**Abhilash Datta**

Artificial Intelligence Engineer

Epicenter News

CSE | Indian Institute of Technology, Kharagpur

Instagram Bot

**July 14, 2021**

# Overview

An instagram bot that can **mass text users**, **receive and read a text back and store it** **somewhere with user details** and much more. Exploring options like instagram API , selenium driver etc.

# Goals Formulated

* **Login into Instagram:** (Achieved)
* **Texting to a Single User:** (Achieved)
* **Uploading Images with Caption:** (On the way)
* **Following Users:** (On the way)
* **Responding to a Single User:** (Achieved)
* **Texting to Multiple Users:** (Achieved)
* **Creating Group and texting in it:** (Achieved)
* **Downloading images with HashTag:** (On the way)
* **Liking and Commenting by HashTag:** (On the way)
* **Logging Out:** (Achieved)

# 

# Specifications

**Platform**: Jupyter Notebook and Python files. Virtual Environment using PIPENV.

**Libraries**: Selenium, Instabot, InstaPy, Time, Pyperclip, Pyautogui, OpenCv

**Softwares:** WindowsChromeDriver

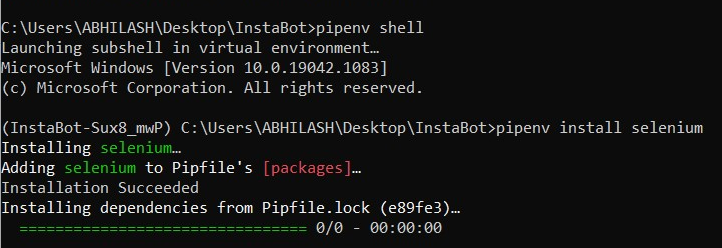
**Low-Level Specs:** Whole program is built in Object Oriented fashion and Modular structure is followed throughout.

Milestones

**July 15, 2021**

## Setting up virtual environment - pipenv

Creating a project directory and setting up a virtual environment using pipenv. Then downloading dependencies required for the project such as selenium. **PIPENV is better than virtualenv as it supports version control.**



## Downloading Chromedriver and Login to Instagram using selenium

There are other libraries too that help login to instagram such as **kora, instapy** but all use selenium as their backend, hence it'll be beneficial to learn selenium from scratch.

**from instapy import InstaPy**

**session = InstaPy(username="<your\_username>", password="<your\_password>")**

**session.login()**

## Running python interactively

**python -i bot.py**

-i flag runs python interactively. This helps to inspect the chromedriver window even after all commands have been run. Otherwise the driver windows shut down as soon as the program ends.

## Sleep vs EC (expected conditions)

Each webpage has its own life cycle, where all the elements load one after the other to assemble the content. So if the element we are targeting is still being processed and hasn't been loaded on the page yet — Selenium won’t be able to detect it.

**Solution 1:**

**from time import sleep**

**sleep(<appropriate amount of time>)**

**Solution 2: expected conditions**

**from selenium.webdriver.support import expected\_conditions as EC**

**from selenium.webdriver.common.by import By**

**from selenium.webdriver.support.wait import WebDriverWait**

**username = WebDriverWait(driver, 10).until(EC.element\_to\_be\_clickable((By.CSS\_SELECTOR, “input[name=’username’]”)))**

Solution 2 is rather advanced, saves time and is compatible for all devices. Hence we’ll carry on with Solution 2.

Milestones

**July 16, 2021**

## Object - oriented - ness and Modular Structure

Creating different files for different functions enhances clarity in code and helps to debug faster. Object oriented code provides proper abstraction.

## Messaging to a single user/group or to multiple users/groups

Program to text to a single user/group has been coded. The function takes a ‘user’ parameter and a ‘message’ parameter. Looping the previous program can send messages to multiple users individually.

## Creating a group with certain users and texting in it

Program very similar to the previous two with just some logical differences.

## Using Action Chains

Action chains can perform a wide range of operations such as hovering, right click, long click etc.

## Message Retrieval

Computer Vision is used for the retrieval of messages. **Pyautogui** library was used for detection of elements so as to navigate to the latest message and copy it to the clipboard. Then **pyperclip** was used to paste the retrieved message to a specified file.