

Report on email_sender.py

The `email_sender.py` script contains a function `send_email()` that sends an email using the Simple Mail Transfer Protocol (SMTP) with the help of the `smtplib` library in Python.

Code Analysis:

The script imports necessary modules:

- `smtplib` for handling email sending via SMTP.
- `MIMEMultipart` and `MIMEText` from `email.mime` for composing the email message with attachments.

The `send_email()` function:

- **Parameters:**
 - `subject`: Subject of the email.
 - `message`: Body of the email.
 - `from_address`: Sender's email address.
 - `to_address`: Recipient's email address.
 - `smtp_server`: SMTP server address.
 - `smtp_port`: Port number for the SMTP server.
 - `smtp_username`: Username for SMTP authentication.
 - `smtp_password`: Password for SMTP authentication.
- Constructs the email message with the provided subject, sender, and recipient.
- Attaches the message body to the email.
- Connects to the SMTP server, initiates TLS encryption, and logs in using the provided credentials.
- Converts the message to a string format and sends the email to the recipient.
- Finally, closes the connection to the SMTP server.

Execution:

The `send_email()` function is called with sample parameters:

- Subject: 'Subject Here'
- Message: 'Message Here'
- Sender's Email: 'your-email@example.com'
- Recipient's Email: 'receiver-email@example.com'
- SMTP Server: 'smtp.example.com'
- SMTP Port: 587
- SMTP Username: 'your-smtp-username'
- SMTP Password: 'your-smtp-password'
- This execution sends an email with the specified details.

The script demonstrates a simple yet effective way to send emails programmatically in Python using SMTP. It can be further extended and customized as per specific requirements.