



**IMST GmbH**

Carl-Friedrich-Gauß-Str. 2-4, D-47475 Kamp-Lintfort

# iOKE868\_LoRaWAN

AN036 - Remote Access Protocol

Version 1.2

**Document State**

final

**Date**

October 2024

**Document ID**

4000/40140/0165

© 2024 IMST GmbH - All rights reserved

## History

Version	Datum	Chapter	Comment
1.0	19.04.2023	all	Initial Version
1.1	19.03.2024	Accessible Resources - Application Status	Reserved Info Field added to Application Status
1.2	02.10.2024	Accessible Resources - Reader Configuration	Byte 0 and 7 of Option Bits added to Reader Configuration

## Aim of this document

This document includes a description of the Remote Access Protocol which is supported by iO881A for LoRaWAN. This protocol enables applications to configure the infrared reading unit over the air from application server side.

### Notation Info

Suffix "b" = binary data

Suffix "h" = hexadecimal data

Without suffix = decimal data

Multi byte / octet fields are considered to be treated as unsigned integers with **Least Significant Byte** first unless explicitly noted

## Content

- 
- [Remote Access Services - Overview](#)
  - [Summary of Services](#)
  - [Accessible Resources](#)
-

# Remote Access Services - Overview

LoRaWAN end nodes like iO881A can be accessed through the LoRaWAN network server by means of so called downlink messages.

Many LoRaWAN network servers provide a RESTful api or gRPC interface which allows to enqueue such messages. Initiating a downlink message requires at least three information elements:

1. A mean to identify the LoRaWAN end device like the unique **64-Bit DeviceEUI**
2. A **LoRaWAN Port** number which is part of the LoRaWAN messages
3. The **message payload**

The next sub chapters describe the **message payload** which transports the service requests or corresponding response messages.

Note: LoRaWAN supports a **guaranteed payload capacity** of  $64 - 13 - 15 = 36$  bytes.

## Simple not segmented Services via LoRaWAN Port 20<sub>h</sub>

The following sub chapters describe the supported services in more detail:

Service	Description
<a href="#">Get</a>	Used to read a resource
<a href="#">Get Sub Items Count</a>	Used to read the number of sub items within a resource e.g. the number of configured Calendar Events
<a href="#">Get Sub Item</a>	Used to read a specific sub item within a list of items
<a href="#">Set</a>	Used to configure a resource
<a href="#">Set Sub Item</a>	Used to configure a specific sub item within a list of items
<a href="#">Add Sub Item</a>	Used to append a new sub item to a list
<a href="#">Delete</a>	Used to delete a resource
<a href="#">Delete Sub Item</a>	Used to delete a single sub item within a list

Table : Remote Access Services

[Back to Top](#)

## Remote Access Services - Details

The following sub chapters include a detailed description of the implemented services:

### Get

This service can be used to read a specific resource. In case of large resources the response might be transmitted as a series of segments which need to be reassembled before further processing.

#### Syntax:

Request: < Get > < Resource ID >

Response < Get Response > < Resource ID > < data >

#### Example: Get DateTime

Request

< Get > < DateTime >

< 01<sub>h</sub> > < 01<sub>h</sub> >

Response

< Get Response > < DateTime > < seconds since epoch ( 1.1.1970, midnight UTC/GMT ) as **unsigned** 32-Bit integer, LSB first >

< 02<sub>h</sub> > < 01<sub>h</sub> > < ( 35 6E 7D 5F )<sub>h</sub> >

## Get Sub Item Count

---

This service can be used to retrieve the number of sub items from a resource.

### Syntax:

Request: < Get Sub Items Count > < Resource ID >

Response < Get Sub Items Count Response > < Resource ID > < Count >

### Example: Get Calender Event Count

Request

< Get Sub Items Count > < Calendar >

< 03<sub>h</sub> > < 02<sub>h</sub> >

Response

< Get Sub Items Count Response > < Calendar > < 4 >

< 04<sub>h</sub> > < 02<sub>h</sub> > < 04<sub>h</sub> >

## Get Sub Item

---

This service can be used to read a single sub item with given index. The index is in the range 0 .. Count - 1

### Syntax:

Request: < Get Sub Item > < Resource ID > < Index >

Response < Get Sub Item Response > < Resource ID > < Index > < data >

### Example: Get Calendar Event Item

Request

< Get Sub Item > < Calendar > < Index : 03<sub>h</sub> >

< 05<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> >

Response

< Get Sub Item Response > < Calendar > < 03<sub>h</sub> > < Calendar Event Item Data >

< 06<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> > < Calendar Event Item Data >

## Set

---

This service can be used to configure a given resource.

For resources which provide a list of sub items the complete list of sub items will be overwritten!

**Syntax:**

Request: &lt; Set &gt; &lt; Resource ID &gt; &lt; data &gt;

Response &lt; Set Response &gt; &lt; Resource ID &gt; &lt; Status Code &gt;

**Example: Set DateTime**

Request

< Set > < DateTime > < seconds since epoch ( 1.1.1970, midnight UTC/GMT ) as **unsigned** 32-Bit integer, LSB first >< 07<sub>h</sub> > < 01<sub>h</sub> > < ( 35 6E 7D 5F )<sub>h</sub> >

Response

&lt; Set Response &gt; &lt; DateTime &gt; &lt; Success &gt;

< 08<sub>h</sub> > < 01<sub>h</sub> > < 00<sub>h</sub> >**Set Sub Item**

---

This service can be used to change a given sub item of a given resource. In case the sub item doesn't exist a new sub item will be created and appended. The returned index will point to the new created item.

**Syntax:**

Request: &lt; Set Sub Item &gt; &lt; Resource ID &gt; &lt; Index &gt; &lt; data &gt;

Response &lt; Set Sub Item Response &gt; &lt; Resource ID &gt; &lt; Index &gt; &lt; Status Code &gt;

**Example: Set Calendar Event Item**

Request

< Set Sub Item > < Calendar > < 03<sub>h</sub> > < Calendar Event Item Data >< 09<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> > < Calendar Event Item Data >

Response

&lt; Set Sub Item Response &gt; &lt; Calendar &gt; &lt; Success &gt;

< 0A<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> > < 00<sub>h</sub> >**Add Sub Item**

---

This service can be used to create and append a new sub item of a given resource. The response contains the index of the new created item.

**Syntax:**

Request: &lt; Add Sub Item &gt; &lt; Resource ID &gt; &lt; data &gt;

Response &lt; Add Sub Item Response &gt; &lt; Resource ID &gt; &lt; Index &gt; &lt; Status Code &gt;

**Example: Add Calendar Event Item**

Request

&lt; Add Sub Item &gt; &lt; Calendar &gt; &lt; Calendar Event Item Data &gt;

< 0B<sub>h</sub> > < 02<sub>h</sub> > < Calendar Event Item Data >

Response

&lt; Add Sub Item Response &gt; &lt; Calendar &gt; &lt; Index: 4 &gt; &lt; Success &gt;

< 0C<sub>h</sub> > < 02<sub>h</sub> > < 04<sub>h</sub> > < 00<sub>h</sub> >

## Delete

---

This service can be used to delete a given resource.

For resources which provide a list of sub items the complete list of sub items will be deleted!

### Syntax:

Request: < Delete > < Resource ID >

Response < Delete Response > < Resource ID > < Status Code >

### Example: Delete All Calendar Events

Request

< Delete > < Calendar all Events >

< 0D<sub>h</sub> > < 02<sub>h</sub> >

Response

< Delete Response > < Calendar all Events > < Success >

< 0E<sub>h</sub> > < 02<sub>h</sub> > < 00<sub>h</sub> >

## Delete Sub Item

---

This service can be used to delete a single sub item of given resource.

### Syntax:

Request: < Delete Sub Item > < Resource ID > < Index >

Response < Delete Sub Item Response > < Resource ID > < Index > < Status Code >

### Example: Delete single Calendar Event Item

Request

< Delete Sub Item > < Calendar > < Index: 3 >

< 0F<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> >

Response

< Delete Sub Item Response > < Calendar > < Success >

< 10<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> > < 00<sub>h</sub> >

[Back to Remote Access Services - Overview](#)

## Summary of Services

This table gives a summary of the implemented services, their service codes and syntax:

1. Unknown Service Request Codes will be answered with a Status Response, corresponding Resource ID and Status Code "Service not available".
2. Service Requests for unknown Resources IDs will be answered with a corresponding Response and Status Code "Resource not found".

#	Name	Request Code	Response Code	Syntax	
				Request	Response
1	Status Response	n.a.	00 <sub>h</sub>		< Status Response > < Resource ID > < Status Code >
2	Get	01 <sub>h</sub>	02 <sub>h</sub>	< Get > < Resource ID >	< Get Response > < Resource ID > < data >
3	Get Sub Items Count	03 <sub>h</sub>	04 <sub>h</sub>	< Get Sub Items Count > < Resource ID >	< Get Sub Items Count Response > < Resource ID > < Count >
4	Get Sub Item	05 <sub>h</sub>	06 <sub>h</sub>	< Get Sub Item > < Resource ID > < Index: 0..Count -1 >	< Get Sub Item Response > < Resource ID > < Index > < data >
5	Set	07 <sub>h</sub>	08 <sub>h</sub>	< Set > < Resource ID > < data >	< Set Response > < Resource ID > < Status Code >
6	Set Sub Item	09 <sub>h</sub>	0A <sub>h</sub>	< Set Sub Item > < Resource ID > < Index > < data >	< Set Sub Item Response > < Resource ID > < Index > < Status Code >
7	Add Sub Item	0B <sub>h</sub>	0C <sub>h</sub>	< Add Sub Item > < Resource ID > < data >	< Add Sub Item Response > < Resource ID > < Index > < Status Code >
8	Delete	0D <sub>h</sub>	0E <sub>h</sub>	< Delete > < Resource ID >	< Delete Response > < Resource ID > < Status Code >
9	Delete Sub Item	0F <sub>h</sub>	10 <sub>h</sub>	< Delete Sub Item > < Resource ID > < Index >	< Delete Sub Item Response > < Resource ID > < Index > < Status Code >

Table : Summary of Services

## Status Codes

The following table lists the possible returned **Status Codes**:

#	Status	Code
1	Success	00 <sub>h</sub>
2	Failure	01 <sub>h</sub>
3	Resource not found	02 <sub>h</sub>
4	Sub Item Index not found	03 <sub>h</sub>
5	Service not available	04 <sub>h</sub>

Table : Status Codes

[Back to Top](#)

# Accessible Resources

The following table lists the accessible resources, their corresponding identifier (Resource ID) and the provided services:

#	Resource	Resource ID	Services							
			Get	Get Sub Item Count	Get Sub Item	Set	Set Sub Item	Add Sub Item	Delete	Delete Sub Item
1	DateTime	01 <sub>h</sub>	yes	no	no	yes	no	no	no	no
2	Calendar Events	02 <sub>h</sub>	no	yes	yes	no	yes	yes	yes	yes
3	Application Status	03 <sub>h</sub>	yes	no	no	no	no	no	no	no
4	OBIS ID Filter	07 <sub>h</sub>	no	yes	yes	no	yes	yes	yes	yes
5	Reader Configuration	0A <sub>h</sub>	yes	no	no	yes	no	no	no	no

Table : Accessible Resources

[Back to Top](#)



## Accessible Resources - DateTime

The DateTime resource enables access to the current date and time of the WM-Bus Range Extender. The DateTime resource can be "Get" and "Set".

The data format looks as follows:

Resource	DateTime
Format	UTC, seconds since 01.01.1970, 00:00:00
Octets	32 Bit, unsigned Integer, LSB first
Example	( 19 9E 64 5F ) <sub>h</sub> 5F649E19 <sub>h</sub> = 1.600.429.593 seconds since 01.01.1970, 00:00:00 "2020-09-18 11:46:33"

Table : DateTime Resource

### Example Set DateTime

< Set > < Resource ID > < DateTime >

< 07<sub>h</sub> > < 01<sub>h</sub> > < ( 19 9E 64 5F )<sub>h</sub> >

Back to [Accessible Resources](#)

## Accessible Resources - Calendar Events

The Calendar resource enables access to the configurable Calendar Events. It is possible to get, set and delete the complete list of all sub items by means of the Tiny Transport Protocol (see User Manual) or to set, get and delete only one sub item per request.

The data format of a single sub item looks as follows:

Resource	Calendar Event				
Format	Event-ID	Filter Group-ID	Repetition Type	Repetition Step Size	Date & Time (UTC)
Octets	8 Bit	16 Bi, LSB first	8 Bit	8 Bit	32 Bit, LSB first
Example	see <a href="#">Application Events</a>	( 01 00 ) <sub>h</sub> : 0001 <sub>h</sub> => Group 1 ( 02 00 ) <sub>h</sub> : 0002 <sub>h</sub> => Group 2 ( 04 00 ) <sub>h</sub> : 0004 <sub>h</sub> => Group 3 ( 08 00 ) <sub>h</sub> : 0008 <sub>h</sub> => Group 4  ( 09 00 ) <sub>h</sub> : 0009 <sub>h</sub> => Group 1 + Group 4 ( 06 00 ) <sub>h</sub> : 0006 <sub>h</sub> => Group 2 + Group 3	03 <sub>h</sub> = Daily	00 <sub>h</sub>	( 19 9E 64 5F ) <sub>h</sub>  see <a href="#">DateTime</a>

Table : Calendar Event

- Event-ID**  
 The event type defines the kind of action to be performed. A list of possible Event Types is given here: [Application Events](#)
- Filter Group-ID**  
 This element is only used in combination with OBIS ID Filters. It defines which OBIS-ID Filter Groups are applied during the meter reading process.  
 The Filter Group-IDs are interpreted bitwise and can be logically ored within these Calendar Events.  
 Note: The value FFFF<sub>h</sub> is reserved and means that all configured OBIS-ID Filter Groups should be applied.
- Repetition Type**  
 The repetition type defines the periodicity of an event:  
 0 = No repetition, single event, can be used for test purpose  
 1 = Every Minute  
 2 = Hourly  
 3 = Daily  
 4 = Weekly  
 5 = Monthly
- Repetition Step Size**  
 The repetition step size is a second parameter which defines the periodicity of an event:  
 Example 1: Repetition Type = 2 ( Hourly ), Repetition Step Size = 2 => Repetition Interval = every 2 + 1 = 3 hours  
 Example 2: Repetition Type = 3 ( Daily ), Repetition Step Size = 0 => Repetition Interval = every 0 + 1 = 1 days
- Date & Time**  
 The date / time element defines when the event should be scheduled for the first time.

### Example Add Calendar Event SubItem

Request:

< Add Sub Item > < Calendar Event > < Calendar Event Sub Item Data: Single Read, Filter Group 1, Daily, first schedule 2020-09-18 11:46:33 >

< 0B<sub>h</sub> > < 02<sub>h</sub> > < ( 51 01 00 03 00 19 9E 64 5F )<sub>h</sub> >

Response:

< Add Sub Item Response > < Calendar Event > < Index: 1 > < Success >

< 0Ch > < 02<sub>h</sub> > < 01<sub>h</sub> > < < 00<sub>h</sub> >

Back to [Accessible Resources](#)

## Accessible Resources - Application Events

The following table lists all application events, which should be used for remote configuration of calendar events

Event Name	Event ID	Description
<b>System Events</b>		
Get Network Time	30 <sub>h</sub>	Request system time via LoRaWAN MAC command. On response the system time will be synchronized.
Send Application Status	31 <sub>h</sub>	Transmit Application Status via LoRaWAN
<b>Meter Reader Events</b>		
Reader Off	50 <sub>h</sub>	Disables the optical reader unit
Single Reading Request	51 <sub>h</sub>	Enables the optical reading unit for a single reading procedure
Continuous Reading Request	52 <sub>h</sub>	Enables the optical reading unit for a continuous reading procedure
Scan OBIS-IDs Request	53 <sub>h</sub>	Enables the optical reading unit for an OBIS-ID scan procedure

Table : Application Events

[Back to Accessible Resources - Calendar Events](#)

## Accessible Resources - Application Status

The Application Status includes the following information elements:

Resource	Application Status								
Format	Date and Time (UTC)	Firmware Version	Last Sync Time	Reset Counter <sup>1)</sup>	Status Bits	Reader Rx Counter <sup>2)</sup>	Reader Error Counter <sup>2)</sup>	Reader Tx Counter <sup>2)</sup>	Reserved Info Firmware 1.3 ff
Octets	32 Bit, LSB first	16 Bit, Minor version first	32 Bit, LSB first	32 Bit, LSB first	16 Bit, LSB first	32 Bit, LSB first	32 Bit, LSB first	32 Bit, LSB first	8 Bit
Example	see <a href="#">DateTime</a>	e.g. ( 07 01 ) <sub>h</sub> v1.7	see <a href="#">DateTime</a>						

- **Date and Time**  
Contains the current date and time in seconds since 01.01.1970 00:00:00
- **Firmware Version**  
Minor and major firmware version
- **Last Sync Time**  
Contains the time stamp of the latest synchronization via local or air interface
- **Reset Counter<sup>1)</sup>**  
Contains the number of device resets
- **Status Bits**  
This field includes several Status Bits:  
  - Bit 0 : 1 = LoRaWAN Stack is not activated
  - Bit 1 : 1 = Network Time is not synchronized
  - Bit 2 : 1 = System Time is not synchronized
  - Bit 3 : 1 = LoRaWAN Activation Procedure is ongoing
  - Bit 4 : 1 = LoRa Configuration is invalid
  - Bit 5 : reserved
  - Bit 6 : 1 = Calendar event list is empty
  - Bit 7 : 1 = OBIS ID filter List is empty
- **Reader Rx Counter<sup>2)</sup>**  
Number of succesful received meter files
- **Reader Error Counter<sup>2)</sup>**  
Number of failed meter readings
- **Reader Tx Counter<sup>2)</sup>**  
Number of forwarded meter readings

### Info

<sup>1)</sup> The Reset Counter is copied to the non-volatile memory earliest 30 seconds after system start.

<sup>2)</sup> The Reader Rx/Tx/Error counters are written into the non-volatile memory earliest 30 seconds after last increment. These counters can be reset by means of an [HCI message](#).

[Back to Accessible Resources](#)

## Accessible Resources - OBIS-ID Filter Items

A single WM-Bus Device Filter Item includes the following information elements:

Resource	OBIS-ID Filter Item							
Format	OBIS-ID Groups						OBIS-ID Mask	Filter Group-ID
	Group A	Group B	Group C	Group D	Group E	Group F		
Octets	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 bit	8 Bit	16 Bit, LSB first
Example 1	1	0	1	8	0	255	FC <sub>h</sub> => all 6 Groups applied	( 01 00 ) <sub>h</sub>
Example 2	0	0	1	8	0	0	30 <sub>h</sub> => only Group C,D	( 01 00 ) <sub>h</sub>

If a list is requested via Get service, then the individual items are transferred in compressed form:

Resource	OBIS-ID Filter Item								
Format	OBIS-ID Mask	OBIS-ID Value						OBIS-ID Mask	Filter Group-ID
Octets	8 Bit	1 - max. 6 Bytes						8 Bit	16 Bit, LSB first
Example 1	FC <sub>h</sub> => all 6 Groups applied	1	0	1	8	0	255	FC <sub>h</sub> => all 6 Groups applied	( 01 00 ) <sub>h</sub>
Example 2	30 <sub>h</sub> => only Group C,D	1			8			30 <sub>h</sub> => only Group C,D	( 01 00 ) <sub>h</sub>

- **OBIS-ID Groups**

A sequence of 6 bytes in total which identifies the requested OBIS-ID to read.

- **OBIS-ID Mask**

This mask defines which of the OBIS-ID Group fields are used for comparison with every decoded meter record.

Bit 7 = Group A

Bit 6 = Group B

Bit 5 = Group C

Bit 4 = Group D

Bit 3 = Group E

Bit 2 = Group F

Bit 1 .. 0 = reserved

- **Filter Group-ID**

This element is only used to group several filter items per Calendar Event. Filter Group-IDs are interpreted bitwise and can be logically ored in a Calendar Event ( see Example [Accessible Resources - Calendar Events](#) ).

Note: The value FFFF<sub>h</sub> is reserved and means that this filter item should be always applied.

### Example: Add OBIS ID Filter Item

Request

< Add Sub Item Request > < OBIS-ID Filter > < OBIS-ID Filter Item Data >

< 0B<sub>h</sub> > < 07<sub>h</sub> > < ( 01 00 01 08 00 FF FC 01 00 )<sub>h</sub> >

Response

< Add Sub Item Response > < OBIS-ID Filter > < Index: 1 > < Success >

< 0C<sub>h</sub> > < 07<sub>h</sub> > < 01<sub>h</sub> > < 00<sub>h</sub> >

### Example: Set Sub OBIS ID Filter Item

Request

< Set Sub Item Request > < OBIS-ID Filter > < Index: 2 > < OBIS-ID Filter Item Data >

< 09<sub>h</sub> > < 07<sub>h</sub> > < 02<sub>h</sub> > < ( 00 00 01 08 00 00 30 01 00 )<sub>h</sub> >

Response

< Set Sub Item Response > < OBIS-ID Filter > < Index: 2 > < Success >

< 0A<sub>h</sub> > < 07<sub>h</sub> > < 02<sub>h</sub> > < 00<sub>h</sub> >

**Example: Get OBIS ID Filter Items**

Request

< Get Request > < OBIS-ID Filter >

< 01<sub>h</sub> > < 07<sub>h</sub> >

Response

< Get Response > < OBIS-ID Filter > < OBIS-ID Filter Item Data 1 > < OBIS-ID Filter Item Data 2 >

< 02<sub>h</sub> > < 07<sub>h</sub> > < ( FC 01 00 01 08 00 ff 01 00 )<sub>h</sub> > < ( 30 01 08 01 00 )<sub>h</sub> >

Back to [Accessible Resources](#)

## Accessible Resources - Reader Configuration

This resource provides some configuration parameters which control the behaviour of the optical reader unit and the radio protocol payload format which is described in document *IOKE868\_LoRaWAN\_AN035\_RadioProtocol*.

Resource	Reader Configuration			
Format	Options Bits	Protocol	Baudrate	Paired Meter-ID ( only for Get Service )
Octets	32 Bit, LSB first	8 Bit	8 Bit	n * 8 Bit
Example	( 06 00 00 00 ) <sub>h</sub> => 0000 0006 <sub>h</sub> : include Meter ID in Meter Data Messages + Status Field	0 = Auto Detection	5 = 9600 bps	

### • Options Bits

This field includes several configuration bits:

Bit 0 : Battery Mode

- 0 = Battery Mode is disabled
- 1 = Battery Mode is enabled

Bit 1 : Include optional Meter-ID Field in Meter Data Messages

- 0 = Meter-ID Field not added
- 1 = Meter-ID Field is added in uplink messages

Bit 2 : Include optional Status Field in Meter Data Messages

- 0 = Status Field not added
- 1 = Status Field is added in uplink messages

Bit 3 : Include optional Time Field in Meter Data Messages

- 0 = Time Field not added
- 1 = Time Field is added in uplink messages

Bit 4 : Omit Unit Information in Meter Object Fields

- When it's possible to derive the unit by means of the OBIS-ID itself this option might be helpful to reduce the uplink payload.
- 0 = Unit Information Field is present in Meter Object Fields
- 1 = Unit Information Field is not added in Meter Object Fields

Bit 5: IEC Number Conversion

- IEC numbers are usually given in ASCII format which requires a single byte per digit.
- This configuration option enables a conversion function from ASCII to integer which helps to reduce the uplink payload.
- 0 = Optimization off
- 1 = Optimization on

Bit 6 : IEC Wake-up Procedure ( IEC Mode only )

This option can be used to configure the fast wake-up procedure which is needed for battery powered meter devices which operate in IEC mode.

- 0 = Fast Wake-up procedure off
- 1 = Fast Wake-up procedure on

Bit 7 : Uplink type

- 0 = always use confirmed uplinks
- 1 = try unconfirmed uplink for status and meter packets

Bits 8 ..31 : reserved

### • Reader Protocol

This parameter defines the applied reader protocol.

- 0 = Auto Detection, the reader unit tries to detect one of the implemented uni- and bidirectional protocols
- 1 = IEC Mode A ( bidirectional, 300 bps )
- 2 = IEC Mode B ( bidirectional, 300 bps + dynamic baudrate change defined by meter )
- 3 = IEC Mode C ( bidirectional, 300 bps + dynamic baudrate change defined by meter )
- 4 = IEC Mode D ( unidirectional, baudrate manufacturer specific e.g.: 9600 bps )
- 5 = reserved

6 = SML ( default, unidirectional, typical with baudrate of 9600 bps )

- **Baudrate**

This parameter can be used to select a specific reader baudrate for meter operating in unidirectional IEC Mode D or SML.

0 = 300 bps

1 = 600 bps

2 = 1200 bps

3 = 2400 bps

4 = 4800 bps

5 = 9600 bps ( default )

6 = 19200 bps

7 = 38400 bps

8 = 57600 bps

9 = 115200 bps

- **Paired Meter-ID**

This information field is only present in the "Get Response" message. It includes the Meter-ID to which the reader is paired.

Back to [Accessible Resources](#)