ReportFile

February 1, 2023

1 Report Automating File Transfer

1.1 1. Use the ftplib library to connect to the external FTP server and list the files in the directory.

ftplib.FTP() initializes a FTP object. Then, we connect to existed server and log in it.

[ ]:
```
# Connect to FTP server
# Create a FTP server "python3 -m pyftpdlib -w --user=username␣
↪--password=password"

# FTP server details
ftp_host = '127.0.0.1'
ftp_user = 'username'
ftp_password = 'password'

ftp = ftplib.FTP()
ftp.connect(ftp_host, 2121)
ftp.login(ftp_user, ftp_password)
```

1.2 2. Use the os library to check for the existence of a local directory where the files will be stored.

os.path.exists(local_dir) returns true:false ? if file exists:if file does not exist. os.makedirs(local_dir) creates a folder.

[ ]:
```
# Check if local directory exists, create if not
if not os.path.exists(local_dir):
    os.makedirs(local_dir)
```

1.3 3. Use a for loop to iterate through the files on the FTP server and download them to the local directory using the ftplib.retrbinary() method.

ftp.nlst() returns a list of file names in the directory specified by the argument. Then it iterates through these files, and move to local directory and remain current files in the origin folder. The method retrbinary() of the FTP class retrieves a file from the server to the local system in binary mode. The method accepts a retrieval command such as RETR, and a callback function along with other paramaters.

[ ]:
```
# List files
files = ftp.nlst()

for file in files:
    with open(local_dir + '/' + file, 'wb') as f:
        ftp.retrbinary("RETR " + file, f.write)
```

1.4 4. Use the shutil library to move the files from the local directory to the internal network.

Open local directory and take a list of files, then it iterates through these files and move them to internal directory. shutil.move() is a method, which recursively moves a file or directory (source) to another location (destination) and returns the destination.

[ ]:
```
# Move files from local directory to internal network
for file in os.listdir(local_dir):
    shutil.move(local_dir + '\\' + file, internal_dir)
```

1.5 5. Use the schedule library to schedule the script to run daily at a specific time.

The approach is the same as in automating email sending*

[ ]:
```
# Set up schedule
schedule.every().day.at("10:00").do(transfer_files)
while True:
    schedule.run_pending()
    time.sleep(1)
```

1.6 6. You can also set up a log file to keep track of the files that have been transferred and any errors that may have occurred during the transfer process.

1.6.1 Log tracking

The approach is the same as in automating email sending*

[ ]:
```
 # open log file
with open(str(os.getcwd()) + "\\log.json", 'r') as log:
    data = json.load(log)

for file in files:
    with open(local_dir + '/' + file, 'wb') as f:
        ftp.retrbinary("RETR " + file, f.write)
    data['Fails'].append(str(file))
json_object = json.dumps(data)

# Write sent files to log file
with open(str(os.getcwd()) + "\\log.json", 'w') as log:
    log.write(json_object)
```

1.6.2 Error tracking

The approach is the same as in automating email sending*

[ ]:
```
try:
    # Set up schedule
    schedule.every().day.at("10:00").do(transfer_files)
    while True:
        schedule.run_pending()
        time.sleep(1)
except Exception as e:  # Catch errors in the code

    # Get current time
    now = datetime.now()
    dt_string = now.strftime("%d/%m/%Y %H:%M:%S")

    # Write an error with the time and description
    with open('error.txt', 'w') as f:
        f.write(f'''The error is occurred at {dt_string}.
           The reason is {e}''')
```