python-icat

A Library for Writing ICAT Clients in Python

Rolf Krahl

ICAT Meeting, Diamond Light Source, March 2015

Introduction

python-icat is a Python library for writing ICAT clients build on top of Suds.

Design Goals

- Keep the general structure and flexibility of Suds.
- Simplify things where possible.
- Try to remove annoying details.
- Add useful client side features.
- Make use object oriented design.

A typical python-icat program might be mistaken for a generic Suds program at first glance. It's just somewhat simpler.

Example: Add a Datafile

Using plain Suds

```
dataset = client.service.search(sessionId,
              "Dataset[name='e201215']")[0]
format = client.service.search(sessionId,
              "DatafileFormat[name='NeXus']")[0]
datafile = client.factory.create("datafile")
datafile.dataset = dataset
datafile.datafileFormat = format
datafile.name = "e201215-7.nxs"
datafile.id = client.service.create(sessionId, datafile)
```

Example: Add a Datafile

Using plain Suds

```
dataset = client.service.search(sessionId,
              "Dataset[name='e201215']")[0]
format = client.service.search(sessionId,
              "DatafileFormat[name='NeXus']")[0]
datafile = client.factory.create("datafile")
datafile.dataset = dataset
datafile.datafileFormat = format
datafile.name = "e201215-7.nxs"
datafile.id = client.service.create(sessionId, datafile)
```

Using python-icat

```
dataset = client.search("Dataset[name='e201215']")[0]
format = client.search("DatafileFormat[name='NeXus']")[0]
datafile = client.new("datafile")
datafile.dataset = dataset
datafile.datafileFormat = format
datafile.name = "e201215-7.nxs"
datafile.create()
```

Example: Config and Login

Using plain Suds

```
url = "https://" + sys.argv[1] + ":" + sys.argv[2] \
    + "/ICATService/ICAT?wsdl"
auth = sys.argv[3]
username = sys.argv[5]
password = sys.argv[7]
client = suds.client.Client(url)
credentials = client.factory.create("credentials")
credentials.entry.append(
        [ { 'key':'username', 'value':username },
          { 'key':'password', 'value':password } ])
sessionId = client.service.login(auth, credentials)
```

Example: Config and Login

Using plain Suds

```
url = "https://" + sys.argv[1] + ":" + sys.argv[2] \
    + "/ICATService/ICAT?wsdl"
auth = sys.argv[3]
username = sys.argv[5]
password = sys.argv[7]
client = suds.client.Client(url)
credentials = client.factory.create("credentials")
credentials.entry.append(
        [ { 'key':'username', 'value':username },
          { 'key':'password', 'value':password } ])
sessionId = client.service.login(auth, credentials)
```

Using python-icat

```
config = icat.config.Config()
conf = config.getconfig()
client = icat.Client(conf.url, **conf.client_kwargs)
client.login(conf.auth, conf.credentials)
```

Config

Default Command Line Arguments

```
usage: login-icat-config.py [options]

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIGFILE, --configfile CONFIGFILE
                        config file
  -s SECTION, --configsection SECTION
                        section in the config file
  -w URL, --url URL     URL to the web service description
  --http-proxy HTTP_PROXY
                        proxy to use for http requests
  --https-proxy HTTPS_PROXY
                        proxy to use for https requests
  --no-proxy NO_PROXY   list of exclusions for proxy use
  -a AUTH, --auth AUTH  authentication plugin
  -u USERNAME, --user USERNAME
                        username
  -p PASSWORD, --pass PASSWORD
                        password
  -P, --prompt-pass     prompt for the password
```

Misc Features

- Integrates an IDS client.
- Dumping and restoring ICAT content to and from a file.
- Hide some incompatibilities between ICAT versions from client programs.
- Improved support for ICAT server exceptions.

Recent changes

Version 0.6.0 (December 2014)

- Add support for ICAT 4.4.0 and IDS 1.3.0.
- Rework the dump file backend API for icatdump and icatrestore.
- Add client method `searchChunked()`.
- Add `no_proxy` configuration support.

Version 0.7.0 (February 2015)

- Add a module icat.query with a class Query that can be used to build ICAT search expressions.

Upcoming

- Add another derived configuration variable `configDir`.
- Values for configuration variables may refer to other configuration variables.

The Query class

Static Example

```
query = Query(client, "Datafile", conditions={
    "name":"= '%s'" % dfname,
    "dataset.name":"= '%s'" % dsname,
    "dataset.investigation.name":"= '%s'" % invname,
}, includes={"datafileFormat", "parameters.type"})
datafile = client.assertedSearch(query)[0]
```

The Query class

Static Example

```
query = Query(client, "Datafile", conditions={
    "name":"= '%s'" % dfname,
    "dataset.name":"= '%s'" % dsname,
    "dataset.investigation.name":"= '%s'" % invname,
}, includes={"datafileFormat", "parameters.type"})
datafile = client.assertedSearch(query)[0]
```

Dynamic Example

```
query = Query(client, "Datafile", order=True)
if startPeriod:
    query.addConditions({"dataset.investigation.startDate":
                         ">= '%s'" % startPeriod})
if endPeriod:
    query.addConditions({"dataset.investigation.startDate":
                         "< '%s'" % endPeriod})
datafiles = client.search(query)
```

System Requirements and Download

System Requirements

- Python 2.6, 2.7, or 3.1 and newer (Python 2.6 requires a patch).
- Suds, either 0.4 or jurko fork, the latter is recommended.
- argparse (in system library in Python 2.7 and 3.2).
- PyYAML only for YAML backend of icatdump and for example scripts.
- lxml only for XML backend of icatdump.

Download

- python-icat 0.7.0 available at `http://code.google.com/p/icatproject/wiki/PythonIcat`
- BSD license.

System Requirements and Download

System Requirements

- Python 2.6, 2.7, or 3.1 and newer (Python 2.6 requires a patch).
- Suds, either 0.4 or jurko fork, the latter is recommended.
- argparse (in system library in Python 2.7 and 3.2).
- PyYAML only for YAML backend of icatdump and for example scripts.
- lxml only for XML backend of icatdump.

Download

- python-icat 0.7.0 available at `http://code.google.com/p/icatproject/wiki/PythonIcat`
- BSD license.

Thank you for your attention! Questions?

Substituting Configuration Variables

Use Case: Custom Configuration File

```
config = icat.config.Config()
config.add_variable('datafileFormats', ("--datafileformats",),
                    dict(help="another configuration file"),
                    default="%(configDir)s/fileformats.xml",
                    subst=True)
conf = config.getconfig()
client = icat.Client(conf.url, **conf.client_kwargs)
client.login(conf.auth, conf.credentials)

with open(conf.datafileFormats, 'r') as f:
    # ...
```