

# Upload Image Session Files with REST API

If I have a dataset I would like to upload into XNAT, and I've properly organized the data on the file system, then uploading the data is very easy.

For our hypothetical situation, I'm uploading a collection of data, each including a T1 and T2 raw DICOM data as well as NIFTI reconstructions of that data. I have it organized by project/subject/session.

```
/data/  
/data/subject1/  
/data/subject1/session1/  
/data/subject1/session1/RAW/  
/data/subject1/session1/RAW/SCAN1/1232132.dcm  
/data/subject1/session1/RAW/SCAN2/1232133.dcm  
/data/subject1/session1/RECON/T1_0343/0343.nifti  
/data/subject1/session1/RECON/T1_0344/0344.nifti
```

I can build a simple script which will translate this directory into a set of REST commands.

```

//if you configure the .xnatPass file in your home directory, you can skip the inclusion of the host, username,
& password
setenv rest_params "--host http://central.xnat.org -u USERNAME -p PASSWORD "

//create subject1
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1"

//create session1
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1?xnat:
mrSessionData/date=01/02/07"

//create SCAN1
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1/scans
/SCAN1?xsiType=xnat:mrScanData&xnat:mrScanData/type=T1"

//upload SCAN1 files...
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1/scans
/SCAN1/resources/DICOM/resources/DICOM?format=DICOM&content=T1_RAW"

//upload SCAN1 files...
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1/scans
/SCAN1/resources/DICOM/files/1232132.dcm" -local /data/subject1/session1/RAW/SCAN1/1232132.dcm

//create SCAN2
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1/scans
/SCAN2?xsiType=xnat:mrScanData&xnat:mrScanData/type=T2"

//create resource collection to put files into
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1/scans
/SCAN2/resources/DICOM?format=DICOM&content=T2_RAW"

//upload SCAN2 files...
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1/scans
/SCAN2/resources/DICOM/files/1232133.dcm" -local /data/subject1/session1/RAW/SCAN2/1232133.dcm

//create reconstruction 1
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1
/reconstructions/session1_recon_0343?xnat:reconstructedImageData/type=T1_RECON"

//create resource collection to put files into
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1
/reconstructions/session1_recon_0343/resources/NIFTI?format=NIFTI"

//upload reconstruction 1 files...
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1
/reconstructions/session1_recon_0343/resources/NIFTI/files/0343.nfti" -local /data/subject1/session1/RECON
/T1_0343/0343.nfti

//create reconstruction 2
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1
/reconstructions/session1_recon_0344?xnat:reconstructedImageData/type=T2_RECON"

//create resource collection to put files into
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1
/reconstructions/session1_recon_0344/resources/NIFTI?format=NIFTI"

//upload reconstruction 2 files...
XNATRestClient $rest_params -m PUT -remote "/data/projects/YOURTEST/subjects/subject1/experiments/session1
/reconstructions/session1_recon_0344/resources/NIFTI/files/0344.nfti" -local /data/subject1/session1/RECON
/T1_0344/0344.nfti

```

When uploading images, it is good form to define the format of the images (DICOM, ANALYZE, etc) and the content type of the data.

Related: [Tips for Uploading Files via REST](#)