ESRF | The European Synchrotron

Announcing

mxcubecore 1.0
and
MXCuBE-Web 4.0

mxcubecore 1.0

On behalf of the MXCuBE developers committee

… and others

mxcubecore

- The main refactoring work finished last autumn. Big congratulations to everybody involved
- New release/git routine, master stable and development on develop branch. Inspired from OneFlow and GitFlow
- Version 1.0.0 of mxcubecore soon available

mxcubecore

- The work with mxcubecore continues

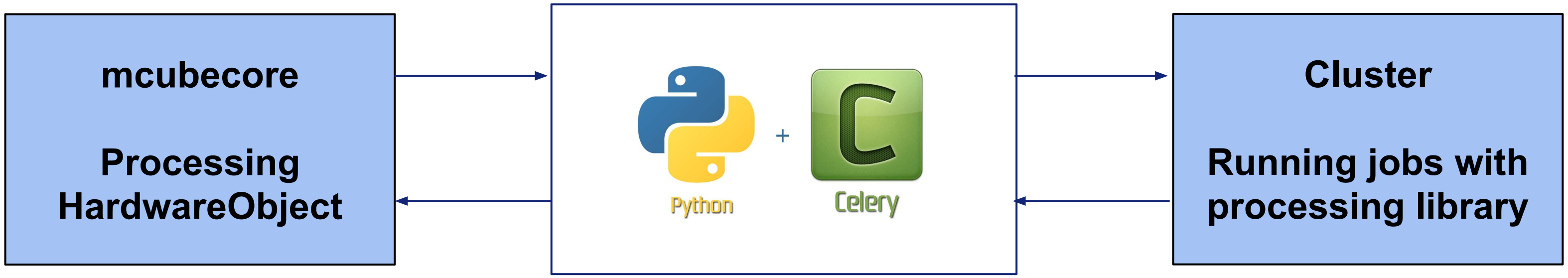
- Two working groups were created based on previous discussions and decisions

- Queue and Workflow working group - with the aim to facilitate porting of automation features such as X-Ray centring
  - At ESRF this has led to the Creation of new Processing hardware object based on Celery
  - Improved QueueEntry based on Pydantic models

- Abstract Diffractometer - Create a common base for diffractometer objects
  - AbstractClass and initial MicroDiffractometer class beeing tested for the new ID29 beamline
  - Arinax have proposed to provide basic implementation for their Microdiffractometer, based on Antonias work

Processing hardware object

- The idea is to provide a new AbstractProcessing object with predefined processing signals
- Default implementation uses Celery to distribute jobs
- A separate python module with a basic set of processing routines is installed on the machine(s) that runs the distributed jobs.


mcubecore
Processing HardwareObject
Python
+
Celery
Cluster
Running jobs with processing library

mxcubecore - continued

The work with mxcubecore continues, apart from the already ongoing work these are topics that have been discussed during the developers meeting.

- AbstractCollect
- Improved signal definitions
- AbstractCentring (based on existing CentringMath)

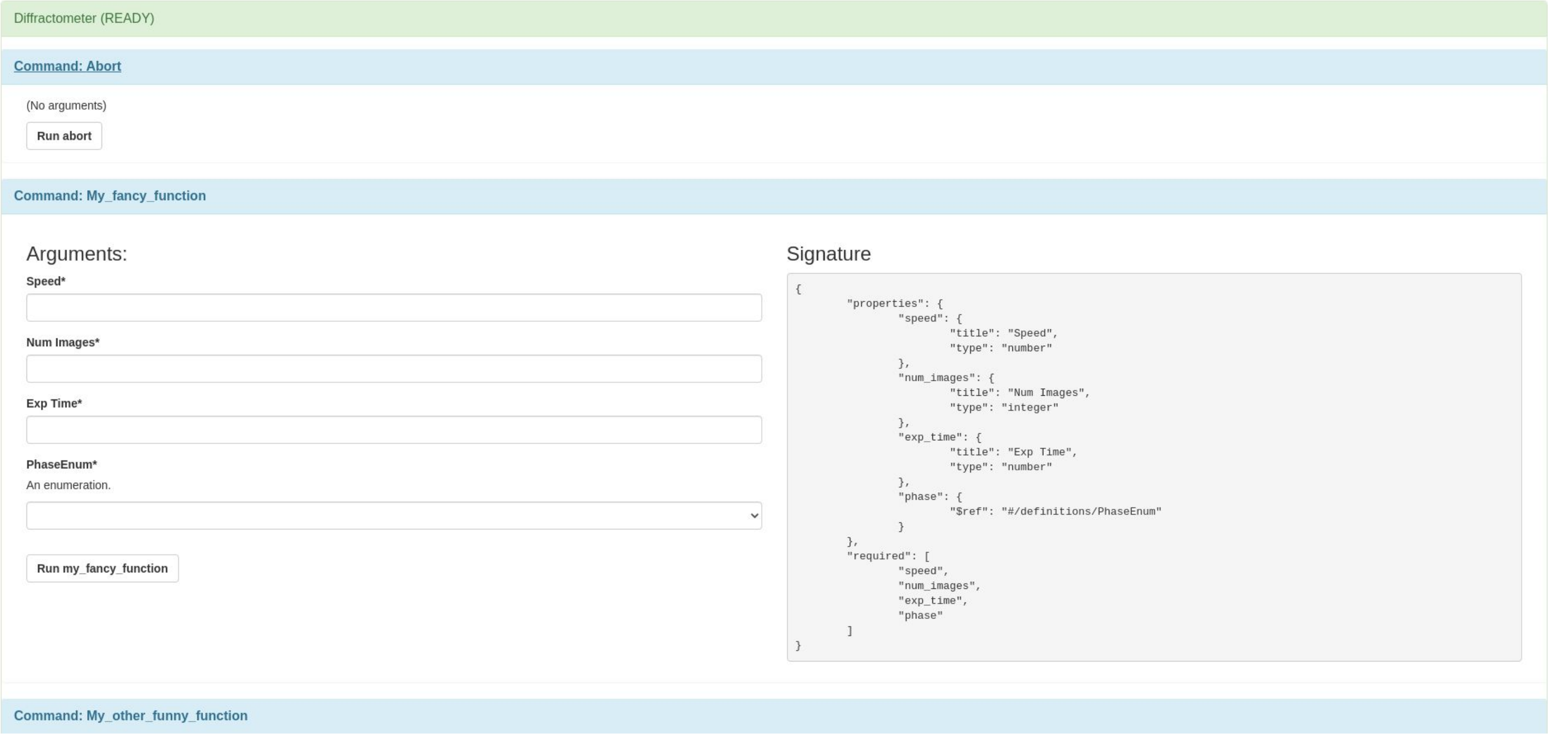
MXCuBE-Web 4

- MXCuBE3 will change name to MXCuBE-Web, first version to be released is MXCuBE-Web v4
- Using new mxcubecore module
- Easier to implement site specific login routine, via login component
- All frontend libraries and build environment have been updated, now using React 17 and Bootstrap 5.
- MXCuBE-Web 4 in use on MASSIF-1

Automatic test/maintenance UI

- Automatic simple test/maintenance UI for type hinted and “exported” (in .xml file) HardwareObject methods

- Possible via typehints, Pydantic and JSONSchema



Diffractometer (READY)

Command: Abort

(No arguments)

Command: My_fancy_function

Arguments:

Speed*

Num Images*

Exp Time*

PhaseEnum*

An enumeration.

Signature

```
{
        "properties": {
                "speed": {
                        "title": "Speed",
                        "type": "number"
                },
                "num_images": {
                        "title": "Num Images",
                        "type": "integer"
                },
                "exp_time": {
                        "title": "Exp Time",
                        "type": "number"
                },
                "phase": {
                        "$ref": "#/definitions/PhaseEnum"
                }
        },
        "required": [
                "speed",
                "num_images",
                "exp_time",
                "phase"
        ]
}
```

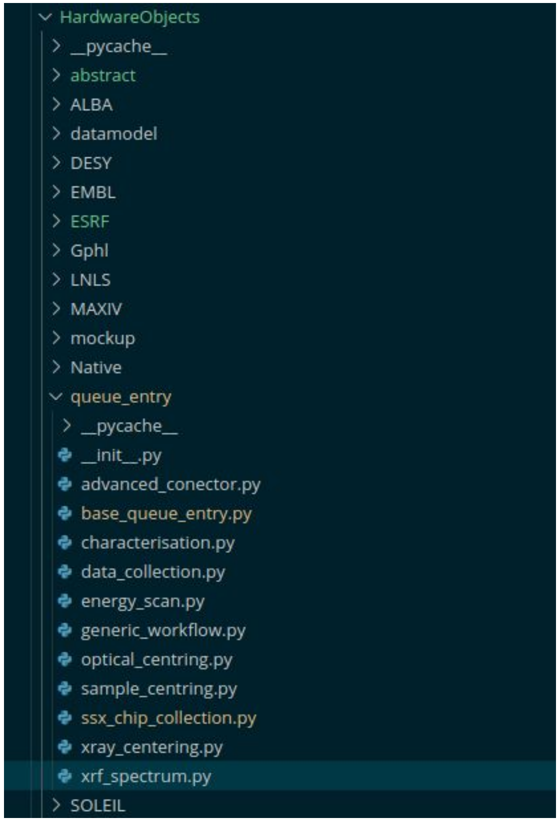
Command: My_other_funny_function

Improved Queue Entry

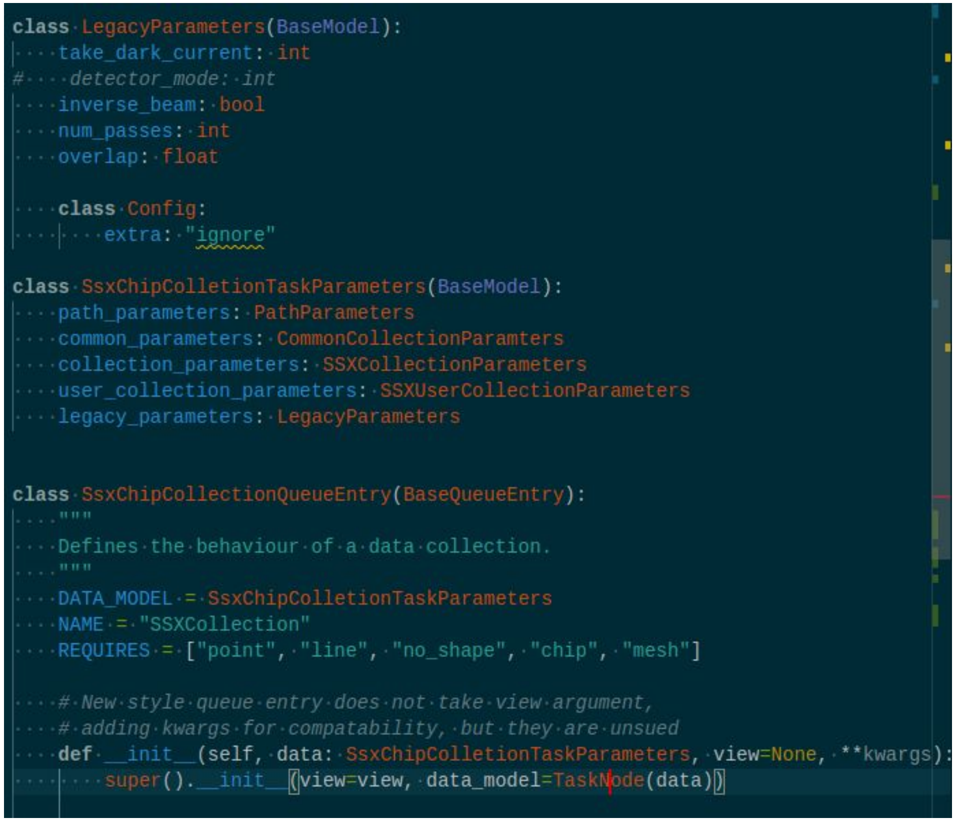
- Same concept used for improved Queue Entry

- Divided current queue_entry.py into idividual .py files
  still imported via queue_entry module (most imports are the same)

- Introduced queue entry that uses Pydantic models to define parameters



```
HardwareObjects
  > __pycache__
  > abstract
  > ALBA
  > datamodel
  > DESY
  > EMBL
  > ESRF
  > Gphl
  > LNLS
  > MAXIV
  > mockup
  > Native
  queue_entry
    > __pycache__
    __init__.py
    advanced_conector.py
    base_queue_entry.py
    characterisation.py
    data_collection.py
    energy_scan.py
    generic_workflow.py
    optical_centring.py
    sample_centring.py
    ssx_chip_collection.py
    xray_centering.py
    xrf_spectrum.py
  > SOLEIL
```

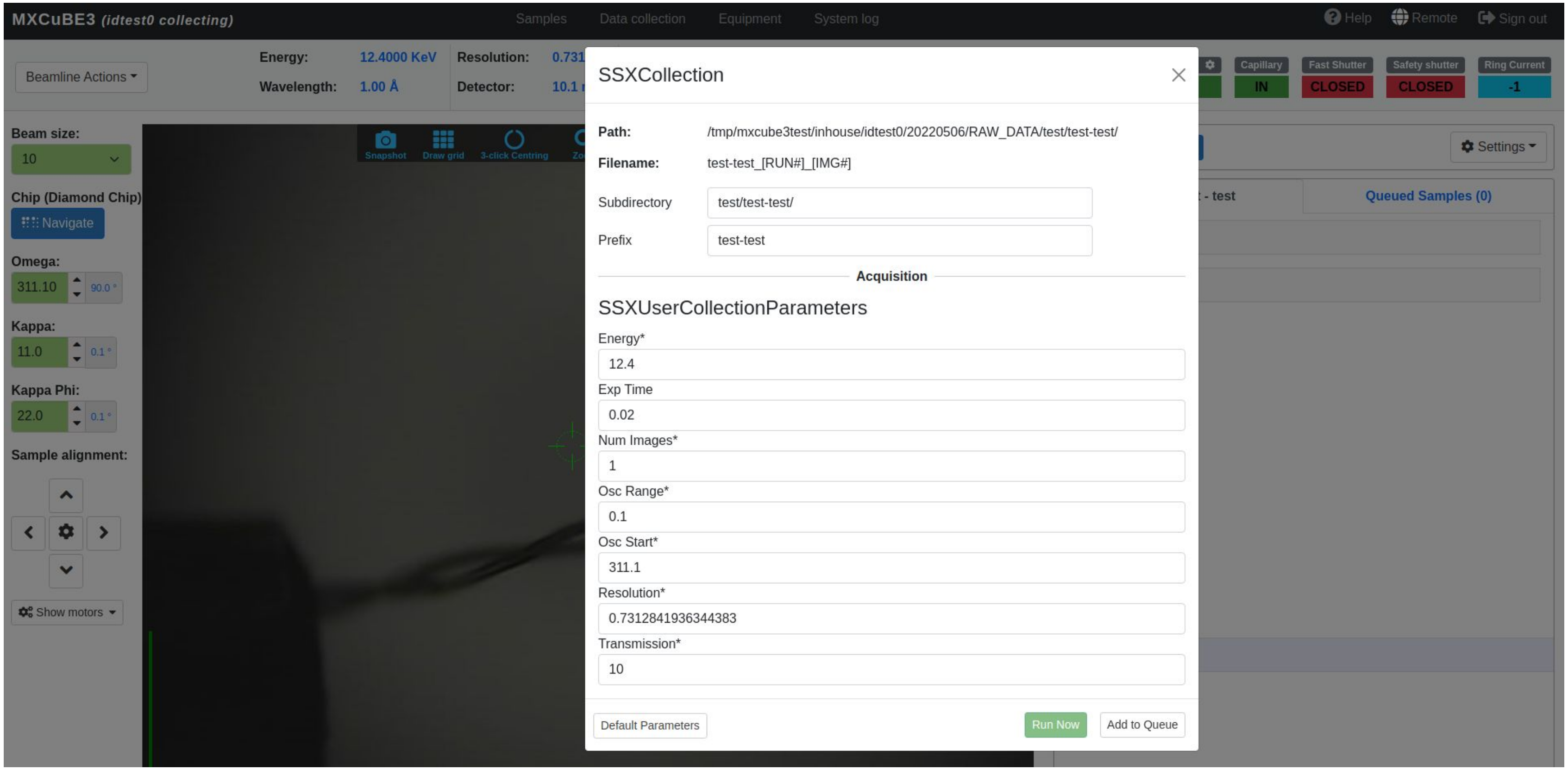


```python
class LegacyParameters(BaseModel):
    take_dark_current: int
#    detector_mode: int
    inverse_beam: bool
    num_passes: int
    overlap: float

    class Config:
        extra: "ignore"

class SsxChipColletionTaskParameters(BaseModel):
    path_parameters: PathParameters
    common_parameters: CommonCollectionParamters
    collection_parameters: SSXCollectionParameters
    user_collection_parameters: SSXUserCollectionParameters
    legacy_parameters: LegacyParameters


class SsxChipCollectionQueueEntry(BaseQueueEntry):
    """
    Defines the behaviour of a data collection.
    """
    DATA_MODEL = SsxChipColletionTaskParameters
    NAME = "SSXCollection"
    REQUIRES = ["point", "line", "no_shape", "chip", "mesh"]

    # New style queue entry does not take view argument,
    # adding kwargs for compatability, but they are unsued
    def __init__(self, data: SsxChipColletionTaskParameters, view=None, **kwargs):
        super().__init__(view=view, data_model=TaskNode(data))
```

Improved Queue Entry

- **Everything needed to make the QueueEntry available is to define the parameters using Pydantic models and add the new entry in the list of available methods in beamline-config.yml**

- **Example default UI (if no bespoke component exists) for example SSXCollection**



MXCuBE3 ***(idtest0 collecting)***

Energy: 12.4000 KeV **Resolution:** 0.731
Wavelength: 1.00 Å **Detector:** 10.1

Beam size: 10

Chip (Diamond Chip)

Omega: 311.10 90.0 °

Kappa: 11.0 0.1 °

Kappa Phi: 22.0 0.1 °

Capillary: IN
Fast Shutter: CLOSED
Safety shutter: CLOSED
Ring Current: -1

SSXCollection

Path: /tmp/mxcube3test/inhouse/idtest0/20220506/RAW_DATA/test/test-test/

Filename: test-test_[RUN#]_[IMG#]

Subdirectory: test/test-test/

Prefix: test-test

Acquisition

SSXUserCollectionParameters

Energy*: 12.4

Exp Time: 0.02

Num Images*: 1

Osc Range*: 0.1

Osc Start*: 311.1

Resolution*: 0.7312841936344383

Transmission*: 10

Braggy

- Braggy diffraction viewer now available to users since January
- New profile tool, lines and circles
- Work being done on packaging

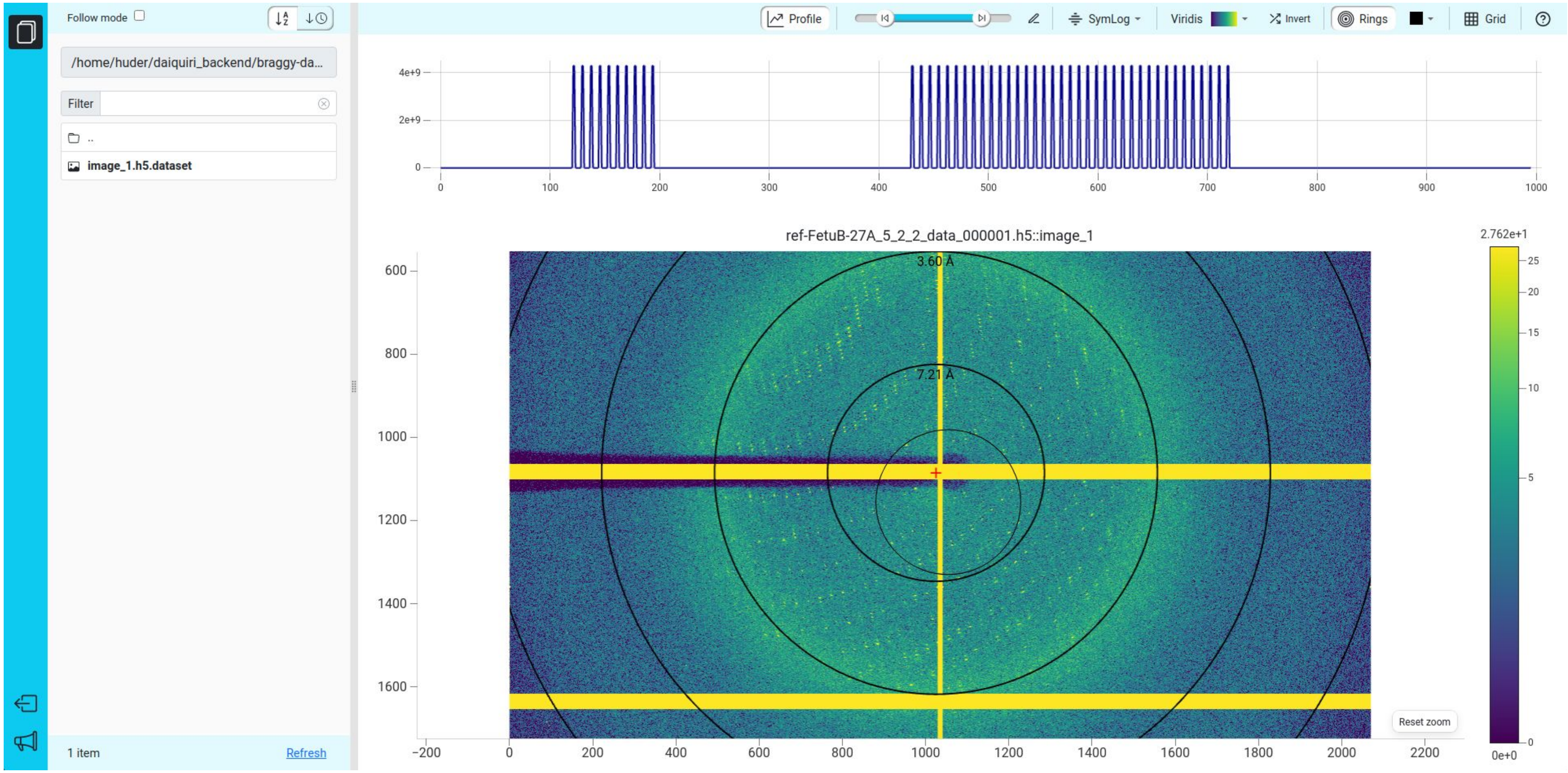

/home/huder/daiquiri_backend/braggy-da...

image_1.h5.dataset

ref-FetuB-27A_5_2_2_data_000001.h5::image_1

1 item

Questions

Thank you for your attention