

Programming for IoT - Lab 5

Matteo Orlando

1 Exercise 1

Develop a RESTful style Catalog of a distributed platform for general purpose services. 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)

1. Update the information of a device
2. Retrieve all the registered devices
3. Retrieve a specific device with a deviceId
4. Register a new user with the following information
 - unique userID
 - name
 - surname
 - email address(es)

5. Retrieve all the registered users
6. 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

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 sensor 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 to periodically (for example every 1 minute) :

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

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

3 Exercise 3

Develop a control application to monitor if the temperature values, coming from a sensor like the one used in the exercise before, is above or below a given threshold. In case the monitored value is outside the given threshold the application needs to perform one or more of the following actions:

- store the event details in a local file that is accessible through an HTTP GET request
- send an HTTP POST request with the event's details to some kind of service (you can fake the actual request but the general structure should make this possible with small future updates)
- send an MQTT message with the details of the event

When this application starts, it **MUST** connect to the catalog in order to obtain the information needed to retrieve the measurements from the sensor (IP address or MQTT topic) and periodically contact the catalog to update the registration

4 Exercise 4

This exercise could be quite useful for your final project, feel free to customize it according to the purpose of our project. The general challenges that need 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 in all the exercises before try to deploy a simple simulation like the one described below

You have developed a system that allows your clients to monitor the temperature in their houses and to switch on and off the light. In your system there are 2 houses with one user each. Each one of these houses has a LED and a temperature sensor. In your system there is also a control application that logs for each user the temperature values outside a common threshold. Each user must be able to obtain his own log 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