

Tips for Uploading Files via REST

Depending on the tool you are using to PUT files to the server, you may need to add an `inbody=true` query string parameter to force XNAT to treat the contents of your HTTP message as the actual file rather than as multi-part form data.

Tools

XnatDataClient

The XnatDataClient is built specifically to facilitate user interaction with the REST API from the command prompt. Uploading files with this tool should be straightforward. Use the `-l` (or `--local`) flag to give the path to the local file you are uploading.

```
XnatDataClient -u USERNAME -p PASSWORD -m PUT -r "http://central.xnat.org/data/PATH/TO/DESTINATION/files" -l filename.txt
```

Curl

Most UNIX operating systems come pre-installed with 'Curl' which is a tool for transferring data to or from a server. Because XNAT Servers are secured, you will need to provide a username and password to access the api. If you use the `-F` tag, the file will be sent to the server as part of a multipart-form message (similar to a browser post). Without the `-F` tag, curl would post the file as the body of the message and you would need to set the mime-type of your message for XNAT to process it correctly (or use the `?inbody=true` parameter). For convenience, the `-F` works effectively.

```
curl -u USERNAME:PASSWORD -X PUT "http://central.xnat.org/data/PATH/TO/DESTINATION" -F "file=@filename.txt"
```

For more information on using curl, please review its documentation (<https://curl.haxx.se/docs/tooldocs.html>).

HTTPIe

This tool was written in python (using the requests library) to mimic the syntax of curl while being better "unix citizen" than curl in how it reads stdin and writes stdout.

```
http -a USERNAME:PASSWORD --form PUT http://central.xnat.org/data/PATH/TO/DESTINATION/files file=@filename.txt
```

See more about HTTPIe at [its home](#) or by reading [its documentation](#).

Compressed file upload

The REST API supports the upload of compressed files that can be extracted on the server after successful upload. There are a few caveats:

- The files cannot cross hierarchical concepts. I.E. You cannot include files for both scans and reconstructions in the same zip file.
- The files cannot cross objects. I.E. You cannot include files for scan A and scan B. They must be within a single scan.
- The files cannot cross catalogs. Within each object you can have multiple catalogs (collections) of files. The files in a zip can only go in a single catalog.

To use compressed uploads, you would use a similar structure to uploading a normal file. Any where that you can upload files via the REST API, you can use the `?extract` option.

```
curl -u USERNAME:PASSWORD -X POST "http://central.xnat.org/data/archive/projects/X/subjects/X/experiments/X/scans/X/files?extract=true" -F "file.zip=@file.zip"
```

If you are including the zipped file as the body of the message (curl's `-d` flag), you should also use the `?inbody=true` attribute. Otherwise XNAT will expect MULTI-PART form data (curl's `-F` flag).

The file name must include the proper file extension. `.zip` for zipped data. `.gz` for a gzipped file. `.tar.gz` for a tar gz. This will be used to properly extract the file.

If the `extract=true` is missing, then the file will be uploaded and stored as is (not extracted).