Application Programming Interface for the Radio Bridge Console

VERSION 1.0
DECEMBER 2018

TABLE OF CONTENTS

1. OVERVIEW

1.1. Introduction

The wireless sensors designed and manufactured by Radio Bridge provide full sensor to cloud solutions for Internet of Things (IoT) applications. The Radio Bridge Console is a web-based device manager that handles the provisioning, monitoring, and configuration of Radio Bridge sensors. The Console is not required in order to use Radio Bridge sensor hardware, but it eliminates much of the work required to properly interface to the sensors and allows end users to get to get up and running quickly. Third party applications can be developed "on top of" the Console with basic APIs. This document provides instructions for third party applications to communicate to the Radio Bridge Console via APIs.

For more information on the Console, visit https://console.radiobridge.com

1.2. Revision History

Table 1 Revision History

Revision	Date	Description
1.0	December 2018	Initial release of the document

1.3. Document Conventions

Table 2 Document Conventions

Font / Icon	Meaning
	Important notes
	Warnings and cautions

2. OVERVIEW

2.1. Message Types

There are two types of messages when communicating with Radio Bridge sensors or the web-based console, uplink and downlink. Uplink messages originate from the sensor and are received by the Console and/or third-party application. The uplink messages communicate sensor events, alerts, heath, and status of the sensor.

Downlink messages originate from the Console and/or third-party application and are received by the sensor. Downlink messages provide configuration information for the sensor, and after receiving this information the sensor will behave accordingly.

2.2. API Types

The APIs used by the Console are RESTful APIs that utilize a simple POST request. These POST requests apply to both the uplink and downlink messages.

2.3. API Location

To utilize the Console API, go to the Radio Bridge Console application at https://console.radiobridge.com and click on the "API" tab along the left side. From there you can use the uplink and downlink APIs as described in the following sections.

3. UPLINK CONFIGURATION

Within the API section of the Console, there are two input fields in the Uplink section as shown in the following table.

Table 3 Uplink API Fields

Field	Description
Uplink URL	The URL of the third-party application as defined by the user
Header Authorization Code	The user defined authorization code added to the header of the POST request

Both fields in the above table are required. Uplink messages will be sent to the user defined URL with the user defined authorization code in the header.

The parameters that are passed with the uplink request are defined in the following table.

Table 4 Uplink Parameters

Parameter	Description
deviceId	Unique Device ID of sensor
deviceName	Device Name defined in the Console
deviceType	Device Type
sensorSeqNumber	Message Sequence Number
eventType	The type of uplink message. This can include a sensor event or one of the following: RESET, SUPERVISOR, TAMPER, RATE_LIMIT_EXCEEDED, CURRENT_SENSOR_STATE, UNKNOWN
decodedMessage	Human readable decode of sensor message
eventTime	Time stamp of the event in UTC
rawBytes	Raw hex data received from the sensor
snr	Signal to Noise Ratio
rssi	Receive Signal Strength Indicator

4. DOWNLINK CONFIGURATION

Within the API section of the Console, there is a single read-only field which specifies the URL to which a third-party application can post downlink parameters to the Console. The downlink URL contains a unique key that is used to authenticate downlink messages posted to this user account.

The parameters that are passed with the downlink request are defined in the following table.

Table 5 Downlink Parameters

Parameter	Description
deviceId	Unique Device ID of the sensor

downlink	The 8-byte downlink message. See the corresponding sensor user manual for more detail.

To verify a successful queueing of a downlink message, there is a JSON response with the following format:

{"status":"true|false","message":"<Success> or <Error> message"}

The two parameters are defined in the following table.

Table 6 Downlink API Response Parameters

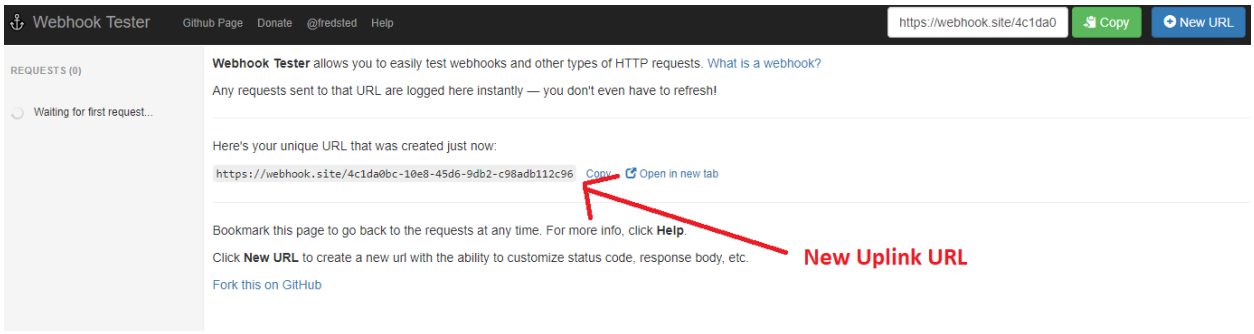
Parameter	Description
status	"true" indicates a successful queueing of a new downlink message. "false" indicates a failure to queue downlink message.
message	A message following the status of the API call indicating either a success or an error message.

5. TUTORIAL AND EXAMPLES

This section will step you through a simple method of testing the API and provide example values to the various parameters.

5.1. Setup

Go to the website https://webhook.site. This is a free webhook tester that provides visualization for the uplink and downlink messages. Click on the "New URL" button and then "Create". This creates a unique URL that can be used to post uplink messages. An example URL might look something like this:

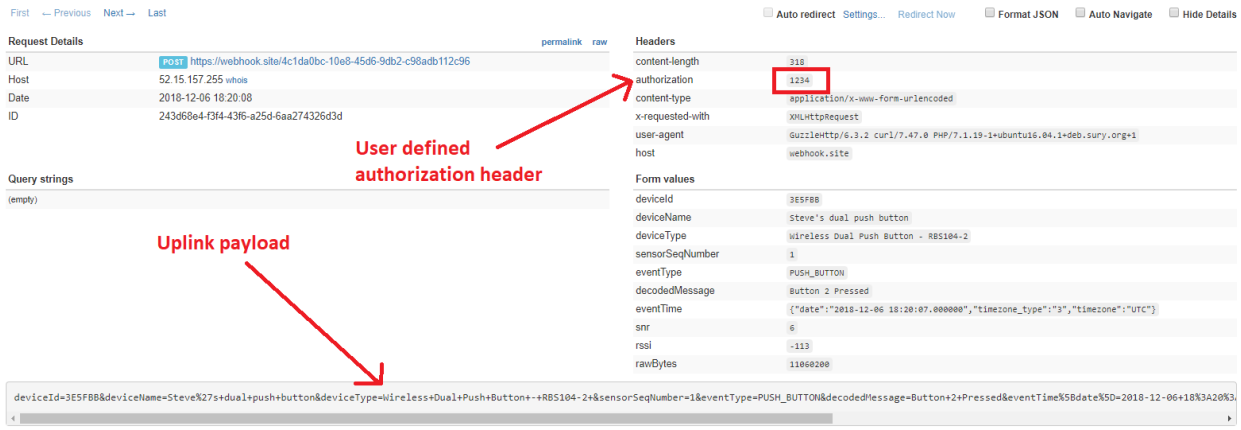


5.2. Configure Uplink

Go back to the Radio Bridge Console and click on the API tab. In the Uplink URL field, enter the URL generated in the last section. In the authorization field, enter any value your application will use for authorization. In this case, simply enter “1234” and click Update.

5.3. Test Uplink

Send a message from a Radio Bridge sensor and observe the data arriving at the webhook tester as shown below:



The sample data received in this example is shown in the following table.

Table 7 Uplink Example Parameters

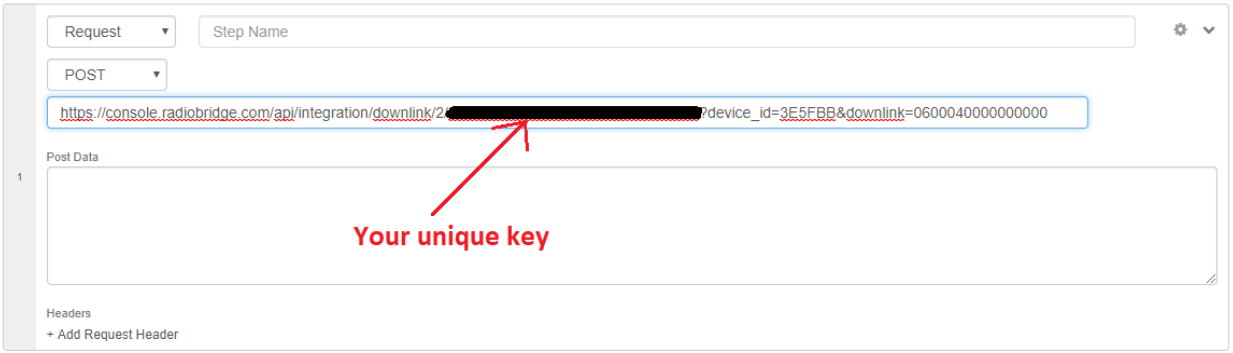
Parameter	Description
deviceId	3E5FBB
deviceName	Steve’s dual push button
deviceType	Wireless Dual Push Button - RBS104-2
sensorSeqNumber	1

eventType	PUSH_BUTTON
decodedMessage	Button 2 Pressed
eventTime	{“date”:”2018-12-06 18:20:07.000000”, “timezone_type”:”3”,”timezone”:”UTC”}
snr	6
rssi	-113
rawBytes	11060200

5.4. Test Downlink

We will send a downlink message: 0600040000000000 which indicates it is a dual push button device (06) with a hold time of 1 second (04) on button 1. Note that there must always be 8 bytes in a downlink message so we add zeros to the end of the message if no parameters are defined for those bytes.

Go to https://apitester.com, select POST, and enter the request string with the device ID and downlink payload as shown in the example below:



Scroll down to Response Body and you should see the following JSON response:

{"status":true,"message":"Event has been added in queue, device will be configured when that wakes up."}

Back in the Radio Bridge Console, navigate to the device ID and the following will be shown under the configuration section:

Refreshes in 00:51 Refresh Now

Event Data	Status	Published At
Configured via API: 0600040000000000	New Config Created	Dec 06 12:02

The downlink payload is queued and will be sent to the sensor.

6. TRADEMARKS AND COPYRIGHT

Radio Bridge™, SubGig®, and BridgeBee® are trademarks of Radio Bridge Inc in the United States.

7. DISCLAIMERS, WARRANTY, AND CUSTOMER SUPPORT

7.1. Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Radio Bridge. Radio Bridge provides this document "as is," without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Radio Bridge may make improvements and/or changes in this manual or in the product(s) and/or the software described in this manual at any time.

7.2. Warranty

To view product warranty information, go to the following website: www.radiobridge.com

7.3. Customer Support

Radio Bridge offers free technical support at:

www.radiobridge.com/forums

Radio Bridge also offers technical support plans and service packages to help our customers get the most out of their Radio Bridge products.

For information on Technical Support plans and pricing, visit us at www.radiobridge.com.