

Get started with Linkspers API

The Linkspers API is a **universal API**. It means that no matter the site architecture, no matter the master system integrator, no matter the hardware installed and integrated, the same queries will be available offering a seamless experience and data structure.

This page will help you to:

- [Deep dive in the documentation](#)
- [Authenticate with Linkspers](#)
- [Execute a first request](#)
- [Understand the API structure](#)
- [Manage user rights](#)

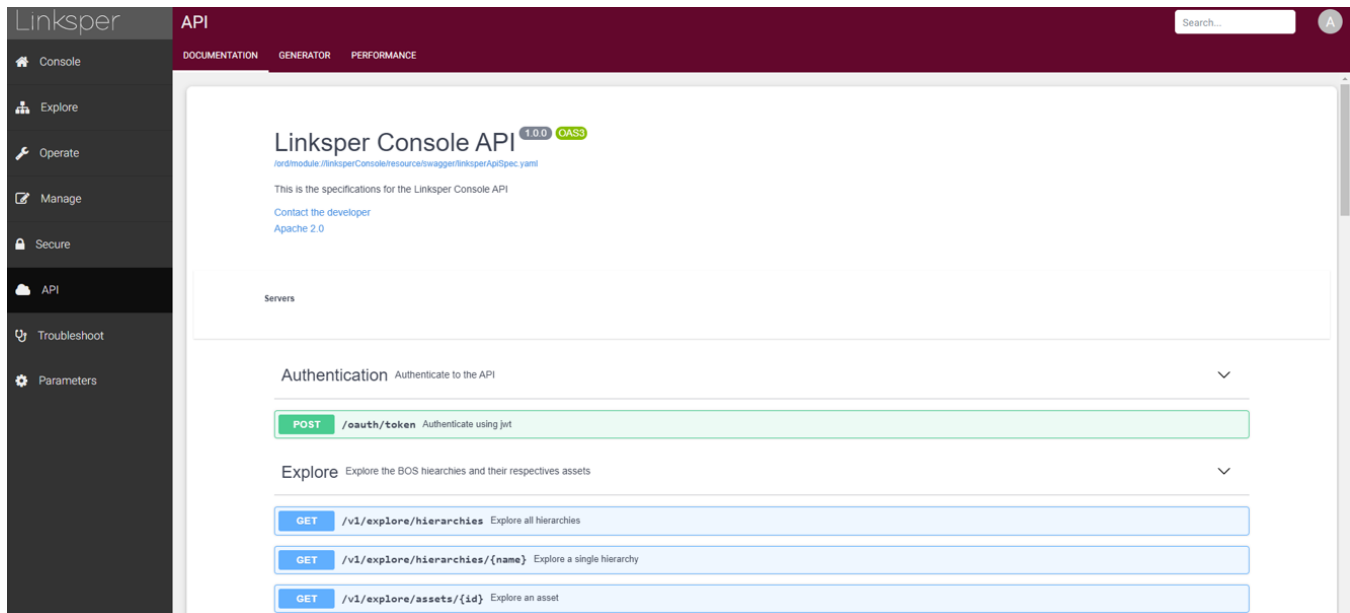
VayanData can provide you access to a sandbox. Please contact us: <https://vayandata.com/contact-us/>

Deep dive in the documentation

Open API

The BOS Linkspers Console API documentation is available here: **Linkspers API**

When you have access rights to the Linkspers Console administration application on the BOS, you also have access to the API documentation (Open API Format) :

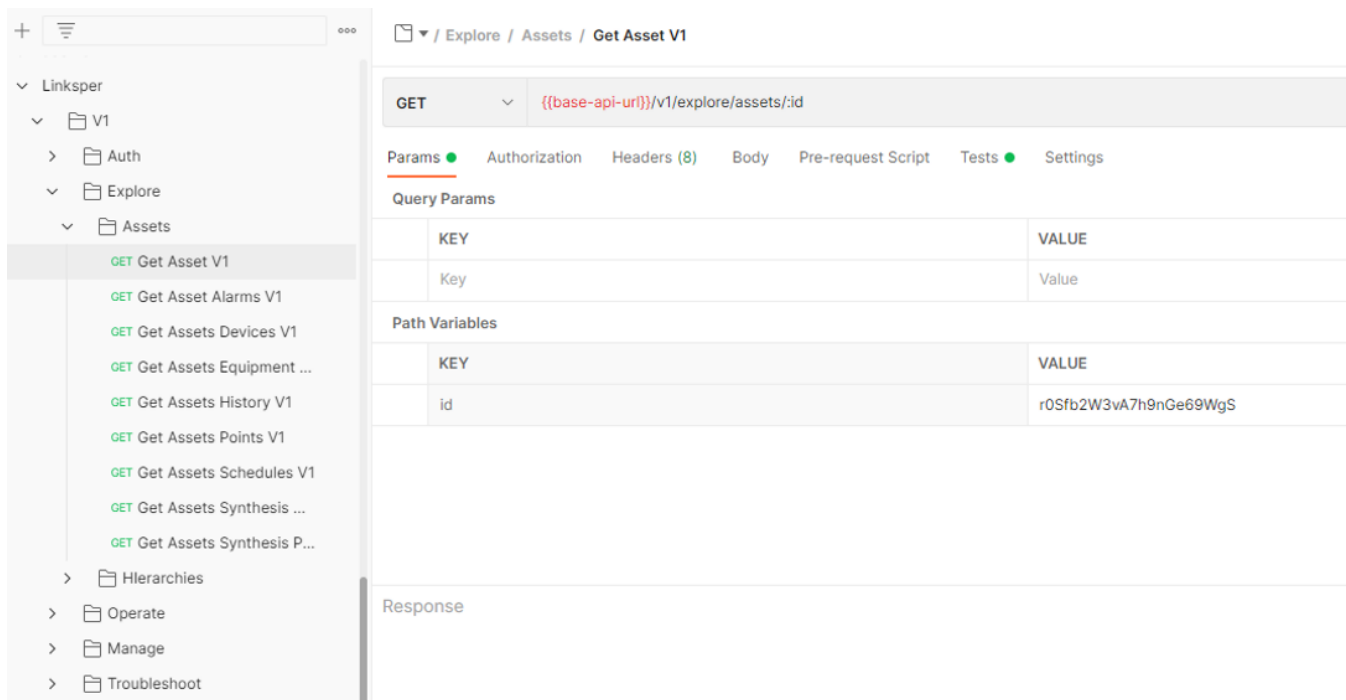


Query collection

In addition to the documentation, a **Postman** collection is available below. This collection provides examples of requests to interact with a Linkspers BOS.



The collection provides an example for each of the endpoints available in the API.



The screenshot shows the Postman interface with the 'Linkspers' collection expanded. The 'Assets' folder is selected, and the 'Get Asset V1' endpoint is highlighted. The endpoint is a GET request with the URL `{{base-api-uri}}/v1/explore/assets/:id`. The 'Params' tab is active, showing 'Query Params' and 'Path Variables'.

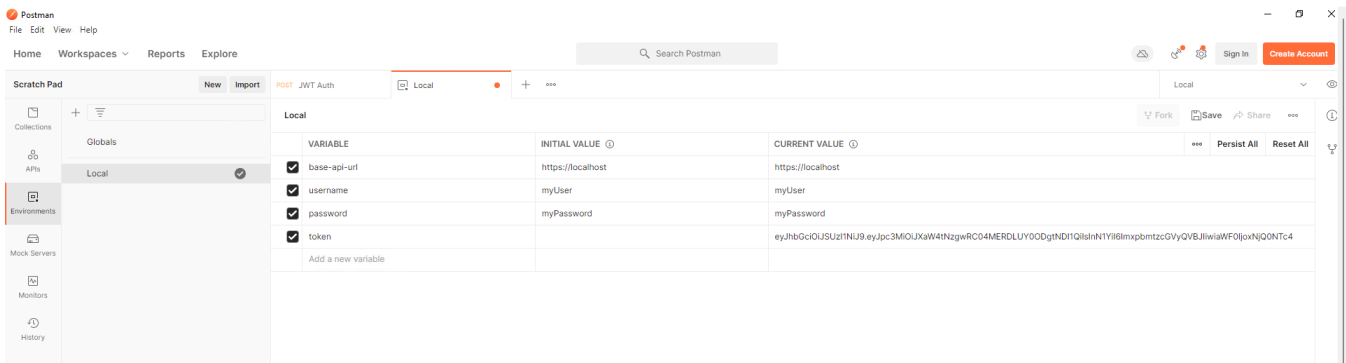
KEY	VALUE
Key	Value

KEY	VALUE
id	r0Sfb2W3vA7h9nGe69WgS

The 'Response' tab is also visible at the bottom.

The collection requires an environment with pre-defined variables to be set-up with Postman accordingly to your Linkspers instance. The three required variables are:

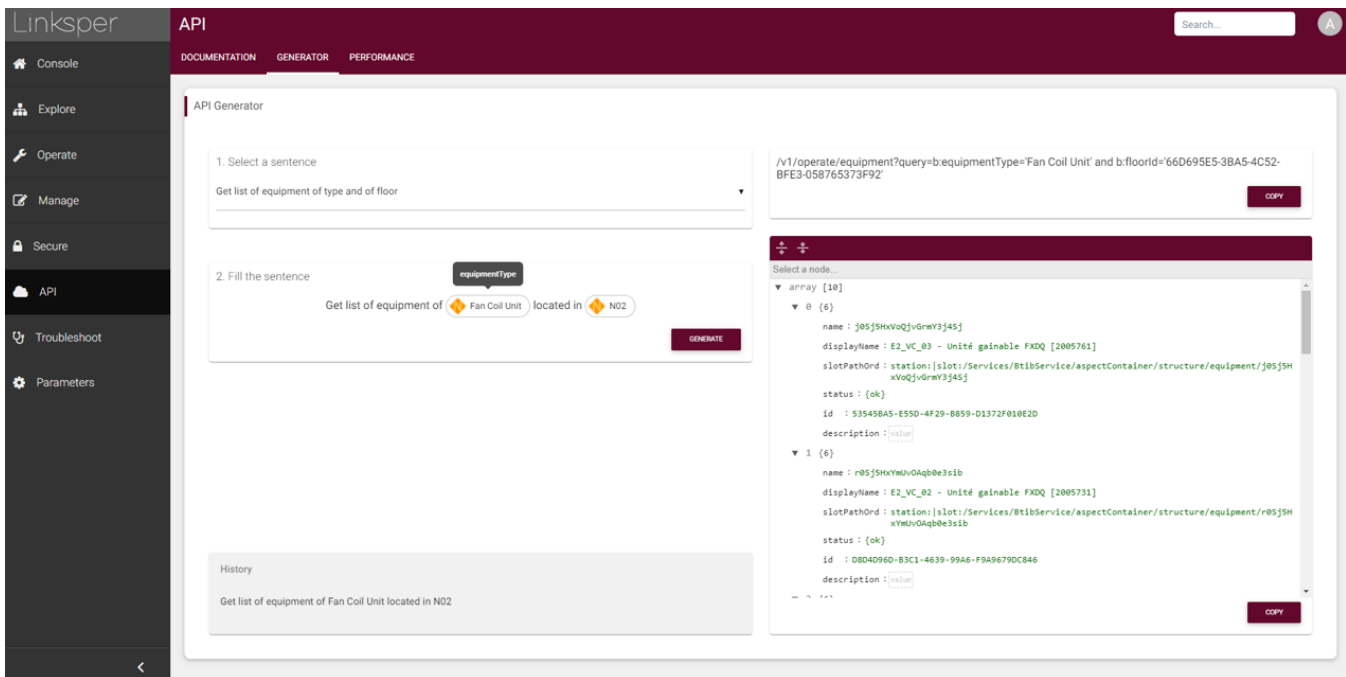
- **base-api-uri**: the local url of Linkspers Center instance (localhost if you are on the same machine)
- **username**: this is the API username given by the BOS integrator
- **password**: Associated password
- **token**: will be filled automatically when the authentication succeeds (see below for more info on the auth)



Testing tools

The BOS Linkspers has a built-in query testing tool called API Generator. The interface allows you to select phrases to complete based on building data and translate them directly into a query.

These are some of the more common examples. Since the API is much broader and generic, the list of phrases is not exhaustive of the API's capabilities.



Authenticate with Linkspers

All communications between a third-party system and the Linkspers BOS are done through the Linkspers Console API and use HTTPS by default. The security is done by using the TLS protocol in version 1.2.

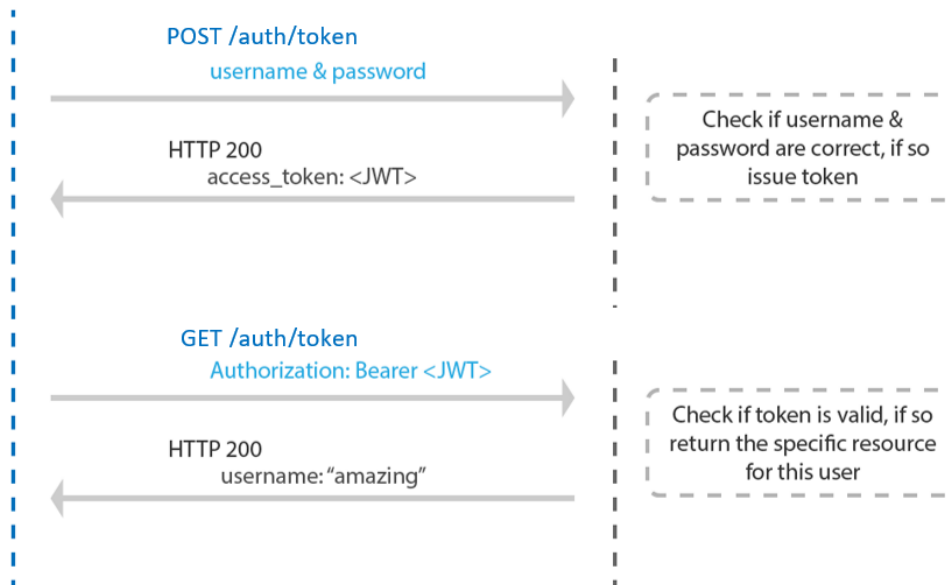
Authentication is required to access the BOS data. This authentication is done by using the JWT technology according to the following scheme.



Browser/Client



Server



A first POST request must be sent to the endpoint: /oauth/token. The body of the request must contain the login and password in JSON format:

```

{
  "username": "{{username}}",
  "password": "{{password}}"
}

```

A token is then returned by the BOS in return.

```

{
  "result": {
    "user": "Service1_API",
    "token": "eyJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJXaW4tRUIxOC9wMzQ3LUY4QjgtQjJGNyIsInN1YiI6ImlnLnZpY2UxX8FQSSIsIm1hdCI6MTYzNzY5MTY4NCwiZmxhZJoxNjM3Njk1Mjg8LCJvYnV0e2Mz20TE200R9.HB8vZ2Mk3g8TX7DQzGsR4jTYeU9tMtIQjMgokVEYkNjcHlIlpkwx1MALNIqy_iJF3Ef_p8dWpZ8BRSYUNMn6AVUcsAGlNgCjHIKrazwCcmWwoaK1Jmpyk9dcVCBFUFl1C9q6g6VSWEqd1jabrl9mm69rlx9utsHM2mMnY1N14k4iRabMPuifK3929tD-goR_oCsHuxFimzPw0RhtDyd7RHVz6hqV1q8lyQomd7q7LvRQzoaKcC6Sk86Ua8a53HDGCF1WHqUNgn9mR0o6W9svwbe1e7axuWa89484_HhU8QeqQP_-smMmE1dz1IN_4J5s41I8MD0nnhDo9zcImvQ"
  }
}

```

This token must be returned as a header for any request to the BOS using the format

Authorization: Bearer < token>

GET
{{base-api-url}}/v1/explore/assets/:id

Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings

Headers
Hide auto-generated headers

	KEY	VALUE
<input checked="" type="checkbox"/>	Authorization ⓘ	Bearer {{token}}

By default the token is valid for 1 hour.

Execute a first request

Once the token is obtained in the previous step, it is possible to trigger its first request:

Example `/v1/explore/hierarchies/Structure?depth=3` allows you to browse the geographical tree of the site

GET

▼

{{(base-api-uri)}}/v1/explore/hierarchies/Structure?depth=4

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Headers

Hide auto-generated headers

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	Authorization ⓘ	Bearer eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJ1aW41RUIxOC0wMzQ3LUY4QigtQjJ	
<input checked="" type="checkbox"/>	Cookie ⓘ	JSESSIONID=2a7321b7a12f8d5f2cfeb75afcaf98f72fb2e60b6bbf8cf6bc.node0	
<input checked="" type="checkbox"/>	Postman-Token ⓘ	<calculated when request is sent>	

▼ / Explore / Hierarchies / Get hierarchy

GET

▼

{{(base-api-uri)}}/v1/explore/hierarchies/Structure?depth=4

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	depth	4	
	Key	Value	Description

Body

Cookies (1)

Headers (9)

Test Results

Status: 200 OK

Tim

Pretty

Raw

Preview

Visualize

JSON

▼

≡

```
1  [
2    "name": "Structure",
3    "descendants": [
4      {
5        "name": "H2K1a2RAXw0x",
6        "displayName": " ",
7        "id": "k0SS2Kq1XzJ6Vf0k8K9mc",
8        "description": "",
9        "lk:assetType": "Model",
10       "descendants": [
11         {
12           "name": "d2K1a9DfoMYg",
13           "displayName": " ",
14           "id": "U0SS2Kq0HaFbawo2CFUCP",
15           "description": "",
16           "lk:assetType": "Model",
17           "descendants": [
18             {
19               "name": "Ascenseurs",
20               "displayName": "Ascenseurs",
21               "id": "g0SpXvHsfGgw3gd4dbreF",
22               "description": "",
23               "lk:assetType": "Equipment",
24               "descendants": [
```

Understand the API structure

The API is structured under 4 main categories:

- **Explore** Discover available data from different perspectives (called "hierarchy"): by a geographic point of view, following the distribution of air through assets...
- **Operate** Deep dive on a particular set of data: look for historical data, alarms, apply commands or create bookings
- **Manage** Apply administrator modifications (change in the model, parameters...)
- **Troubleshoot**: Make sure the system is running properly

Assets

Assets are a common term for both structural parts of a building like a floor, a space or a zone and physical equipment (AHU, LVS, Sensor...). Good to know:

- Each asset may produce data which are collected by the BOS and may provide commands (to act on the equipment) which are available through the BOS as well.
- Each asset has a unique id that can be used across all the queries, a *name* that can be non human friendly and a *display name* which is human friendly.
- Each asset has multiple *metadata* to provide a context, also called *tags*. They are standardized and accessible through the additional field "allTags".
- Each asset has multiple *data types* described below.

See [Linkspers asset types](#) for more information

Explore /Hierarchies

A hierarchy is a perspective, a point of view, to **describe assets relationships** of a building: the geographic structure, the air distribution etc. Linkspers has standardized the hierarchies it offers to third parties, no matter the building, no matter the integrator of the BOS.

The standardized hierarchies are listed here: [Linkspers assets hierarchies](#)

All these hierarchies are not necessary available depending on the systems acquired by the BOS and the level of tagging selected for the site. *Structure* is the only one always available.

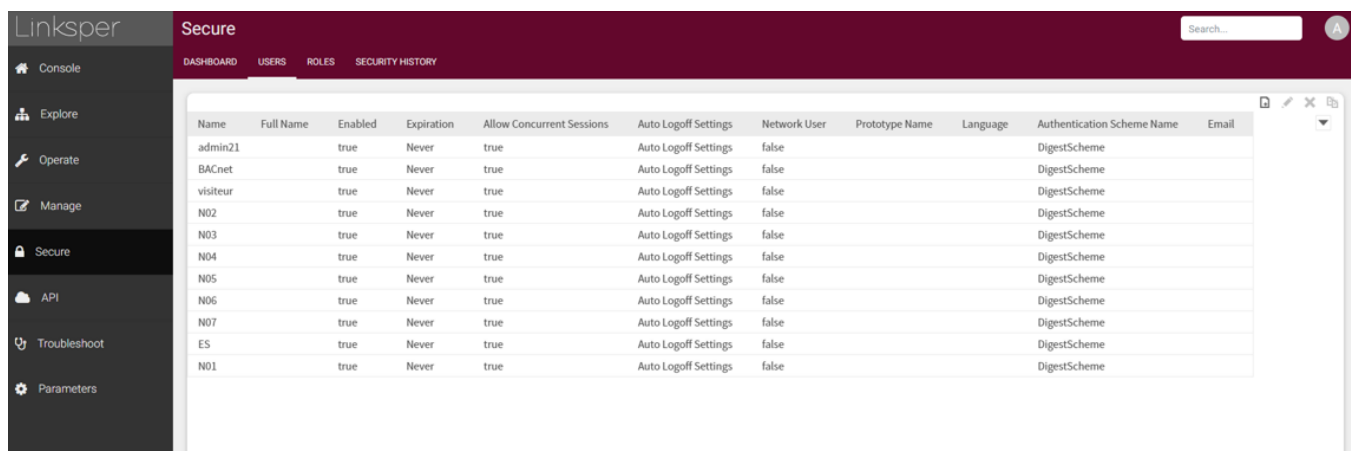
A query to `/v1/explore/hierarchies` will list the available hierarchies.

Manage user rights

Creating a new user to access the API

VayanData recommends the creation of at least one user for API access for each service. Multiple users for the same service can be created, especially if they require different data needs.

To create a new user, access the Linkspers Console application, the "Secure" menu and the "Users" tab.



Name	Full Name	Enabled	Expiration	Allow Concurrent Sessions	Auto Logoff Settings	Network User	Prototype Name	Language	Authentication Scheme Name	Email
admin21		true	Never	true	Auto Logoff Settings	false			DigestScheme	
BACnet		true	Never	true	Auto Logoff Settings	false			DigestScheme	
visiteur		true	Never	true	Auto Logoff Settings	false			DigestScheme	
N02		true	Never	true	Auto Logoff Settings	false			DigestScheme	
N03		true	Never	true	Auto Logoff Settings	false			DigestScheme	
N04		true	Never	true	Auto Logoff Settings	false			DigestScheme	
N05		true	Never	true	Auto Logoff Settings	false			DigestScheme	
N06		true	Never	true	Auto Logoff Settings	false			DigestScheme	
N07		true	Never	true	Auto Logoff Settings	false			DigestScheme	
E5		true	Never	true	Auto Logoff Settings	false			DigestScheme	
N01		true	Never	true	Auto Logoff Settings	false			DigestScheme	

Click on the New button located at the bottom of the page, then

- Enter a name (avoid special characters)
- Check one or more roles (to create new roles, see the next section on managing access rights)

- Select **JwtAuthenticationScheme**
- Set a password

Name	Full Name	Enabled	Expiration	Roles	Allow Concurrent Sessions	Auto Logoff Settings	Network User	Prototype Name	Language	Authentication Scheme Name	A
Service1_API		true	Never	admin	true	Auto Logoff Settings	false	defaultPrototype		JwtAuthenticationScheme	P
<div> <div>Name</div> <div>Service1_API</div> </div>											
<div> <div>Full Name</div> <div></div> </div>											
<div> <div>Enabled</div> <div><input checked="" type="checkbox"/> true</div> </div>											
<div> <div>Expiration</div> <div> <input checked="" type="radio"/> Never Expires <input type="radio"/> Expires On <div>23-Nov-21 11:59 PM</div> </div> </div>											
<div> <div>Roles</div> <div> <input checked="" type="checkbox"/> admin <input type="checkbox"/> Administrator <input type="checkbox"/> IoTIntegrator </div> </div>											
<div> <div>Allow Concurrent Sessions</div> <div><input checked="" type="checkbox"/> true</div> </div>											
<div> <div>Auto Logoff Settings</div> <div> Auto Logoff Enabled <input checked="" type="checkbox"/> true Use Default Auto Logoff Period <input checked="" type="checkbox"/> true Auto Logoff Period <div>0 h 15 m</div> </div> </div>											
<div> <div>Network User</div> <div><input type="checkbox"/> false</div> </div>											
<div> <div>Prototype Name</div> <div>defaultPrototype</div> </div>											
<div> <div>Language</div> <div></div> </div>											
<div> <div>Authentication Scheme Name</div> <div>JwtAuthenticationScheme</div> </div>											
<div> <div>Authenticator</div> <div>Password Authenticator</div> </div>											
<div> <div>Password</div> <div> Password <div>*****</div> Confirm <div>*****</div> </div> </div>											
<div> <div>Password Config</div> <div>User Password Configuration</div> </div>											
<div> <div>Email</div> <div></div> </div>											
<div> <div>Cell Phone Number</div> <div></div> </div>											

You can test the authentication directly in Postman by replacing the variables with the values you entered.

Access rights management

Linkspers has 4 levels of user rights management:

- Roles are associated with each user
- Each role has read, write and invocation rights on 2 levels: operator and administrator
- Each component of the platform (e.g., graph node) is associated with one or more categories.
- Each property of a component can be defined at operator or administrator level.









The creation of new user rights must be done by a Niagara 4 certified system integrator. This rights management is part of the official TCP Niagara training.

Filtering by Endpoint

The Linkspers API is composed of "Endpoints" that allow access to data of various kinds (real-time point readings, historical readings, global commands, etc.). Rights on each endpoint can be defined independently of the user rights. It is thus possible to restrict a user only to reading real time points even though he has user rights to send commands or to read the history of the Niagara viewpoint.

The rights on the endpoints are configured here: Services > WebService > LinkspersV1Servlet > endpoints

Endpoints (Dossier)

▶  get/v1/explore/assets/:id	Composant
▶  get/v1/explore/assets/:id/alarms	Composant
▶  get/v1/explore/assets/:id/devices	Composant
▶  get/v1/explore/assets/:id/equipment	Composant
▶  get/v1/explore/assets/:id/histories	Composant
▶  get/v1/explore/assets/:id/points	Composant
▶  get/v1/explore/assets/:id/schedules	Composant
▶  get/v1/explore/assets/:id/synthesisHistories	Composant
▶  get/v1/explore/assets/:id/synthesisPoints	Composant
▶  get/v1/explore/hierarchies	Composant
▶  get/v1/explore/hierarchies/:name	Composant
▶  get/v1/manage/models	Composant
▶  get/v1/operate/alarms	Composant
▶  get/v1/operate/alarms/:id	Composant
▶  get/v1/operate/components	Composant
▶  get/v1/operate/devices	Composant
▶  get/v1/operate/devices/:id	Composant

Scope of access rights

The permissions granted to a user of the API cover all the functions of the BOS:

- Real time points
- Historicals
- The alarms
- Time programs
- Data models
- BOS application data
- The files