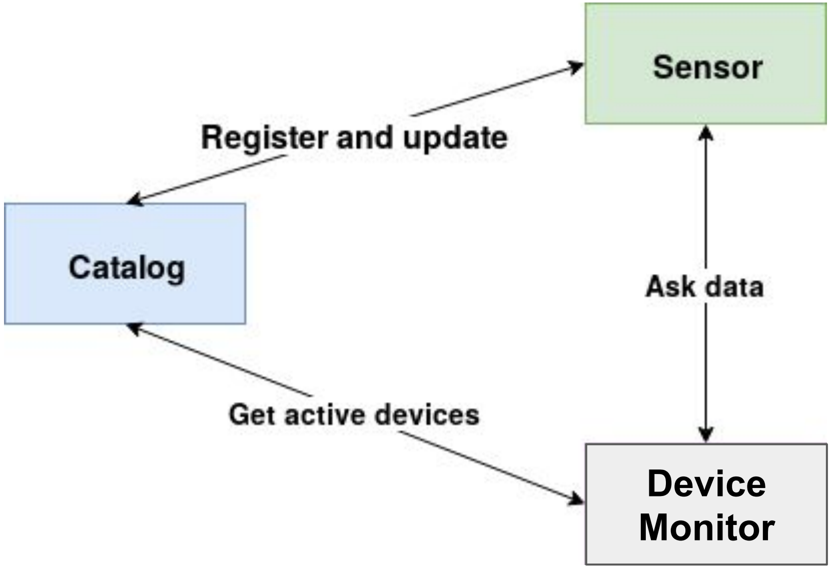
Thingspeak Adaptor

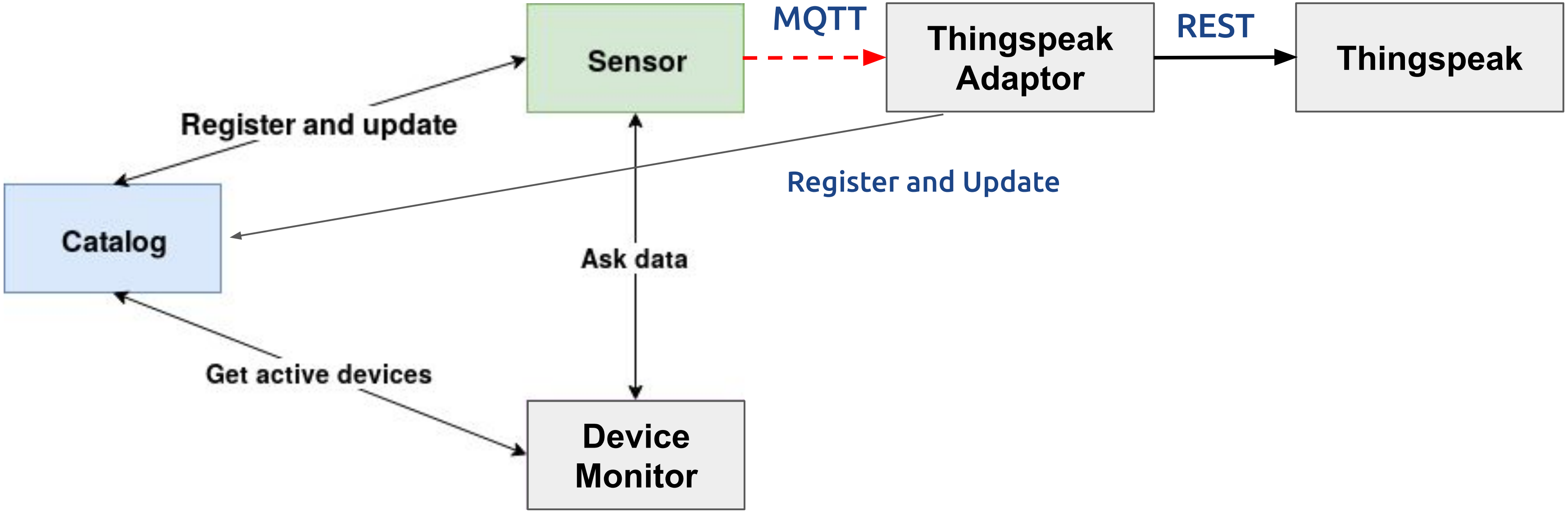
Example Simple Platform V2

Imagine a Platform with 3 actors: *Catalog*, *Sensor*, and *DeviceMonitor*. Each of them will run on its own container



Example Simple Platform V2

Imagine a Platform with 3 actors: *Catalog*, *Sensor*, and *DeviceMonitor*. Each of them will run on its own container



Example Simple Platform: *Sensor*

The *Sensor* is a simple REST client and MQTT publisher for a temperature and humidity sensor. When the *sensor* is launched, it will send a **POST** request to the *Catalog* to register itself, stating which are its settings (IP address, port, accepted methods). Moreover, it will send a **PUT** request periodically (e.g. 1 minute) to the *Catalog* , to let the *Catalog* know that it is alive, and to keep it updated.

What is more, the *Sensor* will publish the measurements periodically every 1 minute.

The settings of *Sensor* are stored in a `settings.json` file.

Example Simple Platform: *Catalog*

The *Catalog* is another REST client. It’s job is to keep and update the list of the available **devices** and **services** (with their settings). Moreover, it may provide this information to other entities that may need them. For example, the *DeviceMonitor* will retrieve the information from the Devices. Everytime the *Catalog* receives a request from a *Sensor* it will add it to the list of the **devices** and will store the timestamp of that request. This list is periodically controlled by the *DeviceMonitor* to check if the last timestamp of each of this devices respects a threshold, if the timestamp is too ”old” the device will be removed from the list. The settings are stored in a file called `settings.json`.

Example Simple Platform: *DeviceMonitor*

The service ***DeviceMonitor*** it’s a simple script to monitor the status of the ***Devices*** . It’s responsible for managing the status of the devices registered in the ***Catalog***.

Example Simple Platform: *TS Adaptor*

The ***Thingspeak Adaptor*** (TS Adaptor) will be an MQTT subscriber of the Sensor measurements, and will upload the information to ***Thingspeak***.

What is Thingspeak?

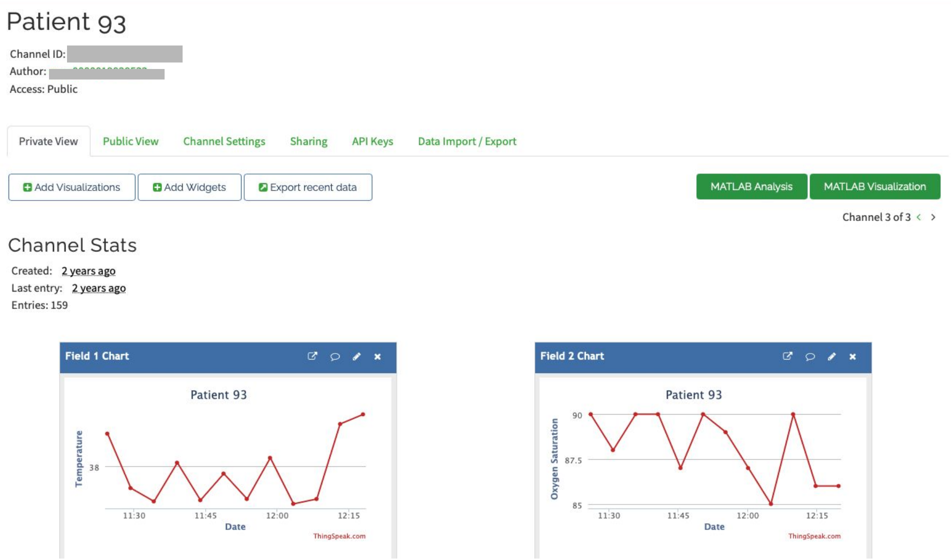
Thingspeak is a MathWORK open-source platform that stores, post-processes and retrieves data. It is possible to send and retrieve data from it in multiple ways. We are going to use Thingspeak as a ***database*** to store our time-series data. You can find more info here

Data in Thingspeak is stored in Channels. You can find more information on how to create Channel, collect and write data to a new Channel here. Inside each Channel, you have to define different Fields.

The first thing to do, is to register to Thingspeak using your `@student.polito` account to have the educational license.

Thingspeak: First steps

Once you are logged in, you can create a new Channel by clicking in the New Channel button. For example, if your project is managing a Smart Home, each Channel can be a room (or a house) and the Fields the different measurements (e.g. temperature, humidity, etc).



Patient 93

Channel ID: [redacted]
Author: [redacted]
Access: Public

Channel Stats

Created: 2 years ago
Last entry: 2 years ago
Entries: 159

Thingspeak: REST APIs

As previously mentioned, there are different ways to send data to channels (MQTT, REST and even from MatLAB functions). Due to previous experiences, it is suggested to “interact” with the Thingspeak platform using REST APIs. Among the possible action, it is possible to create, update and delete channel, write data to them, retrieve their data, etc. You can find more information in the REST API reference.

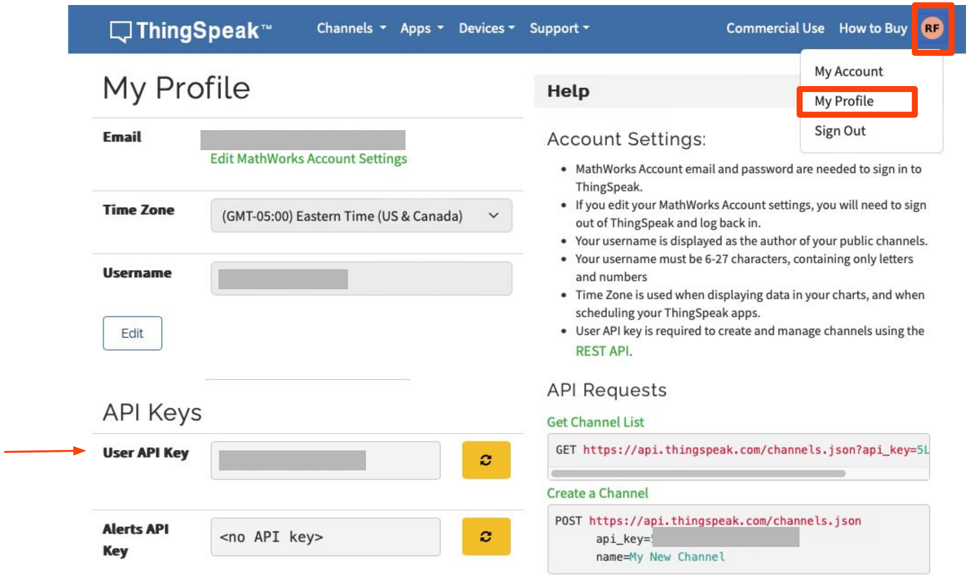
To interact with the Thingspeak REST APIs, there are two main API keys that are required:

- USER API KEY
- Channel API KEY

Thingspeak: USER API KEY

The User API key may be necessary for actions like creating, deleting or editing Channels. This API key is unique for a particular Thingspeak User.

To obtain the User API key, you should go to “My Profile” (as shown in the picture).



My Profile

Email [redacted]
Edit MathWorks Account Settings

Time Zone (GMT-05:00) Eastern Time (US & Canada)

Username [redacted]

Edit

API Keys

User API Key [redacted]

Alerts API Key <no API key>

Help

Account Settings:

- MathWorks Account email and password are needed to sign in to ThingSpeak.
- If you edit your MathWorks Account settings, you will need to sign out of ThingSpeak and log back in.
- Your username is displayed as the author of your public channels.
- Your username must be 6-27 characters, containing only letters and numbers
- Time Zone is used when displaying data in your charts, and when scheduling your ThingSpeak apps.
- User API key is required to create and manage channels using the REST API.

API Requests

Get Channel List

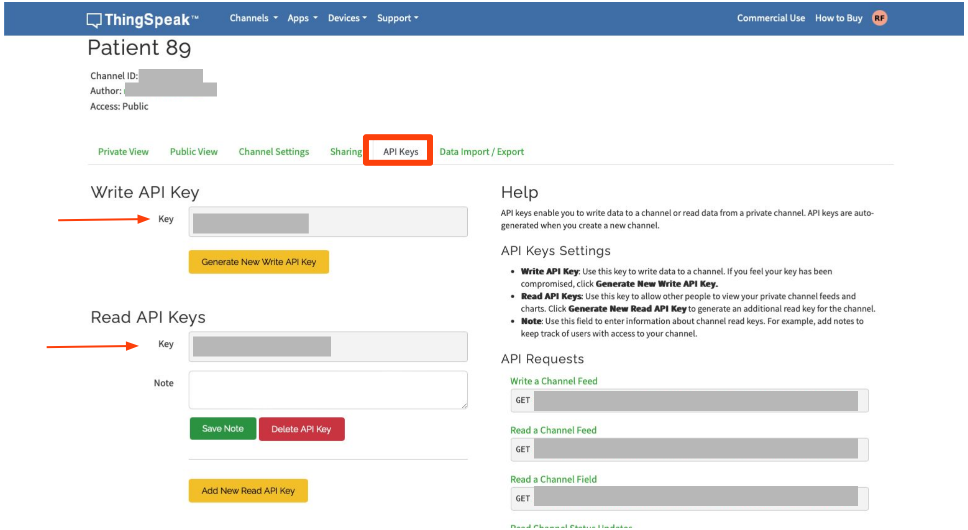
```
GET https://api.thingspeak.com/channels.json?api_key=5L
```

Create a Channel

```
POST https://api.thingspeak.com/channels.json
     api_key=[redacted]
     name=My New Channel
```

Thingspeak: Channel API key

For actions like sending data to a particular channel, or retrieve data from it, the particular Channel API key is required. The Channel API key is specific for each channel. To obtain this API key, once you are inside the channel of interest, click on the “API keys” tab as shown in the figure below



Patient 89

Channel ID: [redacted]
Author: [redacted]
Access: Public

Write API Key

Key [redacted]

Read API Keys

Key [redacted]

Note

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key**: Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys**: Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note**: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

GET [redacted]

Read a Channel Feed

GET [redacted]

Read a Channel Field

GET [redacted]

Read Channel Status Updates

Hint: You can read the Channel API key by doing a GET request to the following link (replacing channel_id with the corresponding channel, and API key with your USER API key): https://api.thingspeak.com/channels/<channel_ID>.json?api_key=<XXXXXXXXXXXXXXXX>

Thingspeak Adaptor

The Thingspeak Adaptor will be our microservice responsible for listening to the sensors measurements (MQTT subscriber), and will upload this measurements to Thingspeak using the REST API keys. You can find an example of Thingspeak Adaptor in `Thingspeak_Adaptor.py`