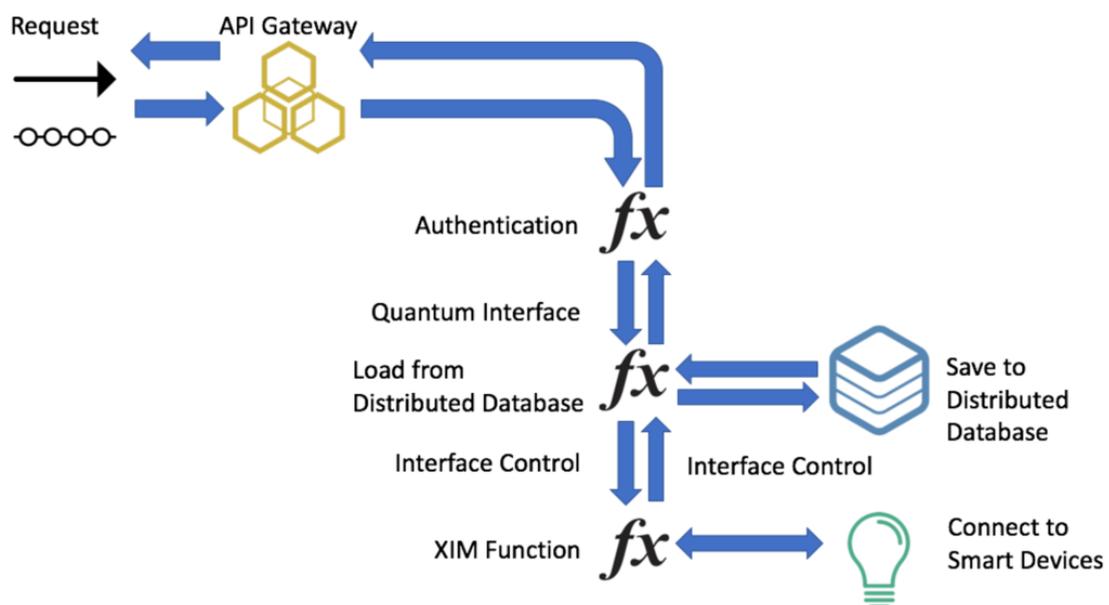


# Quantum XIM Getting Started

This document is for Quantum Developers and Community / V1.0, Aug 9, 2017

## What is Quantum

Quantum is an iPaaS (integration platform as a service) designed for your Smart Home products. With Quantum, you can easily integrate your product with different 3rd party smart home products around the world through a single API call.



## The Benefits of using Quantum

### Easy Integration

You can leave the database and state-control to Quantum and only focus on the controlling of the 3rd party clouds. With Quantum, your product can instantly support all the major Internet smart devices.

### Scalable and Low Latency

Quantum's serverless architecture brings you the best user experience even with a million concurrent requests. No matter how many the concurrent incoming requests are, Quantum gets your result in 100ms ~ 200ms.

## Secure

Quantum is a highly secured platform. Only your user can trigger the service she or he authorized.

## Developers Friendly

Quantum XIM plugins are open source. You can develop your own XIM to support not-yet-supported 3rd party clouds easily.

# Procedures of using Quantum

Quantum access is based on the token that Quantum issues. Once the provisioning sequence is completed, Quantum issues you a token. The token is used for all the access to Quantum.

## The Access Token Owner

Your user is the owner of the token. Whoever owns the token gains the access to the authorized 3<sup>rd</sup> party devices. If your product is on the user's hands, the Quantum token shall be stored in it. If your product is a cloud service, the token shall be managed under the user's account securely.

## Token Time-To-Live

At this moment, Quantum token does not expire. In the future, it will be an option that you can choose the token with a validity time. A refresh token will be needed to renew the token.

## Multiple-Token and Token Revocations

You can obtain multiple tokens for the same user. Each token works equally. You can revoke the issued tokens by deactivating the user. Once a user is deactivated, all previously issued tokens are revoked. To regain the service, you need to obtain another token again.

## App ID and Secret

You will need an app ID/secret to use Quantum. App ID/secret is assigned by Qblinks. Please contact [support@qblinks.com](mailto:support@qblinks.com) to get your app ID/secret. In the future, we will support self-signup/self-manage development console.



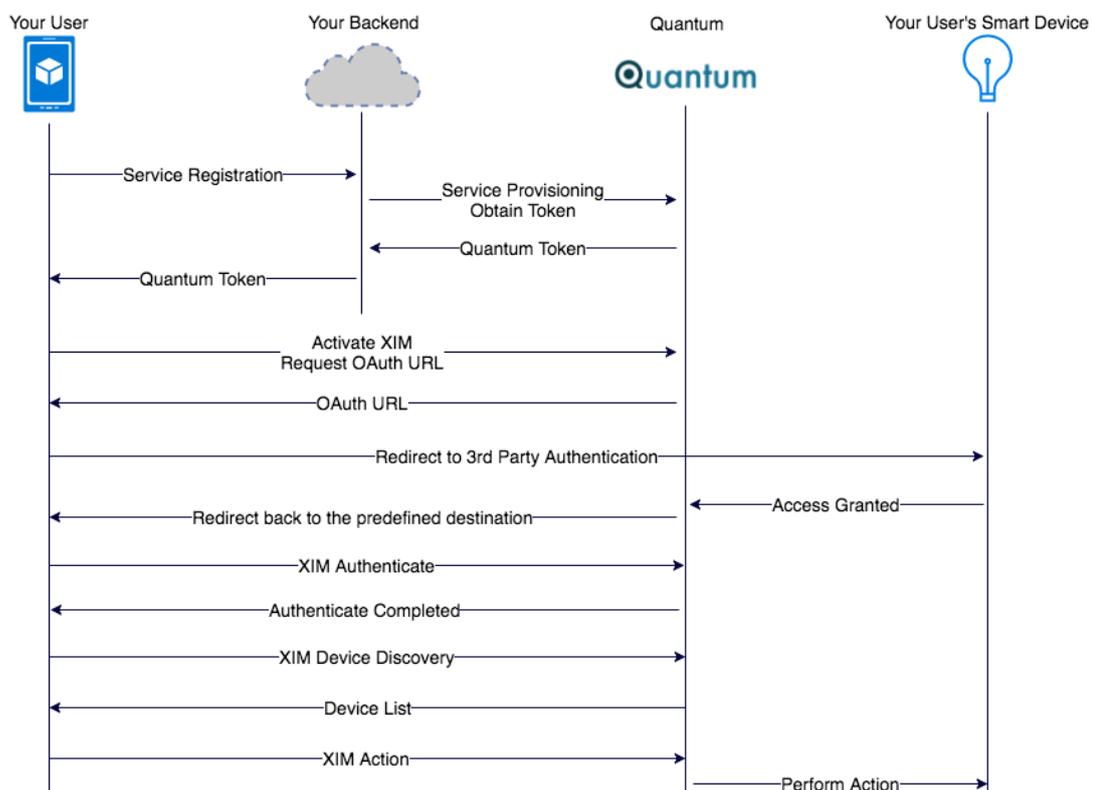
App ID/secret is for the product/service provider only. It is used to perform the provisioning for your users. Never store your app ID/secret outside your backend. Your users are using only the Quantum token to access to Quantum.

### HTTP Access Control Origin for the Application Domain

HTTP Origin is an optional field. With or without HTTP Origin, the Quantum will perform your API requests. However, if you want to use your browser front-end to process your Quantum result, you may need the Application Domain access control. You need to apply your origin with Qblinks. Please contact [support@qblinks.com](mailto:support@qblinks.com) for your allowed origin.

### Procedures of using Quantum

Here is the timing chart of how Quantum works.



## User Provisioning

### Service Provisioning/Obtain Token

GET method

[https://xim.qblinks.com/v1/user/\[user\\_id\]](https://xim.qblinks.com/v1/user/[user_id])



Header

X-Quantum-App-ID: [your\_app\_id]

Authorization: Basic [your\_app\_secret]

Response in json format

```
{
  "result": {
    "err_no": 0,
    "err_msg": "ok"
  },
  "access_token": "[user_quantum_token]",
  "refresh_token": "[user_refresh_token]"
}
```

*Note:*

- 1. refresh token is not currently supported.*
- 2. It may have another err\_no code and err\_msg if there is an error*

## Deactivate Account

DELETE method

[https://xim.qblinks.com/v1/user/\[user\\_id\]](https://xim.qblinks.com/v1/user/[user_id])

Header

X-Quantum-App-ID: [your\_app\_id]

Authorization: Basic [your\_app\_secret]

Response in json format

```
{
  "result": {
    "err_no": 0,
    "err_msg": "ok"
  }
}
```



## XIM Input/Output Structure

All input data is packaged into a JSON format. For example,

```
{
  "xim_type": "light",
  "xim_channel": "lifx",
  "xim_channel_set": 0
}
```

xim\_type, xim\_channel, xim\_channel\_set are mandatory pairs. xim\_type defines the types of the smart devices. The xim\_type determines the action you can perform. xim\_channel is the actual smart device to be chosen. Please use only 0 for xim\_channel\_set.

Once the xim\_type is selected, you will have different JSON pairs you can use as the parameters. Please refer to the properties of the XIM driver you want to use. Please go to <https://github.com/qblinks>

Look into all quantum\_xim drivers. Each driver includes a folder, properties. You can find the properties you can use there. If you want to perform action, check the action properties. If you want to retrieve stat, check the stat properties. The default properties (e.g. xim\_light\_properties.js) is for the output data. The JSON pairs of the output data can only be the subset from the default properties.

## XIM API Calls

All Quantum API calls (other than provisioning API) require some of the following HTTP headers:

(mandatory)	Content-Type: application/json
(mandatory)	Authorization: Bearer [quantum_token]
(optional)	Origin: [your_approved_origin_domain]
(optional)	Accept-Language: [locale, default English]

## XIM Return Message

There are two types of return messages, HTTP response and JSON response. If the Quantum API can be performed, the HTTP response will always 200 regardless



your final XIM request can be performed or not. Its result is always embedded in the responded JSON message. For example,

```
{
  "result": {
    "err_no": 0,
    "err_msg": "ok"
  },
}
```

If there is a response, for example, the /activated API call:

```
{
  "result": {
    "err_no": 0,
    "err_msg": "ok"
  },
  "activated": true
}
```

If there is an error, the error is packaged in the same format. For example,

```
{
  "result": {
    "err_no": 123,
    "err_msg": "xxxxx"
  }
}
```

Note: The "err\_no" will be deprecated in the future. It will be replaced with the error key code such as "err\_code": "quantum.token.invalid".

For the HTTP errors, there are only two possible HTTP errors. 402 Internal Server Error or 408 Timeout. It generally implies there is an error inside the Quantum. If you constantly experience these errors, please contact [support@qblinks.com](mailto:support@qblinks.com)

## Non-Driver API



/activated

/oauthUrl

/unlink

The above APIs are Quantum core level API. You need them to load/unload the drivers.

[/activated](#)

Check if the XIM channel is activated or not.

Response:

“activated”: true            if the channel is activated

“activated”: false        if the channel is not activated

[/unlink](#)

Deactivate the XIM channel

[/oauthUrl](#)

Get the OAuth redirect URL for the user to perform the access grant

Parameter:

"redirect\_url": "url"        URL to go after the user has finished the authorization

Response:

"authorization\_url": "url\_to\_go"    The URL that the user should go

## Channel Authentication

For the first time when the channel is activated that the user grants the access to Quantum, an authentication needs to be performed. This needs to be done only once. Quantum will perform the authentication automatically when the 3<sup>rd</sup> party access needs to be re-established. However, if the action keeps failed, you may need to inform the user to reactivate the channel by unlink-then-activate again. This may happen when the user revokes the access from the 3<sup>rd</sup> party's control panel.

## Step-by-Step Examples

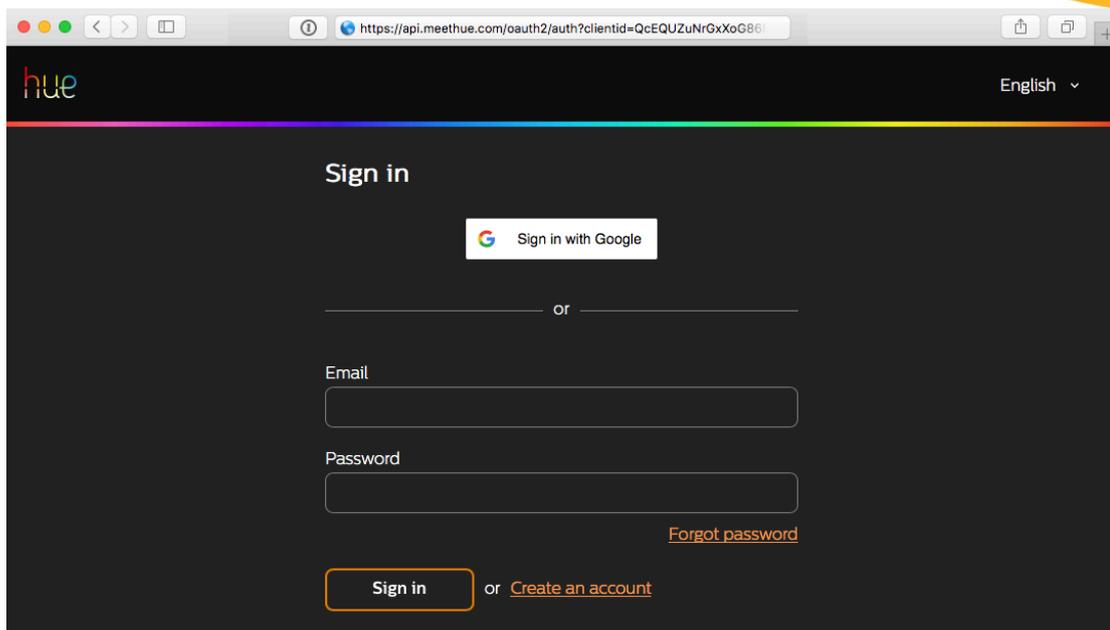
Here is the step by step to use Quantum to control a user's Philips HUE light.

Note1: Some Information is masked out for the security reason.









## 5. Perform Authentication

When the user finishes the authentication, the redirected link you assigned will be loaded. You should perform the authentication once. This is not mandatory. Discovery can perform the authentication automatically, but manually perform the first authentication can speed up the OAuth process.

```
curl -X POST \  
  https://xim.qblinks.com/v1/authenticate \  
  -H 'authorization: Bearer [user_quantum_token]' \  
  -H 'content-type: application/json' \  
  -d '{  
    "xim_type": "light",  
    "xim_channel": "hue",  
    "xim_channel_set": 0  
  }'
```

```
$ curl -X POST https://xim.qblinks.com/v1/authenticate -H 'authorization: Bearer e[redacted]
ciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhcnBfaWQiOiJsZW9sZWUiLCJ1c2VybmFtZSI6InNhbnNvbiU0MHFi
bGlua3MuY29tOm1b2xlZSI6InNjb3BlIjoieGl0IiwiaXNjaXZlbnVlcnVzIjoiMTUzNjAwMDAwLCJp
YXQiOiJlMDI0MjA4NjF9.jRlffEpzVFvis3uTjgP8DkfjVuuC-13CRMxNq3UC-b4' -H 'content-type: application/json' -d '{
{
  "xim_type": "light",
  "xim_channel": "hue",
  "xim_channel_set": 0
}'
{"result":{"err_no":0,"err_msg":"ok"}}$
$
$
```

## 6. Now the channel is activated

```
curl -X POST \
  https://xim.qblinks.com/v1/activated \
  -H 'authorization: Bearer [user_quantum_token]' \
  -H 'content-type: application/json' \
  -d '{
    "xim_type": "light",
    "xim_channel": "hue",
    "xim_channel_set": 0
  }'
```

```
$ curl -X POST https://xim.qblinks.com/v1/activated -H 'authorization: Bearer e[redacted]
iJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhcnBfaWQiOiJsZW9sZWUiLCJ1c2VybmFtZSI6InNhbnNvbiU0MHFi
bGlua3MuY29tOm1b2xlZSI6InNjb3BlIjoieGl0IiwiaXNjaXZlbnVlcnVzIjoiMTUzNjAwMDAwLCJp
YXQiOiJlMDI0MjA4NjF9.jRlffEpzVFvis3uTjgP8DkfjVuuC-13CRMxNq3UC-b4' -H 'content-type: application/json' -d '{
  "xim_type": "light",
  "xim_channel": "hue",
  "xim_channel_set": 0
}'
{"result":{"err_no":0,"err_msg":"ok"},"activated":true}$
$
$
$
```

**Channel is activated**

## 7. Discover the Device List

```
curl -X POST \
  https://xim.qblinks.com/v1/discovery \
  -H 'authorization: Bearer [user_quantum_token]' \
  -H 'content-type: application/json' \
  -d '{
    "xim_type": "light",
```



```
"xim_channel": "hue",
"xim_channel_set": 0
}
```

```
$ curl -X POST https://xim.qblinks.com/v1/discovery -H 'authorization: Bearer e[redacted]
iJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhcHBfaWQiOiJzZW9sZWUiLCJ1c2VybmFtZSI6ImNhbXBvbiU0MHFiYGl
ua3MuY29tOmxb2xlZSIsInNjb3BlIjoieGltIiwiaXhwIjozMTUzNjAwMDAwLCJpYXQiOiE1MDI0MjA4NjF9.j
RlffEpzVFvis3uTjgP8DkfjVuuC-13CRMxNq3UC-b4' -H 'content-type: application/json' -d '{
  "xim_type": "light",
  "xim_channel": "hue",
  "xim_channel_set": 0
}'
{"xim_type":"light","xim_channel":"hue","xim_channel_set":0,"result":{"err_no":0,"err_m
sg":"ok"},"list":[{"device_name":"on Samson's Desk","device_id":"2","is_group":false,"l
ight_type":"color","infrared_support":false,"native_toggle_support":false,"light_status
":{"hue":180,"saturation":83,"brightness":100,"onoff":false}},{"device_name":"on Liming
's Desk","device_id":"3","is_group":false,"light_type":"color","infrared_support":false
,"native_toggle_support":false,"light_status":{"hue":212,"saturation":74,"brightness":1
00,"onoff":false}},{"device_name":"Next to Printer","device_id":"4","is_group":false,"l
ight_type":"color","infrared_support":false,"native_toggle_support":false,"light_status
":{"hue":155,"saturation":100,"brightness":100,"onoff":false}}],"groups":[{"group_name
":"Walkway","group_id":"900002"},{"group_name":"Office","group_id":"900003"}]}
$
$
```

 **List of lights and groups**

Please refer to the properties.js for the format. Each channel type has a different property set. Please go to the XIM drivers for more the information.

```
{
  "xim_type": "light",
  "xim_channel": "hue",
  "xim_channel_set": 0,
  "result": {
    "err_no": 0,
    "err_msg": "ok"
  },
  "list": [
    {
      "device_name": "[device_name_1]",
      "device_id": "[device_id_1]",
      "is_group": false,
      "light_type": "color",
      "infrared_support": false,
      "native_toggle_support": false,
      "light_status": {
```



```
        "hue": 212,  
        "saturation": 74,  
        "brightness": 100,  
        "onoff": false  
    }  
},  
{  
    "device_name": "[device_name_2]",  
    "device_id": "[device_id_2]",  
    "is_group": false,  
    "light_type": "color",  
    "infrared_support": false,  
    "native_toggle_support": false,  
    "light_status": {  
        "hue": 212,  
        "saturation": 74,  
        "brightness": 100,  
        "onoff": false  
    }  
},  
{  
    "device_name": "[device_name_3]",  
    "device_id": "[device_id_3]",  
    "is_group": false,  
    "light_type": "color",  
    "infrared_support": false,  
    "native_toggle_support": false,  
    "light_status": {  
        "hue": 204,  
        "saturation": 100,  
        "brightness": 100,  
        "onoff": false  
    }  
}
```



```

    ],
    "groups": [
      {
        "group_name": "[group_name_1]",
        "group_id": "[group_id_1]"
      },
      {
        "group_name": "[group_name_2]",
        "group_id": "[group_id_2]"
      }
    ]
  }

```

#### 8. Perform an action on the device

```

curl -X POST \
  https://xim.qblinks.com/v1/action \
  -H 'authorization: Bearer [user_quantum_token]' \
  -H 'content-type: application/json' \
  -d '{
    "xim_type": "light",
    "xim_channel": "hue",
    "xim_channel_set": 0,
    "device_id": "[device_id]",
    "action": {
      "brightness":100,
      "hue": 97,
      "saturation": 55,
      "toggle": true
    }
  }'

```



```

$ curl -X POST https://xim.qblinks.com/v1/action -H 'authorization: Bearer e
UzI1NiIsInR5cCI6IkpXVCJ9.eyJhcHBfaWQiOiJsc2VybWVtZSI6InNhbXNvbiU0MHF1bGluas
MuY29tOm1b2xlZSI6InNjb3B1IjoieGl0IiwiaXhwIjozMjUzNjAwMDAwLCJpYXQiOiE1MDI0MjA4NjF9. jRlf
fEpzVFvis3uTjgP8DkfjVuuC-13CRMxNq3UC-b4' -H 'content-type: application/json' -d '{
v  "xim_type": "light",
v  "xim_channel": "hue",
v  "xim_channel_set": 0,
v  "device_id": "4",
v  "action": {
v    "brightness":100,
v    "hue": 97,
v    "saturation": 55,
v    "toggle": true
v  }
v }'

{"result":{"err_no":0,"err_msg":"ok"}}$
$
$

```

**Command to control the light**

## Reference Information

For the API calls, please refer to

<https://mytracmo.com/quantum-api-doc/>

For the XIM Drivers, please refer to

<https://github.com/qblinks>

