

# **AirMapSDK-Embedded**

**version 1.0A**

**AirMap**

**September 29, 2016**



# Contents

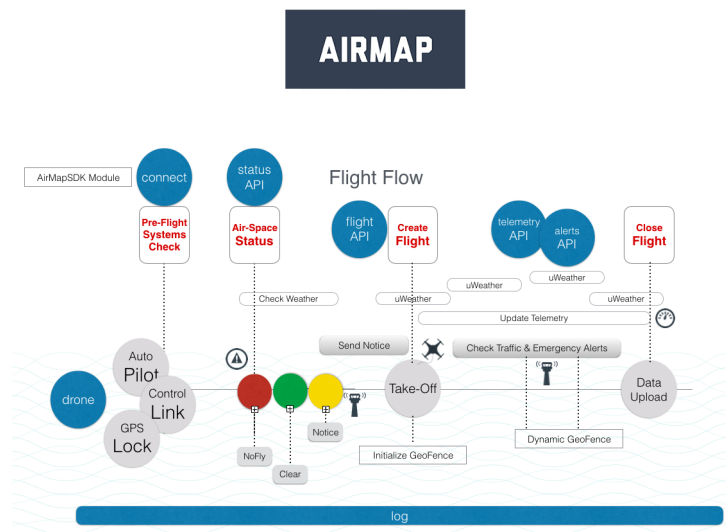
<b>Welcome to AirMap's documentation!</b>	<b>1</b>
airmap Package	2
airdefs Module	2
connect Module	4
statusAPI Module	5
flightAPI Module	6
drone Module	6
log Module	6
telemetryAPI Module	7
alertsAPI Module	7
<b>Indices and tables</b>	<b>7</b>
<b>Index</b>	<b>9</b>
<b>Python Module Index</b>	<b>11</b>



# Welcome to AirMap's documentation!

Airmap -- Airspace Management For Drones.

AirMapSDK is a Python library for interfacing with the [Airmap](#) API.



## Pre-Requisites:

- Airmap developer account
- X-API-Key from your account
- GPS data (fake gps included)
- OAuth token
- pip
- getAirmap.py (copy to /home/root/installs directory or similar)
- AirMapSDK-Embedded is pre-installed

## Wi-Fi Connectivity

```
ifconfig (check for wlan0 or LTE ethX)
cd /etc
sudo wpa_passphrase yourSSID yourPassphrase > wpa-Aero.conf
sudo wpa_supplicant -i wlan0 -c /etc/wpa-Aero.conf &
sudo dhclient wlan0
```

## Update AirMapSDK-Embedded

```
cd /etc/airmap
python getAirmap.py airmap AirMapSDK-Embedded
unzip AirMapSDK-Embedded-master.zip
cd AirMapSDK-Embedded-master
```

## User Example

```
cd /etc/airmap/AirMapSDK-Embedded
python userapp.py
```

## API Example

```
(connect) airmap.connect.set_XAPIKey(xapikey) - Set your X-API-Key from your Airmap account
(connect) airmap.connect.get_boardID() - Retrieve CID information
(connect) airmap.connect.connect() - Check internet connection and endpoint status Returns: True if ready
(statusAPI) airmap.statusAPI.get_status(lat,lon,Weather.on) - Given position retrieve airspace data
(statusAPI) Parse status information including weather, advisories, and maximum flight bounds (see statusAPI documentation)
(user) Process required notifications if needed
(connect) airmap.connect.get_SecureToken() - get security token
```

(flightAPI) airmap.flightAPI.create\_FlightPoint (flight time,lat,lon,publicflight?,sendnotifications?) - Setup the flight and flight time Returns flightID

(flightAPI) airmap.flightAPI.get\_PilotID() - Returns pilotID for this account

(flightAPI) airmap.flightAPI.end\_Flight(flightID) - End the flight specified by flight ID

(flightAPI) airmap.flightAPI.delete\_Flight(flightID) - Delete flight described by flightID

(flightAPI) airmap.flightAPI.get\_FlightList(pilotID) - Get all flights for pilotID

(flightAPI) airmap.flightAPI.cmd\_KillFlights(pilotID) - Delete all flights under specified pilotID

Documentation can be found at <https://developer.airmap.io>

Source code can be found at <https://github.com/...>

Airmap is a trademark of Airmap, Inc.

Contents:

## airmap Package

### **airdefs** Module

airdefs AirMapSDK

Created by AirMap Team on 6/28/16. Copyright (c) 2016 AirMap, Inc. All rights reserved.

*class* airmap.airdefs.**Advisories** (*distance, last\_updated, name, city, color, country, longitude, properties, state, latitude, type, id*)

Advisory group list

**Notes:** Distance, date of last update, name, city, color(status), country, longitude, properties, state, latitude, type, id

*class* airmap.airdefs.**Advisory**

Advisory information (Pre-Flight, In-Flight)

**Notes:** Advisory status color list

*class* **Color**

Flight status color code

**Colors**

Color code

**Parameters:**

- **gray** -- Disabled
- **green** -- Go
- **yellow** -- Advise
- **red** -- NoFly

alias of **Enum**

**enum** (*\*sequential, \*\*named*)

*class* airmap.airdefs.**Globals**

Global settings

**Notes:** Session parameters address, port, timeout, api key, token, pilot id, flight id

**AirConnected** = *False*

**dbgPrint** (*data*)

Debug send to console

**Parameters:** **data** -- Debug or information to send to console

**Returns:** None

`httpPort = 80`

`httpsAddr = 'api.airmap.io'`

`httpsPort = 443`

`keyAddr = 'sso.airmap.io'`

`myFlightID = None`

`myToken = None`

`pilotIDValid = False`

`pilot_id = None`

**strPrint (data)**

Information send to console

**Parameters:** `data` -- information to send to console

**Returns:** None

`telemetryAddr = 'api-aero-telemetry.airmap.com'`

`thisBID = None`

`timeOut = 18`

`xapikey = None`

**class airmap.airdefs.Notify**

Enable notifications

**Parameters:**

- **on** -- Enable notifications
- **off** -- Disable notifications

`off = False`

`on = True`

**class airmap.airdefs.Properties (prop\_name, prop\_value)**

Name to Value Keypairs

**class airmap.airdefs.Public**

Make flight public

**Parameters:**

- **on** -- Public flight
- **off** -- Private flight

`off = False`

`on = True`

**class airmap.airdefs.Requirement**

Notification requirments contact key pair list

**State**

alias of **Enum**

**enum** (\*sequential, \*\*named)

**class** airmap.airdefs.Startup  
Startup configuration

**Notes:**

**class** Drone  
Drone information (ID, Location, Status)

**State**  
alias of **Enum**

**enum** (\*sequential, \*\*named)

**class** airmap.airdefs.Weather  
Weather control parameters

**Parameters:**

- **on** -- Enable weather data
- **off** -- Disable weather data

**off** = 'false'

**on** = 'true'

## **connect** Module

connect AirMapSDK

Created by AirMap Team on 6/28/16. Copyright (c) 2016 AirMap, Inc. All rights reserved.

**class** airmap.connect.Connect

**connect** ()  
Connect to service

**Param:** None

**Returns:** True - if connected otherwise False

**connection** = None  
Connection instance

**Notes:** HTTPS access variable

**get\_SecureToken** ()  
Retrieve security token and refresh

**Param:** None

**Returns:** Token if successful otherwise False

**Todo:** Remove hardcoded token and add token from https endpoint based on CID

**get\_boardID** ()  
Retrieve CID from mmcbk0

**Param:** None

**Returns:** CID otherwise False

**headers** = None



Connection headers security and format

**Notes:** Security and format headers

`os = <module 'os' from '/usr/lib/python2.7/os.pyc'>`  
OS access

**Notes:** Run sys commands

`set_Timeout (time_out)`  
Set https request timeout time

**Parameters:** `time_out` -- Timeout time in seconds

**Returns:** True - Success, False - Fail

`set_XAPIKey (xapikey)`  
Set https request timeout time

**Parameters:** `xapikey` -- X-API-Key from developers account

**Returns:** True - Success, False - Fail

`thisGlobals = <airmap.airdefs.Globals instance at 0x2ab568519c20>`  
Global parameter access

**Notes:** Endpoint address, ports, token, id(s)

## **statusAPI** Module

statusAPI AirMapSDK

Created by AirMap Team on 6/28/16. Copyright (c) 2016 AirMap, Inc. All rights reserved.

`class airmap.statusAPI.Status`

`cmd_ProcessAdvisories ()`

`connection = None`

`get_Advisories ()`

`get_Advisory (id)`

`get_Condition ()`

`get_Humidity ()`

`get_MaxDistance ()`

`get_Precipitation ()`

`get_StatusCode ()`

`get_StatusColor ()`

`get_Temperature ()`

`get_Visibility ()`

`get_WindGusting ()`

`get_WindHeading ()`

```
get_WindSpeed ()

get_status (gps_lat, gps_lon, weather)

headers = None

levelDown (data)

localAdvisories = []

localLevelDown = []

localProperties = []

os = <module 'os' from '/usr/lib/python2.7/os.pyc'>

status_json = None

thisGlobals = <airmap.airdefs.Globals instance at 0x2ab568519d88>
```

## **flightAPI** Module

flightAPI AirMapSDK

Created by AirMap Team on 6/28/16. Copyright (c) 2016 AirMap, Inc. All rights reserved.

```
class airmap.flightAPI.Flight
```

```
    cmd_KillFlights (pilotID)

    connection = None

    create_FlightPoint (time, lat, lon, public, notify)

    delete_Flight (flightID)

    end_Flight (flightID)

    get_FlightList (pilotID)

    get_PilotID ()

    headers = None

    os = <module 'os' from '/usr/lib/python2.7/os.pyc'>

    thisGlobals = <airmap.airdefs.Globals instance at 0x2ab568519ea8>
```

## **drone** Module

drone AirMapSDK

Created by AirMap Team on 6/28/16. Copyright (c) 2016 AirMap, Inc. All rights reserved.

```
class airmap.drone.drone
```

## **log** Module

log AirMapSDK

Created by AirMap Team on 6/28/16. Copyright (c) 2016 AirMap, Inc. All rights reserved.

```
class airmap.log.log
```

## **telemetryAPI** Module

telemetryAPI AirMapSDK

Created by AirMap Team on 6/28/16. Copyright (c) 2016 AirMap, Inc. All rights reserved.

```
class airmap.telemetryAPI.Telemetry
```

```
    connection = None
```

```
    headers = None
```

```
    os = <module 'os' from '/usr/lib/python2.7/os.pyc'>
```

```
    post_Telemetry (flightID, lat, lon, alt, ground_speed, heading, barometer, status, battery,
                    drone, seq, logused)
```

```
    put_Telemetry (flightID, lat, lon, alt, ground_speed, heading, barometer, status, battery,
                   drone, seq, logused)
```

```
    thisGlobals = <airmap.airdefs.Globals instance at 0x2ab568519f80>
```

## **alertsAPI** Module

alertsAPI AirMapSDK

Created by AirMap Team on 6/28/16. Copyright (c) 2016 AirMap, Inc. All rights reserved.

```
class airmap.alertsAPI.alertsAPI
```

## Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)



# Index

## A

Advisories (class in [airmap.airdefs](#))  
Advisory (class in [airmap.airdefs](#))  
Advisory.Color (class in [airmap.airdefs](#))  
AirConnected ([airmap.airdefs.Globals](#) attribute)  
[airmap.airdefs](#) (module)  
[airmap.alertsAPI](#) (module)  
[airmap.connect](#) (module)  
[airmap.drone](#) (module)  
[airmap.flightAPI](#) (module)  
[airmap.log](#) (module)  
[airmap.statusAPI](#) (module)  
[airmap.telemetryAPI](#) (module)  
[alertsAPI](#) (class in [airmap.alertsAPI](#))

## C

[cmd\\_KillFlights\(\)](#) ([airmap.flightAPI.Flight](#) method)  
[cmd\\_ProcessAdvisories\(\)](#) ([airmap.statusAPI.Status](#) method)  
Colors ([airmap.airdefs.Advisory.Color](#) attribute)  
Connect (class in [airmap.connect](#))  
[connect\(\)](#) ([airmap.connect.Connect](#) method)  
[connection](#) ([airmap.connect.Connect](#) attribute)  
    ([airmap.flightAPI.Flight](#) attribute)  
    ([airmap.statusAPI.Status](#) attribute)  
    ([airmap.telemetryAPI.Telemetry](#) attribute)  
[create\\_FlightPoint\(\)](#) ([airmap.flightAPI.Flight](#) method)

## D

[dbgPrint\(\)](#) ([airmap.airdefs.Globals](#) method)  
[delete\\_Flight\(\)](#) ([airmap.flightAPI.Flight](#) method)  
[drone](#) (class in [airmap.drone](#))

## E

[end\\_Flight\(\)](#) ([airmap.flightAPI.Flight](#) method)  
[enum\(\)](#) ([airmap.airdefs.Advisory.Color](#) method)  
    ([airmap.airdefs.Requirement](#) method)  
    ([airmap.airdefs.Startup.Drone](#) method)

## F

[Flight](#) (class in [airmap.flightAPI](#))

## G

[get\\_Advisories\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_Advisory\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_boardID\(\)](#) ([airmap.connect.Connect](#) method)  
[get\\_Condition\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_FlightList\(\)](#) ([airmap.flightAPI.Flight](#) method)  
[get\\_Humidity\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_MaxDistance\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_PilotID\(\)](#) ([airmap.flightAPI.Flight](#) method)  
[get\\_Precipitation\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_SecureToken\(\)](#) ([airmap.connect.Connect](#) method)  
[get\\_status\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_StatusCode\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_StatusColor\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_Temperature\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_Visibility\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_WindGusting\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_WindHeading\(\)](#) ([airmap.statusAPI.Status](#) method)  
[get\\_WindSpeed\(\)](#) ([airmap.statusAPI.Status](#) method)  
[Globals](#) (class in [airmap.airdefs](#))

## H

[headers](#) ([airmap.connect.Connect](#) attribute)  
    ([airmap.flightAPI.Flight](#) attribute)  
    ([airmap.statusAPI.Status](#) attribute)  
    ([airmap.telemetryAPI.Telemetry](#) attribute)  
[httpPort](#) ([airmap.airdefs.Globals](#) attribute)  
[httpsAddr](#) ([airmap.airdefs.Globals](#) attribute)  
[httpsPort](#) ([airmap.airdefs.Globals](#) attribute)

## K

[keyAddr](#) ([airmap.airdefs.Globals](#) attribute)

## L

[levelDown\(\)](#) ([airmap.statusAPI.Status](#) method)  
[localAdvisories](#) ([airmap.statusAPI.Status](#) attribute)  
[localLevelDown](#) ([airmap.statusAPI.Status](#) attribute)  
[localProperties](#) ([airmap.statusAPI.Status](#) attribute)  
[log](#) (class in [airmap.log](#))

## M

[myFlightID](#) ([airmap.airdefs.Globals](#) attribute)  
[myToken](#) ([airmap.airdefs.Globals](#) attribute)

## **N**

Notify (class in airmap.airdefs)

## **O**

off (airmap.airdefs.Notify attribute)

(airmap.airdefs.Public attribute)

(airmap.airdefs.Weather attribute)

on (airmap.airdefs.Notify attribute)

(airmap.airdefs.Public attribute)

(airmap.airdefs.Weather attribute)

os (airmap.connect.Connect attribute)

(airmap.flightAPI.Flight attribute)

(airmap.statusAPI.Status attribute)

(airmap.telemetryAPI.Telemetry attribute)

## **P**

pilot\_id (airmap.airdefs.Globals attribute)

pilotIDValid (airmap.airdefs.Globals attribute)

post\_Telemetry() (airmap.telemetryAPI.Telemetry method)

Properties (class in airmap.airdefs)

Public (class in airmap.airdefs)

put\_Telemetry() (airmap.telemetryAPI.Telemetry method)

## **R**

Requirement (class in airmap.airdefs)

## **S**

set\_Timeout() (airmap.connect.Connect method)

set\_XAPIKey() (airmap.connect.Connect method)

Startup (class in airmap.airdefs)

Startup.Drone (class in airmap.airdefs)

State (airmap.airdefs.Requirement attribute)

(airmap.airdefs.Startup.Drone attribute)

Status (class in airmap.statusAPI)

status\_json (airmap.statusAPI.Status attribute)

strPrint() (airmap.airdefs.Globals method)

## **T**

Telemetry (class in airmap.telemetryAPI)

telemetryAddr (airmap.airdefs.Globals attribute)

thisBID (airmap.airdefs.Globals attribute)

thisGlobals (airmap.connect.Connect attribute)

(airmap.flightAPI.Flight attribute)

(airmap.statusAPI.Status attribute)

(airmap.telemetryAPI.Telemetry attribute)

timeOut (airmap.airdefs.Globals attribute)

## **W**

Weather (class in airmap.airdefs)

## **X**

xapikey (airmap.airdefs.Globals attribute)

# Python Module Index

## ***a***

[airmap](#)

[airmap.airdefs](#)

[airmap.alertsAPI](#)

[airmap.connect](#)

[airmap.drone](#)

[airmap.flightAPI](#)

[airmap.log](#)

[airmap.statusAPI](#)

[airmap.telemetryAPI](#)