

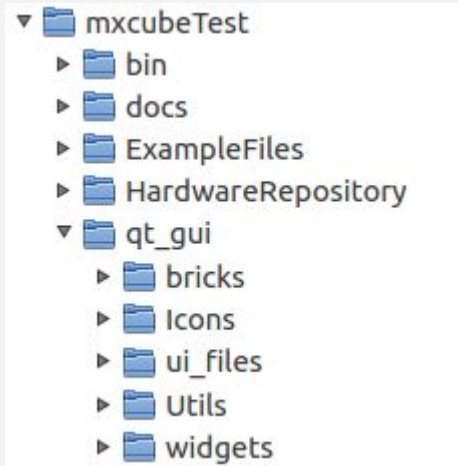
Refactoring Qt version of MXCuBE: structural improvements and new API

Ivars Karpičs (EMBL Hamburg)



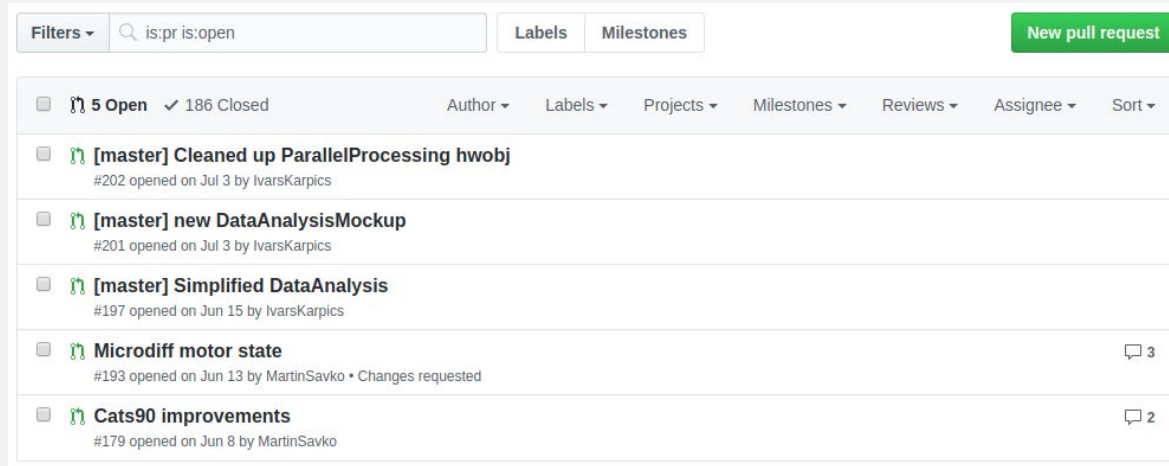
Changes in the MXCuBE project structure

- Introduce new qt_gui subdirectory (could be the submodule).
- web_gui subdirectory also could be as a submodule.
- Top level mxcube directory should not be qt version, but a project with both gui submodules and HardwareRepository.
- Leave Qt3 bricks and widgets for compatibility in the BlissFramework (at some point remove it).








Pull requests

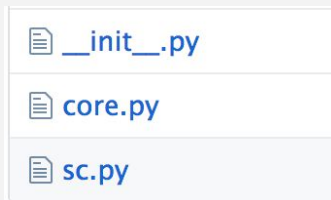
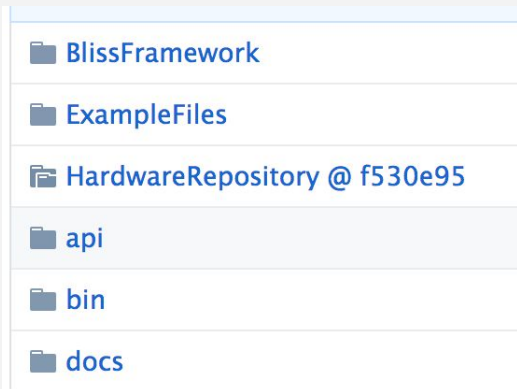
- Not reviewed and accepted pull requests are pending.
- Reviewing of the code is essential.
- Simple guidelines are necessary.



The screenshot shows a GitHub pull request list. At the top, there is a search bar with the text "is:pr is:open", a "Filters" dropdown, and buttons for "Labels" and "Milestones". A green button labeled "New pull request" is in the top right corner. Below the search bar, the list shows 5 open pull requests and 186 closed ones. The pull requests are listed in a table with columns for Author, Labels, Projects, Milestones, Reviews, Assignee, and Sort. The first three pull requests are by IvarsKarpics, and the last two are by MartinSavko.

	Author	Labels	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>  [master] Cleaned up ParallelProcessing hwobj #202 opened on Jul 3 by IvarsKarpics							
<input type="checkbox"/>  [master] new DataAnalysisMockup #201 opened on Jul 3 by IvarsKarpics							
<input type="checkbox"/>  [master] Simplified DataAnalysis #197 opened on Jun 15 by IvarsKarpics							
<input type="checkbox"/>  Microdiff motor state #193 opened on Jun 13 by MartinSavko • Changes requested							3
<input type="checkbox"/>  Cats90 improvements #179 opened on Jun 8 by MartinSavko							2

- First implementation via Qt signals/slots (not capable with web version).
- Second solution based on pydispatcher (almost no feedback).
- A look at api branch.



```
from core import CoreComponent
from core import sample_changer_hwobj
#from sample_changer.GenericSampleChanger import *

class SC(CoreComponent):
    """
    Sample changer core component describes the api between MXCuBE GUI and
    the sample mounting device. Further any device that delivers sample on the
    diffractometer will be called sample changer.
    """

    def __init__(self):
        CoreComponent.__init__(self, "sample_changer")
        sample_changer_hwobj.connect("stateChanged",
                                     self.sc_state_changed_handler)

    # Methods -----

    def mount_sample(self, location, device_name=None, wait=False):
        """
        Mounts sample to the diffractometer. If there is a sample on the
        diffractometer then it is unmounted and requested sample is mounted.
        During the sample mount queue is blocked.
        If the sample mount fails then user is informed with an error message.

        :param LocationStr location: location
        :returns: True if mount successful otherwise False
        :rtype: bool
        """
        if not device_name:
            device = sample_changer_hwobj
        else:
            device = getattr(bl_setup, device_name + "_hwobj")
```

The king is dead, long live the king!

1. Abstract classes is still option to force people use the same core components.
2. Work in progress: AbstractBeamStop, AbstractDoorInterlock, ...

```
AbstractBase.py
AbstractMCA.py
AbstractDataAnalysis.py
AbstractMultiCollect.py
AbstractEnergy.py
AbstractMotor.py
AbstractDetector.py
AbstractAperture.py
AbstractSlits.py
AbstractFlux.py
AbstractAttenuators.py
AbstractEnergyScan.py
AbstractXRFspectrum.py
AbstractCollect.py
```

```
GenericVideoDevice.py
GenericDiffractometer.py
GenericParallelProcessing.py
```

```
LdapLoginMockup.py
TransmissionMockup.py
BeamlineActionsMockup.py
AttenuatorsMockup.py
DoorInterlockMockup.py
CameraMockup.py
BeamstopMockup.py
MDCameraMockup.py
MachInfoMockup.py
MultiCollectMockup.py
XRFspectrumMockup.py
ShapeHistoryMockup.py
MicrodiffInOutMockup.py
ShutterMockup.py
EnergyScanMockup.py
ISPyBRestClientMockup.py
ResolutionMockup.py
DetectorMockup.py
MicrodiffZoomMockup.py
ISPyBClient2Mockup.py
SampleChangerMockup.py
MotorMockup.py
DiffractometerMockup.py
CollectMockup.py
ApertureMockup.py
BeamlineTestMockup.py
BeamInfoMockup.py
FluxMockup.py
EnergyMockup.py
Qt4_VideoMockup.py
MachineInfoMockup.py
SlitsMockup.py
ParallelProcessingMockup.py
```

Thank you for your attention!

Acknowledgments:

- T. Schneider group: G. Bourenkov, T. Schneider, and others.
- Instrumentation group: M. Nikolova, M. Bueno, U. Ristau, and others.
- EMBL Grenoble: J. Sinoir, and others.
- MXCuBE community.