

Team 7

Homework 3

Esposito Giuseppe
s302179

Fanigliulo Sofia
s300751

Pagano Francesco
s299266

1 Exercise 1

Since we are trying to emulate the behavior of various IoT devices that send data about their functioning status to a server, the choices fall upon a REST architecture or a MQTT message protocol for the communication between devices and the general receiver. REST (REpresentational State Transfer) is a web application architecture which enables to work with a variety of data, like media, objects or documents over HTTP; it allows easy scalability and dynamism in development time as well as modifiability of interacting components at runtime and easy portability. MQTT, on the other hand, is a network messaging protocol which allows communication between devices over TCP/IP and follows a publisher-subscriber model and requires the employment of a Broker to correctly retrieve and redirect messages. Despite us dealing with Deepnote, our purpose is to simulate the behavior of various IoT devices communicating with each other, sending and receiving information about battery level and power status constantly; with this perspective in mind, one has to focus on power optimization and efficient communication between parts, and MQTT is the optimal choice to accomplish this: with its protocol header of just 2 *bytes* and its easy implementation, MQTT allows for a lower usage of the device's battery in order to implement effective and continuous communication, which, eventually, results in less maintenance and lowered costs.

2 Exercise 2

In order to develop this API, the implemented strategy was to store information about each device in redis's timeseries with the following format: *mac_address:battery* and *mac_address:power_plugged*. In order to manage such information, the following endpoints were developed:

- **/devices** Retrieves the list of MAC addresses of monitored devices.
- **/device/mac_address** Retrieves and manages battery status information (battery level and power plugged status) about the device with the specified MAC address.

Method	Endpoint	Ref n°
GET	/devices	(1)
GET	/device/mac_address	(2)
DELETE	/device/mac_address	(3)

1. The endpoint **/devices** implements a *GET* method in order to retrieve information about MAC addresses of monitored devices contained inside the corresponding Redis time series, in JSON format.
2. The endpoint **/device/mac_address** also implements a *GET* method in order to retrieve information about the battery level and power plugged status within a specified time-range (stated in the query) of the device stated in the call.
3. In order to delete information about a specific MAC address, the endpoint **/device/mac_address** also implements a *DELETE* method which tells the Redis database to delete timeseries corresponding to the MAC address stated in the call.