Nuki Bridge API

V1.12.3

22.06.2021

Nuki Home Solutions GmbH
Münzgrabenstrasse 92/4, 8010 Graz

1. Introduction

The REST API on the Nuki Bridge offers simple endpoints to list all available Nuki Smart Locks and Nuki Openers, retrieve their current lock state and perform lock operations.

Check for the latest version of this document at our Developer Plattform.

1.1 Abbreviations used

Abbr.	Long form	Description
cm	Continuous Mode	Nuki Opener Mode with Ring to Open continuously activated
lng	Lock 'n' Go	Unlock and lock again automatically
rto	Ring to Open	Nuki Opener State in which ringing the bell activates the electric strike actuation

2. Calling URL

This is the address used to call the available services of the internal webserver.

The IP address is shown in the bridge settings within the Nuki App or can be retrieved from the bridge discovery URL.

The server is listening for incoming requests either on default port 8080 or the configured one if it has been modified within the Nuki App.

2.1 Example

The following base url will be used in upcoming examples:

http://192.168.1.50:8080/

3. Bridge discovery & API activation

Calling the URL https://api.nuki.io/discover/bridges returns a JSON array with all bridges which have been connected to the Nuki Servers through the same IP address than the one calling the URL within the last 30 days. The array contains the local IP address, port, the ID of each bridge and the date of the last change of the entry in the JSON array.

3.1 Example


```
{
  "bridges": [
    {
      "bridgeId":2117604523,"ip":"192.168.1.50","port":8080,"dateUpdated":"2017-06-14
T06:53:44Z"
    }
  ],
  "errorCode":0
}
```


Once a bridge has been discovered on the LAN the API can be activated and the API token retrieved by calling the /auth command. The user has to confirm this request by pressing the button on the bridge. For more details see the description of the /auth command. Alternatively you can activate the API and set the token by managing the Bridge in the Nuki App.

If discovery is disabled via /configAuth or through the Nuki App, the IP is 0.0.0.0 and the port 0. In this case the /auth command fails with HTTP error 403.

3.1.1 Alternative via Nuki App

As an alternative you can activate and manage the Bridge API via the Nuki App by opening *Burger menu* > *Manage my devices* > *Bridge* and follow the described steps:

3.2 Token

We offer two ways of verifying calls to endpoints with a token:

Method	**Usage**
Plain token	You can use the plain token for testing and in private, secured WIFIs or VLANs.

Hashed token	Use if you do not want to send the plain token within your API-calls. **Note:** Only available for the hardware bridge for now.

3.2.1 Parameters

Name	Parameter	Values	Example
Plain token	token	uint8[20]	123456
Timestamp	ts	YYY-MM-DD**T**HH:MM:SS**Z**	2019-03-05T01:06:53Z
Random number	rnr	uint16	4711
Hash	hash	sha256("ts,mr,token")	f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6

sha256("2019-03-05T01:06:53Z,4711,123456") =
f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6

3.2.2 Example calls

Plain token:
http://192.168.1.50:8080/info?token=123456

Hashed token:
http://192.168.1.50:8080/info?ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6

A hashed token will only be valid with a sufficiently current timestamp and can not be reused, to prevent replay attacks. So making two calls with the exact same timestamp will only work with different random numbers.
To debug problems with non synchronous times you can check the current time on the bridge via bridge discovery

4 States and Actions

4.1 Device Types

Nuki device connected to the bridge.

0 ... smartlock *(Nuki Smart Lock)*

2 ... opener *(Nuki Opener)*

4.2 Modes

mode	smartlock	opener	Description
2	door mode	door mode	Operation mode after complete setup
3	-	continuous mode	Ring to Open permanently active

Note: Only modes 2 and 3 can appear in JSON elements, as the HTTP API is not available in the other modes.

4.3 Lock States

Possible lock states (used in Endpoints below).

ID	smartlock	opener
0	uncalibrated	untrained
1	locked	online
2	unlocking	-

3	unlocked	rto active
4	locking	-
5	unlatched	open
6	unlocked (lock ‘n’ go)	-
7	unlatching	opening
253	-	boot run
254	motor blocked	-
255	undefined	undefined

4.4 Lock Actions

Possible lock actions (used in Endpoints below):

ID	smartlock	opener
1	unlock	activate rto
2	lock	deactivate rto
3	unlatch	electric strike actuation
4	lock ‘n’ go	activate continuous mode
5	lock ‘n’ go with unlatch	deactivate continuous mode

4.5 Simple Lock Actions

Possible outcome of a simple lock action (mapping handled in the firmware of the device):

action	smartlock / knob	smartlock / handle	opener
/lock	lock	lock	deactivate rto and cm
/unlock	unlatch	unlock	open

To use this features your Nuki devices need the following firmware version:

Nuki device	Firmware version
Bridge	1.14.0/2.5.0 (or higher)
Smart Lock 1.0	1.8.0 (or higher)
Smart Lock 2.0	2.4.3 (or higher)
Opener	1.3.0 (or higher)

4.6 Doorsensor States

Possible door sensor states (used in Endpoints below).

ID	name
1	deactivated
2	door closed
3	door opened
4	door state unknown
5	calibrating

5. Endpoints

/auth

URL	http://192.168.1.50:8080/auth	
Usage	Enables the api (if not yet enabled) and returns the api token. If no api token has yet been set, a new (random) one is generated. When issuing this API-call the bridge turns on its LED for 30 seconds. The button of the bridge has to be pressed within this timeframe. Otherwise the bridge returns a negative success and no token.	
Response	JSON list containing the success of the authorization	
	token	The api token
	success	Flag indicating the success of the authorization
Errors	**HTTP 403**	Returned if the authentication is disabled
Example-Call	http://192.168.1.50:8080/auth	

Example-Response	{
	"token": “token123”,
	"success": true
	}	

/configAuth

URL	http://192.168.1.50:8080/configAuth
Usage	Enables or disables the authorization via /auth and the

	publication of the local IP and port to the discovery URL (https://api.nuki.io/discover/bridges).	
URL-Parameters	**enable**	Flag (0 or 1) indicating whether or not the authorization should be enabled
	token	The api token configured via the Nuki app when enabling the API
Response	JSON list containing the success of the operation	
	success	Flag indicating the success of the authorization
Errors	**HTTP 400**	Returned if the given value for **enable** is invalid (neither 0 nor 1)
	HTTP 401	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
Example-Calls	http://192.168.1.50:8080/configAuth?enable=0&token=123456 http://192.168.1.50:8080/configAuth?enable=0&ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	

Example-Response	{
	"success": true
	}	

/list

URL	http://192.168.1.50:8080/list	
Usage	Returns a list of all paired Nuki devices	
URL-Parameters	**token**	The api token configured via the Nuki app when enabling the API
Response	JSON array. One item of the following per Nuki device	

	nukiId	ID of the Nuki device	
	deviceType	Nuki device type ● 0 => smartlock *(Nuki Smart Lock*) ● 2 => opener *(Nuki Opener)*	
	name	Name of the Nuki device	
	lastKnownState	JSON list containing the last known lock state of the Nuki device	
		mode	ID of the lock mode (see Modes)
		state	ID of the lock state (see Lock States)
		stateName	Name of the lock state (see Lock States)
		batteryCritical	Flag indicating if the batteries of the Nuki device are at critical level
		keypadBatteryCritical	Flag indicating if the batteries of the paired Nuki Keypad are at critical level
		doorsensorState	ID of the door sensor state
		doorsensorStateName	Name of the door sensor state
		ringactionTimestamp	timestamp of the last ring-action
		ringactionState	Flag indicating if a ring-action is currently occuring or not (reset after 30 seconds)
		timestamp	Timestamp of the retrieval of this lock state

Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
Example-Calls	http://192.168.1.50:8080/list?token=123456 http://192.168.1.50:8080/list?ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	

Example-Response	[{
	"nukiId": 1,
	"deviceType": 0,
	"name": "Home",
	"lastKnownState": {
	"mode": 2,
	"state": 1,
	"stateName": "unlocked",
	"batteryCritical": false,
	"keypadBatteryCritical": false,
	"doorsensorState": 2,
	"doorsensorStateName": "door closed",
	"timestamp": "2018-10-03T06:49:00+00:00" }
	},{
	"nukiId": 2,
	"deviceType": 2,
	"name": "Community door",
	"lastKnownState": {
	"mode": 3,
	"state": 3,
	"stateName": "rto active",
	"batteryCritical": false,
	"ringactionTimestamp": 2020-04-27T16:13:00+00:00”,
	"ringactionState": false,
	"timestamp": "2018-10-03T06:49:00+00:00"
	}
	}]	

/lockState

Warning: /lockstate gets the current state directly from the device and so should not be used for constant polling to avoid draining the batteries too fast. /list can be used to get regular updates on the state, as is it cached on the bridge.

URL	http://192.168.1.50:8080/lockState	
Usage	Retrieves and returns the current lock state of a given Nuki device	
URL-Parameters	**nukiId**	The ID of the Nuki device from which the lock state should be retrieved
	deviceType	Nuki device type (see Device Types; *defaults to 0)*
	token	The api token configured via the Nuki app when enabling the API
Response	JSON list containing the retrieved lock state	
	mode	ID of the lock mode (see Modes)
	state	ID of the lock state (see Lock States)
	stateName	Name of the lock state (see Lock States)
	batteryCritical	Flag indicating if the batteries of the Nuki device are at critical level
	keypadBatteryCritical	Flag indicating if the batteries of the paired Nuki Keypad are at critical level
	doorsensorState	ID of the door sensor state
	doorsensorStateName	Name of the door sensor state
	ringactionTimestamp	timestamp of the last ring-action

	ringactionState	Flag indicating if a ring-action is currently occuring or not (reset after 30 seconds)
	success	Flag indicating if the lock state retrieval has been successful
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
	HTTP 404	Returned if the given Nuki device is unknown
	HTTP 503	Returned if the given Nuki device is offline
Example-Calls	http://192.168.1.50:8080/lockState?nukiId=1&deviceType=0&token=123456 http://192.168.1.50:8080/lockState?nukiId=1&deviceType=&0ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	

Example-Response	{
	"mode": 2,
	"state": 1,
	"stateName": "locked",
	"batteryCritical": false,
	"keypadBatteryCritical": false,
	"ringactionTimestamp": 2020-04-27T16:13:00+00:00",
	"ringactionState": false,
	"doorsensorState": 2,
	"doorsensorStateName": "door closed",
	"success": true
	}	

/lockAction

URL	http://192.168.1.50:8080/lockAction

Usage	Performs a lock action on the given Nuki device	
URL-Parameters	**nukiId**	The ID of the Nuki device which should execute the lock action
	deviceType	Nuki device type (see Device Types; *defaults to 0)*
	action	The desired lock action (see Lock Actions)
	nowait	Flag (0 or 1) indicating whether or not to wait for the lock action to complete and return its result *(optional; defaults to 0)*
	token	The api token configured via the Nuki app when enabling the API
Response	JSON list containing the result of the lock action	
	batteryCritical	Flag indicating if the batteries of the Nuki device are at critical level
	success	Flag indicating if the lock action has been executed successfully
Errors	**HTTP 400**	Returned if the given **action** is invalid
	HTTP 401	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
	HTTP 404	Returned if the given SNuki device is unknown
	HTTP 503	Returned if the given Nuki device is offline
Example-Calls	http://192.168.1.50:8080/lockAction?nukiId=1&deviceType=0&action=1&token=123456 http://192.168.1.50:8080/lockAction?nukiId=1&deviceType=0&action=1&ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	
Example-Response	{	

	<pre>"success": true,
	"batteryCritical": false
	}</pre>

/lock

URL	http://192.168.1.50:8080/lock	
Usage	Send the simple lock action "lock" to a given Nuki device	
URL-Parameters	**nukiId**	The ID of the Nuki device which should execute the lock action
	deviceType	Nuki device type (see Device Types; *defaults to 0)*
	token	The api token configured via the Nuki app when enabling the API
Response	JSON list containing the result of the lock action	
	batteryCritical	Flag indicating if the batteries of the Nuki device are at critical level
	success	Flag indicating if the lock action has been executed successfully
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
	HTTP 404	Returned if the given Nuki device is unknown
	HTTP 503	Returned if the given Nuki device is offline
Example-Calls	http://192.168.1.50:8080/lock?nukiId=1&deviceType=0&token=123456 http://192.168.1.50:8080/lock?nukiId=11&deviceType=0&ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	

Example-Response	<pre>{
	  "success": true,
	  "batteryCritical": false
	}</pre>

/unlock

URL	http://192.168.1.50:8080/unlock	
Usage	Send the simple lock action "unlock" to a given Nuki device	
URL-Parameters	**nukiId**	The ID of the Nuki device which should execute the lock action
	deviceType	Nuki device type (see Device Types; *defaults to 0)*
	token	The api token configured via the Nuki app when enabling the API
Response	JSON list containing the result of the unlock action	
	batteryCritical	Flag indicating if the batteries of the Nuki device are at critical level
	success	Flag indicating if the unlock action has been executed successfully
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
	HTTP 404	Returned if the given Nuki device is unknown
	HTTP 503	Returned if the given Nuki device is offline
Example-Calls	http://192.168.1.50:8080/unlock?nukiId=1&deviceType=0&token=123456 http://192.168.1.50:8080/unlock?nukiId=11&deviceType=0&ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f	

	8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6

Example-Response	<pre>{
	  "success": true,
	  "batteryCritical": false
	}</pre>

/unpair

not available on software bridge

URL	http://192.168.1.50:8080/unpair	
Usage	Removes the pairing with a given Nuki device	
URL-Parameters	**nukiId**	The ID of the Nuki device which should be unpaired
	deviceType	Nuki device type (see Device Types; *defaults to 0)*
	token	The api token configured via the Nuki app when enabling the API
Response	JSON list containing the result of the operation	
	success	Flag indicating if the lock action has been executed successfully
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
	HTTP 404	Returned if the given Nuki device is unknown
Example-Calls	http://192.168.1.50:8080/unpair?nukiId=1&token=123456 http://192.168.1.50:8080/unpair?nukiId=1&ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	

Example-Response	{
	“success”: true
	}

/info

URL	http://192.168.1.50:8080/info		
Usage	Returns all Nuki devices in range and some device information of the bridge itself		
URL-Parameters	**token**	The api token configured via the Nuki app when enabling the API	
Response	JSON list with the result		
	bridgeType	• 1 => Hardware bridge • 2 => Software bridge	
	ids	JSON list containing the ids of the bridge	
		hardwareId	Hardware ID *(hardware bridge only)*
		serverId	Server ID
	versions	JSON list containing the versions of bridge	
		firmwareVersion	Version of the bridges firmware *(hardware bridge only)*
		wifiFirmwareVersion	Version of the WiFi modules firmwarehardware bridge only
		appVersion	Version of the bridge appsoftware bridge only

<table>
<tr><td rowspan="8"></td><td>uptime</td><td colspan="2">Uptime of the bridge in seconds</td></tr>
<tr><td>currentTime</td><td colspan="2">Current timestamp</td></tr>
<tr><td>serverConnected</td><td colspan="2">Flag indicating whether or not the bridge is connected to the Nuki server</td></tr>
<tr><td rowspan="6">scanResults</td><td colspan="2">JSON Array. One item of the following per Nuki device</td></tr>
<tr><td>nukiId</td><td>Nuki device ID</td></tr>
<tr><td>deviceType</td><td>Nuki device type (see Device Types)</td></tr>
<tr><td>name</td><td>BLE-Name of the Nuki device</td></tr>
<tr><td>rssi</td><td>RSSI value</td></tr>
<tr><td>paired</td><td>Flag indicating whether or not a pairing with this

Nuki device has already been established</td></tr>
<tr><td>Errors</td><td>HTTP 401</td><td colspan="2">Returned if the given token is invalid or a hashed token parameter is missing.</td></tr>
<tr><td>Example-Calls</td><td colspan="3">http://192.168.1.50:8080/info?token=123456
http://192.168.1.50:8080/info?ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6</td></tr>
<tr><td>Example-Response</td><td colspan="3"><pre>{
  "bridgeType": 1,
  "ids": {"hardwareId": 12345678, "serverId":
     12345678},
  "versions": { "firmwareVersion": "0.1.0",
     "wifiFirmwareVersion": "0.2.0" },</pre></td></tr>
</table>

	<pre>"uptime": 120,
	"currentTime": "2018-04-01T12:10:11Z",
	"serverConnected": true,
	"scanResults": [{ "nukiId": 10, "type": 0,
	"name": "Nuki_00000010", "rssi": -87,
	"paired": true }, { "nukiId": 11,
	"deviceType": 2, "name": "Nuki_00000011",
	"rssi": -93, "paired": false }]
	}</pre>

/callback

The following endpoints provide methods to register up to 3 http (no https) url callbacks, which will be triggered once the lock state of one of the known Nuki devices changes.

The new lock state will be sent to the callback url by executing a POST request and posting a JSON list in the following format:

```
{"nukiId": 11, "deviceType": 0, "mode": 2, "state": 1, "stateName":
"locked", "batteryCritical": false, "keypadBatteryCritical": false}
```

Nuki device with door sensor capabilities:

```
{"nukiId": 11, "deviceType": 0, "mode": 2, "state": 1, "stateName":
"locked", "batteryCritical": false, "doorsensorState": 2,
"doorsensorStateName": "door closed"}
```

Opener (with ring action capabilities):

```
{"nukiId": 11, "deviceType": 2, "mode": 3, "state": 3, "stateName":
"rto active", "batteryCritical": false, "ringactionTimestamp":
"2020-04-27T16:13:00+00:00", "ringactionState": false}
```

/callback/add

URL	http://192.168.1.50:8080/callback/add	
Usage	Registers a new callback url	
URL-Parameters	**url**	The callback url to be added (no https, url encoded, max. 254 chars)
	token	The api token configured via the Nuki app when enabling the API
Response	JSON list containing the result	
	success	Flag indicating if the url has been added successfully
	message	Contains the reason for the failure if **success** is false

Errors	**HTTP 400**	Returned if the given **URL** is invalid or too long
	HTTP 401	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
Example-Calls	http://192.168.1.50:8080/callback/add?url=http%3A%2F%2F192.168.0.20%3A8000%2Fnuki&token=123456 http://192.168.1.50:8080/callback/add?url=http%3A%2F%2F192.168.0.20%3A8000%2Fnuki&ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	

Example-Response	`{`
	`  "success": true`
	`}`	

/callback/list

URL	http://192.168.1.50:8080/callback/list		
Usage	Returns all registered url callbacks		
URL-Parameters	**token**	The api token configured via the Nuki app when enabling the API	
Response	JSON list with the result		
	callbacks	JSON array. One item of the following per callback	
		id	ID of the callback
		url	URL of the callback
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.	
Example-Calls	http://192.168.1.50:8080/callback/list?token=123456 http://192.168.1.50:8080/callback/list?ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6		

Example-Response	{
	"callbacks": [
	{
	“id”: 0,
	"url": "http://192.168.0.20:8000/nuki"
	},{
	“id”: 1,
	"url": "http://192.168.0.21/test"
	}
	]
	}

/callback/remove

URL	http://192.168.1.50:8080/callback/remove	
Usage	Removes a previously added callback	
URL-Parameters	**id**	The id of the callback to be removed
	token	The api token configured via the Nuki app when enabling the API
Response	JSON list containing the result	
	success	Flag indicating if the url has been added successfully
	message	Contains the reason for the failure if **success** is false
Errors	**HTTP 400**	Returned if the given **url** is invalid or too long
	HTTP 401	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
Example-Calls	http://192.168.1.50:8080/callback/remove?id=0&token=123456 http://192.168.1.50:8080/callback/remove?id=0&ts=2019-03-05	

	T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6

Example-Response	<pre>{
	  "success": true
	}</pre>

6. Maintenance endpoints

The following endpoints are available for maintenance purposes of the hardware bridge. Therefore they are not available on the software bridge.

/log

URL	http://192.168.1.50:8080/log	
Usage	Retrieves the log of the bridge	
URL-Parameters	**offset**	Offset position where to start retrieving log entries *(optional; defaults to 0)*
	count	How many log entries to retrieve *(optional; defaults to 100)*
	token	The api token configured via the Nuki app when enabling the API
Response	JSON array. One item of the following per log entry	
	timestamp	Timestamp of the log entry
	type	Type of the log entry
	some more optional parameters	
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
Example-Calls	http://192.168.1.50:8080/log?token=123456 http://192.168.1.50:8080/log?ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	
Example-Response	[{"timestamp": "2018-10-06T16:46:05+00:00", "deviceType": "..." },	

	{"timestamp": "2018-10-06T16:46:05+00:00", "deviceType": “...” }, …]

/clearlog

URL	http://192.168.1.50:8080/clearlog	
Usage	Clears the log of the bridge	
URL-Parameters	**token**	The api token configured via the Nuki app when enabling the API
Response	No response	
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
Example-Calls	http://192.168.1.50:8080/clearlog?token=123456 http://192.168.1.50:8080/clearlog?ts=2019-03-05T01:06:53Z&nr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	
Example-Response	None	

/fwupdate

URL	http://192.168.1.50:8080/fwupdate	
Usage	Immediately checks for a new firmware update and installs it	
URL-Parameters	**token**	The api token configured via the Nuki app when enabling the API
Response	No response	
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.

Example-Calls	http://192.168.1.50:8080/fwupdate?token=123456 http://192.168.1.50:8080/fwupdate?ts=2019-03-05T01:06:53Z&nr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6
Example-Response	None

/reboot

URL	http://192.168.1.50:8080/reboot	
Usage	Reboots the bridge	
URL-Parameters	**token**	The api token configured via the Nuki app when enabling the API
Response	No response	
Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
Example-Calls	http://192.168.1.50:8080/reboot?token=123456 http://192.168.1.50:8080/reboot?ts=2019-03-05T01:06:53Z&nr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	
Example-Response	None	

/factoryReset

URL	http://192.168.1.50:8080/factoryReset	
Usage	Performs a factory reset	
URL-Parameters	**token**	The api token configured via the Nuki app when enabling the API
Response	No response	

Errors	**HTTP 401**	Returned if the given **token** is invalid or a **hashed token** parameter is missing.
Example-Calls	http://192.168.1.50:8080/factoryReset?token=123456 http://192.168.1.50:8080/factoryReset?ts=2019-03-05T01:06:53Z&rnr=4711&hash=f52eb5ce382e356c4239f8fb4d0a87402bb95b7b3124f0762b806ad7d0d01cb6	
Example-Response	None	

7. Error codes/handling

Specififc errors for endpoints are documented in the respective section. This is an overview of general and specific errors that may occur when using the Bridge API:

Error code	Type	Description	Solution
400	Bad Request	Wrong/missing parameter	Check endpoint documentation for details on expected paramaters and format.
401	Unauthorized	Invalid token or missing hashed token parameter	Recheck if the token is correct or parameters are correctly set.
403	Forbidden	Authentication is disabled	Activate the Bridge API (see 3. Bridge discovery & API activation).
404	Not Found	Unknown Nuki device ID	Recheck the connected device IDs on the Bridge and the device ID used in the request.
503	Service Unavailable	Another request already running on the device	Increase intervals between API calls sent to the Bridge as it can only handle one request at a time.
Failed to connect	Connection refused	Bridge not available at given URL	Check if the Bridge is powered and connected to the Wifi and if IP and Port are correctly set in your request.

8. Frequently Asked Questions

Why are the batteries of my Smart Lock draining so fast when I use the Bridge API?

Most likely you are repeatadly calling /lockAction to get the current state directly from the device, but this should not be used for constant polling to avoid draining the batteries too fast. /list can be used instead to get regular updates on the state, as is it cached on the bridge.

Why do i repeatdly get an Error 503 when calling the Bridge API

The Bridge can only handle one incoming request at a time and you therefore have to serialize repeated requests to the Bridge API. See also: 7. Error codes/handling

Why do API commands sometimes take very long or time out?

The Bridge can only handle one outgoing command at a time and may also have to wait for the reponse of a Nuki actuator. So using several clients (Bridge API, Nuki Apps, Nuki Web) at the same time may lead to delays or timeouts.

9. Changelog

Changelog v 1.12.3

22.06.2021

- Added error code overview and handling section
- Added a Frequently Asked Questions section.

Changelog v 1.12.2

11.06.2021

- Fixed missing values for battery state.

Changelog v 1.12.1

07.05.2021

- Added information on how to activate the API alternatively via Nuki App.

Changelog v 1.12

02.09.2020

- Updated **/lockState** to include the keypadBatteryCritical flag, ringactionState and ringactionTimestamp.
- Updated **/list** to include the keypadBatteryCritical flag, ringactionState and ringactionTimestamp.
- Expanded POST request example for a **/callback** with the keypadBattery flag, ringactionState and ringactionTimestamp.

Changelog v 1.11

08.07.2020

- Introduced **Dorsensor States** for all supported devices.
- Updated **/lockState** to include doorsensorState and doorsensorStateName in the response.
- Updated **/list** to include doorsensorState and doorsensorStateName in the response.
- Added a POST request example for a device with door sensor capabilities to **/callback**.

Changelog v 1.10

07.01.2020

- Introduced **Simple lock actions** for all usecases where the logic should be handled by the device itself.
- Made wording for Nuki devices more general.

Changelog v 1.9

06.05.2019

- Introduced **Device Types** and **Modes** to be able to distinguish between Smart Locks and Nuki Openers and their operating modes.
- Updated **Lock States** to reflect matching and new states for the Nuki Opener.
- Updated **Lock Actions** to reflect matching and new actions for the Nuki Opener and add deviceType parameter.
- Added Opener support to /list and /info endpoints.
- Expanded Callbacks to Nuki Openers and added **deviceType** and **mode**.
- Expanded Callbacks to Nuki Openers and added **deviceType** and **mode**.
- Added deviceType parameter to /unpair.

Changelog v 1.8

07.03.2019

- Introducing the hashed token as a more secure alternative to sending the plain token

Changelog v 1.7

30.03.2018

- Small changes in bridge discovery information

Changelog v 1.6

21.06.2017

- Added bridge discovery