

## 1. 学习目标

1. 理解邮件发送一些名词 邮件服务器 邮箱账号 smtp pop 邮箱发送邮件的基本流程
2. 基于smtp协议实现手动发送邮件

## 2. 概念介绍

### 1. 邮件服务器

邮局：1.接收邮件 2.发送邮件

常见的邮件服务器:163,qq,sina,sohu

### 2. 电子邮件协议

电子邮件的在网络中传输和网页一样需要遵从特定的协议，常用的电子邮件协议包括 SMTP，POP3，IMAP。其中邮件的创建和发送只需要用到 SMTP协议，所以目前对于邮件发送的学习只会涉及到SMTP协议。

SMTP 是 Simple Mail Transfer Protocol 的简称，即简单邮件传输协议。它定义了邮件客户端软件和SMTP邮件服务器之间，以及两台SMTP邮件服务器之间的通信规则。

POP3 邮件服务商专门为每个用户申请的电子邮箱提供了专门的存储空间，SMTP 服务器将接收到的邮件保存到相应用户的存储空间。用户要从邮件服务提供商提供的电子邮箱中获取自己的电子邮件，那么就需要POP3 邮件服务器来完成。全称为 Post Office Protocol，它定义了邮件客户端程序和POP3邮件服务器的通信规则。

## Django使用邮件发送

- settings配置

其实在Python中已经内置了一个smtp邮件发送模块，Django在此基础上进行了简单地封装。

首先，我们需要在项目的settings文件中配置邮件发送参数，分别如下：

# 指定发送邮件的后端模块，大多数情况下照抄

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
```

# 指定发送方的smtp服务器地址，建议使用新浪家的

```
EMAIL_HOST = 'smtp.sina.com'
```

# smtp服务端口，默认为25；

```
EMAIL_PORT = 25
```

# 你在发送服务器的用户名；

```
EMAIL_HOST_USER = 'xxx@sina.com'
```

# 对应用户的密码

```
EMAIL_HOST_PASSWORD = 'xxxxxxxxxx'
```

注意：

1. 邮件公司可能不开放smtp服务
2. 公司要求使用ssl安全机制
3. smtp服务对主机名格式有要求

- 测试邮件发送，新建send\_mail.py文件

在项目根目录下新建一个send\_mail.py文件，然后写入下面的内容：

```
import os
from django.core.mail import send_mail

os.environ['DJANGO_SETTINGS_MODULE'] = 'mysite.settings'

if __name__ == '__main__':

    send_mail(
        '这是邮件的标题',
        '这是邮件的内容!',
        'xxx@sina.cn',
        ['xxx@qq.com'],
    )
```

对于send\_mail方法，第一个参数是邮件主题subject；第二个参数是邮件具体内容；第三个参数是邮件发送方，需要和你settings中的一致；第四个参数是接受方的邮件地址列表。请按你自己实际情况修改发送方和接收方的邮箱地址。

另外，由于我们是单独运行send\_mail.py文件，所以无法使用Django环境，需要通过os模块对环境变量进行设置，也就是：os.environ['DJANGO\_SETTINGS\_MODULE'] = '项目名.settings'

- 发送Html文本

```
import os
from django.core.mail import send_mail, EmailMultiAlternatives

os.environ['DJANGO_SETTINGS_MODULE'] = 'mysite1.settings'

if __name__ == '__main__':

    subject, from_email, to = '来自的测试邮件', '18500230996@sina.cn', 'maoxinyu925@163.com'
    text_content = '欢迎访问www.baidu.com, 祝贺你收到了我的邮件, 有幸收到我的邮件说明你及其幸运'
    html_content = '<p>感谢注册<a href="http://{}/confirm/?code={}"
target=blank>www.baidu.com</a>, \欢迎你来验证你的邮箱, 验证结束你就可以登录了! </p>'
    msg = EmailMultiAlternatives(subject, text_content, from_email, [to])
    msg.attach_alternative(html_content, "text/html")
    msg.send()
```