PyXNAT 101 basic usage of PyXNAT to access an XNAT server.

Creating a connection to a pyxnat server with https:

In [ ]:
```
import pyxnat
xnat = pyxnat.Interface('https://central.xnat.org', user='nosetest
s', password='nose2016')
```

Getting an xnat subject object

There are two methods of retriving a handle to a pyxnat object. The first method uses pyxnat's select methods to walk down the path to the subject.
The second method uses a deterministic URL to directly retrieve the object.

In [4]:
```
# Using pyxnat's object methods to walk down the path.
subject_pyxnat2016 = xnat.select.project('nosetests').subject('pyxn
at2016')
```

In [6]:
```
# Using deterministic URLS.
subject_pyxnat2016 = xnat.select('/projects/nosetests/subjects/pyxn
at2016')
#The URL format for select is "/projects/<projectID or Label>/subje
cts/<subjectID or Label>
```

The deterministic URLs can go as shallow or as deep as you need:

- /projects
  - give a list of all projects
- /projects/project_label
  - get handle to project: project_label
- /projects/project_label/subjects
  - get a list of all subjects in project_label
- /projects/project_label/subjects/subject_label
- /projects/project_label/subjects/subject_label/experiments
- /projects/project_label/subjects/subject_label/experiments/experiment_label
- /projects/project_label/subjects/subject_label/experiments/experiment_label/scans
- /projects/project_label/subjects/subject_label/experiments/experiment_label/scans/scan_label
- /projects/project_label/subjects/subject_label/experiments/experiment_label/scans/scan_label/resour
- /projects/project_label/subjects/subject_label/experiments/experiment_label/scans/scan_label/resour

At each level, you can add "/resources/resource_label" to get access to that level's resources. Only the scan level resources are shown in the above example.

Creating an experiment (xnat:mrSessionData)

In [22]:
```
experiment = subject_pyxnat2016.experiment('pyxnat101')
print "Before create: ", experiment.exists()
#adding experiments='xnat:datatype' allows you to specify the objec
t type during creation.
#for mrSessionData this could be left blank as that's the default,
I put it in for demostration purposes.
experiment.create(experiments='xnat:mrSessionData')
print "After create: ", experiment.exists()
```

```
Before create:  False
After create:  True
```

Setting Attributes on the experiment

PyXNAT is not the best at learning or knowing what attributes are avaiable to set. This is one of the dirty areas of pyxnat. The best way to know what fields are avaiable to set in a given object is to look at the xnat.xsd (or related *.xsd for the object.)

The xnat.xsd can be viewed by doing xnat.get('https://central.xnat.org/schemas/xnat/xnat.xsd (https://central.xnat.org/schemas/xnat/xnat.xsd)')

In [14]:
```
experiment.attrs.mset({
        'xnat:mrSessionData/fieldstrength': '3.0',
        'xnat:mrSessionData/coil': 'head',
        'xnat:mrSessionData/marker': "right"
    })
```

Adding a scan to the experiment.

In [23]:
```
scan = experiment.scan('ScanOne')
print "Before create: ", scan.exists()
scan.create(scans='xnat:mrScanData')
print "After create: ", scan.exists()
```

```
Before create:  False
After create:  True
```

Setting Attributes on the Scan

In [24]:
```
scan.attrs.mset({
        'xnat:mrScanData/parameters/imageType' : 'myType',
        'xnat:mrScanData/series_description': 'The description goes
here',
        'xnat:mrScanData/quality' : 'good'

    })
```

Getting Attributes from the Scan

In [40]:
```
scan.attrs.mget({
        'xnat:mrScanData/ID',
        'xnat:mrScanData/quality'
    })
```

Out[40]:
```
['ScanOne', 'good']
```

Adding DICOM files to a scan

In [26]:
```
dicom_resource = scan.resource('DICOM')
dicom_resource.put_dir('/Users/m085077/XNAT-2016-conference/data/IG
T_GLIOMA/00001/scans/1/DICOM')
```

In [45]:
```
the_files = dicom_resource.files()
for f in the_files:
    print f
```

```
<File Object> 000001.IMA
<File Object> 000002.IMA
<File Object> 000003.IMA
<File Object> 000004.IMA
<File Object> 000005.IMA
<File Object> 000006.IMA
<File Object> 000007.IMA
<File Object> 000008.IMA
<File Object> 000009.IMA
<File Object> 000010.IMA
<File Object> 000011.IMA
<File Object> 000012.IMA
<File Object> 000013.IMA
<File Object> 000014.IMA
<File Object> 000015.IMA
<File Object> 000016.IMA
<File Object> 000017.IMA
<File Object> 000018.IMA
<File Object> 000019.IMA
<File Object> 000020.IMA
<File Object> 000021.IMA
<File Object> 000022.IMA
<File Object> 000023.IMA
<File Object> 000024.IMA
```

In [27]:
```
#Triggering the auto-run pipeline (works but does not do anything f
or the example)
xnat.put('/data/projects/nosetests/subjects/pyxnat2016/experiments/
pyxnat101?triggerPipelines=true')
```

Out[27]: <Response [200]>

Adding NIFTI files to the scan.

In [31]:
```
nifti_resource = scan.resource('NIFIT')
file1 = nifti_resource.file('pyxnat2016_pyxnat101_1_1.nii')
file1.insert('/Users/m085077/XNAT-2016-conference/data/IGT_GLIOMA/0
0001/scans/1/NIFTI/20050111_092149s001a1001.nii')
```

Downloading A file

In [46]:
```
file1.get_copy('/Users/m085077/Downloads/pyxnat2016_pyxnat101_1_1.nii')
```

Out[46]: '/Users/m085077/Downloads/pyxnat2016_pyxnat101_1_1.nii'

Downloading a whole resource

In [49]:
```
dicom_resource.get('/Users/m085077/Downloads/dicom_resource/')
```

Out[49]: '/Users/m085077/Downloads/dicom_resource/DICOM.zip'