

 Search for articles...

All Collections > Install, Setup, and Tech Specs >  Smart Home API - Technical Specifications

Smart Home API - Technical Specifications

Tech Specs and more about our home API



Written by Hydrific Team

Updated over a week ago

Table of contents



While Droplet's API integrations are focused on Home Assistant, they are flexible enough to be used with other home automation systems or personal projects. This document will describe the format and content of the messages Droplet sends over 2 protocols: MQTT and WebSockets.

MQTT

Configuration

The following fields are configurable in the mobile app. The maximum length for all text-based fields is 256 characters.

- Host name. This is the host name (or IP address) of the MQTT broker that the Droplet will try to connect to.
- Port number. The default value is 1883, but can be set to any valid port. (Note: while the port can be set to 8883, Droplet does not support MQTT over SSL.)
- Username and password.
- Discovery prefix. This field is used to create the topic over which Droplet sends its discovery message.

- Status topic. This is the topic Droplet will subscribe to in order to track the health of a relevant client. If your home automation system has no analog of this, it can be left as the default and messages to this topic can be ignored.

Message types

The Droplet sends 3 types of messages: discovery, state, and health. Included as part of each topic is a 4-character identifier, e.g. ABCD.

Discovery

The discovery message is sent once when Droplet connects to the broker, or when it receives a message to the configure status topic with the payload `online`.

Topic: `<discovery_prefix>/device/droplet-<identifier>/config`

Payload: JSON message containing device metadata (name, identifier, manufacturer, model, firmware version), the topics on which the device will send state and health messages, and a description of each of the state messages it will send. The payload makes use of some abbreviations, which can be found here ([MQTT](#)).

Example payload:

```
{
  "avty_t": "droplet-F674/health",
  "stat_t": "droplet-F674/state",
  "dev": {
    "name": "Droplet",
    "ids": "droplet-F674",
    "mf": "Hydrific, part of LIXIL",
    "mdl": "Droplet 1.0",
    "sw": "v1.1.0",
    "sn": "F674"
  },
  "o": {
    "name": "droplet",
    "sw": "v1.1.0",
    "url": "https://hydrificwater.com"
  },
  "cmps": {
    "flow": {
      "p": "sensor",
      "name": "Flow Rate",
```

```
"dev_cla": "volume_flow_rate",
"unit_of_meas": "L/min",
"sug_dsp_prc": 2,
"val_tpl": "{{ value_json.flow }}",
"uniq_id": "flow-F674",
"ic": "mdi:water-circle"
},
"volume": {
  "p": "sensor",
  "name": "Volume",
  "dev_cla": "water",
  "unit_of_meas": "L",
  "stat_cla": "total",
  "lrst_t": "droplet-F674/state",
  "lrst_val_tpl": "{{ utcnow() }}",
  "sug_dsp_prc": 2,
  "val_tpl": "{{ (value_json.volume | float) / 1000 }}",
  "uniq_id": "volume-F674",
  "ic": "mdi:cup-water"
},
"signal_quality": {
  "p": "sensor",
  "name": "Signal Quality",
  "dev_cla": "enum",
  "ent_cat": "diagnostic",
  "val_tpl": "{{ value_json.signal }}",
  "uniq_id": "signal_quality-F674",
  "ic": "mdi:waveform",
  "ops": [
    "Initializing",
    "No Signal",
    "Weak Signal",
    "Strong Signal"
  ]
},
"server_connectivity": {
  "p": "sensor",
  "name": "Server Status",
  "dev_cla": "enum",
  "ent_cat": "diagnostic",
  "val_tpl": "{{ value_json.server }}",
  "uniq_id": "server_connectivity-F674",
  "ic": "mdi:web",
  "ops": [
    "Connected",
```

```
        "Connecting",  
        "Disconnected"  
    ]  
}  
}
```

State

Topic: droplet-<identifier>/state

Payload: JSON message containing one or more of the following fields:

Key	Description	Data type
server	Connectivity status to Hydrific servers	string
signal	Signal quality of the ultrasonic sensor	string
volume	Point-to-point volume (mL), i.e. volume detected by Droplet since the last point published.	float
flow	Current flow rate in L/min.	float

Example payload:

```
{  
  "server": "Connected",  
  "signal": "Strong Signal",  
  "flow": 0.1,  
  "volume": 0.2  
}
```

Health

Topic: droplet-<identifier>/health

Payload: text message reading either `online` or `offline`. Droplet periodically sends an online message. It registers a LWT with the broker to send an offline message on its behalf if it disconnects.

WebSockets

On the Droplet mobile app, the WebSockets API is called “Home Assistant Core.” After enabling this option in the mobile app, there is no additional configuration necessary. Droplet acts as a WebSocket server, listening on port 443.

The [pydroplet](#) package provides an interface to retrieve data from the Droplet. See the [examples directory](#) at the GitHub repository to get started. This package should provide everything you need to read data from Droplet using the WebSockets API. However, if your application precludes the use of this library, it is possible to interface with Droplet without it.

Connection

Droplet can support up to 2 simultaneous connections over WebSockets. TLS is used to encrypt communication between server and client, but the client should not check Droplet’s hostname or verify its certificate. The token should be sent as part of an authorization header. See the library source code for an example of connection.

Message types

Metadata

Within 5 seconds of opening a connection to a new device, Droplet will send a JSON message containing its metadata. Example:

```
{
  "ids": "Droplet-ABCD",
  "mf": "Hydrific, part of LIXIL",
  "mdl": "Droplet 1.0",
  "sw": "v1.0.0",
  "sn": "ABCD"
}
```

State

State messages are identical to those sent over MQTT.

Keepalive

Droplet will send a PING message every 10 seconds. If it does not receive a PONG in reply within 10 seconds, it will close the connection. It should respond to PINGs in a timely manner.

Related Articles



API setup - MQTT with Home Assistant



Smart Home API - Which option is right for me?



Did this answer your question?

