

# Webhook Model

*Webhooks*, often referred to HTTP callbacks, enable LoopCloud to push data to web applications. Webhooks are asynchronous and have triggered events and responses. As with Poll models, which retrieve data from external web services, you can use Webhooks to retrieve data from an external system and push that data to the Loop platform. Webhooks create a web service (HTTP endpoint) that allows you to push data to LoopCloud or a third-party system.

## Webhook Use Cases

In addition to MQTT and LwM2M, LoopCloud supports incoming webhooks as a mechanism for receiving data from external systems.

- Incoming webhooks are basically REST endpoints.
- Transformation rules can convert incoming data to a format supported by the Loop platform. See [Webhook Model Transformation](#).

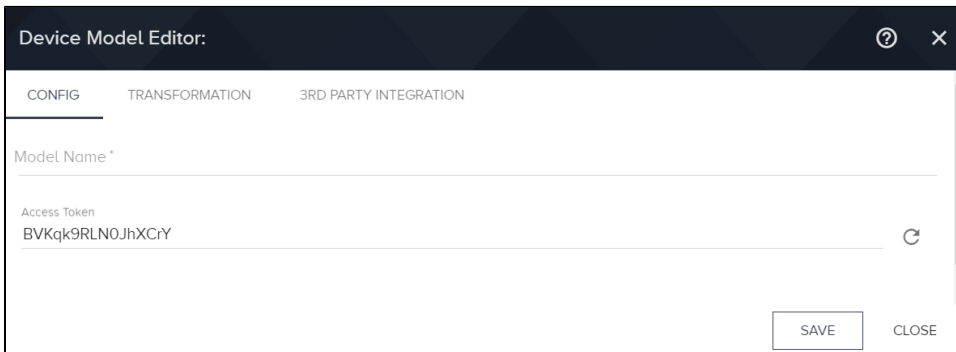
## Example Webhook Case

SigFox acts as an IoT sensor network and it allows users to define Webhooks to invoke upon each sensor metric change. See [SigFox Transformation Example](#).

## Create a Webhook Model

To create a Webhook Model with LoopCloud:

1. Click the Company name at the top of the LoopCloud window.
2. Click the **Project**.
3. In the navigation panel, choose **Configuration > Models**.
4. Select the **WEBHOOK** tab. See also, [Model Types](#).
5. Click **New Model**.



The screenshot shows the 'Device Model Editor' window with three tabs: CONFIG, TRANSFORMATION, and 3RD PARTY INTEGRATION. The CONFIG tab is active. It contains a 'Model Name' field, an 'Access Token' field with the value 'BVKqk9RLNOJhXCrY', and a refresh icon. At the bottom right, there are 'SAVE' and 'CLOSE' buttons.

6. In the **CONFIG** tab:
  - a. Enter a **Model Name**.
  - b. Notice that the Access Token field gets populated automatically.
7. Click **Save**.

**i** When you deploy a device from a Webhook model, the Webhook will be available as a URL in the following format: `https://1itmus.pro/webhook/{projectId}/{deviceId}`

## Webhook Model Transformation

- See [Poll Model to Transform Data](#) for a description of transformation.

## SigFox Transformation Example

You can define transformation rules, as shown in the following example. The following rules convert incoming data into a format supported by the Loop platform.

```

import com.litmusloop.common.device.Payload;
import groovy.json.JsonSlurper;
import org.apache.commons.io.IOUtils;

/**
 * in: byte[] input
 * in: OMNARegistry omnaRegistry
 * out: return List<Payload.Value>
 */
String inputJson = new String(input);

def jsonSlurper = new JsonSlurper();
def json = jsonSlurper.parseText(inputJson);

List<Payload.Value> result = new ArrayList<>();
if (json != null) {
    // sensor id
    String sensorID = json.device
    result.add(new Payload.Value(
        3, // Device
        sensorID,
        2,
        "String",
        sensorID,
        null
    ));
    // Timestamp
    String timestamp = json.time;
    result.add(new Payload.Value(
        3, // Device
        sensorID,
        13,
        "Time",
        timestamp,
        null
    ));
    // RSSI Level
    String rssi = json.rssi;
    result.add(new Payload.Value(
        3305, // Power Measurements
        sensorID,
        5800,
        "Float",
        rssi,
        null
    ));
    // Gas service inlet pressure
    String ainl = json.ainl;
    result.add(new Payload.Value(
        3323, // Pressure
        sensorID,
        5700,
        "Float",
        ainl,
        null
    ));
    // Voltage
    String voltage = Integer.valueOf(json.vdc) / 1000.0;
    result.add(new Payload.Value(
        3316, // OMA Voltage ID
        sensorID, // arbitrary instance ID = sensorID
        5700, // OMA Sensor Value
        "Float",
        voltage,
        null
    ));
    // Meter pulse
    String dinl = json.dinl;
    result.add(new Payload.Value(
        3200, // Digital Input

```

```
        sensorID,  
        5501,  
        "Integer",  
        din1,  
        null  
    });  
    // Meter pulse totalizer  
    String cnt1 = json.cnt1;  
    result.add(new Payload.Value(  
        3200, // Digital Input  
        sensorID,  
        5503,  
        "Integer",  
        cnt1,  
        null  
    ));  
}  
return result;
```