

---

# **pyflightdata Documentation**

**Hari Allamraju**

**Jul 24, 2021**



---

## Contents

---

<b>1</b>	<b>About</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Source code, reporting issues etc</b>	<b>7</b>
<b>4</b>	<b>Fair usage of the API</b>	<b>9</b>
<b>5</b>	<b>Quickstart</b>	<b>11</b>
<b>6</b>	<b>pyflightdata API documentation</b>	<b>13</b>
<b>7</b>	<b>Supporting us</b>	<b>23</b>
<b>8</b>	<b>Indices and tables</b>	<b>25</b>
	<b>Python Module Index</b>	<b>27</b>
	<b>Index</b>	<b>29</b>



Welcome to the documentation for pyflightdata. These docs always track the latest version in pypi. If you see any issues or have questions please reach out to us by raising a ticket on our Github page given below.



# CHAPTER 1

---

## About

---

There are many websites that provide airline flight tracking and related information. One of the popular sites is flightradar24. Many enthusiasts use this site to get data related to an airline or particular flight and analyze it. This is largely a manual process because there is no official API.

The origin of pyflightdata is from one such analysis project. Couple of years ago while working on a small project to analyze airline routes we found that getting the data manually was a huge drag and decided to automate it with simple python scripts. That initial version of the code has been fine tuned and enhanced from time to time till it reached the current state.

pyflightdata is maintained on a best effort basis where we fix issues as and when we find them in our use and from the users. There is no fixed schedule to reconcile with the changes that might happen on flightradar24 and keep up with them. We might add other data sources in the future but the overall interface will be remain same from the users point of view.

Please feel free to reach out to us by raising issues on our github page.





## CHAPTER 2

---

### Installation

---

The best way to install pyflightdata is via pip

```
pip install pyflightdata
```

The dependencies will also be installed, however there have been issues with lxml. You might need to install that separately and resolve installation issues. lxml is used to parse data that is in pure HTML and does not provide a JSON data source.



## CHAPTER 3

---

Source code, reporting issues etc

---

Please reach out to us via the issues page at our github repo <https://github.com/supercoderz/pyflightdata>



## CHAPTER 4

---

### Fair usage of the API

---

The information from flightradar24 is very valuable for users and the team there does a wonderful job of keeping the data correct and live. This API makes simple calls as one would make from a browser to get the data. There is no optimization for performance or throttling.

We trust the users to be aware of that fact that any large scale data extraction done using this API and without reasonable throttling might result in consequences like being blocked from flightradar24 or having the user account suspended.

Also this is just a wrapper for flightradar24 and does not change or add to the terms and conditions already enforced on the users by flightradar24.

Please make sure that whatever you do, you use the API in a fair manner and do not break the terms of flightradar24.



## CHAPTER 5

---

### Quickstart

---

Let's dive in with a quick example.

The main interface to pyflightdata is the FlightData class. This abstracts all the data access mechanism and also maintains the authenticated session to flightradar24 for users who have a paid membership.

Start by initializing the FlightData class and optionally login to the account.

```
1 from pyflightdata import FlightData
2
3 f=FlightData()
4 f.login(foo@bar.com, foobar)
```

Once you login, the session token is stored on the FlightData.AUTH\_TOKEN variable.

The most common use case would be to get information about a particular flight where you know the flight number.

```
1 # get the last 5 flights for AI101
2 f.get_history_by_flight_number('AI101')[-5:]
```

Once you have this information, you might want to inspect the flights before and after this route for a particular aircraft.

```
1 # get the last 5 flights for the aircraft VT-ANL
2 # this is an Air India aircraft
3 api.get_history_by_tail_number('VT-ANL')[-5:]
```

You can also get the information about the aircraft itself like its age etc.

```
1 # get the information for a particular aircraft using tail number
2 api.get_info_by_tail_number('VT-ANL')
```

Internally pyflightdata tracks whether you are authenticated or not and the data returned by the above methods will be restricted accordingly. Users who are not authenticated will only see data for the free limits, which is usually a week.

Please refer to the examples Jupyter notebook in the next section and the API documentation for details on all the other data that you can get using pyflightdata.





**class** pyflightdata.flightdata.FlightData (*email=None, password=None*)

Bases: pyflightdata.common.FlightMixin

FlightData class is the entry point to the API.

This class abstracts the data sources and provides convenient methods to get the various datapoints as JSON lists. At the moment flightradar24 is the only data source.

It is optional to pass in the email and password to login to the site at the time of creating the API object. The login method can be invoked at a later point in the code.

### Parameters

- **email** (*str*) – optional email ID used to login to flightradar24
- **password** (*str*) – password for the user ID

Example:

```
from pyflightdata import FlightData
f=FlightData()
f.login(myemail, mypassword)
```

### clear\_last\_request()

The API maintains a simple state that lets you query further into the history of a given flight number or tail number. However this means that when you call multiple flight numbers in a loop, the state needs to be cleared in between. The API attempts to do this on its own, but also provides this method in case a user wants to call it.

Example:

```
from pyflightdata import FlightData
f=FlightData()
f.get_history_by_flight_number('AI176')
f.clear_last_request()
f.get_history_by_flight_number('AI101')
```

**decode\_metar** (*metar*)

Simple method that decodes a given metar string.

**Parameters** **metar** (*str*) – The metar data

**Returns** The metar data in readable format

Example:

```
from pyflightdata import FlightData
f=FlightData()
f.decode_metar('WSSS 181030Z 04009KT 010V080 9999 FEW018TCU BKN300 29/22_
↪Q1007 NOSIG')
```

**get\_airlines** ()

Returns a list of all the airlines in the world that are known on flightradar24

The return value is a list of dicts, one for each airline, with details like the airline code on flightradar24, call sign, codes etc. The airline code can be used to get the fleet and the flights from flightradar24

**get\_airport\_arrivals** (*iata, page=1, limit=100, earlier\_data=False*)

Retrieve the arrivals at an airport

Given the IATA code of an airport, this method returns the arrivals information.

**Parameters**

- **iata** (*str*) – The IATA code for an airport, e.g. HYD
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned
- **earlier\_data** (*boolean*) – Default false, set to true to get data from earlier in time, mimics similar feature on the site

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_arrivals('HYD')
f.get_airport_arrivals('HYD',page=1,limit=10)
```

**get\_airport\_departures** (*iata, page=1, limit=100, earlier\_data=False*)

Retrieve the departures at an airport

Given the IATA code of an airport, this method returns the departures information.

**Parameters**

- **iata** (*str*) – The IATA code for an airport, e.g. HYD
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned
- **earlier\_data** (*boolean*) – Default false, set to true to get data from earlier in time, mimics similar feature on the site

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_departures('HYD')
f.get_airport_departures('HYD',page=1,limit=10)
```

**get\_airport\_details** (*iata*, *page=1*, *limit=100*)

Retrieve the details of an airport

Given the IATA code of an airport, this method returns the detailed information like lat lon, full name, URL, codes etc.

#### Parameters

- **iata** (*str*) – The IATA code for an airport, e.g. HYD
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_details('HYD')
f.get_airport_details('HYD',page=1,limit=10)
```

**get\_airport\_metars** (*iata*, *page=1*, *limit=100*)

Retrieve the metar data at the current time

Given the IATA code of an airport, this method returns the metar information.

#### Parameters

- **iata** (*str*) – The IATA code for an airport, e.g. HYD
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** The metar data for the airport

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_metars('HYD')
```

**get\_airport\_metars\_hist** (*iata*)

Retrieve the metar data for past 72 hours. The data will not be parsed to readable format.

Given the IATA code of an airport, this method returns the metar information for last 72 hours.

**Parameters** *iata* (*str*) – The IATA code for an airport, e.g. HYD

**Returns** The metar data for the airport

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_metars_hist('HYD')
```

**get\_airport\_onground** (*iata*, *page=1*, *limit=100*)

Retrieve the aircraft on ground at an airport

Given the IATA code of an airport, this method returns the aircraft on the ground at the airport.

**Parameters**

- **iata** (*str*) – The IATA code for an airport, e.g. HYD
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_onground('HYD')
f.get_airport_onground('HYD',page=1,limit=10)
```

**get\_airport\_reviews** (*iata*, *page=1*, *limit=100*)

Retrieve the passenger reviews of an airport

Given the IATA code of an airport, this method returns the passenger reviews of an airport.

**Parameters**

- **iata** (*str*) – The IATA code for an airport, e.g. HYD
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_reviews('HYD')
f.get_airport_reviews('HYD',page=1,limit=10)
```

**get\_airport\_stats** (*iata*, *page=1*, *limit=100*)

Retrieve the performance statistics at an airport

Given the IATA code of an airport, this method returns the performance statistics for the airport.

#### Parameters

- **iata** (*str*) – The IATA code for an airport, e.g. HYD
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_stats('HYD')
f.get_airport_stats('HYD',page=1,limit=10)
```

**get\_airport\_weather** (*iata*, *page=1*, *limit=100*)

Retrieve the weather at an airport

Given the IATA code of an airport, this method returns the weather information.

#### Parameters

- **iata** (*str*) – The IATA code for an airport, e.g. HYD
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_airport_weather('HYD')
f.get_airport_weather('HYD',page=1,limit=10)
```

**get\_airports** (*country*)

Returns a list of all the airports For a given country this returns a list of dicts, one for each airport, with information like the iata code of the airport etc

**Parameters** **country** (*str*) – The country for which the airports will be fetched

Example:

```
from pyflightdata import FlightData
f=FlightData()
f.get_airports('India')
```

**get\_all\_available\_history\_by\_flight\_number** (*flight\_number*, *delay=1*)

Fetch all the available history of a particular aircraft by its flight number.

This method can be used to get all the available history of a particular aircraft by its flight number. It checks the user authentication and returns the data accordingly.

**Parameters**

- **flight\_number** (*str*) – The tail number, e.g. VT-ANL
- **delay** (*int*) – Number of seconds delay between each check for new data, defaults to 1 second

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_all_available_history_by_flight_number('6E375',delay=2)
```

**get\_all\_available\_history\_by\_tail\_number** (*tail\_number*, *delay=1*)

Fetch all the available history of a particular aircraft by its tail number.

This method can be used to get all the available history of a particular aircraft by its tail number. It checks the user authentication and returns the data accordingly.

**Parameters**

- **tail\_number** (*str*) – The tail number, e.g. VT-ANL
- **delay** (*int*) – Number of seconds delay between each check for new data, defaults to 1 second

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_all_available_history_by_tail_number('VT-ANL')
f.get_all_available_history_by_tail_number('VT-ANL',delay=2)
```

**get\_countries** ()

Returns a list of all countries This can be used to get the country name/code as it is known on flightradar24

**get\_fleet** (*airline\_key*)

Get the fleet for a particular airline.

Given a airline code form the get\_airlines() method output, this method returns the fleet for the airline.

**Parameters** **airline\_key** (*str*) – The code for the airline on flightradar24

**Returns** A list of dicts, one for each aircraft in the airlines fleet

**Example::** from pyflightdata import FlightData f=FlightData() #optional login  
f.login(myemail,mypassword) f.get\_fleet('ai-aic')

**get\_flight\_for\_date** (*flight\_number*, *date\_str*)

Fetch a flight by its number for a given date.

This method can be used to get a flight route by the number for a date. The date should be in the YYYYMM-DD format. It checks the user authentication and returns the data accordingly.

#### Parameters

- **flight\_number** (*str*) – The flight number, e.g. AI101
- **date\_str** (*str*) – The date, e.g. 20191116
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_history_by_flight_number('AI101','20191116')
```

#### **get\_flights** (*search\_key*)

Get the flights for a particular airline.

Given a full or partial flight number string, this method returns the first 100 flights matching that string.

Please note this method was different in earlier versions. The older versions took an airline code and returned all scheduled flights for that airline

**Parameters** **search\_key** (*str*) – Full or partial flight number for any airline e.g. MI47 to get all SilkAir flights starting with MI47

**Returns** A list of dicts, one for each scheduled flight in the airlines network

**Example::** from pyflightdata import FlightData f=FlightData() #optional login  
f.login(myemail,mypassword) f.get\_flights('MI47')

#### **get\_flights\_from\_to** (*origin, destination*)

Get the flights for a particular origin and destination.

Given an origin and destination this method returns the upcoming scheduled flights between these two points. The data returned has the airline, airport and schedule information - this is subject to change in future.

#### Parameters

- **origin** (*str*) – The origin airport code
- **destination** (*str*) – The destination airport code

**Returns** A list of dicts, one for each scheduled flight between the two points.

**Example::** from pyflightdata import FlightData f=FlightData() #optional login  
f.login(myemail,mypassword) f.get\_flights\_from\_to('SIN','HYD')

#### **get\_history\_by\_flight\_number** (*flight\_number, page=1, limit=100*)

Fetch the history of a flight by its number.

This method can be used to get the history of a flight route by the number. It checks the user authentication and returns the data accordingly.

You need to query page 1 first, and if there are more pages then increment page number until you get 0 results back. Limit has a max value of 100

#### Parameters

- **flight\_number** (*str*) – The flight number, e.g. AI101
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_history_by_flight_number('AI101')
f.get_history_by_flight_number('AI101',page=1,limit=10)
```

**get\_history\_by\_tail\_number** (*tail\_number*, *page=1*, *limit=100*)

Fetch the history of a particular aircraft by its tail number.

This method can be used to get the history of a particular aircraft by its tail number. It checks the user authentication and returns the data accordingly.

You need to query page 1 first, and if there are more pages then increment page number until you get 0 results back. Limit has a max value of 100

#### Parameters

- **tail\_number** (*str*) – The tail number, e.g. VT-ANL
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_history_by_tail_number('VT-ANL')
f.get_history_by_tail_number('VT-ANL',page=1,limit=10)
```

**get\_images\_by\_tail\_number** (*tail\_number*, *page=1*, *limit=100*)

Fetch the images of a particular aircraft by its tail number.

This method can be used to get the images of the aircraft. The images are in 3 sizes and you can use what suits your need.

#### Parameters

- **tail\_number** (*str*) – The tail number, e.g. VT-ANL
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data



- **limit** (*int*) – Optional limit on number of records returned

**Returns** A dict with the images of the aircraft in various sizes

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_images_by_flight_number('VT-ANL')
f.get_images_by_flight_number('VT-ANL',page=1,limit=10)
```

**get\_info\_by\_tail\_number** (*tail\_number*, *page=1*, *limit=100*)

Fetch the details of a particular aircraft by its tail number.

This method can be used to get the details of a particular aircraft by its tail number. Details include the serial number, age etc along with links to the images of the aircraft. It checks the user authentication and returns the data accordingly.

#### Parameters

- **tail\_number** (*str*) – The tail number, e.g. VT-ANL
- **page** (*int*) – Optional page number; for users who are on a plan with flightradar24 they can pass in higher page numbers to get more data
- **limit** (*int*) – Optional limit on number of records returned

**Returns** A list of dicts with the data; one dict for each row of data from flightradar24

Example:

```
from pyflightdata import FlightData
f=FlightData()
#optional login
f.login(myemail,mypassword)
f.get_info_by_flight_number('VT-ANL')
f.get_info_by_flight_number('VT-ANL',page=1,limit=10)
```

**is\_authenticated** ()

Simple method to check if the user is authenticated to flightradar24

**login** (*email*, *password*)

Login to the flightradar24 session

The API currently uses flightradar24 as the primary data source. The site provides different levels of data based on user plans. For users who have signed up for a plan, this method allows to login with the credentials from flightradar24. The API obtains a token that will be passed on all the requests; this obtains the data as per the plan limits.

#### Parameters

- **email** (*str*) – The email ID which is used to login to flightradar24
- **password** (*str*) – The password for the user ID

Example:

```
from pyflightdata import FlightData
f=FlightData()
f.login(myemail,mypassword)
```

**logout ()**

Logout from the flightradar24 session.

This will reset the user token that was retrieved earlier; the API will return data visible to unauthenticated users

## CHAPTER 7

---

### Supporting us

---

This API is maintained in our free time and we are constantly trying to keep up with the changes in the underlying data source and also add more data/features where possible. We want to do as much as we can to make it easy for you to get and use the data.

If you are a regular user and if this helps you focus on getting your work done rather than worry about how to get the data, then please consider supporting us with a donation at [PayPal](#).

The amount is of your choice, whatever you feel is enough to show your appreciation for the work that goes into pyflightdata.



## CHAPTER 8

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### p

`pyflightdata.flightdata`, [13](#)





## C

`clear_last_request()` (*pyflightdata.flightdata.FlightData method*), 13

## D

`decode_metar()` (*pyflightdata.flightdata.FlightData method*), 13

## F

`FlightData` (*class in pyflightdata.flightdata*), 13

## G

`get_airlines()` (*pyflightdata.flightdata.FlightData method*), 14

`get_airport_arrivals()` (*pyflightdata.flightdata.FlightData method*), 14

`get_airport_departures()` (*pyflightdata.flightdata.FlightData method*), 14

`get_airport_details()` (*pyflightdata.flightdata.FlightData method*), 15

`get_airport_metars()` (*pyflightdata.flightdata.FlightData method*), 15

`get_airport_metars_hist()` (*pyflightdata.flightdata.FlightData method*), 15

`get_airport_onground()` (*pyflightdata.flightdata.FlightData method*), 16

`get_airport_reviews()` (*pyflightdata.flightdata.FlightData method*), 16

`get_airport_stats()` (*pyflightdata.flightdata.FlightData method*), 16

`get_airport_weather()` (*pyflightdata.flightdata.FlightData method*), 17

`get_airports()` (*pyflightdata.flightdata.FlightData method*), 17

`get_all_available_history_by_flight_number()` (*pyflightdata.flightdata.FlightData method*), 17

`get_all_available_history_by_tail_number()` (*pyflightdata.flightdata.FlightData method*), 18

`get_countries()` (*pyflightdata.flightdata.FlightData method*), 18

`get_fleet()` (*pyflightdata.flightdata.FlightData method*), 18

`get_flight_for_date()` (*pyflightdata.flightdata.FlightData method*), 18

`get_flights()` (*pyflightdata.flightdata.FlightData method*), 19

`get_flights_from_to()` (*pyflightdata.flightdata.FlightData method*), 19

`get_history_by_flight_number()` (*pyflightdata.flightdata.FlightData method*), 19

`get_history_by_tail_number()` (*pyflightdata.flightdata.FlightData method*), 20

`get_images_by_tail_number()` (*pyflightdata.flightdata.FlightData method*), 20

`get_info_by_tail_number()` (*pyflightdata.flightdata.FlightData method*), 21

## I

`is_authenticated()` (*pyflightdata.flightdata.FlightData method*), 21

## L

`login()` (*pyflightdata.flightdata.FlightData method*), 21

`logout()` (*pyflightdata.flightdata.FlightData method*), 21

## P

`pyflightdata.flightdata` (*module*), 13