

发送邮件

django 中内置了许多方法可以使开发者方便的进行邮件发送

邮件配置

发送邮件，首先需要在项目的 settings.py 文件下配置邮件服务器连接等信息

```
EMAIL_USE_SSL = True # Secure Sockets Layer 安全套接层，取决于邮件服务器是否开启加密协议
EMAIL_HOST = 'smtp.qq.com' # 邮件服务器地址
EMAIL_PORT = 465 # 邮件服务器端口
EMAIL_HOST_USER = 'account@qq.com' # 登陆邮件服务器的账号
EMAIL_HOST_PASSWORD = 'password' # 登陆邮件服务器的密码
DEFAULT_FROM_EMAIL = EMAIL_HOST_USER # 邮件的发送者
```

注意： EMAIL_HOST_PASSWORD 所设置的登陆邮件服务器的密码为邮件客户端后台所设置的第三方客户端登陆密码，这个值并不是直接账号密码

如：QQ 邮箱的该值密码在网页版 QQ 邮件客户端的设置 -> 账户 -> 开启 POP3/SMTP 服务 | IMAP/SMTP 服务

发送文本邮件

发送普通邮件可以使用 django.core.mail 模块下的 send_mail 函数进行

- send_mail(subject, message, from_email, recipient_list, fail_silently=False, html_message=None)

将邮件发送至 recipient_list 中的每一个收件人处

subject：发送邮件标题

message：发送邮件正文

from_email：发件人邮箱地址

recipient_list：一个字符串列表，每一个数据为接收者的邮箱地址

html_message：如果指定该值，则发送的内容类型为 text/html 为一个 html 邮件内容

- 发送普通邮件视图函数

```

from django.core.mail import send_mail
def send_email(request):
    subject = '邮件测试'
    message = '<a href="http://www.python.org">学习Python, 我很快乐</a>'
    send_mail(
        subject = subject,
        message = message,
        from_email = 'from@qq.com',
        recipient_list = ['recv@qq.com', 'recv1@qq.com']
    )
    return HttpResponse('Down')

```

发送HTML邮件

这里发送的邮件内容为一个HTML的a标签，但是真正接收者接到该邮件时，a标签并不会解释称为真正的a标签，只是一个普通的字符串，这是因为当前发送邮件的类型为text/plain，可以使用html_message参数进行HTML内容的发送

```

from django.core.mail import send_mail
def send_email(request):
    subject = '邮件测试'
    message = '<a href="http://www.python.org">学习Python, 我很快乐</a>'
    send_mail(
        subject = subject,
        message = '', # 该参数为必须参数，必须填写
        html_message = message,
        from_email = 'from@qq.com',
        recipient_list = ['recv@qq.com', 'recv1@qq.com']
    )
    return HttpResponse('Down')

```

发送html格式邮件还可以使用django.core.mail模块下的EmailMultiAlternatives类进行邮件体的构造，然后进行邮件发送

- EmailMultiAlternatives(subject='', body='', from_email=None, to=None)

subject: 邮件标题

body: 邮件内容

from_email: 邮件发送者

to: 邮件接收人列表

- EmailMultiAlternatives.attach_alternative(content, mimetype)

向EmailMultiAlternatives实例中添加mimetype支持的邮件内容

content: 添加的邮件内容

mimetype: 添加内容的mime类型

- EmailMultiAlternatives.send()

发送邮件

```

from django.core.mail import EmailMultiAlternatives
def send_email(request):
    subject = '邮件测试'
    text_message = '学习Python, 我很快乐'
    html_message = '<a href="http://www.python.org">学习Python, 我很快乐</a>'
    email = EmailMultiAlternatives(
        subject = subject,
        body = text_message,
        from_email = '1747266529@qq.com',
        to = ['recv@qq.com', 'recv1@qq.com']
    )
    email.attach_alternative(html_message, 'text/html') # 添加HTML邮件部分
    email.send() # 发送邮件
    return HttpResponse('Down')

```

发送富文本邮件

经常需要我们在邮件中加入图片等静态资源

需要用到 python 中 `email.mime.image` 模块下的 `MIMEImage` 类进行图片内容的构造

这里发送邮件使用 `EmailMessage` 类, 来自于 `django.core.mail` 模块下

并在使用对应该类实例的 `attach` 添加图片资源数据

最后使用实例的 `send` 函数发送邮件

- `EmailMessage(subject='', body='', from_email=None, to=None)`

`subject`: 邮件标题

`body`: 邮件内容

`from_email`: 邮件发送者

`to`: 邮件接收人列表

```

from sendmailpro.settings import STATICFILES_DIRS
import os
from email.mime.image import MIMEImage
from django.core.mail import EmailMessage
def send_email(request):
    subject = '图片邮件测试'
    file_1 = os.path.join(STATICFILES_DIRS[0], 'img/1.png')
    with open(file_1, 'rb') as fp:
        # 打开第一张图片
        image_1 = MIMEImage(fp.read())

    file_2 = os.path.join(STATICFILES_DIRS[0], 'img/2.png')
    with open(file_2, 'rb') as fp:
        # 打开第二张图片
        image_2 = MIMEImage(fp.read())

    body = "<img src='cid:first_id'><br><img src='cid:sec_id'>"
    # 发送邮件主体内容

```

```

image_1.add_header('Content-ID', '<%s>' % 'first_id')
# 通过 CID 对称图片在邮件内容中的位置
image_2.add_header('Content-ID', '<%s>' % 'sec_id')

message = EmailMessage( # 构建发送的邮件主体
    subject=subject,
    body=body,
    from_email='from@qq.com',
    to=['recv@qq.com', 'recv1@qq.com']
)
message.content_subtype = 'html'
message.attach(image_1) # 添加两张图片
message.attach(image_2)
message.send() # 发送邮件
return HttpResponse('Down')

```

发送附件邮件

发送附件，也可以使用 `django.core.mail` 模块下的 `EmailMessage` 类进行附件邮件体构造

通过 `EmailMessage` 实例的 `attach` | `attach_file` 两种函数在邮件主体中添加附件内容

`attach_file` 添加邮件附件直接加入路径即可，但是 `attach` 添加附件内容需要提供附件内容

- `attach(filename=None, content=None, mimetype=None)`

添加附件内容

`filename`: 附件文件名称

`content`: 附件内容

`mimetype`: 附件的 MIME 类型

- `attach_file(path, mimetype=None)`

直接通过路径添加附件

`path`: 附件路径

`mimetype`: 附件的 MIME 类型

```

from sendmailpro.settings import STATICFILES_DIRS
import os
from email.mime.image import MIMEImage
from django.core.mail import EmailMessage
def send_email(request):
    subject = '附件邮件测试'
    email = EmailMessage(
        subject=subject,
        body='这是一封具有图片附件的邮件',
        from_email='from@qq.com',
        to=['recv@qq.com', 'recv1@qq.com']
    )
    file_1 = os.path.join(STATICFILES_DIRS[0], 'img/1.png')

```

```
image_1 = open(file_1, 'rb').read()
email.attach('1.png', image_1, 'image/png') # 使用attach实例函数添加附件内容

file_2 = os.path.join(STATICFILES_DIRS[0], 'img/2.png')
email.attach_file(file_2, mimetype='image/png') # 使用attach_file实例函数添加附件路径

email.send()
return HttpResponse('Down')
```