

Websocket API specification

Connection details

URL	Scheme	Description
host:443	wss	Once the socket is open you can subscribe to channels by sending a subscribe request message.

Authentication

After enabling local API in a mobile app, user gets username and password that can be used to subscribe to local API in current property.

The resulting credentials must be provided in the HTTP Authorization header when opening websocket connection.

Messages

ping

Request (sent by client)

Client can ping server to determine whether connection is alive, server responds with pong. This is an application level ping as opposed to default ping in websockets standard which is server initiated

Payload

Name	Type	Description
event	string	

Example of payload

```
{
  "event": "ping"
}
```

pong

Response (sent by server)

Server pong response to a ping to determine whether connection is alive. This is an application level pong as opposed to default pong in websockets standard which is sent by client in response to a ping

Payload

Name	Type	Description
event	string	

Example of payload

```
{
  "event": "pong"
}
```

telemetry

Publication (sent by server)

Telemetry message

Payload

Name	Type	Description	
event	string		
data	object		
water_flow	float		
leak_status	string	"No Leaks", "Small Leak", "Major Leak"	
status	Array (string)	"OKAY", "AMBIENT_TEMP_FAILED", "BATTERY_FAILED", "PRESSURE_FAILED", "WATER_TEMP_FAILED", "WATERFLOW_FAILED"	
water_temp	float		
ambient_temp	float		
abs_pressure	integer	Absolute pressure (including	
battery_level	string	"okay" "low" "critical" "none"	
probed_at	integer	Unix epoch time in milliseconds	
units	object		
water_flow	string	"gpm", "l/min"	
water_temp	string	"C", "F"	
ambient_temp	string	"C", "F"	
abs_pressure	string	"mbar", "psi"	

Example of payload

```
{
  "event": "telemetry",
  "data": {
    "water_flow": 63.621673583984375,
    "leak_status": "No Leaks",
    "status": ["OKAY"],
    "water_temp": 20.4375,
    "ambient_temp": 20.625,
    "abs_pressure": 4418,
    "battery_level": "okay",
    "probed_at": 1569369606308,
    "units": {
      "water_flow": "ml/min",
      "water_temp": "C",
      "ambient_temp": "C",
      "abs_pressure": "mbar"
    }
  }
}
```

state

Publication (sent by server)

State message

Payload

Name	Type	Description	
event	string		
data	object		
valve_state	string	"open" "closed" "opening" "closing"	
radio_state	string	"connected" "disconnected"	
siren	boolean		FUTURE

Example of payload

```
{
  "event": "state",
  "data": {
    "valve_state": "open",
    "radio_state": "connected",
    "siren": true
  }
}
```

changeState

Request (sent by client)
Change a state attribute

Payload

Name	Type	Description	
event	string		
data	object		
valve_state	string	"open", "closed"	
led_brightness	integer	0-100	FUTURE
siren	boolean	true/false	FUTURE
signal_reboot	boolean	true	FUTURE
sonic_reboot	boolean	true	FUTURE

Example of payload

```
{
  "event": "changeState",
  "data": {
    "valve_state": "open",
    "led_brightness": 100,
    "siren": true,
    "signal_reboot": true,
    "sonic_reboot": true
  }
}
```

Request (sent by client)

Request telemetry message

Payload

Name	Type	Description
event	string	

Example of payload

```
{
  "event": "requestTelemetry"
}
```

requestState

Request (sent by client)

Request state message

Payload

Name	Type	Description
event	string	

Example of payload

```
{
  "event": "requestState"
}
```

error

Response (sent by server)

Payload

Name	Type	Description
event	string	
code	string	
message	string	

Example of payload

```
{
  "event": "error",
  "code": "S11",
  "message": "Invalid valve state: 'wrong_state'. Valid states: 'open',
'closed' "
}
```

Error codes:

G10: Server error

G11: Invalid request

S10: Device unavailable

S11: Invalid valve state