

UTM

The unmanned aircraft system traffic management (UTM) is made to mimic a simple UTM capable of acquiring and providing UAV tracking information etc. for ex. tactical deconfliction and smart route planning.

Usage

- UAV and operator registration
- Reporting of UAV tracking information
- Optional data acquisition

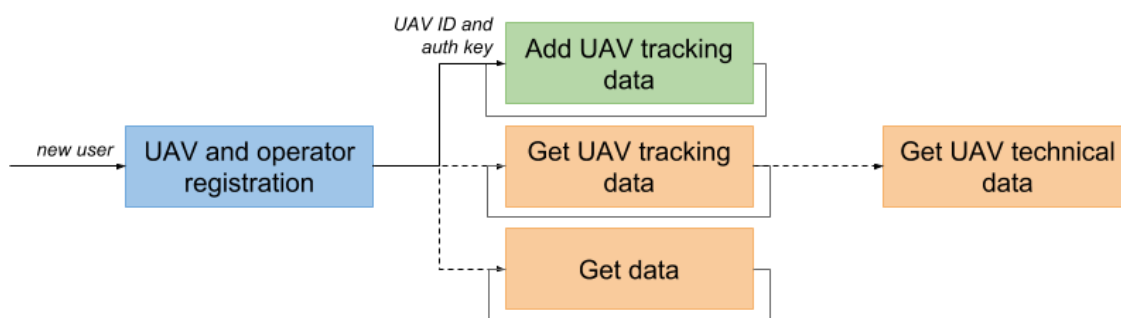


Figure 1: API usage. A dashed line indicates that the API can be used as needed, a solid line indicates required API usage, and the looping grey lines indicate that the API can be used asynchronously.

API

The API is developed based on REST HTTPS requests to support asynchronous use. Safety is ensured using encrypted TLS 1.2 access along with verification for data submission and encrypted storage of specific information. The units used are primarily SI units along WGS84 for geographical positions where the altitude is given in meters.

Register UAV

Description: each UAV and its primary operator needs to be registered in the UTM system. When an UAV and an operator is registered successfully in the system an UAV ID (equivalent to DroneID) is issued along with an authentication key which should be used for adding tracking data to the system etc.

Address: <https://droneid.dk/rmuasd/utm/uav.php>

HTTPS request type: POST

	Parameter description and unit	Parameter name/key
Input	UAV name	uav_name
	UAV weight [kg]	uav_weight_kg
	UAV max. horizontal velocity [m/s]	uav_max_vel_mps
	UAV max. Endurance [s]	uav_max_endurance_s
	GDPR compliance [yes, no]	gdpr_compliance
Optional input <i>Note that the optional inputs are only saved if GDPR compliance is accepted. The optional inputs are additionally encrypted using an asymmetric 2048 bit RSA public key (private key resides offline).</i>	(primary) Operator name	operator_name
	(primary) Operator phone	operator_phone
	(primary) Operator drone certificate	operator_drone_cert
Success response <i>Response code: 201 (created)</i> <i>Data format: JSON</i>	UAV id / DroneID	uav_id
	UAV authentication key required for tracking data submission	uav_auth_key
Error response: input error <i>Response code: 400 (bad request)</i> <i>Data format: int</i>	0	
Error response: server error <i>Response code: 500 (internal server error)</i> <i>Data format: int</i>	0	
Error response: wrong protocol <i>Response code: 505 (HTTP version not supported)</i> <i>Data format: int</i>	0	

Table 1: Register UAV and operator API documentation

Get UAV technical data

Description: get technical information about a specific UAV. The information returned is UAV ID, UAV name, UAV weight, UAV max. velocity, and UAV max. Endurance.

Address: <https://droneid.dk/rmuasd/utm/uav.php>

HTTPS request type: GET

	Parameter description	Parameter name/key
Input	UAV ID	uav_id
Success response <i>Response code: 200 (ok)</i> <i>Data format: JSON</i>	UAV id / DroneID	uav_id
	UAV name	uav_name
	UAV weight [kg]	uav_weight_kg
	UAV max. Velocity [m/s]	uav_max_vel_mps
	UAV max. Endurance [s]	uav_max_endurance_s
Error response: input error <i>Response code: 400 (bad request)</i> <i>Data format: int</i>	0	
Error response: UAV ID not found <i>Response code: 404 (Not Found)</i> <i>Data format: int</i>	0	
Error response: server error <i>Response code: 500 (internal server error)</i> <i>Data format: int</i>	0	
Error response: wrong protocol <i>Response code: 505 (HTTP version not supported)</i> <i>Data format: int</i>	0	

Table 2: Get UAV technical data API documentation

Add UAV tracking data

Description: add current and future tracking information for a specific UAV along with its operational status for deconfliction. The request is verified using the issued UAV ID and UAV

authentication key. The API also sends the tracking information to DroneID (<https://droneid.dk/borger/>).

Address: https://droneid.dk/rmuasd/utm/tracking_data.php

Request type: POST

Recommended rate: 1 Hz

	Parameter description	Parameter name/key
Input <i>If the UAV has reached its goal position the same information is reported for both current and next waypoint.</i>	UAV id	uav_id
	UAV authentication key	uav_auth_key
	UAV operation status	uav_op_status
	Current position latitude [dd]	pos_cur_lat_dd
	Current position longitude [dd]	pos_cur_lng_dd
	Current position altitude [m]	pos_cur_alt_m
	Current heading [deg]	pos_cur_hdg_deg
	Current horizontal velocity [m/s]	pos_cur_vel_mps
	GPS timestamp	pos_cur_gps_timestamp
	Next waypoint position latitude [dd]	wp_next_lat_dd
	Next waypoint position longitude [dd]	wp_next_lng_dd
	Next waypoint altitude [m]	wp_next_alt_m
	Next waypoint outgoing heading [deg]	wp_next_hdg_deg
	Next waypoint outgoing horizontal velocity [deg]	wp_next_vel_mps
	Next waypoint ETA [UNIX/EPOCH]	wp_next_eta_epoch
Optional input	UAV battery SOC <i>Set to -1 if not reported</i>	uav_bat_soc
Success response <i>Response code: 202</i>	1	

<i>(accepted)</i> <i>Data format: int</i>	
Error response: input error <i>Response code: 400 (bad request)</i> <i>Data format: int</i>	0
Error response: authentication failed <i>Response code: 412 (precondition failed)</i> <i>Data format: int</i>	0
Error response: wrong protocol <i>Response code: 505 (HTTP version not supported)</i> <i>Data format: int</i>	0

Table 3: Add UAV tracking data API documentation

Get UAV tracking data

Description: get tracking information for UAVs in a given time interval calculated back from the request time. The time interval is adjustable using the time delta parameter, ex. a time delta of 180 results in the API returning tracking information received during the last 180s (3min). If a specific UAV ID is of interest it can be provided as a parameter. The API also pulls data from DroneID (<https://droneid.dk/borger/>) but this data only contains UAV ID, time EPOCH, current position and altitude and the other parameters are therefore set to -1 if it is a DroneID entry.

Address: https://droneid.dk/rmuasd/utm/tracking_data.php

Request type: GET

	Parameter description	Parameter name/key
Optional input	Time delta [s] <i>Default: 120, max: 600</i>	time_delta_s
	UAV ID	uav_id
Success response <i>Response code: 200 (ok)</i> <i>Data format: JSON</i>	UAV id / DroneID	uav_id
	UAV name	uav_op_status
	UAV operation status <i>See 'UAV operation status'</i>	uav_bat_soc

	Server EPOCH time tracking entry	time_epoch
	Current position latitude [dd]	pos_cur_lat_dd
	Current position longitude [dd]	pos_cur_lng_dd
	Current position altitude [m]	pos_cur_alt_m
	Current heading [deg]	pos_cur_hdg_deg
	Current velocity [m/s]	pos_cur_vel_mps
	GPS timestamp	pos_cur_gps_timestamp
	Next waypoint position latitude [dd]	wp_next_lat_dd
	Next waypoint position longitude [dd]	wp_next_lng_dd
	Next waypoint altitude [m]	wp_next_alt_m
	Next waypoint outgoing heading [deg]	wp_next_hdg_deg
	Next waypoint outgoing velocity [deg]	wp_next_vel_mps
	Next waypoint ETA [UNIX/EPOCH]	wp_next_eta_epoch
Error response: No data available <i>Response code: 404 (Not Found)</i> <i>Data format: int</i>	0	
Error response: server error <i>Response code: 500 (internal server error)</i> <i>Data format: int</i>	0	
Error response: wrong protocol <i>Response code: 505 (HTTP version not supported)</i> <i>Data format: int</i>	0	

Table 4: Get UAV tracking data API documentation

Get data

Description: the get data API is able to provide specific UAS relevant data which includes static no-fly zones, rally points etc.

Address: <https://droneid.dk/rmuasd/utm/data.php>

Request type: GET

Parameter: data_type

Data description	Parameter value	Success response <i>Response code: 200 (ok)</i>
Static no-fly zones Visualization: https://droneid.dk/rmuasd/utm/data/kml-visualizer.html	'data_type' = 'static_no_fly'	Response format: KML Each no-fly zone is defined by a placemark consisting of a polygon made of coordinates.
Dynamic no-fly zones	'data_type' = 'dynamic_no_fly'	Response format: JSON Data per dynamic no-fly zone: <ul style="list-style-type: none"> • Internal server ID: int_id • Name: name • Geometry: geometry • Coordinates: geometry <ul style="list-style-type: none"> ◦ <i>Polygon format: longitude [dd], latitude [dd] where coordinate pairs are separated by a space</i> ◦ <i>Circle format: longitude [dd], latitude [dd], radius [m]</i> • Start EPOCH time: valid_from_epoch • Expiration EPOCH time: valid_to_epoch
Rally points New rally points can be added by contacting tolu@mmmi.sdu.dk	'data_type' = 'rally_points'	Response format: JSON Data per rally point: <ul style="list-style-type: none"> • ID: int_id • Name: name • Latitude [dd]: lat_dd • Longitude [dd]: lng_dd • Relative altitude above ground [m]: alt_rel_m <ul style="list-style-type: none"> ◦ <i>Not 0 when rally point is located on ex. a building roof</i> • Safe landing radius [m]: safe_radius_m

ADS-B <i>Note that the ADS-B data is from a receiver station in Odense</i>	‘data_type’ = ‘adsb’	Response format: JSON Data per aircraft: <ul style="list-style-type: none"> • Timestamp [yyyy-MM-dd HH:mm:ss]: timestamp • Timestamp [UNIX/EPOCH]: timestamp_epoch • Aircraft ICAO address: icao <ul style="list-style-type: none"> ◦ https://junzisun.com/adb/ • Latitude [dd]: lat_dd • Longitude [dd]: lng_dd • Altitude [m]: alt_m • Heading [deg]: hdg_deg • Velocity [m/s]: vel_mps
Server time	‘data_type’ = ‘server_time’	Response format: int Data: <ul style="list-style-type: none"> • Server time [UNIX/EPOCH]

Table 5: Get data API documentation

Error response

- **internal server error**
 - Response code: 500 (internal server error)
 - Data format: int
 - Response: 0
- **wrong protocol**
 - Response code: 505 (HTTP version not supported)
 - Data format: int
 - Response: 0
- **input error**
 - Response code: 400 (bad request)
 - Data format: int
 - Response: 0

UAV operation status'

- *Status -1: unknown*
- Status 0: emergency, systems failing
- Status 1: life-critical medical transport (AED, medicine)
- Status 2: disaster and accident governmental operation
- Status 3: medical transport (samples, medicine)
- Status 4-9: reserved
- Status 10: goods transport
- Status 11: photo- and videography operation

- Status 12-20: reserved
- Status 21: no payload operation

The underlined items are the operation status' though to be applicable for a blood sample delivery system.