列组合。  
　　这也是为什么不少接口测试用例采用 excel 表格来编写，因为参数的排列组合用表格呈现是最简单快捷的。  
　　还有一个例子说明用例越简练越好。  
　　例如列表接口有个翻页的功能。它有一个入参：lastTopicId ，返回值里有一个 isMore 的字段。lastTopicId 表示从哪个 topic 开始显示， isMore 表示是否存在下一页（1为有，0为无）。  
　　针对这个接口的其中一个用例，需要验证 isMore 字段在最后一页时是否为0。我一开始的思路是像功能那样，不断翻页，直到达到一个很大的页数或者到达 isMore 为 0 的情况。由于页数不确定，因此就算这个很大的页数设得非常大也没有十足的把握绝对不会超出这个页数。所以思路本身有问题。  
　　和 Monkey 和我的同事交流后，他指出正确的思路应该是：  
　　通过一个可信的途径（从服务器数据库直接计算，或者有一个端口告诉你最有一页的 lastTopicId 是什么），用一个步骤知道怎么一次性去到最后一页。  
**直接去到最后一页**  
　　验证 isMore 参数值  
　　有些东西开发可以很方便地给到我们，那么我们没有必要那么艰难地自己去计算出来，而是直接让开发增加一个接口让我们能直接获取到数据就好。让接口测试做得更好，开发的协助是分不开的。  
　　做好接口测试并不像之前想象得那么简单，但也没有刚开始的时候做得那么艰难。不管怎样，既然开始了，那就要想办法把它做好。接下来我会使用新的设计用例思路做剩下的接口测试用例，并修改 Java 代码让其支持使用 excel 作为用例。感悟中如果有不对的地方，欢迎大家及时提出。