# 1软件测试的定义

软件测试就是利用测试工具按照测试方案和流程对产品进行功能和性能测试，甚至根据需要编写不同的测试工具，设计和维护测试系统，对测试方案可能出现的问题进行分析和评估。

# 2软件测试的方法

白盒测试，通过去看程序的具体运行，去找出运行程序的问题，需要对代码有一定的了解。

黑盒测试，在不知道代码细节的前提下，站在用户的角度上，去检查所开发的软件或app功能是否符合需求，或者说达没达到我们之前预期所想要的结果。

灰盒测试，是介于白盒测试和黑盒测试之间的一种方法，不光是看产品的功能是不是符合我们所需要的功能，而且也会去看代码是不是有问题，有的时候输出结果是对的，但是内部可能是巧合或者其它原因导致，可能在运行过程中受其它影响就会出现错误，所以这时就需要灰盒测试方法。

其它测试方法还包括单元测试、集成测试、系统测试、回归测试、验收测试、等价划分法、边界值方法、因果图法、判定表法、正交排列法等等。并且还会用到自动化测试工具，其中需要掌握的测试工具包括：性能自动化测试工具loadrunner、功能自动化测试工具QTP和现在需要了解的手机移动端测试方法。

Appium是一款开源的跨平台测试自动化工具。

# 3软件测试的工作内容

（1）使用各种测试技术和方法来测试和发现软件中存在的软件缺陷；

（2）测试人员将发现的缺陷编写成正式的缺陷报告，提交给开发人员进行缺陷的确认和修复；

（3）测试人员需要分析软件质量。在测试完成后，测试人员需要根据测试结果来分析软件质量，包括缺陷率、缺陷分布、缺陷修复趋势等。给出软件各种质量特性包括有功能性、可靠性、易用性、安全性、时间与资源特性等的具体度量；

（4）测试人员为了更好更有效地进行测试，保证测试工作质量，需要在执行测试工作之前首先需要设计测试用例，形成测试用例报告。

# 4 Web功能测试

（1）链接测试：在这方面主要考虑所有链接页面是否存在，所有链接是否按指示的那样确实链接到了该链接的页面，还要测试web应用系统上没有孤立的页面；

（2）表单测试：表单一般指在界面进行和数据提交操作的，包括新增和修改数据，它具体还要涉及到输入框测试、下拉框测试、图片、视频、Excel、TXT等文件上传测试，表单提交按钮测试等；

（3）搜索测试：搜索条件一般为2种情况:输入框、下拉框。对于多个条件的页面搜索可以按照下面的编号顺序去进行测试；

（4）删除测试：删除时要测试具体删除条件，例如没选择数据进行删除时界面是否会有提示，选择一条数据与多条数据时有什么不同提示；

（5）cookies/session测试 ：cookies/session通常用来存储用户信息，cookie通过在客户端记录信息确定用户身份，session通过在服务器端记录信息确定用户身份；

（6）数据库测试：在web应用技术中，我们在进行功能测试的任何操作都可以归溯到是对于数据库的增删改查。所以数据库测试实际就是通过前面讲过的测试手段来完成，只不过关注点是数据库表及表字段值的变化。

# 5 Web性能测试

## 5.1 [web](javascript:;" \t "http://www.51testing.com/html/44/_self)性能测试指标

[web](javascript:;" \t "http://www.51testing.com/html/44/_self)性能测试基本性能指标：

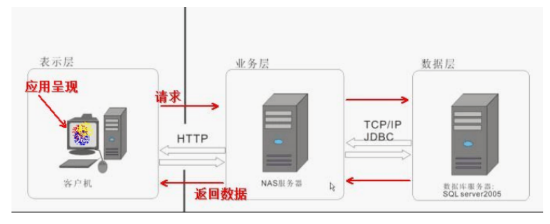
（1）客户发送请求；

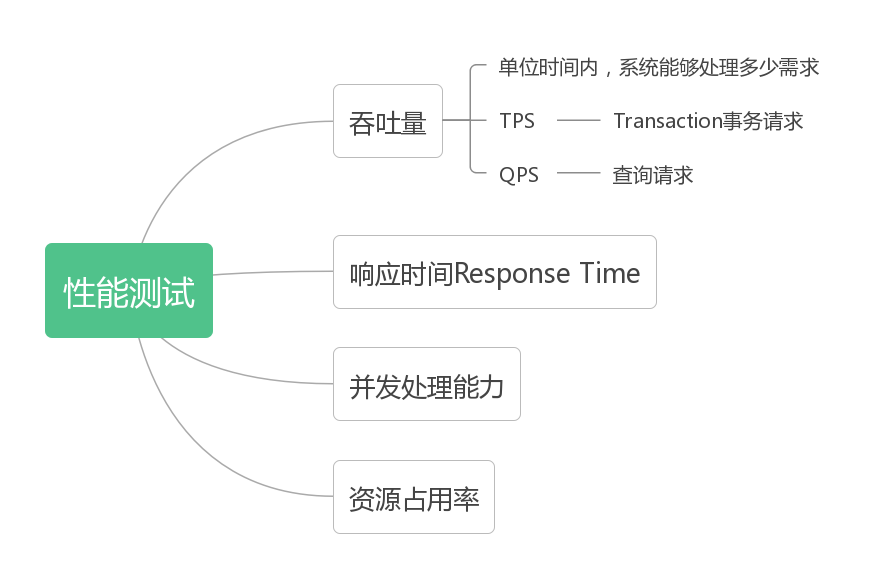
（2）web [server](javascript:;" \t "http://www.51testing.com/html/44/_self)接受到请求，进行处理；

（3）web server向DB获取数据；

（4）web server生成用户的object(页面)，返回给用户。给客户发送请求开始到最后一个字节的时间称为响应时间（第三步不包括在每次请求处理中）。

性能测试主要关注四个指标：吞吐量、响应时间Response Time、并发处理能力、资源占用率。





1.事务（Transaction）

在web性能测试中，一个事务表示一个“从用户发送请求->web server接受到请求，进行处理-> web server向DB获取数据->生成用户的object(页面)，返回给用户”的过程，一般的响应时间都是针对事务而言的。

2.请求**响应时间**

请求响应时间指的是从客户端发起的一个请求开始，到客户端接收到从服务器端返回的响应结束，这个过程所耗费的时间。在某些工具中，响应通常会称为“TTLB”，即"time to last byte"，意思是从发起一个请求开始，到客户端接收到最后一个字节的响应所耗费的时间，响应时间的单位一般为“秒”或者“毫秒”。一个公式可以表示：响应时间＝网络响应时间+应用程序响应时间。标准可参考国外的3/5/10原则：

（1）在3秒钟之内，页面给予用户响应并有所显示，可认为是“很不错的”；

（2）在3-5秒钟内，页面给予用户响应并有所显示，可认为是“好的”；

（3）在5-10秒钟内，页面给予用户响应并有所显示，可认为是“勉强接受的”；

（4）超过10秒就让人有点不耐烦了，用户很可能不会继续等待下去。

3.事务响应时间

事务可能由一系列请求组成,事务的响应时间主要是针对用户而言,属于宏观上的概念，是为了向用户说明业务响应时间而提出的.例如:跨行取款事务的响应时间就是由一系列的请求组成的.事务响应时间是直接衡量系统性能的参数。

4.**并发用户数**

并发一般分为2种情况。一种是严格意义上的并发，即所有的用户在同一时刻做同一件事情或者操作，这种操作一般指做同一类型的业务。比如在信用卡审批业务中，一定数目的用户在同一时刻对已经完成的审批业务进行提交；还有一种特例，即所有用户进行完全一样的操作，例如在信用卡审批业务中，所有的用户可以一起申请业务，或者修改同一条[记录](javascript:;" \t "http://www.51testing.com/html/44/_self)。

另外一种并发是广义范围的并发。这种并发与前一种并发的区别是，尽管多个用户对系统发出了请求或者进行了操作，但是这些请求或者操作可以是相同的，也可以是不同的。对整个系统而言，仍然是有很多用户同时对系统进行操作，因此也属于并发的范畴。

可以看出，后一种并发是包含前一种并发的。而且后一种并发更接近用户的实际使用情况，因此对于大多数的系统，只有数量很少的用户进行“严格意义上的并发”。对于WEB性能测试而言，这2种并发情况一般都需要进行测试，通常做法是先进行严格意义上的并发测试。严格意义上的用户并发一般发生在使用比较频繁的模块中，尽管发生的概率不是很大，但是一旦发生性能问题，后果很可能是致命的。严格意义上的并发测试往往和[功能测试](javascript:;" \t "http://www.51testing.com/html/44/_self)关联起来，因为并发功能遇到异常通常都是程序问题，这种测试也是健壮性和稳定性测试的一部分。

用户并发数量：关于用户并发的数量，有2种常见的错误观点。一种错误观点是把并发用户数量理解为使用系统的全部用户的数量，理由是这些用户可能同时使用系统；还有一种比较接近正确的观点是把在线用户数量理解为并发用户数量。实际上在线用户也不一定会和其他用户发生并发，例如正在浏览网页的用户，对服务器没有任何影响，但是，在线用户数量是计算并发用户数量的主要依据之一。

5.**吞吐量**

指的是在一次性能测试过程中网络上传输的数据量的总和。吞吐量/传输时间,就是吞吐率。

6.TPS（transaction per second）

每秒钟系统能够处理的交易或者事务的数量。它是衡量系统处理能力的重要指标。

7.点击率

每秒钟用户向WEB服务器提交的HTTP请求数。这个指标是WEB应用特有的一个指标：WEB应用是“请求-响应”模式,用户发出一次申请,服务器就要处理一次,所以点击是WEB应用能够处理的交易的最小单位。如果把每次点击定义为一个交易,点击率和TPS就是一个概念。容易看出,点击率越大,对服务器的压力越大。点击率只是一个性能参考指标,重要的是分析点击时产生的影响。需要注意的是,这里的点击并非指鼠标的一次单击操作,因为在一次单击操作中,客户端可能向服务器发出多个HTTP请求。

8.**资源利用率**

指的是对不同的系统资源的使用程度,例如服务器的CPU利用率,磁盘利用率等。资源利用率是分析系统性能指标进而改善性能的主要依据,因此是WEB性能测试工作的重点。

资源利用率主要针对WEB服务器、操作系统、数据库服务器、网络等,是测试和分析瓶颈的主要参考。在WEB性能测试中,更根据需要采集相应的参数进行分析。

1. **通用指标**（指Web应用服务器、[数据库](javascript:;" \t "http://www.51testing.com/html/44/_self)服务器必需测试项)

|  |  |
| --- | --- |
| **指标** | **说明** |
| ProcessorTime | 服务器CPU占用率，—般平均达到70%时，服务就接近饱和 |
| Memory Available Mbyte | 可用内存数，如果测试时发现内存有变化情况也要注意，如果是内存泄漏 |
| Physicsdisk Time | 物理磁盘读写时间情况 |

1. **Web服务器指标**

|  |  |
| --- | --- |
| **指标** | **说明** |
| Requests Per Second (Avg Rps) | 平均每秒钟响应次数=总请求次数/秒数 |
| Avg time to last byte per terstion  (mstes) | 平均每秒业务脚本的迭代次数,有人会把上面那个混淆 |
| Successful Rounds | 成功的请求 |
| Failed Requests | 失败的请求 |
| Successful Hits | 成功的点击次数 |
| Failed Hits | 失败的点击次数 |
| Hits Per Second | 每秒点击次数 |
| Successful Hits Per Second | 每秒成功的点击次数 |
| Failed Hits Per Second | 每秒失败的点击次数 |
| Attempted Connections | 尝试链接数 |

1. **数据库服务器性能指标**

|  |  |
| --- | --- |
| **指标** | **说明** |
| User 0 Connections | 用户连接数，也就是数据库的连接数量 |
| Number of deadlocks | 数据库死锁 |
| Butter Cache hit | 数据库Cache的命中情况 |

1. **系统的瓶颈定义**

|  |  |  |
| --- | --- | --- |
| **性能项** | **命令** | **指标** |
| CPU限制 | vmstat | 当%user+%sys超过80%时 |
| 磁盘I/O限制 | vmstat | 当%iowait超过40%(AIX4.3.3或更高版本)时 |
| 应用磁盘限制 | lostat | 当%tm\_act超过70%时 |
| 虚存空间少 | Lsps, -a | 当分页空间的活动率超过70%时 |
| 换页限制 | lostat,stat | 虚存逻辑卷%tm\_act超过I/O(iostat)的30%，激活的虚存率超过CPU数量(vmstat)的10倍时 |
| 系统失效 | Vmstat,sar | 页交换增大、CPU等待并运行队列 |

1. **稳定系统的资源状态**

|  |  |  |
| --- | --- | --- |
| **性能项** | **资源** | **评价** |
| CPU占用率 | 70% | 好 |
|  | 85% | 坏 |
|  | 90%+ | 很差 |
| 磁盘I/O | <30% | 好 |
|  | <40% | 坏 |
|  | <50%+ | 很差 |
| 网络 | <30%带宽 | 好 |
| 运行队列 | <2\*CPU数量 | 好 |
| 内存 | 没有页交换 | 好 |
|  | 每个CPU每秒10个页交换 | 坏 |
|  | 更多的页交换 | 很差 |

通俗理解：日访问量、常用页面最大并发数、同时在线人数、访问响应时间

## **5.2 网页响应时间测试标准**

1.响应时间

指的是客户发出请求到得到响应的整个过程的时间。在某些工具中，请求响应时间通常会被称为“TTLB”(Time to laster byte) ,意思是从发起一个请求开始，到客户端收到最后一个字节的响应所耗费的时间。

响应时间的计算模型:

响应时间=网络传输时间（请求）+服务器处理时间（一层或是多层)+网络传输时间〈响应)+页面前段解析时间

简化的浏览器响应时间的计算模型;

浏览器响应时间=服务器响应时间＋页面装载时间＋页面渲染时间

页面渲染时间主要包含两个部分:

页面渲染时间=脚本执行时间＋浏览器引擎渲染时间

2.指导思路

“2-5-10原则”，简单说．就是当用户能够在2秒以内得到响应时，会感觉系统的响应很快;

当用户在2秒以内得到响应时,会感觉系统的响应很快;

当用户在2-5秒之间得到响应时，会感觉系统的响应速度还可以;

当用户在5-10秒以内得到响应时，会感觉系统的响应速度很慢，但是还可以接受;

而当用户在超过8秒后仍然无法得到响应时，会感觉系统糟透了，或者认为系统已经失去响应，而选择离开这个Web站点，或者发起第二次请求。

2秒-首页，5秒-普通页面,10秒-复杂查询页面。

在进行性能测试时,“合理的响应时间”取决于用户的需求，而不能依据测试人员自己设想来决定。

终端用户响应时间:

就是指从这个用户触发一个页面请求到这个页面被完全展示的时间，有时也被称为浏览器响应时间。终端用户响应时间是终端用户对一个应用性能的直观感受。它由三部分组成:

页面请求与下载时间（简称页面下载时间)。

服务器响应时间。

浏览器处理及渲染时间。

用一个公式来表示，那就是：

终端用户响应时间=页面下载时间+服务器响应时间＋浏览器处理及渲染时间

网络延迟、HTTP请求数、并发度和消耗的总延迟时间的关系是：

总延迟时间=网络延迟×HTTP请求数/并发度

页面下载时间的简化模型就是:

页面下载时间=在带宽上消耗的时间＋在网络延迟上消耗的时间=页面尺寸/网络带宽＋(网络延迟×HTTP请求数)│并发度

在这个简化模型中,有些因子没有被考虑︰

DNS查询时间;

HTTP请求建立时间;

HTTP连接的保持状态;

浏览器、服务器在处理传输时消耗的时间；带宽对并发度的影响。

基于这个模型，我们只要测量出页面的尺寸、HTTP 请求数和并发度就可以推测在不同网络条件下的页面下载时间。

## **5.3 测试案例**

最近某公司一个项目，是个门户网站，需要做性能测试，根据项目特点定出了主要测试项和测试方案：

第一种是测试几个常用页面能接受的最大并发数(用户名参数化，设置集合点策略)

第二种是测试服务器长时间压力下，用户能否正常操作(用户名参数化，迭代运行脚本)

第三种则需要测试服务器能否接受10万用户同时在线操作，如果是用IIS做应用服务器的话，单台可承受的最大并发数不可能达到10万级，那就必须要使用集群，通过多台机器做负载均衡来实现；如果是用websphere之类的应用服务器的话，单台可承受的最大并发数可以达到10万级，但为性能考虑还是必须要使用集群，通过多台机器做负载均衡来实现；通常有1个简单的计算方式，1个连接产生1个session，每个session在服务器上有个内存空间大小的设置，在NT上是3M，那么10万并发就需要300G内存，当然实际使用中考虑其他程序也占用内存，所以准备的内存数量要求比这个还要多一些.还有10万个用户同时在线，跟10万个并发数是完全不同的2个概念。这个楼上已经说了.但如何做这个转换将10万个同时在线用户转换成多少个并发数呢？这就必须要有大量的历史日志信息来支撑了。系统日志需要有同时在线用户数量的日志信息，还需要有用户操作次数的日志信息，这2个数据的比例就是你同时在线用户转换到并发数的比例。另外根据经验统计，对于1个JAVA开发的WEB系统（别的我没统计过，给不出数据），一般1台双CPU、2G内存的服务器上可支持的最大并发数不超过500个（这个状态下大部分操作都是超时报错而且服务器很容易宕机，其实没什么实际意义），可正常使用（单步非[大数据](javascript:;" \t "http://www.51testing.com/html/44/_self)量操作等待时间不超过20秒）的最大并发数不超过300个.假设你的10万同时在线用户转换的并发数是9000个，那么你最少需要这样的机器18台，建议不少于30台.当然，你要是买个大型服务器，里面装有200个CPU、256G的内存，千兆光纤带宽，就算是10万个并发用户，那速度，也绝对是嗖嗖的。

另外方案一下，光设置全部进入运行状态就需要接近6个小时。具体的可以拿1个系统来压一下看看，可能会出现以下情况：

（1）服务器宕机；

（2）客户端宕机；

（3）从某个时间开始服务器拒绝请求，客户端上显示的全是错误；

（4）勉强测试完成，但网络堵塞或测试结果显示时间非常长.假设客户端和服务器之间百兆带宽，百兆/10000=10K，那每个用户只能得到10K，这个速度接近1个64K的MODEM上网的速度；另外以上分析全都没考虑系统的后台，比如数据库、中间件等.

（1）服务器方面：上面说的那样的PC SERVER需要50台；

（2）网络方面：按每个用户50K，那至少5根百兆带宽独享，估计仅仅网络延迟就大概是秒一级的；

（3）如果有数据库，至少是[Oracle](javascript:;" \t "http://www.51testing.com/html/44/_self)，最好是SYSBASE，SQLSERVER是肯定顶不住的.数据库服务器至少需要10台4CPU、16G内存的机器；

（4）如果有CORBA，那至少再准备10台4CPU、16G内存的机器；再加上负载均衡、防火墙、路由器和各种软件等，总之没个1000万的资金投入，肯定搞不定。

这样的门户系统，由于有用户权限，所以并不像jackie所说大多是静态页面。但只要是多服务器的集群，那么我们就可以通过1台机器的测试结果来计算多台机器集群后的负载能力的，最多额外考虑一下负载均衡和路由上的压力，比如带宽、速度、延迟等.但如果都是在1台机器上变化，那我们只能做一些指标上的计算，可以从这些指标上简单判断一下是否不可行，比如10万并发用户却只有1根百兆带宽，那我们可以计算出每个用户只有1K带宽，这显然是不可行的。但实际的结果还是需要测试了才知道，毕竟系统压力和用户数量不是线性变化的。

这一类系统的普遍的成熟的使用，以及很多软件在方案设计后就能够大致估算出系统的性能特点，都导致了系统在软件性能方面调优的比例并不大（当然不完全排除后期针对某些代码和配置进行优化后性能的进一步提高），更多的都是从硬件方面来考虑，比如增加内存、硬盘做RAID、增加带宽、甚至增加机器等。

网络[技术](javascript:;" \t "http://www.51testing.com/html/44/_self)中的10M 带宽指的是以位计算， 就是 10M bit /秒 ，而下载时的速度看到的是以字节（Byte）计算的，所以10M带宽换算成字节理论上最快下载速度为： 1.25 M Byte/秒!

# 6 Web功能测试实操

一、输入框

1、字符型输入框：

a.字符型输入框：英文全角、英文半角、数字、空或者空格、特殊字符“~！@#￥%……&\*？[]{}”特别要注意单引号和&符号.禁止直接输入特殊字符时，使用“粘贴、拷贝”功能尝试输入;

b.长度检查：最小长度、最大长度、最小长度-1、最大长度+1、输入超工字符比如把整个文章拷贝过去;

c.空格检查：输入的字符间有空格、字符前有空格、字符后有空格、字符前后有空格;

d.多行文本框输入：允许回车换行、保存后再显示能够保存输入的格式、仅输入回车换行，检查能否正确保存（若能，检查保存结果，若不能，查看是否有正常提示）；

e.安全性检查：输入特殊字符串（null,NULL, ,javascript,)、document.write(“abc”)、hello）

2、数值型输入框：

a.边界值：最大值、最小值、最大值+1、最小值-1

b.位数：最小位数、最大位数、最小位数-1最大位数+1、输入超长值、输入整数

c.异常值、特殊字符：输入空白（NULL）、空格或"~!@#$%^&\*()\_+{}|[]:"<>?;’,./?;:’-=等可能导致系统错误的字符、禁止直接输入特殊字符时，尝试使用粘贴拷贝查看是否能正常提交、word中的特殊功能，通过剪贴板拷贝到输入框，分页符，分节符类似公式的上下标等、数值的特殊符号如∑，㏒，㏑，∏，+，-等、

输入负整数、负小数、分数、输入字母或汉字、小数（小数前0点舍去的情况，多个小数点的情况）、首位为0的数字如01、02、科学计数法是否支持1.0E2、全角数字与半角数字、数字与字母混合、16进制，8进制数值、货币型输入（允许小数点后面几位）、

d.安全性检查：不能直接输入就copy

3、日期型输入框：

a.合法性检查：(输入0日、1日、32日)、月输入[1、3、5、7、8、10、12]、日输入[31]、月输入[4、6、9、11]、日输入[30][31]、输入非闰年，月输入[2]，日期输入[28、29]、输入闰年，月输入[2]、日期输入[29、30]、月输入[0、1、12、13]

b.异常值、特殊字符：输入空白或NULL、输入~！@#￥%……&\*（）{}[]等可能导致系统错误的字符

c.安全性检查：不能直接输入，就copy，是否数据检验出错？

4、信息重复:在一些需要命名,且名字应该唯一的信息输入重复的名字或ID,看系统有没有处理,会否报错,重名包括是否区分大小写,以及在输入内容的前后输入空格,系统是否作出正确处理.

二、搜索功能

若查询条件为输入框，则参考输入框对应类型的测试方法

1、功能实现：

a.如果支持模糊查询，搜索名称中任意一个字符是否能搜索到

b.比较长的名称是否能查到

c.输入系统中不存在的与之匹配的条件

d.用户进行查询操作时，一般情况是不进行查询条件的清空，除非需求特殊说明

2、组合测试：

a .不同查询条件之间来回选择，是否出现页面错误（单选框和多选框最容易出错）

b.测试多个查询条件时，要注意查询条件的组合测试，可能不同组合的测试会报错

三、添加、修改功能

a.特殊键：（1）是否支持Tab键 （2）是否支持回车键

b.提示信息：（1）不符合要求的地方是否有错误提示

c.唯一性：（1）字段唯一的，是否可以重复添加，添加后是否能修改为已存在的字段（字段包括区分大小写以及在输入的内容前后输入空格，保存后，数据是否真的插入到数据库中，注意保存后数据的正确性）

d.数据正确性：

（1）对编辑页的每个编辑项进行修改，点击保存，是否可以保存成功，检查想关联的数据是否得到更新

（2）进行必填项检查（即是否给出提示以及提示后是否依然把数据存到数据库中；是否提示后出现页码错乱等）

（3）是否能够连续添加（针对特殊情况）

（4）在编辑的时候，注意编辑项的长度限制，有时在添加的时候有，在编辑的时候却没有（注意要添加和修改规则是否一致）

（5）对于有图片上传功能的编辑框，若不上传图片，查看编辑页面时是否显示有默认的图片，若上传图片，查看是否显示为上传图片

（6）修改后增加数据后，特别要注意查询页面的数据是否及时更新，特别是在首页时要注意数据的更新

（7）提交数据时，连续多次点击，查看系统会不会连续增加几条相同的数据或报错

（8）若结果列表中没有记录或者没选择某条记录，点击修改按钮，系统会抛异常

四、删除功能

a.特殊键：

（1）是否支持Tab键

（2）是否支持回车键

b.提示信息：

（1）不选择任何信息，直接点击删除按钮，是否有提示

（2）删除某条信息时，应该有确认提示

c.数据实现：

（1）是否能连续删除多个产品

（2）当只有一条数据时，是否可以删除成功

（3）删除一条数据后，是否可以添加相同的数据

（4）如系统支持批量删除，注意删除的信息是否正确

（5）如有全选，注意是否把所有的数据删除

（6）删除数据时，要注意相应查询页面的数据是否及时更新

（7）如删除的数据与其他业务数据关联，要注意其关联性（如删除部门信息时，部门下游员工，则应该给出提示）

（8）如果结果列表中没有记录或没有选择任何一条记录，点击删除按钮系统会报错.

五、注册、登陆模块

a.注册功能：

（1）注册时，设置密码为特殊版本号，检查登录时是否会报错

（2）注册成功后，页面应该以登陆状态跳转到首页或指定页面

（3）在注册信息中删除已输入的信息，检查是否可以注册成功.

b.登陆 功能：

（1）输入正确的用户名和正确的密码

（2）输入正确的用户名和错误的密码

（3）输入错误的用户名和正确的密码

（4）输入错误的用户名和错误的密码

（5）不输入用户名和密码（均为空格）

（6）只输入用户名，密码为空

（7）用户名为空，只输入密码

（8）输入正确的用户名和密码，但是不区分大小写

（9）用户名和密码包括特殊字符

（10）用户名和密码输入超长值

（11）已删除的用户名和密码

（12）登录时，当页面刷新或重新输入数据时，验证码是否更新

六、上传功能

功能实现：

（1）文件类型正确、大小合适

（2）文件类型正确，大小不合适

（3）文件类型错误，大小合适

（4）文件类型和大小都合适，上传一个正在使用中的图片

（5）文件类型大小都合适，手动输入存在的图片地址来上传

（6）文件类型和大小都合适，输入不存在的图片地址来上传

（7）文件类型和大小都合适，输入图片名称来上传

（8）不选择文件直接点击上传，查看是否给出提示

（9）连续多次选择不同的文件，查看是否上传最后一次选择的文件

七、查询结果列表

功能实现：

（1）列表、列宽是否合理

（2）列表数据太宽有没有提供横向滚动

（3）列表的列名有没有与内容对应

（4）列表的每列的列名是否描述的清晰

（5）列表是否把不必要的列都显示出来

（6）点击某列进行排序，是否会报错（点击查看每一页的排序是否正确）

（7）双击或单击某列信息，是否会报错

八、返回键检查

a.一条已经成功提交的记录，返回后再提交，是否做了处理

b.检查多次使用返回键的情况，在有返回键的地方，返回到原来的页面多次，查看是否会出错

九、回车键检查

在输入结果后，直接按回车键，看系统如何处理，是否会报错

十、刷新键检查

在Web系统中，使用刷新键，看系统如何处理，是否会报错

十一、直接URL链接检查

在Web系统中，在地址栏直接输入各个功能页面的URL地址，看系统如何处理，是否能够直接链接查看（匿名查看），是否有权限控制，是否直接执行，并返回相应结果页；

十二、界面和易用性测试

1、风格、样式、颜色是否协调

2、界面布局是否整齐、协调（保证全部显示出来的，尽量不要使用滚动条

3、界面操作、标题描述是否恰当（描述有歧义、注意是否有错别字）

4、操作是否符合人们的常规习惯（有没有把相似的功能的控件放在一起，方便操作）

5、提示界面是否符合规范（不应该显示英文的cancel、ok，应该显示中文的确定等）

6、界面中各个控件是否对齐

7、日期控件是否可编辑

8、日期控件的长度是否合理，以修改时可以把时间全部显示出来为准

9、查询结果列表列宽是否合理、标签描述是否合理

10、查询结果列表太宽没有横向滚动提示

11、对于信息比较长的文本，文本框有没有提供自动竖直滚动条

12、数据录入控件是否方便

13、有没有支持Tab键，键的顺序要有条理，不乱跳

14、有没有提供相关的热键

15、控件的提示语描述是否正确

16、模块调用是否统一，相同的模块是否调用同一个界面

17、用滚动条移动页面时，页面的控件是否显示正常

18、日期的正确格式应该是XXXX-XX-XX或XXXX-XX-XX XX:XX:XX

19、页面是否有多余按钮或标签

20、窗口标题或图标是否与菜单栏的统一

21、窗口的最大化、最小化是否能正确切换

22、对于正常的功能，用户可以不必阅读用户手册就能使用

23、执行风险操作时，有确认、删除等提示吗

24、操作顺序是否合理

25、正确性检查：检查页面上的form, button, table, header, footer,提示信息，还有其他文字拼写，句子的语法等是否正确

26、系统应该在用户执行错误的操作之前提出警告，提示信息

27、页面分辨率检查，在各种分辨率浏览系统检查系统界面友好性

28、合理性检查：做delete, update, add, cancel, back等操作后，查看信息回到的页面是否合理

十三、兼容性测试

兼容性测试不只是指界面在不同操作系统或浏览器下的兼容，有些功能方面的测试，也要考虑到兼容性,包括操作系统兼容和应用软件兼容，可能还包括硬件兼容。比如涉及到ajax、jquery、javascript等技术的，都要考虑到不同浏览器下的兼容性问题。

十四、链接测试

主要是保证链接的可用性和正确性，它也是网站测试中比较重要的一个方面。

可以使用特定的工具如XENU来进行链接测试。

a.导航测试导航描述了用户在一个页面内操作的方式，在不同的用户接口控制之间，例如按钮、对话框、列表和窗口等；或在不同的连接页面之间.通过考虑下列问题，可以决定一个Web应用系统是否易于导航：导航是否直观？Web系统的主要部分是否可通过主页存取？Web系统是否需要站点地图、搜索引擎或其他的导航帮助？在一个页面上放太多的信息往往起到与预期相反的效果.Web应用系统的用户趋向于目的驱动，很快地扫描一个Web应用系统，看是否有满足自己需要的信息，如果没有，就会很快地离开.很少有用户愿意花时间去熟悉Web应用系统的结构，因此，Web应用系统导航帮助要尽可能地准确.导航的另一个重要方面是Web应用系统的页面结构、导航、菜单、连接的风格是否一致.确保用户凭直觉就知道Web应用系统里面是否还有内容，内容在什么地方.Web应用系统的层次一旦决定，就要着手测试用户导航功能，让最终用户参与这种测试，效果将更加明显

b.图形测试在Web应用系统中，适当的图片和动画既能起到广告宣传的作用，又能起到美化页面的功能。一个Web应用系统的图形可以包括图片、动画、边框、颜色、字体、背景、按钮等.图形测试的内容有：

1.要确保图形有明确的用途，图片或动画不要胡乱地堆在一起，以免浪费传输时间。Web应用系统的图片尺寸要尽量地小，并且要能清楚地说明某件事情，一般都链接到某个具体的页面

2.验证所有页面字体的风格是否一致

3.背景颜色应该与字体颜色和前景颜色相搭配

4.图片的大小和质量也是一个很重要的因素，一般采用JPG或GIF压缩，最好能使图片的大小减小到30k以下

5.最后，需要验证的是文字回绕是否正确。如果说明文字指向右边的图片，应该确保该图片出现在右边.不要因为使用图片而使窗口和段落排列古怪或者出现孤行。通常来说，使用少许或尽量不使用背景是个不错的选择.如果您想用背景，那么最好使用单色的，和导航条一起放在页面的左边.另外，图案和图片可能会转移用户的注意力。

十五、业务流程测试（主要功能测试）

业务流程，一般会涉及到多个模块的数据，所以在对业务流程测试时，首先要保证单个模块功能的正确性，其次就要对各个模块间传递的数据进行测试，这往往是容易出现问题的地方，测试时一定要设计不同的数据进行测试。

十六、安全性测试

（1）SQL注入（比如登陆页面）

（2）XSS跨网站脚本攻击：程序或数据库没有对一些特殊字符进行过滤或处理，导致用户所输入的一些破坏性的脚本语句能够直接写进数据库中，浏览器会直接执行这些脚本语句，破坏网站的正常显示，或网站用户的信息被盗,构造脚本语句时，要保证脚本的完整性

　document.write(“abc”)

（3）URL地址后面随便输入一些符号，并尽量是动态参数靠后

（4）验证码更新问题

（5）现在的Web应用系统基本采用先注册，后登陆的方式.因此，必须测试有效和无效的用户名和密码，要注意到是否大小写敏感，可以试多少次的限制，是否可以不登陆而直接浏览某个页面等

（6）Web应用系统是否有超时的限制，也就是说，用户登陆后在一定时间内（例如15分钟）没有点击任何页面，是否需要重新登陆才能正常使用

（7）为了保证Web应用系统的安全性，日志文件是至关重要的.需要测试相关信息是否写进了日志文件、是否可追踪

（8）当使用了安全套接字时，还要测试加密是否正确，检查信息的完整性

（9）服务器端的脚本常常构成安全漏洞，这些漏洞又常常被黑客利用。所以，还要测试没有经过授权，就不能在服务器端放置和编辑脚本的问题。

十七、性能测试

a.连接速度测试

用户连接到Web应用系统的速度根据上网方式的变化而变化，他们或许是电话拨号，或是宽带上网.当下载一个程序时，用户可以等较长的时间，但如果仅仅访问一个页面就不会这样。如果Web系统响应时间太长（例如超过5秒钟），用户就会因没有耐心等待而离开。

另外，有些页面有超时的限制，如果响应速度太慢，用户可能还没来得及浏览内容，就需要重新登陆了.而且，连接速度太慢，还可能引起数据丢失，使用户得不到真实的页面。

b.负载测试

负载测试是为了测量Web系统在某一负载级别上的性能，以保证Web系统在需求范围内能正常工作.负载级别可以是某个时刻同时访问Web系统的用户数量，也可以是在线数据处理的数量.例如：Web应用系统能允许多少个用户同时在线？如果超过了这个数量，会出现什么现象？Web应用系统能否处理大量用户对同一个页面的请求？

c.压力测试负载测试应该安排在Web系统发布以后，在实际的网络环境中进行测试。因为一个企业内部员工，特别是项目组人员总是有限的，而一个Web系统能同时处理的请求数量将远远超出这个限度，所以，只有放在Internet上，接受负载测试，其结果才是正确可信的。进行压力测试是指实际破坏一个Web应用系统，测试系统的反映.压力测试是测试系统的限制和故障恢复能力，也就是测试Web应用系统会不会崩溃，在什么情况下会崩溃.黑客常常提供错误的数据负载，直到Web应用系统崩溃，接着当系统重新启动时获得存取权.压力测试的区域包括表单、登陆和其他信息传输页面等。

十八.web测试需要了解的知识

a.HTTP/HTTPS协议

你应该去了解什么是http协议？

什么是GET, POST, session, cookie等

Get与Post的区别是什么?

session与cookie的区别是什么?

什么是无状态？

浏览器机制

理解浏览器在处理javascript及渲染CSS的机制

了解IE与其他浏览器的差异

同一浏览器，各版本的区别

浏览器在加载javascript，CSS有时在前面有时在后面，为什么？

加载顺序会对视觉和使用上有什么影响呢？

各种浏览器使用的内核分别是什么？

b.web架构

也许你会说这是架构师的事儿，没错，基本是他们的活儿，但是理解了web架构能让我们测试的更深入。你要知道：

软件出错时怎么个报警法？是否有详尽的log记录？

服务器缓存机制如何？

数据库如何主从同步，如何备份的？

集群如何处理session的？

c.web性能

你应该知道的web性能知识：

web前端的性能极大影响了用户，如何观察这些数据？CSS和图片的合并压缩的意义

了解浏览器cache及服务端cache

对于图片请求过多的网站，为何要把图片放置在不同的域名下，最好使用CDN？

确认你的网站有一个 favicon.ico 文件放在网站的根下，如 /favicon.ico。每当有用户收藏网站/网页时，浏览器会自动请求这个文件，就算这个图标文件没有在你的网页中明显说明，浏览器也会请求。如果你没有这个文件，就会出大量的404错误，这会消耗你的服务器带宽，服务器返回404页面会比这个ico文件可能还大。

知道单个页面的http请求数越少越好

顺序加载和异步加载的优劣，何时需要使用AJAX？懒加载的意义，用于何处？

如何使用性能测试工具Jmeter/LR等开展性能测试？

d.用户体验

网站的功能只是说实现了什么，而用户体验则诠释了做的有多好，用户使用起来是否有难度，是否会爱上这个网站（当然12306除外，咯咯）

访问网站的用户操作行为是怎么样的？页面的访问频率占比如何？因为测试的精力和侧重点也要根据这个数据而定

网站部署时是否会影响到用户使用，如何避免？

不要直接显示不友好的错误提示，是否有友好的提示信息？

web应用不能泄漏用户的隐私信息

页面是在当前页打开还是另开一个tab？

页面元素的布局如何影响到用户体验的？

e.使用工具

1.HttpWatch，基于IE的网络数据分析工具，包括网页摘要，Cookies管理，缓存管理，消息头发送/接受，字符查询，POST 数据和目录管理功能等

2.FireBug，用途同上，基于firefox的

3.Yslow，前端网站性能工具，显示测试结果的分析,分为等级、组件、统计信息

4.Fiddler，强大的web前端调试工具，它能记录/拦截所有客户端和服务器的http和https请求，允许你监视，设置断点，甚至修改输入输出数据，也可用于安全测试

5.Chrome扩展程序：浏览器兼容性检测工具，分析网站的兼容性情况

[Chrome浏览器兼容性检测工具（chrome插件）](https://www.cnblogs.com/HughTan/archive/2013/05/07/3064711.html)

近日 Google Chrome 组发起了浏览器兼容性检测工具(Chrome扩展)开源项目，能够自动检测网页中的兼容性问题。此工具已经实现了检测2010年9月Google在 Devfest北京会议上推出的w3help.org站点内所描述的部分兼容性问题。是前端兼容性工作必备辅助开发工具。

目前的版本一共检查了 40 个跨浏览器开发过程中可能会遇到的问题，其中包括文本渲染、表格、布局以及脚本等几个方面。所有已知的兼容性问题列表见<http://w3help.org/zh-cn/causes/>，目前只实现了一部分，其它的还正在开发。

工具安装:

<https://chrome.google.com/webstore/detail/fcillahbnhlpombgccogflhmgocfifma?hl=zh-CN>

<http://www.w3help.org/tools/compatibility-detector.crx>

开源代码站点:

<https://code.google.com/p/compatibility-detector/>

讨论邮件组:

compatibility-detector-discuss@googlegroups.com

<https://groups.google.com/group/compatibility-detector-discuss>

说明PPT:

<http://www.slideshare.net/dskyline/compatibility-detector-tool-of-chrome-extensions>

中文说明PDF：

<http://www.w3help.org/tools/compatibility-ppt.pdf>

PPT内对应样例：<http://www.w3help.org/tools/comp-sample.zip>

# 7 移动端测试

**一、移动端性能监测的主要途径**

移动端性能监测的主要途径有三种：

一是开发工具自带的监测工具，例如xcode自带的instrument，Android studio自带的Android monitor；

二是使用第三方SDK;

三是自行开发检测代码。

三种途径各有利弊。开发工具自带的监测工具，包含了很多强大的监测功能，且持续迭代更新，使用方便，为开发阶段的性能测试提供强有力的支持。但是只能在开发工具内部使用，不能独立使用在其他产品周期内。专门用于性能监测和用户行为、属性分析的第三方SDK，比如Bugly，OneAPM，听云，Firebase，把它们接入项目可以进行性能监测，这些第三方的工具工作原理都是类似的：引入SDK和相关库，在程序入口处启动监测，在关键函数之前和之后自动埋点记录上报。然而使用第三方SDK的缺点也是非常明显的，首先是可定制性差，我们需要的一些指标无法收集，SDK收集的我们又不完全需要，很有可能为了简单的几个值，让安装包增大许多。除性能指标外，SDK是否统计了其他数据，我们并不完全知道，涉及到安全性问题。自行在项目中植入监测代码的优势就是安全可靠，并且可定制化指标，有针对性。缺点就是需要开发成本，而且在代码中监测App的性能本身可能也会带来额外的性能损耗。

**二、移动端核心性能指标**

App的性能问题包括崩溃、网络请求错误或超时、响应速度慢、列表滚动卡顿、流量大、耗电等。而导致App性能低下的原因有很多，除去设备硬件和软件的外部因素，其中大部分是开发者错误地使用线程、锁、系统函数、编程范式、数据结构等导致的。即便是最有经验的程序员，也很难在开发时就能避免所有导致性能低下的“坑”，因此解决性能问题的关键是在于能不能尽早地发现和定位这些“坑”。衡量一个App的性能有很多方面的指标，不同App的特征会有不同的侧重指标。其中，核心的性能检测指标(普遍会监测的指标）有

内存占用量（MEM）

CPU使用率（CPU）

页面刷新帧率（FPS）

网络状况（Net）

流量消耗（Flow）

卡顿（Flag）

崩溃（Crash）

启动时间

**三、移动端性能测试的需求**

不同迭代周期对移动端性能测试的需求

一个App的迭代生命周期，可大体分为开发期、测试期、上线期。

开发时期，需要能够检查性能异常点并通知给开发者；

测试时期，需要使用测试工具产生性能测试报告；

上线时期，通过监控平台上报性能数据，实现线上问题定位和追查。

不同的迭代周期对于测试工具具有不同需求：

开发和测试阶段：直观看到性能测试的结果，一个可以全面检测App性能指标的本地监测工具就可以满足需求。

上线阶段：我们对于用户使用的应用无法做到像针对测试阶段的应用的可控程度，所以需要通过引入性能监测SDK的方式，在运行阶段将运行在用户设备中的应用性能指标上传至监控平台，便于监测应用线上的性能表现并定位问题。

**四、主流移动端性能测试工具简述**

1.Xcode-Instrument

Instrument是Apple官方提供的一个强大的内置在xcode中的性能调试工具集.其中，Activity Monitor（活动监视器）：监控进程级别的CPU,内存,磁盘，网络使用情况，可以得到应用程序在手机运行时总占用的内存大小；Core Animation（图形性能）：这个模块显示程序显卡性能、CPU使用情况以及页面刷新帧率；Network：用链接工具分析程序如何使用TCP/IP和UDP/IP链接；Energy Log：耗电量监控.Instrument主要用于在调试过程中随时发现问题，及时优化，但是这个工具只能供有应用源码的程序员使用，无法测量用户真实使用场景下的性能。

2.Android monitor

Android Studio内置了四种性能监测工具Memory Monitor、Network Monitor、CPU Monitor、GPU Monitor，可以使用这些工具监测App的状态.Memory Monitor工具主要是用来监测App的内存分配情况，判断是否存在内存泄漏；Network Monitor是用于显示App网络请求的状态；GPU Monitor工具可以将进行UI渲染工作所花的时间表现出来；CPU Monitor可以对代码中的方法进行检测。　 3.Matrix

Matrix是一款微信研发并日常使用的应用性能接入框架，支持iOS, macOS和Android.Matrix 通过接入各种性能监控方案，对性能监控项的异常数据进行采集和分析，输出相应的问题分析、定位与优化建议，从而帮助开发者开发出更高质量的应用.数据监测结果存储到后台，没有直观的展示界面.Matrix-iOS/　　macOS主要监控崩溃、卡顿和爆内存.Matrix-android监控范围包括：应用安装包大小，帧率变化，启动耗时，卡顿，慢方法，SQLite 操作优化，文件读写，内存泄漏等。

4.GT

GT（随身调）是直接运行在手机上的“集成调测环境”。利用GT，仅凭一部手机，无需连接电脑，即可对App进行快速的性能测试(CPU、内存、流量、电量、帧率/流畅度等)、开发日志的查看、Crash日志查看、网络数据包的抓取、App内部参数的调试、真机代码耗时统计等。GT支持iOS和Android两个手机平台，其中：iOS版是一个Framework包，必须嵌入App工程，编译出带GT的App才能使用；iPhone和iPad应用都能支持。Android版由一个可直接安装的GT控制台App和GT SDK组成.GT控制台可以独立安装使用；SDK需嵌入被调测的应用、并利用GT控制台进行信息展示和参数修改。

5.Emmagee

Emmagee是一款实用、方便的性能测试工具，适用于指定的Android App，它可以监控CPU、内存、网络流量、电池电流和状态（某些设备不受支持）.此外，它还支持自定义收集数据的时间间隔，在浮动窗口中呈现实时进程状态等.　　安装之后，打开Emmagee会显示出目前系统已经安装的应用（可在设置中配置间隔）.选择一个应用之后，点击“开始测试”按钮，Emmagee会自动启动被测应用，并在最上方显示流量、CPU、内存等数据，在该状态下可任意使用被测应用，Emmagee会自动进行相关数据的记录，终止测试之后会生成相应的报告，并保存在Android目录中。

6.Soloπ

Soloπ是支付宝在移动端上实现的一套无线化、非侵入、免Root的 Android 专项测试方案.直接操控手机，即可实现自动化的功能、性能、兼容性、以及稳定性测试等工作.支持 CPU、内存、FPS、流量等常规指标的实时获取，同时支持将性能数据记录下来，存储到本地并通过报表形式展示，支持性能加压。　 7.Testin

Testin性能测试，通过自动化的测试工具，在上线前模拟正常、峰值以及异常等多种场景，从而对App的各项性能指标进行测试.包括：负载测试、压力测试、配置测试、并发测试、容量测试、可靠性测试.　　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情况下，系统所能承受的最大负载量。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，并以此来获得系统能够提供的最大服务级别。配置测试：通过测试被测试软件的软硬件配置，获得不同配置的性能指标，用于选择最佳的设备及参数配置。并发测试：测试多个用户同时访问同一个应用、同一个模块或者数据记录时是否存在死锁或者其他性能问题。容量测试：在确定的软、硬件条件下，构造一定数量的虚拟用户访问，从而获取不同数量级别的服务器性能指标，以确定数据库能够处理的最大会话能力、最大容量等。可靠性测试：采用系统能够支持的最大并发用户数，持续执行一段时间（7\*24小时），通过综合分析执行指标、资源监控指标，来确定系统处理最大工作量的性能情况。

8.Bugly

Bugly是腾讯推出的一款免费的崩溃收集工具.面向移动开发者提供专业的 Crash 监控、崩溃分析等质量跟踪服务.移动开发者 ( Android / iOS ) 可以通过监控，快速发现用户在使用过程中出现的 Crash (崩溃)、Android ANR( application not response) 和 iOS 卡顿，并根据上报的信息快速定位和解决问题.用户只要登陆 Bugly 网站，就可以清晰的看到被监测产品有多少 Crash，影响了多少用户的使用，并可以根据 Bugly 提供的 Crash 日志进行问题修复。

9.OneAPM

Mi是OneAPM针对移动设备上App推出的移动应用性能监控工具。Mi展示移动应用的交互性能，崩溃率，HTTP 错误率，网络错误率等核心性能指标、用户访问轨迹、用户动作等行为数据，以及活跃用户等运营数据.从用户动作、网络请求、崩溃三个核心指标对用户体验进行评估，并结合应用版本、操作系统、设备、地域等多个维度总览用户分布情况.完整记录每个用户的访问轨迹，通过查看单次用户访问，回溯过程中出现的缓慢动作，错误请求和崩溃.详尽展示某类崩溃影响的设备、操作系统、应用版本、发生次数、影响用户等信息，还原崩溃发生轨迹，定位崩溃代码行。

10.听云App

听云App是听云平台提供的移动端性能监测工具.监控真实用户使用过程中的崩溃、错误、卡顿、网络性能差等问题。可以跟踪App应用移动设备端用户进行屏幕操作时的交互性能；统计多种错误类型,直观定位到具体报错URL、以及崩溃时的堆栈信息；可以深入追踪HTTP错误、网络错误和崩溃的原因，提高App质量。

**五、主流移动端性能测试工具横向对比**

针对不同的产品迭代周期，选取了目前业内比较成熟的工具，对比各自的优缺点。





各工具性能指标覆盖情况对比如下表：



**六、结论**

不同的移动端性能测试工具各有优劣势，在工具选择上，需要综合考虑主要监测的性能指标和工具的特长。对于研发测试阶段适用的工具，GT、Emmagee、soloπ在andriod端可直接安装apk进行测试，方便快捷。对于线上用户使用情况的监控，有些工具指标覆盖比较全面，但是对某些指标的监测不够深入，可考虑叠加另外一个或多个工具进行组合使用。

移动端测试工具分享：

抓包工具

Lyrebird - 移动端的抓包及Mock平台（美团）。

spy-debugger - 适用于任何手机移动端webview的一站式页面调试、抓包工具。

nohost - 基于 Whistle 实现的多账号多环境远程配置及抓包调试平台。（腾讯）

性能测试

PerfDog - 一款用于采集手机（Android&iOS）性能数据的 PC 端工具。（腾讯）

mobileperf - 基于 Python 实现的 Android 性能采集工具。（阿里）

Pyroscope - 基于 Go 的开源实时性能分析平台.能帮你找出代码的性能问题、CPU 使用过高的原因，还有丰富的图表和调用树展示。

LoliProfiler - 用于 Android 游戏和应用程序的C/C ++ 内存分析工具。（腾讯）

MemoryLeakDetector - 西瓜视频安卓团队开发的原生内存泄漏监控工具。（字节）

Tailor - 西瓜视频安卓团队开发的一款通用的hprof裁剪压缩工具。（字节）

Booster - 一款易于使用、轻量级、功能强大且可扩展的质量优化工具包，专为移动应用程序设计.首要目标是解决APP复杂度增加带来的质量问题，如性能、稳定性、包大小等。（滴滴）

自动化

跨平台

**Appium - 用于 iOS、Android 和 Windows Apps 的自动化。**

Detox - 端到端的移动应用测试和自动化框架。

Karate - UI/API 自动化测试框架。

**Android**

SoloPi - 支持录制回放、一机多控和性能测试的 Android 自动化测试工具.（蚂蚁）

scrcpy- 显示和控制你的 Android 设备.（Genymobile）

minicap - 从 Android 设备中流式传输实时屏幕捕获数据.（STF）

adb\_shell - ADB 的 Python 实现，具有 Shell 和 FileSync 功能.

python-adb - Python ADB + Fastboot实现.（Google）

ADB Event Mirror - 通过命令行实现 Android 设备的一机多控.

ADB Enhanced - 增强 ADB 命令.

uiautomator2 - Android Uiautomator2 Python包装器.（openatx）

Moby - E2E 测试框架，用于具有复杂环境要求的测试.（Google）

Docker-Android - Android in docker 解决方案，支持 noVNC 和视频录制.

Fastbot-Android - Android APP 稳定性测试工具.（字节）

Shot - 适用于 Android 的屏幕截图测试库。

**iOS**

WebDriverAgent - iOS 的 WebDriver 服务实现，可用于远程控制 iOS 设备.

python-wda - Facebook WebDriverAgent Python 客户端库.（openatx）

tidevice - iOS 自动化命令行工具.（阿里）

libimobiledevice - 使用本机协议与iOS设备上的服务进行通信.

pymobiledevice - libimobiledevice 跨平台软件库的 python 实现.

idb - 用于自动化 iOS 模拟器和设备的命令行工具.

EarlGrey - iOS UI自动化测试框架.（Google）

KIF - 保持功能正常 - iOS 功能测试框架.

SwiftMonkey - 对 iOS 应用程序进行随机 UI 测试的框架.

Fastbot-iOS - iOS APP 稳定性测试工具.（字节）

Mixbox - iOS UI 测试框架.

微信小程序

Minium - 微信小程序自动化框架, 提供了 Python 和 JavaScript 版本.（腾讯）

测试平台

STF - 基于 Agular 和 Node.js 实现的移动设备管理平台 .

atxserver2 - 基于 Python 实现的移动设备管理平台.

DoraemonKit - 一款功能齐全的客户端（ iOS 、Android、微信小程序、Flutter ）研发助手.（滴滴）

史上最全测试开发工具推荐（含自动化、APP性能、稳定性、抓包神器）

在本篇文章中，将给大家推荐14款日常工作中经常用到的测试开发工具神器，涵盖了自动化测试、APP性能测试、稳定性测试、抓包工具等.

一、UI自动化测试工具

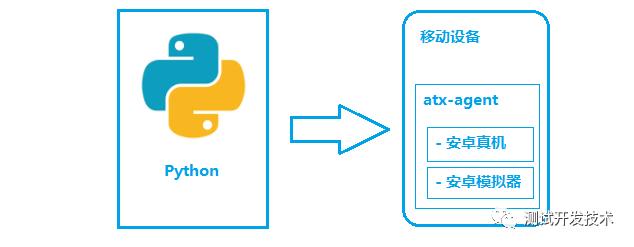
1. uiautomator2

Github地址

https://github.com/openatx/uiautomator2介绍:

openatx开源的ui自动化工具，支持Android和iOS.主要面向的编程语言是Python，API设计简洁易用，在开源社区也是很受欢迎.

原理图：



安装：

pip install --upgrade --pre uiautomator2# Or you can install directly from github sourcegit clone https://github.com/openatx/uiautomator2pip install -e uiautomator22. Appium

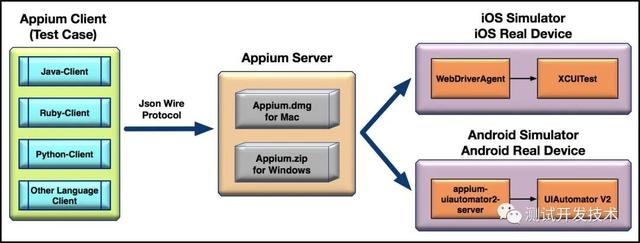
Github地址：

<https://github.com/appium/>

appium介绍:

基于Node.js的一款非常成熟的APP UI自动化工具，支持多种语言开发，在测试圈非常流行，可以说是APP UI自动化领域的标杆产品.如果你不知道Appium，都不好意思说自己干测试这行.

原理图：



3. ATX-Test

Github地址：

https://github.com/pengchenglin/ATX-Test介绍:

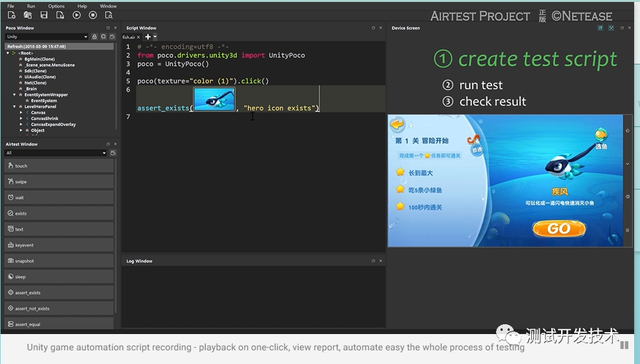
基于uiautomator2和unittest封装的一个UI自动化测试框架.项目作者已经把框架设计、可配置化、底层设备连接、日志模块、报告模块都做好了，使用者只需要编写PageObject和脚本逻辑即可，大大提高了uiautomator2的易用性.不仅如此，该项目现在还支持Monkey运行.

4. Airtest

Github地址：

https://github.com/AirtestProject/Airtest介绍:

网易开源的一个ui自动化测试工具，通过图像识别和UI控件检索技术，使得游戏测试简单快捷.支持Android、iOS、Windows、Unity、Cocos2dx、白鹭引擎、微信小程序，上手简单，游戏测试的朋友可以体验体验.



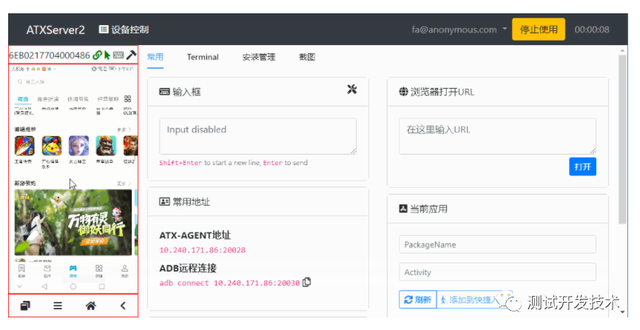
5. ATXServer2

Github地址：

https://github.com/openatx/atxserver2介绍:

支持Android和iOS的移动设备管理平台.可以通过web界面，察看测试手机的状态，可以做到远程操控手机，对于测试兼容性具有较大帮助。部署方便，支持docker-compose部署。



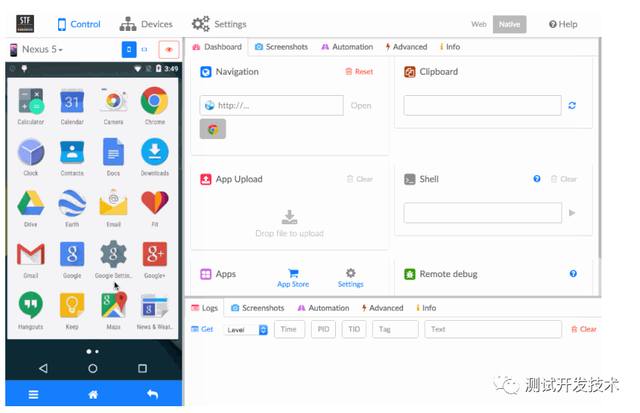


6. STF

Github地址：

https://github.com/openstf/stf介绍:

远程移动设备管理平台，支持Android、iOS真机管理，功能非常强大.



7. Appetizer

官网地址：

https://www.appetizer.io/cn/介绍:

使用十分方便.可以进行APP UI自动化测试，也可以支持APP稳定性测试，提供GUI界面操作，易用性极强，满足日常UI自动化需要.可惜只有部分组件开源。



二、APP稳定性测试工具

8. UICrawler

Github地址

https://github.com/lgxqf/UICrawler介绍:

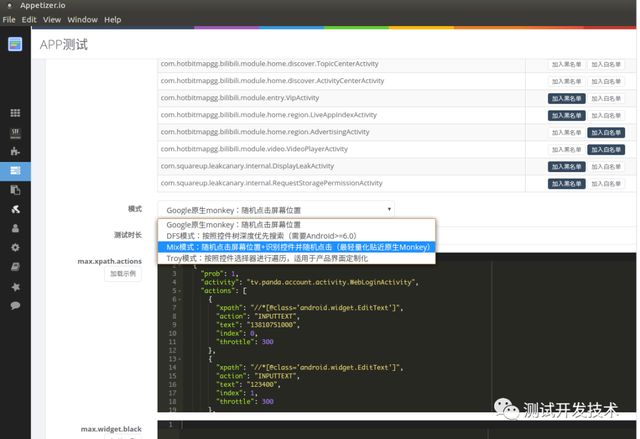
基于Appium的App UI遍历 & Monkey 工具，支持iOS和Android

9. Maxim

Github地址

https://github.com/zhangzhao4444/Maxim介绍:

号称是智能monkey，用过的人都觉得真香.支持原生monkey、遍历控件、黑白名单等功能，而且也封装了更加易解析的日志文件.有了它之后，已经可以不需要再用原生monkey了。



10. AppCrawler

Github地址

https://github.com/seveniruby/AppCrawler介绍:

一个基于自动遍历的app爬虫工具. 支持android和iOS, 支持真机和模拟器. 最大的特点是灵活性. 可通过配置来设定遍历的规则.

三、APP性能测试工具

11. SoloPi

Github地址

https://github.com/alipay/SoloPi介绍:

SoloPi是一个无线化、非侵入式的Android自动化工具，公测版拥有录制回放、性能测试、一机多控三项主要功能，能为测试开发人员节省宝贵时间（官方介绍）.一般用于APP的性能测试比较多.

12. GT

Github地址

https://github.com/Tencent/GT介绍:

腾讯开源的APP的随身调试平台，可对APP进行快速的性能测试(CPU、内存、流量、电量、帧率/流畅度等等)、开发日志的查看、Crash日志查看、网络数据包的抓取、APP内部参数的调试、真机代码耗时统计等.



利用 GT，仅凭一部手机，无需连接电脑，即可对 APP 进行快速的性能测试(CPU、内存、流量、电量、帧率/流畅度等等)、开发日志的查看、Crash 日志查看、网络数据包的抓取、APP 内部参数的调试、真机代码耗时统计等.如果您觉得 GT 提供的功能还不够满足您的需要，您还可以利用 GT 提供的基础 API 自行开发有特殊功能的 GT 插件，帮助您解决更加复杂的 APP 调试问题.

GT 支持 iOS 和 Android 两个手机平台，其中：

iOS 版是一个 Framework 包，必须嵌入 APP 工程，编译出带 GT 的 APP 才能使用；iPhone 和 iPad 应用都能支持.Android 版由一个可直接安装的 GT 控制台 APP 和 GT SDK 组成，GT 控制台可以独立安装使用，SDK 需嵌入被调测的应用、并利用 GT 控制台进行信息展示和参数修改。

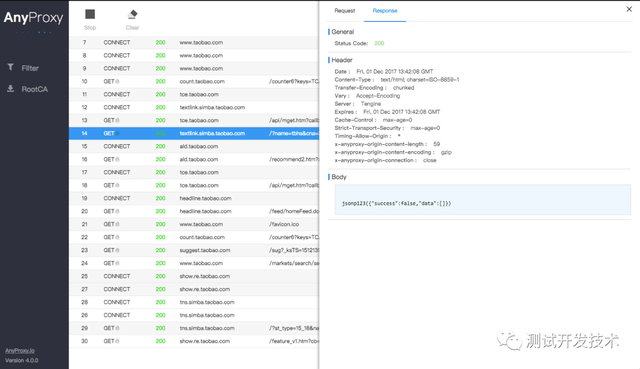
四、抓包工具

13. AnyProxy

Github地址

https://github.com/alibaba/anyproxy介绍:

阿里开源的抓包工具，基于Node.js可进行二次开发，使用场景：app爬虫、接口数据分析等.

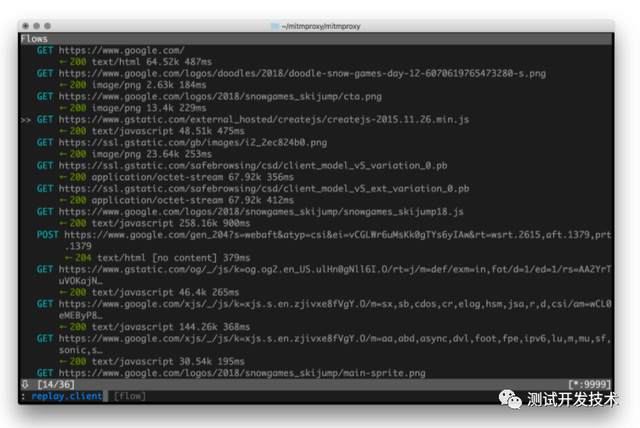


14. mitmproxy

Github地址

https://github.com/mitmproxy/mitmproxy介绍:

基于Python的抓包工具，对于pythoner更加友好，主要作用是对抓包数据进行采集供后续分析。



# 8 测试工具包

常用的测试工具大体分为10类：

一、 测试管理工具

二、接口测试工具

三、性能测试工具

四、C/S自动化工具

五、白盒测试工具

六、代码扫描工具

七、持续集成工具

八、网络测试工具

九、app自动化工具

十、web安全测试工具

一、测试管理工具

1.TestDirector(大而全)

2.**jira**(简单好用)

3.Quality Center(复杂，收费)

4.禅道（简单好用）

5.bugzilla(功能简单)

6.svn(代码和文档管理工具)

7.vss类似svn

8.**git**，同svn，但是多分支管理比svn好

9.Note（大而全，费用太贵）

10.CQ(ClearQuest-IBM产品-大而全)

二、接口测试工具

**1.Jmeter（开源）**

**2.postman**

3.SoapUI

推荐使用 jmeter 和 postman

jmeter是一款100%纯Java编写的免费开源的工具，它主要用来做性能测试，相比loadrunner来说，它内存占用小，免费开源，轻巧方便、无需安装，越来越被大众所喜爱。Postman是谷歌的一款接口测试插件，它使用简单，支持用例管理，支持get、post、文件上传、响应验证、变量管理、环境参数管理等功能，可以批量运行，并支持用例导出、导入。

三、性能测试工具

1.**loadrunner**，大而全，要学精通还是有点难度，重量级工具

2.**jmeter** 基于java平台的性能开源测试工具，其实也很强大，而且比较好用

3.Web bench 一个简单的web基准指标测试工具

4.Load UI，一款开源的压力测试工具，支持图形化

5.httperf 一款高性能的web性能测试工具

6.Siege 一款开源的压力和指标测试工具

7.Gatling

前两种是比较常用的

四、C/S自动化工具

1.qtp (录制回放和脚本编辑)，用到的是vb语言

2.winrunner IBM产品类似qtp

3.autoit 在窗口定位上做到很不错

五、白盒测试工具

1.jtest java语言的单元测试框架

2.JUnit 验证java的工具

3.cppunit 跨平台的c++单元测试框架

4.gtest 跨平台的c++单元测试框架

5.PhpUnit Php

6.BoundsChecker C++,Delphi API和OLE错误检查、指针和泄露错误检查、内存错误检查

7.TrueTime C++,Java,Visual Basic 代码运行效率检查、组件性能的分析

六、代码扫描工具

1.Coverity源代码静态分析工具

2.cppcheck c++静态扫描工具

3.gcover代码覆盖率工具

4.findbugs：基于字节码分析，大量使用数据流分析技术，侧重运行时错误检测，如空指针引用等

5.SonarLint

6.TscanCode

七、持续集成工具

1.jenkins

2.Hudson

八、网络测试工具

1.思博伦 目前流行的一款网络自动化测试商用平台了(而且能够完全顶替loadrunner)，基本上能够满足所有的网络产品测试需求了，不过很贵

2.Ixia 也是对网络设备进行性能和压力测试的平台

3.wireshark 数据包抓取分析和回放测试工具

4.tc 网络丢包和试验模拟工具，非常好用

5.iperf 用来测试tcp和udp的网络质量

6.tcpping工具工作在 TCP 层，通过发送伪造的 TCP SYN 包并侦听来自服务器或中间设备返回的 SYN/ACK 或 RST

九、app自动化工具

**1.appium 这个应该算是目前最流行的基于app的自动化测试框架了**

2.instruments ios平台下的自动化测试框架，用java语言写的

3.uiautomator安卓自动化测试框架，基本上支持安卓的所有事件操作

4.Monkey 安卓自带的测试工具

5.Monkey Runner Monkey改进版，支持自己编写脚本测试，用Python语言

6.Robotium 一款国外的Android自动化测试框架，用法比较简单

十、web安全测试工具

金融服务和银行业一直是安全漏洞的受害者，因为会破坏了大量敏感的用户数据。然而，金融服务是每个人的必备品。所以在这里我们列出了一些安全测试工具，用于构建一个健壮的应用程序。

1.appscan 算是用的非常多的一款工具了，扫描后能够将绝大部分的漏洞找出来（IBM Security AppScan Standard）

2.Netsparker Community Edition 这个程序可以检测SQL注入和跨页脚本事件.牛逼的是还能提供解决方案

3.Websecurify 这是个简单易用的开源工具，此程序还有一些人插件支持，可以自动检测网页漏洞.运行后可生成多种格式的检测报告

4.Wapiti 这是一个用Python编写的开源的工具，可以检测网页应用程序，探测网页中存在的注入点

5.N-Stalker Free Version 此工具可一次检测100个以上的页面，包括跨页脚本的检测

6.skipfish 这是一个轻量级的安全测试工具，处理速度很快，每秒可处理2000个请求

7.Scrawlr HP的一款免费软件，可检测SQL注入漏洞

8.Watcher: 这个是Fiddler的插件，可在后台静默运行，可检测跨域提交等

9.WebScarab 这个实际上是一个代理软件，有很多功能，可以检测XSS跨站脚本漏洞、SQL注入漏洞等

10.抓包工具:fiddler

11.burpsuite：暴力破解、抓包工具

# 9 人工智能测试方法

## 9.1了解人工智能

### 9.1.1人工智能的定义

人工智能AI（Artificial Intelligence）是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。通俗的说，就是让机器可以像⼈类⼀样有智能，让机器看得懂、听得懂、会思考、能决策、能⾏动，实现原来只有⼈类才能完成的任务。人工智能利用机器学习技术，通过对现有的经过处理（筛选、消噪、过滤等）的数据，不断进行矫正（设置阀值等方法）机器模型的输出，此过程称为训练，期望通过训练可以得到在未来新数据上有良好表现的模型，从而投入生产。

⼈⼯智能（AI）的本质是通过软件来实现特定的算法。

⼀个优秀的⼈⼯智能系统，应该具有三个方面的特征：知识运用的能力、从数据或经验中学习的能力、处理不确定性的能力。

### 9.1.2 人工智能应用领域

（1）计算机视觉：使用计算机及相关设备对生物视觉的一种模拟，通过对采集的图片或视频进行处理以获得相应场景的三维信息。如应用在图像识别领域的高速车牌识别、人脸识别等。

（2）自然语言处理（NLP）：用计算机对自然语言的形、音、义等信息进行处理，即对字、词、句、篇章的输入、输出、识别、分析、理解、生成等的操作和加工。如电子病历的症状提取。

（3）[语音识别](https://so.csdn.net/so/search?q=%E8%AF%AD%E9%9F%B3%E8%AF%86%E5%88%AB&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/aasdad1/article/details/_blank)：人工智能在语音识别方面的应用相对较好，如siri、多邻国读音识别等。

（4）个性化推荐：如亚马逊、今日头条根据用户阅读历史做的推荐系统，利用人工智能进行调参数等。

（5）深度学习：深度学习是基于现有的数据进行学习操作，其动机在于建立、模拟人脑进行分析学习的神经网络，它模仿人脑的机制来解释数据。

（6）数据挖掘：数据挖掘可以分为直接数据挖掘和间接数据挖掘。

### 9.1.3 人工智能应用详解

#### 9.1.3.1 自然语言处理

1.学习目标

• 了解自然语言处理基本知识

• 掌握循环神经网络算法

• 掌握自然语言处理关键技术

• 了解自然语言处理的应用

2.自然语言处理介绍

（1）什么是自然语言

• 以语音为物质外壳，由词汇和语法两部分组成的符号系统。文字和声音是语言的两种属性。

• 语言是人类交际的工具，是人类思维的载体；人类历史上以语言文字形式记载和流传的知识占人类知识总量的80%以上。

• 是约定俗成的，有别于人工语言，比如Java、C++等程序设计语言。

（2）什么是自然语言处理

• 自然语言处理（Natural Language Processing，NLP)是人工智能和语言学领域的学科分支，它研究实现人与计算机之间使用自然语言进行有效通信的各种理论和方法。

• 自然语言处理就是，利用计算机为工具对人类特有的书面形式和口头形式的自然语言的信息，进行各种类型处理和加工的技术。----冯志伟

• 自然语言处理可以定义为研究在人与人交际中以及在人与计算机交际中的语言问题的一门学科。自然语言处理要研制表示语言能力和语言应用的模型，建立计算框架来实现这样的语言模型，提出相应的方法来不断完善这样的语言模型，根据这样的语言模型设计各种实用系统，并探讨这些实用系统的评测技术。----Bill Manaris

（3）自然语言处理的基本方法

• 能力模型-基于规则的方法

• 通常是基于语言学规则的模型，建立在人脑中先天存在语法通则这一假设的基础上，认为语言是人脑的语言能力推导出来的，建立语言模型就是通过建立人工编辑的语言规则集来模拟这种先天的语言能力。

• 又称“理性主义的”语言模型，代表人物有Chomsky、Minsky。

o 制定语法规则（通用）：<noun名词><verb动词><adj形容词>

o 制定语义规则（场景化）：

o 使用规则进行校验

• 建模步骤：

o 语言学知识形式化

o 形式化规则算法化

o 算法实现

• 应用模型–基于统计的方法

• 根据不同的语言处理应用而建立的特定语言模型，通常是通过 建立特定的数学模型来学习复杂的、广泛的语言结构，然后利用统计学、模式识别和机器学习等方法来训练模型的参数，以扩大语言使用的规模。

• 又称“经验主义”语言模型，代表人物有Shannon、 Skinner。

o 从大量的英文文本中统计词语的组合搭配频次

o 构建统计模型计算两个句子的概率

• 建模步骤

o 大规模真实语料库中获得不同层级语言单位上的统计信息。

o 依据较低级语言单位上的统计信息运用相关的统计推理技术，来计算较高级语言单位上的统计信息。

自然语言处理研究方向

• 自然语言处理是计算机科学领域以及人工智能领域的一个重要的研究方向，是一门交叉性学科，包括了语言学、计算机科学、数学、心理学、信息论、声学……

• 音位学

• 描述音位的结合规律，说明音位怎样形成语素。

• 举例：delete file x -> dilet#fail#eks

• 形态学

• 描述语素的结合规律，说明语素怎样形成单词。

• 举例：unusually -> un+usual+ly

• 词汇学

• 描述词汇系统的规律，说明单词本身固有的语义特性和语法特性。

• 举例：delete file x -> delete(VERB) file(NOUN) x(ID)

• 句法学

• 描述单词或词组之间的结构规则，说明单词或词组怎样构成句子。

• 举例：delete file x -> (delete (file x)\_NP)\_VP

• 语义学

• 描述句子中各成分之间的语义关系，怎样从构成句子的各个成分推导出整个句子的语义。

• 举例：delete file x -> DELETE(FILE(id=x))

• 语用学

• 描述与情景有关的情景语义说明 怎样推导出句子具有的与周围话语有关的各种含义。

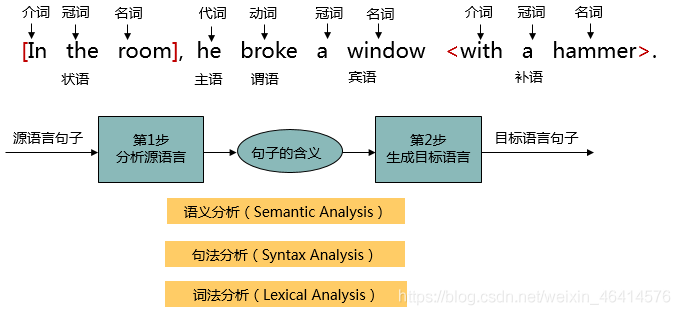
• 举例：感染病毒->生物病毒 or 计算机病毒

自然语言处理的三个层面

• 词法分析：包括分词、词性标注、命名实体识别等。

• 句法分析：包括句法结构分析和依存关系分析等。短语结构句法体系，依存结构句法体系，深层文法句法分析。

• 语义分析：最终目的是理解句子表达的真实语义



难点

• 词法歧义：

• 分词：词语的切分边界比较难确定。

o 严守一/把/手机/关/了

o 严守/一把手/机关/了

• 词性标注：同一个词语在不同的上下文中词性不同。

o 我/计划/v 考/研/

o 我/完成/了/计划/n

• 命名实体识别：人名、专有名称、缩略词等未登录词的识别困难。

o 高超/nr/a 华明/nr/nt 移动/nt/v

• 句法歧义：句法层面上的依存关系受上下文的影响。

• 咬死了猎人的狗

o 那只狼咬死了猎人的狗

o 咬死了猎人的狗失踪了

• 语义歧义：

• At last, a computer understands you like your mother.

o 含义1：计算机会像你的母亲那样很好的理解你。

o 含义2：计算机理解你喜欢你的母亲。

o 含义3：计算机会像理解你母亲那样去理解你。

• 语用歧义：

• “你真坏”

o 当对干了坏事的成年人说时，是一种严厉的苛责。

o 当妈妈对淘气的儿子说时，实际表达的是对儿子的一种疼爱。

o 当恋爱中的女孩对男友说时，则是女孩在男友面前撒娇的一种表现。

## 9.2 AI测试

### 9.2.1 AI测试的特点

AI测试是对基于AI的应用系统的测试。与任何系统⼀样，基于AI的系统具有功能性和⾮功能性需求。某些传统的测试⽅法也会在AI软件测试中用到，如在代码级的单元测试，以及AI系统的⼀些性能测试、可伸缩性测试、安全性测试也是要做的。

目前AI测试主要是对⼈⼯智能（AI）应用最为⼴泛的四个领域的系统测试，即对自然语言处理、图像识别、推荐系统、机器学习这四个方面的测试。

由于AI系统与传统系统所采用的技术有很多的不同，如机器学习、逻辑程序、规则引擎、演绎分类器、神经网络等，以及AI系统的特点是不确定性和概率性、对⼤数据具有依赖性、随机性的输⼊/输出、难以预测所有应用场景、需要从过去的⾏为中不断自我学习。传统的⽅法和测试⼯具将不能满足测试这些系统的需要。

### 9.2.2 AI测试的内容和类型

一般这些项目都要测试什么，要进行什么类型的测试。

1.模型评估测试

模型评估主要是测试模型对未知新数据的预测能力，即泛化能力。泛化能力越强，模型的预测能力表现越好。而衡量模型泛化能力的评价指标，就是性能度量(performance measure)。性能度量一般有错误率、准确率、精确率、召回率等。

2.稳定性/鲁棒性测试

稳定性/鲁棒性主要是测试算法多次运行的稳定性；以及算法在输入值发现较小变化时的输出变化。如果算法在输入值发生微小变化时就产生了巨大的输出变化，就可以说这个算法是不稳定的。

3.系统测试

将整个基于算法模型的代码作为一个整体，通过与系统的需求定义作比较，发现软件与系统定义不符合或与之矛盾的地方。

系统测试主要包括以下三个方面：

（1）项目的整体业务流程

（2）真实用户的使用场景

（3）数据的流动与正确

4.接口测试

接口测试是测试系统组件间接口的一种测试。接口测试主要用于检测外部系统与系统之间以及内部各个子系统之间的交互点。测试的重点是要检查数据的交换，传递和控制管理过程，以及系统间的相互逻辑依赖关系等。

5.文档测试

文档测试是检验用户文档的完整性、正确性、一致性、易理解性、易浏览性。

在项目的整个生命周期中，会得到很多文档，在各个阶段中都以文档作为前段工作成果的体现和后阶段工作的依据。为避免在测试的过程中发现的缺陷是由于对文档的理解不准确，理解差异或是文档变更等原因引起的，文档测试也需要有足够的重视。

6.性能测试

7.白盒测试–代码静态检查

8.竞品对比测试

如果有涉及时，可针对做竞品对比测试，清楚优势和劣势。比如AI智能音箱产品。

9.安全测试

发布上线后，线上模型监控

1. 测试数据
2. 不管是[机器学习](https://so.csdn.net/so/search?q=%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/aasdad1/article/details/_blank)，推荐系统，图像识别还是自然语言处理，都需要有一定量的测试数据来进行运行测试。
3. 算法测试的核心是对学习器的泛化误差进行评估。为此是使用测试集来测试学习器对新样本的差别能力。然后以测试集上的测试误差作为泛化误差的近似。测试人员使用的测试集，只能尽可能的覆盖正式环境用户产生的数据情况。正式环境复杂多样的数据情况，需要根据上线后，持续跟进外网数据。算法模型的适用性一定程度上取决于用户数据量，当用户量出现大幅增长，可能模型会随着数据的演化而性能下降，这时模型需要用新数据来做重新训练。

上线只是完成了一半测试，并不像APP或者WEB网站测试一样，测试通过后，发布到正式环境，测试工作就完成了。

测试集如何选取很关键，一般遵循两个原则：

测试集独立同分布

测试数据的数量和训练数据的比例合理；不能使用训练数据来做为测试数据，此为独立。

测试数据需要和训练数据是同一个分布下的数据，此为分布。

举个例子，训练数据中正样本和负样本的分布为7：3，测试数据的分布也需要为7：3，或者接近这个分布，比较合理

测试数据的数量和训练数据的比例合理

当数据量比较小时，可以使用 7 ：3 训练数据和测试数据

(西瓜书中描述 常见的做法是将大约 2/3 ~4/5 的样本数据用于训练，剩余样本用于测试）

或者 6：2 : 2 训练数据，验证数据和测试数据。

如果只有100条，1000条或者1万条数据，那么上述比例划分是非常合理的。如果数据量是百万级别，那么验证集和测试集占数据总量的比例会趋向于变得更小。如果拥有百万数据，我们只需要1000条数据，便足以评估单个分类器，并且准确评估该分类器的性能。假设我们有100万条数据，其中1万条作为验证集，1万条作为测试集，100万里取1万，比例是1%，即：训练集占98%，验证集和测试集各占1%。对于数据量过百万的应用，训练集可以占到99.5%，验证和测试集各占0.25%，或者验证集占0.4%，测试集占0.1%。

一般算法工程师会将整个数据集，自己划分为训练集、验证集、测试集。或者训练集、验证集 等等。（这里的测试集是算法工程师的测试数据）

算法工程师提测时，写明自测时的准确率或其他指标。测试人员另外收集自己的测试集。

测试数据可以测试人员自己收集。或者公司的数据标注人员整理提供。或者爬虫。外部购买。

测试人员可以先用算法工程师的测试集进行运行测试查看结果。再通过自己的测试集测试进行指标对比。

### 9.2.3测试分析

人工智能归根结底也是利用对历史数据的处理训练出可以在将来数据上有良好输出的模型。

对于测试而言，应该关心数据模型在对待正常数据、边界数据、异常数据作为输入时，模型的输出是否能够符合期望。

### 9.2.4测试方法

改变测试集：如输入与训练时一样的数据、与训练时完全不同的数据、训练时的边界值等，看是否达到期望输出。

如在安卓平台运行的代码：通过不断点击运行、以及快速退出和快速进入、处理大量数据、空数据等观察性能指标的上升等。

模型是否有良好的用户交互

人工智能发展目前有一定的技术限制，但是无论如何都不能造成应用crash、卡死、内存溢出等现象

具体使用时，应有良好的告知用户的提示，不能一直loading等

模型是否能够根据处理数据的量从少到多而自动不断优化、调整输出

观察模型输出是否是一直不变化的

经过多次改变输入（百次计算），再进行回归测试，观察输出是否有一定程度调优（或者更差了）

模型在处理数据时的效率（学习过程，cpu占用率、内存消耗等）

模型有没有人性化的参数调整入口，供运营人员以及测试人员对上线后、上线前进行调整

模型上线后应具有一定的参数调整能力（例如某些权重的调整等。业界今日头条的某些推荐方案一旦效果好，据说会立刻将所有的模型进行模拟升级（切换到相同的模式）

风险控制，当发现严重问题时如何良好的控制线上的模型，对其进行开关以及升级操作，如上线后若发现难以控制的风险，如需要紧急下线（政策等影响）等问题时，需要有立刻关闭的功能以及关闭前对用户的良好的提示功能

若此模型并不是单独使用，有没有良好的兼容性（兼容其他模型），遇到错误的使用时如何变现（以及提示方式）

如模型需要与其他模型进行合作才能工作，那应当分开单独进行测试，此模型应该具有良好的接口，和期望输出。测试方法参照上面。然后再测试与其他模型共同工作时的效果

### 9.2.3 AI测试举例

#### 9.2.3.1语音识别部分

输入正常的语音

输入有杂音的语音

输入空白语音

输入不同语言的语音

输入长时间语音

输入重复语音

结论：训练好的模型应在使用上满足一定程度的场景，不能答非所问的太离谱

#### 9.2.3.2自优化测试

将测试集分成2部分（或多部分），第一次输入第一部分然后观察结果，然后再输入第二部分，然后再次输入第一部分，观察输出是否有优化的体现。

#### 9.2.3.3性能部分

在进行语音识别时，观察cpu、内存等占用情况

在语音识别结束观察cpu、内存有没有释放等情况

观察识别的时间长短

#### 9.2.3.4友好度测试

观察在使用时弱网络情况下的提示

观察识别出错时的提示

观察正常时的提示

#### 9.2.3.5风险测试

功能入口测试（展示和关闭），是否能通过远程直接关闭或开启语音识别功能，升级等。

# 10 Jemeter测试实战

## 10.1 Jemeter性能测试报告

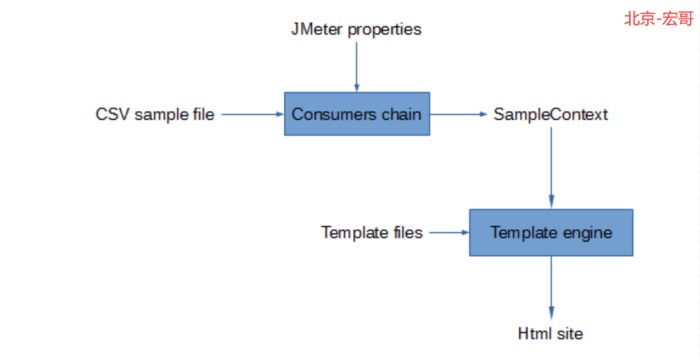
### 1.简介

 　　相对于Loadrunner，Jmeter其实也是可以有测试报告产出的，虽然一般都不用（没有Loadrunner的报告那么强大是一方面），但是有小伙伴们私下问，那宏哥还是顺手写一下吧，今天我们就来学习下，如何输入HTML格式的JMeter测试报告。前面已经介绍， 如果要做性能测试，需要在GUI上设计好你的Test Plan，设置各种场景和负载值，包括多少个线程，多少个用户，循环多少次。设置好了保存，然后用命令行去启动性能测试，查看相关测试结果。

### 2.Jmeter生成HTML格式报告的架构

仪表板生成引擎是基于示例操作流程的模块化功能。

下图可以表示这些过程：



在此视图中，可以看到：

产生样品的来源（例如CSV文件）。

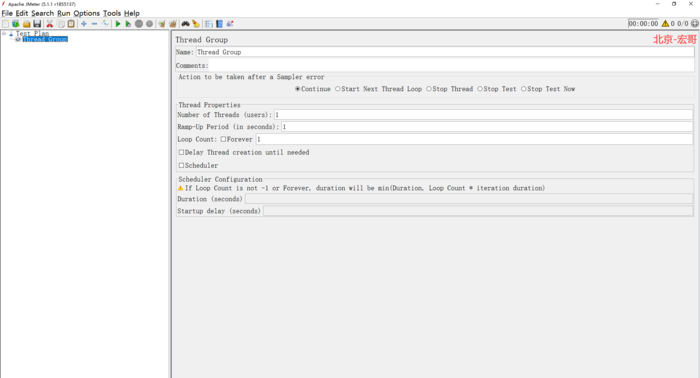
一系列称为消费者的项目，对通过该项目的样本进行操作（例如，筛选，排序，计算等）。

执行上下文，称为样本上下文，在其中存储使用者计算的结果。

一组名为出口器的项目，它们使用示例上下文的内容为用户生成最终结果（例如HTML页面生成）。

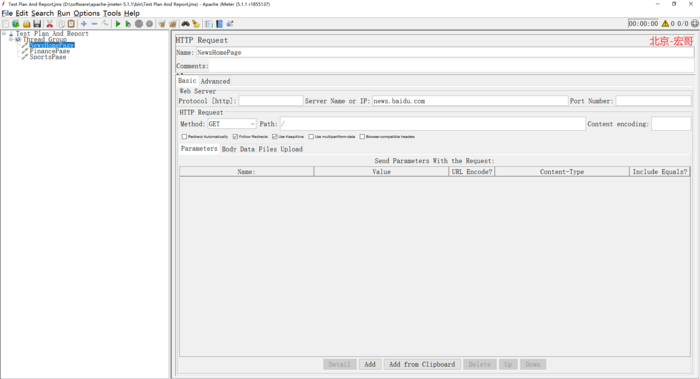
### 3.具体步骤

1. 打开图像（gui）界面，新建一个Thread Group，如下图所示：

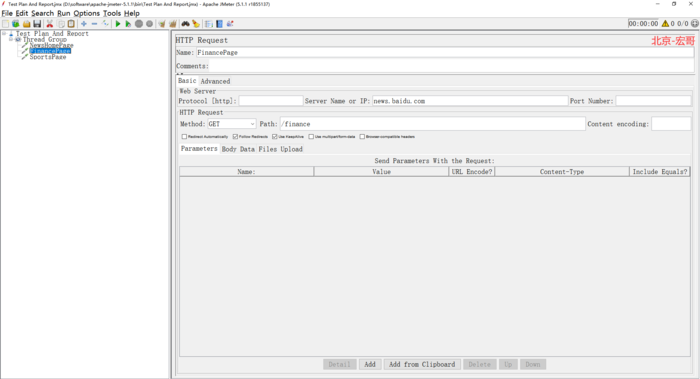


2. 新建三个Http Request，分别是百度新闻首页，和百度新闻财经，百度新闻体育，如下图所示：

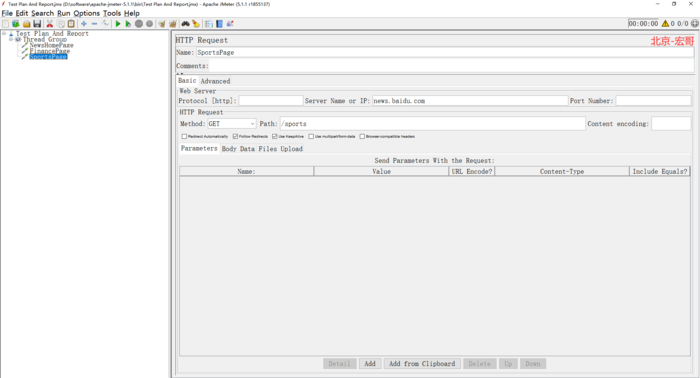
百度新闻首页



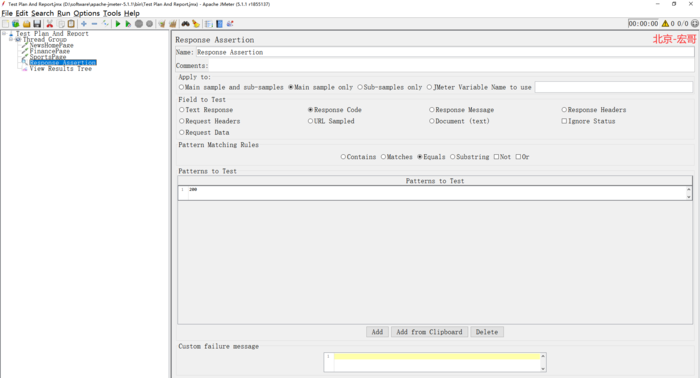
百度新闻财经



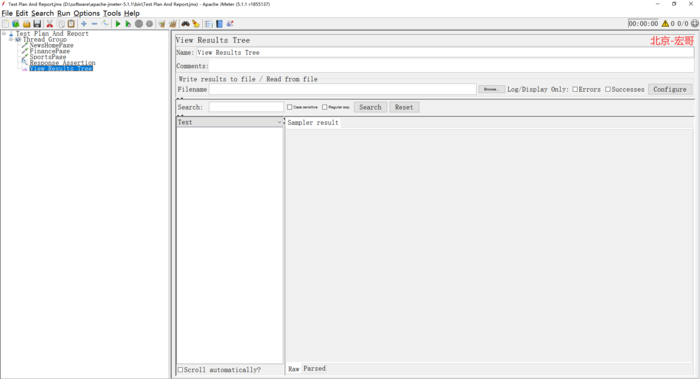
百度新闻体育



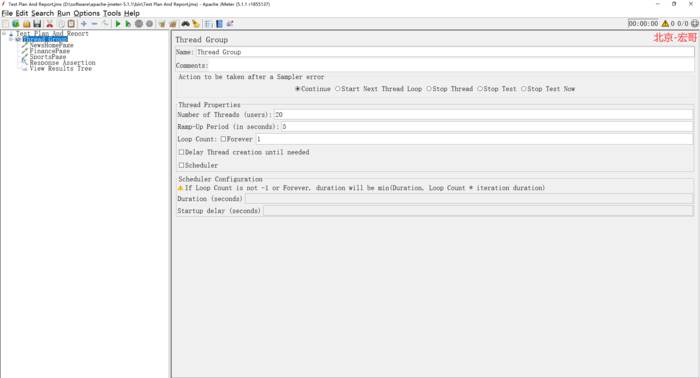
3. 添加一个Assertion,选择response assertion，如下图所示：



4. 添加一个View Results Tree.如下图所示：



5. Thread Group设置如下图所示：

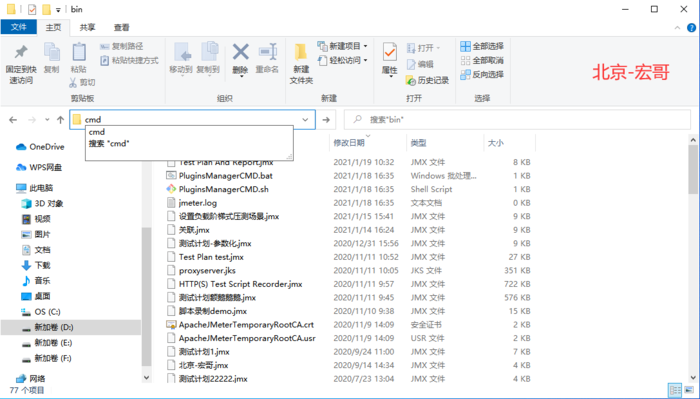


这里，你在图形界面的时候，最好设置都是1，测试通过后，再改成多个用户，然后保存本地xx.jmx文件

6. 保存到本地，关闭JMeter。

7.本地先在桌面创建一个文件夹HttpReports，用来保存HTML报告。

8. cmd启动性能测试，如下图所示：



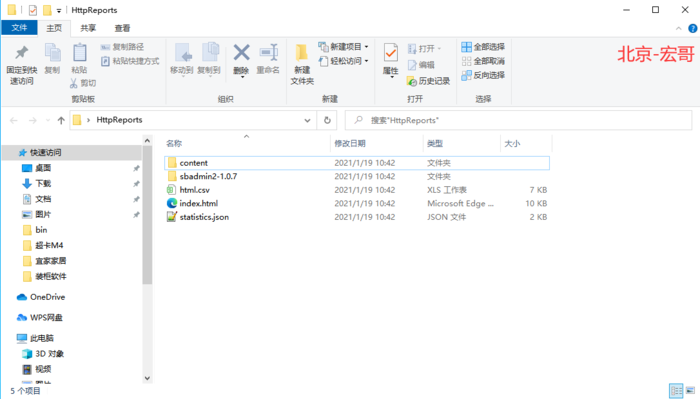
9.cmd 先要cd到jmeter的bin目录，然后输入以下命令：

jmeter -n -t D:\software\apache-jmeter-5.1.1\bin\Test Plan And Report.jmx -l C:\Users\DELL\Desktop\HttpReports\html.csv -e -o C:\Users\DELL\Desktop\HttpReports

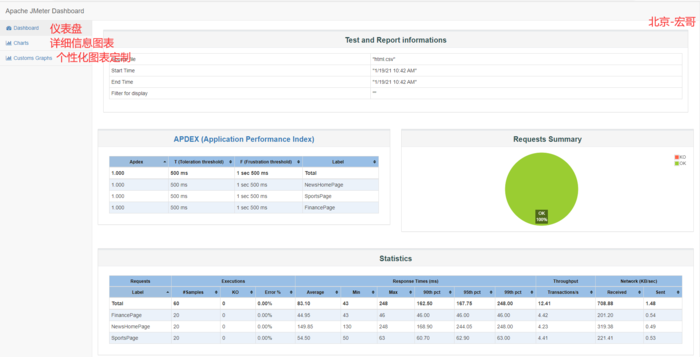
三个路径依次是：xxx.jmx文件路径，输出xxx.csv文件路径，要保存的http report的文件路径，运行结果如下图所示：



10. 查看HTTP 报告，在桌面的HttpReports文件夹，生成了多个文件，如下图所示：



11.用浏览器打开这个html文件，自己随便点击，看看报告，如下图所示：



### 4.****图表信息详解****

Jmeter5.0后测试报告分为三部分，Dashboard、Charts和Customs Graphs，Jmeter5.0以下版本没有第三项的。下面分开讲解。

****4.1 Dashboard（概览仪表盘）****

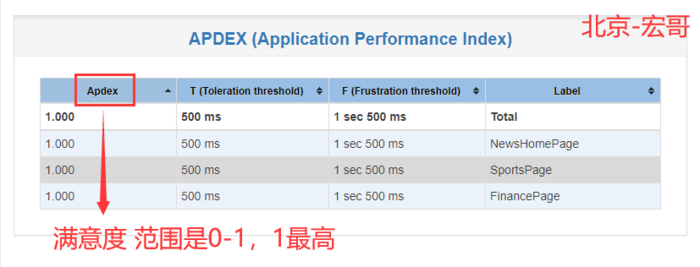
****①、Test and Report informations****

Test and Report informations：测试和报告信息: 测试结果保存文件/测试开始时间/测试结束时间/展示过滤器。



****②、APDEX (应用性能指标)****

APDEX(Application Performance Index)：应用程序性能满意度的标准，范围在 0-1之间，1表示达到所有用户均满意，可以在配置文件设置。关于APDEX的相关信息，请参考这里：[应用性能指标](http://oneapm.udesk.cn/hc/articles/515" \t "https://www.likecs.com/_blank)；英文原文，参考这里：[Apdex-Wikipedia](https://en.wikipedia.org/wiki/Apdex" \t "https://www.likecs.com/_blank)



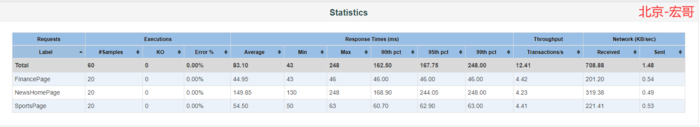
****③、Requests Summary****

Requests Summary: 请求的通过率(OK)与失败率(KO)，百分比显示。



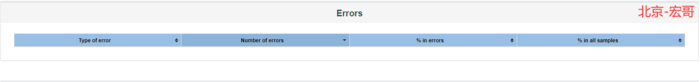
****④、Statistics****

统计表在一个表中提供每个事务的所有指标的摘要，包括3个可配置的百分位数。Statistics: 数据分析，基本将 Summary Report 和 Aggrerate Report 的结果合并。



****⑤、Errors****

错误表提供了所有错误及其在总请求中所占比例的摘要。Errors: 错误情况，依据不同的错误类型，将所有错误结果展示。



****⑥、Top 5 Errors by sampler****

前5个错误（按Sampler列出）表为每个Sampler（默认情况下不包括Transaction Controller）提供前5个错误：Top 5 Errors by sampler：Top5错误信息采样



****4.2 Charts（详细信息图表）****

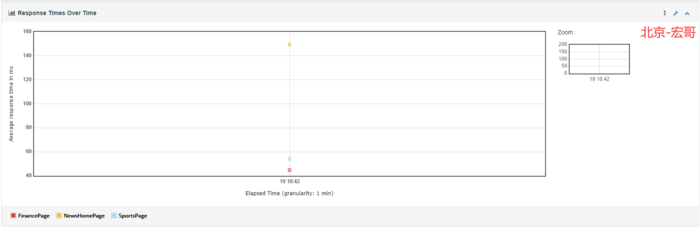
Charts里如图所示分为三大块：时间维度信息(Over Time)、吞吐量(Thorughput)、响应时间(Response Times)

****PS：****由于详细信息图表有点多，这里我挑几个性能测试过程中比较关键的图表解析！

****Over Time****

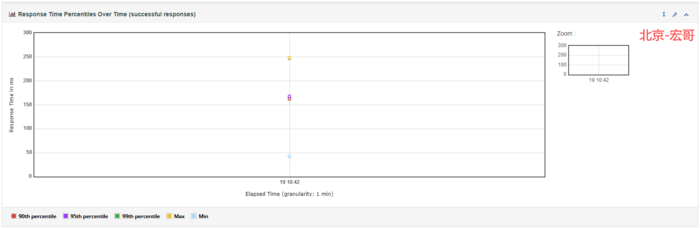
****①、Response Times Over Time（脚本运行期间的响应时间变化趋势图）****

说明：可以根据响应时间和变化和TPS以及模拟的并发数变化，判断性能拐点的范围。

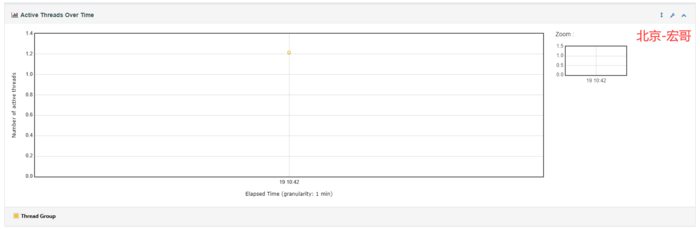


****②、** Response Time Percentiles Over Time (successful responses)**

说明：脚本运行期间成功的请求响应时间百分比分布图，可以理解为聚合报告里面不同%的数据，图形化展示的结果。

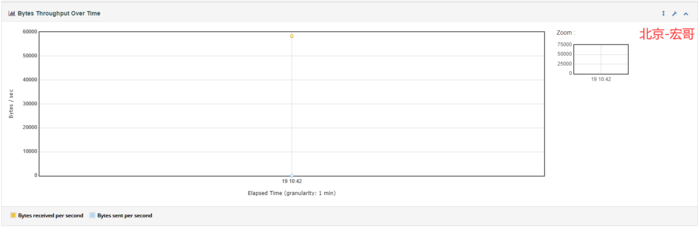


****③、**Active Threads Over Time（脚本运行时间内的活动线程分布）**



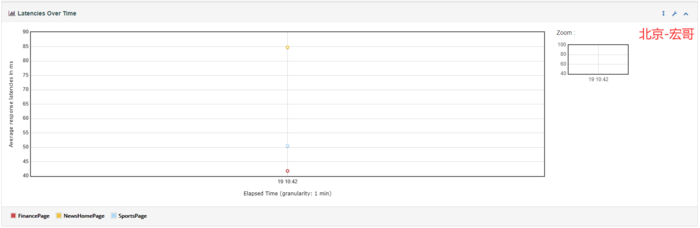
****④、Bytes Throughput Over Time（脚本运行期间的吞吐量变化趋势图）****

说明：在容量规划、可用性测试和大文件上传下载场景中，吞吐量是很重要的一个监控和分析指标。

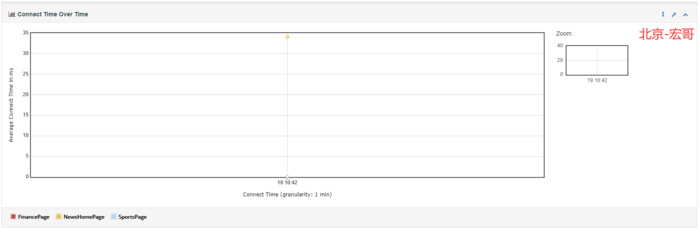


**⑤、Latencies Over Time（脚本运行期间的响应延时变化趋势图）**

说明：在高并发场景或者强业务强数据一致性场景，延时是个很严重的影响因素。

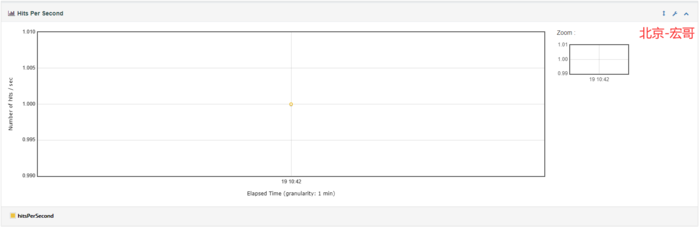


****⑥Connect Time Over Time（脚本运行时间内平均连接时间）****

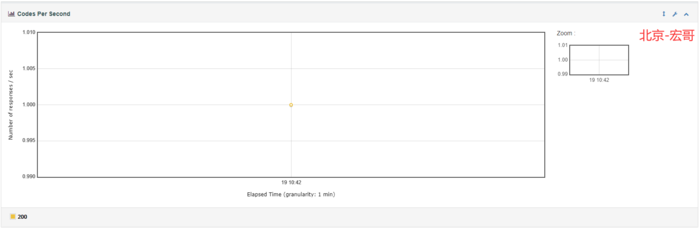


****Throughput****

****①、Hits Per Second (excluding embedded resources) 每秒点击数曲线****

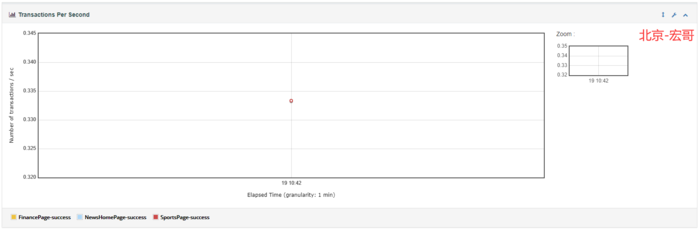


****②、Codes Per Second (excluding embedded resources)每秒状态码分布曲线****

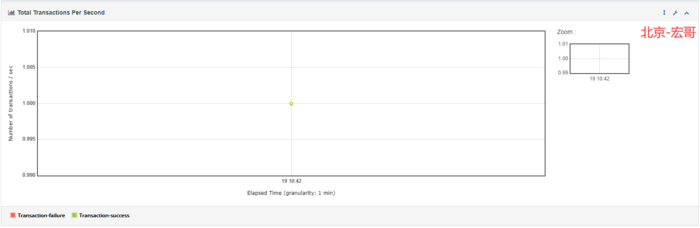
********

****③、Transactions Per Second 每秒事物数曲线****

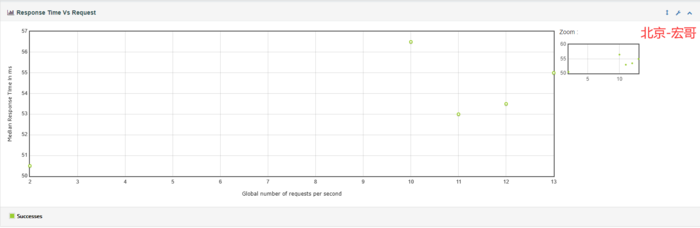
说明：每秒事务数，即TPS，是性能测试中很重要的一个指标，它是用来衡量系统处理能力的一个重要指标。



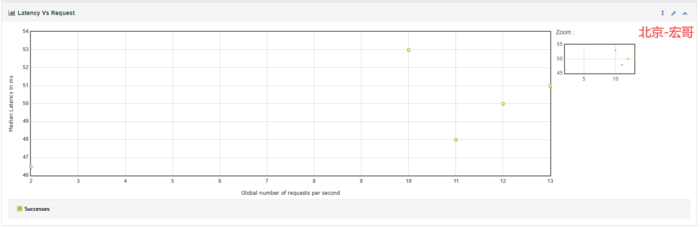
****④、**Total Transactions Per Second每秒通过事务总数**



****⑤、Response Time Vs Request 响应时间中值与每秒请求数关系曲线****



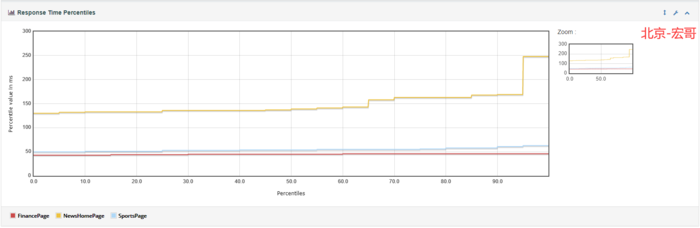
****⑥、Latency Vs Request 延迟时间中值与每秒请求数关系曲线****



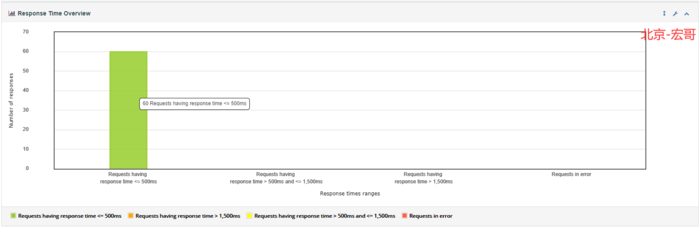
****Response Times****

****①、**Response Time Percentiles（响应时间百分比分布曲线图）**

说明：即响应时间在某个范围内的请求在所有请求数中所占的比率，相比于平均响应时间，这个值更适合用来衡量系统的稳定性。

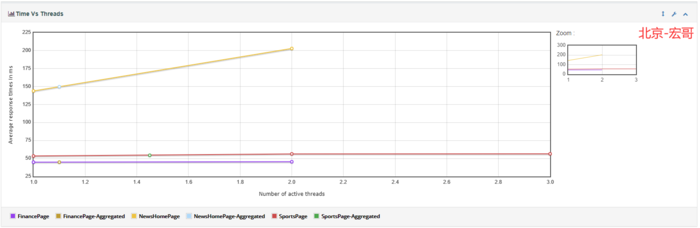


****②、Response Time Overview 响应时间概述柱状图****

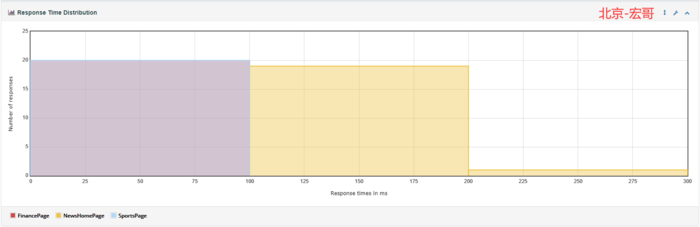


**③、Time Vs Threads（平均响应时间和线程数的对应变化曲线）**

说明：可以通过这个对应的变化曲线来作为确定性能拐点的一个参考值。



****④、Response Time Distribution响应时间分布图****

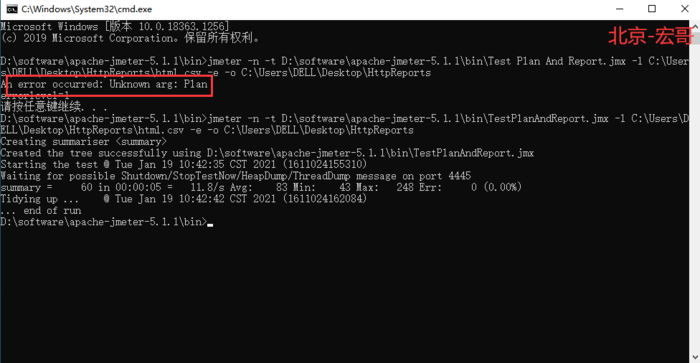
********

****4.3 Customs Graphs****

因为宏哥这里没有设置所以是空的，你可以根据自己需求定制自己的，具体的步骤请参考：https://jmeter.apache.org/usermanual/generating-dashboard.html#top

### 5.小结

1.在输入命令运行的时候，宏哥遇到了找不到测试计划（An error occurred: Unknown arg: Plan），宏哥第一感觉就是测试计划的名字中间有空格（其他可能情况：特殊字符、路径错误、中文等等吧）去掉后就可以运行了。



2.运行过程和制作会遇到4445端口是否允许通过防火墙或者安全性提示的信息，允许通过就可以了。

3.自定义配置

JMeter3.0在bin目录新增了reportgenerator.properties文件保存了所有关于图形化HTML报告生成模块的默认配置，要变更配置，建议不要直接编辑该文件，而是推荐在user.properties中去配置和覆盖。

①、总体配置

总体配置都是以jmeter.reportgenerator.为前缀，如：jmeter.reportgenerator.overall\_granularity=60000

overall\_granularity：定义采样点粒度，默认为60000ms，通常在稳定性以外的测试中，我们可能需要定义更细的粒度，比如1000ms，我们可以在user.properties文件末尾添加如下配置：

# Change this parameter if you want to change the granularity of over time graphs.

jmeter.reportgenerator.overall\_granularity=6000

report\_title:定义报告的标题，我们可能需要将标题定义为实际测试项名称

apdex\_satisfied\_threshold：定义Apdex评估中满意的阈值(单位ms)

apdex\_tolerated\_threshold: 定义Apdex评估中可容忍的阈值

Apdext = (Satisfied Count + Tolerating Count / 2) / Total Samples

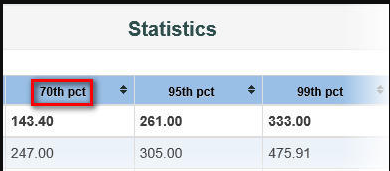
另外，在jmeter.properties中，有关于集合报告中的三个百分位的默认值：

aggregate\_rpt\_pct1 : Defaults to 90

aggregate\_rpt\_pct2 : Defaults to 95

aggregate\_rpt\_pct3 : Defaults to 99

可以在user.properties中对其进行覆盖，如：aggregate\_rpt\_pct1 = 70，效果如下：



②、图表配置

每个图表配置都是以jmeter.reportgenerator.graph.<图表名称>.为前缀。

classname 图表的实现类，如果有自己定制的实现，将该配置的值写为自定义实现类的类名

property.set\_granularity 设置图标的采样点粒度，不配置时默认使用总体配置中的粒度设置

③、输出配置

输出配置都以jmeter.reportgenerator.exporter为前缀。

property.output\_dir 配置默认的报告输出路径。在命令行可以用-o选项来设置特定的路径覆盖该配置。

html.series\_filter 用于过滤展示内容。如在user.properties添加如下配置：

jmeter.reportgenerator.exporter.html.series\_filter=(^Login)(-success|-failure)?

则最后的报告只展示名为Login这个取样器的数据。该配置包含两部分，(-success|-failure)?是Transactions per second图表所依赖的配置。前面部分接受一个正则表达式用来过滤。

④、报告定制

JMeter的HTML报告生成时是使用了固定的模板，模板文件路径为./bin/report-template。

进入该目录可以看到报告的每个页面都有一个.fmkr模板文件，包括index.html.fmkr和./content/pages路径下的几个文件。通过查看这些模板文件，就可以知道怎样去进行报告的轻度定制，比如将一些文本修改得更易懂，或者修改为中文等

⑤、页面的title

默认为"Apache JMeter Dashboard"

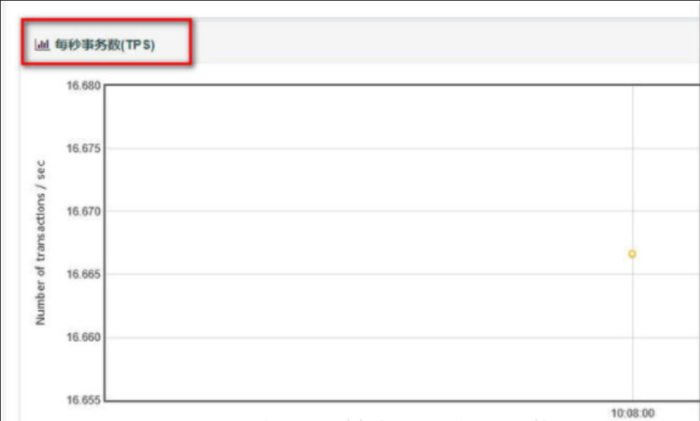
可以由reportgenerator.properties中的jmeter.reportgenerator.report\_title来统一定义，这种方式就是所有页面的title都使用同一个。

也可以直接修改对应的.fmkr文件中的title标签中双引号内的值，如<title>${reportTitle!"想要设置的title"}</title>，这中方式可以为每个页面单独定义title

图表的名称

当前版本下，各图表的名称是直接在模板文件中定义，要修改也是直接修改模板文件中对应元素的值即可

如要修改Transactions Per Second图表的名称，可以直接在./content/pages/Throughput.html.fmkr文件中修改，效果如下图

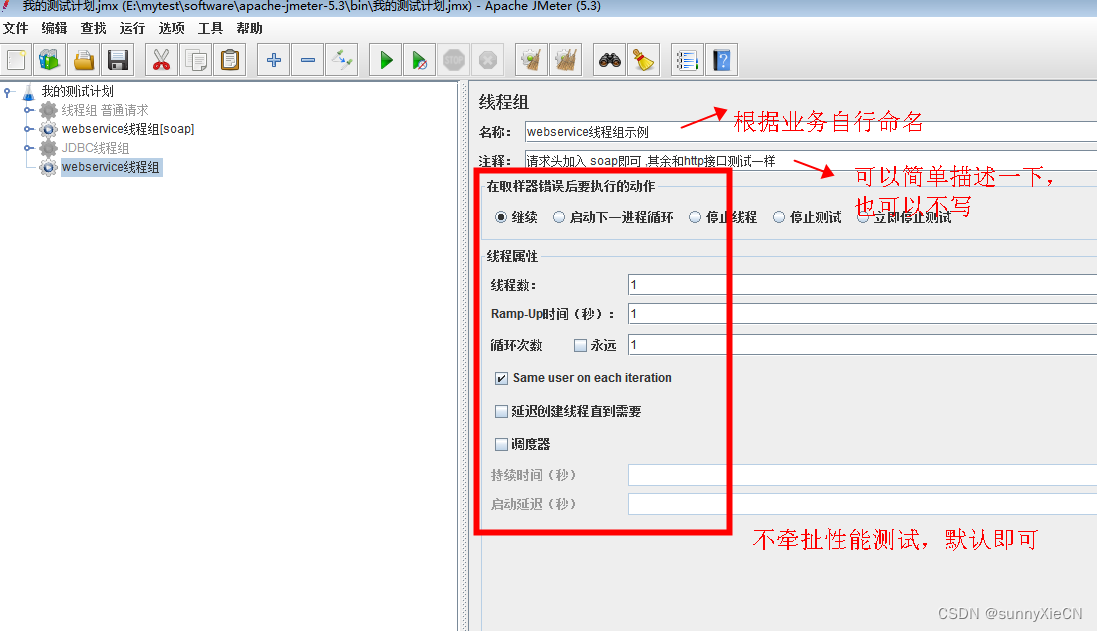


原文链接：https://www.cnblogs.com/du-hong/p/14293243.html

## 10.2 jmeter测试webservice接口

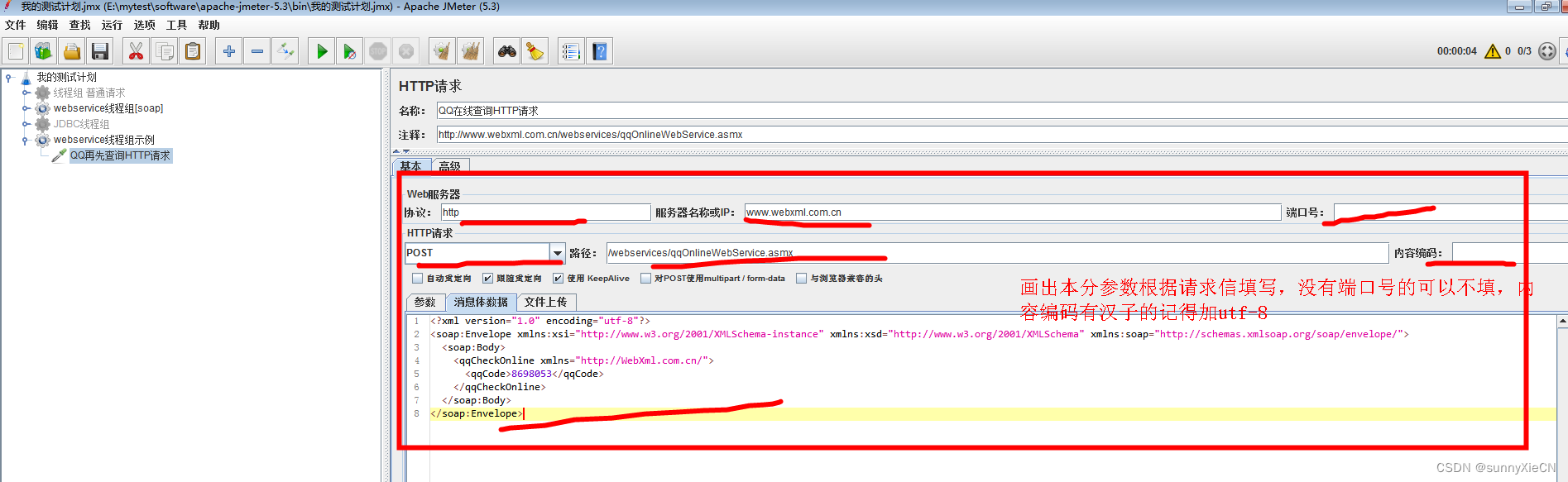
1、创建线程组

选中测试计划——添加——线程（用户）——线程组，创建好如下：



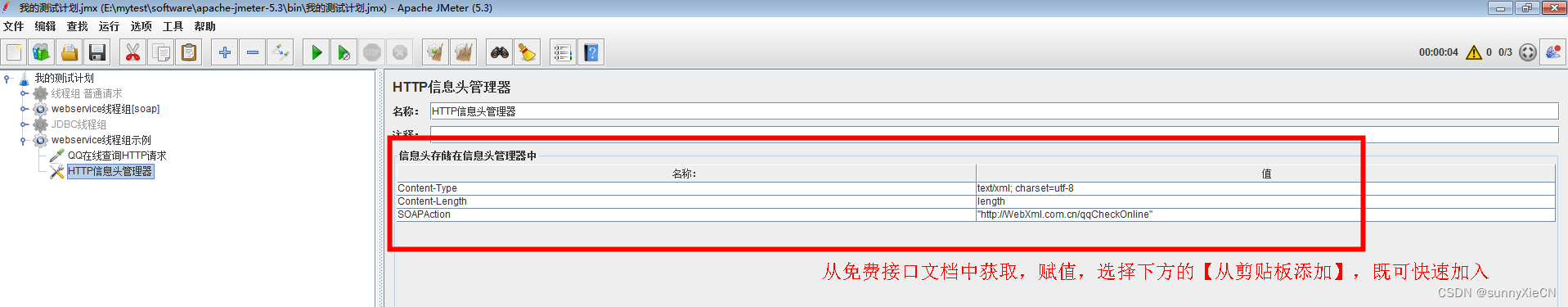
2、创建HTTP请求【body的格式和普通接口有点不一样，其余信息一致】

选中上一步创建的线程组右键—添加—取样器—HTTP请求



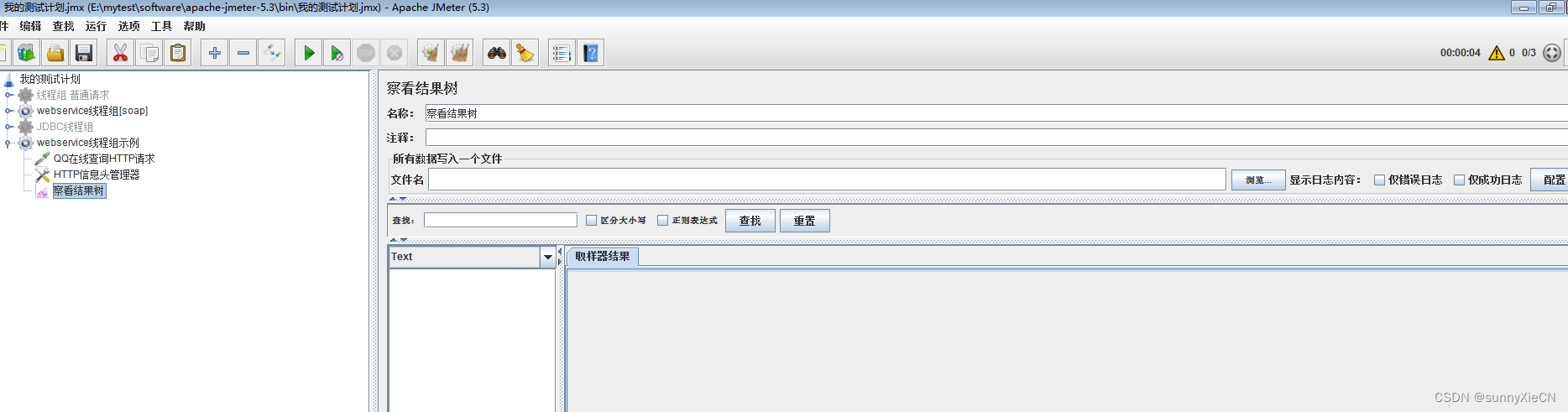
3.创建HTTP信息头管理器【比普通接口多SOAPAction参数，其余和普通接口一致】

选中上一步创建的线程组右键——添加——配置元件——HTTP信息头管理器

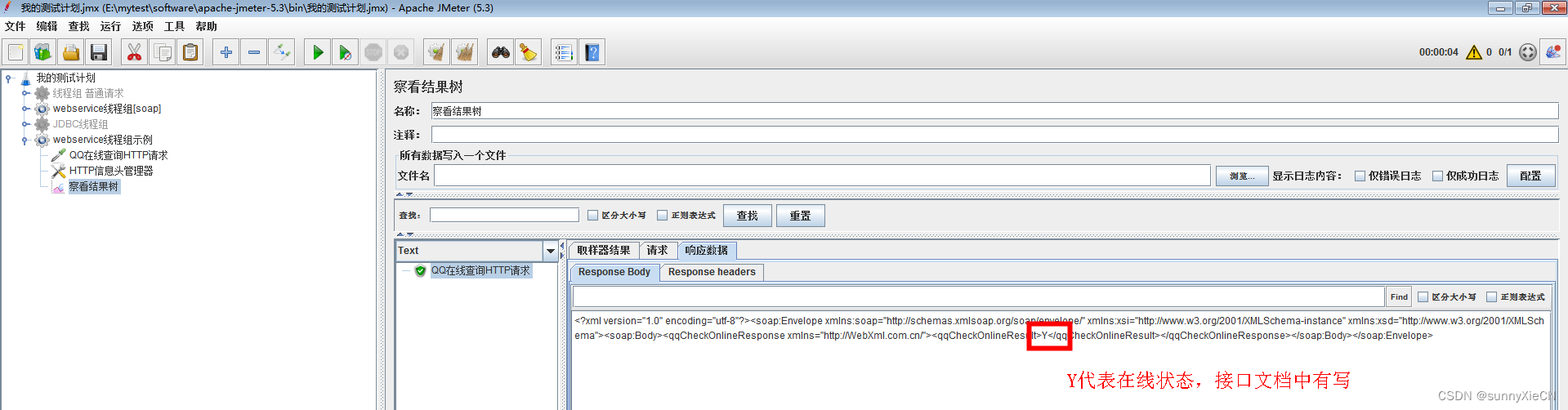


4.创建察看结果树

选中上一步创建的线程组右键——添加——监听器——查看结果树



5.运行（选中单个线程组——右键——运行，或者，直接点击工具栏的绿色箭头运行），返回值是xml格式，不是普通的接口的json格式



到这里，一个webservice接口就调用成功了。

## 10.3 20个免费webservice接口

测试扛把子必须了解的20个免费webservice接口

（1）天气预报Web服务，数据来源于中国气象局

Endpoint

http://www.webxml.com.cn/WebServices/WeatherWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/WeatherWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/WeatherWebService.asmx?wsdl

（2）IP地址来源搜索 WEB 服务（是目前最完整的IP地址数据）

Endpoint

http://www.webxml.com.cn/WebServices/IpAddressSearchWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/IpAddressSearchWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/IpAddressSearchWebService.asmx?wsdl

（3）随机英文、数字和中文简体字 WEB 服务

Endpoint

http://www.webxml.com.cn/WebServices/RandomFontsWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/RandomFontsWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/RandomFontsWebService.asmx?wsdl

（4）中国邮政编码 <-> 地址信息双向查询/搜索 WEB 服务

Endpoint

http://www.webxml.com.cn/WebServices/ChinaZipSearchWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/ChinaZipSearchWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/ChinaZipSearchWebService.asmx?wsdl

（5）验证码图片 WEB 服务 支持中文、字母、数字 图像和多媒体

Endpoint

http://www.webxml.com.cn/WebServices/ValidateCodeWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/ValidateCodeWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/ValidateCodeWebService.asmx?wsdl

（6）Email 电子邮件地址验证 WEB 服务

Endpoint

http://www.webxml.com.cn/WebServices/ValidateEmailWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/ValidateEmailWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/ValidateEmailWebService.asmx?wsdl

（7）中文简体字 <-> 繁体字转换 WEB 服务

Endpoint

http://www.webxml.com.cn/WebServices/TraditionalSimplifiedWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/TraditionalSimplifiedWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/TraditionalSimplifiedWebService.asmx?wsdl

（8）中文 <-> 英文双向翻译 WEB 服务

Endpoint

http://www.webxml.com.cn/WebServices/TranslatorWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/TranslatorWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/TranslatorWebService.asmx?wsdl

（9）火车时刻表 WEB 服务 （第六次提速最新列车时刻表）

Endpoint

http://www.webxml.com.cn/WebServices/TrainTimeWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/TrainTimeWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/TrainTimeWebService.asmx?wsdl

（10）中国股票行情数据 WEB 服务（支持深圳和上海股市的基金、债券和股票）

Endpoint

http://www.webxml.com.cn/WebServices/ChinaStockWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/ChinaStockWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/ChinaStockWebService.asmx?wsdl

（11）即时外汇汇率数据 WEB 服务

Endpoint

http://www.webxml.com.cn/WebServices/ExchangeRateWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/ExchangeRateWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/ExchangeRateWebService.asmx?wsdl

（12）腾讯QQ在线状态 WEB 服务

Endpoint

http://www.webxml.com.cn/webservices/qqOnlineWebService.asmx

Disco

http://www.webxml.com.cn/webservices/qqOnlineWebService.asmx?disco

WSDL

http://www.webxml.com.cn/webservices/qqOnlineWebService.asmx?wsdl

（13）中国电视节目预告（电视节目表） WEB 服务

Endpoint

http://www.webxml.com.cn/webservices/ChinaTVprogramWebService.asmx

Disco

http://www.webxml.com.cn/webservices/ChinaTVprogramWebService.asmx?disco

WSDL

http://www.webxml.com.cn/webservices/ChinaTVprogramWebService.asmx?wsdl

（14）外汇-人民币即时报价 WEB 服务

Endpoint

http://www.webxml.com.cn/WebServices/ForexRmbRateWebService.asmx

Disco

http://www.webxml.com.cn/WebServices/ForexRmbRateWebService.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/ForexRmbRateWebService.asmx?wsdl

（15）中国股票行情分时走势预览缩略图 WEB 服务

Endpoint

http://www.webxml.com.cn/webservices/ChinaStockSmallImageWS.asmx

Disco

http://www.webxml.com.cn/webservices/ChinaStockSmallImageWS.asmx?disco

WSDL

http://www.webxml.com.cn/webservices/ChinaStockSmallImageWS.asmx?wsdl

（16）国内飞机航班时刻表 WEB 服务

Endpoint

http://www.webxml.com.cn/webservices/DomesticAirline.asmx

Disco

http://www.webxml.com.cn/webservices/DomesticAirline.asmx?disco

WSDL

http://www.webxml.com.cn/webservices/DomesticAirline.asmx?wsdl

（17）中国开放式基金数据 WEB 服务

Endpoint

http://www.webxml.com.cn/WebServices/ChinaOpenFundWS.asmx

Disco

http://www.webxml.com.cn/WebServices/ChinaOpenFundWS.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/ChinaOpenFundWS.asmx?wsdl

（18）股票行情数据 WEB 服务（支持香港、深圳、上海基金、债券和股票；支持多股票同时查询）

Endpoint

http://www.webxml.com.cn/WebServices/StockInfoWS.asmx

Disco

http://www.webxml.com.cn/WebServices/StockInfoWS.asmx?disco

WSDL

http://www.webxml.com.cn/WebServices/StockInfoWS.asmx?wsdl

（19）API市场

云市场

https://market.aliyun.com/data

（20）聚合数据

https://www.juhe.cn/

（21）极速数据

<https://www.jisuapi.com/>

## 10.4 Jemeter测试webservice天气预报接口

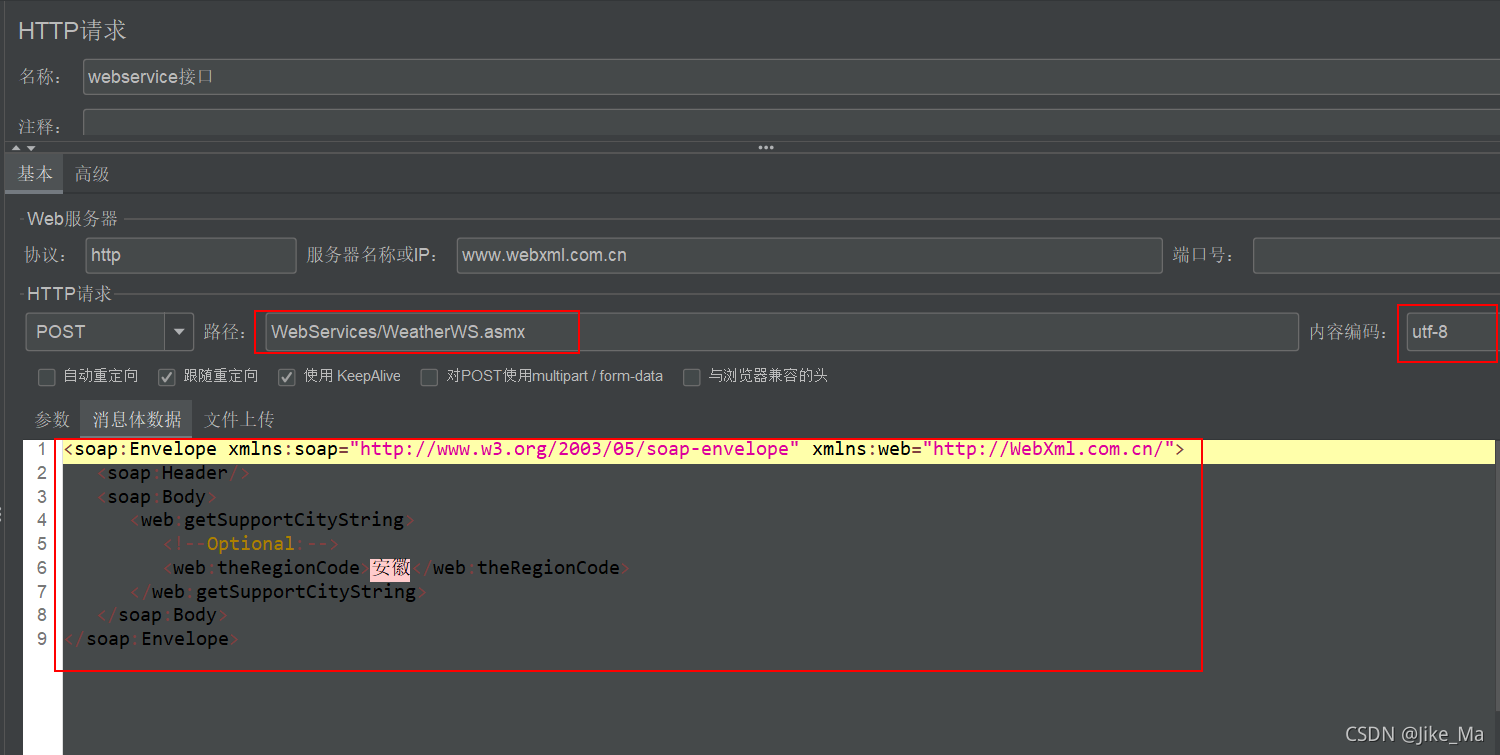
a. 新建Http Request取样器：

1. 路径取到接口地址的?前面；

2. 参数选SoapUI中的请求参数(全部复制并粘贴过来)；

3. 参数中有中文，所以内容编码选 utf-8；

4. 请求方式选POST；



5. 消息体数据设置成如下格式，其中${city}设置成动态参数调用

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:web="http://WebXml.com.cn/">

<soap:Header/>

<soap:Body>

<web:getSupportCityString>

<web:theRegionCode>${city}</web:theRegionCode>

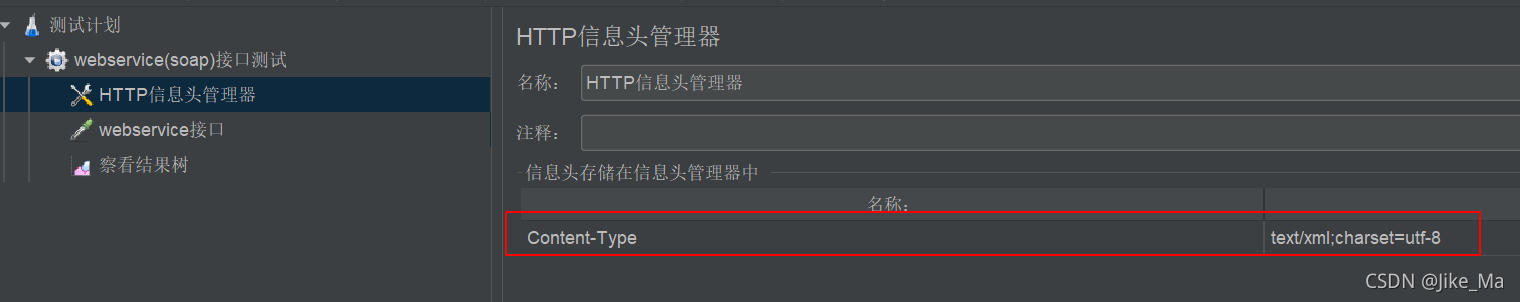
</web:getSupportCityString>

</soap:Body>

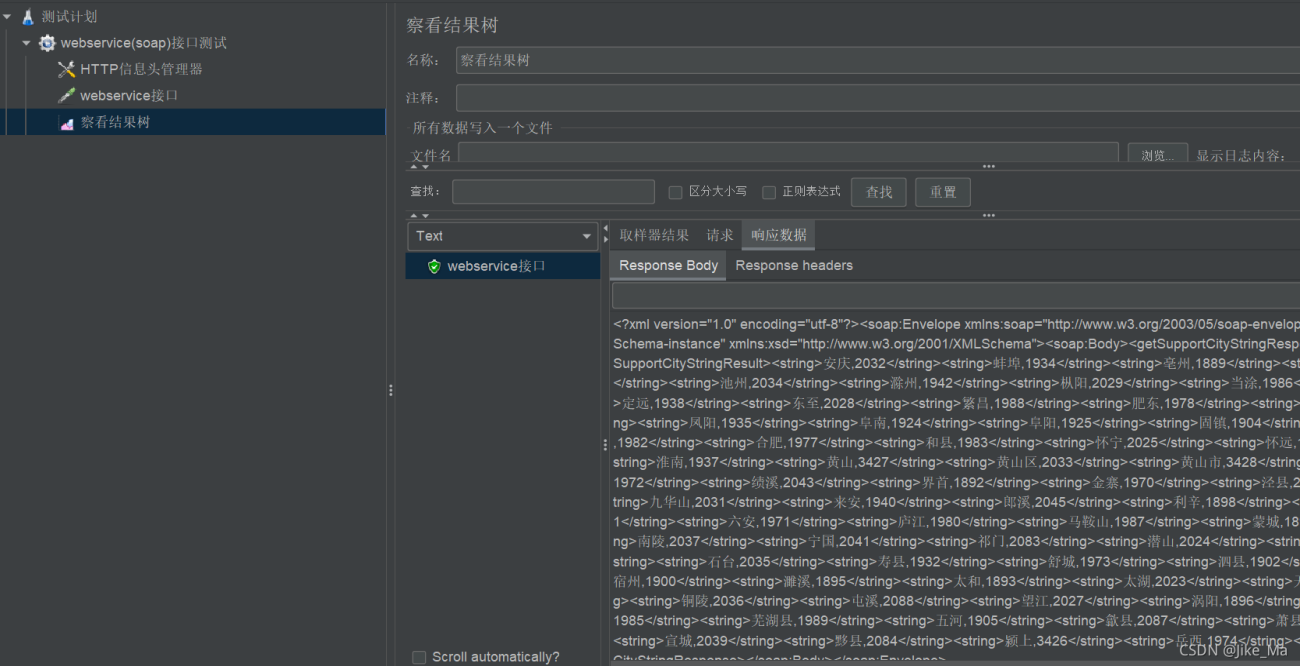
</soap:Envelope>

b. 添加 HTTP 信息头管理器，由于webservice接口不是http

协议的,是soap协议的，所以在请求头中加“Content-Type:text/xml;charset=utf-8”

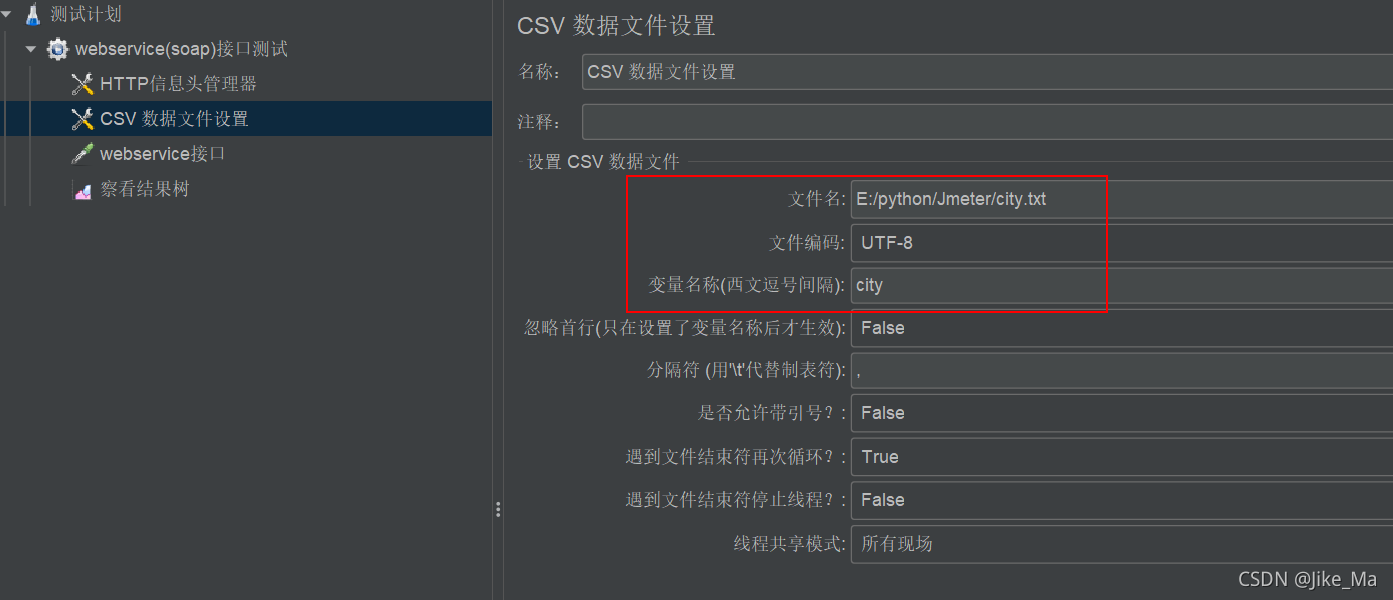


c. 运行请求，查看结果树，成功返回数据



3. Jmeter实现webservice接口参数化

a. 添加 配置元器件：CSV Data set Config，并创建一个 city.txt ,存5个省市的名称

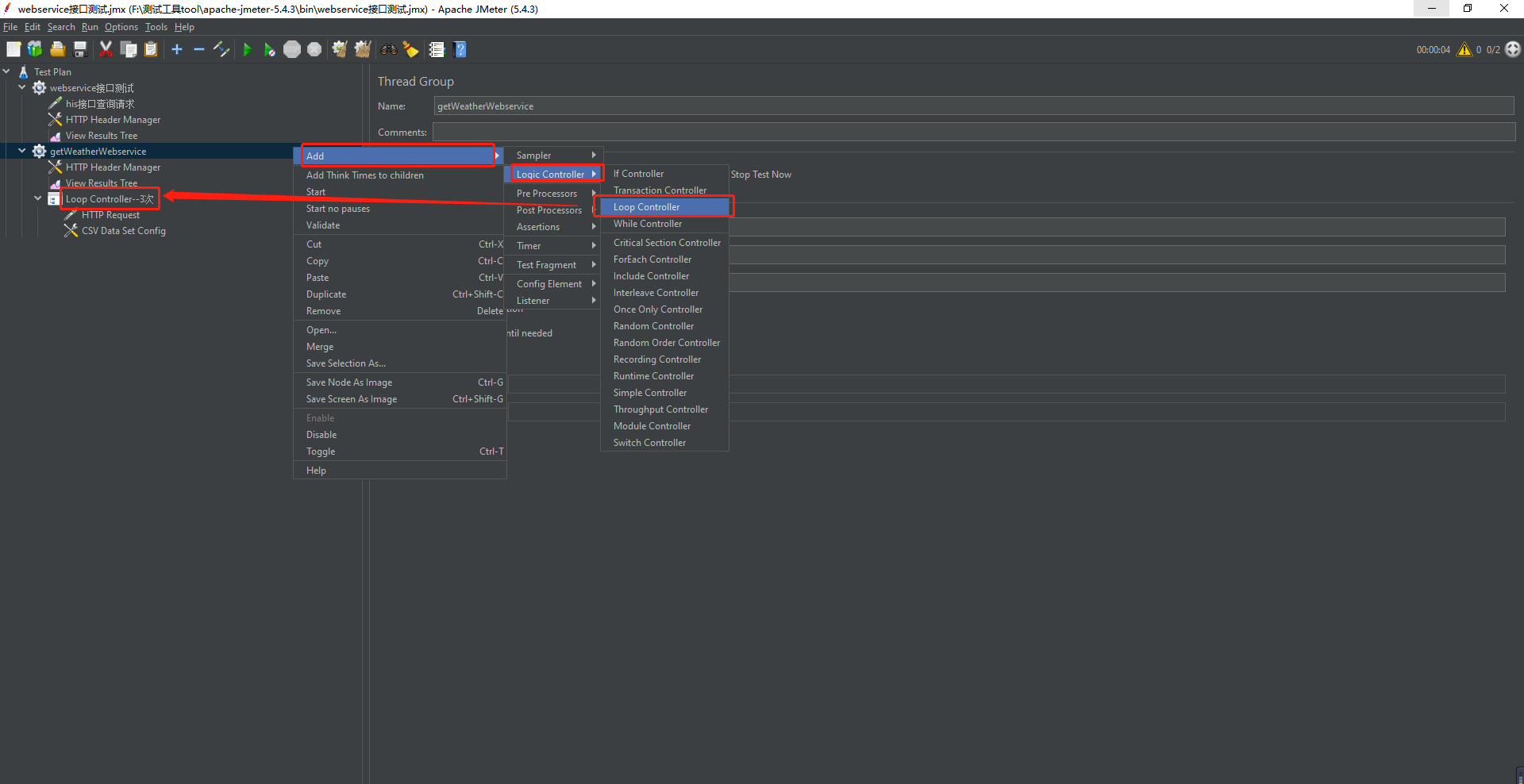


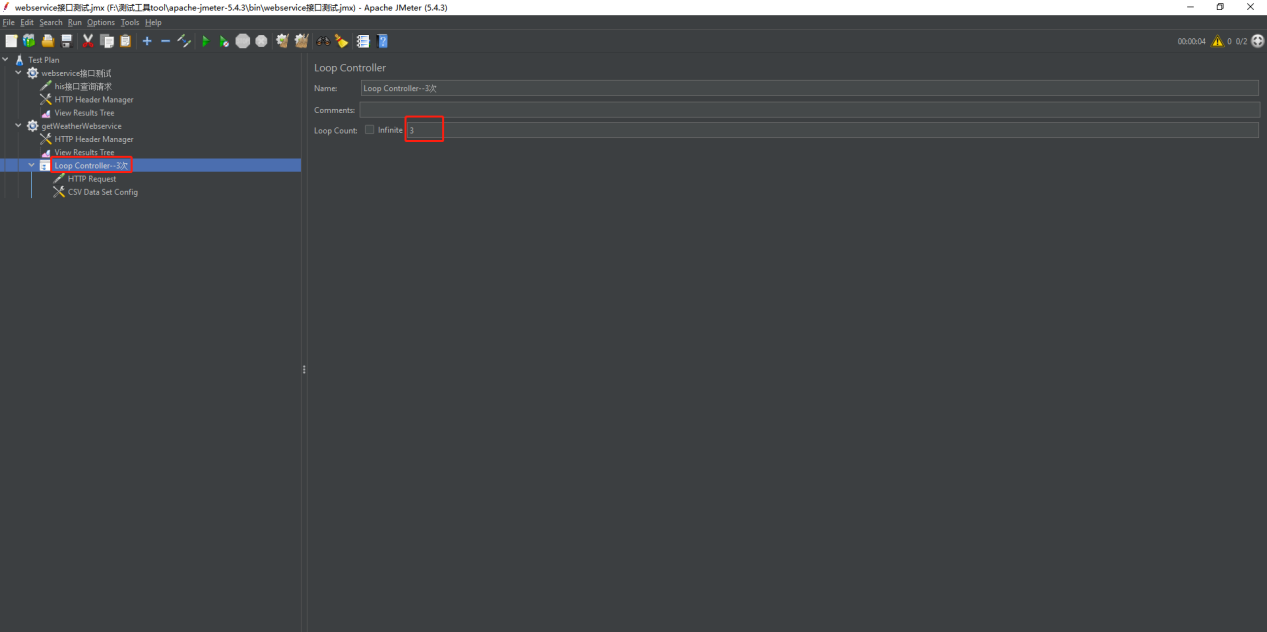
b.将webservice接口中的参数使用变量替代



c.设置一个循环控制器

线程组设置3次循环后，将设置的HTTP Request和CSV Data set Config拖动到循环控制器，即可测试所有参数数据。





# 11 Selenium自动化测试框架搭建（PC端）

## 11.1安装Python

选择Python版本后，进入Python官方网站下载适配机型版本： [http://www.python.org/](https://link.zhihu.com/?target=http://www.python.org/" \t "https://zhuanlan.zhihu.com/p/_blank)

## 11.2安装setuptools和pip

setuptools下载地址： [https://pypi.python.org/pypi/setuptools](https://link.zhihu.com/?target=https://pypi.python.org/pypi/setuptools" \t "https://zhuanlan.zhihu.com/p/_blank)

pip下载地址：[https://pypi.python.org/pypi/pip](https://link.zhihu.com/?target=https://pypi.python.org/pypi/pip" \t "https://zhuanlan.zhihu.com/p/_blank)

解压后打开cmd进入相应目录，执行命令安装：

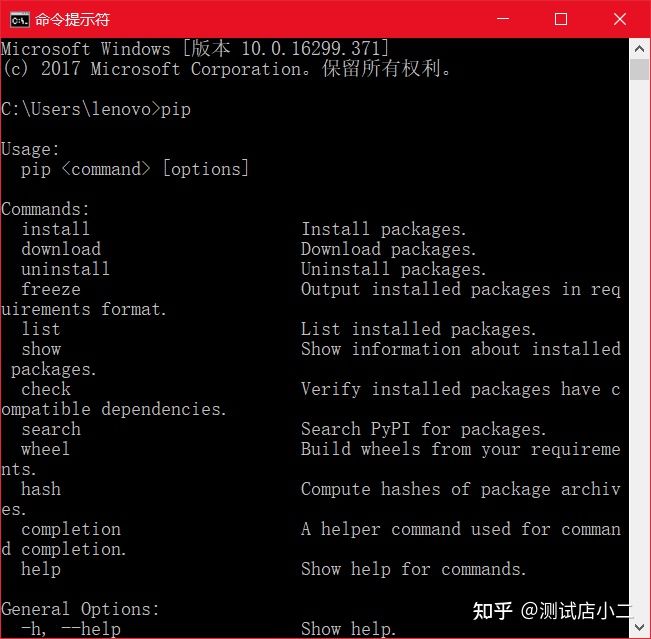
setuptools：

C:\package\setuptools-7.0>python [setup.py](https://link.zhihu.com/?target=http://setup.py" \t "https://zhuanlan.zhihu.com/p/_blank) install

　 pip：C:\package\pip-1.5.6>python [setup.py](https://link.zhihu.com/?target=http://setup.py" \t "https://zhuanlan.zhihu.com/p/_blank) install

安装完执行命令（pip）验证安装结果：C:\Users\lenovo>pip

当弹出如下内容是表示安装成功：



或者进入到python安装目录安装pip：

python -m ensurepip

python -m pip install --upgrade pip

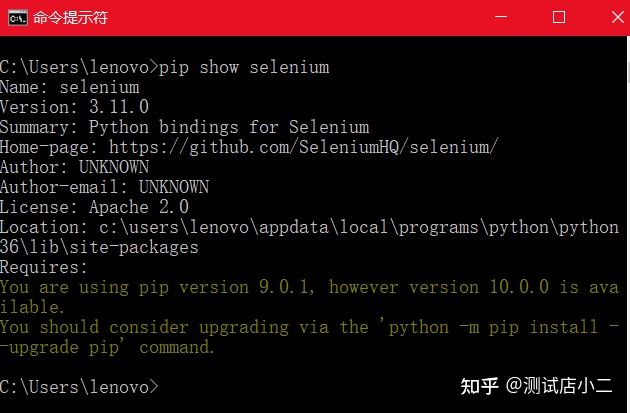
## 11.3安装selenium

（1）执行命令安装：C:\Users\lenovo>pip install selenium

或者指定版本安装：C:\Users\lenovo>pip install selenium==2.48.0

（2）查看当前版本信息：C:\Users\lenovo>pip show selenium

弹出如下信息表示安装成功：



（3）如需卸载，命令：C:\Users\lenovo>pip uninstall selenium

## 11.4安装浏览器驱动webdriver

selenium之chromedriver与chrome版本映射表：[selenium之 chromedriver与chrome版本映射表（更新至v2.46）](https://link.zhihu.com/?target=https://blog.csdn.net/huilan_same/article/details/51896672" \t "https://zhuanlan.zhihu.com/p/_blank)

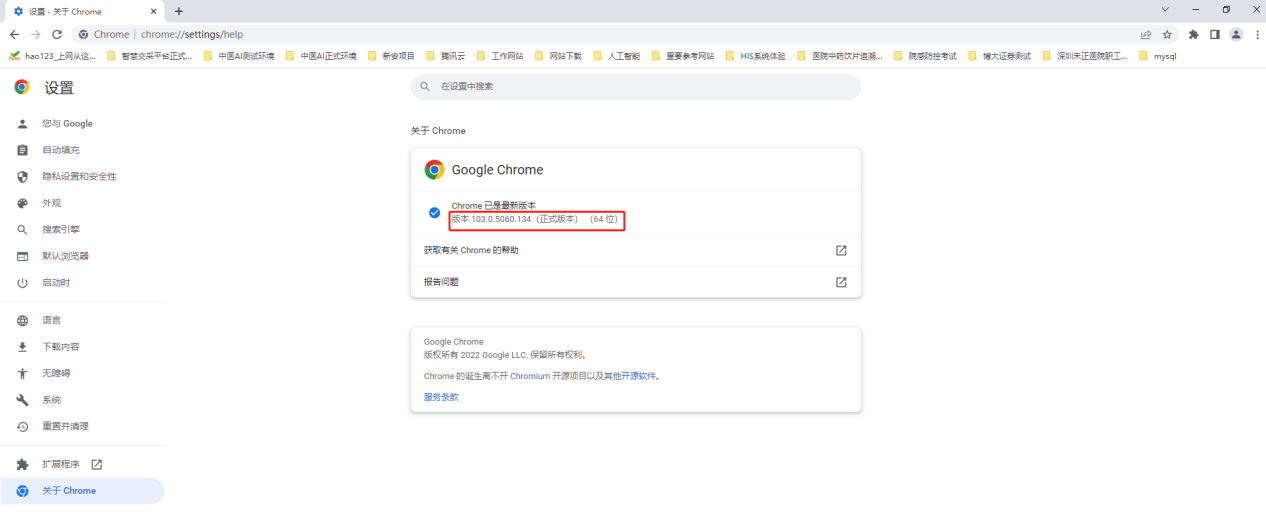
谷歌chromedriver下载地址：[http://chromedriver.storage.googleapis.com/index.html](https://link.zhihu.com/?target=http://chromedriver.storage.googleapis.com/index.html" \t "https://zhuanlan.zhihu.com/p/_blank)

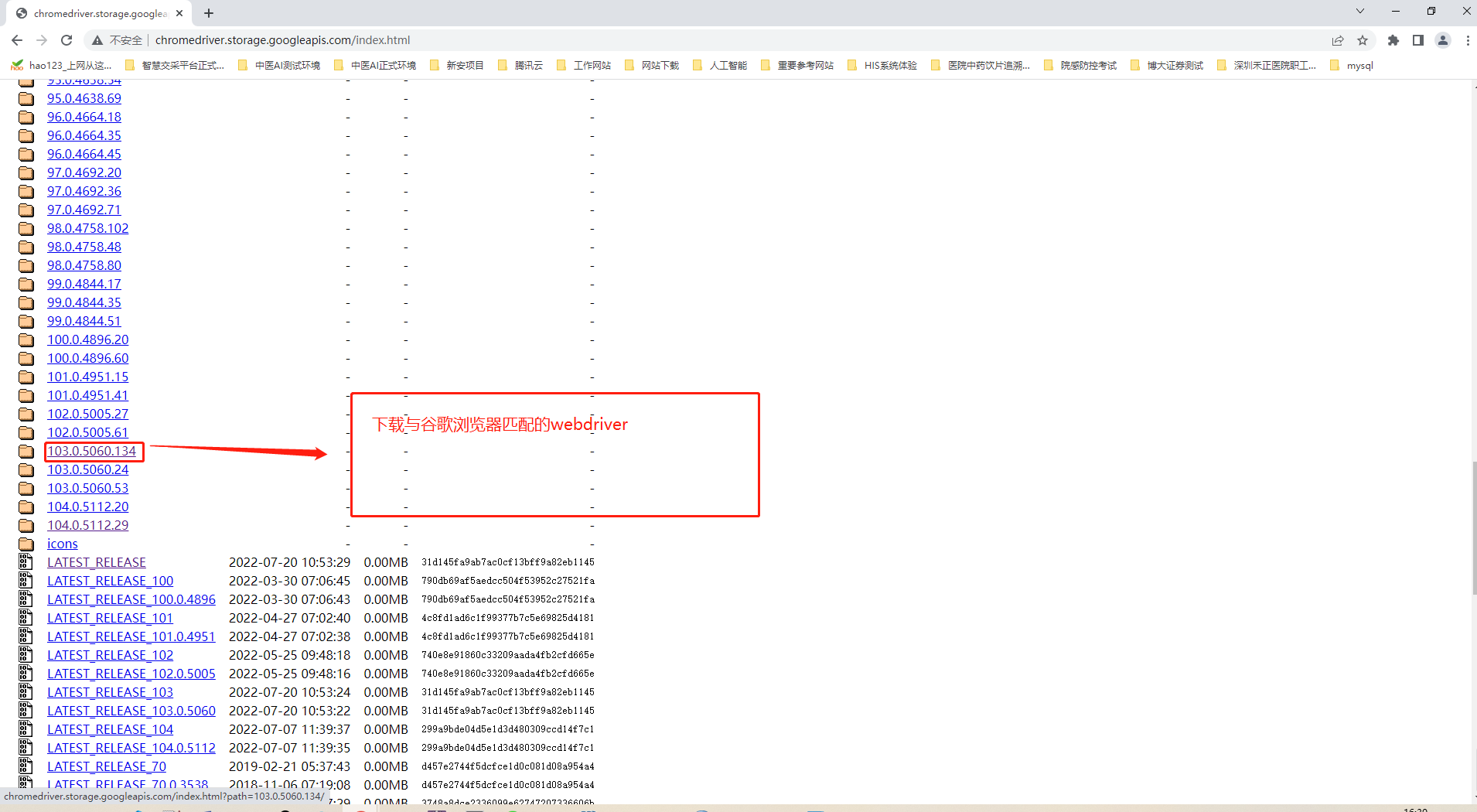
火狐geckodriver下载地址：

https://github.com/mozilla/geckodriver/releases

注意：chromedriver的选择应根据chrome浏览器 版本映射表下载对应版本，否则会报错。

下载完chromedriver后，解压得到.exe文件，把此文件放入Python路径下，跟Python.exe在同一路径，如图：





至此：selenium自动化测试环境搭建完毕，可以尝试编写第一个自动化测试脚本（基于Python）了。

## 11.5第一个自动化测试脚本

coding=utf-8　　　　 　　　　　　　　　　　　　　　　　　　 #防乱码，方便添加中文注释

from selenium import webdriver　　　　　　　　　　　　　　　 #导入selenium的webdriver包

driver= webdriver.Chrome()或者driver= webdriver.Firefox()

/\*option = webdriver.ChromeOptions()

# 防止打印一些无用的日志

option.add\_experimental\_option("excludeSwitches", ['enable-automation', 'enable-logging'])

driver = webdriver.Chrome(chrome\_options=option)

driver.get(r'https://192.168.1.1')]\*/

#把webdriver的chrome对象赋值给变量driver

driver.get(“https://www.baidu.com”)　　　　 　　　　　　　　　 #获得浏览器对象后，通过get（）方法向浏览器发送URL地址

driver.find\_element\_by\_id(“kw”).send\_keys(“selenium”) 　　　　 #通过页面元素定位到百度搜索框并输入selenium关键字搜索

driver.find\_element\_by\_id(“su”).click() 　　　　　　　　　　　　 #通过页面元素定位到“百度一下”tab并发送点击事件

driver.quit() 　　　　 　　　　　　　　　　　　　　　　　　 #关闭浏览器

# 12 Appium+Python自动化环境搭建（移动端）

参考操作链接： https://copyfuture.com/blogs-details/202207120540449328

# 20.常见的二十种软件测试方法详解

**一.单元测试（模块测试）**

单元测试是对软件组成单元进行测试。其目的是检验软件组成单位的正确性。测试对象是：模块。

对模块进行测试，单独的一个模块测试，属于静态测试的一类

* 测试阶段：编码后或者编码前（TDD）
* 测试对象：最小模块
* 测试人员：白盒测试工程师或开发工程师（测源码）
* 测试依据：代码和注释+详细设计文档
* 测试方法：白盒测试（因为要测源码）
* 测试内容：模块接口测试（测试模块里面的参数传递是否正确）、局部数据结构测试（测试变量的作用域范围）、路径测试（if-else 判断必须覆盖所有分支）、错误处理测试、边界测试（ for 循环）

**二.集成测试**

集成测试也称联合测试，将程序模块采用适当的集成策略组装起来，对系统的接口（白盒测试）以及集成后的功能（黑盒测试进行正确性检测的一种测试。集成主要目的是检查软件单位之间的接口是否正确。

* 测试阶段：一般单元测试之后进行
* 测试对象：模块间的接口
* 测试人员：白盒测试工程师或开发工程师
* 测试依据：单元测试的模块+概要设计文档
* 测试方法：黑盒测试与白盒测试相结合
* 测试内容：模块之间数据传输、模块之间功能冲突、模块组装功能正确性、全局数据结构、单模块缺陷对系统的影响

**三.系统测试**

将软件系统看成是一个系统的测试。包括对功能、性能以及软件所运行的软硬件环境进行测试。时间大部分在系统测试执行阶段，包括回归测试和冒烟测试。

* 测试阶段：集成测试通过之后
* 测试对象：整个系统（软、硬件）
* 测试人员：黑盒测试工程师（对功能测试）
* 测试依据：需求规格说明文档
* 测试方法：黑盒测试
* 测试内容：功能、界面、可靠性、易用性、性能、兼容性、安全性等
* 回归测试(Regression Testing)

1. **回归测试**

回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。

有了代码修改后就进行回归测试，根据测试阶段确定回归范围。

在整个软件测试过程中占有很大的工作量比重，软件开发的各个阶段都会进行多次回归测试。随着系统的庞大，回归测试的成本越来越大，通过选择正确的回归测试策略来改进回归测试的效率和有效性是很有意义的。

1. **冒烟测试**

这一术语源自硬件行业。

对一个硬件或硬件组件进行更改或修复后，直接给设备加电。如果没有冒烟，则该组件就通过了测试。也可以理解为该种测试耗时短，仅用一袋烟功夫足够了。

冒烟测试的对象是每一个新编译的需要正式测试的软件版本，目的是确认软件基本功能正常，可以进行后续的正式测试工作。冒烟测试的执行者是版本编译人员。

**概念：对核心主干流程进行测试，如果成功，就认为成功**

作用：判断是否接受测试的标准，若核心主干都走不通，那么直接打下去。

1. **验收测试**

买到新手机，一般会有7天包退，一个月包换，我们会尽量在7天内把手机的所有功能都试一遍。

验收测试是部署软件之前的最后一个测试操作。它是技术测试的最后一个阶段，也称为交付测试。验收测试的目的是确保软件准备就绪，按照项目合同、任务书、双方约定的验收依据文档，向软件购买都展示该软件系统满足原始需求。

* 测试阶段：系统测试通过之后
* 测试对象：整个系统（包括软硬件）。
* 测试人员：主要是最终用户或者需求方。
* 测试依据：用户需求、验收标准
* 测试方法：黑盒测试（对功能进行测试）
* 测试内容：同系统测试(功能…各类文档等)

1. **静态测试（不运行程序本身，测试文档）**

静态测试是指不运行被测程序本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。‘对需求规格说明书、软件设计说明书、源程序做结构分析、流程图分析、符号执行来找错。

1. **动态测试**

动态测试方法是指通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率、正确性和健壮性等性能。大多数软件测试工作都属于动态测试。

1. **手工测试**

就是由人去一个一个的输入测试用例，然后观察结果，和机器测试相对应，属于比较原始但是不可缺少的一个步骤。

总结优缺点：

（1）优点：自动化无法替代探索性测试、发散思维结果的测试。

（2）缺点：执行效率慢，量大易错。

1. **自动化测试**

就是在预设条件下运行系统或应用程序，评估运行结果，预先条件应包括正常条件和异常条件。

简单说自动化测试是把以人为驱动的测试行为转化为机器执行的一种过程。

自动化实施步骤：

1.完成功能测试，版本基本稳定

2.根据项目特性，选择适合项目的自动化工具，并搭建环境

3.提取手工测试的测试用例转化为自动化测试的用例

4.通过工具、代码实现自动化的构造输入，自动检测输出结果是否符合预期

5.生成自动测试报告

6.持续改进，脚本优化。

1. **业务测试**

业务测试是测试人员把系统各个模块串接起来运行、模拟真实用户实际的工作流程，满足用户需求定义的功能来进行测试的过程。

例如查看邮件：

登录网站-输入用户名、密码登录-进入收件箱-查到邮件-点击打开-查阅-关闭邮件-退出邮箱-关闭网站

业务测试关注需求和用户

所有业务流程进行测试，包过主干流程，分支流程，甚至更小的流程

测不同的业务，必须对项目的需求特别了解

1. **界面测试**

界面测试（简称UI测试)，测试用户界面的功能模块的布局是否合理、整体风格是否一致、各个控件的放置位置是否符合客户使用习惯，此外还要测试界面操作便捷性、导航简单易懂性，页面元素的可用性，界面中文字是否正确，命名是否统一，页面是否美观，文字、图片组合是否完美等。

1. **文档测试**

文档的术语：验证术语的正确性

文档的正确性：输入用户名和密码，点击登录，退出系统

文档的完整性：输入用户名和密码

文档的一致性：前后描述不一致，第一种是出现欢迎界面，第二种说出现填写资料页面

文档的易用性：文档的可读性

1. **兼容性测试**

大家经常上网，同一网站在不同的浏览器上表现不一样

WEB测试 ；APP测试

兼容性主要是指软件之间能否很好的运作，会不会有影响软件和硬件之间能否发挥很好的效率工作，会不会影响导致系统的崩溃。

* 平台测试
* 浏览器测试
* 软件本身能否向前或者向后兼容
* 测试软件能否与其它相关的软件兼容
* 数据兼容性测试

最常见的就是浏览器的兼容性测试，不同浏览器在css，js解析上的不同会导致页面的显示不同。

1. **易用性测试**

易用性（Useability）是交互的适应性、功能性和有效性的集中体现。

手机拔打电话功能不放在首页，放在一个目录下边，点击三四次才可以找到拔打电话功能，这个功能好用吗？

1. **性能测试**

检查系统是否满足需求规格说明书中规定的性能。

通常表现在以下几个方面：

* 对资源利用（如内存、处理机周期等）进行的精确度量
* 对执行间隔
* 日志事件（如中断，报错）
* 响应时间
* 吞吐量（TPS）
* 辅助存储区（例如缓冲区、工作区的大小等）
* 处理精度等进行的监测

1. **易用性测试**

易用性（Useability）是交互的适应性、功能性和有效性的集中体现。易用性属于人体工程学的范畴，人体工程学（ergonomics）是一门将日常使用的东西设计为易于使用和实用性强的学科。

手机拔打电话功能不放在首页，放在一个目录下边，点击三四次才可以找到拔打电话功能，这个功能好用吗？

在某些大厂会有专门的部门来进行易用性测试，又叫用户体验测试。

**十八. 安装测试**

测试程序的安装、卸载

典型的是app的安装、卸载

**十九.安全测试**

安全测试是一个相对独立的领域，需要更多的专业知识。例如web的安全测试，需要熟悉各种网络协议

TCP\HTTP，防火墙，CDN，熟悉各种操作系统的漏洞，熟悉路由器等。从软件来说，熟悉各种攻击手段，例如

SQL注入、Xss等。

作为web入门测试，可以IBM的appscan。

**二十.内存泄漏测试**

电脑打开的东西太多，机器反应慢甚至死机，重启之后就好了，过会同样的问题出现

很多软件系统都存在内存泄露的问题，尤其是缺乏自动垃圾回收机制的“非托管”语言编写的程序，例如C、CH、Delphi等。从用户使用的角度来看，内存泄露本身不会造成什 么危害，一般用户可能根本不会感觉到内存泄露的存在。但是内存泄露是会累积的，只要执 行的次数足够多，最终会耗尽所有可用内存，使软件的执行越来越慢，最后停止响应。可以 把这种软件的问题比喻成软件的“慢性病”。

造成内存泄露的原因有很多，最常见的有以下几种。

1.分配完内存之后忘了回收。  
2.程序写法有问题，造成没办法回收。  
3.某些API函数的使用不正确，造成内存泄露。  
4.没有及时释放。

**内存泄漏的检测：**

1、对于不同的程序可以使用不同的方法来进行内存泄露的检查，还可以使用一些专门的工具来进行内存问题的检查，例如MemProof. AQTime、Purify、BundsChecker等。 有些开发工具本身就带有内存问题检查机制．要确保程序员在编写程序和编译程序的时候打开这些功能。

2、通过代码扫描分析工具来检查

# 参考文档：

<https://www.163.com/dy/article/H4PDFDC60552EQVK.html>

<https://blog.csdn.net/weixin_54696666/article/details/113094996>

测试学习链接：<https://ceshiren.com/t/topic/15695>

进阶测试：https://ceshiren.com/t/topic/15695