

发送邮件是在web中比较常见的，比如：注册验证码，密保邮箱，用户购买提示，广告推送等等。

flask中使用 flask-mail第三方扩展：

```
pip install Flask-Mail
```

1.邮箱设置

这里使用163邮箱做为演示，注册好163邮箱有需要将smtp协议开启来，才能通过第三方客户端登录。

注意回到旧版操作

The screenshot shows the 163 Webmail settings interface. A red arrow points to the '设置' (Settings) button in the top navigation bar. Another red arrow points to the 'POP3/SMTP/IMAP' link in the sidebar. A third red arrow points to the '客户端授权密码' (Client Authorization Password) link in the sidebar. A fourth red arrow points to the 'POP3/SMTP/IMAP' section in the main content area. The main content area displays settings for POP3/SMTP/IMAP, including checkboxes for 'POP3/SMTP服务' and 'IMAP/SMTP服务'. A note at the bottom says '温馨提示：请使用授权码登录第三方邮件客户端' (Tip: Please log in with an authorization code for third-party email clients). Below the content are '保存' (Save) and '取消' (Cancel) buttons.

POP3协议允许电子邮件客户端下载服务器上的邮件，但是在客户端的操作（如移动邮件、标记已读等），不会反馈到服务器上，比如通过客户端收取了邮箱中的3封邮件并移动到其他文件夹，邮箱服务器上的这些邮件是没有同时被移动的。

而IMAP提供webmail 与电子邮件客户端之间的双向通信，客户端的操作都会反馈到服务器上，对邮件进行的操作，服务器上的邮件也会做相应的动作。

同时，IMAP像POP3那样提供了方便的邮件下载服务，让用户能进行离线阅读。IMAP提供的摘要浏览功能可以让你在阅读完所有的邮件到达时间、主题、发件人、大小等信息后才作出是否下载的决定。此外，IMAP 更好地支持了从多个不同设备中随时访问新邮件。

操作位置	操作内容	IMAP	POP3
收件箱	阅读、标记、移动、删除邮件等	客户端与邮箱更新同步	仅客户端内
发件箱	保存到已发送	客户端与邮箱更新同步	仅客户端内
创建文件夹	新建自定义的文件夹	客户端与邮箱更新同步	仅客户端内
草稿	保存草稿	客户端与邮箱更新同步	仅客户端内
垃圾文件夹	接收误移入垃圾文件夹的邮件	支持	不支持
广告邮件	接收被移入广告邮件夹的邮件	支持	不支持

总之，IMAP 整体上为用户带来更为便捷和可靠的体验。POP3 更易丢失邮件或多次下载相同的邮件，但 IMAP 通过邮件客户端与 webmail 之间的双向同步功能很好地避免了这些问题。

然后设置客户端授权码，点击开启这里会要求使用手机验证码。

目前均为手机向指定地方发送指定验证码

发送完成后就开启了----同时记住授权码

POP3/SMTP/IMAP

开启服务:

IMAP/SMTP服务	已开启 关闭
POP3/SMTP服务	已开启 关闭

POP3/SMTP/IMAP服务能让你在本地客户端上收发邮件, [了解更多 >](#)

温馨提示: 在第三方登录网易邮箱, 可能存在邮件泄露风险, 甚至危害Apple或其他平台账户安全

收取选项:

- 收取最近30天邮件
- 收取全部邮件

温馨提示: 收取大量邮件, 会耗费您更多的流量, 建议您选择“收取最近30天邮件”

通知提醒:

- 开启客户端删除邮件提醒
当邮件客户端大量删除邮件时, 系统会发送提醒信息

授权密码管理:

授权码是用于登录第三方邮件客户端的专用密码。
适用于登录以下服务: 您开启的服务 (例如POP3/IMAP/SMTP)、Exchange/CardDAV/CalDAV服务,

使用设备	启用时间	操作
设备1	2022.10.27	删除

[新增授权密码](#)

2.邮箱参数

参数	默认设置	说明
----	------	----

参数	默认设置	说明
MAIL_SERVER	default 'localhost'	服务器地址
MAIL_PORT	default 25	端口
MAIL_USE_TLS	default False	是否启用传输层安全 (Transport Layer Security, TLS) 协议， 默认为 False
MAIL_USE_SSL	default False	是否启用安全套接层 (Secure Sockets Layer, SSL) 协议， 默认为 False
MAIL_DEBUG	default app.debug	开启调试
MAIL_USERNAME	default None	邮箱名
MAIL_PASSWORD	default None	邮箱密码
MAIL_DEFAULT_SENDER	default None	邮件发件人， 默认为 None， 也可在 Message 对象里指定
MAIL_MAX_EMAILS	default None	邮件批量发送个数上限， 默认为 None
MAIL_SUPPRESS_SEND	default app.testing	默认为 app.testing， 如果为 True，则不会真的发送邮件， 供测试用

3. 案例

```
# -*- encoding: utf-8 -*-
from flask import Flask
from flask_mail import Mail, Message

app = Flask(__name__)

app.config.update(
    DEBUG=True,
    MAIL_SERVER='smtp.163.com', # 服务器地址
    MAIL_PORT=25, # 邮件服务器端口
    MAIL_USE_TLS=True, # 使用tls 安全协议
    MAIL_USERNAME='', # 你的邮箱
    MAIL_PASSWORD='', # 授权码
)

mail = Mail(app)

@app.route('/')
def index():
    pass
```

```
# subject 邮件标题
# sender 发送方
# recipients 接收方列表，如果多个接收方，那么是群发。
# cc 抄送列表
# bcc 密件抄送
# attachments 附件实例列表
# body 邮件文本正文
# html 网页形式的正文
msg = Message(subject="来自 flask", sender=app.config['MAIL_USERNAME'], recipients=['denhu290858879@163.com'], cc=['2908588789@qq.com'])
# 邮件内容
msg.body = "Flask test mail"
# html格式的邮件
msg.html = '<a href="http://127.0.0.1:5000/index">激活</a>'
# 添加附件 content_type对照表: http://tool.oschina.net/commons/ 附件就是发送文件
msg.attach(filename='demo/mall.py', content_type='text/html', data=open('mall.py', 'r').read())
# 发送邮件
mail.send(msg)
return "Sent Succeed"

if __name__ == '__main__':
    app.run()
```