orange™ Business Services

Live Objects

Orange

User manual

orange Business Services

Table of content

1. Introduction

1.1. Context

Live Objects is software as a service (SaaS) who provides a set of tools for Machine To Machine (M2M) and Internet of things (IoT).

Main features provided by Live Object are:

- Advanced management of connected devices (provisioning, supervision, configuration, firmware update, campaign…)
- Application programming interfaces (API) to connect devices and business applications, and manage all Live Objects features
- Message routing
- Data management (event/state processing, dataviz, search, data message enrichment)
- Data storage

Important: Depending on subscribed offer (LoRa, Discover, Express, Advanced, Premium) some features may be available or not. In this document all features are presented.

1.2. Purpose of this document

This user manual is a support to the use of Orange Live Objects portal. Some features only available by API are not described here, but in the developer guide referenced in the table below.

1.3. References

N°	title	reference
Orange documents		
1	Live Objects Developer Guide (API)	https://liveobjects.orange-business.com/doc/html/lo_manual.html
2	Live Objects complete training	https://liveobjects.w3b-experience.net/app/uploads/2018/07/liveobjectscompletetrainingandapisv7.pdf
3	Swagger (API reference)	https://liveobjects.orange-business.com/#/swagger

2. Live Objects portal user manual

2.1. Access

2.1.1. Get a Live Objects account

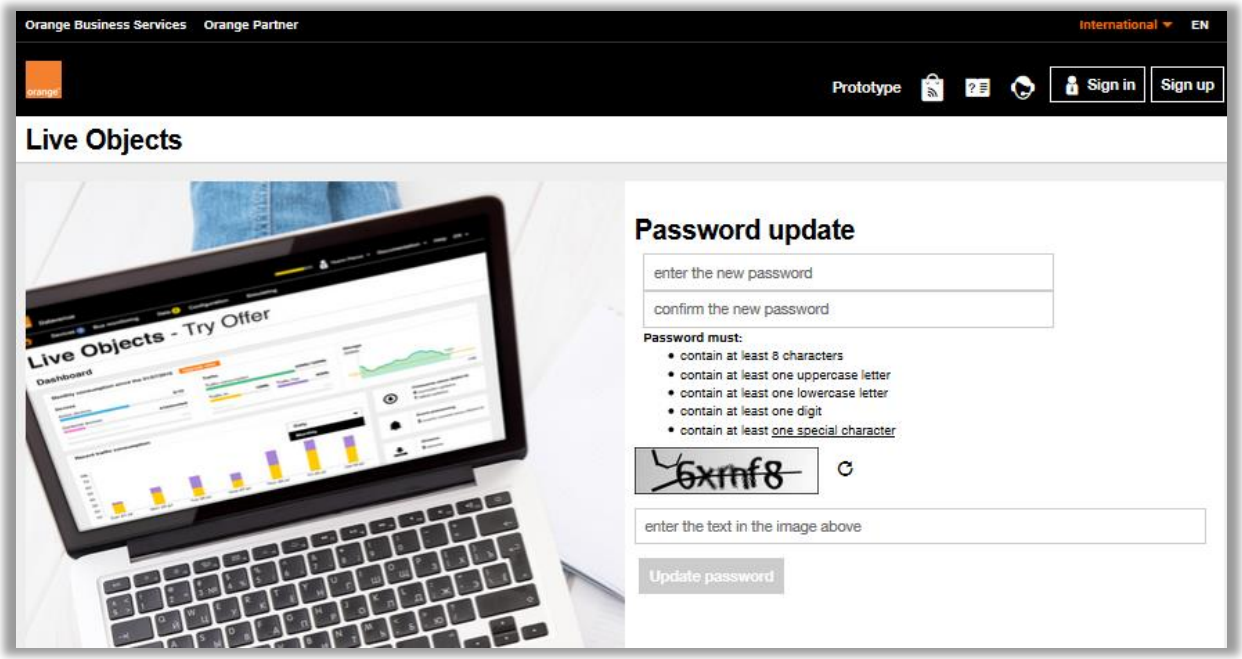
You can get a trial account (Discover) online at https://liveobjects.orange-business.com/#/request_account.

If you want an account with LoRa connectivity included, check the corresponding checkbox. In this case the creation of the account is not immediate, and you will be notified by email when ready.

Then, you can contact a commercial interlocutor at liveobjects.contact@orange.com

2.1.2. Credentials

The creation of your Live Objects account triggers an activation email. This email contains an activation link valid for 7 days. When you will activate your account you will have to set your password.



The password must contain 8 characters with uppercase, lowercase, numbers and special characters (/*-+.?,;:§!%ùµ*$@_()').

Then a new email is issued to confirm your account creation. It contains the identifier you will use to log into Live Objects portal.

2.1.3. Log into Live Objects portal

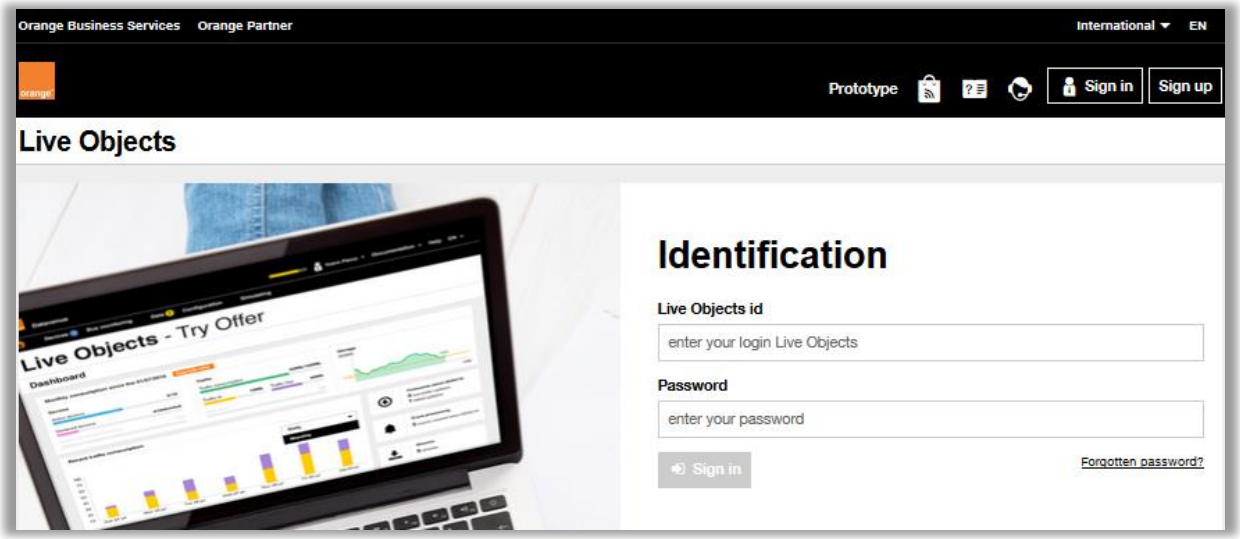
2.1.3.1. Supported browsers

Supported browsers are:

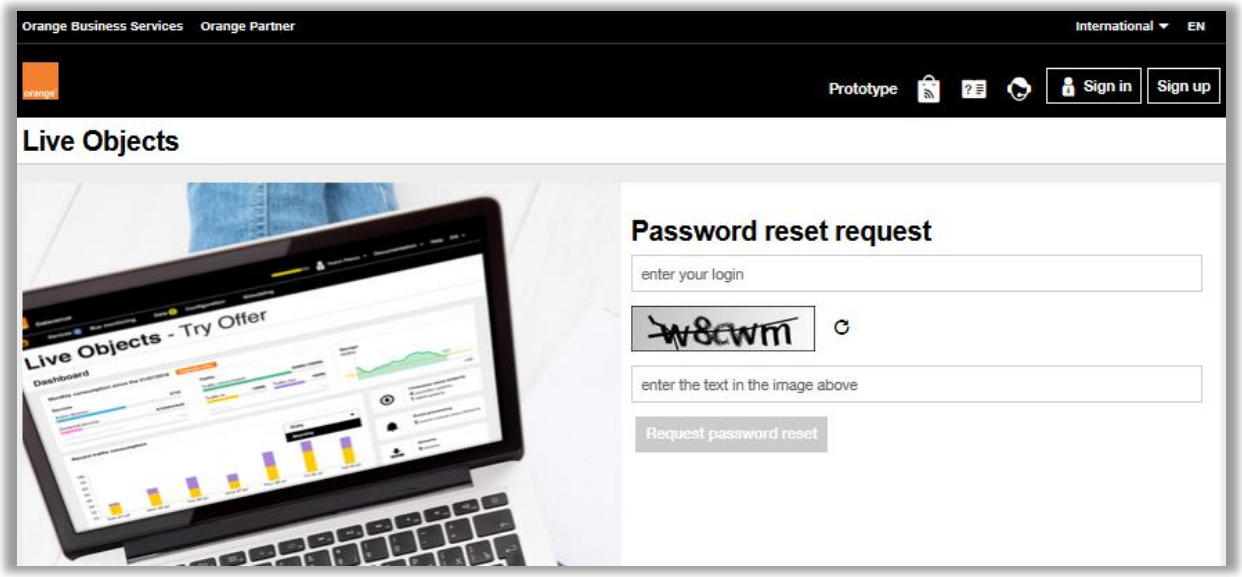
- Firefox 38 and over
- IE 11,
- Edge 13 and over
- Chrome 49 and over

2.1.3.2. Login

Go to https://liveobjects.orange-business.com



Connect with the identifier you have received in your welcome email, and the password you set during the activation process. If you don't remember it, use the link "forgotten password?"



To reset your password type your login into the form, then an email is issued to the email address of the account. This email contains a reset link valid only for 10 minutes.

2.2. Organization of the portal

2.2.1. Header

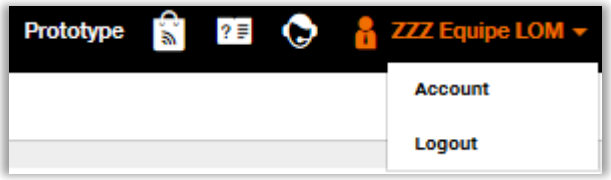
The header allows diving into the main areas of Live Objects portal:

- « Dashboard» is the welcome page, which gives a synthetic view of the account activity
- « Devices » to manage the fleet of devices
- « Data » to access the data stored into Live Objects
- « Configuration » for Live Objects account configuration (API keys, users, message queues, decoders…)

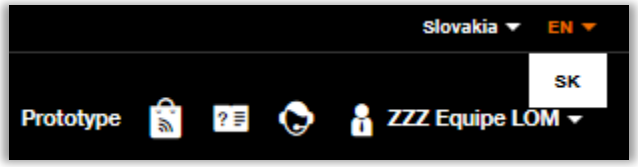
It also gives access to number of resources and support:

- One link « Prototype » where you can find some development kit (SDK)
- One link to the Orange market place where you will find a large choice of devices, modules and gateway usable with Live Objects.
- One link to a set of documentation on Live Objects: FAQ, developer guide, this guide, tutorials, videos, and tools for developers (Postman collections, swagger, code samples)
- One link to log-off, and to get to your account information's:
    - Tenant ID (to be communicated when calling to customer support)

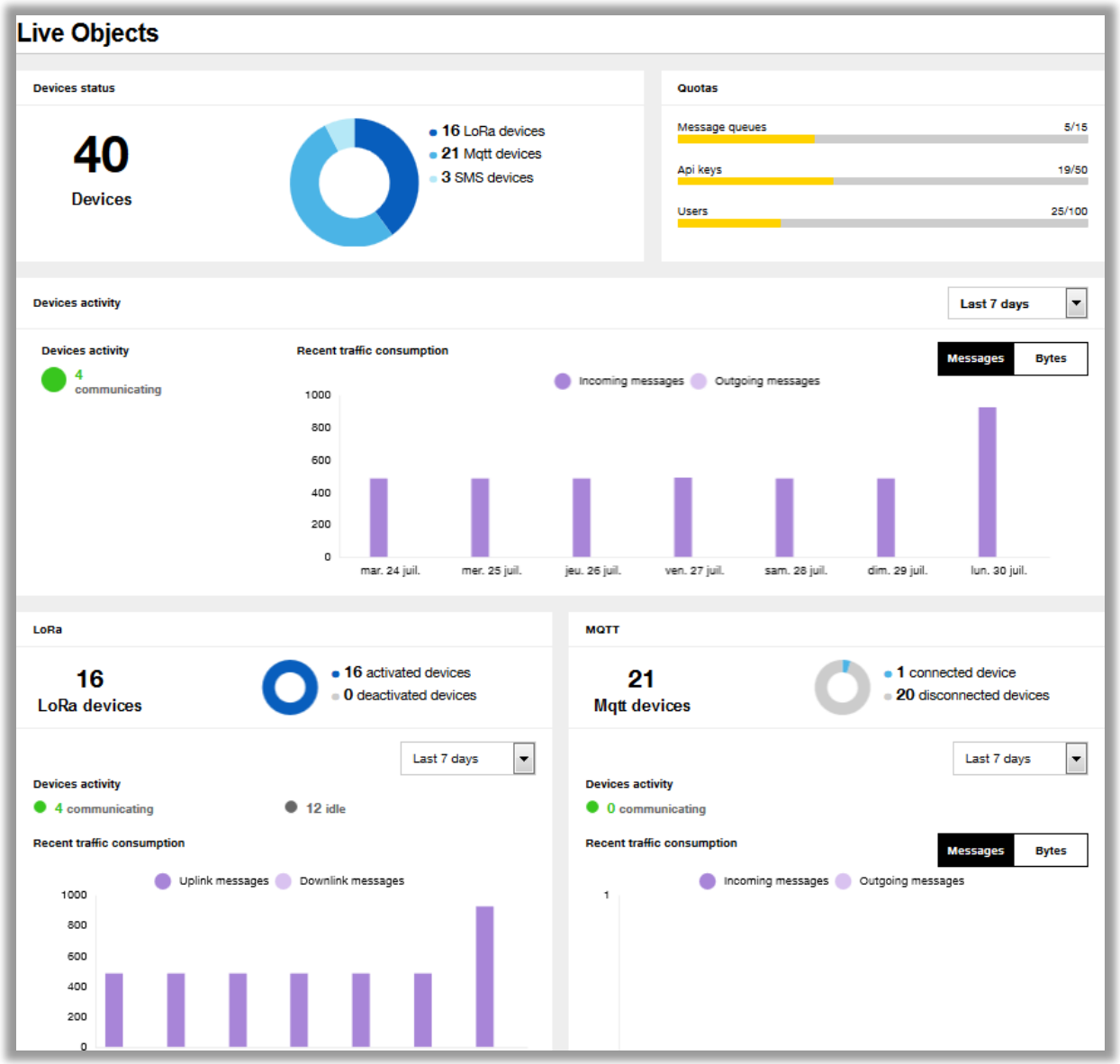
- Name of the account, date of creation and email of the principal user (used for the creation of the Live Objects account)



Depending on your country you can switch at any moment from English to you native language. Live Objects refresh automatically

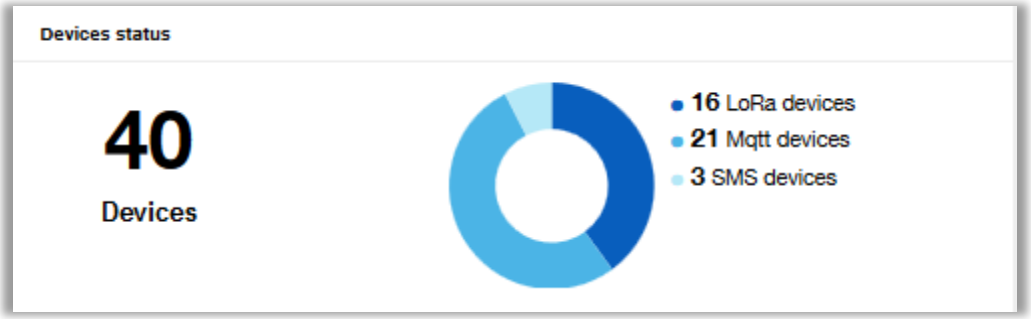


2.3. Dashboard



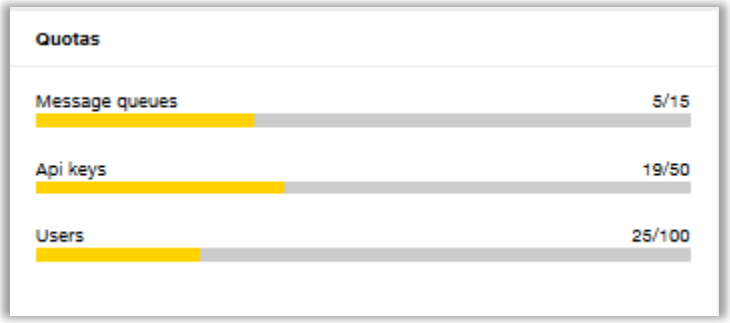
The dashboard is the welcome page of the portal. It gives a synthetic view of the account activity

2.3.1. Total number of devices per connectivity



This view gives the number of devices currently managed per connectivity.

2.3.2. Quotas

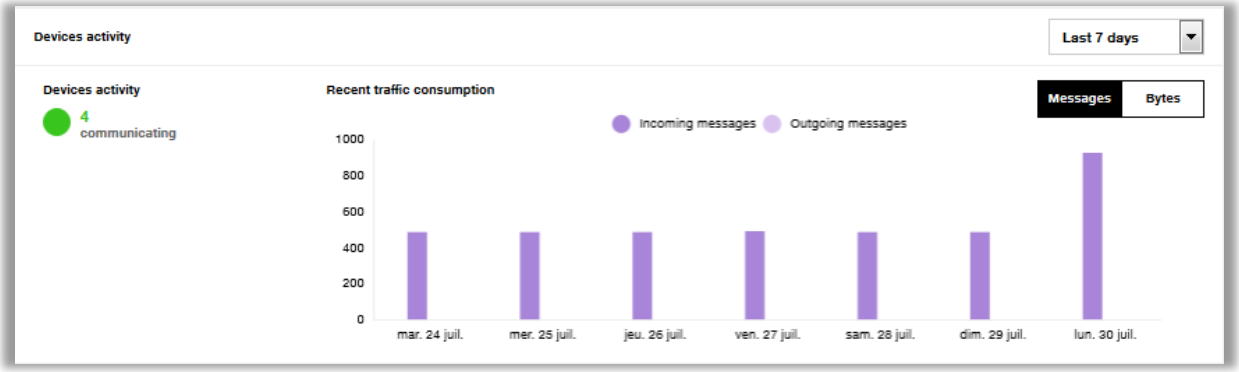


These quotas depend on the offer that has been subscribed.

- Message queues : Number of queues created / Max number of queues allowed
- API keys: Number of key created / Max number of key allowed
- Users: Number of users created / Max number of users allowed
-

To manage queues, keys and users go to “configuration” menu.

2.3.3. Fleet activity



This view allows selecting for a given period (last 7 days, last 4 weeks, or las 12 months) the global activity of the Live Objects account:

- number of devices that have communicated over the period
- traffic in number of messages (in/out) or in volume (bytes)

« incoming messages» are any messages entering into the platform. Examples:

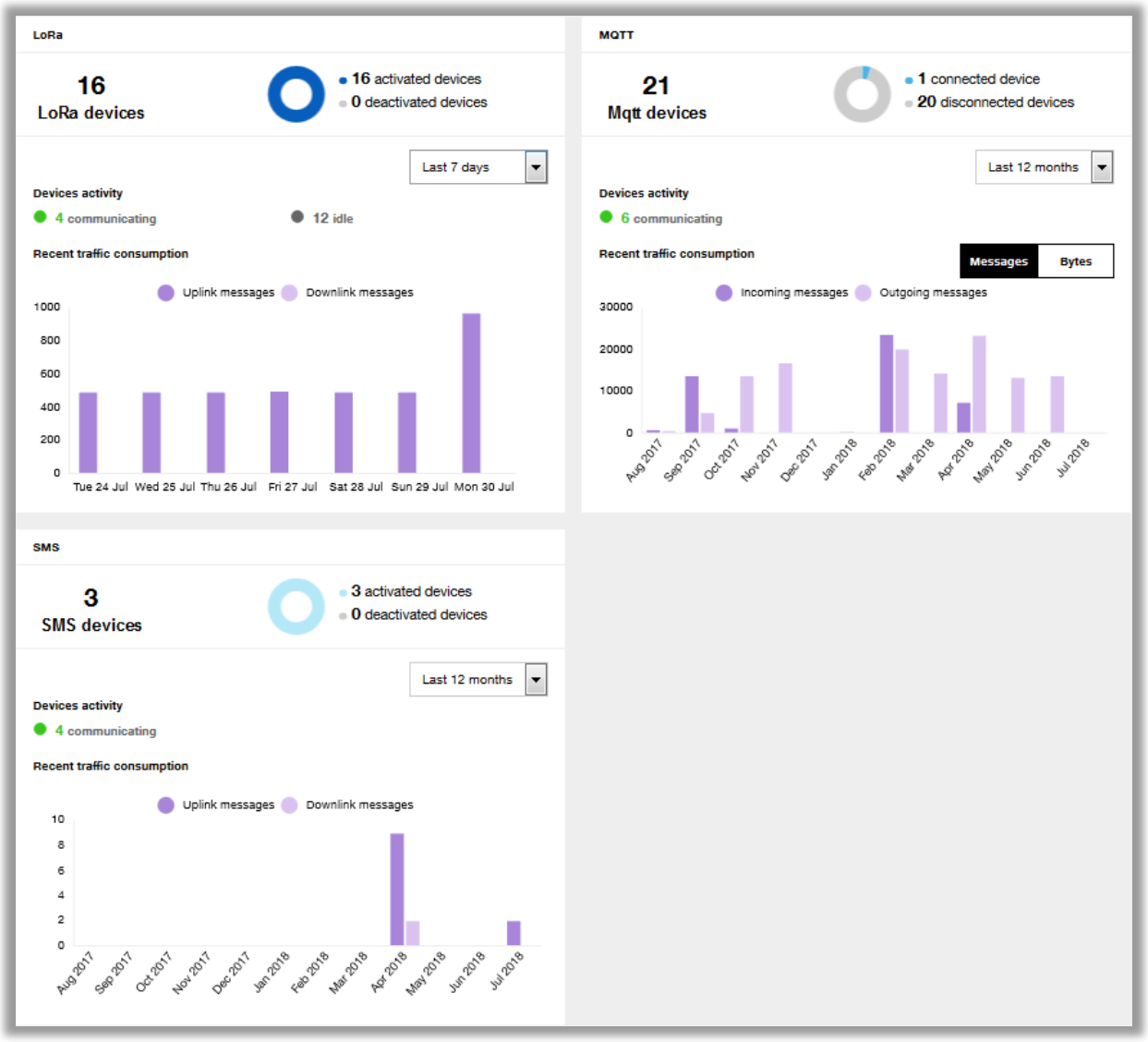
- uplink messages from device to the platform
- a business application on top of Live Objects which send REST API requests (POST, PUT, PATCH) to the platform (ex: search request, send command …)

« outgoing messages» are any messages issued by the platform to the outside. Examples:

- A downlink command from the platform to a device
- Answers to REST API call (GET)
- Consumption of messages by MQTT

2.3.4. Detailed activity

A synthesis of the activity is also available per connectivity



For each connectivity:

- The number of devices **currently** managed (16 LoRa, et 21 MQTT and 3 SMS in the example above)
- Statistics over the **selected period**. Note that on this period, the number of devices could vary from the current state of the fleet, depending on additions and deletions that may have occurred in the meantime. It's the case in the example above: During the last 12 months we have more communicating SMS devices than we have currently in fleet.

Some informations are specific to connectivity.

For LoRa, are displayed:

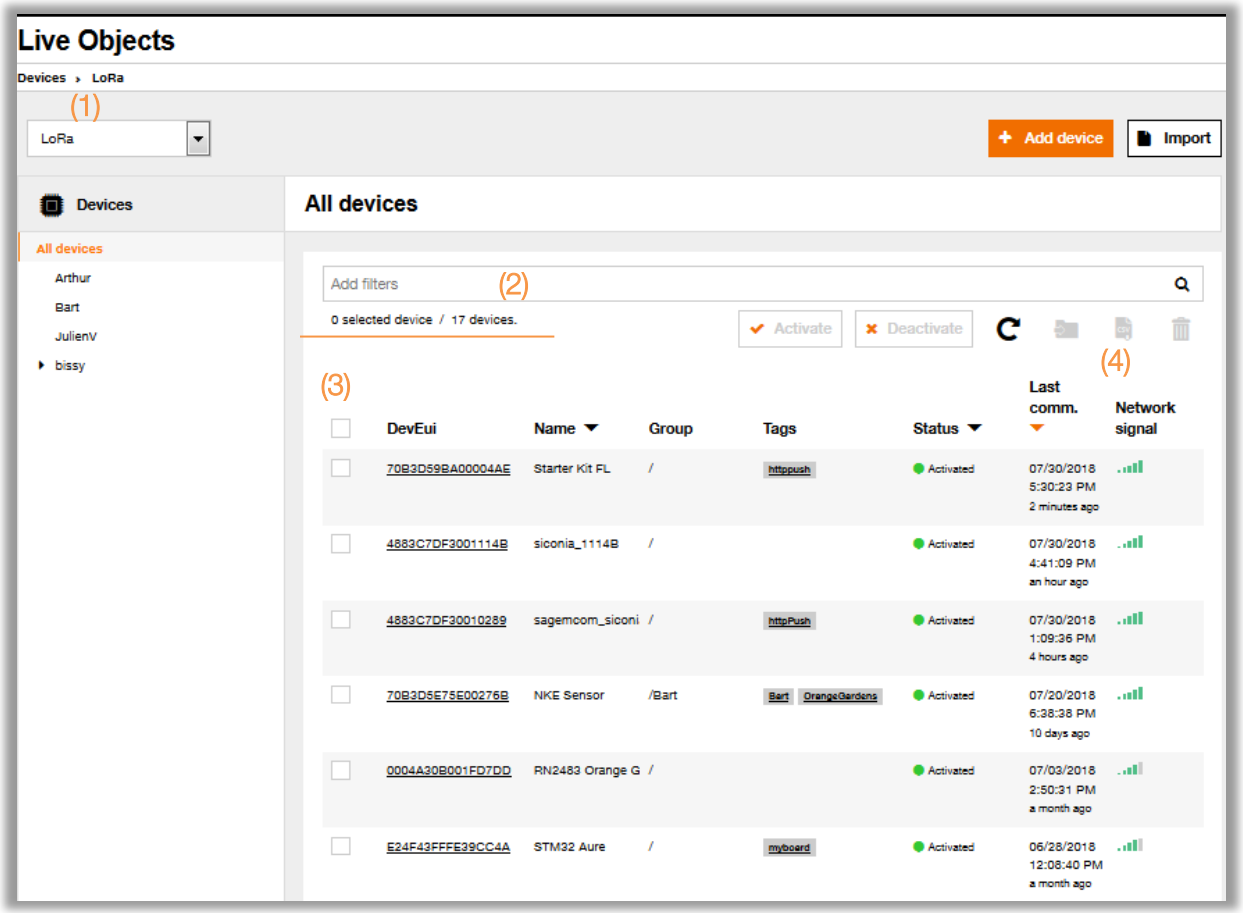
- An « activated » device means that the device has been registered into the network and is ready to communicate from the network point of view. This doesn't mean that the device has already communicated nor he "JOIN" successfully. For example a device created in Live Objects with bad parameters (devEUI, appEUI, AppKey), will be activated in the network with these bad parameters and then could not JOIN
- A « deactivated » device means that any communication on the network has been prohibited for this device. It's a voluntary action which has been made by an administrator of the fleet (See management of Lora Devices)
- Uplink messages from the devices to the platform
- Downlink messages sent from the platform to the devices (commands)

for MQTT :

- Number of connected devices. It's all the devices which have a MQTT connection currently active
- The number of devices that have communicated over the given period
- Number of incoming and outgoing messages. Note : this is the number of all MQTT messages, exchanged between the platform and devices or business applications on top of Live Objects. Between Live Objects and devices are counted data messages and device management messages; Between Live Objects and business applications real time data messages consumption.

2.4. Device management

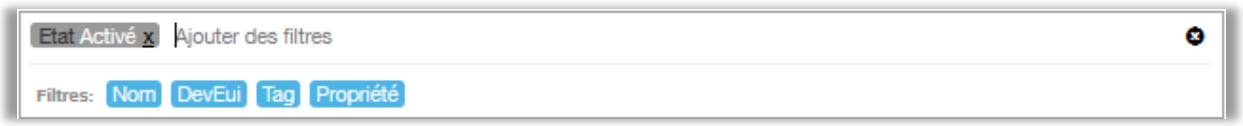
You can manage your fleet of devices in the menu « Devices ».



2.4.1. Display the fleet per connectivity

Because devices have characteristics proper to their connectivity, you have to first select the connectivity in the dropdown list (1)

You can filter the list of devices with several criteria. Click on the field « Add filters » to choose them (2)

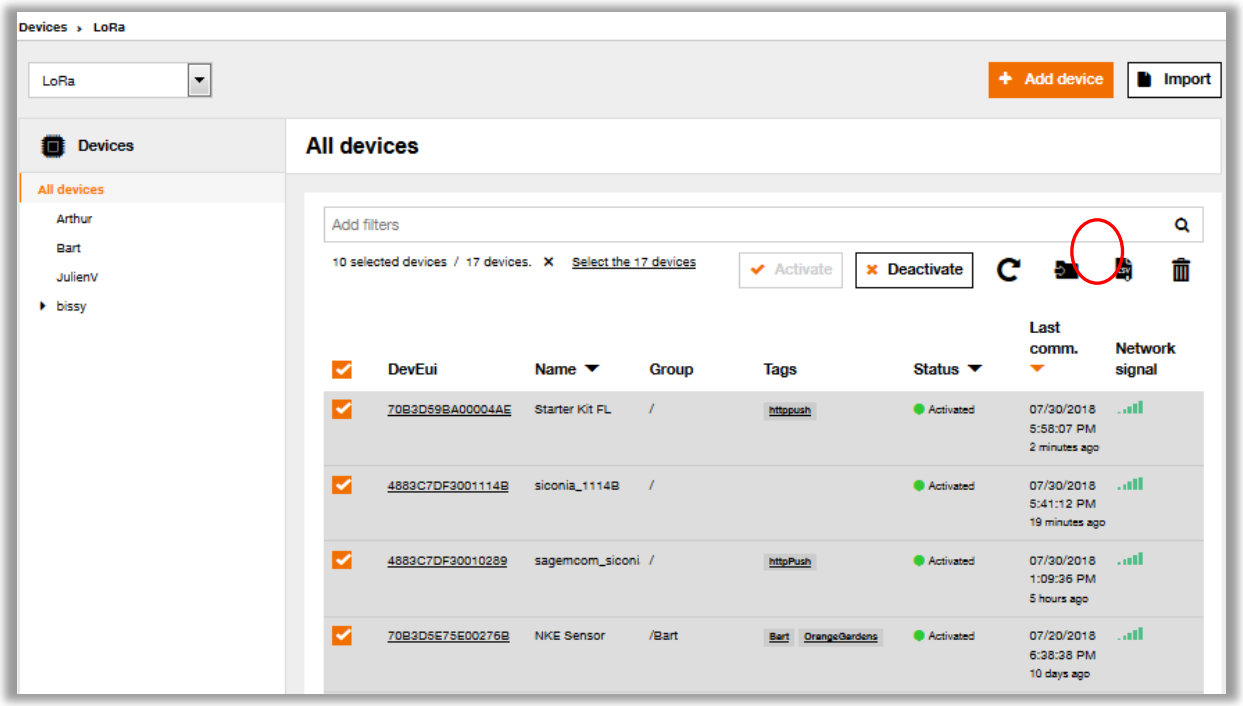


You can select devices unitary, per page or totally, by using the checkbox. The number of selected devices / total number of devices is recalled in the top of the list (3)

You can make actions on the selection: export to a csv file, move from group, deletion (4)

2.4.2. Export a list of devices

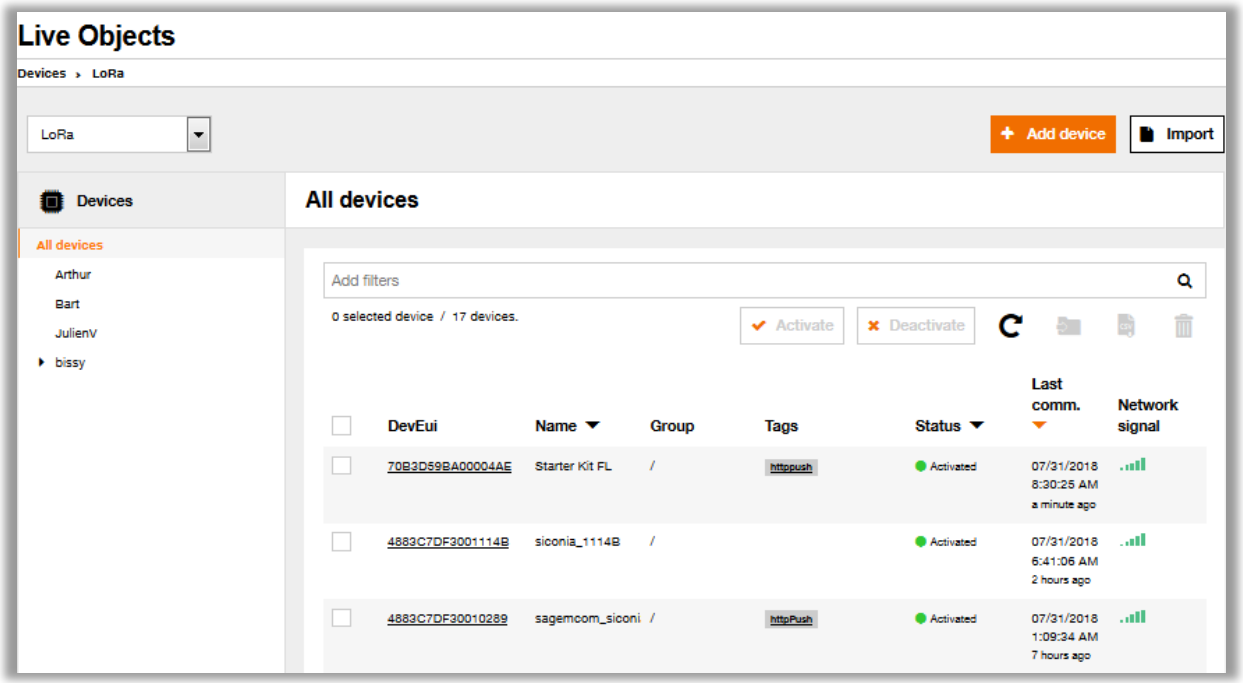
You can export the list of your devices in a csv file. For that go to the menu « Devices » then select the type of equipment (LoRa, MQTT or SMS). Then select the devices to export (all or a subset) and click on the icon « export »



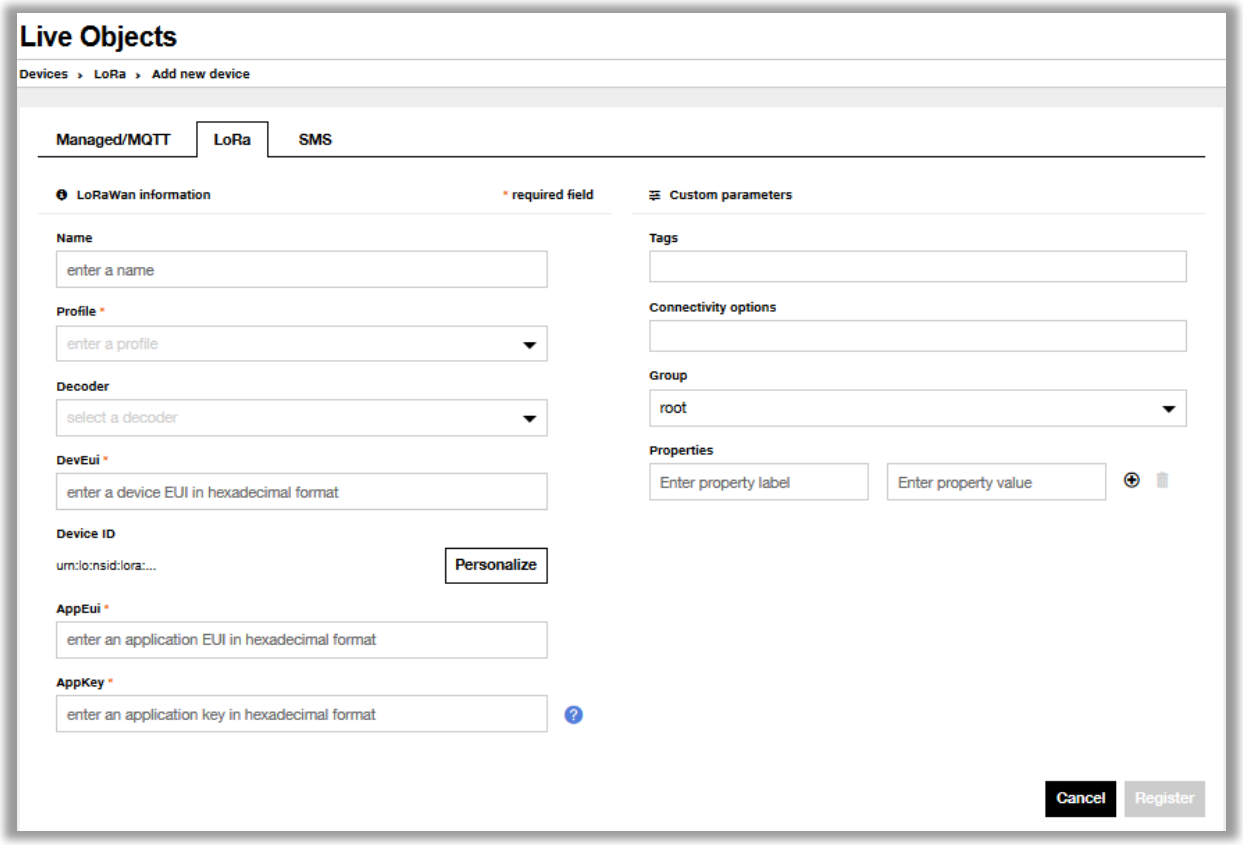
2.4.3. LoRa devices

2.4.3.1. Add a LoRa device

To add a LoRa device, click on the button “Add device”



The creation form opens with the parameters specific to a LoRa device



Name : Name of the device

Profile : The « Device profile » contains some predefined network parameters as the « spreading factor » that the device will use for JOIN, Class and RX frame. Select in the list the profile corresponding to your device. If you don't find the model corresponding to your device choose the generic profile Generic_classA_RX2SF12 (if your device is a class A) or Generic_classC_RX2SF12 (if your device is a class C)

Decoder : Decoders allow translating a payload from hexadecimal into data understandable by humans and indexable. Select in the list the decoder corresponding to your device. If there is no one, you can create your own in the menu "Configuration". See "manage decoders" paragraph.

DevEUI : The unique identifier of the device in the LoRa network. It is provided by the device vendor.

Device ID: The unique device identifier in Live Objects (URN). By default, it is built in the namespace « lora » with the devEUI as device identifier. You can custom the device ID with your own namespace instead of « lora » and with a device identifier different than devEUI

AppEUI, AppKey : These parameters belong to the owner of the device or the application used with it (See LoraWan specification)

Tags: Tags are free labels that you can attribute to your devices. You can use tags to ease sorting or identification in the fleet

Connectivity options: This field is displayed only if connectivity options have been activated on your Live Objects account. See paragraph « Add a connectivity option»

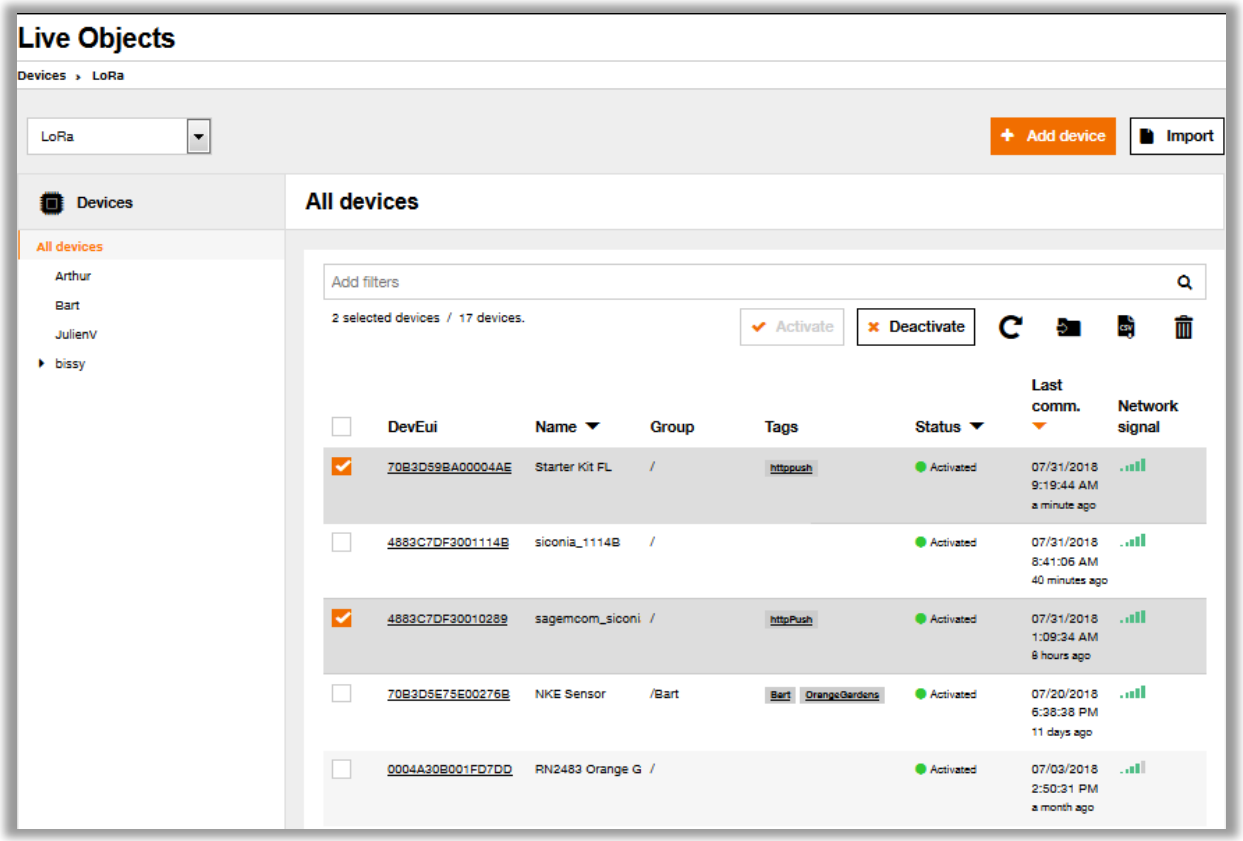
Group: Select here the group you want that your device belongs to. See paragraph « Groups »

Properties : You can define some properties for your devices. Properties allow you to enrich you device management.

Note : Orange recommends « Over the Air » activation (OTA) on its network. Only this activation mode is proposed in the Live Objects portal. If you need « activation by personalization » (ABP) please contact you customer support.

2.4.3.2. Deactivate and reactivate a LoRa device

It is possible to prevent one or more LoRa devices from communicating on the network temporarily without having to delete them. To do this, select the devices concerned by ticking the corresponding box in the list, and click on the "Disable" button.



To reactivate one or more devices, proceed in the same way by clicking on the "Activate" button.

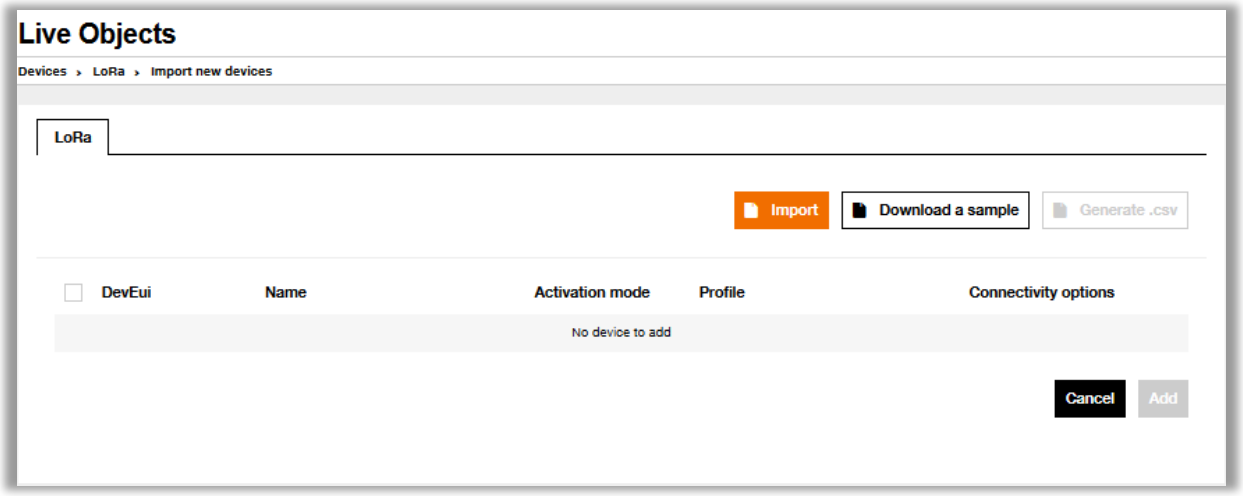
2.4.3.3. Delete a LoRa device

It is possible to delete LoRa device from the platform. This action will also remove it from the Orange LoRa network. The data that have been issued by the device are not deleted and remain accessible in the menu “Data”.

To delete one or more devices, select them by ticking the corresponding box in the list, and click on the "Delete" icon.

2.4.3.4. Mass import of LoRa devices

To bulk import LoRa devices, click on the "import" button available in the LoRa devices list view.



The import file must be in CSV format. An example is available by clicking on the "Download an example" button.

	A	B	C	D	E	F	G	H	I	J
1	groupPath	name	lora_profile	lora_activationType	lora_devEUI	lora_appEUI	lora_appKey	tags	lora_encoding	lora_connectivityOptions
2	/	Device01	Generic_classA_RX2SF9	OTAA	1234567800000000	1234567800000000	################	Paris		TDOA,ACKUL
3	/france/lyon	Device02	Generic_classA_RX2SF9	OTAA	1234567800000000	1234567800000000	################	Neuf,Lyon		TDOA,ACKUL

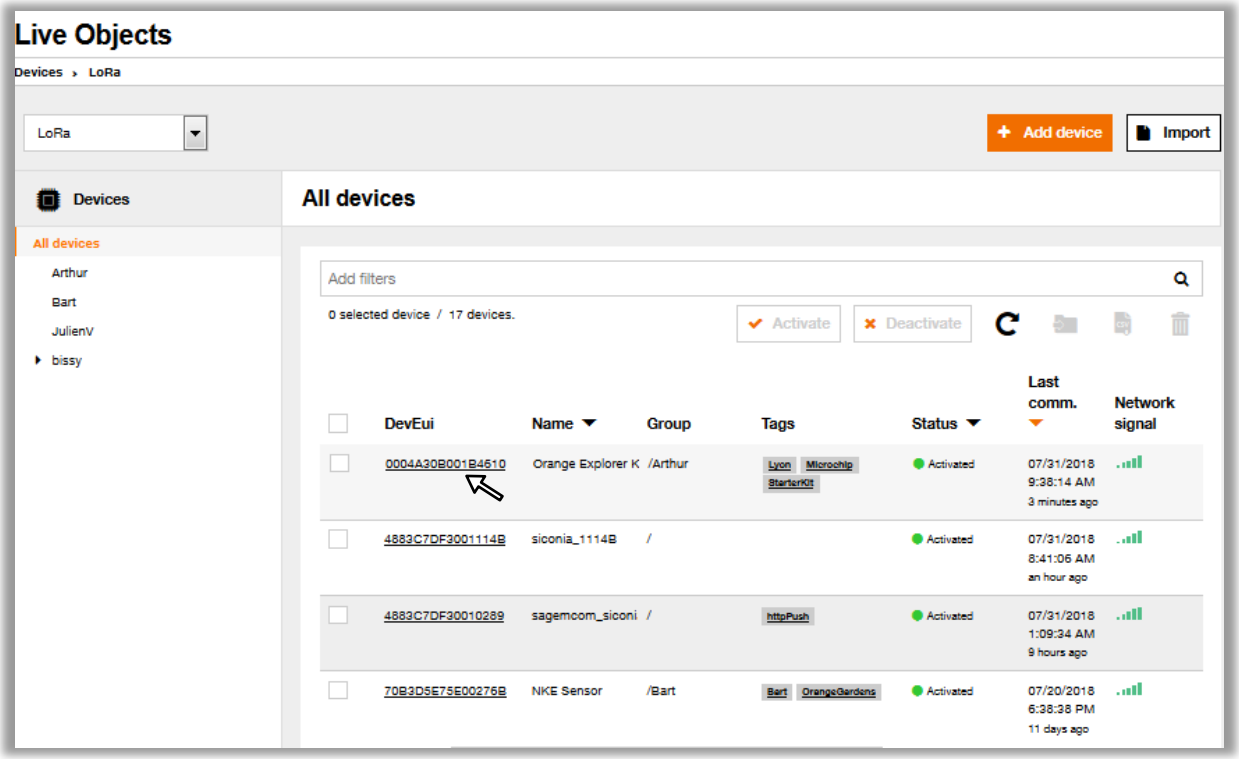
The fields to be filled in are the same as those to be provided during the unitary creation.

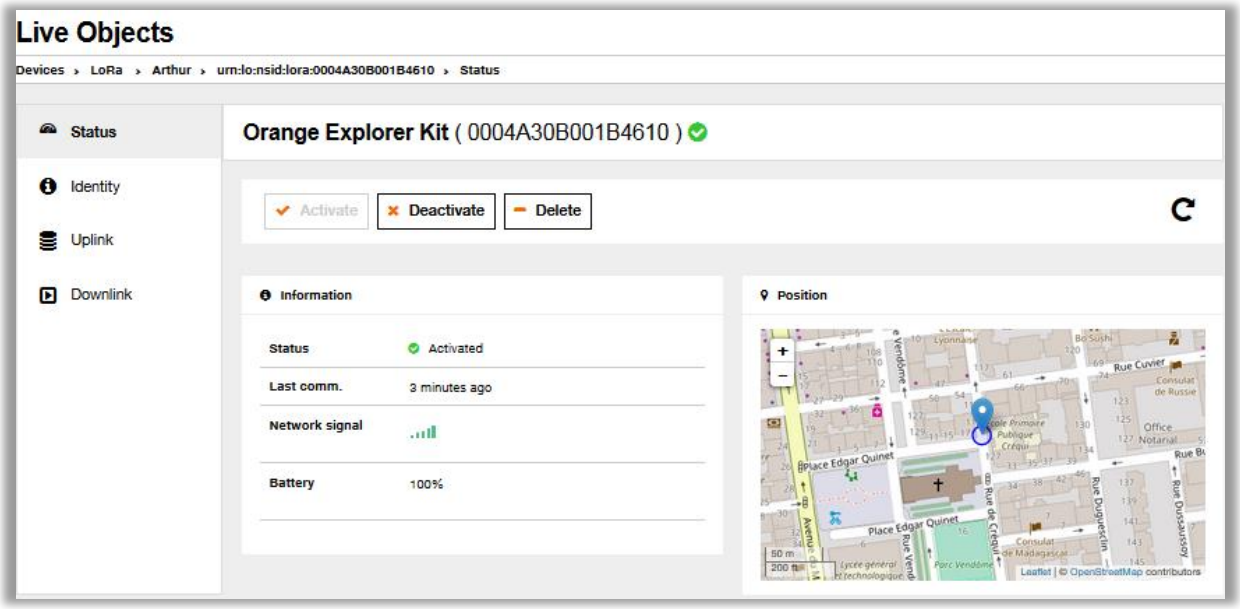
Once your import file is ready you can start mass importing by clicking the "Import" button

A first verification of the file is then performed. This verification concerns the validity of the syntax, the presence of the required fields, as well as the structure of the file. The result of the analysis is displayed in the import screen. Correct any lines in error by re-importing the file if necessary and click on the "Add" button to proceed with the actual creation.

2.4.3.5. Details of a LoRa device

To access detailed information about a LoRa device, click on the desired device in the main list of the "Park" menu.



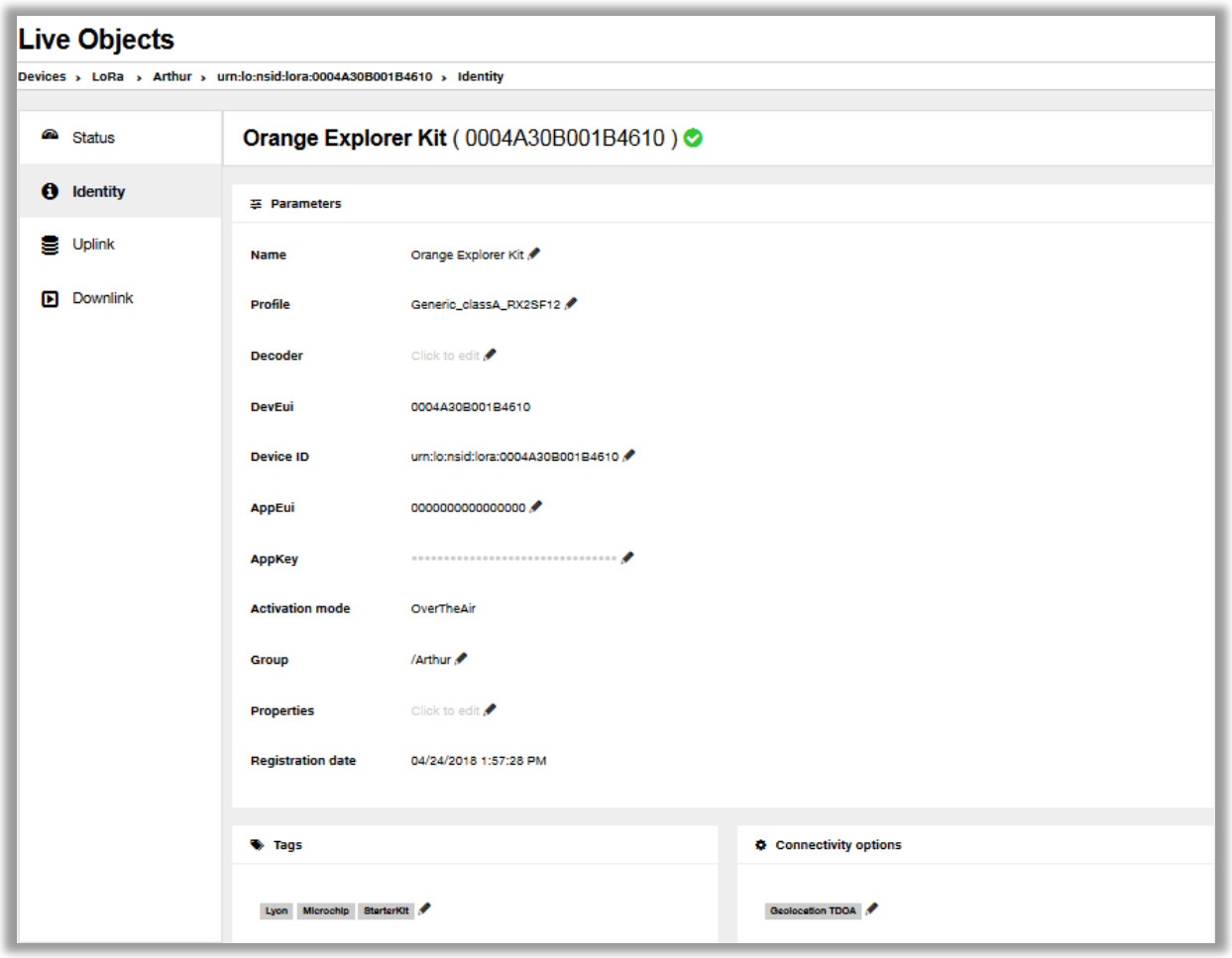


The general **status** of the device is displayed:

- its state: activated or deactivated (see definitions in the paragraph « deactivate and reactivate a LoRa device »)
- its date of last communication (based on the timestamp field applied by the LoRa gateway that has forwarded the message)
- the network signal: it is a quality indicator of the connectivity calculated by Live Objects based on various network parameters reported by the device (RSSI, SNR, PER ...)
- the battery level: It is based on the value transmitted to the network by the device (LoraWan standard)
- the geolocation of the device on a map with a circle showing the accuracy. The detail is available by clicking on:
  - date of the last location
  - GPS coordinates

2.4.3.6. Modify a LoRa device

To modify a LoRa device go to the details of the device and click on the “identity”.



Informations entered during the creation of the device are displayed. It is possible to modify the name, the profile, the decoder, the tags as well as the keys appEUI and appKey

Changing the devEUI is not possible.

Note:

- The appKey is never displayed in clear
- The name, and the appEUI, appKey keys are not editable if the device is disabled.
- The connectivity options must have been activated in your Live Objects account by your customer service to be selectable. See next paragraph.

2.4.3.7. Add a connectivity option

The connectivity options available are:

- TDOA geolocation: Under coverage conditions* this type of geolocation allows a much greater accuracy than a standard geolocation
- "Ack Uplink": Allows the device to request an acknowledgment from the network for each uplink issued

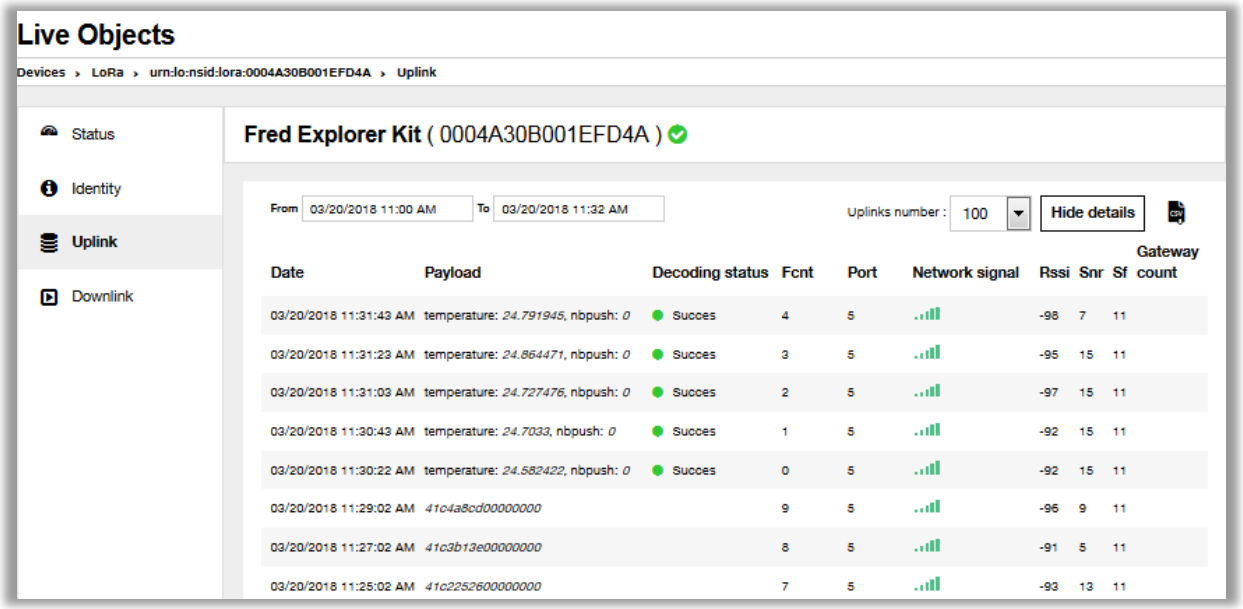
(Contact your customer service)*

To get these options:

1) contact your customer service to activate them in your Live Objects account
2) then you'll be able to add these options to your devices (See paragraph « Modify a LoRa device »)

2.4.3.8. See data issued by a LoRa device (uplink)

To view the data reported by LoRa device go to the device details and click on "Uplink"



The chronological list of messages sent by the device is displayed along with the associated network information:

- The "Payload" column represents the application content sent by the device (no network message) in hexadecimal format. This content can be decoded by associating an appropriate decoder to the device.
- The "Decoding Status" column indicates whether a decoding was requested and whether it was correctly performed or not
- The "Gateway Number" column indicates the number of gateways that relayed the message (known also as "macro-diversity")

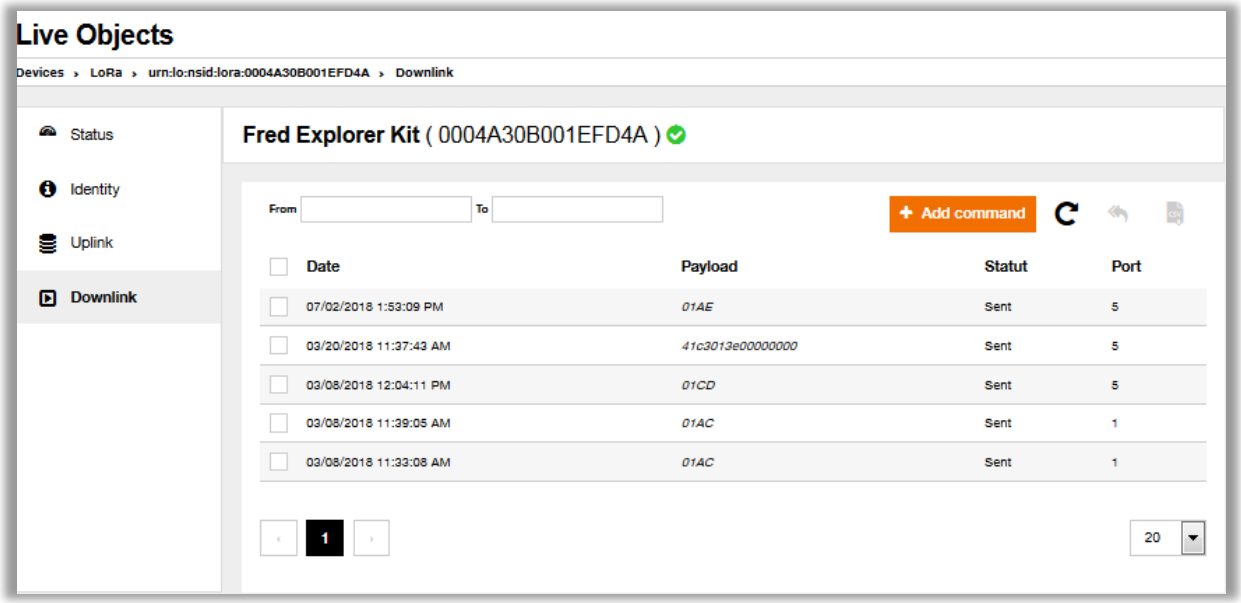
You can use the filter to search collected messages over a period of time.

A selector is set to 100 by defaults to limit the number of uplink to display in a page. The most recent 1000 can be displayed at most

This list can also be exported in a CSV file.

2.4.3.9. Send a command to a LoRa device (downlink)

To send a command to a LoRa device go to the device details and click on the "Downlink".



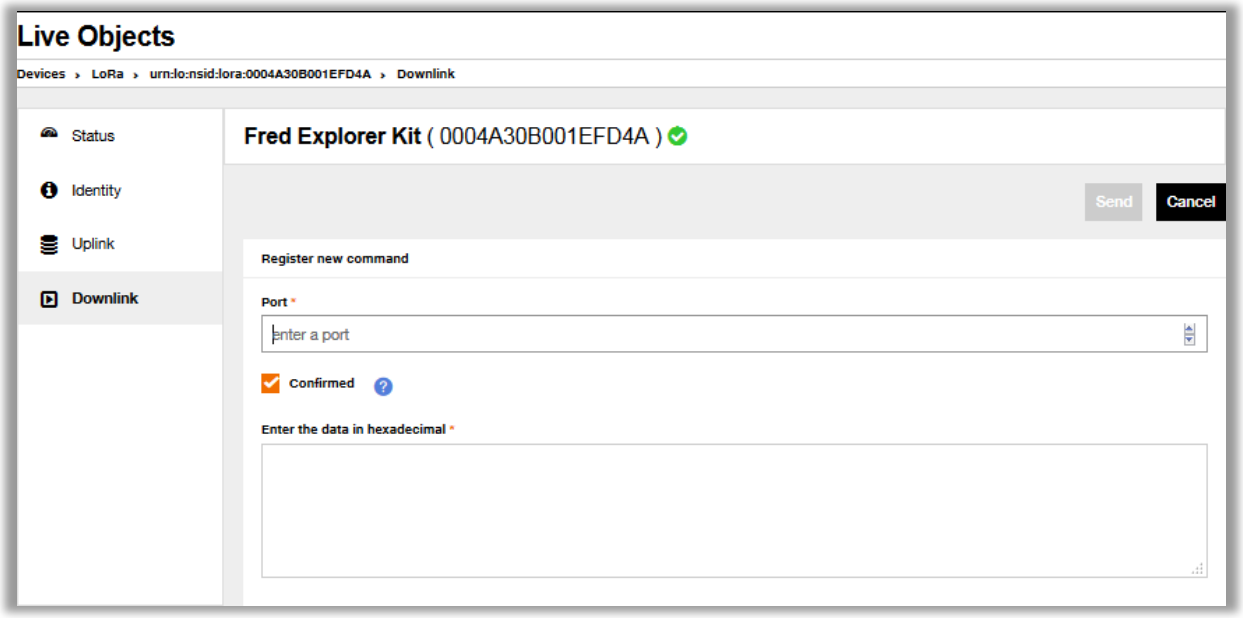
The history of commands sent to the device is displayed in a chronological order. The "status" column displays the status of the command and can have the following values:

- Sent: means that the command has been taken over by the network, which will take care of routing it to the device by implementing, if necessary retry mechanisms. In the current version this information does not constitute an acknowledgment of receipt by the device
- Error: the command was not accepted by the network. It will not be delivered to the device

The history of commands is exportable in a csv file.

A command can be sent again without having to reenter it, for that, you just have to select it in the list and to click on the icon "Repeat"

To send a new order, click the "Add Order" button



- select the port used by the equipment to communicate (it must be greater than 0). Sending MAC commands (port 0) is not possible via Live Objects.
- if the command must be acknowledged by the device (check the "Confirmed" box)
- the data of the command in hexadecimal format (the validity of the format is checked)

2.4.4. MQTT devices

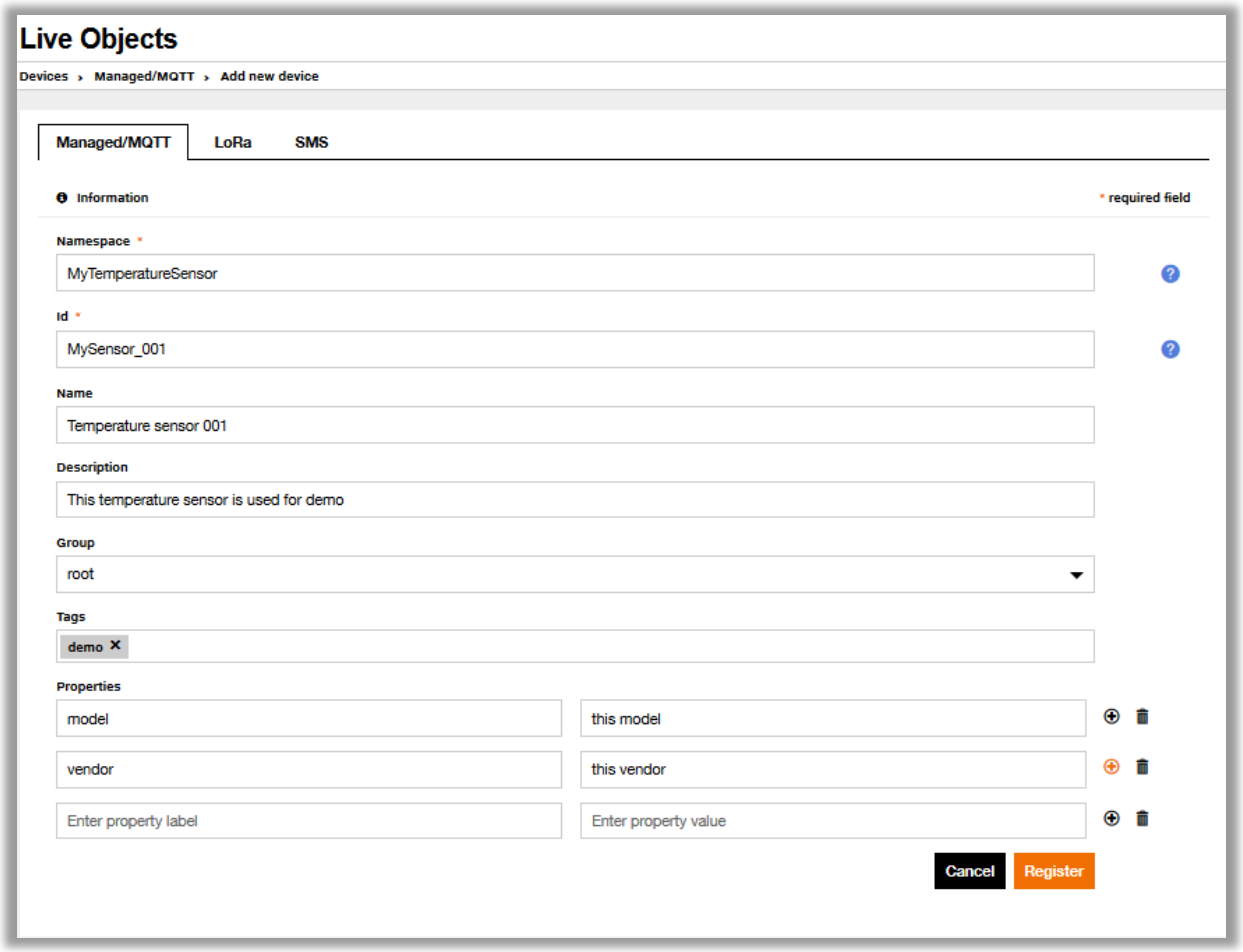
2.4.4.1. Add an MQTT device

Adding an MQTT device to Live Objects is automatic. When connecting to Live Objects for the first time, the device registers itself and uploads its own informations (firmware version, parameters). For this the equipment must have been properly configured. (See the developer guide https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html#MQTT_MODE_DEVICE)

A number of attributes can be added to a device to facilitate its management in Live Objects (groups, properties, tags). These attributes are not stored at the device level but remain at the platform level.

It is also possible to add an MQTT device to Live Objects before it first connects to the platform. This allows for example to prepare a fleet by setting in advance the attributes above.

To add an equipment MQTT click on the button "Add an equipment"



Namespace : The namespace provides a consistent naming for a set of devices. For example all devices of the same manufacturer or same type

Id : The Id identifies the device inside the namespace. The couple (namespace, id) constitutes the unique identifier of the device in Live Objects

Name : Allow to give a familiar name to the device

Description : A comment field to enter a short description of the device

Group : Select a group in which to add the device. By default the device is added to the root group. (See paragraph « Groups »)

Tags : Free labels that can be used to ease sorting and identification of devices in the fleet

Properties: Free key-value couples that can be added to devices to enrich device management

2.4.4.2. Delete an MQTT device

It is possible to delete an MQTT device from the platform. This action will remove it from the Live Objects device manager (history, properties, groups etc.). Data issued by the device are not deleted and remain accessible in the "Data" menu.

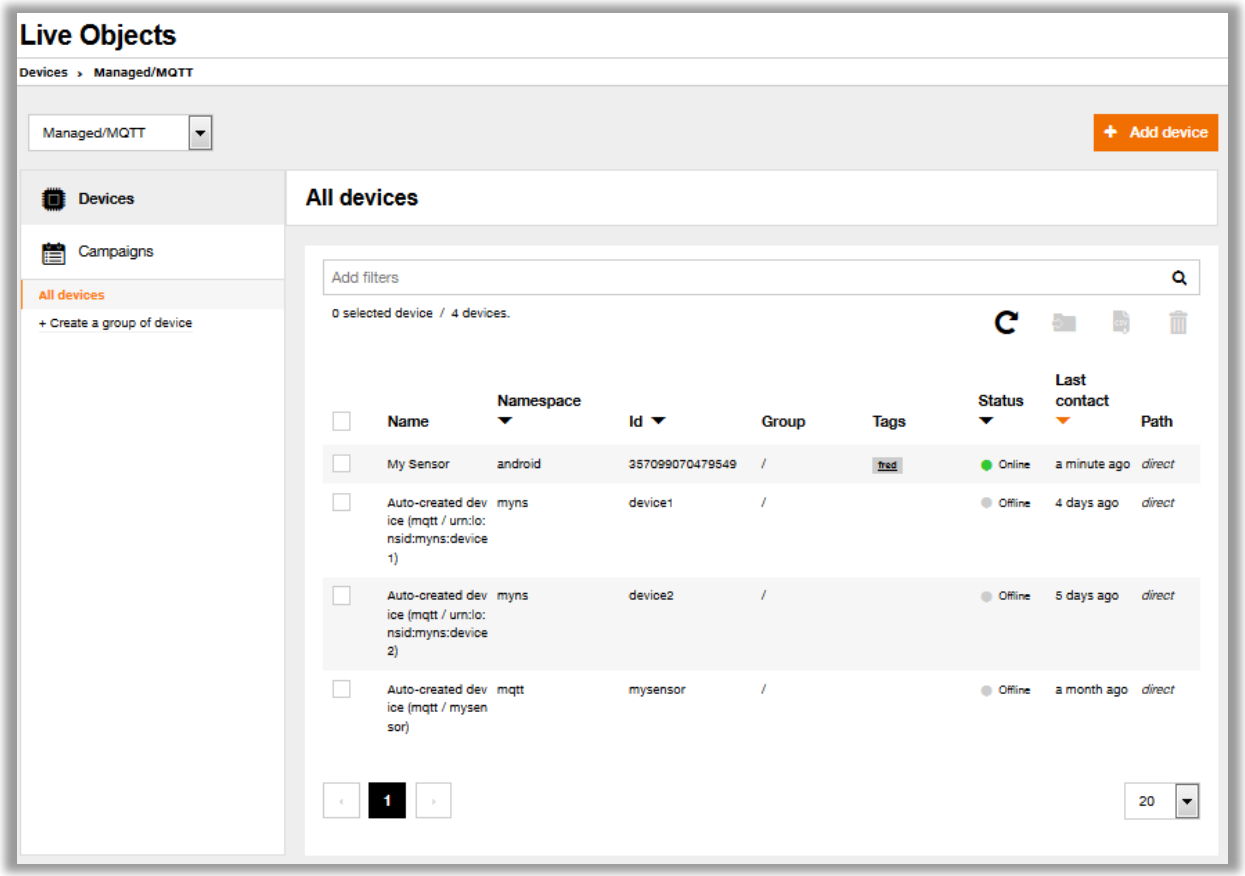
To delete one or more devices, select the devices concerned by ticking the corresponding box in the list, and click on the "Delete" icon.

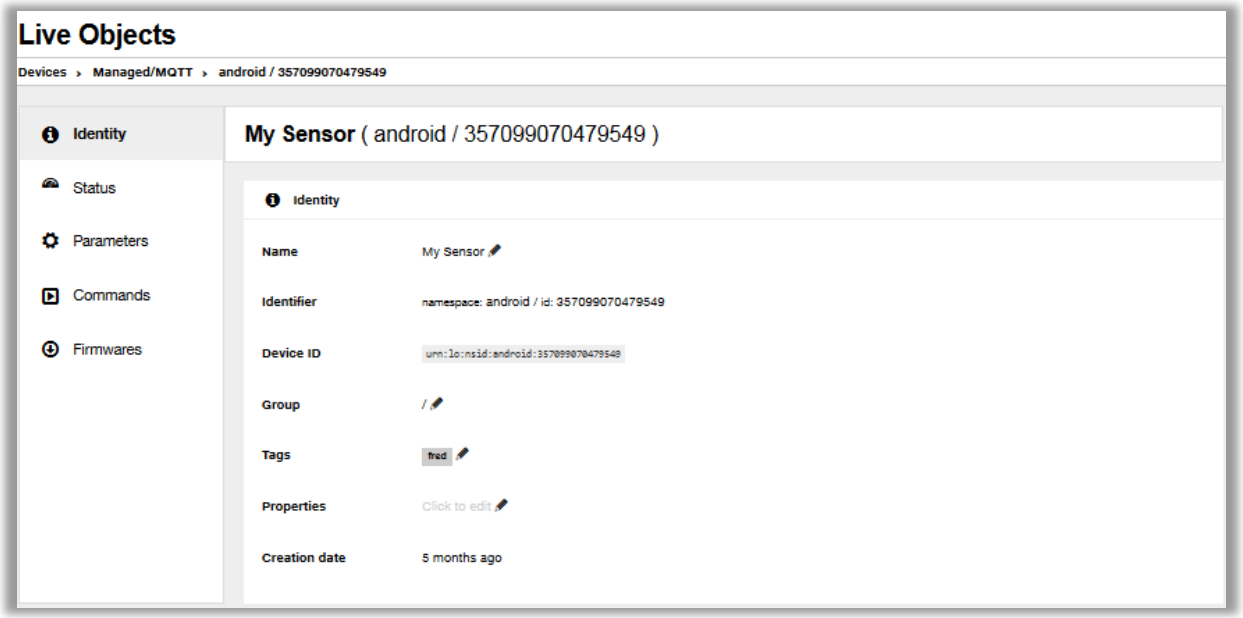
2.4.4.3. Mass import of MQTT devices

Bulk import of MQTT devices is not yet available. It is planned in coming versions.

2.4.4.4. Details of an MQTT device

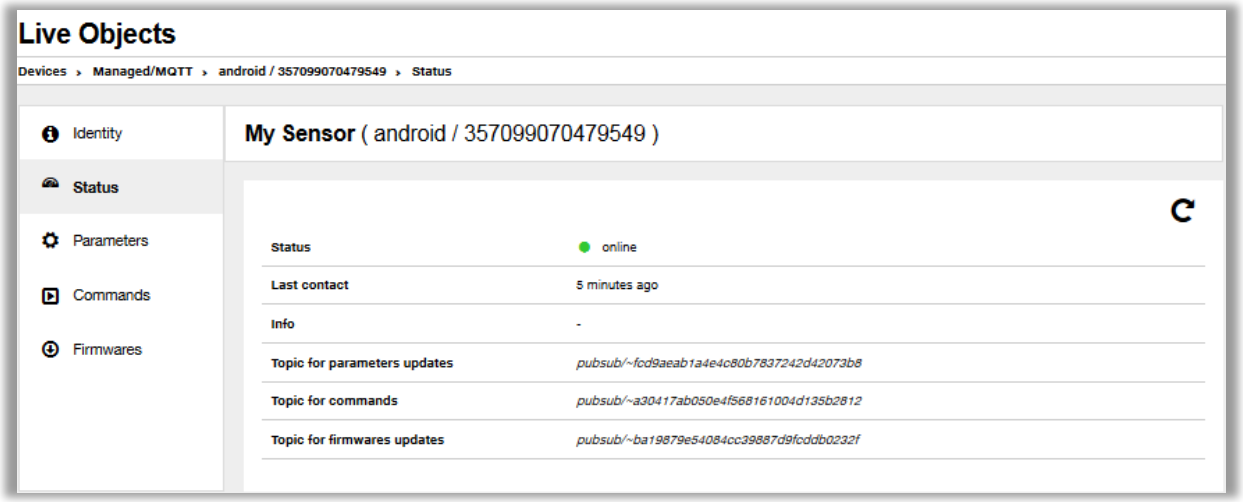
To access detailed informations about an MQTT device, select it in the list.





The menu **Identity** displays the informations entered during the creation of the device in Live Objects, as well as the date of its creation. It allows you to modify these informations with the exception of the identifier / URN (Unique Resource Name) which is the unique identifier used by the device to connect to MQTT.

The menu **Status** gives connections details



It indicates whether the device is connected or not, and when it was last connected. The field "info" is no more used and will be removed.

If the device is available for a parameter update, a command send, or a firmware update, then the corresponding topics are displayed. Note that these are internal topics auto generated when the device subscribes to the following topics:

- `dev/cfg/upd` (topic to receive parameter update requests)
- `dev/cmd` (topic to receive commands)
- `dev/rsc/upd` (topic to receive firmware update requests)

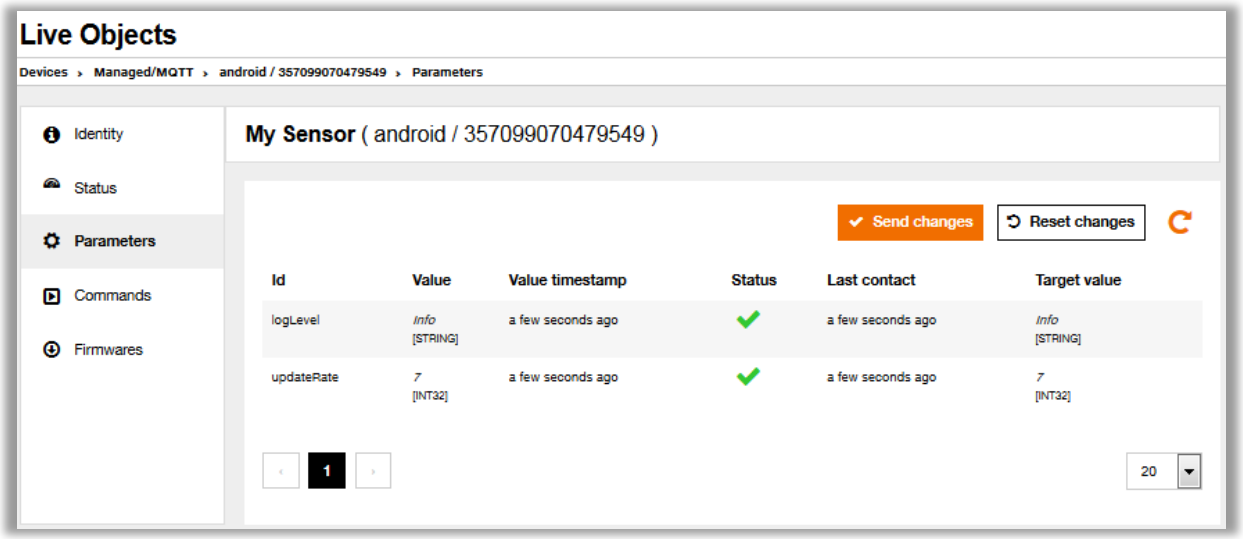
Internal topics are not to use.

See the developer guide for more details on how MQTT devices work
: https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html#MQTT_MODE_DEVICE

2.4.4.5. Update parameters of an MQTT device

A device can have specific parameters as factory parameters, or applications embedded directly in its micro code. These settings can be managed in Live Objects if the device support it. The device must have previously informed Live Objects about these parameters (publish to dev/cfg. See developer guide for details)

The visualization and the update of the parameters is possible in the menu "Parameters"

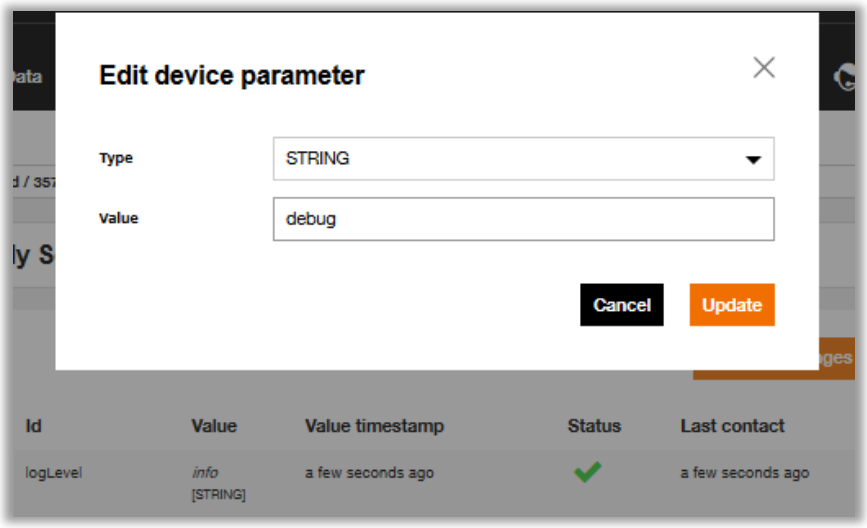


Informations displayed are :

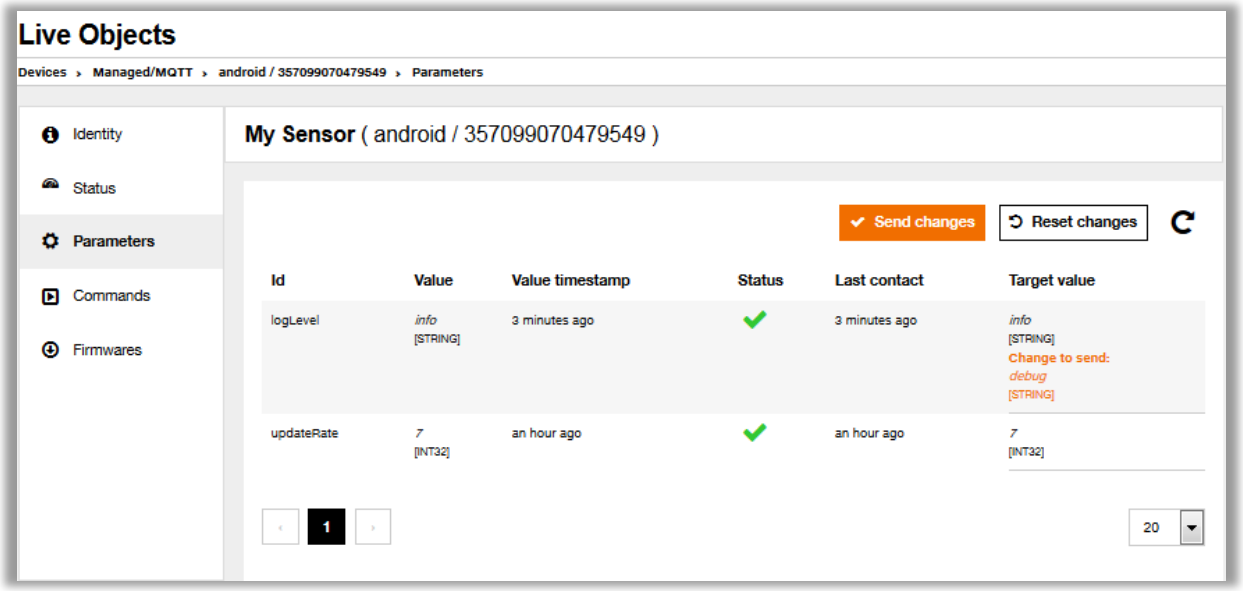
- the name of the parameter (Id)
- its current value and its type
- its last update
- its status :
    - « new » means that the parameter has been reported by the device for the first time and has never been synchronized with Live Objects
    - « pending» means that an update request is pending to be delivered to the device, and will be delivered when the device will be ready to receive it (by subscribing to topic dev/cfg/upd)

- « sent» means that the update request has been received by the device (subscribe to dev/cfg/upd), but the device didn't confirmed yet the update (publish to dev/cfg with the new value of parameter). See developer guide for details
- « sync » means that the parameter has been successfully updated by Live Objects. The value of the parameter announced by the device on topic dev/cfg is the same as the target set in Live Objects. The device and Live Objects have the same value of parameter.
- "failed" means that the update request has been received by the device, but the device has answered to Live Objects with a different value of the parameters than those set as the target. See developer guide for details

To send an update request, click on a parameter and set its target value



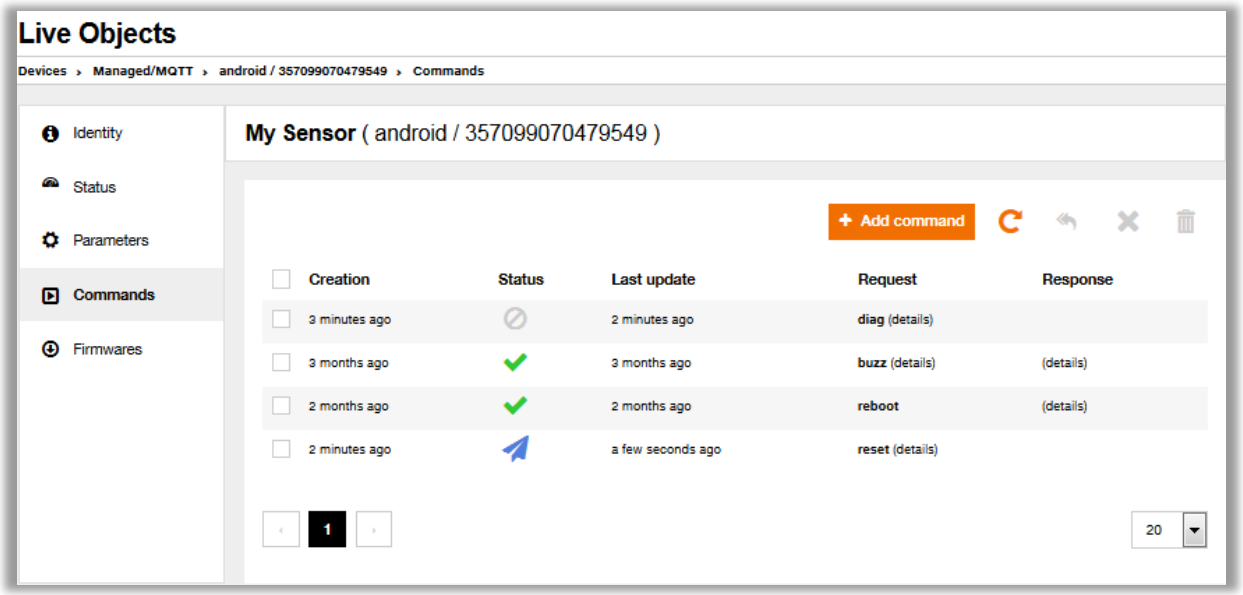
Click on button « Update » to validate the request, and repeat the operation for each parameter you want to change.



Changes to apply are listed in column « target value » with mention “change to send”. At this moment the requests are not yet sent. To send it to the device, you must click on the button « Send changes »

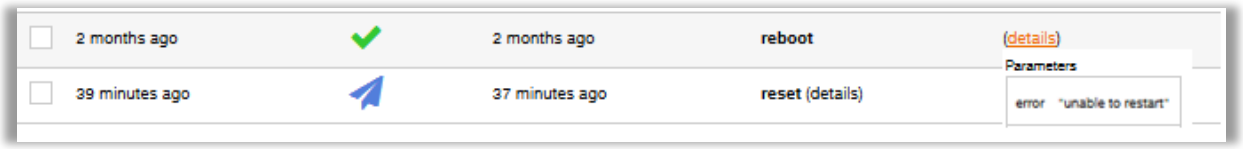
2.4.4.6. Send a command to an MQTT device

To send a command to a MQTT device, go to « Commands »

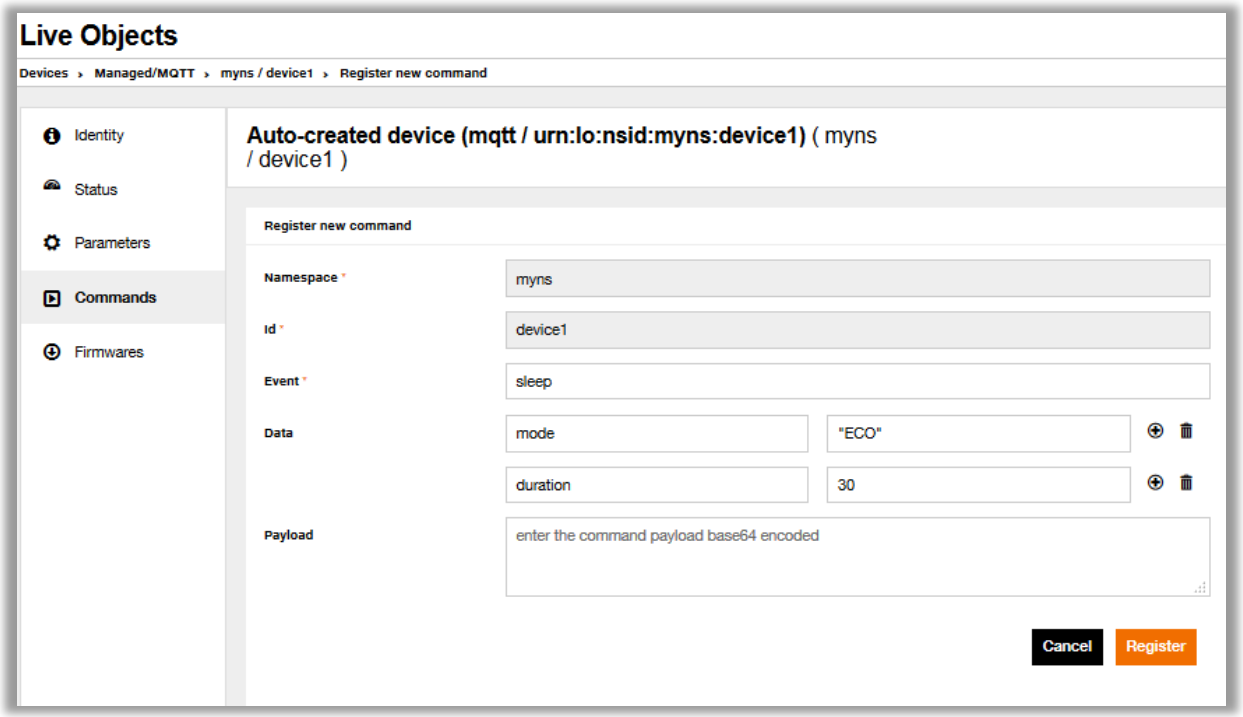


The history of commands is displayed. A command can have several statuses:

- “pending”: The command has been registered into Live Objects but is pending to be delivered to the device. For that the device must be connected and subscribe to topic dev/cmd
- “sent”: The command has been received by the device (subscription to dev/cmd), but the device didn’t acknowledged it yet (by publishing to dev/cmd/res)
- “Canceled”: The command has been canceled from Live Objects before having been received by the device
- “processed” The command has been processed by the device and acknowledged to Live Objects. Note: “Processed” doesn’t indicate the result of the processing of the command by the equipment. This result is an information that the device can communicate in its response:



To send a new command click on the button « Add command »



The name of the command and its possible values depend on the specifications of your device. Type its name into the field "Event"

The content of the command has to be typed in fields ”Data” as couples key-value. The values that are strings must be typed with quotation marks

Field « Payload » is for future usage. It is not used.

Click on button « Register » to send the command.

Note : as long as the device has not received the command (status pending), it is possible to cancel it, by clicking on the icon « Cancel »

This command can be sent back later, like any other command in the history by selecting it and clicking on the icon "Repeat".

Finally commands can be deleted. To do this select commands to delete and click on the icon «Delete.

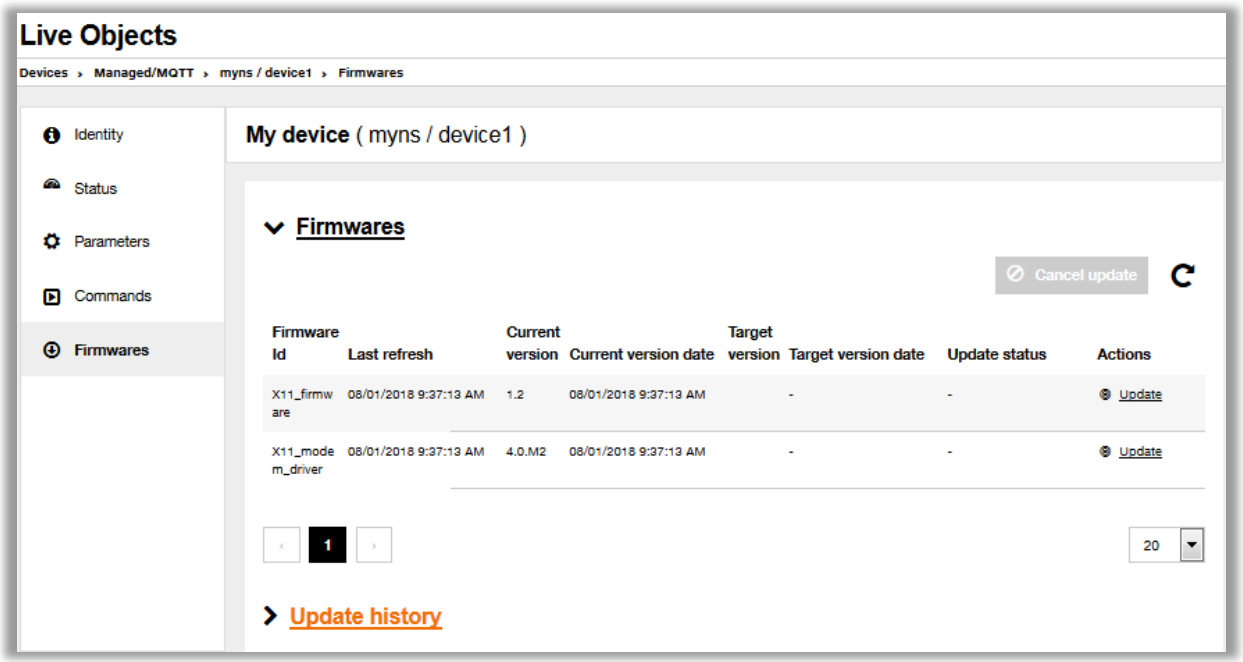
NB: Deleting a command that has been sent ("sent" status) does not send a cancellation order to the device. It removes the command from the history and from the Live Objects device manager.

2.4.4.7. Update the firmware of an MQTT device

To update the firmware of an MQTT device go to the device details and select "Firmware".

The update screen is divided into two parts. The first part presents the firmwares that the device has announced and if update operations are in progress. The second part is the history of updates that have been made.

Note: Live Objects can manage several firmwares for a same device, if this device has several internal components (modem, core chipset etc) with each their own firmwares.

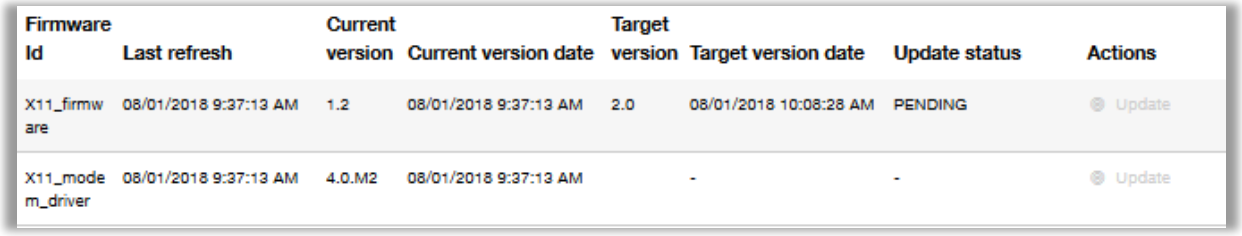


Prerequisites: To be able to update the firmware of your device,:

- The device must support this operation,
- This device has previously announced to the platform the name of its firmwares and its current versions (publish to topic dev/rsc). See developer guide.
- The firmware versions to manage have been previously created in the Live Objects Firmware Manager and the corresponding binaries imported into the platform. (See the "Configuration" menu for firmware management)

To start the update, select the firmware and click on the "Update" link. In the pop-up window, choose the target version and click on the "Save" button

The status of the request changes to "pending". This means that the request is waiting for reception by the device (the device must subscribe to topic dev/rsc/upd to receive the request). See developer guide https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html#MQTT_DEV_RSC_UPD



Firmware Id	Last refresh	Current version	Current version date	Target version	Target version date	Update status	Actions
X11_firmware	08/01/2018 9:37:13 AM	1.2	08/01/2018 9:37:13 AM	2.0	08/01/2018 10:08:28 AM	PENDING	Update
X11_modem_driver	08/01/2018 9:37:13 AM	4.0.M2	08/01/2018 9:37:13 AM		-	-	Update

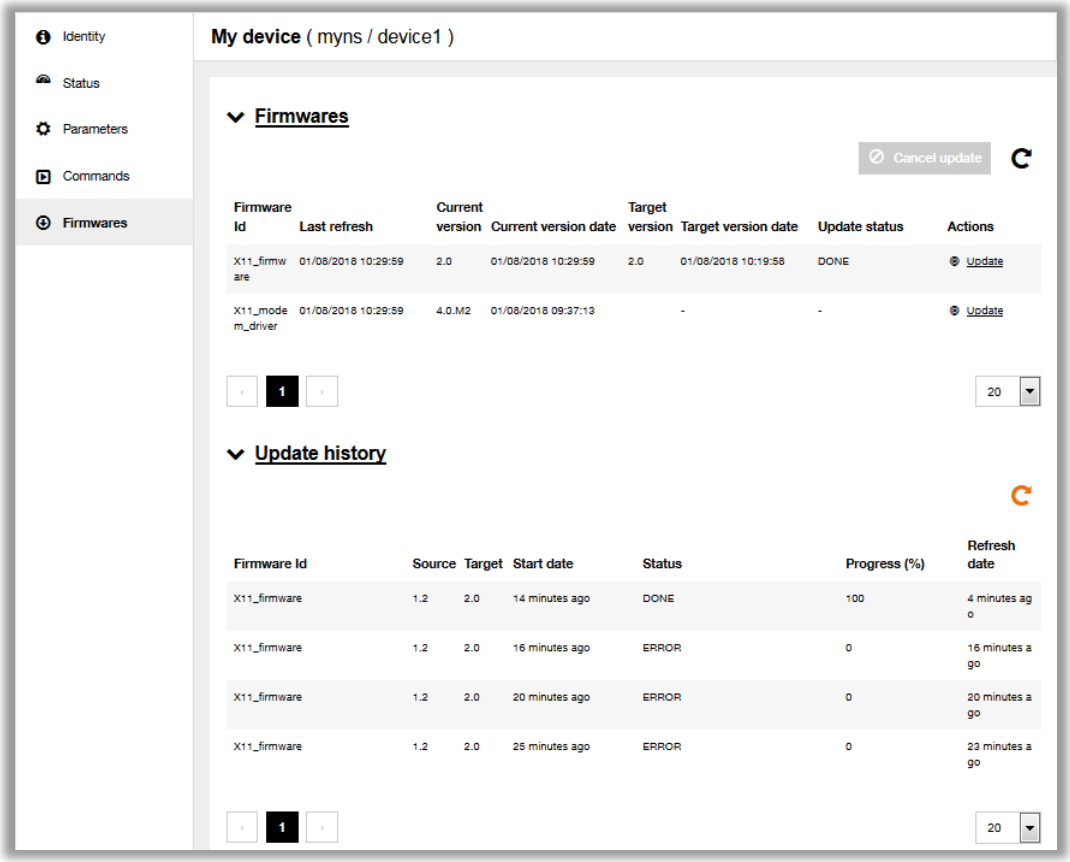
Once received by the device the request goes to "preparing_asset - 0%". This means that the device has received the request but has not yet responded if it accepts it. (To accept the request the device must answer by publishing on topic dev/rsc/upd/res)

Firmware Id	Last refresh	Current version	Current version date	Target version	Target version date	Update status	Actions
X11_firmware	01/08/2018 09:37:13	1.2	01/08/2018 09:37:13	2.0	01/08/2018 10:08:28	PREPARING_ASSET	Update
X11_modem_driver	01/08/2018 09:37:13	4.0.M2	01/08/2018 09:37:13		-	-	Update

If the device accepts the request, it turns to « waiting_transfert_info ». Once the device starts the download of the firmware, a percentage will show the progress of the transfer

Firmware Id	Last refresh	Current version	Current version date	Target version	Target version date	Update status	Actions
X11_firmware	08/01/2018 9:37:13 AM	1.2	08/01/2018 9:37:13 AM	2.0	08/01/2018 10:19:58 AM	WAITING_TRANSFER_INFO	Update
X11_modem_driver	08/01/2018 9:37:13 AM	4.0.M2	08/01/2018 9:37:13 AM		-	-	Update

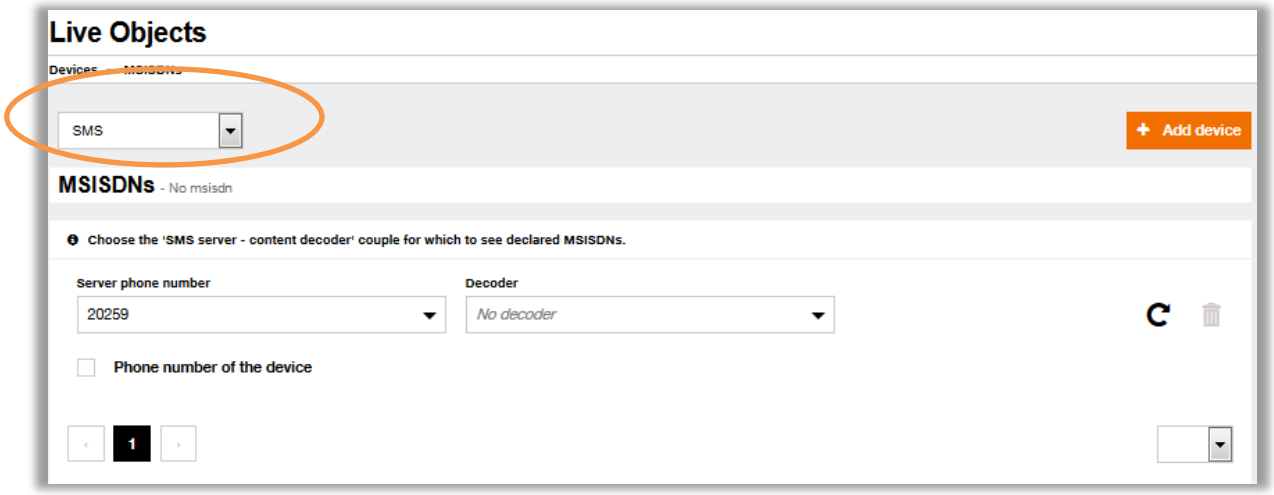
Once the transfer is made, and the device has installed the firmware, the device goes announces to Live Objects its new version (by publishing on topic dev/rsc). If the version announced is the same as the target value that has been set, the process ends successfully and the status changes to "done". The history is also updated.



2.4.5. SMS devices

You can connect to Live Objects devices that communicate with SMS. These devices must have Orange SIM cards.

To manage SMS devices go to menu "Device" and select "SMS" from the drop-down list.



The "server phone number" is a configuration assigned when creating your Live Objects account if your offer contains this option. It is the "short code" that will be used by your devices to communicate the Live Objects platform.

Several short-codes may be configured if needed.

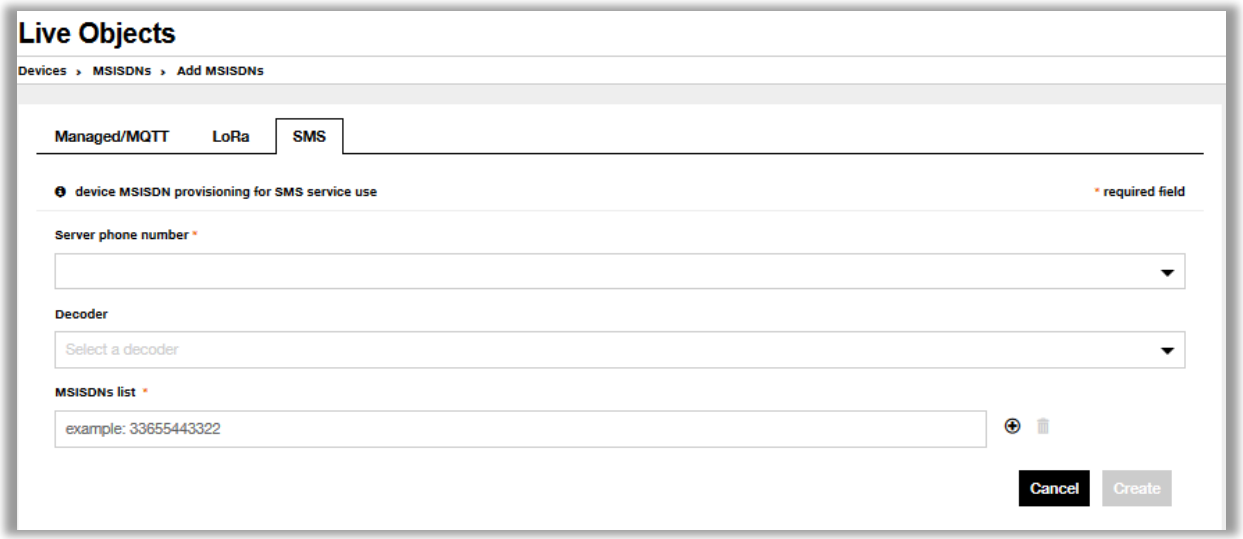
The decoder field provides a list of private decoders (accessible only by your Live Objects account) or public (accessible by all Live Objects customers).

You can define your own private decoders in the "Configuration" menu. See paragraph “Manage decoders”

To list your SMS devices, you have to select first the corresponding decoder.

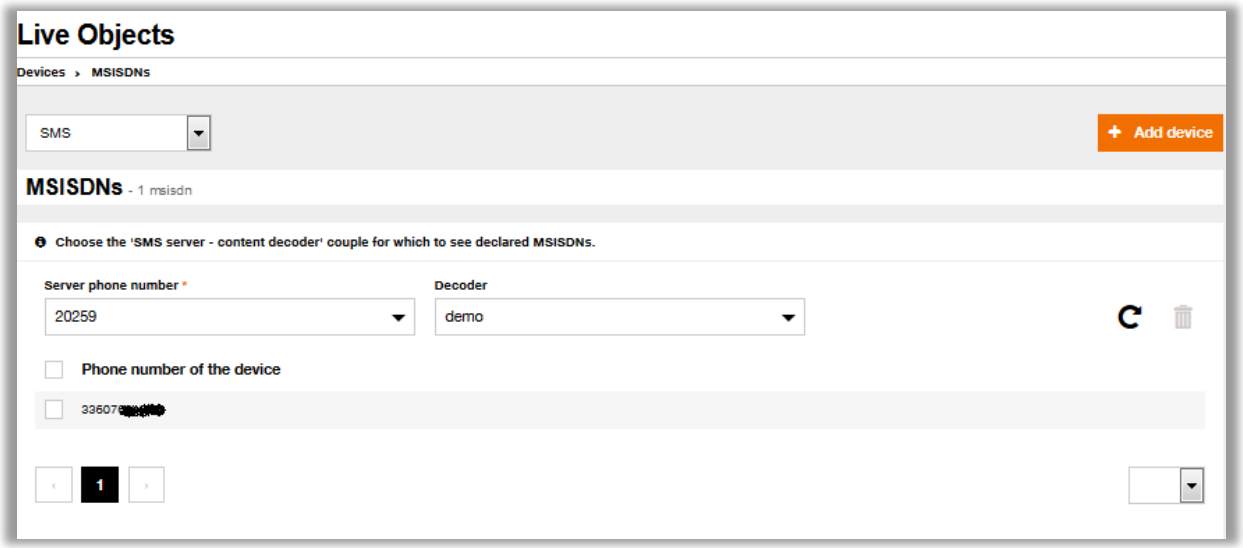
2.4.5.1. Add an SMS device

To add a new SMS device, click on the "Add a device" button in the "SMS" main menu.



First select the "Server Phone Number" (short code that will be used by your SMS device to communicate with the platform), then a decoder in the list if the payload needs to be decoded.

Enter the msisdn of your device in the format shown in the example. Do not forget to validate the entry of each msisdn by clicking on the "+" icon, then validate the creation by clicking on the button "Create". Your device is added to the list.



Caution: The list of devices displayed corresponds to the pair ("Server telephone number", "Decoder") selected. If you have created other devices with other couples (for example, same

server but with a different decoder), you will have to select the right pair to see the devices in the list.

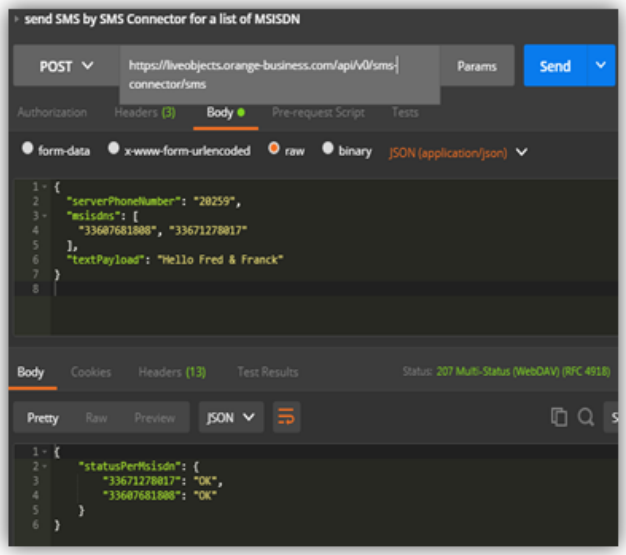
2.4.5.2. Delete an SMS device

To delete an SMS device, simply select it in the list and click on the "Delete" icon.

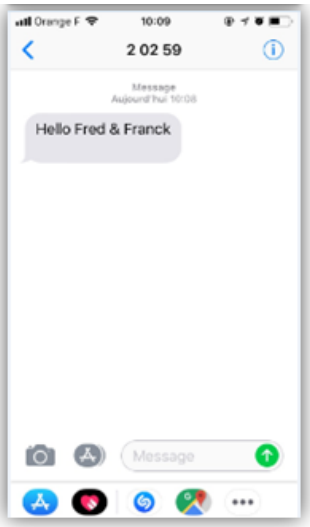
2.4.5.3. Send a command to an SMS device

In the current version, only the creation / deletion of SMS devices is possible with Live Objects portal.

The sending of command is possible with API. To do this, go to the API documentation (swagger) : https://liveobjects.orange-business.com/swagger-ui/index.html#!/SMS_Connector/sendSMSUsingPOST)



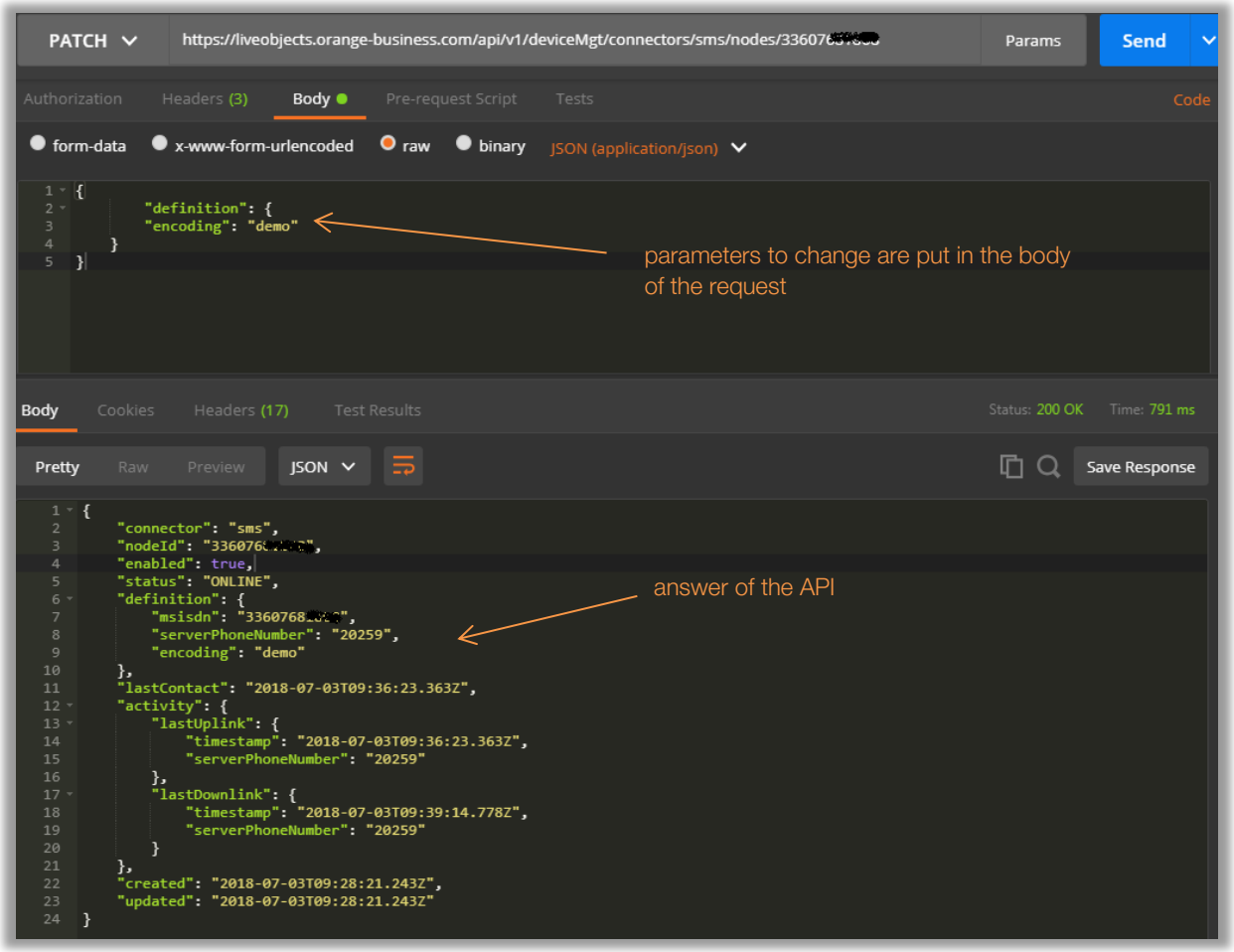
Message received on the device



2.4.5.4. Modify an SMS device

In the current version, only the creation / deletion of SMS device is possible with Live Objects portal. The modification of a device (for example, to change a decoder, must be done with the API https://liveobjects.orange-business.com/swagger-ui/index.html#!/Device_management_-_Connector_nodes_-_V1/updateNodeUsingPATCH

Example for changing the decoder of a device :



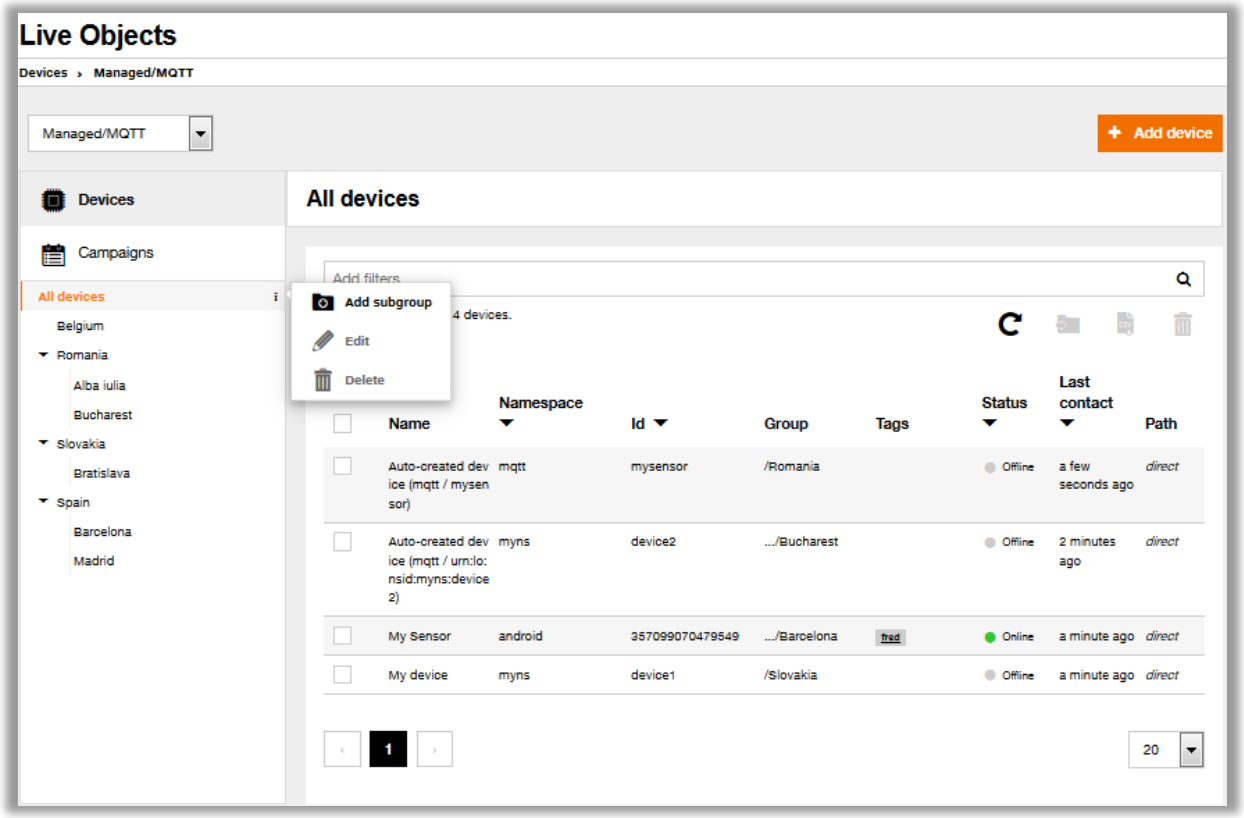
2.4.6. Groups

You can organize your fleet with groups of devices.

You can create, rename, or delete a group, or a subgroup by clicking at the root "All Devices" or an existing group or subgroup.

Note: It is not possible to delete a group that contains at least one device.

By clicking on a group you will see the list of devices that are part of it. A device can only belong to one group at a time.



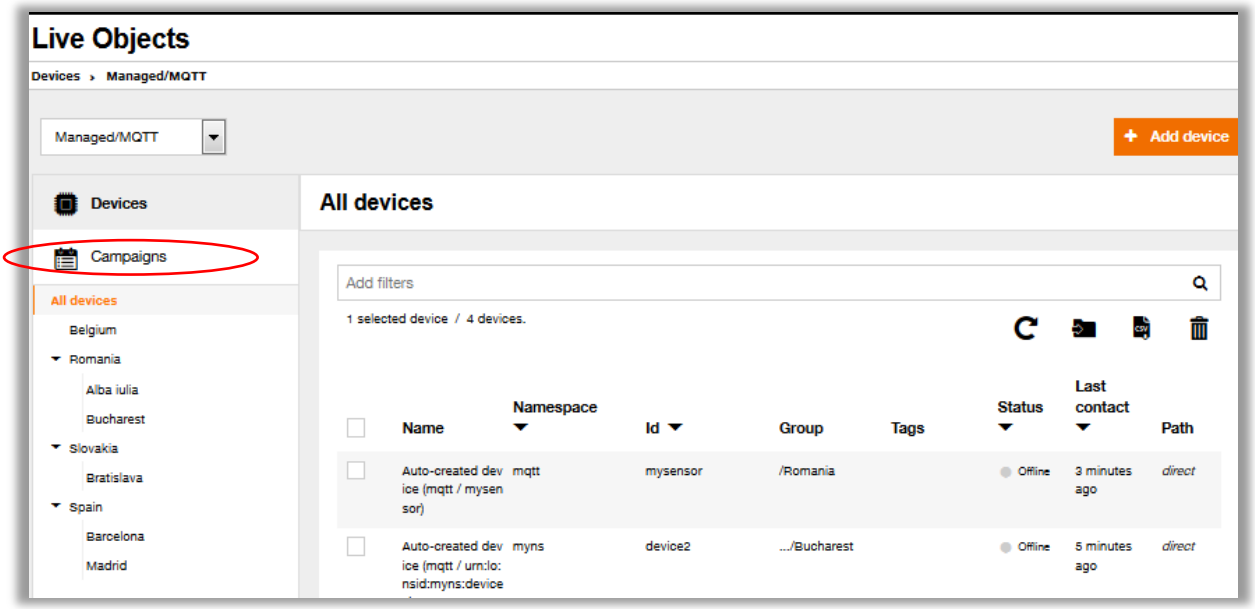
To move a device from one group to another, go to the device details, or select it from the list and click on the "Move" icon.

2.4.7. Campaign management

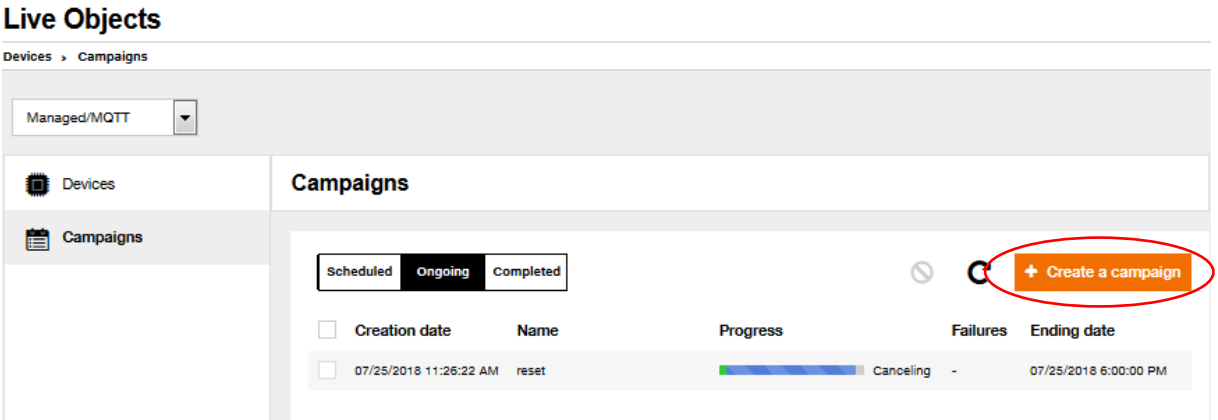
Campaigns enable to plan, execute, and track bulk device management operations. Supported operations are command send, configuration update, firmware update. Campaign management is currently available only for MQTT devices.

2.4.7.1. Create a campaign

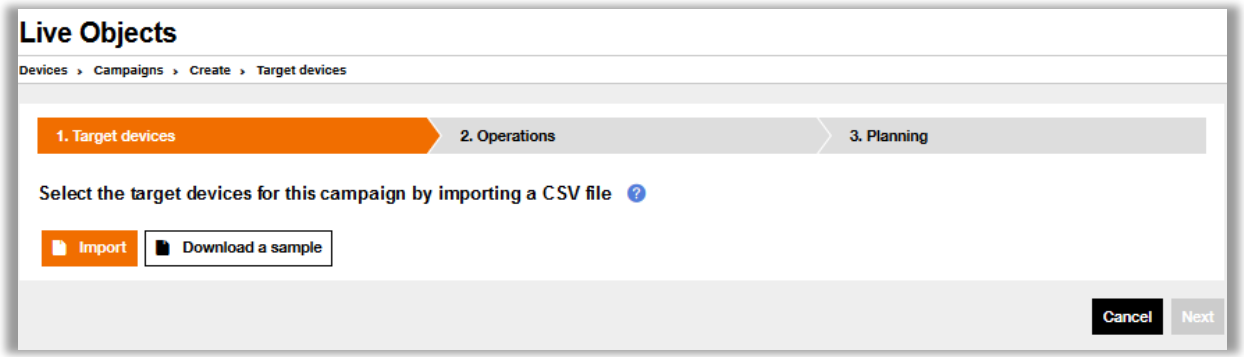
To manage campaign go to menu « Devices », and click on « Campaigns »,



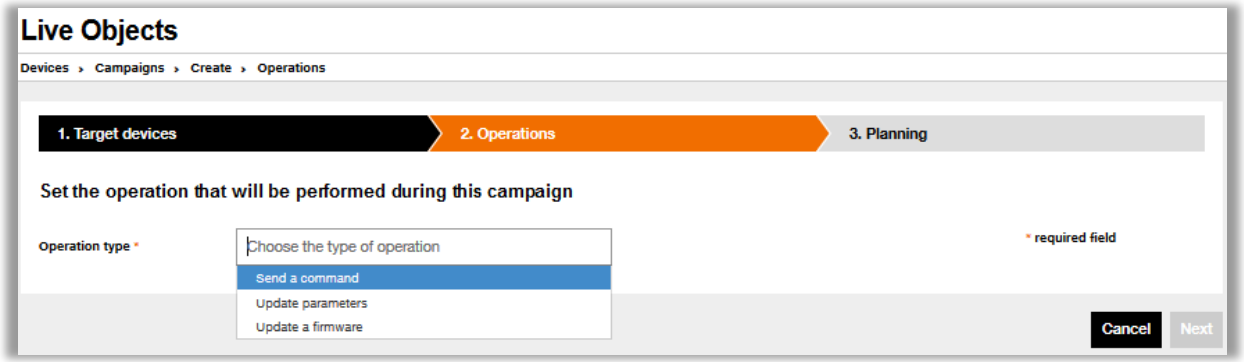
then click on « Create a campaign»



The first step is to select the targets. This can be done either by importing a list of existing devices with a csv export of the MQTT fleet (see the paragraph "Export a list of devices"), or by importing a list of devices that does not yet exist in Live Objects; in this case the import file will contain only the column "Id".

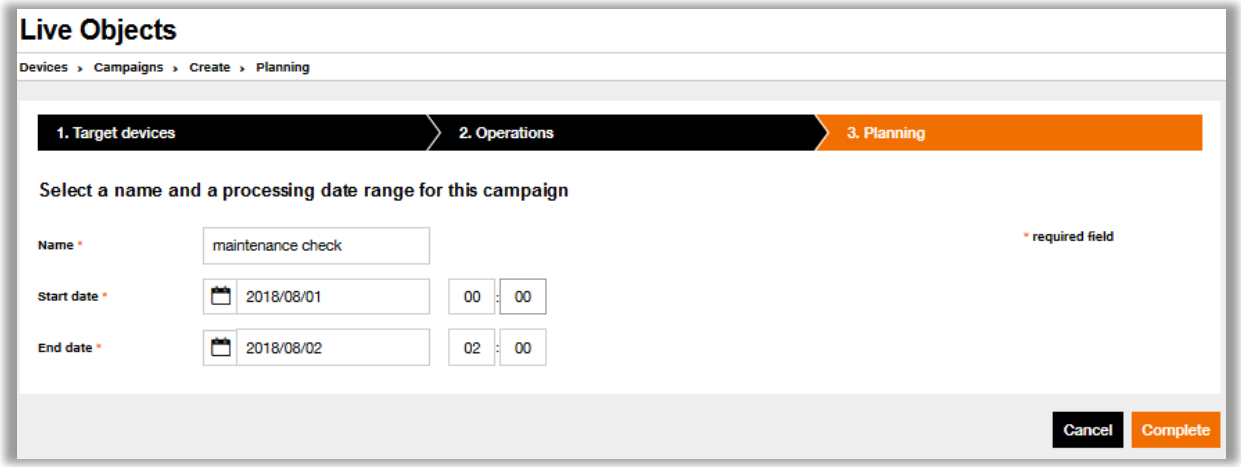


The second step is to indicate the type of operation that will be applied to the selected devices, and to provide the parameters corresponding to the operation (See the paragraph "MQTT devices" for a description of these operations).



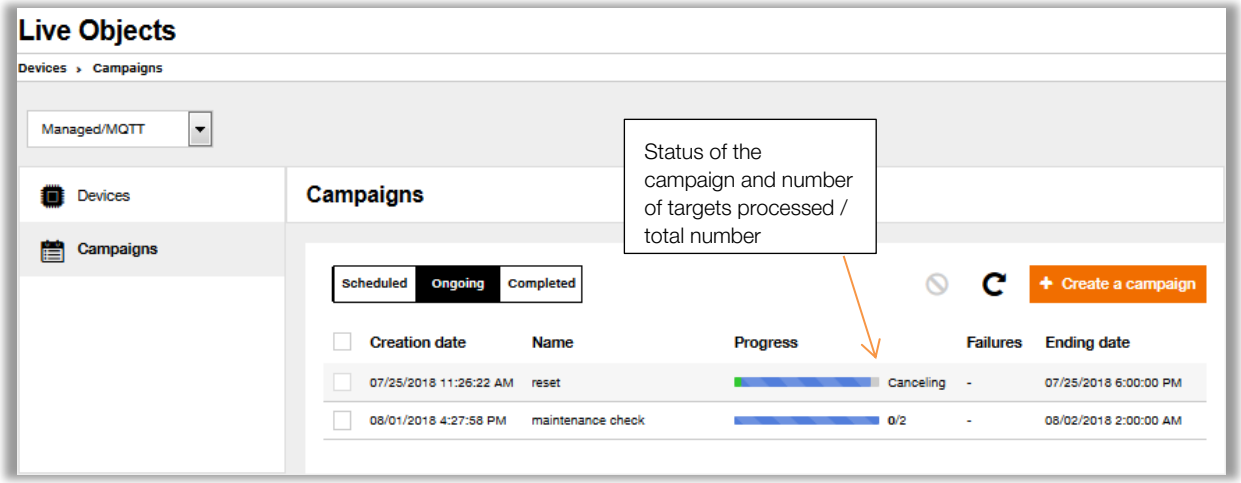
NB: The Live Objects portal allows you to choose only one operation to apply to target devices. If you need to combine within the same campaign several operations per device (e.g. a command, followed by a firmware update) you can do it using the APIs https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html#_campaign_management and https://liveobjects.orange-business.com/swagger-ui/index.html#!/Campaign_management

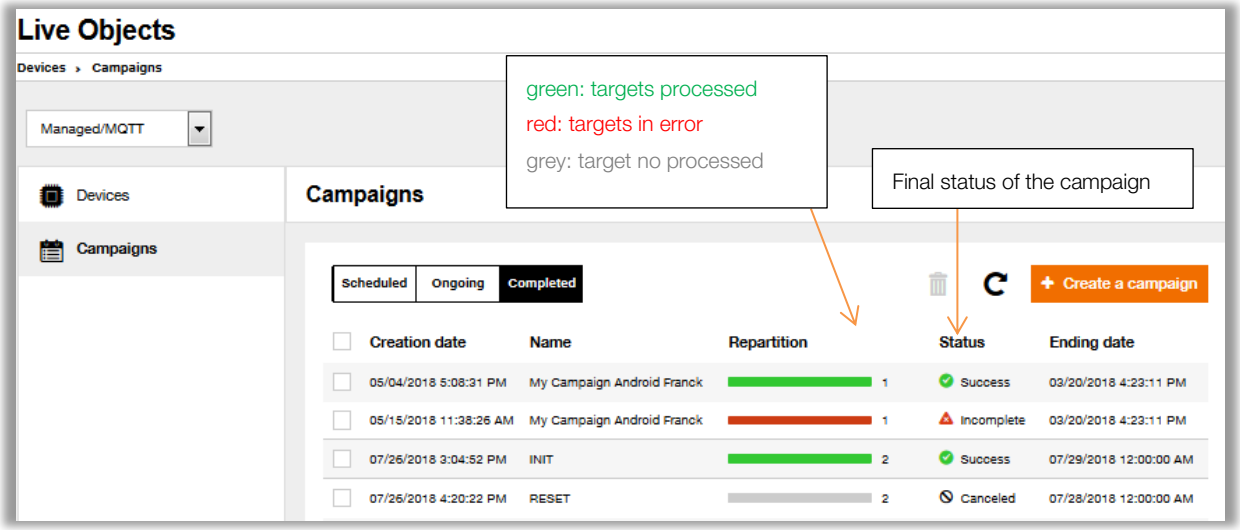
The third step consists in plan the campaign.



2.4.7.2. Follow a campaign

Campaigns are sorted depending on their planning: « Scheduled», « Ongoing» et « Completed». Click on the corresponding tab to display them.





2.4.7.3. Status of a campaign

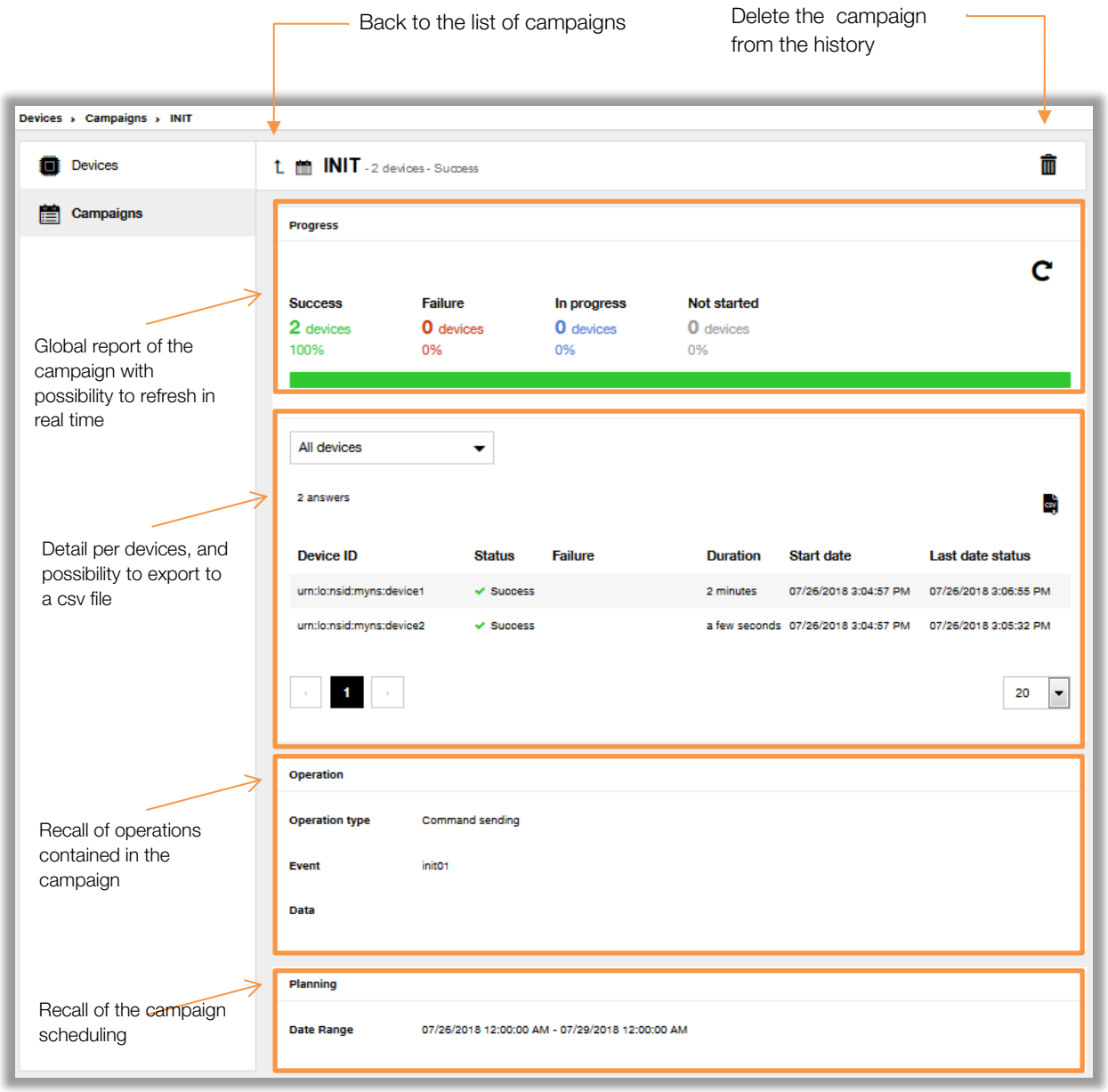
Status	Meaning
Scheduled	the start date is not yet reached
Ongoing	• the start date is reached AND • the end date is not yet reached and at least one action is still ongoing on a target When a campaign is running, its progress is displayed: number of targets reached / total number of targets, as well as failed targets
Canceling	the campaign is being canceled. Waiting for the current actions to finish. Actions that did not start will not start
Canceled	the campaign has been canceled. Some scheduled actions may not have been completed
Success	the campaign ended without error: All targets were reached and the operations successfully completed* (* See corresponding paragraphs for the actions concerned)
Incomplete	the end date has expired, but targets could not be reached, or some operations failed.

2.4.7.4. Status of a target device

Status (EN)	Meaning
Success	the operations were carried out successfully* on the target device (* See the corresponding paragraphs for the concerned actions)
Not started	no operation has been started yet on the target device
In progress	an operation is ongoing on the target device
Failure	an operation has failed on the target device

2.4.7.5. Campaign report

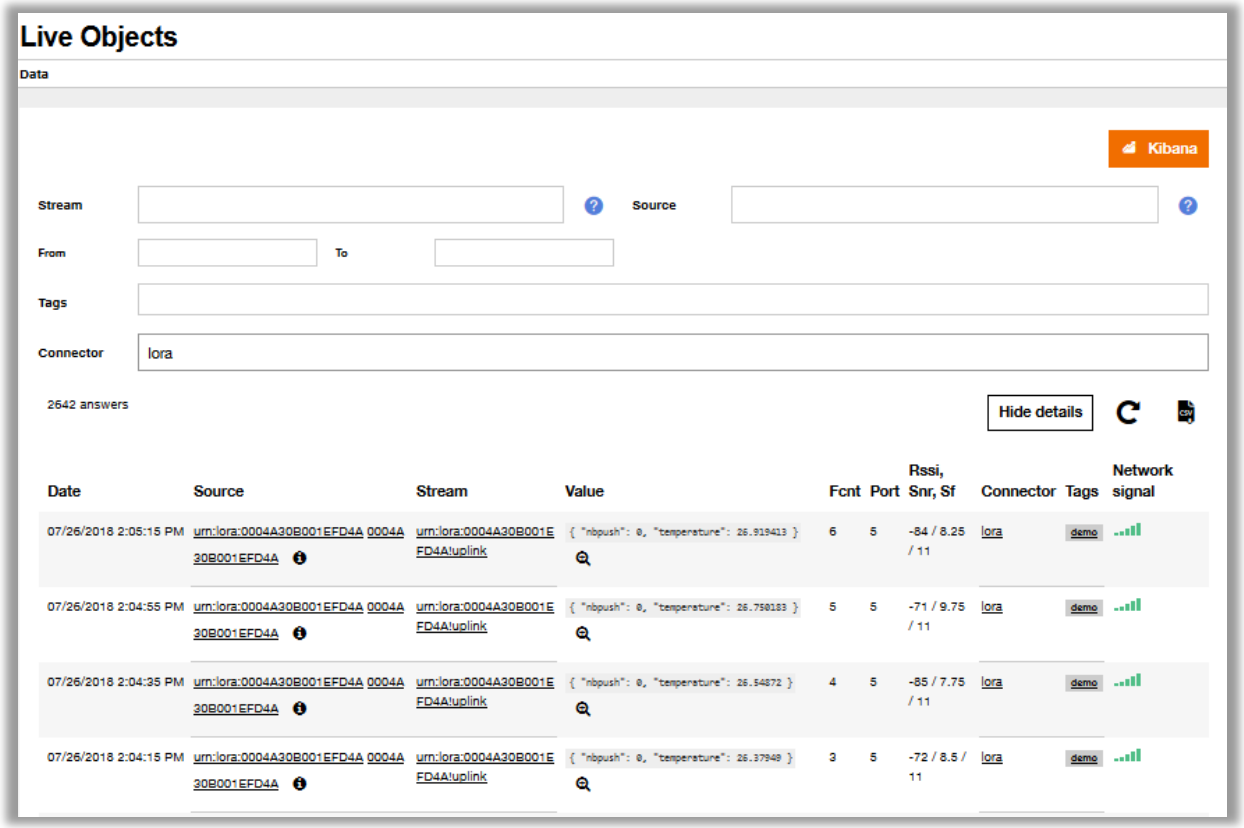
The detailed status of a campaign can be consulted at any moment. For that just click on it on the list.



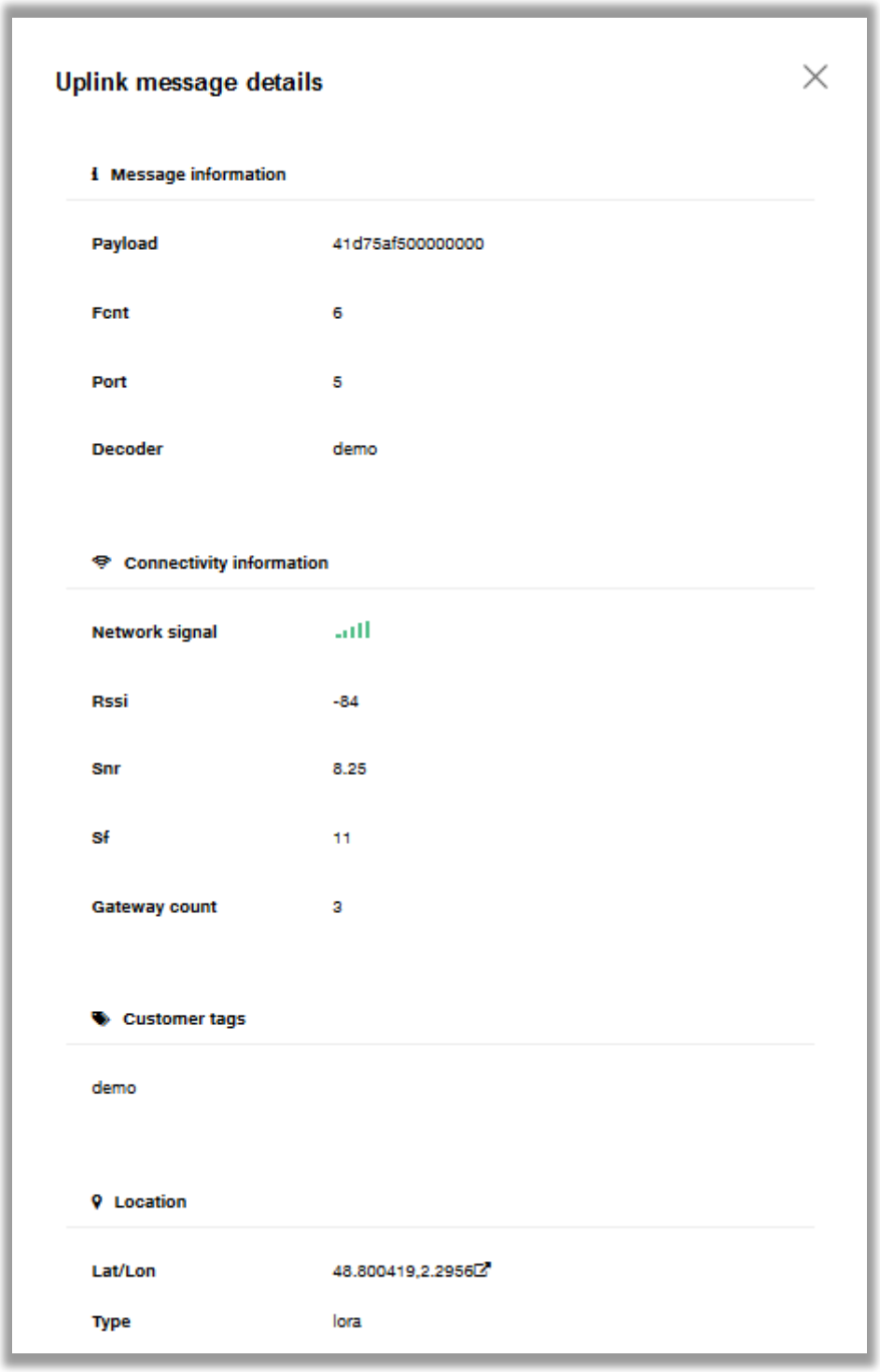
2.5. Data

Data stored in Live Objects can be visualized in the menu « Data »

The data display can be filtered using different fields:
• **Stream:** This is the time series in which the data are stored. A stream can contain all the data sent by the same device (default for LoRa), or all data of a given type (e.g. "temperatures", "pressure" ...)
• **Source:** The source that have produced the data (e.g. URN of a device)
• **From / To:** The time range to search
• **Tags:** Labels inserted in the data message by the device that issued it.
• **Interface:** This is the type of interface that your device uses to communicate with the platform. It can be "lora", "mqtt", "http", "sms"



For each data displayed in the table:
• **Date:** This is the date of the data communicated by the device (or the gateway in the case of Lora), thanks to the "timestamp" field of the data message. (NB: If this field not provided, the date is the date of arrival of the data in the platform)
• **Tags:** Like devices, data can be tagged. This tag is inserted in the data message by the device that emits it.
• **Value:** This is the "data message" produced by the device. Details are available by clicking on it.
• Other details specific to the type of device (LoRa or MQTT connectivity)



Uplink message details

Message information

Payload	41d75af500000000
Fcnt	6
Port	5
Decoder	demo

Connectivity information

Network signal	
Rssi	-84
Snr	8.25
Sf	11
Gateway count	3

Customer tags

demo

Location

Lat/Lon	48.800419,2.2956
Type	lora

Decoded data


```
{
  "nbpush": 0,
  "temperature": 26.919413
}
```

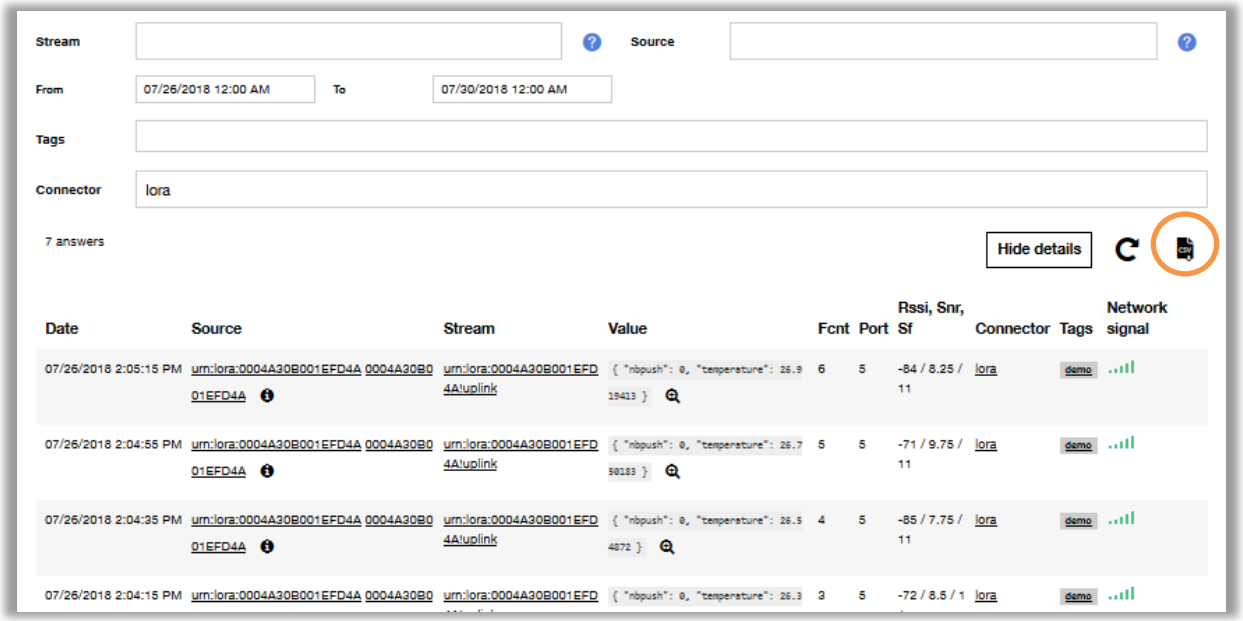

Details


```
{
   "metadata": {
      "source": "urn:lora:0004A30B001EFD4A",
      "connector": "lora",
      "encoding": "demo",
      "network": {
         "lora": {
            "signalLevel": 5,
            "rssi": -84,
            "gatewayCnt": 3,
            "sf": 11,
            "port": 5,
            "snr": 8.25,
            "location": {
               "alt": 0,
               "accuracy": 1500,
               "lon": 2.2956,
               "lat": 48.800419
            },
            "fcnt": 6,
            "devEUI": "0004A30B001EFD4A"
         }
      }
   },
   "streamId": "urn:lora:0004A30B001EFD4A!uplink",
   "created": "2018-07-26T12:05:15.606Z",
   "extra": null,
   "location": {
      "provider": "lora",
      "alt": 0,
      "accuracy": 1500,
      "lon": 2.2956,
      "lat": 48.800419
   },
   "model": "lora_v0",
   "id": "5b59b8fb7676a76f448f17db",
   "value": {
      "nbpush": 0,
      "temperature": 26.919413,
      "payload": "41d75af500000000"
   },
   "timestamp": "2018-07-26T12:05:15.318Z",
   "tags": [
      "demo"
   ]
}
```


Ok

2.5.1. Data export

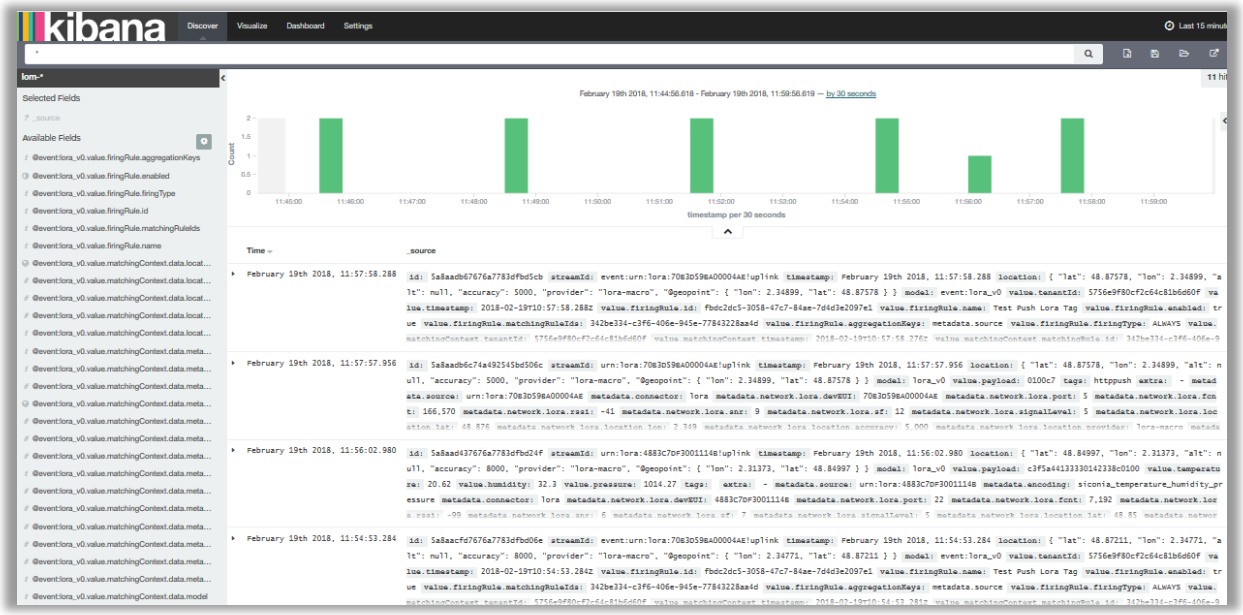
Filtered data can be exported to a csv file. Just click on the “Export” icon

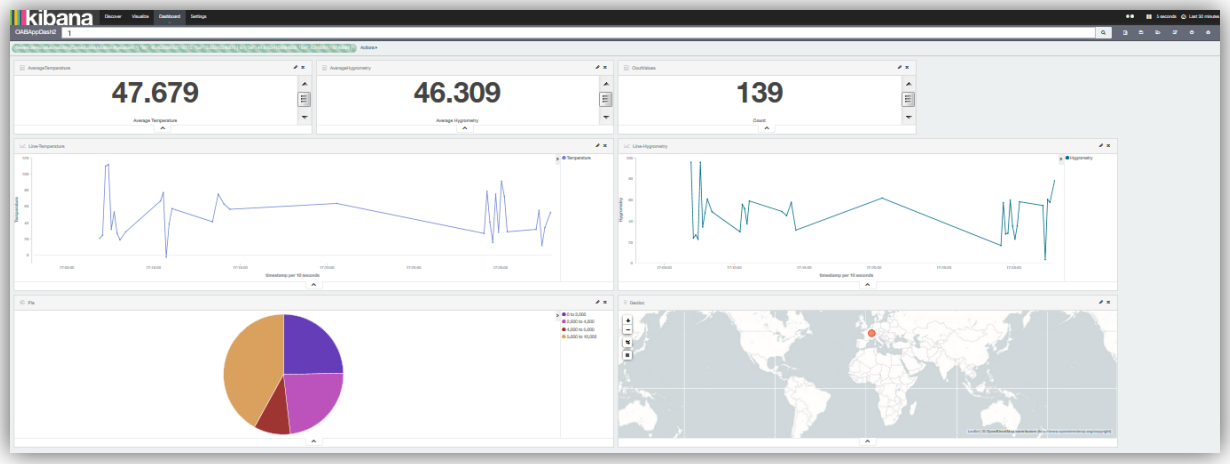


2.5.2. Advanced visualization

Live Objects relies on "Kibana" an open source tool for data mining and dashboarding. It is a based on "Elastic Search" technology that allows dynamic dashboards and advanced data search. Go to https://liveobjects.orange-business.com/#/cms/ressources-dashboards to learn more about Kibana and other tools usable with Live Objects

To launch Kibana click on the button "Kibana"





Prerequisites for MQTT devices to have data been indexed by Elastic Search:

- the "model" field must be entered in the data message returned by the device. This field provides information on the structure field “value” of the data message and allows the indexing of the data contained in this field.
- if the field “model” is not present in the data message, the data in "value" will not be indexed
- if the structure of the field “value” changes (for example a "temperature" field that goes from integer to string), the value of the field "model" must also be changed. The same data model cannot be used to index data of different structures.
- the value of the field "model" is not imposed by Live Objects, it is provided by the device

See developer guide for detailed description of a data message:
https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html#MQTT_DEV_DATA

For Lora devices

- the payload must be decoded so that the data can be indexed
- the decoder must also specify a field “model” corresponding to the structure of the decoded data, for the same reasons as explained above

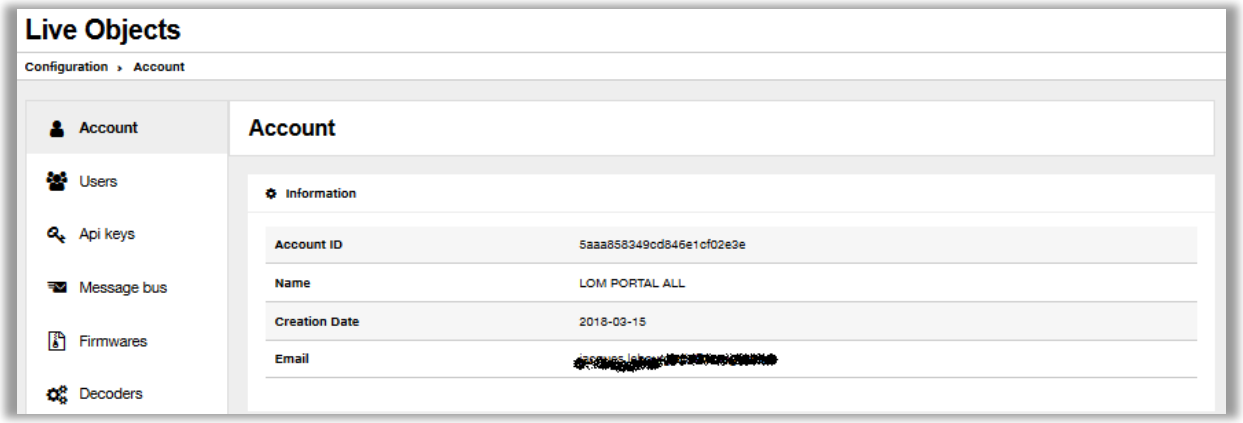
2.6. Configuration

The menu « Configuration » allows the management of your Live Objects account.

2.6.1. Identification of the account

Here are the informations that identify your Live Objects account:

- **Account ID:** It’s the Live Objects tenant ID. To provide when calling customer support
- **Name :** The name of the Live Objects tenant
- **Creation date :** Creation date of the Live Objects tenant
- **Email :** The contact email address used for the account creation



2.6.2. Roles and rights in Live Objects

Rights in Live Objects are managed through "roles". Each role defines a scope of possible actions on one or more features of the product. These roles can be assigned to users, or API keys used by devices or applications to access to Live Objects.

Two user profiles are available in Live Objects :

- « administrator » : it has all administrator rights on the Live Objects account,
- « user» : it has mainly read rights

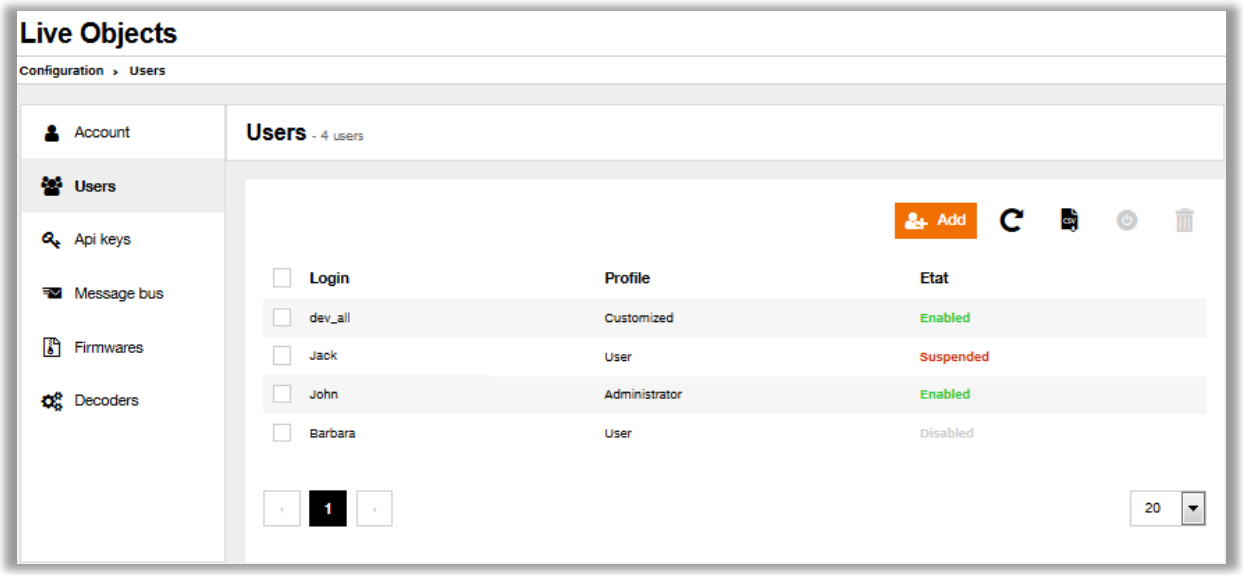
It is also possible to define a custom profile, by selecting each role manually. (See paragraph « Manage users»)

The table below summarizes the roles attributed to each profile:

Technical value (Read/Write)	Description	Administrator	User
API_KEY_R/W	API keys management	R/W	R/W
USER_R/W	Live Objects users management	R/W	
SETTINGS_R/W	Live Objects account configuration (for future use)	R/W	R/W
DEVICE_R/W	Device Management	R/W	R
DATA_R/W	Collected Data Access	R/W	R
DATA_PROCESSING_R/W	Data Decoders and Events Processing management	R/W	R
KIBANA_R	Access to Kibana	R	
BUS_CONFIG_R/W	Management of routing keys and message queues (fifo)	R/W	R
BUS_R/W	Access to message bus access using MQTT or HTTP	R/W	R/W
DEVICE_ACCESS	MQTT Device mode access (allow to create API keys for devices)	R/W	R/W
CAMPAIGN_R/W	Management of massive device management operations	R/W	R

2.6.3. Manage users

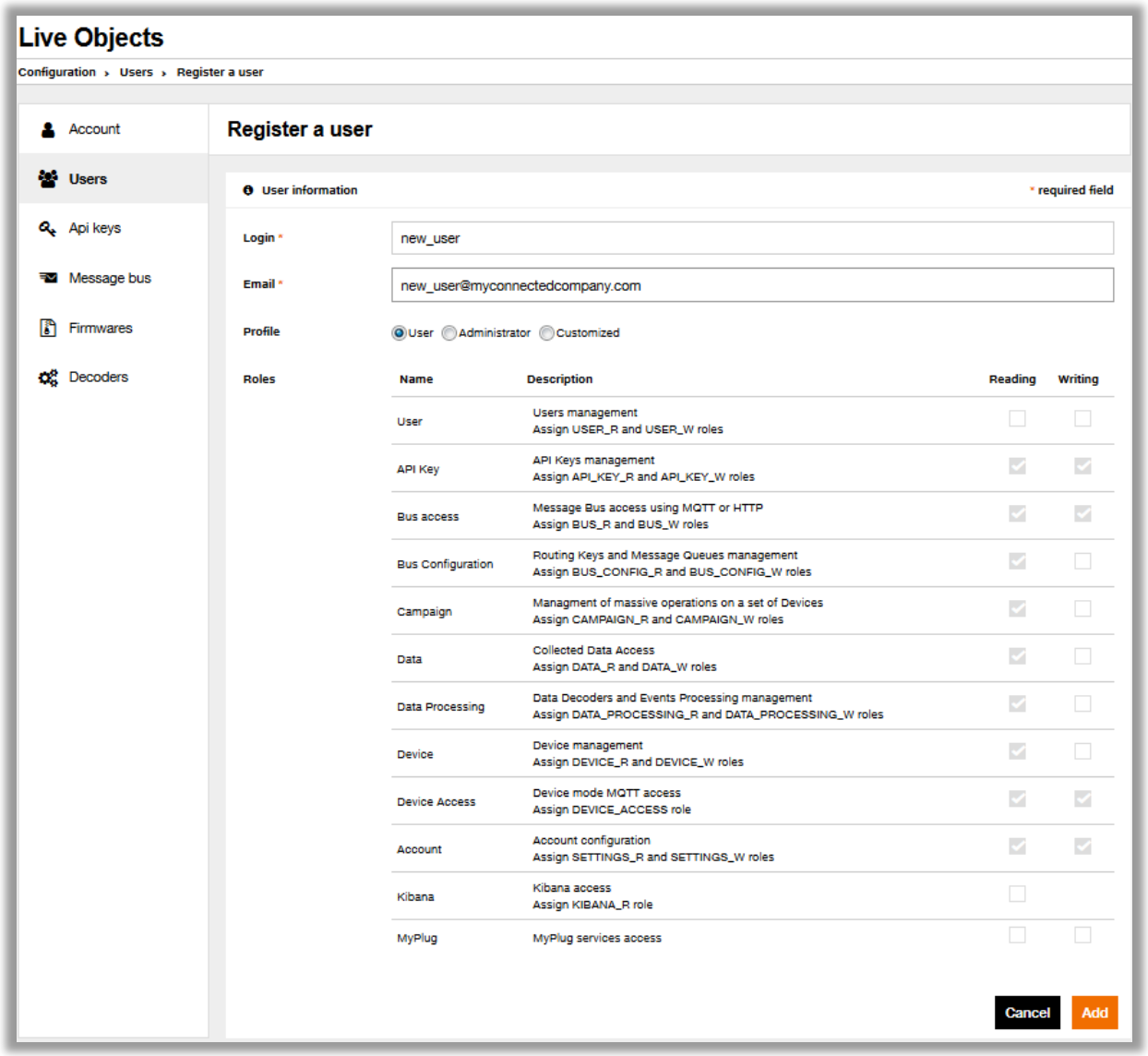
To manage Live Objects users go to menu “Users”.



Enabled: Means the user has access to Live Objects
Suspended: Means the access to Live Objects for the user has been suspended by an admin user of the account
Disabled: Means that an activation email has been sent to the user, but the user didn’t activate it yet his

2.6.3.1. Create a user

To create a new user click on the button “Add”



Login is used to log into Live Objects Portal.

Email is the contact email address where the activation email and password recovery emails will be sent

Profile allows to select predefined right profiles, or to selects custom rights

Click on button « Add » to validate the user creation. An activation email is immediately sent. This email contains an activation link with **7 days validity**.

As long as the user has not activated his access by clicking on the activation link, his status remains "inactive".

If the user has not validated his access within 7 days, it will be necessary to send again an activation email. This can be done by selecting the user(s) concerned and clicking on the "Send activation email" icon, or in the user's details by clicking on the "Send activation email" button.

2.6.3.2. Suspend / reactivate a user

It is possible to suspend access to the service for a user temporarily. For this go to the details of the user by clicking on it in the main list and click on the button "Suspend". The suspension is immediate, the user can no longer log in to the Live Objects portal.

To reactivate a suspended user, go to the user's details and click the "Enable" button. The reactivation is immediate and the user can connect again to the Live Objects portal.

2.6.3.3. Delete a user

To delete a user select it from the main list and click on the "Delete" icon. The deletion is immediate and definitive.

2.6.4. Manage API keys

API keys allow:

- to connect MQTT devices to the platform
- business applications to use Live Objects services through APIs (MQTT and REST):
    - Data mining (Search)
    - Real-time data consumption
    - Device management
    - Live Objects configuration (users, decoders etc.)

All features available through the Live Objects portal are also available through the REST APIs (see https://liveobjects.orange-business.com/#/swagger)

The MQTT APIs are reserved for:
- connecting MQTT devices (“device mode”)
- real time data consumption by business applications (“bridge mode” with payload username)

See developer guide https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html#MQTT_API

As for users, it is possible to assign roles to API keys, to allow certain features or not. For example:

- an API key used by an MQTT device only requires the DEVICE_ACCESS role

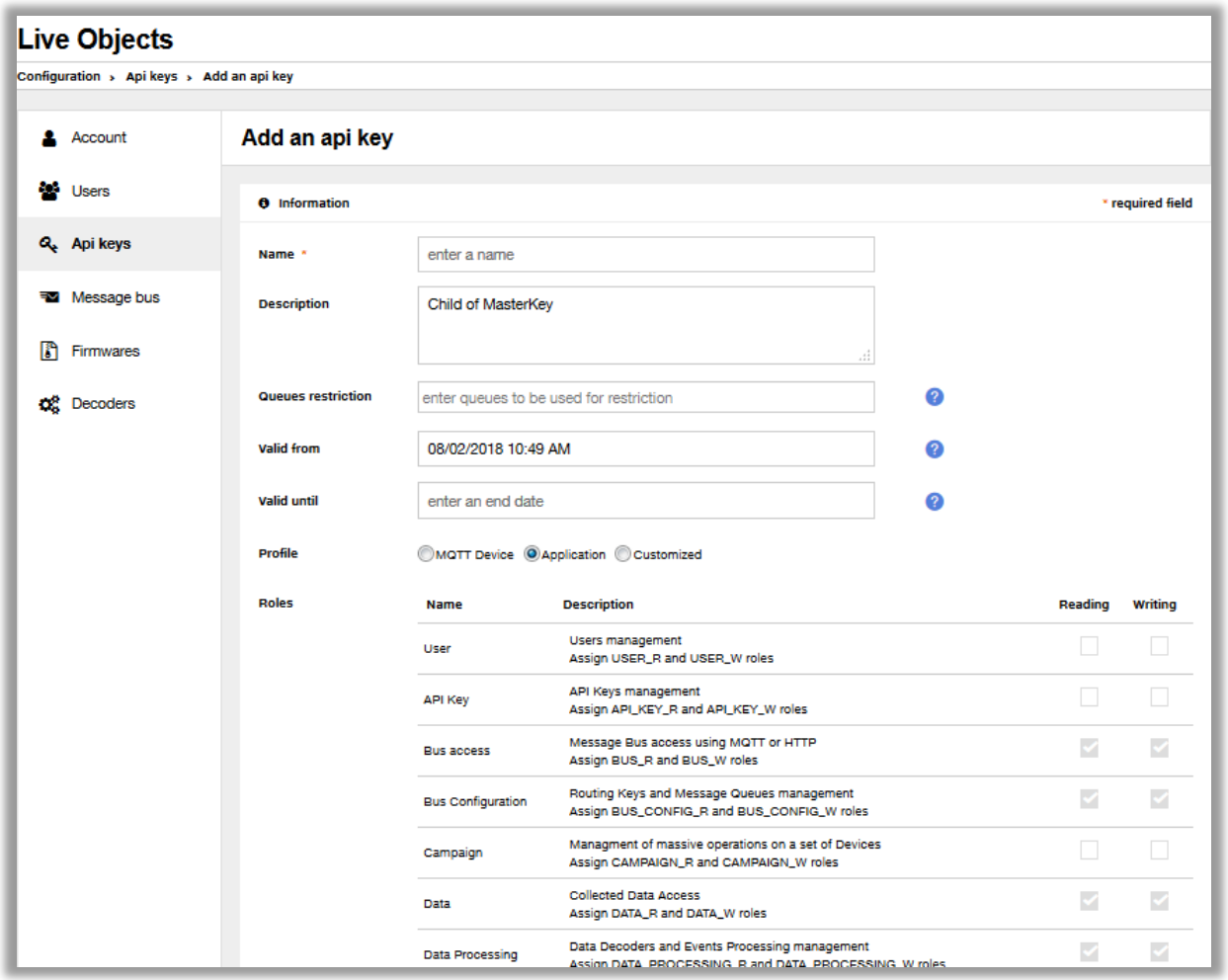
- a business application that needs to use the device management APIs will require the role DEVICE_R (if simple consultation) and DEVICE_R + DEVICE_W (if consultation and modification)
- a business application that needs to consume real-time data with MQTT will need the BUS_R role

NB: for each REST API, the necessary role is indicated in the swagger (https://liveobjects.orange-business.com/#/swagger).

To view all the API keys created in your Live Objects account, go to the "API Keys" section of the "Configuration" menu.

2.6.4.1. Create an API key

To create a new API key, click on the button “Add” in the menu “Api keys”



When creating an API key you can choose to assign to it a predefined rights profile based on your usage, or a custom profile.

The "MQTT Equipment" profile is the profile to select for the keys used for connecting MQTT devices.

The "Application" profile is the profile to be selected for the keys used by application servers on top of Live Objects for the usual use cases: real-time data consumption, configuration of message queues, management of decoders and "event processing".

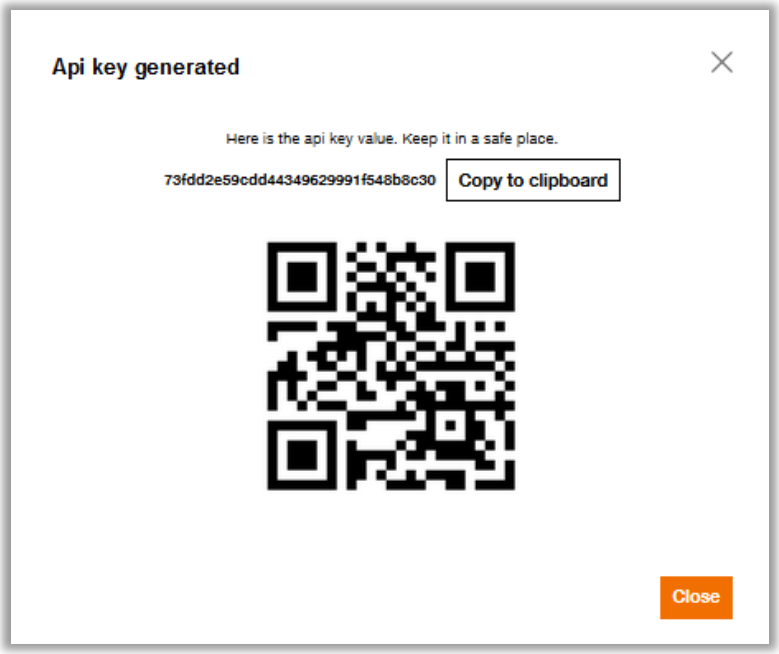
For more advanced needs ("Device Management", campaign management for example), use the "Custom" profile and select the appropriate rights.

You can also set a validity period for your key. At the end of the period the key will be automatically deactivated.

It is also possible to restrict an API key to one or more message queues (fifo). This means that this key can only be used for the consumption of routed messages into the specified queues and only that. The consumption of messages on other fifo, or the use of other APIs, including REST, with this key will be impossible.

This feature allows you to share data from some devices with a partner without giving access to other part of you Live Objects tenant.

To validate the creation of the key, click on the "Create" button. Your key is generated in the form of an alphanumeric sequence and a QR code.



Important note: The key is not stored in Live Objects. Only an hash is kept in the platform. It is your responsibility to keep it in a safe place. In case of loss the key cannot be found in Live

Objects, it will have to be deleted or regenerated and the systems using it (MQTT device, applications) will have to be updated with the new key.

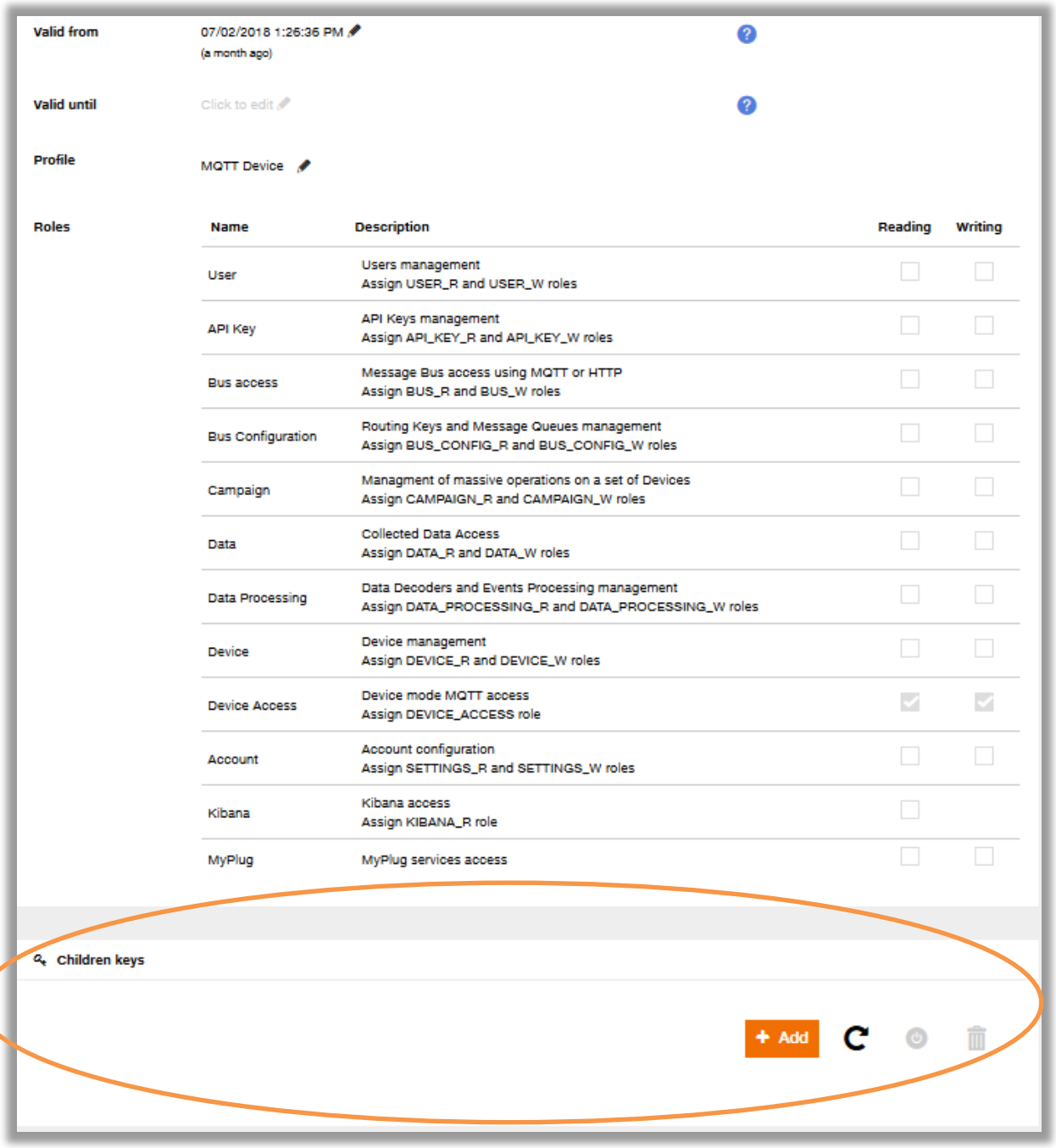
2.6.4.2. Hierarchy of API Keys

It is possible to create children of an API key. These children keys will automatically inherit roles from the parent key as well as its validity. It is not possible to add roles to a child that are not owned by the parent key.

Disabling the parent key automatically disables all of its children.

The use of child keys can be useful for managing groups and subgroups of API keys. For example, a single API key for a device group, with a daughter key for each, will disable access to Live Objects for that device group in one click.

To create a child key, simply select the a key by clicking on it in the main list, then click on the "Add" button in the modification screen, at the bottom of the page.


Valid from
07/02/2018 1:26:35 PM
(a month ago)
Valid until
Click to edit
Profile
MQTT Device
Roles
Name
Description
Reading
Writing
User
Users management
Assign USER_R and USER_W roles
API Key
API Keys management
Assign API_KEY_R and API_KEY_W roles
Bus access
Message Bus access using MQTT or HTTP
Assign BUS_R and BUS_W roles
Bus Configuration
Routing Keys and Message Queues management
Assign BUS_CONFIG_R and BUS_CONFIG_W roles
Campaign
Managment of massive operations on a set of Devices
Assign CAMPAIGN_R and CAMPAIGN_W roles
Data
Collected Data Access
Assign DATA_R and DATA_W roles
Data Processing
Data Decoders and Events Processing management
Assign DATA_PROCESSING_R and DATA_PROCESSING_W roles
Device
Device management
Assign DEVICE_R and DEVICE_W roles
Device Access
Device mode MQTT access
Assign DEVICE_ACCESS role
Account
Account configuration
Assign SETTINGS_R and SETTINGS_W roles
Kibana
Kibana access
Assign KIBANA_R role
MyPlug
MyPlug services access
Children keys
+ Add

2.6.4.3. Modify an API key

To modify a key, select it in the main list.

You can modify :

- the name
- the description
- the profile and roles (you need to regenerate the key to take effect)
- the validity period
- the queue restriction
- the children keys (add/delete)

2.6.4.4. Regenerate an API key

You can regenerate a key from the key edit screen by clicking the "Regenerate" button. A new key value will then be generated and the old value will be obsolete. If you regenerate a key (for example, if you lose it), you must update the systems that use it (MQTT devices, applications).

2.6.4.5. Disable / enable an API key

You may need to temporarily disable an API key. This is possible from the key edit screen by clicking on the "Disable" button. API access becomes impossible for any system using this key. Conversely, a key can be instantly reactivated by clicking on the "Enable" button present in the key details screen, or in the main list by clicking on the "Enable" icon.

2.6.4.6. Delete an API key

To delete one or more API keys, select them in the main list and click on the "Delete" icon.

The deletion is irreversible, and access to any system using the key(s) is impossible.

NB: Deleting a parent key will automatically delete all its children keys.

2.6.5. Manage message queues (fifo)

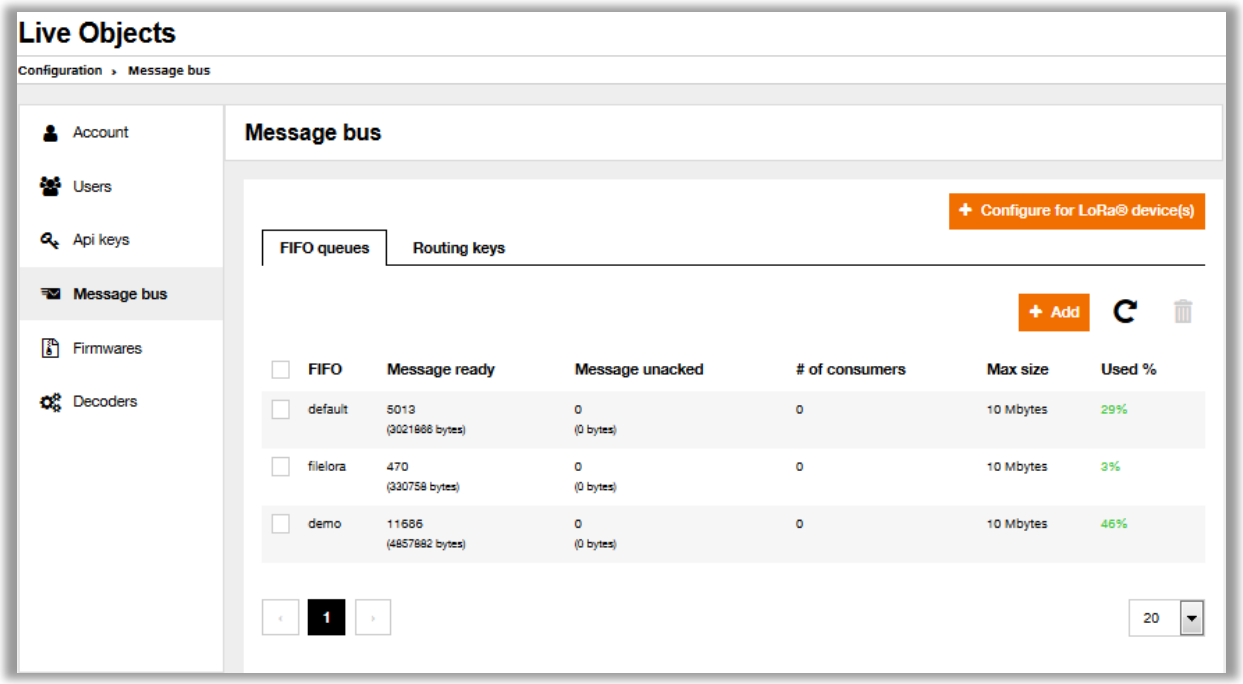
Message queues (or fifo) are useful for applications using MQTT to consume data in real time. They persist the messages in case connection cut for example.

These queues can be filled by the following routing keys (or "binding"):

- Messages of all devices: ~event.v1.data.new.#
- Messages of all LoRa devices: ~event.v1.data.new.urn.lora.#
- Messages of one LoRa device: ~event.v1.data.new.urn.lora.*{DevEUI}*.#
- Messages of all MQTT devices: ~event.v1.data.new (without **.#** at the end)
- Messages of all SMS devices: ~event.v1.data.new.urn.msisdn.#
- Messages of one SMS device: ~event.v1.data.new.urn.msisdn.*{msisdn}*/#

Note: Routing keys will evolve in a future version to allow a greater flexibility of routing (by device, by group, by type) as well as a unification of the "pattern" of key. Backward compatibility will be ensured with the current keys. In particular routing messages of a specific MQTT device will become possible.

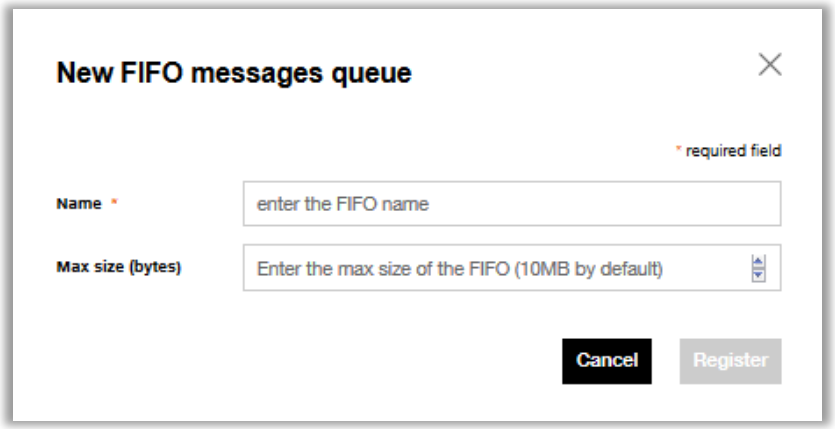
To manage message queues go to the "Message Bus" section of the "Configuration" menu.



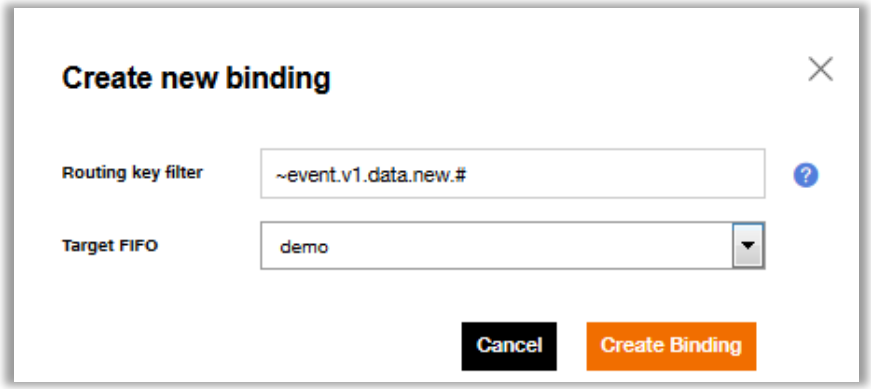
2.6.5.1. Create a message queue

Creating a message queue is done in two steps:

1. Creating the queue: Click on the button "Add", in the tab “FIFO queues”



2. Then go to the tab « Routing keys » and add a routing key to the fifo.



Repeat this action if you want to add several routing key to a same fifo.

See previous paragraph for the list of possible routing keys

Note that in the case of LoRa device, a shortcut allows this operation to be carried out in a single step. To do this click on the button "Configure for LoRa device(s)"

To route messages from a particular LoRa device to the queue, simply enter its devEUI in the corresponding field. If this field is left blank, messages from all LoRA devices in the fleet will be routed in the queue.

2.6.5.2. Modify a message queue

It is possible to add routing keys to an existing message queue. To do this go to the "Routing Keys" tab, and click on the "Add" button. Enter the routing key and target queue (see previous paragraph)

2.6.5.3. Delete a message queue

To delete a message queue, select it in the main list and click on the "Delete" icon. The deletion of the queue is definitive.

2.6.5.4. Limitations on message queues

Limitations on message queues depend on the offer you have subscribed. These limitations relate to:

- The number of queues you can create
- The overall storage allowed for fifo creation (typically 10MB x nb of fifo)

When a queue is full, new incoming messages replace older ones.

2.6.6. Manage firmwares

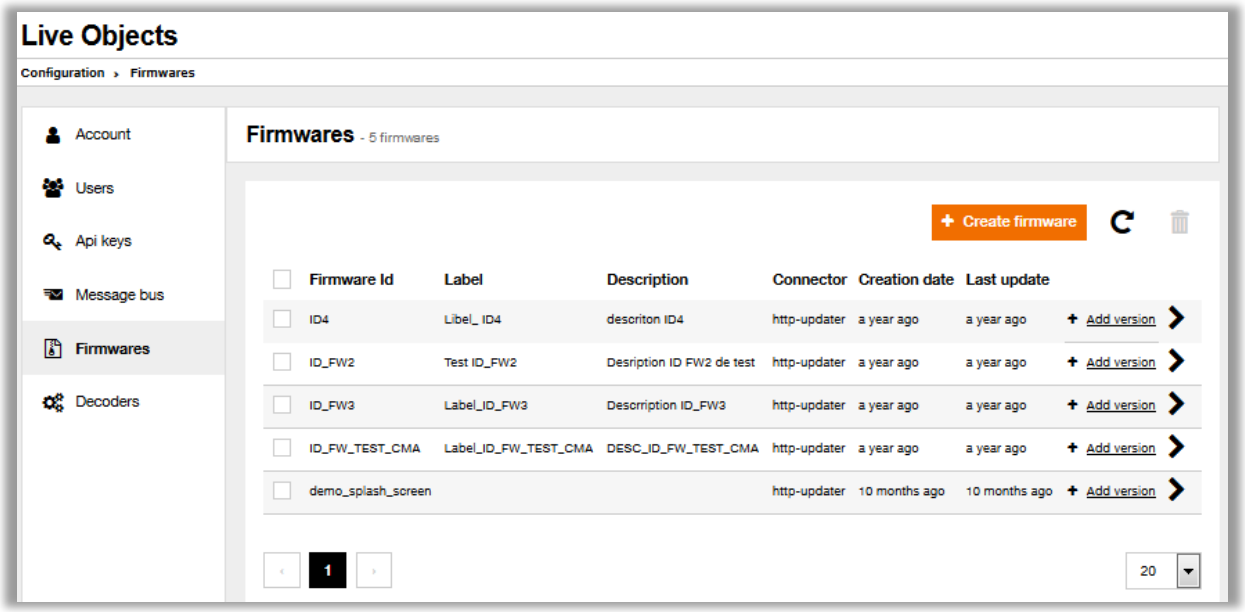
Firmware update requires that the device is connected with MQTT and implement the Live Objects "device management" logic (see the developer manual https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html#MQTT_DEV_RSC)

To be able to update the firmware of your devices, you must first have created the corresponding firmware(s) in Live Objects. This consists of two steps:

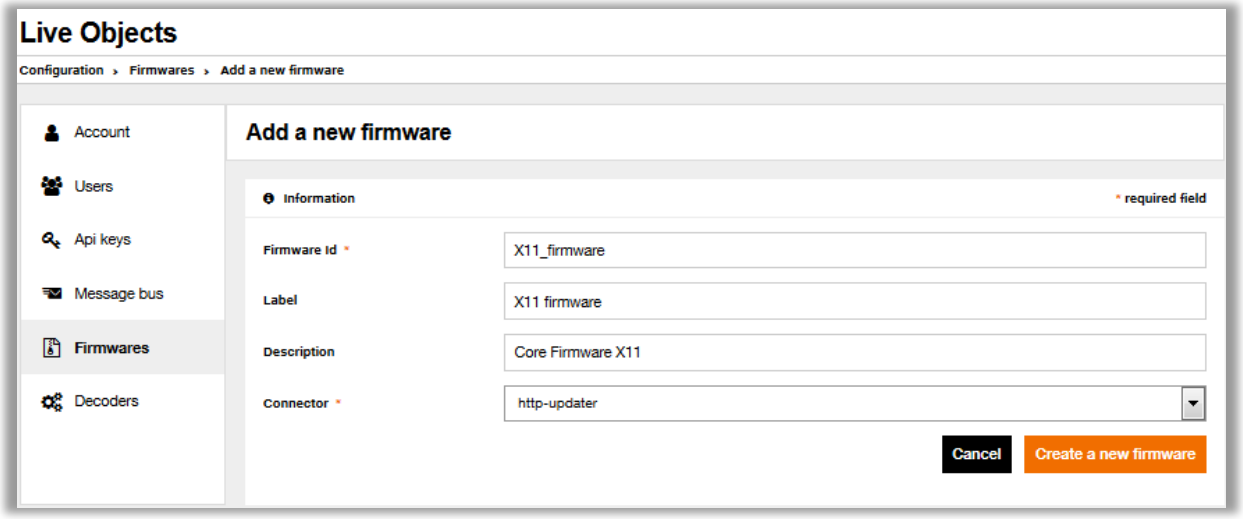
1. Creation of the firmware with the same ID as that communicated by the device
2. Create the different versions attached to this firmware and import into Live Objects the corresponding binaries.

Step 1

To create a new firmware, go to « Firmwares » in the menu « Configuration » and click on the button “Create firmware”

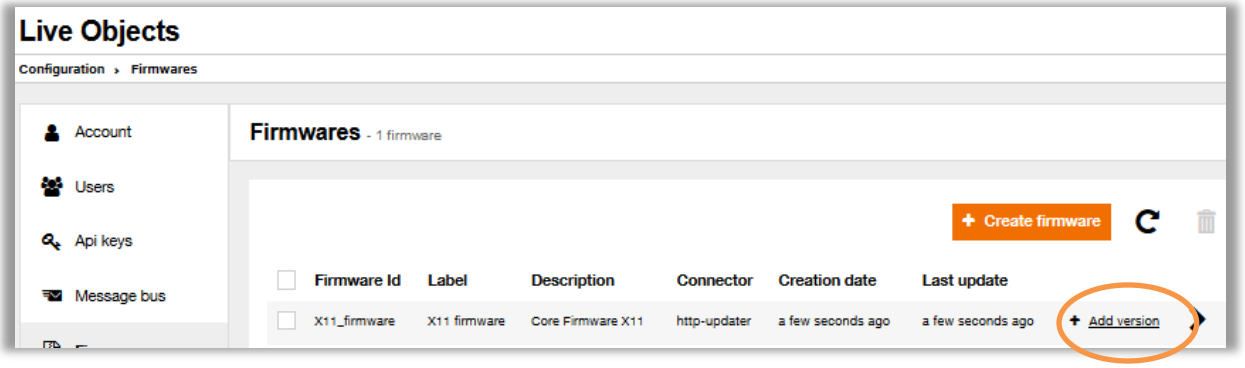


Enter the same firwmare Id as the one reported by your device, and select the "http-updater" connector. Click the "Create a new firmware" button to complete step 1.



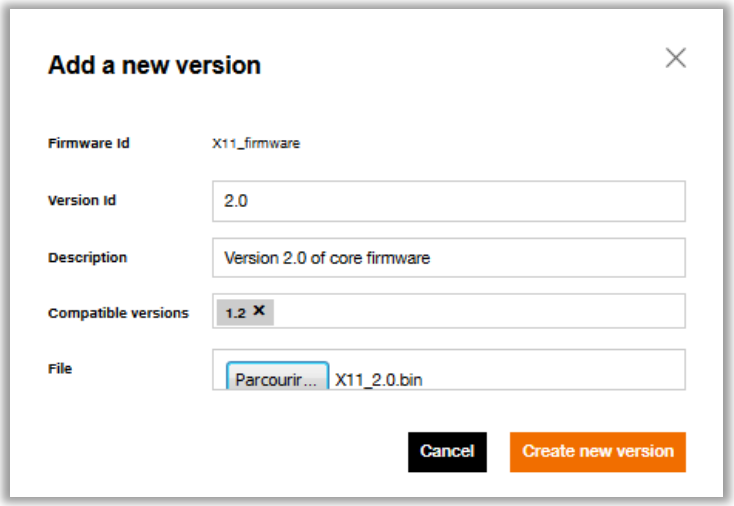
Step 2

In the main list, click on « Add version».

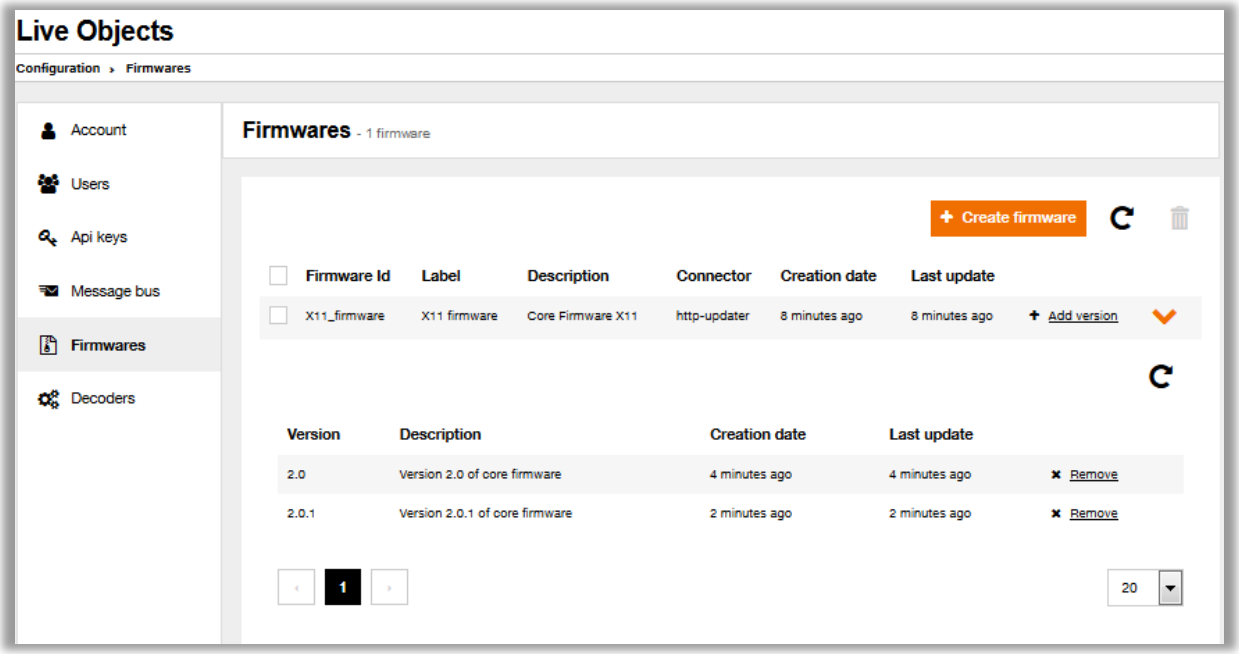


Indicate the version's id in coherence with the one communicated by your device, then import the binary file corresponding to the firmware.

NB: You have the possibility to manage the compatibility between versions of a same firmware. To do this, enter the old versions of the firmware compatible with this new version in the "Compatible version" field. If you do not specify any version, this means that the new version can be installed on any previous version.



The available versions of a firmware are listed when clicking on the icon corresponding to the line of the firmware.



You can now program a firmware update. To do this, go to the “Devices” menu. (See section "Update the firmware of an MQTT device".

2.6.7. Manage decoders

A decoder allows translating data issued by a device from a "computer" form (e.g. hexadecimal payload of a LoRa device) into a human understandable and indexable data. This translation also allows usage of advanced services based on data, such as event processing. These advanced features are currently only available with API (See developer manual https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html)

2.6.7.1. Type of decoders

Decoders available in Live Objects are of 2 types:

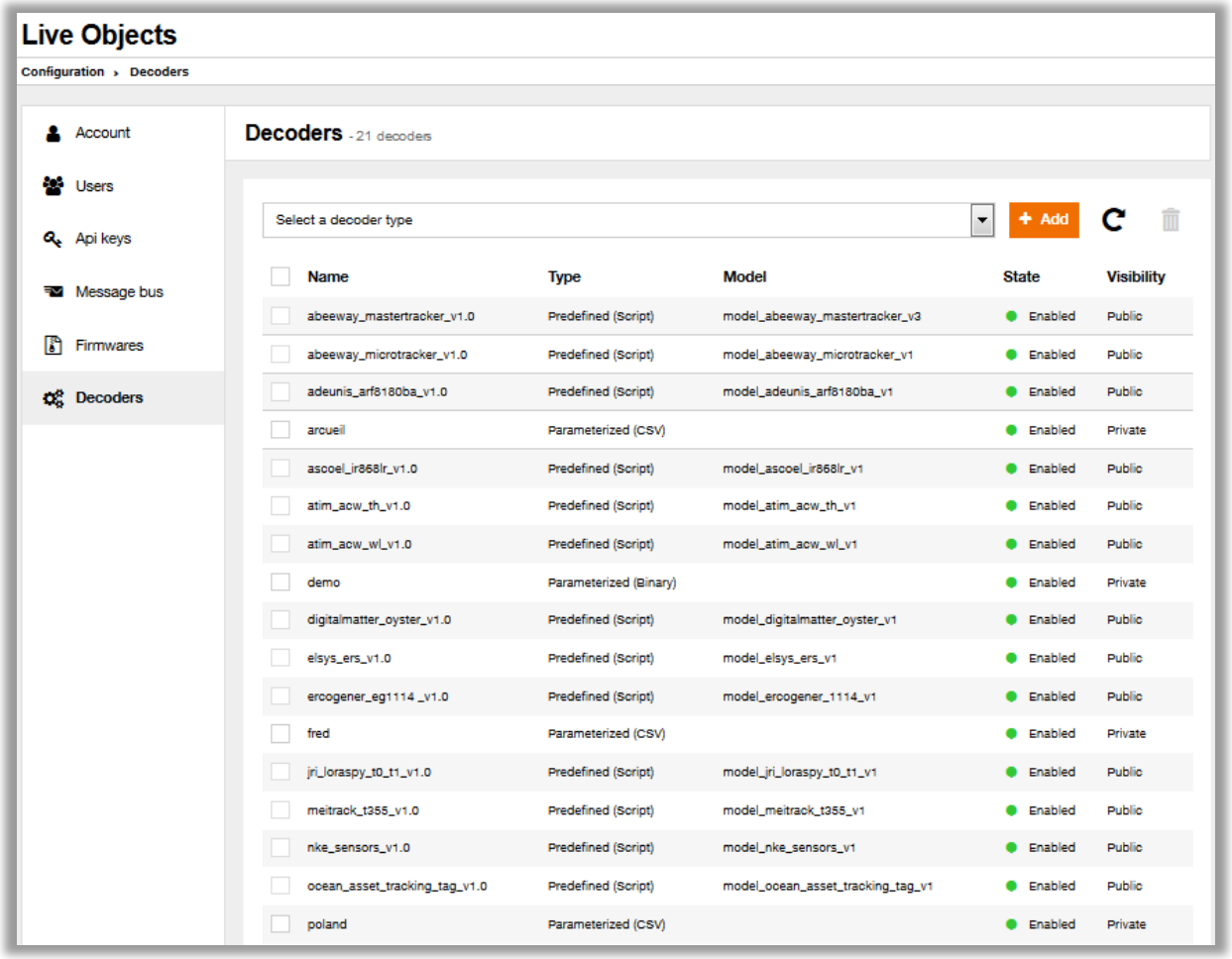
- "Configurable" decoders: These are binary decoders (for hexadecimal payloads), or csv (text payload with separator). These decoders are suitable for devices whose payload structure does not vary. You can create your own configurable decoders yourself (see paragraphs after)
- "Programmable" decoders (Script): These decoders are intended for devices whose payload structure are complex and cannot be processed by a configurable decoder. For security reasons, the deployment of this type of decoder can only be done by Orange teams (see paragraph "Create a programmable decoder")

2.6.7.2. Visibility of decoders

Decoders can be "private", that is to say visible and usable only in your Live Objects tenant; this is the case of the configurable decoders that you create yourself.

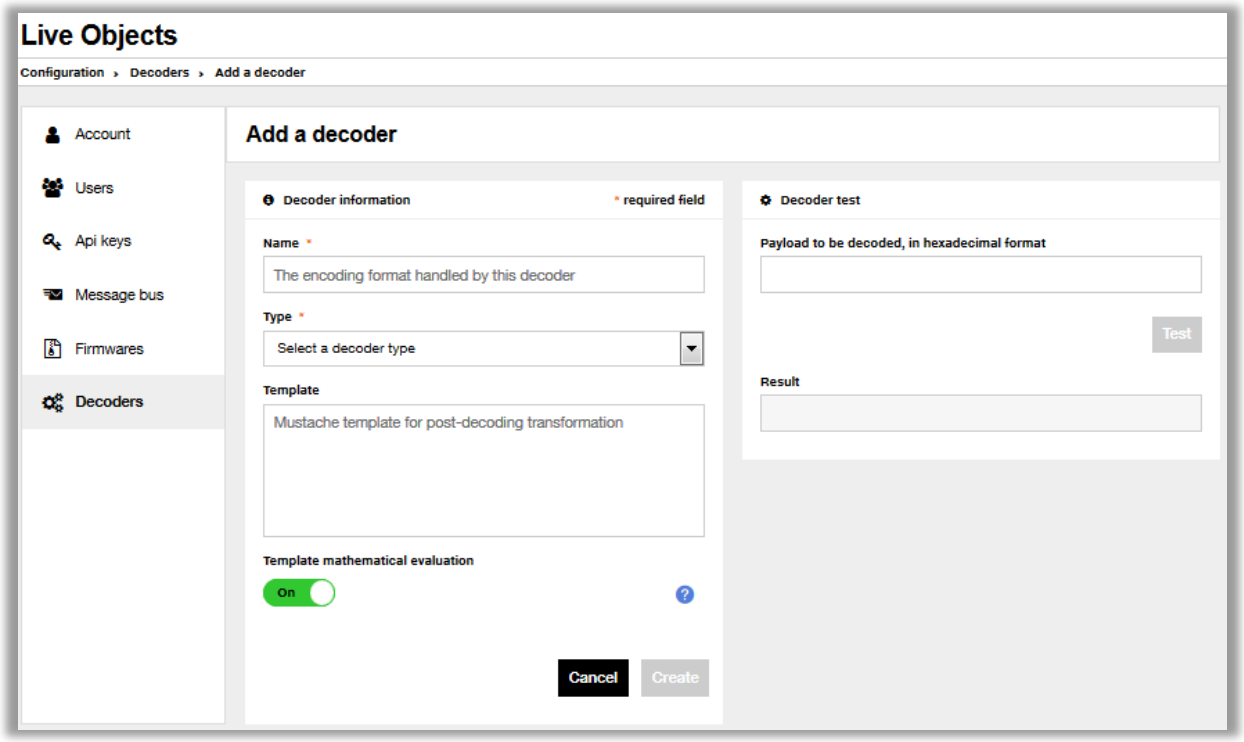
Orange also offers as part of its catalog of objects https://iotmarket.orange.com a list of predefined decoders adapted to different hardware. These decoders are called "public" that is to say, visible and usable by all Live Objects customer.

The list of public and private decoders is displayed in the menu "Decoders".


Live Objects
Configuration › Decoders
Account
Users
Api keys
Message bus
Firmwares
Decoders
Decoders - 21 decoders
Select a decoder type
+ Add
Name
Type
Model
State
Visibility
abeeway_mastertracker_v1.0 Predefined (Script) model_abeeway_mastertracker_v3 Enabled Public
abeeway_microtracker_v1.0 Predefined (Script) model_abeeway_microtracker_v1 Enabled Public
adeunis_arf8180ba_v1.0 Predefined (Script) model_adeunis_arf8180ba_v1 Enabled Public
arcueil Parameterized (CSV) Enabled Private
ascoel_ir868lr_v1.0 Predefined (Script) model_ascoel_ir868lr_v1 Enabled Public
atim_acw_th_v1.0 Predefined (Script) model_atim_acw_th_v1 Enabled Public
atim_acw_wl_v1.0 Predefined (Script) model_atim_acw_wl_v1 Enabled Public
demo Parameterized (Binary) Enabled Private
digitalmatter_oyster_v1.0 Predefined (Script) model_digitalmatter_oyster_v1 Enabled Public
elsys_ers_v1.0 Predefined (Script) model_elsys_ers_v1 Enabled Public
ercogener_eg1114 _v1.0 Predefined (Script) model_ercogener_1114_v1 Enabled Public
fred Parameterized (CSV) Enabled Private
jri_loraspy_t0_t1_v1.0 Predefined (Script) model_jri_loraspy_t0_t1_v1 Enabled Public
meitrack_t355_v1.0 Predefined (Script) model_meitrack_t355_v1 Enabled Public
nke_sensors_v1.0 Predefined (Script) model_nke_sensors_v1 Enabled Public
ocean_asset_tracking_tag_v1.0 Predefined (Script) model_ocean_asset_tracking_tag_v1 Enabled Public
poland Parameterized (CSV) Enabled Private

2.6.7.3. Create a binary decoder (configurable)

To create your own configurable decoder, click on the button « Add ».

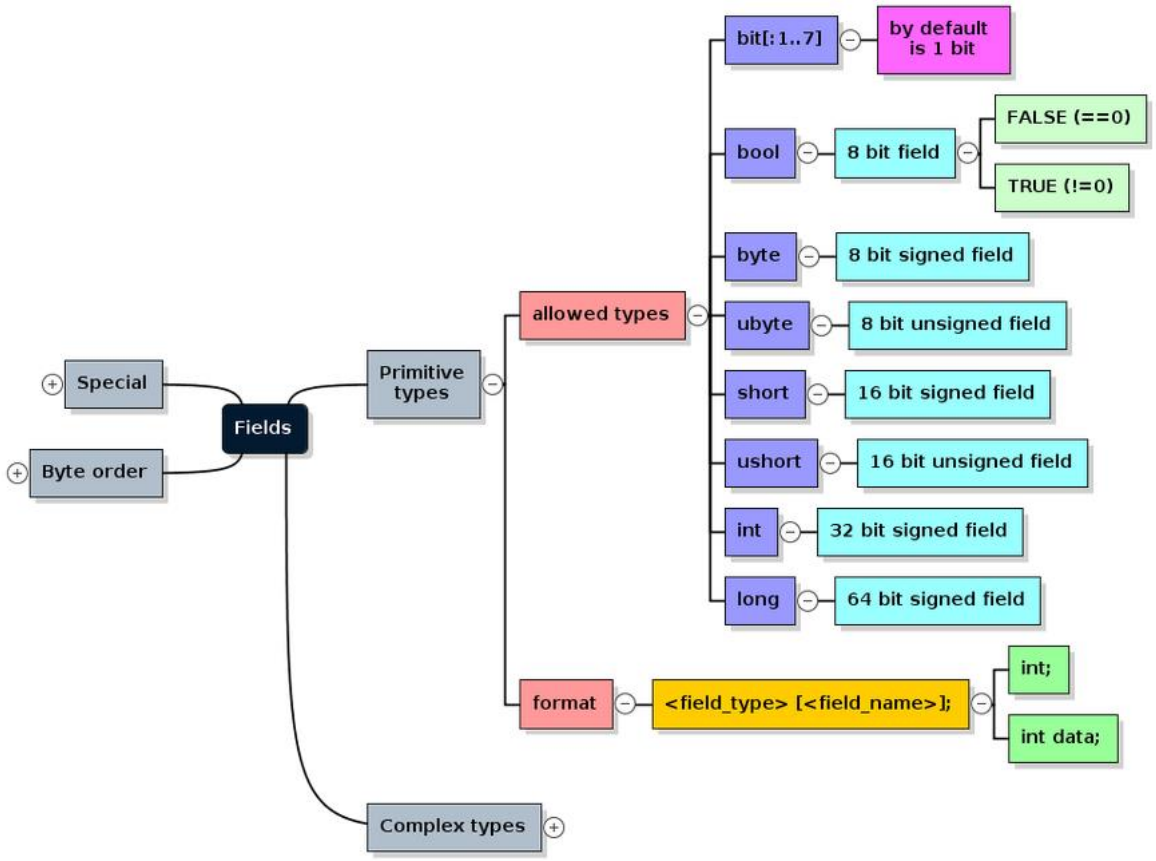


Decoder creation is in two step:

1. Definition and test
2. Creation

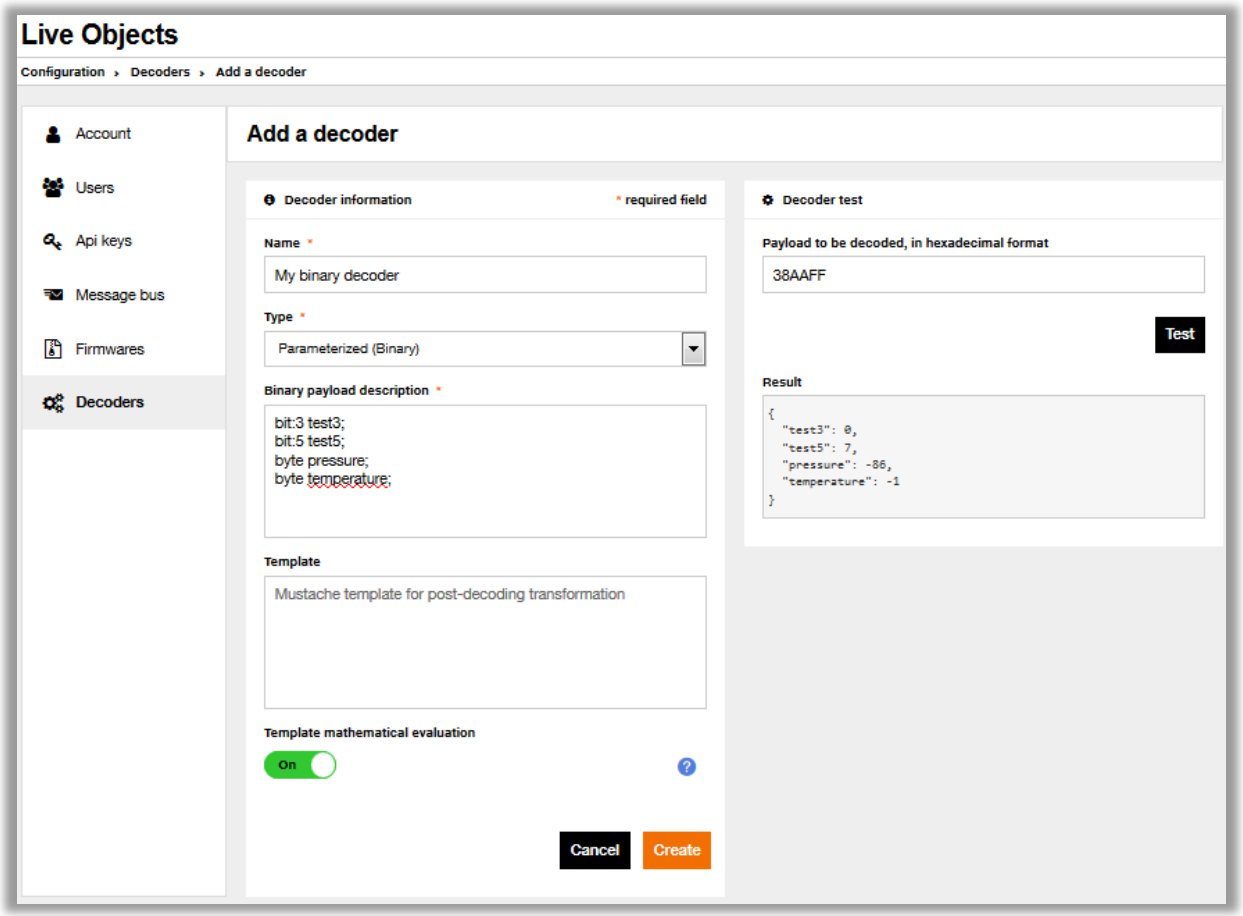
Step 1: Definition of the decoder

In this step you have to describe the structure of the payload. Live Objects relies on the Java Binary Block Parser (JBBP) for the description of binary structures. The different types are available here https://github.com/raydac/java-binary-block-parser#primitive-types



In the field « Binary payload description» enter the structure of your payload as follows: `variable type;`

In the example given below, the variable "test3" is coded on 3 bits, the variable "test5" on 5 bits and the variables "pressure" and "temperature" on 1 byte each. This constitutes a payload of 3 bytes in total.



It is recommended to test the decoder before creating it. To do this, do not immediately click on the "Create" button, but first enter a test payload in the field "Payload to be decoded, in hexadecimal format" and click on the "Test" button. In the above example the 38AAFF payload will match the decoded values

```
{
  "test3": 0,
  "test5": 7,
  "pressure": -86,
  "temperature": -1
}
```

Indeed 38AAFF matches to the binary sequence 00111000 10101010 11111111

38 AA FF

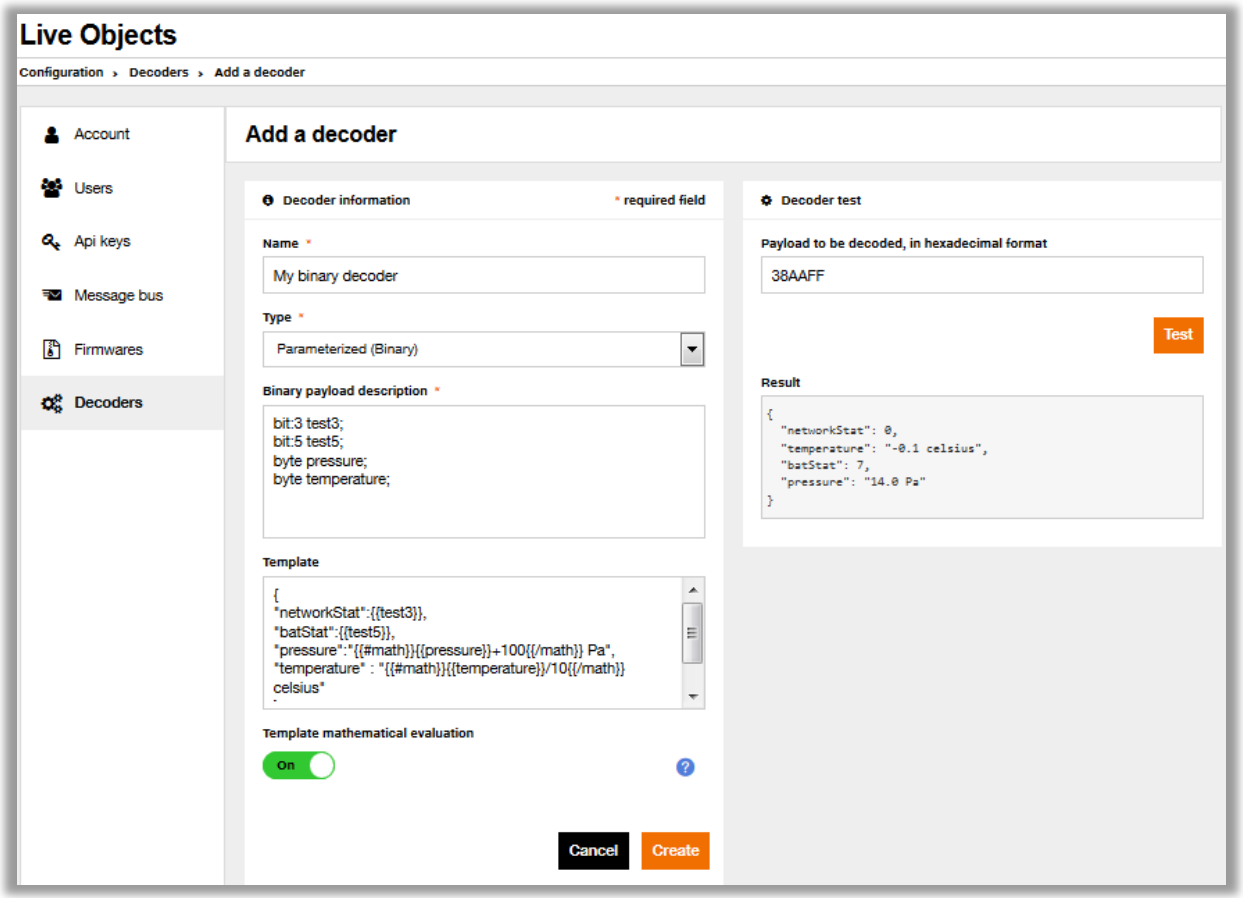
First byte: the three first bits are 000 that equals to 0 in decimal (value of « test3"), the following five are 00111 that equals to 7 in decimal (value of "test5"), so the entire byte 00111000 equals to **38 in hexadecimal**

Second byte: AA in hexa equals -86 in decimal (value of "pressure")

Third byte: FF in hexa equals -1 in decimal (value of "temperature")

It is possible to format the decoded data and to apply arithmetical operations (offset application, unit conversion ...). This is possible through the use of templates. The templates are based on the "mustache" formalism (see http://mustache.github.io/)

If we take the previous example we may want to give a particular meaning to the values represented by "test3" and "test5" and convert the pressure and temperature values and display them with the correct unit.



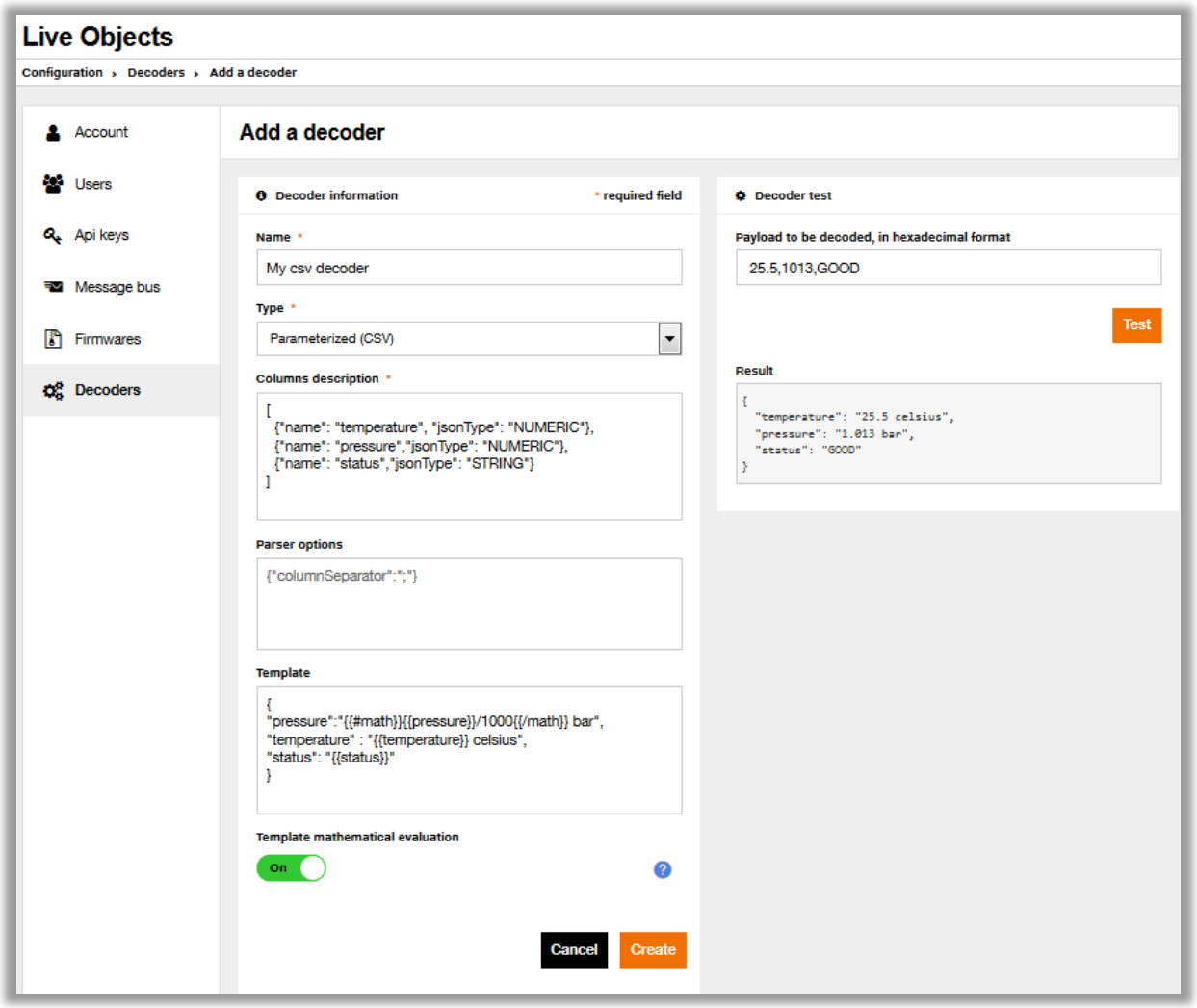
The selector "Template mathematical evaluation" allows to restrict the mathematical evaluation to the tags {{math}} only (value off), or to extend it to the whole template (value on)

Step 2: Creation of the decoder

Once the decoder has been validated, you can create it by clicking on the "Create" button. It becomes immediately available for use by your devices (see the paragraph "Assign a decoder to a device")

2.6.7.4. Create a CSV decoder (configurable)

To create a csv decoder, proceed in the same way as for a binary one. The structure of the payload is simply composed of "Name of the variable" (name) and "type" (jsonType), and a separator character. It must be entered in the "Columns descriptions" field.



Possible types are: NUMERIC (integer, long, float double), STRING (UTF-8), and BOOLEAN (“true”/ “false”)

The field « Parser options» allows specify different options of csv format. If left blank the default separator is the comma.

See the developer guide for the complete list of available formatting options https://liveobjects.orange-business.com/doc/html/lo_manual_v2.html#_csv_decoding

It is also possible to define a template to format decoded data, in the same way as for the binary decoder (see "Create a binary decoder")

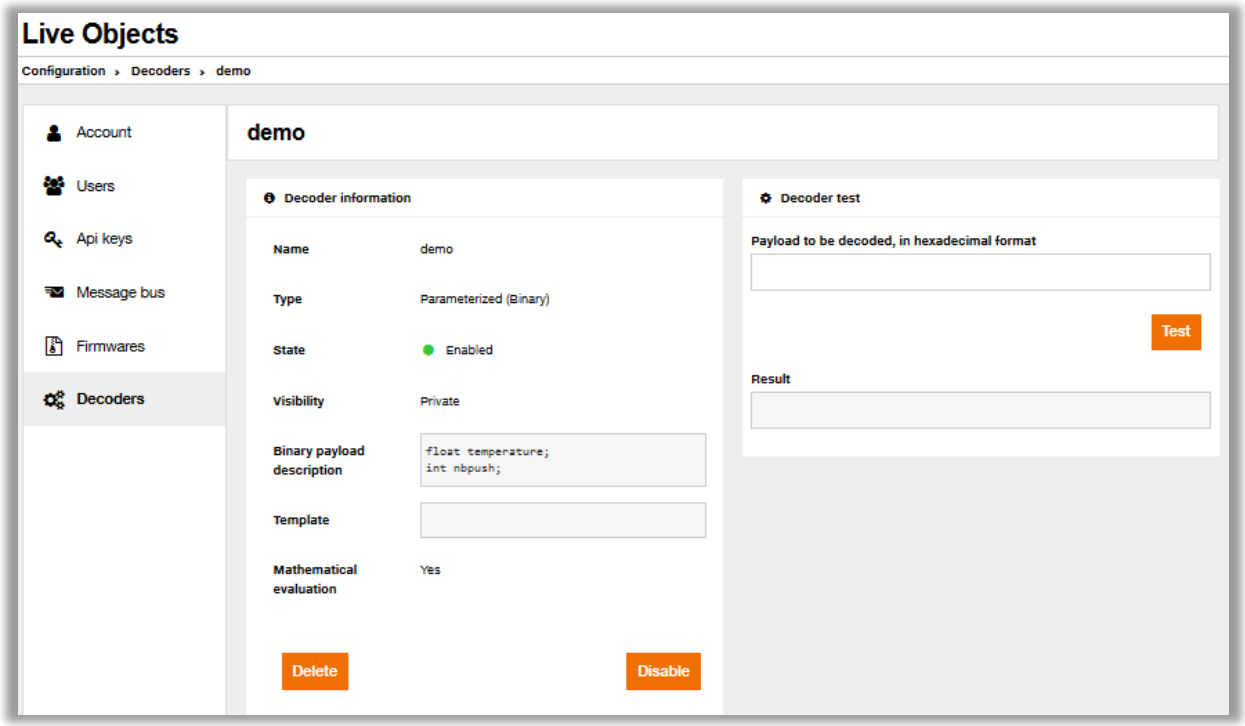
2.6.7.5. Create a programmable decoder

If your device produces payloads with complex and variable structures, you can use programmable decoders. The creation of this type of decoder requires action from Orange teams. If you cannot find a public decoder for your hardware, please ask to liveobjects.support@orange.com.

2.6.7.6. Disable/enable a decoder

When it is created, a decoder is automatically activated, but it is possible to deactivate it for temporarily suspend the decoding for all devices that uses it.

To disable/enable a decoder, select it in the main list, then click on the "Disable / Enable" button.



Note: You can disable only your own decoders.

2.6.7.7. Assign a decoder to a device

You can assign decoders only to LoRa or SMS devices.

You can assign a decoder to a LoRa device at its creation by selecting the decoder from the proposed list (see paragraph "Add a Lora Device"), or to an existing device (see paragraph "Modify a LoRa device"). NB: Once the decoder is associated with to the device the next payloads will be automatically decoded. The payloads raised before the association of the decoder remain not decoded.

You can assign a decoder to an SMS device at its creation (see paragraph "Adding an SMS device"). To add a decoder to an existing SMS device you must use the API (see paragraph "Modify an SMS device"); then decoding will work the same way as for LoRa devices.

3. Resources

A set of resources is available from your Live Objects portal: https://liveobjects.orange-business.com/#/faq

You will find:

- Practical answers to frequently asked questions (FAQ)
- Video tutorials giving you a quick start to the different features of Live Objects
- The complete development guide for Live Objects. This is the documentation for developers and integrators.
- A complete training kit for Live Objects
- The REST API documentation (swagger). This is an interactive documentation that allows you to manipulate the APIs with your Live Objects account
- Postman collections to facilitate use of APIs
- Code samples for devices (SDK) and for business applications (samples)
- Recommended tools for BI, analytics and dataviz that can be easily used with Live Objects
- An "About" section in which you can find the current version and its content (Release Note)

You can also join the Live Objects community on the developer forum https://stackoverflow.com/tags/orange-api