ABB WAQT HAI
AGEY BARHNE KA

الخدمت
ALKHIDMAT

بنو قابل

اپنی دنیا آپ پیدا کر

THE LARGEST
SCHOLARSHIP PROGRAM
FOR KARACHI'S YOUTH

Python Scripts Collection For File Management

Umer Saeed

BSc Electrical Engineering, MS Data Science
Department of Information System
School of Business and Economics
University of Management & Technology, Lahore, Pakistan
Sr. RF Optimization and Planning Engineer
umersaeed1981@gmail.com

Clean up Files and Directories in a specified Folder Path

- The purpose of this code is to;

  - **Clear Files and Directories:** It aims to remove all files and directories within a specified folder path.
  - **Prevent Errors**: It checks if the folder path exists before attempting any operations to prevent potential errors.
  - **Recursive Removal:** If a directory is found, it removes it along with all its contents recursively.
  - **Clean-Up for Analysis:** This code might be used as a preparatory step before conducting analysis on a dataset, ensuring a clean slate for new data.

Clean up Files and Directories in a Specified Folder Path

```python
# import required Libraries
import os
import shutil

# folder path
folder_path = "D:/abc/def/xyz"

# Ensure the folder path exists before proceeding
if os.path.exists(folder_path):
    # Iterate over the contents of the folder
    for item in os.listdir(folder_path):
        item_path = os.path.join(folder_path, item)

        # If it's a file, remove it
        if os.path.isfile(item_path):
            os.remove(item_path)

        # If it's a directory, remove it and its contents recursively
        elif os.path.isdir(item_path):
            shutil.rmtree(item_path)
else:
    print(f"The folder path '{flder_path}' does not exist.")
```

Batch Clearing of Files and Folders in Multiple Paths

- The purpose of this code is to;

  - **Iterate Through Multiple Folder Paths:** It contains a list of folder paths, and it iterates through each path to perform the file and folder deletion operation.
  - **Ensure Path Existence:** Before attempting any operations, it checks if the specified folder path exists to avoid potential errors.
  - **Recursively Delete Directories:** If a directory is encountered, it removes the directory along with all its contents in a recursive manner.
  - **Provide Feedback for Nonexistent Paths:** If a specified folder path does not exist, it prints a message indicating this.

Batch Clearing of Files and Folders in Multiple Paths

```python
# import libraries
import os
import shutil
# user define funcations
def delete_all_files_and_folders(folder_path):
    # Ensure the folder path exists before proceeding
    if os.path.exists(folder_path):
        # Iterate over the contents of the folder
        for item in os.listdir(folder_path):
            item_path = os.path.join(folder_path, item)

            # If it's a file, remove it
            if os.path.isfile(item_path):
                os.remove(item_path)

            # If it's a directory, remove it and its contents recursively
            elif os.path.isdir(item_path):
                shutil.rmtree(item_path)
    else:
        print(f"The folder path '{folder_path}' does not exist.")
```

Batch Clearing of Files and Folders in Multiple Paths

```
# List of folder paths
folder_paths = ["D:/Advance_Data_Sets/License/Target",
                "D:/Advance_Data_Sets/Output_Folder"]

# Delete all files and folders in each path
for path in folder_paths:
    delete_all_files_and_folders(path)
```

Batch File Transfer Based on Excel File Information

- The purpose of this code is to;
  - **Set Working Directory:** It sets the current working directory to a specified path.
  - **Read Excel File:** It reads an Excel file named 'Files_Info.xlsx' that contains information about source and target file paths.
  - **Iterate Through Excel Rows:** It iterates through the rows of the DataFrame obtained from the Excel file.

Batch File Transfer Based on Excel File Information

- **Move Files:** For each row, it checks if the source file exists. If it does, it creates the target directory if it doesn't exist and then moves the file.

- **Provide Feedback for Nonexistent Files:** If a source file doesn't exist, it prints a message indicating this.

- **Completion Message:** After processing all rows, it prints a message indicating that the file transfer process is complete.

Batch File Transfer Based on Excel File Information

- **Files_info.xlsx:** Format of the mentioned File is given below;

File	sr_Path	target_Path
UMTS_CE_Utilization_31082023.zip	D:/Advance_Data_Sets/Congestion/CE_Utilization	D:/Advance_Data_Sets/Congestion/CE_Utilization/08_Aug_2023
GSM_Cell_Houlry_31082023.zip	D:/Advance_Data_Sets/KPIs_Analysis/Hourly_KPIs_Cell_Level/GSM	D:/Advance_Data_Sets/KPIs_Analysis/Hourly_KPIs_Cell_Level/GSM/Aug
LTE_Cell_Hourly_31082023.zip	D:/Advance_Data_Sets/KPIs_Analysi/Hourly_KPIs_Cell_Level/LTE	D:/Advance_Data_Sets/KPIs_Analysis/Hourly_KPIs_Cell_Level/LTE/08_Aug2023
UMTS_Cell_Hourly_31082023.zip	D:/Advance_Data_Sets/KPIs_Analysis/Hourly_KPIs_Cell_Level/UMTS	D:/Advance_Data_Sets/KPIs_Analysis/Hourly_KPIs_Cell_Level/UMTS/08_Aug2023
GSM_BTS_Hourly_31082023.zip	D:/Advance_Data_Sets/TXN_DataSets/GSM_BTS_Hourly	D:/Advance_Data_Sets/TXN_DataSets/GSM_BTS_Hourly/08_Aug
LTE_Hourly_BTS_Level_31082023.zip	D:/Advance_Data_Sets/TXN_DataSets/LTE_TNL_BTS_Hourly	D:/Advance_Data_Sets/TXN_DataSets/LTE_TNL_BTS_Hourly/08_Aug2023
LTE_Nationwide_RX_31082023.zip	D:/Advance_Data_Sets/TXN_DataSets/LTE_Transmission_Link_Capacity	D:/Advance_Data_Sets/TXN_DataSets/LTE_Transmission_Link_Capacity/08_Aug
UMTS_BTS_Hourly_31082023.zip	D:/Advance_Data_Sets/TXN_DataSets/UMTS_Hourly_BTS	D:/Advance_Data_Sets/TXN_DataSets/UMTS_Hourly_BTS/08_Aug
IPPath_Hourly_31082023.zip	D:/Advance_Data_Sets/TXN_DataSets/UMTS_Transmission_Hourly/IPPATH_Hourly	D:/Advance_Data_Sets/TXN_DataSets/UMTS_Transmission_Hourly/IPPATH_Hourly/08_Aug2023
IPPool_Hourly_31082023.zip	D:/Advance_Data_Sets/TXN_DataSets/UMTS_Transmission_Hourly/IPPOOL_Hourly	D:/Advance_Data_Sets/TXN_DataSets/UMTS_Transmission_Hourly/IPPOOL_Hourly/08_Aug2023

Batch File Transfer Based on Excel File Information

```python
# import libraries
import os
import shutil
import pandas as pd
#set the Path
path = 'D:/Advance_Data_Sets/File_Paths'
os.chdir(path)
# Read the Excel file
df = pd.read_excel('Files_Info.xlsx')
# Iterate through the rows
for index, row in df.iterrows():
    source_file = os.path.join(row['sr_Path'], row['File'])
    target_path = row['target_Path']

    # Check if the source file exists
    if os.path.exists(source_file):
        # Create the target directory if it doesn't exist
        os.makedirs(target_path, exist_ok=True)

        # Move the file
        shutil.move(source_file, target_path)
    else:
        print(f"The file '{source_file}' does not exist.")

print("File transfer complete.")
```

Batch File Deletion Based on Excel File Information

- The purpose of this code is to;

  - **Set Working Directory:** It sets the current working directory to a specified path.
  - **Read Excel File:** It reads an Excel file named 'DA_Files.xlsx' that contains information about the files to be deleted.
  - **Iterate Through Excel Rows:** It iterates through the rows of the DataFrame obtained from the Excel file.
  - **Delete Files:** For each row, it checks if the specified file exists. If it does, it deletes the file and provides a success message. If it doesn't exist, it prints a message indicating this.

Batch File Deletion Based on Excel File Information

- **Completion Message:** After processing all rows, it provides a summary of the deletion process.
- **DA_Files.xlsx:** Format of the mentioned File is given below;

File
D:/Advance_Data_Sets/HW_Scripts/LTE_Main_Diversity/LTE_Main_Diversity_07092023.zip
D:/Advance_Data_Sets/HW_Scripts/UMTS_Main_Diversity/UMTS_Main_Diversity_07092023.zip
D:/Advance_Data_Sets/KPIs_Analysis/BH_KPIs_Cell_Level/GSM/GSM_Cell_BH_07092023.zip
D:/Advance_Data_Sets/KPIs_Analysis/BH_KPIs_Cell_Level/UMTS/UMTS_Cell_BH_07092023.zip
D:/Advance_Data_Sets/KPIs_Analysis/DA_KPIs_Cell_Level/GSM/GSM_Cell_DA_07092023.zip
D:/Advance_Data_Sets/KPIs_Analysis/DA_KPIs_Cell_Level/LTE/LTE_Cell_DA_07092023.zip
D:/Advance_Data_Sets/KPIs_Analysis/DA_KPIs_Cell_Level/UMTS/UMTS_Cell_DA_07092023.zip
D:/Advance_Data_Sets/KPIs_Analysis/Inter_BSC_HSR/KPIs/INTER_BSC_HSR_07092023.zip
D:/Advance_Data_Sets/SLA/Cluster_DA_KPIs/GSM_Cluster_DA_07092023.zip
D:/Advance_Data_Sets/SLA/Cluster_BH_KPIs/GSM_Cluster_BH_07092023.zip
D:/Advance_Data_Sets/TA_Analysis/GSM/DA/TA_Analysis_DA_07092023.zip
D:/Advance_Data_Sets/TA_Analysis/GSM/Hourly/TA_Analysis_Hourly_07092023.zip
D:/Advance_Data_Sets/TXN_DataSets/GSM_BTS_DA/GSM_BTS_DA_07092023.zip
D:/Advance_Data_Sets/TXN_DataSets/LTE_TNL_BTS_DA/LTE_BTS_DA_07092023.zip
D:/Advance_Data_Sets/TXN_DataSets/UMTS_DA_BTS/UMTS_BTS_DA_07092023.zip
D:/Advance_Data_Sets/TXN_DataSets/UMTS_Transmission_DA/IPPATH_DA/IPPath_DA_07092023.zip
D:/Advance_Data_Sets/TXN_DataSets/UMTS_Transmission_DA/IPPOOL_DA/IPPool_DA_07092023.zip

Batch File Deletion Based on Excel File Information

```python
# import libraries
import os
import shutil
import pandas as pd
#set the Path
path = 'D:/Advance_Data_Sets/File_Paths'
os.chdir(path)
# Read the Excel file
df = pd.read_excel('DA_Files.xlsx')
# Iterate through the rows and delete the files
for index, row in df.iterrows():
    file_path = row['File']
    if os.path.exists(file_path):
        os.remove(file_path)
        print(f"{file_path} deleted successfully")
    else:
        print(f"{file_path} does not exist")
```

Batch File Transfer Based on Excel Information

- The purpose of this code is to;

  - **Set Working Directories:** It sets two different working directories, first to read an Excel file named 'p_name.xlsx' and then to perform file operations in the directory specified by 'E:/test_path'.
  - **Read Excel File:** It reads an Excel file named 'p_name.xlsx' which contains information about source and target folders.
  - **Iterate Through Excel Rows:** It iterates through the rows of the DataFrame obtained from the Excel file.
  - **Transfer Files:** For each row, it looks for files in the source folder that have names starting with the value in the 'File' column. When found, it moves these files to the corresponding target folder.

Batch File Transfer Based on Excel Information

- The purpose of this code is to;

- **Note:** The code assumes that the target folder structure is similar to the source folder structure and that files with specific names should be moved from source to target
- **p_name.xlsx:** Format of the mentioned File is given below;

File	Target_Folder	Sr_Folder
UMTS_CE_Utilization_	D:/Advance_Data_Sets/Congestion/CE_Utilization	E:/test_path
LTE_Main_Diversity_	D:/Advance_Data_Sets/HW_Scripts/LTE_Main_Diversity	E:/test_path
UMTS_Main_Diversity_	D:/Advance_Data_Sets/HW_Scripts/UMTS_Main_Diversity	E:/test_path
GSM_Cell_BH_	D:/Advance_Data_Sets/KPIs_Analysis/BH_KPIs_Cell_Level/GSM	E:/test_path
UMTS_Cell_BH_	D:/Advance_Data_Sets/KPIs_Analysis/BH_KPIs_Cell_Level/UMTS	E:/test_path
GSM_Cell_DA_	D:/Advance_Data_Sets/KPIs_Analysis/DA_KPIs_Cell_Level/GSM	E:/test_path
LTE_Cell_DA_	D:/Advance_Data_Sets/KPIs_Analysis/DA_KPIs_Cell_Level/LTE	E:/test_path

Batch File Transfer Based on Excel Information

```python
# import libraires
import os
import shutil
import pandas as pd
#set the Path
path = 'D:/Advance_Data_Sets/File_Paths'
os.chdir(path)
# Read the Excel file
df = pd.read_excel('p_name.xlsx')
#set the Path (Path must be same format)
path = 'E:/test_path'
os.chdir(path)
# Iterate through the rows in the DataFrame
for index, row in df.iterrows():
    source_folder = row['Sr_Folder']
    target_folder = row['Target_Folder']

    for file_name in os.listdir(source_folder):
        if file_name.startswith(row['File']):
            source_path = os.path.join(source_folder, file_name)
            target_path = os.path.join(target_folder, file_name)
            shutil.move(source_path, target_path)
```