1. **Python的SMTP**

**第一部分**

1. **内容的引入**
2. 项目中如何通过给用户的邮件发验证码？
3. 如何把相关信息推荐到用户的邮件中？
4. **邮件发送/接收的三大协议**
5. SMTP（Simple Mail Transfer Protocol）即[简单邮件传输协议](https://baike.baidu.com/item/%E7%AE%80%E5%8D%95%E9%82%AE%E4%BB%B6%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE" \t "https://baike.baidu.com/item/SMTP/_blank)

它是一组用于由源地址到目的地址传送[邮件](https://baike.baidu.com/item/%E9%82%AE%E4%BB%B6" \t "https://baike.baidu.com/item/SMTP/_blank)的规则，由它来控制信件的中转方式。[SMTP协议](https://baike.baidu.com/item/SMTP%E5%8D%8F%E8%AE%AE" \t "https://baike.baidu.com/item/SMTP/_blank)属于[TCP/IP协议簇](https://baike.baidu.com/item/TCP/IP%E5%8D%8F%E8%AE%AE%E7%B0%87" \t "https://baike.baidu.com/item/SMTP/_blank)，它帮助每台[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA" \t "https://baike.baidu.com/item/SMTP/_blank)在发送或中转信件时找到下一个目的地。通过SMTP协议所指定的[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/SMTP/_blank),就可以把E-mail寄到收信人的服务器上了，整个过程只要几分钟。SMTP[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/SMTP/_blank)则是遵循SMTP协议的发送[邮件服务器](https://baike.baidu.com/item/%E9%82%AE%E4%BB%B6%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/SMTP/_blank)，用来发送或中转发出的[电子邮件](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E9%82%AE%E4%BB%B6" \t "https://baike.baidu.com/item/SMTP/_blank)

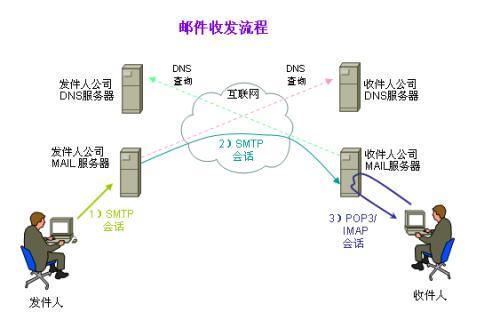
2、POP3

POP3是Post Office Protocol 3的简称，即邮局协议的第3个版本,它规定怎样将个人计算机连接到Internet的邮件服务器和下载电子邮件的电子协议。它是因特网电子邮件的第一个离线协议标准,POP3允许用户从服务器上把邮件存储到本地主机（即自己的计算机）上,同时删除保存在邮件服务器上的邮件，而POP3服务器则是遵循POP3协议的接收邮件服务器，用来接收电子邮件的

3、IMAP

IMAP全称是Internet Mail Access Protocol，即交互式邮件存取协议，它是跟POP3类似邮件访问标准协议之一。不同的是，开启了IMAP后，您在电子邮件客户端收取的邮件仍然保留在服务器上，同时在客户端上的操作都会反馈到服务器上，如：删除邮件，标记已读等，服务器上的邮件也会做相应的动作。所以无论从浏览器登录邮箱或者客户端软件登录邮箱，看到的邮件以及状态都是一致的。

**三、邮件收发流程**



**四、使用Python开发发送邮件的知识点和模块**

**4.1、SSL**

SSL(Secure Sockets Layer 安全套接层),及其继任者传输层安全（Transport Layer Security，TLS）是为网络通信提供安全及数据完整性的一种安全协议。TLS与SSL在传输层对网络连接进行加密。SSL为Netscape所研发，用以保障在Internet上数据传输的安全，利用数据加密(Encryption)的技术，可确保数据在网络上的传输过程中不会被截取及窃听。SSL协议位于TCP/IP协议与各种应用层协议之间，为数据通讯提供安全支持。SSL默认的端口号465,设置发件服务器端口号. 有SSL(465)和非SSL(25)两种形式。

**4.2、[smtplib与email模块](http://www.cnblogs.com/freely/p/6859117.html)**

4. 2.1 smtplib模块

SMTP是发送邮件的协议，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。smtplib模块定义了一个简单的SMTP客户端，可以用来在互联网上发送邮件。

定义的类有如下：

class smtplib.SMTP([host[, port[, local\_hostname[, timeout]]]])  
class smtplib.SMTP\_SSL([host[, port[, local\_hostname[, keyfile[, certfile[, timeout]]]]]])

Python SMTP对象使用sendmail方法发送邮件，语法如下：

SMTP.sendmail(from\_addr, to\_addrs, msg[, mail\_options, rcpt\_options]

参数说明：

 from\_addr: 邮件发送者地址。

 to\_addrs: 字符串列表，邮件接收地址。

 msg: 发送消息

这里要注意一下第三个参数，msg是字符串，表示邮件。我们知道邮件一般由标题，发信人，收件人，邮件内容，附件等构成，发送邮件的时候，要注意msg的格式。这个格式就是smtp协议中定义的格式。

4.2.2、email模块

emial模块用来处理邮件消息，其包括的类有class email.mime.text.MIMEText(\_text[, \_subtype[, \_charset]])：#MIME文本对象，其中\_text是邮件内容，\_subtype邮件类型，可以是text/plain（普通文本邮件），html/plain(html邮件), \_charset编码，可以是gb2312等等。

MIME(Multipurpose Internet Mail Extensions)多用途互联网邮件扩展类型,它是一个互联网标准，扩展了电子邮件标准，使其能够支持：

非ASCII字符文本；非文本格式附件（二进制、声音、图像等）；由多部分（multiple parts）组成的消息体；包含非ASCII字符的头信息（Header information）。

4.2.3、注意的地方

使用qq的邮件服务器需要注意的两个地方主要是：

1.协议问题 使用465端口 SSL 协议

2.口令问题 出现SMTPAuthenticationError 主要的原因就是口令和帐号信息不对，这里我们使用qq服务器发送 需要先到邮箱里设置独立密码(必须), 然后开启。

**五、开通smtp服务**

1. 设置独立账号
2. 开通smtp服务

成功开启POP3/SMTP服务,在第三方客户端登录时，密码框请输入以下授权码：

yhevfpvfpioofaeg

关于授权码:

http://service.mail.qq.com/cgi-bin/help?subtype=1&&id=28&&no=1001256

**第二部分**

1. **邮件发送步骤**

6.1编写邮件发送的基本思路：

1、导入模块

2、设置邮件会话对象

3、设置邮件信息(邮件内容，邮件类型，标题，发送人，接收人）

4、登录发送邮箱

5、发送邮件

6.2 简单的HTML邮件发送

具体实现

#coding=utf-8

#1、导入模块

import smtplib

from email.mime.text import MIMEText

import traceback

result = False

#2、邮件会话(SSL)对象

session = smtplib.SMTP\_SSL("smtp.qq.com", 465, 30)

#3、设置邮件信息(包括正文、邮件类型 、标题、发送人、接收人)

try:

# 设置正文为符合邮件格式的HTML内容

msg = MIMEText("<h1>Hi</h1><p>test mail from czf3</p>", 'html')

# 设置邮件标题

msg['subject'] = "This is a python stmp test mail"

# 设置发送人

msg['from'] = "12345678@qq.com"

# 设置邮件接收人

msg['to'] = "12345678@qq.com"

#4、 登陆邮箱

session.login("12345678@qq.com", "czf\_qq\_96944")

#5、 发送邮件

session.sendmail("12345678@qq.com","12345678@qq.com", msg.as\_string())

result = True

except Exception:

traceback.print\_exc()

finally:

print(result)

6.3 纯文本邮件发送

*#encoding=utf-8  
#使用stmp发送一封简单邮件  
#1、导入模块***import** email  
**from** email.mime.text **import** MIMEText  
**import** smtplib  
**import** traceback  
mark=**False  
try**:  
 session=smtplib.SMTP\_SSL(**"smtp.qq.com"**,465,30)  
 msg=MIMEText(**"test email smtp this is a test email! "**,**"plain"**)  
 msg[**'subject'**]=**'Python 发送邮件测试'** msg[**'from'**]=**'1634217611@qq.com'** msg[**'to'**]=**'1634217611@qq.com'** session.login(**'1634217611@qq.com'**,**"yhevfpvfpioofaeg"**)  
 session.sendmail(**'1634217611@qq.com'**,**'1634217611@qq.com'**,str(msg))  
 mark=**True  
except** Exception **as** e:  
 print(traceback.print\_exc())  
**if** mark==**True**:  
 print(**"邮件发送成功!"**)

**第三部分**

**七、创建一个带附件的邮件**

email模块中：

class email.mime.multipart.MIMEMultipart([\_subtype[, boundary[, \_subparts[, \_params]]]]：#MIMEBase的一个子类，多个MIME对象的集合，\_subtype默认值为mixed。boundary是MIME ultipart的边界，默认边界是可数的。

Base64（text/base64）是一种用64个字符来表示任意二进制数据的方法，好处是编码后的文本数据可以在邮件正文、网页等直接显示。

如何构建一个附件：

1、传送当前目录下的 czf.txt 文件

att1 = MIMEText(open('czf.txt', 'rb').read(), 'base64', 'utf-8')

2、指定附件的"Content-Type"等于application/octet-stream

att1["Content-Type"] = 'application/octet-stream'

3、指定附件的att1["Content-Disposition"] = 'attachment; filename="czf.txt"'

# 这里的filename可以任意写，写什么名字，邮件中显示什么名字

att1["Content-Disposition"] = 'attachment; filename="czf.txt"'

4、把附件放到 MIMEMultipart对象下

message.attach(att1)

#coding=utf-8

#1、导入模块

import smtplib

from email.mime.text import MIMEText

import traceback

from email.mime.multipart import MIMEMultipart

result = False

#2、邮件会话(SSL)对象

session = smtplib.SMTP\_SSL("smtp.qq.com", 465, 30)

try:

#3、创建一个带附件的实例

message = MIMEMultipart()

#4、设置邮件信息(包括正文、邮件类型 、标题、发送人、接收人)

# 设置发送人

message['From'] = "12345678@qq.com"

# 设置邮件接收人

message['To'] = "12345678@qq.com"

# 设置邮件标题

message['Subject'] = 'Python SMTP 邮件附件测试'

#邮件正文内容

message.attach(MIMEText('这是Python 邮件附件发送测试……', 'plain', 'utf-8'))

# 5、构造附件1，传送当前目录下的 czf.txt 文件

att1 = MIMEText(open('czf.txt', 'rb').read(), 'base64', 'utf-8')

att1["Content-Type"] = 'application/octet-stream'

# 这里的filename可以任意写，写什么名字，邮件中显示什么名字

att1["Content-Disposition"] = 'attachment; filename="czf.txt"'

message.attach(att1)

#5、 登陆邮箱

session.login("12345678@qq.com", "czf\_qq\_96944")

#6、 发送邮件

session.sendmail("12345678@qq.com","12345678@qq.com", message.as\_string())

result = True

except Exception:

traceback.print\_exc()

finally:

print(result)

**第四部分**

**八、在邮件中带图片**

邮件体被分为多个段，每个段又包含段头和段体两部分，这两部分之间也以空行分隔。常见的multipart类型有三种：multipart/mixed, multipart/related和multipart/alternative。从它们的名称，不难推知这些类型各自的含义和用处。它们之间的层次关系可归纳为下图所示：

+------------------------- multipart/mixed ---------------------------+  
|   
| +----------------- multipart/related ------------------+   
| | | |  
| | +----- multipart/alternative ------+ +----------+ | +------+ |  
| | | | 内嵌资源 | | 附件 | |  
| | | +------------+ +------------+ | +----------+ | +------+ |  
| | | | 纯文本正文 | | 超文本正文 | | | |  
| | | +------------+ +------------+ | +----------+ | +------+ |  
| | | | 内嵌资源 | | 附件 | |  
| | +----------------------------------+ +----------+ | +------+ |

+---------------------------------------------------------------------+  
可以看出，如果在邮件中要添加附件，必须定义multipart/mixed段；如果存在内嵌资源（内嵌资源是在邮件里面展示的图片，视频），至少要定义multipart/related段；如果纯文本与超文本共存，至少要定义multipart/alternative段。什么是“至少”？举个例子说，如果只有纯文本与超文本正文，那么在邮件头中将类型扩大化，定义为multipart/related，甚至multipart/mixed，都是允许的。

#coding=utf-8

#1、导入模块

import smtplib

from email.mime.text import MIMEText

import traceback

from email.mime.multipart import MIMEMultipart

from email.mime.image import MIMEImage

result = False

#2、邮件会话(SSL)对象

session = smtplib.SMTP\_SSL("smtp.qq.com", 465, 30)

try:

#3、创建一个可以带图片和超链接的MIMEMultipart对象

message =MIMEMultipart('related')

#4、设置邮件信息(包括正文、邮件类型 、标题、发送人、接收人)

# 设置发送人

message['From'] = "12345678@qq.com"

# 设置邮件接收人

message['To'] = "12345678@qq.com"

# 设置邮件标题

message['Subject'] = 'Python SMTP 邮件附件测试'

#邮件正文内容

mail\_msg = """

<p>Python 邮件发送测试...</p>

<p><a href="http://www.rupeng.com">如鹏教育网</a></p>

<p>图片演示：</p>

<p><img src="cid:image1"></p>

"""

msgAlternative = MIMEMultipart('alternative')

message.attach(msgAlternative)

msgAlternative.attach(MIMEText(mail\_msg, 'html', 'utf-8'))

#5、指定图片为当前目录

fp = open('logo.png', 'rb')

msgImage = MIMEImage(fp.read())

fp.close()

#6、定义图片ID，在HTML文本中引用

msgImage.add\_header('Content-ID', '<image1>')

message.attach(msgImage)

#7、 登陆邮箱

session.login("12345678@qq.com", "czf\_qq\_96944")

#8、 发送邮件

session.sendmail("12345678@qq.com","12345678@qq.com", message.as\_string())

result = True

except Exception:

traceback.print\_exc()

finally:

print(result)