

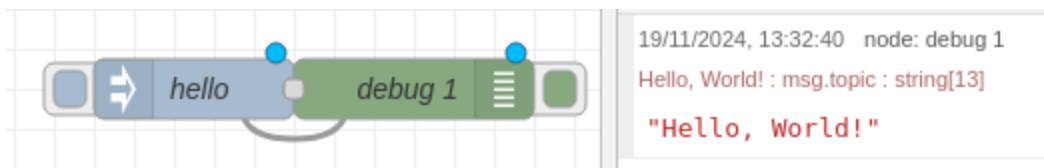
Node-Red

28th November

1 Hello World

How I did it:

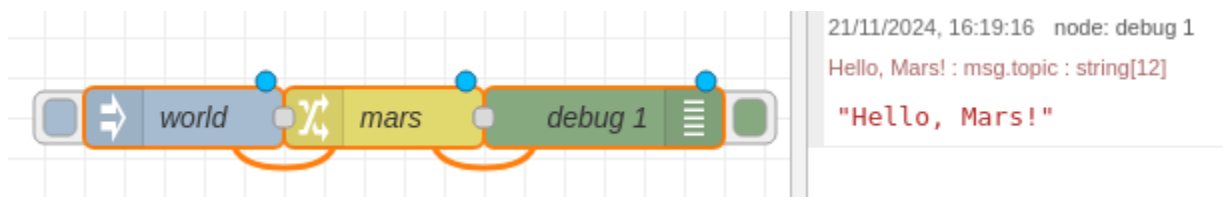
1. Drag in two nodes: Inject and Debug
2. Change Inject's msg.topic to "Hello, World!"
3. Let the Debug node output whatever is in the msg.topic.



2 Hello Mars

What changed:

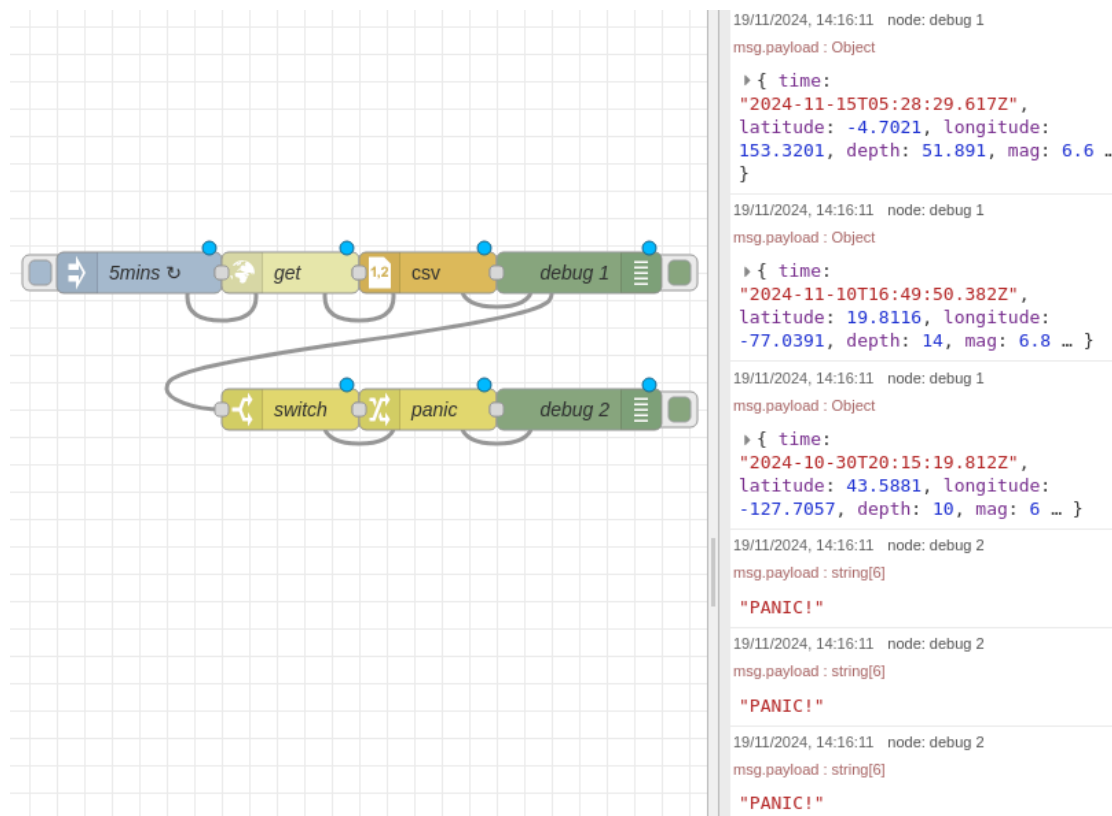
1. Insert Change Node between Inject and Debug Node.
2. Let the Change node search for "World" and change it to "Mars".



3 Earthquakes

Observations:

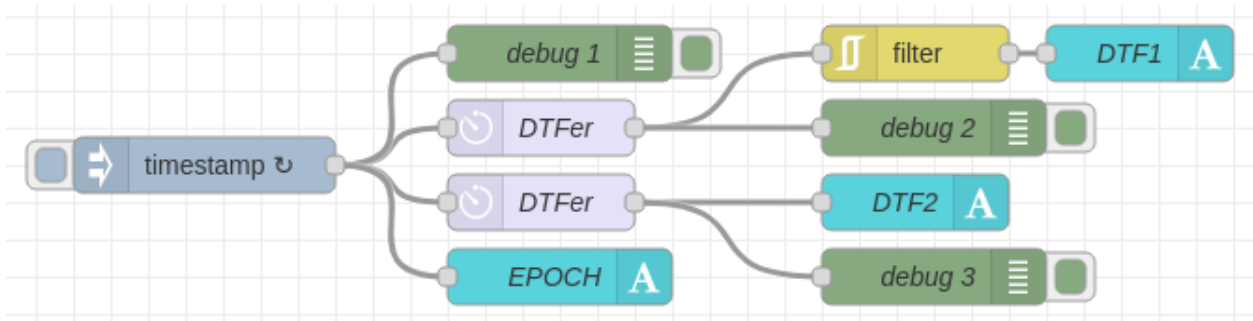
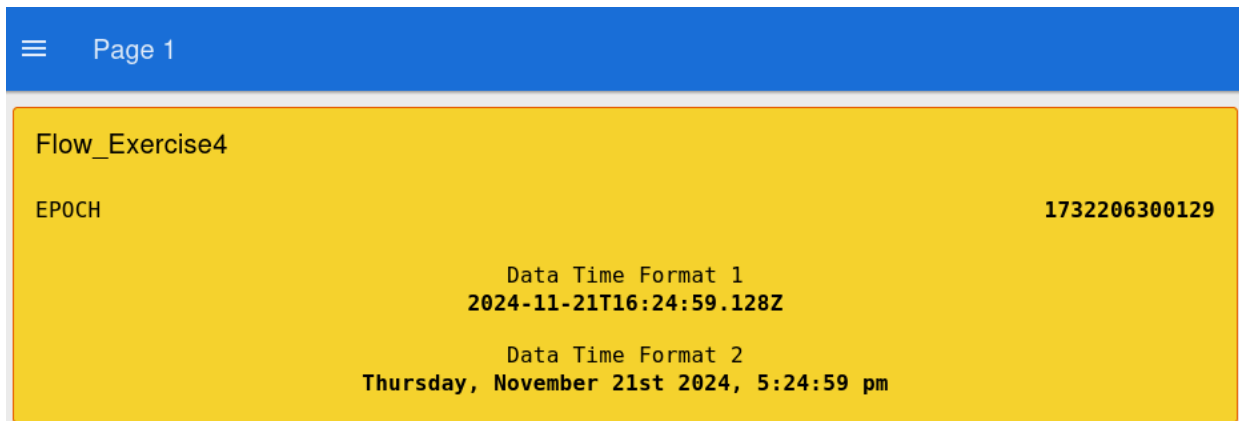
1. Inject Node triggers flow every 5 minutes.
2. CSV Node parses the data into a JSON format.
3. Debug Node outputs the parsed earthquake data.
4. Switch Node filters earthquakes with magnitude ≥ 6 (for example).
5. Thresholds ≥ 1 , ≥ 2 , ≥ 3 , ≥ 4 , ≥ 5 , and ≥ 6 resulted in three "PANIC!" messages for each earthquake that matched the condition.
6. Thresholds ≥ 7 , ≥ 8 , ≥ 9 , and ≥ 10 did not generate any "PANIC!" messages.



4 Date and Time Dashboard

What I did:

1. Wonder what the filter is for.
2. Set the first format to default (ISO8601)
3. Set the second format to `dddd, MMMM Do YYYY, h:mm:ss a`
4. Create an aesthetic theme :)



5 Date and Time Dashboard Pro Max 9000

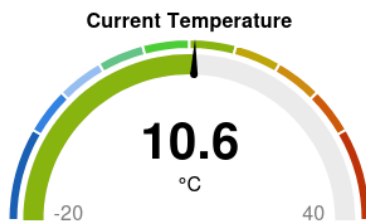
Steps taken:

1. Inject HTTP Request:
<https://api.openweathermap.org/data/2.5/weather?q=Konstanz,de&APPID={API-KEY}>
2. Insert JSON Node that parses into JS and back.
3. Insert Change Node for Current Temperature `$round(payload.main.temp - 273.15, 1)`, Humidity and Temperature (for Chart).

4. Insert function node:

```
const {  
  name,  
  coord: { lat, lon },  
  weather: [{ main, description }]  
} = msg.payload;  
msg.payload = `The weather in ${name} at coordinates: ${lat}, ${lon} is  
${main} (${description}).`;   
  
return msg;
```

5. Wait for Charts to aggregate data.



Current Weather
The weather in Konstanz at coordinates: 47.6603, 9.1758 is Clouds (few clouds).

