Behind the ESS public operations screen

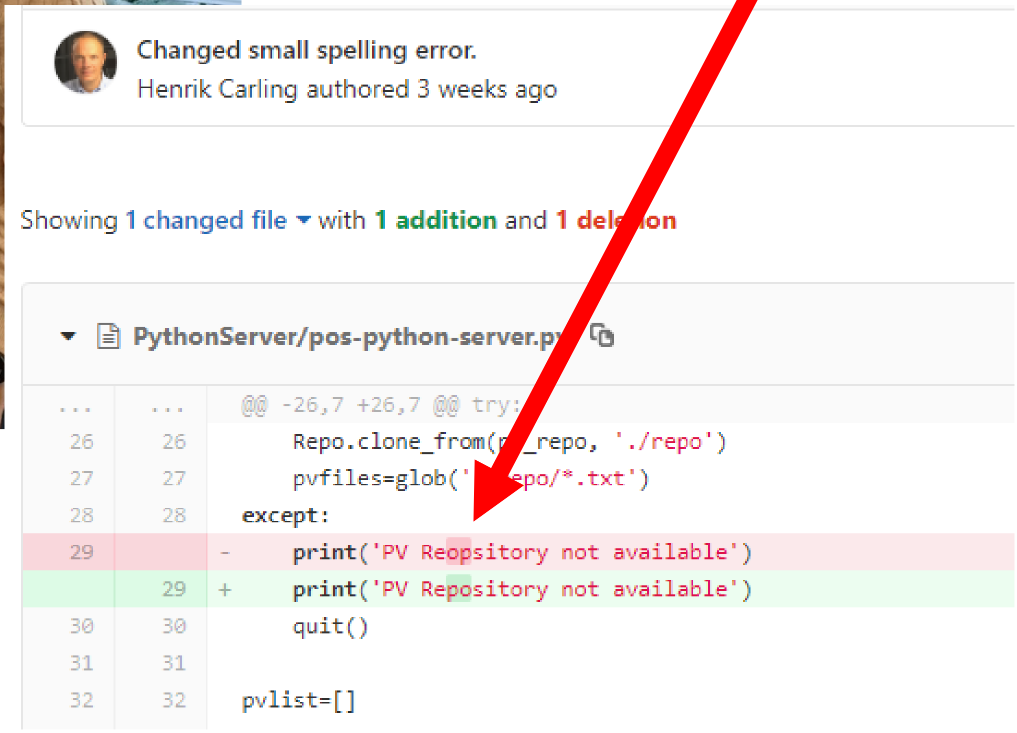
Emanuele Laface

Henrik Carling

ESS/ICS

Date: 2019-06-05

Background



Changed small spelling error.

Henrik Carling authored 3 weeks ago

Showing 1 changed file with 1 addition and 1 deletion

PythonServer/pos-python-server.py

```diff
@@ -26,7 +26,7 @@ try:
     Repo.clone_from(p_repo, './repo')
     pvfiles=glob('./repo/*.txt')
 except:
-    print('PV Reopsitory not available')
+    print('PV Repository not available')
     quit()

 pvlist=[]
```

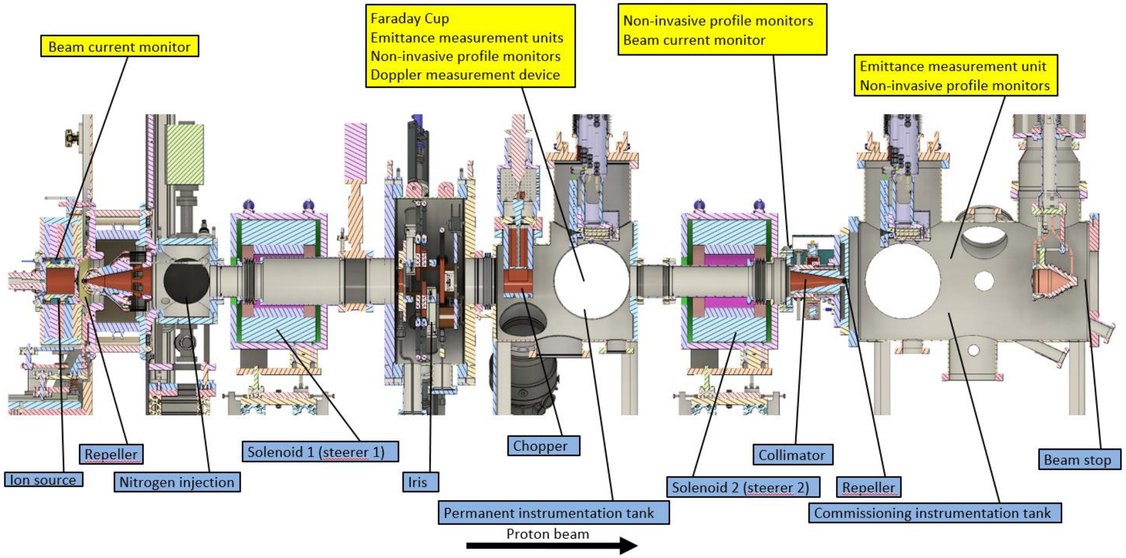
Background

Integrated Control Systems

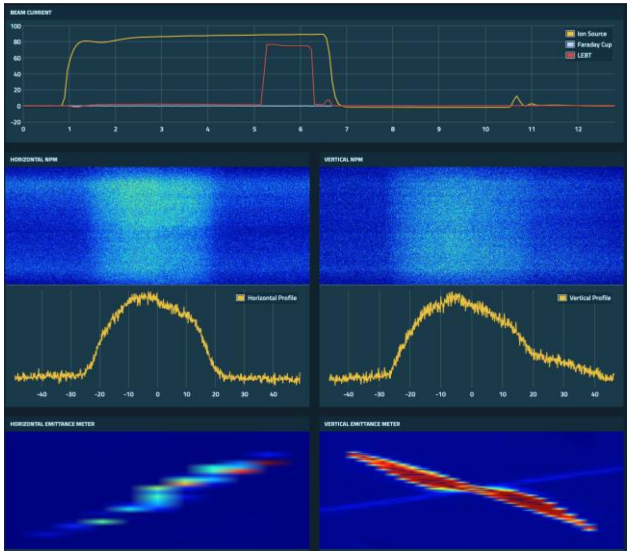
- Use case 1: quick access to operational status overview for ESS staff from anywhere
- Use case 2: provide “on-line” operational status of the ESS facility to the public/internet

- Programmer idea 1: display EPICS data with the minimum possible quantity of code
- Programmer idea 2: provide extensible public http/json access to EPICS data

- Ambition: create a tool for simple EPICS data representation for everyone at ESS

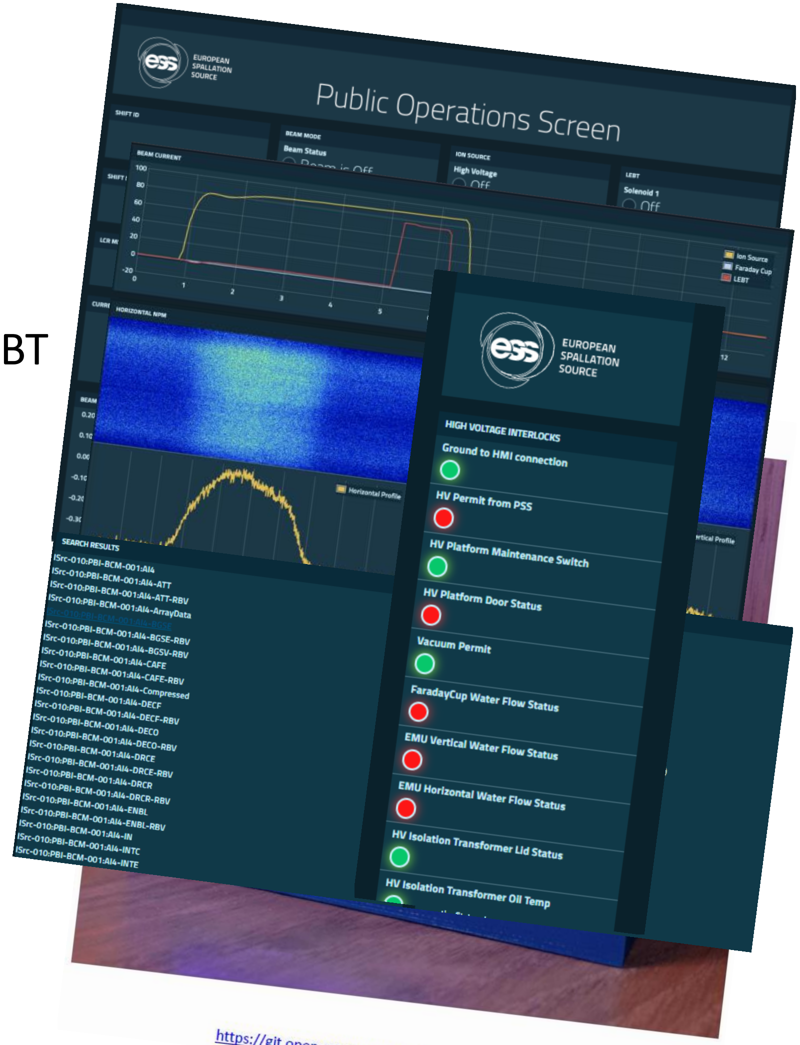


Beam current monitor
Faraday Cup
Emittance measurement units
Non-invasive profile monitors
Doppler measurement device
Non-invasive profile monitors
Beam current monitor
Emittance measurement unit
Non-invasive profile monitors
Ion source
Repeller
Nitrogen injection
Solenoid 1 (steerer 1)
Iris
Chopper
Permanent instrumentation tank
Proton beam
Solenoid 2 (steerer 2)
Collimator
Repeller
Commissioning instrumentation tank
Beam stop



Current status

- Current status
  - Three customized screens
    - Operation
    - Interlocks status
    - Beam instruments of the Ion source and LEBT
  - Graphic profile customization for ESS
  - Search engine for PVs
    - Requires some additional Python code
  - Python interface that exposes JSON files
    - Accessible from the internet



SEARCH RESULTS

```
ISrc-010:PBI-BCM-001:AI4
ISrc-010:PBI-BCM-001:AI4-ATT
ISrc-010:PBI-BCM-001:AI4-ATT-RBV
ISrc-010:PBI-BCM-001:AI4-ArrayData
ISrc-010:PBI-BCM-001:AI4-BGSE
ISrc-010:PBI-BCM-001:AI4-BGSE-RBV
ISrc-010:PBI-BCM-001:AI4-BGSV-RBV
ISrc-010:PBI-BCM-001:AI4-CAFE
ISrc-010:PBI-BCM-001:AI4-CAFE-RBV
ISrc-010:PBI-BCM-001:AI4-Compressed
ISrc-010:PBI-BCM-001:AI4-DECF
ISrc-010:PBI-BCM-001:AI4-DECF-RBV
ISrc-010:PBI-BCM-001:AI4-DECO
ISrc-010:PBI-BCM-001:AI4-DECO-RBV
ISrc-010:PBI-BCM-001:AI4-DRCE
ISrc-010:PBI-BCM-001:AI4-DRCE-RBV
ISrc-010:PBI-BCM-001:AI4-DRCR
ISrc-010:PBI-BCM-001:AI4-DRCR-RBV
ISrc-010:PBI-BCM-001:AI4-ENBL
ISrc-010:PBI-BCM-001:AI4-ENBL-RBV
ISrc-010:PBI-BCM-001:AI4-IN
ISrc-010:PBI-BCM-001:AI4-INTC
ISrc-010:PBI-BCM-001:AI4-INTE
```

HIGH VOLTAGE INTERLOCKS

Ground to HMI connection 🟢

HV Permit from PSS 🔴

HV Platform Maintenance Switch 🟢

HV Platform Door Status 🔴

Vacuum Permit 🟢

FaradayCup Water Flow Status 🔴

EMU Vertical Water Flow Status 🔴

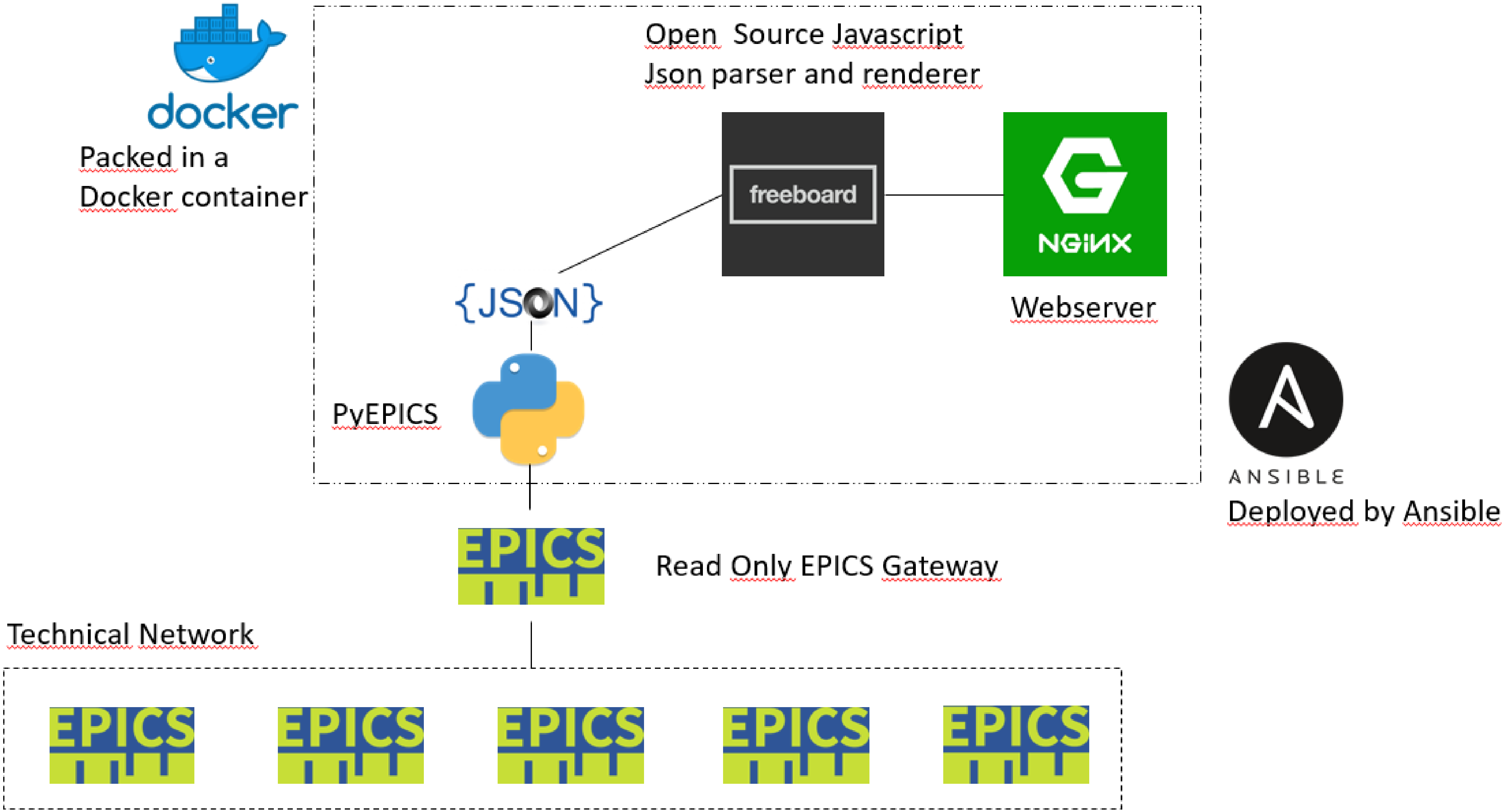
EMU Horizontal Water Flow Status 🔴

HV Isolation Transformer Lid Status 🟢

HV Isolation Transformer Oil Temp

https://git.open-source.se/henrikcarling/ESS_monitor

Current deployment



Packed in a Docker container

Open Source Javascript Json parser and renderer

Webserver

PyEPICS

Deployed by Ansible

Read Only EPICS Gateway

Technical Network

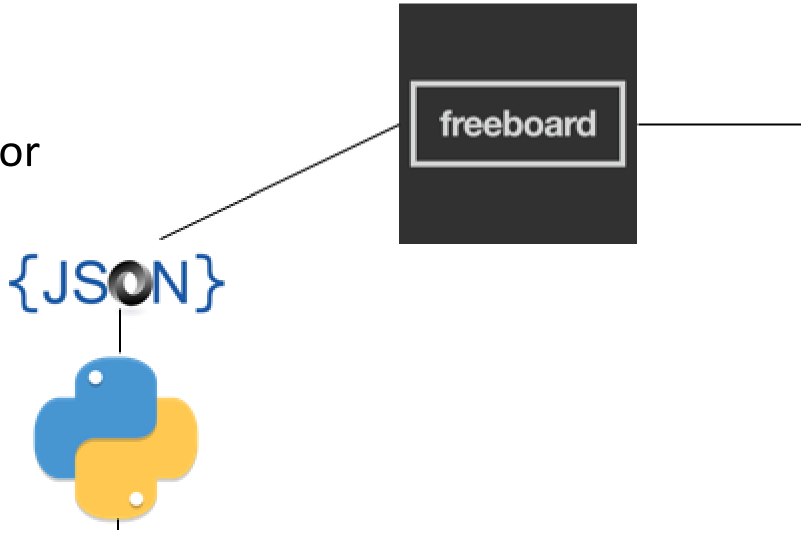
Programmers perspective

- The code is basically a translator from EPICS to Json. (Example: https://pos.esss.lu.se/data/getPublicData.json)

- The JavaScript/rendering part is done through the Freeboard editor without writing code (or very little for some customization)

  https://github.com/Freeboard/freeboard

Open Source Javascript Json parser and renderer

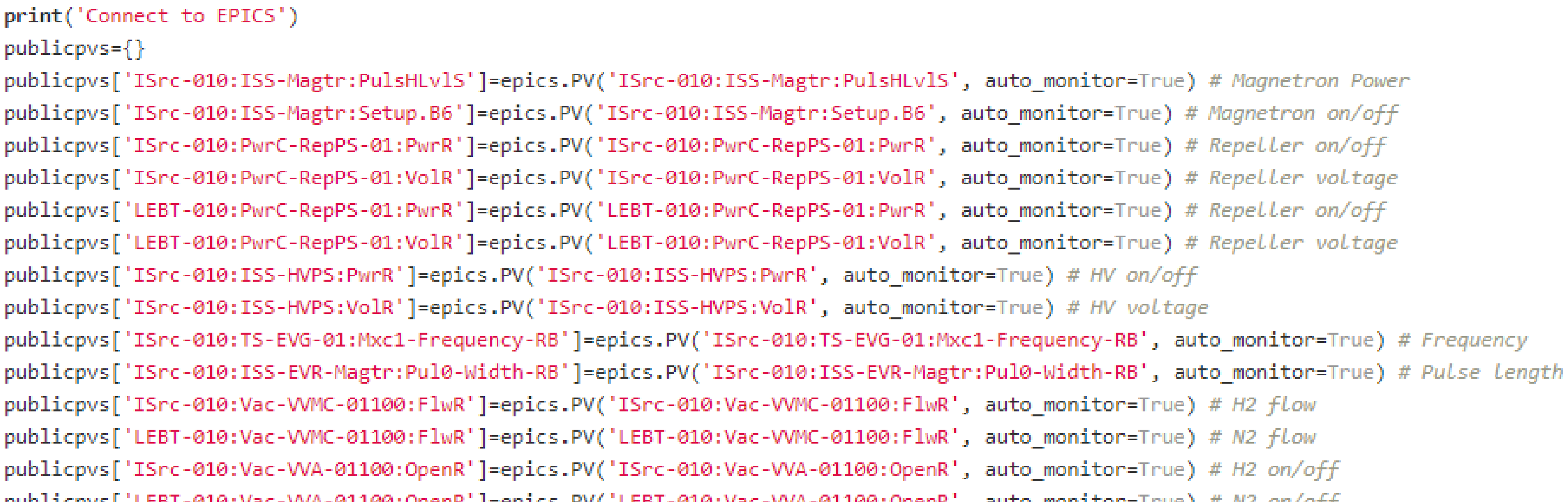


PyEPICS

- The graphic customization is done through a single CSS style sheet

Next steps

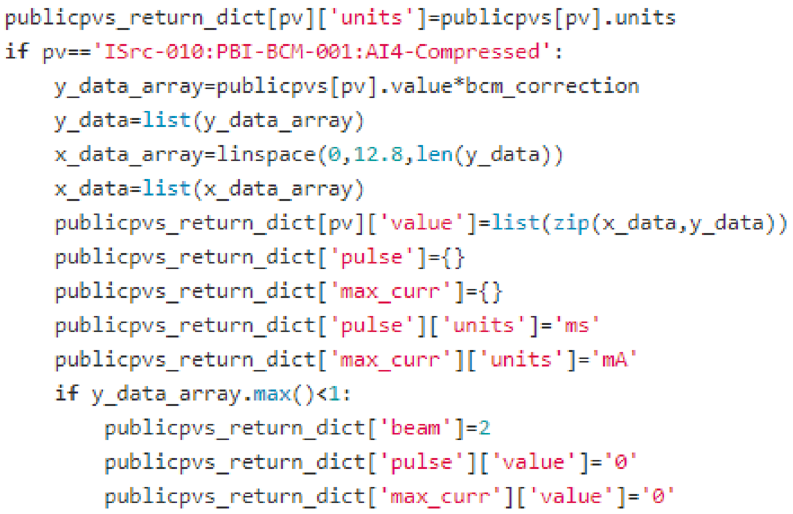
- Improvement potential
  - Polling/monitor
  - The Python polling system uses the monitor function of the pyepics class
    - Any other method times out when the PV is not available - can hang the polling thread
    - The timeout happens also in monitor mode but the variable is always accessible
    - The drawback of the monitor function is that data is received in Python at the speed of the machine: 14 Hz but I am interested in visualizing them at 1 Hz, overloading the network and the CPU of the Python server.
    - Can possibly be solved by IOC filtering?



```
print('Connect to EPICS')
publicpvs={}
publicpvs['ISrc-010:ISS-Magtr:PulsHLvlS']=epics.PV('ISrc-010:ISS-Magtr:PulsHLvlS', auto_monitor=True) # Magnetron Power
publicpvs['ISrc-010:ISS-Magtr:Setup.B6']=epics.PV('ISrc-010:ISS-Magtr:Setup.B6', auto_monitor=True) # Magnetron on/off
publicpvs['ISrc-010:PwrC-RepPS-01:PwrR']=epics.PV('ISrc-010:PwrC-RepPS-01:PwrR', auto_monitor=True) # Repeller on/off
publicpvs['ISrc-010:PwrC-RepPS-01:VolR']=epics.PV('ISrc-010:PwrC-RepPS-01:VolR', auto_monitor=True) # Repeller voltage
publicpvs['LEBT-010:PwrC-RepPS-01:PwrR']=epics.PV('LEBT-010:PwrC-RepPS-01:PwrR', auto_monitor=True) # Repeller on/off
publicpvs['LEBT-010:PwrC-RepPS-01:VolR']=epics.PV('LEBT-010:PwrC-RepPS-01:VolR', auto_monitor=True) # Repeller voltage
publicpvs['ISrc-010:ISS-HVPS:PwrR']=epics.PV('ISrc-010:ISS-HVPS:PwrR', auto_monitor=True) # HV on/off
publicpvs['ISrc-010:ISS-HVPS:VolR']=epics.PV('ISrc-010:ISS-HVPS:VolR', auto_monitor=True) # HV voltage
publicpvs['ISrc-010:TS-EVG-01:Mxc1-Frequency-RB']=epics.PV('ISrc-010:TS-EVG-01:Mxc1-Frequency-RB', auto_monitor=True) # Frequency
publicpvs['ISrc-010:ISS-EVR-Magtr:Pul0-Width-RB']=epics.PV('ISrc-010:ISS-EVR-Magtr:Pul0-Width-RB', auto_monitor=True) # Pulse length
publicpvs['ISrc-010:Vac-VVMC-01100:FlwR']=epics.PV('ISrc-010:Vac-VVMC-01100:FlwR', auto_monitor=True) # H2 flow
publicpvs['LEBT-010:Vac-VVMC-01100:FlwR']=epics.PV('LEBT-010:Vac-VVMC-01100:FlwR', auto_monitor=True) # N2 flow
publicpvs['ISrc-010:Vac-VVA-01100:OpenR']=epics.PV('ISrc-010:Vac-VVA-01100:OpenR', auto_monitor=True) # H2 on/off
publicpvs['LEBT-010:Vac-VVA-01100:OpenR']=epics.PV('LEBT-010:Vac-VVA-01100:OpenR', auto_monitor=True) # N2 on/off
```

Next steps

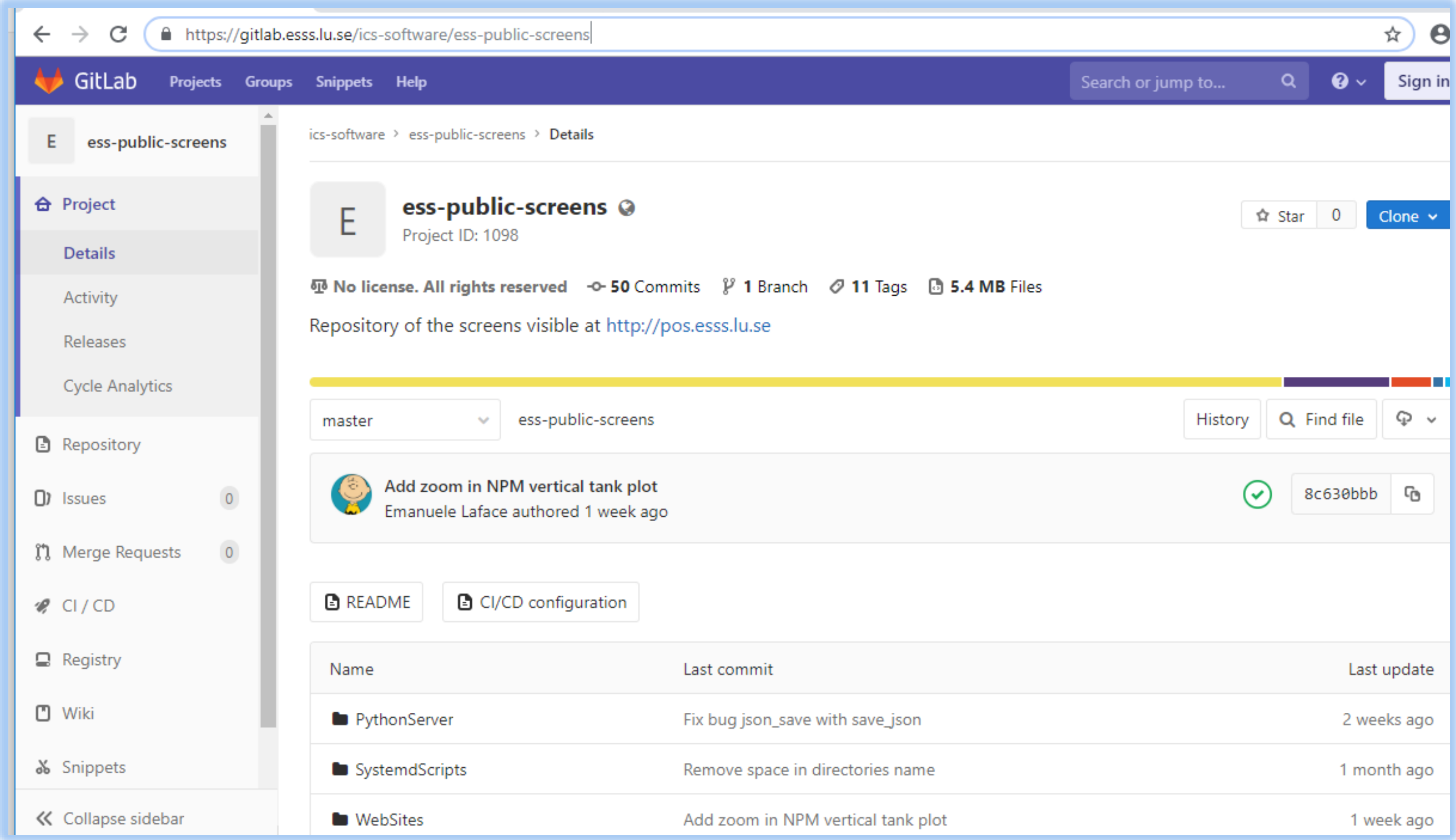
- Improvement potential
  - Flexibility
    - A user that wants a specific interface has to ask for the Python to JSON translator implementation for that set of PVs.
    - The ambition is to create an automatic system for the user to drop the list of PVs somewhere (git?) and have the JSON generated every second.
    - In this way the user can create the interface in Freeboard (or elsewhere) without writing any code



```python
publicpvs_return_dict[pv]['units']=publicpvs[pv].units
if pv=='ISrc-010:PBI-BCM-001:AI4-Compressed':
    y_data_array=publicpvs[pv].value*bcm_correction
    y_data=list(y_data_array)
    x_data_array=linspace(0,12.8,len(y_data))
    x_data=list(x_data_array)
    publicpvs_return_dict[pv]['value']=list(zip(x_data,y_data))
    publicpvs_return_dict['pulse']={}
    publicpvs_return_dict['max_curr']={}
    publicpvs_return_dict['pulse']['units']='ms'
    publicpvs_return_dict['max_curr']['units']='mA'
    if y_data_array.max()<1:
        publicpvs_return_dict['beam']=2
        publicpvs_return_dict['pulse']['value']='0'
        publicpvs_return_dict['max_curr']['value']='0'
```

Open project

- https://gitlab.esss.lu.se/ics-software/ess-public-screens



https://gitlab.esss.lu.se/ics-software/ess-public-screens

ics-software › ess-public-screens › Details

ess-public-screens

Project ID: 1098

No license. All rights reserved **50** Commits **1** Branch **11** Tags **5.4 MB** Files

Repository of the screens visible at http://pos.esss.lu.se

Add zoom in NPM vertical tank plot

Emanuele Laface authored 1 week ago

8c630bbb

Name	Last commit	Last update
PythonServer	Fix bug json_save with save_json	2 weeks ago
SystemdScripts	Remove space in directories name	1 month ago
WebSites	Add zoom in NPM vertical tank plot	1 week ago

Live demo

- https://pos.esss.lu.se/