

## RealTraffic Application Programmable Interface (API)

You can access traffic and weather data via the RealTraffic API. The responses are given in JSON strings. Please make sure you use gzip compression setting in the HTTPS requests so as not to incur unnecessary data transfer cost at my end. Input data for the API is always expected as HTTP POST data.

Please refer to the easily readable python example script for a practical implementation of this.

The sequence to establish a session and retrieve data works as follows:

- authenticate
- retrieve the GUID and use the GUID in all subsequent requests
- loop:
  - fetch traffic
  - fetch weather
- deauthenticate when done. Please don't forget this, it helps keeping the server less cluttered with session data from the users.

The GUID will remain in memory on the server for 120 seconds after the last use. If more than 120 seconds elapse between your calls, you will need to authenticate again and obtain a new GUID. Thus, session lifetime is 120 seconds.

Rate limitations apply on a per license basis.

**Note in the cURL command line examples I'm explicitly setting the header to accept gzip compression and pipe the output into gunzip rather than using the --compressed option which does the same thing but in a simpler call. This is purely to remind you to please use gzip compression in all of your calls!**

## Authentication

Address: <https://rtw.flyrealtraffic.com/v3/auth>

### POST parameters

license: the user's RealTraffic license

software: the name and version of your software.

### Example call

```
curl -sH 'Accept-encoding: gzip' -d "license=AAA-BBB-CCC-  
DDD&software=MyCoolSoftware 1.0" -X POST https://rtw.flyrealtraffic.com/v3/auth |  
gunzip -
```

## Responses

*License ok*

```
{ "status": 200, "type": 2, "rrl": 2000, "expiry": 1957899487 "GUID": "c0245e11-  
8840-46f9-9cf9-985183612495", "server": "rtw03a" }
```

type: 1 = standard, 2 = professional

**rrl**: request rate limit in ms. Make sure you sleep that amount of time before your next request or you will get a request rate violation

**expiry**: UNIX epoch second of the expiration date of the license

**GUID**: the session GUID you need to store and use in any communications going forward

**server**: the internal name of the server handling the request. Please indicate this name in case you run into repeatable problems using the API. DO NOT use this servername to connect to the API, always use the generic rtw name. Using a specific servername not only upsets the load balancing, it also will cause your application to fail as servers are taken offline/swapped/etc.

#### *Internal server errors*

```
{ "status": 500, "message": "No database connection", "server": "rtw05a" }
```

Status 500 errors will have a message associated with them giving you more information about what's wrong. It would be unusual to see any of these messages during normal operations.

#### *License not found*

```
{ "status": 404, "message": "License not found", "server": "rtw03a" }
```

#### *Too many sessions*

```
{ "status": 405, "message": "Too many sessions", "server": "rtw03a" }
```

You can only have a single session connected at any given time. Close any other RT clients you may have running, wait 10 seconds, and retry.

#### *License expired*

```
{ "status": 402, "message": "License expired", "type": 1, "expiry": 1520363968, "server": "rtw03a" }
```

## Deauthentication

Address: <https://rtw.flyrealtraffic.com/v3/deauth>

### POST parameters

GUID: the session GUID

### Example call

```
curl -sH 'Accept-encoding: gzip' -d "GUID=26cb1a52-169b-425e-8df6-7713e5c34835" -X POST https://rtw.flyrealtraffic.com/v3/deauth | gunzip -
```

### Responses

*OK*

```
{ "status": 200, "message": "Deauth success", "Session length": 1572, "server": "rtw01a" }
```

Note that the session length contains the duration of the session you just closed in number of seconds.

## Traffic

Address: <https://rtw.flyrealtraffic.com/v3/traffic>

Traffic can be retrieved by giving a latitude/longitude box and a time offset in minutes from real-time.

### POST parameters

GUID: the GUID obtained from the auth call  
querytype: should be set to locationtraffic  
bottom: southern latitude  
top: northern latitude  
left: western longitude  
right: eastern longitude  
toffset: time offset in minutes

### Example call

```
curl -sH 'Accept-encoding: gzip' -d "GUID=76ff411b-d481-470f-9ce5-5c3cbc71a276&top=-16&bottom=-17&left=145&right=146&querytype=locationtraffic&toffset=0" -X POST https://rtw.flyrealtraffic.com/v3/traffic | gunzip -
```

### Responses

#### *Internal server errors*

```
{ "status": 500, "message": "No database connection", "server": "rtw05a" }
```

Status 500 errors will have a message associated with them giving you more information about what's wrong. It would be unusual to see any of these messages during normal operations.

If the GUID is no longer valid on the server, this message is returned:

```
{ "status": 401, "message": "GUID not found, authorize please.", "server": "rtw01a" }
```

If a request is made that is not available for the current license type, the following message is returned. This would for example be returned if historical data is requested with a standard license.

```
{ "status": 403, "message": "No data for this license type.", "server": "rtw01a" }
```

#### *Request rate violation*

```
{ "status": 406, "message": "Request rate violation: 1.5", "server": "rtw05a" }
```

This shows you sent a request too quickly after the last request for traffic and you should wait for the number of milliseconds between requests that was indicated in the auth request return variable rrl (request rate limit). In this example, the server tells you that it's been 1.5 seconds since the last request was received.

On success, a JSON object of the following form is sent:

```
{"7c1522": ["7c1522", -16.878799, 145.66705, 103.16, 3025, 127.3, 1752, "X2", "A139", "VH-EGK", 1658644400.67, "MRG", "", "RSCU510", 0, -
```

```

768,"RSCU510","adsb_icao",3350,140,148,0.224,1.06,7.38,93.87,100.46,-
800,"none","A7",1014.4,1024,1008,,,"tcas",8,186,1,9,2,0.1,-
9.3,0,0,91,20,,1], "7c68a1": ["7c68a1",-
16.754288,145.693311,156.07,2325,165.2,6042,"X2","E190","VH-
UYB",1658644401.01,"BNE","CNS","QFA1926",0,-
960,"QF1926","adsb_icao",2625,173,182,0.272,-0.09,-1.41,146.6,153.18,-
928,"none","A3",1014.4,3712,2896,,,"autopilot|approach|tcas",8,186,1,9,2,0.1,-
20.2,0,0,123,19,,1], "7c49c2": ["7c49c2",-
16.866531,145.743906,0,75,0,,,"X2","C208","VH-
OUS",1658644389.19,,,"","OUS",0,0,,,"","adsb_icao",,,,,,,,,,"","A1",,,,,,"",8,186,,,,
11.9,-24.9,0,0,,,,1], "7c7aa3": ["7c7aa3",-16.8705,145.747,247,0,0,,,"F","B738","VH-
YID",1658644396,"CNS","SYD","VOZ1424",1,0,"VA1424","adsb_other",null,null,null,null
,null,null,null,null,null,null,"null","null",null,null,null,null,null,"null",null,null,null,n
ull,null,null,null,null,null,null,null,null,null,null,1], "full_count": 6839, "source":
"MemoryDB ", "rrl": 2000, "status": 200, "dataepoch": 16738882 }

```

Each aircraft has a state vector entry, with the transponder's hex ID as the key. Additional fields returned include:

- full\_count: The number of aircraft being tracked in the system at the requested point in time
- source: MemoryDB or DiskDB
- rrl: The number of milliseconds you should wait before sending the next traffic request. During times of heavy load, you may be requested to poll less frequently. If you don't honour this request, you will receive request rate violation errors.
- status: 200 for success, any other status indicates an error
- dataepoch: the epoch seconds in UTC of the data delivered

The data fields for each aircraft entry are as follows:

Example:

```

["7c68a1",-16.754288,145.693311,156.07,2325,165.2,6042,"X","E190","VH-
UYB",1658644401.01,"BNE","CNS","QFA1926",0,-
960,"QF1926","X_adsb_icao",2625,173,182,0.272,-0.09,-1.41,146.6,153.18,-
928,"none","A3",1014.4,3712,2896,,,"autopilot|approach|tcas",8,186,1,9,2,0.1,-
20.2,0,0,123,19,,,1]

```

- hexid (7c68a1)
- latitude (-16.754288)
- longitude (145.693311)
- track in degrees (156.07)
- barometric altitude in ft (std pressure) (2325)
- Ground speed in kts (165.2)
- Squawk / transponder code (6042)
- Data source: "X" (the provider code where the data came from)
- Type (E190)
- Registration (VH-UYB)
- Epoch timestamp of last position update (1658644401.01)
- IATA origin (BNE)
- IATA destination (CNS)

- ATC Callsign (QFA1926)
- On ground (0)
- Barometric vertical rate in fpm (-928)
- Flight number
- Message source type (X\_adsb\_icao) – see below for all possible types
- Geometric altitude in ft (=GPS altitude) (2625)
- Indicated air speed / IAS in kts (173)
- True air speed / TAS in kts (182)
- Mach number (0.272)
- Track rate of turn (-0.09) negative = left
- Roll / Bank (-1.41) – negative = left
- Magnetic heading (146.6)
- True heading (153.18)
- Geometric vertical rate in fpm (-928)
- Emergency (none)
- Category (A3) – see below for all possible categories
- QNH set by crew in hPa (1014.4)
- MCP selected altitude in ft (3712)
- Autopilot target altitude in ft (2896)
- Selected heading (empty)
- Selected autopilot modes (AP on, approach mode, TCAS active)
- Navigation integrity category (8) – see below for categories
- Radius of containment in meters (186)
- Navigation integrity category for barometric altimeter (1)
- Navigation accuracy for Position (9)
- Navigation accuracy for velocity (2)
- Age of position in seconds (0.1)
- Signal strength reported by receiver (-20.2 dbFS, -49.5 indicates a source that doesn't provide signal strength, e.g. ADS-C positions)
- Flight status alert bit (0)
- Flight status special position identification bit (0)
- Wind direction (123)
- Wind speed (19)
- SAT/OAT in C (none)
- TAT (none)
- Is this an ICAO valid hex ID (1)

The “message source type” field is preceded by ?\_ where ? contains any alphabetical code to indicate the data provider the data came from, and optionally is preceded by ID\_ if the data is interpolated data. The remainder can contain the following values:

- est: an estimated position
- adsb: a simplified ADS-B position only providing position, speed, altitude and track.
- adsb\_icao: messages from a Mode S or ADS-B transponder.
- adsr\_icao: rebroadcast of an ADS-B messages originally sent via another data link
- adsc: ADS-C (received by satellite downlink) – usually old positions, check tstamp.

- mlat: MLAT, position calculated by multilateration. Usually somewhat inaccurate.
- other: quality/source unknown.
- mode\_s: ModeS data only, no position.
- adsb\_other: using an anonymised ICAO address. Rare.
- adsr\_other: rebroadcast of 'adsb\_other' ADS-B messages.

#### Transmitter categories:

- A0 – No information
  - A1 – Light ( < 15,500 lbs)
  - A2 – Small (15,500 – 75,000 lbs)
  - A3 – Large (75,000 – 300,000 lbs)
  - A4 – High vortex generating Acft (e.g. B757)
  - A5 – Heavy ( > 300,000 lbs)
  - A6 – High Performance ( > 5G accel, > 400 kts)
  - A7 – Rotorcraft
- 
- B0 – No information
  - B1 – Glider/Sailplane
  - B2 – Lighter-than-air
  - B3 – Parachutist/Skydiver
  - B4 – Ultralight/Hangglider/Paraglider
  - B5 – Reserved
  - B6 – Unmanned Aerial Vehicle / Drone
  - B7 – Space/Trans-atmospheric vehicle
- 
- C0 – No Information
  - C1 – Surface vehicles: Emergency
  - C2 – Surface vehicles: Service
  - C3 – Point obstacles (e.g. tethered balloon)
  - C4 – Cluster obstacle
  - C5 – Line obstacle
  - C6-7 – Reserved
- 
- D0 – No information
  - D1-7 – Reserved

## Weather

Address: <https://rtw.flyrealtraffic.com/v3/weather>

Weather (METARs and TAFs) and the local weather can be retrieved using this call.

### POST parameters

GUID: the GUID obtained from the auth call  
 querytype should be set to "locwx"

toffset: time offset in minutes  
lat: latitude in degrees  
lon: longitude in degrees (west = negative)  
alt: altitude in ft  
airports: a list of airports for which to retrieve METARs, delimited by pipe (|)

### Example call

```
curl -sH 'Accept-encoding: gzip' -d "GUID=76ff411b-d481-470f-9ce5-5c3cbc71a276&lat=47.5&&lon=5.5&alt=37000&airports=LSZB|LSZG|LFSB&querytype=locwx&toffset=0" -X POST https://rtw.flyrealtraffic.com/v3/traffic | gunzip -
```

### Responses

#### *Internal server errors*

```
{ "status": 500, "message": "No database connection", "server": "rtw05a" }
```

#### *Request rate violation*

```
{ "status": 406, "message": "Request rate violation: 1.5", "server": "rtw05a" }
```

This shows you sent a request too quickly after the last request for weather and you should wait for the number of milliseconds between requests that was indicated in the auth request return variable rrl (request rate limit). In this example, the server tells you that it's been 1.5 seconds since the last request was received.

**Note** that the locWX query only returns data when sufficient distance has elapsed between the last call and the current call, or if more than 60 seconds have elapsed since the last call, or if it is the initial call.

This will return the following JSON output:

```
{ 'ICAO': 'LSZB',  
  'QNH': 1015,  
  'METAR': 'LSZB 070320Z AUTO VRB01KT 9999 FEW035 SCT075 03/02 Q1015',  
  'locWX': {  
    'Info': '2023-11-07_0408Z',  
    'SLP': 1016.05,  
    'WSPD': 84.54,  
    'WDIR': 241.16,  
    'T': -53.43,  
    'ST': 7.49,  
    'SVis': 24135,  
    'SWSPD': 7.56,  
    'SWDIR': 157.09,  
    'DZDT': 0.0157,  
    'LLCC': 27.2,  
    'MLCC': 3.7,  
    'HLCC': 0.0,  
    'TPP': 9652.2,  
    'PRR': 0.0,  
    'Profiles': 'RTFX1          ^N4748.0          ^E00742.0  
^FL518 242/045 -54 ^FL498 241/052 -53 ^FL478 240/058 -52 ^FL458 239/064 -52  
^FL438 238/069 -51 ^^'},  
    'AM': ['LSZH 070320Z AUTO VRB02KT 9999 FEW036 BKN041 07/05 Q1015  
NOSIG', 'LSGG 070320Z AUTO 22007KT 9999 FEW048 BKN071 08/04 Q1016 NOSIG', 'LFSB  
070330Z AUTO 16006KT CAVOK 06/05 Q1015 TEMPO FEW030TCU']  
  }
```

The fields are:

- **ICAO:** the ICAO code of the nearest airport,
- **QNH:** the reported pressure in hPa, often to within 0.1 hPa precision,
- **METAR:** contains the full METAR received,
- **TA:** the transition altitude in ft
- **locWX:** is the location weather at the present position, and contains the following information:
  - **Info** contains the timestamp if data is present, if no data is present it contains the reason for no data. Valid reasons are:
    - **TinyDelta:** Means that less than one minute has elapsed since the last query, or the lateral/vertical distance to the last query is less than 10NM / 2000ft.
    - **error:** Means there was an error on the server, the description after error contains more information.
    - If no data is provided (because not enough time or distance has elapsed to the last call), all fields are set to -1
  - **ST:** surface temperature in C
  - **SWSPD:** surface wind speed in km/h
  - **SWDIR:** surface wind direction in km/h
  - **SVis:** surface visibility in meters
  - **PRR:** precipitation rate on ground in mm/h.
    - < 0.5: none - or drizzle if not zero
    - < 2.5: light
    - < 7.5: moderate
    - > 7.5: heavy
  - **LLCC:** low level cloud cover in percent (lowest third of troposphere)
  - **MLCC:** medium level cloud cover in percent (medium third of troposphere)
  - **HLCC:** high level cloud cover in percent (top third of troposphere)
  - **DZDT:** vorticity of atmospheric layer. Larger values are indicative of turbulence. As a rule of thumb:
    - < 0.05: still air
    - < 0.5: light
    - < 1: medium (spills coffee)
    - > 1: strong (unattached objects and people go flying)
    - > 2: severe (unable to retain positive aircraft control at all times – block altitude needed)
  - **T:** OAT/SAT in C
  - **WDIR:** wind direction in degrees
  - **WSPD:** wind speed in km/h
  - **TPP:** tropopause height in meters
  - **SLP:** sea level pressure in hPa
  - **Profiles:** contains a vertical cross section / profile at the current location, formatted as “Aerowinx Format D”. This is a common output format used in flight planning and dispatch software. The format contains a line with caret symbols as newline placeholders. In the example above, the line reads as



follows:

```
RTFX1  
N3737.2  
W12034.8  
FL381 265/071 -51  
FL361 267/075 -52  
FL341 269/078 -53  
FL321 269/082 -51  
FL301 268/086 -48
```

The first line contains the waypoint name (RT Fix 1)

The second and third lines contain the coordinates of the fix

Lines 4 – 8 contain the altitude (or FL) in 2000ft increments above and below the current FL, and the Winddirection/Windspeed and OAT in C for each of those altitudes.

- **AM:** additional metars. Contains a list of an additional 6 nearest METARs in a forward looking direction (if available). You can use these METARs to gain additional understanding of the weather surrounding the aircraft.