AirMapSDK

version 1.0A

Airmap

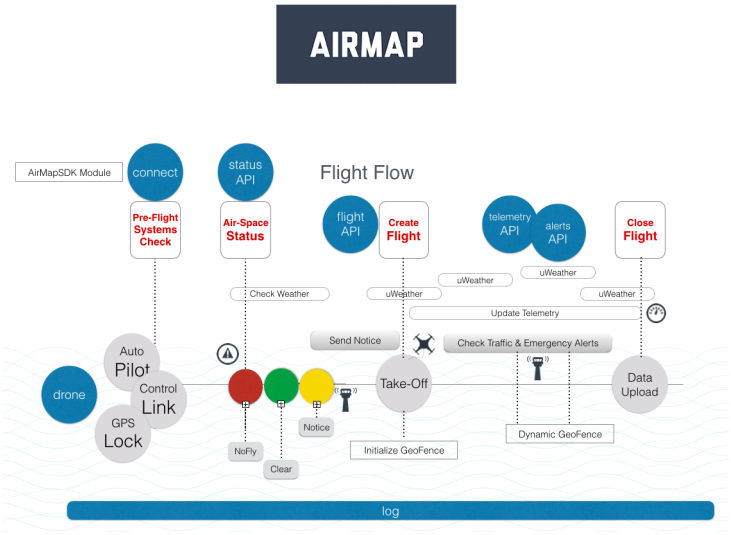
September 26, 2016

Contents

Welcome to AirMap's documentation!

Airmap -- Airspace Management For Drones.

AirMapSDK is a Python library for interfacing with the Airmap API.



Pre-Requistes:

Airmap developer account
X-API-Key from your account
GPS data (fake gps included)
OAuth token
getAirmap.py (copy to /home/root/installs directory or similar)
python requests package

Install requests* | Insert USB drive with requests-master.zip | sudo mount /dev/sda1 /mnt | cp /mnt/requests-master.zip /home/root/new_dir | cd /home/root/new_dir | unzip requests-master.zip | cd requests-master | python setup.py install

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

Install simplejson

cd /home/root
mkdir installs
cd installs
python getAirmap.py simplejson simplejson
unzip simplejson-master.zip
cd simplejson-master
python setup.py install

Install AirMapSDK

cd /home/root
mkdir Installs
cd installs
python getAirmap.py ricardoairmap EmbeddedSDK
unzip EmbeddedSDK-master.zip
cd EmbeddedSDK-master

User Example

cd /home/root/Installs/EmbeddedSDK-master
python userapp.py

API Example

(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 retreive 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*)
Adivosry group list

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

class airmap.airdefs.**Advisory**
Advisoy information (Pre-Flight, In-Flight)

Notes: Advisory status color list

class **Color**
Flight status color code

Colors
Color code

Parameters:
- **gray** -- Disbabled
- **green** -- Go
- **yellow** -- Advise
- **red** -- NoFly

alias of **Enum**

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

class airmap.airdefs.**Globals**
Global settings

Notes: Session parametes address, port, timeot, 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'*

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

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_CIDID ()
Retrieve CID from mmcblk0

Param: None
Returns: CID otherwise False

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

headers = *None*
Connection headerssecurity 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 0x2afd1f5346c8>*
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 0x2afd1f549830>*

flightAPI Module

flightAPI AirMapSDK

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 0x2afd1f544f38>*

drone Module

drone AirMapSDK

class airmap.drone.**drone**

log Module

log AirMapSDK

class airmap.log.**log**

telemetryAPI Module

telemetryAPI AirMapSDK

class airmap.telemetryAPI.**Telemetry**

connection = *None*

headers = *None*

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

put_Telemetry (*flightID*, *lat*, *lon*)

thisGlobals = *<airmap.airdefs.Globals instance at 0x2afd1f544f80>*

alertsAPI Module

alertsAPI AirMapSDK

class airmap.alertsAPI.**alertsAPI**

Indices and tables

- *genindex*
- *modindex*
- *search*

Index

Python Module Index

a