



# How to **develop for Hue?**

Develop

Get Started

Application Design Guidance

**Hue API**

Hue Entertainment

Tools and SDKs

Overview

## 1. Lights 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

### 1.1. Get all lights

URL	<code>/api/&lt;username&gt;/lights</code>
Method	<code>GET</code>
Version	1.0
Permission	Whitelist

#### 1.1.1. Description

Gets a list of all lights that have been discovered by the bridge.

#### 1.1.2. Response

Returns a list of all lights in the system.

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

#### 1.1.3. Sample Response

```
{
  "1": {
    "state": {
      "on": false,

```

On this page:

[1.1. Get all lights](#)

[1.2. Get new lights](#)

[1.3. Search for new lights](#)

[1.4. Get light attributes and state](#)

[1.5. Set light attributes \(rename\)](#)

[1.6. Set light state](#)

[1.7. Delete lights](#)

Remote API Quick start guide	"bri": 1, "hue": 33761, "sat": 254, "effect": "none", "xy": [ 0.3171, 0.3366 ], "ct": 159, "alert": "none", "colormode": "xy", "mode": "homeautomation", "reachable": true
Remote Authentication	}, "swupdate": { "state": "noupdates", "lastinstall": "2018-01-02T19:24:20"
Remote Hue API - Error Messages	}, "type": "Extended color light", "name": "Hue color lamp 7", "modelid": "LCT007", "manufacturername": "Philips", "productname": "Hue color lamp", "capabilities": { "certified": true, "control": { "mindimlevel": 5000, "maxlumen": 600, "colorgamuttype": "B", "colorgamut": [ [ 0.675, 0.322 ], [ 0.409, 0.518 ], [ 0.167, 0.04 ] ], "ct": { "min": 153, "max": 500 } } "streaming": { "renderer": true, "proxy": false } }, "config": { "archetype": "sultanbulb", "function": "mixed", "direction": "omnidirectional"
Error messages	}, "uniqueid": "00:17:88:01:00:bd:c7:b9-0b",
Message Structure and Response	
Supported Devices	
API Documentation Changelog	
Glossary terms	

```

    "swversion": "5.105.0.21169"
  }
}

```

## 1.2. Get new lights

URL	<code>/api/&lt;username&gt;/lights/new</code>
Method	<code>GET</code>
Version	1.0
Permission	Whitelist

### 1.2.1. Description

Gets a list of lights that were discovered the last time a search for new lights was performed. The list of new lights is always deleted when a new search is started.

### 1.2.2. Response

Name	Type	Description
Lastscan	string	Returns "active" if a scan is currently on-going, "none" if a scan has not been performed since the bridge was powered on, or else the date and time that the last scan was completed in ISO 8601:2004 format <code>(YYYY-MM-DDThh:mm:ss)</code> .

### 1.2.3. Sample Response

```

{
  "7": {"name": "Hue Lamp 7"},
  "8": {"name": "Hue Lamp 8"},
  "lastscan": "2012-10-29T12:00:00"
}

```

## 1.3. Search for new lights

URL	<code>/api/&lt;username&gt;/lights</code>
Method	<code>POST</code>
Version	1.0

Permission	Whitelist
------------	-----------

### 1.3.1. Description

Starts searching for new lights.

The bridge will open the network for 40s. The overall search might take longer since the configuration of (multiple) new devices can take longer. If many devices are found the command will have to be issued a second time after discovery time has elapsed. If the command is received again during search the search will continue for at least an additional 40s.

When the search has finished, new lights will be available using the get new lights command. In addition, the new lights will now be available by calling get all lights or by calling get group attributes on group 0. Group 0 is a special group that cannot be deleted and will always contain all lights known by the bridge.

### 1.3.2. Sample Body (Optional)

```
{ "deviceid": ["45AF34", "543636", "34AFBE"] }
```

Note : The maximum number of serial numbers in any request is 10.

### 1.3.3. Response

Contains a list with a single item that details whether the search started successfully.

### 1.3.4. Sample Response

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

## 1.4. Get light attributes and state

URL	<code>/api/&lt;username&gt;/lights/&lt;id&gt;</code>
Method	<code>GET</code>
Version	1.0
Permission	Whitelist

### 1.4.1. Description

Gets the attributes and state of a given light.

### 1.4.2. Response

Name	Description
------	-------------

state	Details the state of the light, see the state table below for more details.
type	A fixed name describing the type of light e.g. "Extended color light".
name	A unique, editable name given to the light.
modelid	The hardware model of the light.
uniqueid	Unique id of the device. The MAC address of the device with a unique endpoint id in the form: AA:BB:CC:DD:EE:FF:00:11-XX
manufacturername	The manufacturer name.
luminaireuniqueid	Unique ID of the luminaire the light is a part of in the format: AA:BB:CC:DD-XX-YY. AA:BB;, ... represents the hex of the luminaireid, XX the lightsource position (incremental but may contain gaps) and YY the lightpoint position (index of light in luminaire group). A gap in the lightpoint position indicates an incomplete luminaire (light search required to discover missing light points in this case).
streaming	Current light supports streaming features
renderer	Indicates if a lamp can be used for entertainment streaming as renderer
proxy	Indicates if a lamp can be used for entertainment streaming as a proxy node
swversion	An identifier for the software version running on the light.

The state object contains the following fields:

Name	Type	Description
on	bool	On/Off state of the light. On=true, Off=false

bri	uint8	<p>Brightness of the light. This is a scale from the minimum brightness the light is capable of, 1, to the maximum capable brightness, 254.</p>
hue	uint16	<p>Hue of the light. This is a wrapping value between 0 and 65535. Note, that hue/sat values are hardware dependent which means that programming two devices with the same value does not guarantee that they will be the same color. Programming 0 and 65535 would mean that the light will resemble the color red, 21845 for green and 43690 for blue.</p>
sat	uint8	<p>Saturation of the light. 254 is the most saturated (colored) and 0 is the least saturated (white).</p> <p>The x and y coordinates of a color in CIE color space.</p> <p>The first entry is the x coordinate and the second entry is the y coordinate. Both x and y are between 0 and 1. Using CIE xy, the colors can be the same on all lamps if the coordinates are within every lamps gamuts (example: "xy":[0.409,0.5179] is the same color on all lamps). If not, the lamp will calculate it's closest color and use that. The CIE xy color is absolute, independent from the hardware.</p>
xy	list 2..2 of float 4	

ct	uint16	<p>The Mired Color temperature of the light. 2012 connected lights are capable of 153 (6500K) to 500 (2000K).</p> <p>The alert effect, which is a temporary change to the bulb's state. This can take one of the following values:</p> <p>“none” – The light is not performing an alert effect.</p> <p>“select” – The light is performing one breathe cycle.</p> <p>“lselect” – The light is performing breathe cycles for 15 seconds or until an <code>"alert": "none"</code> command is received. Note that this contains the last alert sent to the light and not its current state. i.e. After the breathe cycle has finished the bridge does not reset the alert to “none”.</p>
alert	string	
effect	string	<p>The dynamic effect of the light, can either be “none” or “colorloop”. If set to colorloop, the light will cycle through all hues using the current brightness and saturation settings.</p>
colormode	string 2, 2	<p>Indicates the color mode in which the light 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 light supports at least one of the values.</p>

reachable	bool	Indicates if a light can be reached by the bridge.
-----------	------	--

### 1.4.3. Sample Response

```
{
  "state": {
    "hue": 50000,
    "on": true,
    "effect": "none",
    "alert": "none",
    "bri": 200,
    "sat": 200,
    "ct": 500,
    "xy": [0.5, 0.5],
    "reachable": true,
    "colormode": "hs"
  },
  "type": "Living Colors",
  "name": "LC 1",
  "modelid": "LC0015",
  "swversion": "1.0.3"
}
```

### 1.4.4. Notes

Note the usage of the colormode parameter: There are 3 ways of setting the light color: xy, color temperature (ct) or hue and saturation (hs). A light may contain different settings for xy, ct and hs, but only the mode indicated by the colormode parameter will be certain to give the active light color.

Also note that some light state attributes are only present for specific light types. See supported lights link for more information or supported lights json link for JSON response examples.

## 1.5. Set light attributes (rename)

URL	<code>/api/&lt;username&gt;/lights/&lt;id&gt;</code>
Method	<code>PUT</code>
Version	1.0
Permission	Whitelist

### 1.5.1. Description

Used to rename lights. A light can have its name changed when in any state, including when it is unreachable or off.

### 1.5.2. Body arguments

Name	Type	Description	
name	string 0, 32	The new name for the light. If the name is already taken a space and number will be appended by the bridge e.g. "Bedroom Light 1".	Required

### 1.5.3. Body

```
{"name": "Bedroom Light"}
```

### 1.5.4. Response

A response to a successful **PUT** request contains confirmation of the arguments passed in. Note: If the new value is too large to return in the response due to internal memory constraints then a value of "Updated." is returned.

### 1.5.5. Sample Response

```
[{"success": {"/lights/1/name": "Bedroom Light"}}]
```

## 1.6. Set light state

URL	<code>/api/&lt;username&gt;/lights/&lt;id&gt;/state</code>
Method	<b>PUT</b>
Version	1.0
Permission	Whitelist

### 1.6.1. Description

Allows the user to turn the light on and off, modify the hue and effects.

### 1.6.2. Body arguments

Name	Type	Description	
on	bool	On/Off state of the light. On=true, Off=false	Optional
bri	uint8	The brightness value to set the	Optional

		<p>light to. Brightness is a scale from 1 (the minimum the light is capable of) to 254 (the maximum).</p> <p>Note : a brightness of 1 is not off.</p> <p>e.g. "brightness": 60 will set the light to a specific brightness</p>	
hue	uint16	<p>The hue value to set light to. The hue value is a wrapping value between 0 and 65535. Both 0 and 65535 are red, 25500 is green and 46920 is blue.</p> <p>e.g. "hue": 50000 will set the light to a specific hue.</p>	Optional
sat	uint8	<p>Saturation of the light. 254 is the most saturated (colored) and 0 is the least saturated (white).</p>	Optional
xy	list 2..2 of float 4	<p>The x and y coordinates of a color in CIE color space. The first entry is the x coordinate and the second entry is the y coordinate. Both x and y must be between 0 and 1. If the specified coordinates are not in the CIE color space, the closest color to the coordinates will be chosen.</p>	Optional
ct	uint16	<p>The Mired color temperature of the light. 2012 connected lights are capable of 153 (6500K) to 500 (2000K).</p>	Optional
alert	string	<p>The alert effect, is a temporary change to the bulb's state, and has one of the following values:</p> <ul style="list-style-type: none"> <li>"none" - The light is not performing an alert effect.</li> <li>"select" - The light is performing one breathe cycle.</li> <li>"lselect" - The light is performing breathe cycles for 15 seconds or until</li> </ul>	Optional

		<p>an <code>"alert": "none"</code> command is received. Note that this contains the last alert sent to the light and not its current state. i.e. After the breathe cycle has finished the bridge does not reset the alert to "none".</p>	
effect	string	<p>The dynamic effect of the light. Currently "none" and "colorloop" are supported. Other values will generate an error of type 7. Setting the effect to colorloop will cycle through all hues using the current brightness and saturation settings.</p>	Optional
transitiontime	uint16	<p>The duration of the transition from the light's current state to the new state. This is given as a multiple of 100ms and defaults to 4 (400ms). For example, setting <code>transitiontime:10</code> will make the transition last 1 second.</p>	Optional
bri_inc	-254 to 254	<p>Increments or decrements the value of the brightness. <code>bri_inc</code> is ignored if the <code>bri</code> attribute is provided. Any ongoing <code>bri</code> transition is stopped. Setting a value of 0 also stops any ongoing transition. The bridge will return the <code>bri</code> value after the increment is performed.</p>	Optional
sat_inc	-254 to 254	<p>Increments or decrements the value of the sat. <code>sat_inc</code> is ignored if the <code>sat</code> attribute is provided. Any ongoing <code>sat</code> transition is stopped. Setting a value of 0 also stops any ongoing transition. The bridge will return the <code>sat</code> value after the increment is performed.</p>	Optional
hue_inc	-65534 to 65534	<p>Increments or decrements the value of the hue. <code>hue_inc</code> is ignored if the <code>hue</code> attribute is provided. Any ongoing color transition is stopped. Setting a</p>	Optional

		value of 0 also stops any ongoing transition. The bridge will return the hue value after the increment is performed. Note if the resulting values are < 0 or > 65535 the result is wrapped. For example:	
		<pre>{"hue_inc": 1}</pre>	
		on a hue value of 65535 results in a hue of 0.	
		<pre>{"hue_inc": -2}</pre>	
		on a hue value of 0 results in a hue of 65534.	
		Increments or decrements the value of the ct. ct_inc is ignored if the ct attribute is provided.	
ct_inc	-65534 to 65534	Any ongoing color transition is stopped. Setting a value of 0 also stops any ongoing transition. The bridge will return the ct value after the increment is performed.	Optional
		Increments or decrements the value of the xy. xy_inc is ignored if the xy attribute is provided. Any ongoing color transition is stopped. Setting a value of 0 also stops any ongoing transition. Will stop at it's gamut boundaries. The bridge will return the xy value after the increment is performed. Max value [0.5, 0.5].	
xy_inc	list 2..2 of float 4		Optional

### 1.6.3. Sample Body

```
{
  "hue": 50000,
  "on": true,
  "bri": 200
}
```

### 1.6.4. Response

A response to a successful PUT request contains confirmation of the arguments passed in. Note: If the new value is too large to return in the response due to internal memory constraints then a value of

"Updated." is returned.

### 1.6.5. Sample Response

```
[
  {"success":{"/lights/1/state/bri":200}},
  {"success":{"/lights/1/state/on":true}},
  {"success":{"/lights/1/state/hue":50000}}
]
```

### 1.6.6. Notes

A light cannot have its hue, saturation, brightness, effect, ct or xy modified when it is turned off. Doing so will return 201 error.

There are 3 methods available to set the color of the light - hue and saturation (hs), xy or color temperature (ct). If multiple methods are used then a priority is used: xy > ct > hs. All included parameters will be updated but the 'colormode' will be set using the priority system.

## 1.7. Delete lights

URL	/api/<username>/lights/<id>
Method	DELETE
Version	1.7
Permission	Whitelist

### 1.7.1. Description

Deletes a light from the bridge.



Connect with us



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

©2018-2019 Signify Holding. All rights reserved.

