Python

Developing KakaoTalk automation with Python

Progress Report : 1

Date : 2023/11/24
Name : Dong Ho Han
ID : 203734

1. Introduction

1) Background

While looking at examples for automation, I often saw automation systems that send emails, but I did not see any cases of automating KakaoTalk, which is closest to our everyday life. I thought that if it was well-made, it would be much more valuable than code that automatically sends emails, so I chose this topic.

2) Project goal

To enable automatic sending of pre-written messages to specific friend lists..

3) Differences from existing programs

When I looked at some automatic email sending programs, the processes of launching the program and logging in were not automated. In this project, I plan to automate the process from launching the program to closing the program.

2. Functional Requirement

1) Function 1

- Auto-launch and login (login information is included in the code in advance)

(1) Detailed function

- Add a function to close and launch KakaoTalk for initialization

2) Function 2

- Friend search function – Find friends that are pre-entered in the code

-

3) Function 3

- Content input and transmission – Send messages to friends by automating keyboard and mouse controls

(1) Detailed function

- Add a function to close the chat window after sending a message

3. Progress

1) Function implementation

(1) function name

- Turn off KakaoTalk function/Turn on KakaoTalk function/login funtion

- I have implemented the function to turn KakaoTalk on and off. I have also written a function to automate the login process by receiving data from the user. The login function works fine on my computer, but I am aware that it may not work depending on the resolution and computer environment. I plan to overhaul the login algorithm before the next schedule.

- Using loops, conditional statements, functions, modules, libraries

```
#필요할법한 모듈들을 긁어 모아봤습니다.
import cv2 as cv
from difflib import get_close_matches
import subprocess
import os
import time
import pyautogui
import pyperclip

#아이디와 비밀번호를 입력받습니다.
```

```
ID = input("아이디를 입력하세요 : ")
PASSWORD = input("비밀번호를 입력하세요 : ")

#카카오톡을 키는 것을 자동화 하는 함수
def open_kakao():
    try:
        path = r"C:\Program Files
(x86)\Kakao\KakaoTalk\KakaoTalk.exe"
        subprocess.Popen(path)
        print('카카오톡을 열어볼까용~!')
    except:
        print("ㅠㅠ 카카오톡이 안열려요. 뭔가
잘못됐네요.")


#카카오톡을 끄는 것을 자동화 하는 함수
def kill_kakao():
     os.system("TASKKILL /F /IM KakaoTalk.exe")

#카카오톡 로그인을 자동화하는 함수
def login_kakao():
    button_location =
pyautogui.locateOnScreen('images/prac.png',
confidence=0.9)
    button_location_2 =
pyautogui.locateOnScreen('images/login_login.p
ng', confidence=0.9)
    if button_location is None and
button_location_2 is None:
```

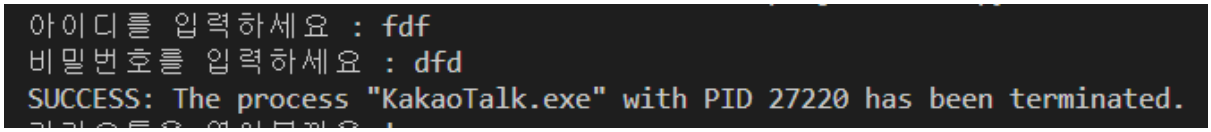
```
        print("패스워드 버튼 찾기 실패 ㅠㅠ")
    elif button_location is not None:
        button_point =
pyautogui.center(button_location)
        pyautogui.click(button_point.x,
button_point.y)
        pyautogui.write(PASSWORD)
        pyautogui.press('enter')
    elif button_location_2 is not None:
        button_point =
pyautogui.center(button_location_2)
#     pyautogui.moveTo(button_point.x,
button_point.y, duration=0)
#     pyautogui.move(0, -45, 0.5,
pyautogui.easeInQuad)  # Move 45 pixels Up
        pyautogui.doubleClick(button_point.x,
button_point.y-45)
        pyautogui.write(PASSWORD)
        pyautogui.press('enter')

# 원활한 디버깅을 위해 추가 구현해본 함수
def login_kakao2():
    # 사용자의 입력을 받는다.
    username = input("기기에 따라 카카오톡
실행속도가 다릅니다. 프로그램이 실행되면 아무키나
누르고 엔터를 누르세요 ")
```

```
#main 부분 구성

#  카카오톡이 실행되고 있을수 있으므로 일단 끄고 초기화시킵니다.
kill_kakao()

#  카카오톡 프로그램을 실행시킵니다.
open_kakao()

# 이미지 인식을 바탕으로 로그인 합니다.

login_kakao()
```

2) Test results

(1) opening/closing kakao program

-



4. Changes in Comparison to the Plan

Nothing

5. Schedule