### Python邮件发送

### SMTP（Simple Mail Transfer Protocol）

* 即简单邮件传输协议。它是一组用于从源地址到目的地址传输邮件的规范，通过它来控制邮件的中转方式。SMTP 协议属于TCP/IP协议簇，它帮助每台计算机在发送或中转信件时找到下一个目的地。SMTP 服务器就是遵循 SMTP 协议的发送邮件服务器。

### SMTP 认证

* SMTP 认证，简单地说就是要求必须在提供了账户名和密码之后才可以登录 SMTP 服务器，这就使得那些垃圾邮件的散播者无可乘之机。
* 增加 SMTP 认证的目的是为了使用户避免受到垃圾邮件的侵扰。

更多资料： <http://help.163.com/09/1223/14/5R7P6CJ600753VB8.html>

### smtplib模块

Python内置对SMTP的支持，可以发送纯文本邮件、HTML邮件以及带附件的邮件。

Python对SMTP支持有smtplib和email两个模块，email负责构造邮件，smtplib负责发送邮件。

注意：使用前需要开启SMTP服务

案例：使用163邮箱来结合smtp模块发送邮件 准备工作：

import smtplib #发送邮件模块

from email.mime.text import MIMEText #定义邮件内容

from email.header import Header #定义邮件标题

#发送邮箱服务器

smtpserver='smtp.163.com'

#发送邮箱用户名密码

user='yuexiaolu2015@163.com'

password='…'

#发送和接收邮箱

sender='yuexiaolu2015@163.com'

receive='yuexiaolu2015@126.com'

#发送邮件主题和内容

subject='Web Selenium 自动化测试报告'

content='<html><h1 style="color:red">我要自学网，自学成才!</h1></html>'

#HTML邮件正文

msg=**MIMEText**(content,'html','utf-8')

msg['Subject']=**Header**(subject,'utf-8')

msg['From']='yuexiaolu2015@163.com'

msg['To'] = 'yuexiaolu2015@126.com'

#SSL协议端口号要使用465

smtp = smtplib.**SMTP\_SSL**(smtpserver, 465)

**#HELO** 向服务器标识用户身份

smtp.**helo**(smtpserver)

#服务器返回结果确认

smtp.**ehlo**(smtpserver)

#登录邮箱服务器用户名和密码

smtp.**login**(user,password)

**print**("开始发送邮件...")

smtp.**sendmail**(sender,receive,msg.as\_string())

smtp.**quit**()

**print**("邮件发送完成！")

### 发送带附件的邮件

### 案例：发送E:\Python\_script\目录下 logo.png图片文件到指定的邮箱

import smtplib #发送邮件模块

from email.mime.text import MIMEText #定义邮件内容

from email.mime.multipart import MIMEMultipart #用于传送附件

#发送邮箱服务器

smtpserver='smtp.163.com'

#发送邮箱用户名密码

user='yuexiaolu2015@163.com'

password='070337shu'

#发送和接收邮箱

sender='yuexiaolu2015@163.com'

receives=['yuexiaolu2015@126.com','yuexiaolu2015@sina.com']

#发送邮件主题和内容

subject='Web Selenium 附件发送测试'

content='<html><h1 style="color:red">我要自学网，自学成才!</h1></html>'

#构造附件内容

send\_file=**open**(r"E:\Python\_script\logo.png",'rb').**read**()

att=**MIMEText**(send\_file,'base64','utf-8')

att["Content-Type"]='application/octet-stream'

att["Content-Disposition"]='attachment;filename="logo.png"'

#构建发送与接收信息

msgRoot=**MIMEMultipart**()

msgRoot.**attach**(MIMEText(content, 'html', 'utf-8'))

msgRoot['subject']=subject

msgRoot['From']=sender

msgRoot['To'] = ','.**join**(receives)

msgRoot.**attach**(att)

#SSL协议端口号要使用465

smtp = smtplib.**SMTP\_SSL**(smtpserver, 465)

**#HELO** 向服务器标识用户身份

smtp.**helo**(smtpserver)

#服务器返回结果确认

smtp.**ehlo**(smtpserver)

#登录邮箱服务器用户名和密码

smtp.**login**(user,password)

**print**("Start send email...")

smtp.**sendmail**(sender,receives,msgRoot.as\_string())

smtp.**quit**()

**print**("Send End！")

### 整合测试报告发送

### 案例：获取…\Test\_Baidu\test\_report目录下最新的测试报告

import os #用于访问操作系统功能的模块

#报告存放位置

report\_dir='./test\_report'

**#os**.**listdir**() 方法用于返回指定的文件夹包含的文件或文件夹的名字的列表

lists=os.**listdir**(report\_dir)

#按时间顺序对该目录文件夹下面的文件进行排序

lists.**sort**(key=lambda fn:os.path.getatime(report\_dir+'\\'+fn))

**print**(lists)

**print**("latest report is :"+lists[-1])

#输出最新报告的路径

file=os.path.join(report\_dir,lists[-1])

**print**(file)

Python os模块相关知识： <http://www.cnblogs.com/MnCu8261/p/5483657.html>

### lambda 介绍 <http://www.cnblogs.com/evening/archive/2012/03/29/2423554.html>

发送测试报告

案例：将E:\Python\_script\unittest\Test\_Baidu生成的最新测试报告发送到指定邮箱。

**import** unittest

**from** BSTestRunner **import** BSTestRunner

**import** time

**import** smtplib *#发送邮件模块*

**from** email.mime.text **import** MIMEText *#定义邮件内容*

**from** email.header **import** Header *#定义邮件标题*

**import** os

**def** **send\_mail**(latest\_report):

f=open(latest\_report,'rb')

mail\_content=f.read()

f.close()

smtpserver='smtp.163.com'

*# 发送邮箱用户名密码*

user = 'yuexiaolu2015@163.com'

password = '…'

*# 发送和接收邮箱*

sender = 'yuexiaolu2015@163.com'

receives = ['yuexiaolu2015@126.com', 'yuexiaolu2015@sina.com']

*# 发送邮件主题和内容*

subject = 'Web Selenium 自动化测试报告'

*# HTML邮件正文*

msg = MIMEText(mail\_content, 'html', 'utf-8')

msg['Subject'] = Header(subject, 'utf-8')

msg['From'] = sender

msg['To'] = ','.join(receives)

smtp = smtplib.SMTP\_SSL(smtpserver, 465)

*# HELO 向服务器标识用户身份*

smtp.helo(smtpserver)

*# 服务器返回结果确认*

smtp.ehlo(smtpserver)

*# 登录邮箱服务器用户名和密码*

smtp.login(user, password)

print("Start send Email...")

smtp.sendmail(sender, receives, msg.as\_string())

smtp.quit()

print("Send Email end!")

**def** **latest\_report**(report\_dir):

lists = os.listdir(report\_dir)

*# 按时间顺序对该目录文件夹下面的文件进行排序*

lists.sort(key=**lambda** fn: os.path.getatime(report\_dir + '\\' + fn))

print(("new report is :" + lists[-1]))

file = os.path.join(report\_dir, lists[-1])

print(file)

**return** file

**if** \_\_name\_\_ == '\_\_main\_\_':

test\_dir='./test\_case'

report\_dir='./test\_report'

discover = unittest.defaultTestLoader.discover(test\_dir, pattern="test\*.py")

now = time.strftime("%Y-%m-%d %H\_%M\_%S")

report\_name = report\_dir + '/' + now + 'result.html'

**with** open(report\_name, 'wb') **as** f:

runner = BSTestRunner(stream=f, title="Test Report", description="baidu search")

runner.run(discover)

f.close()

*#h获取最新测试报告*

latest\_report=latest\_report(report\_dir)

*#发送邮件报告*

send\_mail(latest\_report)

### 163邮箱发生失败的常见问题

<http://help.163.com/09/1224/17/5RAJ4LMH00753VB8.html>

补充知识点——By 元素定位

from selenium import webdriver

from selenium.webdriver.common.by import By

from time import sleep

driver=webdriver.**Firefox**()

driver.**get**("http://www.baidu.com/")

driver.**implicitly\_wait**(5)

driver.**find\_element**(By.ID,'kw').**clear**()

driver.**find\_element**(By.NAME,'wd').**send\_keys**("Selenium ")

driver.**find\_element**(By.CLASS\_NAME,'s\_ipt').**send\_keys**("自学网 ")

driver.**find\_element**(By.CSS\_SELECTOR,"#kw").**send\_keys**("自动化测试")

**sleep**(3)

driver.**find\_element**(By.ID,'su').**click**()

**sleep**(3)

driver.**quit**()

* find\_element([By.ID](http://By.ID),"loginName")
* find\_element([By.NAME](http://By.NAME),"SubjectName")
* find\_element(By.CLASS\_NAME,"u-btn-levred")
* find\_element(By.TAG\_NAME,"input")
* find\_element(By.LINK\_TEXT,"退出")
* find\_element(By.PARTIAL\_LINK\_TEXT,"退")
* find\_element(By.XPATH,".//\*[@id='Title")
* find\_element(By.CSS\_SELECTOR,"[type=submit]")

补充知识点——方法的参数个数

**def** **fun\_args1**(args):

print("args is %s" %args)

**def** **fun\_args2**(args1,args2):

print("args is %s and %s" %(args1,args2))

**def** **fun\_var\_args**(\*args):

**for** value **in** args:

print("args:", value)

*# fun\_args1('51zxw')*

*# fun\_args1()*

*# fun\_args2('51zxw','Python')*

*# fun\_args2('51zxw')*

fun\_var\_args("Python")

fun\_var\_args("hello","51zxw")

fun\_var\_args("Selenium",'Python','51zxw')

fun\_var\_args()

Page Object

Page Object是Selenium自动化测试项目开发实践的最佳设计模式之一，通过对界面元素和功能模块的封装减少冗余代码，同时在后期维护中，若元素定位或功能模块发生变化，只需要调整页面元素或功能模块封装的代码，提高测试用例的可维护性。

[BasePage.py](http://BasePage.py)

**from** time **import** sleep

**class** **Page**():

'''页面基础类'''

*#初始化*

**def** **\_\_init\_\_**(self, dirver):

self.base\_url = 'http://localhost'

self.driver = dirver

self.timeout = 10

*#打开不同的子页面*

**def** **\_open**(self, url):

url\_ = self.base\_url + url

print("Test page is： %s" %url\_)

self.driver.maximize\_window()

self.driver.get(url\_)

sleep(2)

**assert** self.driver.current\_url == url\_, 'Did ont land on %s' % url\_

**def** **open**(self):

self.\_open(self.url)

*#元素定位方法封装*

**def** **find\_element**(self,\*loc):

**return** self.driver.find\_element(\*loc)

[LoginPage.py](http://LoginPage.py)

**from** BasePage **import** \*

**from** selenium.webdriver.common.by **import** By

**class** **LoginPage**(Page):

'''首页登录页面'''

url='/'

*#定位器*

username\_loc=(By.NAME,'username')

password\_loc=(By.NAME,'password')

submit\_loc=(By.NAME,'Submit')

*#用户名输入框元素*

**def** **type\_username**(self,username):

self.find\_element(\*self.username\_loc).clear()

self.find\_element(\*self.username\_loc).send\_keys(username)

*#密码输入框元素*

**def** **type\_password**(self,password):

self.find\_element(\*self.password\_loc).clear()

self.find\_element(\*self.password\_loc).send\_keys(password)

*#登录按钮元素*

**def** **type\_submit**(self):

self.find\_element(\*self.submit\_loc).click()

*#登录功能模块封装*

**def** **test\_user\_login**(driver,username,password):

'''测试用户名密码是否可以登录'''

login\_page=LoginPage(driver)

login\_page.open()

login\_page.type\_username(username)

login\_page.type\_password(password)

login\_page.type\_submit()

loin\_test.py

**from** LoginPage **import** \*

**from** selenium **import** webdriver

driver=webdriver.Firefox()

username = '51zxw'

password = '123456'

test\_user\_login(driver, username, password)

sleep(3)

driver.quit()

参考文档：

<http://www.liaoxuefeng.com/>

<http://blog.csdn.net/menglei8625/article/details/7721746>

<http://blog.csdn.net/bravezhe/article/details/7659198>

<http://blog.csdn.net/spritzdance/article/details/5362220>