ROCK SEVEN MOBILE SERVICES - LOCATION TECHNOLOGY

# RockSTAR XML API

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

Version 3.0 - Updated 14th February 2014

**support@rock7mobile.com**

# Background

The RockSTAR XML API enables remote systems to interact with the RockSTAR satellite communication service.  It is intended to allow integration of the device into third-party back-office systems.

# Fair Use

While there are no hard limits in place, please use this API sensibly:

> Version 3.0 of this document introduces a Push API, which reduces or eliminates the need to 'poll' the API.  New integrations are *strongly* encouraged to make use of this feature.  While there are no immediate plans to deprecate the existing GetPositions method, it should be noted that the Push API will be more actively developed, and is likely to offer additional functionality in future.

- Please don't repeatedly request information that you've already received.
- Please be considerate when choosing polling intervals.  How often is really necessary for your application?
- Rock Seven reserve the right to immediately terminate your access if your usage is considered to be affecting performance of the API for other customers.

# Revision History

**3.0**
- Added Push API, enabling developers to have MO messages POSTed to a web service.
- No impact on existing integrations.

# 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 | 15 digits |
| device_type | Type of device.  This will usually be set to LEOPARD which refers to the family of products including RockSTAR and Yellowbrick v3. | LEOPARD ROCKBLOCK |
| 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 |
| transmit_time | Timestamp that the message was transmitted. | e.g. 2014-02-14T13:55:16Z |

| Parameter | Description | Example/Possible values |
|---|---|---|
| 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<br>MANUAL<br>ACTIVATION<br>CONFIG_REPORT<br>WAYPOINT<br>ACKNOWLEDGE<br>BLUETOOTH_LOSS<br>COLLISION<br>COUNTDOWN<br>DEAD_MAN<br>GEOFENCE<br>BUTTON<br>CANCEL_ALERT<br>POWER_LOSS<br>TEMPERATURE |
| source | An indicator of whether the transmission includes a valid GPS fix. | GPS<br>IRIDIUM |
| lat | GPS latitude | 50.88886 |
| lon | GPS longitude | -1.29198 |

| Parameter | Description | Example/Possible values |
|---|---|---|
| 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 |

# Position Reports

**To return ALL positions for ALL devices in your account:**
https://secure.rock7mobile.com/API2/GetPositions?username=xxxx&password=yyyy&pretty=yes

**To return all positions after our unique ID 5201300, for all devices:**
https://secure.rock7mobile.com/API2/GetPositions?id=5201300&username=xxxx&password=yyyy&pretty=yes

**To return ALL positions for tracker serial 2322:**
https://secure.rock7mobile.com/API2/GetPositions/2322?username=xxxx&password=yyyy&pretty=yes

**To return positions after our unique ID 5201300 for tracker serial 2322:**
https://secure.rock7mobile.com/API2/GetPositions/2322?id=5201300&username=xxxx&password=yyyy&pretty=yes

**To return positions after MOMSN 2450 for tracker serial 2322:**
https://secure.rock7mobile.com/API2/GetPositions/2322?momsn=2450&username=xxxx&password=yyyy&pretty=yes

**To return positions for all trackers in group with keyword 'team-two', after our unique ID 5201300:**
https://secure.rock7mobile.com/API2/GetPositions?group=team-two&id=5201300&username=xxxx&password=yyyy&pretty=yes

Your account manager will provide you with your username and password parameters.  Note that the pretty=yes parameter simply makes the output a bit more human-readable.

There is usually a limit of 1000 positions returned per request.  It is anticipated that this service will be polled quite frequently and so this will not present any problem.  If you think you've hit the limit, make another request with id param set to the maximum id seen so far, and repeat until you have fewer than 1000 positions returned.

An example of the output:

```
<TrackerData>
    <device serial="1149">
          <position id="5174721" momsn="831" at="2012-04-11T17:15:00Z" lat="51.58872" lon="0.31921" sog="0.1" cog="000"
alt="109" temp="19.0" batt="98" alert="false" />
```

The 'at' parameter will always be UTC.  Lat/Lon in decimal degrees.  Speed is in knots.  Altitude in metres.  Temperature in degrees.  Battery in percent.

# 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.

# MO (Mobile Originated) Messages

You will receive MO messages in the same way that you receive position reports, by calling /GetPositions

An example:

```
<position id="5410158" momsn="2587" at="2012-05-25T08:45:05Z" lat="18.88754" lon="39.82386" source="GPS" sog="16.0" cog="151"
alt="27" temp="28.0" batt="91" alert="false">
      <message ackRequest="false">What shall we do with the drunken sailor?</message>
</position>
```

The <message /> element will be present if there was a text message sent with a report. The ackRequest parameter (true/false) indicates whether or not the sender requested an acknowledgement that the message had been read by somebody (a human).  Note that there is no way of sending an 'acknowledgement' back to the device - you should simply send a regular message saying "Acknowledged", or more helpfully "*Acknowledged - 007 is on his way. Stay calm.*"

It is possible that the **<position />** element can contain a parameter **messageAck=123**.   This means that the user of the device has acknowledged a message that was sent to them.  The number refers to the message receipt ID, which is returned to you after you submit a message to the **/SendMessage** API method, described next.

# MT (Mobile Terminated) Messages

To send a message to a device, 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" \
           "http://secure.rock7mobile.com/API2/SendMessage/2322"

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

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

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.

# Configuration Reports

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

**To return the last-known configuration for tracker serial 2322:**
https://secure.rock7mobile.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>
```

Note that the **<Config />** element can also appear within a **<position />** element as part of a **/GetPositions** response.

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://secure.rock7mobile.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://secure.rock7mobile.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<br>off | Whether regular, automatic position reporting is switched on or off. |
| <Tracking /> | freq | trackingFreq | continuous<br>burst<br>5<br>10<br>15<br>20<br>30<br>60<br>90<br>120<br>180<br>240<br>360<br>480<br>720 | Frequency that the tracker will attempt to transmit it's position, in minutes. |
| <Tracking /> | burstFixFreq | trackingBurstFixFreq | 5<br>10<br>15<br>20<br>30<br>60<br>120<br>300<br>600<br>900 | How often will a GPS fix be recorded, when in burst mode, in seconds. |

| XML element | XML attribute | equivalent POST param | Allowed values | Notes |
|---|---|---|---|---|
| <Tracking /> | burstTransmitFreq | trackingBurstTransmitFreq | 1<br>2<br>5<br>10<br>15<br>30<br>60 | How often will a set of fixes be transmitted, when in burst mode, in minutes. |
| <Mailbox /> | state | mailboxState | on<br>off | Whether regular, automatic mailbox checking is switched on or off. |
| <Mailbox /> | freq | mailboxFreq | continuous<br>5<br>10<br>15<br>20<br>30<br>60<br>90<br>120<br>180<br>240<br>360<br>480<br>720 | Frequency that the tracker will attempt to check it's mailbox, in minutes. |
| <Alert /> | timerState | alertTimerState | on<br>off | Whether the timer alert function is switched on or off. |
| <Alert /> | timerTimeout | alertTimerTimeout | 5<br>10<br>15<br>30<br>60<br>120<br>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<br>off | Whether the dead man alert function is switched on or off. |
| <Alert /> | deadManTimeout | alertDeadManTimeout | 5<br>10<br>15<br>30<br>60<br>120<br>240 | The number of minutes after which the dead-man timer must be reset, otherwise an alert will be triggered. |
| <Alert /> | temperatureState | alertTemperatureState | on<br>off | Whether the temperature alert function is switched on or off. |
| <Alert /> | temperatureFreq | alertTemperatureFreq | 1<br>2<br>3<br>5<br>10<br>15<br>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<br>-35<br>-30<br>-25<br>-20<br>-15<br>-10<br>-5<br>0<br>5<br>10<br>15<br>20<br>25<br>30<br>35<br>40<br>45<br>50 | The cold limit setting for the temperature alert. |

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

| XML element | XML attribute | equivalent POST param | Allowed values | Notes |
| --- | --- | --- | --- | --- |
| <Alert /> | geofenceCentre | alertGeofenceCentre | yes<br>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<br>off | Whether the power loss alert function is switched on or off. |
| <Bluetooth /> | state | bluetoothState | on<br>off | Whether Bluetooth is switched on or not. |
| <System /> | encryption | systemEncryption | on<br>off | Whether AES256 encryption is enabled or not. |
| <System /> | logging | systemLogging | on<br>off | Whether debug logging is switched on or not. |
| <GPS /> | accuracy | gpsAccuracy | 2D<br>3D | Whether to accept 2D fixes, or to require a 3D fix. |
| <GPS /> | hotMode | gpsHotMode | on<br>off | Whether GPS hot mode is switched on or off. |
| <GPS /> | fixesRequired | gpsFixesRequired | 1<br>5<br>10<br>30 | How many consecutive good fixes are required before the position is accepted. |
| <GPS /> | earlyness | gpsEarlyness | 20<br>40<br>60<br>120<br>180<br>240 | How many seconds does the GPS switch on before the scheduled transmission time. |
| <ScreenLock /> | state | screenLockState | on<br>off | Whether the screen is locked. |

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