User's Manual

Python Flight Tracker

Written by Steven Dang

Table of Contents

I. Introduction

The purpose of this designed flight tracker is to provide basic, necessary flight information up to three days before/after today's date. Additional features in this flight tracker include layouts to the terminal output, notification system, and live tracking. A concept used to collect flight information is called web scraping, and the programming language used for this flight tracker is Python. Since the program performs web scraping, let it be known that the information collected is from FlightStats by Cirium. We do not own any flight information, and it belongs to its rightful owner, FlightStats by Cirium.

II. Downloading and Opening File

If you are downloading from the GitHub website, all you have to do is click on the green button that says "Code". Once you have clicked on the green button, click the last button on the dropdown menu that says "Download ZIP."

We highly recommend you use an integrated development environment (IDE) to open the code. Some IDEs don't run Python, so make sure the IDE you have can run Python. If your IDE can run Python, make sure you install Python in your IDE. Running Python codes on the Terminal itself is acceptable, but it can be challenging if you want to run the program multiple times. If you are using a Windows operating system, it is advisable to download Visual Studio Code.

Open your IDE, and open the "PythonFlightTracker.py." Your IDE will show errors, and that's perfectly okay because we are going to install a few libraries needed for this program.

III. Installing Python Libraries

There are three libraries that you will have to install.

- Requests
- BeautifulSoup
- Rich

In order to install these libraries, open up your IDE's terminal. When you open the terminal up, enter the following one at a time:

- pip install requests
- pip install beautifulsoup4
- pip install rich

If you are wondering about the time and os module, those have already been installed when you first installed Python to your IDE or terminal.

Once you have installed those libraries, you're ready to run the program!

IV. Running the Program

If you are using an IDE, look at their instructions and see how you're supposed to run the Python file. Most of the time, you will see a play button that allows you to execute the file. If you are using the terminal independently (not integrated with an IDE), you can paste your code onto the terminal and press enter.

Now, your program should start running!

What's introduced to you is an introduction message that users should keep in mind when running the program. Then, it's followed by three prompts that require the user's input.

First Input: The first input requires you to enter a flight number. The flight number must be in IATA format, which consists of two letters followed by 1-4 digits. For example, if you are flying with Southwest Flight 123, the IATA code is WN123, and the ICAO code is SWA123.

Second Input: The second input requires you to enter the date in the MM-DD-YYYY format. As mentioned before, you can search for a flight three days before or after the present date. The flight tracker won't be able to provide any information beyond those limits.

Third Input: The third input is a simple yes or no prompt since the questions ask if you want to be notified one hour before the flight's arrival. If people plan to pick up their family or friends from the airport, they have the option to be notified by the program that the flight is arriving soon.

Once you have entered those three inputs correctly, your flight information should pop up in the terminal!

V. Expected Results

After you have inputted your flight, date, and notification option, a function that is coded into the program will check to see if it can web-scrape the flight information. If the program can't web scrape any information about the flight, it will print an error message and bring the user back to the introduction.

If the function verifies that there's flight information available, it will display all the necessary flight information. Keep in mind the following information about the program:

- If there's an active flight, the program will refresh every 60 seconds to provide correct, live flight information.
- If the flight has already arrived or been canceled, the auto-refresh will stop.
- If the flight operates multiple times a day, the moment the plane has arrived, the program will switch to the next flight.
- Not all flights will get live tracking due to limitations from the Flightstats website.
- The program will notify the user that the flight is arriving soon only if the user opts to get notified.
- The flight status block is color-coded:
    - Green = On-Time
    - Yellow = Delayed
- Red = Canceled

If you want to exit the program while it's auto-updating information, you can always press Ctrl + C to terminate the program.