

Guia Node Red

Node Red es una plataforma basada en JavaScript, es una herramienta de programación que conecta varios dispositivos a la vez, tanto de hardware como de servicios de internet. Se trata de un motor de flujos que cuenta con un enfoque IoT y esta condición permite definir flujos de servicios a través de protocolos como MQTT.



¿Qué es un objeto?

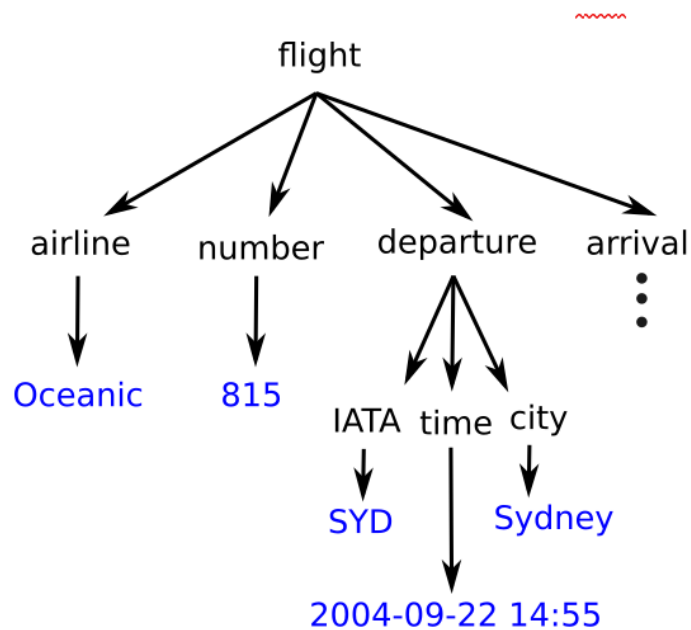
Un objeto en JavaScript es un contenedor de propiedades, donde una propiedad tiene un nombre y un valor. El nombre de una propiedad puede ser una cadena de caracteres, incluso una vacía. El valor de la propiedad puede ser cualquier valor que podamos utilizar en JavaScript, excepto `undefined`. Para más información, acceder al siguiente enlace:

<https://www.arkaitzgarro.com/javascript/capitulo-8.html#:~:text=Un>

Un objeto es una forma de agrupar datos (Ejemplo):

```
var flight = {
  airline: "Oceanic",
  number: 815,
  departure: {
    IATA: "SYD",
    time: "2004-09-22 14:55",
    city: "Sydney"
  },
  arrival: {
    IATA: "LAX",
    time: "2004-09-23 10:42",
    city: "Los Angeles"
  }
}
```

};



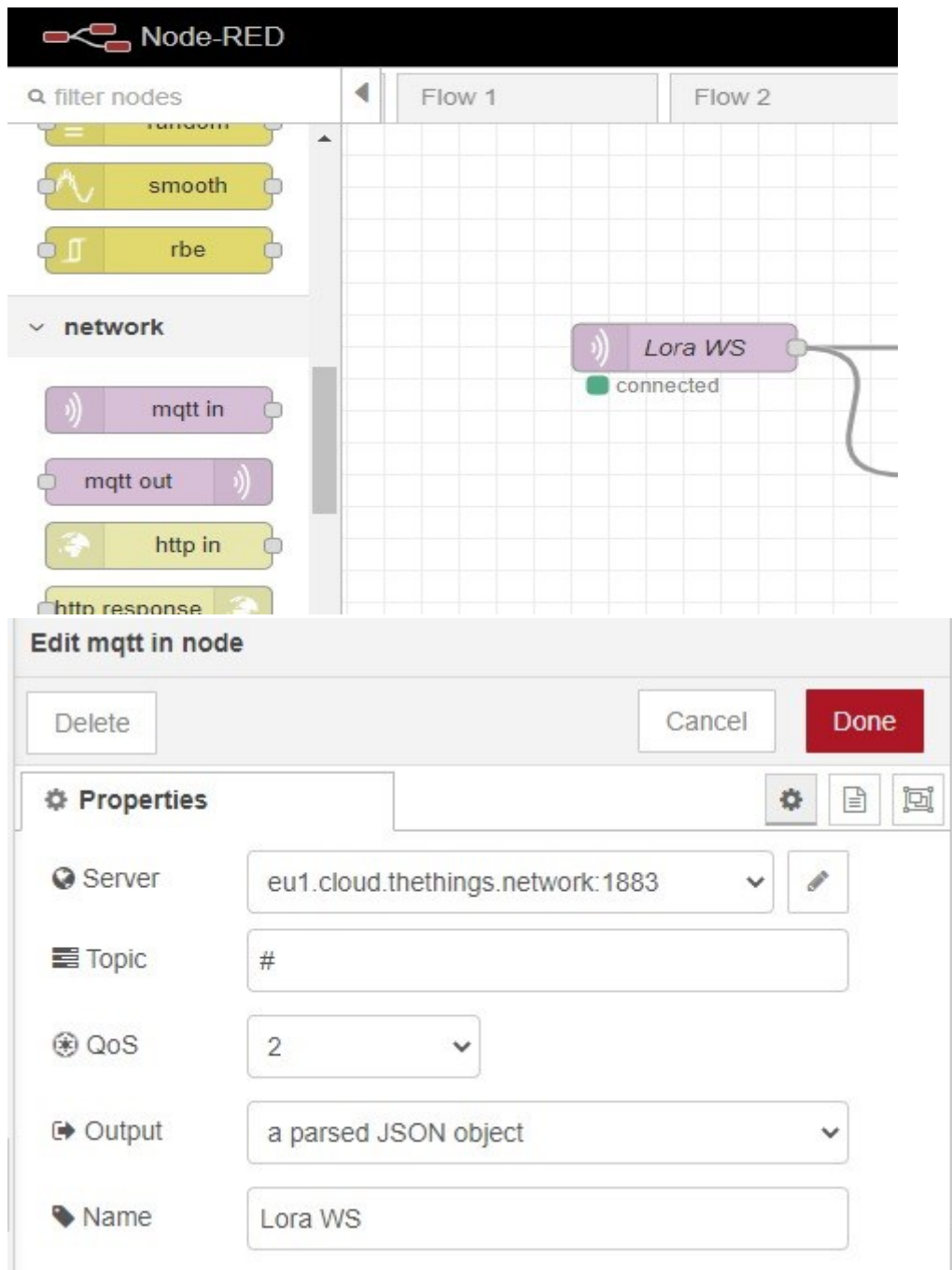
Para acceder a un valor deberemos navegar a través del objeto hasta llegar al valor que queremos:

```
[flight.departure.city] (<http://flight.departure.city>) = Sydney
```

La propiedad es el nombre identificativo (como el nombre de una variable) el valor puede ser: Cadena, numero, otro objeto, un array.

Node RED: VP2-MediaLab

Para crear esta red empezamos creando un node *mqtt* que tiene como parámetros de entrada la información necesaria para conectarse con TTN y obtener la información enviada por Lora.



TTN genera un objeto con cada mensaje recibido de lora, como se puede ver a continuación:

```
{
  "end_device_ids": {
    "device_id": "test1-estacion",
    "application_ids": {
      "application_id": "vp2-medialab"
    },
    "dev_eui": "70B3D57ED00515DF",
    "join_eui": "0000000000000000",
    "dev_addr": "260B4C71"
  },
}
```

```

"correlation_ids": [
  "as:up:01G4CTPRTQ10YX4FWMGFWW3DYZ",
  "gs:conn:01G3R44DV2YZHBXNEC47HE87X8",
  "gs:up:host:01G3R44DVAK0D1AS1ZTWM6PT3K",
  "gs:uplink:01G4CTPRM716CP4HPAJWJK9Q6P",
  "ns:uplink:01G4CTPRM8SSV5MNNJ8WB3K59K",
  "rpc:/ttn.lorawan.v3.GsNs/HandleUplink:01G4CTPRM82TBQ7PW631ND187N",
  "rpc:/ttn.lorawan.v3.NsAs/HandleUplink:01G4CTPRTQEZARF5N7N6A4137H"
],
"received_at": "2022-05-31T10:31:11.447962151Z",
"uplink_message": {
  "session_key_id": "AYEZqItL2qvV9JrEXAEXew==",
  "f_port": 1,
  "f_cnt": 5,
  "frm_payload": "AAAAAAEAAAAAAAAA=",
  "decoded_payload": {
    "bme280": {
      "humedad": 0,
      "presion": 0,
      "temperatura": 0
    },
    "precipitacion": 0,
    "viento": {
      "direccion": "SW",
      "velocidad": 0
    }
  }
},
"rx_metadata": [
  {
    "gateway_ids": {
      "gateway_id": "mk-gateway-p",
      "eui": "353036202A004F00"
    },
    "time": "2022-05-31T10:31:03.545703Z",
    "timestamp": 3888031092,
    "rssi": -111,
    "channel_rssi": -111,
    "snr": -5.5,
    "uplink_token":
"ChoKGaMbWstZ2F0ZXdhcS1wEgglMDYgKgBPABD0yvq9DhoLCO/c15QGEMe3rHIgoJrZhZSOngeQdAjn3NeUBhDYiJuEA==",
    "channel_index": 2
  },
  {
    "gateway_ids": {
      "gateway_id": "gateway-medialab",
      "eui": "3133303757005D00"
    },
    "time": "2022-05-31T10:29:58.478685Z",
    "timestamp": 3776540934,
    "rssi": -47,

```

```

        "channel_rssi": -47,
        "snr": 7.5,
        "location": {
            "latitude": 43.524006,
            "longitude": -5.635203,
            "source": "SOURCE_REGISTRY"
        },
        "uplink_token":
        "Ch4KHAoQZ2F0ZXdhcSltZWRpYWxhYhIIMTMwN1cAXQAQhuLliA4aCwjv3NeUBhCGmb12I
        PD+hdv0ip4BKgwIptzXlAYQyM6g5AE=",
        "channel_index": 2
    }
],
"settings": {
    "data_rate": {
        "lorawan": {
            "bandwidth": 125000,
            "spreading_factor": 7
        }
    },
    "coding_rate": "4/5",
    "frequency": "868500000",
    "timestamp": 3888031092,
    "time": "2022-05-31T10:31:03.545703Z"
},
"received_at": "2022-05-31T10:31:11.240654629Z",
"consumed_airtime": "0.061696s",
"network_ids": {
    "net_id": "000013",
    "tenant_id": "ttn",
    "cluster_id": "eu1",
    "cluster_address": "eu1.cloud.thethings.network"
}
}

```

A partir de este mensaje, si queremos acceder a la precipitación o cualquier otro parámetro se puede obtener la información a través del siguiente mensaje:

```
objeto.uplink_message.decoded_payload.precipitacion
```

Función de node red parte 1

Adicionalmente al mensaje de TTN hay que tener en cuenta que en node RED los mensajes de un bloque a otro se envían a través del objeto **msg.payload** o **msg.query**

```
msg.payload.uplink_message.decoded_payload.precipitacion
```

Después de obtener la información, se crear un **msg.payload** y una función donde se extraen los datos que necesitamos y se generan los objetos **msg.payload** y **msg.query** con la información extraída.

Si nos fijamos en la primera parte del código de la función lo que haremos será crear una variable para cada valor obtenido a partir del mensaje de Lora:

```
// estas variables salen de decodificar el mensaje lora en el apartado de TTN
if (msg.payload.hasOwnProperty("uplink_message"))
{

var devid = msg.payload.end_device_ids.device_id
var app = msg.payload.end_device_ids

//Variables obtenidas de decodificar el mensaje Lora en la parte de TTN
var precipitacion = msg.payload.uplink_message.decoded_payload.precipitacion
var direccion = msg.payload.uplink_message.decoded_payload.viento.direccion
var velocidad = msg.payload.uplink_message.decoded_payload.viento.velocidad
var humedad = msg.payload.uplink_message.decoded_payload.bme280.humedad
var presion = msg.payload.uplink_message.decoded_payload.bme280.presion
var temperatura= msg.payload.uplink_message.decoded_payload.bme280.temperatura
node.warn(app)

msg.payload= {
  precipitacion:precipitacion,
  direccion:direccion,
  velocidad:velocidad,
  temperatura: temperatura,
  presion: presion,
  humedad:humedad,

  id:id_d
}
```

Funcion de node red parte 2

Para poder enviar los datos al servidor usaremos el protocolo HTTP con el método GET

Como funciona el metodo GET

El método get es el mismo que se utiliza para las páginas web podemos dividir una URL en varias partes:

<https://example.com/?product=shirt&color=blue&newuser&size=m>



Nosotros necesitaremos crear una URL como esa para que los datos lleguen al servidor.

Como la primera parte (servidor) es siempre la misma solo crearemos la parte de los parámetros

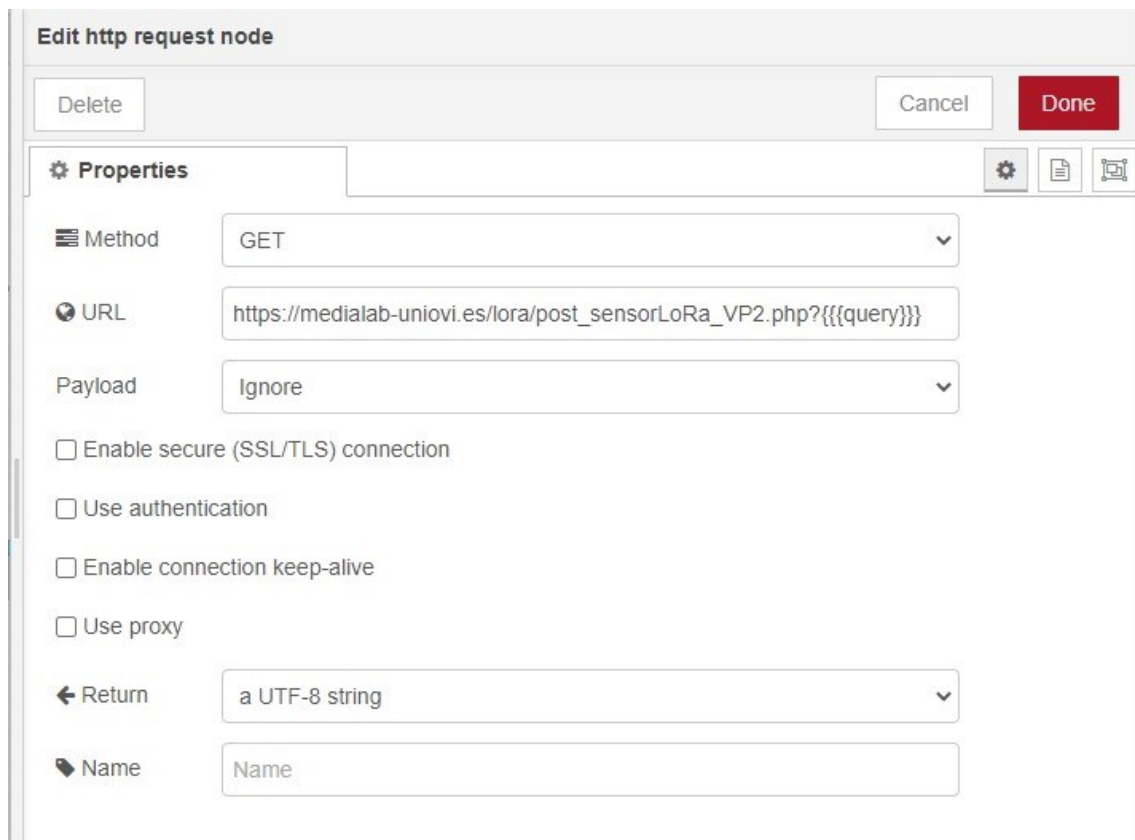
```
var p = msg.payload.precipitacion.toString()
var dir = msg.payload.direccion.toString()
var v = msg.payload.velocidad.toString()
var t = msg.payload.temperatura.toString()
var pr = msg.payload.presion.toString()
var h = msg.payload.humedad.toString()
var id = msg.payload.id.toString()

str="p="+ p + "&d=" + dir + "&v="+ v + "&id=" + id + "&t=" + t+ "&pr="
+ pr + "&h=" + h//

msg.query = str
return msg
```

Enviar los datos

El servidor se especifica en la parte de URL de la configuración del request node



Edit http request node

Delete Cancel Done

Properties

Method: GET

URL: https://medialab-uniovi.es/lora/post_sensorLoRa_VP2.php?{{{query}}}

Payload: Ignore

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

Return: a UTF-8 string

Name: Name

El diagrama final de Node RED para la estación VP2:

