< Back

Congrats Daniel! This project has been marked as completed.

Project Rating

★★★☆☆

Teacher's Comment

"Good"

-

Was this helpful?

Community Link

Publish to Community

Edit Your Project

Last Submitted

Previous Submissions

6th Jan 2024 Open Link

Start Project

Submit Your Project

Learn how to submit your project ▶

Paste your project URL

Submit Project

Class Summary

This project is based on your last class PRO-C103

View Class Summary

PRO-C103: WATCH FILE SYSTEM EVENTS Completed

In Class 103, We Learned About Files System Events And How To Monitor Those At Operating System Level. You Automated Folder/Files Movement As Soon As The File Is Downloaded, And Segregated It Into Different Folders Based On The Type Of The File By Tracking The File Creation Event. In This Project You Will Write A Program To Track Different File System Events.

Goal of the Project:

In Class 103, we learned about files system events and how to monitor those at operating system level. You automated folder/files movement as soon as the file is downloaded, and segregated it into different folders based on the type of the file by tracking the file creation event. In this project you will write a program to track different file system events.

This project is a continuation of project 102, Make sure to finish that before attempting this one.

Story:

Have wondered how your system notifies you what is happening to the files in your system?

Operating Systems like Windows keep track of all the files present in the system. This helps to track malicious files like viruses which get downloaded through the net. Viruses modify the existing files or delete important files from the system to corrupt the system! Automating the process to track the changes in the system helps to track if such activities have happened or not!

***This is just for your reference. We expect you to apply your own creativity to the project.**

Getting Started:

1. Create a folder named **Project102** in VSC.

2. Create a new file named **file_system_events_tracker.py**.

3. **Document Ref(watchdog events) :**
   https://pythonhosted.org/watchdog/api.html#module-watchdog.events

Specific Tasks to complete the Project:

1. Import Libraries:
   - import sys
   - import time

Ask a doubt to your tea

HELP

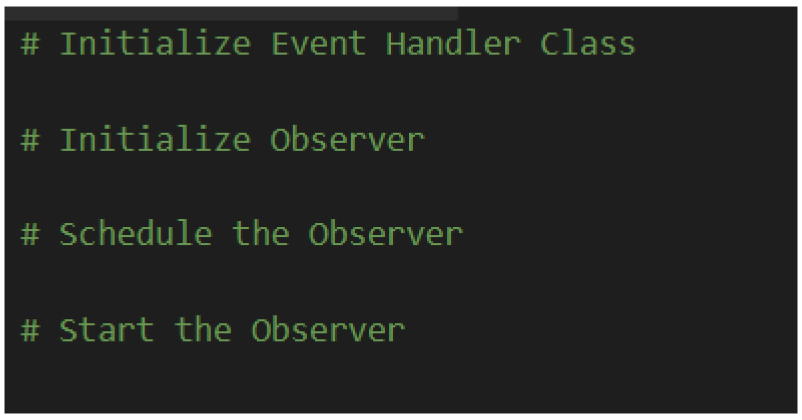
- import random
- import os
- import shutil
- from watchdog.observers import Observer
- from watchdog.events import FileSystemEventHandler

2. Set the path for the directory to track changes

   from_dir = “<Set path for tracking file system events>”

3. Write a **FileEventHandler** Class with following methods to track file system eventsb (See Hint 1):
   - **on_created():** Called when a file or a directory is created.
   - **on_modified():** Called when a file or directory is modified.
   - **on_moved():** Called when a file or a directory is moved or renamed.
   - **on_deleted():** Called when a file or directory is deleted.

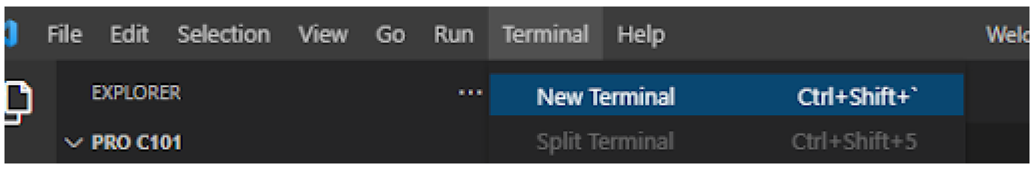
4. Set the **observer** to watch the changes:



```
# Initialize Event Handler Class

# Initialize Observer

# Schedule the Observer

# Start the Observer
```

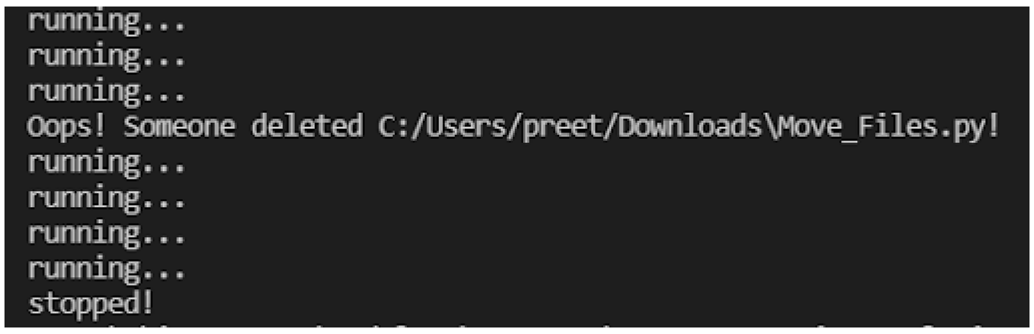
5. Add the code to stop the **observer** program when any key is pressedb (See Hint 2).

6. Run the code using Terminal.



- Run the file using **python file_system_events_tracker.py / py file_system_events_tracker.py**

7. Try to delete some file from the path set for tracking in code and see the event details:



```
running...
running...
running...
Oops! Someone deleted C:/Users/preet/Downloads\Move_Files.py!
running...
running...
running...
running...
stopped!
```

Note: Try out with different file events too.

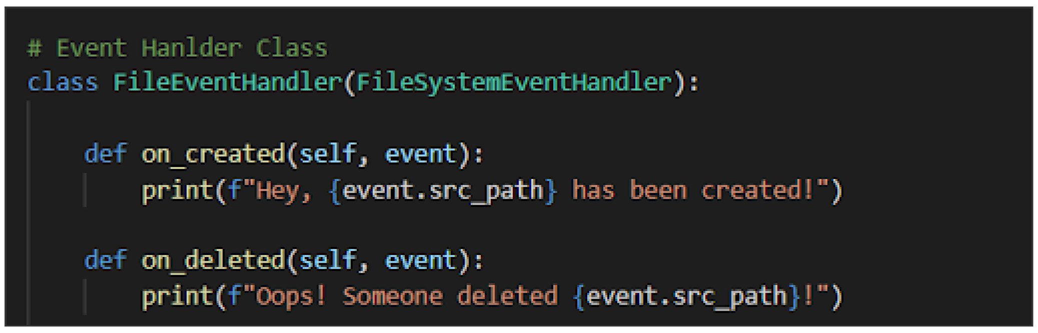
Submitting the Project:

1. **SAVE** all the changes made to the project.
2. Click on **"Run"** once to check if it is working.
3. Open the GitHub create a repository named **Project103**
4. Upload a file **Move_File.py** and click **Commit Changes**
5. Copy this link and submit it in the Student Dashboard Projects panel against the correct class number.

Hints:

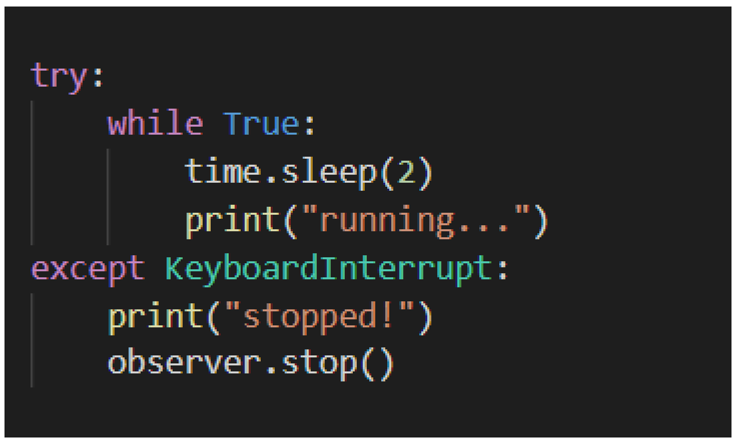
1. Writing **FileEventHandler** Class:
   Note: The below hint only shows two methods, **on_created()** and **on_deleted()**. Other

two methods **on_modified()** and **on_moved()** can be written in the similar way.



```
# Event Hanlder Class
class FileEventHandler(FileSystemEventHandler):

    def on_created(self, event):
        print(f"Hey, {event.src_path} has been created!")

    def on_deleted(self, event):
        print(f"Oops! Someone deleted {event.src_path}!")
```

2. Code to stop **observer** on key press:



```
try:
    while True:
        time.sleep(2)
        print("running...")
except KeyboardInterrupt:
    print("stopped!")
    observer.stop()
```

HELP