IoT Soft Box Starter Kit

Getting started with the Greenhouse example

orange™

Table of contents

1. Introduction

1.1. Document purpose

This document is a guide to IoT Soft Box SDK for ARM presenting the following:

- Overview
- Greenhouse

1.2. Reference documents

#	Origin	Title
1	Orange	Datavenue Live Objects - complete guide (1.4.1.)
2	ARM mbed	FRDM-K64F devlopment board
3	ARM mbed	mbed documentation

2. Overview

2.1. What is Live Objects?

Live Objects is one of the products belonging to Orange Datavenue service suite.

Live Objects is a software suite for IoT / M2M solution integrators offering a set of tools to facilitate the interconnection between **devices** or **connected « things »** and **business applications**.

The main features provided are:

- **Connectivity interfaces** (public and private) to collect data, send command or notification from/to IoT/M2M devices,
- **Device management** (supervision, configuration, ressources, firmware, etc.),
- **Message Routing** between devices and business applications,
- **Data Management**: Data Storage with Advanced Search features.

Read Datavenue Live Objects - complete guide to have a full description of architecture and services provided by Live Objects platform.

2.2. ARM mbed OS

ARM mbed OS is an open source embedded operating system designed specifically for the "things" in the Internet of Things.

It includes robust security foundations; standard based communication capabilities, drivers for sensors, I/O devices and connectivity. mbed OS is built as a modular, configurable software stack so it can be customized for the device used for development.
To find out more about the mbed OS:

- https://developer.mbed.org/blog/entry/Introducing-mbed-OS-5/
- https://developer.mbed.org/

Live Objects iotsoftbox-mqtt library is fully compatible with the ARM mbed platform.

2.3. IoT Soft Box

The Live Objects IoT Soft Box is a Software Development Kit (SDK) to help developers make easy usage of Live Objects platform.

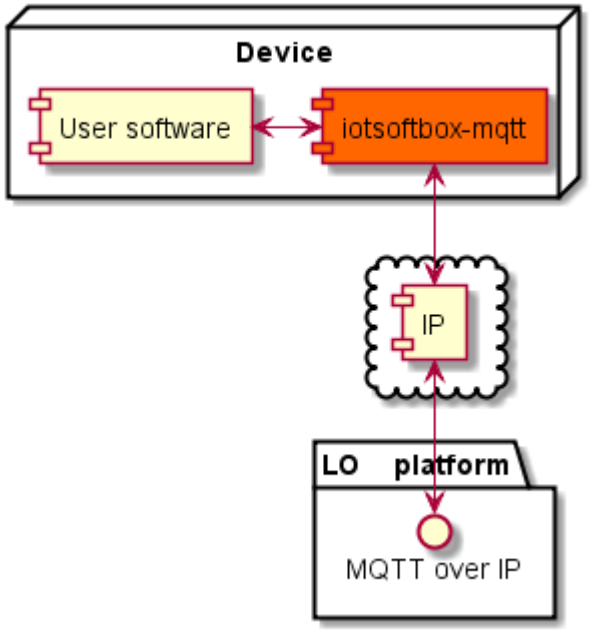


Figure 1 – IoT Soft Box integration in a system

The Live Objects platform is able to manage different formats (MQTT, HTTP …) and several low level protocols (SMS, IP …). The Live Objects IoT Soft Box is designed to work with MQTT over TCP w/o TLS.

The IoT Soft Box can run on devices connected to Internet through Ethernet, Wi-Fi, GPRS or any other IP connection.

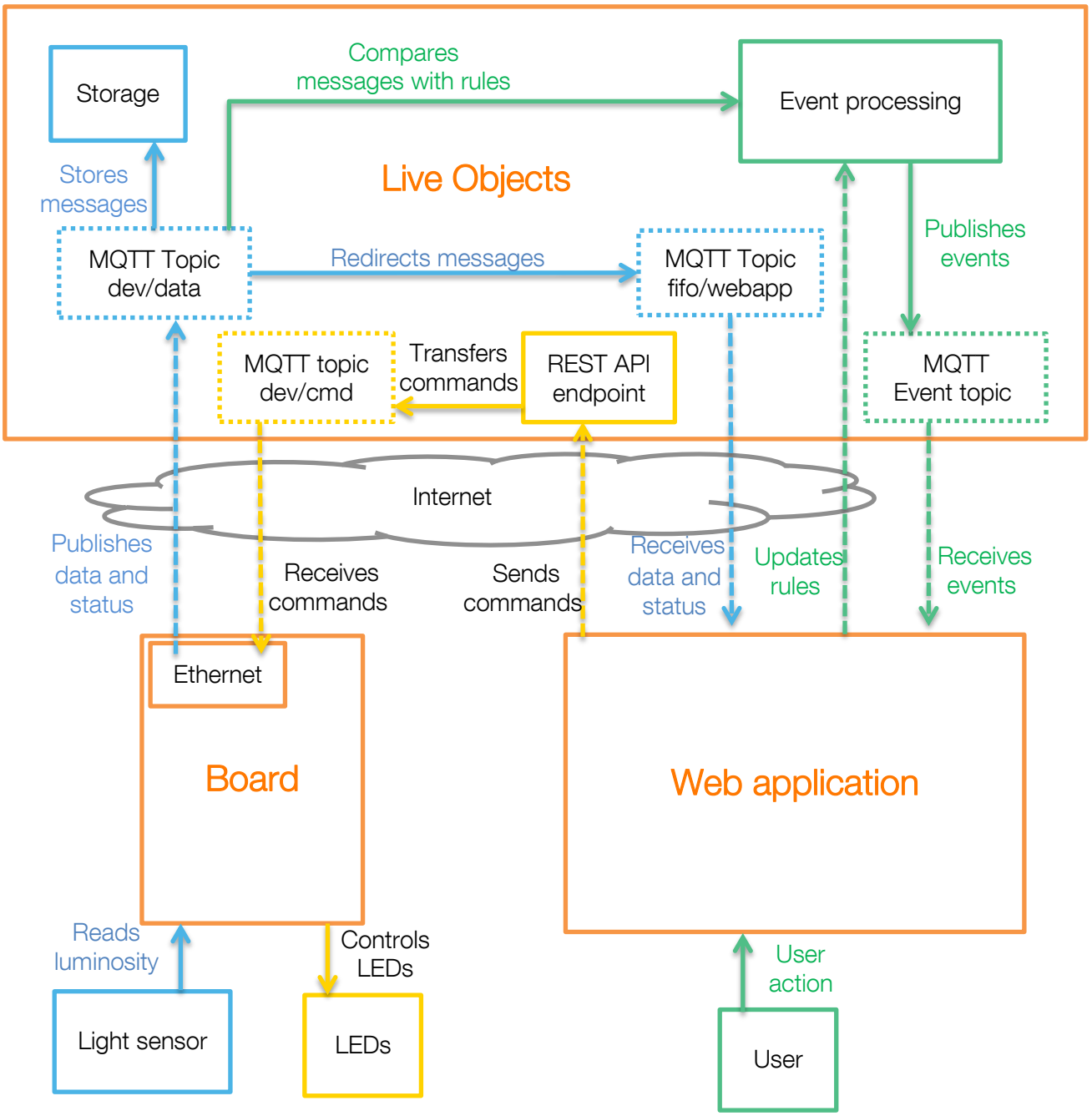
The library (iotsoftbox-mqtt) is linked to the following third-party existing libraries:

- Embedded MQTT C/C++ Client Libraries (eclipse paho). This library is available here.
- JSMN, a simple C library only used to parse the received JSON messages. The JSMN is available here.
- Mbed TLS is used to include cryptographic and SSL/TLS capabilities in embedded devices.

3. Greenhouse

3.1. Principles

The following diagram shows how the Greenhouse example works and how messages are transmited.



3.2. Hardware Environment

3.2.1. Components

To test our SDK with a compatible hardware, please use:

- Our IoT Soft Box hardware kit
- 4 LEDs
- RJ45 cable

or

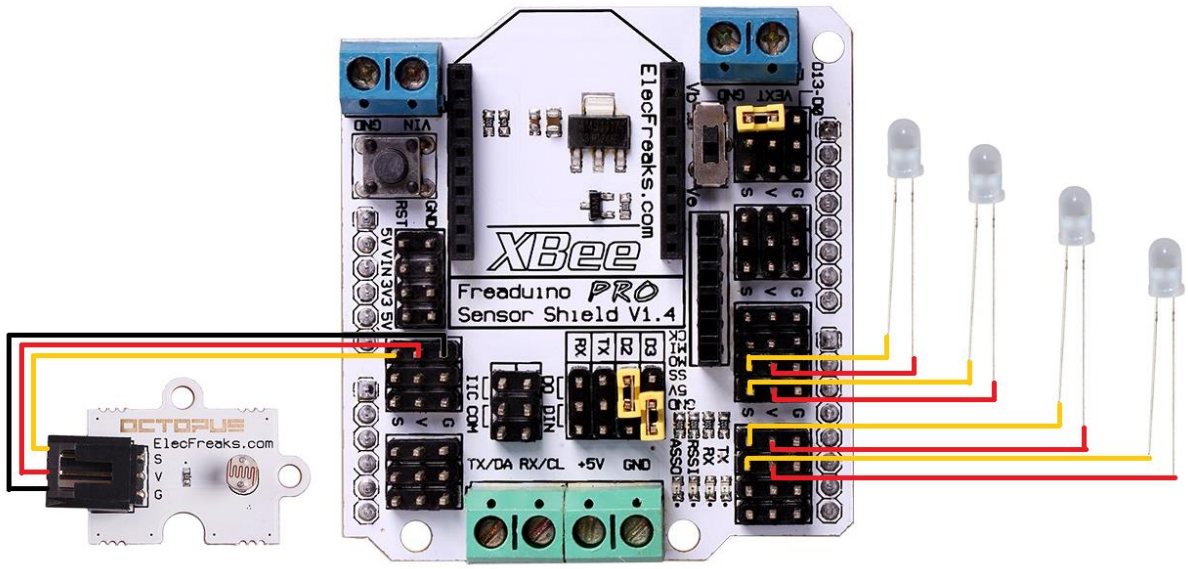
- The NXP development board : Freescale K64F
- Arduino shield
- 4 LEDs
- Analog light sensor
- RJ45 cable

3.2.2. Wiring

Plug the shield on the development board and wire the components as written in the table below.

Component	Arduino pin name	K64F pin name
Led 1	D5	PTA2
Led 2	D4	PTB23
Led 3	D3	PTA1
Led 4	D2	PTB9
Analog light sensor	A0	PTB2

Your connections should look like this:



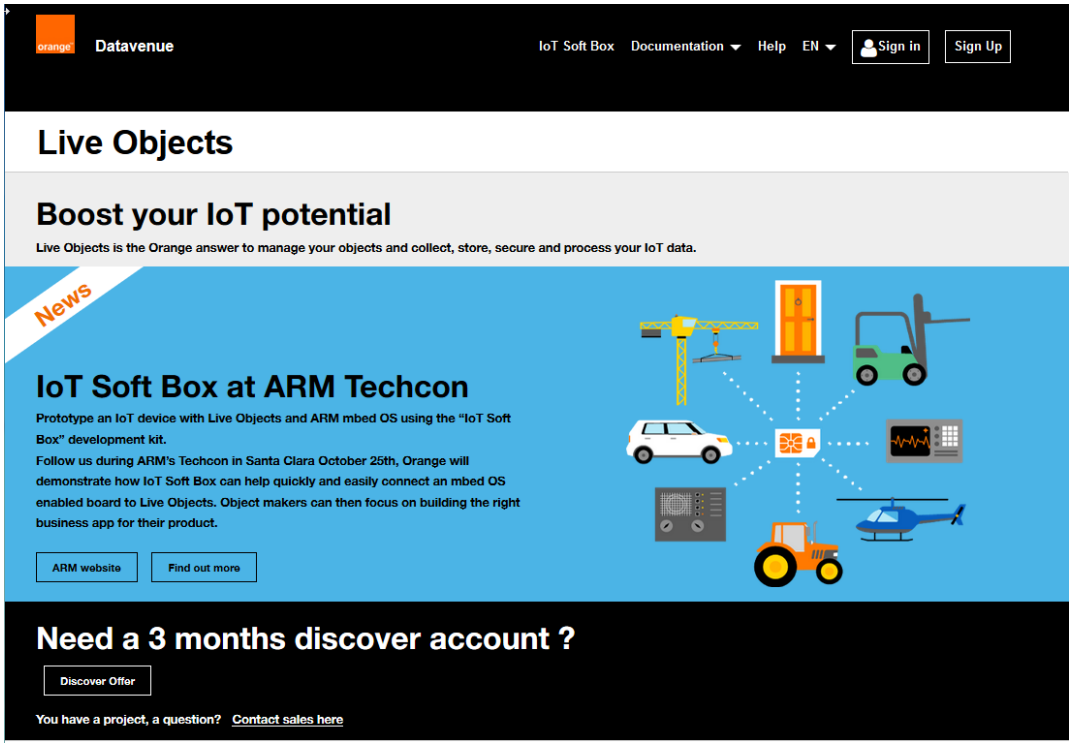
Depending on the caracteristics of your LEDs, you may have to add a resistor in series with each LED.

3.3. Access to Live Objects

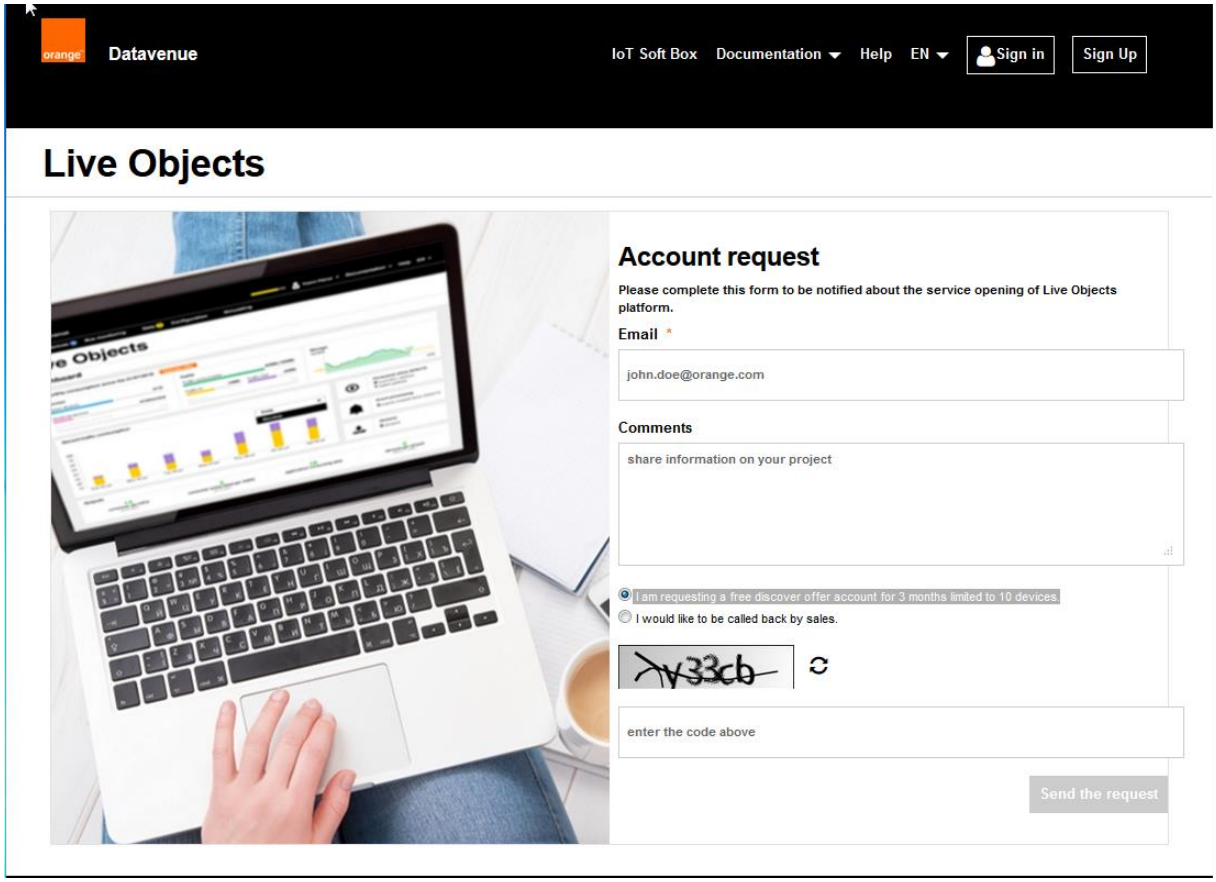
3.3.1. Account creation

In order to use Live Objects, you need to have a dedicated account on the service.

1. Go to Live Objects portal (https://liveobjects.orange-business.com/).



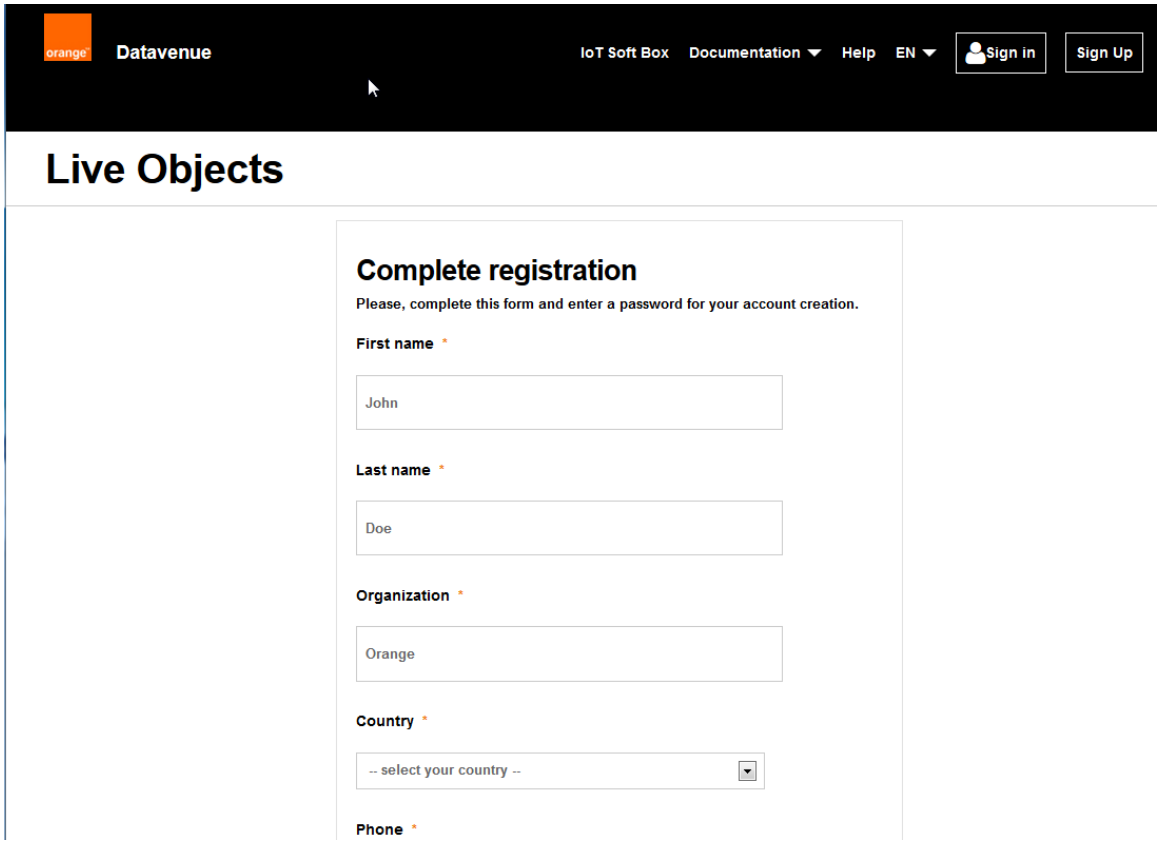
2. Click on '**Discover Offer**' button (or Sign Up) and fill the form, checking option 'I am requesting a free discover offer account for 3 months limited to 10 devices'.



3. Then you will receive an e-mail to activate your Live Objects account.

> Hello,
>
> You just subscribed to Live Objects portal.
>
> Please, use this link to activate your account 'john' and define your password.
>
> 'john' Account activation
>
> Thank you for your trust.
>
> Orange Business Service Customer Support.
>
> This is an automatically generated email, please do not reply.

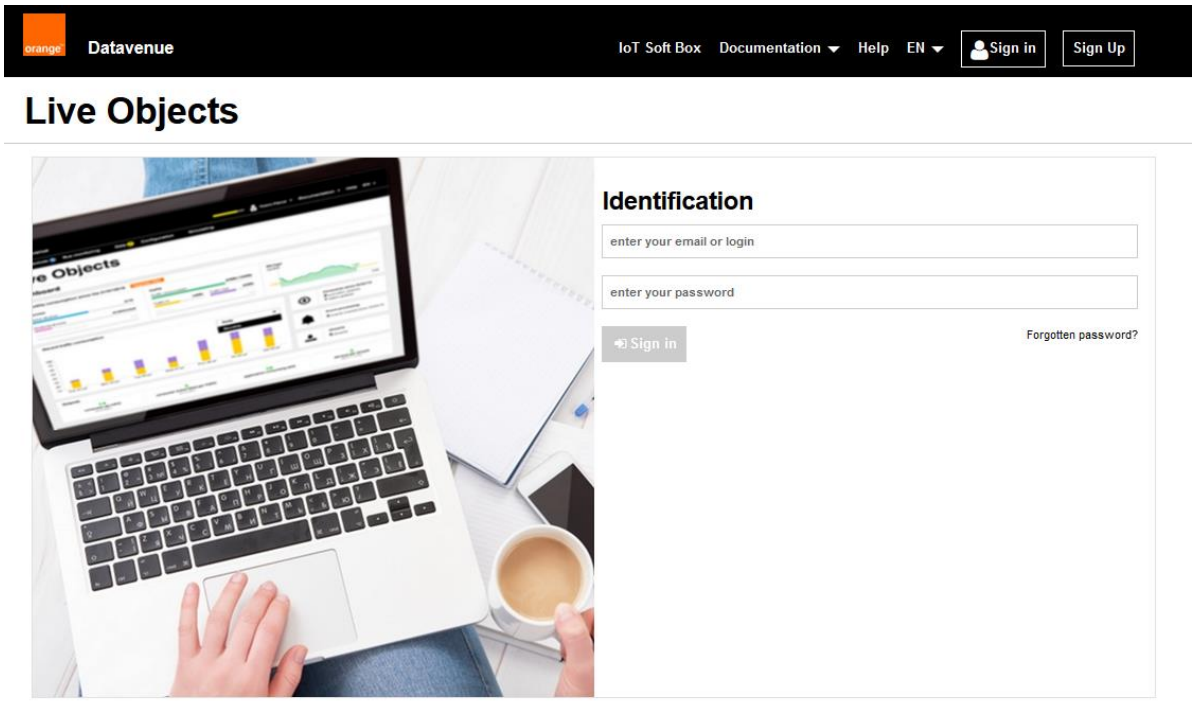
4. Follow the link, fill the form, and click on ‘Validate’.



5. Now, you can go back to Datavenue Live Objects portal and sign in. Once logged, select the 'configuration' tab to create a new API key.

3.3.2. Log in

To log in to Live Objects web portal, connect to liveobjects.orange-business.com using your web-browser:

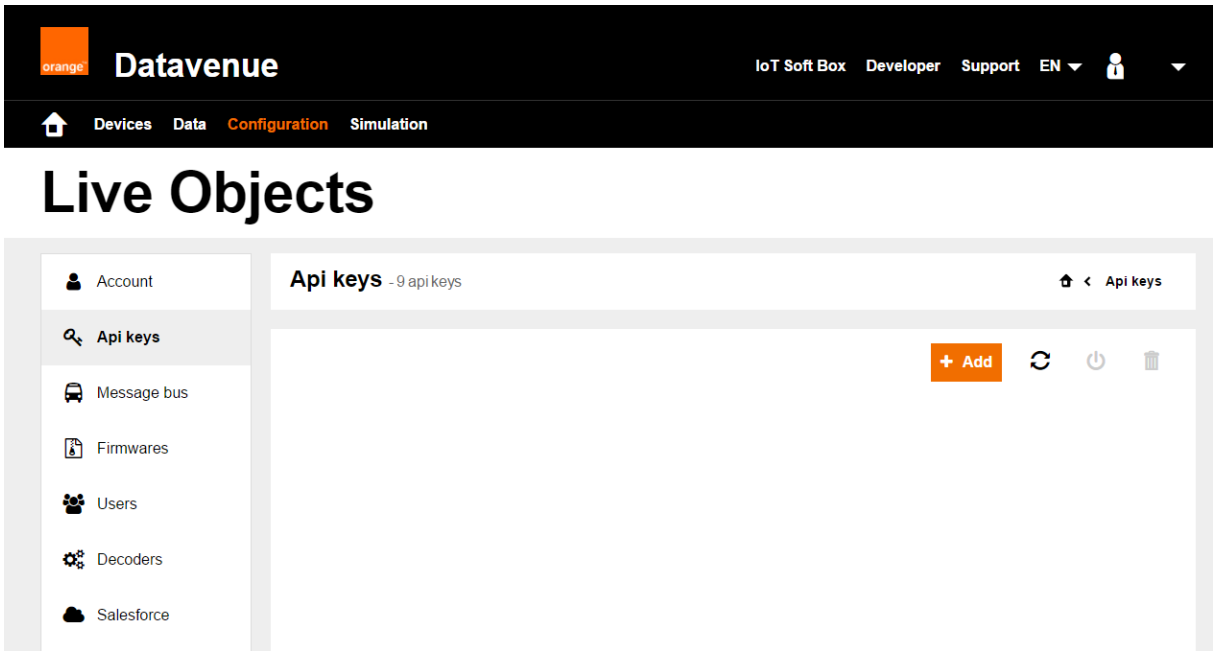


1. Fill the “Log in” form with your credentials:
    - your email address,
    - the password set during the activation phase,
2. Then click on the “Log in” button.

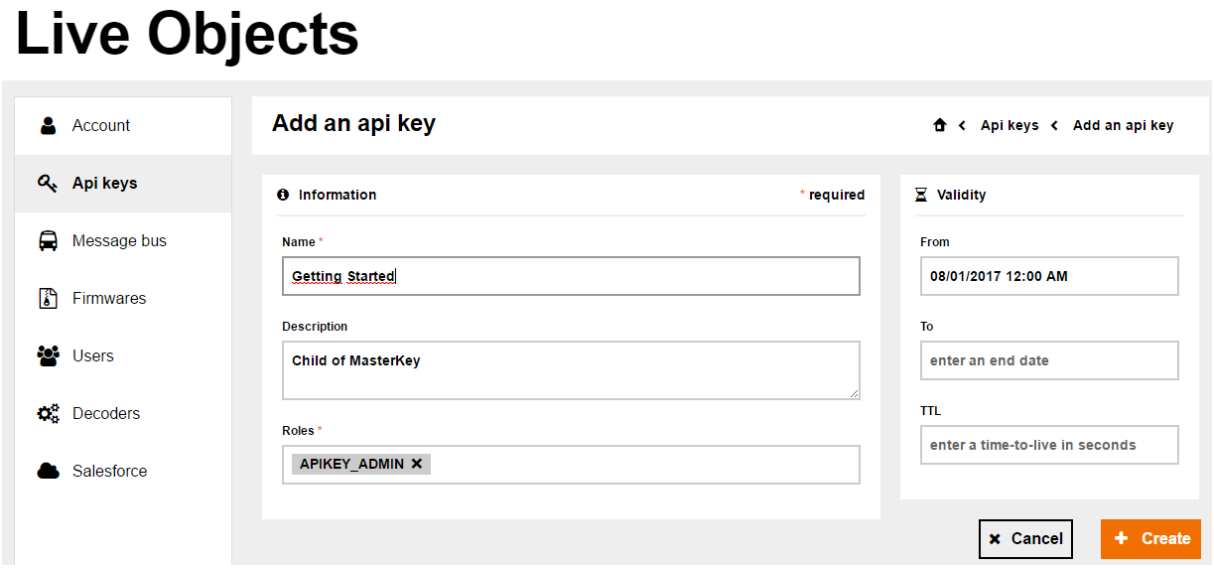
If the credentials are correct, a success message is displayed and you are redirected to your “home” page:

3.3.3. API Key creation

To get a device or an application communicating with Live Objects, you will need to create an API Key in the “Configuration” menu. On the left menu, click on "API keys" and create a new API key. This key will be necessary to set up a connection with the public interfaces (MQTT and REST) of Live Objects.



As a security measure, you cannot retrieve the API Key again after closing the API key creation results page. So, note it down to work with the MQTT client, during the scope of this getting started.



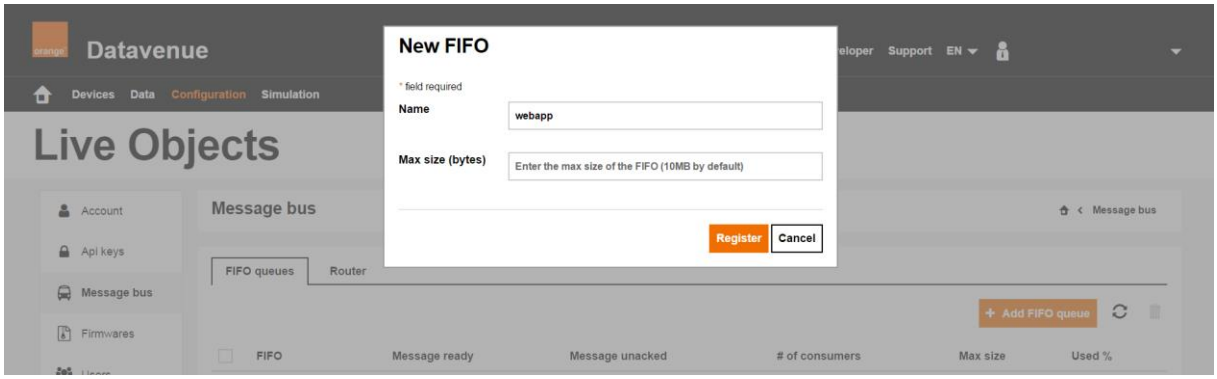
When you press create, the following windows will appear. It contains the api key as a text and a QR code. Don't forget to save the key because it won't appear again.

3.3.4. Route messages

To use Greenhouse, you will need to redirect messages from the board to the web application. The board will publish messages to the special "dev/data" topic in Live Objects platform and they will be stored by Live Objects.

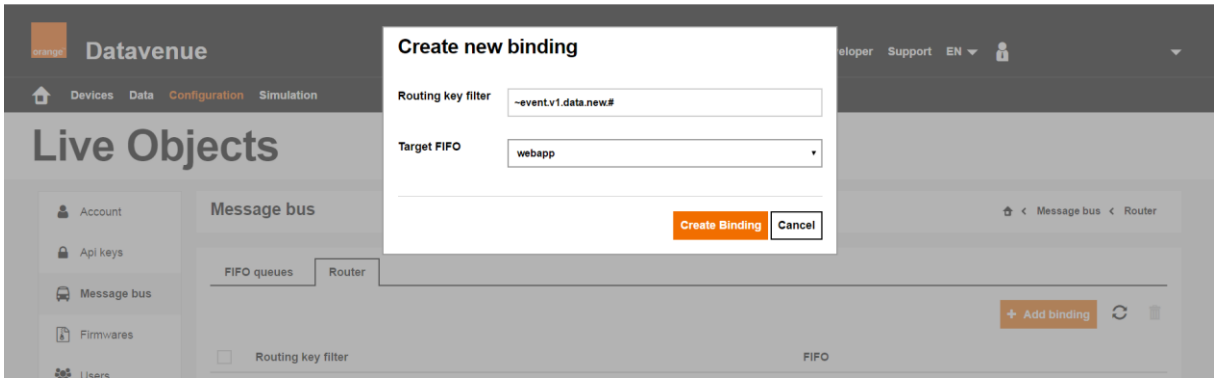
As the web application retrieves messages as they come, you need to create a route from the "dev/data" topic to another topic to which the web application can subscribe.

To do this, enter the "Configuration" menu and click on "Message bus" in the left menu. Then create a new queue by clicking on +Add FIFO queue. Enter "webapp" in the Name field and click on Register.



The FIFO queue you created is the topic to which the web application will subscribe in order to get messages.

To link the two topics, enter the "Configuration" menu and click on "Message bus" in the left menu. Then go to the "Router" tab, click +Add binding, enter "~event.v1.data.new.#" as the routing key filter, select "webapp" as the target FIFO and click on Create Binding.



Now all your incoming messages are redirected to the FIFO "webapp" and messages are available in the topic "fifo/webapp" until a subscriber read them.

3.4. Embedded program

3.4.1. Introduction

A good way to discover Live Objects features is to use our Live Objects IoT examples. The next part introduce the 'greenhouse' example.

The 'greenhouse' embedded application:

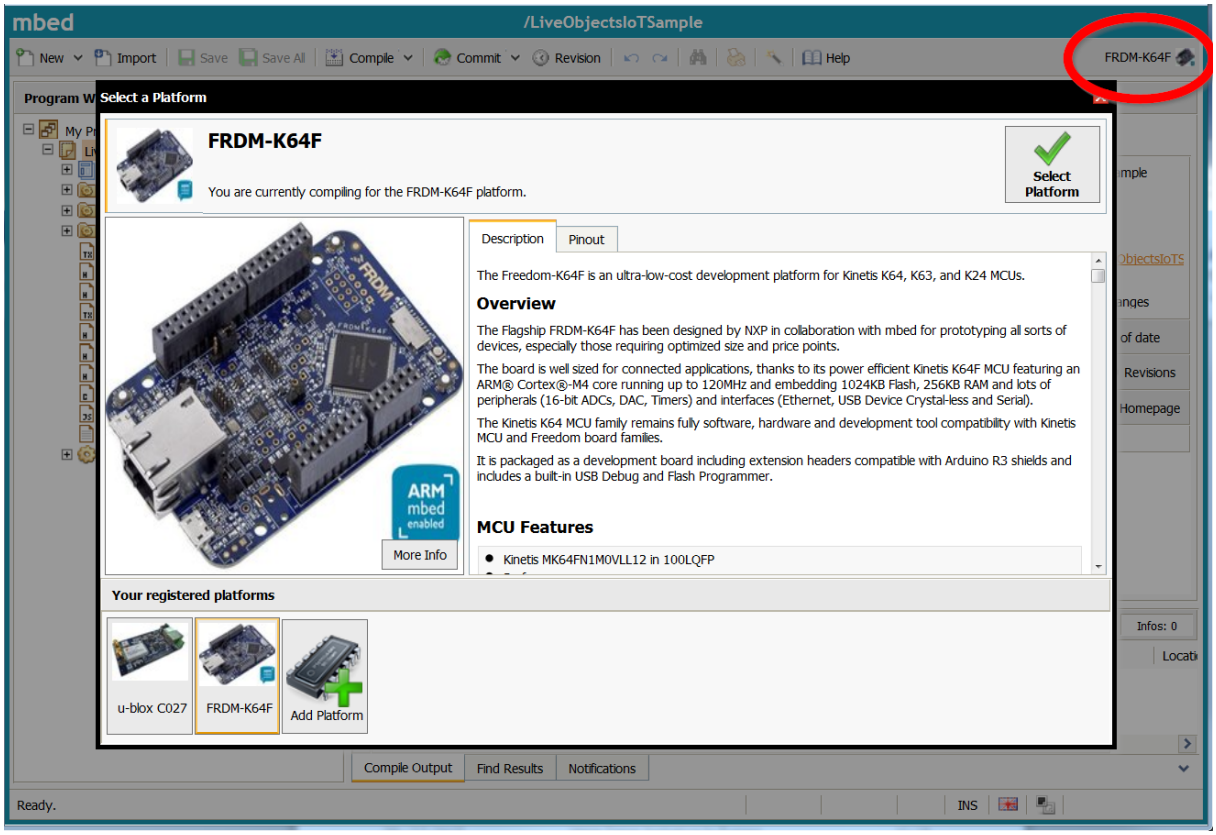
1. Connect to Datavenue Live Objects Platform, using:
    - an optional secure connection (TLS)
    - the LiveObjects mode: Json+Device
2. Publish
    - The current Status/Info
    - The current Configuration Parameters
    - The current Resources
3. Subscribe to Live Objects topics to receive notifications
    - Configuration Parameters update request
    - Resource update request
    - Command request
4. Then it:
    - Reads the temperature from the sensor and publish it every 2 seconds
    - Waits for events from Live Objects platform to:
        - Update configuration parameters
        - Process a command
    - Restart at step 2 if the connection is lost

3.4.2. Getting started with mbed online compiler

To help you use mbed online compiler, you can also visit the following page: getting started with blinky example..

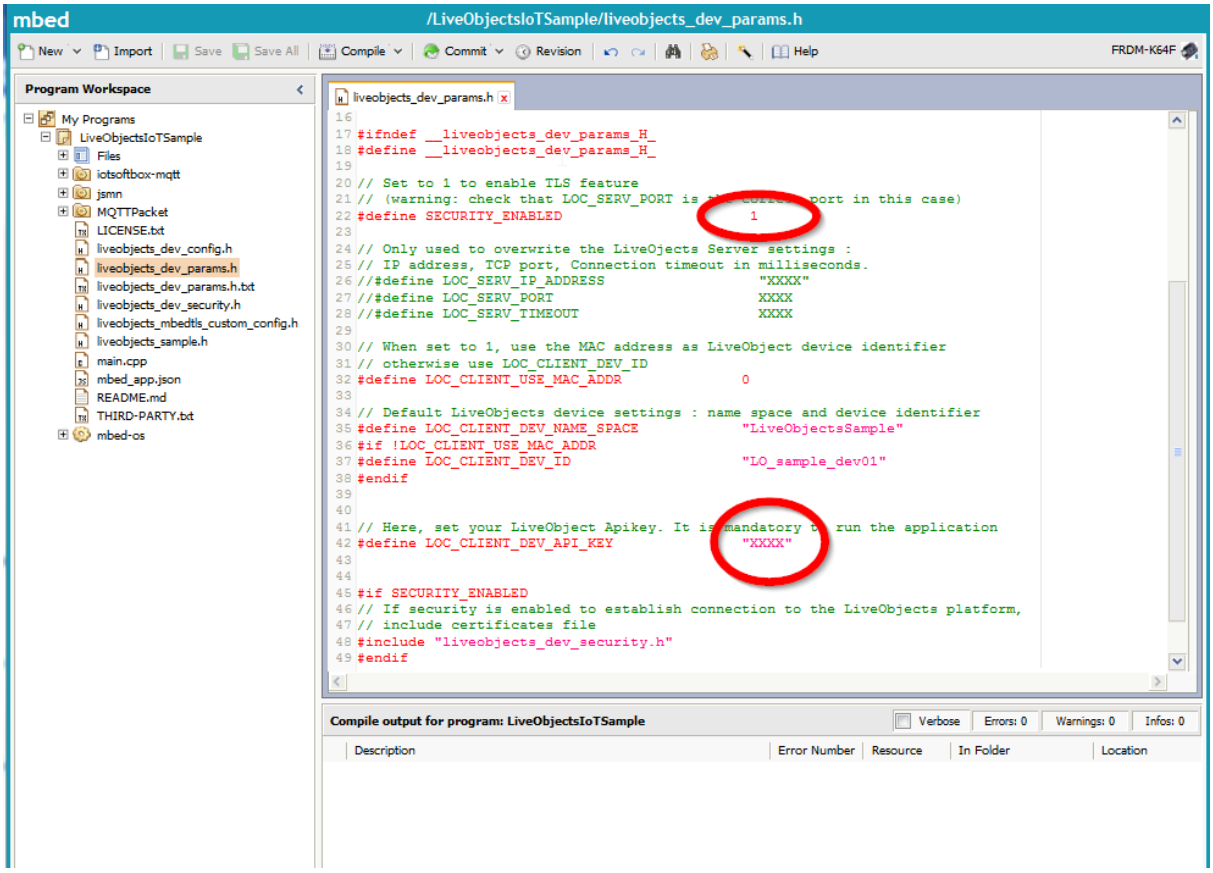
The procedure is the following:

- Join the mbed OS Developer community by creating your own developer account.
- Click on the top right corner to select the target board: FRDM-K64F



- Click on “Import” and type “liveobjects-iotsoftbox” in the programs’ search field. There are at least two main projects:
  - liveobjects-iotsoftbox-basic: A Live Objects ‘basic’ sample application using this iotsoftbox-mqtt library.
  - liveobjects-iotsoftbox-greenhouse: A Live Objects ‘greenhouse’ demonstrator using this iotsoftbox-mqtt library. The web ‘greenhouse’ application is located here.
- Import the ‘greenhouse’ application by double clicking on the program’s name

Edit the header file liveobjects_dev_params.h to set your Live Objects Tenant API key. And save the header file.



Your program is ready. For the record, SECURITY_ENABLED means that the connexion to LO will be secured. If it is set to 0, your data will not be encrypted. We recommend you to always leave the flag to 1.

3.4.3. Flash the board

To run the program on your board, the procedure is the following:

- Compile the application
- Download/Save the generated binary file liveobjects-iotsoftbox-greenhouse_K64F.bin on your computer
- Connect your K64F board to your computer with the USB cable. The mbed board should be shown as “mbed removable storage”. And an mbed serial port is up.

If you met any issue with mbed driver, especially on Windows system, see Windows serial configuration for full details about setting up Windows for serial communication with your mbed Microcontroller.

- Drag and drop the binary file to the board (via the ‘mbed’ storage)
- Start an terminal application on the mbed serial port (configuration is: 9600 baud, 1-bit stop) to see debug messages
- Connect your K64F board to your network (with a DHCP server)

(If firewall: outgoing TCP ports are 8883 or 1883).

- Push the reset button on your board

The application should be running. And you should see the green LED of your board turning on and off. It blinks quickly at the beginning, then it blinks slowlier when it is connected to Live Objects.

One of the first debug messages contains the MAC address of your board, note it as you will need it later. If you do not have a terminal application, you can also get this information in Live Objects platform, in the "Devices" tab, the MAC adresse of your board is its "Id".

3.5. Web application

3.5.1. Introduction

The web application:

- Connects to Live Objects in the json+bridge mode
- Subscribes to 2 topics:
  - Fifo/webapp to get data messages
  - Router/~event/v1/data/eventprocessing/fired to get event messages
- Then it:
  - Displays the sensor's data in a time graph
  - Waits for a user action:
    - It sends a command to the board tu control manually the LEDs
    - It creates/updates 2 matching rules and 2 firing rules so Live Objects generates events depending on these rules. For each event, it sends a command to the board to control automatically the LEDs

3.5.2. Configuration

You do not need any development environment to run the web application.

The sources are available on Github in the "greenhouse-web" repository:

https://github.com/Orange-OpenSource/LiveObjects-iotSoftbox-mqtt-mbed-examples

To configure the web application so that it can communicate with your board:

- Download the sources
- Open the configuration file: js/common.js
- Write your API key in the API_KEY variable
- Write your board's MAC address in the ASSET_ID variable

It does not require a server to run it. You may use it on your on PC.

3.6. Application Monitoring/Testing

There are several ways to monitor or/and to test the greenhouse application:

- Go to your Live Objects user account on Live Objects Portal
- Go to Live Objects Swagger User Interface.
- Serial Terminal is used by embedded sample application to print debug/trace messages.

3.6.1. From Live Objects Portal

From Live Objects:

- In the “Devices” tab, you can:
  - See your board
  - Check its status
  - Send commands
- In the “Data” tab, you can:
  - See the history of you board’s messages

3.6.2. From the web application

The web application is a dashboard in which you can:

- See the timeline of the luminosity in real time, the graph automatically refreshes every time there is a new message from the board
- See the number of the board’s LEDs that are currently switched on
- Send commands to the board to control the LEDs
- Configure rules depending on the luminosity to control automatically the LEDs