

Uplink Data Message

Contents

- [Basic verbosity](#)
- [Extended verbosity](#)

Basic verbosity

An "Uplink Data Message" is a LoRaWAN message received from a registered device on the Network Server which is sent to the defined [Application Output](#).

- Every received message will be sent the defined endpoint in the configured verbosity in a JSON format.

Every uplink JSON message corresponds to one frame produced by your device, It contains not only the raw application payload, but also additional meta-data.

- If APPSKEY is assigned to a device, the data field will be populated with decrypted data, and encdata will be omitted.
- If APPSKEY is missing, the encdata field will be populated with encrypted payload, and data will be omitted.

To change the verbosity have a look at the [Radio Verbosity](#) page.

Message format [Typescript interface notation](#)

```
{
  cmd      : 'rx'; // identifies type of message, rx = uplink message
  EUI      : string; // device EUI, 16 hex digits (without dashes)
  ts       : number; // server timestamp as number (milliseconds from Linux epoch)
  ack      : boolean; // acknowledgement flag as set by device
  bat      : number; // device battery status, response to the DevStatusReq
  LoRaWAN MAC Command
  fcnt     : number; // frame counter, a 32-bit number
  port     : number; // port as sent by the end device

  encdata? : string; // data payload (APPSKEY encrypted hex string)
              // only present if APPSKEY is not assigned to device

  data?    : string; // data payload (decrypted, plaintext hex string)
              // only present if APPSKEY is assigned to device
}
```

Example Message

```
{
  "cmd" : "rx",
  "EUI" : "0102030405060708",
  "ts"  : 1470850675433,
  "ack" : false,
  "bat" : 254
  "fcnt" : 1,
  "port" : 1,
  "data" : "0102AABB"
}
```

Parameter	Type	Description
<code>cmd</code>	string	identifies type of message, always 'rx' for uplink data messages
<code>EUI</code>	string	device EUI, 16 hex digits (without dashes)

ts	number	server timestamp as a number of milliseconds from Linux epoch
ack	boolean	acknowledgement flag as set by device
bat	number	device battery status, response of the DevStatusReq LoRaWAN MAC Command
fcnt	number	frame counter, a 32-bit integer number
port	number	port number as sent by the end device
encdata	string (optional)	encrypted data payload as hexadecimal string, only present if APPSKEY is not assigned to device
data	string (optional)	decrypted data payload as hexadecimal string, only present if APPSKEY is assigned to device

Extended verbosity

The LORIoT Network Server can deliver extended radio information to your application.

The extended radio information is reported as seen by the first gateway that received the frame.

Message format [Typescript interface notation](#)

```

{
  cmd      : 'rx'; // identifies type of message, rx = uplink message
  EUI      : string; // device EUI, 16 hex digits (without dashes)
  ts       : number; // server timestamp as number (milliseconds from Linux epoch)
  ack      : boolean; // NEW! acknowledgement flag as set by device
  fcnt     : number; // frame counter, a 32-bit number
  port     : number; // port as sent by the end device

  encdata? : string; // data payload (APPSKEY encrypted hex string)
                // only present if APPSKEY is not assigned to device

  data?    : string; // data payload (decrypted ,plaintext hex string)
                // only present if APPSKEY is assigned to device

  // extended radio information

  freq     : number; // radio frequency at which the frame was received, in Hz
  dr       : string; // radio data rate - spreading factor, bandwidth and coding rate
                // e.g. SF12 BW125 4/5
                // UPDATED: previously the field was called 'sf'
  rssi     : number; // frame rssi, in dBm, as integer number
  snr      : number; // frame snr, in dB, one decimal place
}

```

Example Message

```

{
  "cmd" : "rx",
  "EUI" : "0102030405060708",
  "ts"  : 1470850675433,
  "ack" : false,
  "fcnt" : 1,
  "port" : 1,
  "data" : "0102AABB",
  "freq" : 868500000,
  "dr"   : "SF12 BW125 4/5",
  "rssi" : -130,
  "snr"  : 1.2
}

```

Parameter	Type	Description
cmd	string	identifies type of message, always 'rx' for uplink data messages

<code>EUI</code>	string	device EUI, 16 hex digits (without dashes)
<code>ts</code>	number	server timestamp as a number of milliseconds from Linux epoch
<code>ack</code>	boolean	acknowledgement flag as set by device
<code>bat</code>	number	device battery status, response of the DevStatusReq LoRaWAN MAC Command
<code>fcnt</code>	number	frame counter, a 32-bit integer number
<code>port</code>	number	port number as sent by the end device
<code>encdata</code>	string (optional)	encrypted data payload as hexadecimal string, only present if APPSKEY is not assigned to device
<code>data</code>	string (optional)	decrypted data payload as hexadecimal string, only present if APPSKEY is assigned to device
<code>freq</code>	number	radio frequency at which the frame was received, in Hz
<code>dr</code>	string	radio data rate - spreading factor, bandwidth and coding rate, e.g. SF12 BW125 4/5
<code>rssi</code>	number	frame rssi, in dBm, as integer number
<code>snr</code>	number	frame snr, in dB, one decimal place