**Python+Appium自动化真机启动app**

网站:

<https://blog.csdn.net/asaasa1/article/details/109332602?utm_medium=distribute.pc_relevant.none-task-blog-title-6&spm=1001.2101.3001.4242>

网站:https://www.cnblogs.com/lucky0425/p/10955252.html

1.连接手机(测试机)

通过USB线连接Android手机和电脑，打开手机的开发者模式，开启USB调试模式（方法自行百度）

打开cmd输入命令adb devices，显示连接手机的udid即为连接成功

输入adb devices后如果List of devices attached下为空，说明手机连接电脑没有成功。可做如下尝试：

1.拔掉手机重新连接

2.更换USB线

3.重启adb。先杀死adb进程adb kill-server，再重启adb start-server

4.检查adb版本，版本太低则更新

2.获取测试手机及被测app的相应信息

1.platformName，声明被测手机系统是ios还是Android

2.platformVersion，即测试手机的Android版本号。可在手机设置里面查看，也可以通过命令adb shell getpropro.build.version.release查看

3.deviceName，即测试手机的名称。可在手机设置里面查看，也可以通过命令adb devices -l 查看，其中model即为连接的测试机的名称。

4.appPackage，即被测app的包名。

5.appActivity，即被测app的launcherActivity。通过命令adb shell dumpsys activity | findstr “mResume”查看（先在手机上打开app，再输入命令）。以今日头条app为例。  
6.如果手机系统在Android 8.1以下，则需使用命令adb shell dumpsys activity | findstr “mFocus”查看。

Appium启动app

1.电脑打开appium，点击“Start Server”  
2.出现如下页面，再点击“Start Inspector Session”按钮  
3.输入第二步获取的配置内容，点击“Start Session”按钮（可先点击3所指的按钮保存，下次直接选择即可）。  
4.此时，如果测试机是第一次连接appium的话，手机上会提示安装两个应用：Appium settings、io.appium.uiautomator2.server，容许安装，且必须安装成功。

再次点击上图中4所指的“Start Session”按钮，我们会发现手机上会自动启动今日头条app，且appium会出现如下窗口。

1. 至此，我们就成功地通过appium启动了测试机上的app。我们可以在这个页面查看app当前页面的元素的属性，如下图，3所指便是弹窗【我知道了】按钮的元素属性，用于后面自动化测试过程中元素定位。

**Windows里搭建redis**

1. 先下载redis包
2. 解压redis包(右击)
3. cd切换到redis里
4. 输入redis-server.exe redis.windows.conf

查看客户端

1. cd 切换到redis里
2. 输入命令redis-cli.exe -h 127.0.0.1 -p 6379
3. Key \* 查看所有的星

get name set name “zhangsan” flushdb

**Liux下搭建**

1. 先下载redis包



1. tar zxf 压缩包
2. cd redis
3. make
4. cd src
5. ./redis-server ../redis启动

7.输入命令redis-cli.exe -h 127.0.0.1 -p 6379

8.Key \* 查看所有的星

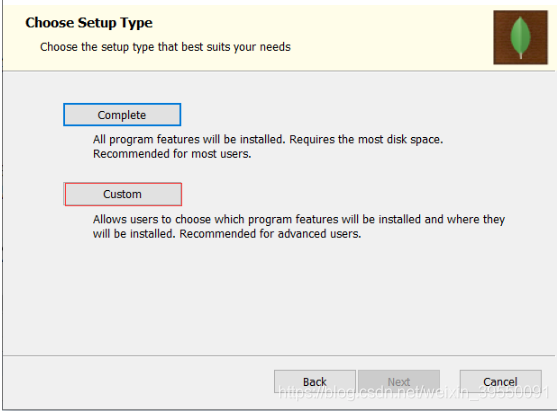
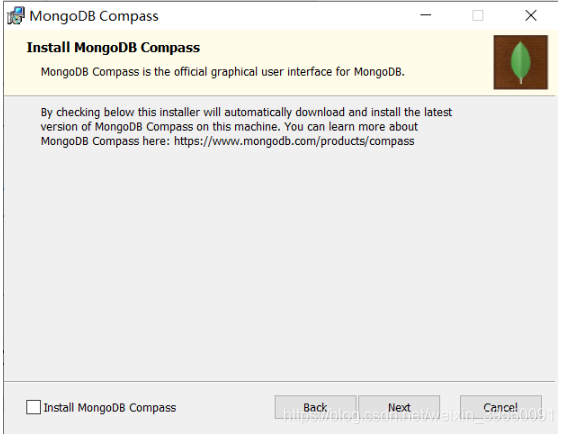
# **mongodb安装教程**

# **下载**

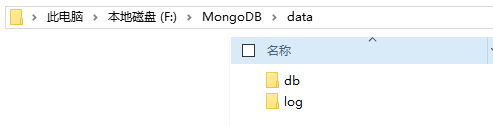
给出官网链接：[MongoDB官网下载链接](https://www.mongodb.com/download-center/community)

https://blog.csdn.net/weixin\_39550091/article/details/105618482?utm\_medium=distribute.pc\_relevant.none-task-blog-title-7&spm=1001.2101.3001.4242

**安装**

找到下载的文件，然后双击。会进行到如下图的情况（进行安装路径的配置）。建议选择（Custom）。我这里选择F盘。  
  
继续点下一步，会遇到如下选项。不要勾选【Install MongoDB Compass】这个选项。否则安非常慢。  
  
然后进行下一步，直到完成安装。

# **配置**

1. 首先在data的文件夹下建立两个空文件夹：db文件夹和log文件夹。如下图：  
   
2. 然后在log文件夹下新建mongo.log文件，如图：  
   
3. 添加环境变量。把F:\MongoDB\bin添加到环境变量中的【系统变量】中。
4. 打开cmd，运行mongo。如图：  
   IMG_260
5. 启动服务。运行命令：

mongod --dbpath E:\MongoDB\data\db

如图：  
IMG_261  
温馨提示：提前创建好db文件夹，才可以启动成功。

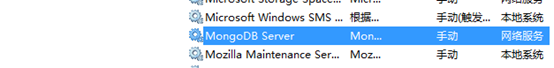
1. 打开浏览器，输入

127.0.0.1:27017

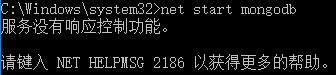
可以得到如图：  
IMG_262  
表示启动成功。

1. 如果每次都这样启动服务，未免太麻烦，我们可以设置成开机自动启动，也可以设置为手动启动。我个人倾向于手动启动。  
   以【管理员身份】打开cmd。输入如下命令：

mongod --dbpath “F:\MongoDB\data\db” --logpath “F:\MongoDB\data\log\mongo.log” -install -serviceName “MongoDB”

如图：  
IMG_263  
如果没有报错，说明服务已经添加到服务里了。可以【Win+R】键然后输入【services.msc】命令，查看服务。如图：  
IMG_264  
然后可以改为【手动】，如图：  
  
继续在cmd窗口输入(仍然是以管理员身份打开的cmd)：

net start mongodb

如图：  
  
遇到这种情况，不要慌。跟着步骤照做，保准能用。

1. 同样在以【管理员身份】打开的cmd中，运行以下命令删除服务

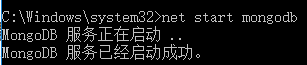
sc delete mongodb

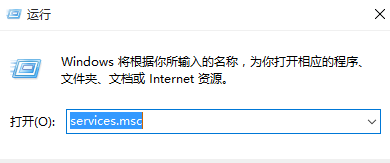
IMG_267  
2. 再运行一次配置命令

mongod --dbpath “F:\MongoDB\data\db” --logpath “F:\MongoDB\data\log\mongo.log” -install -serviceName “MongoDB”

再次运行

net start mongodb

如图：  


1. 查看服务状态。如图：  
     
   IMG_270
2. 停止服务。命令

net stop mongodb

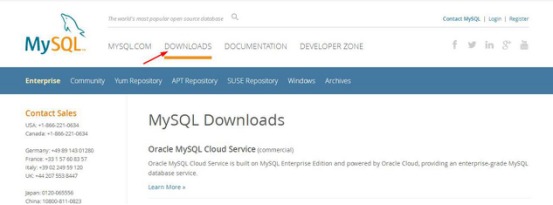
如图：  
IMG_271  
IMG_272  
配置完成。【如果想要再次打开使用MongoDB的话，同样用管理员身份打开cmd，然后输入命令：net start mongodb】

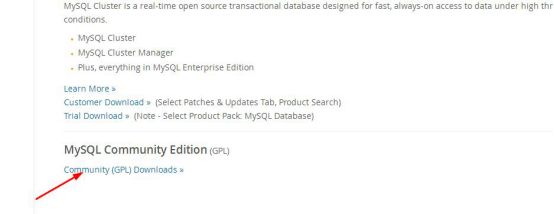
**Linux里MySQL的搭建**

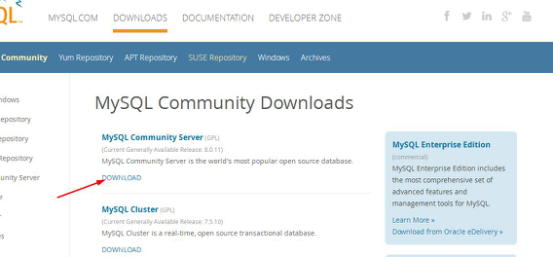
1、首先关闭linux的防火墙，执行命令

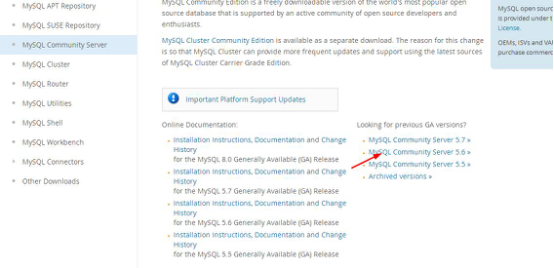
chkconfig iptables off

2、从mysql官网上下载自己适合的mysql版本https://dev.mysql.com/downloads/mysql/5.6.html#downloads，进入mysql官网，依次点击











3、下载后的mysql文件

mysql-5.6.40-linux-glibc2.12-i686.tar.gz

 将下载好的mysql压缩文件放置在linux的/usr/local文件夹下，解压该压缩文件

tar -zxvf mysql-5.6.40-linux-glibc2.12-i686.tar.gz

将解压后的文件重命名为mysql

mv mysql-5.6.40-linux-glibc2.12-i686 mysql

4、创建mysql用户组及用户

groupadd mysql
useradd -r -g mysql mysql

5、进入到mysql目录，执行添加MySQL配置的操作

cp support-files/my-medium.cnf /etc/my.cnf
或：
cp support-files/my-default.cnf /etc/my.cnf

是否覆盖？按y 回车

6、编辑/etc/my.cnf文件；

vi /etc/my.cnf

在my.cnf文件中添加或者修改相关配置，更改完成后保存退出

1 # For advice on how to change settings please see
2 # http://dev.mysql.com/doc/refman/5.6/en/server-configuration-defaults.html
3 # \*\*\* DO NOT EDIT THIS FILE. It's a template which will be copied to the
4 # \*\*\* default location during install, and will be replaced if you
5 # \*\*\* upgrade to a newer version of MySQL.
6
7 [mysqld]
8
9 # Remove leading # and set to the amount of RAM for the most important data
10 # cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
11 # innodb\_buffer\_pool\_size = 128M
12
13 # Remove leading # to turn on a very important data integrity option: logging
14 # changes to the binary log between backups.
15 # log\_bin
16
17 # These are commonly set, remove the # and set as required.
18 basedir = /usr/local/mysql
19 datadir = /usr/local/mysql/data
20 port = 3306
21 # server\_id = .....
22 socket = /tmp/mysql.sock
23 character-set-server = utf8
24 skip-name-resolve
25 log-err = /usr/local/mysql/data/error.log
26 pid-file = /usr/local/mysql/data/mysql.pid
27
28 # Remove leading # to set options mainly useful for reporting servers.
29 # The server defaults are faster for transactions and fast SELECTs.
30 # Adjust sizes as needed, experiment to find the optimal values.
31 # join\_buffer\_size = 128M
32 # sort\_buffer\_size = 2M
33 # read\_rnd\_buffer\_size = 2M
34
35 sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES

7、在mysql当前目录下设定目录的访问权限（注意后面的小点，表示当前目录）

chown -R mysql .
chgrp -R mysql .
scripts/mysql\_install\_db --user=mysql

问题一：-bash: scripts//mysql\_install\_db: /usr/bin/perl: 坏的解释器: 没有那个文件或目录

解决方案：yum -y install perl perl-devel

问题二：FATAL ERROR: please install the following Perl modules before executing scripts//mysql\_install\_db:

Data::Dumper

解决方案：yum -y install autoconf

chown -R root .
chown -R mysql data

8、初始化数据（在mysql/bin或者mysql/scripts下有个 mysql\_install\_db 可执行文件初始化数据库），进入mysql/bin或者mysql/scripts目录下，执行下面命令

./mysql\_install\_db --verbose --user=root --defaults-file=/etc/my.cnf --datadir=/usr/local/mysql/data --basedir=/usr/local/mysql --pid-file=/usr/local/mysql/data/mysql.pid --tmpdir=/tmp

9、启动mysql，进入/usr/local/mysql/bin目录，执行下面命令

./mysqld\_safe --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=root &

注意，如果光标停留在屏幕上，表示启动成功，需要我们先关闭shell终端，再开启一个新的shell终端，不要执行退出操作。如果出现 mysql ended这样的语句，表示Mysql没有正常启动，你可以到log中查找问题.

10、设置开机启动，新开启shell中断后，进入mysql目录，执行下面命令

cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
cp /usr/local/mysql/support-files/mysql.server /etc/rc.d/init.d/mysql

chmod 700 /etc/init.d/mysql
chkconfig --add mysqld
chkconfig --level 2345 mysqld on
chown mysql:mysql -R /usr/local/mysql/

重启linux

reboot

查看mysql状态

service mysqld status

1. 添加远程访问权限

（1）、添加mysql命令

ln -s /usr/local/mysql/bin/mysql /usr/bin （mysql的安装路径）

 （2）、更改访问权限

登录mysql,执行下面命令
mysql -uroot -p
密码为空直接回车,运行以下两条命令

切换数据库

use mysql;

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'root' with grant option;

Flush privileges;

 退出mysql

exit

重启linux,就完成了

Reboot

**Postman和jmeter的区别**

**1.创建接口用例集（没区别）**

Postman是Collections，Jmeter是线程组，没什么区别。

**2.步骤的实现（有区别）**

Postman和jmeter都是创建http请求  
区别1：postman请求的请求URL是一个整体，jmeter分成了4个部分（协议、主机、端口、路径）  
区别2：postman可以在请求中直接填写请求头信息，  jmeter需要通过添加http请求头管理器添加请求头  
区别3：对于cookie，postman可以对cookie做管理，但是jmeter只需添加http cookie管理器即可完成cookie的处理，并且是自动处理cookie信息，所以jmeter的cookie管理更简单

Postman在pre-request script可以添加前置请求，获取响应数据，比较容易进行json结果的处理，很方便的提取json数据——————jmeter不仅可以处理json数据，（json提取器），还可以提取其他数据（正则表达式提取器）

### **3数据用例的实现**

区别1：jmeter比较适合进行数据与操作分离，而postman比较适合把数据和操作放在一起，显然postman操作更简单，jmeter更便于维护

区别2： postman也支持csv数据文件的导入，但是每次执行时都需要收工加载数据文件。不方便（所以只能做半自动化）  
  Jmeter可以进行完全自动化，特别是引入ant后效果更明显

### **4断言的实现**

区别1：Postman有很多自带的断言函数，直接引用即可，操作非常方便。。。 jmeter也自带断言组件，操作非常直观。 区别： postman用函数断言， jmeter用元件进行断言

区别2：jmeter支持正则表达式断言，postman不支持

区别3：Jmeter的断言更丰富。 postman需要通过编程来实现同样的效果，所以难度更大

### **5执行**

区别：默认执行，postman不能保存结果，jmeter可以报存结果  
Postman可以通过newman实现批量执行和保存结果，jmeter可以通过ant实现批量执行和保存结果

### **6其他**

Postman比较适合做手工接口测试，因为简单，可以实现半自动化  
Jmeter比较适合自动化接口测试，因为功能强大并且可以保存脚本，批量执行设置很容易  
Postman一般用来做接口测试，用来发现BUG，验证后台程序  
Jmeter一般用来做自动化测试，做冒烟测试。

**appium自动化测试**

1. 安卓SDK需要 JDK 环境；首先下载安装jdk（1.8版本）
2. 在我的电脑D盘；在我的电脑属性高级系统设置-环境变量-新建一个JAVA-HOME复制jdk目录，然后在path文件下新建一个复制jdk的bin目录这样jdk环境变量就配置成功了，在控制板上输入java -version来检查jdk下载安装成功和查看。
3. 然后下载安装Android SDK；安装完成后，需要配置一下添加一个环境量ANDROID\_HOME，设置值为sdk目录，还要配置环境变量PATH ，加入adb所在目录；在运行程序的电脑上用USB线连接上我测试的安卓手机

4.进入手机设置->关于手机，不断点击版本号菜单（7次以上），退出到上级菜单，在开发者模式中，启动USB调试，手机界面弹框中点击确定允许USB调试。连接好以后，打开命令行窗口，执行adb devices -l命令来列出连接在电脑上的安卓设备。通过傲软投屏将手机APP投屏到运行程序的电脑上，启动appium service，在cmd中通过命令nox\_adb.exe connect 手机ip+端口号连接手机，启动Android Sdk包中的uiautomateviewer.bat，通过adb把手机屏幕投屏到uiautomateviewer里

**Get和post的区别**  
POST和GET都是向服务器提交数据，并且都会从服务器获取数据。  
区别：  
1）传送方式：get通过地址栏传输，post通过报文传输  
2）传送长度：get参数有长度限制（受限于url长度），而post无限制  
3）GET产生一个TCP数据包（对于GET方式的请求，浏览器会把http header和data一并发送出去，服务器响应200返回数据），POST产生两个TCP数据包（对于POST，浏览器先发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 ok返回数据）  
4）get请求参数会被完整保留在浏览历史记录里，而post中的参数不会被保留  
5）在做数据查询时，建议用GET方式；而在做数据添加、修改或删除时，建议用post方式

****你平常做接口测试的过程中发现过哪些bug?****  
****答：****  
常规错误，接口没实现，没按约定返回结果，边界值处理出错等。  
输入异常值（空值、特殊字符、超过约定长度等），接口抛错，没做封装处理；  
输入错误的参数、多输入、少输入参数，接口可能出现的错误；  
安全性问题，如明文传输、返回结果含有敏感信息，没对用户身份信息做校验，没做恶意请求拦截等；  
性能问题，如接口并发插入多条相同操作，响应时间过长，接口压测出现瓶颈等；

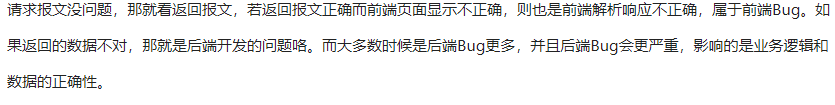
****当一个接口出现异常时候，你是如何分析异常的？****  
****答：****  
先抓包，用fiddler（charles）工具抓包，或者浏览器上F12调试工具；APP上的话，那就用Fiddler做代理，通过手机设置代理去看请求和返回报文；  
查看后端日志，如Linux系统通过xhell连上服务器，查看接口日志，查看是否有报错信息（命令：tail -f 日志文件）；

## **什么是接口测试？为什么要做接口测试？**

　　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。

由于如今的系统复杂度不断上升，传统的测试方法成本急剧增加且测试效率大幅下降，所以就要做接口测试。同时，接口测试相对容易实现自动化持续集成，且相对UI自动化也比较稳定，可以减少人工回归测试人力成本与时间，缩短测试周期，支持后端快速发版需求。接口持续集成是为什么能低成本高收益的根源。现在很多系统前后端架构是分离的，从安全层面来说，只依赖前端进行限制已经完全不能满足系统的安全要求（绕过前面实在太容易）， 需要后端同样进行控制，在这种情况下就需要从接口层面进行验证。前后端传输、日志打印等信息是否加密传输也是需要验证的，特别是涉及到用户的隐私信息，如身份证，银行卡等。

**如何分析一个bug属于前台和后台**



**接口测试的流程**

