AFF Consumer Guide

Introduction

This guide covers integrating with the spidertracks API / AFF feed. The data in the spidertracks AFF response, how to query the feed, and how to setup an AFF feed in the spidertracks website are all covered.

Examples of querying the feed, including code, requests, and responses are all provided. Please fill out the form at the end of the document in order to have this enabled for you.

What is AFF?

AFF stands for Automated Flight Following, a standard developed by the US Forest Service. The protocol is described and defined on the aff.gov website, under the Contractual Requirements section. AFF provides a standard format for transmitting real-time aircraft tracking information via HTTP requests and responses.

AFF is designed as a polling system, where clients poll the target system, e.g. spidertracks servers, for updates. Clients may not make requests more than once every 30 seconds.

The AFF specification requires point data to be available for a minimum of two weeks.

Spidertracks AFF Feeds

Spidertracks AFF Endpoints

Spidertracks has two AFF end points:

- the Standard feed (https://go.spidertracks.com/api/aff/feed)
- the Plus feed (https://go.spidertracks.com/api/aff/feed/plus)

The only difference between the two feeds is that the plus feed adds the buttonmode telemetry to the feed. Most public consumers will not require this, so should hit the standard feed end point.

The buttonmode may be useful for customers doing integrations with other systems which need the extra point information. See Feed Request and Response Elements for a detailed explanation of the feed data.

Publicity

Publicity controls who can see (and therefore send to) the account you have created. Setting an integration feed to *private* allows only organisations that the account is a member of to feed data to that integration account. This is typically used for internal monitoring software within an organisation. Setting an integration account to *public* allows all users of the spidertracks system to make their data available to your data feed as your feeds name will show up in the list of publicly available feeds. All spidertracks users can send to the feed, but only the owner of the feed can query the feed for data. **A public account does *not* make the information fed through this integration account public**. The configuration of publicity is performed by spidertracks support staff.

How to query AFF data

To query a spidertracks AFF feed, you need to make a POST request to either the Standard or Plus feed URLs listed in Spidertracks AFF Endpoints.

When making the request, there are several things to do:

- The request is authenticated with BASIC auth, using your GO website login. Make sure the request has the BASIC auth header set.
- The Content-Type header must be included for the request and be set to either application/xml or text/xml.
- The body element of the msgRequest element should be set to the desired query time.

An AFF feed query body looks like the following

The attributes and values that you need to set when making the request are highlighted in red below

```xml
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="https://aff.gov/affSchema" sysId="Adin" rptTime="2014-01-13T23:06:26Z" version="2.23">
   <msgRequest to="spidertracks" from="Adin" msgType="Data Request" subject="Async" dateTime="2014-01-13T23:06:26Z">
      <body>2014-01-12T21:01:00Z</body>
</msgRequest>
</data>
```

To ensure you do not miss points when querying the AFF feed, use the following process.

1. Make the initial request to the feed.
2. Process returned points and save the highest dataCtrDateTime for the next request.
3. Request positions using the highest dataCtrDateTime read in the previous request as the body date
4. Repeat the process from 2.

Please also note:

- A feed should not be queried more than once every 30 seconds.
- The AFF specification requires data to be kept for a minimum of two weeks. Spidertracks does not guarantee that data will be available for more than two weeks previous.
- It is helpful if you set the User-Agent header on the request to something that identifies your organisation. This will assist spidertracks with solving any issues you encounter.
- The spidertracks feed endpoints are https URLs so you will need to ensure the go.spidertracks.com SSL certificate is trusted by your system.
- Responses are limited to 1000 points at a time. If there are 1500 points in the time frame you are requesting, the first request will return the oldest 1000 points in chronological order. In order to get the next 500 points, make another request using the most recent point time you received as the query body in the second request.
  See the following request / response example:

```xml
<!-- First request -->
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="https://aff.gov/affSchema" sysId="Adin" rptTime="2014-01-13T23:06:26Z"
version="2.23">
  <msgRequest to="Spidertracks" from="Adin" msgType="Data Request" subject="Async"
dateTime="2014-01-13T23:06:26Z">
    <body>2014-01-12T21:01:00Z</body>
  </msgRequest>
</data>
<!-- First response, contains 1000 points -->
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<data xmlns="https://www.aff.gov/affSchema" version="2.23" sysID="spidertracks"
rptTime="2014-01-13T23:06:277Z">
  <posList listType="Async">
   <!-- 999 points with a dateTime pre 2014-01-13T21:07:00Z -->
    <acPos esn="300034012609560" UnitID="300034012609560" source="GPS" fix="3D" HDOP="12"
dateTime="2014-01-13T21:07:00Z" dataCtrDateTime="2014-01-13T21:07:00Z"
dataCtr="spidertracks">
      <!-- ... -->
    </acPos>
  </posList>
</data>

<!-- Second request, using the most recent point time from the first response -->
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="https://aff.gov/affSchema" sysId="Adin" rptTime="2014-01-13T23:06:58Z"
version="2.23">
  <msgRequest to="Spidertracks" from="Adin" msgType="Data Request" subject="Async"
dateTime="2014-01-13T23:06:58Z">
    <body>2014-01-13T21:07:00Z</body>
  </msgRequest>
</data>
<!-- Second response, contains 500 points -->
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<data xmlns="https://www.aff.gov/affSchema" version="2.23" sysID="spidertracks"
rptTime="2014-01-13T23:06:59Z">
  <posList listType="Async">
    <acPos esn="300034012609560" UnitID="300034012609560" source="GPS" fix="3D" HDOP="12"
dateTime="2014-01-13T21:09:00Z" dataCtrDateTime="2014-01-13T21:09:00Z"
dataCtr="spidertracks">
      <!-- ... -->
    </acPos>
   <!-- 499 points with a dateTime post 2014-01-13T21:09:00Z -->
  </posList>
</data>
```

Example AFF Feed Access

In order to test a consumer account is successfully configured to send data, a test client can be used. Spidertracks recommends the rest client available through https://code.google.com/p/rest-client/ as this is completely compatible with the spidertracks system. To use this client please follow the instructions at the end of this user guide.

In order to assist with getting started, the following examples for querying AFF are provided:

- bash script
- Powershell script
- .NET Console Application
  The console application is named SpiderTracksOnlineAFFTestConsole.exe, and has an associated configuration file named SpiderTracksOnlineAFFTestConsole.exe.config. You can save your username and password in the config file, otherwise you will have to enter it each time you run the application. You can also change the feed URL in the config file.
  The AFF feed request and response can be saved to disk. The application will ask you if you want to do this. If you say yes, request.xml and response.xml files will be created in the current directory, containing the request and response xml.

You will need to fill in a few variables, such as your user name and password, in the bash and Powershell scripts before they will work. There are comments in both scripts explaining what to fill in and what the script is doing.

The above scripts and application can help verify that you have the correct credentials and have data in the feed. Alternatively, you can use an HTTP client such as the Chrome plugin Postman Rest Client or the Java program rest-client to assist with making the request.

Example Requests and Responses

The following is an example request body. Note that the date-time in the body is the important element that determines what is returned in the response. The date-time format for this as per the XML date time format.

AFF Request

```xml
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="https://aff.gov/affSchema" sysId="BCFS" rptTime="2014-01-13T21:00:27Z"
version="2.23">
  <msgRequest to="spidertracks" from="BCFS" msgType="Data Request" subject="Async"
dateTime="2014-01-13T21:00:27Z">
    <body>2014-01-13T21:00:27Z</body>
  </msgRequest>
</data>
```

Standard AFF Feed Response

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<data xmlns="https://www.aff.gov/affSchema" version="2.23" sysID="spidertracks"
rptTime="2014-01-13T21:15:26Z">
  <posList listType="Async">
    <acPos esn="300034012609560" UnitID="300034012609560" source="GPS" fix="3D" HDOP="12"
dateTime="2014-01-13T21:05:00Z" dataCtrDateTime="2014-01-13T21:05:20Z" dataCtr="spidertracks">
      <Lat>-40.354868333333336</Lat>
      <Long>175.61191</Long>
      <altitude units="meters">44</altitude>
      <speed units="meters/sec">0</speed>
      <heading units="Track-True">120</heading>
      <telemetry name="trackid" source="spider" type="xsd:integer" value="895" />
      <telemetry name="registration" source="spidertracks" type="xsd:string" value="HBEAT" />
    </acPos>
    <acPos esn="300034012609560" UnitID="300034012609560" source="GPS" fix="3D" HDOP="12"
dateTime="2014-01-13T21:07:00Z" dataCtrDateTime="2014-01-13T21:07:04Z" dataCtr="spidertracks">
      <Lat>-40.35488</Lat>
      <Long>175.61188</Long>
      <altitude units="meters">37</altitude>
      <speed units="meters/sec">0</speed>
      <heading units="Track-True">76</heading>
      <telemetry name="trackid" source="spider" type="xsd:integer" value="895" />
      <telemetry name="registration" source="spidertracks" type="xsd:string" value="HBEAT" />
    </acPos>
  </posList>
</data>
```

Plus AFF Feed Response

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<data xmlns="https://www.aff.gov/affSchema" version="2.23" sysID="spidertracks"
rptTime="2014-01-13T21:15:26Z">
  <posList listType="Async">
    <acPos esn="300034012609560" UnitID="300034012609560" source="GPS" fix="3D" HDOP="12"
dateTime="2014-01-13T21:05:00Z" dataCtrDateTime="2014-01-13T21:05:20Z" dataCtr="spidertracks">
      <Lat>-40.354868333333336</Lat>
      <Long>175.61191</Long>
      <altitude units="meters">44</altitude>
      <speed units="meters/sec">0</speed>
      <heading units="Track-True">120</heading>
      <telemetry name="trackid" source="spider" type="xsd:integer" value="895" />
      <telemetry name="registration" source="spidertracks" type="xsd:string" value="HBEAT" />
      <telemetry name="buttonmode" source="spider" type="xsd:integer" value="0" />
    </acPos>
    <acPos esn="300034012609560" UnitID="300034012609560" source="GPS" fix="3D" HDOP="12"
dateTime="2014-01-13T21:07:00Z" dataCtrDateTime="2014-01-13T21:07:04Z" dataCtr="spidertracks">
      <Lat>-40.35488</Lat>
      <Long>175.61188</Long>
      <altitude units="meters">37</altitude>
      <speed units="meters/sec">0</speed>
      <heading units="Track-True">76</heading>
      <telemetry name="trackid" source="spider" type="xsd:integer" value="895" />
      <telemetry name="registration" source="spidertracks" type="xsd:string" value="HBEAT" />
      <telemetry name="buttonmode" source="spider" type="xsd:integer" value="0" />
      <telemetry name="description" source="spider" type="xsd:string" value="0" />
    </acPos>
  </posList>
</data>
```

Note the addition of the buttonmode telemetry in the Plus AFF feed response.

What information is available via AFF?

Feed Request and Response Elements

The following definitions are based on the AFF Schema and https://www.aff.gov/contract/current/AFF_XML_Whitepaper.pdf

Values column are elements or attributes that do not change their value, example column is for elements or attributes that may change their value.

AFF Feed Description				
element	**attribute**	**description**	**value**	**example**
data		The root element of AFF requests and responses.		
	sysID	Used to identify the server that generated the data element. You can set this to whatever you want when you make the request, eg SAR. This will be spidertracks when you are receiving a response.	spidertracks	
	rptTime	The time the response to your request was generated on the spidertracks server, or the time you generated your request.		2013-10-07T19:24:35Z
	version	The AFF protocol version. Spidertracks will return 2.23 as this is the version used in the current contract specs.	2.23	
msgRequest				
	to	sysID of the requested system, should be spidertracks.	spidertracks	

	from	sysID of the requesting system, preferably your organisation name if making a request to spidertracks AFF server.		YourOrganisationName
	dateTime	The time the request was made.		2013-10-07T19:24:35Z
	msgType	Expected to be Data Request.	Data Request	
	subject	Must be Async	Async	
posList				
	listType	The Sync type has been phased out as of 2006, so only Async used now.	Async	
acPos				
	esn	The ESN of reporting equipment, or other unique ID - responses from spidertracks will set this to be the IMEI of the Spider		300034012609560
	UnitID	In the spidertracks AFF feed, this will always be the same as the esn.		300034012609560
	source	The originator of position reported (e.g. GPS, Loran) - will be "GPS" in spidertracks AFF responses.	GPS	
	fix	The indicator of position fix type	3D	
	HDOP	The accuracy of the GPS fix. This is sent as 10x the actual value. So an HDOP of 12 is actually 1.2.		
	dataCtr	Either the company name of the equipment or the name of the dispatch center		spidertracks
	dataCtrDateTime	The time the position was inserted into the data center database, ie the time the spidertracks servers processed the Iridium point.		2013-10-07T19:24:05Z
	dateTime	The time of position aquisition, in the case of spidertracks, the time the point was generated on a spider		2013-10-07T19:23:58Z
Lat		The Latitude component of the GPS position in Decimal Degrees format		-40.35487
Long		The Longitude component of the GPS position in Decimal Degrees format		175.61191
altitude		The altitude component of the GPS position		542
	units	Spidertracks will always return the altitude in meters	meters	
speed		The speed component of the GPS position		49
	units	Spidertracks will always return the speed in meters/sec	meters/sec	
heading		The heading component of the GPS position		
	units	Spidertracks will always return the heading as Track-True	Track-True	76
telemetry		Telemetry elements allows custom data to be included in a position report. There is no limit on the number of telemetry elements that may be included.		
	name	The name of the telemetry		registration
	source	The source of the telemetry value. In the spidertracks feed, this will either be spider or spidertracks. spider is used if the information is generated dynamically on the Spider, and spidertracks indicates it is a static value set through the GO website.		spidertracks

	type	A simple XSD type, eg xsd:string, xsd:integer, xsd:double		xsd:string
	value	The value of the custom telemetry		HBEAT

For more information on any of the above, please see the AFF Schema that defines the elements and attributes. All of the elements listed above will be present in every request to

The following are telemetry elements that spidertracks adds to the feeds.

Telemetry Types						
Feed Availability	**Description**	**Name**	**Source**	**Type**	**Value**	**Example**
Standard, Plus	The current track id of the Spider. This rolls over back to the minimum once the maximum is reached.	trackid	spider	xsd:integer	0 - 65,535	<telemetry name="trackid" source="spider" type="xsd:integer" value="889" />
Standard, Plus	The registration of the aircraft, as specified in the Aircraft Settings > Registration field in the GO website	registration	spidertracks	xsd:string	<User defined>	<telemetry name="registration" source="spidertracks" type="xsd:string" value="HBEAT" />
Plus	The button mode of the particular point. Button mode values are listed in the table below.	buttonmode	spider	xsd:integer	See table below for values	<telemetry name="buttonmode" source="spider" type="xsd:integer" value="0" />
Plus	The event description of the particular point. Description values are listed in the table below.	description	spider	xsd:string	See table below for values	<telemetry name="buttonmode" source="spider" type="xsd:string" value="Landing" />

Button Modes

When an event occurs that causes the Spider to send a location report, the reason for the sending (event reason) is included in the location report. The value sent is specific to the event that occurred, and is often caused by the buttons on the Spider having been pressed. For this reason it is referred to as Button mode, but it is a more general value than just for Button press events. The values that may be sent are shown in the following table.

Button Modes		
Name	**Value**	**Description**
Normal Point	0	A normal spidertracks point. No special meaning.
NOW point	15	A point that was triggered by the Now function in the GO website.
Geofence 0	40	Geofence event with geofence 0
Geofence 1	41	Geofence event with geofence 1
Geofence 2	42	Geofence event with geofence 2
Geofence 3	43	Geofence event with geofence 3
Geofence 4	44	Geofence event with geofence 4

Geofence 5	45	Geofence event with geofence 5
Watch On	50	Watch was turned on
Watch Off	51	Watch was turned off
Watch On Resend	52	Watch was turned on but the previous Watch on point hasn't been acknowledged as received
Watch Off Resend	53	Watch was turned off but the previous Watch off point hasn't been acknowledged as received
Blocks On	55	Blocks On Event
Blocks Off	56	Blocks Off Event
Automated Watch On	57	Automated Watch On Point (caused by passing through transition speed)
Speed Up	58	Speed Up point (Passing through transition speed - accelerating)
Slow Down	59	Slow Down point (Passing through transition speed less 5 knots - decelerating)
Mark 1	61	Mark 1 button press event
Mark 2	62	Mark 2 button press event
Mark 3	63	Mark 3 button press event
Mark 4	64	Mark 3 button press event
Radius	68	Turned on Radius
Heading Change	69	Heading change event
Rate of Climb	80	Rate of Climb Event (Rate of Climb Trigger)
Rate of Climb safe	81	Rate of Climb Un-trigger Event (Safe)
Rate of Descent	82	Rate of Descent Event (Rate of Descent Trigger)
Rate of Descent safe	83	Rate of Descent Un-trigger Event (Safe)
Altitude Change	84	Altitude Change Event (Altitude Trigger)
SOS	111	SOS button press event

Description Values

Event	Value	Event Description
Engine On	Engine On	Engine On Event
Engine Off	Engine Off	Engine Off Event
Take Off	Take Off	Take Off Event
Landing	Landing	Landing Event
Start of drop	Start of drop	Start of drop Event
End of drop	End of drop	End of drop Event

Note: Speed transition events can be configured in the organisation Settings > Aircraft Settings screen on the GO website.

If you require more information about Button Mode values, please contact support@spidertracks.com.

Handling Error Responses

If there is a problem with the request, the response from Spidertracks will include a msgList element that contains a description of the error encountered. The HTTP response code will still be 200.

Example AFF error response

```
<data xmlns="https://www.aff.gov/affSchema" version="2.23" sysID="spidertracks"
rptTime="2014-01-13T23:06:26Z">
 <msgList>
  <msg to="Adin" from="spidertracks" msgType="ERROR" subject="Invalid request"
dateTime="2014-01-13T23:06:26Z">
   <body>Server does not support Sync requests.</body>
  </msg>
 </msgList>
</data>
```

Heartbeat Data

In order to ensure the spidertracks service is running, a heartbeat Spider runs 24/7, configured to send a point every two minutes. There is an option when setting up an AFF feed to send the heartbeat Spider point data to your feed. This provides testing data for you, without having to run your own Spider. You can request the heartbeat data be turned off once you have your system up and running. The heartbeat Spider track id will change every 24 hours, approximately.

The heartbeat Spider has an IMEI of 300034012609560 and its registration is "HBEAT".

Setting up an AFF feed

Definitions

Consumer: spidertracks user account that has been granted AFF consumer privileges, and can query the AFF endpoint.

Consumer Connection Account

An account on the spidertracks website must be created with a valid email address and password. This email address is also used to pass any information about updates, issues, etc. that is associated with the data feed system. It is therefore important to ensure that the email address used can be monitored by staff involved with maintaining the consumer software using the account to retrieve the data feed.
Once this account has been created, the password can be managed by logging into spidertracks as a normal user (please see spidertracks support for more information).

Requesting Service Establishment

Once an account has been created, a request for integration access needs to be lodged with spidertracks. Please complete the form at the end of this guide and return it to spidertracks. Spidertracks will advise you once this has been processed and integration access has been established for this account.

Consumer Step Summary

1. Create an account on the spidertracks website
2. Complete the Request to use spidertracks API at the bottom of this page and email it to support@spidertracks.com.
   Spidertracks will then configure your data feed which will enable you to continue with steps 3 – 5
3. Create the software service that will retrieve data from the API
4. Configure any Spiders that are to send location information to the consumer account

5. Confirm correct request/response processing with spidertracks

Spider Configuration

In order for an aircraft/Spider to have its location information included in a consumers data feed, it must be subscribed to that data feed. This is managed through the Aircraft - AFF/API menu selection for the Organisation that owns the Spiders. Figure 1 shows the web interface for the AFF/API page to configure the service.

To configure an aircraft/Spider to send location information, select it in the selection box at the top of the page. Then simply click the checkbox beside the consumer you wish to subscribe the aircraft/Spider too.

Note here that there are the two types of consumers shown. The list of public consumers is quite long, and none of these accounts are a member of this Organisation, however the aircraft/Spider location information can still be sent to the consumer. This Organisation also has a private consumer established for it, and this is shown in the Private AFF Consumers area.
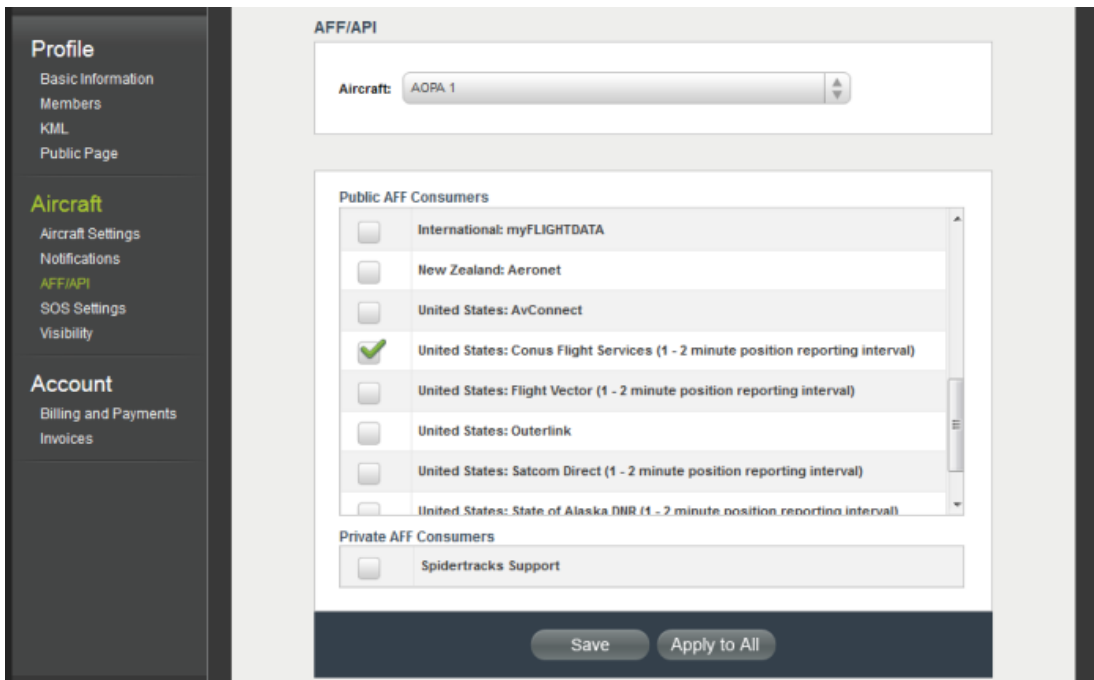

Figure 1 - AFF/API Set up Page

Once one aircraft/Spider has been configured using the interface, its settings can be applied to all aircraft for the Organisation if required. This provides a fast mechanism for configuring the majority of a fleet. Once 'Apply to All' has been clicked, individual aircraft/Spiders can be customised as required.

Request to use spidertracks API

Please use this form for requesting Spider Tracks Ltd (STL) to set your personal account as an Integration data consumer. All sections must be filled out.

Company / Personal Details

These details will be used to contact you for all support issues relating to the Integration service. Note you will also receive information on the account email as well.

Name	Company	Contact Email	Contact Phone
_____	_____	_____	_____

Publicity

Do you wish to be able to receive data from Organisations that your account is a member of (Private Consumer), or from any Organisation (Public Consumer)? Please circle:

Private Public

Consumer Display Name

What name do you want displayed for the account? We suggest your company name for public consumers and a description of the integration services use for private consumers.

Name	_____

Country or Region

At this time STL does not limit which public consumers are visible to an Organisation by country. However if you are focused on a particular country (or region) this should be included in the name. If this is the case which country or region are you associated with. If any country the please enter *International*.

Country or Region	_____

Reporting Interval

When subscribing to your integration service, you can specify customers to set a minimum reporting interval that the unit can send. If this is the case please specify this below. Note, all aircraft subscribed to this consumer will have this applied to their settings. Please circle yes or no.

No   Yes   If yes, please state the reporting interval required: _____

Please send this completed form through to support@spidertracks.com. It you have any questions about any sections, please send an email to

support@spidertracks.com.