

PyICAT

An ICAT WebService client module

12. March 2013 | Christian Felder
JCNS | Scientific Computing

PyICAT

Part I: Motivation

12. March 2013 | Christian Felder
JCNS | Scientific Computing

PyICAT

Table of contents

- Part I: Motivation
- Part II: Application Interface and programs
- Part III: Installation and Summary

12. March 2013

2 | 18

Motivation

- ICAT web service description language file
 - create, createMany
 - search, get
 - update
 - delete, deleteMany
 - + types, session management, version info
- “To understand exactly how the data manipulation calls work requires an understanding of the schema.” [icatproject.org]
 - “rather thin layer on top of a relational DBMS” [icatproject.org]
 - generic interface
 - knowledge about object relations required

12. March 2013

4 | 18

Motivation

Goal

layer on top of ICAT's WSDL

- data manipulation calls for each type (methods)
- calls should include all relations (arguments)

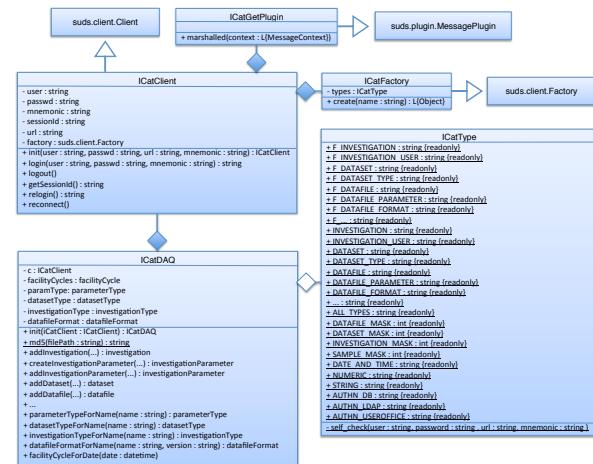
benefits

- reduces knowledge requirements about the schema.
- higher code density
 - less errors due to wrong/missing attributes and/or relationships

12. March 2013

5 | 18

Application Interface pyICAT module



12. March 2013

7 | 18

PyICAT

Part II: Application Interface and programs

12. March 2013 | Christian Felder

JCNS | Scientific Computing

Application Interface Simple example (I/II)

examples/SimpleExample.py

```

1 #!/usr/bin/env python
# standard library
import datetime
import time
# local library
6 from pylICAT import ICatDAQ, ICatClient, ICatType
c = ICatClient("root", "secret", "https://apps.jcns.fz-juelich.de:5443/" +
               "ICATService/ICAT?wsdl", "db")
daq = ICatDAQ(c)
11 facility = c.service.search(c.getSessionId(), ICatType.FACILITY +
                               "[name='JCNS'])[0]
instrument = c.service.search(c.getSessionId(), ICatType.INSTRUMENT +
                               "[name='KWS-1'])[0]

```

12. March 2013

8 | 18

Application Interface

Simple example (II/II)

```

cycle = daq.facilityCycleForDate(datetime.datetime.today())
investigationId = "p" + str(time.time())[1:10]
inv = daq.addInvestigation(facility , instrument , cycle , "visit0",
                            investigationId ,
                            daq.investigationTypeForName("simulation"),
                            "Example_Investigation",
                            datetime.datetime(2013, 3, 4),
                            datetime.datetime(2013, 3, 15))
sample = daq.addSample(inv , "Empty_Cell")
dataset = daq.addDataSet(inv , sample , "Example_Dataset",
                         daq.datasetTypeForName("raw") , False ,
                         "example_dataset.description",
                         "/tmp/example.dataset")
file = daq.addDatafile(dataset , "/tmp/example.dataset/zero.gz",
                       datetime.datetime.today(),
                       "jcnsfns.jcns.frm2:/path/to/destination/zero.gz",
                       daq.datafileFormatForName("GZIP"))
daq.addDatafileParameter(12 , daq.parameterTypeForName("wavelength") , file)
daq.updateDatasetStatus(dataset , True)
c.logout()

```

12. March 2013

9 | 18

Programs

Overview

- **icatconfig**
 - default values for icatdaq
 - plugin class mapping (instrument → plugin)
 - store login credentials
- **icatdaq**
 - main **metadata acquisition program**
 - list available facilities and instruments
 - **import facility cycles from csv file**
 - integrated **logging support** (-d, --debug)
 - integrated **cProfile support**
- **icattest**
 - connection/login test for a given **ICAT WebService**
 - basic tests, e.g. remote type checks

12. March 2013

10 | 18

Programs

icatdaq - list available facilities and instruments

```

user@localhost:~% icatdaq -s apps.jcns.fz-juelich.de:5443 \
-u db/icat42 -p secret -l
#login as: db/icat42
#available instruments

JCNS      Juelich Centre for Neutron Science
-----
BIODIFF   Single crystal DIFFractometer for BIOlogical macromolecules
DNS       Diffuse Neutron Scattering
J-NSE     Juelich Neutron Spin Echo Spectrometer
KWS-1     Kleinwinkelstreuung I
KWS-2     Kleinwinkelstreuung II
KWS-3     Kleinwinkelstreuung III
MARIA     MAgnetic Reflectometer with Incident Angle
SPHERES   SPectrometer for High Energy RESolution

TUM       Technische Universitaet Muenchen
-----
SANS-I    Small Angle Neutron Scattering I

```

12. March 2013

11 | 18

Programs

icatdaq - metadata acquisition using a plugin

```

user@localhost:~ % icatdaq -s apps.jcns.fz-juelich.de:5443 \
-u db/icat42 -p secret -f JCNS -i maria \
/tmp/example_dataset
#login as: db/icat42
#use plugin: pyICAT.plugins.pyfrid.MARIADAQ
/tmp/example_dataset/test_0_0.dev
/tmp/example_dataset/test_0_0.gz
/tmp/example_dataset/test_0.set
/tmp/example_dataset/test_1_0.dev
/tmp/example_dataset/test_1_0.gz
...

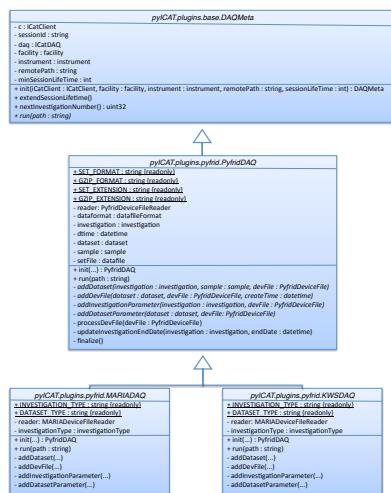
```

12. March 2013

12 | 18

Application Interface

pyICAT.plugins module



12. March 2013

13 | 18



Configuration program icatconfig

```
user@localhost:~ % icatconfig -c
ICAT configuration file [/home/user/.pyICAT/icat.conf]:
default hostname [None]: apps.jcns.fz-juelich.de
default port [None]: 5443
default facility [None]: JCNS
default mnemonic [None]: ldap
remote base url [None]: jcnsnfs.jcns.frm2:/Archive/project/instruments/FRM2
[plugin mnemonic/]username password (press ENTER to skip): db/icat42 secret
...
[plugin mnemonic/]username password (press ENTER to skip):
instrument plugins.module.class (press ENTER to skip): \
kws-3 plugins.pyfrid.KWSDAQ
...
instrument plugins.module.class (press ENTER to skip):
```

12. March 2013

15 | 18

Configuration file icat.conf

```
[icatuser]
db/icat42 = secret

[defaults]
remote_base_url = jcnsnfs.jcns.frm2:/Archive/project/instruments/FRM2
port = 5443
hostname = apps.jcns.fz-juelich.de
facility = JCNS
mnemonic = ldap

[plugins]
kws-3 = plugins.pyfrid.KWSDAQ
kws-2 = plugins.kws.KWSDAQ
kws-1 = plugins.kws.KWSDAQ
maria = plugins.pyfrid.MARIADAQ
```

12. March 2013

14 | 18



PyICAT

Part III: Installation and Summary

12. March 2013 | Christian Felder

JCNS | Scientific Computing

Installation and Summary

current release: v1.6.0

- Prerequisites and third party modules
 - python, version ≥ 2.6 (version 3.0 not yet tested)
 - SUDS (tested with version 0.4.1)
- Installation
 - python distutils package

```
python setup.py install [--prefix=$PREFIX]
```
- Plugins
 - inherit from pyICAT.plugins.base.DAQMeta
 - store plugin in directory named plugins
 - create plugins/_init__.py

```
__all__ = ["mypluginmodule"]
```
- configure plugin class mapping (instrument → plugin)
`icatconfig`

12. March 2013

17 | 18

Thanks for your attention

Christian Felder

Jülich Centre for Neutron Science
Scientific Computing

c.felder@fz-juelich.de

<http://apps.jcns.fz-juelich.de/doku/sc/pyicat>

12. March 2013

18 | 18