

# Discussion on PyJapcScout + DataScout

**D. Gamba, G. Sterbini – 28/01/2022**

# PyJapcScout

- **PyJapcScout** is a wrapper on top of **PyJapc** with the following main additional features:
  - Acquired data is fully pythonic and “simple” (to ease conversion/data saving)
  - Acquired data can be (automatically) saved with **parquet data format** using **datascout** package
    - Data can also be converted (or loaded) as **awkward arrays**: this could be the future data format for all our acquisitions/data analysis
- **PyJapcScout is now released in accpy repository (version 0.1.0 from Yesterday)**

```
source /acc/local/share/python/acc-py/base/pro/setup.sh
acc-py venv ./testvenv
source ./testvenv/bin/activate
python -m pip install pyjapcscout
python
>>>from pyjapcscout import PyJapcScout
>>>myPyJapc = PyJapcScout(incaAcceleratorName='CTF')
>>>print(myPyJapc.getSimpleValue('CA.BHB0900/SettingPPM#current',selectorOverride=''))
```

- **(Poor) API documentation:**

<https://acc-py.web.cern.ch/gitlab/abpcomputing/sandbox/pyjapcscout/docs/v0.1.0/>

# datascout

- **datascout** is a simple collection of sweet functions to load/save/convert
  - No Java and/or CERN-control-system-related dependences
  - Used by pyjapcscout to "save" data to file in the control room
  - Should be used **by user** to "load" data for post-processing on his/her **favorite environment** (laptop, SWAN, ...)
- **datascout** is "fully released" (new version 0.0.3 from Yesterday) on [acc-py repository](#), i.e. it can be installed as:

```
# To have access to acc-py repository, if needed:  
#pip install git+https://gitlab.cern.ch/acc-co/devops/python/acc-py-pip-config  
pip install datascout
```

```
#or from gitlab directly:  
pip install git+https://gitlab.cern.ch/abpcomputing/sandbox/datascout.git
```

- (poor) API Documentation:
  - <https://acc-py.web.cern.ch/gitlab/abpcomputing/sandbox/datascout/docs/v0.0.3/>

# Plans and Questions

- **Initial idea was to give inputs for a PyJapc V3, but the latter might not happen**
  - Different plans in CCS for a new Python API to access devices (“**PyDA**”)
    - Maybe **data saving** format we found could be integrated in this new API?
- **Other ideas we had in mind in the past:**
  - making data “aggregators” (e.g. a la **UCAP**) and send **to NXCALS**
    - Did not invest much time in exploring this... seems complicate (on the short term)
  - Use some local library **to store data** (parquet) on **HDFS** then accessible via PySpark the NXCALS cluster
    - In principle possible today – not know if this is a good practice
- **Questions:**
  - Is there the need for **pyjapcscout** on the **short/mid/long term**?
    - Should we invest more time to **add features**?
    - Is there something that can/should **go in PyJapc**?
    - Were **other solution found**?
  - Is **Parquet** and the present **data format + datascout** a **must/useful/used solution**?
    - Should we invest more time to **add features**?