



How to **develop for Hue?**

Develop

Get Started

Application Design Guidance

Hue API

Hue Entertainment

Tools and SDKs

Overview

Supported Devices

Datatypes and Time Patterns

The hue system supports several devices. These are divided into lights, luminaires and sensors (ZigBee and CLIP).

1. Lights API

Supported lights

On this page:

2. Groups API

Within the Hue system we have several light types. These light types may have different lighting characteristics and attributes. The attributes of the lights will be described in this section.

[Supported lights](#)

3. Schedules API

NOTE : The supported light table will not be maintained due to the large expansion and dynamic nature of new/updated Hue products. Instead, the information is readily available in the Light properties via API. The table is still available for legacy purposes. [Please read for more info](#) such as modelid, archetype, function, direction, etc.

[Supported sensors](#)

4. Scenes API

5. Sensors API

6. Rules API

You can retrieve the light types with their attributes by sending a “Get light attributes and state” request to the bridge, which is described in the Lights API. This will return a list of attributes amongst which the “type” and “modelid” are important to determine the characteristic of the light. The “type” is a fixed name, which represents the ZigBee Light Link Device ID number and name, and describes the functionality that the light supports, which are:

8. Info API
(deprecated as of 1.15)

- On/off light (ZigBee Device ID: 0x0000) , supports groups, scenes and on/off control
- Dimmable light (ZigBee Device ID: 0x0100) , which supports groups, scenes, on/off and dimming.
- Color temperature light (ZigBee Device ID: 0x0220) , which supports groups, scenes, on/off, dimming, and setting of a color temperature.

9. ResourceLinks API

10. Capabilities API

Remote API Quick start guide

Remote Authentication

Remote Hue API – Error Messages

Error messages

Message Structure and Response

- Color light (ZigBee Device ID: 0x0200) , which supports groups, scenes, on/off, dimming and color control (hue/saturation, enhanced hue, color loop and XY)
- Extended Color light (ZigBee Device ID: 0x0210) , same as Color light, but which supports additional setting of color temperature

Modelid identifies the hardware model of the light. The following table illustrates the Hue models which are currently available and can be identified.

*Note that the Living Color Lights (i.e. LLC006 and LLC007) can be identified, but are not officially supported as Friends of Hue Lights because they can not be commissioned the same way as Friends of Hue lights.

Supported Light Types

Product Name	Device ID (Type)	Model ID	Color Gamut	Hue / Friends of Hue
Hue bulb A19	0x0210 (Extended Color Light)	LCT001, LCT007	B	Yes
Hue bulb A19	0x0210 (Extended Color Light)	LCT010, LCT014, LCT015, LCT016	C	Yes
Hue Spot BR30	0x0210 (Extended Color Light)	LCT002	B	Yes
Hue Spot GU10	0x0210 (Extended Color Light)	LCT003	B	Yes
Hue BR30 Richer Colors	0x0210 (Extended Color Light)	LCT011	C	Yes
Hue BR30 White Ambience	0x0220 (Color Temperature Light)	LTW011	2200K-6500K	Yes
Hue LightStrips	0x0200 (Color Light)	LST001	A	Yes
Hue Living Colors Iris	0x0200 (Color Light)	LLC010	A	Yes
Hue Living Colors Bloom	0x0200 (Color Light)	LLC011, LLC012	A	Yes

Supported Devices

API Documentation Changelog

Glossary terms

Product Name	Device ID (Type)	Model ID	Color Gamut	Hue / Friends of Hue
Living Colors Gen3 Iris*	0x0200 (Color Light)	LLC006	A	No
Living Colors Gen3 Bloom, Aura*	0x0200 (Color Light)	LLC005, LLC007, LLC014	A	No
Disney Living Colors	0x0200 (Color Light)	LLC013	A	Yes
Hue White	0x0100 (Dimmable Light)	LWB004, LWB006, LWB007	-	Yes
Hue White lamp	0x0100 (Dimmable Light)	LWB010, LWB014	-	Yes
Color Light Module	0x0210 (Extended Color Light)	LLM001	B	Yes
Color Temperature Module	0x0220 (Color Temperature Light)	LLM010, LLM011, LLM012	2200K-6500K	Yes
Hue A19 White Ambiance	0x0220 (Color Temperature Light)	LTW001, LTW004, LTW010, LTW015	2200K-6500K	Yes
Hue ambiance spot	0x0220 (Color Temperature Light)	LTW013, LTW014	2200K-6500K	Yes
Hue Go	0x0210 (Extended Color Light)	LLC020	C	Yes
Hue LightStrips Plus	0x0210 (Extended Color Light)	LST002	C	Yes
Hue color candle	0x0210 (Extended Color Light)	LCT012	C	Yes
Hue	0x0220	LTW012	2200K-6500K	Yes

Product Name	Device ID (Type)	Model ID	Color Gamut	Hue / Friends of Hue
ambiance candle	(Color Temperature Light)			

Supported Lamp Types

Product Name	Device ID (Type)	Model ID	Color Gamut	Hue / Friends of Hue
Hue ambiance pendant	0x0220 (Color Temperature Light)	LTP001, LTP002, LTP003, LTP004, LTP005, LTD003	2200K-6500K	Yes
Hue ambiance ceiling	0x0220 (Color Temperature Light)	LTF001, LTF002, LTC001, LTC002, LTC003, LTC004, LTC011, LTC012, LTD001, LTD002	2200K-6500K	Yes
Hue ambiance floor	0x0220 (Color Temperature Light)	LFF001	2200K-6500K	Yes
Hue ambiance table	0x0220 (Color Temperature Light)	LTT001	2200K-6500K	Yes
Hue ambiance downlight	0x0220 (Color Temperature Light)	LDT001	2200K-6500K	Yes
Hue white wall washer	0x0100 (Dimmable Light)	LDF002	-	Yes
Hue white ceiling	0x0100 (Dimmable Light)	LDF001	-	Yes

Product Name	Device ID (Type)	Model ID	Color Gamut	Hue / Friends of Hue
Hue white floor	0x0100 (Dimmable Light)	LDD002	-	Yes
Hue white table	0x0100 (Dimmable Light)	LDD001	-	Yes
Hue white 1-10V	0x0100 (Dimmable Light)	MWM001	-	Yes

Multisource Luminaires - using the Color Light Module (LLM001).

Product Name	Number of light modules	Model ID luminaire group
Hue Beyond Table	2	HBL001
Hue Beyond Pendant	4	HBL002
Hue Beyond Ceiling	4	HBL003
Hue Entity Table	2	HEL001
Hue Entity Pendant	2	HEL002
Hue Impulse Table	2	HIL001
Hue Impulse Pendant	2	HIL002

Multisource Luminaires - using the Color Temperature Module (LLM010/LLM011/LLM012).

Product Name	Number of light modules	Model ID luminaire group
Hue Phoenix Centerpiece	3	HML001
Hue Phoenix Ceiling	3	HML002

Product Name	Number of light modules	Model ID luminaire group
Hue Phoenix Pendant	2	HML003
Hue Phoenix Wall	1	HML004
Hue Phoenix Table	1	HML005
Hue Phoenix Downlight	1	HML006

The “Friendly name for hue” item is the name of the light as soon it has been discovered by the bridge. A sequence number is added to distinguish multiple lamps.

As indicated in the table, lights may have different color gamuts. The following color gamuts are supported by the Hue lights:

Gamut A

Color	x	y
Red	0.704	0.296
Green	0.2151	0.7106
Blue	0.138	0.08

Gamut B

Color	x	y
Red	0.675	0.322
Green	0.409	0.518
Blue	0.167	0.04

Gamut C

Color	x	y
Red	0.692	0.308
Green	0.17	0.7
Blue	0.153	0.048

The following CIE figure illustrates the three gamut ranges the hue lights have.

1.1 On/Off light

Top level attributes

type	string	device type of the light
name	<i>string</i> 0..32	Human readable name of the light. Names has to be unique.
modelid	<i>string</i> 6..32	This parameter uniquely identifies the hardware model of the device for the given manufacturer.
productid	<i>string</i> 1..32	Uniquely identifying hardware models for the given manufacturer. Not available for all devices.
swversion	<i>string</i> 8..16	This parameter uniquely identifies the software version running in the hardware.
swconfigid	<i>string</i> 1..16	Uniquely identifying configuration of the software running in the hardware. Not available for all devices.
state	object	The object representing the state of the light. Supported attributes are listed for each supported light class in 5Supported lights
config	object	Additional configuration of a light
capabilities	object	Lists capabilities of the light
swupdate	object	
state object attributes		
reachable	bool	Indicates if a lamp can be reached by the smart bridge
on	bool	On/Off state of light. On=True, Off=False.
transitiontime	<i>uint16</i>	Duration of the transition for this API command only in 100 millisecond steps. This is an optional write-only parameter with a default value of 4 (400 milliseconds).
alert	string	<p><i>Alert effect value. A PUT with other unknown values shall generate error 7. Note the alert resource contains the last alert sent to the light and not its current state. The effect will also stop if any other action is performed changing the light.</i></p> <p><i>“none”</i> <i>The lamp shall stop performing all alert effects.</i></p> <p><i>“select”</i> <i>The lamp shall perform one breathe cycle.</i></p>

		<i>“lselect”</i>	<i>The lamp shall perform breathe cycles for 15 seconds or until the “alert”:”none” command is received.</i>
		<i>homeautomation</i>	<i>Device can be controlled by updating /state, light output is reflected in /state</i>
<i>mode</i>	enum	<i>streaming</i>	<i>Device is in use by a streaming client and cannot be controlled by updating /state and light output is not reflected in /state</i>
swupdate object attributes			
<i>state</i>	String		State of software update for this device.
		<i>notupdatable</i>	<i>System cannot update this device or determine if it is out-of-date.</i>
		<i>noupdates</i>	<i>No update available nor known.</i>
		<i>transferring</i>	<i>Bridge knows there is an update is available. But not yet downloaded from portal or finished transferring to device.</i>
		<i>readytoinstall</i>	<i>Software is ready to install (ie transferred to device).</i>
		<i>installing</i>	<i>Software update is installing. Note that the device might not be usable for 30-60s during installation.</i>
		<i>batterylow</i>	<i>Battery is too low for update.</i>
		<i>imagerejected</i>	<i>Device rejected installing image.</i>
		<i>error</i>	<i>There is an issue installing the software.</i>
<i>lastinstall</i>	time		Time of last software installation.

Capabilities object attributes			
<i>certified</i>	<i>bool</i>	1.23	This device is Hue certified
<i>streaming</i>	<i>object</i>	1.22	Present of light supports streaming features
<i>renderer</i>	<i>bool</i>	1.22	Indicates if a lamp can be used for entertainment streaming as renderer

proxy	bool	1.22	Indicates if a lamp can be used for entertainment streaming as a proxy node
-------	------	------	---

Top level attributes

type	string	device type of the light
name	string 0..32	Human readable name of the light. Names has to be unique.
modelid	string 6..32	This parameter uniquely identifies the hardware model of the device for the given manufacturer.
productid	string 1..32	Uniquely identifying hardware models for the given manufacturer. Not available for all devices.
swversion	string 8..16	This parameter uniquely identifies the software version running in the hardware.
swconfigid	string 1..16	Uniquely identifying configuration of the software running in the hardware. Not available for all devices.
state	object	The object representing the state of the light. Supported attributes are listed for each supported light class in 5 Supported lights
capabilities	object	Lists capabilities of the light
swupdate	object	
state object attributes		
reachable	bool	Indicates if a lamp can be reached by the smart bridge
on	bool	On/Off state of light. On=True, Off=False.
transitiontime	uint16	Duration of the transition for this API command only in 100 millisecond steps. This is an optional write-only parameter with a default value of 4 (400 milliseconds).
alert	string	<i>Alert effect value. A PUT with other unknown values shall generate error 7. Note the alert resource contains the last alert sent to the light and not its current state. The effect will also stop if any other action is performed changing the light.</i>
		<i>“none” The lamp shall stop performing all alert effects.</i>

		<code>"select"</code>	<i>The lamp shall perform one breathe cycle.</i>
		<code>"lselect"</code>	<i>The lamp shall perform breathe cycles for 15 seconds or until the "alert": "none" command is received.</i>
<code>mode</code>	<code>enum</code>	<code>homeautomation</code>	<i>Device can be controlled by updating /state, light output is reflected in /state</i>

Config object attributes

			The shape of the light (bulb, luminaire)
		<code>classicbulb</code>	Oldschool light bulb shape, used for traditional white bulbs and unknown lights
		<code>sultanbulb</code>	The "Hue" bulb shape, our most recognizable shape. E27/A19/B22 etc
<code>archetype</code>	<code>enum</code>	1.24	
		<code>spotbulb</code>	For all lights with GU/PAR kind of bulb. Typically used in spot like lights
		<code>floodbulb</code>	BR30 ceiling bulbs. Typically found in the US.
		<code>candlebulb</code>	For smaller screw fittings,

		typically used for decorative lights.
huebloom	↳	If product fits the Bloom line
hueiris	↳	If the product fits the Iris line
huego	↳	If the product fits the Go line
huelightstrip	▪	If the product is a lightstrip
hueplay	↳	If the product fits the Play line
vintagebulb		If the product fits the vintagebulb line
christmastree		If the product fits the christmastree line
huecentris		
huelightstriptv		
groundspot	↳	A spot like luminaire that is attached to the ground, e.g. outdoor spot
bollard	↳	A "pillar" like luminaire that is mounted on the floor, e.g. outdoor bollard

tablewash	<ul style="list-style-type: none"> A luminaire that aims to “wash the wall in a certain color”
tableshade	<ul style="list-style-type: none"> Any kind of table-top light that is not a wash or flexible, e.g. hue beyond
floorshade	<ul style="list-style-type: none"> For most floor mounted lights. Primary purpose is often decorative, mostly used indoors.
floorlantern	<ul style="list-style-type: none"> Mostly used outdoors, lights that primarily function to illuminate their surroundings.
flexiblelamp	<ul style="list-style-type: none"> A “pixar” kind of lamp, typically seen in desk lights.
recessedfloor	<ul style="list-style-type: none"> A luminaire that is embedded (flush or semi-flush) into the floor.
walllantern	<ul style="list-style-type: none"> A lantern-type for wall attached luminaires, typically found outdoors.

			For wall attached luminaires that serve a more decorative function.
	wallshade	*	
			A Spot-like type for wall attached luminaires, typically found in transit areas or to highlight objects hanging on the wall.
	wallspot	*	
			A round(ish) luminaire that hangs suspended from the ceiling.
	pendanround	*	
			An elongated luminaire that hangs suspended from the ceiling.
	pendantlong	*	
ceilingground			A round(ish) luminaire that is directly mounted to the ceiling.
ceilingsquare			An square(ish) luminaire that is directly mounted to the ceiling.

singlespot	A luminaire that consists of only one spot kind of shape that is attached to the ceiling.		
doublespot	A luminaire that consists of more than one spot kind of shape that is attached to the ceiling.		
		recessedceiling	A luminaire that is embedded (flush or semi-flush) into the ceiling.
			The major purpose of the light.
function	enum	functional	The major purpose is functional lighting. Examples include ceiling luminaires, spots, white bulbs.
		decorative	The major purpose is decorative lighting with visible impact on space ambiance. Examples

1.24

				include shades, candle, color bulbs.
			mixed	There is no clear major purpose. The use changes depending on the use case and can be functional or decorative. Examples include lantern, desk lamp, lightstrip.
			unknownfunction	Other function (none of the above).
				The major direction of the light.
			omnidirectional	Light is emitted in all directions (omni-directional). Examples include Hue Go, lightstrip, spot, lantern, bollard.
direction	enum	1.24	upwards	Light emits upwards only. Examples include groundspot, recessfloor, Iris, Bloom.
			downwards	Light emits downwards only. Examples include ceiling lamp, wallspot.

			horizontal	Light emits to the sides in at least 2 directions or to all directions.
			vertical	Light emits up and down. Examples include wall lanterns, pendants that also have light on top.
			unknowndirection	Other direction (none of the above).
startup	object	1.26		Object describing the startup behavior of a light.
swupdate object attributes				
state	String	1.20		State of software update for this device.
			<i>notupdatable</i>	<i>System cannot update this device or determine if it is out-of-date.</i>
			<i>noupdates</i>	<i>No update available nor known.</i>
			<i>transferring</i>	<i>Bridge knows there is an update is available. But not yet downloaded from portal or finished transferring to device.</i>
			<i>readytoinstall</i>	<i>Software is ready to install (ie</i>

			<i>transferred to device).</i>
		<i>installing</i>	<i>Software update is installing. Note that the device might not be usable for 30–60s during installation.</i>
		<i>batterylow</i>	<i>Battery is too low for update.</i>
		<i>imagerejected</i>	<i>Device rejected installing image.</i>
		<i>error</i>	<i>There is an issue installing the software.</i>
lastinstall	time	1.20	Time of last software installation.

Capabilities object attributes

certified	<i>bool</i>	1.23	This device is Hue certified
streaming	<i>object</i>	1.22	Present of light supports streaming features
renderer	<i>bool</i>	1.22	Indicates if a lamp can be used for entertainment streaming as renderer
proxy	<i>bool</i>	1.22	Indicates if a lamp can be used for entertainment streaming as a proxy node
control	<i>object</i>	1.24	Control capabilities of light
maxlumen	<i>int</i>	1.24	Maximum lumen output. Might not be reached at all color points.

Startup object attributes

			Mode of the startup behavior
mode		safety	lights go back to Philips “bright light” safety setting (100%)

			brightness @ 2700K)
		powerfail	light keeps the setting when power failed. If light was off it stays off
<i>enum</i>	1.26	lastonstate	light keeps the setting when power failed. If light was off it returns to the last on state
		custom	custom settings defined in custom settings. Will be automatically set when providing “customsettings”.
			Not available for “On/Off Light”
		unknown	custom setting is not supported
		configured	true if the startup settings are committed to the device, false if not.
	<i>bool</i>	1.28	If this attribute is not present (<1.28) the bridge does not ensure the settings are committed.

Top level attributes

type	<i>string</i>	1.0	device type of the light
name	<i>string</i> 0..32	1.0	Human readable name of the light. Names has to be unique.
modelid	<i>string</i> 6..32	1.0	This parameter uniquely identifies the hardware model of the device for the given manufacturer.
productid	<i>string</i> 1..32	1.13	Uniquely identifying hardware models for the given manufacturer. Not available for all devices.
swversion	<i>string</i> 8..16	1.0	This parameter uniquely identifies the software version running in the hardware.
swconfigid	<i>string</i> 1..16	1.13	Uniquely identifying configuration of the software running in the hardware. Not available for all devices.
state	<i>object</i>	1.0	The object representing the state of the light. Supported attributes are listed for each supported light class

			in 5 Supported lights
config	object	1.24	Additional configuration of a light
capabilities	object	1.22	Lists capabilities of the light
swupdate	object	1.20	
state object attributes			
reachable	bool	1.0	Indicates if a lamp can be reached by the smart bridge
on	bool	1.0	On/Off state of light. On=True, Off=False.
transitiontime	uint16	1.0	Duration of the transition for this API command only in 100 millisecond steps. This is an optional write-only parameter with a default value of 4 (400 milliseconds).
		1.0	<i>Alert effect value. A PUT with other unknown values shall generate error 7. Note the alert resource contains the last alert sent to the light and not its current state. The effect will also stop if any other action is performed changing the light.</i>
alert	string		<p><i>“none”</i> <i>The lamp shall stop performing all alert effects.</i></p> <p><i>“select”</i> <i>The lamp shall perform one breathe cycle.</i></p> <p><i>“lselect”</i> <i>The lamp shall perform breathe cycles for 15 seconds or until the “alert”:“none” command is received.</i></p>
mode	enum	1.22	<i>homeautomation</i> <i>Device can be controlled by updating /state, light output is reflected in /state</i>

1.2 Dimmable light

Top level attributes			
state object attributes			
All attributes in 1.1 On/Off light			
<i>bri</i>	<i>uint8</i>	1.0	Brightness value to set lamp to. Note: Brightness represents a scale from minimum brightness the

			lamp is capable of 0 to maximum capable brightness 255. Brightness 0 is not off but the minimum dim level of the device.
control	<i>object</i>	1.24	Control capabilities of light.
	mindimlevel	<i>int</i>	Indication of the dimming capabilities of a light.
startup object attributes			
customsettings	<i>object</i>	1.26	only present if mode is on custom settings. Writing this object with valid attributes changes mode to “custom”. Changing the mode to any other setting than “custom” removes this object
bri	<i>uint8</i>	1.26	Startup brightness. If not present on write, the default is 100% (254) in custom mode

1.3 Color temperature light

type	“Color light”		
state object attributes			
All attributes of 1.1 On/Off light			
colormode	<i>string</i> 2..2	1.0	Indicates the colormode in which the lamp is working, this is the last command type it received. Values are “ct” for Color Temperature.
ct	<i>uint16</i>	1.0	indicates the Mirek color temperature value a lamp shall be set to. The end value is also what shall be returned in the response. If a lamp is not capable to move to the target ct value then it shall still return success but with actual ct value achieved. If ct is incremented outside the capable range it shall remain on the limit value. (2012 connected lamps are capable of 153 (6500K) to 500 (2000K).
control	<i>object</i>	1.24	Control capabilities of light.

ct	<i>object</i>	1.24	
min	<i>int</i>	1.24	Min CT value
max	<i>int</i>	1.24	Max CT value
ct	<i>uint16</i>	1.26	Startup color temperature the light will go to when powered on. Cannot be used in combination with xy. If not present the default is 2700K in custom mode.

1.4 Color light

state object attributes

All attributes of [1.2 Dimmable light](#)

<i>colormode</i>	<i>string</i> 2..2	1.0	Indicates the colormode in which the lamp is working, this is the last command type it received. Values are “hs” for Hue and Saturation, “xy” for XY and “ct” for Color Temperature. This parameter is only present when the lamp supports at least one of the values.
<i>effect</i>	<i>string</i>	1.0	Code for dynamic effect of light. “none” and “colorloop” supported. A PUT with other unknown values shall generate error 7. When a colorloop stops, the light will be set to the same color it was set before the color loop was started. However, the color mode will always be “hs” (Zigbee limitation)
<i>hue</i>	<i>uint16</i>	1.0	Hue value to set lamp to.
<i>sat</i>	<i>uint8</i>	1.0	Saturation value to set lamp to.
<i>xy</i>	<i>list</i> 2..2 of <i>float4</i>	1.0	indicates the xy values a lamp shall move to. First entry in array is x and second entry is y. Both must be provided or incorrect value for attribute is returned. Both x and y must be between 0.0000 and 000. 5 th decimal place is truncated.
<i>control</i>	<i>object</i>	1.24	Control capabilities of light.

colorgamut	<i>Array</i> 3..3 of [float, float]	1.24	Color gamut of color bulb [[red_x_max, red_y_max], [green_x_max, green_y_max], [blue_x_max, blue_y_max]]
colorgamuttype	<i>enum</i>	1.24	A Living colors & lightstrip v1 gamut B Hue generation 1 gamut C Hue full colors gamut other Other/not properly defined gamuts
xy	<i>list</i> 2..2 of float4	1.26	Startup color the light will go to when powered on. Cannot be used in combination with ct. If not present and ct is not supported the default is xy representation of 2700K in custom mode.

1.5 Extended color light

state object attributes

All attributes of [1.3 Color Temperature light](#) and [1.4 Color light](#)

Supported sensors

Hue supports sensors and switches like presences sensors and wall switches. These devices are exposed under Sensors API. This page gives an overview of supported sensors and corresponding attributes.

1. ZigBee sensors

1.1 ZGP Switch (Hue Tap)

type	ZGP Switch
Status	Supported in 1.3
Description	A ZigBee Green Power (ZGP) sensor reporting button presses and optionally releases of one or multiple buttons.
Supported toplevel attributes	
name	Human readable name.

type	“ZGPSwitch”	
modelid	“ZGPSWITCH”, “SWT001”	
manufacturername	The manufacturer name	
uniqueid	Unique id of switch.	
Supported config attributes		
on		
Supported state attributes		
buttonevent	Code of last switch event. Events are guidelines for integration Zigbee-Application. Bridge does not verify/disallow values.	
	Button id	Tap Button
	34	Button 1
	16	Button 2
	17	Button 3
	18	Button 4

1.2 ZLL Switch (Hue Dimmer Switch)

type	ZLLSwitch
Status	Supported in 1.8
Description	A ZigBee (ZLL) sensor reporting button presses and optionally releases of one or multiple buttons.
Supported toplevel attributes	
name	Human readable name.
type	“ZLLSwitch”
modelid	“RWL020” (US) “RWL021” (EU)
manufacturername	“Philips”
productid	Uniquely identifying hardware models for the given manufacturer. Not available for all devices.
uniqueid	Unique id of switch.
swconfigid	Uniquely identifying configuration of the software running in the hardware. Not available for all devices.

Supported config attributes			
on			
battery	Attribute will be available if the ZLL Switch is battery operated.		
alert	<p>Alert resource contains the last alert sent to the device and not its current state.</p> <p>“none” – The device stops performing all alert effects.</p> <p>“select” – The device performs one indication cycle.</p> <p>“lselect” – The device performs indication cycle for 15 seconds or until the “alert:none” command is received.</p>		
reachable			
Supported state attributes			
buttonevent	Code of last switch event. Events are guidelines for integration Zigbee-Application. Bridge does not verify/disallow values.		
	Button	Action	Dimmer Button
	1000	INITIAL_PRESS	Button 1 (ON)
	1001	HOLD	
	1002	SHORT_RELEASED	
	1003	LONG_RELEASED	
	2000	INITIAL_PRESS	Button 2 (DIM UP)
	2001	HOLD	
	2002	SHORT_RELEASED	
	2003	LONG_RELEASED	
	3000	INITIAL_PRESS	Button 3 (DIM DOWN)
	3001	HOLD	
	3002	SHORT_RELEASED	
	3003	LONG_RELEASED	
	4000	INITIAL_PRESS	Button 4 (OFF)
	4001	HOLD	
	4002	SHORT_RELEASED	

	4003	LONG_RELEASED
lastupdated	Last time the sensor state was updated	

1.3 ZLL Presence (Hue Motion Sensor)

type	ZLLPresence
Status	Supported in 1.6
Description	A sensor detecting presence in the vicinity of the sensor. Any change of presence is reported to the bridge.
Supported toplevel attributes	
name	Human readable name.
type	“ZLLPresence”
modelid	“SML001”
manufacturername	“Philips”
productid	Uniquely identifying hardware models for the given manufacturer. Not available for all devices.
uniqueid	Unique id of presence sensor.
swconfigid	Uniquely identifying configuration of the software running in the hardware. Not available for all devices.
Sensor specific state attributes	
presence	True if sensor detects presence.
Supported config attributes	
on	
battery	Attribute will be available if the ZLL Presence is battery operated.
alert	Alert resource contains the last alert sent to the device and not its current state. “none” – The device stops performing all alert effects. “select” – The device performs one indication cycle. “lselect” – The device performs indication cycle for 15 seconds or until the “alert:none” command is received.
reachable	

sensitivity	Sensitivity of the sensor. Value in the range $0..sensitivitymax$.
sensitivitymax	Maximum level for the sensitivity of the sensor. Total number of supported levels are $sensitivitymax+1$
Supported state attributes	
lastupdated	Last time the sensor state was updated

1.4 ZLL Temperature

type	ZLLTemperature
Status	Supported in v1.6.0
Description	A sensor measuring the current ambient temperature.
Supported toplevel attributes	
All toplevel attributes in 3. General Sensor resource	
Sensor specific state attributes	
temperature	1.6 int32 Current temperature in 0.01 degrees Celsius. (3000 is 30.00 degree) Bridge does not verify the range of the value.

2. CLIP sensors

2.1 CLIP Switch

type	CLIPSwitch
Status	Supported in 1.3
Description	A sensor reporting button presses and optionally releases of one or multiple buttons.
Supported toplevel attributes	
All toplevel attributes in 3. General Sensor resource	
Supported config attributes	
Sensor specific state attributes	
buttonevent	1.3 buttonevent code (state)

2.2 CLIP OpenClose

type	CLIPOpenClose
------	---------------

Status	Supported in 1.3
Description	Sensor detecting whether a switch or a contact is open resp. closed. Any change of the binary state is reported to the bridge, while the sensor is required to stay minimal 1s in either state.
Supported toplevel attributes	
All toplevel attributes in 3. General Sensor resource	
Supported config attributes	
on	
reachable	
battery	Present if provided during POST or PUT
url	Present if provided during POST or PUT
Sensor specific state attributes	
open	True if switch is currently open

2.3 CLIP Presence

type	CLIPPresence
Status	Supported in 1.3
Description	A sensor detecting presence in the vicinity of the sensor. Any change of presence is reported to the bridge.
Supported toplevel attributes	
All toplevel attributes in 3. General Sensor resource	
Supported config attributes	
on	
reachable	
battery	Present if provided during POST or PUT
url	Present if provided during POST or PUT
Sensor specific state attributes	
presence	True if sensor detects presence

2.4 CLIP Temperature

type	CLIPTemperature
Status	Supported in 1.3
Description	A sensor measuring the current ambient temperature. The sensor reports the temperature when the delta between the current and the last reported values exceeds a given threshold and the last report has been send at least 10s ago.
Supported toplevel attributes	
All toplevel attributes in 3. General Sensor resource	
Supported config attributes	
on	
reachable	
battery	Present if provided during POST or PUT
url	Present if provided during POST or PUT
Sensor specific state attributes	
temperature	Current temperature in 0.01 degrees Celsius. (3000 is 30.00 degree) Bridge does not verify the range of the value.

2.5 CLIP Humidity

type	CLIPHumidity
Status	Supported in 1.3
Description	A sensor measuring the current ambient humidity.
Supported toplevel attributes	
All toplevel attributes in 3. General Sensor resource	
Supported config attributes	
on	
reachable	
battery	Present if provided during POST or PUT
url	Present if provided during POST or PUT
Sensor specific state attributes	
humidity	Current humidity 0.01% steps (e.g. 2000 is 20%)The bridge does not enforce range/resolution.

2.6 Daylight Sensor

type	Daylight
Status	Supported in 1.3
Description	A sensor indicating the switch between daylight and night for a given location. This sensor is implemented in the bridge and is always available.
Supported toplevel attributes	
All toplevel attributes in 3. General Sensor resource	
Supported config attributes	
on	
long	GPS coordinate longitude in decimal degrees DDD.DDDD{W E} with leading zeros required ending with W or E e.g. 000.3295W “none” . In future versions this may change to null.
	Is not returned anymore due to privacy reasons see “configured” to check if already set.
lat	GPS coordinate latitude in decimal degrees DDD.DDDD{N S} with leading zeros required e.g. 010.5186N ending with N or S “none”.In future versions this may change to null.
	Is not returned anymore due to privacy reasons see “configured” to check if already set.
configured	True if the valid GPS coordinates have been set.
sunriseoffset	Timeoffset in minutes to sunrise. Daylight changes to true sunriseoffset minutes after sunrise. Values: -120..120min, default 30min. In case this cause overlap with sunset, daylight will be constantly: <ul style="list-style-type: none"> • true if next sunrise is moved before sunset • false if next sunrise is moved after sunset
sunsetoffset	Timeoffset in minutes to sunset. Daylight changes to true sunsetoffset minutes after sunset. Values: -120..120min, default -30min. . In case this cause overlap with sunset, daylight will be constantly: <ul style="list-style-type: none"> • false if next sunset is moved before sunrise • true if next sunset is moved after sunrise
Sensor specific state attributes	

daylight	True between sunrise and sunset at the given GPS location
----------	---

2.7 CLIP Lightlevel and ZLL Lightlevel

type	CLIPLightlevel ZLLLightlevel	
Status	Supported in 1.14	
Description	A sensor indicating the ambient light level at the sensor location.	
Supported toplevel attributes		
All toplevel attributes in 3. General Sensor resource		
Supported config attributes		
tholddark	unit16	Threshold the user configured to be used in rules to determine insufficient lightlevel (ie below threshold). Default value 16000
tholdoffset	unit16	Threshold the user configured to be used in rules to determine sufficient lightlevel (ie above threshold). Specified as relative offset to the “dark” threshold. Shall be >=1. Default value 7000
Sensor specific state attributes		
lightlevel	unit16	Light level in $10000 \log_{10}(\text{lux}) + 1$ measured by sensor. Logarithm scale used because the human eye adjusts to light levels and small changes at low lux levels are more noticeable than at high lux levels.
dark	bool	lightlevel is at or below given dark threshold.
daylight	bool	lightlevel is at or above light threshold (dark+offset).

2.7.1 Typical light levels for indoor use cases

Example	Lux	MeasuredValue
Outdoor: Overcast moonless night sky	0.0001	0
Outdoor: Bright moonlight	1	1
Home: Night light	2	3000
Home: Dimmed light	10	10000
Home: ‘Cosy’ living room	50	17000
Home: ‘Normal’ non-task light	150	22000

Home: Working / reading	350	25500
Home: Specialized tasks, Inside daylight	700	28500
Home: Maximum to avoid glare	2000	33000
Outdoor: Clear daylight	> 10000	> 40000
Outdoor: Brightest direct sunlight	120000	51000

2.8 CLIP Generic Flag Sensor

type	CLIPGenericFlag
Status	Supported in 1.3
Description	A generic sensor object for 3 rd party IP sensor use. E.g. the portal can make use of a Generic sensor to indicate IFTTT events. The sensor can be created by POST. Once created, only the created attributes can be updated by PUT.
Supported toplevel attributes	
All toplevel attributes in 3. General Sensor resource	
Supported config attributes	
on	
reachable	
battery	Present if provided during POST or PUT
url	Present if provided during POST or PUT
Sensor specific state attributes	
flag	Boolean flag indicating sensor state

2.9 CLIP Generic Status Sensor

type	CLIPGenericStatus
Status	Supported in 1.3
Description	A generic sensor object for 3 rd party IP sensor use. E.g. the portal can make use of a Generic sensor to indicate IFTTT events. The sensor can be created by POST. Once created, only the created attributes can be updated by PUT.
Supported toplevel attributes	

All toplevel attributes in <i>3. General Sensor resource</i>	
Supported config attributes	
on	
reachable	
battery	Present if provided during POST or PUT
url	Present if provided during POST or PUT
Sensor specific state attributes	
status	Integer field indicating sensor state

3. General Sensor Resource

Top level attributes		
name	string 0..32	The human readable name of the sensor, can be changed by the user. Is not allowed to be empty on change.
type	string 0..32	Type name of the sensor.
modelid	string 6..32	This parameter uniquely identifies the hardware model of the device for the given manufacturer.
manufacturername	<i>string</i> 6..32	The name of the device manufacturer (Zigbee sensor manufacturer name, resp. IP device manufacturer)
uniqueid	string 6..32	Unique id of sensor. Should be the MAC address of the device. Extended with an unique endpoint id in form AA:BB:CC:DD:EE:FF:00:11-XX-YYYY with -YYYY optional With AA,BB,.. hex numbers of mac address. For CLIP sensors, XX should be used to distinguish multiple sensors on one device. If MAC address of the device is not available a random number of this length should be used.
swversion	string 1..16	This parameter uniquely identifies the software version running in the hardware. Optional for CLIP Sensors. Not applicable for ZGP devices
state	object	The state object with attributes corresponding to the sensor type Attribute values represents initial state of sensor after creation

config object The configuration object with attributes corresponding to the sensor type. Attribute values represents configuration information.

config object attributes

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
battery	uint8	The current battery state in percent, only for battery powered devices. Not present when not provided on creation or modification (CLIP sensors).
alert	string	Alert effect value. A PUT with unknown values generates error 7. Note the alert resource contains the last alert sent to the device and not its current state.
usertest	bool	Activates or extends user usertest mode of device for 120 seconds. False deactivates usertest mode. Returns the value last sent to or received from the device. In usertest mode, sensors report changes in state faster and indicate state changes on device LED (if applicable).

url	string 0..64	Optional URL of the CLIP sensor. Not present when not provided on creation or modification.
pending	List 0..n of string	<p>Array of config parameters which is not yet committed to sensor. As long as the attribute is listed here, the configuration attribute value listed on GET does not take effect and might return to previous value. A subsequent PUT on listed attribute might return error 10.</p> <p>Example:</p> <pre>{“sensitivity”:3,“pending”:[“sensitivity”]}</pre> <p>(sensitivity 3 is been written on a sensor resource but not yet committed to device)</p>
ledindication	bool	<p>Turns device LED during normal operation on or off. Devices might still indicate exceptional operation (Reset, SW Update, Battery Low)</p> <p>Optional, only used for ZLL sensors.</p>
state object attributes		
<sensor_state>	<data type>	Sensor specific. Can be one or multiple attributes. Can be written over CLIP only for CLIP sensors
lastupdated	time	Last time (based on /config/utc) the sensor send state data reflected in the state field. No value change is required to update the field. “none” (asof 1.x.0 <i>null</i>) when not initialized/no recent

		update has been received since the last bridge power cycle
capabilities object attributes		
certified	bool	This device is Hue certified
primary	bool	This sensor resource represents the device. In case of a combined sensors (same MAC address portion in uniqueID), name and device information should be used from the resource with “primary”:true.

Multisource luminaires

Multisource luminaires are luminaires with multiple light modules. The bridge uses the groups and lights resources to store multisource luminaires.

When a MSL is added the bridge automatically creates a group of type “Luminaire” which stores all the Light Identifiers which form part of the luminaire, groups of type “Lightsource” which stores the different light sections (of the luminaire) and the lights which are part of the luminaire.

Note: You can synchronously control the Luminaire/Lightsource by sending group commands.

An example configuration for a Hue Beyond is shown below. A Hue Beyond contains 2 independent light sources, an uplight and a downlight, with the uplight containing 1 light, and the downlight containing 3 lights.

Example MSL groups configuration

```
{
  "2": {
    "name": "Hue Beyond 2",
    "lights": [
      "4",
      "6",
      "7",
      "11"
    ],
    "type": "Luminaire",
    "modelid": "HBL002",
    "action": {
      "on": false,
      "bri": 0,

```

```
        "hue": 0,
        "sat": 0,
        "effect": "none",
        "xy": [
            0,
            0
        ],
        "ct": 0,
        "alert": "none",
        "colormode": "hs"
    }
},
"3": {
    "name": "Hue Beyond Up",
    "lights": [
        "4"
    ],
    "type": "LightSource",
    "action": {
        "on": false,
        "bri": 0,
        "hue": 0,
        "sat": 0,
        "effect": "none",
        "xy": [
            0,
            0
        ],
        "ct": 0,
        "alert": "none",
        "colormode": "hs"
    }
},
"4": {
    "name": "Hue Beyond Down",
    "lights": [
        "6",
        "7",
        "11"
    ],
    "type": "LightSource",
    "action": {
        "on": false,
        "bri": 0,
        "hue": 0,
        "sat": 0,
        "effect": "none",
        "xy": [
            0,
            0
        ],
        "ct": 0,
        "alert": "none",
        "colormode": "hs"
    }
}
}
```

Note that if the MSL has recently been added to the bridge, some

light identifiers can be null if they are not yet available. e.g.

```
"lights": [  
  "4",  
  "6",  
  null,  
  "11"  
]
```

Example MSL lights configuration

```
{  
  "4": {  
    "state": {  
      "on": false,  
      "bri": 0,  
      "hue": 0,  
      "sat": 0,  
      "effect": "none",  
      "xy": [  
        0,  
        0  
      ],  
      "ct": 0,  
      "alert": "none",  
      "colormode": "hs",  
      "reachable": false  
    },  
    "type": "Extended color light",  
    "name": "Hue Beyond Up 2",  
    "modelid": "LLM001",  
    "uniqueid": "00:17:88:01:00:b6:3f:b5-0b",  
    "swversion": ""  
  },  
  "6": {  
    "state": {  
      "on": false,  
      "bri": 0,  
      "hue": 0,  
      "sat": 0,  
      "effect": "none",  
      "xy": [  
        0,  
        0  
      ],  
      "ct": 0,  
      "alert": "none",  
      "colormode": "hs",  
      "reachable": false  
    },  
    "type": "Extended color light",  
    "name": "Hue Beyond Down 1",  
    "modelid": "LLM001",  
    "uniqueid": "00:17:88:01:00:b6:3f:ed-0b",  
    "swversion": ""  
  },  
  "7": {  
    "state": {  
      "on": false,  

```

```

        "bri": 0,
        "hue": 0,
        "sat": 0,
        "effect": "none",
        "xy": [
            0,
            0
        ],
        "ct": 0,
        "alert": "none",
        "colormode": "hs",
        "reachable": false
    },
    "type": "Extended color light",
    "name": "Hue Beyond Down 2",
    "modelid": "LLM001",
    "uniqueid": "00:17:88:01:00:60:0a:d3-0b",
    "swversion": ""
},
"11": {
  "state": {
    "on": false,
    "bri": 0,
    "hue": 0,
    "sat": 0,
    "effect": "none",
    "xy": [
      0,
      0
    ],
    "ct": 0,
    "alert": "none",
    "colormode": "hs",
    "reachable": false
  },
  "type": "Extended color light",
  "name": "Hue Beyond Down 3",
  "modelid": "LLM001",
  "uniqueid": "00:17:88:01:00:b9:c3:06-0b",
  "swversion": ""
}
}

```

4. Switches

4.1 General Switch Resource

All toplevel attributes in General Sensor resource.			
diversityid	<i>UUID</i>	1.30	An UUID string used by Hue app to fetch branded UI information from the Hue cloud.

inputs	<i>array of input objects</i>	1.30	<p>An array containing input definitions relevant for the sensor.</p> <p>Supported Types: ZLLRelativeRotary, ZLLSwitch, ZGPSwitch</p>
events	<i>array of event objects</i>	1.30	<p>Object containing list of events supported by the input.</p> <p>Content of the events objects is depending on the input type.</p>
archetype	<i>string</i>	1.31	<p>The default function of the input. If there is printing on the physical input, this can be reflected as the function.</p> <p>Currently the following functions are defined:</p>
			<p>none, dim, on_off, on, off, dim_up, dim_down, up, down, up_down, scene, dots</p>
repeatintervals	<i>array of uint</i>	1.30	<p>If an input supports repeating of “holding” or “turning” (ie “repeat”) events, it can schedule the next repetition of the event in one of the given timeintervals</p> <p>(events are e.g. “initial_press” for</p>

			<p>button, “repeat” for rotary.)</p> <p>e.g. Empty array means input supports no repeat intervals.</p> <p>e.g. [300, 800, 1500] means that the device can switch between 300, 800 and 1500 ms interval for next expected event.</p>
--	--	--	---

4.2 Button Inputs (ZGPSwitch, ZLLSwitch)

event object attributes (sensor type = “ZLLZGPSwitch” or “ZLLSwitch”)			
buttonevent	<i>uint16</i>	1.31	number of a button event which can be send by the switch
eventtype	<i>string</i>	1.31	<p>User interaction triggering this button event.</p> <p>Currently the following event types are defined for type button:</p> <p>initial_press, repeat, short_release, long_release, double_short_release</p>

4.3 Rotary Inputs (ZLLRelativeRotary)

event object attributes (sensor type = “ZLLRelativeRotary”)			
rotaryevent	<i>uint16</i>	1.30	number of a rotary event which can be send by
eventtype	<i>string</i>	1.30	<p>User interaction triggering this rotary event.</p> <p>Currently the following event types are defined for type rotary:</p> <p>START, STOP, REPEAT</p>

4.4 ZLL Relative Rotary

Supported as of 1.30

A sensor reporting rotary dial actions of a single rotary element on a switch. A device can contain multiple rotary elements and multiple buttons (the latter represented in a ZLLSwitch) Elements belonging to the same physical device can be identified through unique ID.

All toplevel attributes in *General Switch Resource*

rotaryevent	<i>uint8</i>	1.30	Parameter with information on last rotary action, whether it is an event that marks the start or stop of the rotation action, or if it is a repeat message that gives updates during the rotation action.
expectedrotation	<i>int16</i>	1.30	Expected Rotation in the time until the next expected repeat command. In permille of a full rotation. Positive value is clockwise rotation, negative value is counter clockwise rotation.
expectedeventduration	<i>uint16</i>	1.30	Time milliseconds until the next expected repeat command.

```
{
  "1": {
    "state":{
      "rotaryevent": 2,
      "expectedrotation": 800,
      "expectedeventduration":300,
      "lastupdated":"2013-03-25T13:32:34",
    },
    "name": "Rotary 1",
    "modelid":"WR0001",
    "uniqueid":"01:23:45:67:89:AB-12",
    "manufacturername": "Philips",
    "swversion":"1.0",
    "type": "ZLLRelativeRotary"
    "diversityid": "2c3a75ff-55c4-4e4d-8c44-82d330b8eb9b",
    "capabilities": {
      "certified": true,
      "inputs": [
        {
          "id" : 1,
          "archetype": "on",
          "repeatintervals" : [ 300,700 ],

```

```
    "events": [  
      {"rotaryevent ":0, "eventtype":"start" },  
      {"rotaryevent ": 1, "eventtype":"stop" },  
      {"rotaryevent ": 2, "eventtype":"repeat" }  
    ]  
  }  
}}
```



Connect with us



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

©2018-2019 Signify Holding. All rights reserved.