

# MicroX on premise Troubleshooting

MicroX: <https://www.snap4city.org/738>

To get updated version of the slides go to <https://www.snap4city.org/577>

From Snap4City:

- We suggest you read the TECHNICAL OVERVIEW:
  - <https://www.snap4city.org/download/video/Snap4City-PlatformOverview.pdf>
- Development: <https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf>
- See Client-Side Business Logic Widget Manual:  
<https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf>
- <https://www.snap4city.org>
- <https://www.snap4solutions.org>
- <https://www.snap4industry.org>
- <https://twitter.com/snap4city>
- <https://www.facebook.com/snap4city>
- <https://www.youtube.com/channel/UC3tAO09EbNba8f2-u4vandg>

Coordinator: Paolo Nesi, [Paolo.nesi@unifi.it](mailto:Paolo.nesi@unifi.it)

DISIT Lab, <https://www.disit.org>

DINFO dept of University of Florence,

Via S. Marta 3, 50139, Florence, Italy

Phone: +39-335-5668674

Access Level: public

Date: 25-09-2023

Version: 1.0



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

## Summary

Install docker and docker-compose .....	4
MicroX creation and on premise setup .....	4
First steps.....	4
Deploy to customer domain .....	5
Enable HTTPS.....	5
Error in keycloak .....	8
IoT App won't load.....	8
Orion/Nifi subscription lost .....	9
Widget/wssserver errors.....	9
Changing user passwords .....	10
Close all doors.....	10
Redirect base URL to URL/dashboardSmartCity .....	11
Adding a nodered app .....	11
Possible error following addition of IoTApp.....	12
Change ownership of IoTApp .....	12
Change display of menu items .....	12
Change portal name (i.e. Snap4Asymmetric).....	12
Edit footer and logo Dashboard .....	12
Change row/col limit of widgets.....	13
KIBANA Dashboard .....	13
Set SMTP configuration .....	13
Clone models .....	13
Clone static attributes .....	14
Mapping local folder with docker-compose to a volume inside a container.....	14
Possible errors and solutions.....	16
All quiet... docker-compose ps all up but then log in from browser and white screen with no errors on console.....	17
Orion's delegation to the group .....	17
No microx IOT app is accessible from the submenu .....	17
Orion is not receiving data from nodeRED (1).....	18
Orion is not receiving data from nodeRED (2).....	18
Orion stalls (following a system crash).....	18
change root psw and nifi should be updated.....	19
you change root psw and the device is not there in the data-inspector .....	20
Data crashes on NiFi in enrich-data.....	20

NiFi does not save on opensearch/elasticserach ..... 21

The device is not in the service map ..... 21

Device does not appear in data-inspector ..... 21

Change session timeout ..... 22

Cleaning up the tokens of an iot app..... 23

Set up tunnelling for access to NIFI interface..... 23

Broker subscription list..... 24

Enabling a user to CSBL ..... 24

Increasing the default refresh time of Micro X ..... 24

Update dictionary after adding new value type..... 24

If api iot-search doesn't work..... 24

Empty authentication on Node-RED stream ..... 25

Clean up the db from the sample data in the MicroXs. .... 26

IOTApp Flow ..... 26

## Install docker and docker-compose

- Docker: <https://docs.docker.com/engine/install/debian/>
- Docker-compose: <https://docs.docker.com/compose/install/>

## MicroX creation and on premise setup

From the page [https://www.snap4city.org/docker-generator/selecting\\_model](https://www.snap4city.org/docker-generator/selecting_model) create your required version:

- IP must be the IP of the machine (VM) running Docker: on Unix verify IP with `ifconfig`
- Then download the installation files, unzip and follow the guide

**Installation** (<https://www.snap4city.org/drupal/node/738>):

- unzip the archive
- navigate to the archive, then run with `sudo` (or better `su -`)
- then run:
  - `./setup.sh`
  - `docker-compose up -d`
  - `./post-setup.sh`

At <http://dashboard/dashboardSmartCity> we find the homepage. Note: The IP of the VM must be added to the `/etc/hosts` file (or `C:\Windows\System32\drivers\etc\hosts` for Windows) of the client.

*In `post-setup.sh`, line #4, change to `./update-ontology.sh localhost`*

## First steps

- log in as `rootadmin`
- **[IMPORTANT]** take ownership of the IoT broker by clicking `edit` and confirm `<<<<`
- delegate the broker to the group
- log in as `areamanager`
- Define a device model by indicating the delegated broker,
  - set `dateObserved` (timestamp, timestamp in ms, string) to allow reading as time series
  - Council (en) also put a data like I put `lateemperatures` at (temperatures, Celsius, float)
- Build an IoT device, using the model defined
- from `nodered`, create a flow to send data to the device, setting up the connection with the broker, the keys, and defining the necessary fields in the message
  - **[IMPORTANT]** `dateObserved` must have.
    - `type = string,`
    - `value = new Date().toISOString()`

(en) -> to send The data I recommend to go to `IoTDevices` and press the `+` key and then take the `ngsi v2` payload click it and it opens the content to put in the `nodered` function and then send the data to the broker

Ps I put the json of the stream at the bottom of this doc

!!!! remember to put the subnature

- Create a dashboard to read data from the IoT device
- possibility of creating a KPI as well

- **orion broker** sometimes remains in a state of continuous restart.
  - manually stop, remove and restart the orion container
    - `docker-compose stop orion`
    - `docker-compose rm orion`
    - `docker-compose up -d orion`

*Bringing orion back into proper execution.*  
*In order to get to the correct creation of device model and iot device (the broker `iotobsf` created did not appear in the device creation screens) it was necessary to restart the individual containers.*
- Sometimes the dashboard creation method did not allow to select the created iot devices. The problem may be related to the **dashboard-cron** container: the control scripts for searching for new devices were stuck due to the file `running2.txt` being incorrectly removed. By manually removing the file the scripts ran and the devices found.

## Deploy to customer domain

Operations performed for Asymmetric.

The first step is to verify the domain name associated with the IP of the server on which to run the installation. If there is an error (wrong domain, e.g., `e.g.,.com` instead of `www.esempio.com`) you will experience problems accessing the platform.

*Homepage link:* <http://<domain-name>/dashboardSmartCity>

*When creating MicroX, it is useful to change all service passwords.*

*Note that as you change the password for rootadmin, it is changed in the **dashboardbuilder** configuration that it uses to take private sensor data but not in LDAP: so access to the portal is always done with the default password, but since it is not the same as the password used for dashboardbuilder, this can cause communication errors on the sensors/devices.*

*Solutions:*

- *change password for rootadmin in LDAP (see later)*
- *reset the default password for rootadmin in `dashboard-builder-con/personaldata.ini` and restart **dashboard-cron** (and possibly **proxy** to handle the new internal container IP)*

## Enable HTTPS

In the MicroX generator, select `https` in the **##base-protocol##** field.

Then follow the guide to install certbot/letsencrypt and enable https on nginx proxy

- [Nginx and Let's Encrypt with Docker in Less Than 5 Minutes | by Philipp | Medium](#)
- [How to handle HTTPS using Nginx, Let's encrypt and Docker - Mindsers Blog](#)

*Broadly speaking, to enable https, in addition to adding certbot to the compose, you need to edit `nginx.conf`. The less intuitive part is the need to create dummy certificates to allow nginx to start, then remove them and make the request to `letsencrypt` to get the real certificates.*

In the `docker-compose.yml` add certbot and indicate shared volumes between certbot and nginx to store certificates

```

proxy:
  image: nginx
  logging:
    driver: json-file
    options:
      max-file: '10'
      max-size: 100m
  networks:
    default:
      aliases:
        - www.cityconn.cloud
  ports:
    - "80:80"
    - "443:443"
  restart: unless-stopped
  command: "/bin/sh -c 'while ;; do sleep 6h & wait $$(!); nginx -s reload; done & nginx -g \"daemon off;\""
  volumes:
    - ./nginx-proxy-conf:/etc/nginx/conf.d:rw
    - ./certbot/conf:/etc/nginx/ssl:ro
    - ./certbot/www:/var/www/certbot:ro
certbot:
  image: certbot/certbot
  entrypoint: "/bin/sh -c 'trap exit TERM; while ;; do certbot renew; sleep 12h & wait $$(!); done;'"
  volumes:
    - ./certbot/conf:/etc/letsencrypt:rw
    - ./certbot/www:/var/www/certbot:rw

```

In *nginx.conf* add the configuration for https (port 443) and edit the one for http (port 80)

```

server {
    listen 80;
    server_name www.cityconn.cloud;

    location /.well-known/acme-challenge/ {
        root /var/www/certbot;
    }
    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name www.cityconn.cloud;

    ssl_certificate /etc/nginx/ssl/live/www.cityconn.cloud/fullchain.pem;
    ssl_certificate_key /etc/nginx/ssl/live/www.cityconn.cloud/privkey.pem;

    root /var/www/html;

    index index.html index.htm index.nginx-debian.html;

    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_http_version 1.1;
    proxy_read_timeout 10m;

    location = / {
        rewrite ^ https://www.cityconn.cloud/dashboardSmartCity/ redirect;
    }
    location / {
        proxy_pass "http://dashboard-builder/";
    }
    location /phpldapadmin/ {
        proxy_pass "https://myldap:443/";
    }
    location /ServiceMap/api/v1/iot/ {
        proxy_pass "http://servicemap:8080/iot/";
    }
    location /ServiceMap/ {
        proxy_pass "http://servicemap:8080/ServiceMap/";
    }
    location /wsserver {
        proxy_set_header Upgrade $http_upgrade;

```

```

        proxy_set_header Connection "upgrade";

        proxy_pass "http://wssserver/wssserver";
    }
    location /superservicemap/ {
        proxy_pass "http://servicemap:8080/superservicemap/rest/";
    }
    location /auth/ {
        proxy_pass "http://keycloak:8088/auth/";
    }
    location /datamanager/ {
        proxy_pass "http://personaldata:8080/datamanager/";
    }
    location /kibana/ {
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_http_version 1.1;
        proxy_pass "http://opensearch-dashboards:5601/";
    }
    location /synoptics/ {
        proxy_pass "http://synoptics:3002/";
    }
    location /synoptics/socket.io/ {
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_pass "https://synoptics:3001/socket.io/";
    }

    location /iotapp/iotapp-001/ {
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";

        proxy_pass "http://iotapp-001:1880/iotapp/iotapp-001/";
    }
    location /iotapp/iotapp-002/ {
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";

        proxy_pass "http://iotapp-002:1880/iotapp/iotapp-002/";
    }
}

```

*Note: proxy\_passes all remain unchanged in http/https*

So

- create the dummy certificates

```

docker-compose run --rm --entrypoint "\
openssl req -x509 -nodes -newkey rsa:4096 -days 1\
-keyout '/etc/letsencrypt/live/www.cityconn.cloud/privkey.pem' \
-out '/etc/letsencrypt/live/www.cityconn.cloud/fullchain.pem' \
-subj '/CN=localhost'" certbot

```

- start the compose (especially the proxy)

```
docker-compose restart proxy
```

- delete fictitious certificates

```

docker-compose run --rm --entrypoint "\
rm -Rf /etc/letsencrypt/live/www.cityconn.cloud && \

```

```
rm -Rf /etc/letsencrypt/archive/www.cityconn.cloud && \
rm -Rf /etc/letsencrypt/renewal/www.cityconn.cloud.conf" certbot
```

- Make the application for royal certificates

```
docker-compose run --rm certbot certonly --webroot --webroot-path
/var/www/certbot/ -d www.cityconn.cloud
```

Error in keycloak

*Gets error: Invalid parameter: redirect\_uri*

Add to compose for keycloak in environment:

- KEYCLOAK\_FRONTEND\_URL: https://<domain\_name>/auth/.
- PROXY\_ADDRESS\_FORWARDING: "true"

As in the example below

```
keycloak:
  command:
    -Djboss.socket.binding.port-offset=8
  depends_on:
    ldap-server:
      condition: service_started
  environment:
    KEYCLOAK_PASSWORD: sRqdREu20xBTJzdN
    KEYCLOAK_USER: admin.
    KEYCLOAK_FRONTEND_URL: https://www.cityconn.cloud/auth/
    PROXY_ADDRESS_FORWARDING: "true"
  image: disitlab/preconf-keycloak:v4
  logging:
    driver: json-file
    options:
      max-file: '10'
      max-size: 100m
  # ports:
  # - published: 8088
  # target: 8088
  restart: unless-stopped
  volumes:
    - keycloak:/opt/jboss/keycloak/standalone:rw
```

Then recreate the container with `docker-compose up -d keycloak` and possibly restart the proxy.

IoT App does not load

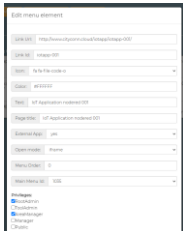
It is a link problem in the DB in the menu table where for IoT Apps http is specified and not https.



To correct, either log into the DB and edit the entry, or from the administration interface identify the menu entry

10230	IoT Application nodered2 001	http://www.cityconn.cloud/iotapp/iotapp-001/	IoT Application nodered2 001	EDIT DELETE	iotapp-001	MainMenuSubmenus	1035
-------	------------------------------	--	------------------------------	----------------	------------	------------------	------

so with EDIT you change http to https



### Edit menu element

Link Url:

### Orion/Nifi subscription lost

If on orion the data comes in, but the device doesn't show up in data inspector, it should be seen subscription to nifi by trying

```
curl http://localhost:1026/v2/subscriptions
```

In the case of an empty array in response, the broker's registration on nifi has been lost. To restore it

- login as rootadmin
- go to broker's edit
- edit the subscription url slightly and then remove it by putting it back as in the original

Running curl again should result in an array like

```
[{"id": "62d5112d75ae1864255a642d", "description": "orion-1 nifi", "status": "active", "subject": {"entities": [{"idPattern": ".*", "typePattern": ".*"}], "condition": {"attrs": []}}, "notification": {"attrs": []}, "onlyChangedAttrs": false, "attrsFormat": "normalized", "http": {"url": "http://nifi:1030/ingestngsi"}}
```

### Widget/wsserver errors

If in HTTPS, edit the `dashboard-builder-conf/webSocketServer.ini` file by changing the options

- `leads` on 443 and
- `protocol` on wss

In `iotapp-00X/settings.js`, edit

```
wsServerUrl: 'ws://www.cityconn.cloud/wsserver'
```

at

```
wsServerUrl: 'wss://www.cityconn.cloud/wsserver'
```

In *nginx.conf* edit

```
location /wsserver {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    proxy_pass "http://wsserver/wsserver";
}
```

In

```
location /wsserver {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    proxy_pass "http://wsserver:443/wsserver";
}
```

### Change user password

Changing user passwords is done from **ldap**, using **phpldapadmin** reachable at the URL

<https://www.cityconn.cloud:6443/phpldapadmin>

Access to phpldapadmin may be blocked due to invalid https certificate: simply accept the invalid certificate and browser access should work.

In addition to the users `userrootadmin`, `userareamanager`, etc., it is also possible to change the password of the keycloak admin. calmly we explain further

change psw admin keycloak go to `addressinstallation/auth/admin`

for user password change go to the snap4 front-end

left menu

user management and auditing

Dashboard builder local users

**[NIC NOTE: changing the root psw causes inconsistencies that need to be documented]**

### Close all doors

By default, all containers in the compose have a port mapping that maps internal Docker network ports to externally accessible ports. Although useful during deployment, once in production all ports must be closed for security changes. The only container that must be externally accessible is the proxy for obvious reasons.

To close the ports, simply comment out/remove the port mappings from the compose file and restore the containers with `docker-compose up -d`.

Following the port closure, the site may be unreachable. In that case, try restarting dashboard-builder first, then the proxy to update the internal IPs associated with the recreated containers.

## Redirect base URL to URL/dashboardSmartCity

In nginx.conf add

```
location = / {
    rewrite ^ https://www.cityconn.cloud/dashboardSmartCity/ redirect;
}
```

## Adding a nodered app

In this configuration to add a new nodered app `nr3-003` you need to:

1. change the `docker-compose.yml` file and copy the rows defining `nodered-nr2-iotapp-001` and change any reference to `nr2-001` into `nr3-002`
2. copy the folder `iotapp-001nr2` into `iotapp-002nr3` and check the Unix permission to files and directories, and files and directories ownership
3. edit the file `iotapp-002nr3/settings.js` and replace any reference to `001nr2` to `002nr3`

```
- line 19: var myappid='iotapp-002'
- line 131: httpRoot: '/iotapp/iotapp-002',
```
4. edit the file `apache-proxy-conf/nginx-proxy-conf/nginx.conf` and copy the rows related with `001nr2` to `002nr3`
5. connect to the db on port 3306 user/passwordx and
6. add a new row in `profiledb.ownership` with the following SQL command

```
INSERT INTO
profiledb.ownership(username,elementid,elementtype,elementname,elementurl,
elementdetails,created) values('usermanager','nr3','AppID','nodered
3','http://dashboard/iotapp/nr3/', '{"edgeway_type":
"linux_Linux_4.9.0-8-amd64"}',now());
```

```
INSERT INTO
profiledb.ownership(username,elementId,elementType,elementName,elementUrl,
elementDetails,created) values('userareamanager','iotapp-
002','AppID','nodered
2','http://www.cityconn.cloud/iotapp/002/', '{"edgeway_type':'linux_Lin
ux_4.9.0-8-amd64'}',now());
```

7. perform the following SQL command to add a menu option to easily access the app:

```
INSERT INTO
Dashboard.MainMenuSubmenus(menu,linkUrl,linkid,icon,text,privileges,userTy
pe,externalApp,openMode,iconColor,pageTitle,menuorder,organizations)
VALUES (1035,'http://dashboard/iotapp/nr3/', 'iotappnr3', 'fa fa-file-code-
o', 'IoT Application nodered3', ['\RootAdmin\',
'\Manager\'],'any','yes','iframe','#FFFFFF','IoT Application nodered3',
3, ['\Organization\','\DISIT\','\Other\']');
```

```
INSERT INTO
Dashboard.MainMenuSubmenus(menu,linkUrl,linkid,icon,text,privileges,userType,externalApp,openMode,iconColor,pageTitle,menuorder,organizations)
VALUES (1035,'https://www.cityconn.cloud/iotapp/iotapp-002/', 'iotapp-002', 'fa fa-file-code-o', 'IoT Application nodered 002', '[\'RootAdmin\', \'AreaManager\']', 'any', 'yes', 'iframe', '#FFFFFF', 'IoT Application nodered 002', 1, '[\'Organization\', \'DISIT\', \'Other\']');
```

8. issue command `docker-compose up -d` to bring the new container up
9. restart the dashboard container and the proxy:
10. `docker-compose restart dashboard-builder`
11. `docker-compose restart proxy`

If all it is ok, you should see from the menu of `'usermanager'` `'userareamanager'` the new application and you can connect to it.

**Note:** if you delete the `dashboarddb` volume you will lose the changes made on the db, to make them more permanent you may change the `database/profiledb.sql` file and `database/dashboard-menu.sql` adding the two SQL instructions.

**Note:** Please consider that login on a nodered app is allowed to the owner (as stated on the ownership table) and to any user with RootAdmin role.

Possible error following addition of IoTApp.

Both `iotapp 001` and the new `002` fail to write to orion reporting the following error:

```
failed to update, reason: {"statusCode":401, "headers":{"x-content-type-options": "nosniff", "x-xss-protection": "1; mode=block", "cache-control": "no-cache, no-store, max-age=0, must-revalidate", "pragma": "no-cache", "expires": "0", "strict-transport-security":"max-age=31536000 ; includeSubDomains", "x-frame-options": "DENY", "transfer-encoding": "chunked", "date": "Fri, 22 Jul 2022 08:42:35 GMT", "connection": "close"}, "payload":{"result":false, "message":"The passed access token is not valid"}
```

**Solution:** restart `orionbrokerfilter`.

**Note:** Restarting the whole compose may not work, as if `orionbrokerfilter` starts before `keycloak` is ready it cannot retrieve keys for ownerships.

Change ownership of IoTApp

Need to change the ownership in the `profiledb.ownership` db for the `iotapp` of interest, then re-access the IoTApp with the new owner to do the `acesstoken` refresh

Change display of menu items

You can change which menu items to display for which users by going to change the values in the `privileges` column in the `Dashboard.MainMenu` and `Dashboard.MainMenuSubmenus` tables.

Change portal name (i.e. `Snap4Asymmetric`)

In the DB edit in the `Dashboard.Domains` table the column `'claim'`

Edit footer and logo Dashboard

In `dashboard-builder`, in the folder `/var/www/html/dashboardSmartCity/view` edit the file `index.php`. To ensure persistence in case of container rebuild:

- copy folder from container to host  
`docker cp <container_ID>:/var/www/html/dashboardSmartCity/view/* ./dashboard-builder-view/.`
- Add in compose the mount of the copied folder  
`- ./dashboard-builder-view:/var/www/html/dashboardSmartCity/view:rw`
- Do up the container and then edit the

## Change row/col limit of widgets

Change `max_row` and `max_col` in the `Dashboard.Widgets` table.

## KIBANA Dashboard

In `postsetup.sh` edit

```
curl -u admin:f89Ux63JkcyN9ofh -XPOST
"http://localhost/kibana/api/saved_objects/_import?overwrite=true" -H "osd-xsrf:
true" -H "securitytenant: global" --form file=@osd-dashboard.ndjson
```

at

```
curl -u admin:f89Ux63JkcyN9ofh -XPOST
"https://www.cityconn.cloud/kibana/api/saved_objects/_import?overwrite=true" -H
"osd-xsrf: true" -H "securitytenant: global" --form file=@osd-dashboard.ndjson
```

## Set SMTP configuration

In `servicemap-conf/servicemap.properties` set the following fields,

```
#smtp settings CONFIGURE
mailfrom=me@email.com
smtp=$#smtp-host#$
portSmt=$#smtp-port#$
authSmt=$#smtp-auth#$
#authTypeSmt=TLS
#userSmt=
#passwdSmt=
```

**e.g.:**

```
#smtp settings CONFIGURE
mailfrom=dashboard@xxxxxcloud
smtp=smtps.aruba.it
portSmt=465
authSmt=$#smtp-auth#$
authTypeSmt=TLS
userSmt=<MANCA>
passwdSmt=xxxx@2022
```

Then restart **servicemap**

## Clone models

- Models in the `iotdb.model` table of the `dashboarddb`
- Ownership in the `profiledb.ownership` table of the `dashboarddb`

## Clone static attributes

- Clone the tables `processloader_db.dictionary_table` and `processloader_db.dictionary_relations`
- Update VIRTUOUS as follows

Importing the dictionary requires re-synchronization of `servicemap`, using `update-valuetypes.sh` in `servicemap-conf`, which should contain

```
cd "$(dirname "$0")"
echo -n "$(date -Iseconds) get valuetypes from processloader... "
curl -sS http://localhost:90/processloader/get_valuetypes.php > valuetypes.vt

STATUS="$(cmp --silent valuetypes.vt valuetypes-last.vt ; echo $?)" # "$?" gives exit status for
each comparison

if [ "$STATUS" -ne "0" ]; then
    echo CHANGED valuetypes upload to virtuoso

    /usr/local/bin/docker-compose exec -T virtuoso-kb isql-v localhost dba
/root/servicemap/valuetypes.vt

    cp valuetypes.vt valuetypes-last.vt
else
    echo SAME valuetypes
fi
```

Before running the script, check the URL response: for the installation on Asymmetric, the correct URL was [https://www.cityconn.cloud/processloader/get\\_valuetypes.php](https://www.cityconn.cloud/processloader/get_valuetypes.php)

It is then necessary to change the virtuoso password (e.g., `dba`) if it was changed during the creation of the microX.

Finally, to force the upload, and force the KB to update even if the script finds no changes, you need to delete the `valuetypes-last.vt` file

## Mapping local folder with docker-compose to a volume inside a container

in the docker compose for example in the dashboard-builder section

between volumes add

```
- developmentfiles:/var/www/html:rw
```

then at the bottom of docker-compose add this

```
developmentfiles:
```

```
  driver: local
```

```
  driver_opts:
```

type: none

device: /home/ubuntu/developmentfiles

o: bind

and redo the docker-compose up -d

and in the folder you will find all the files ;)

## Possible errors and solutions

### How to create the broker

- 1) as ip put the internal ip of the machine ifconfig from terminal and port 1026!
- 2) as access link option A) put the public ip or B) always the internal one with port 8443

### How to access virtuoso

via ssh do a tunnelling on port 8890 and access the interface from browser

if you put the sparql slash you can do the queries <http://localhost:8890/sparql>

like this

```
select distinct ?c {  
  ?s to ?c.  
}
```

To see if there is anything in kb

### How to handle the problem No registration in the context broker No registration in kb when we create the device

- 1) redo the procedure and do F12 and go to network - fetch x. preview and go to the bottom to see the whole error for good
- 2) check the logs in servicemap-iot-conf insert if there is anything
- 3) `sudo docker-compose exec servicemap bash`
- 4) then `cd logs` and go to the last access
- 5) `cat localhost_access_log.2023-08-30.txt | grep iot` and see what happens

### How to increase the timeout time before the platform kicks you out

- 1) turn off the dashboard builder container.
- 2) in the folder where the configuration files reside go to the folder indicated in the quotes "dashboard-builder-conf"
- 3) Open the file "general.ini"
- 4) for the values `sessionDuration[dev]`, `sessionDuration[test]` and `sessionDuration[prod]` set the desired value (in seconds, so one day is worth 86400)
- 5) save the file
- 6) restart the container



"docker-compose down dashboard-builder", to be run in the console in the folder where the docker-compose.yaml is located

to bring it up you replace down with up

you may want to omit dashboard-builder; in that case pull down everything or pull up everything (pulling up something already running has no effect)

for "up" the console will be occupied by the output of the container(s), if more than one, to go into detached mode you can use the -d flag

All quiet... docker-compose ps all up but then log in from browser and white screen with no errors on console

Solution: there is an entry in the domain database that is left 'dashboard' and needs to be changed to the one specified... maybe in future versions of the builder it will be fixed automatically however the steps to recover it are:

1 find in the docker-compose the psw of the mysql db

2A access either with workbench or

2B do docker bash in the dashboard container and run mysql -u root -p

-use dashboard

- select \* from Domains and check if everything is ok

- possibly do the field update

Orion's delegation to the group

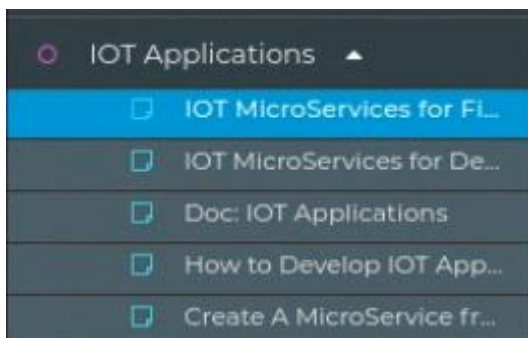
During the first rootadmin login, after taking over Orion (by doing edit without editing), it may sometimes happen that it is not possible to delegate the broker to a group.

**Error: *Delegated groupname can't be empty***

**Solution:** Perform restart of the personaldata container.

No microx IOT app is accessible from the submenu

Error:



**Solution:** Performed docker-compose restart.

## Orion does not receive data from nodeRED (1)

The node that subscribes and sends data on Orion gives the following error

```
"failed to update, reason: {"statusCode":401, "headers":{"x-content-type-options": "nosniff", "x-xss-protection": "1; mode=block", "cache-control": "no-cache, no-store, max-age=0, must-revalidate", "pragma": "no-cache", "expires": "0", "strict-transport-security":"max-age=31536000 ; includeSubDomains", "x-frame-options": "DENY", "transfer-encoding": "chunked", "date": "Tue, 19 Jul 2022 10:43:08 GMT", "connection": "close"}, "payload":{"result\":"false,\"message\":"The passed access token is not valid"}"
```

**Solution:** verify that the password for rootadmin is consistent between the password that was set in the microX generator (and that is used by dashboard-builder by reading the dashboard-builder-conf/personaldata.ini file to access private devices) with the password set in ldap/keyclock.

## Orion does not receive data from nodeRED (2)

NodeRED fails to communicate with Orion, giving the following error:

```
"failed to update, reason: {"errno": "ECONNREFUSED", "code": "ECONNREFUSED", "syscall": "connect", "address": "5.189.175.163", "port":8443}"
```

**Solution:** check that the connection parameters to Orion use the container name and not the IP (especially if port mapping has been removed to close ports). Then in the tab with the connection parameters to Orion (see figure).

Enter the name of the Orion container in IP (e.g. orion-001) and the name of the brokerfilter in AccessLink (e.g. orionbrokerfilter-001). Then restart the containers.

## Orion stalls (following a system crash)

As a result of crashes, Orion may stall by reporting the following error:

```
orion-001_1 | time=2022-11-16T11:26:03.258Z | lvl=ERROR | corr=N/A | trans=N/A | from=N/A | srv=N/A | subsrv=N/A | comp=Orion | op=contextBroker.cpp[432]:pidFile | msg=PID-file '/tmp/contextBroker.pid' found. A broker seems to be running already
```

**Solution:** delete and recreated the container with

```
docker-compose stop orion
docker-compose rm orion
docker-compose up -d orion
```

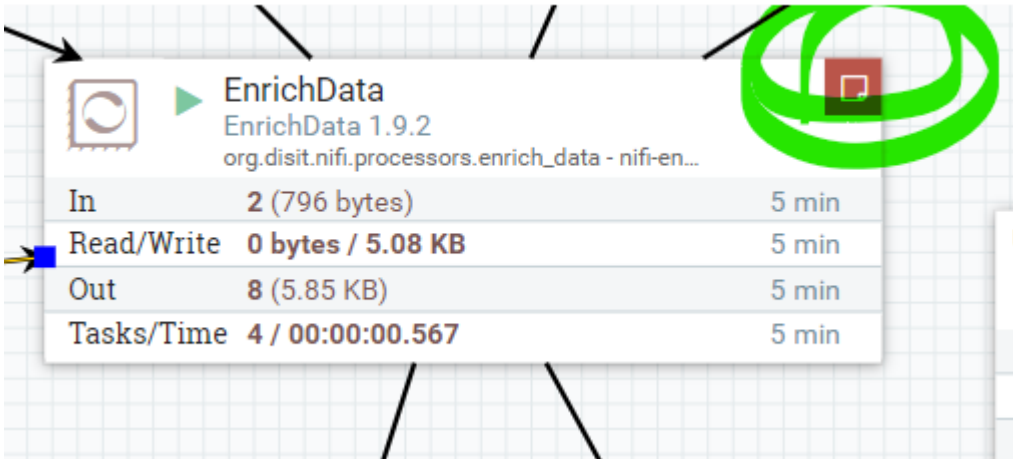
or with

```
docker-compose up -d --force-recreate --no-deps orion
```

change root psw and nifi should be updated

do tunnelling <https://localhost:9090>

then in enrich date you double click on the

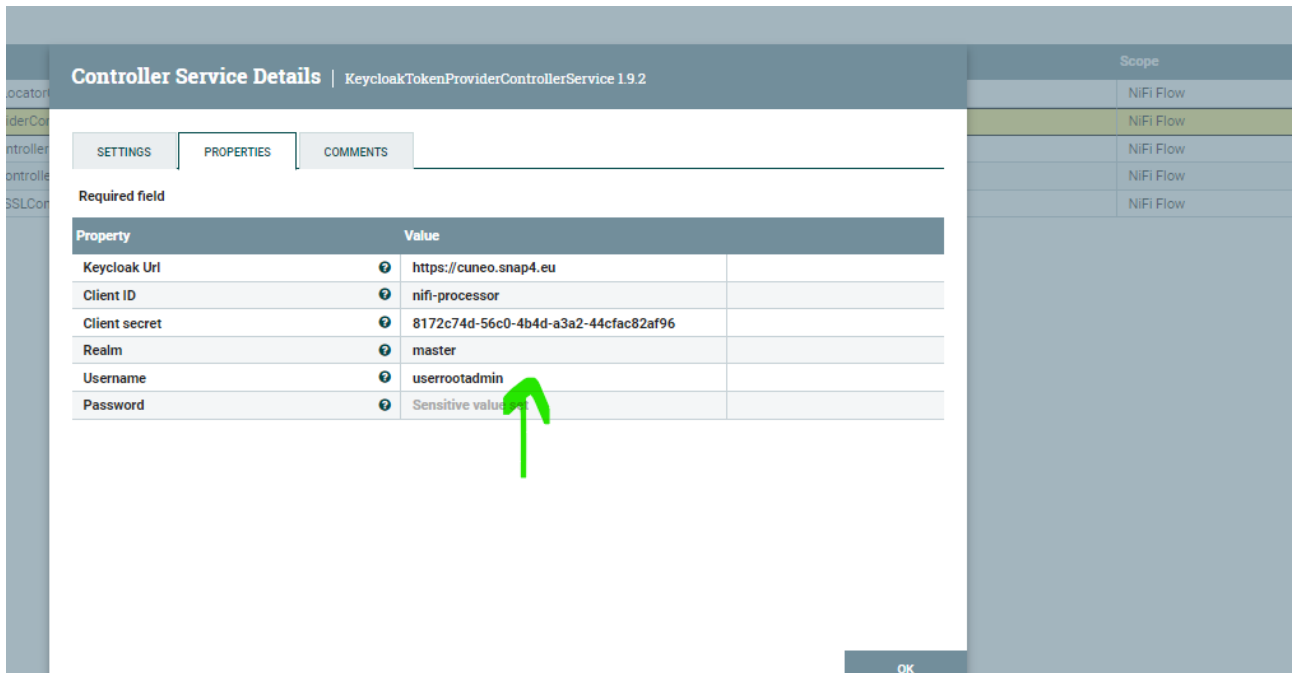


Property	Value
Enrichment Source Client Service	ServiceMapOAuthControllerService2

NIFI Flow Configuration

GENERAL CONTROLLER SERVICES

Name	Type	Bundle	State	Scope
IOTDirectoryOAuthLocatorControllerService	IOTDirectoryOAuthLocatorControllerService 1.9.2	org.disit.nifi.processors.enrich_data - nifi-resource-locator-servi...	Enabled	NIFI Flow
KeycloakTokenProviderControllerService	KeycloakTokenProviderControllerService 1.9.2	org.disit.nifi.processors.enrich_data - nifi-qaath-token-provider-s...	Enabled	NIFI Flow
OwnershipOAuthControllerService	OwnershipOAuthControllerService 1.9.2	org.disit.nifi.processors.enrich_data - nifi-ownership-client-serv...	Enabled	NIFI Flow
ServiceMapOAuthControllerService2	ServiceMapOAuthControllerService 1.9.2	org.disit.nifi.processors.enrich_data - nifi-servicemap-client-serv...	Enabled	NIFI Flow
StandardRestrictedSSLContextService	StandardRestrictedSSLContextService 1.16.2	org.apache.nifi - nifi-ssl-context-service-nar	Enabled	NIFI Flow



reactivates all lightning

also change it in the compose that of orionbrokerfilter

do stop first then rm the docker-compose orionbrokerfilter-001

then up -d

you change root psw and the device is not there in the data-inspector

go inside the dashboard-builder-conf folder

in personalData.ini change to userrootadmin psw with new one

### Data crashes on NiFi in enrich-data

Passing data from Orion through NiFi hangs in enrich-data, with the following error:

```
EnrichData[id=03fcba02-017c-1000-10b5-1b63ebcd9871] {"reason":
"org.apache.http.client.HttpResponseException while handling the Servicemap response body. Routing
to
FAILURE_RELATIONSHIP", "content": {"id": "testWind", "type": "sensor", "dateObserved": {"type": "string", "va
```

```

"ue":"2022-07-
19T11:07:20.261Z","metadata":{},"latitude":{"type":"float","value":"43.79221","metadata":{}}, "longi
tude":{"type":"float","value":"11.30587","metadata":{}}, "model":{"type":"string","value":"windModel"
,"metadata":{}}, "wind":{"type":"float","value":"8.176945692257064", "metadata":{}}, "date_time":
"2022-07-19T11:07:20.532Z"}, "exception":
"org.disit.nifi.processors.enrich_data.enrichment_source.EnrichmentSourceException:
org.apache.http.client.HttpResponseException while handling the Servicemap response body.", "cause":
"org.apache.http.client.HttpResponseException: status code: 400", "ff-uuid": "433c8483-b384-48df-
9ba9-ccdd7990a79b", "Servicemap_response":{" \n \failure : \ERROR\", \n \httpcode\" : 400, \n \n
message\" : \invalid apikey\", \n \n \"apiDoc\" : \http://www.disit.org/6991\" \n}"}

```

**Solution:** enrich-data cannot access servicemap, due to problems with APIKEY (which can be changed during MicroX generation). To fix the apikey, it must be removed from enrich-data's settings.

**Note:** This correction was done directly, so it is not perfectly clear how to go about removing the apikey from enrich-data.

### NiFi does not save to opensearch/elasticserach

As with enrich-data, in this case the NiFi stream crashes on the node that goes to write to opensearch. The problem is that the password for opensearch (\$#opensearch-admin-pwd#\$ in placeholder) is not properly set.

**Solution:** change the password.

**Notes:** again the operation was done by us.

### The device is not in the service map

Error: an alert appears with " no access to http://...link..al device/Organization/testdevice"

But the device is visible in the data inspector

solution: check and remember to put the public device or else in service-map it doesn't show!

### Device does not appear in data-inspector

The created device does not appear in data inspector: running2.txt is not there and the logs in **dashboard-cron** report

```
# cat health.log
```

```
PHP Warning: syntax error, unexpected '$' in ../conf/ssl_expose.php on line 2
```

```
in /var/www/html/dashboardSmartCity/config.php on line 27
```

```
PHP Warning: Invalid argument supplied for foreach() in /var/www/html/dashboardSmartCity/config.php on line 29
```

```
Starting HealthinessCheck SCRIPT at: 2022-07-19 15:45:01+02:00
```

```
    Updating : blueCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
    Updating : greenCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
    Updating : redCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
    Updating : state at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
    Updating : whiteCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```
    Updating : yellowCode at: 2022-07-19 15:45:06+02:00 --> healthiness = false
```

```

1 FINISHED HEALTH CHECK FOR DEVICE: FirstAid
End HealthinessCheck SCRIPT at: 2022-07-19 15:45:06

# cat feed-iot.log

PHP Warning: syntax error, unexpected '$' in ../conf/ssl_expose.php on line 2

in /var/www/html/dashboardSmartCity/config.php on line 27

PHP Warning: Invalid argument supplied for foreach() in /var/www/html/dashboardSmartCity/config.php
on line 29

*** Starting IOT_Sensor_FeedDashboardWizard SCRIPT at: 2022-07-19 15:50:01

1 - IOT DEVICE: , MEASURE:

2 - IOT DEVICE: http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind, MEASURE:
http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind/dateObserved

End IOT_Sensor_FeedDashboardWizard SCRIPT at: 2022-07-19 15:50:01

# cat feed-iot-app.log

PHP Warning: syntax error, unexpected '$' in ../conf/ssl_expose.php on line 2

in /var/www/html/dashboardSmartCity/config.php on line 27

PHP Warning: Invalid argument supplied for foreach() in /var/www/html/dashboardSmartCity/config.php
on line 29

*** Starting FeedT IO_App SCRIPT at: 2022-07-19 15:51:02

End Feed IOT_App SCRIPT at: 2022-07-19 15:51:02

```

In the dashboard.wizard table of the DB the device appears but set to old

```

| 15715 | From IOT Device to KB | Sensor | IoTSensor | Organization:orion-1:testWind | single_marker
| http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind | NULL | NULL | sensor_map
| no | direct | yes | yes | NULL | NULL |
api/v1/?serviceUri=http://www.disit.org/km4city/resource/iot/orion-
1/Organization/testWind&format=json | false | NULL | 2022-07-19 16:05:01 | private | Organization |
43.79221 | 11.30587 | NULL | aGxzcz11VUM1V1hUdUkveE50REZIUT09 | | | old |

| 15716 | From IOT Device to KB | Sensor | IoTSensor | dateObserved | Organization:orion-1:testWind
| single_marker | http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind | NULL |
NULL | time | no | direct | yes | yes | NULL | NULL |
api/v1/?serviceUri=http://www.disit.org/km4city/resource/iot/orion-
1/Organization/testWind&format=json | false | NULL | 2022-07-19 16:05:01 | private | Organization |
43.79221 | 11.30587 | timestamp | aGxzcz11VUM1V1hUdUkveE50REZIUT09 | | | old |

| 15717 | From IOT Device to KB | Sensor | IoTSensor | wind | Organization:orion-1:testWind |
single_marker | http://www.disit.org/km4city/resource/iot/orion-1/Organization/testWind | NULL |
NULL | float | no | direct | yes | NULL |
api/v1/?serviceUri=http://www.disit.org/km4city/resource/iot/orion-
1/Organization/testWind&format=json | false | NULL | 2022-07-19 16:05:01 | private | Organization |
43.79221 | 11.30587 | m/s | aGxzcz11VUM1V1hUdUkveE50REZIUT09 | | | old |

```

The field a old means that healthiness has failed. The healthiness is based on apikey servicemap. The problem is that apikey servicemap assumes it is a hexadecimal, while the generator generates random letters numbers.

**Solution:** update the servicemap version to v4.3, then put the old values back to null

## Change session timeout

**\*\*** How to increase the timeout time before the platform kicks you out

1. Turn off the container dashboard builder;
2. in the folder where the configuration files reside go to the folder indicated in the quotes "dashboard-builder-conf";
3. open the "general.ini" file;
4. for the values sessionDuration[dev], sessionDuration[test] and sessionDuration[prod] set the desired value (in seconds, so one day is worth 86400);
5. save the file;
6. restart the container;

"docker-compose down dashboard-builder", to be run in the console in the folder where the docker-compose.yaml is located

to bring it up you replace down with up

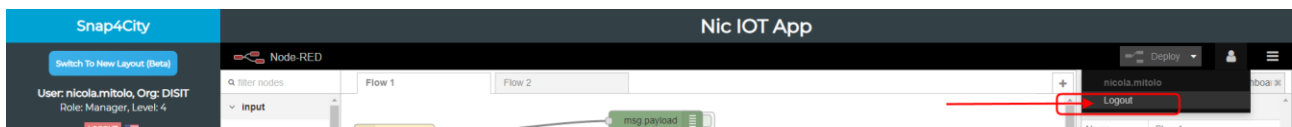
you may want to omit dashboard-builder; in that case pull down everything or pull up everything (pulling up something already running has no effect)

for "up" the console will be occupied by the output of the container(s), if more than one, to go into detached mode you can use the -d flag

### Cleaning up the tokens of an iot app

```
docker-compose stop iotapp-xxx
cd iotapp-xxx
ls -la
rm refresh_token*
cd ...
docker-compose start iotapp-xxx
```

Enter IOTApp and be sure to logout from the icon in the upper right corner



Re-enter IOTApp to make sure you are logging in with the correct user.

### Set up tunnelling for access to NIFI interface.

```
ssh -L 9090:localhost:9090 username@IP
```

```
>Yes
```

```
>password
```

access the URL <https://localhost:9090>

Login with password inside the docker-compre

user: admin

### Broker subscription list

Broker subscription

Go to *IP:1026/v2/subscriptions*

### Enabling a user to CSBL

Add the user name in the *TrustedUsers* table of the *Dashboard* database.

### Increase the default refresh time of the Micro X

It can happen that although the device healthness values are set with low values, in the data inspector the healthness dot remains green even though the data is not coming in. This may be due to a refresh time value set by default during the Micro X configuration that must then be re-edited to make it consistent with the desired healthness values.

**[BUG BEING SOLVED]**

### Update dictionary after adding new value type

If value types are added to the dictionary, there may be errors when creating devices with the new value types.

Go to the `servicemap-conf` folder

Update the dba user's virtuoso password in the `./update-valuetypes.sh` script

The dba user's password is found in the `docker-compose.yml` file and is randomly generated .

Save the script and execute it.

Possibly it may be useful to put a cron every 10 minutes as in Snap4City.

### If api iot-search does not work

if api iot-search does not return anything, check that nifi is writing data to right index, right index is *iot-device-state* and not *device-iot-organization*

Connect to nifi and check the put-elastic at the top that saves the last status of the devices, if it is wrong:

1. stop the put-elastic
2. open kibana as userrootadmin to go into dev tools and send the following



```
POST _reindex
{
  "source":{
    "index": "device-iot-organization"
  },
  "dest":{
    "index": "iot-device-state"
  }
}
```

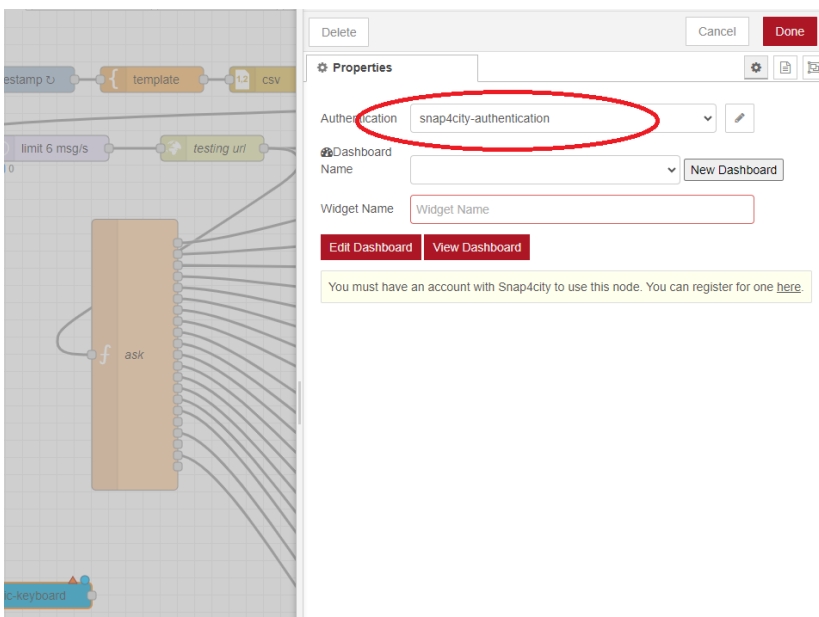
Which is used to re-index any data already on the right index

3. at this point go back to nifi and change in the put-elastic block the name of the index to save to and put the right value **iot-device-state** and save
4. restart the put-elastic

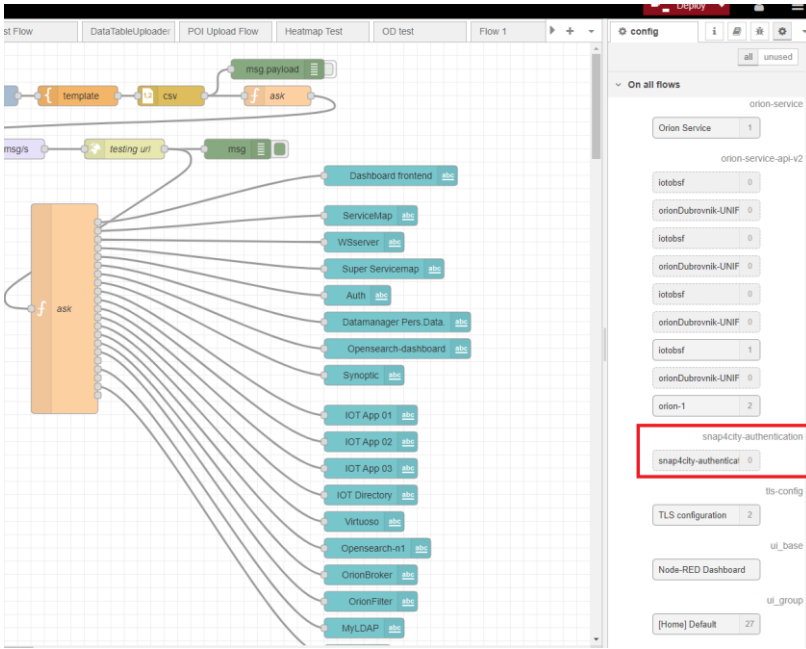
(be careful not to change the other put-elastic that saves the data)

### Empty authentication on Node-RED stream.

In the iot apps of the Micro Xs there is an authentication object (empty) that automatically sets itself as authentication to user blocks.



To solve delete from the settings the authentication as in the image



(Fabrizio is resolving)

Clean up the db from the sample data in the MicroXs.

the rows with high\_level\_type = "External Service" in the DashboardWizard table in the Dashboard database

To erase them all:

```
DELETE FROM Dashboard.DashboardWizard WHERE high_level_type = "External Service";
```

Repeat for heatmaps and "Special events" if not of interest

## IOTApp Flow

```
[{"id":"955e4a58ca1c42e4","type":"tab","label":"Flow
1","disabled":false,"info":"","env":[]},{id:"75a8edb4987afe1d","type":"inject","z":"955e4a58ca1c42e
4","name":"","props":[{"p":"payload"},{"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"once
Delay":0.1,"topic":"","payload":"","payloadType":"date","x":120,"y":140,"wires":[["d62e1d387d5c970
3"]]},{"id":"89ee7a76aa229b75","type":"orion-out-api-
v2","z":"955e4a58ca1c42e4","name":"","tls":"","service":"97a0f29a.0bf4f","entype":"","enid":"","user
k1":"","passk2":"","tenant":"","servicepath":"","apikey":"","basicAuth":"","x":440,"y":160,"wires":[["211
7e58e9449612d"]]},{"id":"83fce0279032f172","type":"debug","z":"955e4a58ca1c42e4","name":"","a
ctive":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","
statusType":"auto","x":410,
"y":260,"wires":[]},{id":"d62e1d387d5c9703","type":"function","z":"955e4a58ca1c42e4","name":"","f
unc":"adesso = new Date().toISOString()\nmsg.payload =
{id:\\"dajedevicel\", \"type\":\\"testl\", \"dateObserved\":\\"typel\":\\"stringl\", \"value\":now},\\"latemperature
l\":\\"typel\":\\"floatl\", \"value\":50.1}\n\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":180,"y":240,"wires":[["83fce0279032f
172","89ee7a76aa229b75"]]},{"id":"2117e58e9449612d","type":"debug","z":"955e4a58ca1c42e4","
name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":
```

```
"","statusType":"auto","x":630,"y":160,"wires":[]},{id:"97a0f29a.0bf4f","type":"orion-service-api-v2","contextbroker":"orion-001","selectedContextbroker":"","name":"orion-001","authentication":"","url":"orionbrokerfilter-001","port":"8443"}]
```