POLITECNICO DI MILANO

DEPARTMENT OF ELECTRONICS, INFORMATION AND BIOINGINEERING

Course:

Internet of Things

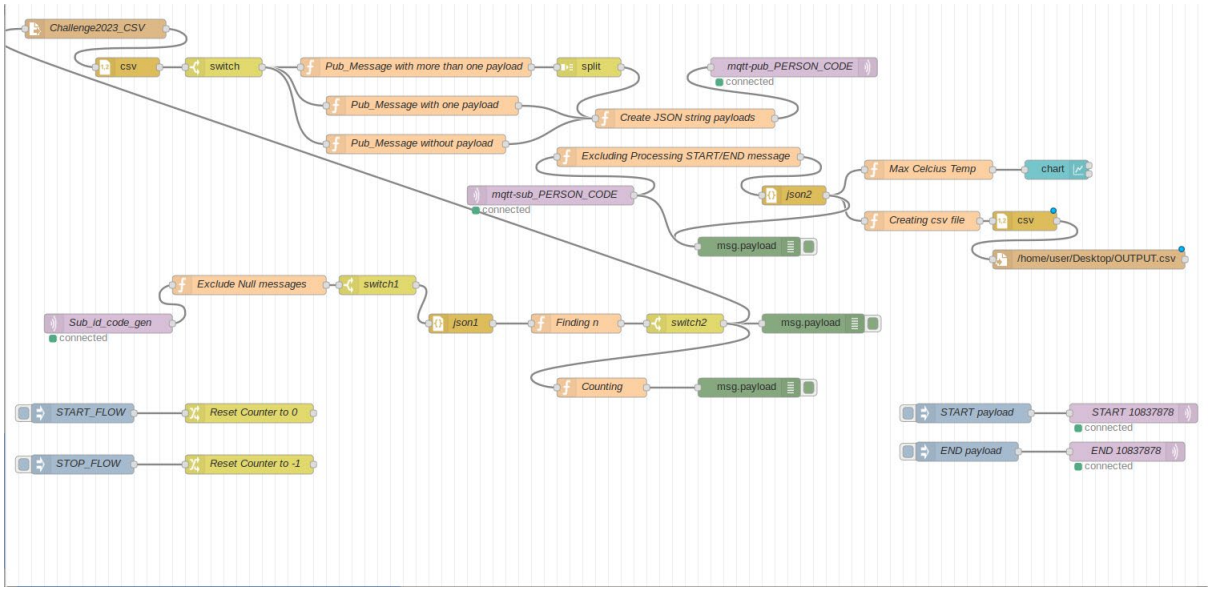
Challenge 2:

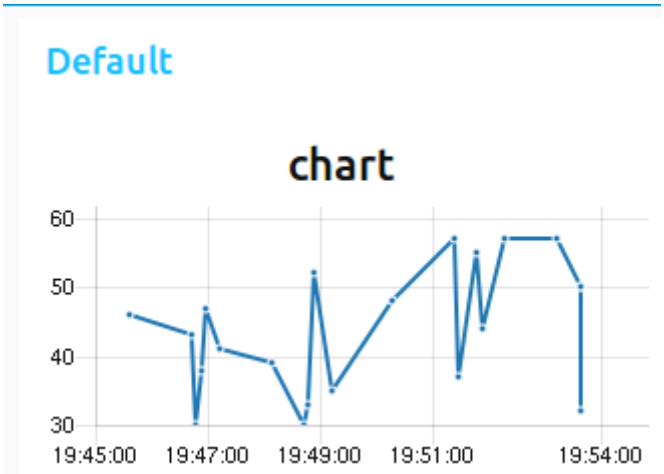
Node-Red

Professor. Dr. REDONDI

Tutor: Dr. Antonio Boiano

ACADEMIC YEAR 2022-2023





1. Sub_id_code_gen: this node is a mqtt-in node used to subscribe and receive messages with id which the client sends every 5 seconds. (Generally, mqtt-in node can be used for receiving messages from the topic that we specified in the node properties also we can specify the QoS which by default is set to 2).
2. switch1: a switch node is used to prevent subscribe messages from proceeding at the beginning of our Flow. (So, if the global counter isn't equal to -1, the flow can proceed.). (in general, this node can be used to route messages base on the properties and constraints we define in the node property).
3. STOP_FLOW: this node provokes the "Reset Counter to -1" node that sets the global counter to -1 and with the combination of "switch1" node, they stop receiving messages from id_code_generator topic. (That means first, we set the global counter value to -1 by using "STOP_FLOW" node and because the value of the global variable is persistent, we "deploy" our flow and now we are ready to press the "START_FLOW" node to begin receiving messages). (in general, this inject node can be used to inject a

message to the flow. In the node property section, we can define if we want to inject at intervals or even the kind of payload we want to inject).

4. Reset counter to -1: is a switch node that changes the value of the global counter to -1. (generally, “change” node can be used to set, change, move or delete properties of a message for example).
5. START_FLOW: this node just triggers the “Reset Counter to 0” node by sending a timestamp.
6. Reset Counter to 0: Change the value of the global counter to 0.
7. Exclude Null messages: is just a simple function that excludes the null messages and messages that are malformed (by checking if the messages has “id” key) we might receive from id_code_generator topic. (a JavaScript function can be used to filter and block the messages goes through it).
8. Json1: this json node converts JSON strings we receive from id_code_generator topic to JavaScript object (So, in the next node we can easily access the value of “id” key inside the messages). (in general, Json node converts between JavaScript objects and JSON string in either direction)
9. Finding n: is a function that accumulates “n” from ids.
10. Switch2: this switch stops routing messages when the global counter counts 100 messages.
11. Counting: is a function node that defines the global counter. (That returns "Counter is= " message to the debug node which is useful for keep tracking of the counter value in real time).
12. Challenge2023_CSV: a file-in node is used to read the csv file, then we used a csv node to convert the csv formatted strings to JavaScript object so we can perform filtering on those values using JavaScript in functions.
13. Pub_Message with more than one payload: this node is a function node that checks if there is messages that contains “publish message”, and then it checks if the col10 is available and if it has },{ means it contains more than 1 publish messages in that field, so we replace all of }, symbols with }@ symbols.
14. Split: a split node is used to split those messages that contain more than 1 payload in their col10, to separated ones base on where is @ symbol. (In general, split node splits a message payload in the sequence of messages base on the type that can be String, array or object).
15. Pub_Message with one payload: this function only routes messages that have one payload in their col10.
16. Pub_Message without payload: this function only routes messages that have col10, but it is empty.
17. Create JSON string payloads: this is one of the main functions in our flow. It construct messages with timestamp,id and payload (which we find from 3 previous parallel functions).
18. mqtt-pub_PERSON_CODE: this is a mqtt-out node used to publish messages that previously we formatted as timestamp,id,payload to topic /polimi/iot2023/challenge2/10837878. (In general, with mqtt-out node we can

connect to the broker and publish messages on a topic. Also, we can specify QoS for publish messages or specify username, password if needed.)

19. START payload: this injection node publishes a message with payload START to the next node.
20. START 10837878: is a mqtt publish node that sends the previous START payload to the topic /polimi/iot2023/challenge2/10837878. (We do it before starting to process 100 messages).
21. END payload and END 10837878: these two nodes do the same thing, and we use them at the end when processing 100 messages is finished.)
22. mqtt-sub_PERSON_CODE: this node is used to subscribe to topic with PERSON_CODE which is /polimi/iot2023/challenge2/10837878.
23. Excluding Processing START/END message: it excludes processing "START payload" and "END payload" nodes when they are triggered.
24. json2: it converts the JSON string messages received from the "mqtt-sub_PERSON_CODE" into JavaScript objects. (So later, we can easily access the value of "unit" and "range" keys).
25. Max Celsius Temp: this function only filters the messages with "unit" key equal to Celsius and then it calculates the max value in the "Range" (which is an array of 2 values).
26. Chart: the numbers received from the previous node are plotted as a chart. (Chart can be used to plot input values on a chart. We can select different charts like bar chart or pie chart from property window).
27. Creating csv file: in this function we filter messages that have Celsius in their payload. Then we pass it to csv node to write strings on the output node.
28. Debug node: "Our best friend in Node-Red!". By default it displays each msg.payload that goes to it as input. It is very useful to observe what is the output payload of the previous node. By default, it shows debug window, but you can change its property to also show system console or node status.

Thank you for your time.