

AN-0004-WACS Sigfox backend configuration-EN-R08



Summary

Introduction	3
Requirements	3
Device type	3
Callbacks	3
DATA callback - BIDIR	3
SERVICE callback - DATA_ADVANCED	4
SERVICE callback - ACKNOWLEDGE	7
Revisions	8

Introduction

This document describes how to configure the callback to integrate the Sigfox backend with the Ayga dots platform.

Requirements

- Access to Sigfox backend configurations with rights to create Device type and callback

Device type

The downlink mode must be configured as “CALLBACK”

Downlink data

Downlink mode **CALLBACK** For more details on Downlink modes, please refer to documentation.

Expression must either include hexadecimal encoded bytes (ex: deadbeefcafebabe) or the following variables: - {time} 4 bytes - {tapid} 4 bytes - {rssi} 2 bytes - {roaming} 1 byte

Downlink data in hexa ⓘ

Callbacks

The device type to be used by WS2-0xx devices must have the following callbacks configured:

- DATA callback - BIDIR
- Service callback - DATA_ADVANCED
- Service callback - ACKNOWLEDGE

DATA callbacks					
Downlink	Enable	Channel	Subtype	Batch	Information
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		BIDIR	<input type="checkbox"/>	[GET] https://dots-api.ayga.com.br/sigfox/downlink?device={device}&data={data}&ack={ack}&station={station}&seqNumber={seqNumber}

SERVICE callbacks					
Enable	Channel	Subtype	Batch	Information	
<input checked="" type="checkbox"/>		DATA_ADVANCED	<input type="checkbox"/>	[POST] https://dots-api.ayga.com.br/sigfox	
<input checked="" type="checkbox"/>		ACKNOWLEDGE	<input type="checkbox"/>	[POST] https://dots-api.ayga.com.br/sigfox/ack	

DATA callback - BIDIR

- Type: DATA - BIDIR
- Channel: URL
- Custom payload config:
- Url pattern:
https://dots-api.ayga.com.br/sigfox/downlink?device={device}&data={data}&ack={ack}&seqNumber={seqNumber}
- Use HTTP Method: GET
- Headers

- header: apiKey
- value: (the business apiKey in which the devices are registered on the Ayga dots platform must be used)

Callbacks

Type

Channel

Custom payload config

URL syntax: `http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...`
 Available variables: `device, time, snr, station, data, rssi, seqNumber, deviceTypeid, ack, longPolling`
 Custom variables:

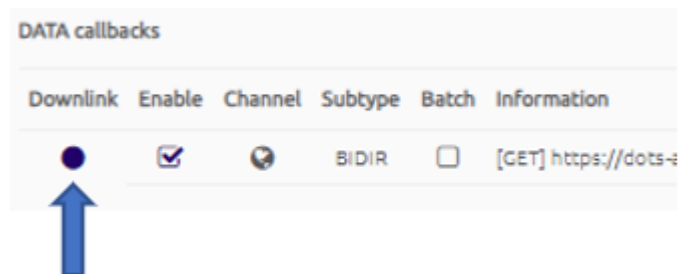
Url pattern

Use HTTP Method

Send SNI (Server Name Indication) for SSL/TLS connections

Header	Value
apiKey	your key
header	value

Please confirm that downlink is enabled for BIDIR callback.



SERVICE callback - DATA_ADVANCED

- Type: SERVICE- DATA_ADVANCED
- Channel: URL
- Custom payload config:
- Url pattern: `https://dots-api.ayga.com.br/sigfox`
- Use HTTP Method: POST
- Headers
 - header: apiKey
 - value: (the business apiKey in which the devices are registered on the Ayga dots platform must be used)
- Content type: `application/json`
- Body:

```
{
  "Device": "{device}",
  "Counter": "{seqNumber}",
  "Time": "{time} ",
  "Data": "{data} ",
  "LinkQuality": {linkQuality},
  "LQI": "{lqi}",
  "OperatorName": "{operatorName}",
  "CountryCode": {countryCode}
}
```

- If `computedLocation` is available, Ayga dots will use it as fall back option, please add it on Body:


```
{
  "Device" : "{device}",
  "Counter" : "{seqNumber}",
  "Time" : "{time}",
  "Data" : "{data}",
  "Location" : {computedLocation},
  "LinkQuality" : {linkQuality},
  "LQI" : "{lqi}",
  "OperatorName" : "{operatorName}",
  "CountryCode" : {countryCode}
}
```

Callbacks

Type: **SERVICE** | **DATA_ADVANCED**

The DATA ADVANCED callback is delivered with a delay of approximately 30 seconds.

Channel: **URL**

Custom payload config 

URL syntax: `http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...`
 Available variables: `device, time, data, seqNumber, lqi, linkQuality, fixedLat, fixedLng, operatorName, countryCode, deviceTypeId`
 Custom variables:

Uri pattern: `https://dots-api.ayga.com.br/sigfox`

Use HTTP Method: **POST**

Send SNI: (Server Name Indication) for SSL/TLS connections

Headers: header value

Content type: **application/json**

Body

```

{
  "Device" : "{device}",
  "Counter" : "{seqNumber}",
  "Time" : "{time}",
  "Data" : "{data}",
  "LinkQuality" : {linkQuality},
  "Lqi" : "{lqi}",
  "OperatorName" : "{operatorName}",
  "CountryCode" : {countryCode}
}
    
```

SERVICE callback - ACKNOWLEDGE

- Type: SERVICE- ACKNOWLEDGE
- Channel: URL
- Custom payload config:
- Url pattern: <https://dots-api.ayga.com.br/sigfox/ack>
- Use HTTP Method: POST
- Headers
 - header: apiKey
 - value: (the business apiKey in which the devices are registered on the Ayga dots platform must be used)
- Content type: application/json
- Body:

```
{
  "Device" : "{device}",
  "Time" : "{time}",
  "InfoCode" : "{infoCode}",
  "InfoMessage" : "{infoMessage}",
  "DownlinkAck" : "{downlinkAck}",
  "DownlinkOverusage" : "{downlinkOverusage}",
  "Station" : "not available"
}
```

Callbacks

Type: SERVICE | ACKNOWLEDGE

Channel: URL

URL syntax: <http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...>
Available variables: device, time, infoCode, infoMessage, downlinkAck, downlinkOverusage, downlinkPayload

Url pattern: <https://dots-api.ayga.com.br/sigfox/ack>

Use HTTP Method: POST

Send SNI (Server Name Indication) for SSL/TLS connections

Headers: header value

Content type: application/json

Body

```
{
  "Device" : "{device}",
  "Time" : "{time}",
  "InfoCode" : "{infoCode}",
  "InfoMessage" : "{infoMessage}",
  "DownlinkAck" : "{downlinkAck}",
  "DownlinkOverusage" : "{downlinkOverusage}",
  "Station" : "not available"
}
```