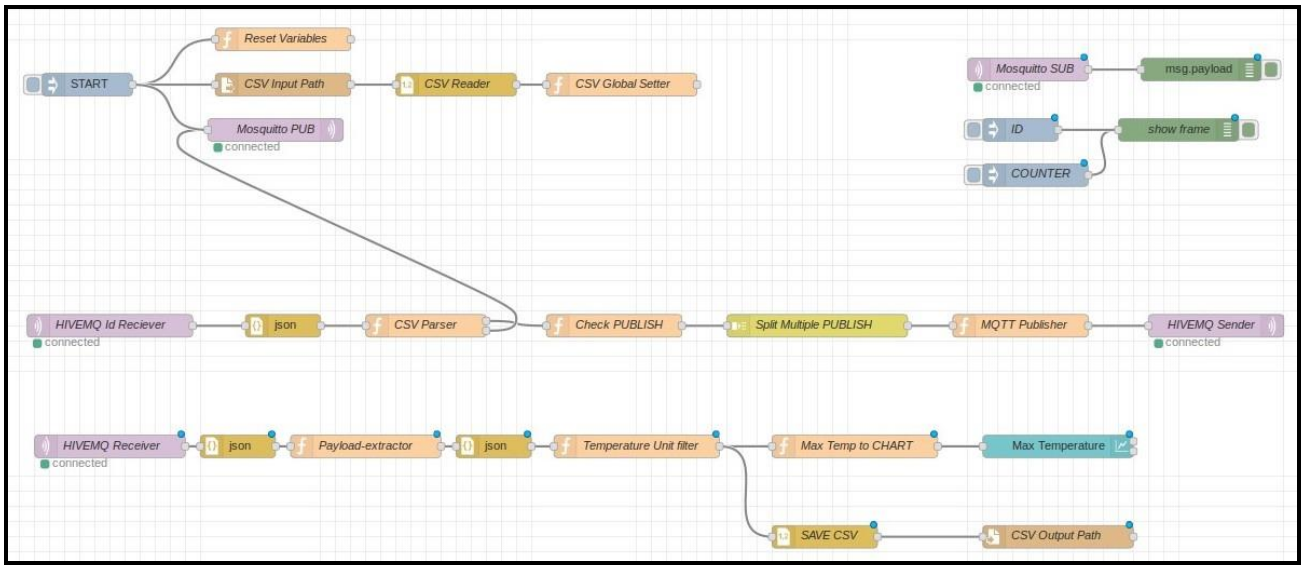
IoT 2023 CHALLENGE 2

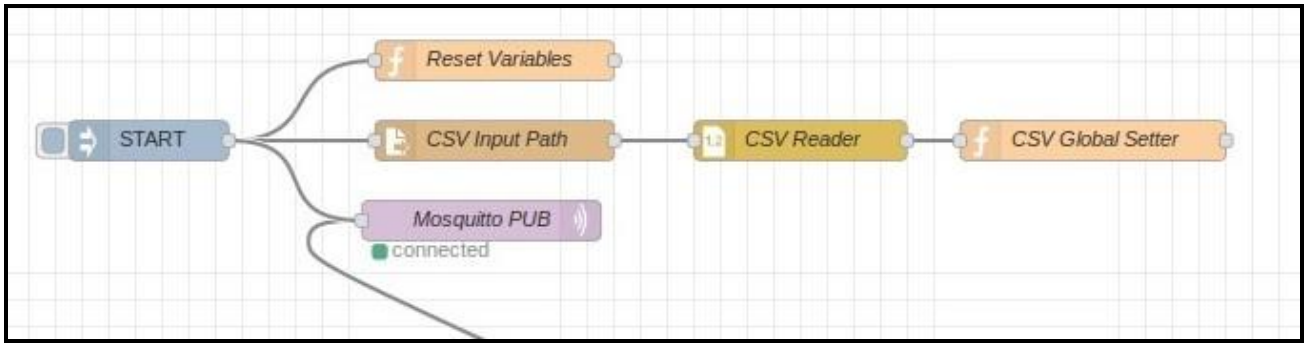
(1) Name: Nicolò Pisanu Person Code: 10827469

(2) Name: Marco Viviani Person Code: 10528650

Global flow:



1) Set-up section flow:

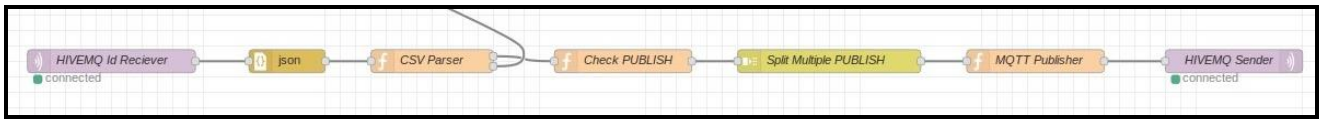


- **START:** inject node that triggers several actions:
  - It publishes a “START” message at topic “/polimi/iot2023/challenge2/10528650” on broker “test.mosquitto.org” (**“Mosquitto PUB”**);

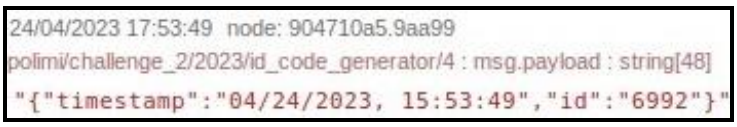
    24/04/2023 17:53:40 node: 78051e14.efbe38
    /polimi/iot2023/challenge2/10528650 : msg.payload : string[5]
    "START"

  - Calls the **“Reset Variables”** function: it sets a Boolean “ON” variable to start and stop the execution flow, a variable “counter” in order to count the ID messages in input from the broker “polimi/challenge_2/2023/id_code_generator” and a variable “id” to store their IDs;
  - Loads into a flow-variable the “challenge2023_2.csv” file as a string array.

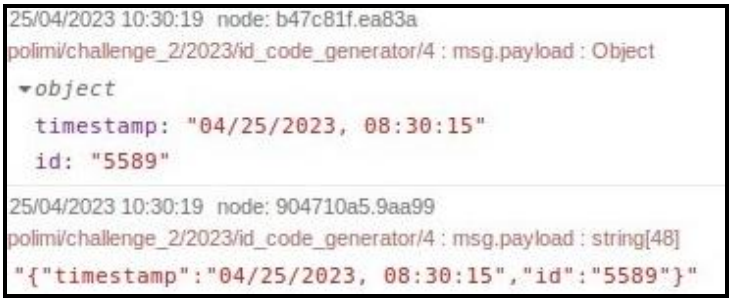
2) Publisher filter flow



- **HIVEMQ ID Receiver:** every 5 seconds ID MQTT Messages are received through the broker "broker.hivemq.com" at the topic "polimi/challenge_2/2023/id_code_generator" and are of type:



- **Json:** node that converts a JSON string in its JavaScript object representation. We used it often to easily access message payload fields;

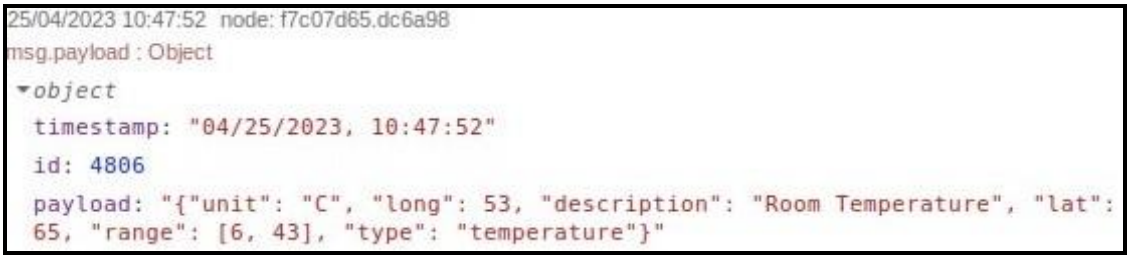


- **CSV Parser:** this node performs several important tasks. It extracts the message ID (also incrementing a counter) from which it calculates the number of the packet it will extract from the CSV file. This, however, after checking that the flow is active (through the boolean flow-variable "ON") and that the counter has not exceeded the 100 messages requested in reception;

- **Check PUBLISH:** checks that the incoming packet contains at least one Publish Message and if it has payload. If packet contains multiple Publish messages, this node extracts and sends them as an array of payloads (managed by the following **"split"** node). If some of the payload do not appear, it sends an empty payload ("") ;

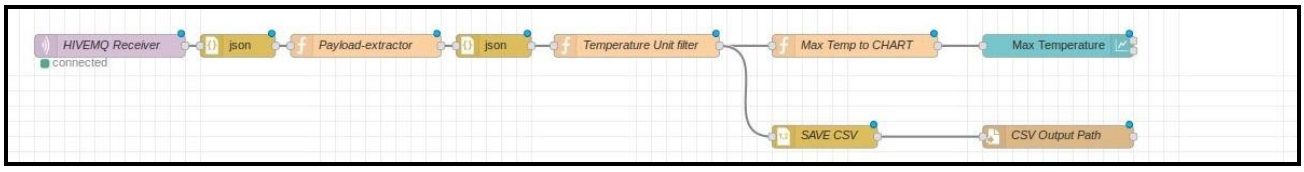
- **Split Multiple PUBLISH:** splits the incoming array of payloads into a sequence of messages;

- **MQTT Publisher:** this function takes care of adding timestamp and ID fields to the payload of the input message. The ID is retrieved by the ID flow-variable setted in the **"CSV Parser"** node;

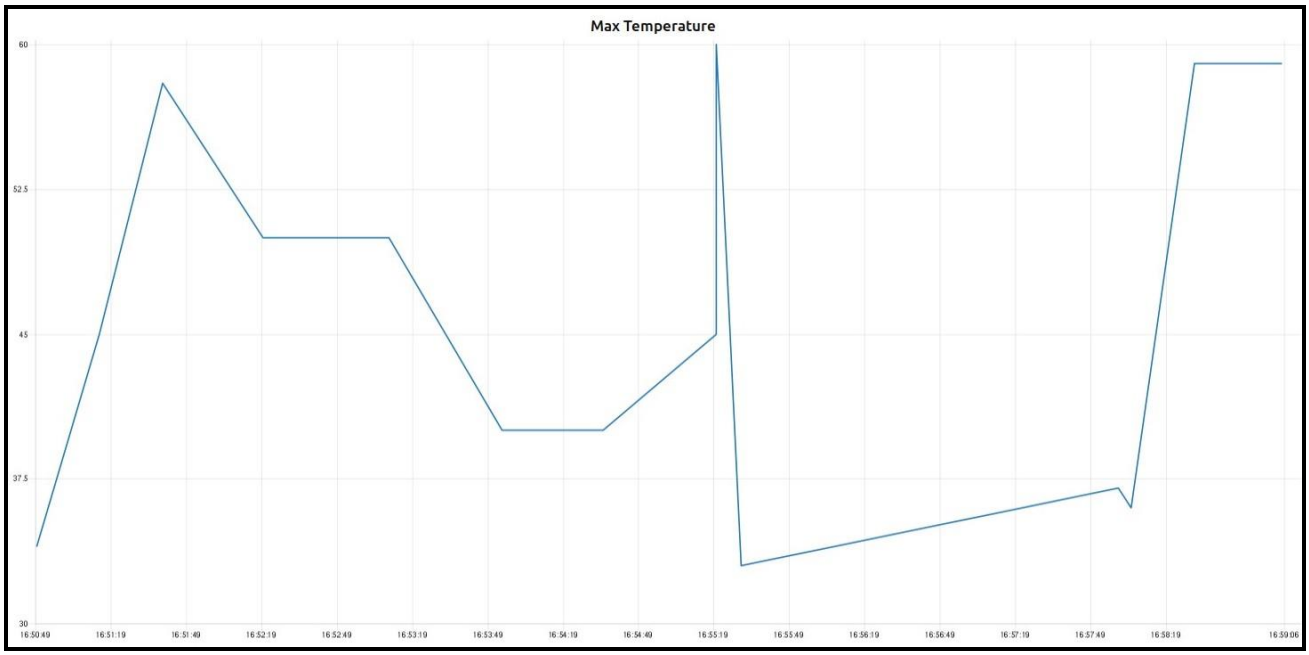


- **HIVEMQ Sender:** MQTT Publisher node that publishes messages through the server "broker.hivemq.com" at the topic "/polimi/iot2023/challenge2/10528650";

3) Temperature filter flow



- **HIVEMQ Receiver:** MQTT subscribed at the topic "/polimi/iot2023/challenge2/10528650" through the server "broker.hivemq.com";
- **Payload-extractor:** this node extracts the payload of one single Publish message if it has payload and it is not empty;
- **Temperature Unit filter:** filters only the messages with temperature in Celsius unit of measure;
- **Max Temp to CHART:** this node extracts the maximum temperature from the second position of the "range" array and sends it to the "Max Temperature" chart;
- **Max Temperature:** plots the maximum temperature input value in a time-temperature chart shown in the DASHBOARD;
- **SAVE CSV / CSV Output Path:** saves the payload string of such filtered messages in CSV where each column represents a payload field.



4) Helper flow:

- Allows us to monitor ID and counter values at will and Subscriber node flow that receives "START" and "END" messages;