

AirMapSDK

version 1.0A

Airmap

September 22, 2016

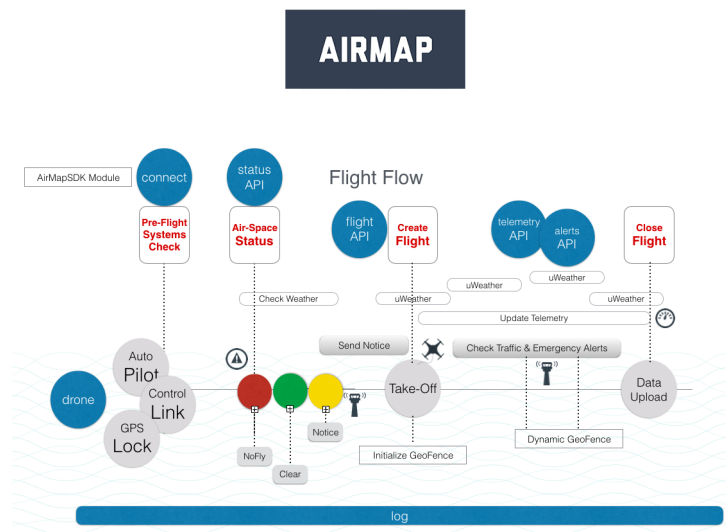
Contents

Welcome to AirMap's documentation!	1
AirMap Package	2
airdefs Module	2
connect Module	3
statusAPI Module	4
flightAPI Module	5
drone Module	5
log Module	6
telemetryAPI Module	6
alertsAPI Module	6
Indices and tables	6
Index	7
Python Module Index	9

Welcome to AirMap's documentation!

Airmap -- Airspace Management For Drones.

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



Pre-Requires:

Airmap develop account

X-API-Key from your account

GPS data (fake gps included)

OAuth token

Access Flow

(connect) `airmap.connect.set_Timeout(16)` - Set cloud access timeout in seconds

(connect) `airmap.connect.set_XAPIKey(xapikey)` - Set your X-API-Key from your Airmap account

(connect) `airmap.connect.get_CIDID()` - 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)

(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

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

Adivosry group list

class airmap.airdefs.**Advisory**

Advisoy information (Pre-Flight, In-Flight)

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

AirConnected = *False*

dbgPrint (*data*)

Debug send to console

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

Returns: *None*

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*

thisCID = *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

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'
```

```

class airmap.connect.Connect

    connect ()

    connection = None

    get_CIDID ()

    get_SecureToken ()

    headers = None

    levelDown (data)

    localAdvisories = []

    localLevelDown = []

    localProperties = []

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

    set_Timeout (time_out)

    set_XAPIKey (xapikey)

    status_json = None

    thisGlobals = <airmap.airdefs.Globals instance at 0x2b8bea9b1488>

```

statusAPI Module

```

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 0x2b8beaff4c68>

```

flightAPI Module

```

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

    localAdvisories = []

    localLevelDown = []

    localProperties = []

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

    status_json = None

    thisGlobals = <airmap.airdefs.Globals instance at 0x2b8beaff4d88>

```

drone Module

```

class airmap.drone.drone

```

log Module

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

telemetryAPI Module

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

alertsAPI Module

```
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)
[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_CIDID\(\)](#) ([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)
[httpsAddr](#) ([airmap.airdefs.Globals](#) attribute)
[httpsPort](#) ([airmap.airdefs.Globals](#) attribute)

K

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

L

[levelDown\(\)](#) ([airmap.connect.Connect](#) method)
 ([airmap.statusAPI.Status](#) method)
[localAdvisories](#) ([airmap.connect.Connect](#) attribute)
 ([airmap.flightAPI.Flight](#) attribute)
 ([airmap.statusAPI.Status](#) attribute)
[localLevelDown](#) ([airmap.connect.Connect](#) attribute)
 ([airmap.flightAPI.Flight](#) attribute)
 ([airmap.statusAPI.Status](#) attribute)
[localProperties](#) ([airmap.connect.Connect](#) attribute)
 ([airmap.flightAPI.Flight](#) attribute)
 ([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)

P

pilot_id (airmap.airdefs.Globals attribute)

pilotIDValid (airmap.airdefs.Globals attribute)

Properties (class in airmap.airdefs)

Public (class in airmap.airdefs)

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.connect.Connect attribute)

(airmap.flightAPI.Flight attribute)

(airmap.statusAPI.Status attribute)

strPrint() (airmap.airdefs.Globals method)

T

telemetryAPI (class in airmap.telemetryAPI)

thisCID (airmap.airdefs.Globals attribute)

thisGlobals (airmap.connect.Connect attribute)

(airmap.flightAPI.Flight attribute)

(airmap.statusAPI.Status 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](#)