



How to **develop for Hue?**

Develop

Get Started

Application Design Guidance

Hue API

Hue Entertainment

Tools and SDKs

Overview

2. Groups 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

2.1. Get all groups

URL	<code>/api/<username>/groups</code>
Method	<code>GET</code>
Version	1.0
Permission	Whitelist

2.1.1. Description

Gets a list of all groups that have been added to the bridge. A group is a list of lights that can be created, modified and deleted by a user.

2.1.2. Response

Returns a list of all groups in the system, each group has a name and unique identification number.

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

2.1.3. Sample Response

```
{
  "1": {
    "name": "Group 1",
    "lights": [
      "1",
      "2"
    ]
  }
}
```

On this page:

[2.1. Get all groups](#)

[2.2. Create group](#)

[2.3. Get group attributes](#)

[2.4. Set group attributes](#)

[2.5. Set group state](#)

[2.6. Delete Group](#)

[2.7. General Group Resource](#)

Remote API Quick start guide

Remote Authentication

Remote Hue API - Error Messages

Error messages

Message Structure and Response

Supported Devices

API Documentation Changelog

Glossary terms

```

    ],
    "type": "LightGroup",
    "action": {
      "on": true,
      "bri": 254,
      "hue": 10000,
      "sat": 254,
      "effect": "none",
      "xy": [
        0.5,
        0.5
      ]
    },
    "ct": 250,
    "alert": "select",
    "colormode": "ct"
  }
},
"2": {
  "name": "Group 2",
  "lights": [
    "3",
    "4",
    "5"
  ]
},
"type": "LightGroup",
"action": {
  "on": true,
  "bri": 153,
  "hue": 4345,
  "sat": 254,
  "effect": "none",
  "xy": [
    0.5,
    0.5
  ]
},
"ct": 250,
"alert": "select",
"colormode": "ct"
}
}
}

```

2.1.4. Notes

The following groups are allowed on the bridge:

Group	API version	Description
0	1.0	A special group containing all lights in the system, and is not returned by the 'get all groups' command. This group is not visible, and cannot be created, modified or deleted using the API.
Luminaire	1.4	Multisource luminaire group A lighting installation of default groupings of hue lights. The bridge will pre-install these

		groups for ease of use. This type cannot be created manually. Also, a light can only be in a maximum of one luminaire group. See multisource luminaires for more info.
Lightsource	1.4	LightSource group A group of lights which is created by the bridge based on multisource luminaire attributes of Zigbee light resource.
LightGroup	1.4	LightGroup group A group of lights that can be controlled together. This the default group type that the bridge generates for user created groups. Default type when no type is given on creation.
Room	1.11	Room A group of lights that are physically located in the same place in the house. Rooms behave similar as light groups, except: (1) A room can be empty and contain 0 lights, (2) a light is only allowed in one room and (3) a room isn't automatically deleted when all lights in that room are deleted.
Entertainment	1.22	Represents an entertainment setup Entertainment group describe a group of lights that are used in an entertainment setup. Locations describe the relative position of the lights in an entertainment setup. E.g. for TV the position is relative to the TV. Can be used to configure streaming sessions. Entertainment group behave in a similar way as light groups, with the exception: it can be empty and contain 0 lights. The group is also not automatically recycled when lights are deleted. The group of lights can be controlled together as in LightGroup.
Zone	1.30	Zones describe a group of lights that can be controlled together. Zones can be empty and contain 0 lights. A light is allowed to be in multiple zones.

If the group is of type "Luminaire" then this is the Unique ID of the Luminaire in format AA:BB:CC:DD. If the group is of type "Lightsource" then it has the format AA:BB:CC:DD-XX, where XX is the lightsource position.

Allowed Room classes (case sensitive):

Living room	Kitchen	Dining	Bedroom	Kids bedroom	Bathroom	Nursery
Recreation	Office	Gym	Hallway	Toilet	Front door	Garage
Terrace	Garden	Driveway	Carport	Other		

Support starting [1.30](#)

Home	Downstairs	Upstairs	Top floor	Attic	Guest room	Staircase
Lounge	Man cave	Computer	Studio	Music	TV	Reading
Closet	Storage	Laundry room	Balcony	Porch	Barbecue	Pool

2.2. Create group

URL	/api/<username>/groups
Method	POST
Version	1.0
Permission	Whitelist

2.2.1. Description

Creates a new group containing the lights specified and optional name. A new group is created in the bridge with the next available id.

2.2.2. Sample Body

```
{
  "lights": [
    "1",
    "2"
  ],
  "name": "bedroom",
  "type": "LightGroup"
}
```

Note: For room creation the room class has to be passed, without class it will get the default: "Other" class.

```
{
  "name": "Living room",
  "type": "Room",
  "class": "Living room",
  "lights": [
    "3",
    "4"
  ]
}
```

```
}
}
```

2.2.3. Sample Response

```
[{"success":{"id":"1"}}]
```

2.3. Get group attributes

URL	<code>/api/<username>/groups/<id></code>
Method	<code>GET</code>
Version	1.0
Permission	Whitelist

2.3.1. Description

Gets the group attributes, e.g. name, light membership and last command for a given group.

2.3.2. Response

Name	Type	Description
action	object	The light state of one of the lamps in the group.
lights	array of light IDs	The IDs of the lights that are in the group.
name	string 0, 32	A unique, editable name given to the group.
type	string	If not provided upon creation "LightGroup" is used. Can be "LightGroup", "Room" or either "Luminaire" or "LightSource" if a Multisource Luminaire is present in the system.
modelid	string	Uniquely identifies the hardware model of the luminaire. Only present for automatically created Luminaires.

uniqueid	string	Unique Id in AA:BB:CC:DD format for Luminaire groups or AA:BB:CC:DD-XX format for Lightsource groups, where XX is the lightsource position.
class	string 1, 32	Category of Room types. Default is: Other.

2.3.3. Sample Response

```
{
  "action": {
    "on": true,
    "hue": 0,
    "effect": "none",
    "bri": 100,
    "sat": 100,
    "ct": 500,
    "xy": [0.5, 0.5]
  },
  "lights": [
    "1",
    "2"
  ],
  "state": {"any_on": true, "all_on": true},
  "type": "Room",
  "class": "Bedroom",
  "name": "Master bedroom",
}
```

2.3.4. Notes

“all_on” indicates all lights within the group are ON (true) or OFF (false). “any_on” is true when one or more lights within the group is ON. Otherwise, when all are off, false is returned. 3 common scenarios exist:

1. If all lights within the group are ON, then “all_on” and “any_on” are true.
2. If any light within the group is ON, then “any_on” is true while “all_on” is false.
3. If all lights within the group are OFF, then “all_on” and “any_on” are false.

2.4. Set group attributes

URL	<code>/api/<username>/groups/<id></code>
Method	PUT

Version	1.0
Permission	Whitelist

2.4.1. Description

Allows the user to modify the name, light and class membership of a group.

2.4.2. Body arguments

Name	Type	Description	
name	string 0..32	The new name for the group. If the name is already taken a space and number will be appended by the bridge e.g. "Custom Group 1".	Optional
lights	array of light IDs	The IDs of the lights that should be in the group. This resource must contain an array of at least one element. Each element can appear only once. A light id must be an existing light resource in /lights.If an invalid light ID is given, error 7 will be returned and the group not created.	Optional
class	string 1..32	Category of the Room type. Default is "Other".	

2.4.3. Sample Body

```
{"name":"Bedroom", "lights":["1"]}
```

2.4.4. 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.

2.4.5. Sample Response

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

2.5. Set group state

URL	<code>/api/<username>/groups/<id>/action</code>
Method	PUT
Version	1.0
Permission	Whitelist

Modifies the state of all lights in a group.

User created groups will have an ID of 1 or higher; however a special group with an ID of 0 also exists containing all the lamps known by the bridge.

2.5.2. Body arguments

Name	Type	Description	
on	bool	On/Off state of the light. On=true, Off=false	Optional
bri	uint8	Brightness is a scale from 0 (the minimum the light is capable of) to 254 (the maximum). Note: a brightness of 0 is not off.e.g. "brightness": 60 will set the light to a specific brightness.	Optional

hue	uint16	The hue value is a wrapping value between 0 and 65535. Both 0 and 65535 are red, 25500 is green and 46920 is blue.e.g. "hue": 50000 will set the light to a specific hue.	Optional
sat	uint8	Saturation of the light. 254 is the most saturated (colored) and 0 is the least saturated (white).	Optional
xy	list 2..2 of float 4	The x and y coordinates of a color in CIE color spaceThe 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.	Optional
ct	uint16	The Mired Color temperature of the light. 2012 connected lights are capable of 153 (6500K) to 500 (2000K).	Optional
alert	string	The alert effect, which is a temporary change to the bulb's state, and has one of the	Optional

		<p>following values: "none" - The light is not performing an alert effect. "select" - The light is performing one breathe cycle. "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> <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
effect	string		
transitiontime	uint16	<p>The duration of the transition from the light's current state to</p>	Optional

			<p>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>	
<code>bri_inc</code>	-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
<code>sat_inc</code>	-254 to 254		<p>Increments or decrements the value of the <code>sat</code>. <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. hue_inc is ignored if the hue attribute is provided. Any ongoing color transition is stopped. Setting a 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:</p> <pre> {"hue_inc": 1} </pre> <p>on a hue value of 65535 results in a hue of 0.</p> <pre> {"hue_inc": -2} </pre> <p>on a hue value of 0 results in a hue of 65534.</p>	Optional
ct_inc	-65534 to 65534	<p>Increments or decrements the value of the ct. ct_inc is ignored if the ct attribute is provided. Any ongoing color transition is stopped. Setting a value of 0 also stops any ongoing transition. The</p>	Optional

		bridge will return the ct value after the increment is performed.	
		Increments or decrements the value of the xy. xy_inc is ignored if the xy attribute is provided. Any ongoing color transition is stopped. Will stop at it's gamut boundaries. Setting a value of 0 also stops any ongoing transition. The bridge will return the xy value after the increment is performed.	
xy_inc	-0.5 to 0.5		Optional
scene	string	The scene identifier if the scene you wish to recall.	Optional

2.5.3. Sample Body

```
{
  "on": true,
  "hue": 2000,
  "effect": "colorloop"
}
```

Note: Use group <id> 0 to recall a scene for all lights (which are part of the scene), or use another group <id> if you want to recall the scene for a specific group of lights. E.g. Using group 2 would recall the scene for all lights that are in group 2 AND are part of the specified scene.

```
{
  "scene": "AB34EF5"
}
```

2.5.4. 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.

2.5.5. Sample Response

The response details the success of sending each state parameter to the group. Note that the success is not reported for each light bulb and a "success" for the group does not guarantee that the lights actually changed as they may be unreachable or not capable of the requested change.

```
[
  {"success":{"address": "/groups/1/action/on", "value": true}},
  {"success":{"address": "/groups/1/action/effect",
"value":"colorloop"}},
  {"success":{"address": "/groups/1/action/hue", "value":6000}}
]
```

or for scene recall:

```
[
  {"success":{"address": "/groups/1/action/scene", "value": "AB34EF5"}}
]
```

2.5.6. 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.

2.6. Delete Group

URL	<code>/api/<username>/groups/<id></code>
Method	DELETE
Version	1.0
Permission	Whitelist

2.6.1. Description

Deletes the specified group from the bridge.

2.6.2. Response

The response details whether the group was successfully removed from the bridge.

2.6.3. Sample Response

```
[{
  "success": "/groups/1 deleted."
}]
```

2.6.4. Notes

It is not possible to delete a group of type “LightSource” or “Luminaire” This will return a type 302 error.

2.7. General Group Resource

Top level attributes

name	string 0..32	1.0	Human readable name of the group. If name is not specified one is generated for you (default name is “Group”)
type	string	1.4	Type of the Group. If not provided on creation a “LightGroup” is created. Supported types: <ul style="list-style-type: none"> LightGroup 1.4 Default Luminaire 1.4 multisource luminaire LightSource 1.4 multisource luminaire Room 1.11 Represents a room Entertainment 1.22 Represents an entertainment setup Zone 1.30 Represents a zone
lights	array of light ids	1.0	The ordered set of light ids from the lights which are in the group. This resource shall contain an array of at least one element with the exception of the “Room” type: The Room type may contain an empty lights array. Each element can appear only once. Order of lights on creation is preserved. A light id must be an existing light resource in /lights. If an invalid lights resource is given, error 7 shall be returned and the group is not created. There shall be no

			change in the lights.
			Light id can be <i>null</i> if a group has been automatically create by the bridge and a light source is not yet available
			The ordered set of sensor ids from the sensors which are in the group. The array can be empty.
sensors	array [0..#sensors] of sensor ids	1.27	A sensor id must be an existing sensor resource in /sensors. If an invalid sensor resource is given, error 7 shall be returned and the group is not created.
action	object	1.0	Is used to execute actions on all lights in a group.
state	object	1.12	Contains a state representation of the group
presence	object	1.27	Only exists if sensors array contains a presence sensor of type "ZLLPresence", "CLIPPresence" or "Geofence". This object contains a state object which contains the aggregated state of the sensors
lightlevel	object	1.28	Only exists if sensors array contains a light sensor of type "ZLLLightlevel" or "CLIPLightLevel". This object contains a state object which contains the aggregated state of the sensors
recycle	bool	1.12	When true: Resource is automatically deleted when not referenced anymore in any resource link. Only on creation of resource. "false" when omitted.
presence object attributes			

state	object	1.27	
lastupdated	time	1.27	Last time the combined state was changed
presence	bool	1.27	Any sensor (i.e one or more) in the group detected presence
presence_all	bool	1.27	All sensors in the group detected presence
light level object attributes			
state	object	1.28	
lastupdated	time	1.28	Last time the combined state was updated
dark	bool	1.28	There is not sufficient light in the group (for at least one sensor)
dark_all	bool	1.28	All sensors do not detect sufficient light
daylight	bool	1.28	There is sufficient light in the group (for all sensors)
daylight_any	bool	1.28	Some sensors detect there is sufficient light
lightlevel	int	1.28	Average light level in the group
lightlevel_min	int	1.28	Minimum measured light level
lightlevel_max	int	1.28	Maximum measured light level



Connect with us



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

©2018-2019 Signify Holding. All rights reserved.