PHILIPS

hue PERSONAL WIRELESS LIGHTING

How to **develop for Hue?**

Get Started

The fastest way to learn how to build apps which control the hue system is to use the simple test web app built into every bridge. This lets you directly input commands and send them to the lights. You can look at the source HTML and JavaScript code for directions on how to do something different.

Follow 3 Easy Steps

Step 1

First make sure your bridge is connected to your network and is functioning properly. Test that the smartphone app can control the lights on the same network.

Step 2

Then you need to discover the IP address of the bridge on your network. You can do this in a few ways.

NOTE – When you are ready to make a production app, you need to discover the bridge automatically using Hue Bridge Discovery Guide .

1. Use a UPnP discovery app to find Philips hue in your network.
2. Use our broker server discover process by visiting https://discovery.meethue.com
3. Log into your wireless router and look Philips hue up in the DHCP table.
4. Hue App method: Download the official Philips hue app. Connect your phone to the network the hue bridge is on. Start the hue app(iOS described here). Push link connect to the bridge. Use the app to find the bridge and try controlling lights. All working — Go to the settings menu in the app. Go to My Bridge. Go to Network settings. Switch off

On this page:

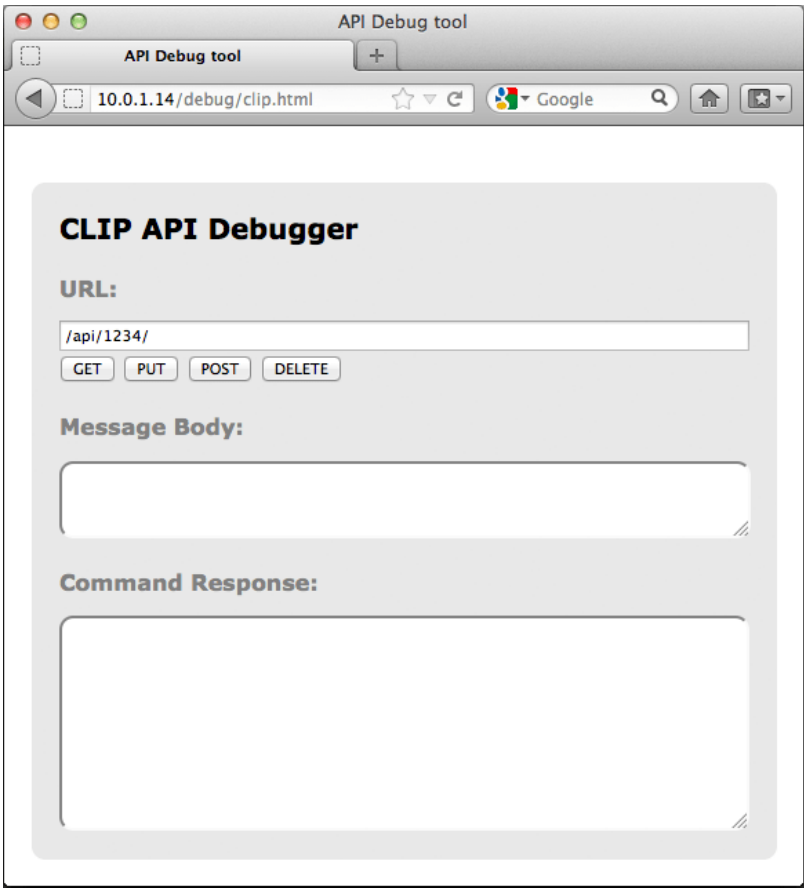
the DHCP toggle. The ip address of the bridge will show. Note the ip address, then switch DHCP back on

Step 3

Once you have the address load the test app by visiting the following address in your web browser.

`https://<bridge ip address>/debug/clip.html`

You should see an interface like this.



Using this debugger utility you can populate the components of an HTTPS call – the basis of all web traffic and of the hue RESTful interface.

1. URL: this is actually the local address of a specific resource (thing) inside the hue system. It could be light, a group of lights or many more things. This is the object you'll be interacting with in this command.

2. A body: this is the part of the message which describes what you want to change and how. Here you enter, in JSON format, the resource name and value you'd like to change/add.
3. A method: here you have a choice of the 4 HTTPS methods the hue call can use.
GET: this is the command to fetch all information about the addressed resource
PUT: this is the command to modify an addressed resource

POST: this is the command to create a new resource inside the addressed resource
DELETE: this is the command to deleted the addressed resource
4. Response: In this area you'll see the response to your command. Also in JSON format.

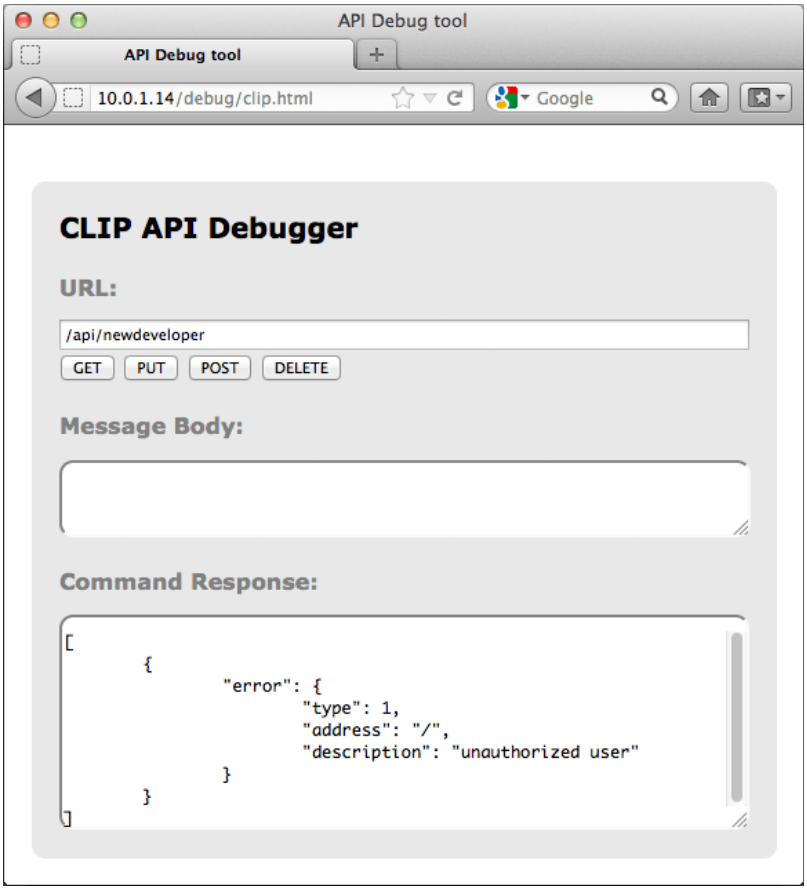
So let's get started...

First let's do a very simple command and get information about your hue system.

Fill in the details below leaving the body box empty and press the `GET` button.

URL	`/api/newdeveloper`
Method	`GET`

You should see a response like below:



Congratulations you've just sent you first CLIP command!

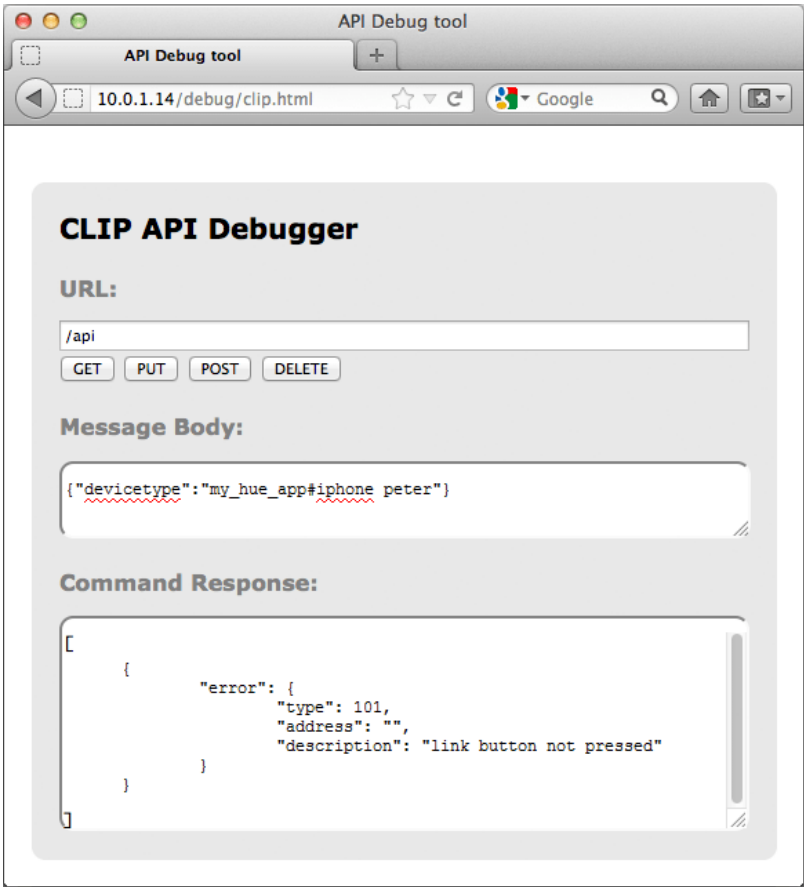
Now this is the command to fetch all information in the bridge. You didn't get much back and that's because you're using an unauthorized username "newdeveloper".

We need to use the randomly generated username that the bridge creates for you. Fill in the info below and press the POST button.

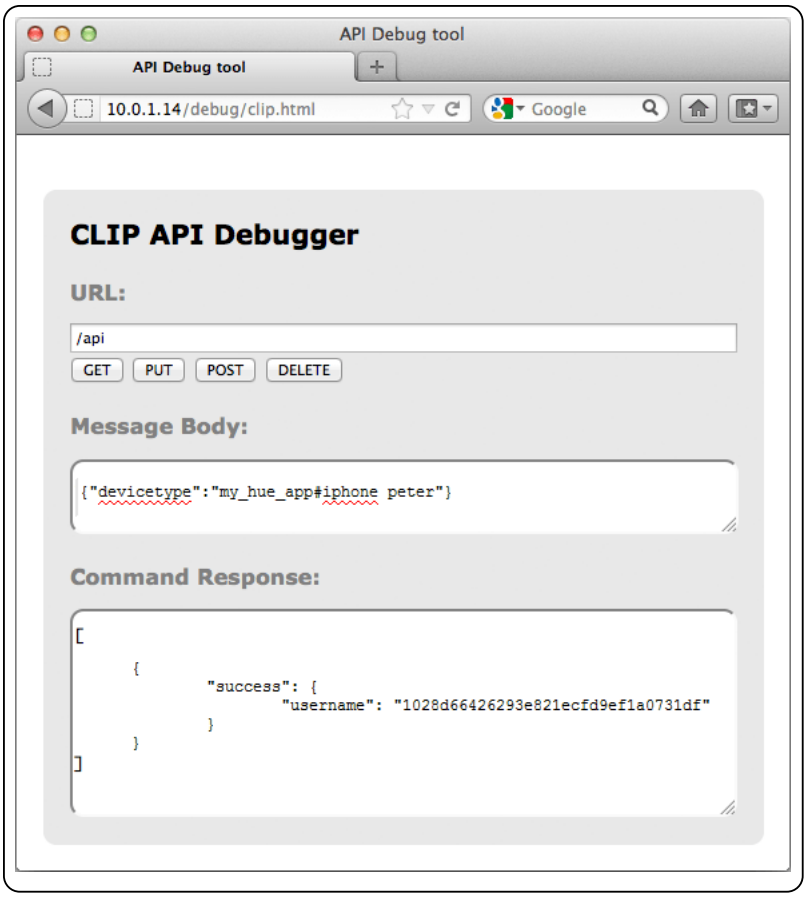
URL	`/api`
Body	`{"devicetype":"my_hue_app#iphone peter"}`
Method	`POST`

This command is basically saying please create a new resource inside /api (where usernames sit) with the following properties.

When you press the POST button you should get back an error message letting you know that you have to press the link button. This is our security step so that only apps you want to control your lights can. By pressing the button we prove that the user has physical access to the bridge.



Go and press the button on the bridge and then press the `POST` button again and you should get a success response like below.



Congratulations you've just created an authorized user `(1028d66426293e821ecfd9ef1a0731df)`, which we'll use from now on! Now if you do the first GET command again you should get a whole lot more information about what lights you have and their states. This data is all in JSON format so can be easily processed by your applications.

Turning a light on and off

Okay now that we have a username with permission to use the system lets start having some fun.

Each light has its own URL. You can see what lights you have with the following command:

Address	`https://<bridge ip address>/api/1028d66426293e821ecfd9ef1a0731df/lights`
Method	`GET`

You should get a JSON response with all the lights in your system and their names.

Now let's get information about a specific light. The light with id 1.

Address	`https://<bridge ip address>/api/1028d66426293e821ecfd9ef1a0731df/lights/1`
Method	`GET`

In this response you can see all of the resources this light has. The most interesting ones are inside the state object as these are the ones we'll have to interact with to control the light.

Lets' start with the "on" attribute. This is a very simple attribute that can have 2 values: true and false. So let's try turning the light off.

Address	`https://<bridge ip address>/api/1028d66426293e821ecfd9ef1a0731df/lights/1/state`
Body	`{"on":false}`
Method	`PUT`

Looking at the command you are sending we're addressing the "state" object of light one and telling it to modify the "on" value inside it to false (or off). When you press the `PUT` button the light should turn off. Change the value in the body to true and the light will turn on again.

Now let's do something a bit more fun and start changing some colors. Enter the command below.

Address	`https://<bridge ip address>/api/1028d66426293e821ecfd9ef1a0731df/lights/1/state`
Body	`{"on":true, "sat":254, "bri":254,"hue":10000}`
Method	`PUT`

We're interacting with the same "state" attributes here but now we're modifying a couple more attributes. We're making sure the light is on by setting the "on" resource to true. We're also making sure the saturation (intensity) of the colors and the brightness is at its maximum by setting the "sat" and "bri" resources to 254. Finally we're telling the system to set the "hue" (a measure of color) to 10000 points (hue runs from 0 to 65535). Try changing the hue value and keep pressing the `PUT` button and see the colour of your light changing running through different colors.

Now you understand the basics of the commands you can send to hue through this tool – but we can also send the commands as part of an app. Intrigued now? Read more at Core Concepts (developer account required).

Connect with us