PHILIPS

hue PERSONAL WIRELESS LIGHTING

How to **develop for Hue?**

Develop   Get Started   Application Design Guidance   **Hue API**   Hue Entertainment   Tools and SDKs

Overview

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

7   Configuration   API

7.1. Create user

URL	/api
Method	POST
Version	1.0
Permission	All

7.1.1. Description

Creates a new user. The link button on the bridge must be pressed and this command executed within 30 seconds.

Once a new user has been created, the user key is added to a 'whitelist', allowing access to API commands that require a whitelisted user. At present, all other API commands require a whitelisted user.

We ask that published apps use the name of their app as the devicetype.

7.1.2. Body arguments

Name	Type	Description	Required
devicetype	string 0..40	<application_name># <devicename> application_name string 0..20,	Required

On this page:

7.1. Create user

7.2. Get configuration

7.3. Modify configuration

7.4. Delete user from whitelist

7.5. Get full state (datastore)

Name	Type	Description	Required
		devicename string 0..19 (Example: my_hue_app#iphone peter)	
generate clientkey	bool	When set to true, a random 16 byte clientkey is generated and returned in the response. This key is encoded as ASCII hex string of length 32.	Optional

7.1.3. Sample Body

```
{"devicetype": "my_hue_app#iphone peter"}
```

7.1.4. Response

Contains a list with a single item that details whether the user was added successfully along with the username parameter. If successful the username should be stored for future API calls.

7.1.5. Sample Response

```
[{"success":{"username": "83b7780291a6ceffbe0bd049104df"}}]
```

7.1.6. Notes

The link button on the bridge must have been recently pressed for the command to execute successfully. If the link button has not been pressed a 101 error will be returned.

7.2. Get configuration

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

7.2.1. Description

Returns list of all configuration elements in the bridge. Note all times are stored in UTC.

7.2.2. Response

Name	Type	Description

Name	Type	Description
name	string 4..16	Name of the bridge. This is also its uPnP name, so will reflect the actual uPnP name after any conflicts have been resolved.
swupdate	object	Contains information related to software updates. Deprecated in 1.20
swupdate2	object	Contains information related to software updates.
whitelist	object	A list of whitelisted user IDs.
portalstate	object	Object representing the portal state.
apiversion	string	The version of the hue API in the format <major>.<minor>.<patch>, for example 1.2.1
swversion	string	Software version of the bridge.
proxyaddress	string 0..40	No longer available as of `1.21`. IP Address of the proxy server being used. A value of "none" indicates no proxy. as of `1.37` do not allow update anymore. Always returns "none".
proxyport	uint16	No longer available as of `1.21`. Port of the proxy being used by the bridge. If set to 0 then a proxy is not being used. as of `1.37` do not allow update anymore. Always returns 0.
linkbutton	bool	Indicates whether the link button has been pressed within the last 30 seconds. Starting `1.31`, Writing is only allowed for Portal access via cloud application_key.
ipaddress	string	IP address of the bridge.
mac	string	MAC address of the bridge.
netmask	string	Network mask of the bridge.
gateway	string	Gateway IP address of the bridge.
dhcp	bool	Whether the IP address of the bridge is obtained with DHCP.
portalservices	bool	This indicates whether the bridge is registered to synchronize data with a portal account. When setting portalservices to true it shall come with a terms and conditions (www.meethue.com/terms) and privacy notice (www.meethue.com/privacy). A user shall accept it before setting portal services to true.
UTC	string	Current time stored on the bridge.
localtime	string	The local time of the bridge. "none" if not available.

Name	Type	Description
timezone	string 0..32	Timezone of the bridge as OlsenIDs, like "Europe/Amsterdam" or "none" when not configured.
zigbeechannel	uint8	The current wireless frequency channel used by the bridge. It can take values of 11, 15, 20,25 or 0 if undefined (factory new).
modelid	string 6..32	This parameter uniquely identifies the hardware model of the bridge (BSB001, BSB002).
bridgeid	string 16	The unique bridge id. This is currently generated from the bridge Ethernet mac address.
factorynew	bool	Indicates if bridge settings are factory new.
replacesbridgeid	string	If a bridge backup file has been restored on this bridge from a bridge with a different bridgeid, it will indicate that bridge id, otherwise it will be *null*.
datastoreversion	string	The version of the datastore.
starterkitid	string	Name of the starterkit created in the factory.

As of `1.20` returned values of /config/swupdate are deprecated:

Name	Type	Description
checkforupdate	boolean	Setting this flag to true lets the bridge search for software updates in the portal. After the search attempt, this flag is set back to false. Requires portal connection to update server.
devicetypes	object	Details the type of updates available.
updatestate	uint8	can only be updated manually from state 2 into state 3. All other transitions are invalid and will return an error 3. Updating this constitutes acceptance by the app of the software update. 0 No update available. 1 Downloading available updates. 2 Updates are ready to be installed. 3 Installing updates.
notify	bool	Can only be updated when its state is true and it is being set to false. All other transitions are invalid and will return an error 3. is set to true when software has been installed successfully. The app should inform the user that the updates are installed and shall reset this flag to false to avoid multiple user notifications.
url	string	URL to release notes.

Name	Type	Description
text	string	Brief release notes

As of `1.20` returned values of /config/swupdate/devicetypes are deprecated:

Name	Type	Description
bridge	boolean	Update available for the bridge.
lights	array	List of lights to be updated or empty if no update available.
sensors	array	List of sensors to be updated or empty if no update available

Starting `1.20` /config/swupdate2 returns:

Name	Type	Description
bridge	object	Software update object of bridge.
checkforupdate	boolean	Setting this flag true lets the bridge search for software update at the portal. After the search attempt, this flag is reset to false. Requires portal connection to update software. If software update server cannot be reached /config/internetservices/swupdate will be "disconnected".
state	string	State of software update for the system <table><tr><td>unknown</td><td>System does not know if new updates are available (no connection to update server for multiple days).</td></tr><tr><td>noupdates</td><td>No updates available.</td></tr><tr><td>transferring</td><td>Updates are being transferred to devices.</td></tr><tr><td>anyreadytoinstall</td><td>At least one SW update can be installed.</td></tr><tr><td>allreadytoinstall</td><td>All (known) SW updates can be installed.</td></tr><tr><td>installing</td><td>System update is installing. System and/or Devices might not be available for a few minutes.</td></tr></table>
install	bool	Writing "true" triggers installation of software updates when

Name	Type	Description
		in state anyreadytoinstall or allreadytoinstall.
autoinstall	object	Automatic update configuration object.
		Timestamp of last change in system configuration
lastchange	time	• last software configuration update requires additional software to be transferred (noupdates -> transferring) • last successful transfer of a software image to a device
		last successful installation of a software image on a device
lastinstall	time	Time of last software update.

As of `1.20` /config/swupdate2/autoinstall returns:

Name	Type	Description
on	boolean	Indicates if automatic update is activated. Default is false
		T[hh]:[mm]:[ss] Local time of day.
updatetime	Trunctime	The bridge auto. updates for bridge and zigbee devices. The installation time will be randomized between updatetime and updatetime+T01:00:00. Default is T14:00:00.

As of `1.20` /config/internetservices returns:

Name	Type		Description
internet	String	Connected	Bridge has a connection to Internet.
		Disconnected	Bridge cannot reach the Internet.
remoteaccess	String	Connected	If remote CLIP is available.
		Disconnected	If remoteaccess is unavailable, reasons can be portalservices are false or no remote connection is available.
time	String	Connected	Time was synchronized with internet service.
		Disconnected	Internet time service was not reachable for 48hrs.
swupdate	String	Connected	swupdate server is available.
		Disconnected	swupdate server was not reachable in the last 24 hrs.

/config/backup returns:

Name	Type	Description

Name	Type	Description
		Status of backup/restore

idle	R	No backup/restore ongoing.
startmigration	R, W	Can be written in "idle". Creates a backup file which can be retrieved via the backup interface and puts the bridge in "fileready_disabled" state. Indicates that a backup file for migration is being created. CLIP is not available for some time after this command.
fileready_disabled	R	Indicates that a backup file is available and that this bridge has been disabled due to a migration procedure. The bridge can be activated again by a factory reset or power cycle.
prepare_restore	R	Indicates that the a backup file has been sent to the bridge and the bridge is in the process of preparing it for restoring.
restoring	R	Indicates that the bridge is in the process of restoring the backup file.

status — String

Specifies the last error source if the backup has detected an internal error. Cleared at the start of a backup import or export.		
0	None	
1	Export failed	
2	Import failed	

errorcode — int

Note : The Hue Bridge Transfer is done by the Philips Hue App.
Typically, 3rd Party apps should not concern themselves with this.

7.2.3. Sample Response

```
{
    "name": "Philips hue",
    "zigbeechannel": 15,
    "mac": "00:17:88:00:00:00",
    "dhcp": true,
    "ipaddress": "192.168.1.7",
    "netmask": "255.255.255.0",
    "gateway": "192.168.1.1",
    "proxyaddress": "none",
    "proxyport": 0,
    "UTC": "2014-07-17T09:27:35",
    "localtime": "2014-07-17T11:27:35",
```

```json
        "timezone": "Europe/Madrid",
        "whitelist": {
            "ffffffffe0341b1b376a2389376a2389": {
                "last use date": "2014-07-17T07:21:38",
                "create date": "2014-04-08T08:55:10",
                "name": "PhilipsHueAndroidApp#TCT ALCATEL ONE TOU"
            },
            "pAtwdCV8NZId25Gk": {
                "last use date": "2014-05-07T18:28:29",
                "create date": "2014-04-09T17:29:16",
                "name": "MyApplication"
            },
            "gDN3IaPYSYNPWa2H": {
                "last use date": "2014-05-07T09:15:21",
                "create date": "2014-05-07T09:14:38",
                "name": "iPhone Web 1"
            }
        },
        "swversion": "01012917",
        "apiversion": "1.3.0",
        "swupdate": {
            "updatestate": 0,
            "url": "",
            "text": "",
            "notify": false
        },
        "linkbutton": false,
        "portalservices": false,
        "portalconnection": "connected",
        "portalstate": {
            "signedon": true,
            "incoming": false,
            "outgoing": true,
            "communication": "disconnected"
        }
    }
```

swupdate contains additional details:

```json
    "swupdate": {
        "updatestate": 2,
        "checkforupdate": false,
        "devicetypes": {
            "bridge": true,
            "lights": [
                "1",
                "2",
                "3"
            ]
        },
        "url": "",
        "text": "010000000",
        "notify": false
    }
```

As of 1.20 swupdate is deprecated, with swupdate2 contains additional details:

```json
    "swupdate2": {
        "checkforupdate": false,
```

```
                    "lastchange": "2017-06-21T19:44:36",
                    "bridge": {
                        "state": "noupdates",
                        "lastinstall": "2017-06-21T19:44:18"
                    },
                    "state": "noupdates",
                    "autoinstall": {
                        "updatetime": "T14:00:00",
                        "on": false
                    }
                },
                "internetservices": {
                    "internet": "connected",
                    "remoteaccess": "connected",
                    "time": "connected",
                    "swupdate": "connected"
                },
                "factorynew": false,
                "replacesbridgeid": null,
                "backup": {
                    "status": "idle",
                    "errorcode": 0
                }
```

7.3. Modify configuration

URL	`/api/<username>/config`
Method	`PUT`
Version	1.0
Permission	Whitelist

7.3.1. Description

Allows the user to set some configuration values.

7.3.2. Body arguments

Name	Type	Description	Required
proxyport	uint16	Port of the proxy being used by the bridge. If set to 0 then a proxy is not being used.	Optional
name	string 4..16	Name of the bridge. This is also its uPnP name, so will reflect the actual uPnP name after any conflicts have been resolved.	Optional
swupdate	object	Contains information related to software updates.	Optional

Name	Type	Description	Required
proxyaddress	string 0..40	IP Address of the proxy server being used. A value of "none" indicates no proxy.	Optional
linkbutton	bool	Indicates whether the link button has been pressed within the last 30 seconds. Starting `1.31`, Writing is only allowed for Portal access via cloud application_key.	Optional
ipaddress	string	IP address of the bridge.	Optional
netmask	string	Network mask of the bridge.	Optional
gateway	string	Gateway IP address of the bridge.	Optional
dhcp	bool	Whether the IP address of the bridge is obtained with DHCP.	Optional
UTC	string	Current time in UTC. Only modifiable when no internet connection is available to the bridge.	Optional
timezone	string	The bridge timezone.	Optional
touchlink	boolean	Perform a touchlink action if set to true, setting to false is ignored. When set to true a touchlink procedure starts which adds the closest lamp (within range) to the ZigBee network. You can then search for new lights and lamp will show up in the bridge. This field is Write–Only so it is not visible when retrieving the bridge Config JSON.	Optional
zigbeechannel	uint8	The wireless frequency channel used by the bridge. It can take values of 11, 15, 20 or 25.	Optional

7.3.3. Sample Body

```
{"name":"My bridge"}
```

7.3.4. 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.

7.3.5. Example Response

```
[{"success":{"/config/name":"My bridge"}}]
```

7.4. Delete user from whitelist

URL	/api/<application_key>/config/whitelist/<element>
Method	DELETE
Version	1.0
Permission	Whitelist; Starting 1.31.0: Cloud application_key

7.4.1. Description

Deletes the specified user, <element>, from the whitelist.

7.4.2. Response

The response details whether the user was successfully removed from the whitelist.

7.4.3. Sample Response

```
[{
    "success": "/config/whitelist/1234567890 deleted."
}]
```

7.5. Get full state (datastore)

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

7.5.1. Description

This command is used to fetch the entire datastore from the device, including settings and state information for lights, groups, schedules and configuration. It should only be used sparingly as it is resource intensive for the bridge, but is supplied e.g. for synchronization purposes.

7.5.2. Response

Name	Type	Description
lights	object	A collection of all lights and their attributes.
groups	object	A collection of all groups and their attributes.
config	object	All configuration settings.
schedules	object	A collection of all schedules and their attributes.
scenes	object	A collection of all scenes and their attributes.
sensors	object	A collection of all sensors and their attributes.
rules	object	A collection of all rules and their attributes.

7.5.3. Sample Response

```
{
    "lights": {
        "1": {
            "state": {
                "on": false,
                "bri": 0,
                "hue": 0,
                "sat": 0,
                "xy": [0.0000, 0.0000],
                "ct": 0,
                "alert": "none",
                "effect": "none",
                "colormode": "hs",
                "reachable": true
            },
            "type": "Extended color light",
            "name": "Hue Lamp 1",
            "modelid": "LCT001",
            "swversion": "65003148"
        },
        "2": {
            "state": {
                "on": true,
                "bri": 254,
                "hue": 33536,
                "sat": 144,
                "xy": [0.3460, 0.3568],
                "ct": 201,
                "alert": "none",
                "effect": "none",
                "colormode": "hs",
                "reachable": true
            },
            "type": "Extended color light",
            "name": "Hue Lamp 2",
            "modelid": "LCT001",
            "swversion": "65003148"
```

```
            }
        },
        "groups": {
            "1": {
                "action": {
                    "on": true,
                    "bri": 254,
                    "hue": 33536,
                    "sat": 144,
                    "xy": [0.3460, 0.3568],
                    "ct": 201,
                    "effect": "none",
                    "colormode": "xy"
                },
                "lights": ["1", "2"],
                "name": "Group 1"
            }
        },
        "config": {
            "name": "Philips hue",
            "mac": "00:00:88:00:bb:ee",
            "dhcp": true,
            "ipaddress": "192.168.1.74",
            "netmask": "255.255.255.0",
            "gateway": "192.168.1.254",
            "proxyaddress": "",
            "proxyport": 0,
            "UTC": "2012-10-29T12:00:00",
            "whitelist": {
                "1028d66426293e821ecfd9ef1a0731df": {
                    "last use date": "2012-10-29T12:00:00",
                    "create date": "2012-10-29T12:00:00",
                    "name": "test user"
                }
            },
            "swversion": "01003372",
            "swupdate": {
                "updatestate": 0,
                "url": "",
                "text": "",
                "notify": false
            },
            "linkbutton": false,
            "portalservices": false
        },
            "swupdate2": {
                "checkforupdate": false,
                "lastchange": "2017-06-21T19:44:36",
                "bridge": {
                    "state": "noupdates",
                    "lastinstall": "2017-06-21T19:44:18"
                },
                "state": "noupdates",
                "autoinstall": {
                    "updatetime": "T14:00:00",
                    "on": false
                }
            },
        "schedules": {
```

```
        "1": {
            "name": "schedule",
            "description": "",
            "command": {
                "address": "/api/<username>/groups/0/action",
                "body": {
                    "on": true
                },
                "method": "PUT"
            },
            "time": "2012-10-29T12:00:00"
        }
    }
}
```

Connect with us

Contact   Terms & Conditions   Privacy   Product Security