Programming for IoT - Lab 5

Rafael N. Fontana

1 Exercise 1

Develop a RESTful style Catalog for a general purpose distributed platform. Identify the most suitable HTTP methods (among GET, POST, PUT and DELETE) and develop the web services to:

1. Retrieve information about IP address and port of the message broker in the platform
2. Add a new device with the following information
    - unique deviceID
    - end-points (i.e. Rest Web Services and/or MQTT topics)
    - available resources (e.g. Temperature, Humidity and Motion sensor)
    - "insert-timestamp" when this device was added

    (**SUGGESTION:** to avoid synchronization issues, this attribute is managed and updated only by the Catalog according to its system clock)
3. Update the information of a device
4. Retrieve all the registered devices
5. Retrieve a specific device with a deviceID
6. Register a new user with the following information
    - unique userID
    - name
    - surname
    - email address(es)
    - telegram chat id
7. Retrieve all the registered users
8. Retrieve a specific user with a certain userID

This information is stored in a JSON file and all the information among the actors in the platform must be exchanged in JSON format.

Implement an additional feature of the Catalog to remove all the devices with "insert-timestamp" higher than two minutes. The Catalog has to take this action periodically (for example every 1 minute).

2 Exercise 2

Modify one of the sensors developed in the previous lab (lab 4 exercises 1 and 2) in order to be able to invoke the RESTful Catalog developed in Exercise 1 periodically (for example every 1 minute) to:

- either registers as new device or
- refresh the old registration by updating theirs "insert-timestamp".

NOTE: *During the refresh of an old device registration, the Catalog has to update also the "insert-timestamp".*

3 Exercise 3

Modify Exercise 3 of Lab 4 in order to make it able to perform REST requests to the Catalog developed in the first exercise. When this application starts, it **MUST** connect to the Catalog to obtain the information needed to retrieve the measurements from the sensor (i.e. IP address or MQTT topic). Moreover, the application **MUST** periodically send a request to the Catalog to update its registration information.

4 Exercise 4

This exercise could be quite useful for your final project, feel free to customize it according to the purpose of your project. The general challenges that needs to be assessed are:

- *having multiple users for your project*
- *having multiple sensors for each user*
- *each user must be able to see his own information*

Using the application developed for the previous exercises, try to run a simple simulation like the one described below:

You have developed a system that allows your clients to monitor the temperature in their houses. Moreover, the system enables the clients to switch on and off the lights. In your system, there are 2 houses, with one user each. Each one of these houses has an LED and a temperature sensor. In your system, there is also a control application that notifies each user when the temperature values are outside a common threshold. Each user must be able to obtain his own notification of the temperatures and must be able to see and set the status of his light.

4.1 Bonus improvements

- each user has its own custom threshold for the temperatures
- each user has more than one temperature sensor and light