

Nuki Web API

V1.3.0

02.03.2020

Nuki Home Solutions GmbH
Münzgrabenstrasse 92/4, 8010 Graz

1. Introduction	3
1.1 Abbreviations used	3
1.2 Wording	4
1.3 Device IDs	4
2. Calling URL	5
3. Swagger Interface	5
3.1 Available endpoints	6
3.2 Example API call through Swagger	8
4. Authentication	12
4.1 Curl call from the Swagger example:	12
4.2 When to use which type of Authentication?	12
4.3 API Tokens	12
4.3.1 Scopes	14
4.4 OAuth 2	15
4.4.1 “Code Flow” OAuth 2 Authentication Example	15
4.4.1.1 Authorization Code Link	15
4.4.1.2 User Authorizes Application	17
4.4.1.3 Application Receives Authorization Code	17
4.4.1.4 Application Requests Access Token	18
4.4.1.5 Application Receives Access Token	18
4.4.1.6 Token Refresh	18
4.4.2 “Implicit” OAuth 2 authentication example	19
4.4.2.1 Authorization Code Link	19
4.4.2.2 User Authorizes Application	19
4.4.2.3 Receive Access Token Via Callback URL	19
4.5 Advanced API integration	19
5. Smart Lock States	20
6. Smart Lock Actions	21
6.1 Simple Lock Actions	22
7. Changelog	22
Changelog v.1.3.0	22
Changelog v.1.2.1	22
Changelog v.1.2.0	22
Changelog v.1.1.1	22

1. Introduction

The Nuki Web API offers various ways to interact with a Nuki Smart Lock. The API transmits all commands directly through a permanent HTTPS/TLS connection to the corresponding Nuki bridge, which forwards it via Bluetooth to the Smart Lock for execution. Responses are directly fed back into Nuki Web.

All commands are performed with the server-stored Nuki Web Authentication Key which was created when Nuki Web has been initially activated by the Smart Lock administrator. Because of having its own Authentication Key, Nuki Web acts independently of other clients (e.g. Nuki iOS or Android App).

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
ms	Milliseconds	One thousandth of a second
rto	Ring to Open	Nuki Opener State in which ringing the bell activates the electric strike actuation

1.2 Wording

Smartlock

Endpoints containing "Smartlock" are used for all Nuki devices and are only kept that way for legacy reasons. All supported device types use the same endpoints. Only the Nuki Opener has additional ones for intercom compatibility.

Swagger

The 3. Swagger UI is a tool we use to automatically generate documentation from our OpenAPI definition for visual interaction and easier testing for you.

API token

We use 4.3 API Tokens as Authorization Bearer for calls to the Web API.

OAuth2

4.4 OAuth 2 is an open standard we use to grant applications access to Nuki Web users devices without sharing passwords.

1.3 Device IDs

The Web API expects Nuki Device IDs to be sent as an integer. Device IDs in HEX wherefore need to be converted to DEC. For the **Nuki Opener** this needs the type as a prefix to ensure unqiue IDs for Web API usage

device	type	ID (HEX) example	ID (HEX) example with prefix	calulated ID (DEC) example
keyturner	0	1A2B3C4D	1A2B3C4D	439041101
opener	2	1A2B3C4D	21A2B3C4D	9028975693

2. Calling URL

The Nuki Web API can be found under the URL <https://api.nuki.io>.

3. Swagger Interface

The Swagger Interface at <https://api.nuki.io/> lists all API commands with its input and output parameters. Next to just listing the commands the Interface also allows to easily perform API commands.

When adding support for the Nuki Opener the new Smart Lock *type = 2 ... Opener* has been introduced. Smart Lock states and Smart Lock actions are mapped for the new usecase.

New Smart Lock actions as well as a new Smart Lock *mode = 3 ... continuous mode* have been added, to activate, deactivate and signal the status of a *Continuous Ring to Open Mode* for the Nuki Opener.

Additionally a new **Opener Advanced Config** has been introduced, which is used for advanced settings for the Nuki Opener instead of the Smartlock Advanced Config.

For all details check the [Smart Lock States](#) and [Smart Lock Actions](#) sections and the updated models at <https://api.nuki.io/#!/Smartlock/>

3.1 Available endpoints

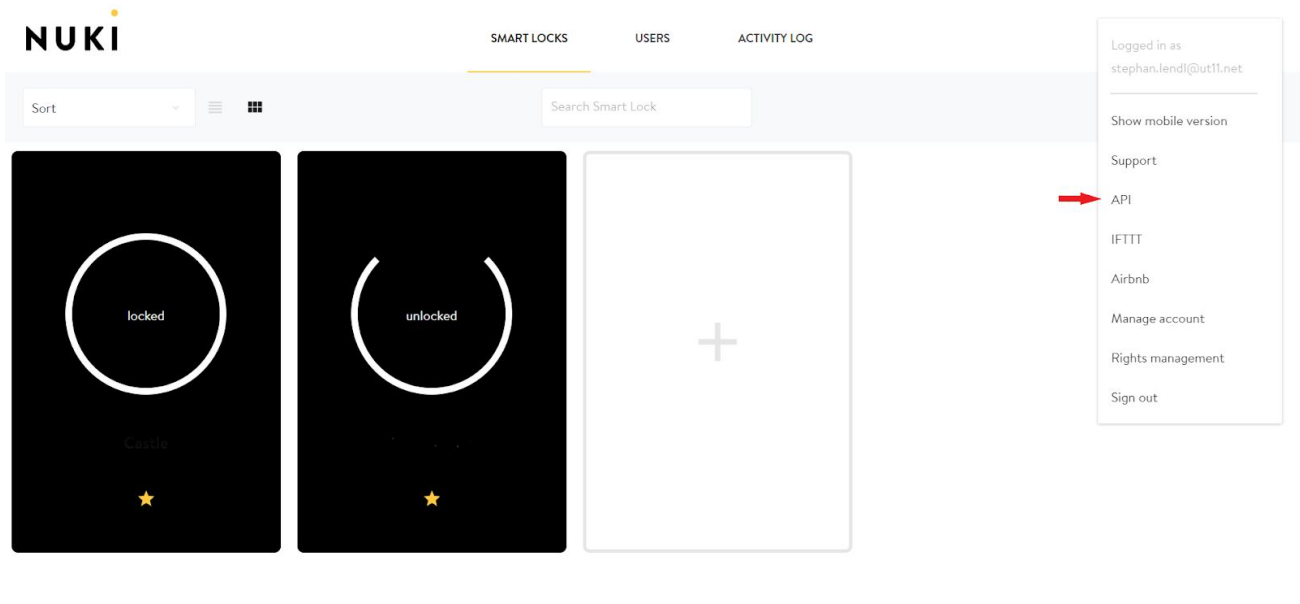
Path	Usage	Available options	Description	Scope needed
Account	Nuki Web account	POST, GET, PUT, DELETE	Handle Nuki Web accounts and sub-accounts, OTP settings and password reset.	account
AccountSubscription	Nuki Box subscriptions	POST, GET	Check and edit Nuki Box subscription tokens.	account
AccountUser	Nuki device users	POST, GET, PUT, DELETE	Create, edit and delete (email based) Nuki device users to which authorizations can be assigned.	account, smartlock, auth
Address	Nuki device grouping	POST, GET	Connecting an array of Nuki devices to an address object for Nuki Box subscriptions and short rental.	account
AddressReservation	Short rental integration	GET, POST	Handle bookings for connected listings from short rental integrations.	account

AddressToken	Nuki Box subscriptions	POST, GET	Create and check Nuki Box subscriptions	-
ApiKey	Manage Web API keys	POST, GET, PUT, DELETE	Create, edit and delete API keys for the Nuki Web API.	account
Company	Nuki Partner network	GET	List companies from Nukis partner network.	-
Notification (BETA)	Push notifications and webhooks	POST, GET, PUT, DELETE	Create, edit and delete webhooks for activity log entries and errors.	smartlock.log
Opener	Opener compatibility check and installation	GET	List Opener compatible intercoms per brand.	-
Service	Short Rental	POST, GET	Link, unlink and sync available short rental integration services.	account
Smartlock	Nuki devices	POST, GET, PUT, DELETE	Manage Nuki devices and device settings.	smartlock, (smartlock.readOnly)
SmartlockAuth	Authorizations	POST, GET, PUT, DELETE	Create, edit and delete authorizations	smartlock.auth

			on Nuki devices.	
SmartlockLog	Activity Log	GET	Retrieve log files from Nuki devices	smartlock.log
Subscription	Nuki Box subscriptions	GET	Check for valid Nuki Box subscriptions.	account

3.2 Example API call through Swagger

Log into Nuki Web, go to MENU > API, activate the Nuki Web API and copy your OAuth 2 API key.



Nuki Web API

The Nuki Web API makes it easy for programmers to integrate Nuki's features into other applications.

OAuth2 API key & URL

Creating your own application that requires access to Nuki?

OAuth2 API key



OAuth2 redirect URL

API tokens

API tokens provide full access to Nuki, so keep them safe.

Generate API token

Deactivate Nuki Web API

Deactivating the Nuki Web API immediately renders it unusable and deletes all API tokens.

Deactivate Nuki Web API

Go to <https://api.nuki.io>, paste the OAuth 2 API key, select scopes you want to grant this key and log into the Swagger interface.



Enter your api key here

Authorize

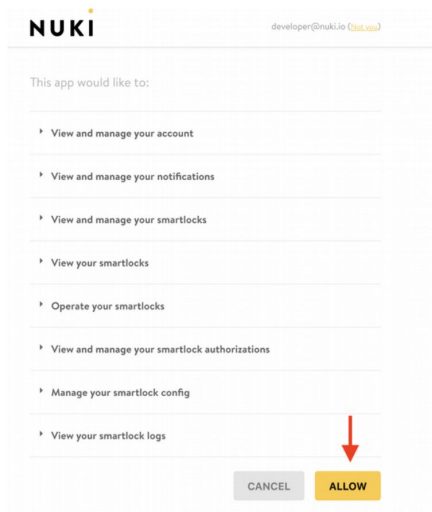
Nuki API

Created by Nuki
See more at <https://nuki.io>
[Contact the developer](#)
[Apache 2.0](#)

Account	Show/Hide	List Operations	Expand Operations
AccountSubscription	Show/Hide	List Operations	Expand Operations
AccountUser	Show/Hide	List Operations	Expand Operations
Address	Show/Hide	List Operations	Expand Operations
AddressReservation	Show/Hide	List Operations	Expand Operations
AddressToken	Show/Hide	List Operations	Expand Operations
ApiKey	Show/Hide	List Operations	Expand Operations
Company	Show/Hide	List Operations	Expand Operations
Service	Show/Hide	List Operations	Expand Operations
Smartlock	Show/Hide	List Operations	Expand Operations
SmartlockAuth	Show/Hide	List Operations	Expand Operations
SmartlockLog	Show/Hide	List Operations	Expand Operations
Subscription	Show/Hide	List Operations	Expand Operations

[BASE URL: / , API VERSION: v1]

VALID 



Choose an API function you want to execute, e.g. the GET /smartlock command in order to obtain a list of Smart Locks on this account:

Smartlock Show/Hide List Operations Expand Operations

GET /smartlock Get a list of smartlocks

Response Class (Status 200)
successful operation

Model Example Value

```
[
  {
    "smartlockId": 0,
    "accountId": 0,
    "type": 0,
    "authId": 0,
    "name": "string",
    "favorite": true,
    "config": {
      "name": "string",

```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
authId	<input type="text"/>	Filter for authId	query	integer
type	<input type="text"/>	Filter for type	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Not authorized		

[Try it out!](#)

PUT /smartlock Create a smartlock

You will also get the corresponding cURL call and the response from the API.

4. Authentication

Successful API calls require appropriate authorization: The bearer token (also see <https://swagger.io/docs/specification/authentication/bearer-authentication/>) needs to be present in each request to the API. There are several ways on how to obtain a valid bearer token, which we will describe in the upcoming section.

4.1 Curl call from the Swagger example:

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer c2c0981ffcab78eecd13c8b7ae9fdec4706045bdbb17b1ef06a335b832f36641322c5c3357b7fe47' 'https://api.nuki.io/smartlock'
```

4.2 When to use which type of Authentication?

API Tokens	When you use the API to access your own Nuki Web account with your own Smart Locks only.
OAuth 2	<p>When you are offering an application to your users which grants your server/application the right to operate the Smart Lock of a user.</p> <p>When your users have no technical experience and you want to offer a simple login to your services without the need for the user to generate API tokens and copy them around.</p> <p>When you need short term access to a users Nuki Web information for your (mobile) web app. In this case use the implicit authentication flow.</p>

4.3 API Tokens

Log into your Nuki Web account, go to MENU > API and create a new API Token. Use this API Token as Authorization Bearer.

Generate new API token

API token name

Which rights would you like to grant this token?

- View and manage account
- View and manage notifications
- View and edit Smart Locks
- View Smart Locks
- Operate Smart Locks
- View and manage Smart Lock authorizations
- Manage Smart Lock configuration
- View Smart Lock activity log

Cancel Generate

Copy the API token into the clipboard and store it in a secure way. It gives permanent access to all rights you did grant to it:

Generate new API token

API token name

API token

d5a94f81466076dcea0568376354e7ea21b9

Which rights would you like to grant this token?

- View and manage account
- View and manage notifications
- View and edit Smart Locks
- View Smart Locks
- Operate Smart Locks
- View and manage Smart Lock authorizations
- Manage Smart Lock configuration
- View Smart Lock activity log

Token was copied to the clipboard.

Close

Use it as the “Authorization: Bearer” in your API calls:

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer API_token' 'https://api.nuki.io/smartlock'
```

API tokens do not expire, but they are destroyed when the password of the corresponding Nuki Web account changes.

4.3.1 Scopes

Scope	Label	Description
account	View and manage account	Edit the Nuki Web user. Create, edit and delete Nuki Web sub-users. Create, edit and delete API keys.
smartlock	View and edit devices	Add, view, edit and remove devices to/from Nuki Web (API).
smartlock.readonly	View devices	Show Nuki devices in Nuki Web (API).
smartlock.config	Manage device configuration	Change device settings in Nuki Web (API).
smartlock.action	Operate devices	Operate devices via Nuki Web (API).
smartlock.create	Create devices	Add Nuki devices to Nuki Web (API).
smartlock.auth	View and manage authorizations	Create, edit and delete authorizations on a Nuki device via Nuki Web (API).
smartlock.log	View activity logs and get log notifications	Retrieve logs from Nuki devices via Nuki Web (API) and manage webhooks for the Web API.

4.4 OAuth 2

We support the Authorization grants “[Code Flow](#)” and “[Implicit](#)”. When using “Implicit” the access token expires after one hour.

If you follow the “Code Flow” scheme you will need a client secret in order to receive an access token. Client secrets are issued only by Nuki. Please apply via [Nuki Web](#) to get yours.

Note that you need to create a Nuki Web account by adding a Nuki device to Nuki Web via the Nuki App first.

For an introduction of OAuth 2 have a look at this:

<https://www.digitalocean.com/community/tutorials/an-introduction-to-OAuth-2#authorization-grant>

4.4.1 “Code Flow” OAuth 2 Authentication Example

4.4.1.1 Authorization Code Link

```
https://api.nuki.io/oauth/authorize?response_type=code&client_id=CLIENT_ID&redirect_uri=CALLBACK_URL&scope=SCOPES
```

CLIENT_ID is your **OAuth 2 API Key** from Nuki Web > MENU > API

CALLBACK_URL is your callback URL to which users will be redirected after they successfully logged in. You can restrict the allowed CALLBACK_URL to e.g. your domain by inserting it into Nuki Web > MENU > API > **OAuth 2 Redirect URL**. If you leave this field empty, every CALLBACK_URL is allowed.

Nuki Web API

The Nuki Web API makes it easy for programmers to integrate Nuki's features into other applications.

OAuth2 API key & URL

Creating your own application that requires access to Nuki?

OAuth2 API key



OAuth2 redirect URL

API tokens

API tokens provide full access to Nuki, so keep them safe.

[Generate API token](#)

Deactivate Nuki Web API

Deactivating the Nuki Web API immediately renders it unusable and deletes all API tokens.

[Deactivate Nuki Web API](#)

SCOPES is a list of scopes that you want to request from the user for your application. You can see which scope is needed for which API command on the Swagger frontend at <https://api.nuki.io>

All parameters need to be URL encoded ([Online URL encoder/decoder](#)).

Example Authorization Call:

```
http://api.nuki.io/oauth/authorize?response_type=code&redirect_uri=https%3A%2F%2Ftest.com&client_id=v7kn_NX7vQ7VjQdXFGK43g&scope=account%20notification%20smartlock%20smartlock.readOnly%20smartlock.action%20smartlock.auth%20smartlock.config%20smartlock.log
```


4.4.1.2 User Authorizes Application

The image shows two screenshots from the NUKI application. The top screenshot is the login page, featuring the NUKI logo, a sign-in prompt, and input fields for email and password, with a yellow 'SIGN IN' button. The bottom screenshot is an authorization consent screen, also with the NUKI logo, showing a list of permissions the app requests, such as 'View and manage your account' and 'Operate your smartlocks', with 'CANCEL' and 'ALLOW' buttons at the bottom.

NUKI

Sign in with your Nuki Web account

Email

Email

Password

Password

SIGN IN

NUKI developer@nuki.io (Net you)

This app would like to:

- View and manage your account
- View and manage your notifications
- View and manage your smartlocks
- View your smartlocks
- Operate your smartlocks
- View and manage your smartlock authorizations
- Manage your smartlock config
- View your smartlock logs

CANCEL **ALLOW**

4.4.1.3 Application Receives Authorization Code

Your user will be redirect to the provided callback URL:

`CALLBACK_URL?code=AUTHORIZATION_CODE`

Redirect URL from the example above:

```
https://www.test.com/?code=d69dc5bdfbae822707a3bbc3a8ea2f1a9f6053d5%717592822654
```

4.4.1.4 Application Requests Access Token

Your application/server posts to the following URL to receive the final access token:

```
curl -X POST -d 'client_id=CLIENT_ID client_secret=CLIENT_SECRET
grant_type=authorization_code code=AUTHORIZATION_CODE redirect_uri=CALLBACK_URL'
https://api.nuki.io/oauth/token
```

CLIENT_ID is your **OAuth 2 API Key** from Nuki Web > MENU > API

CALLBACK_URL is your callback URL to which users will be redirected after they successfully logged in. You can restrict the allowed **CALLBACK_URL** to e.g. your domain by inserting it in to Nuki Web > MENU > API > **OAuth 2 Redirect URL**. If you leave this field empty, every **CALLBACK_URL** is allowed.

CLIENT_SECRET is your client secret received from the [Web API settings in Nuki Web](#).

4.4.1.5 Application Receives Access Token

You will receive something like this as response from the server:

```
{"access_token":"ACCESS_TOKEN","token_type":"bearer","expires_in":2592000,"refresh_token":"REFRESH_TOKEN"}
```

You can use this **ACCESS_TOKEN** to make requests to the API in the same way as with API token authentication:

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer
ACCESS_TOKEN' 'https://api.nuki.io/smartlock'
```

4.4.1.6 Token Refresh

After your access token expires you will receive an “Invalid Token Error” from the API. You can use the **REFRESH_TOKEN** received in step 5 to get a new **ACCESS_TOKEN** by posting the following URL:

```
curl -X POST -d
`grant_type=refresh_token&client_id=CLIENT_ID&client_secret=CLIENT_SECRET&refresh_token=REFRESH_TOKEN` https://api.nuki.io/oauth/token
```

CLIENT_ID is your **OAuth 2 API Key** from Nuki Web > MENU > API

CLIENT_SECRET is your client secret received from the [Web API settings in Nuki Web](#).

REFRESH_TOKEN is your refresh token received together with your last access token (step 5)

4.4.2 “Implicit” OAuth 2 authentication example

4.4.2.1 Authorization Code Link

```
https://api.nuki.io/oauth/authorize?response_type=token&client_id=CLIENT_ID  
redirect_uri= scope=SCOPES
```

CLIENT_ID is your **OAuth 2 API Key** from Nuki Web > MENU > API.

CALLBACK_URL is your callback URL to which users will be redirected after they successfully logged in. You can restrict the allowed **CALLBACK_URL** to e.g. your domain by inserting it in to Nuki Web > MENU > API > **OAuth 2 Redirect URL**. If you leave this field empty, every **CALLBACK_URL** is allowed.

SCOPES is a list of scopes that you want to request from the user for your application. You can see which scope is needed for which API command on the Swagger frontend at <https://api.nuki.io>.

See the example of the [“code flow” authorization](#) for a detailed description of the parameters.

4.4.2.2 User Authorizes Application

Same as with [“code flow” authorization](#)

4.4.2.3 Receive Access Token Via Callback URL

Your user will be redirected to the provided callback URL:

```
CALLBACK_URL?token=ACCESS_TOKEN
```

Your application needs to extract the **ACCESS_TOKEN** from the URL and can afterwards use this **ACCESS_TOKEN** for up to one hour to make requests to the API in the same way as with API token authentication:

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer  
ACCESS_TOKEN' ' https://api.nuki.io/smartlock '
```

4.5 Advanced API integration

You can apply via [Nuki Web API management](#)

to get your OAuth 2 client secret, or apply for advanced access to the Web API.

Currently this is only used in the **Short Rental** usecase to access `PUT /smartlock/auth/advanced` which can be used to create Nuki invite codes. Those which will be returned via webhook instead of triggering an email by Nuki. The codes still need to be redeemed from within the Nuki App, but integrators can handle the channels and messages used to deliver it to a guest themselves.

5. Smart Lock States

Name	smartlock	opener
mode	<p>The current operation state of the Nuki Smart Lock</p> <p>0 uninitialized 1 pairing 2 door (default) 3 - 4 maintenance</p>	<p>The current operation state of the Nuki Opener</p> <p>0 uninitialized 1 pairing 2 door (default) 3 continuous 4 maintenance</p>
state	<p>The current state of the Nuki Smart Lock</p> <p>0 uncalibrated 1 locked 2 unlocking 3 unlocked 4 locking 5 unlatched 6 unlocked (lock'n'go) 7 unlatching 253 - 254 motor blocked 255 undefined</p>	<p>The current state of the intercom control within Nuki Opener</p> <p>rto ... Ring to Open</p> <p>0 untrained 1 online 2 - 3 rto active 4 - 5 open 6 - 7 opening 253 boot run 254 - 255 undefined</p>
trigger	<p>The trigger, that caused the state change within the Nuki Smart Lock</p> <p>0 system (bluetooth) 1 manual 2 button 3 automatic 4 - 5 -</p>	<p>The trigger, that caused the state change within the Nuki Opener</p> <p>0 system (bluetooth) 1 manual 2 button 3 automatic 4 - 5 -</p>

	6 -	6 continuous mode
lastAction	1 unlock 2 lock 3 unlatch 4 lock'n'go 5 lock'n'go with unlatch 6 - 7 -	1 activate rto 2 deactivate rto 3 electric strike actuation 4 - 5 - 6 activate cm 7 deactivate cm

Note: trigger-types 4 and 5 are Box-only

6. Smart Lock Actions

LockActions are used as parameter in lock commands or show up as lastAction for states or state changes.

Name	smartlock	box	opener
action	1 unlock 2 lock 3 unlatch 4 lock'n'go 5 lock'n'go with unlatch 6 - 7 -	1 unlock 2 - 3 - 4 - 5 - 6 - 7 -	1 activate rto 2 deactivate rto 3 electric strike actuation 4 - 5 - 6 activate cm 7 deactivate cm

6.1 Simple Lock Actions

Instead of sending a lockAction ID as a parameter with [POST /smartlock/{smartlockId}/action](#), separate endpoints can be used to send simple "lock" or "unlock" commands to a device.

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)

7. Changelog

Changelog v.1.3.0

02.03.2020

- Introduced [6.1 Simple Lock Actions](#) for all usecases where the logic should be handled by the device itself.
- Added information on [1.3 device ID](#) usage.
- Added [3.1 Available endpoints](#) to the Swagger part.
- Added a description of the available [4.3.1 Scopes](#) to the API token section.
- Added general naming conventions with [1.2 Wording](#).

Changelog v.1.2.1

14.01.2020

- Introduced the new section [4.5 Advanced API integration](#) to cover additional scopes which can only be accessed after registration and verification.

Changelog v.1.2.0

31.05.2019

- Added support for the Nuki Opener to the Web API
- Added chapters for [Smart Lock States](#) and [Actions](#) to show differences between the Nuki Smart Lock and the Nuki Opener.
- Noted changes and adding of new OpenerAdvancedSettings in section [Swagger interface](#).

Changelog v.1.1.1

30.08.2018

- Fixed some missing links
- Fixed some typos and unclear text