

上海时年信息科技有限公司

python与游戏测试

主讲人：小白

2018年4月17日



目录/Contents

01

数据库操作

02

调用接口

03

服务器操作

04

调用cmd

05

UI自动化测试

06

参考文献

和其他语言一样，python可以直接对数据库进行操作。这里介绍如何用python对mysql数据库进行操作

python3使用的库是PyMySQL，Python2中则使用mysqldb。

●准备：

1.PyMySQL 库安装：pip install PyMySQL

2.mysql数据库已存在。

●连接数据库

- # 打开数据库连接
- `conn= pymysql.connect("123.123.10.10","testuser","test123","TESTDB")`
- # 使用 `cursor()` 方法创建一个游标对象 `cursor`
- `cur = conn.cursor()`

●对数据库进行操作

- # 使用 `execute()` 方法执行 SQL 查询
- `cur.execute("SELECT VERSION()")`
- # 使用 `fetchone()` 方法获取单条数据，及获得刚刚的执行结果
- `data = cur.fetchone()`

● 注意事项

- 如果是插入、更新、删除等操作，需要在`cur.execute()`执行后，进行`conn.commit()`操作，否则只会在程序执行的时候结果会变动，但实际数据库的值不会变(e.g.)；另外可以在`execute`操作后加事务控制：如果`execute`执行失败，则执行 `cur.rollback()`进行回滚
- 在连接数据库结束后，需要执行`conn.close()`操作

- httplib 直接处理HTTP /HTTP请求和响应
- urllib 构建在httplib上
- pycurl C语言编写的，速度很快，比urllib和httplib都快
- requests API 非常简洁好用

●准备：

- 1.requests库安装：pip install requests
- 2.已使用抓包工具获得接口信息

```
import requests
...

resp = requests.get('http://www.mywebsite.com/user')
resp = requests.post('http://www.mywebsite.com/user')
resp = requests.put('http://www.mywebsite.com/user/put')
resp = requests.delete('http://www.mywebsite.com/user/delete')
```

```
import requests
url = "http://119.29.114.101:8004/game?acid=1814&ver=2.1.1"
data1 = json.dumps({"data": {"activityId": "0", "token": "75336cd5374ae4685612bba717ed2f26", "uid": "7-813808"},
                    "sign": "7b559922114c2b07dce24e052e685c6a", "ts": 59})
r1 = requests.post(url, data1)
# 取返回值
b1 = (r1.json()["data"]["addDiamond"])
```

```
▼ r1 = {Response} <Response [200]>
  [9] _content = {str} '{"code":4000,"msg":"登录超时，请尝试重新登录","ts":59}'
  [9] _content_consumed = {bool} True
  [9] _next = {NoneType} None
  [9] apparent_encoding = {str} 'utf-8'
  > [9] connection = {HTTPAdapter} <requests.adapters.HTTPAdapter object at 0x0000
  [9] content = {str} '{"code":4000,"msg":"登录超时，请尝试重新登录","ts":59}'
  > [9] cookies = {RequestsCookieJar} <RequestsCookieJar[]>
  > [9] elapsed = {timedelta} 0:00:00.167539
  [9] encoding = {str} 'UTF-8'
  > [9] headers = {CaseInsensitiveDict} {'Date': 'Sun, 15 Apr 2018 07:44:23 GMT', 'Conte
```

```
r1 = requests.post(url, data1)  r1: <Response [200]>
# 取返回值
b1 = (r1.json()["code"])  b1: 4000
```

python中的paramiko模块是用来实现ssh连接到远程服务器上的库，在进行连接的时候，可以用来执行命令，也可以用来上传文件。

●准备：

- 1.paramiko库安装：pip install paramiko
- 2.已知服务器的地址、账号密码。

- 获取连接：

```
ssh = paramiko.SSHClient()
```

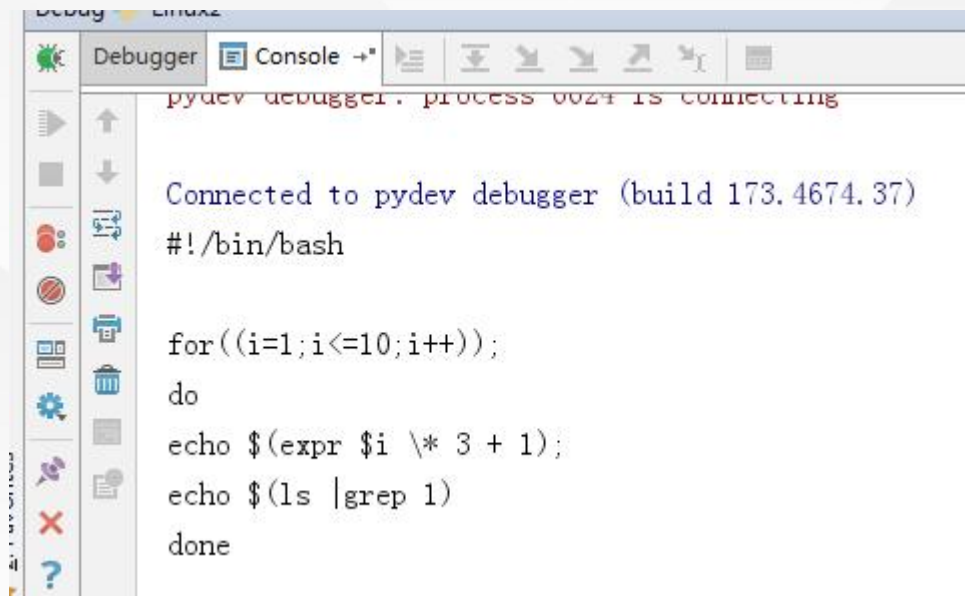
```
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
ssh.connect(HostIP, port, username, passwd)
```

- 执行命令：

```
return=ssh.exec_command('cat xiaobaitest/1.sh')  
print(return)
```

```
[root@VM_34_100_centos ~]# cat xiaobaitest/1.sh  
#!/bin/bash  
  
for((i=1;i<=10;i++));  
do  
echo $(expr $i \* 3 + 1);  
echo $(ls |grep 1)  
done
```



- 上传文件：

```
ftp = conn.open_sftp()
```

```
ftp.put(inpath,outpath)#如inpath='Linux.py',outpath='/root/xiaobaitest/Linux.py'
```

```
ftp.close()
```

- 下载文件：

```
ftp = conn.open_sftp()
```

```
ftp.get(inpath,outpath)
```

```
ftp.close()
```

```
[root@vm_34_100-centos xiaobaitest]# ll
总用量 28
-rw-r--r-- 1 root root 100 4月  9 21:07 1.sh
drwxr-xr-x 4 root root 4096 3月 18 23:22 2
-rw-r--r-- 1 root root 236 4月 13 15:37 for.sh
-rw-r--r-- 1 root root 419 4月 16 00:47 Linux.py
-rw-r--r-- 1 root root 111 3月 19 06:56 sql.sh
drwxr-xr-x 3 root root 4096 3月 16 00:08 test
-rwxr-xr-x 1 root root 115 4月 13 15:45 test.sh
```

- python的OS模块：

1.os.system(cmd)

```
1 import os
2 os.system('date')
```

2.os.popen(cmd)

```
1 import os
2 nowtime = os.popen('date')
3 print nowtime.read()
```

- commands模块

getstatusoutput 获取执行命令的状态值以及返回信息

```
1 import commands
2 status, output = commands.getstatusoutput('date')
3 print status      # 0
4 print output      # 2016年 06月 30日 星期四 19:26:21 CST
```

- subprocess模块：

主要用来取代一些旧的模块方法，如os.system、os.spawn*、os.popen*、commands.*等。subprocess通过子进程来执行外部指令，并通过input/output/error管道，获取子进程的执行的返回信息。

```
import subprocess
a=subprocess.Popen("tasklist").communicate()
print(a)
```

subprocess.Popen(args, bufsize=0, executable=None, stdin=None, stdout=None, stderr=None, preexec_fn=None, close_fds=False, shell=False, cwd=None, env=None, universal_newlines=False, startupinfo=None, creationflags=0)

- adb命令封装：
比如查看当前apk的内存占用

```
def get_mem_using(package_name="com.shinian.wineleven.uc"):
    """查看apk的内存占用
    :param package_name: com.TimeYears.tbball
    :return: 单位KB
    """
    if not package_name:
        package_name = get_apk_info()[0]
    result = run_cmd("adb shell dumpsys meminfo {}".format(package_name))
    info = re.search('TOTAL\W+\d+', str(result)).group()
    mem = ''
    try:
        mem = info.split()
    except Exception as e:
        print(info)
        print(e)
    return str(float(mem[-1])/1000)+"MB"
for i in range(100):
    print(get_mem_using())
```

323.265MB
323.289MB
323.237MB
323.282MB
323.31MB
324.464MB
325.346MB
325.334MB

- 按键精灵：

背景：小游戏

```
For i=1 to 50
    RunApp"com.TimeYears.tball"
    Delay 25000
    SnapShot "/sdcard/DCIM/" & i & ".png"
    KillApp "com.TimeYears.tball"
    Delay 5000
Next
```

- ATX :

原理：基于OpenCV的图像识别技术，使用图像识别完成游戏的自动化。

准备：

Python版本限制 ≥ 2.7 && ≤ 3.5

Android 4.1+

iOS 9.0+

adb已安装

安装：

pip install --upgrade --pre atx

pip install opencv_contrib_python

```
d.start_app('com.shinian.wineleven.uc',  
time.sleep(10))  
d.click_image(u"选区.1280x720.png")  
d.click_image(u"新区.1280x720.png")  
d.click_image(u"进入游戏.1280x720.png")  
d.click_image(u"确认.1280x720.png")  
d.click_image(u"曼联.1280x720.png")  
d.click_image(u"创建球队.1280x720.png")  
time.sleep(10))
```

API有c.click(),c.swipe()等

网址：

<https://github.com/NetEaseGame/ATX/blob/master/docs/API.md>

- AirTest :

原理：基于图像识别的 Airtest 框架、基于UI控件搜索的 Poco 框架

例：



- CMD :
测试部门培训分享:Python脚本分享-古力特
- AirTest :
https://github.com/AirtestProject/Airtest/blob/master/README_zh.rst
- ATX :
测试部门培训分享:ATX-贝隆
<https://github.com/NetEaseGame/ATX/blob/master/docs/API.md>
- 其他 :
<https://testerhome.com/topics/7723>



谢谢欣赏

— THANK YOU —

