Change Log

Version	Release Date	Note
1.0	2019-03-28	Initial version
1.1	2019-08-15	Updated for terrestrial data integration

Description

Spire's AirSafe Historical API gives access to historical data of both satellite and terrestrial ADS-B data. The satellite data commences in November 2018 and terrestrial data in August 2019.

These are the simplified steps to use the AirSafe Historical API:

1. Submit PUT requests with any filter parameters to the API to trigger a data extract.
2. The response from the PUT request returns a job_id.
3. Poll on a GET request to the API specifying the job_id.
4. When the job is completed a set of download URLs is returned. The data is retrieved from these URLs.

GET STARTED

Authentication

Before accessing the APIs, Spire must provide users with an API token for the AirSafe Historical API.

Host https://api.airsafe.spire.com/archive/job

API Token (Example) AbcDEfghIJklmn0pQrstuvWXYZ012345

Header Value Authorization: Bearer your_token

Curl example

The PUT request will export the data and prepare it for download.

```
curl -X PUT
'https://api.airsafe.spire.com/archive/job?time_interval=2019-08-12T00:00:00Z/2019-08-13T00:00:00Z'
-H 'Authorization: Bearer your_token'   -H 'Content-Length: 0'
```

The above PUT request will return a response similar to this one:
{"job_id":"HZTSJ9Uml_1yR_kagq7daooZwSA__CSV_0","job_state":"RUNNING"}

The job_id can be used to check the state and to retrieve the download urls for the exported data.

curl 'https://api.airsafe.spire.com/archive/job?job_id=HZTSJ9Uml_1yR_kagq7daooZwSA__CSV_0' -H 'Authorization: Bearer your_token'

If the job is not finished yet this will just repeat the status we received from the PUT, but at some point the job is finished and will return this:
{"job_id":"HZTSJ9Uml_1yR_kggq7paooZwZQ__CSV_0","job_state":"DONE","download_urls":[...]}

PUT request

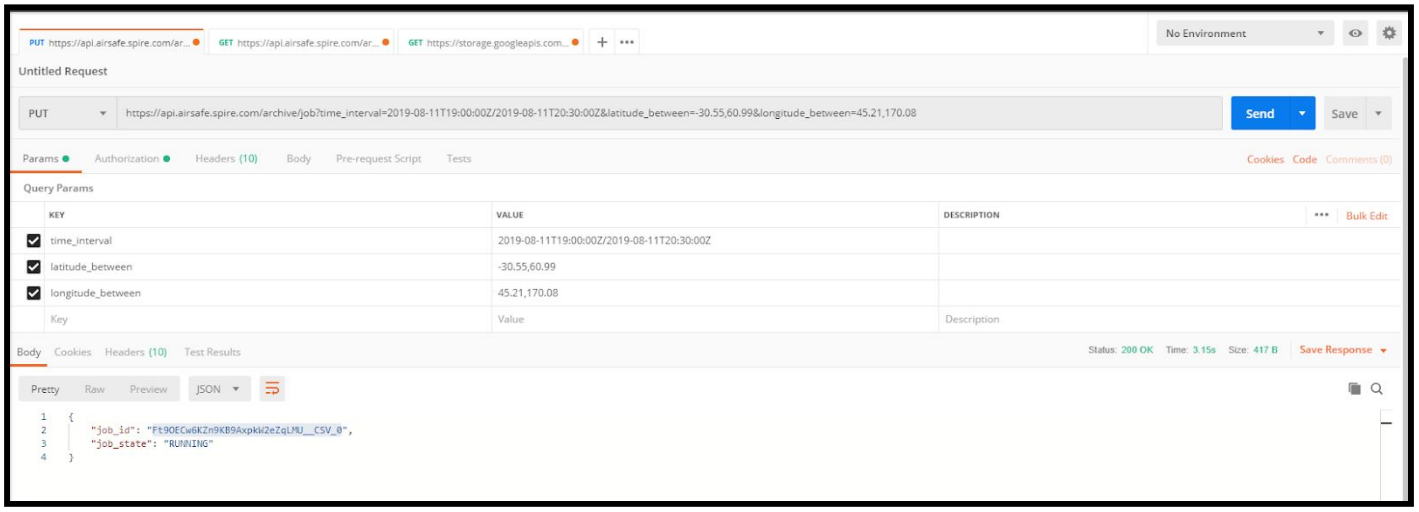
PUT requests requires a time_interval. Other parameters as noted in the table below can be used to narrow the data returned.

Query Parameters	Required?	Type	Description
time_interval	yes	timestamp	• Format: ISO 8601 Time Interval. Note the right-hand limit is excluded, so requesting two consecutive intervals does not lead to data duplication. For example, intervals "2019-01-**01**T00:00:00Z/P1D" and "2019-01-**02**T00:00:00Z/P1D" return non-intersecting data. • Any timezone is accepted, as long as the date-time is in ISO 8601 formal, for example: "2019-01-01T00:00:00-08:00". • Interval can be specified using two date-times, e.g. "2019-01-01T00:00:00Z/2019-01-02T00:00:00Z". • Period can be specified as the first part. The date-time then would mark the end of requested period. E.g. "P1D/2019-01-02T00:00:00Z". At least one of the parts must be a date-time.
icao_address	no	string	comma separated list (limit results to any of the provided values)
callsign	no	string	comma separated list (limit results to any of the provided values)

latitude_between	no	float	• Two numbers, separated by a comma. • First value must be smaller than the second, specifies south-to-north range. • First value is inclusive, last value is exclusive.
longitude_between	no	float	• Two numbers, separated by a comma. • Specifies west-to-east range. E.g. range [170, -170) is a 20° range crossing anti-meridian, and [-170, 170) is a 340° range covering most of the globe. • First value is inclusive, last value is exclusive.
altitude_baro_between	no	integer	• Barometric altitude in ft, two integers separated by a comma. • First value is inclusive, last value is exclusive.
out_format	no	string	Specifies the format of the downloadable files. Must be one of these options: • "CSV" (encoded as UTF-8, and separated by a comma) • "JSON" (encoded as UTF-8 and new line delimited) • "AVRO" Default is CSV. JSON means new line delimited JSON.
compression	no	string	For CSV or JSON: • GZIP For AVRO: • DEFLATE • SNAPPY If the parameter is not send or the string is empty, it defaults to no compression. An error will be raised if the compression algorithm is not compatible with the selected format.
ingestion_time_interval	no	timestamp	Ingestion time records the timestamp when a record was made live into the database. • Format: ISO 8601 Time Interval. Note the right-hand limit is excluded, so requesting two consecutive intervals does not lead to data duplication. For example, intervals "2019-01-**01**T00:00:00Z/P1D" and "2019-01-**02**T00:00:00Z/P1D" return non-intersecting data.

			• Any timezone is accepted, as long as the date-time is in ISO 8601 formal, for example: “2019-01-01T00:00:00-08:00”. • Interval can be specified using two date-times, e.g. “2019-01-01T00:00:00Z/2019-01-02T00:00:00Z”. • Period can be specified as the first part. The date-time then would mark the end of requested period. E.g. “P1D/2019-01-02T00:00:00Z”. At least one of the parts must be a date-time.

An example screenshot of PUT request with time_interval (required), latitude_between (optional), and longitude_between (optional) parameters.

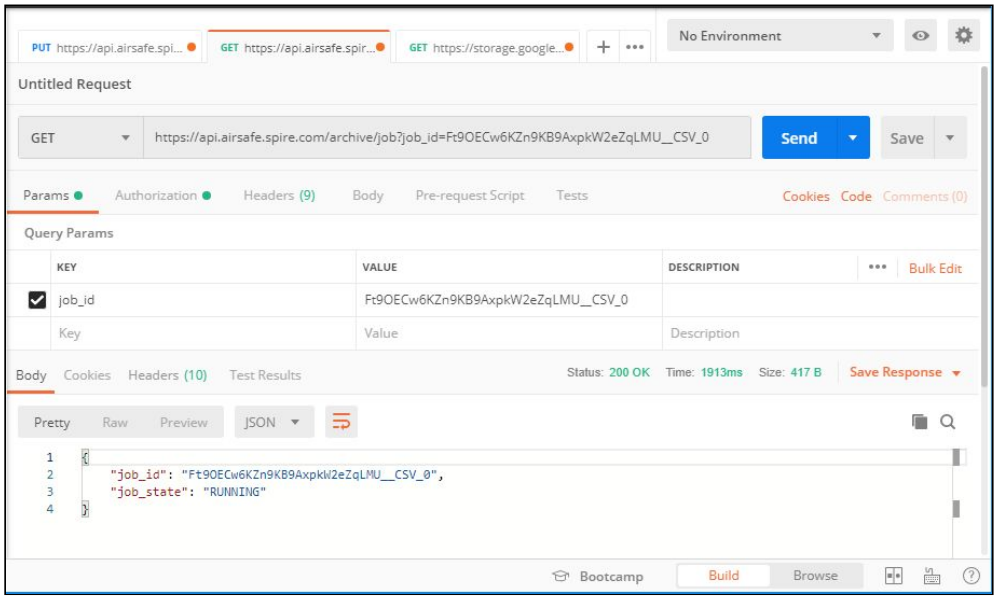


Response to PUT request

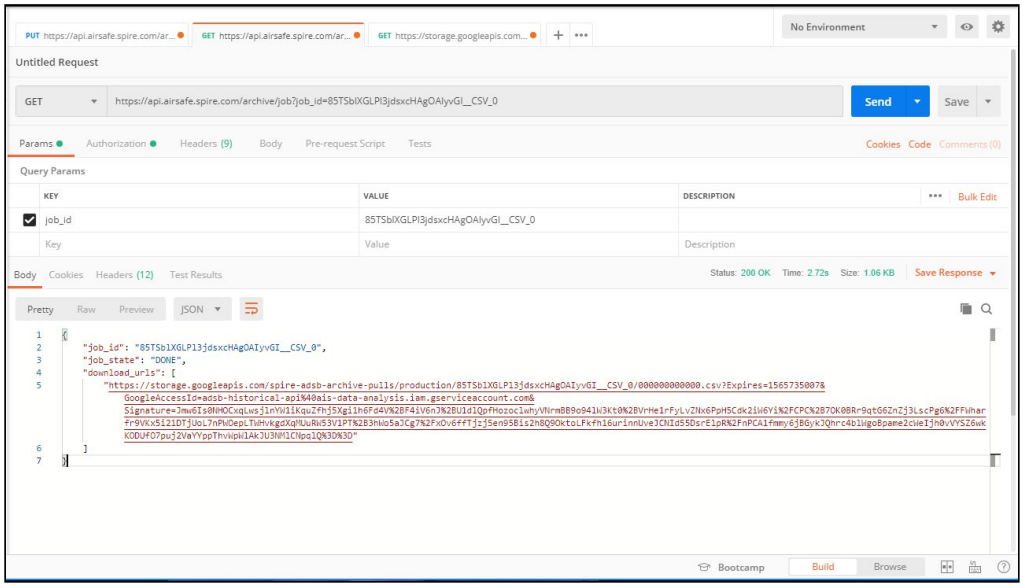
Please note the job_id value in the screenshot above. That's what you are looking for.

GET request

The only parameter for the GET request is the job_id. Like the screenshot below, it will initially say “RUNNING”. This is the system fetching the data in the background. Any export job is expected to finish within 5 minutes in total (typically around 10 seconds), regardless of data sizes, thanks to parallel processing on all stages of data extraction. In other words, give it some time before hitting ‘Send’ to check the Status.



When the job has finished running and you’ve hit SEND to refresh, you will get a result with one or more URLs from which to retrieve your data files.



Output Data Format & Description

Data will be provided to users data in the form of CSV or JSON with optional compression.

The maximum size of each data file is 1GB. The data will be returned in one or multiple files and should be joined together to get one large file with all of the exported data. The files will be available for 24 hours to be downloaded.

The fields in this table are taken directly from ADS-B messages:

Field	Type	Description	Example
callsign	string	call sign	"RAM200"
icao_address	string	hexadecimal representation of ICAO 24-bit address	"02013F"
latitude	float	latitude in degrees, between -90 and 90 (both inclusive)	-26.80491819
longitude	float	longitude in degrees, between -180 (exclusive) and 180 (inclusive)	28.64151588
altitude_baro	integer	barometric altitude in feet	38000
heading	float	aircraft heading in degrees (0 is North)	273.200012207
speed	float	aircraft speed in knots	477.8
vertical_rate	integer	vertical rate of ascent/descent, in ft/min	1020
squawk_code	string	Mode-A squawk code	6617

These fields are merged into each record to give additional information:

Field	Type	Description	Example
timestamp	timestamp	timestamp when the message was received by ADS-B receiver	"2019-06-06T07:55:25Z"
source	string	Terrestrial or FM value for satellite	“FM83”
ingestion_time	timestamp	timestamp when record was stored to the database	"2019-06-06T07:55:25Z"

collection_type	string	collection type	"satellite" or "terrestrial"

The following fields are enhanced data and may be present in the record:

Aircraft information

Field	Type	Description	Example
icao_actype	string	aircraft type (ICAO code)	"B738"
tail_number	string	aircraft Registration	"A6-EEQ"

Flight information

Field	Type	Description	Example
flight_number	string	flight number (IATA format)	"WN9002"
origin_airport_icao	string	origin airport ICAO code	"EHAM"
destination_airport_icao	string	destination airport ICAO code	"KDTW"
scheduled_departure_time_utc	timestamp	scheduled departure time, format is in ISO 8601	"2019-06-06T07:55:25Z"
scheduled_arrival_time_utc	timestamp	scheduled arrival time, format is in ISO 8601	"2019-06-06T07:55:25Z"
estimated_arrival_time_utc	timestamp	estimated arrival time, format is in ISO 8601	"2019-06-06T07:55:25Z"

Contact Us

Please contact Customer Service <cx@spire.com> for any questions.