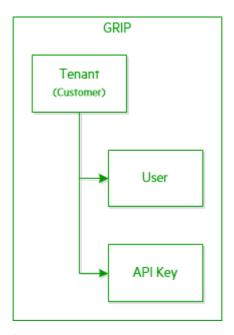# KPN Things Developer Manual

API access ✩✩

> ⚠ ✩✩ **Early access feature!**
> API access to KPN Things is currently in beta! We try to notify users in advance about breaking changes, but we cannot guarantee continuity of this service just yet.

API access to KPN Things is managed by GRIP, the identify management platform of KPN. Your user account used to access the KPN Things Portal is also managed by GRIP. Learn more about the account you use to access KPN Things.

In GRIP there are **API Keys**. These are separate entities in a tenant (a customer environment). That means an API key is not user bound, but it is rather a separate entity.

API Keys are (like Users) part of a GRIP Tenant and represent an entity that can log into KPN applications.
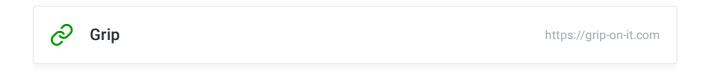
Create your API Key in GRIP

> ⓘ It is best practice to create individual API Keys for each instance that will access the KPN Things API's.
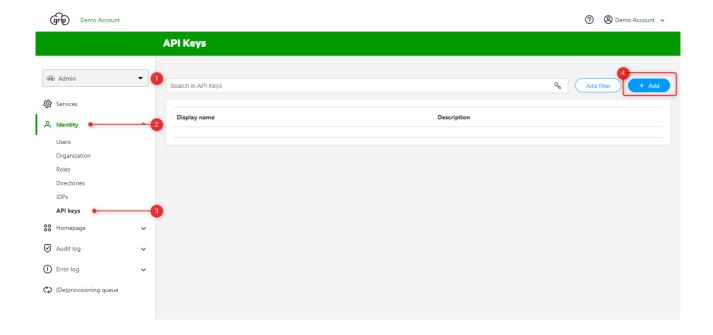
Log into the GRIP Portal as administrator.

- If you have a **KPN Developer** account, you will be administrator in the GRIP Portal.
- If you have a **MijnKPN Zakelijk** account, you should ask your local administrator for access.

Learn more about the account you use to access KPN Things.

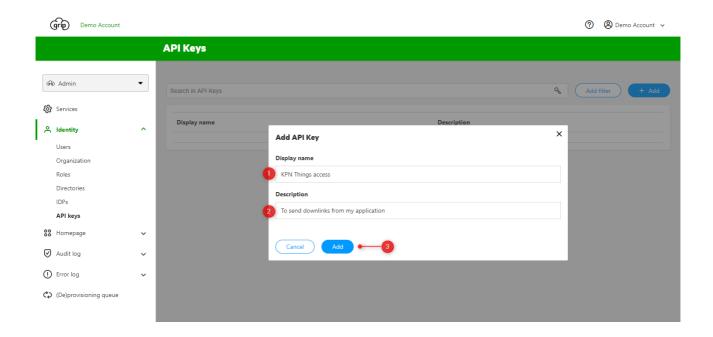| 🔗 **Grip**	https://grip-on-it.com

Go to API Key management

Then, open the **Admin** panel using the role drop down (#1 in the screenshot), click on **Identity** in the side menu (#2 in the screenshot) and click on **API keys** (#3 in the screenshot). This brings you to the API Key management page. Now, click **Add** to add a new API Key (#4 in the screenshot).

Create new API Key

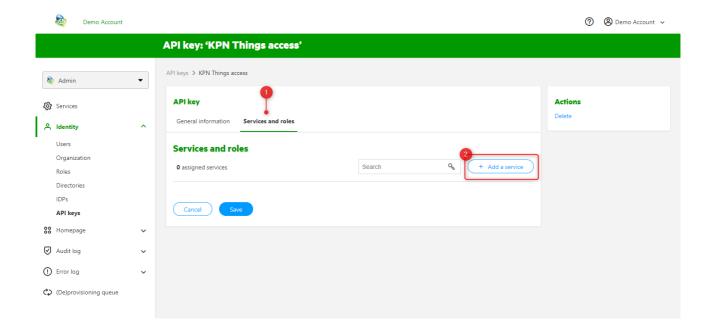In the following pop-up, give your new API Key a *Display name* and a *Description*. Then click **Add** to create your new API Key.
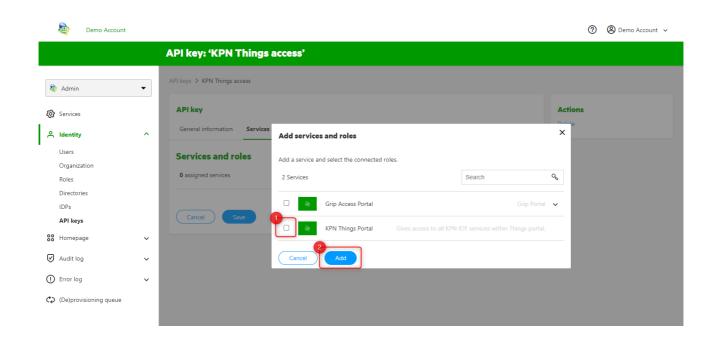

Give new API Key access to KPN Things
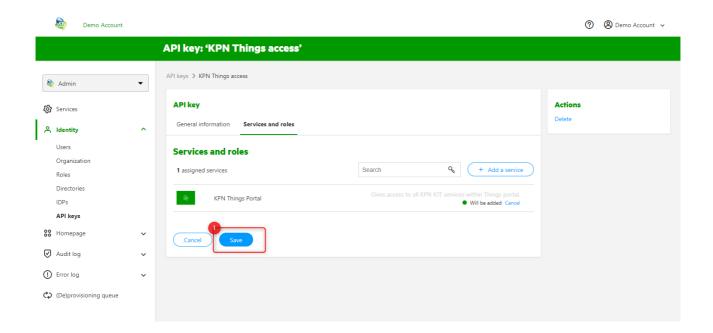
You need to give your newly created API Key access to the KPN Things Portal service before you can use it. For this, click on **Services and roles** (#1 in the screenshot), and click on **Add a service** (#2 in the screenshot).

A modal will open. Click on the **checkbox** next to KPN Things Portal (#1 in the screenshot) and hit **Add** (#2 in the screenshot).

Then, click **Save** to finally give your API Key access to KPN Things.

Fetch credentials of new API Key

Now, when you click back on **General information**, you will see the *ID* and *Secret key* of your new API Key.

> ⓘ **Copy-paste** the key's ID and Secret key values for later use. You will need these values every time you request access to KPN Things.
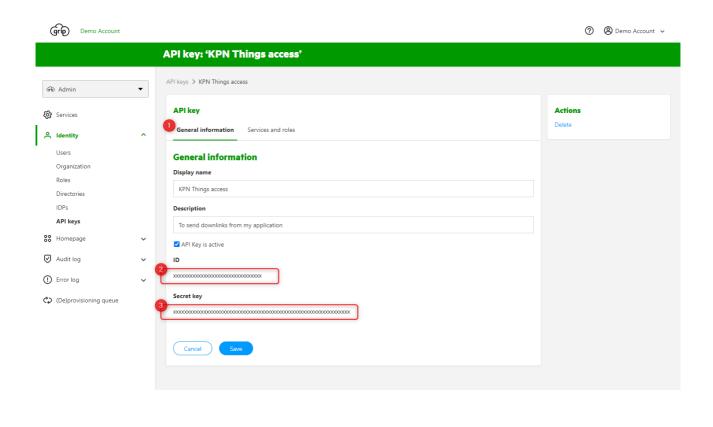

API call to get access token for API access

> ⓘ Don't want to manually implement the API calls? Use our published Postman collection to quickly generate code snippets for all published API calls!

After the previous step you have an API ID and API Secret key. Each time you want to access the KPN Things API's, you need to request an access token from the GRIP API using these API Key's ID and Secret key.

The access token you get is **valid for one hour**, and can be used as a bearer token to access KPN Things API's directly. That means, you can use the following Authorization header when accessing KPN Things API's:

```
    Authorization: Bearer {accessToken}
```

There is no refresh token. To regain access after the validity of your access token has
expired, just request another access token through the GRIP API.

You can find more information about the access method we employ here on the Grip
Knowledge Base website:

🔗 **Machine 2 Machine - Grip Knowledge Base**   https://kb.grip-on-it.com/en/service-
integration/machine2machine/

Collect required information for API call to request access token

The following values should be known to be able to request access tokens through the
GRIP API:

Value	Description
Tenant ID	The unique identifier of your GRIP environment. Visit the GRIP Portal on https://www.grip-on-it.com. After logging in, your Tenant ID will be in the URL of the home page: `https://www.grip-on-it.com/a/{tenantId}/homepage` It will be a string of lowercase letters and numbers.
Application ID	The ID of the service you want to access, being KPN Things. Used as audience in your client credentials request. For KPN Things it is this: `4dc82561-f65f-523g-dek9-6c79ec314f02`
Client ID	The API Key's ID you got from the previous step.
Client Secret	The API Key's Secret key you got from the previous step.

`POST` **Request KPN Things access token through GRIP API**

https://auth.grip-on-it.com**/v2/{tenantId}/oidc/idp/c1/token**

Request

Headers

Content-Type
REQUIRED

`string`

application/json; charset=utf-8

Body Parameters

grant_type
REQUIRED

`string`

client_credentials

audience
REQUIRED

`string`

The Application ID (see above)

client_id
REQUIRED

`string`

The API Key's ID

client_secret
REQUIRED

`string`

The API Key's Secret key

Response

● 200: OK

The `access_token` in the response can be used as bearer token to access KPN Things API's.

```
1  {
2      "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
3      "token_type": "Bearer",
4      "expires_in": 3600
5  }
```

Error	Solution
Feature user does not exist anymore.	You should check your API Key ID or Secret, they may be incorrect.
Feature service does not exist anymore.	You should add the KPN Things Portal service to your API Key through the GRIP Portal.
SAML request error: "idpId" not found.	Your tenantId (which is a path variable) is incorrect.

More information in the GRIP Knowledge Base (documentation).

Perform API calls

ⓘ Don't want to manually implement the API calls? Use our published Postman collection to quickly generate code snippets for all published API calls!

After obtaining an access token through the GRIP API, you can access the KPN Things API's!

You can for instance list your Devices.

`GET` Example: List devices

https://api.kpnthings.com**/api/devices**

Now you have an access token, execute the following call to list your Devices

Request

Headers

Authorization
REQUIRED

`string`

Bearer *{accessToken}*

Accept
REQUIRED

`string`

application/vnd.kpnthings.device.v1.response+json

Query Parameters

currentPage
OPTIONAL

`number`

0

pageSize
OPTIONAL

`number`

20

search
OPTIONAL

`string`

(**case sensitive** partial string search)

sortBy
OPTIONAL

`string`

name

sortOrder
OPTIONAL

`string`

ASC

Response

● 200: OK

A paged list of all your Devices.

```json
1  {
2      "content": [
3          {
4              "uuid": "xxxxxx",
5              "name": "My first own device",
6              "description": "",
7              "metadata": null,
8              "deviceSpecificationUUID": "d2723e99-1fe4-4d60-9b6d-8061
9              "barcode": null
10         },
11         ...
12     ],
13     "last": true,
14     "totalPages": 1,
15     "totalElements": 2,
16     "sort": [
17         {
18             "direction": "ASC",
19             "property": "name",
20             "ignoreCase": false,
21             "nullHandling": "NATIVE",
22             "ascending": true,
23             "descending": false
24         }
25     ],
26     "first": true,
27     "numberOfElements": 2,
28     "size": 20,
29     "number": 0
30  }
```

Response codes

Code	Description

☐ 200 OK	Your request was correct.
☐ 201 Created	You created an object with your request.
☐ 204 No Content	Your request was correct, but there is no information to return, for instance after deleting an object.
☐ 302 Found	Your access token is incorrect or does not have access to KPN Things. Please check your API Key in GRIP.
☐ 400 Bad Request	Something in your request is not correctly formatted. For instance your body structure, your accept header or a query parameter.
☐ 401 Forbidden	Your access token is correct, but you do not have the authorization to access that resource.

`POST` Send downlink to device

https://api.kpnthings.com**/api/actuator/downlinks**

Body: [{"bn":"urn:dev:DEVEUI:{devEUI}:", "n":"payloadHex", "vs": "{payload}"}]

Request

Headers

Accept

REQUIRED

application/vnd.kpnthings.actuator.v1.response+json

Content-Type

REQUIRED

application/json

Query Parameters

externalReference

OPTIONAL

`string`
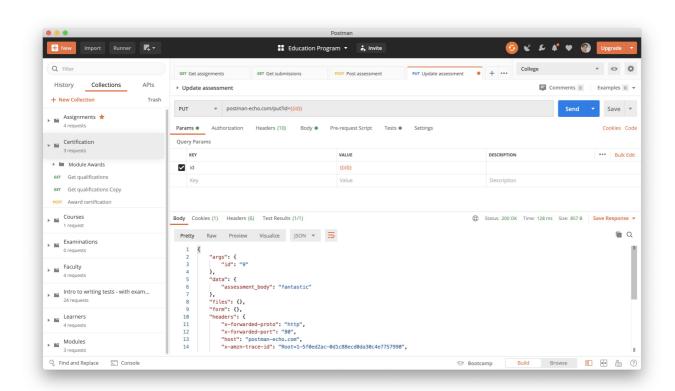
Optional reference for you to choose

port

REQUIRED

`integer`

LoRa FPort to send data on

Response

● 200: OK

Full documentation of KPN Things API's

Currently the API's of KPN Things are documented using Postman. Postman is an easy to use API development tool.

Postman API Collection

🔗 KPN Things API's - public documentation   https://documenter.getpostman.com/view/1978616/TVRg7pvv

Postman environment variables

To get the Postman collection working, you need to create a Postman environment with the following variables:

Variable	Value
gripBaseUrl	`https://auth.grip-on-it.com`
tenantId	Your grip tenant id
applicationId	`4dc82561-f65f-523g-dek9-6c79ec314f02`
keyId	Your key id
keySecret	Your key secret
baseUrl	`https://api.kpnthings.com`

SSL certificate information

The SSL server certificates of KPN Things and GRIP are provided by the new *PKI Overheid CA* hierarchy. On kpn.com you can learn more about the KPN certificates.

If required for your security implementation, all certificate files of the chain can be downloaded here:

- Root: Staat der Nederlanden EV Root CA
- Intermediate: Staat der Nederlanden Domein Server CA 2020
- Intermediate: KPN PKIoverheid Server CA 2020