Netherlands Institute for Radio Astronomy

Monitoring of staging activity through a Python API

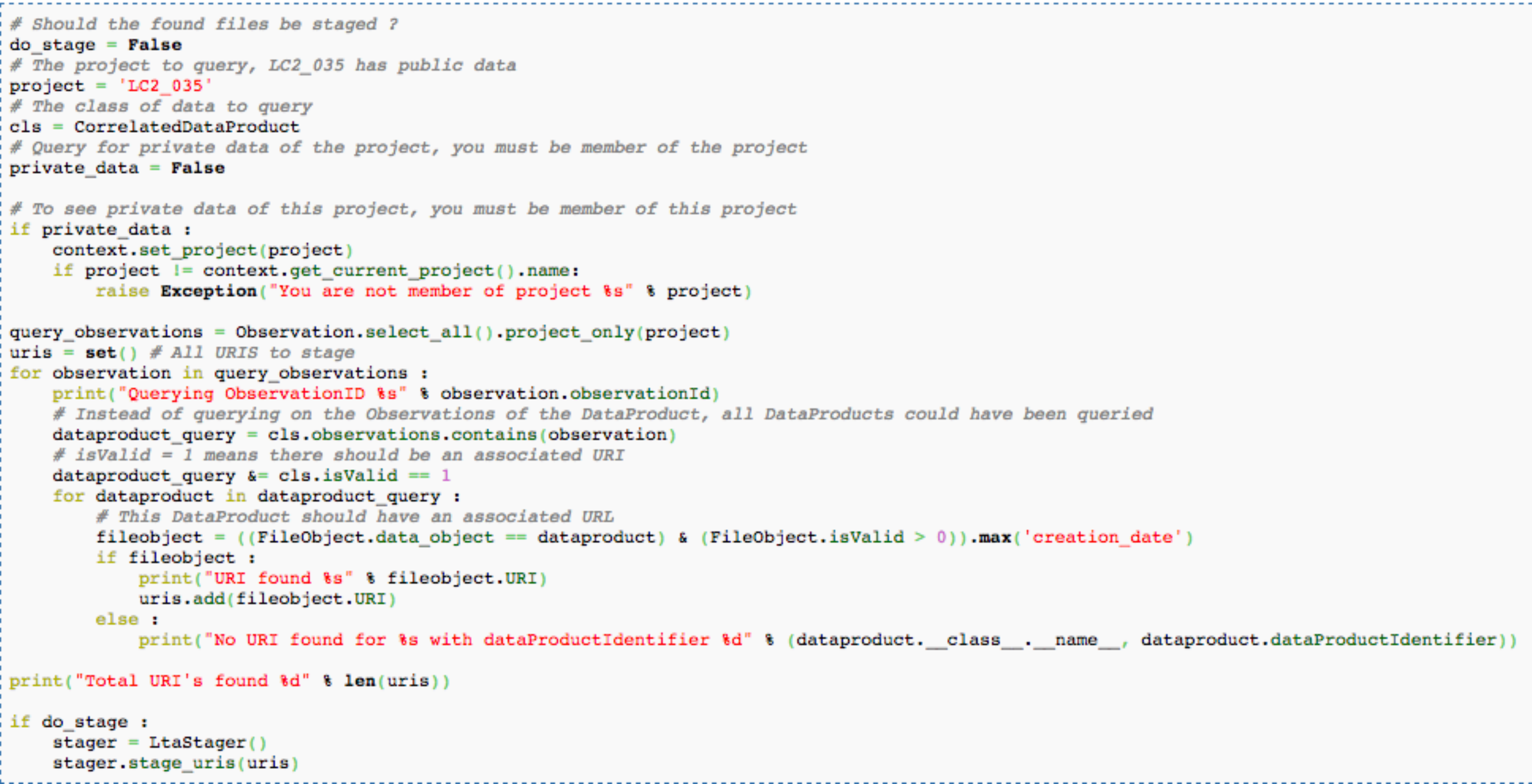
L. Cerrigone

Access the LTA catalog:

- web interface (general users)
- DBview (SQL) (advanced users)
- AstroWise Python interface (advanced users)

Instructions on how to set up the python LTA client are given in the Wiki http://www.lofar.org/operations/doku.php?id=public:lta_tricks#astrowise_python_interface

One of the examples in the Wiki



```
# Should the found files be staged ?
do_stage = False
# The project to query, LC2_035 has public data
project = 'LC2_035'
# The class of data to query
cls = CorrelatedDataProduct
# Query for private data of the project, you must be member of the project
private_data = False

# To see private data of this project, you must be member of this project
if private_data :
    context.set_project(project)
    if project != context.get_current_project().name:
        raise Exception("You are not member of project %s" % project)

query_observations = Observation.select_all().project_only(project)
uris = set() # All URIS to stage
for observation in query_observations :
    print("Querying ObservationID %s" % observation.observationId)
    # Instead of querying on the Observations of the DataProduct, all DataProducts could have been queried
    dataproduct_query = cls.observations.contains(observation)
    # isValid = 1 means there should be an associated URI
    dataproduct_query &= cls.isValid == 1
    for dataproduct in dataproduct_query :
        # This DataProduct should have an associated URL
        fileobject = ((FileObject.data_object == dataproduct) & (FileObject.isValid > 0)).max('creation_date')
        if fileobject :
            print("URI found %s" % fileobject.URI)
            uris.add(fileobject.URI)
        else :
            print("No URI found for %s with dataProductIdentifier %d" % (dataproduct.__class__.__name__, dataproduct.dataProductIdentifier))

print("Total URI's found %d" % len(uris))

if do_stage :
    stager = LtaStager()
    stager.stage_uris(uris)
```

Thanks to Joern Kuensmoeller, we now have a python module that allows users to check the status of their staging requests. The module can be used along with the python scripts to browse the catalog or just on any staging request put in through the web interface.

stage(*surls*)

It takes in a list of surls, queue a staging request for those urls, and outputs the ID of the request

get_status(*stageid*)

It tells the user if a request is queued, in progress or finished (success)

abort(*stageid*)

It allows users to end a staging request

get_surls_online(*stageid*)

It gives a list of the surls that have been staged for the relative request. The list is updated whenever a new surl comes on line.

get_srm_token(*stageid*)
The srm token is useful to interact directly with the SRM site through GRID/SRM tools.

reschedule(*stageid*)
If a request failed, it can be rescheduled.

get_progress()
No input needed. It returns the statuses of all the requests owned by the user. Users with special privileges will see all requests.

The module will be made available on the Wiki, in the section about the AstroWise Python interface