

Python - Enviando e-mail

Um aplicativo que gerencia e entrega e-mail pela Internet é chamado de "servidor de e-mail". Simple Mail Transfer Protocol (SMTP) é um protocolo que trata do envio de e-mail e do roteamento de e-mail entre servidores de e-mail. É um padrão da Internet para transmissão de e-mail.

Python fornece o módulo `smtplib`, que define um objeto de sessão de cliente SMTP que pode ser usado para enviar e-mails para qualquer máquina da Internet com um daemon de ouvinte SMTP ou ESMTP.

Função `smtplib.SMTP()`

Para enviar um email, você precisa obter o objeto da classe SMTP com a seguinte função -

```
import smtplib

smtpObj = smtplib.SMTP( [host [, port [, local_hostname]]] )
```

Aqui estão os detalhes dos parâmetros -

- **host** - Este é o host que executa seu servidor SMTP. Você pode especificar o endereço IP do host ou um nome de domínio como `tutorialspoint.com`. Este é um argumento opcional.
- **port** - Se você estiver fornecendo o argumento do host, será necessário especificar uma porta onde o servidor SMTP está escutando. Normalmente esta porta seria 25.
- **local_hostname** - Se o seu servidor SMTP estiver em execução na sua máquina local, você poderá especificar apenas `localhost` como opção.

O objeto SMTP possui os seguintes métodos -

- **connect(host, port, source_address)** - Este método estabelece conexão com um host em uma determinada porta.
- **login(user, password)** - Faça login em um servidor SMTP que requer autenticação.
- **quit()** - encerra a sessão SMTP.
- **data(msg)** - envia dados da mensagem ao servidor.
- **docmd(cmd, args)** - envia um comando e retorna seu código de resposta.
- **ehlo(name)** - Nome do host para se identificar.
- **starttls()** - coloca a conexão com o servidor SMTP no modo TLS.

- **getreply()** - obtém uma resposta do servidor que consiste no código de resposta do servidor.
- **putcmd(cmd, args)** - envia um comando ao servidor.
- **send_message(msg, from_addr, to_addrs)** - converte a mensagem em um bytestring e a passa para o sendmail.

O módulo smtpd

O módulo **smtpd** que vem pré-instalado com Python possui um servidor de depuração SMTP local. Você pode testar a funcionalidade do e-mail iniciando-o. Na verdade, ele não envia e-mails para o endereço especificado, ele os descarta e imprime seu conteúdo no console. Executar um servidor de depuração local significa que não é necessário lidar com a criptografia de mensagens ou usar credenciais para fazer login em um servidor de e-mail.

Você pode iniciar um servidor de depuração SMTP local digitando o seguinte no prompt de comando -

```
python -m smtpd -c DebuggingServer -n localhost:1025
```

Exemplo

O programa a seguir envia um e-mail fictício com a ajuda da funcionalidade smtpplib.

```
import smtpplib

def prompt(prompt):
    return input(prompt).strip()

fromaddr = prompt("From: ")
toaddrs = prompt("To: ").split()
print("Enter message, end with ^D (Unix) or ^Z (Windows):")

# Add the From: and To: headers at the start!
msg = ("From: %s\r\nTo: %s\r\n\r\n"
       % (fromaddr, ", ".join(toaddrs)))
while True:
    try:
        line = input()
    except EOFError:
        break
    if not line:
        break
    msg = msg + line

print("Message length is", len(msg))
```

```
server = smtplib.SMTP('localhost', 1025)
server.set_debuglevel(1)
server.sendmail(fromaddr, toaddrs, msg)
server.quit()
```

Basicamente usamos o método `sendmail()`, especificando três parâmetros -

- **The sender** - Uma string com o endereço do remetente.
- **The receivers** - Uma lista de strings, uma para cada destinatário.
- **The message** - Uma mensagem como uma string formatada conforme especificado nos vários RFCs.

Já iniciamos o servidor de depuração SMTP. Execute este programa. O usuário é solicitado a inserir o ID do remetente, os destinatários e a mensagem.

```
python example.py
From: abc@xyz.com
To: xyz@abc.com
Enter message, end with ^D (Unix) or ^Z (Windows):
Hello World
^Z
```

O console reflete o seguinte log -

```
From: abc@xyz.com
reply: retcode (250); Msg: b'OK'
send: 'rcpt TO:<xyz@abc.com>\r\n'
reply: b'250 OK\r\n'
reply: retcode (250); Msg: b'OK'
send: 'data\r\n'
reply: b'354 End data with <CR><LF>.<CR><LF>\r\n'
reply: retcode (354); Msg: b'End data with <CR><LF>.<CR><LF>'
data: (354, b'End data with <CR><LF>.<CR><LF>')
send: b'From: abc@xyz.com\r\nTo: xyz@abc.com\r\n\r\nHello
World\r\n.\r\n'
reply: b'250 OK\r\n'
reply: retcode (250); Msg: b'OK'
data: (250, b'OK')
send: 'quit\r\n'
reply: b'221 Bye\r\n'
reply: retcode (221); Msg: b'Bye'
```

O terminal no qual o servidor SMTPD está sendo executado mostra esta **saída** -

```
----- MESSAGE FOLLOWS -----  
b'From: abc@xyz.com'  
b'To: xyz@abc.com'  
b'X-Peer: ::1'  
b''  
b'Hello World'  
----- END MESSAGE -----
```

Usando SMTP do Gmail

Vejamos o script abaixo, que usa o servidor de e-mail **smtp** do Google para enviar uma mensagem de e-mail.

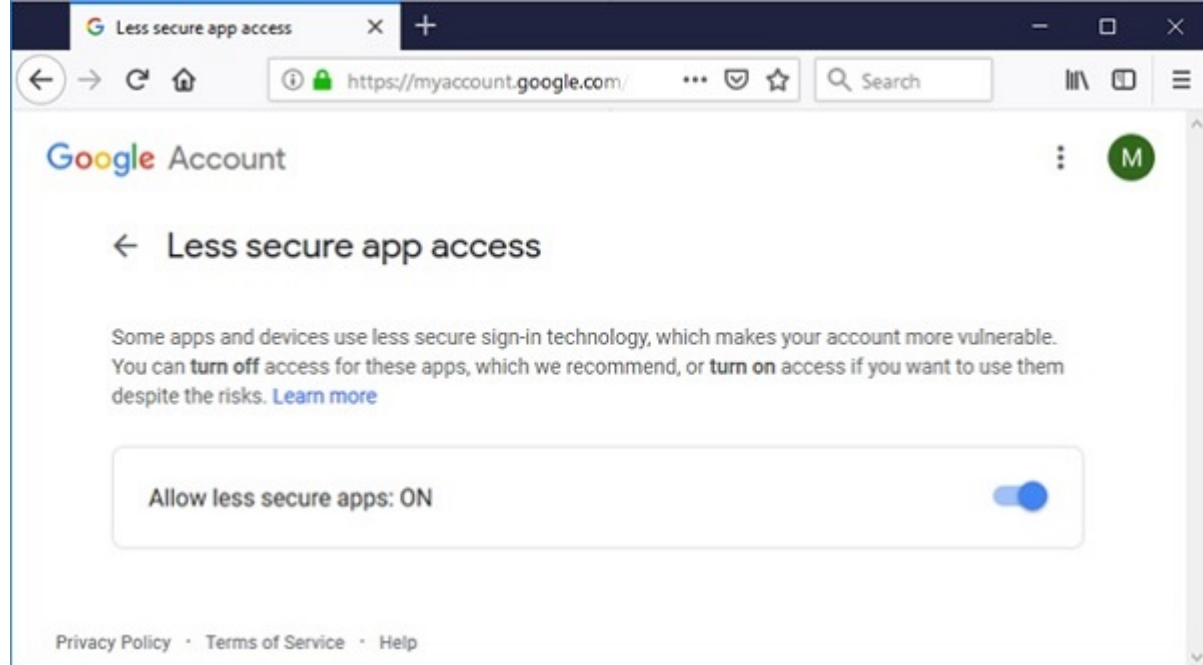
Primeiro de tudo, o objeto SMTP é configurado usando o servidor SMTP do Gmail e a porta 527. O objeto SMTP então se identifica invocando o comando ehlo(). Também ativamos o Transport Layer Security para a mensagem de correio de saída.

Em seguida, o comando login() é invocado passando credenciais como argumentos para ele. Finalmente, a mensagem de correio é montada anexando-lhe um cabeçalho no formato prescrito e é enviada usando o método sendmail(). O objeto SMTP é fechado posteriormente.

```
import smtplib  
content="Hello World"  
mail=smtplib.SMTP('smtp.gmail.com', 587)  
mail.ehlo()  
mail.starttls()  
sender='mvl@gmail.com'  
recipient='tester@gmail.com'  
mail.login('mvl@gmail.com', '*****')  
header='To: '+recipient+'\n'+ 'From: ' \  
+sender+'\n'+ 'subject:testmail\n'  
content=header+content  
mail.sendmail(sender, recipient, content)  
mail.close()
```

Antes de executar o script acima, a conta do Gmail do remetente deve ser configurada para permitir 'aplicativos menos seguros'. Visite o seguinte link.

<https://myaccount.google.com/lesssecureapps> Defina o botão de alternância mostrado como LIGADO.



Se tudo correr bem, execute o script acima. A mensagem deve ser entregue na caixa de entrada do destinatário.