Rock Seven XML API

A quick-start, example-led guide to the Rock Seven XML API

Version 4.1 - Updated 18th August 2017

support@rock7.com

ROCK SEVEN
LOCATION COMMUNICATION

Background

The Rock Seven XML API enables remote systems to interact with the RockSTAR, RockFLEET and RockAIR devices. It is intended to allow integration of the device into third-party back-office systems.

Revision History

4.1

- Corrected XML configuration elements.

4.0

- Added RockAIR specific methods for confirming Watch Me mode.
- Deprecated /GetMessages polling method

3.3

- Added SendAppMessage method for i360 CONNECT App.
- Added additional push API POST parameters for i360 CONNECT App.
- If you don't know what i360 is, don't worry - it's not for you.

3.2

- Added SendRawMessage method

3.1

- Added SendSerialMessage method.
- Replaced old URL references.

Push API

The Push API will send every mobile-originated message from a device to your own web service as an HTTP POST. The parameters included in the POST can vary depending on the type of device and the type of message, as described below.

You may configure multiple Push API endpoints via the Rock 7 Core user interface. Please contact your account manager if you require any assistance.

Response

It is essential that your web service implementation of the Push API endpoint responds correctly. Our service requires an HTTP **200** response to indicate successful delivery - our service will then consider the message to be delivered, and no further retry attempts will be made.

Any other HTTP response will be considered a 'temporary' failure, and our service will exponentially back off and retry for up to 1 week.

POST parameters

The table below describes the parameters which MAY be included.

Note that if the trigger parameter is set to CONFIG_REPORT, then you may also receive all/any of the parameters listed in the "*Configuration attribute reference*" later in this document.

Parameter	Description	Example/Possible values
imei	IMEI of Iridium device (only present for ROCKBLOCK device types)	15 digits
device_type	Type of device. This may be set to LEOPARD which refers to the family of products including RockSTAR and Yellowbrick v3. TIGERSHARK refers to the family of devices including RockFLEET and YB3i. GRIFFIN refers to RockAIR.	LEOPARD ROCKBLOCK TIGERSHARK GRIFFIN
serial	The serial number of the device.	An integer
momsn	The MOMSN of the SBD transmission. The MOMSN is a counter stored in the Iridium device, incremented upon transmission.	An integer

Parameter	Description	Example/Possible values
transmit_time	Timestamp that the message was transmitted.	e.g. 2014-02-14T13:55:16Z
at	GPS timestamp of transmitted position.	e.g. 2014-02-14T13:54:48Z
iridium_longitude	The approximate longitude of the device, derived by the Iridium satellites.	e.g. -5.1841
iridium_latitude	The approximate latitude of the device, derived by the Iridium satellites.	e.g. 50.9009
cep	The accuracy, in km, of the Iridium derived position.	An integer
trigger	The reason for this transmission to have been triggered.	ROUTINE MANUAL ACTIVATION CONFIG_REPORT WAYPOINT ACKNOWLEDGE BLUETOOTH_LOSS COLLISION COUNTDOWN DEAD_MAN GEOFENCE BUTTON CANCEL_ALERT POWER_LOSS TEMPERATURE GENERIC BLE_RAW MAILBOX_CHECK APP_MESSAGE WATCHING_START_REQUEST WATCHING_STOP_REQUEST

Parameter	Description	Example/Possible values
source	An indicator of whether the transmission includes a valid GPS fix.	GPS IRIDIUM
lat	GPS latitude	50.88886
lon	GPS longitude	-1.29198
sog	Speed over ground	e.g. 0.1
cog	Course over ground	e.g. 302
alt	GPS altitude (in metres)	e.g. 22
temp	Temperature	e.g. 4.5
battery	Battery level, in percent	e.g. 99
message	A text message sent from the device.	e.g. All fine
ack_request	The id of a text message sent FROM the device, to be used when acknowledging.	e.g. 123
message_ack	The id of a text message previously sent TO the device, which the device is acknowledging.	e.g. 321
alert	Indicator of whether or not this transmission was triggered by an alert condition.	true, false
waypoint	The name of the waypoint, as selected by the device user.	e.g. Bridge

Parameter	Description	Example/Possible values
appMessageAddress	The comma-separated list of addresses for recipients specified by the user of an i360 CONNECT App.	someone@gmail.com, 4477701233321
appMessageContent	The message content for a message sent using the i360 CONNECT App.	

Alerts

If a tracker transmits an 'alert' message, you will now see an **alert='true'** parameter, as well as an **alertType="BUTTON"** parameter. The alertType parameter could have any of these values:

Value	Meaning
BUTTON	The user has pressed the red alert button
BT_LOST	The bluetooth connection has been lost
COLLISION	A collision alert
COUNTDOWN	The countdown timer has timed-out
DEAD_MAN	The dead-man timer wasn't reset in time
GEOFENCE	The tracker has moved too far from it's geofence centre
CANCEL	The user has cancelled any previous alerts (this is a menu option on the tracker).
POWER_LOST	External power (5V) has been disconnected
TEMPERATURE	Temperature outside preset range.

Important note regarding 'Iridium' position reports:

We can report transmissions to you that may not have a valid GPS fix. Typically this happens if it has a poor view of the sky, but there are other occasions when it will transmit a message without a GPS position. In this case, we do get an approximate position from the Iridium network. This ranges in accuracy - often it can be within a few km, or it may be 200km out.

There is a **source="GPS"** or **source="IRIDIUM"** parameter in every <position/> element. If it's an Iridium position, you won't get sog, cog or alt parameters. You will get a **cep="123"** parameter, which indicates the accuracy (in km) of the reported position.

It's necessary for us to tell you about Iridium positions, because it is possible that the tracker can report an alert with no GPS fix.

MT (Mobile Terminated) Messages

To a RockSTAR/YB3 Screen

To send a message to a device screen, you need to call **/SendMessage**

Note: this must be an HTTP POST. ***GET won't work.***

An example:

```
$ curl -d "username=xxx&password=yyy&message=Hello%20World&ackRequest=yes" \
            "https://core.rock7.com/API2/SendMessage/2322"

<?xml version="1.0" encoding="UTF-8"?>
<SendMessage><receipt messageId="2110" /></SendMessage>
```

The POST parameters are message=blah and ackRequest=yes/no which would cause the device to prompt the user to acknowledge receipt/reading.

The POST should be UTF-8 encoded.

If successful, you'll see the <receipt /> element containing the unique ID for this MT message. This will match the messageAck parameter discussed above when the device user acknowledges.

To a RockFLEET/YB3i Serial field application

To send an MT message which will be available to your field application using the serial API, you need to call **/SendSerialMessage**

Note: this must be an HTTP POST. ***GET won't work.***

An example:

```
$ curl -d "username=xxx&password=yyy&data=8a8c8b" \
            "https://core.rock7.com/API2/SendSerialMessage/2322"

<?xml version="1.0" encoding="UTF-8"?>
<SendSerialMessage><receipt messageId="21199" /></SendSerialMessage>
```

To RockCONNECT Application (Raw message)

To send an MT raw message which will be available to your application using the RockCONNECT framework, you need to call **/SendRawMessage**

Note: this must be an HTTP POST. ***GET won't work.***

An example:

```
$ curl -d "username=xxx&password=yyy&data=8a8c8b" \
            "https://core.rock7.com/API2/SendRawMessage/2322"

<?xml version="1.0" encoding="UTF-8"?>
<SendRawMessage><receipt messageId="21199" /></SendRawMessage>
```

To i360 CONNECT Application (BLE text message)

To send a message which will be available to the user’s i360 messaging application, you need to call **/SendAppMessage**

Note: this must be an HTTP POST. ***GET won't work.***

The POST should be UTF-8 encoded.

An example:

```
$ curl -d
“username=xxx&password=yyy&address=sender@domain.com&message=%D0%97%D0%B4%D1%80%D0%B0%D0%B2%D1%81%D1%82%D0%B2%D1%83%D0%B9%D1%8
2%D0%B5” \
            "https://core.rock7.com/API2/SendAppMessage/2322"

<?xml version="1.0" encoding="UTF-8"?>
<SendAppMessage><receipt messageId="21199" /></SendAppMessage>
```

Watch Me Request Confirmation

To confirm to a RockAIR user that their “Watch Me” request has been received, and that the system is now “Watching” them:

```
$ curl -d "username=xxx&password=yyy" “https://core.rock7.com/API2/SendWatchStartConfirm/2322"
```

Similarly, to confirm to a RockAIR user that their “Stop Watching Me” request has been received, and that the system is no longer monitoring them:

```
$ curl -d "username=xxx&password=yyy" "https://core.rock7.com/API2/SendWatchStopConfirm/2322"
```

Configuration Reports

You can request the last-known configuration for a particular device.

To return the last-known configuration for tracker serial 2322:

https://core.rock7.com/API2/GetConfig/2322?username=xxxx&password=yyyy

An example of the response:

```
<TrackerData>
      <device serial="2802">
            <Config>
                  <Info software="1.9.9" pcb="C" reportAt="2012-09-04T07:52:49Z"/>
                  <Tracking state="on" freq="60" burstFixFreq="5" burstTransmitFreq="1" commandAt="2012-09-04T07:52:49Z"/>
                  <Mailbox state="off" freq="30" reportAt="2012-09-04T07:52:49Z"/>
                  <Alert timerState="off" timerTimeout="5" deadManState="off" deadManTimeout="5" temperatureState="off"
                  temperatureFreq="1" temperatureCold="10" temperatureHot="40"geofenceState="off" geofenceFreq="1"
            geofenceRadius="500" powerLossState="off" reportAt="2012-09-04T07:52:49Z"/>
                  <Bluetooth state="off" reportAt="2012-09-04T07:52:49Z"/>
                  <System encryption="off" logging="no" reportAt="2012-09-04T07:52:49Z"/>
                  <GPS accuracy="2D" hotMode="off" fixesRequired="5" earlyness="20" reportAt="2012-09-04T07:52:49Z"/>
                  <ScreenLock state="off" pin="1234" reportAt="2012-09-04T07:52:49Z"/>
            </Config>
      </device>
</TrackerData>
```

Notice that every config element will indicate the time that these options were either sent to or received from the tracker, with a **commandAt** or **reportAt** attribute

Requesting Configuration

To request the tracker with serial 2322 to transmit a configuration report:

https://core.rock7.com/API2/RequestConfig/2322?username=xxxx&password=yyyy

Note that the configuration request will be queued on the network, and will be received by the tracker during it's next routine communication with the network. For example, a tracker that is set to transmit every 4 hours, will only receive any waiting requests or commands every 4 hours.

Sending Configuration

To configure the tracker with serial 2322 to transmit every 15 minutes, HTTP POST:

https://core.rock7.com/API2/SendConfig/2322

POST parameters:

trackingState=on&trackingFreq=15&username=xxxx&password=yyyy

All possible POST parameters are listed in the table below.

Configuration attribute reference

All elements and parameters from the XML are optional. There are a few extra parameters that can also be sent. The following table explains all:

XML element	XML attribute	equivalent POST param	Allowed values	Notes
<Tracking />	state	trackingState	on off	Whether regular, automatic position reporting is switched on or off.
<Tracking />	freq	trackingFreq	burst 15s 30s 1 2 3 4 5 6 8 10 12 15 20 30 60 90 120 180 240 360 480 720 1440	Frequency that the tracker will attempt to transmit it's position. Mostly in minutes, but there are 15 and 30 second modes too.

XML element	XML attribute	equivalent POST param	Allowed values	Notes
<Tracking />	burstFixFreq	trackingBurstFixFreq	5 10 15 20 30 60 120 300 600 900 1200	How often will a GPS fix be recorded, when in burst mode, in seconds.
<Tracking />	burstTransmitFreq	trackingBurstTransmitFreq	1 2 5 10 15 30 60	How often will a set of fixes be transmitted, when in burst mode, in minutes.
<Tracking />	cellFreq	trackingCellularFreq	see *trackingFreq*	Normal tracking frequency when in cellular context
<Tracking />	cellBurstFixFreq	trackingCellularBurstFixFreq	see *trackingBurstFixFreq*	How often will a GPS fix be recorded, when in burst mode, in seconds. Cellular context.
<Tracking />	cellBurstTransmitFreq	trackingCellularBurstTransmitFreq	see *trackingBurstTransmitFreq*	How often will a set of fixes be transmitted, when in burst mode, in minutes. Cellular context
<Tracking />	distressFreq	trackingDistressFreq	see *trackingFreq*	Normal tracking frequency when in distress context
<Tracking />	distressBurstFixFreq	trackingDistressBurstFixFreq	see *trackingBurstFixFreq*	How often will a GPS fix be recorded, when in burst mode, in seconds. Distress context.
<Tracking />	distressBurstTransmitFreq	trackingDistressBurstTransmitFreq	see *trackingBurstTransmitFreq*	How often will a set of fixes be transmitted, when in burst mode, in minutes. Distress context

XML element	XML attribute	equivalent POST param	Allowed values	Notes
<Tracking />	activitySenseMode	activitySenseMode	off power bump sog bump_sog	Activity sense mode
<Tracking />	activitySenseThreshold	activitySenseThreshold	integer (min: 18, max: 54)	Activity sense threshold
<Mailbox />	state	mailboxState	on off	Whether regular, automatic mailbox checking is switched on or off.
<Mailbox />	freq	mailboxFreq	5 10 15 20 30 60 90 120 180 240 360 480 720	Frequency that the tracker will attempt to check it's mailbox, in minutes.
<Alert />	timerState	alertTimerState	on off	Whether the timer alert function is switched on or off.
<Alert />	timerTimeout	alertTimerTimeout	5 10 15 30 60 120 240	The number of minutes after which the timer alert will be triggered.

XML element	XML attribute	equivalent POST param	Allowed values	Notes
<Alert />	deadManState	alertDeadManState	on off	Whether the dead man alert function is switched on or off.
<Alert />	deadManTimeout	alertDeadManTimeout	5 10 15 30 60 120 240	The number of minutes after which the dead-man timer must be reset, otherwise an alert will be triggered.
<Alert />	temperatureState	alertTemperatureState	on off	Whether the temperature alert function is switched on or off.
<Alert />	temperatureFreq	alertTemperatureFreq	1 2 3 5 10 15 30	The frequency, in minutes, that the tracker will check the temperature.

XML element	XML attribute	equivalent POST param	Allowed values	Notes
<Alert />	temperatureCold	alertTemperatureCold	-40 -35 -30 -25 -20 -15 -10 -5 0 5 10 15 20 25 30 35 40 45 50	The cold limit setting for the temperature alert.

XML element	XML attribute	equivalent POST param	Allowed values	Notes
<Alert />	temperatureHot	alertTemperatureHot	-40 -35 -30 -25 -20 -15 -10 -5 0 5 10 15 20 25 30 35 40 45 50	The hot limit setting for the temperature alert.
<Alert />	geofenceState	alertGeofenceState	Off Simple Polygon	Geofencing mode
<Alert />	geofenceFreq	alertGeofenceFreq	1 2 3 5 10 15 30	The frequency, in minutes, that the tracker will check it's position with respect to it's geofence boundary.
<Alert />	geofenceRadius	alertGeofenceRadius	500 1000 2000 3000 5000	The radius in metres of the geofence boundary.

XML element	XML attribute	equivalent POST param	Allowed values	Notes
<Alert />	geofenceCentre	alertGeofenceCentre	yes no	If this is set to 'yes', it instructs the tracker to set its geofence centre to its current location. This option is relevant to sending configuration only.
<Alert />	powerLossState	alertPowerLossState	on off	Whether the power loss alert function is switched on or off.
<Bluetooth />	state	bluetoothState	on off	Whether Bluetooth is switched on or not.
<System />	encryption	systemEncryption	on off	Whether AES256 encryption is enabled or not.
<System />	logging	systemLogging	on off	Whether debug logging is switched on or not.
<GPS />	accuracy	gpsAccuracy	2D 3D	Whether to accept 2D fixes, or to require a 3D fix.
<GPS />	hotMode	gpsHotMode	on off	Whether GPS hot mode is switched on or off.
<GPS />	fixesRequired	gpsFixesRequired	1 5 10 30	How many consecutive good fixes are required before the position is accepted.
<GPS />	earlyness	gpsEarlyness	20 40 60 120 180 240	How many seconds does the GPS switch on before the scheduled transmission time.
<ScreenLock />	state	screenLockState	on off	Whether the screen is locked.

XML element	XML attribute	equivalent POST param	Allowed values	Notes
<ScreenLock />	stealthModeState	stealthModeState	on off	Whether stealth mode is enabled
<ScreenLock />	pin	screenLockPin	0-9999	Screen unlock code
<Info />	software	software		Software version (e.g. 1.2.3)