

OKdo Air Quality Kit for Raspberry Pi - Guide



Intro

Overview

Difficulty: 3/5, Time Needed: ~60 min

Got the new OKdo Air Quality Kit? Let's get started!

Make an air quality monitoring device with your Raspberry Pi and two sensors which will measure eCO₂ (*Carbon dioxide equivalent*), VOC (*Total Volatile Organic Compounds*), Temperature and Humidity. All of this data will be sent to OKdo Cloud and you'll be able to check it (current and historical) using your web browser or your smartphone. Let's go:

What you'll do

- Connect the hardware
- Create the device on OKdo Cloud
- Flash an OS to your Raspberry Pi
- Configure your Raspberry Pi
- Deploy the Air Quality project script
- Make the Air Quality script run automatically at boot
- Create a pretty interface to visualize your data on OKdo Cloud

Hardware needed

- [Raspberry Pi 3 Model B+](#) (or newer)
- [Grove Base HAT for Raspberry Pi](#)
- [Grove VOC and eCO₂ Sensor \(SGP30\)](#)
- [Grove Temperature & Humidity Sensor Pro](#) (also called DHT or DHT22)
- microSD Card (preferably Class 10)
- microSD Card Reader
- Monitor with HDMI cable
- USB Keyboard

Setup

OKdo Cloud

You'll need to create the device on [OKdo Cloud](#) so you can use your Raspberry Pi to send data from its sensors to it.

- Go to your Ground
- | • *You'll have a default "Playground" ground if you just created your account*
- Click "+ New Device"
- Choose "Raspberry Pi Air Quality Kit"
- Choose a name and click Connect

Assets **TVOC**, **eCO2**, **Temperature** and **Humidity** are already present and your Raspberry Pi will send sensor data to them.

Hardware

- Make sure your Raspberry Pi isn't plugged to power
- Connect the Grove Base HAT to your Raspberry Pi
- | • *Make sure to use the included spacers and screws - it'll be more stable*
- Using the included Grove connectors, connect the *SGP30* sensor to any one of three **I2C** ports on the Grove Base HAT
- Connect the *Temperature&Humidity Sensor Pro* to port **D22** on the Grove Base HAT
- | • *You can use any digital port, but be sure to change the pin in the code later*

Raspberry Pi

Flashing the OS

In this step, you'll flash the operating system for your Raspberry Pi so you can start working in it.

- Insert the SD card you'll use for the Raspberry Pi into your computer
- [Download the latest OS for Raspberry Pi](#)
- | • *- This is the "Lite" version of the OS, which means it doesn't have a graphical user interface - it only has command line, which is good since you're going to run the Raspberry Pi headless (without monitor/keyboard/mouse)*
- *If you already have a system installed, please make sure it's at least Raspbian Buster because older Raspbians (e.g. Stretch) will need additional steps of upgrading python and a few other things.*
- [Download Balena Etcher](#) which will be used to flash the OS to the SD card
- Open Etcher and select the downloaded image, select the SD card and click Flash
- Once done, insert SD card into your Raspberry Pi

- Plug in your monitor, keyboard and power cable into your Raspberry Pi
- *You'll see a lot of command line output on your monitor. Once you see "raspberrypi login:", you can continue with the steps below*

Initial System Configuration

Now you'll configure the Raspberry Pi so that you have a functional system ready to be used without a monitor or a keyboard.

- Enter `pi` (default user) and then enter `raspberrypi` (default password). You're now logged in to your Raspberry Pi
- Enter `sudo raspi-config` to enter the *Raspberry Pi software configuration tool*
- Go to *Network Options > WiFi > Choose your country > Enter your WiFi Network Name > Enter your WiFi Network Password*
- From the main screen, go to *Change User Password* and change your password
- *This is important - leaving the default password on an SSH-enabled system is a security risk*
- From the main screen, go to *Interfacing Options*. Enable **SSH**, **I2C** and **1-Wire**
- Click *Finish* and reboot your Raspberry Pi
- Once it boots, login with `pi` as user and your new password
- Execute `ifconfig` and make note of your IP Address (should be under network interface "wlan0" as "inet X.X.X.X")
- Make sure your computer is connected to the same network as your Raspberry Pi
- You can now disconnect your monitor and keyboard and use an SSH client such as [PuTTY](#) instead to connect to your Raspberry Pi and control it from your computer.
 - Open PuTTY
 - In the "Host name (or IP Address)", enter the IP you noted earlier
 - Click *Open* and accept the host key
- *In case you're using a Linux or macOS machine, open terminal and execute `ssh pi@your_ip_address` and you'll be prompted to enter the user and password*
 - Login as user "pi" with the password you defined earlier
 - Done!
- Your Raspberry Pi is now running "headless" (without monitor/keyboard/mouse) and you're ready for the rest of the guide!

Script Installation

This is the part where you'll actually read and send data from your sensors!

- Now you'll need to install libraries required to run the script. Run the following commands to do so:
 - `sudo apt-get install python-pip python3-pip`
- *Installs "pip", which is a Python package installer. Enter "y" when asked to confirm installation.*
 - `curl -sL https://github.com/Seeed-Studio/grove.py/raw/master/install.sh | sudo bash -s -`
- *Installs "[grove.py](#)", a dependency for sensor libraries*
 - `sudo pip3 install allthingstalk seeed-python-dht seeed-python-sgp30`
- *Installs the library for communicating with OKdo, Temperature & Humidity Sensor and the VOC and eCO2 (SGP30) Sensor*
- Once done, enter the following commands:
 - `sudo nano /usr/bin/okdo_airquality.py`
- You're now in a text editor and are creating a new file
 - Now copy and paste the code from below into that file (paste it in terminal):

https://drive.google.com/a/allthingstalk.eu/file/d/1qQMP4ljmDhAPDXr0spMdcplVsXWn2E_r/view?usp=drivesdk

- Edit the `DEVICE_ID` variable to contain your device ID
- *To get your Device ID, on [OKdo Cloud](#), go to your Device > Settings > Authentication > Device ID*
 - Edit the `DEVICE_TOKEN` variable to contain your device token
- *To get your Device ID, on [OKdo Cloud](#), go to your Device > Settings > Authentication > Device Tokens*
 - In case you didn't connect the Temperature/Humidity sensor according to this guide, change the `DHT_PIN` variable to your pin number.
 - While the editor is open, press CTRL+X to exit the editor, then press "y" to save the file and press enter when prompted for a name.
- You can now run the python script:
 - `python3 /usr/bin/okdo_airquality.py`
- *The sensor data is printed out in your terminal and sent to your OKdo Cloud every 30 seconds.*
 - To stop the script, press CTRL+C

Script Auto-Start

Now you'll make the system take care of the script, so it makes sure it:

- Runs at boot
- Runs only after an internet connection was established
- Is restarted the script if it stops working

You'll do that by adding the script as a service in the "systemd" directory:

- `sudo nano /etc/systemd/system/okdo_airquality.service`
- Copy/Paste the following (make sure the formatting stays the same):

```
[Unit]
Description=OKdo Air Quality Project
After=network.target
```

```
[Service]
ExecStart=/usr/bin/okdo_airquality.py
Restart=always
RestartSec=5
```

```
[Install]
WantedBy=multi-user.target
```

- Press CTRL+X, Confirm with "y" and confirm the name.
- `sudo chmod +x /usr/bin/okdo_airquality.py` - Adds permissions
- `sudo systemctl daemon-reload` - Reloads services
- `sudo systemctl start okdo_airquality` - Starts our script
- `sudo systemctl enable okdo_airquality` - Enables our script to start at boot
- `sudo systemctl status okdo_airquality` - Check the status.
- Make sure you see `Active: active (running)`
- *And use `sudo systemctl stop okdo_airquality` if you want to stop the service, or `sudo systemctl disable okdo_airquality` if you wish to stop it from being executed at boot.*
- Check that your sensor data is being uploaded by going to your [OKdo Cloud](#).
- Now you can restart your Raspberry Pi, and it will automatically start uploading sensor data upon boot!

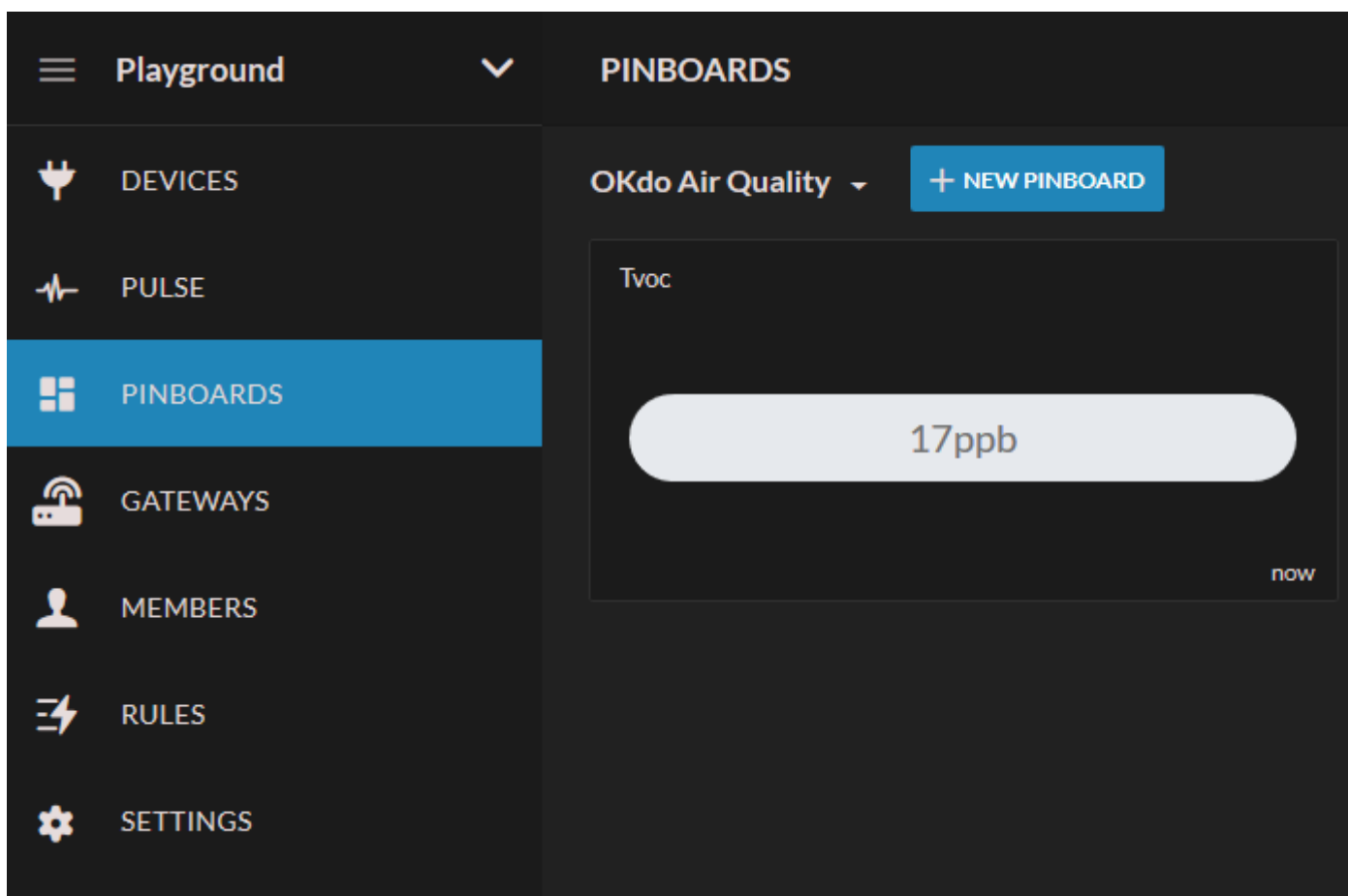
Final

You now have a fully functional air quality measuring device!
Let's do the final stuff to make it look nice.

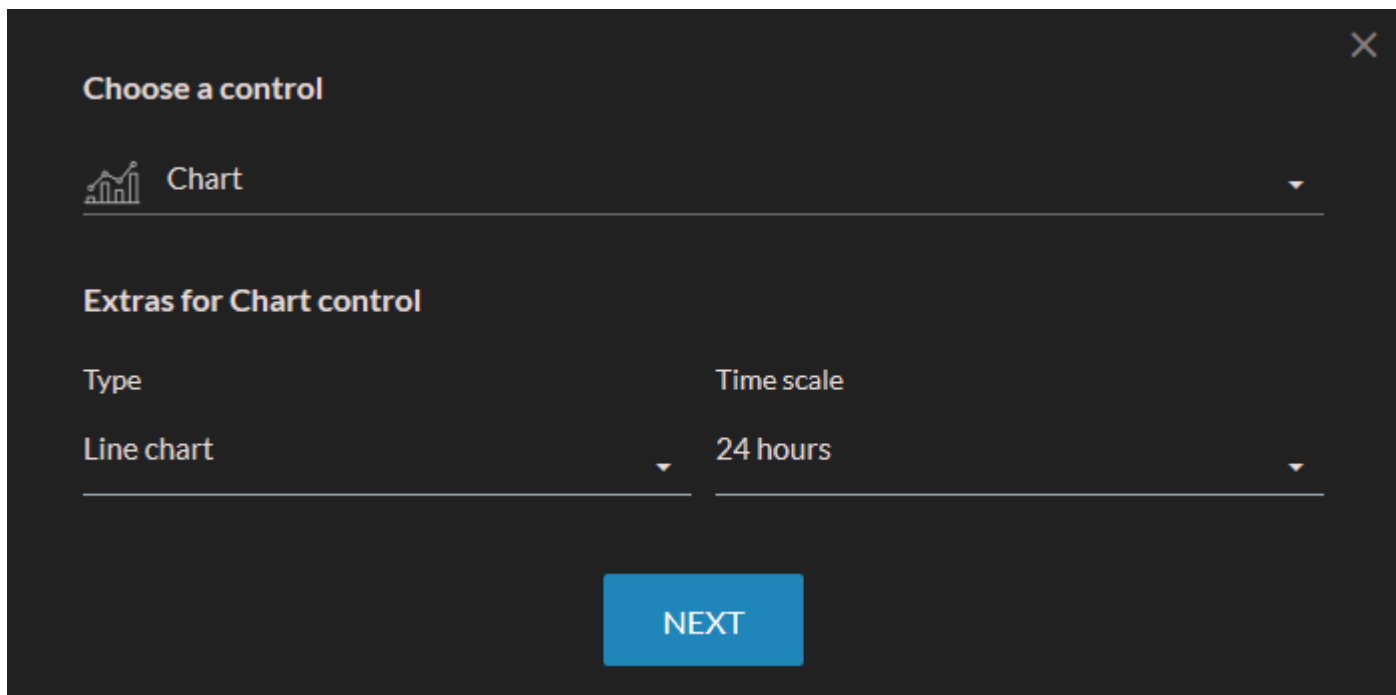
Create a Pinboard

Make the data more presentable by creating a Pinboard to show it all:

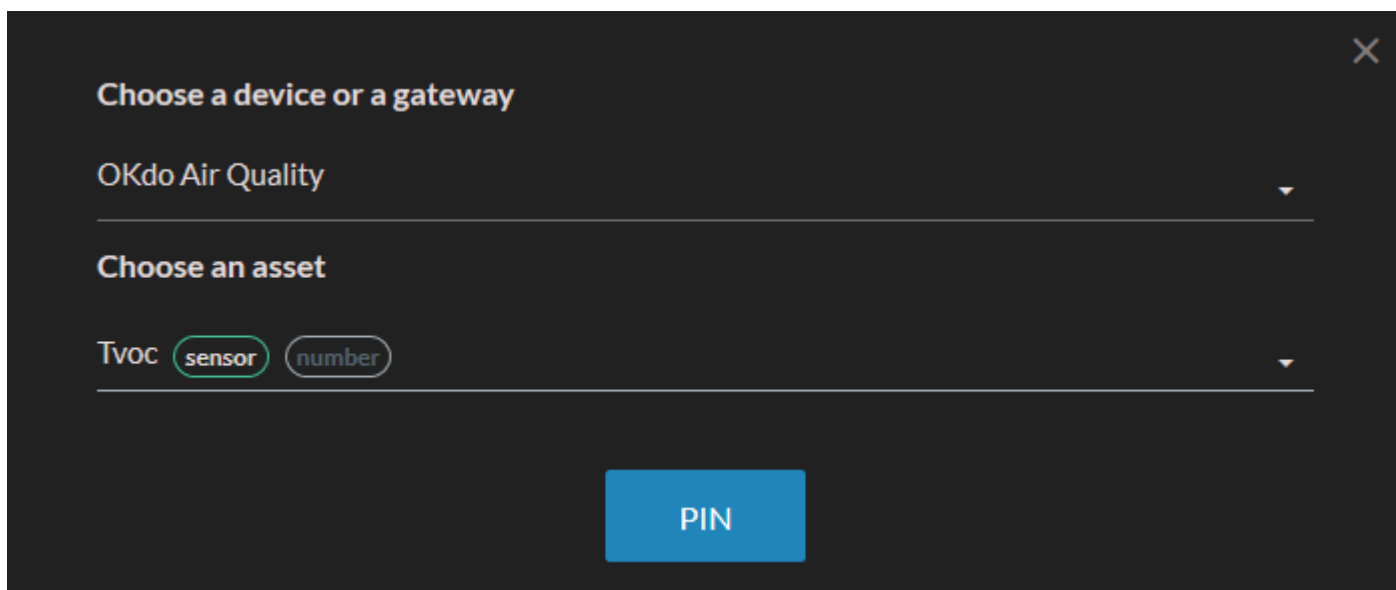
- Go to your [OKdo Cloud](#)
- Select your Ground
- Go to your *Devices* and click on the air quality device you created
- In the upper right corner, click “+ *Create Pinboard*”
- Select all 4 assets you have and click “*Create Pinboard*”
- You should now see the data visualized:



- Now let's create historical charts: Edit your Pinboard (pencil icon)
- Click “New Pin” and set *control* to “*Chart*”, *type* to “*Line chart*” and *Time scale* to “*24 hours*”. You can of course play with these settings whenever you want.



- Now select your device and select the asset you wish to show on the Pinboard.



- Repeat this step the remaining 3 assets you have and you should see charts showing data from all of your sensors!

You can also rearrange and change control's size, so you can get something like this:



PINBOARDS



OKdo Air Quality ▾

+ NEW PINBOARD

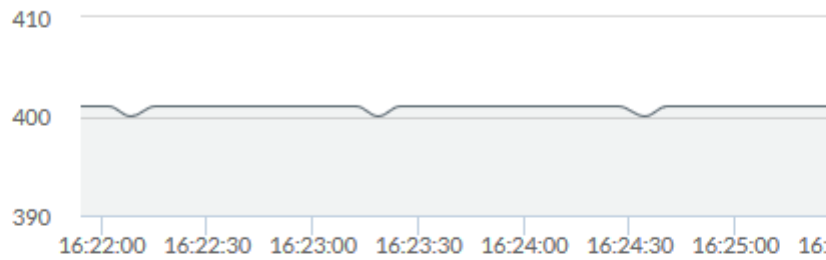


Eco2

401ppm

now

Eco2

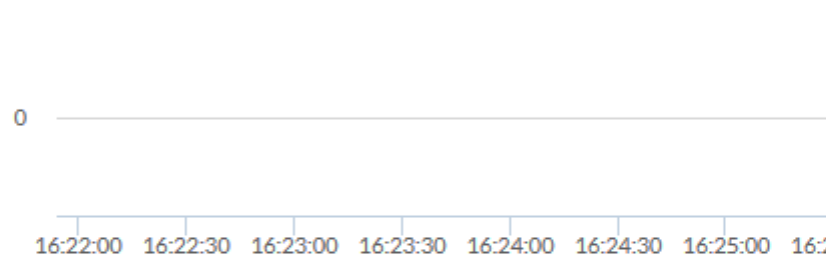


Tvoc

0ppb

now

Tvoc

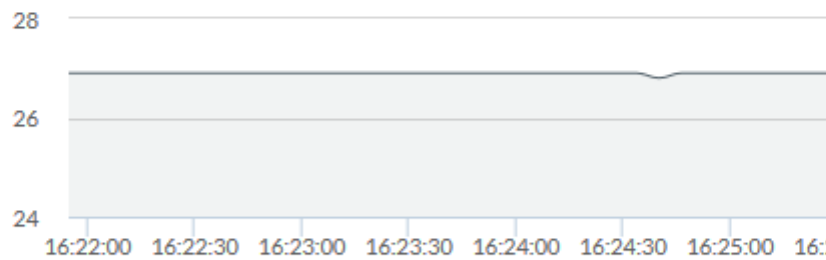


Temperature

26.9C

now

Temperature

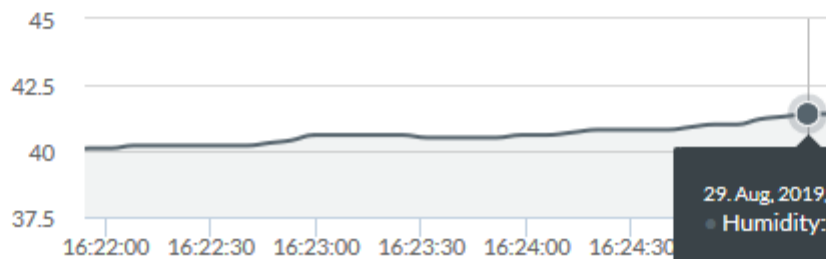


Humidity

41.6%

now

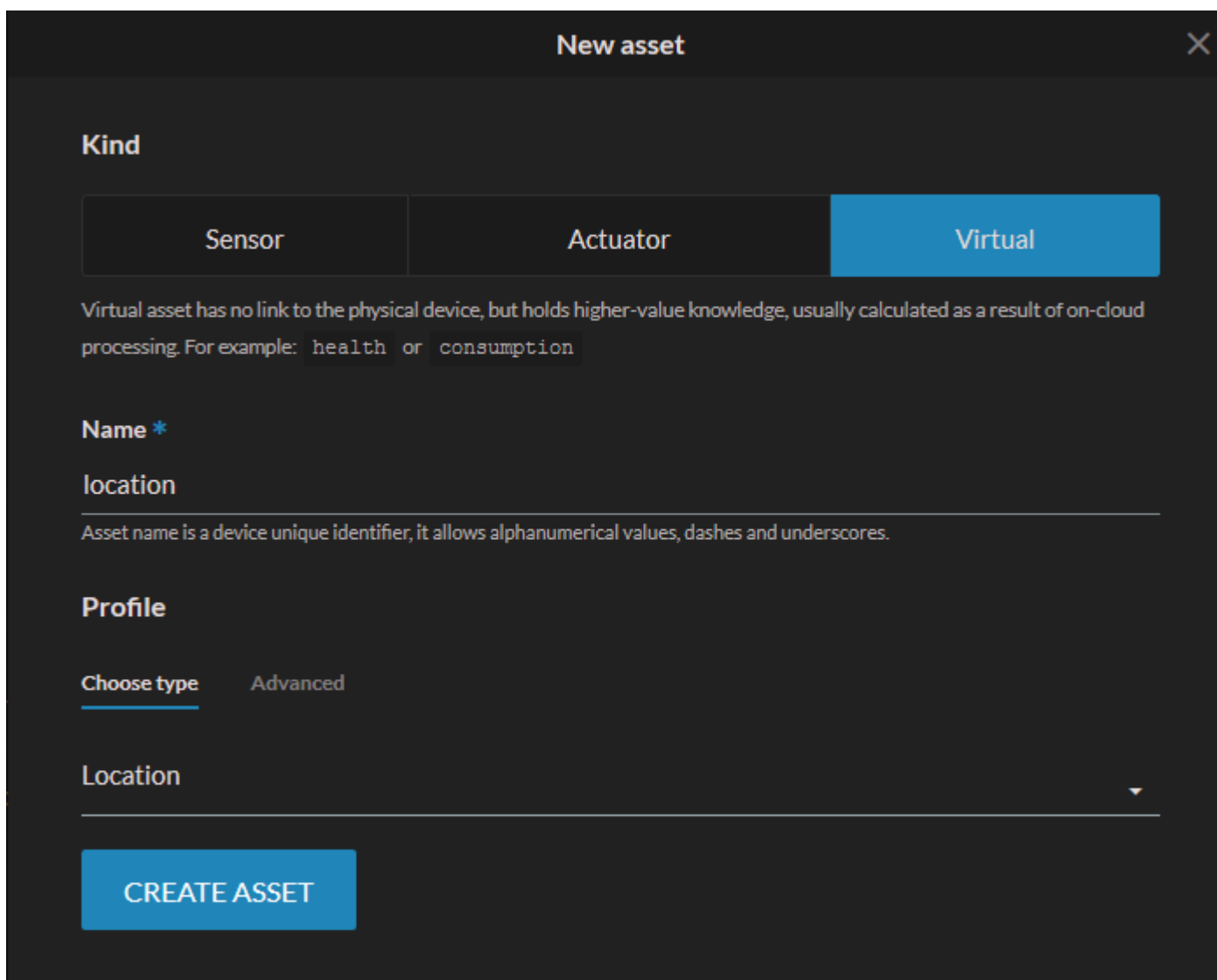
Humidity



Share Your Data

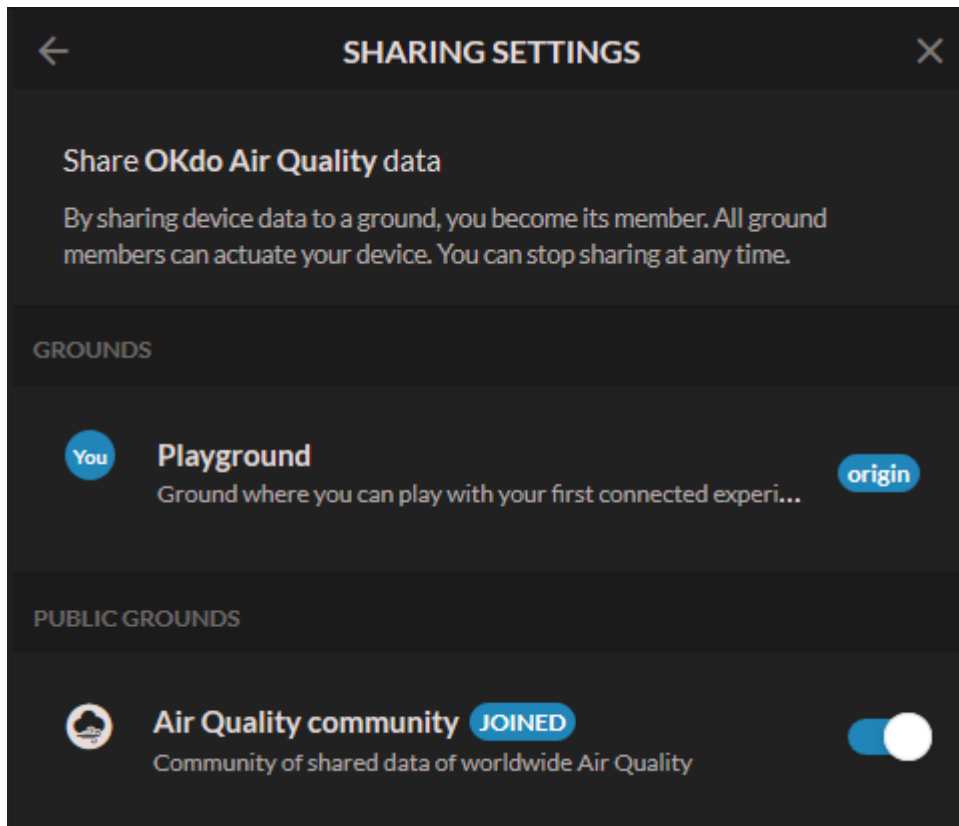
Help others be aware of air quality in your area by sharing your data with the OKdo Air Quality Community:

- In the OKdo Cloud select your newly configured device.
- Click “+ New Asset” and fill it out as shown:



The screenshot shows a dark-themed 'New asset' form. At the top, there's a title 'New asset' and a close button (X). Below the title, there's a section labeled 'Kind' with three buttons: 'Sensor', 'Actuator', and 'Virtual'. The 'Virtual' button is highlighted in blue. Below this, there's a text description: 'Virtual asset has no link to the physical device, but holds higher-value knowledge, usually calculated as a result of on-cloud processing. For example: health or consumption'. The next section is 'Name *' with a text input field containing 'location'. Below the input field, there's a note: 'Asset name is a device unique identifier, it allows alphanumerical values, dashes and underscores.'. The 'Profile' section has two tabs: 'Choose type' (selected) and 'Advanced'. Below the tabs, there's a 'Location' dropdown menu. At the bottom, there's a large blue button labeled 'CREATE ASSET'.

- Set your Location by clicking the *location* asset, and click `Use my location`, or choose your location on a map (does not have to be precise if you're not comfortable with it).
- Now simply go back to your device screen, click “Settings” > “Share data” and activate the toggle to share data with the *Air Quality community*.



Enjoy your new Air Quality device!