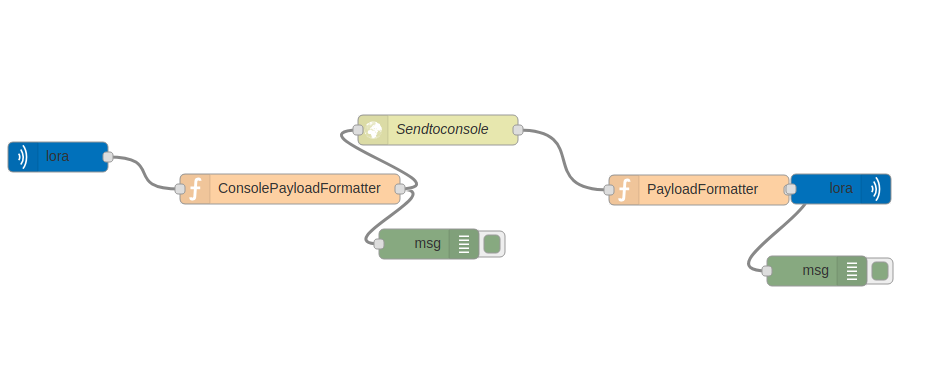
**Node-Red to Python Example**

**Original Node-red Flow**

The image above shows a simple Node-Red flow for Radiobridge that

1. Takes LoRa up-links in

2. Formats the data and prints the data to the debug platform

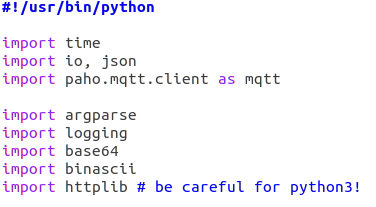
3. Send the data to online console through an HTTP requests

4. Receives response from the HTTP requests that contains a downlink to be queued and formats the downlink response data

5. Sends the downlink to the conduit and prints the data to the debug port

The python script will replicate the major parts of the Node-Red flow. In order to do so in a python we use an MQTT broker to connect to Lora server and listens for uplinks as well as schedule downlinks to the end device.

For python, paho-mqtt is the library for using an MQTT broker. Other libraries needed for this application are

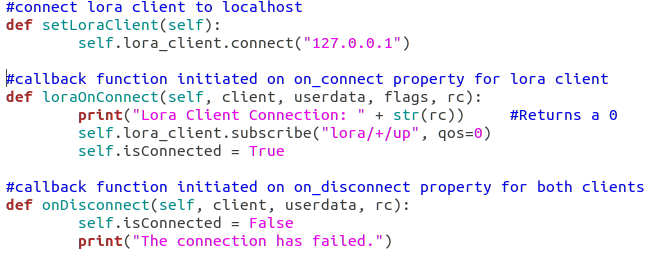


The httplib library will perform the http requests in the application.

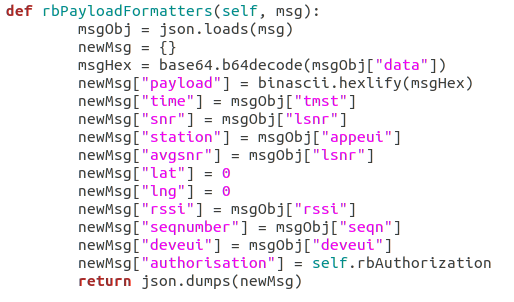
Converting to Python code:

These methods of the class set the mqtt client, and define the behavior when the server is connected and disconnected. The loraOnConnect function also subscribes to listen for uplinks to the gateway.

**1. Takes LoRa up-links in**

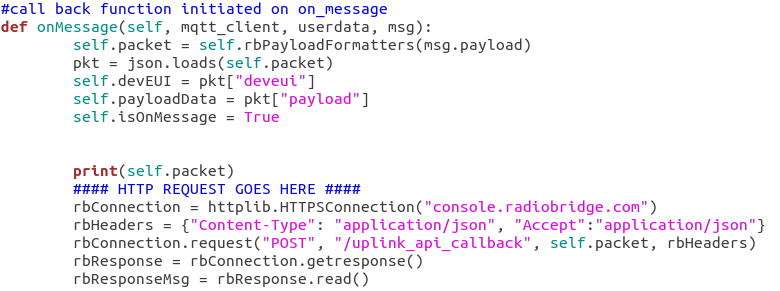


**2. Formats the data and prints the data to the debug platform**



This method is initialized whenever an uplink is received. The method formats the incoming uplink payload and sends that data to a remote online server; in this case the Radiobridge console. The http request response is formatted, printed and published the downlink to the down to the gateway.

**3. Send the data to online console through an HTTP requests**

**4. Receives response from the HTTP requests that contains a downlink to be queued and formats the downlink response data**

**5. Sends the downlink to the conduit and prints the data to the debug port**