GABRIELE ROMANATO

Come creare un server SMTP di base con Python

Un server SMTP (“Simple Mail Transfer Protocol”) è un componente fondamentale per l’invio di email. Python, grazie alla sua libreria standard, rende la creazione di un server SMTP di base un processo semplice e veloce. In questo articolo, vedremo come utilizzare il modulo `smtpd` e `asyncio` di Python per creare un server SMTP di base.

Il modulo `smtpd` è utilizzato per gestire il protocollo SMTP, mentre `asyncio` è usato per creare un loop di eventi asincroni.

```
import asyncio
from aiosmtpd.controller import Controller
```

La classe handler è responsabile della gestione delle email ricevute dal server.

```
class EmailHandler:
    async def handle_DATA(self, server, session, envelope):
        print(f"Messaggio ricevuto da: {envelope.mail_from}")
        print(f"Destinatari: {envelope.rcpt_tos}")
        print(f"Contenuto del
messaggio:\n{envelope.content.decode('utf8',
errors='replace')}")
        return '250 Message accepted for delivery'
```

In questo esempio, il metodo `handle_DATA` stampa i dettagli del messaggio ricevuto, inclusi mittente, destinatari e contenuto.

Con il modulo `aiosmtpd`, possiamo avviare un server SMTP utilizzando la classe `Controller`.

```
def start_server():
    handler = EmailHandler()
    controller = Controller(handler, hostname='127.0.0.1',
port=1025)
    controller.start()
    print("Server SMTP in esecuzione su 127.0.0.1:1025")

if __name__ == "__main__":
    start_server()
    try:
        asyncio.get_event_loop().run_forever()
    except KeyboardInterrupt:
        print("\nServer SMTP arrestato.")
```

Questo script avvia un server SMTP che ascolta sulla porta 1025 e utilizza l'host locale (127.0.0.1).

Puoi testare il tuo server SMTP utilizzando strumenti come telnet, curl o uno script Python. Ecco un esempio di script per inviare una mail al server SMTP:

```
import smtplib

server = smtplib.SMTP('127.0.0.1', 1025)
server.sendmail(
    'mittente@example.com',
    ['destinatario@example.com'],
    "Subject: Test\n\nQuesto è un messaggio di test."
)
server.quit()
```

Conclusione

In questo articolo, abbiamo visto come creare un server SMTP di base con Python utilizzando il modulo `aiosmtpd`. Questo approccio è utile per testare applicazioni email in locale o imparare di più sul protocollo SMTP. Con alcune modifiche, puoi espandere le funzionalità del server e adattarlo alle tue esigenze specifiche.