



How to **develop for Hue?**

Develop

Get Started

Application Design Guidance

Hue API

Hue Entertainment

Tools and SDKs

Overview

5. Sensors API

Datatypes and Time Patterns

1. Lights API

2. Groups API

3. Schedules API

4. Scenes API

5. Sensors API

6. Rules API

7 Configuration API

8. Info API
(deprecated as of 1.15)

9. Resourcelinks API

10. Capabilities API

5.1. Get all sensors

URL	<code>/api/<username>/sensors</code>
Method	<code>GET</code>
Version	1.3
Permission	Whitelist

5.1.1. Description

Gets a list of all sensors that have been added to the bridge.

5.1.2. Response

Returns a list of all sensors in the system. Each sensor has a name and unique identification number.

If there are no sensors then the bridge will return an empty object, {}.

5.1.3. Sample Response

```
{
  "1": {
    "state": {
      "daylight": false,
      "lastupdated": "2014-06-27T07:38:51"
    }
  },

```

On this page:

[5.1. Get all sensors](#)

[5.2. Create sensor](#)

[5.3. Find new sensors](#)

[5.4. Get New Sensors](#)

[5.5. Get Sensor](#)

[5.6. Update Sensor](#)

[5.7. Delete Sensor](#)

[5.8. Change Sensor Config](#)

[5.9. Change Sensor State](#)

Remote API Quick start guide	
Remote Authentication	
Remote Hue API - Error Messages	
Error messages	<pre> "config": { "on": true, "long": "none", "lat": "none", "sunriseoffset": 50, "sunsetoffset": 50 }, "name": "Daylight", "type": "Daylight", "modelid": "PHDL00", "manufacturername": "Philips", "swversion": "1.0" }, "2": { "state": { "buttonevent": 0, "lastupdated": "none" }, "config": { "on": true }, "name": "Tap Switch 2", "type": "ZGPSwitch", "modelid": "ZGPSWITCH", "manufacturername": "Philips", "uniqueid": "00:00:00:00:00:00:40:03:50-f2" } } </pre>
Message Structure and Response	
Supported Devices	
API Documentation Changelog	
Glossary terms	

5.2. Create sensor

URL	<code>/api/<username>/sensors</code>
Method	<code>POST</code>
Version	1.3
Permission	Whitelist

5.2.1. Description

Allows the creation of sensors.

5.2.2. Body arguments

Name	Type	Description	
name	string 1..32	The human readable name of the sensor, can be changed by the user. Is not allowed to be empty on change.	Required
modelid	string	This parameter uniquely	Required

Name	Type	Description	
	6..32	identifies the hardware model of the device for the given manufacturer.	
swversion	string 1..16	This parameter uniquely identifies the software version running in the hardware.	Required
type	string 1..32	Type name of the sensor	Required
uniqueid	string 6..32	Unique id of the sensor. Should be the MAC address of the device.	Required
manufacturername	string 6..32	The name of the device manufacturer.	Required
state	object	The state object with attributes corresponding to the sensor type Attribute values represents initial state of sensor after creation.	Optional
config	object	The configuration object with attributes corresponding to the sensor type. Attribute values represents configuration information.	Optional
recycle	bool	When true: Resource is automatically deleted when not referenced anymore in any resource link. Only for CLIP sensors on creation of resource. "false" when omitted.	Optional

Config Object Attributes

Name	Type	Description
on	bool	Turns the sensor on/off. When off, state changes of the sensor are not reflected in the sensor resource. Default is "true"
reachable	bool	Indicates whether communication with devices is possible. CLIP Sensors do not yet support reachable verification. Mandatory for all Sensors except ZGPSwitch, Daylight

Name	Type	Description
battery	uint8	The current battery state in percent, only for battery powered devices. Not present when not provided on creation (CLIP sensors).

5.2.3. Sample Body

```
{
  "state": {
    "presence": false
  },
  "name": "IP Camera living room",
  "modelid": "IPSENSOR",
  "swversion": "1.0",
  "type": "CLIPPpresence",
  "uniqueid": "12345678",
  "manufacturername": "TheManufacturer"
}
```

5.2.4. Response

Contains a list with a single item that details whether the sensor was added successfully.

5.2.5. Sample Response

```
[{
  "success": {"id": "4"}
}]
```

5.2.6. Notes

The following errors can occur upon sensor creation:

code	description
501	Not allowed to create sensor type
502	Sensor list is full.
5,6,7	Unsupported attribute for sensor type. Illegal sensor value.

5.3. Find new sensors

URL	<code>/api/<username>/sensors</code>
Method	<code>POST</code>
Version	1.3
Permission	Whitelist

5.3.1. Description

Starts a search for new sensors.

5.3.2. Sample Response

```
[ { "success": { "/sensors": "Searching for new devices" } } ]
```

5.4. Get New Sensors

URL	<code>/api/<username>/sensors/new</code>
Method	<code>GET</code>
Version	1.3
Permission	Whitelist

5.4.1. Description

Finds all new sensors since the last scan.

Name	Type	Description
		Either:
lastscan	string	<ul style="list-style-type: none"> time – Date last scan completed in ISO 8601:2004 active – Scan currently ongoing none – No scan has taken place since last reboot

5.4.2. Sample Response

```
{
  "7": { "name": "Hue Tap 1" },
  "8": { "name": " Button 3" }
  "lastscan": "2013-05-22T10:24:00"
}
```

5.5. Get Sensor

URL	<code>/api/<username>/sensors/<id></code>
Method	<code>GET</code>
Version	1.3
Permission	Whitelist

5.5.1. Description

Gets the sensor from the bridge with the given id.

5.5.2. Sample Response

```
{
  "state":{
    "buttonevent": 34,
    "lastupdated":"2013-03-25T13:32:34",
  },
  "name": "Wall tap 1",
  "modelid":"ZGPSWITCH",
  "uniqueid":"01:23:45:67:89:AB-12",
  "manufacturername": "Philips",
  "swversion":"1.0",
  "type": "ZGPSwitch"
}
```

5.6. Update Sensor

URL	<code>/api/<username>/sensors/<id></code>
Method	<code>PUT</code>
Version	1.3
Permission	Whitelist

5.6.1. Description

Renames the sensor in the bridge for the supplied id. A sensor can have its name changed when it is in any state, unreachable/off etc.

5.6.2. Sample Body

```
{"name": "Bedroom Tap"}
```

5.6.3. Sample Response

```
[{"success":{"/sensors/2/name": "Bedroom Tap"}}]
```

5.7. Delete Sensor

All sensors can be deleted.

URL	<code>/api/<username>/sensors/<id></code>
Method	<code>DELETE</code>
Version	1.3
Permission	Whitelist

5.8. Change Sensor Config

URL	<code>/api/<username>/sensors/<id>/config</code>
Method	<code>PUT</code>
Version	1.3
Permission	Whitelist

5.8.1. Description

The allowed configuration parameters depend on the sensor type.

5.8.2. Sample Body

```
{
  "on": true
}
```

5.8.3. Sample Response

```
[
  {"success": {"/sensors/2/config/on": true}}
]
```

5.9. Change Sensor State

URL	<code>/api/<username>/sensors/<id>/state</code>
Method	<code>PUT</code>
Version	1.3
Permission	Whitelist

5.9.1. Description

Used to allow the state of a CLIP sensor to be updated.

5.9.2. Sample Body

```
{
  "presence": false
}
```

5.9.3. Sample Response

```
[
  {"success": {"/sensors/1/state/presence": false}}
]
```

5.9.4. Notes

All values outside of range shall return error 7. All configuration change parameters are optional. The call will always return error 8 if the sensor is not a CLIP sensor.



Connect with us



[Contact](#) [Terms & Conditions](#) [Privacy](#) [Product Security](#)

©2018-2019 Signify Holding. All rights reserved.