

Mails in Python senden

Dominik Pataky

7. Januar 2018

Python-Kurs 2017

1. Grundlagen: Mails senden

Das Modul `smtplib`

2. Komplexere Mails senden

Das Modul `email`

Die folgenden Folien enthalten eine praktische Anleitung zum Senden von Mails in Python.

Grundlagen: Mails senden

Das Modul `smtplib` definiert eine SMTP*-Client Session, die genutzt werden kann, um von jedem beliebigen, internetfähigen Gerät E-Mails zu verschicken.

*SMTP steht für **S**imple **M**ail **T**ransfer **P**rotocol und ist das Standard-Protokoll zum E-Mail Versand.

Verbindung zum Server

Die `smtplib` kann sich zu einem SMTP-Server verbinden

```
1 smtplib.SMTP(host='', port=0, local_hostname=None, [timeout, ]  
  source_address=None)
```

Der Server kann zum einen als einheitlicher **String** (inkl. Port) oder einzeln als **host** und **port** angegeben werden.

Login auf dem Server

Heutzutage arbeiten die meisten SMTP-Server mit **TLS**, um eine sichere Verbindung zu gewährleisten. Diese muss mithilfe der Methode `starttls()` hergestellt werden.

Der eigentliche Login erfolgt im Anschluss durch:

```
SMTP.login(user, password, *, initial_response_ok=True)
```

Der Parameter `initial_response_ok` kann in unserem Fall vernachlässigt werden.

Senden der Mail

Das tatsächliche Versenden der Mail funktioniert dann mit folgender Methode:

```
SMTP.sendmail(from_addr, to_addrs, msg, mail_options=[],  
              rcpt_options=[])
```

Jedoch lässt sich das ganze auch mit einem **MIME**-Objekt vereinfachen, auf das später noch eingegangen wird. Dieses wird mit `send_message()` versendet.

Zum Schluss darf nicht vergessen werden, die Verbindung zum SMTP-Server wieder zu schließen.

Dies geschieht mit der Methode `quit()` oder man stellt die Verbindung mithilfe eines *Filehandlers* her.

Die vollständige Serverkommunikation

```
1 import smtplib
2
3 def main():
4     message = buildmessage()
5     server = 'Servername'
6     port = 587 # This is the default Port of an smtp server
7
8     try:
9         with smtplib.SMTP(host=server, port=port) as smtpObj:
10             smtpObj.starttls()
11             smtpObj.login('user', 'pass')
12             smtpObj.send_message(message)
13     except SMTPException:
14         print("Error: unable to send email.")
15
16 if __name__ == '__main__':
17     main()
```

Auf `buildmessage()` wird im folgenden Teil eingegangen.

Komplexere Mails senden

Um mehr Möglichkeiten zur Gestaltung der E-Mail zu haben, lohnt sich die Verwendung des Moduls `email`.

Mit `email.mime` lassen sich Emails individuell bauen und zusammensetzen. Außerdem kann man mehrteilige Mails und Mails mit Anhängen (z.B. Bildern) erstellen.

Die Klasse `MIMEText`

Die Klasse `email.mime.text.MIMEText()` erstellt ein MIME Objekt, welches hauptsächlich aus Text besteht und einfach dem SMTP-Objekt übergeben werden kann:

```
email.mime.text.MIMEText(_text, _subtype='plain', _charset=None)
```

_text Ein String, der den Inhalt der Nachricht enthält.

_subtype Der Untertyp des Objekts, per default `plain`

charset Der Zeichensatz, der zur Kodierung der Zeichen verwendet werden soll. Standardmäßig `us-ascii` oder `utf8`, abhängig von dem eingegebenen Text.

Die Meta-Felder

Wenn das `MIMEText` Objekt instanziiert ist, muss dieses mit weiteren Informationen ergänzt werden:

```
1  # set the sender of your mail
2  message['From'] = 'Sender name <sender@example.com>'
3  # set the receiver of your mail
4  message['To'] = 'Receiver name <receiver@example.com>'
5
6  message['Subject'] = 'Betreff der E-Mail'
```

From Daten des Absenders (nur Mail oder Name und Mail)

To Daten des Empfängers (nur Mail oder Name und Mail)

Subject Betreff der Mail

Die Message lässt sich außerdem noch um einen **Cc** oder einen **Bcc** erweitern.

Das fertige Mail-Skript

```
1 from email.mime.text import MIMEText
2
3
4 def buildmessage():
5     message = MIMEText('Text der E-Mail.')
6
7     # set the sender of your mail
8     message['From'] = 'Sender name <sender@example.com>'
9     # set the receiver of your mail
10    message['To'] = 'Receiver name <receiver@example.com>'
11
12    message['Subject'] = 'Betreff der E-Mail'
13
14    return message
```