## **简述通过Python如何做接口回归测试**

当时我们在做接口测试的回归测试的时候使用的是pytharm来测的

我首先把requests包安装上导进去，requests的安装有两种方式，一是在底下的terminal命令窗口输入pip install requests ,另外一种是在file目录下的settings下的project目录下的project interpreter 点击右侧的＋，然后搜索requests，然后进行安装，安装成功之后

我们首先，建一个python网页，然后在控制台输入import requests 导入这个库，然后根据接口文档要求进行写脚本。

当时我们先定义这个接口文档需要的key values 的值，然后requests.get()把url和key values放进去地址和传参，最后给它赋一个值，例如a，然后打印它的响应码和响应报文，这里的响应报文是text,响应码是status\_code,中间用逗号隔开，然后给它附一个值。

打印这个值。接着检查返回结果的请求值，成功是200，错误是404、500等。然后检查返回数据的正确性与格式是否正常。

我们开发那边经常返回的是json,格式。当然也有xml格式的，还有html格式的。

获取的方式有两种一种是get,一种是post,他们两个的区别是如果用post，就需要在参数前面加一个date,就可以了。Get传送的数据量较小，不能大于2kB,post传输数据量较大一般默认为不受限制；get安全性低，post的安全性高；get执行效率比post方法好。传参的时候一定要按照接口文档来写

这个是要区分大小写，否则就只会返回响应码，而不返回报文了，这一点是非常值得注意的。

接口的性能我们也很重视，因为这个直接影响客户的体验，如果半天不返回值，那么我就会果断放弃和开发那边沟通，我们就是这样来相互测接口的。

还可以用for循环这个所有的接口，然后一次性打印出来。For循环的格式呢是：

咱们做接口测试的时候，用for循环的时候呢咱们首先也是先定义这个接口文档需要的key values值，给它们赋一个值，例如是prams,然后咱们接着定义这个url的地址，也给它赋一个值，例如就叫url,后面的接着咱们开始for i in prams，然后requests.get(里面放它的prams和url)。然后也给它附一个值，例如是b ,接着咱们开始打印它的响应报文和响应码。Print(b.text,b.status\_code)这是第一个，第二个的时候咱们还是同样的办法，只是接口文档的url地址换一下就可以了。这里需要注意的是咱们接口文档那的地址需要用咱们本机的IP地址，端口号也是本机tomcat的默认端口号8080.

## **简述通过Python如何做自动化从测试**

**ui自动化**

我说一下我们之前做那个自动化测试的时候，是怎样做的吧：

首先我们做自动化测试的时候呢是用的火狐的浏览器，我们先在浏览器上安装一个xpath,来做一个功能的实现，然后呢我们还需要下载一个火狐的驱动，来实现访问的功能。谷歌、IE是同样的套路，这里需要注意的是，咱们的驱动要和咱们的浏览器是一个版本因为如果不是一个版本的话会报错的，它这个谷歌隔一段时间它就容易自动更新，所以呢这也是我不喜欢用谷歌的原因。

然后咱们接着说项目，我们之前用的是patharm最新版和python3.6。还有版本控制器我们用的是svn.先在svn把项目下载下来，然后让开发把项目给我把项目跑起来然。

接着咱们开始在pycharm里面编写代码。先建一个python网页。在控制台输入pip install selenium 来安装这个库，咱们还可以输入pip show selenium 来查看它的版本。接着咱们在命令窗口输入 from selenium import webdriver 来调入这个库,还有就是要调入一个时间的库 import time 库。这个库呢是为了让它们以后如果页面加载不出来的时候让他休眠用的。因为有的时候如果不让他休眠一下很容易报错。这个是比较值得注意的。

接着咱们启动浏览器，使用的是webdriver.firefox(),给他一个赋一个值比如“browser”，然后咱们继续使用browser.get(),括号里面给你需要访问的域名，这个域名要用引号引起来。接着咱们让它先跑起来看看能不能访问咱们域名里面的内容，如果成功的话，咱们继续下一步，获取的方式呢有by\_name、by\_xpath、by\_id 等等。咱们功能的话是能用name就用name,不能用name，看哪个好用就用哪个。接着在xpath里面写你上面那个火狐浏览器的xpatn点击复制过来的值，然后，登陆注册或者修改的按钮呢咱们后面跟的是.click .然后咱们的文本输入框里面统一跟的是.send keys(),括号里面放你的值。这里需要注意的是这个值需要用‘’，引起来，并且这个引号它遵循内单外双，或者是内双外单的原则。还是上面的那个问题，让它在用的时候呢必须先休眠一下，time.sleep(2)括号里面跟时间。如果不休眠很可能页面加载不出来报错。我们也经常遇到这种问题。然后再有就是如果有弹窗的话用browser.switch\_to\_alter.accept()

最后的话呢是给他一个browser.quit(),直接退出浏览器。

## **总结Linux中的常用命令**

[账户 @ IP地址/本机 当前所在目录] Linux命令

* 1. ifconfig 查看ip地址
  2. Pwd 查看当前的完整目录/路径
  3. ping ip地址 查看两台计算机是否可以相互通信 ctrl+c退出--注意:windows ,Linux 防火墙问题
  4. root账户是超级管理员 密码 跟centos一样
  5. su 切换账户 su root 提示输入密码
  6. ctrl+c 或者 ctrl+z 退出正在执行命令
  7. clear 清屏
  8. 切换目录/文件夹
     1. cd / 切换到根目录
     2. cd ../ 切换到上一层目录
     3. cd usr 切换到 usr 目录
     4. cd usr/java/abc 切换到 usr 下的 java 下的 abc 目录
     5. cd /abc 切换到 根目录下的 abc 目录
  9. 查看文件夹中的内容
     1. ll 显示文件夹的详细内容 权限(可读R可写W 可执行 X) , 创建账号,创建时间 等等
     2. ls 只显示文件夹名称
  10. 创建文件夹
      1. mkdir 文件名称
      2. mkdir yp 创建一个文件夹名称为 yp
      3. mkdir yizu/abc 前提条件:yizu文件夹必须存在
      4. mkdir /usr/aa 在根目录下的 usr 下 创建 aa文件夹
  11. 删除文件夹
      1. rmdir 文件夹名称 rmdir sz 删除sz文件夹
      2. rm -rf 文件名称 -rf 不要询问并且强制删除
  12. 创建文件
      1. touch 文件名.后缀名 touch 1.txt
      2. vi 文件名.后缀名 通过文件编辑器[记事本]间接创建文件
  13. 编辑文件
      1. vi 文件名.后缀名 vim 文件名.后缀名----功能强大
      2. i 或者 insert键 进入编辑模式
      3. esc键 退出编辑模式
      4. :wq w保存 q退出
      5. :wq! !强制 w保存 q退出
      6. :w 保存 不退出
      7. :q! 强制退出 不保存
      8. :w! 强制保存 不退出
  14. 查看文件
      1. 文件名 --适合小文件 cat hi.java
      2. more 文件名 more hi.java 查看文本文件的内容【分屏查看可以使用空格键或enter键或上下键查看】
      3. less 文件名 less 22.txt 查看文件内容 【分屏查看可以使用空格键或enter键或上下键查看】
  15. 移动文件 [剪切] move
      1. mv 目录/源文件 目标文件/新文件名
      2. mv /usr/yp/python/1.txt /usr/yp 将根目录 下 usr 下 yp 下 python 中的1.txt 移动到 根目录 下 yp 中
      3. mv /usr/yp/python/1.txt /usr/yp/111.txt 移动文件同时修改文件名称
  16. 复制文件 copy
      1. cp 目录/源文件 目标文件/新文件名
      2. cp /user/yp/python/hi.java /usr/yp 复制文件
      3. cp /user/yp/python/hi.java /usr/yp/hello.java 复制文件同时修改文件名
  17. 删除文件 rm
      1. rm 文件名 删除文件 提示:询问是否删除
      2. rm -rf 文件夹 一般不提示用户直接删除[包括子文件夹,文件等],-r提示功能
  18. 删除空目录
      1. rmdir 文件夹名称
      2. 注意: 只针对空文件夹
  19. 防火墙相关命令
      1. 启动和关闭 ---重启Linux系统之后生效
      2. 启动 chkconfig iptables on
      3. 关闭 chkconfig iptables off
      4. 启动和关闭 --立即后生效,但是重启Linux失效
      5. 启动 service iptables start
      6. 关闭 service iptables stop
      7. 查看iptables防火墙状态
      8. service iptables status
      9. 重启防火墙
      10. service iptables restart
  20. 重启,关机命令---root超级管理员
      1. 重启
         1. reboot
         2. shutdown -r now 立即重启
         3. shutdown -r 5 5分钟之后重启 ---时间自定义
         4. shutdown -r 10:30 在时间是10:30分时重启 --时间自定义
      2. 关机
         1. init 0
         2. halt
         3. poweroff
         4. shutdown -h now 立即关机
         5. shutdown -h 5 5分钟后关机
  21. 查询Linux系统是32位还是64位
      1. getconf LONG\_BIT
  22. 进程相关命令
      1. ps 命令用户查看当前正在执行的进程
      2. ps -ef 查看所有进程
      3. grep 搜索
         1. ps -ef | grep java 表示查看所有进程里CMD是java的进程信息
         2. ps -aux | grep java 显示java进程的 状态 -aux运行状态
      4. 杀死关闭进程
         1. kill -9 PID -9表示强制并立即停止 --- PID指的是进程ID
         2. kill -9 2662

（23）查看日志

①tail -n 1000 显示最后1000行

②Tail -n +1000 从1000行开始显示，显示1000行以后的

③Head -n 1000 显示前1000行

④Tail -f -n 100 catalina.out liunx 查看日志后100行

⑤Cat filename | tail -n +200 | head -n 200 从200行显示，显示200行 从200~399

## **简述在Linux中如何实现nginx+Tomcat的集群**

当一台Tomcat不能够承载请求数量的时候，有两种解决方案：横向扩展和纵向扩展

横向扩展：通过硬件解决

纵向扩展：通过技术解决

搭nginx+Tomcat的集群就是纵向扩展，通过技术来解决的

把nginx上传到Linux上我们一般都是在usr的文件夹新建一个nginx的文件夹

用wgt++网址路径或者是用ssh

将下载好的nginx安装包上传到Linux，要先安装依赖环境，gcc，pcre，zlib，openssl（哦破思奥） 这四种依赖环境，再将上传的压缩包进行解压

先用./configure进行初始化，再用make进行编译，然后make install进行安装

在 /usr/local/nginx/sbin 目录中启动nginx

在浏览器中输入IP地址就可以看到welcome to nginx

解压四台Tomcat 分别重命名，在conf文件里进行 vi sever.xml 共修改3处端口号，在配置nginx目录下的conf文件，编辑conf文件夹里的nginx.conf 在nginx.conf里找到gzip on 在下一行写配置文件的，访问域名

## **简述性能测试，压力测试，并发测试，负载测试的区别**

性能测试：指在多个用户对服务器或者项目进行访问时，服务器的整体性能检测

压力测试：指多个用户在一个时间段内对服务器访问，不断增加人数直到服务器出现崩溃

并发测试：在同一时间点上同时向服务器发起请求

负载测试：不断增加人数，直到服务器不满足指标为止（cpu、内存、磁盘读写）

要注意的是：压力测试时要通过不断增加用户量，直到服务器崩溃或者宕机而负载测试只是需要测试出服务器在满足一定指标的前提下最大访问量

压力测试： 是测试服务器在某一个时间段中所承受最大访问量，而并发测试则是测试服务器在某一个时间点上所能承受的最大访问量。

所以综上所述：压力值>负载值 压力值>并发值

## **简述如何通过jmeter如何做自动化测试**

用jmeter做自动化的时候，结合badboy一起使用，先用badboy录制脚本，导入到jmeter中，做一个csv参数化，（csv参数化，首先在任意目录下创建一个文本文档，然后把文档当做一个参数化的文档，给它填写一下参数化的内容，并保存，在线程组右键在二级菜单中选择配置原件，中的一个csv的参数化，在file name中填写他的一个绝对路径，在文档编码中填写utf-8 的防止中文乱码，在下一行填写参数化文档的别名，在下一行填写列和列之间的一个分隔符（英文标点符号），然后在请求的列表中用EL表达式进行参数化。）创建断言来验证程序返回的结果，断言就是对测试的结果进行预判，如果结果和预测值匹配则正常。最后添加查看结果树，看返回的值是否正确。

## **简述如何通过jmeter如何性能测试**

用jmeter做性能的时候，结合badboy第三方软件使用，先用badboy录制脚本，导入到jmeter中，然后在线程组中填写线程数，当时我们公司用的是八个监听器，参看结果树，聚合报告，图形结果，每秒点击量，每秒事物树，服务器的吞吐量，服务器的响应时间，服务器硬件变化。

## **简述jmeter如何做并发测试**

1）将启动的线程在执行之前进行集合，达到多人次同事并发访问

2）使得jmeter可以做并发测试

3）首先在线程组上鼠标右键选择‘添加’，在弹出的二级菜单中选择‘定时器’，接着在弹出的三级菜单中选择 ‘synchroninzing timener 在弹出的窗口的 number of simulated  users to group by 属性中填写集合点的人数，在timeout in milesecond属性中设置集合点的超事时间，该时间的单位是毫秒，给时间设置时要通过测试场景的计算得出：集合点超时时间>集合人数\*（线程数/启动时间）

4）当集合点的集合人数满足时，则集合点释放

5）当集合点的超时时间满足时，则集合点释放

6）两个集合点策略满足其中一个即可，不管哪个满足集合点，都会释放掉，因为这两个集合点策略为或者的关系

7）当集合点的人数一致不满足人数时，在时间超出集合点的超时时间时，也会释放掉

## **简述jmeter的断言，脚本关联的操作步骤**

断言

1.断言: 对未知的结果进行预判

2.用处: 使jmeter能够对逻辑性缺陷进行判断

3.操作步骤:

3.1 首先要明确的是常用的断言是响应断言，所以断言一般是加在需要进行判断的响应数据的请求上

3.2 在需要进行断言的请求上鼠标右键，在弹出的下拉列表中选择“断言”

3.3 在二级菜单中选择“响应断言”

3.4 在弹出框中默认选择“响应断言”，然后在模式匹配规则中有:

包括：返回结果包括你指定的内容

匹配：（好像跟Equals查不多，弄不明白有什么区别）

Equals：返回结果与你指定结果一致

Substring：返回结果是指定结果的字串

否：不进行匹配

默认选择“substring”

3.5 点击测试模式下面的“添加”按钮，然后在文本域中填写断言内容

3.6 添加“查看结果树”监听器，运行线程组

3.7 在“查看结果树”的请求列表中查看请求的状态，此时会发现，符合断言信息的请求为绿色，

不符合断言信息的请求则为红色

Jmeter的脚本关联：边界提取器，首先需要在脚本中添加--后置处理器--边界提取器

在边界提取器中填写引用名称、左边界、右边界。通过“添加—后置处理器—正则表达式提取器”来获取数据，且注意正则表达式提取器附在需要获取数据的某个采样器之下，数据一般从“查看结果树“的响应数据获取。

jmeter之脚本关联

1. 脚本关联: 将上一个脚本请求后响应数据作为下一个请求的参数

2. 优点: 可是实现动态化获取响应数据作为参数，使得测试场景更加真实有效

3. 关键点: 1. 后置处理器 2. 正则表达式提取器

4. 操作步骤:

4.1 首先在要获取响应信息的请求上鼠标右键，在添加的二级菜单中选择“后置处理器”

4.2 然后在后置处理器的二级菜单中选择“正则表达式提取器”

4.3 在正则表达式提取器的窗口中设置正则表达式内容

4.4 首先，在引用名称属性中自定义引用名称

4.5 然后在正则表达式属性中填写获取内容时的正则表达式。

## **简述jmeter如何做性能分析**

一、图形结果: 图形结果监听器中显示的是服务器的偏离和样本数目

1.样本数目: 所有线程执行完所有请求的个数，

计算公式:样本数目 = 线程数 \* 请求个数

2.最新样本: 最后一个线程执行最后一步请求所花费的时间。（单位: 毫秒）

3.平均: 所有线程执行完所有请求后的平均时间（单位:毫秒）

4.偏离: 该值反应的是服务器的稳定性，当该值越大时，表示服务器的性能越

不稳定，反之，则比较稳定

5.吞吐量: 单位时间内服务器对于客户端的数据的发送和接收的数据量

6.中值: 将所有请求的时间按照一定顺序进行排序，然后取出最中间的时间，

该时间是平均时间。

二、查看结果树: 每个请求都为绿色说明请求执行成功。取样器里的请求响应码为200，

说明服务器响应成功。通过查看结果树中每个请求的相应数据得出，

服务器成功响应一个请求。

三、聚合报告: 展示的是服务器响应的时间和容错率以及服务器吞吐量（接收和发送的数据量

1）Label：脚本代码或者http请求

2）Samples：当前请求执行的人数

3）Average: 服务器平均响应时间（单位:毫秒）

4）Median: 服务器响应时间的中值，与图形结果中的中值意义一致（单位:毫秒）

5)90%Line: 百分之九十用户的响应时间小于该时间

6)Min: 服务器最小响应时间（单位:毫秒）

7)Max: 服务器最大响应时间（单位:毫秒）

8)Error%: 错误的请求占总请求的百分比

9)Throughput: 服务器的吞吐量（单位: 个数/sec）

10)Received: 服务器接收的数据量

11)Sent: 服务器发送的数据量

四、Bytes Throughput Over Time(单位时间内服务器吞吐量变化趋势图)：服务器的吞吐量随着点击 量变化而变化，变化关系是呈正比关系

五、Hits per second（每秒点击量变化）：服务器的每秒点击量随着点击量变化而变化，变化关系是呈正比关系

六. PerfMon Metrics collector（服务器硬件变化）：根据图形结果查看服务器是否稳定

七、 Response times over time（服务器单位时间内响应时间变化）：查看通过响应时间的最高

请求，是否小于基准测试，来分析服务器的稳定性。

八、Transactions per Second（每秒通过的事务数）：每秒事务数的变化趋势和每秒点击量的

变化趋势成正比关系

1、通过查看图形结果的偏离

2、查看结果树的请求是否执行成功

3、聚合报告中的平均响应时间、最大值、容错率，百分之九十的响应时间

4、查看服务器吞吐量、每秒点击量、每秒事务数是否是正比关系

5、Cpu的占用率