



# *Cumulocity Connector*

## *PLCnext App*

|          |            |
|----------|------------|
| Editor   | MuOl       |
| Reviewer | JoKo       |
| Revision | 01         |
| Date     | 22.04.2022 |

Index

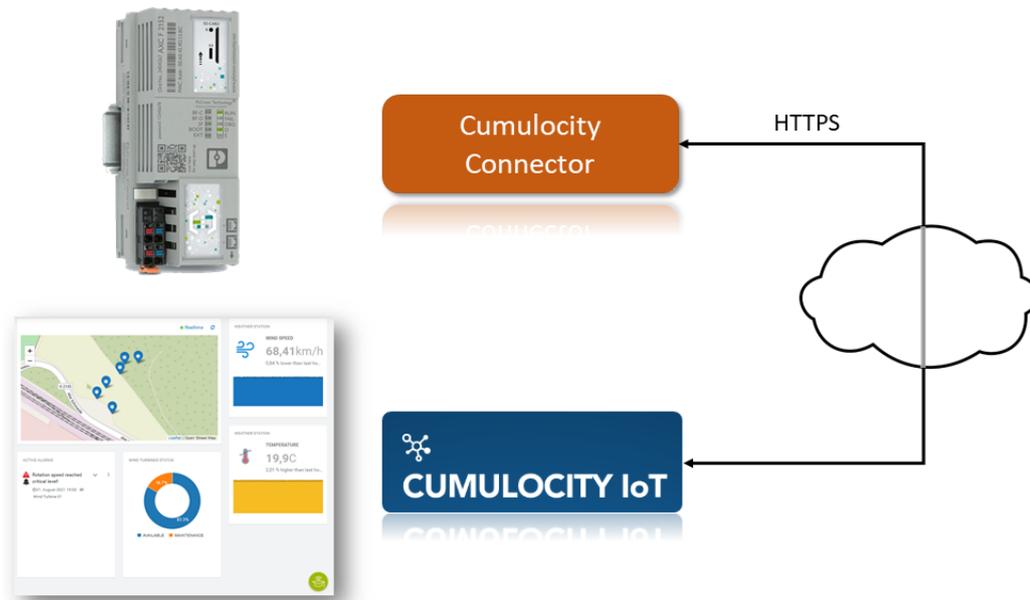
| Rev | Date       | Editor | Changes                            |
|-----|------------|--------|------------------------------------|
| 01  | 22.04.2022 | MuOI   | The first version of this document |
|     |            |        |                                    |
|     |            |        |                                    |
|     |            |        |                                    |
|     |            |        |                                    |

# Content

|  |           |
|--|-----------|
| <b>1. Introduction</b>   | <b>4</b>  |
| 1.1. Feature Set   | 5         |
| 1.2. Requirements  | 5         |
| <b>2. Cumulocity Connector App Installation</b>                              | <b>6</b>  |
| 2.1. App Installation Steps  | 6         |
| 2.2. App Configuration Files   | 8         |
| <b>3. Cumulocity IoT Device Registration</b>                                 | <b>11</b> |
| 3.1. Free Trial Setup (optional)   | 11        |
| 3.2. Device Registration   | 13        |
| <b>4. Cumulocity IoT Device Management</b>                                   | <b>17</b> |
| 4.1. Info  | 17        |
| 4.2. Measurements  | 18        |
| 4.3. Alarms  | 18        |
| 4.4. Configuration   | 19        |
| 4.5. Control   | 23        |
| 4.6. Events  | 24        |
| 4.7. Location  | 24        |
| 4.8. Service Monitoring  | 25        |
| 4.9. Identity  | 26        |
| <b>5. Troubleshooting</b>  | <b>27</b> |
| 5.1. SDKException: Http status code: 401                                     | 27        |
| 5.2. UnknownHostException  | 27        |
| 5.3. SokPlcConnectionException: Error connecting to PLC                      | 28        |
| 5.4. CommonRemotingFatalException: Communication error - no data from server | 28        |

## 1. Introduction

The Cumulocity Connector App is in terms of *Internet of Things* (IoT) a device agent, since it is connecting PLCnext devices produced by Phoenix Contact with the Cumulocity IoT platform which is a product from Software AG. Therefore, on the one hand the App will communicate over the device specific protocol with the PLCnext device and on the other hand the App will forward and fetch data over HTTPS with Cumulocity. For exchanging data with Cumulocity the Cumulocity data model must be used and therefore, the App will transform the device specific data.



## 1.1. Feature Set

The Cumulocity IoT platform has a lot of different device features out of the box. Some of them are also depending on the communication protocol used for device integration. The current feature set supported by the Cumulocity Connector App are:

- Alarms
- Events
- Location & Tracking
- Measurements
- Restart
- Service Monitoring

## 1.2. Requirements

To start with the installation of the Cumulocity Connector App and performing the device registration, following tools are required:

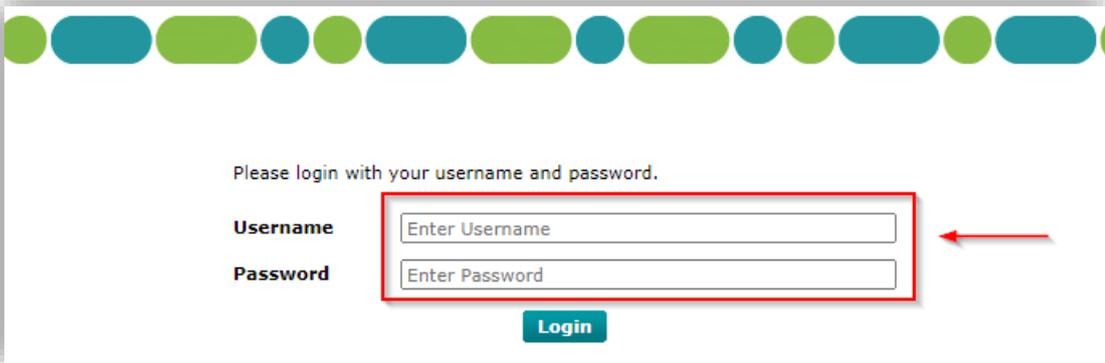
- Cumulocity Connector App
- Phoenix Contact PLCnext Device
- Web Browser
- SFTP Client
- Cumulocity IoT Platform

## 2. Cumulocity Connector App Installation

This chapter describes how the Cumulocity Connector App is installed on the PLCnext device. For that the Web-based Management (WBM) of PLCnext devices is used in which static and dynamic device information can be accessed and certain device settings can be modified which also includes installing, uninstalling, starting and stopping apps. The WBM can be opened if a proper Ethernet connection to the PLCnext device has been established. For more information on establishing a connection to WBM, please look at the official WBM documentation by Phoenix Contact: <https://www.plcnext.help/te/WBM/WBM.htm>

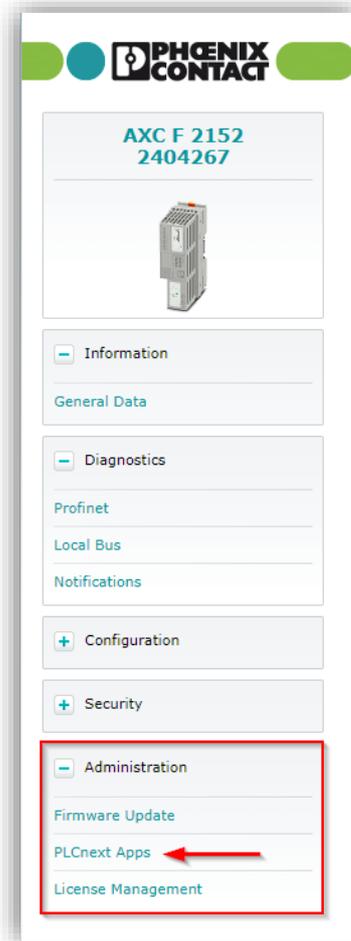
### 2.1. App Installation Steps

- Start your PLCnext device
- Connect the PLCnext device via Ethernet with your laptop/pc
- Open the web browser of your choice and enter the PLCnext device URL <https://IP.address.of.controller> (e.g. <https://192.168.10.1/>).
- Login to WBM
  - Enter your username in the “Username” input field
  - Enter your password in the “Password” input field
  - To open WBM, click on the “Login” button

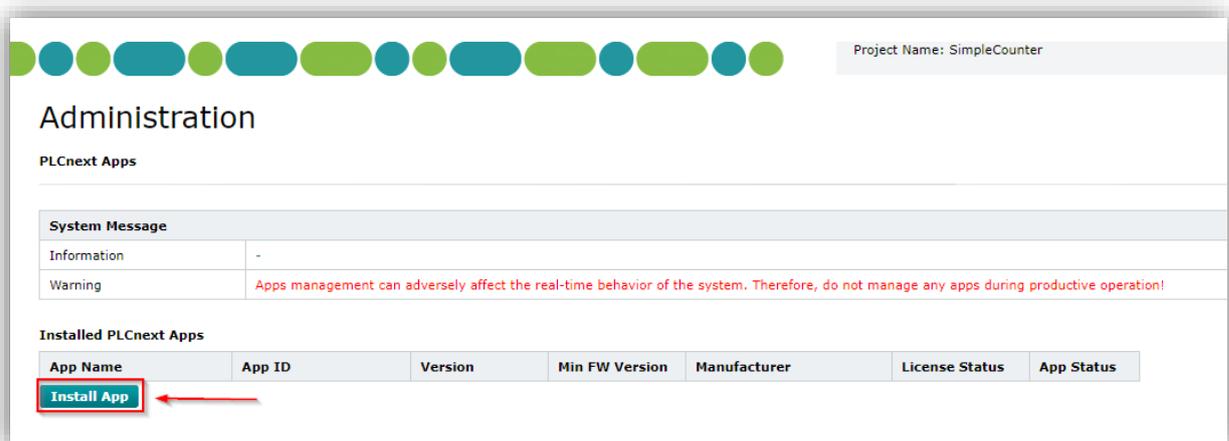


The screenshot shows a web browser window with a login form. At the top, there is a decorative header with a row of colored circles in shades of green and teal. Below the header, the text "Please login with your username and password." is displayed. The form consists of two input fields: "Username" and "Password". The "Username" field contains the placeholder text "Enter Username" and the "Password" field contains "Enter Password". A red rectangular box highlights both input fields, and a red arrow points to the right side of the box. Below the input fields is a blue "Login" button.

- On the Administration panel select the "PLCnext Apps" tab

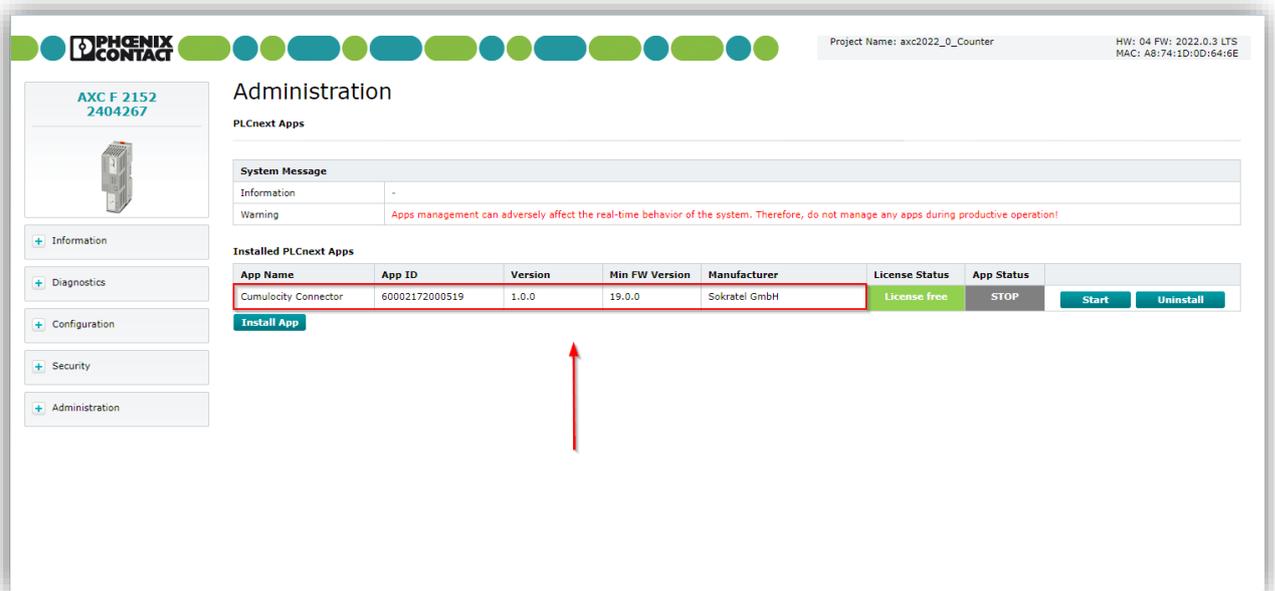


- Click on the "Install App" button



- In the file explorer that opens, select CumulocityConnector.app

- Click on the “Open” button
- Wait until the installation is completed
- Start the app by pressing the “Start” button



**Note:** We need to start the app now, so we can access the config templates. After proper configuration, the app must be restarted.

## 2.2. App Configuration Files

The Cumulocity Connector App configuration files application.properties and logback.xml must be placed in the proper directory of the PLCnext device:

- Connect to your PLCnext device with SFTP client (e.g. WinSCP)
- On the PLCnext device move to folder `"/opt/plcnext/appshome/data/"` and create here a folder with name `"60002172000519"` if it does not already exist (from here the app will load all configurations at startup)
- Copy the application.properties and logback.xml contained in the `"/opt/plcnext/apps/60002172000519/config-templates"` folder into `"/opt/plcnext/appshome/data/60002172000519"`
- Modify the config files in the `"/opt/plcnext/appshome/data/60002172000519"` directory so that at least all mandatory fields are set

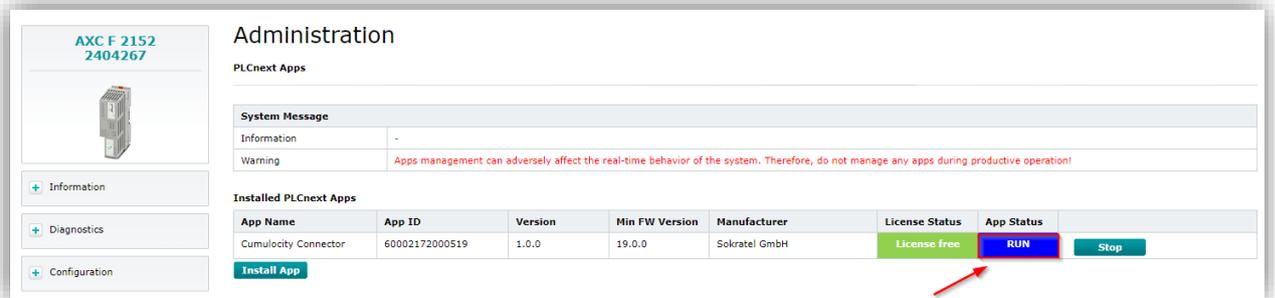
There is no need in modifying the logback.xml file since it holds the default logging configuration. In contrast to that, the application.properties file must be adapted. The most critical configuration must be done in line 4, 20, 22, 31, 32 and 35. All configuration fields are documented in the properties file itself, so please have a look:

```

1 #-----Mandatory configuration for connection to c8y platform-----
2
3 # Address of the platform. Provided by the deployment process, no default value, e.g. https://env757303.eu-latest.cumulocity.com/
4 C8y.baseUrl=
5
6 # These settings Control the device bootstrap process of the Cumulocity-Connector.
7 # In general, the default settings are sufficient, and should not be Changed.
8 # Contact product support (https://cumulocity.com/guides/<latest-release>/welcome/contacting-support/)
9 # in case the bootstrap credentials are different.
10
11 # Tenant ID to be used for device bootstrap, DO NOT CHANGE THIS UNLESS YOU KNOW EXACTLY WHAT YOU ARE DOING.
12 C8y.bootstrap.tenantId=management
13 # Credentials for the device bootstrap user, DO NOT CHANGE THIS UNLESS YOU KNOW EXACTLY WHAT YOU ARE DOING.
14 C8y.bootstrap.username=devicebootstrap
15 C8y.bootstrap.password=Fhdt1bb1f
16
17 #-----Mandatory configuration for the c8y device-----
18
19 # Name of the device, no default value, e.g. phoenix-plc-device
20 sok.c8y.device.name=
21 # Note: The same device identifier should be used for device registration on Cumulocity IoT, no default value, e.g. phoenix-plc-device
22 sok.c8y.device.identifier=
23
24 #-----Mandatory configuration for connecting to PLC-----
25
26 # A PLCnext user with the "Admin" role must be entered here.
27 # We recommend to create a dedicated user for this App.
28 # If you want the App to encrypt these information at startup use the encrypt tag, e.g.:
29 # sok.phx.plc.username=DEC(your_username) -- after startup --> sok.phx.plc.username=ENC(7dWn/EQBuB9cLquG6mJQJu9cUSLa2sy9CSx8uWIIc6U=)
30 # sok.phx.plc.password=DEC(your_password) -- after startup --> sok.phx.plc.password=ENC(ZNOxB85EjkKzeqkyRtLHJ1EZCTGhFEa9pSh9dqPuNnI=)
31 sok.phx.plc.username=
32 sok.phx.plc.password=
33 # Enabling or disabling encryption for RSC communication, configured on PLC: /etc/plcnext/device/System/RscGateway/RscGateway.settings
34 # MAKE SURE THAT THIS VALUE IS IDENTICAL WITH THE RSCGATEWAY SETTINGS ON THE PLC, no default value, e.g. false
35 sok.phx.plc.encrypted=
36
37 #-----Optional configuration for the c8y device-----
38
39 # Type of the C8y device, default: PhoenixPLC
40 sok.c8y.device.type=
41 # The identifier of the device hardware model, no default value, e.g. BCM270899
42 sok.c8y.device.hardware.model=
43 # The identifier of the hardware revision, no default value, e.g. 000e
44 sok.c8y.device.hardware.revision=
45 # The hardware serial number of the device, no default value, e.g. 00133700e2f5ad4d
46 sok.c8y.device.hardware.serialNumber=
47 # Position: reports the geographical location of an asset in terms of latitude(position.lat),
48 # longitude(position.lng) and altitude(position.alt), no default value, e.g. 13.37, 10.000115, 53.676344
49 sok.c8y.device.position.alt=
50 sok.c8y.device.position.lng=
51 sok.c8y.device.position.lat=
52 # Expected device Communication interval in minutes, default: 10
53 sok.c8y.device.response.interval=
54
55 # -----Optional configuration for connecting to PLC-----
56 # If you are not using the default PLC configuration for RSC (/etc/plcnext/device/System/RscGateway/RscGateway.settings),
57 # then you may need to set your specific RSC configuration here:
58
59 # Ip address of the PLC, default: 127.0.0.1
60 sok.phx.plc.ip=
61 # Determines the TCP listener port, default: 41100
62 sok.phx.plc.port=
63 # Connection timeout (milliseconds), default: 10000
64 sok.phx.plc.connecttimeout=
65 # The remoting receive timeout (milliseconds), default: 10000
66 sok.phx.plc.receiveivetimeout=
67
68 #-----Other configuration, do not change this-----
69
70 spring.main.lazy-initialization=true

```

After a proper configuration has been set, re-start the App in the WBM by pressing the stop button, wait, and then pressing the start button again. The App status RUN is then indicated in the column App Status.



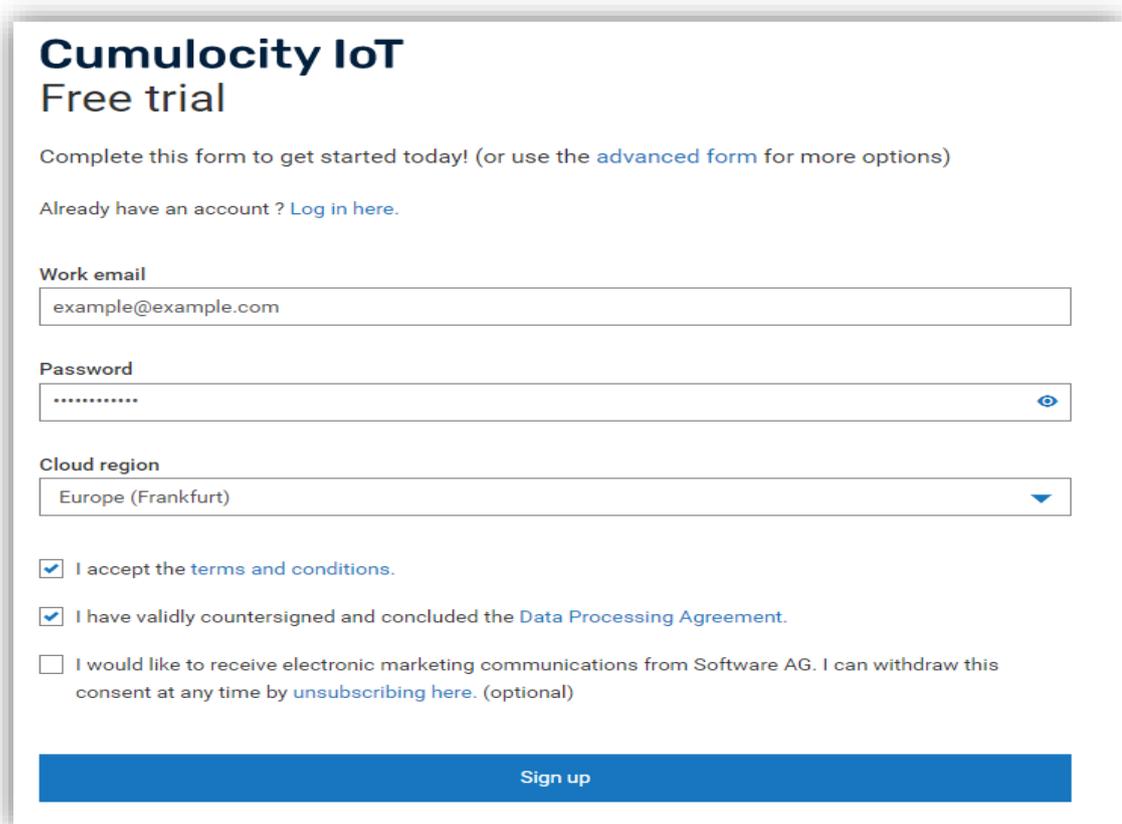
**Note:** Make sure that your PLCnext device has proper network configuration so it can reach the Cumulocity IoT platform configured in line 4 of the application.properties file. If you do not have a Cumulocity IoT platform yet, you can sign up for a free trial as described in the next chapter.

### 3. Cumulocity IoT Device Registration

This chapter contains a guide on how to create a free Cumulocity IoT trial which should be done if you do not have a Cumulocity IoT platform yet. After that the device registration process is shown for integrating the PLCnext device on the platform.

#### 3.1. Free Trial Setup (optional)

- Start free Cumulocity IoT trial on Software AG page: <https://signup.softwareag.cloud/#/?product=cumulocity>
- Complete this form



**Cumulocity IoT**  
**Free trial**

Complete this form to get started today! (or use the [advanced form](#) for more options)

Already have an account ? [Log in here.](#)

**Work email**  
example@example.com

**Password**  
..... 

**Cloud region**  
Europe (Frankfurt) 

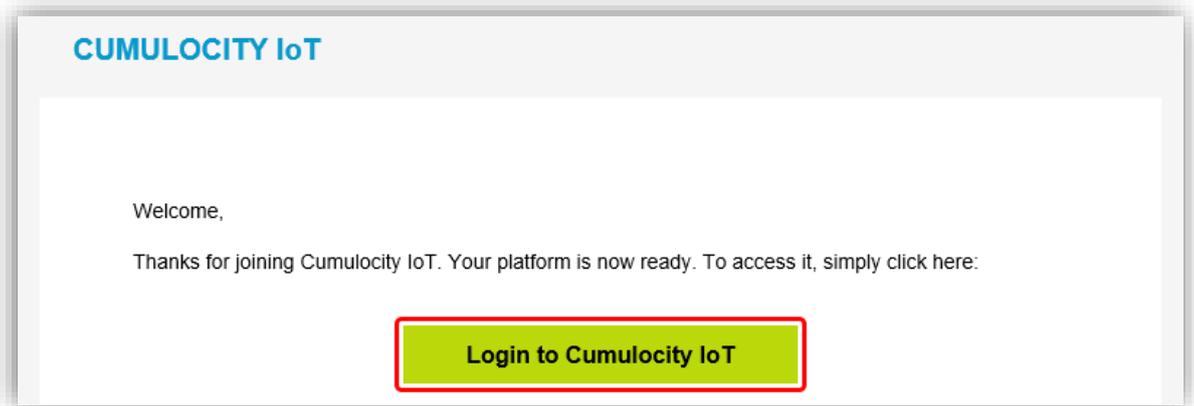
I accept the [terms and conditions](#).

I have validly countersigned and concluded the [Data Processing Agreement](#).

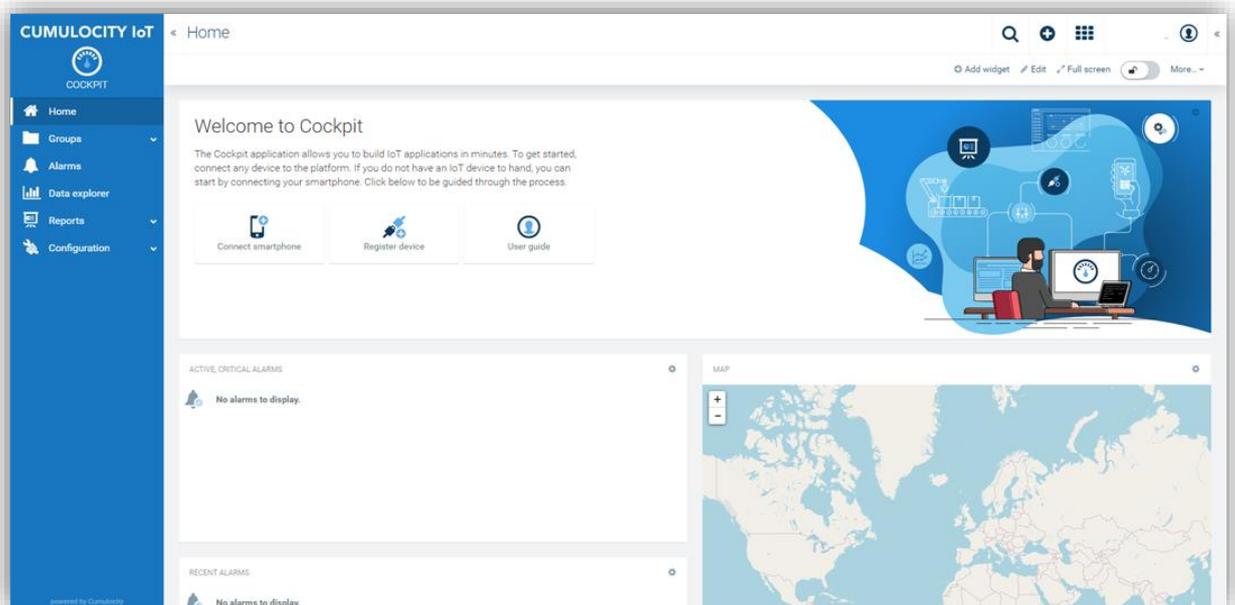
I would like to receive electronic marketing communications from Software AG. I can withdraw this consent at any time by [unsubscribing here](#). (optional)

**Sign up**

- Wait a few minutes after the registration, open your email account and look for an email with the subject "Welcome to Cumulocity IoT". Open it and click "Login to Cumulocity IoT".



- Upon successful login you should be redirected to the Cumulocity Cockpit.

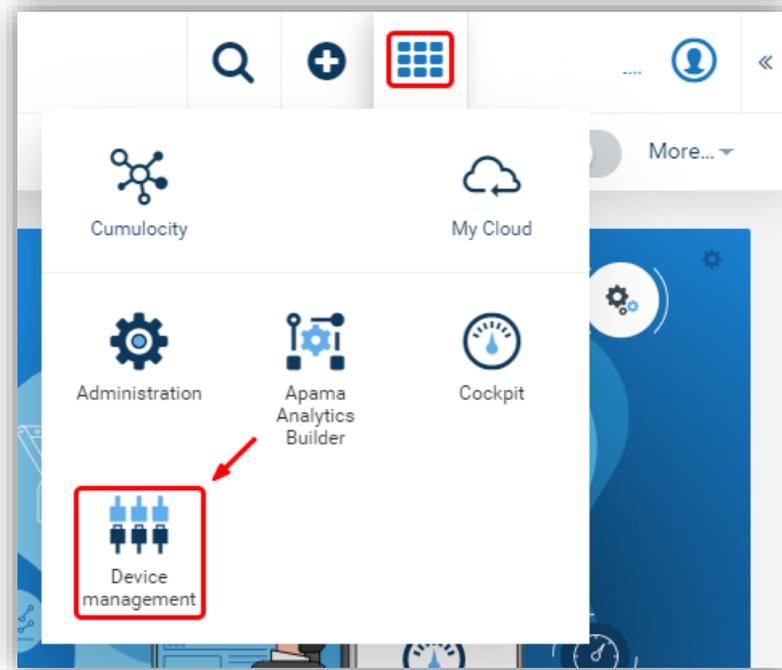


Now you are ready to integrate a PLCnext device which has a running Cumulocity Connector App with proper configuration.

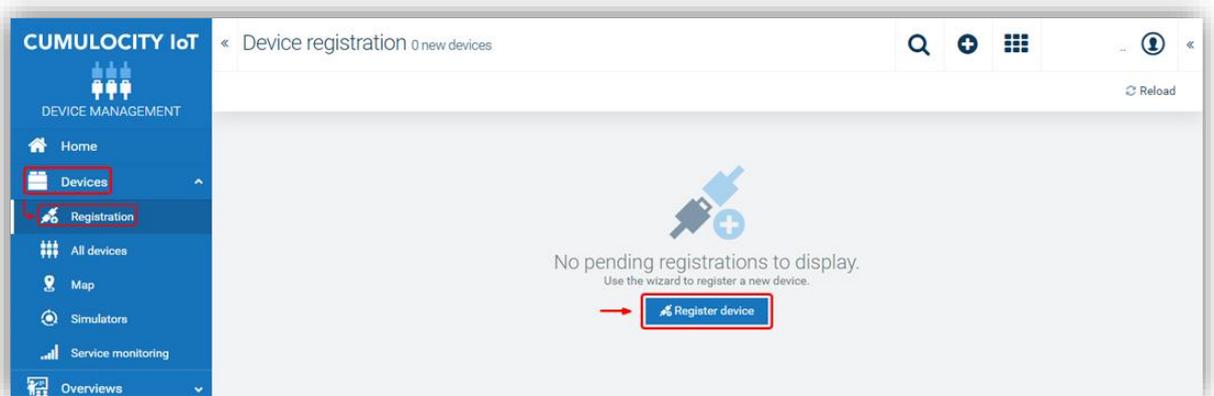
### 3.2. Device Registration

Registration of PLCnext device on Cumulocity IoT:

- Find the “Application switcher” in the top-right corner of the Cumulocity dashboard and go to “Device management”.

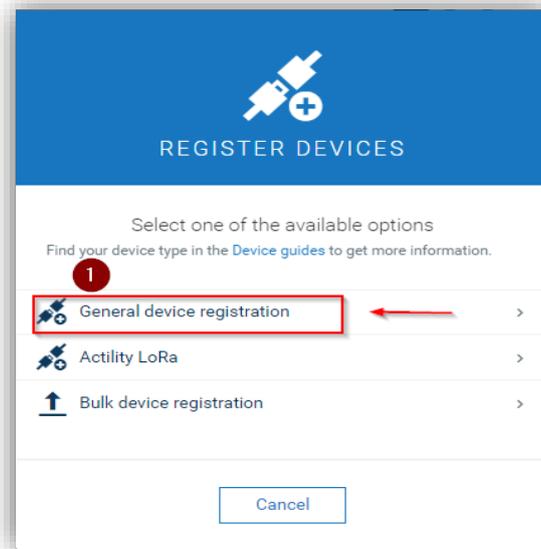


- Then expand the “Devices” tab in the left-hand menu, go to “Registration” and click “Register device”.

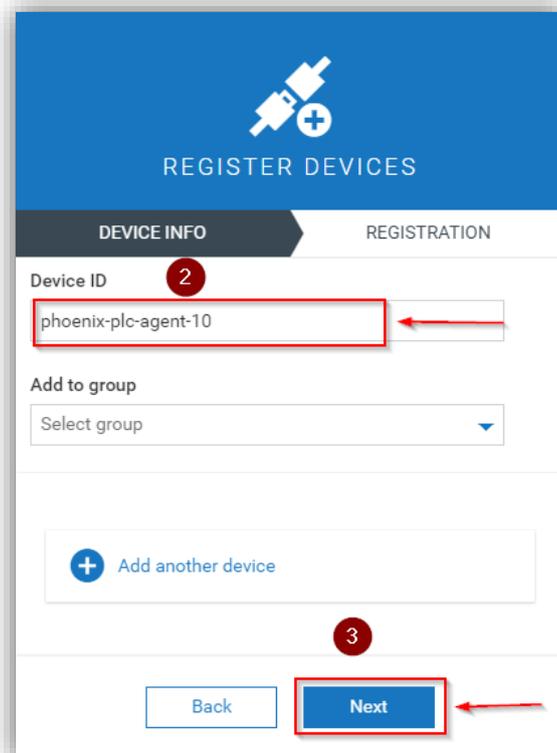


Device registration steps:

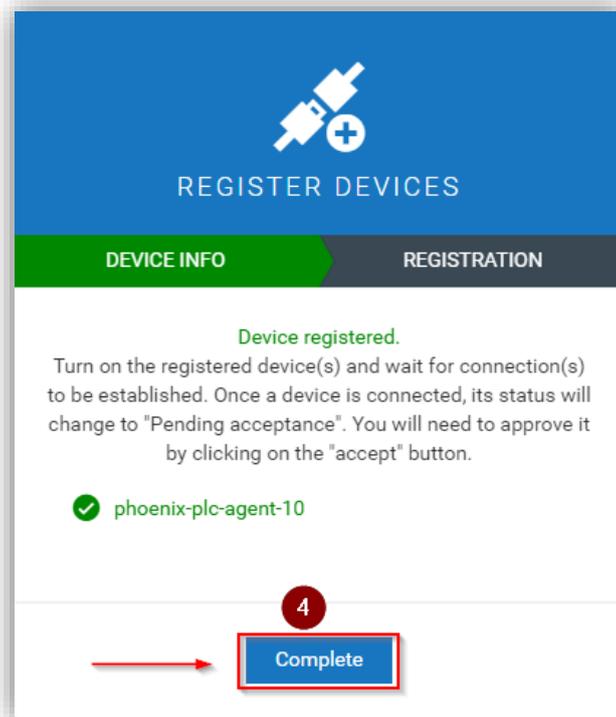
- **Step 1:** Choose “General device registration”



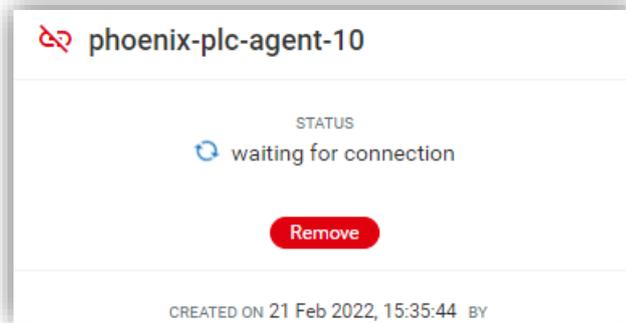
- **Step 2:** In the *Device ID* field enter the device identifier configured in line 22 of the application.properties.
- **Step 3:** Click “Next” to register the device



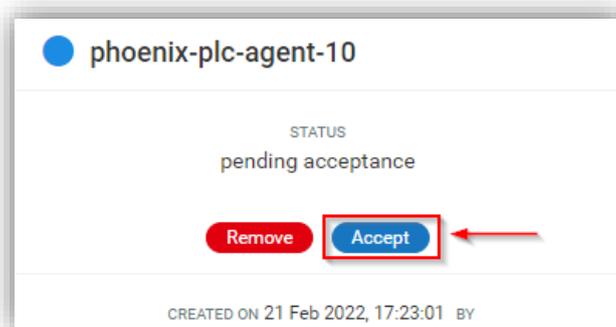
- **Step 4:** Click “Complete” to finish the registration.



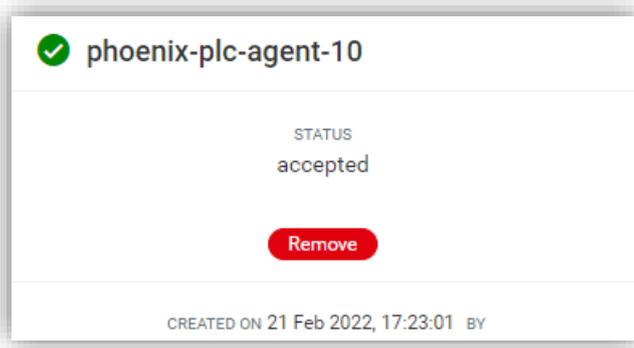
Cumulocity IoT is now waiting for the device to be start communication.



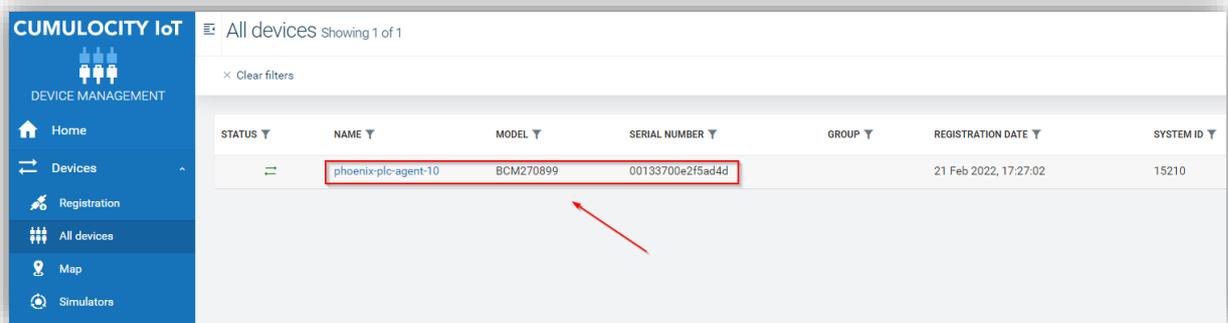
Since we already have the Cumulocity Connector App running, a pending device acceptance should be displayed soon.



Click on the "Accept" button and the status of the device will change to "accepted".



Now the PLCnext device was successfully registered. Click on the “All devices” tab under “Devices” on the right side of the page. A new device with the name configured in line 20 of the application.properties is shown.



## 4. Cumulocity IoT Device Management

This chapter describes the device management features available on the Cumulocity IoT platform for the PLCnext device.

### 4.1. Info

In the info section all relevant device information is visible. The connection status of the device as well as device specific hardware information can be seen if these have been configured in the application.properties.

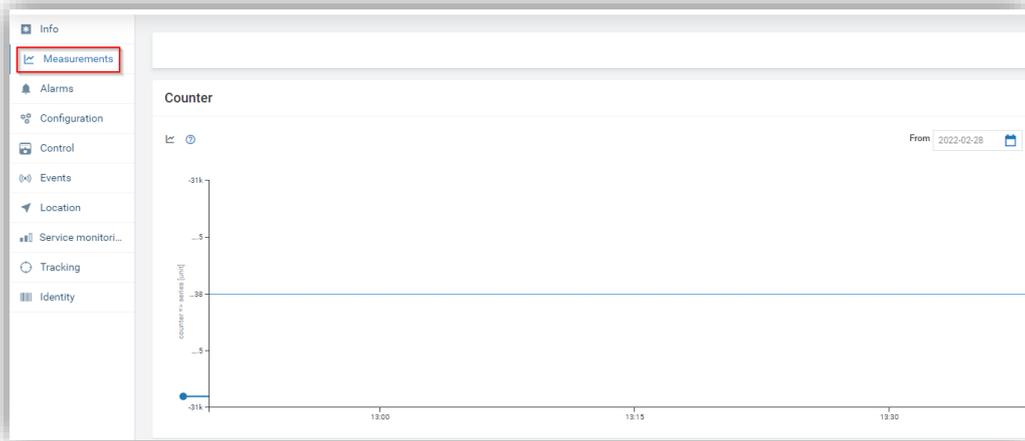
The screenshot displays the 'Info' page for a device in the Cumulocity IoT platform. The left sidebar contains navigation options: Info (highlighted), Alarms, Configuration, Control, Events, Location, Service monitoring, and Identity. The main content area is divided into several sections:

- Notes:** A megaphone icon and the text "No notes yet. Edit".
- DEVICE STATUS:** A green circular icon with two white arrows pointing in opposite directions. Below it, the text reads: "Send connection: online", "Push connection: active", and "Last communication: 20 Apr 2022, 14:16:07".
- Configuration:** A list of settings:
  - Required interval: 10 minutes (with an edit icon)
  - Maintenance:  (disabled)
  - Owner:  device\_phoenix-plc-... (with an edit icon)
- DEVICE DATA:** A table with the following information:
 

| DEVICE DATA   |                          |
|---------------|--------------------------|
| ID            | 9512                     |
| Name          | phoenix-plc-agent-10     |
| Type          | PhoenixPLC               |
| Last updated  | 2022-04-20T12:19:13.934Z |
| Creation time | 2022-04-20T12:07:59.396Z |

## 4.2. Measurements

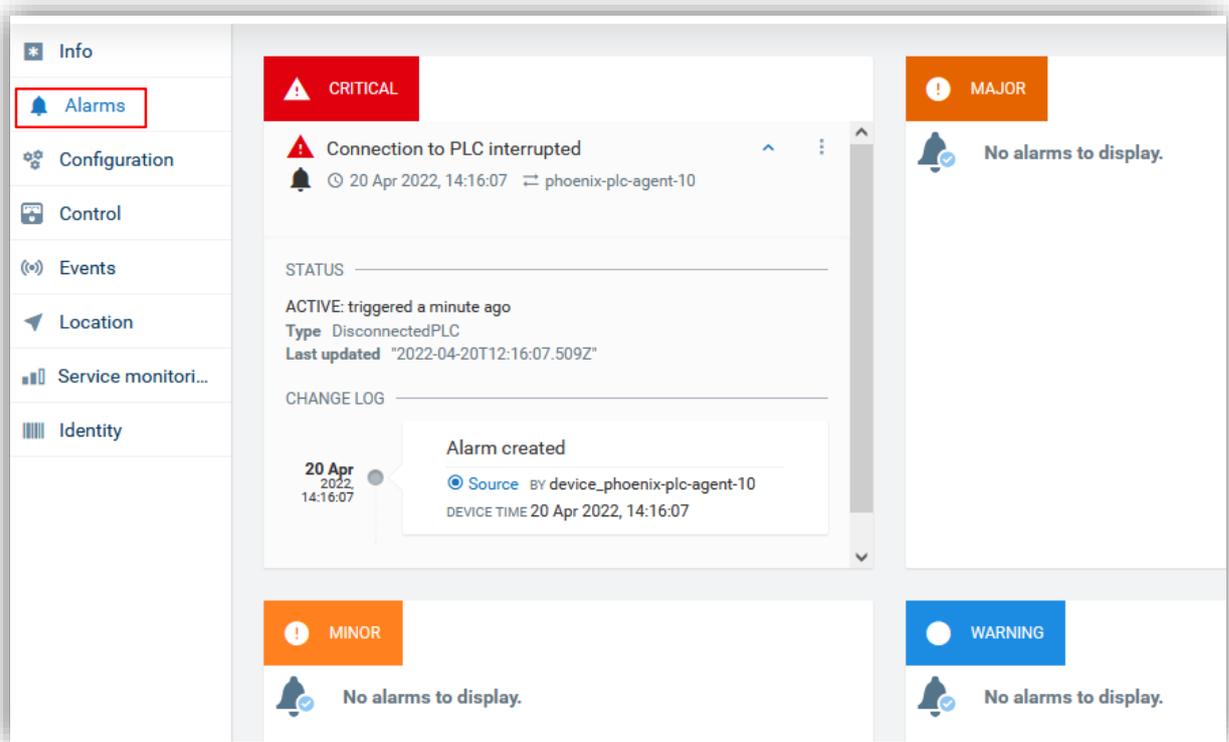
Looking at the device in Cumulocity IoT for the first time, the measurements section is not visible due to the fact that there are no measurements created per default for the device. Specific signals of interest can be set in the configuration section and then if a measurement should be created for a signal it would be visible here.



## 4.3. Alarms

In the alarms section all currently active alarms are shown. There is one alarm of type “c8y\_UnavailabilityAlarm” which will be automatically created by Cumulocity IoT if the connection to the PLCnext device (Cumulocity Connector App) is interrupted, so that no data was exchanged in the minimum required time interval.

In case that the Cumulocity IoT connection is fine but the Cumulocity Connector App is disconnected to the PLCnext device a specific alarm of type “DisconnectedPLC” is created. It is automatically cleared if connection is established again.

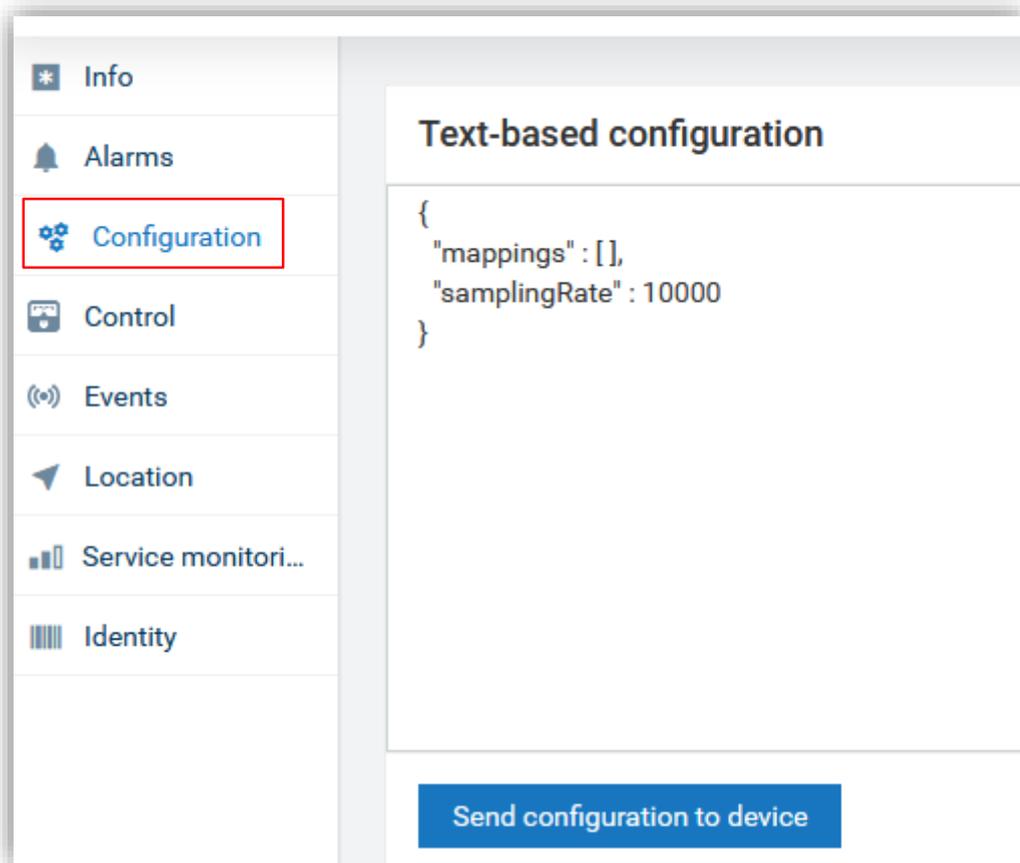


#### Alarms List

| Text   | Type                    | Description   |
|--|-------------------------|---|
| No data received from device within required interval. | c8y_UnavailabilityAlarm | Created if connection between Cumulocity IoT and Cumulocity Connector App is interrupted. |
| Connection to PLC interrupted                          | DisconnectedPLC         | Created if connection between Cumulocity Connector App and PLCnext device is interrupted. |

#### 4.4. Configuration

In the configuration section a specific JSON text configuration can be submitted which contains all PLCnext signals which should be transformed and send to Cumulocity IoT. The default configuration holds a configuration with an empty mapping list.



In this list a mapping holds the PLCnext signal of interest defined by the GDS port and the mapping rule(s) on how to transfer this signal to Cumulocity IoT. Here is an example configuration with all three different mappings:

```
{
  "mappings": [
    {
      "port": "Arp.Plc.Eclr/SomeCounter",
      "eventMapping": {
        "type": "CounterEvent",
        "text": "Counter value: ${value}, read at: ${time}"
      },
      "overriddenSamplingRate": 60000
    },
    {
      "port": "Arp.Plc.Eclr/CpuTempCritical",
      "alarmMapping": {
        "text": "CPU temperature ciritcal",
        "type": "CPUTemp",
        "severity": "CRITICAL"
      }
    },
    {
      "port": "Arp.Plc.Eclr/CpuLoadCore1",
      "measurementMapping": {
        "unit": "%",
        "type": "CPULoad",
        "fragmentName": "sok_CPULoad",
        "series": "Core1"
      },
      "overriddenSamplingRate": 15000
    }
  ],
  "samplingRate": 10000
}
```

Modify the configuration according to your preferences and click the „Send the configuration to device“ button. If the JSON syntax is valid and all mandatory fields are present, the configuration will be enabled, highlighted by a configuration update operation which has a successful state. If the JSON configuration is not correct or there was a configuration update issue the configuration update operation will fail.

The different elements of the JSON configuration are explained in more details in the following:

## Configuration Structure

| Field        | Type                     | Mandatory | Description  |
|--------------|--------------------------|-----------|--|
| mappings     | array of signal mappings | yes       | Holds the signals that should be sampled as well as the transformation rules on how the signals should be presented in Cumulocity. |
| samplingRate | integer                  | yes       | Global rate in which the signals should be sampled.  |

## Signal Mapping Structure

| Field                  | Type                | Mandatory | Description  |
|------------------------|---------------------|-----------|--|
| port                   | string              | yes       | Defines the signal of interest identified by the PLCnext GDS port name.    |
| eventMapping           | event mapping       | no        | Defines how the signal should be mapped to a Cumulocity event.             |
| alarmMapping           | alarm mapping       | no        | Defines how the signal should be mapped to a Cumulocity alarm.             |
| measurementMapping     | measurement mapping | no        | Defines how the signal should be mapped to a Cumulocity measurement.       |
| overriddenSamplingRate | integer             | no        | Specific rate in which the signal of the signal mapping should be sampled. |

**Note:** At least one mapping (event, alarm or measurement) should be present.

## Event Mapping Structure

| Field           | Type             | Mandatory | Description   |
|-----------------|------------------|-----------|---|
| type            | string           | yes       | Holds the Cumulocity event type.  |
| text            | string           | yes       | Holds the Cumulocity event text. The placeholder <code>{value}</code> can be used to insert the value of the signal and the placeholder <code>{time}</code> can be used to insert the reading time. |
| staticFragments | array of strings | no        | Static fragments which should be added to the Cumulocity event.   |

**Note:** Event mappings are possible for numeric, text and boolean signals.

## Alarm Mapping Structure

| Field    | Type   | Mandatory | Description   |
|----------|--------|-----------|---|
| type     | string | yes       | Holds the Cumulocity alarm type.  |
| text     | string | yes       | Holds the Cumulocity alarm text.  |
| severity | string | yes       | Holds the Cumulocity alarm severity (CRITICAL, MAJOR, MINOR or WARNING) |

|                 |                  |    |   |
|-----------------|------------------|----|---|
| staticFragments | array of strings | no | Static fragments which should be added to the Cumulocity alarm. |
|-----------------|------------------|----|---|

**Note:** Alarm mappings are possible for boolean and numeric signals. For numeric values the alarm will be active for values > 0 and inactive for values <= 0.

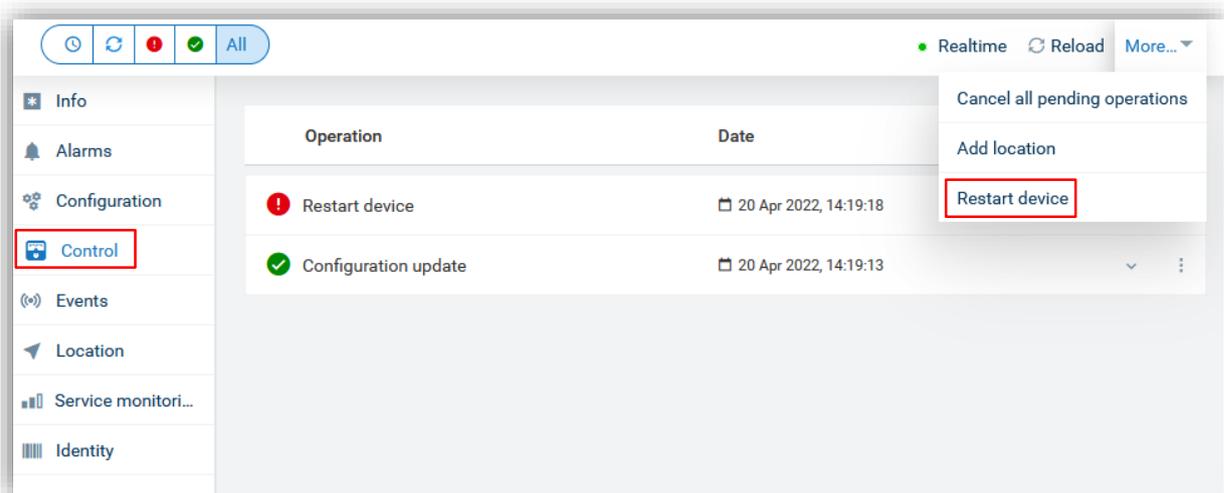
### Measurement Mapping Structure

| Field           | Type             | Mandatory | Description   |
|-----------------|------------------|-----------|---|
| type            | string           | yes       | Holds the Cumulocity measurement type.                                |
| series          | string           | yes       | Holds the Cumulocity measurement series.                              |
| unit            | string           | no        | Holds the Cumulocity measurement unit.                                |
| fragmentName    | string           | yes       | Holds the Cumulocity fragment name.                                   |
| staticFragments | array of strings | no        | Static fragments which should be added to the Cumulocity measurement. |

**Note:** Measurement mappings are only possible for numeric signals.

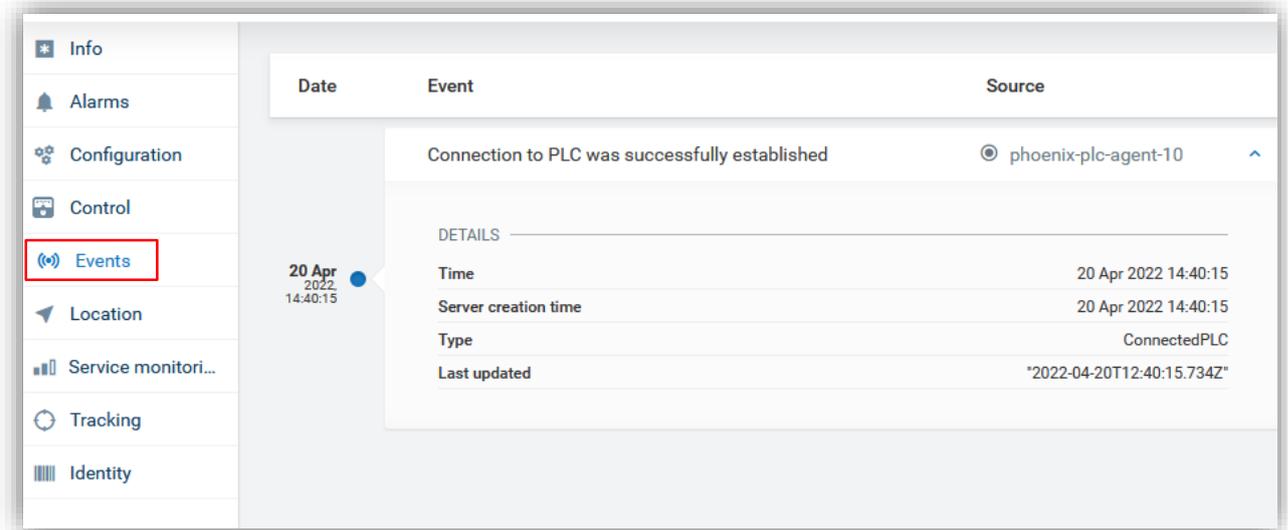
## 4.5. Control

In the control section all operations related to the device are shown. There are currently two supported operations: configuration update and device restart. The device restart can be triggered by clicking on “Restart device” under “More”.



## 4.6. Events

In the event section all device related events are shown. Per default there is only one event created when the Cumulocity Connector App has successfully connected to the PLCnext device.

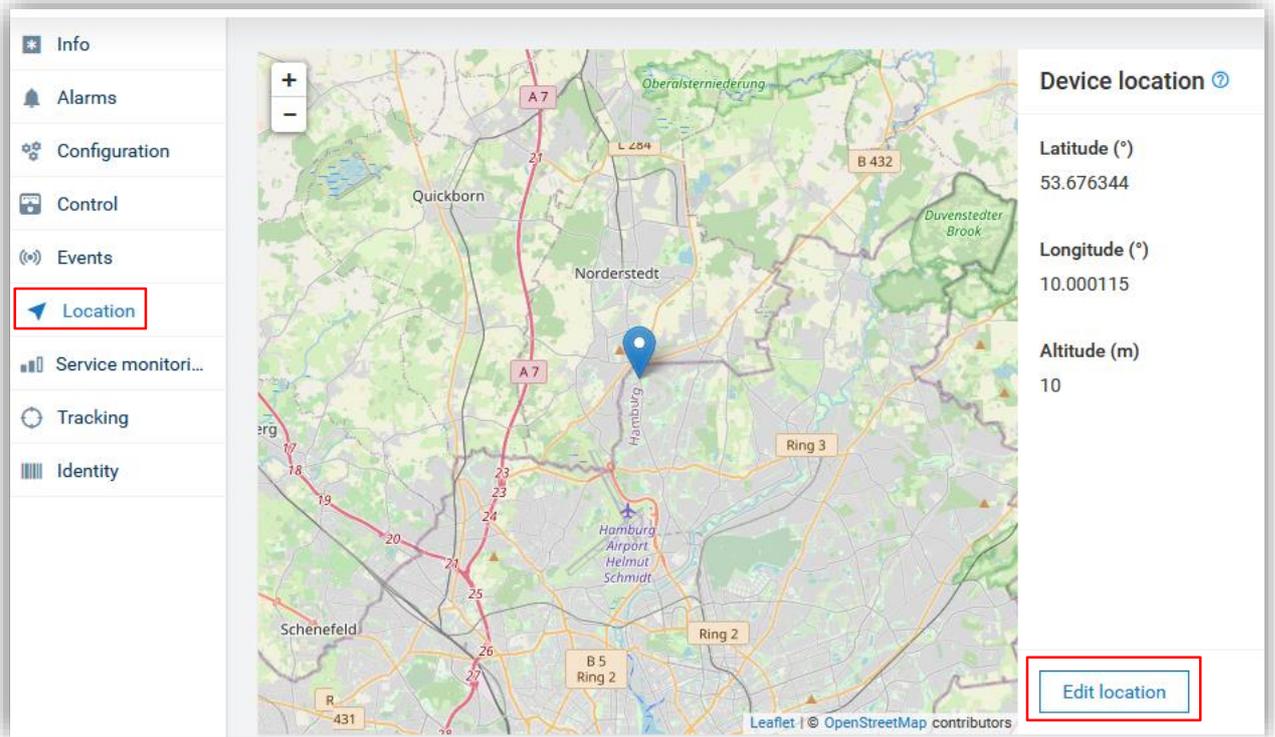


Events List:

| Text   | Type         | Description   |
|--|--------------|---|
| Connection to PLC was successfully established | ConnectedPLC | This event is created when the Cumulocity Connector App connects successfully to the PLCnext device - either first time or due to a re-connect. |

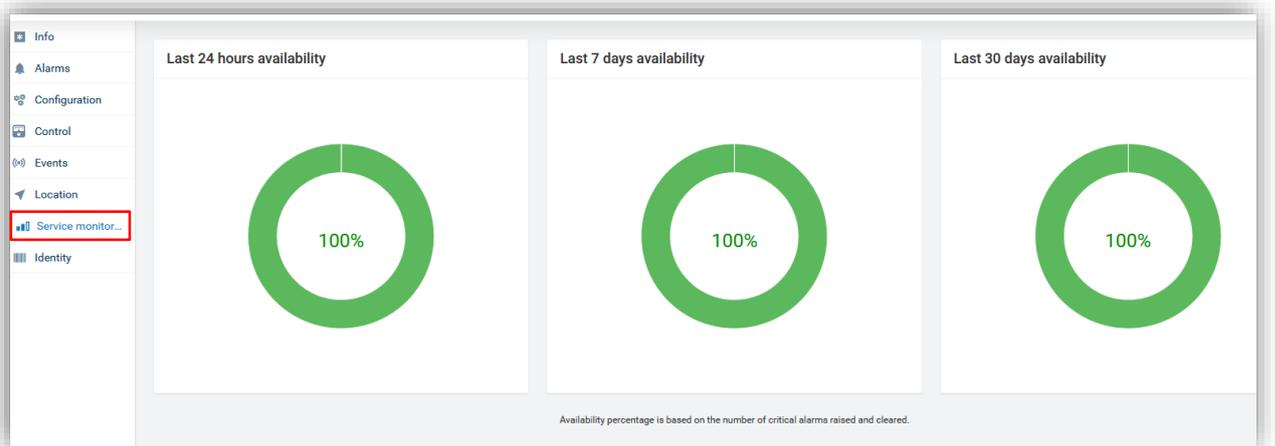
## 4.7. Location

In the location section the geographically location of the device can be seen which was initially configured at the application.properties. By using the "Edit location" button the device location can be updated.



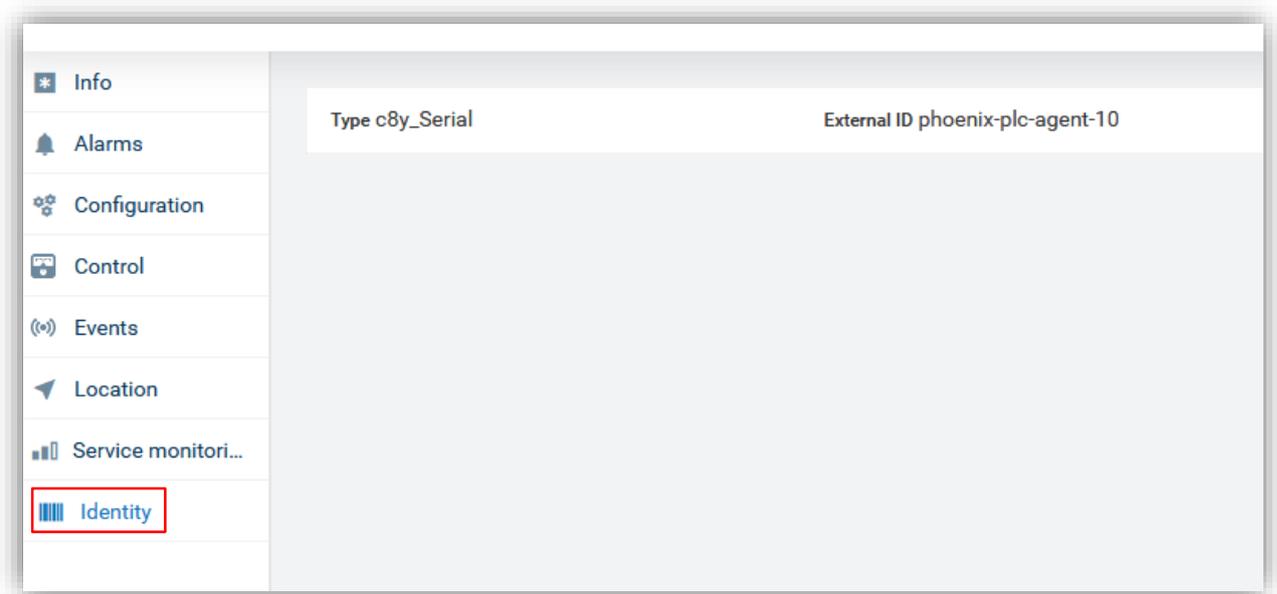
## 4.8. Service Monitoring

The service monitoring section visualizes the device availability in pie charts for the last 24 hours, last 7 days and last 30 days.



## 4.9. Identity

In the identity section all external identifiers can be seen. Per default one external identifier is attached to the device with type "c8y\_Serial" and value of the device identifier configured in the application.properties.



## 5. Troubleshooting

### 5.1. SDKException: Http status code: 401

This error can appear if the provided “C8Y.baseUrl” configuration in the application.properties file is incorrect.

Additionally, it could also happen that the Cumulocity device credentials for the App are removed from the platform. If this is the case the “database.db” file on the PLCnext device in folder “/opt/plcnext/appshome/data/60002172000519” must be deleted, the App must be restarted and a new device registration must be done.

```

Caused by: com.cumulocity.sdk.client.SDKException: Http status code: 401
{error="security/Unauthorized",message="Invalid credentials!",info="https://www.cumulocity.com/guides/reference-guide/#a-name-error-reporting-a-error-reporting",details="null"}
at com.cumulocity.sdk.client.ResponseParser.checkStatus(ResponseParser.java:75) ~[java-client-1009.9.0.jar:na]
at com.cumulocity.sdk.client.ResponseParser.parse(ResponseParser.java:57) ~[java-client-1009.9.0.jar:na]
at com.cumulocity.sdk.client.RestConnector.get(RestConnector.java:124) ~[java-client-1009.9.0.jar:na]
at com.cumulocity.sdk.client.PlatformImpl.getPlatformApi(PlatformImpl.java:282) ~[java-client-1009.9.0.jar:na]
at com.cumulocity.sdk.client.PlatformImpl.getInventoryApi(PlatformImpl.java:295) ~[java-client-1009.9.0.jar:na]
at com.cumulocity.sdk.client.PlatformImpl$$FastClassBySpringCGLIB$$751b2209.invoke(<generated>) ~[java-client-1009.9.0.jar:na]
at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:218) ~[spring-core-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.invokeJoinpoint(CglibAopProxy.java:771) ~[spring-aop-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:163) ~[spring-aop-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:749) ~[spring-aop-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.aop.support.DelegatingIntroductionInterceptor.doProceed(DelegatingIntroductionInterceptor.java:136) ~[spring-aop-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.aop.support.DelegatingIntroductionInterceptor.invoke(DelegatingIntroductionInterceptor.java:124) ~[spring-aop-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:186) ~[spring-aop-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:749) ~[spring-aop-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercept(CglibAopProxy.java:691) ~[spring-aop-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at com.cumulocity.sdk.client.PlatformImpl$$EnhancerBySpringCGLIB$$5764dc09.getInventoryApi(<generated>) ~[java-client-1009.9.0.jar:na]
at com.cumulocity.microservice.api.CumulocityClientFeature.inventoryApi(CumulocityClientFeature.java:132) ~[microservice-platform-api-1009.9.0.jar:na]
at com.cumulocity.microservice.api.CumulocityClientFeature$$EnhancerBySpringCGLIB$$209f6920.CGLIB$inventoryApi$10(<generated>) ~[microservice-platform-api-1009.9.0.jar:na]
at com.cumulocity.microservice.api.CumulocityClientFeature$$EnhancerBySpringCGLIB$$209f6920.invoke(<generated>) ~[microservice-platform-api-1009.9.0.jar:na]
at org.springframework.cglib.proxy.MethodProxy.invokeSuper(MethodProxy.java:244) ~[spring-core-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at org.springframework.context.annotation.ConfigurationClassEnhancer$BeanMethodInterceptor.intercept(ConfigurationClassEnhancer.java:331) ~[spring-context-5.2.9.RELEASE.jar:5.2.9.RELEASE]
at com.cumulocity.microservice.api.CumulocityClientFeature$$EnhancerBySpringCGLIB$$209f6920.inventoryApi(<generated>) ~[microservice-platform-api-1009.9.0.jar:na]
at sun.reflect.GeneratedMethodAccessor52.invoke(Unknown Source) ~[na:na]
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[na:1.8.0_291]
at java.lang.reflect.Method.invoke(Method.java:498) ~[na:1.8.0_291]
at org.springframework.beans.factory.support.SimpleInstantiationStrategy.instantiate(SimpleInstantiationStrategy.java:154) ~[spring-beans-5.2.9.RELEASE.jar:5.2.9.RELEASE]
... 36 common frames omitted

```

### 5.2. UnknownHostException

This error indicates that the PLCnext device is not able to resolve the DNS name of the Cumulocity platform. Check the network configuration of the PLCnext device and try to ping the platform after re-configuration. In case of a cloud-based Cumulocity platform a proper internet connection must be available.

```

Caused by: javax.ws.rs.ProcessingException: java.net.UnknownHostException: env757303.eu-latest.cumulocity.com
    at org.glassfish.jersey.client.internal.HttpUrlConnector.apply(HttpUrlConnector.java:267) ~[jersey-client-2.33.jar!/:na]
    at org.glassfish.jersey.client.ClientRuntime.invoke(ClientRuntime.java:297) ~[jersey-client-2.33.jar!/:na]
    at org.glassfish.jersey.client.JerseyInvocation.lambda$invoke$0(JerseyInvocation.java:630) ~[jersey-client-2.33.jar!/:na]
    at org.glassfish.jersey.client.JerseyInvocation.call(JerseyInvocation.java:665) ~[jersey-client-2.33.jar!/:na]
    at org.glassfish.jersey.client.JerseyInvocation.lambda$runInScope$3(JerseyInvocation.java:659) ~[jersey-client-2.33.jar!/:na]
    at org.glassfish.jersey.internal.Errors.process(Errors.java:292) ~[jersey-common-2.33.jar!/:na]
    at org.glassfish.jersey.internal.Errors.process(Errors.java:274) ~[jersey-common-2.33.jar!/:na]
    at org.glassfish.jersey.internal.Errors.process(Errors.java:205) ~[jersey-common-2.33.jar!/:na]
    at org.glassfish.jersey.process.internal.RequestScope.runInScope(RequestScope.java:390) ~[jersey-common-2.33.jar!/:na]
    at org.glassfish.jersey.client.JerseyInvocation.runInScope(JerseyInvocation.java:659) ~[jersey-client-2.33.jar!/:na]
    at org.glassfish.jersey.client.JerseyInvocation.invoke(JerseyInvocation.java:629) ~[jersey-client-2.33.jar!/:na]
    at org.glassfish.jersey.client.JerseyInvocation$Builder.method(JerseyInvocation.java:408) ~[jersey-client-2.33.jar!/:na]
    at org.glassfish.jersey.client.JerseyInvocation$Builder.get(JerseyInvocation.java:308) ~[jersey-client-2.33.jar!/:na]
    at com.cumulocity.sdk.client.RestConnector.getClientResponse(RestConnector.java:150) ~[java-client-1009.9.0.jar!/:na]
    at com.cumulocity.sdk.client.RestConnector.get(RestConnector.java:123) ~[java-client-1009.9.0.jar!/:na]
    at com.cumulocity.sdk.client.PlatformImpl.getPlatformApi(PlatformImpl.java:282) ~[java-client-1009.9.0.jar!/:na]
    at com.cumulocity.sdk.client.PlatformImpl.getInventoryApi(PlatformImpl.java:205) ~[java-client-1009.9.0.jar!/:na]
    at com.cumulocity.sdk.client.PlatformImpl$FastClassBySpringGILB$5$12209.invoke(<generated>) ~[java-client-1009.9.0.jar!/:na]
    at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:218) ~[spring-core-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.invokeJoinpoint(CglibAopProxy.java:771) ~[spring-aop-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:163) ~[spring-aop-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:749) ~[spring-aop-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at org.springframework.aop.support.DelegatingIntroductionInterceptor.doProceed(DelegatingIntroductionInterceptor.java:136) ~[spring-aop-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at org.springframework.aop.support.DelegatingIntroductionInterceptor.invoke(DelegatingIntroductionInterceptor.java:124) ~[spring-aop-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:186) ~[spring-aop-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:749) ~[spring-aop-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercept(CglibAopProxy.java:691) ~[spring-aop-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]
    at com.cumulocity.sdk.client.PlatformImpl$EnhancerBySpringGILB$5$e0f30c8a.getInventoryApi(<generated>) ~[java-client-1009.9.0.jar!/:na]
    at com.cumulocity.microservice.api.CumulocityClientFeature.inventoryApi(CumulocityClientFeature.java:132) ~[microservice-platform-api-1009.9.0.jar!/:na]
    at com.cumulocity.microservice.api.CumulocityClientFeature$EnhancerBySpringGILB$5$2f5e8d.CGLIB$InventoryApi$3(<generated>) ~[microservice-platform-api-1009.9.0.jar!/:na]
    at com.cumulocity.microservice.api.CumulocityClientFeature$EnhancerBySpringGILB$5$47f5c4b.invoke(<generated>) ~[microservice-platform-api-1009.9.0.jar!/:na]
    at org.springframework.cglib.proxy.MethodProxy.invokeSuper(MethodProxy.java:244) ~[spring-core-5.2.9.RELEASE.jar!/:5.2.9.RELEASE]

```

### 5.3. SokPlcConnectionException: Error connecting to PLC

This problem could occur if the device RSC connection settings are wrong. Please re-check these config parameters:

- sok.phc.plc.username
- sok.phc.plc.password
- sok.phx.plc.ip
- sok.phx.plc.port

```

com.sokratel.phx.plc.connection.SokPlcConnectionException: Error connecting to PLC
    at com.sokratel.phx.plc.connection.PlcConnectionService.connect(PlcConnectionService.java:66) ~[phx-0.0.1-SNAPSHOT.jar!/:0.0.1-SNAPSHOT]
    at com.sokratel.phx.plc.connection.PlcConnectionManagerRunner.triggerConnect(PlcConnectionManagerRunner.java:55) ~[phx-0.0.1-SNAPSHOT.jar!/:0.0.1-SNAPSHOT]
    at com.sokratel.phx.plc.connection.PlcConnectionManagerRunner.run(PlcConnectionManagerRunner.java:36) ~[phx-0.0.1-SNAPSHOT.jar!/:0.0.1-SNAPSHOT]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) [na:1.8.0_312]
    at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308) [na:1.8.0_312]
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$301(ScheduledThreadPoolExecutor.java:180) [na:1.8.0_312]
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:294) [na:1.8.0_312]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_312]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_312]
    at java.lang.Thread.run(Thread.java:748) [na:1.8.0_312]
Caused by: com.phoenixcontact.adc.commonremoting.CommonRemotingException: Connection refused (Connection refused)
    at com.phoenixcontact.adc.commonremoting.impl.a.d.connect(SourceFile:218) ~[commonremoting-implementation-1.0.281.0.jar!/:1.0.281.0]
    at com.phoenixcontact.arp.system.rsc.ServiceManager.connect(SourceFile:268) ~[rsc-common-1.0.281.0.jar!/:1.0.281.0]
    at com.sokratel.phx.plc.connection.PlcConnectionService.connect(PlcConnectionService.java:57) ~[phx-0.0.1-SNAPSHOT.jar!/:0.0.1-SNAPSHOT]
    ... 9 common frames omitted
Caused by: java.net.ConnectException: Connection refused (Connection refused)
    at java.net.PlainSocketImpl.socketConnect(Native Method) ~[na:1.8.0_312]
    at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350) ~[na:1.8.0_312]
    at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206) ~[na:1.8.0_312]
    at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188) ~[na:1.8.0_312]
    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392) ~[na:1.8.0_312]
    at java.net.Socket.connect(Socket.java:607) ~[na:1.8.0_312]
    at sun.security.ssl.SSLSocketImpl.connect(SSLSocketImpl.java:288) ~[na:1.8.0_312]
    at com.phoenixcontact.arp.system.rsc.b.d.connect(SourceFile:117) ~[rsc-common-1.0.281.0.jar!/:1.0.281.0]
    at com.phoenixcontact.adc.internal.commonremoting.a.i.connect(SourceFile:94) ~[commonremoting-implementation-1.0.281.0.jar!/:1.0.281.0]
    at com.phoenixcontact.adc.internal.commonremoting.a.m.a(SourceFile:92) ~[commonremoting-implementation-1.0.281.0.jar!/:1.0.281.0]
    at com.phoenixcontact.adc.commonremoting.impl.a.d.connect(SourceFile:200) ~[commonremoting-implementation-1.0.281.0.jar!/:1.0.281.0]
    ... 11 common frames omitted

```

### 5.4. CommonRemotingFatalException: Communication error - no data from server

This problem can occur if the RSC encryption setting “sok.phx.plc.encrypted” is not matching the device configuration, so re-check this config parameter.

```
Caused by: com.phoenixcontact.ade.commonremoting.CommonRemotingFatalException: Communication error - no data from server.  
    at com.phoenixcontact.ade.internal.commonremoting.a.a.f.a(SourceFile:63) ~[commonremoting-implementation-1.0.281.0.jar!/1.0.281.0]  
    at com.phoenixcontact.ade.internal.commonremoting.a.a.a.a(SourceFile:1114) ~[commonremoting-implementation-1.0.281.0.jar!/1.0.281.0]  
    at com.phoenixcontact.ade.commonremoting.impl.a.d.b(SourceFile:544) ~[commonremoting-implementation-1.0.281.0.jar!/1.0.281.0]  
    at com.phoenixcontact.ade.commonremoting.impl.a.d.connect(SourceFile:153) ~[commonremoting-implementation-1.0.281.0.jar!/1.0.281.0]  
    at com.phoenixcontact.arp.system.rsc.ServiceManager.connect(SourceFile:181) ~[rsc-common-1.0.281.0.jar!/1.0.281.0]  
    at com.phoenixcontact.arp.system.rsc.ServiceManager.connect(SourceFile:218) ~[rsc-common-1.0.281.0.jar!/1.0.281.0]  
    at com.sokratel.phx.plc.connection.PlcConnectionService.connect(PlcConnectionService.java:61) ~[phx-0.0.1-SNAPSHOT.jar!/0.0.1-SNAPSHOT]  
    ... 9 common frames omitted  
Caused by: java.io.IOException: There is no more data because the end of the stream has been reached.  
    at com.phoenixcontact.ade.internal.commonremoting.a.e.a(SourceFile:1152) ~[commonremoting-implementation-1.0.281.0.jar!/1.0.281.0]  
    at com.phoenixcontact.ade.internal.commonremoting.a.a.c.a(SourceFile:68) ~[commonremoting-implementation-1.0.281.0.jar!/1.0.281.0]  
    at com.phoenixcontact.ade.internal.commonremoting.a.a.f.a(SourceFile:59) ~[commonremoting-implementation-1.0.281.0.jar!/1.0.281.0]  
    ... 15 common frames omitted
```