目 录

测试中的"望闻问切"法

◆ 作者：流氓贵族

“望闻问切”四诊法由扁鹊发明。“四诊法”（望、闻、问、切），是中医诊病的基本方法。根据相关史籍记载，四诊法是由扁鹊发明的，当时扁鹊称之为“望色、听声、写影及切脉”。扁鹊在诊视疾病中，通过望诊（看看他的脸色等）、闻诊（听听病人最近做了什么事情后生病）、问诊（问问有没有干导致生病的一些事情）和切诊（看他的脉搏）。这些诊断技术，体现在史书所记载他治病的案例中。他精于望色，即通过望色，来判断病证及其病程演变和预后。

“望闻问切”对于人们来说并不陌生，即使不是医学出身的人，对于这种方法也是略知一二。“望闻问切”的方法是用于诊断人的身体疾病的，那么可以用类似的方法来测试软件系统寻找 bug 吗？我觉得，虽说测试软件系统和给人诊病看起来谬之千里，类比着来看也可以说是差之毫厘。

“望闻问切”四诊法

1、四诊法之望诊——分析需求，确认测试用例

望诊，是对病人的神、色、形、态、舌象等进行有目的的观察，以测知内脏病变，中医通过大量的医疗实践，逐渐认识到机体外部，特别是面部、舌质，舌苔与脏腑的关系非常密切。如果脏腑阴阳气血有了变化，就必然反映到体表。正如《灵枢·本脏篇》所说：“视其外应，以知其内脏，则知所病矣。

所谓“望诊”，就是观察病人的神、色、形、态的变化。“神”是精神、神气状态；“色”是五脏气血的外在荣枯色泽的表现；“形”是形体丰实虚弱的征象；“态”是动态的灵活呆滞的表现。这就是对病人面目、口、鼻、齿、舌和苔、四肢、皮肤进观察，以了解病人的“神”。

现在的软件繁多，几乎每天都有新出版的软件，软件之间的竞争也是日益激烈，这

时候软件无明显 bug，潜在 bug 少已经不再是软件能够脱颖而出的条件了，那只是软件能够推向市场并被用户接受的基本条件，因此软件开发公司只能想对策来解决已经面临或是即将面临的问题。仅仅简单的业务已经满足不了用户了，软件开发公司纷纷在附加功能上做文章，在基础功能的基础上增添能够吸引用户关注的功能。此时，通用的测试用例便不能满足测试的标准，测试人员需要在确认需求后，分析新增功能的测试点，增加测试用例，给软件量身定做一套测试用例以达到测试标准。

个人观点，测试用例可以从实现功能的角度和用户体验的角度，分别使用边界值法、等价法、错误猜想法等测试方法设计测试用例。

2、四诊法之闻诊——执行测试用例，设计测试数据找出 bug

闻诊，包括听声音和嗅气味两个方面。主要是听患者语言气息的高低、强弱、清浊、缓急……等变化，以分辨病情的虚实寒热。

所谓“闻诊”，是指听病人说话的声音、呼吸、咳嗽、呕吐、呃逆、嗳气等的声动，还要以鼻闻病人的体味、口臭、痰涕、大小般发出的气味。

测试的基本职能就是找出系统中存在的 bug。通常测试人员会首先执行预先制定好的测试用例、设计测试数据，找出系统中的 bug 确保系统中没有严重级别的 bug，不会影响系统的正常使用，保证通用功能的功能性、安全性、易用性等方面。

这里要注意，不是执行完预先制定好的测试用例就可以的。由于需求多种多样，功能众多，在使用软件的时候会遇到许许多多的思考不到的路径，尽管在需求分析阶段我们已经充分的考虑了软件使用场景路径的可能性，我们必须承认一定有我们所遗漏的测试点，当然测试并不是保证系统没有任何的 bug，在这一点上即使是最有经验的测试工程师也不敢说经他手测试过的系统没有任何的问题。基于这样的事实上，我们就需要随时的补充测试用例，执行补充测试用例，测试软件，力求软件在其服役期无限趋近于无可遇见的 bug。

3、四诊法之问诊——分析 bug 产生原因，寻找类似 bug

问诊，是通过询问患者或其陪诊者，以了解病情，有关疾病发生的时间、原因、经过、既往病史、患者的病痛所在，以及生活习惯、饮食爱好等与疾病有关的情况，均要通过问诊才能了解，故问诊是了解病情和病史的重要方法之一，在四诊中占有重要的位置。

所谓“问诊”，就是问病人起病和转变的情形，寒热、汗、头身感、大小便、饮食、胸腹、耳、口等各种状况。

一个优秀的测试工程师不仅仅在于能够测试出 bug，还要注意训练和加强对于 bug 的敏感度，也就是能够遇见 bug 的存在，这种方法在测试的理论中叫做错误推断法，错误推断法并不是胡乱的猜测而是根据测试工程自身的经验和对于整个系统业务流程的熟悉程度作出的 bug 预测和推断。这也就是为什么测试工程师职业是一个越老越吃香的职业。

我们在测试的过程中不要仅仅止步于表面，测出 bug 就视为大功告成，测试的宗旨在于尽可能找出系统中存在的 bug，而不是保证系统毫无 bug，在这一点上没有哪一个测试工程师敢于这样说，当让也不会有测试主管以测试工程师测试过的系统必须无 bug 作为断定测试工程师好坏的标准。

俗话说实干不如巧干，一个 bug 的产生一定是有其自身的原因的，或是编程技术方面的原因；或是业务方面的逻辑原因；或是不同的编程语言差异以及和环境、平台、浏览器等的不兼容的原因。测试出 bug 后，要养成分析 bug 的习惯，分析 bug 的产生原因和出现的预制条件，再在类似的场景测试看有没有同种类型的 bug 存在。这种方法也可以在回归测试中使用，可以有效的提高测试的效率。

4、四诊法之切诊——总结 bug，更新测试用例库

切诊是指用手触按病人身体，藉此了解病情的一种方法。切脉又称诊脉，是医者用手指按其腕后挠动脉搏动处，借以体察脉象变化，辨别脏腑功能盛衰，气血津精虚滞的一种方法。正常脉象是寸、关、尺三部都有脉在搏动，不浮不沉，不迟不数，从容和缓，柔和有力，流利均匀，节律一致，一息搏动四至五次，谓之平脉。

所谓“切诊”，就是脉诊和触诊。脉诊就是切脉，掌握脉象。触诊，就是以手触按病人的体表病颁部分，察看病人的体温、硬软、拒按或喜按等，以助诊断。

测试工作中最为重要也最容易被忽视的就是总结。“温故而知新，可以为师矣”是出自孔子的《论语》，意思是：“温习旧知识从而得知新的理解与体会，凭借这一点就可以成为老师了。不要求做到“可以为师矣”，但一定要做到“温故而知新”。在测试的执行程度达到可以结束的标准之后，要养成总结 bug 的习惯，及时更新测试用例库，为测试工作留下宝贵的工作文档，更要为以后继续开展测试工作打好基础。

四诊法中的 4 种诊疾方法，不是孤立存在的，是相互联系的，是阴阳五行、藏象经

络、病因病机等基础理论具体运用。测试中的四诊法也一样，四种方法要合理的关联运用，不要教条死板的准守，可以根据实际的状况适当的增减。Bug 发现的阶段越早，修复 bug 的成本越低，所以我们力求将 bug 消灭在萌芽阶段，需求分析在这里就尤为的重要，当然测试的其他阶段也是不能忽视的。

LoadRunner 脚本编写介绍之 HTTP 篇

◆搜狗测试：曹承臻

做过 loadrunner 性能测试的同学都用过脚本录制功能，但是在脚本录制时经常会遇到以下问题:

- 无法掉起 IE 浏览器
- 调起 IE 后，action 事件始终为 0
- 调起 IE 后，窗口卡死
- 升级 IE 或更换 IE 版本后，不能录制了。
- ……

这些情况在网上很容易搜到解决方案。例如:

- LoadRunner 录制自身的缺陷。
- LoadRunner 支持 IE 版本的局限。
- 系统默认浏览器的设置错乱。
- 安全软件拦截。
- IE 插件设置。
- 系统兼容。
- ……

但是按照网上操作下载，你会发现，没有什么卵用，没有一个可以解决问题的~

鉴于此，古人云，靠人不如靠己，既然不能录制，那就自己写脚本。

(一)准备工作:

1) 熟悉抓包工具使用（这里推荐 httpanalyzer）。

由于 httpanalyzer 是基于 hook 抓包，loadrunner 是基于代理录制，这样不会冲突，loadrunner 回放时，httpanalyzer 可以抓到回放请求。

2) 了解自己测试的需求，页面测试还是接口级测试？

这决定函数参数为 http 还是 html。

3) 了解 server 想要的东西是什么？

这决定服务器是否会处理发上去的请求。如果不符合请求，如没有 cookie，UA 不对，这样服务器就不会处理，达不到打压的目的。

4) 了解 LoadRunner 使用的脚本语言基本格式。

参数、变量转换，转义符等。

5) 了解 LoadRunner 的几个基本函数。

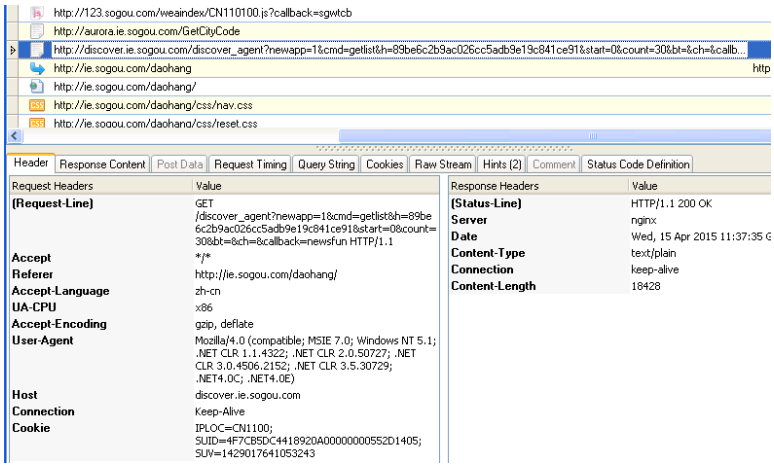
- Get 请求用 web_url
- Post 请求用 web_custom_request

6) 形成自己查找 LoadRunner 函数及其使用方法的习惯。

Loadrunner 的函数帮助文档很完善，每个函数都有使用使用实例，所以养成自己查找的习惯，可以省去打扰别人的麻烦。

(二)录制请求:

1) 打开 httpanalyzer，点击开始录制按钮，开始操作要录制的应用。



2) 找到要打压的请求或请求组，查看 request headers 信息。

如果是 get 类型的请求，可以使用 web_url 函数来实现，不同参数的取值参照抓包请求的字段。

(三)如果是 post 类型的请求，可以使用 web_custom_request 函数来实现，post 的内容需要写在 body 参数中。如果上传的是文件或二进制，需要使用 bodyfilepath 参数，将文件路径引入进去。

```
web_url("down.jsp",
//   "URL=http://download.ie.sogou.com/se/sogou_explorer_6.0_1104.exe",
  "URL=http://download.ie.sogou.com/se/Yidian_3.0_1009.exe",
  "Resource=1",
  "RecContentType=application/octet-stream",
  "Referer=",
  "Snapshot=t3.inf",
  LAST);
```

Post Josn

```
"URL=http://www.sogou.com",
"Method=POST",
"Resource=0",
"RecContentType=application/json",
"Referer=",
"Mode=HTTP",
"EncType=application/json",
"Body={\"user\":{\"uid\":\"C8-9C-DC-70-BD-B2\"}}",
LAST);
```

Post 文件

```
web_custom_request("recv_p2pcrash.php",
    "URL=http:[illegible]'recv_p2pcrash.php?software=se&ver=5.1.7.4145",
    "Method=POST",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=http://setest.cn/AutoForm/post-crash.html",
    "Snapshot=t24.inf",
    "Mode=HTTP",
    "BodyFilePath=C://Documents and Settings//Administrator//Local Settings//Temp//setask.zip",

    LAST);
```

学习更多 LoadRunner 脚本开发技术，推荐教程 http://www.atstudy.com/course/27

我的自动化历程
--我对自动化测试的一些理解和认识

◆ 作者：刘琛梅

在换了一家新工作之后，新公司的项目组也要开始做自动化了。这段时间做了一些基本的自动化工作，让我有些感触，也再次印证了我之前的一些观点。

我是自动化测试坚决的簇拥者。记得我刚做测试，第一次接触自动化测试，就觉得好神奇，当时就把手头的工作都自动化了——不过那时所谓的自动化，也就是写尽量写一些脚本来帮我操作被测对象，然后自己还要去检查测试结果对不对。尽管很 low，但也给了初出茅庐的我巨大的成就感，此后我还养成个习惯，凡是遇到繁琐的测试任务，或是管理任务，我就思考能不能写几个脚本尽量自动化完成。

后来，在第一家公司，我就开始向各位自动化前辈（本部门，外部门）学习，并开始着手搞自动化，我自己包含热情的搞了至少有三大轮，但不避讳的说，全失败了。当然，最后我们团队的自动化还是做得不错的，但是在我同事，加上一个自动化团队的带领下搞出来的（这也是我很钦佩的牛人之一）。我们达到了“工厂化”的水平，能够支持所有的功能，回归，转测试，发布测试的工作。

这期间我也不断在总结我自动化失败的原因。从我们一听到自动化这个词开始，就觉得这项技术特别牛逼，不吝给它各种赞美之词，是提高测试效率的银弹。但是，自动化真的没有你想的那么好，自动化本身的投入也是巨大的。我第一次自动化失败的原因，就是没有意识到它的投入--自动化应该是一个团队的行为，领导需要给它留工时，团队成员也需要有一定的时间投入，靠员工加班，靠员工的激情是做不成自动化的（当然，写几个脚本还是可以，如果你认为几个脚本就是自动化了除外）

从意识到自动化需要高投入开始，我就开始思考效率和效益这个问题，也就是自动化的策略。我们是应该追求自动化率，还是该追求别的什么？这也是我第二次的失败经历：我没有想清楚自动化的目的，又要追求效率和收益，于是我追求自动化率，在这种情况下，必然是简单的内容先自动化，于是出现了大量的测边界值的脚本。这样的自动化率是上去了，但是效果呢？

那次自动化经历还暴露了别的问题，就是脚本的适应性问题，开发的参数，命令稍有变化，我需要改大量的脚本，当你率还不错的时候，真是一场灾难，有种自己挖坑自己跳的感觉。

有了这次弱爆了的失败经历，开始重视自动化的目标和策略：目前你定位自动化是回归测试，还是功能测试还是别的什么测试？然后优先开发这部分脚本，而不是管它三七二十一先做起来。另外，追求率是没有意义的，测得爽才是真的好。

还有一个经验就是，自动化也需要设计，也需要规范，需要框架。

有了这些教训，第三轮的自动化实践看起来就像样多了，也慢慢有了效果，但在运行过程中，我又发现了一个致命的问题：自动化脚本误判！跑出来的结果明明是 pass，但实际上是失败的！omg！甚至还出了网上问题！

这段时间我也读了大量的测试书，记得之在一本当时很流行的测试书中也看到这样的问题，他们的解决方法是记录整个测试过程，但是这又引入新的问题，内容太多，分析不完。我对书中的做法表示怀疑，觉得这还算自动化么。

带着问题去思考和学习总是特别有效。公司中牛人很多，和前辈们讨论交流，发现原来这些问题，是可以通过写好自动化的检查函数来改善甚至避免的。这让我认识到自动化的难点不是让脚本模拟测试者的操作，能够运行起来，而是在 check。很多人都喜欢把自动化测试比作“机器人”。自动化测试中模拟测试者的操作，是这个自动化机器人的“手”，而 check 就是机器人的“大脑”。check 没有做好，自动化就不够可靠，做也是白做。有同学认为单元测试和接口测试不用太关心对 check 的设计，我认为这也是不正确的。Check 的设计对 UI 和 CLI 自动化会比单元测试和接口测试更为重要一些，仅此而已。

自动化不是个人行为，要让一个团队每个人都能快速写好自动化，把 check 做到位，是自动化管理的难点。一个经验就是，针对业务特点来总结有哪些 check 类型，然后对这些类型来封装函数，让大家就可以根据用例的情况来用这些函数。

纪录测试过程也是需要的，对可能的测试结果分级，设计各种全局调试开关，做出分层级的测试结果报告。除了“成功”和“失败”的状态，还可以加入一个“怀疑”的状态，总结测试时的定位手段和思路，让脚步可以有针对性的抓取更多的定位信息，而不是一出现问题就只有重跑一遍脚本，这不仅提高了自动化测试的效率，还可以提升产品的可测试性水平。

把这些都做好后，自动化就变的舒服多了。后来我的实践还证明，做好 check 的设计还是提高 UI 和 CLI 自动化测试率的方法。自动化测试走上正轨后，我们又开始思考各种小改进，比如自动回填结果，自动生成脚本等等。

随着自动化测试进入正轨，团队也开始尝到自动化测试的甜头，自动化测试率也开始稳步上升。但这时又开始出现了新问题——我们发现自动化率到一定程度后，自动化率就很难上去了，就像减肥中遇到的平台期一样。另外还有个问题，就是和手工测试相比，自动化能够执行的用例比较少，但占用的资源量却很大，执行效率不高。

我们对当时团队的状况进行分析：自动化率上不去，是限于流程，是因为对新功能来说，无论是 UI 和 CLI 还是接口自动化，都无法做到项目一开始就自动化。开发对 UI 和 CLI 和接口确定的比较晚，而且修改还很随意。新功能在项目中，可以自动化的时间很短，最后到项目结束，新功能特别是 UI 和 CLI 的自动化，也只能做到一些基本功能的测试，只能满足冒烟要求。自动化测试主要还是针对稳定的老功能的回归测试。但在没有自动化的时候，我们并不会执行那么多老功能的回归测试。当时团队有些同学也对这种为什么要占用这么多资源，花这么多功夫进行这么大规模的回归测试提出了质疑，并提出应该想办法提高新功能测试的自动化率的诉求。

自动化测试占资源多的原因和我们的产品特点有关，我们的产品业务需要的部署比较复杂，有些用例需要控制多个设备来进行。我们当时还没有做到让脚本在不同的部署上跑，也无法在一个测试集中引入不同的测试部署环境。

这时领导的巨大能力就体现出来了。对第一个问题，在领导的支持下，我们修改了研发流程改了，要求项目先设计出 CLI 和 web，先确定出 CLI 的回显和 web 的显示，严格评审后就开始自动化，在项目中不得随便修改自动化，就算要修改也需要通知测试团队。这大大增加了自动化的可编写时间，让自动化可以做新功能的测试了，自动化变得更实用了。为了更好的自动化，我们对测试用例的设计也做了调整，使其更适合自动化。

第二个问题，我们引入测试床的技术，在一个床上可以包容多套部署。此后自动化开始工厂化运作，有专门的环境，可以 7*24 小时不间断跑。我们也像产品开发一样来自动化脚本，用 svn 来管理脚本，评审脚本，不断重构和优化自动化框架和脚本，自动化也运作的越来越好了。

这段经历让我认识到，自动化其实不是测试的单方面能搞定的事情，需要领导支持，还有开发的理解和配合。自动化的框架确实非常重要，但是一个牛逼的框架，不是因为技术有多新多牛，而是能够切实的解决问题。

测试女巫之控制鼠标键盘篇

◆作者：王平平

一、前言:

上一期我们介绍了 Pywinauto 此模块，根据此模块的学习我们可以实现自动化控制运行在 Windows 上的应用程序。这次我们再以 PyUserInput 为例，详细地说明通过这个模块如何控制我们最常用的鼠标和键盘，尤其是包含在此模块中的各个函数如何模拟我们常用的鼠标和键盘。虽然在第四十二期和第三十三期中有介绍了控制 Windows 和控制浏览器的模块，看样子是可以实现很多我们工作上的自动化，但是我们还是需要学习一些辅助性的模块化的学习，例如模拟鼠标和键盘的模块 PyUserInput，因为有时有的操作，使用 Selenium 或者 Pywinauto 操作起来很复杂，或者很难操作，反而使用鼠标或者键盘操作更简单。

这个 PyUserInput 相对于 Selenium 或者 Pywinauto 简单很多，所以这期内容的学习将是一趟轻松之旅~

所以大家一起启动学习模式，一起为改变枯燥的工作努力吧！

二、第一阶段：工作需求

学习必须要有理由，这个问题在上一期已经做了详细说明，所以同样学习 PyUserInput 也要有充足的工作需求才能说服老板。

所以对于 PyUserInput 的“工作需求”，女巫总结如下:

1) 在测试一些 Module 的项目，需要调用 dos 对话框，如果使用 pywinauto 这个模块调用 dos 对话框，会很复杂。因为换一个思路，我们模拟鼠标键盘来调用 dos 对话框，工作会简单需要，会大大提高工作效率。

2) 在测试路由器的 WIFI 连接性能测试时，需要进入 Windows 的控制面板->网络和 Internet->网络连接，如果需要使用 pywinauto 也会非常复杂，而且等你花费很大

的力气去学习如何使用 pywinauto 实现上述的功能会发现，根本无法实现。同样换一个思路，我们通过模拟鼠标和键盘，工作逻辑会减化很多，也会大大提供我们的工作效率

三、第二阶段：PyUserInput 安装步骤说明

PyUserInput 的本质是通过先通过 pymouse 和 pykeyboard 的函数模拟人的操作，将人力的操作抽象成一个个的函数，通过这些函数来实现自动化控制鼠标和键盘

Pymouse 和 Pykeyboard 顾名思义是用于模拟鼠标和键盘的模块。Pymouse 和 Pykeyboard 之前是两个模块，目前已经合并为一个模块即 PyUserInput。

1. 配置开发环境安装说明

1) PyUserInput 官网

https://pypi.python.org/pypi/PyUserInput

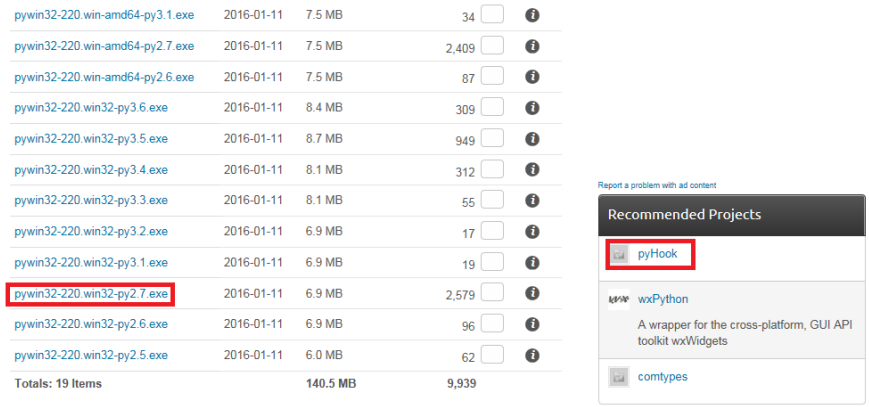
注意: 首先你要明确你的操作系统是什么，对于 Mouse 是可以同时应用在 Windows，Linux 以及苹果系统中，对于 Keyboard 只能应用在 Windows 和 Linux 系统中，苹果系统还在开发中。最重要的是要了解在不同的系统中，PyUserInput 此模块在安装前需要安装的模块是什么。对于 Windows 操作系统，需要先安装 pywin32 和 phHook 此两个模块。

2) Pywin32 官网

https://sourceforge.net/projects/pywin32/files/pywin32/

注意：pywin32 要与你当前安装的 python 版本一致，且需要与 PC 的操作系统保持一

致如下图：我们安装的 Python 版本是 Python27，还要注意，官网上也给你推荐了 pyHook 的链接。



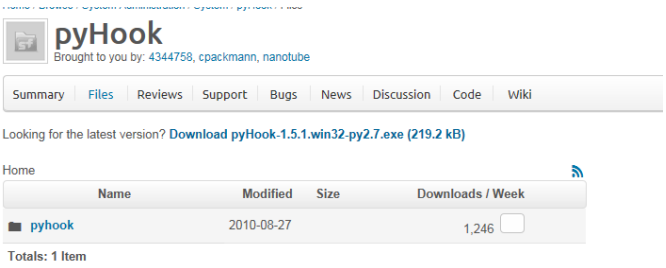
pywin32-220.win-amd64-py3.1.exe	2016-01-11	7.5 MB	34	
pywin32-220.win-amd64-py2.7.exe	2016-01-11	7.5 MB	2,409	
pywin32-220.win-amd64-py2.6.exe	2016-01-11	7.5 MB	87	
pywin32-220.win32-py3.6.exe	2016-01-11	8.4 MB	309	
pywin32-220.win32-py3.5.exe	2016-01-11	8.7 MB	949	
pywin32-220.win32-py3.4.exe	2016-01-11	8.1 MB	312	
pywin32-220.win32-py3.3.exe	2016-01-11	8.1 MB	55	
pywin32-220.win32-py3.2.exe	2016-01-11	6.9 MB	17	
pywin32-220.win32-py3.1.exe	2016-01-11	6.9 MB	19	
pywin32-220.win32-py2.7.exe	2016-01-11	6.9 MB	2,579	
pywin32-220.win32-py2.6.exe	2016-01-11	6.9 MB	96	
pywin32-220.win32-py2.5.exe	2016-01-11	6.0 MB	62	
Totals: 19 Items		140.5 MB	9,939	

我的操作系统是 32 位所以我们需要下载上图中红框标出的版本。所以需要确认你到底需要 download 哪个版本需要同时确认上述两个信息。

3) pyHook 官网

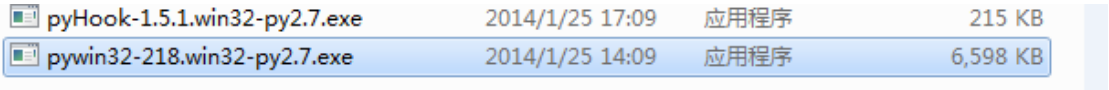
https://sourceforge.net/projects/pyhook/files/

注意：pyHook 并没有根据 Python 版本和操作系统的不同而不同，所以直接下载即可。

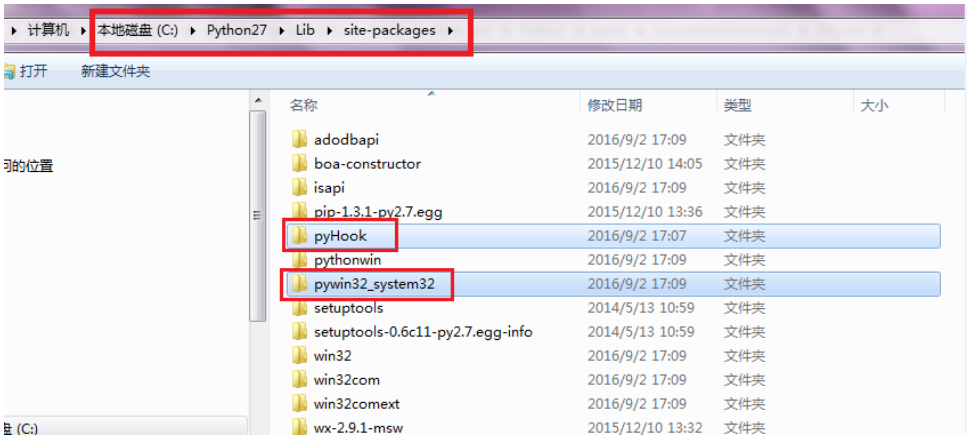


2. 安装步骤

1) Pywin32 和 PyHook 下载后如下图：即全是 exe 文件，直接双击，点击下一步即可。

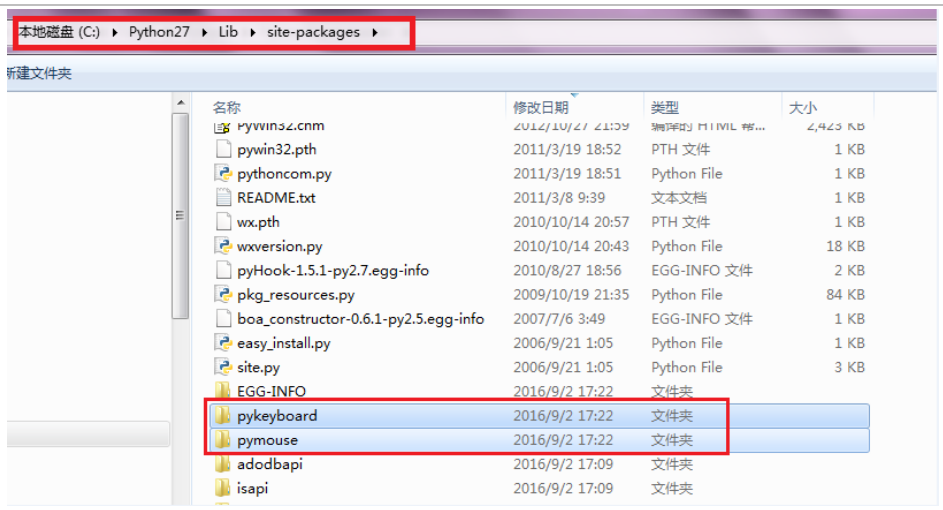


安装完毕后可以到此路径下找到 pyHook 和 pywin32 的文件夹，就说明已经安装成功。



2) PyInput 下载后如下图，安装步骤与 selenium 和 pywinauto 一样的安装步骤，这里就不再赘述。

安装完毕后可以到此路径下找到 pykeyboard 和 pymouse 的文件夹，就说明已经安装成功如下图：



四、第三阶段：如何模拟鼠标的基本动作

无论学习鼠标还是键盘甚至 Selenium，Pywinauto 学习资料的来源永远都是官网和源代码，官网的主要作用是：学习官网给你传递的学习这个 Module 的学习逻辑；而源代码的作用是：源代码中涵盖了所有函数的说明。所以如果想根据需求找寻函数来实现这个需求，可以在源代码中寻找。

所以一定要根据不同的需求进行选择你的学习资料。

1. 如何创建 mouse 和 keyboard 的对象

资料来源是官网：https://pypi.python.org/pypi/PyUserInput

How to get started

After installing PyUserInput, you should have pymouse and pykeyboard modules in your python path. Let's make a mouse and keyboard object:

```python
from pymouse import PyMouse
from pykeyboard import PyKeyboard

m = PyMouse()
k = PyKeyboard()
```

2. pymouse 学习资料介绍

1) 官网

只给出了两个函数的例子：即 screen_size()即得到屏幕的 size 以及 click()函数

```
x_dim, y_dim = m.screen_size()
m.click(x_dim/2, y_dim/2, 1)
k.type_string('Hello, World!')
```

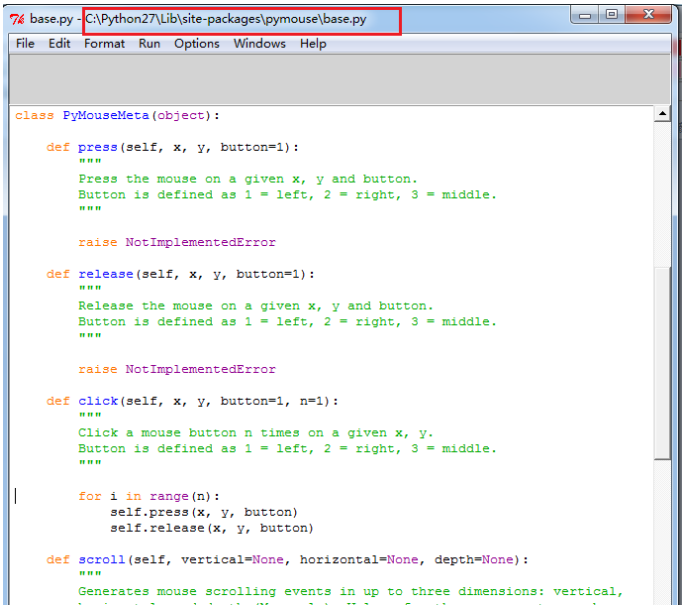
2) 源代码:

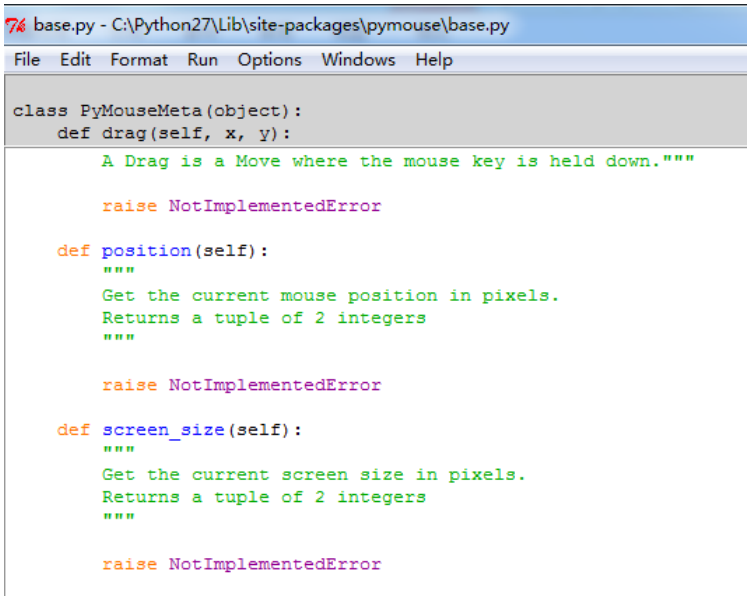
可以看出有更多的函数:

例如: 按下鼠标按键: press()。

松开鼠标: Release()等, 大家可以根据自己的需求去选择函数。

从我的经验看, 用得做多的函数是: position(): 即获得当前鼠标所在的坐标。click(self, x, y, button=1, n=1), 其中 x, y 为希望点击的坐标, button 为 1 为点击左键, 2 为点击右键, 3 为点击中键。





3. 函数使用说明

1) position()

得到当前鼠标的位置坐标

例如:

m=PyMouse()

#实例化 PyMouse 类为 m

a=m.position()

#使用 Position 函数得到当前鼠标的位置坐标

print (str(a))

#将此函数返回的结果变为 string，并打印出来

最后得到的结果如下:

<64,70>即当前鼠标的坐标

2) move(x, y)

将鼠标移到坐标为(x,y)的位置

例如:

m.PyMouse()

#实例化 PyMouse 类为 m

m.move(230,343)

#移到坐标为(230,343)的位置

3) click(x, y, number)

将鼠标移到并点击坐标为(x,y)的位置

如果 number 为 1 则为鼠标左键点击

如果 number 为 2 则为鼠标右键点击

如果 number 为 3 则为鼠标中键点击

例如:

m.PyMouse()

#实例化 PyMouse 类为 m

m.click(230,343,1)

#移到坐标为(230,343)的位置并点击鼠标左键

m.click(230.343,2)

#移到坐标为(239,343)的位置并点击鼠标右键

m.click(230.343,3)

#移到坐标为(239,343)的位置并点击鼠标中键

4) screen_size()

得到当前电脑设置的分辨率

例如:

m.PyMouse()

#实例化 PyMouse 类为 m

a=m.screen_size()

#得到当前电脑的分辨率

print (str(a))

#将此分辨率打印出来

最后得到的结果如下:

<1366, 768>

5) press(x,y)

长压坐标为 x,y 位置的区域

例如:

m.PyMouse()

#实例化 PyMouse 类为 m

m.press(230,343)

#长压坐标为 230,343 的区域

6) release(x,y)

释放坐标为 x,y 位置的区域

例如:

m.PyMouse()

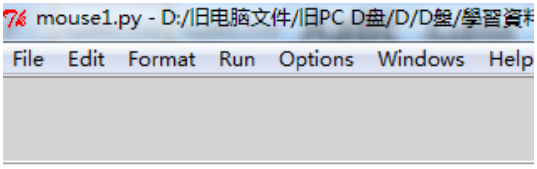
#实例化 PyMouse 类为 m

m.release(230,343)

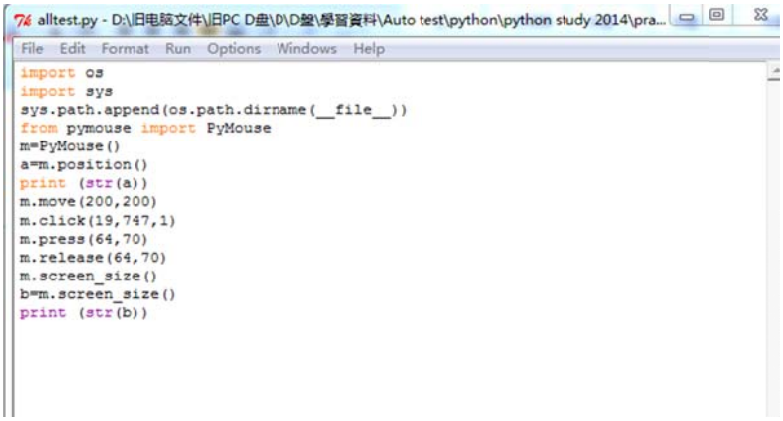
#释放坐标为 230,343 的区域

4. 例子

比较特殊的是：在运行这个脚本前需要先把鼠标移到你希望点击的位置，m.click(14,754,1)，其中 14 和 754 就是根据 m.postion()这个函数得到，而 m.postion()获得的参数是通过得到当前鼠标的位置得到。




```
from pymouse import PyMouse
import time
m=PyMouse()
a=m.position()
print (str(a))
m.click(14,754,1)
```



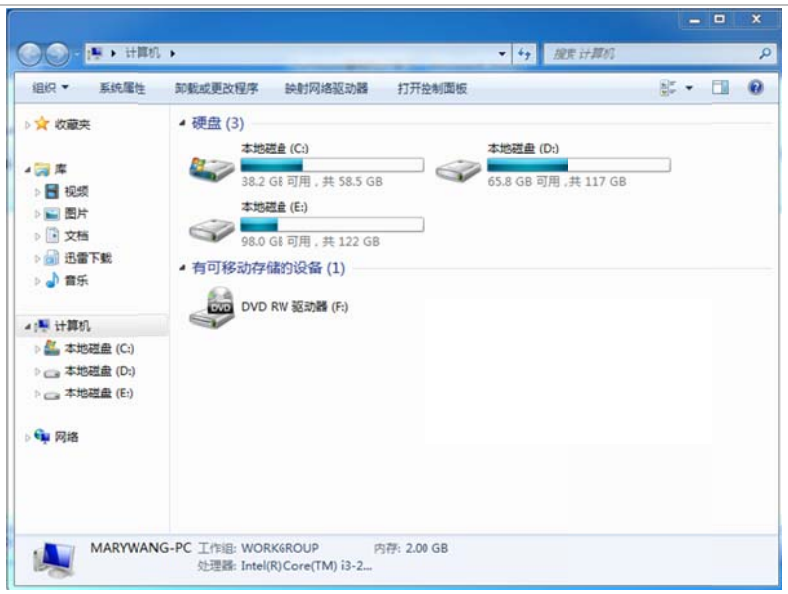
5. 问题解决

1) 问题描述

Windows 桌面的标签，例如：计算机，网络等，必须点击两次才能进入其文件夹中，但是 pymouse 提供的函数没有双击的功能，只有单击的功能，如何实现进入计算机或者网络这样的需求？

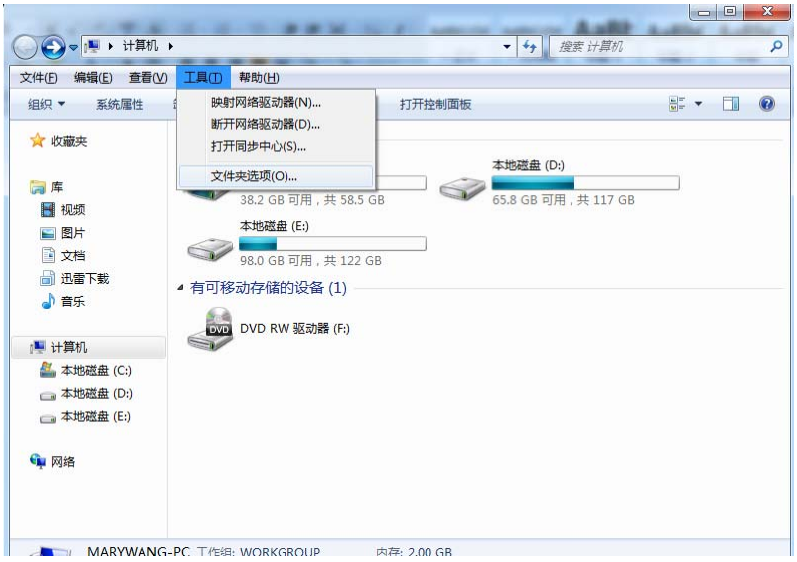
2) 问题解决

既然 pymouse 端无法解决此问题，应该考虑从 windows 端来解决此问题：进入计算机如【图 3】



【图 3】

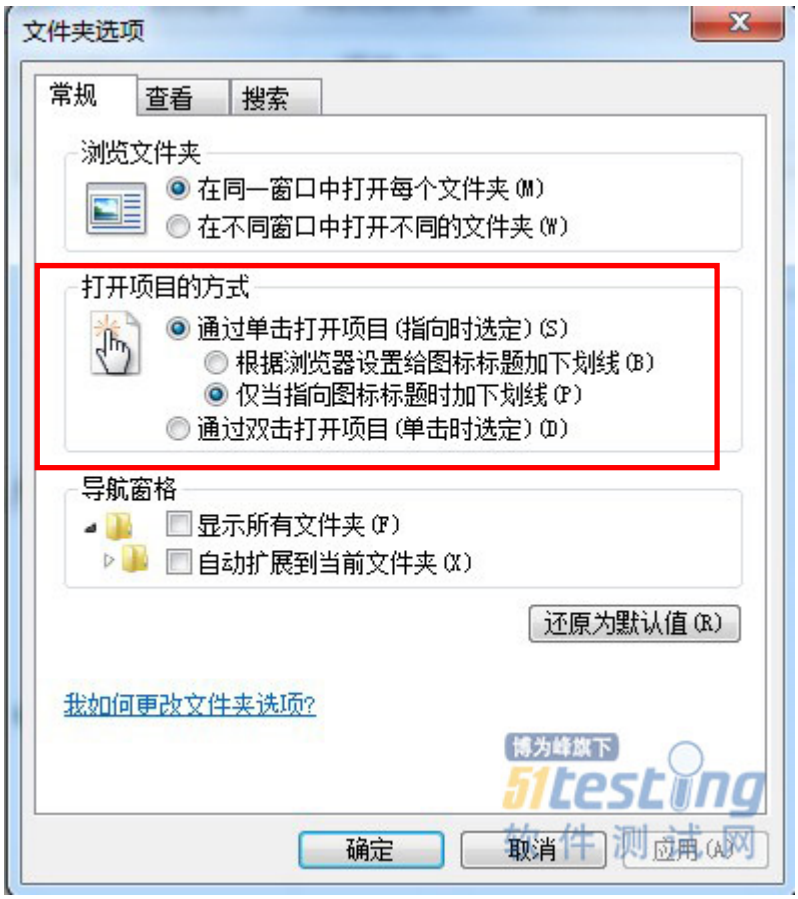
点击 Alt+F 出现如【图 4】，点击工具选择“文件夹选项”



【图 4】

出现如【图 5】，选择“通过单击打开项目(指向时选定)”

点击确定



【图 5】

按照上述步骤即可实现通过单击打开 windows 中所有项目，包括文件夹，或者一些文档(例如 Word, Excel 等)

例子，进入计算机界面

```
import os
import sys
import time
sys.path.append(os.path.dirname(__file__))
from pymouse import PyMouse
from pykeyboard import PyKeyboard
m=PyMouse()
a=m.position()
print (str(a))
time.sleep(1)
m.click(25,37,1)
```

五、第四阶段：如何模拟键盘的基本动作

1. pykeyboard 学习资料介绍

1) 官网

官网上的信息分两类，第一类是点击键盘的普通 key 例如：字母或者一个字串；第二类是如何点击键盘上的 special key（组合键或者数字键盘，Home_key 等）

```python
# pressing a key
k.press_key('H')
# which you then follow with a release of the key
k.release_key('H')
# or you can 'tap' a key which does both
k.tap_key('e')
# note that that tap_key does support a way of repeating keystrokes with a interval time between each
k.tap_key('l',n=2,interval=5)
# and you can send a string if needed too
k.type_string('o World!')
```

and it supports a wide range of special keys:

```python
#Create an Alt+Tab combo
k.press_key(k.alt_key)
k.tap_key(k.tab_key)
k.release_key(k.alt_key)

k.tap_key(k.function_keys[5]) # Tap F5
k.tap_key(k.numpad_keys['Home']) # Tap 'Home' on the numpad
k.tap_key(k.numpad_keys[5], n=3) # Tap 5 on the numpad, thrice
```

2) 源代码：可以看出有更多的资讯

常用函数的源代码：

base.py - C:\Python27\Lib\site-packages\pykeyboard\base.py

File Edit Format Run Options Windows Help

```
class PyKeyboardMeta(object):
    def press_key(self, character=''):
        """Press a given character key."""
        raise NotImplementedError

    def release_key(self, character=''):
        """Release a given character key."""
        raise NotImplementedError

    def tap_key(self, character='', n=1, interval=0):
        """Press and release a given character key n times."""
        for i in xrange(n):
            self.press_key(character)
            self.release_key(character)
            time.sleep(interval)

    def type_string(self, char_string, interval=0):
        """A convenience method for typing longer strings of characters."""
        for i in char_string:
            time.sleep(interval)
            self.tap_key(i)

    def special_key_assignment(self):
        """Makes special keys more accessible."""
        raise NotImplementedError

    def lookup_character_value(self, character):
        """
        If necessary, lookup a valid API value for the key press from the
        character.
        """
        raise NotImplementedError
```

如何引用一些常用的特殊按键

windows.py - C:\Python27\Lib\site-packages\pykeyboard\windows.py

File Edit Format Run Options Windows Help

```
class PyKeyboard(PyKeyboardMeta):
    def special_key_assignment(self):
        Special Key assignment for windows
        """
        #As defined by Microsoft, refer to:
        #http://msdn.microsoft.com/en-us/library/windows/desktop/dd375731(v=vs.85).aspx
        self.backspace_key = VK_BACK
        self.tab_key = VK_TAB
        self.clear_key = VK_CLEAR
        self.return_key = VK_RETURN
        self.enter_key = self.return_key  # Because many keyboards call it "Enter"
        self.shift_key = VK_SHIFT
        self.shift_l_key = VK_LSHIFT
        self.shift_r_key = VK_RSHIFT
        self.control_key = VK_CONTROL
        self.control_l_key = VK_LCONTROL
        self.control_r_key = VK_RCONTROL
        #Windows uses "menu" to refer to Alt...
        self.menu_key = VK_MENU
        self.alt_l_key = VK_LMENU
        self.alt_r_key = VK_RMENU
        self.alt_key = self.alt_l_key
        self.pause_key = VK_PAUSE
        self.caps_lock_key = VK_CAPITAL
        self.capital_key = self.caps_lock_key
        self.num_lock_key = VK_NUMLOCK
        self.scroll_lock_key = VK_SCROLL
        #Windows Language Keys,
        self.kana_key = VK_KANA
        self.hangeul_key = VK_HANGEUL # old name - should be here for compatibility
        self.hangul_key = VK_HANGUL
        self.junjua_key = VK_JUNJA
        self.final_key = VK_FINAL
        self.hanja_key = VK_HANJA
```

如何引用 numpad 以及 FX 上的特殊按键

```
self.keypad_keys = {'Space': None,
                    'Tab': None,
                    'Enter': None,  # Needs Fixing
                    'F1': None,
                    'F2': None,
                    'F3': None,
                    'F4': None,
                    'Home': VK_NUMPAD7,
                    'Left': VK_NUMPAD4,
                    'Up': VK_NUMPAD8,
                    'Right': VK_NUMPAD6,
                    'Down': VK_NUMPAD2,
                    'Prior': None,
                    'Page_Up': VK_NUMPAD9,
                    'Next': None,
                    'Page_Down': VK_NUMPAD3,
                    'End': VK_NUMPAD1,
                    'Begin': None,
                    'Insert': VK_NUMPAD0,
                    'Delete': VK_DECIMAL,
                    'Equal': None,  # Needs Fixing
                    'Multiply': VK_MULTIPLY,
                    'Add': VK_ADD,
                    'Separator': VK_SEPARATOR,
                    'Subtract': VK_SUBTRACT,
                    'Decimal': VK_DECIMAL,
                    'Divide': VK_DIVIDE,
                    0: VK_NUMPAD0,
                    1: VK_NUMPAD1,
                    2: VK_NUMPAD2,
                    3: VK_NUMPAD3,
                    4: VK_NUMPAD4,
                    5: VK_NUMPAD5,
                    6: VK_NUMPAD6,
                    7: VK_NUMPAD7,
                    8: VK_NUMPAD8,
                    9: VK_NUMPAD9}
self.numpad_keys = self.keypad_keys
```

2. 函数使用说明

1) type_string(self, char_string, char_interval=0):

输入长字串的非常方便的方法

➢ 参数说明:

Char_string-> 输入字串的内容，注意需要字串内容需要使用单引号

Char_interval-> 在输入多个字符之间，需要设置的间隔时间，如使用此函数时没有设置此参数，则此参数默认为 0

➢ 此函数实际上就是多个 tap_key 函数的循环

```
for i in char_string:
        time.sleep(char_interval)
        self.tap_key(i)
```

➢ 例如:

k=PyKeyboard()

#实例化 PyKeyboard 类

k.type_string('cmd')

#在当前位置输入 cmd

2) tap_key(self, character='', n=1):

Press 和 release 给定的字串，n 次

➢ 参数说明:

点击按键的名称

N 为点击的次数，如使用此函数时没有给出此参数的定义则默认为点击 1 次

➢ 此函数实际上就是 press 和 release 函数的组合

```
for i in xrange(n):
self.press_key(character)
self.release_key(character
```

➢ 例如:

k=PyKeyboard()

#实例化 PyKeyboard 类

k.tap_key(k.numpad_keys[5], n=3)

#输入 3 个数字 5

3) press_key(self, character=''):

长压给定的按键

➢ 参数说明:

Character 点击按键的名称

➢ 例如:

k=PyKeyboard()

#实例化 PyKeyboard 类为 m

k.press_key(k.alt_key)

#长压 Alt 按键

4) release_key(self, character=''):

释放给定的按键

➢ 参数说明:

Character 点击按键的名称

➢ 例如:

k=PyKeyboard()

#实例化 PyKeyboard 类为 m

k.release_key(k.alt_key)

#释放 Alt 按键

5) numpad_keys

数字键盘的名称

➢ 说明:

它与 keypad_keys 是相等的，它是一个字典类型的数据类型，如果输入电脑键盘中的数字可以与 tap_key(character='', n=1)一起使用

➢ 例子:

即：如果输入 1 个数字 4

k=PyKeyboard()

#实例化 PyKeyboard 类

k.tap_key(k.numpad_keys[4], n=1)

#输入 1 个数字 4

如果输入 k.tap_key(k.keypad_keys[4],n=1)也是可以正常工作的

➢ 请看下图为官网中此 keyboard 的说明

```
self.keypad_keys = {'Space': None,
                    'Tab': None,
                    'Enter': None,  # Needs Fixing
                    'F1': None,
                    'F2': None,
                    'F3': None,
                    'F4': None,
                    'Home': VK_NUMPAD7,
                    'Left': VK_NUMPAD4,
                    'Up': VK_NUMPAD8,
                    'Right': VK_NUMPAD6,
                    'Down': VK_NUMPAD2,
                    'Prior': None,
                    'Page_Up': VK_NUMPAD9,
                    'Next': None,
                    'Page_Down': VK_NUMPAD3,
                    'End': VK_NUMPAD1,
                    'Begin': None,
                    'Insert': VK_NUMPAD0,
                    'Delete': VK_DECIMAL,
                    'Equal': None,  # Needs Fixing
                    'Multiply': VK_MULTIPLY,
                    'Add': VK_ADD,
                    'Separator': VK_SEPARATOR,
                    'Subtract': VK_SUBTRACT,
                    'Decimal': VK_DECIMAL,
                    'Divide': VK_DIVIDE,
                    0: VK_NUMPAD0,
                    1: VK_NUMPAD1,
                    2: VK_NUMPAD2,
                    3: VK_NUMPAD3,
                    4: VK_NUMPAD4,
                    5: VK_NUMPAD5,
                    6: VK_NUMPAD6,
                    7: VK_NUMPAD7,
                    8: VK_NUMPAD8,
                    9: VK_NUMPAD9}
self.numpad_keys = self.keypad_keys
```

如下图如果需要使用调用数字键盘，输入数字可以使用

上述内容，但是完全 可以使用更简单的方法:

k.type_string(‘number’)来实现

6) enter_key

键盘中 Enter key 的名称

➢ 说明:

触发键盘中的 Enter 键，它的 key name 即为 enter_key,它需要与 tap_key(函数)一起使用

➢ 例子:

点击运行，在运行的框中输入“cmd”,并点击 Enter key 进入 dos 界面

m=PyMouse()

#实例化 PyMouse 类

k = PyKeyboard()

#实例化 PyKeyboard 类

m.click(19,747,1)

#点击“开始”按钮

k.type_string('cmd')

#在运行的输入框中输入命令”cmd”

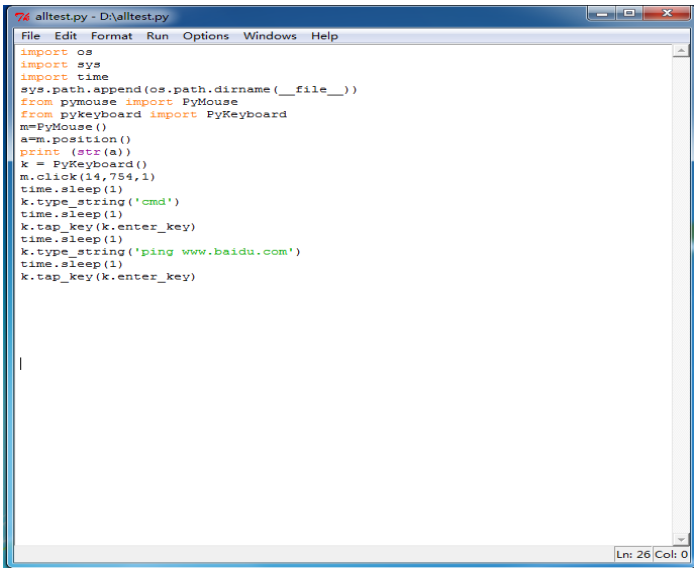
k.tap_key(k.enter_key)

#点击 Enter_key 进入 dos 界面

请看下图中对于 enter_key 的说明

```
def special_key_assignment(self):
    """
    Special Key assignment for windows
    """
    #As defined by Microsoft, refer to:
    #http://msdn.microsoft.com/en-us/library/windows/desktop/dd375731(v=vs.85).aspx
    self.backspace_key = VK_BACK
    self.tab_key = VK_TAB
    self.clear_key = VK_CLEAR
    self.return_key = VK_RETURN
    self.enter_key = self.return_key  # Because many keyboards call it "Enter"
    self.shift_key = VK_SHIFT
    self.shift_l_key = VK_LSHIFT
    self.shift_r_key = VK_RSHIFT
    self.control_key = VK_CONTROL
    self.control_l_key = VK_LCONTROL
    self.control_r_key = VK_RCONTROL
    #Windows uses "menu" to refer to Alt...
```


3. 给出一个完成特定任务的例子：如下图



此脚本的意义是:

点击运行，输入 cmd,点击 Enter key

进入 dos 界面，输入 ping www.baidu.com，点击 enter,开始进行 Ping 的操作。

六、总结

我们这一期介绍的是 PyMyInput 这个模块中的 Pymouse 和 Pykeyboard 这两个模块如何学习，这两个模块比较简单，不管是学习逻辑还是函数的复杂程度都是比较简单的，但是也告诉大家不管学习复杂的模块还是简单的模块，总体的学习逻辑是一致的：即学习资料永远都是两方面：官网和源代码，官网上隐藏了一个非常宝贵的珍宝：即学习逻辑；即官网会根据初学者的学习 逻辑将这些模块的资料进行整理；但是有一点一定要注意：官网上的资料尤其是一些函数是不全的，它只会列出常用的一些函数，尤其学习者可以参看

Pymouse 在官网上的一些函数介绍，真的非常的少。如果想要在官网上找不到你想要的函数，就要到源代码中寻找，如果是需要另外下载，另外安装的模块，则可以在 C:\Python27\Lib\site-packages 这个固定的路径下寻找，如果不需要另外下载安装，则直接在 C:\Python27\Lib\这个固定的路径下寻找即可。

还有大家在解决工作中的具体问题是，思路一定要灵活，对于具体问题一定要先考虑使用哪个模块比较合理，比较节省人力物力，这样你才能与老板真的有共同语言：即一定要切记：一定要用最少的人力物力，实现你的自动化！

参考文献:

1. Python 官网：http://www.python.org/getit/
2. PyUserInput 的官网：https://pypi.python.org/pypi/PyUserInput(此网站包括如何使用，以及其中的一些常用函数的介绍)
3. Pywin32 下载官网：https://sourceforge.net/projects/pywin32/files/pywin32/
4. Pyhook 下载官网：https://sourceforge.net/projects/pyhook/files/

场景法——基于用户行为的测试方法

◆ 作者：千 里

面试官在面试时问得最多的问题之一是关于 XXX 功能，你是如何测试的？很多测试小白面对关于某个功能如何测试时，感到非常苦恼。可见，关于某个功能如何测试，是软件测试的重点也是难点。

从软件测试的定义与方法来看，软件测试就是将用户所有对软件的操作进行验证，如果没有错误，则可说明软件是正确的。因为我们有足够的理由告诉用户，他的操作是不会出现问题，因为我们做过相同的测试。

但是这种穷举测试因为受制于时间、成本和资源是不可能完成的，所以我们需要使用等价类划分、边界值、因果图判定表、场景法、错误猜测等方法将无限的测试情形变成有限的测试用例。

而本文主要探讨场景法，因为场景法是工作中最为常用的一种方法，是一种基于用户行为的测试方法，可以简单理解为：对于某个功能点用户可能执行的操作有哪些，测试就需要验证哪些，如果用户不可能有某个操作行为自然也不属于测试范围。

如何使用用户行为分析法进行用例设计呢？

我们常见的用户行为包括以下 4 个部分

1. 用户将使用进行正常操作
2. 用户使用该功能进行非法操作
3. 用户不仅关心功能的正确实现，还关心用户体验。
4. 某些特殊情况下的用户行为，如环境异常，用户并发等。

场景法则一：有输入项的常规功能，只需要分析用户如何输入，会产生哪些操作场

景即可设计出测试用例。

例一：删除邮件

从【删除】功能而言，用户可能选择一封邮件进行删除，也可能同时选择多封邮件进行删除操作，也可能会出现未选择邮件时误点击了删除按钮。基于这些用户操作，需要设计的测试用例如下：

1. 单选一封邮件进行删除
2. 多选若干封邮件进行删除
3. 直接点击全选框，是否会进行全选操作并全部删除
4. 不选择邮件，直接删除邮件，是否给出用户相应提示

场景法则二：对于没有输入框的功能，也可以使用用户行为分析，分析会产生哪些使用场景来设计测试用例。

例二：抽奖活动

抽奖功能分析，通过点击转盘来得到相应的奖项。在这功能中，用户所中奖项是存在概率的，概率在该功能中犹为重要，其次可能存在多人同时摇转盘以及某个奖项没有了的情况。

案例如下：

1. 并发多个用户，同时进行抽奖，存在中奖与未中奖的情况形成概率。
2. 某个奖项无奖品了，该奖项不再被用户中奖。
3. 中奖与未中奖给用户的提示信息
4. 中奖用户再度参与抽奖，其中奖概率是否降低的考虑
5. ……

场景法则三：对于复杂业务，同样可以使用流程分析法进行分析与设计，按照用户对功能的使用场景分析即得到测试场景。

例三：网上银行转账功能

图 1：录入转账信息

图 2：转账安全认证

转账功能分析：

通过选择付款账号、填写转账金额、输入收款账户（含收款账号、收款人姓名、开户行）以及其他附加信息（手机号、备注等），填写好之后，进入安全认证页面选择安全工具、填写安全密码，最后提交转账。

测试分析与设计思路:

1. 正常流程类设计（确保系统实现了业务需求）
2. 异常流程类设计（确保边界等异常均做出了必要的限制）
3. 其他考查点（如转账还会涉及手续费、汇路选择等）
4. 输入框验证类
5. 用户体验等非功能考查

详细测试分析如下:

1. 付款账号、金额、收款账号、安全认证全部输入正确的，可以转账成功
   1) 付款账号是借记卡，也可以是活期存折
   2) 金额在(0,余额]之内，同时需要对边界值进行设计
   3) 收款账号、收款人姓名、开户行需要全部对应
   4) 如果收款账号是行内账号，只需要进行账号与姓名的匹配即可，系统不再需要选择开户行信息
   5) 收款账号需要考虑：行内账号，跨行转账，同城、异地转账，借记卡、信用卡账号，同名账户（本人，本行）。转活期存折、转定期存折（变定期）转定期存单（不能）
   6) 安全认证：U 盾（插入 U 盾输入密码），手机验证码（发短信到手机，输入验证码进行验证）
2. 转账无效/失败的情况说明:
   1) 付款账号金额不够、即转账金额大于账户余额
   2) 信用卡不能转出，定期存折不能转出。一般会在账号选择的时候，进行屏蔽。
   3) 转出账户被销户，冻结、挂失等异常状态，不能转出。
   4) 小于 0，等于 0 要测试，为非数字（该不是重点）
   5) 转入账户：收款账号，收款人姓名，开户行三者的任意不匹配
   6) 正常状态、挂失状态的卡可以转入，冻结、销户卡（不存在的卡）不能转入

7) U 盾密码错误、使用别人的 U 盾，验证码错误、超时、为空都会转账失败

3. 特殊的业务场景（手续费相关）

1) 手续费 XXX 封顶（比如 50 元封顶）

2) 最低手续费（比如最低收 1 元）

3) 手续费折扣（头 3 笔免手续费，手续费 5 折）

4. 其他特殊场景:

1) 转账当时限额、单笔限额（具体安全认证工具有关）

2) 转对公账号

3) 转账汇路考虑，普通转账，实时转账的测试

4) 外币转账

5) 转入账户与转出账户相同（结果为转账失败）

6) 转账超时导致转账失败（手续费照扣）

5. 输入框验证和用户体验等非功能考虑

1) 为了不让文章写起来特别长，故此部分略去。

场景法则四：对于可以使用其他测试设计方法的功能，同样可以使用场景法，一切功能都可以使用场景法功能。

例四：登录功能

对于测试界最为熟悉的功能----登录，等价类方法必讲的案例。也适合使用场景法来

设计测试用例，即将等价类设计的每一个结果认为是一个场景即可。

对于用户而言，用户有如下使用场景:

1. 用户名、密码全部正确输入的场景
2. 用户输入了错误密码的场景
3. 用户输入了不存在的用户名（错误用户名）的场景
4. 用户名未输入，即用户名为空的场景
5. 密码未输入，即密码输入为空的场景
6. 还有用户连续输入了错误密码的场景

可以发现，对于登录功能，场景法设计的测试用例与等价类设计的测试用例结果相同。

场景法总结:

场景法是一种非常实用的用例设计方法，同时也非常简单。只需要分析用户如何使用某个功能，将使用行为转化为测试用例即可，符合测试人员的思维，降低了测试设计的难度。

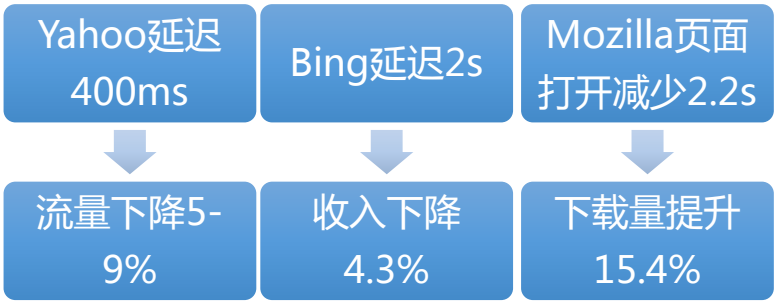
而且场景法可以适用任何功能的测试，无论是有无输入项，无论简单或复杂功能，只需要清楚功能的操作行为，了解功能的业务规则即可完成用例设计。

无线测试之 H5 测试方法（二）：性能测试

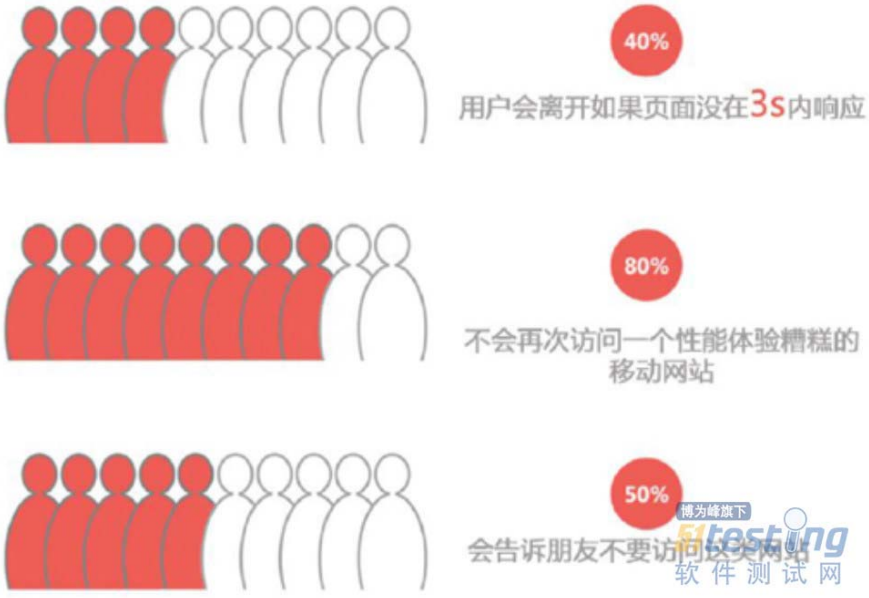
◆极测：youjiang

一、性能测试的重要性

性能对产品造成什么影响？



性能对用户造成什么影响？看下图，它直接导致了用户的流失！



而 H5 的性能体验原先就不及 native，H5 的性能测试不管是从商业角度上，还是从产品本身体验上看，都非常重要。

二、性能测试工具

1) Chrome DevTools（推荐）

i.支持移动端 H5 在 PC 远程调试;

ii.集成了 PageSeed;

iii.提供 network 面板，展示瀑布流视图，各类资源清晰罗列，还提供缩略图，方便查看图片大小尺寸和冗余或缺失;

ix.提供 timeline 面板，展示 CPU、内存、FPS 等性能数据

同类型的还有 fiddler、charles，下文就以 Chrome DevTools 为例来说。

2) WebPageTest

i.提供首屏时间、首字节时间、全界面加载时间的工具，并提供对应时间点的截图

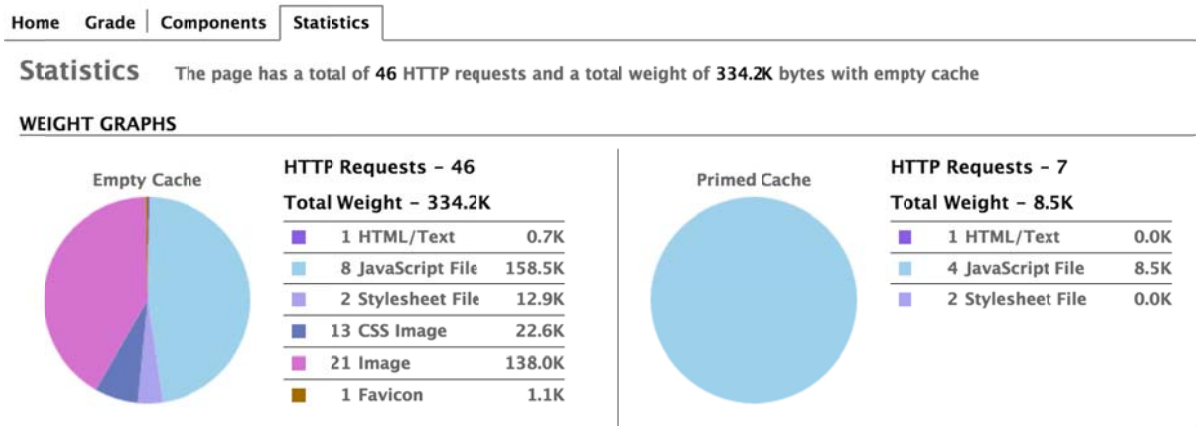
ii.不仅提供瀑布流视图，还提供连接视图，清晰展示了并发请求的 http 连接以及请求资源;

iii.提供优化建议 checklist，详细标出各个资源是否可优化：如压缩、缓存、发布 cdn、设置 GZip 等;

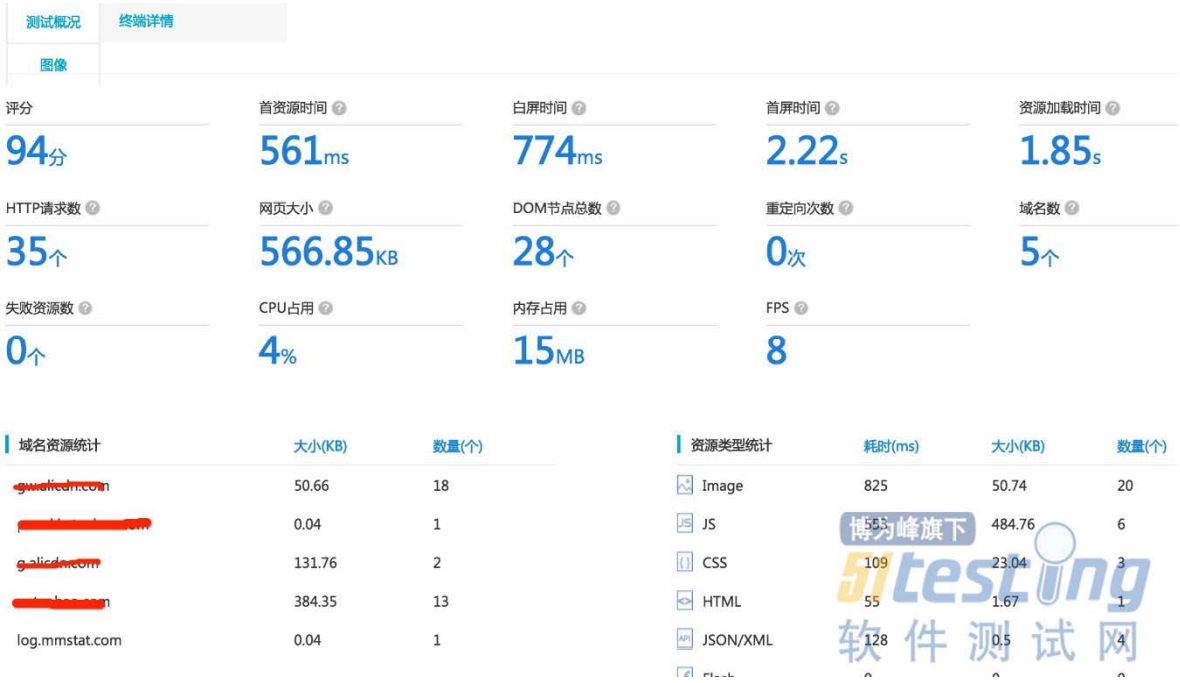
ix.提供测试过程中的视频演示

但是，他不是专门为移动端 H5 测试的，没有关注浏览器引起的内存、CPU 变化、FPS 等。

3) 可以跑分的网站性能工具 YSlow



4) **阿里云测平台**，见下图为测试概况



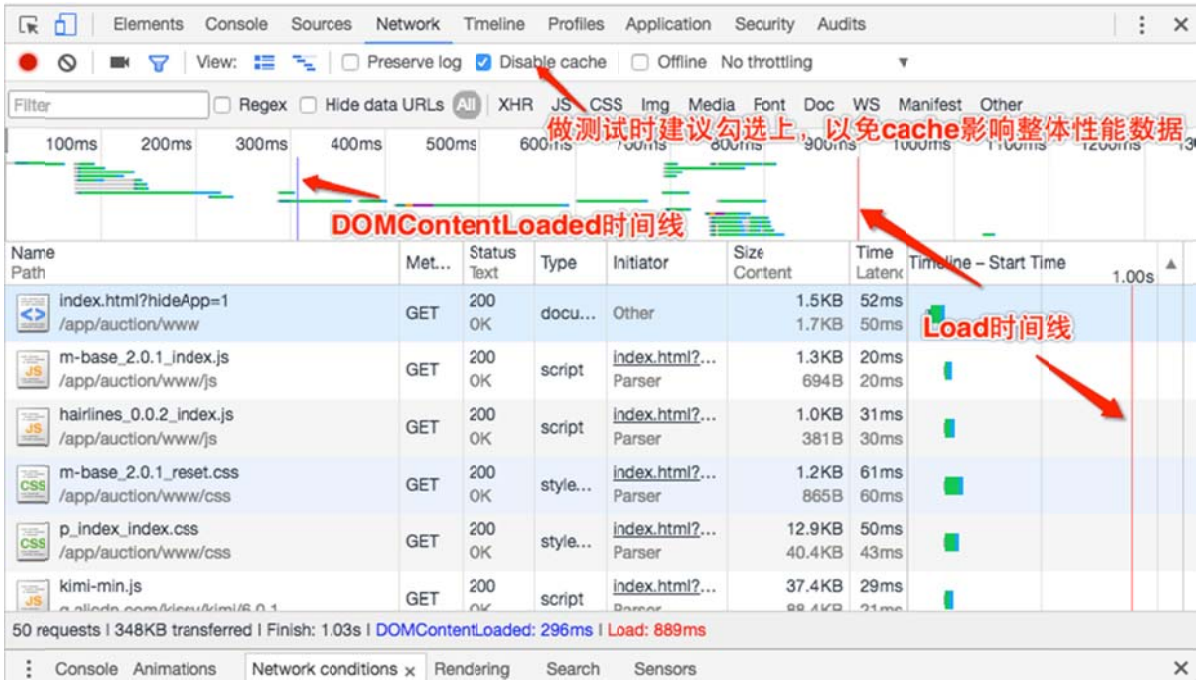
三、性能测试点

接下来，我们分别从加载性能、渲染性能、crash 角度来做系统的性能测试。下文中，我们规定:

- 表示测试校验点
- 表示性能优化点

3.1、加载性能

用 chrome 自带的开发者工具-》network 来查看下整体的性能数据，见下图



图中最下面的性能数据包括有请求数、请求资源总量，完全加载时间、onDOMContentLoaded 时间，load 时间。

PS：当 html 及 script 资源就位就会触发 onDOMContentLoaded 事件;

Load 事件则包括了外部 script，图片，iframe 标签触发，并嵌套内容下载完毕等（异步资源不包括在这个范围内）;

Finish 表示首屏完全加载所需时间。

再观察，资源类型包括有：XHR、js、css、img、media、font、Doc……

我们的 H5 加载性能和这些数据都有关。

3.1.1 总体性能指标

- 2G/3G 下，首屏不超过 200k，二次加载不超过 50k
- WIFI 下，首屏不超过 300k，二次加载不超过 50k
- 首屏 js 请求<5 个，css 请求<5 个
- 页面完全加载时间<2s
- 单个请求资源<50k，响应时间<2s
- 避免 30*/40*/50*的 http status

（正常的请求：200，200（fromcache）为图片缓存未超过缓存超时时间直接取本地缓存，304not modified 为未发生变化的响应）

3.1.2 域名收敛

- 静态资源域名收敛
- 图片域名收敛

✓ 为什么要做域名收敛？因为以下技术服务于我们的白名单域名：

1) httpdns 是面向无线端的域名解析服务，与传统走 UDP 协议的 DNS 不同，httpdns 基于 HTTP 协议。基于 HTTP 的域名解析，减少域名解析部分的时间并解决 DNS 劫持的问题。

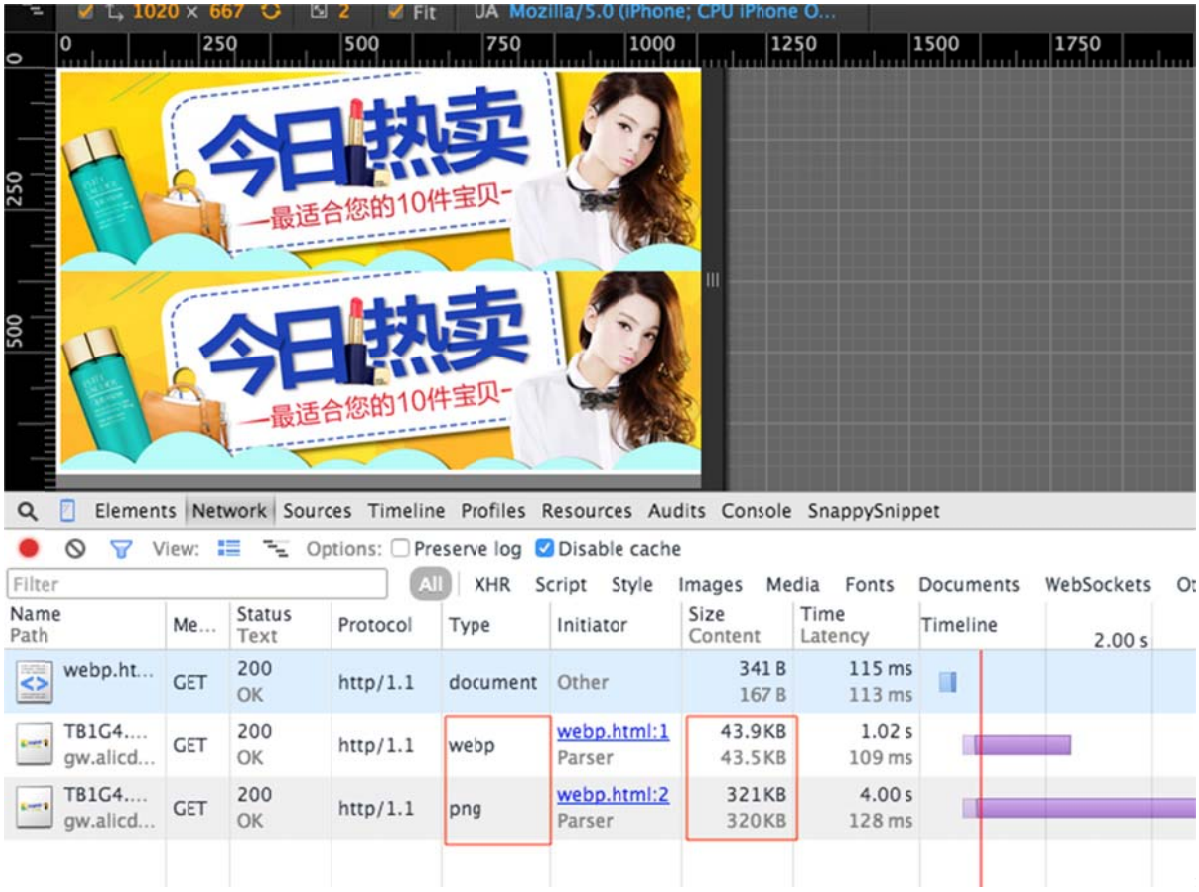
2) 提供 SPDY 协议的底层支持，从而完成多路复用的加密全双工通道，显著提升非 wifi 环境下的网络体验。

3.1.3 图片性能

- 大 banner 图：最多允许 45K，尽量控制在 40K 以下；
- 小图：最多允许 10K，尽量控制在 8K 以下；
- 位于首屏外的图片需要进行懒加载；
- 添加 webp 后缀：是一种支持有损压缩和无损压缩的图片文件格式，派生自图像编码格式 VP8。根据 Google 官方的数据，无损压缩后的 WebP 比 PNG 文件少了 26％ 的文件大小。

我们做一个测试，同一张图，分别有加上 webp 和没有加上 webp，结果见图

43.93k 和 321k 的大小差距非常明显！为用户、为公司省了 70%流量费哦！



- 图片质量：根据网络的不同，可以显示不同质量的商品图片,这样减小网络流量，加快图片的访问速度. 要调整图片质量只需要在原 uri 中增加图片质量调节参数 (q 或者 Q). Q 是绝对值, q 是相对值. 比如一个原图的质量已经被调节成 90%. 则增加 Q90 后缀，图片不会发生变化，增加 q90，则会被调节成 90% * 90% = 81%。

现在用的比较多的是 Q75。

✧ **注意点：**如果开发给图片强制规定尺寸，也加了后缀，可能会导致图片显示不出来

- png 图片不支持图片质量调整，css 中的 png 背景图应小于 10k、应 base64
- gif 图不允许使用
- 图片域名收敛，通过 cdn 加载

一些更好的优化建议

✓ 支持对展现区域内所需的图片尺寸适配，

✓ 支持根据当前的网络状况，给出适当的压缩适配，

- ✓ 支持对 CDN 图片做锐化处理（在某些场景下以低质量的图片配合锐化的效果，可以节省流量的同时，提升图片质量。）
- ✓ 支持缓存组件，同一张图片多次访问只下载一次
- ✓ 支持业务订制自己的强弱网络下的图片规则（灵活配置）

3.1.4 css&js 性能

- 使用 Gzip 或 deflate 压缩 css、js 代码，压缩后的大小可以降低至原来的 1/3 以下，有效节约流量
- 尽量将 js、css 合并以减少请求次数，但有时候需要根据实际情况做合理合并
- 针对不同屏幕的设备优化 css 样式
- 域名收敛，通过 cdn 加速加载
- 后缀加时间戳（时间戳不变则命中缓存，时间戳变更则更新代码）
- 注意资源排序位置，

一般情况下，CSS 要放到 html 代码的开头的 head 标签结束前；js 放到</body>前，这样的话，js 的加载不会影响初始页面的渲染。但需要结合实际情况来。

3.1.5 服务端接口

- 首屏接口请求数不超过 1 个，对首屏后端请求做合理合并
- 减少后端数据量，去掉无关紧要的数据返回

看下面的例子，服务端的耗时在所有资源里面最长，严重影响了页面加载时间。

后端接口直接影响着内容显示&图片加载，mtop 的响应速度 & 不被干扰，非常非常重要。

此外 MTOP 接口令牌过期问题（虽然重复请求能解决问题，但是重抓去的数据上看开销不少。）

3.1.6 离线缓存

HTML5 引入了应用程序缓存，这意味着 web 应用可进行缓存。

一般的策略是，在 wifi 下用户无感知下默默将页面资源(包括 html、css、js、jpg)等缓存起来，并可在没有网络连接时进行访问。应用程序缓存为应用带来三个优势:

i.离线浏览：用户可在应用离线时浏览它们。

ii.速度：已缓存资源加载得更快。

iii.减少服务器负载：浏览器将只从服务器下载更新过或更改过的资源。

- 大量使用离线缓存会使 hybrid app 总量非常大，建议通过个性化算法等技术合理有效地使用。

3.1.7 Iconfont

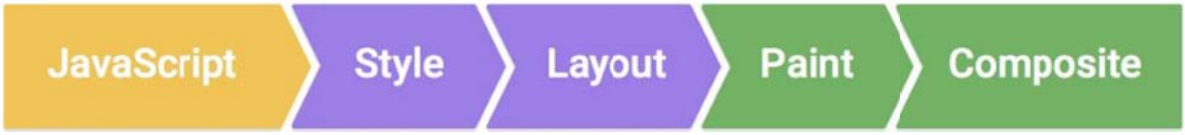
可以认为是一种矢量类型的操作字体。如果页面中有较多的操作图标，可以考虑使用 iconfont 来替代图片资源。iconfont 对于前端来说有很多优点：自由变化大小、矢量不失真、自由修改颜色、可以添加一些视觉效果如 阴影、旋转、透明度、兼容 IE6。

- css 中 iconfont 不使用 woff 文件
- css 中 iconfont 的 ttf 字体文件需要 base64

3.2 渲染性能

用户都希望他们访问的 web 应用是可交互且运行流畅的，这就要求页面不但要能被快速加载，还要能流畅运行：页面的滚动要快速响应手指的动作，动画和交互效果更要如丝般顺滑。

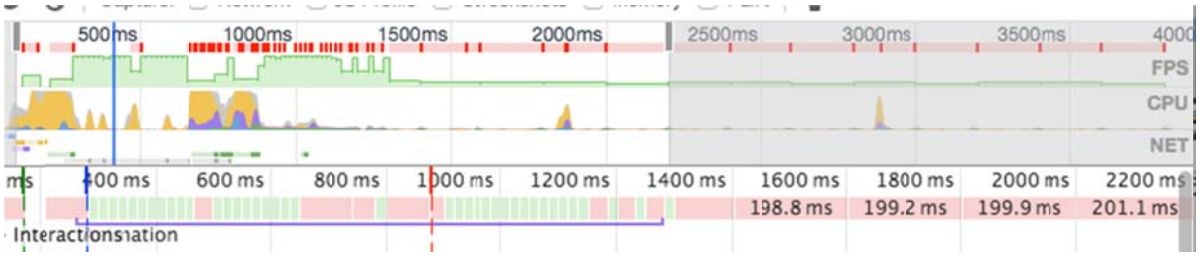
前端的代码是如何转换成屏幕上的显示呢？包含 5 个关键步骤:



从帧率高低直接影响体验感知，而上图 5 个因素影响了帧率

3.2.1 帧率

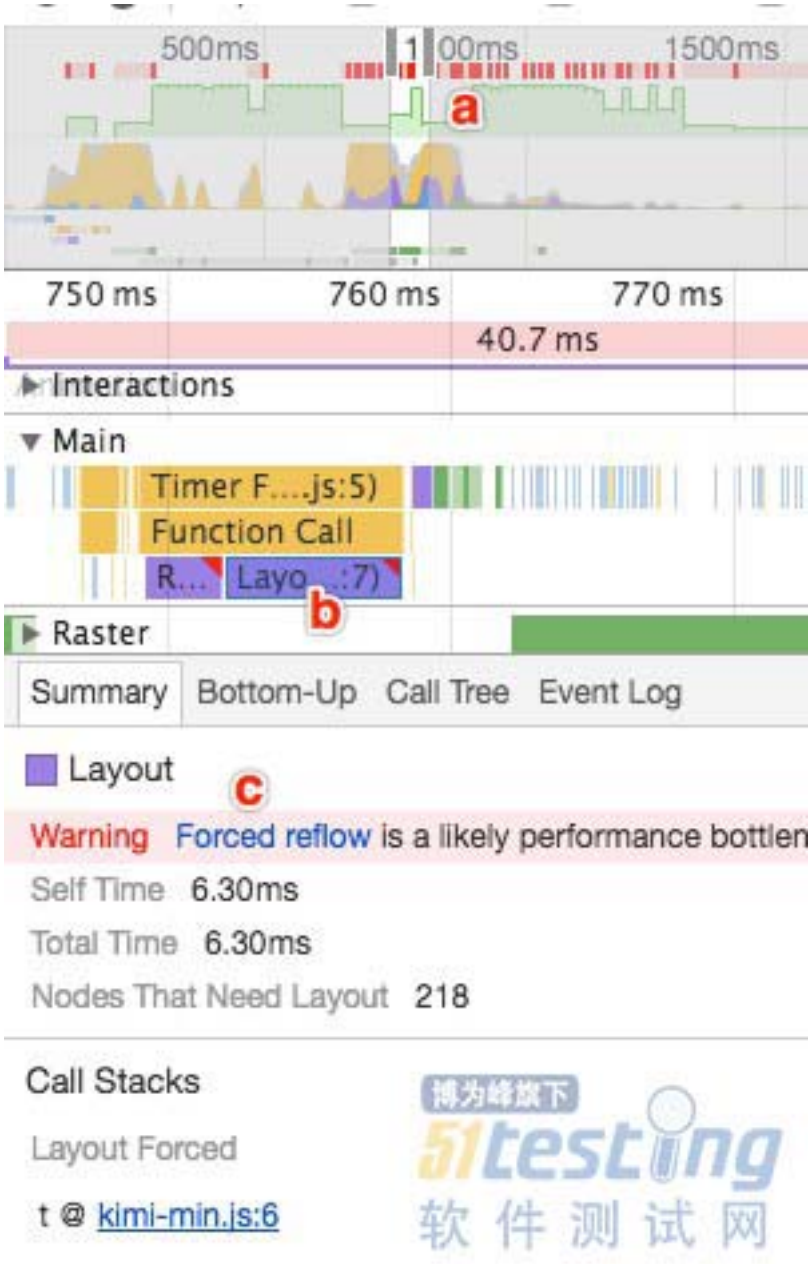
首先使用 Chrome 的 Timeline，滚动页面，进行 Record。你会得到如下类似结果，其中绿色的波浪线就是页面的帧率：



- 帧率>60 会有流畅的感知
- 帧率>30 是在低端机下也必须达到的，否则卡顿很明显

此外，破浪线越高表示帧率越高，波浪线越低，表示帧率越低，同时，帧率区域上边标红一行区域，表示有问题的帧，凡是标红的帧都是存在问题的，排查问题时，需要着重关注帧率低和标红的地方。

你需要逐一排查标红的帧，以下图为例



a. 选中红色出问题的帧区间

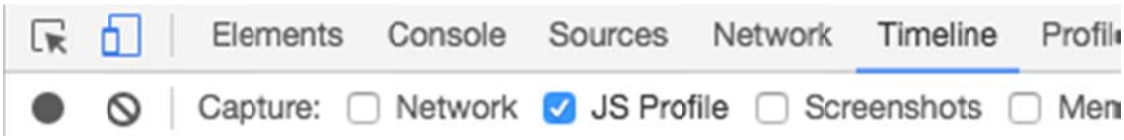
b. 带红色角标的帧，即是有问题的帧，选中该帧

c. 可以看到详细的耗时和简单的问题描述

可以看出，本例中，改帧 layout 触发布局，耗时 6.3ms，导致了卡顿

3.2.2 JavaScript

如果你发现了运行时间很长的 JavaScript 代码，那么你可以开启 DevTools 中顶部的 JS profiler 选项，开启后，会看到更多的细节:



优化建议:

- 使用 requestAnimationFrame
- 降低代码复杂度或者使用 Web Workers

3.2.3 render

render 部分包括 Recalculate Style 和 Layout，如果你发现 render 部分耗时较长，需要分别从这两部分进行分析。如果你这一帧，触发了强制 layout，Timeline 会用红色角标标出，这是需要进行优化的地方。

Style 优化建议:

- 降低样式计算和复杂度
- 使用块、元素、修饰语

Layout 优化建议:

- 避免大规模、复杂的布局
- 尽可能避免触发布局
- 使用 flexbox 替代老的布局模型
- 避免强制同步布局事件的发生
- 避免快速连续的布局

3.2.4 Paint

Paint（绘制）其实是生成元素呈现的像素的过程。例如，一个有着灰色背景，有文字的元素，当浏览器 Paint 它时，是决定哪些像素填充背景，哪些像素填充文字，然后浏览器将这些像素存入位图（Bitmap）中。

在页面的整个被解析、执行、渲染的过程中，Paint 通常来说是代价最高的一步，

尽量减少 Paint 时间，甚至避免 Paint 的发生，对页面性能的提升有着很重要的作用。

Paint 优化建议:

- 提升元素渲染层为合成层
- 使用 transform or opacity 实现动画
- 减少绘制区域
- 降低绘制复杂度

3.2.5 Composite

渲染层合并，由上一步可知，对页面中 DOM 元素的绘制是在多个层上进行的。在每个层上完成绘制过程之后，浏览器会将所有层按照合理的顺序合并成一个图层，然后显示在屏幕上。对于有位置重叠的元素的页面，这个过程尤其重要，因为一旦图层的合并顺序出错，将会导致元素显示异常。

Composite 优化建议:

- 提升动画效果的元素
- 使用 transform 或者 opacity 来实现动画效果
- 减少绘制区域
- 合理管理合成层
- 防止层爆炸

3.3 Crash

APP 在发生某些严重错误时，会发生崩溃/闪退。这种现象称为 Crash。

Crash 发生的原因很多，而一般 app 的 crash 日志实际对 H5 页面问题排查帮助不大，所以主要依靠经验总结。

3.3.1 HybirdAPP 雷区

页面触发 HybirdApp 的「雷区」导致的 Crash。app 安卓、iphone 的 WebView 实现中存在一些 bug(诸如对某些实验性特性的支持不够完善等), 当 H5 页面的相关内容(不仅包括 JS，还包括样式等) 踩到了「雷区」，就会触发 Crash。

这种 Crash 一般仅仅在某些机型上发生，是难以避免的。需依赖 app 的来跟进修复，

在此之前，只能以案例+总结的方式进行。

3.3.2 jsheap

页面 jsheap 使用内存过多是 h5 页面导致 APP Crash 的最常见原因之一。当页面所消耗的存储太高的时候，APP 就会闪退。

需要经过多次试验，拿到单个页面的内存消耗峰值数据。

3.3.3 layer 层爆炸

通过 opacity、animation、positon 等方式创建层，即便是 1w 个，页面也没有明显变化；但是使用 transform 创建 2k~5k 个层，页面会卡顿几秒后立即闪退

3.4.4 并行请求过多

并发请求也是存在响应问题的，Fetch API 和 CSS Resource 并发 1k 请求没有出现问题，但是 XHR 和 Script Resource 请求，问题特别明显，虽然没有导致页面闪退，但是页面已经进入了假死状态。

3.4.5 调用未定义的 jsbridge

参考性能指标

拿了某主流 app 的性能指标来作参考

页面首屏加载请求的数量	<80
页面首屏加载总大小	<=300KB
页面二次加载大小	<=50KB
首次打开页面下载内容的整体大小 (Gzip 后)	<= 1500KB
首次打开页面到可交互时的时间(DOM Ready)	<=4s
首次打开页面完全加载所需时间	<=5s
服务器响应时间(RESPONSE TIME)	<=300ms
单个静态资源大小（如 CSS、JS、图片）	<=50KB
Assets 4xx/5xx 错误	没有
所有 Assets 必须存放在 CDN	所有资源都存放在 CDN
非首屏出现的图片建议懒加载	做懒加载

	（页面打开时,在首屏 1024px 高度后面不建议加载出的图片）
页面自动跳转次数	<=1
开始渲染时间	非移动信号网络下,开始渲染时间不超过 500ms
	3G/4G 环境下,开始渲染时间不超过 1s

相关文章:

无线测试之 H5 测试方法（1）：功能测试

3 个月的测试生涯

◆ 作　者：兔子闹

这是我的首份测试工作，于一家主要做银行项目和保险项目的 40 人左右的小公司，干了近乎 3 个月了。

刚上班没多久就去外地出差，因为是个银行项目，需用到它们银行内网才行，这次出差也算是我测试人生的正式开始吧。去了之后，项目开发经理会给一份测试工作表，具体到日，每日的工作量基本上已经安排出来（但是具体每天做那几个模块自己决定），然后开发经理会告诉你这个系统目前的一些现状（说些啥我也都没怎么听懂），至于需求分析什么的书面资料统统没有，开发经理都没有一份完整的需求，因为工作越早做完我就可以越早回去，索性我也就不再搜寻那些书面资料，直接上手测试工作，测试用例的模板是之前老师给的，手头里就仅有开发经理给的系统网址、用户名、密码，登陆进去之后就看到整个系统了，每个模块大体的看下就明白界面设计大致风格。再找开发经理问下这边的测试工作是个什么流程以及测试工作中需要注意的一些问题啥的，**其实功能测试在实际工作中就是写测试用例，寻找问题，提交问题，以及后续跟踪**（我们公司用的是“禅道”，开发经理会给用户名和密码，已经设好了权限，我在上面提交 bug 和测试用例，开发人员登陆进去进行修改再指派给测试人员，你可以了解下这个系统），至于那些问题所在、总结报告啊啥的暂时我还没碰到，不过就算需要到时候在禅道上直接导出再修改就行了（事后居然得到经理的表扬，说我测得很细，哈哈哈，这是对我工作的肯定啊，感觉很开心，因为身为摩羯座的我天生就是严谨的人）。

出差回到公司，每天上班第一件事，登陆“禅道”，查看自己提交的 bug 的一些后续情况。开发忙着写代码，测试就翻翻系统看看，在网上找点最新关于测试的资料以及一些金融知识进行学习（毕竟是银行项目，作为测试员还是有必要了解些相关知识的，好在自己本专业是财务管理，所以了解起来并不是很费劲），因为是家小公司，所以工作氛

围还是蛮好的，大家都是各干各的事、一起讨论某些问题，偶尔大家也可以聊聊天开开玩笑都是允许的，我喜欢这样的工作环境，很少会有加班。一般有新需求的话老板直接找相关开发人家进会议室去讨论，每天都会问下进度并指导相关工作的进行。

干了这么久，发现理论跟实践还是很有差距的：

1) 曾经以为搞测试必须熟悉需求啥的，干了才知道没有需求文档测试工作也照样进行；

2) 曾经以为测试分为性能测试和功能测试，干了才知道测试原来分为功能测试和业务测试（仅针对我们公司啊）。功能测试测功能的实　　现(功能肯定能实现，其实主要就是界面的一些问题)；业务测试是测数据的正确性（业务测试人员用公式计算出的结果跟计算机跑出来　　的结果进行比对）；性能测试是开发人员，他们自己有一套属于他们的测试办法（后续我再找开发探探）。

3) 曾经以为写测试用例是件很麻烦的事情，干了才知道其实也就那样，大概步骤写了就行了，没必要写的像我们培训老师讲的那样；

至于工作必备：一台笔记本，一个被测系统，一个 excel 测试模板，一个红蜻蜓截图工具，一个 QQ 就开始工作了，一个靠垫

1，得了解整个系统设计风格，当你打开多个界面的时候你就能发现一些界面小问题。

2，红蜻蜓截图工具很好使，截图之后方便在上面做标记。

3，登陆 QQ 就是方便你跟开发人员进行沟通。由于我没有需求说明书、操作手册等资料，有些要测的东西需要数据，数据不是乱输的，得找开发要他们录入数据库的，还有些操作也得问清楚他们呢。

4，坐上个把小时还是全身酸疼，有个靠垫能让自己好点，中午还可以抱着它休息。

测试中遇到的问题：

1，自己这里遇到问题而开发没有遇到，这样的话就得找寻多台机子去试，看是自己的原因还是其他什么原因，主动找寻多种原因来让开发心服口服

2，提交 bug 时尽可能的写出详细步骤、并且把你找寻 bug 的图片保留下来，以防开发日后问你这个问题怎么显出来的(之前有个开发就这样，我从出差的地方回来很久了，他来问我给他提的问题他咋没有看到，我都回来了，那个项目我在这边也打不开（因为

是银行项目，用它们内网），天呐，得亏那些 bug 图片我还留在手中，不然我真的想不起来，其实最后的原因就是版本更新把那个去掉了，他基本上每个 bug 都问我，毕竟自己做的项目自己最熟悉，而且步骤截图的很清楚了啊，怎么就懒得看呢）

3，有些缺陷自己知道但就是不知道怎么用言语简洁明了的表达出来

4，很重要的一点就是怎么提高自己的测试效率，之前我是边写测试用例边提交 bug，想节省时间但是发现这样干没有节约时间反而浪费很多时间，一旦干起来真的是受不了任何的打扰，就是别人不打扰自己也能被这些逼疯，很多时候就是用例写了一步遇到问题了那就去提交 bug，但是 bug 一找就又根本停不下来，就开始使劲的找 bug 把用例给忘了，这样下来就不同步，返回来再写测试用例时，详细数据还得从刚才的 bug 中找寻，很浪费时间。再后来我就先写测试用例，测试用例一遍搞完（期望结果和实际结果一并写完）遇到实际结果与期望结果不符的切实际结果有问题的就直接用红色显示实际结果并截图保存，对了，在写测试用例的时候我是照着系统来写，所以一些界面问题我也直接截屏画重点给保存放在文件夹里，等到测试用例写完，打开“禅道”提交 bug，再自己刚才保存图片的文件夹和测试用例拿出来，其实像有些界面上的问题看一下截好的图片就能提交，像那些需要数据的就从测试用例中复制粘贴在提交 bug 的详细步骤里。（适合自己的就是最好的吧，目前还在摸索期，你们若是有好的也可共享下）。

5，有时候自己不太明白的一定要问清楚，不要怕别人嘲笑，更不要怕挨批评，别在意太多，不管出了多少错，都可以把原因归结为我们太年轻、没经验，别人不说、我们怎么会成长，老板他越给你找事，你就努力改正改到他无话可说，你就完美了，到时你得成长一大截呢。也别觉得公司小、工资低了什么的，都还年轻，小公司成长机会大，脚踏实地最重要，人不要老跟别人比，跟自个比，看自己成长多少、进步多少。加油

这份工作我还会继续下去，并且我还会学习更多的相关知识……软件测试这条道路上我希望探索的更长更深!!!

如何给产品需求做“体检”

◆搜狗测试：王 丹

作为软件测试工程师，我们不仅要扮演质量检测员，还要扮演用户，要有用户的思维，同时还要有医生的敏锐度，寻找出产品需求中潜在的“隐患”，防患于未然。一个产品能否上线，上线后是否被用户喜爱？这些问题与我们测试工程师息息相关，因为我们是产品上线前最后关卡的守卫员。那么如何给产品需求做一个全面的体检，将病变的细胞消灭在萌芽状态，减少修复成本，产品更健康，用户更喜爱？下面的清单就是我们需要检查的项目，让我们一项一项的按序完成它吧。

一、 “体检”的目的

总的来说，给产品需求做“体检”主要有以下 4 个目的:

1）寻找需求的缺陷，比如某些场景没有考虑到

2）已有的需求是否符合需求的目的

3）已有的产品在易用性、易操作性、易学习性等方面体验度是否足够好

4）提出新的需求或建议，让用户体验更好

二、 “体检”项目

2.1、产品需求的一致性

定义：一致性是指与其它软件需求或相关标准规范不相矛盾

1）系统规范：产品的风格要尽量与系统本身的系统风格保持一致，这样使得产品更专业，也不会让习惯了系统风格的用户感觉不适

举例：表情投稿界面添加颜文字输入框中，输入内容只有点击键盘一个一个字符删除，OS 系统输入框均有删除 icon，一键删除全部内容

原需求：没有提到在投稿界面添加颜文字中加入清空已输入内容功能

新需求：添加颜文字输入框增加删除 icon

2）APP 整体设计规范：同一个产品中类似功能要界面风格一致，这样既规范，产品品质也高

举例：表情商城顶部的 tab 不管文字个数多少，所分割的按键大小一致，新增的皮肤商城顶部 tab 根据文字个数分割按键大小

原需求：设计稿上皮肤商城和表情商城顶部 Tab 大小不是一致的

新需求：设计修改皮肤商城和表情商城顶部精品 Tab 等大小保持一致

3）借鉴竞品类似功能逻辑：竞品类似功能如优于我们，可以借鉴与其保持一致

举例：在 Email、URL 输入框下，输入首字母大写的概率很小，一般情况下都是小写。对比其他输入法，在 Email、URL 输入框下，shift 键不高亮

原需求：在句首时切换/调起英文键盘，shift 默认高亮

新需求：英文适配键盘（URL、Email）下 shift 键不默认高亮

2.2、产品需求的可行性

定义：产品提出的需求可开发实现，且实现的效果达到需求提出的目的，且可验证

1）无二义性：对所有需求说明都只能有一个明确统一的解释，由于自然语言极易导致二义性，所以尽量把每项需求用简洁明了的语言表达出来

A、文字本身存在的二义性

举例：原文案“请输入一个昵称”，“一个”这个词用在这里不合适，语义理解不唯

一，用户可能会有是否可以输入两个昵称，昵称是自己的还是其他什么的昵称等疑问

原需求：在投稿界面未输入昵称时，点击提交提示文案“请输入一个昵称”

新需求：修改提示文案：请输入您的昵称

B、词语背后的扩展义导致的二义性

举例：4.0.0 增加泛灵犀功能，搜索框中提示文案中的推荐两个字，在现实生活中语义中含有部分贬义，推荐的东西一般都不怎么好

原需求：输入时智能推荐服务，点击即刻分享

新需求：输入即时匹配服务，点击瞬间分享

2）健壮性：需求的说明中是否对可能出现的异常进行了分析，并且对这些异常进行了容错处理。此外需求是否符合用户场景

A、针对异常情况进行容错处理

举例：4.0.0 版本新增泛灵犀功能，有主动和被动入口之分，网络情况有有网、无网、网络超时之分

原需求：只针对网络正常、无网的情况有正在加载和无网的提示，网络超时情况未提及

新需求：增加了网络超时的时间限制和提示语

B、符合用户实际使用场景

举例：删除自定义皮肤将其他非自定义皮肤也一起删除

原需求：在自定义皮肤界面点击全部删除按钮后，全部皮肤界面的所有皮肤也会被删除

新需求：在自定义皮肤 TAB 中点击全部删除，确认后自定义皮肤全部删除（已启用不可删除），但是全部皮肤中还有其他下载皮肤&内置皮肤。在全部皮肤中点击全部删除，确认后两个 TAB 中下载/自定义皮肤全部删除

3）可测试性：每项需求都能通过设计测试用例或其它的验证方法来进行测试

2.3、产品需求的易用性

1）易理解：让用户在使用新功能前体验，视觉了解功能

A、新功能展现的直观性

举例：新增花漾字功能时，需要开启完全访问开关才能使用

原需求：未开启花漾字开关时，界面只有一个开关按钮，用户不知道花漾字里面有什么内容，用户因为不知道里面有什么内容，可能永远不开启开关，也不会使用

新需求：未开启花漾字开关，界面文字置灰，可以看到花漾字界面包含的内容。用户可能看到了内容，觉得很有趣，开始使用此功能。类似销售中常使用的用户先体验后购买的营销策略

B、文案的简洁，通俗易懂

举例：4.0.0 版本输入法在键盘中新增了语音开启完全访问提示

原需求：文案内容是“开启「完全访问」才能将语音转文字发给好友哦”，在单手键盘下，键盘调节到最小时，因为文案内容多，部分文字遮挡，显示为开启「完全访问」...音转文字发给好友哦

新需求：提示文案就是提示用户的，当内容显示不全，用户就无法明白提示的内容，也没有达到我们增加提示的目的。所以文案更改为“开启「完全访问」语音转文字发给好友”，在单手键盘模式键盘调节到最小的时候也可以全部内容显示

2）易操作：用户使用某一功能用最少的步骤达到目的或者保留用户已有使用习惯，且操作结果可感知

A、减少用户操作

举例：4.0.0 版本新增斗图功能，到收藏斗图个数达到上限，继续收藏会弹出整理斗图的按钮

原需求：点击整理斗图按钮跳转到我的表情斗图界面

新需求：点击整理斗图按钮跳转到我的表情斗图编辑界面

B、与用户操作习惯保持一致

举例：当空格选择联想词打开时，显示通讯录信息后，点击空格键可以打开通讯录信息详情界面。同理，根据此需求泛灵犀联想也可以这样做

原需求：没有提到当显示泛灵犀联想时，点击空格键怎么处理

新需求：当空格选择联想词打开后，有泛灵犀联想时点击空格进入到泛灵犀界面

C、对类似于点击类的操作要给用户一个反馈

举例：4.0.0 版本新增斗图功能，可以进行收藏、复制等操作

原需求：点击收藏和复制按钮后没有任何提示

新需求：点击收藏和复制按钮出现已收藏和已复制提示语

3）吸引性：功能设计出来的效果是否对用户有吸引力，能否产生使用收益

举例：快捷短语新增表白 tab，内容普通不新颖，用户直接输入所花费时间可能比使用快捷短语时间还短，无法吸引用户

原需求：快捷短语“表白”Tab 里面句子“ 我喜欢你、我爱你、我不能没有你”等等

新需求：所有内容全部更换，换成比较有意思的句子，如：戒烟容易，戒酒太难等

备注：

原需求：产品给出的原始需求

新需求：提出需求建议后，更改后的需求

拓展学习：

访问博为峰网校《软件需求精讲》课程，可以对需求有个全面清晰的认识！试听>>

小试牛刀—完整实例带你探究 LR 性能测试

◆ 作　者：姜林斌

一、性能测试理论知识

(ps: 我们先来了解下性能测试理论方面知识)

1.1、性能测试及其目的

性能测试的定义:

通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。

性能测试的手段:

是通过模拟真实业务从而向服务器发送大量并发请求进而对被测系统产生负载，分析被测系统在不同压力下的表现。

我们进行性能测试的常见目的如下:

a: 评估系统的性能（在局域网测试环境或生产环境下，通过测试结果的分析评估当前系统的服务级别）。

b: 定位性能瓶颈（通过性能测试找出影响系统整体性能的关键步骤或过程，为系统调优提供方向性依据）。

c: 验证调优结果(通过比对优化后和优化前的测试结果，确认性能优化策略是否生效)。

1.2、性能测试的种类细分

1.2.1、压力测试:

通过逐步增加系统负载，测试系统性能的变化，并最终确定在什么负载条件下系统

性能处于失效状态来获得系统能提供的最大服务级别的测试。

压力测试是逐步增加负载，使系统某些资源达到临界点。

1.2.2、负载测试:

通过逐步增加系统负载，测试系统性能的变化，并最终确定在满足性能指标的前提下，系统所能够承受的最大负载量的测试。

1.2.3、稳定性测试:

通过给系统加载一定的业务压力（如 CPU 资源在 70%～90%的使用率）的情况下，运行一段时间，检查系统是否稳定。因为运行时间较长，所以通常可以测试出系统是否有内存泄露等问题。

1.2.4、容量测试:

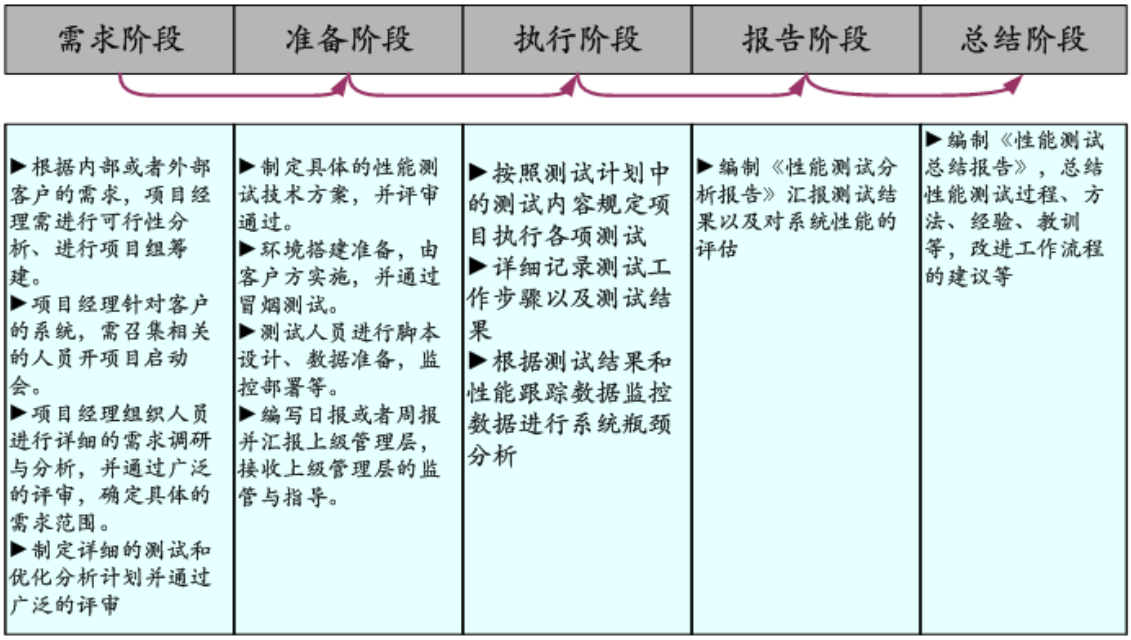
在一定的软、硬件条件下，在数据库中构造不同数量级的记录数量，通过运行一种或多种业务场景，在一定虚拟用户数量的情况下，获取不同数量级别的性能指标，从而得到数据库能够处理的最大会话能力、最大容量等。

1.2.5、配置测试:

通过对被测试软件的软硬件配置的测试。配置测试能充分利用有限的软硬件资源，发挥系统的最佳处理能力，同时可以将其与其他性能测试类型联合应用，为系统调优提供参考。

1.3、性能测试的实施流程

性能测试流程体系框架图



需求阶段	准备阶段	执行阶段	报告阶段	总结阶段
▶根据内部或者外部客户的需求，项目经理需进行可行性分析、进行项目组筹建。 ▶项目经理针对客户的系统，需召集相关的人员开项目启动会。 ▶项目经理组织人员进行详细的需求调研与分析，并通过广泛的评审，确定具体的需求范围。 ▶制定详细的测试和优化分析计划并通过广泛的评审	▶制定具体的性能测试技术方案，并评审通过。 ▶环境搭建准备，由客户方实施，并通过冒烟测试。 ▶测试人员进行脚本设计、数据准备，监控部署等。 ▶编写日报或者周报并汇报上级管理层，接收上级管理层的监管与指导。	▶按照测试计划中的测试内容规定项目执行各项测试 ▶详细记录测试工作步骤以及测试结果 ▶根据测试结果和性能跟踪数据监控数据进行系统瓶颈分析	▶编制《性能测试分析报告》汇报测试结果以及对系统性能的评估	▶编制《性能测试总结报告》，总结性能测试过程、方法、经验、教训等，改进工作流程的建议等

(PS:在实施性能测试的过程中，整体工作流程是 1:分析性能测试需求--→2:设计性能测试方案→3:开发性能测试脚本-→4: 搭建性能测试环境-→5: 执行测试-:6: 分析结果后多轮测试进行验证优化-→7: 编写性能测试报告-→8: 编写性能测试总结报告)

二、性能需求分析

(ps: 以我之前做过的一个小需求逐步开始吧~~)

目前公司开发人员 15 名，测试人员 7 名；使用 TeamFoundation 进行文档和测试用例以及 bug 的管理，团队考虑使用开源版禅道系统代替现有的 teamfoundation。

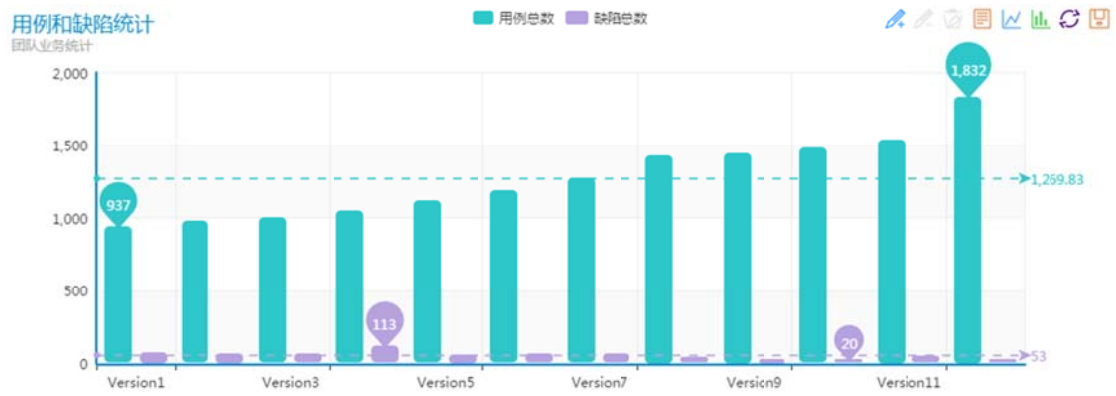
此次性能测试活动目标如下:

1: 能否使用禅道开源版(8.0.1)代替 TeamFoundation 进行项目活动管理。

2: 评估禅道开源版(8.0.1)在一定的服务器硬件配置(cpui5 3.3GHz+内存 8G)下的最大负载。

（ps: 分析团队历史数据）

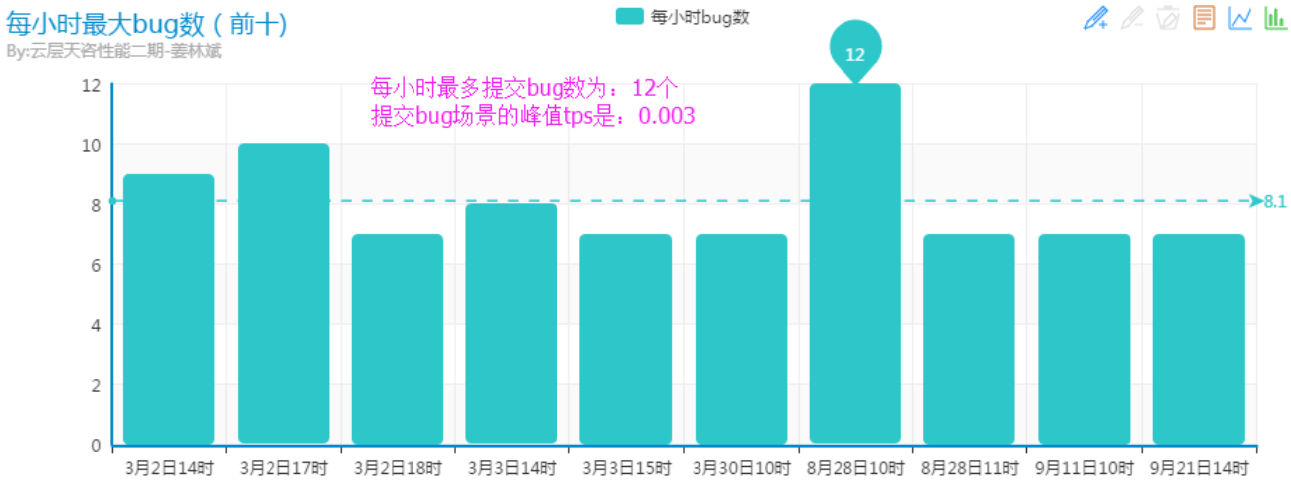
- **每版本测试用例数统计:**



(ps:

在 12 个迭代版本中 最多新建用例 1832 个)

- **单日提交 bug 数统计:**



- **每小时提交 Bug 数统计:**

- **以近一年来 12 个版本的数据进行分析，团队现状如下：**

版本最大 Case 数：1832 单个版本最大 Bug 数：113

每月 Release 一个新版本，即：每 22 个工作日进行一次发布。

在每次 releae 过程中 编写测试用例的时间为：3 天 执行测试的时间为 7 天(一轮回归测试)。

按峰值大致算出团队目前各场景的事务数如下：

团队目前各场景事务统计

场景名称	每日事务数	TPS
添加用例	1832/3日=610/日	0.0212
执行用例	1832/7日=261/日	0.0091
提交Bug	(峰值)12/3600s	0.003
解决Bug	同提交bug(ps:要求测试过程中bug日结)	0.003
关闭Bug	同提交bug(ps：要求测试过程中bug生命周期<=24小时)	0.003

(ps:测试过程中对峰值 tps 也需要留 20%的富余)

- **团队成员对业务及禅道环境的要求：**

禅道开源版(8.0.1)负载测试需求

场景名称	响应时间	CPU使用率	内存使用率	磁盘读/写时间占比	事务成功率
登陆禅道	小于5s	小于85%	小于90%	小于90%	等于100%
添加用例					
执行用例					
提交Bug					
确认Bug					
解决Bug					
关闭Bug					

（ps:性能需求分析是性能测试流程中的第一步,如果这一步做好了 接下来的测试方案设计，脚本开发，测试执行，测试报告都会轻松很多； 反推也是成立的，如果不清楚需求是什么 后面多的一切都是白做！

另外：收集需求数据的途径有 1：运维拉取生产环境的历史数据。2：参考竞品。3：对数据增量可以进行容量建模。

切记一点：千万别用什么所谓的二八原则，没有数据依据一切都是胡扯！）

三、性能测试方案设计

（ps：完整的性能测试方案 直接拿去用吧~ 嘿嘿~）

3.1、测试目的、范围与目标

3.1.1、测试目的

本次性能测试的主要目的在于：

- 测试已完成系统的综合性能表现，检验交易或系统的处理能力是否满足系统运行的性能要求；
- 发现交易中存在的性能瓶颈，并对性能瓶颈进行修改；
- 模拟发生概率较高的单点故障，对系统得可靠性进行验证；

3.1.2、测试功能范围

序号	业务名称	优先级	备注
1	登录系统	中	
2	添加测试用例	高	
3	执行用例	高	
4	提交 Bug	高	
5	解决 Bug	高	
6	关闭 Bug	高	
8	确认 Bug	高	
9			
10			

3.1.3、测试指标范围

*/*** 明确列出说明本次测试需要关注的测试指标的定义及范围，不需要关注的测试指标也应列出。下面的内容供参考。***/*

本次性能测试需要获得的性能指标如下：

- 交易的响应能力：即在单交易负载和模拟生产交易情况的混合场景负载压力情况下，系统的响应时间。
- 每秒处理事务数：即应用系统在单位时间内完成的交易量（TPS）。
- 系统可支持的并发用户数量。

本次性能测试的限制性指标为：

- 系统资源使用情况：在正常压力下，应用服务器和数据库服务器的 CPU、Memory 占用率应分别低于 80%、80%，数据库存储空间和文件系统空间占用率应低于 80%。
- 交易的成功率：交易成功率不低于 99.5%。

本次性能测试不需要关注的指标:

- 业务流程/路径覆盖率。
- 业务数据的完整、正确性。
- 其他诸如系统易用性、可管理性等属于专项测试的内容。

3.1.4、测试目标

*///***明确本次测试各功能项的测试指标需要达到的测试目标，该目标须由项目组提出或最终确认。***///*

- 针对不同类型交易的单业务事务平均响应时间
- 针对不同类型交易的单业务事务 TPS 值
- 在负载情况下的单业务事务平均响应时间
- 在负载情况下的单业务事务 TPS 值
- 在负载情况下的系统综合 TPS 值

3.2、测试资源

3.2.1、系统生产环境物理架构

*///***说明本项目生产环境的物理架构，可以以物理架构图或网络拓扑图的方式。***///*

3.2.2、性能测试环境物理架构

*///*说明本项目性能测试环境的物理架构，可以以物理架构图的方式。*///*

可使用 Visio 画出测试环境和生产环境的网络拓扑图。

测试环境的网络拓扑图(单 Web 服务器+单数据库服务器)很简单在此省略。

3.2.3、性能测试环境与生产环境资源对比

说明本项目测试环境与生产环境的差异，确定性能测试环境的软硬件资源，包括待测系统各组成部分的配置。下表供参考，非强制使用。

服务器	性能测试环境（规划）		生产环境（规划）	
	硬件配置	软件配置及 IP	硬件配置	软件配置
Web&DB 服务器	Cpu:i5 Disk:500G Memory: 8G	192.168.10.206	Cpu:i3 Disk:500G Memory: 8G	Os:Win7(64 位) WebServer:Apache2.4 DB:MySql5.5 PHP: 5.4.19
负载生成服务器	Cpu:i3 Disk:500G Memory:8G Net:100Mb 局域网	192.168.10.188 OS:Win7(64 位) LR 11	--	--

3.3、测试准备

3.3.1、测试环境安装

*/*说明本次测试的测试环境安装情况。*/*

XAMPP 集成环境:

控制面板 1.2.6

phpmyadmin 版本: Version 4.0.8

php 版本: PHP 5.4.19 (cli) (built: Aug 21 2013 01:12:03)

apache 版本: Server version: Apache/2.4.4 (Win32)

mysql 版本: Ver 5.5.32

负载机: LoadRunner 11(patch3+patch4)(192.168.10.206)

3.3.2、测试工具

*/*说明本次测试使用到的测试工具和监控工具。*/*

浏览器: IE11，Chrome43

协议抓包: HttpWatch9.3

性能脚本: LoadRunner 11

监控工具: Monyog MySQL 和 LoadRunner11 Controller。

测试数据图表生成: ECharts

……

目录:

你是如何成为一名捉虫师的？

--测试校招漫谈

◆作者：刘琛梅

转眼又到一年校招季。在近几年的校招工作中，我发现在面试测试职位时，我越来越遇不到男同学了，几乎清一色都是女同学。而且几乎所有的选择做测试的女同学，都强调说自己细心，有责任心，就算做重复、低级的工作也不会有怨言。发现这个现象后，我就会在校招时有意识的找一些男同学聊天，聊得不错的，就会说你来做测试吧，谁知这些男同学几乎都会面露难色，表示自己不合适不合适，说测试是女生该做的职业……

我这才知道，原来在同学们现在的观念中，测试变成了一个“女性专属”的职业。

后来我向几位高校教计算机老师朋友吐槽，说怎么感觉现在的同学们都认为测试是没有什么技术含量的工作，好像在观念中都是适合女生的职业。这时，几个要好的老师朋友就给我说，哎呀，你不要生气呐，我们还真是这样给同学分析的，专业课差一些的，编程能力不强的，都可以从测试开始做，简单一些，等积累够了能力，再转开发。这我真是惊得都嘴都合不上了。

再看看百度、知乎，好吧……“枯燥无趣”、“能力要求低”、“不需要编程技能”、“轻松”等等都变成了测试的标签。

作为做了十年测试，至今还是对测试充满感情的老人，真的觉得非常不开心。

很多同学在面测试时都会提到自己足够细心、耐心。但细心、耐心绝对不是只有测试才需要的独特素质。不是只有测试才有重复劳动，很多时候开发要做的事情比测试更重复，更需要耐心。

有很多同学认为测试不需要写代码，这也是不对的，你对你的被测对象都不了解，如何更好的理解你的被测对象呢。更不是只有自动化才要涉及代码。测试至少要会编写脚本，通过脚本来帮我们模拟用户的各种操作，模拟业务，帮我们更好的来做一些稳定性、压力、可靠性方面的测试，更充分的测试产品，这就是测试者的基本能力，和你做

开发还是做测试真是一点关系都没有。

是不是基础不好的，专业课不好的同学就更适合做测试呢？在我看来，答案也是否定的。和开发相比，测试需要从系统的角度来思考问题，这就需要各种专业知识。在公司中，我们经常看到一个测试身边会围站几个开发，测试抛一个问题出来，几个开发就开始从各自的角度进行分析判断，然后测试会再根据开发的分析继续抛一些问题出来，如果没有良好的专业基础，测试怎么能够快速做出分析并继续引导问题的定位？

所以我一直认为测试和开发，对校招的同学来说，在细心程度、编程基础、专业基础上的要求都应该是一样的。除此之外，还有一些对测试来说，特别重要的特质，就是“独立分析和思考能力（白话说法就是有主见）”，“逻辑能力”，“坚持和韧性”，以及“良好的沟通协调”和“文档编写能力”。

我认为对测试来说，“有主见”是最为重要的一项素质。试想一位测试人员，如果没有主见，别人（如开发）怎么说，他就怎么测，表面上看起来很和谐的样子，但最后一定是产品的灾难。

这是因为当一个完整的用户需求，被打散拆解后，由不同开发去完成，就像大家都在拧一个魔方一样，很难不乱。然而当局者迷，开发自己往往还不能看清产品的魔方已经乱了。这时就需要测试能够完整、独立的基于用户来分析需求，能够基于用户需求去测试，去发现产品魔方的问题。

对用户需求进行测试，可以归为黑盒测试。不知道从什么时候开始，黑盒测试变成了最没有技术含量的测试。很多同学在面试的时候，都会说自己准备做 1、2 年低级的黑盒测试，然后就去做高级的白盒测试。其实黑盒测试真的没有大家想的那么简单，相反，黑盒测试很难，难在对需求的理解和对系统的整体把握。

独立思考和分析能力（有主见），也是我在面试中最看重的能力。对我来说，你可以不懂数据库，不懂 linux，不懂网络，这些都不是难题，只要你肯学，短期就能学，但“主见”可不是一下子就学得来。

其次是逻辑能力。

大多数产品功能的预期没有标准答案，需求也不可能写得很细。很多时候测试判断输出的标准，就是是否符合逻辑，另外测试一般熟悉多个特性，还可以从整体上来判断功能输出是否有明显的矛盾，是否符合产品的整体风格。此外，逻辑性强的同学一般思

路也比较清晰。测试设计需要强大思路，沟通协调也是需要以逻辑为基础的。所以这项能力我也觉得尤为重要。

坚持和韧性，是因为搭建环境，执行，复现问题，需要的绝对不仅是耐心，还有韧性。特别是复现问题，尤其需要坚持和韧性。

最后就是沟通协商和良好的文档编写能力。测试需要有很强的推动力。从测试数据中很容易发现研发的问题，如何去改善这些问题，如何持续改进，这些都需要靠沟通协商和文档编写。

测试和开发倾向的能力确实有所差别。如果你有主见，逻辑性强，沟通表达能力和分析能力不错，我建议你试试测试。如果你性格中还有些好奇，追求完美，你一定要试试测试。另外我还发现我身边的测试朋友大多很文艺，喜欢艺术的特别多（我自己也喜欢画画），范儿很足，如果你也是这样，就一起来玩吧。但是如果你是因为能力不足，基础差而想退而求其次去选择测试，我的建议是 NO，你应该去提升能力，打牢基础。如果同学想更好的了解测试，可以去较大的公司实习，多和做测试的师兄师姐沟通，不要光问度娘，也不要太信老师的话，要自己看看这个世界究竟是怎样的，再来独立分析。

当然测试行业也需要男女平衡，我希望更多的男生能够来做测试。其实我最佩服的几位测试达人，都是男性。

你心中好的测试用例是怎样的？

◆ 作　者：刘琛梅

测试用例对测试来说就是日常的工作，重要性无需过多描述。但你心中好的测试用例是怎样的呢？需要满足怎样的标准呢？

在我的理解中，优秀的测试用例，应该是内容（业务需求）+形式（组织设计）都要棒棒的，在覆盖需求的前提下，有“设计感”（我喜欢的设计感是兼顾效率，减少冗余，能够以比较少的测试用例，来覆盖多的测试内容，当然，设计感还有很多），注重用例结构层次的设计，易于维护，能够被复用。为此，我设计了这个好用例的 check:

1	覆盖度（MFQ）	25%	
1.1	测试用例对需求的覆盖是否全面？（比如测试用例中有需求和用例的对应关系，有完善的评审记录等）		
1.2	测试用例中考虑了哪些测试类型，是否考虑完全？		
1.3	测试用例是否进行了功能交互分析？		
2	设计方法	25%	
2.1	使用了哪些测试用例设计方法（如PPDCS），以及方法的选择是否适合		
2.2	设计用例时是如何控制用例的粒度的？（是否会有有些用例很大很粗，有些用例很细很小的情况）		
3	用例的组织	25%	
3.1	用例的组织条理是否清晰？（需要分层分级来组织用例）		
3.2	测试者是否能够快速找到某条测试用例？		
3.3	用例在组织上是否存在较为明显的冗余		
3.4	用例的组织方式是否易于后续扩展		
4	用例的描述	25%	
4.1	测试用例的结构是否完整（包括用例编号，标题，预置条件，测试数据，测试步骤，预期结果，说明等）		
4.2	通过测试用例标题是否可以让测试者了解到这个测试用例的意图（例如“在怎样的条件下，谁做了怎样的事情，得到了怎样的结果”来描述。不希望是一些参数的堆积）		
4.3	测试用例中是否又引用了别的测试用例（不希望在测试用例中又引用别的测试用例，这样不利于测试计划的安排，还会对后期的用例修改，维护和移植带来麻烦）		
4.4	测试用例中的测试步骤和预期结果是否能够很好的对应起来		
4.5	测试步骤的描述是否过于笼统而导致用例无法执行（例如反复，多次，长时间，大量等）		

在我的《四步设计测试用例》课程中，也为大家分享了一些测试用例的设计技巧，如需深入学习，可以查看我的课程 http://www.atstudy.com/course/83

服务端测试笔记

◆作者：脚踏实地

一、面向的人员

1. 所有希望从事服务端测试的人员；
2. 刚刚从事服务端测试，但还有些许困惑的同学。

二、能力的要求

1. 服务端完全不了解；
2. 服务端测试零基础；
3. Linux、mysql 使用零基础。

三、学习目的

1. 了解服务器测试环境，能根据具体业务搭建环境；
2. 掌握常用的 Linux 和 mysql 命令；
3. 能独立完成使用 bingo 框架开发的后端项目的测试任务；
4. 能使用 phpunit 框架编写自动化 case。

四、测试环境搭建

作为测试人员，需要最近本的测试环境是如何搭建的，以及一些配置项是什么意思。

1. LAMP 介绍

所谓的 LAMP 指的是 Linux+Apach+mysql+php，LAMP 三个源码包是自行下载，分别安装的，最后配置环境变量，这样就完成 linux 下配置 php 编译环境，才能进行 php 开发。

2. Lighttpd 介绍

- Lighttpd 与 Apache 区别

1) 在访问纯静态对象时，Lighttpd 速度更快，更理想；

2) Lighttpd 吞吐量更大，能使用较少服务器提供与 Apache 相同访问量的服务；

3) Lighttpd 具有非常低的性能开销，cpu 占用率低。

- Lighttpd 的安装

方法一： windows 进行下载，再将其传输到 Linux 环境进行解压安装

（1）进入 http://www.lighttpd.net/download 网站，下载 lighttpd 安装包至 windows 环境；

（2）打开 Linux 系统，进入终端对需要传输的安装包进行选择，输入如下命令：

rz –be;

（3）解压压缩包：tar zxvf 安装包名称，如

tar zxvf lighttpd-1.4.39.tar.gz;

方法二： 使用 Linux 下的 wget 进行获取

（1）访问 http://www.lighttpd.net/download 网站；

（2）右击想要下载的文件，选择复制链接地址，如 http://download.lighttpd.net/lighttpd/releases-1.4.x/lightt pd-1.4.39.tar.gz;

（3）使用 wget 命令对链接地址进行下载：wget 链接地址，如

wget http://download.lighttpd.net/lighttpd/releases-1.4.x/lighttpd-1.4.39.tar.gz

（4）对压缩包进行解压：tar zxvf 安装包名称，如 tar zxvf lighttpd-1.4.39.tar.gz;

3. Lighttpd 的配置

（1）新建一个安装目录：mkdir lighttpd 安装文件夹名称，如

mkdir lighttpd_install;

（2）获取当前文件夹的绝对地址：#pwd

（3）进入源码文件:cd 源码文件夹名称，如，cd lighttpd-1.4.39;

（4）运行 configure 配置文件：./configure --prefix= lighttpd 安装文件夹绝对地址，如，./configure --prefix=/home/zyy/Desktop/lighttpd_install/

注：prefix 取的是 lighttpd 安装文件夹的绝对路径，如果是相对地址，则不能配置成功;

（5）进行编译：make

如果编译不成功，报错如下:

ERROR: pcre-config not found, install the pcre-devel package or build with --without-pcre

这是说明缺少一个 pcre 包，需要下载安装 pcre 包

具体安装 pcre 包步骤见下:

- 进入 http://downloads.sourceforge.net/project/pcre/
- 复制想要下载的 pcre 压缩包的链接地址，如

https://sourceforge.net/projects/pcre/files/latest/download?source=files

- 下载 pcre 压缩包：wget 压缩包链接地址，如

wget https://sourceforge.net/projects/pcre/files/latest/download?source=files

如果下载失败，出现如下报错:

ERROR: certificate common name “*.sourceforge.net” doesn’t match requested host name “sourceforge.net”.

To connect to sourceforge.net insecurely, use ‘--no-check-certificate’.

- 在以上命令中间加上--no-check-certificate ，如

wget --no-check-certificate

https://sourceforge.net/projects/pcre/files/latest/download?source=files

- 下载成功，解压压缩包：tar zxvf 压缩包名称，如

tar zxvf pcre-8.38.tar.bz2

如果解压失败，出现如下报错:

ERROR: gzip: stdin: not in gzip format

tar: Child returned status 1

tar: Error is not recoverable: exiting now

- 去掉解压命令中间的 z，重新解压：tar xvf 压缩包名称，如

tar xvf pcre-8.38.tar.bz2

- 解压成功，新建安装目录 mkdir pcre 安装文件夹名称，如

mkdir pcre_install

- 获取 pcre 安装文件夹所在位置的绝对地址：#pwd 命令
- 进入源码文件：cd pcre 源码文件名称，如 cd pcre-8.38
- 运行 configure 配置文件：./configure --prefix=pcre 安装文件夹所在位置的绝对地址，如./configure --prefix=/home/zyy/Desktop/pcre_install/

注：prefix 所取的是 pcre 安装文件夹所在位置的绝对路径，若换成相对地址则不能配置成功;

- 如果配置不成功，出现如下报错:

  You need a C++ compiler for C++ support,

- 安装 gcc-c++，运行以下命令：yum install -y gcc gcc-c++
- 如果和原有 gcc 版本冲突，需要将 gcc，cpp, libgomp 等等依赖先删除掉，运行以下命令：rpm -Va --nofiles --nodigest
- 重新安装 gcc-c++，运行以下命令：yum install gcc-c++
- 安装成功，重新运行配置文件:

./configure --prefix=/home/zyy/Desktop/pcre_install/

- 配置完成进行 pcre 编译，运行如下命令：make;
- 编译成功进行安装，运行如下命令：make install;
- 安装成功，进入 pcre 安装目录下的 bin 目录，运行如下命令:

cd ../pcre_install/bin/

- 获取 bin 目录的绝对路径，如：#pwd

获取的 pcre 安装目录下的 bin 目录绝对路径为：/home/zyy/Desktop/pcre_install/bin

- 进入.bashrc 配置文件，如，#vi ~/.bashrc

注：~表示根目录

- 按 i 进入编辑模式，将绝对路径添加到 PATH 行，如

export PATH=/home/zyy/Desktop/pcre_install/bin:$PATH

注：export 区分大小写，若不小心写成 Export，系统会报错-bash: Export: command not found

- 按 Esc 退出编辑模式，输入：wq 进行保存退出
- 执行.bashrc 文件，运行以下命令：source ~/.bashrc

如果没有报错则说明 pcre 安装配置成功

（6）退出 bin 文件，进入文件夹 lighttpd-1.4.39，如 cd ../lighttpd-1.4.39

（7）重新执行配置文件

./configure --prefix=/home/zyy/Desktop/lighttpd_install/

（8）配置失败，缺少 zlib-devel 相关文件，出现如下报错信息：

configure: error: zlib-headers and/or libs where not found, install them or build with --without-zlib

（9）下载安装 zlib-devel 相关文件，运行以下命令同时安装，如 yum install gcc glib2-devel openssl-devel bzip2-devel gzip-devel zlib-devel

（10）安装完成，重新进行配置

./configure--prefix=/home/zyy/Desktop/lighttpd_install/

（11）配置成功，进行编译，运行以下命令：make

（12）编译成功进行安装：make install;

（13）安装完成之后需要将 doc/config 目录下 lighttpd.conf 拷贝到相应的目录下并做修改。

- 回到 lighttpd 安装目录：cd ../lighttpd_install/
- 建立一个安装目录对配置文件进行管理：mkdir config
- 进入配置文件夹：cd config/
- 将 doc/config 目录下 lighttpd.conf 拷贝到当前目录

cp -r ../../lighttpd-1.4.39/doc/config/* .

注意：此时拷贝的是整个文件夹内容，需要有参数-r，其中后面的.指的是当前目录。

（14）获取配置文件夹的绝对路径，运行命令：#pwd

获取的绝对路径：/home/zyy/Desktop/lighttpd_install/config

（15）编辑 lighttpd.conf 文件：vi lighttpd.conf

- 设置以下路径:

```
var.log_root    = "/home/zyy/Desktop/lighttpd_install/log"
var.server_root = "/home/zyy/Desktop/lighttpd_install/"
var.state_dir   = "/home/zyy/Desktop/lighttpd_install/"
var.home_dir    = "/home/zyy/Desktop/lighttpd_install/"
var.conf_dir    = "/home/zyy/Desktop/lighttpd_install/config"
```

- 查找 sp modules.conf 复制其内容:

```
server.modules = ("mod_access",)
```

- 找到 include "modeles.config"，在下面输入以上复制的内容，并在其前面输入#将其注释掉。
- 修改端口号，一般改为 8000 到 9000 之间的任意值，例如修改为 8099;

注意：该端口号需要记住，后面验证需要使用端口号。

- ipv6 一般不使用，可以设置成"disable";
- 下面的用户名和组名可以写成自己的名字：如都设成"zyy";
- 修改完成输入：wq 保存退出

（16）创建实例程序，重启 lighttpd，看 lighttpd 是否搭建成功

- 回到 lighttpd_install 目录，运行命令：cd ..

- 新建 log 目录作为日志文件夹，htdocs 目录存放源文件:

mkdir log

mkdir htdocs

- 进入 htdocs 目录，新建 index.html 文件:

cd htdocs

touch index.html

- 编辑 index.html 文件:

vi index.html

输入 hello 进行保存退出

- 回到 lighttpd_install 目录，运行 cd ..
- 重启 lighttpd，运行以下命令

./sbin/lighttpd -f config/lighttpd.conf -m lib/

- 重启失败，出现如下报错:

opening errorlog failed: Permission denied

说明权限不够

- 修改 lighttpd 安装目录的所有者和组的权限为 zyy，如

chown -R zyy:zyy /home/zyy/Desktop/lighttpd_install/

- 重启 lighttpd，没报错的话就说明 lighttpd 配置成功，如

./sbin/lighttpd -f config/lighttpd.conf -m lib/

（17）验证配置，获取 IP，执行以下命令：ifconfig

（18）打开浏览器，访问以下链接出现 hello 说明安装配置成功

ip: 端口号/index.html

4. php

1）安装 php-5.6.2

（1）进入 http://php-fpm.org/downloads 下载相应的源码包;

注：只有 5.3.3 之前的版本安装需要

（2）进入 http://museum.php.net/php5;

（3）复制 php-5.6.2 的链接地址;

（4）下载 php 压缩包：wget 链接地址，如

wget http://museum.php.net/php5/php-5.6.2.tar.gz;

（5）解压：tar zxvf php-5.6.2.tar.gz;

- 配置 php-5.6.2

（1）新建 php 安装文件夹：mkdir php_install;

（2）进入 php 源码目录 cd php-5.6.2

（3）对 php 进行配置，如下:

./configure -prefix=/home/zyy/Desktop/php_install

（4）若没有安装 libxm 包，会出现如下报错

error: xml2-config not found. Please check your libxml2 installation

（5）检查是否安装了 libxm 包，运行以下命令:

rpm -qa |grep libxml2

有的话重新安装，运行

yum install libxml2

yum install libxml2-devel -y

（6）重新配置文件

./configure -prefix=/home/zyy/Desktop/php_install

4） 编译 php-5.6.2，要使 16 个进程并行编译，运行 make -j16 ；

5）安装 make stall;

6）修改文件名称

- 安装之后检查是否有 php.ini-development 文件，运行 ll;
- 将其拷贝至 php_install 目录下 lib 目录下

cp php.ini-development ../php_install/lib/

- 进入复制的目录，检查是否复制成功

```
cd ../php_install/lib/
ll
```

- 将 php.ini-development 文件重命名为 php.ini 文件

```
mv php. ini-development php.ini
ll
```

7）修改 lighttpd 的配置文件

- 打开 lighttpd 的配置文件

vi lighttpd_install/config/lighttpd.conf

- 查找 server.modules，添加 "mod_fastcgi";
- 在最后加上

```
fastcgi.server = (
".php" => ((
"host" => "127.0.0.1",
"port" =>"9005",
"bin-path" => "/home/zyy/Desktop/php_install/bin/php-cgi"
)))
```

注：bin-path 是安装的 bin 文件的绝对路径，需要根据实际安装目录进行设置。

- 修改端口号 8098;

8）重启 lighttpd

./sbin/lighttpd -f config/lighttpd.conf -m lib/

9）验证配置成功

- 进入 lighttpd_install 下的 htdocs

cd lighttpd_install/htdocs

- 打开页面

vi index.php

- 输入 i 进入编辑模式，输入以下内容:

```
<?php
phpinfo();
?>
```

输入：wq 进行保存退出。

- 浏览器验证，输入 http://localhost：8098/index.php

出现 PHP Version 5.6.2 配置参数的相关信息说明配置成功。

五、Linux 常用命令讲解

1. ls 命令

-a:列出当前目录下的所有文件和目录，包括以点开头的隐藏目录;

-l:列出文件和目录的详细信息;

-l | grep '^d':筛选满足条件的子目录;

注：^代表开头

-l | grep '^d'| wc -l:计算当前目录下子目录的数量

注：-l 记录一共有多少行；-c 表示记录一共有多少个字符。

sed: 列出文件的绝对路径

ls | sed "s:^:`pwd`/:g"

g 代表全局替换

2. cd 目录名

1）/表示根目录;

2）~表示当前用户目录;

3）..表示上一级目录;

4）-表示返回进入此目录之前所在的目录。

3. mkdir 命令

1）-p: 递归创建目录，如 mkdir -p test1/test11

2）-m:带权限创建目录，如 mkdir -m 777 test2

采用 ll 查看一下，可以看到，test2 的权限是可读可写可执行 rwxrwxrwx，一般情况下新建的目录是缺少写的权限的，即 rwxrwxr-x。

4. rm 命令

1)-r: 递归删除目录: 如 test 目录里面还有 test1，删除 test 目录就需要使用 rm -r test;

2)-i: 温和删除目录，提示用户确认是否删除，避免误删: 如删除 test 目录使用 rm -ri test 就会有提示

3）-f: 强制删除文件: 如想要删除目录 test，使用 rm -rf test 无任何提示。

4）rm -rf *.bar: 可以删除正则表达式匹配的文件

5）rmdir 可以删除空目录。

5. mv 命令

mv 源文件或者目录 目标文件或者目录

6. cp 命令

cp 源文件或者目录 目标文件或者目录

7. which 命令

which 可执行文件名称: 用于查找可执行文件的目录;

8. whereis 命令

Whereis 文件名: 定位可执行文件、源码文件以及帮助文件的目录;

9. find 命令

1）find . -type d ： 查找当前目录下的所有目录;

2）find . -name ‘*.log’: 查找指定文件所在的位置;

3）find . -name ‘*.log’-exec mv {} .. \;将当前目录下的.log 文件移至上一目录下;

注：find 后面可以跟-exec 执行后面的命令，此命令必须以;结尾，其中\是防止;进行转义。

4）find . -name '*.log' |xargs -i mv {} test: 将当前目录以及其子目录下的以.log 结尾的文件通过 mv 命令移至 test 目录下。

注：-i 表示前边的输出用{}代替。

10. windows 与 linux 之间传递文件的命令:

1）rz -bey: 从 windows 选择文件传递至 linux 下;

2）sz 文件名称：从 linux 传递文件至 windows 环境下，文件存放在虚拟机设置的下载目录下。

11. tar 命令

1）解压：tar -zxvf 文件名.tar.gz;

tar jxvf 文件名.tar.bz2;

2）压缩：tar -zcvf 文件名.tar.gz 压缩文件或者目录名字

tar jcvf 文件名.tar.bz2 压缩文件或者目录名字.

12. 空间查看命令 du

1）du -s: 显示文件所占的空间总的大小;

2）du -h: 以可读方式显示每个文件所占的大小。

13. diff 命令

1）diff 文件 1 文件 2: 查看两个文件的不同

2）vimdiff 文件 1 文件 2: 查看两个文件的不同

14. wc 命令

1）wc -l:计算字符数;

2）wc -w:计算字数;

3）wc -c:计算行数;

15. ps、grep、kill 命令组合

ps -aux |grep vim |kill -9 PID

16. ping 命令：测试目的主机的连通性

ping ip 或者域名

17. wget 命令

wget 下载文件的链接地址

18. scp 命令

scp 原路径 目标路径

六、mysql 常用命令讲解

1. 连接数据库

mysql -hIP -P 端口 -u 用户名 -p 密码

2. 数据库操作

1）新建数据库：mysql>create database databasename;

2）查看数据库：mysql>show databases;

3）删除数据库：mysql>drop database databasename;

4）进入数据库：mysql>use databasename;

5）退出数据库：mysql>exit databasename;

3. 表操作：

1）新建表格:

mysql> create table test1(name varchar(50),age char(10),sex int,area varchar(128));

2） 修改表格:

mysql>alter table tablename add column id int(10) primary key auto_increacement;将 id 设为标的主键。

3）删除表格:

mysql>drop table tablename;

4）添加表格数据:

mysql>insert into tablename（column1，column2, column3……） values(‘value1’, ‘value2’，‘value3’ ……);

注：前面的行数要与相应的数值在数量上对应上，否则会报错。

5）查看表格数据:

select * from tablename

6）删除表格数据:

delete from tablename where condition

7）更新表格数据:

update tablename set column1=value1 where condition

4. 查询语句:

select [all|distinct|top] *|expression[as output_name][,…]

[from tablename][where condition][group by expression[,…]]

[having condition[,…]][order by expression[asc|desc]]

[limit{count|all}]

注意事项:

1）where 跟的表达式可以是比较运算符（>|=|<）、范围运算符（between… and…|not between… and… ）、列表(in|not in)、（模式匹配（link）等。

2）limit{count|all}是需要筛选的记录数，如 limit 2 表示的是筛选两条记录。

例如:

- select * from tablename limit 2;

显示整个表格的两条记录;

- select * from tablename where id>1 limit 2;

显示 id>1 的两条记录;

- select * from tablename where id between 1 and 4 limit 2;

显示 id 在 1 到 4 之间的两条记录;

3）模式匹配 link，有两种通配符，一种是%表示 0 个或者多个字符，_表示单个字符。

例如:

- select *from tablename where title like “%le%” limit 2;

筛选出标题含有 le 的两条记录，前面后面都可以有 0 个或者多个字符。

- select * from tablename where title like “_le_” limit 2;

筛选的是标题含有 le 且前面含有一个字符，后面含有一个字符的 2 条记录。

- select rating,count（*）from tablename group by rating

按照 rating 列进行分组统计数量

- select rating,count(*) from tablename group by rating having rating >9.0

将分组的数据满足条件输出，可以用 group by 配合 having 进行使用。

- select rating,count(*) from tablename group by rating having rating >9.0 order by rating desc

将分组的满足条件的数据进行排序，就可以使用 order by 进行使用。

5. 备份数据库命令:

1）mysqldump -hip -Pport -u 用户名 -p 密码 数据库名 表名>导出的文件名

运用 source 导出的表的绝对路径

移动设备的配置测试

◆ 作 者：枫 叶

手机测试

手机软件应用在我们的生活中正逐渐流行起来。最近一个由安德里亚·史密斯在将来上发布的研究着重指出作为一个社会群体我们对我们的手机应用沉迷多少，以至于就像所说的“一些人承认一天用超过 50 的应用”。我们随处可见他们，排队等待，在街上散步，或者甚至参加运动会。实际上，82%的响应者，人们认为他们无法超过一天没有他们的手机应用并且如果这个示例反应社会性，然后这些移动应用需要工作得正确、持续并且符合用户需求。

越来越多的应用被创造。人们不仅要测试这些应用也要有如何在各种设备上测试这些产品的需要正成为一项相当的挑战。这篇文章将会描述一些对测试员来说在他们的手机应用测试项目里考虑的适配测试。

上个月，我参加了一个关于使用脸谱作为国内的应用的手机设备适配测试的周末美国测试员会议。我们有一系列的设备，苹果手机，安卓手机，平板和安卓平板的组合。虽然各种移动设备和操作系统，测试在使用相同的移动应用方面体验了彻底的不同。结果：测试们关注着开放的体验并且当他们回去测试他们自己的移动应用时可以去拓宽他们的视野。

我们周末的测试会议发现了以下一些观点：

- 整理新闻供应的发布在使用的设备上呈现出不同。
- 脸谱应用在设备可显示区域的尺寸大小上显示信息，这个区域上平板比手机展现更多信息。

- 在所有三个配置（平板，台式机/笔记本和手机）默认显示和功能设置改变了，像好友列表，刷新设置，时间戳发布
- 搜索功能在平板上与台式机/浏览器应用表现不同
- 了解了如何在设备配置上干活并不意味着你将自动知道怎样去在另一个配置上展示一个功能-为那些从配置到配置做切换的用户。

你想过在一个设备屏幕上出现的东西和使用一个不同设备的输出的测试吗？甚至在不同的安卓手机中，你将会看到在视觉区域的物理尺寸的不同。当设计你的测试用例时，你要做的不仅是考虑手机应用的通常真实评估，而且应用的视觉在不同尺寸设备上的每个可识别呈现。脸谱本土应用填满了在 4 英寸屏幕和 5.5 英寸屏幕的安卓手机屏幕吗？现在问题出来了，你是如何对这些不同做自动化测试呢？你应该使其自动化吗？这个一个测试或许不值得自动化，尤其是如果这部分代码从一个版本发布没有被修改。不是所有的测试需要被自动化而且由于手机应用对公司来说正显得如此重要，因此公司去生产、测试项目需要很细心地计划并且什么时候投入为手机测试执行自动化是很关键的。

和在你的手机移动应用上的脸谱应用吗？甚至所有的配置共享相同的操作系统，他们是几乎完全不同的应用或者代码版本呈现出根本的不同。所以你怎样基于在一个手机应用上计划你的测试？这些是你去考虑的一些事；设备的不同配置，设备的旋转，展现出的是什么并且它会改变可见的功能吗？这些可能不仅在一个发布里计划一次的测试而且应该在某一时刻来说是发布的一部分。

在手机应用版本，图标取决于配置而呈现出不同。一些为“易用性”而考虑的测试和从一个到另一个配置的转换需要被考虑。什么组合成易用性？谁决定这个定义？目前这些因素必须被在设计和开始代码前被“测试”出来。记得，作为一个测试，你需要有被清楚定义的需求或者在你的应用如何在每一个配置上使用的一个清晰的理解。如果不这样，一个“无缝体验”的缺乏会对一个公司的市场声誉有破坏性的影响。

什么是可学习性？你的用户典型地从一个配置切换到另一个吗？包含可视和功能的配置之间的转换应该被考虑成为发布的一部分。随着一些移动手机应用比他们的对应平板电脑版本更不同，这种切换对用户来说舒服吗？为“舒适”“可用性”的测试是一项主观的认为。手机测试者需要知道更多关于他们的用户和他们如何使用这个应用。这就是销售，市场和其他一些面向客户的团队成员们能与测试们的经验和用户故事分享的地方。

因为我们已进化到使用这么多的手机应用，人类的爱好和偏见已经在我们的思想中建立。我们有不同在展示，使用性，反馈时效和功能性上的期望。那些不是更经常使用笔记本和台式机去指导我们的日常活动而是而是更典型地同样使用我们的手机应用因为一些人没有拥有笔记本或者台式机。他们在使用上的偏见是完全不同的，因为“易用性”有一个不同的意义。我们的测试因此需要考虑不同的在配置上的期望。

当使用手机应用时网络连接性也是另一个配置的测试考虑。一个平板电脑比如说，通常在固定的地点比如电视机前的扶手椅或者在喜欢的咖啡店里。只要连接被建立，因为缺乏移动几乎没有波动。当你步行或者在一个移动的自行车使用网络有多频繁？如果你的应用要求网络连接，你将需要在配置基础上增加适当的测试。

对于移动设备和移动应用测试存在的特殊性有多少不同的种类？这篇文章是在移动设备上的对通用功能性，使用性，不同配置的呈现考虑的引荐。为什么一个开发设计一个移动应用会在一个桌面应用或者平板应用出现很大不同？所有的测试没有应用到所有的配置。使用性的定义需要在需求上被细心地量化并且应用的使用性可能依赖于客户所期待的使用应用的特殊市场。与你的利益相关者紧密工作会学到更多只要你想学的关于用户/顾客和他们的观点。

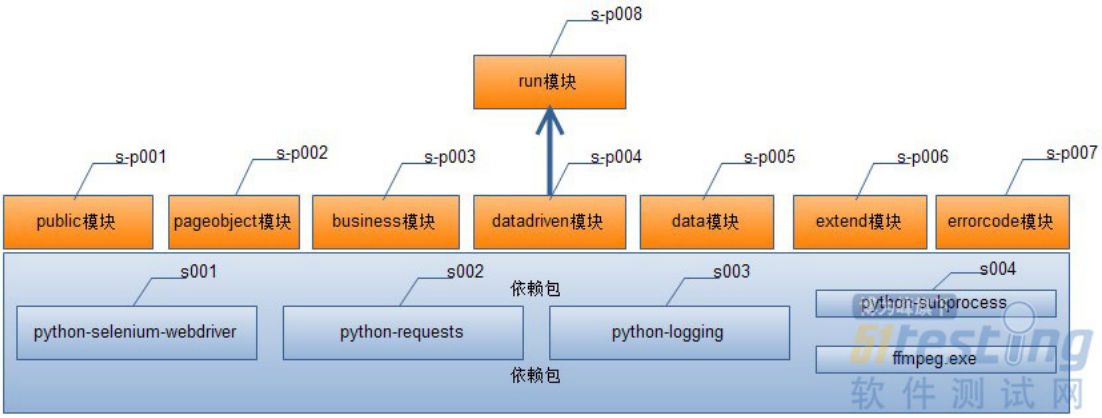
最后，继续在移动设备和移动应用上实践测试。越多时间花在移动应用上的测试，越多灵感（更好的心理模型）一个手机测试者将得到的就像采用的那些测试种类。不同性能测试，通知测试和网络通信测试会使用，和传统的功能和行为测试。学习有更多种类的测试在测试图形用户界面功能性之外对策划你的手机测试项目是关键的。

拓展学习：安卓 APP 测试之性能测试

用 Pageobject 的方式使用 Webdriver

◆作者：文 字

一、代码引用结构



S001：主要调用 webdriver 中以下文件及类，这些类是 webdriver 运行的基本依赖。Public 模块对此类依赖进行了再次封装。

```
import selenium.webdriver as webdriver
import selenium.webdriver.common.by as by
import selenium.webdriver.support.wait as Wait
import selenium.webdriver.support.expected_conditions as Expect
from selenium.webdriver.common.action_chains import ActionChains
```

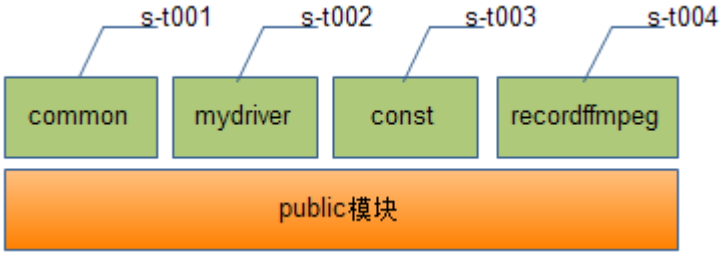
S002: requests 包主要供调用 api 或者执行 web-api(浏览器中执行的请求)使用。extend 模块实现了其再次封装以及实例的实现。

S003：自定义了日志输出的模式，在 public 模块中实现了相类的类。

S004：subprocess 模拟调用 ffmpeg.exe 执行命令，实现录像的操作。在 public 模块中实现了相关的类。

二、具体模块的实现

- **S-p001：public 模块主要包括以下几个文件**



S-t001：common 文件主要包括以下方法

def log(words, filename=time.strftime('%Y-%m-%d') + ".log", mode="a"):

写自定义日志文件

def to_request_cookies(webdriver_cookies):

将 webdriver cookies 转换成 requests cookies

S-t002：mydriver 文件主要包括以下方法等等。

其主要目的是增强查找对象的可靠性，以及在 webdriver 操作层对可重复使用的方法进行公共抽象。

def wait_find_element(self, tupleBy, time = 10, hlight=False):

尝试 5 次等待寻找对象，且每次对象的超时时间为 10 秒，如果 hlight 为 true 则高亮对象。找到对象则返回该对象。

def wait_find_elements(self, tupleBy, time = 10, hlight=False):

尝试 5 次等待寻找多个对象，且每次对象的超时时间为 10 秒，如果 hlight 为 true 则高亮对象。找到对象则返回该对象 list。

def wait_subchild_element(self,father_element,tupleBy, times = 10, hlight=False):

多层对象的查找

def wait_get_style_display(self,typleBy):

获取对象的 style 中的 display 值

def wait_find_element_mousover(self,tupleBy):

将光标移动到对象中

下例展示了 wait_find_element 的实现

```
def wait_find_element(self, tupleBy, time = 10, hlight=False):
    ok = False
    i = 1
```

```
            while (i <= 5):
                try:
                    wait_element = Wait.WebDriverWait(self.maindr, time).until(lambda maindr:
        maindr.find_element(tupleBy[0], tupleBy[1]))
                    com.log("find element    ok    " + tupleBy[0] + "        " + tupleBy[1])
                    if hlight:
                        self.high_light(wait_element)
                    return wait_element
                except Exception as e:
                    if (i == 5):
                        com.log("find element    fail    " + tupleBy[0] + "        " + tupleBy[1] + "
        " + str(e))
                finally:
                    i = i + 1
```

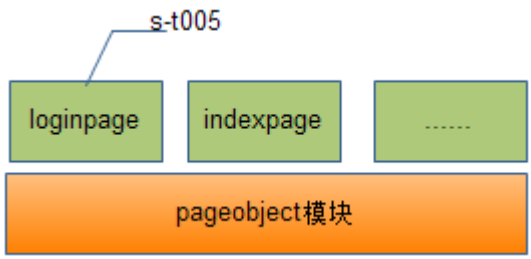
S-t003：const 即常用常量

如，HOST = "https://test.myfoscam.com"

S-t004：在 subprocess 中执行 ffmepg 命令

```
def start(self,filename):开始录像
self.save_filename = self.record_dir() + "\\"+filename+".avi"
self.delete_current_record()
cmd = "%s -draw_mouse 1 -offset_x 0 -offset_y 0 -f GDIgrab -i
desktop %s" %(self.get_ffmpeg_path(),self.save_filename)
# print "cmd = %s" % cmd
self.process = subprocess.Popen(cmd, stdout=subprocess.PIPE,
stderr=subprocess.PIPE,universal_newlines=True)

def stop(self):
停止录像
```

- **S-p002：pageobject 模块用以管理待操作对象**



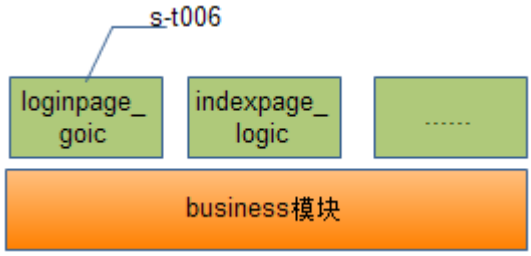
S-t005：pageobject 模块用以页面的方式来管理操作的对象。并且 pageobject 模块，调用 S-t002 中的 wait_find_element 方法用以驱动相关方法。其主要划分方法为，查找方式为类的属性、操作的过程为 sub 方法，检查的方式为 check 方法。其主要目的，是方便对象被修改后易于维护，且易用被其它地方调用的时候能够重用。

其具体实现如下例所示：

```
'''首页'''
class MainPage(mydriver.MyDriver):
myBy = by.By()
'''登录用户名'''
findByID_user_email = (myBy.ID, "user-email")
'''登录密码'''
findByID_user_password = (myBy.ID, "user-password")
'''登录按钮'''
findByID_login_button = (myBy.ID, "login-button")

'''登录错误提示语层'''
findByID_error_msg = "error-msg"

'''首页标题'''
check_main_title_string    = "Foscam Cloud Service"

'''登录过程'''
def sub_login(self, user_email, user_password):
    self.switch_default()

    object_user_email = self.wait_find_element(self.findByID_user_email)
    object_user_password = self.wait_find_element(self.findByID_user_password)
    object_login_button = self.wait_find_element(self.findByID_login_button)

    if object_login_button == None or object_user_email == None or object_user_password ==
None:
        return    None
    else:
        '''先清除输入框中的内容'''
        object_user_email.clear()
        object_user_password.clear()

        '''输入内容'''
        object_user_email.send_keys(user_email)
        object_user_password.send_keys(user_password)
        object_login_button.click()
```

```
    '''如果出现等待中则继续等待，直到尝试了 10 次后停止
       在登录时，期望 False
    '''
    def wait_login(self,bool_expect = True):
        bool_wait = True
        wait_times = 3
        i = 0
        while ( i <= wait_times):
            try:
                self.switch_default()
                if bool_expect == False:
                    expect_time = 2
                else:
                    expect_time = 10
                object_error_msg =
self.wait_find_element(self.findByID_login_button,time=expect_time)
                if object_error_msg == None:
                    bool_wait = False
                else:
                    if bool_expect == False:
                        time.sleep(10)
                        pass
                    else:
                        return object_error_msg
            except Exception as e:
                if i == wait_times:
                    return bool_wait
            finally:
                i += 1
        return bool_wait

    """
    检查登录失败时的提示语内容是否与指定的指符串一致
    """
    def check_login_faiure_string(self, faiure_string):
        object_error_msg = self.wait_login()
        if    object_error_msg != False:
            if faiure_string in object_error_msg.getText():
                return True
```

- **S-p003：business 模块用以构成测试的判断逻辑**



S-t006：用以调用 pageobject 中的方法，构成测试逻辑。其主要包括二部分，一是测试逻辑方法，二是回滚逻辑处理（即测试的期待结果与实际结果不一致时，如何回滚到初始该测试方法之前）。并且调用 errorcode 模拟中的预置错误码信息。详细的错误，可快速排查失败的原因。

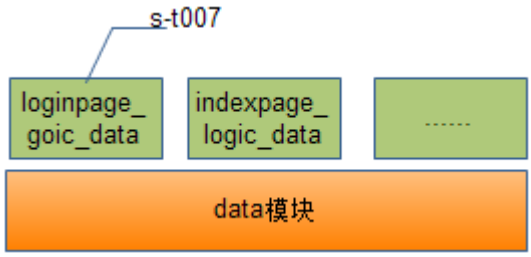
其具体实现如下所示：

logic_login2()为具体测试实现的逻辑，logic_login()为具有回滚操作的逻辑，实际供调用是 logic_login()。

```
class LogicLogin(mydriver.MyDriver):
def __init__(self):

    self.main_page = pageobject.main.MainPage()
    self.index_page = pageobject.index.Index()

def logic_login(self, user_email, user_password, faiure_string, expect_test_result=True,
                expect_rollback_check=True):
    flag = self.logic_login2(user_email = user_email,user_password =
user_password,faiure_string=faiure_string,expect_test_result =
expect_test_result,expect_rollback_check = expect_rollback_check)

    if expect_rollback_check:
        self.index_page.sub_logout(user_email)
        '''判断回滚是否成功，如果成功，则进行下一次数据的测试，否则 raise 出异常'''
        if self.main_page.wait_login() == False:
            raise "登录可能失败，并且回滚到登录界面时也可能失败了"
    return flag

"""
    判断逻辑:
    1.输入用户名、密码
    2.根据 expect_test_result(期待结果）进行逻辑判断
        expect_test_result == True 时
        如果登录界面不存在，并且跳转到的主页面中存在 Live Video 导航和用户名时，则
测试成功，写结果，并回到登录界面
```

```
            如果

            expect_test_result == False 时
            如果登录界面存在，且错误提示字符串存在时，则测试成功，写结果；
            否则，如果登录界面不存在了，则再次进行回滚操作
        """

    def logic_login2(self,user_email, user_password, faiure_string, expect_test_result = True,
expect_rollback_check = True):
        '''将当前对象 self.maindr 赋给 MainPage、Index 类的 maindr 属性'''
        self.main_page.maindr = self.maindr
        self.index_page.maindr = self.maindr

        result_flag = public.const.Fail
        bool_rollback = False
        '''登录过程'''
        self.main_page.sub_login(user_email, user_password)
        if expect_test_result == True:
            '''如果登录界面不存在'''
            bool_wait_login = self.main_page.wait_login(bool_expect = False)
            if bool_wait_login == False:
                '''并且用户名和导航栏中的字符串再次进行检查，存在'''
                # if self.index_page.check_login_success(user_email) == True:
                #     result_flag = public.const.Pass
                #     bool_rollback = True
                if self.index_page.check_login_success(user_email) == False:
                    return
com.result_flag(result_flag,errorcode.ERROR_LOGON_REASON_HAVNT_LOGINPAGE)
        else:
            if self.main_page.wait_login() != False:
                if self.main_page.check_login_faiure_string(faiure_string) == False:
                    return com.result_flag(result_flag,
errorcode.ERROR_LOGON_REASON_HAVNT_TIPS)
            else:
                bool_rollback = True
            """尝试进行回滚"""
            if bool_rollback == True:
                '''回滚到登录界面，供下次测试时再使用'''
                # if index_page.check_login_success(user_email) == True:
                self.index_page.sub_logout(user_email)

        """获取 webdriver 的 cookies，并赋给全局变量，供 webextend 中使用"""
        public.const.MyCookies = self.index_page.get_cookies_to_request_cookies()

        return com.result_flag(public.const.Pass, errorcode.ERROR_CODE)
```

- **S-p004：data 模块准备测试数据**

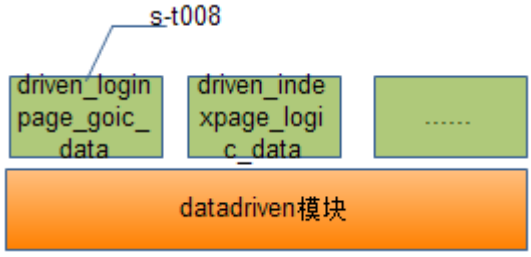


S-t007：以 key-value 的方式准备测试数据。该测试数据，可供非代码编辑人员照例准备。

如下例所示：

```
import public.const
 class DataLogin():
      @staticmethod
      def data_login():
          data_login = [{"testId": "login_001", \
                         "testTitle": "正确的用户名正确的密码登录", \
                         "user_email": public.const.UserName, \
                         "user_password": public.const.PassWord, \
                         "faiure_string": "",\
                        "expect_test_result": True,\
                        "expect_rollback_check":False,\
                        }]
          data_login.append({"testId": "login_002", \
                         "testTitle": "正确的用户名正确的密码登录", \
                         "user_email": public.const.UserName, \
                         "user_password": public.const.PassWord, \
                         "faiure_string": "",\
                        "expect_test_result": True,\
                        "expect_rollback_check":True,\
                        })
          return data_login
```

- **S-p005：datadriven 模块以业务数据来驱动测试逻辑**

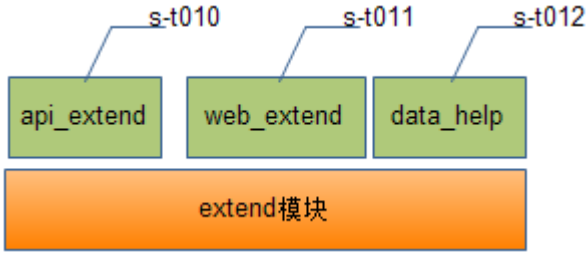


S-t008：结合 business 模块中的测试逻辑，以及 data 模块中的数据，进行多次驱动测试。并同时调用 public 模块中的 RecordScreen 类对每一个测试过程进行录屏，如果测试成功则删除掉该录屏文件，否则保存下来，供结果分析使用。同时调用相关的写入数据库的方法，将相关测试结果及准备的测试数据保存其来。

具体示例如下：

```
class DrivenLogin(mydriver.MyDriver):

def __init__(self):
    self.data_login = testdata.main_page_data.DataLogin.data_login()
    self.result = business.main_page_logic.LogicLogin()

def driven_login(self):
    '''将当前对象 self.maindr 赋给 logicLogin 类的 maindr 属性'''
    self.result.maindr = self.maindr
    for data in self.data_login:
        #    开始录屏
        record = RecordScreen()
        record.start(data['testId'])

        result_flag = self.result.logic_login(data['user_email'],\
                data['user_password'],\
                data['faiure_string'], \
                expect_test_result = data['expect_test_result'], \
                expect_rollback_check = data['expect_rollback_check'],\
                                                    )

        #    结束录屏
        record.stop()

        #    如果是 OK 的结果，则录屏文件自动消毁
        if result_flag['result_flag'] == public.const.Pass:
            record.delete_current_record()
        """根据结果，写库"""
        result_flag(result_flag,data )
```

- **S-p006：extend 模块扩展测试逻辑**



S-t010：供调用接口查询 web 界面中不存在的数据，供测试逻辑进行检查，或一些特殊的操作，如：

```
import requests
def del_ipc(ipc_id):
url =
public.const.Api_HOST+"/gateway&ipcSettingId=%s" %(public.const.OpenID,public.const.AccessTok
en,ipc_id)
r = requests.get(url,verify = False)
response = r.json()
# public.common.MyLog.log("删除设备返回=%s" % response)
# print response
if response['errorCode'] == '':
    return True
```

S-t011：供调用 web 界面操作时的请求，供一些特殊的测试场景进行相关的检查或特殊操作。同时该方法可以实现 web api 的自动化测试（web 界面访问时存在 session 控制，不同的页面 session 不一定能够多次使用，在使用 requests 测试 web api 时获取一些界面的 session 可能较困难，如果结合 webdriver 则实现起来比较容易），如：

```
import requests
import public.const
import  time
"""订单列表刷新，检查订单列表中的状态是否为 5，否则每隔 10 秒去获取，直到 1 个小时后
结束"""
def check_order_cancle(macaddr,currency,product_type,times = 60):
    url = public.const.HOST + "/order/ancel"
    body = {"page":"1","language":"ENU"}
    bool_return = False
    i = 0
    while True:
        r = requests.post(url,data=body,cookies=public.const.MyCookies,verify = False)
        response = r.json()
        if len(response) > 0:
            for order in response:
                if order['ipcMac'] == macaddr:
```

```
                    if order['currency'] == currency :
                        if order['tradeStatus'] == 5:
                            return True
            time.sleep(60)
            if i == times:
                break
            i = i + 1
        return False
```

注意：该方法能够执行是由于测试逻辑代码（上例有）执行了 web-api cookies 到 requests cookies 的转换。

public.const.MyCookies = self.index_page.get_cookies_to_request_cookies()

S-p007：errorcode 模块

Errorreason：错误码的常量文件，如：

```
# coding=utf-8
ERROR_CODE = ""
ERROR_LOGON_REASON_HAVNT_LOGINPAGE = "登录界面仍存在"
ERROR_LOGON_REASON_HAVNT_USERNAME = "用户名没有显示在导航栏中"
ERROR_LOGON_REASON_HAVNT_TIPS = "登录失败时的错误提示语没有找到"
ERROR_ADDDEV_REASON_HAVNT_SEARCHPAGE = "1 分钟都没有进入搜索界面"
ERROR_ADDDEV_REASON_HAVNT_SEARCHDEV = "1 分钟都没有搜索到相关设备"
ERROR_ADDDEV_REASON_HAVNT_MANUALLY = "没有进入手动输入 UID/DDNS/IP 的界面"
```

S-p008：run 模块

调用 datadriven 的方法，驱动测试。

```
import datadriven.main_page_driven
import datadriven.choose_page_driven
import datadriven.purchase_function_driven

import selenium.webdriver as driver
import public.const
```

```
import time
import extend.apiextend

if __name__ == "__main__":

    """首登录一下，供调用 api 获取 openId 和 accessToken"""
    extend.apiextend.login()
    """启动 webdriver"""
    start_driver = driver.Ie(executable_path =
r"IEDriverServer.exe",log_level='INFO',log_file=r"driver_"+time.strftime('%Y-%m-%d') + ".log")
    '''最大化窗口'''
    start_driver.maximize_window()
    start_driver.get(public.const.HOST)

    '''-------------------------------------------------登录'''
    datadriven_login = datadriven.main_page_driven.DrivenLogin()
    datadriven_login.maindr = start_driver
    datadriven_login.driven_login()
    '''-------------------------------------------------登录----------------结束'''
```