Datalad XNAT

Release 0.2.1+0.gc332eea.dirty

DataLad team

Sep 25, 2023

CONTENTS

This is documentation for the DataLad extension, `datalad-xnat`, that equips DataLad with additional functionality to work with XNAT servers. Use it to [COMPLETE ME]

The extension was created during the Juelich Brain Hackathon 2021 and wouldn't have been possible without a dedicated team of volunteers. If you want to get in touch or on board as well, please see our *contributing guidelines*.

CHAPTER

ONE

DOCUMENTATION OVERVIEW

1.1 Introduction

1.1.1 XNAT

XNAT is an open source platform purpose-built for imaging data and data associated with it. In addition to hosting and cataloging the data, XNAT can assist with triggering quality assurance tasks and other workflows.

1.1.2 Goal of the extension

[COMPLETE ME] What did we set out to do?

1.1.3 What can I use this extension for?

[COMPLETE ME] Usecases for the extentions

1.1.4 What can I not use this extension for?

The list of valid use cases for this extension is much shorter than the list of invalid use cases. You for example will not be able to open a bottle of your favourite beverage with it (sorry). But here is a list of invalid use cases that may be most relevant to know about:

- You can not use this extension as a *replacement* for an XNAT instance. It requires access to one, and tracks the data available on this instance.
- You *should* not use this extension to share data that you retrieved from an XNAT server publicly. While DataLad datasets are ideal for sharing large amounts of data publicly, the data from an XNAT server is usually not to be shared publicly - for example due to, but not limited to, privacy concerns. Please think twice before you attempt to do this.

1.2 Quickstart

1.2.1 Requirements

DataLad and `datalad-xnat` are available for all major operating systems (Linux, MacOS, Windows 10[1]). The relevant requirements are listed below.

An XNAT account with appropriate permissions
: You need access to an XNAT server to able to interact with it, and appropriate permissions to access the projects you are interested in. Keep your XNAT instance's URL and your user name and password to your account close by.

DataLad
: If you don't have DataLad and its underlying tools (git, git-annex) installed yet, please follow the instructions from the datalad handbook.

1.2.2 Installation

`datalad-xnat` is a Python package available on pypi and installable via pip.

```
# create and enter a new virtual environment (optional)
$ virtualenv --python=python3 ~/env/dl-xnat
$ . ~/env/dl-xnat/bin/activate
# install from PyPi
$ pip install datalad-xnat
```

1.2.3 Getting started

Here's the gist of some of this extension's functionality. Checkout the *Tutorial* for more detailed demonstrations.

Start by creating and initializing a new DataLad dataset to track a specific XNAT project. This example uses the XNAT central instance with anonymous credentials for the project DCMPHANTOM.

```
$ datalad create dcm_phantom
$ cd dcm_phantom
$ datalad xnat-init https://central.xnat.org --credential anonymous --project DCMPHANTOM
```

After initialization, run `xnat-update` to download all files for the project.

```
$ datalad xnat-update --credential anonymous --subject CENTRAL_S01742
```

HELP! I'm new to this!

If you are confused about the words DataLad dataset, please head over to the DataLad Handbook for an introduction to DataLad.

If you are confused about the words project, experiment, or session in the context of XNAT, take a look at the *Glossary* or in the XNAT documentation.

---

[1] While installable for Windows 10, the extension may not be able to perform all functionality documented here. Please get in touch if you are familiar with Windows to help us fix bugs.

1.3 Tutorial

1.3.1 Authentication

The authentication process

Typical interactions with an XNAT instance require a user name and a password. When you initialize a project using `datalad xnat-init` for a given XNAT URL you will thus be prompted to supply those credentials in the command line:

```
$ datalad xnat-init <myxnatinstance>
  You need to authenticate with '<myxnatinstance>' credentials. <myxnatinstance>/app/
↪template/Register.vm provides information on how to gain access
  user: <myusername>
  password: <mypassword>
  password (repeat): <mypassword>
```

Afterwards, these credentials are stored internally in your systems keyring under the credential name `datalad-<myxnatinstance>`, and subsequent interactions to this XNAT instance will authenticate automatically using the stored credentials.

Multiple different credentials

If you have multiple XNAT instances with different user names or passwords you want to authenticate against, the name of the credential should automatically authenticate you with the correct user password combination based on the XNAT URL. If you nevertheless want to enfore a specific credential to be used, you can supply the `--credential <name>` parameter to `xnat-init`. If `<name>` matches an existing credential in your keyring, the given credential will be used for authentication. If `<name>` does not match an existing credential, you will be prompted for user name and password, and the supplied credentials will be saved under the `<name>` you specified.

Authenticating as anonymous

Typical interactions with an XNAT instance require a user name and a password. Some XNAT instances, however, allow anonymous access, such as XNAT central. In order to authenticate as an anonymous user, supply the special value `anonymous` to the `--credential` parameter.

```
$ datalad xnat-init --credential anonymous https://central.xnat.org
  [INFO   ] Querying https://central.xnat.org for projects available to user anonymous
  No project name specified. The following projects are available on https://central.
↪xnat.org for user anonymous:
  [...]
```

If things go wrong during authentication

Unauthorized Errors: If the authentication process fails, `datalad xnat-init` will throw an error:

```
xnat_init(error): . (dataset) [Request to XNAT server failed: Unauthorized]
```

In this case, read on in the last paragraph on how to update your credential.

Faulty XNAT URLs: If the provided XNAT URL is fault, and can be appropriately reached, you may see an error like this:

```
xnat_init(error): . (dataset) [During authentication the XNAT server sent␣
↪MissingSchema(Invalid URL 'myxnatinstance/data/JSESSION': No schema supplied. Perhaps␣
↪you meant http://<wrongurl>/data/JSESSION?)]
```

In this case, double check the URL you provided. Open an issue if you need help or think that you found a bug.

Updating credentials

"Oh no, I accidentally mistyped my password!" If you supplied wrong credentials, or previously working credentials expired and stopped working, you can re-enter new credentials with the configuration `datalad.credentials.force-ask=1`:

```
$ datalad -c datalad.credentials.force-ask=1 xnat-init <url>
You need to authenticate with [...]
user: <user>
password: <password>
```

Alternatively, find your system's secure Keyring (your systems credential store) and remove or replace your password in there.

1.3.2 Tracking a project

1.3.3 Internals: Understanding xnat-init

Understanding the dataset configurations

One of the main functions of the `datalad xnat-init` command is to create a dataset-internal configuration with information about the XNAT instance, the directory structure for downloaded files, and the project to track. This configuration is what determines the looks and feel of the final dataset, in particular the presence and location of *subdataset*s, and the imaging files you will be able to retrieve afterwards.

The individual components of these configurations are spread over two different places in your dataset:

1. a DataLad configuration within `.datalad/config`
2. a provider configuration within `.datalad/providers/`

Both of these configurations are dataset-specific, i.e., configure only the behavior of this particular dataset, not other datasets you may have on your system.

The provider configuration

The first configuration created by `datalad xnat-init` is a so-called "provider configuration". Provider configurations are small, plain text files that configure how a specific service provider shall be accessed. You can find general information on them in the DataLad handbook.

The provider configuration created by `datalad xnat-init` lives in `.datalad/providers/xnat-<name>.cfg`, where `name` is a placeholder for an arbitrary identifier. The default name is `.datalad/providers/xnat-default.cfg`, but in case of datasets that track projects from multiple different XNAT instances the identifier allows to differentiate between them.

We can take a look into an exemplary configuration file:

```
[provider:xnat-default]
url_re = https://xnat.kube.fz-juelich.de/.*
credential = xnat.kube.fz-juelich.de
authentication_type = http_basic_auth

[credential:xnat.kube.fz-juelich.de]
type = user_password
```

It specifies the URL to the XNAT instance supplied during `datalad xnat-init`, and determines the credential (such as authentication with a user name and a password) and authentication (such HTTP Basic Authentication) type.

The DataLad configuration

The second configuration created by `datalad xnat-init` is a DataLad configuration that configures the dataset to use a given provider configuration. It is included in the `.datalad/config file` and looks like this:

```
[datalad "xnat.default"]
        url = https://xnat.kube.fz-juelich.de
        project = phantoms
        path = {subject}/{session}/{scan}/
        credential-name = xnat.kube.fz-juelich.de
```

Note how it identifies a provider configuration via its <name>, and how it includes the configuration about which XNAT project to track.

1.4 Contributing

If you have any questions, comments, bug fixes or improvement suggestions, feel free to contact us via our Github page. Before contributing, be sure to read the contributing guidelines.

1.5 Acknowledgments

DataLad development is being performed as part of a US-German collaboration in computational neuroscience (CR-CNS) project "DataGit: converging catalogues, warehouses, and deployment logistics into a federated 'data distribution'" (Halchenko/Hanke), co-funded by the US National Science Foundation (NSF 1912266) and the German Federal Ministry of Education and Research (BMBF 01GQ1905). Additional support is provided by the German federal state of Saxony-Anhalt and the European Regional Development Fund (ERDF), Project: Center for Behavioral Brain Sciences, Imaging Platform.

DataLad is built atop the git-annex software that is being developed and maintained by Joey Hess.

The extension was created during the Juelich Brain Hackathon 2021 and wouldn't have been possible without a dedicated team of volunteers.

1.6 Glossary

credentials
: Something used to authenticate to a server, typically a username and a password. Datalad-xnat may prompt you for your credentials or use ones which have been previously saved (in datalad configuration or system keyring, see datalad docs on credential management). Some XNAT servers allow anonymous access (without checking credentials).

experiment
: In XNAT terms, an experiment is an event by which data is acquired. This data can be imaging data, or non-imaging data. It exists within the context of a *project*, but can be registered into multiple *project*s. Most experiments will be imaging *session*s.

project
: A project is used to define a collection of data stored in XNAT. These often correlate directly to an IRB approved study, or a multi-site data acquisition program. Within XNAT, the project is used to define a security structure for data. Users are given certain permissions for data within certain projects -- thus, as a user you may not have permissions for all projects on a given XNAT instance.

session
: In XNAT, an image session is a specific kind of an *experiment* which contains image data. A session groups together multiple scans, where a scan corresponds to a DICOM series or BIDS data acquisition / run.

subdataset
: A DataLad dataset contained within a different DataLad dataset (the parent or DataLad *superdataset*).

subject
: A subject is anyone who participates in a study, and exist within the context of a *project*. Subjects can be registered in multiple *project*s (e.g., to capture longitudinal data from various studies).

superdataset
: A DataLad dataset that contains one or more levels of other DataLad datasets (DataLad *subdataset*s).

XNAT
: XNAT is an open source imaging informatics platform developed by the Neuroinformatics Research Group at

Washington University. It facilitates common management, productivity, and quality assurance tasks for imaging and associated data. Imaging centers can operate an XNAT instance to manage their imaging acquisitions. Typically, they require a user name and password to gain access.

1.7 Walk-through connectomeDB

This walk-through tutorial shows how to obtain the subjects in a project of the ConnecotmeDB It is assumed that you have a working installation of datalad-xnat and you have accepted the data user agreement in connectomeDB.

Create a Datalad dataset. Here we'll call the dataset *hcp*

```
datalad create hcp
```

Move into the *hcp* dataset.

```
cd hcp
```

Initialize XNAT to track ConnectomeDB, and list all the projects in ConnectomeDB.We will use the project *WU_L1A_Subj*.

```
datalad xnat-init https://db.humanconnectome.org --project WU_L1A_Subj
```

If a dataset was already initialized before, you will need to force the initialization.

```
datalad xnat-init https://db.humanconnectome.org --project MGH_DIFF --force
```

Obtain all subjects within the project.

```
datalad xnat-update
```

Download all data that belongs to a subject, here subject ConnectomeDB_S01439. Make sure that you have enough free space on your disk.

```
datalad xnat-update -s ConnectomeDB_S01439
```

Now the data should start to download.

1.8 Command line reference

`datalad-xnat` has three main commands: `xnat-init` for for configuring a dataset to track XNAT projects, `xnat-update` for updating and retrieving files from tracked XNAT projects, and `xnat-query-files` for for querying available files on an XNAT server. Find out more about each command below.

1.8.1 datalad xnat-init

Synopsis

```
datalad xnat-init [-h] [-F PATHFMT] [-p ID] [-s ID] [-e ID] [-c LABEL] [--credential␣
↪NAME] [-f] [--interactive] [-d DATASET] [--version] url
```

Description

Initialize an existing dataset to track an XNAT project

Examples

Initialize a dataset in the current directory:

```
% datalad xnat-init http://central.xnat.org:8080
```

Initialize with anonymous access (no credentials used):

```
% datalad xnat-init https://central.xnat.org --credential anonymous
```

Use credentials previously stored as <NAME>:

```
% datalad xnat-init https://central.xnat.org --credential <NAME>
```

Track a specific XNAT project, without credentials:

```
% datalad xnat-init https://central.xnat.org --project Sample_DICOM --credential␣
↪anonymous
```

Options

url

XNAT instance URL.

-h, --help, --help-np

show this help message. --help-np forcefully disables the use of a pager for displaying the help message

-F *PATHFMT*, --pathfmt *PATHFMT*

Specify the directory structure for the downloaded files, and if/where a subdataset should be created. The format string must use POSIX notation and must end with a slash ('/'). To include the subject, session, or scan values, use the following format: {subject}/{session}/{scan}/ To insert a subdataset at a specific directory level use '//': {subject}/{session}//{scan}/. [Default: '{subject}/{session}/{scan}/']

-p ID, --project ID

accession ID of a single XNAT project to track.

-s ID, --subject ID

accession ID of a single subject to track.

-e ID, --experiment ID

accession ID of a single experiment to track.

-c LABEL, --collection LABEL

limit updates to a specific collection/resource. Can be given multiple times.

--credential NAME

name of the credential providing a user/password combination to be used for authentication. The special value 'anonymous' will cause no credentials to be used, and all XNAT requests to be performed anonymously. The credential can be supplied via configuration settings 'datalad.credential.<name>.{user|password}', or environment variables DATALAD_CREDENTIAL_<NAME>_{USER|PASSWORD}, or will be queried from the active credential store using the provided name. If none is provided, the host-part of the XNAT URL is used as a name (e.g. 'https://central.xnat.org' -> 'central.xnat.org'). Constraints: value must be a string or value must be NONE

-f, --force

force (re-)initialization.

--interactive

enables interactive configuration based on XNAT queries. Default: enabled in interactive sessions.

-d DATASET, --dataset DATASET

specify the dataset to perform the initialization on. Constraints: Value must be a Dataset or a valid identifier of a Dataset (e.g. a path) or value must be NONE

--version

show the module and its version which provides the command

Authors

datalad is developed by Michael Hanke <michael.hanke@gmail.com>.

1.8.2 datalad xnat-query-files

Synopsis

```
datalad xnat-query-files [-h] [-p ID] [-e ID] [-s ID] [--credential NAME] [--version] url
```

Description

Query an XNAT server for projects, or an XNAT project for subjects

Use this command to get a list of available projects at an XNAT instance for a given URL, or to get a list of subjects inside a specific project at the given XNAT instance.

Examples

Get a list of projects for a given XNAT instance::

```
% datalad xnat-query http://central.xnat.org:8080
```

Get a list of subject for a given XNAT project::

```
% datalad xnat-query http://central.xnat.org:8080 -p myproject
```

Options

url

XNAT instance URL to query.

-h, --help, --help-np

show this help message. --help-np forcefully disables the use of a pager for displaying the help message

-p ID, --project ID

accession ID of a single XNAT project to track.

-e ID, --experiment ID

accession ID of a single experiment to track.

-s ID, --subject ID

accession ID of a single subject to track.

--credential NAME

name of the credential providing a user/password combination to be used for authentication. The special value 'anonymous' will cause no credentials to be used, and all XNAT requests to be performed anonymously. The credential can be supplied via configuration settings 'datalad.credential.<name>.{user|password}', or environment variables DATALAD_CREDENTIAL_<NAME>_{USER|PASSWORD}, or will be queried from the active credential store using the provided name. If none is provided, the host-part of the XNAT URL is used as a name (e.g. 'https://central.xnat.org' -> 'central.xnat.org'). Constraints: value must be a string or value must be NONE

--version

show the module and its version which provides the command

Authors

datalad is developed by Michael Hanke <michael.hanke@gmail.com>.

1.8.3 datalad xnat-update

Synopsis

```
datalad xnat-update [-h] [-p ID] [-s ID] [-e ID] [-c LABEL] [--credential NAME] [-f] [--
↪reckless fast] [--ifexists {overwrite|skip}] [-J NJOBS] [-d DATASET] [--version]
```

Description

Update files for a subject(s) of an XNAT project.

This command expects an xnat-init initialized DataLad dataset. The dataset may or may not have existing content already.

Options

-h, --help, --help-np

show this help message. --help-np forcefully disables the use of a pager for displaying the help message

-p ID, --project ID

accession ID of a single XNAT project to track.

-s ID, --subject ID

accession ID of a single subject to track.

-e ID, --experiment ID

accession ID of a single experiment to track.

-c LABEL, --collection LABEL

limit updates to a specific collection/resource. Can be given multiple times.

--credential NAME

name of the credential providing a user/password combination to be used for authentication. The special value 'anonymous' will cause no credentials to be used, and all XNAT requests to be performed anonymously. The credential can be supplied via configuration settings 'datalad.credential.<name>.{user|password}', or environment variables DATALAD_CREDENTIAL_<NAME>_{USER|PASSWORD}, or will be queried from the active credential store using the provided name. If none is provided, the host-part of the XNAT URL is used as a name (e.g. 'https://central.xnat.org' -> 'central.xnat.org'). Constraints: value must be a string or value must be NONE

-f, --force

force (re-)building the addurl tables.

--reckless fast

Update the files in a potentially unsafe way. Supported modes are: ["fast"]: No content verification or download. Will only register the urls. Constraints: value must be one of ('fast',)

--ifexists {overwrite|skip}

Flag for addurls. Constraints: value must be one of ('overwrite', 'skip')

-J NJOBS, --jobs NJOBS

how many parallel jobs (where possible) to use. "auto" corresponds to the number defined by 'datalad.runtime.max-annex-jobs' configuration item NOTE: This option can only parallelize input retrieval (get) and output recording (save). DataLad does NOT parallelize your scripts for you. Constraints: value must be convertible to type 'int' or value must be NONE or value must be one of ('auto',) [Default: 'auto']

-d DATASET, --dataset DATASET

specify the dataset to perform the update on. Constraints: Value must be a Dataset or a valid identifier of a Dataset (e.g. a path) or value must be NONE

--version

show the module and its version which provides the command

Authors

datalad is developed by Michael Hanke <michael.hanke@gmail.com>.

1.9 Python API

`datalad-xnat` has three main commands that are exposed as functions via `datalad.api` and as methods of the `Dataset` class: `xnat_init` for configuring a dataset to track XNAT projects, `xnat_update` for updating and retrieving files from tracked XNAT projects, and `xnat_query-files` for querying available files on an XNAT server. Find out more about each command below.

xnat_init(url[, pathfmt, project, subject, ...])	Initialize an existing dataset to track an XNAT project
xnat_query_files(url[, project, experiment, ...])	Query an XNAT server for projects, or an XNAT project for subjects
xnat_update([project, subject, experiment, ...])	Update files for a subject(s) of an XNAT project.

1.9.1 datalad.api.xnat_init

datalad.api.**xnat_init**(*url*, *pathfmt='{subject}/{session}/{scan}/'*, *project=None*, *subject=None*, *experiment=None*, *collection=None*, *credential=None*, *force=False*, *interactive=None*, *dataset=None*)

Initialize an existing dataset to track an XNAT project

Examples

Initialize a dataset in the current directory:

```
> xnat_init("http://central.xnat.org:8080")
```

Initialize with anonymous access (no credentials used):

```
> xnat_init("https://central.xnat.org", credential="anonymous")
```

Use credentials previously stored as <NAME>:

```
> xnat_init("https://central.xnat.org", credential="<NAME>")
```

Track a specific XNAT project, without credentials:

```
> xnat_init("https://central.xnat.org", project="Sample_DICOM", credential=
↪"anonymous")
```

Parameters

- **url** -- XNAT instance URL.
- **pathfmt** -- Specify the directory structure for the downloaded files, and if/where a subdataset should be created. The format string must use POSIX notation and must end with a slash ('/'). To include the subject, session, or scan values, use the following format: {subject}/{session}/{scan}/ To insert a subdataset at a specific directory level use '//': {subject}/{session}//{scan}/. [Default: '{subject}/{session}/{scan}/']
- **project** -- accession ID of a single XNAT project to track. [Default: None]
- **subject** -- accession ID of a single subject to track. [Default: None]
- **experiment** -- accession ID of a single experiment to track. [Default: None]
- **collection** -- limit updates to a specific collection/resource. Multiple collections can be specified as a list. [Default: None]
- **credential** (*str or None, optional*) -- name of the credential providing a user/password combination to be used for authentication. The special value 'anonymous' will cause no credentials to be used, and all XNAT requests to be performed anonymously. The credential can be supplied via configuration settings 'datalad.credential.<name>.{user|password}', or environment variables DATALAD_CREDENTIAL_<NAME>_{USER|PASSWORD}, or will be queried from the active credential store using the provided name. If none is provided, the host-part of the XNAT URL is used as a name (e.g. 'https://central.xnat.org' -> 'central.xnat.org'). [Default: None]
- **force** (*bool, optional*) -- force (re-)initialization. [Default: False]
- **interactive** (*bool, optional*) -- enables interactive configuration based on XNAT queries. Default: enabled in interactive sessions. [Default: None]
- **dataset** (*Dataset or None, optional*) -- specify the dataset to perform the initialization on. [Default: None]
- **on_failure** (*{'ignore', 'continue', 'stop'}, optional*) -- behavior to perform on failure: 'ignore' any failure is reported, but does not cause an exception; 'continue' if any failure occurs an exception will be raised at the end, but processing other actions will continue for as long as possible; 'stop': processing will stop on first failure and an exception is

raised. A failure is any result with status 'impossible' or 'error'. Raised exception is an IncompleteResultsError that carries the result dictionaries of the failures in its *failed* attribute. [Default: 'continue']

- **result_filter** (*callable or None, optional*) -- if given, each to-be-returned status dictionary is passed to this callable, and is only returned if the callable's return value does not evaluate to False or a ValueError exception is raised. If the given callable supports ***kwargs* it will additionally be passed the keyword arguments of the original API call. [Default: None]
- **result_renderer** -- select rendering mode command results. 'tailored' enables a command- specific rendering style that is typically tailored to human consumption, if there is one for a specific command, or otherwise falls back on the the 'generic' result renderer; 'generic' renders each result in one line with key info like action, status, path, and an optional message); 'json' a complete JSON line serialization of the full result record; 'json_pp' like 'json', but pretty-printed spanning multiple lines; 'disabled' turns off result rendering entirely; '<template>' reports any value(s) of any result properties in any format indicated by the template (e.g. '{path}', compare with JSON output for all key-value choices). The template syntax follows the Python "format() language". It is possible to report individual dictionary values, e.g. '{metadata[name]}'. If a 2nd-level key contains a colon, e.g. 'music:Genre', ':' must be substituted by '#' in the template, like so: '{metadata[music#Genre]}'. [Default: 'tailored']
- **result_xfm** (*{'datasets', 'successdatasets-or-none', 'paths', 'relpaths', 'metadata'} or callable or None, optional*) -- if given, each to-be-returned result status dictionary is passed to this callable, and its return value becomes the result instead. This is different from *result_filter*, as it can perform arbitrary transformation of the result value. This is mostly useful for top- level command invocations that need to provide the results in a particular format. Instead of a callable, a label for a pre-crafted result transformation can be given. [Default: None]
- **return_type** (*{'generator', 'list', 'item-or-list'}, optional*) -- return value behavior switch. If 'item-or-list' a single value is returned instead of a one-item return value list, or a list in case of multiple return values. *None* is return in case of an empty list. [Default: 'list']

1.9.2 datalad.api.xnat_query_files

datalad.api.**xnat_query_files**(*url*, *project=None*, *experiment=None*, *subject=None*, *credential=None*)

Query an XNAT server for projects, or an XNAT project for subjects

Use this command to get a list of available projects at an XNAT instance for a given URL, or to get a list of subjects inside a specific project at the given XNAT instance.

Examples

Get a list of projects for a given XNAT instance::

```
> xnat_query("http://central.xnat.org:8080")
```

Get a list of subject for a given XNAT project::

```
> xnat_query("http://central.xnat.org:8080", project="myproject")
```

Parameters

- **url** -- XNAT instance URL to query.
- **project** -- accession ID of a single XNAT project to track. [Default: None]
- **experiment** -- accession ID of a single experiment to track. [Default: None]
- **subject** -- accession ID of a single subject to track. [Default: None]
- **credential** (*str or None, optional*) -- name of the credential providing a user/password combination to be used for authentication. The special value 'anonymous' will cause no credentials to be used, and all XNAT requests to be performed anonymously. The credential can be supplied via configuration settings 'datalad.credential.<name>.{user|password}', or environment variables DATALAD_CREDENTIAL_<NAME>_{USER|PASSWORD}, or will be queried from the active credential store using the provided name. If none is provided, the host-part of the XNAT URL is used as a name (e.g. 'https://central.xnat.org' -> 'central.xnat.org'). [Default: None]
- **on_failure** (*{'ignore', 'continue', 'stop'}, optional*) -- behavior to perform on failure: 'ignore' any failure is reported, but does not cause an exception; 'continue' if any failure occurs an exception will be raised at the end, but processing other actions will continue for as long as possible; 'stop': processing will stop on first failure and an exception is raised. A failure is any result with status 'impossible' or 'error'. Raised exception is an IncompleteResultsError that carries the result dictionaries of the failures in its *failed* attribute. [Default: 'continue']
- **result_filter** (*callable or None, optional*) -- if given, each to-be-returned status dictionary is passed to this callable, and is only returned if the callable's return value does not evaluate to False or a ValueError exception is raised. If the given callable supports ***kwargs* it will additionally be passed the keyword arguments of the original API call. [Default: None]
- **result_renderer** -- select rendering mode command results. 'tailored' enables a command- specific rendering style that is typically tailored to human consumption, if there is one for a specific command, or otherwise falls back on the the 'generic' result renderer; 'generic' renders each result in one line with key info like action, status, path, and an optional message); 'json' a complete JSON line serialization of the full result record; 'json_pp' like 'json', but pretty-printed spanning multiple lines; 'disabled' turns off result rendering entirely; '<template>' reports any value(s) of any result properties in any format indicated by the template (e.g. '{path}', compare with JSON output for all key-value choices). The template syntax follows the Python "format() language". It is possible to report individual dictionary values, e.g. '{metadata[name]}'. If a 2nd-level key contains a colon, e.g. 'music:Genre', ':' must be substituted by '#' in the template, like so: '{metadata[music#Genre]}'. [Default: 'tailored']
- **result_xfm** (*{'datasets', 'successdatasets-or-none', 'paths', 'relpaths', 'metadata'} or callable or None, optional*) -- if given, each to-be-returned result status dictionary is passed to this callable, and its return value becomes the result instead. This is different from *result_filter*, as it can perform arbitrary transformation of the result value. This is mostly useful for top- level command invocations that need to provide the results in a particular format. Instead of a callable, a label for a pre-crafted result transformation can be given. [Default: None]
- **return_type** (*{'generator', 'list', 'item-or-list'}, optional*) -- return value behavior switch. If 'item-or-list' a single value is returned instead of a one-item return value list, or a list in case of multiple return values. *None* is return in case of an empty list. [Default: 'list']

1.9.3 datalad.api.xnat_update

datalad.api.**xnat_update**(*project=None*, *subject=None*, *experiment=None*, *collection=None*, *credential=None*, *force=False*, *reckless=None*, *ifexists=None*, *jobs='auto'*, *dataset=None*)

Update files for a subject(s) of an XNAT project.

This command expects an xnat-init initialized DataLad dataset. The dataset may or may not have existing content already.

Parameters

- **project** -- accession ID of a single XNAT project to track. [Default: None]
- **subject** -- accession ID of a single subject to track. [Default: None]
- **experiment** -- accession ID of a single experiment to track. [Default: None]
- **collection** -- limit updates to a specific collection/resource. Multiple collections can be specified as a list. [Default: None]
- **credential** (*str or None, optional*) -- name of the credential providing a user/password combination to be used for authentication. The special value 'anonymous' will cause no credentials to be used, and all XNAT requests to be performed anonymously. The credential can be supplied via configuration settings 'datalad.credential.<name>.{user|password}', or environment variables DATALAD_CREDENTIAL_<NAME>_{USER|PASSWORD}, or will be queried from the active credential store using the provided name. If none is provided, the host-part of the XNAT URL is used as a name (e.g. 'https://central.xnat.org' -> 'central.xnat.org'). [Default: None]
- **force** (*bool, optional*) -- force (re-)building the addurl tables. [Default: False]
- **reckless** (*{None, 'fast'}, optional*) -- Update the files in a potentially unsafe way. Supported modes are: ["fast"]: No content verification or download. Will only register the urls. [Default: None]
- **ifexists** (*{None, 'overwrite', 'skip'}, optional*) -- Flag for addurls. [Default: None]
- **jobs** (*int or None or {'auto'}, optional*) -- how many parallel jobs (where possible) to use. "auto" corresponds to the number defined by 'datalad.runtime.max-annex-jobs' configuration item NOTE: This option can only parallelize input retrieval (get) and output recording (save). DataLad does NOT parallelize your scripts for you. [Default: 'auto']
- **dataset** (*Dataset or None, optional*) -- specify the dataset to perform the update on. [Default: None]
- **on_failure** (*{'ignore', 'continue', 'stop'}, optional*) -- behavior to perform on failure: 'ignore' any failure is reported, but does not cause an exception; 'continue' if any failure occurs an exception will be raised at the end, but processing other actions will continue for as long as possible; 'stop': processing will stop on first failure and an exception is raised. A failure is any result with status 'impossible' or 'error'. Raised exception is an IncompleteResultsError that carries the result dictionaries of the failures in its *failed* attribute. [Default: 'continue']
- **result_filter** (*callable or None, optional*) -- if given, each to-be-returned status dictionary is passed to this callable, and is only returned if the callable's return value does not evaluate to False or a ValueError exception is raised. If the given callable supports ***kwargs* it will additionally be passed the keyword arguments of the original API call. [Default: None]
- **result_renderer** -- select rendering mode command results. 'tailored' enables a command- specific rendering style that is typically tailored to human consumption, if there

is one for a specific command, or otherwise falls back on the the 'generic' result renderer; 'generic' renders each result in one line with key info like action, status, path, and an optional message); 'json' a complete JSON line serialization of the full result record; 'json_pp' like 'json', but pretty-printed spanning multiple lines; 'disabled' turns off result rendering entirely; '<template>' reports any value(s) of any result properties in any format indicated by the template (e.g. '{path}', compare with JSON output for all key-value choices). The template syntax follows the Python "format() language". It is possible to report individual dictionary values, e.g. '{metadata[name]}'. If a 2nd-level key contains a colon, e.g. 'music:Genre', ':' must be substituted by '#' in the template, like so: '{metadata[music#Genre]}'. [Default: 'tailored']

- **result_xfm** (*{'datasets', 'successdatasets-or-none', 'paths', 'relpaths', 'metadata'} or callable or None, optional*) -- if given, each to-be-returned result status dictionary is passed to this callable, and its return value becomes the result instead. This is different from *result_filter*, as it can perform arbitrary transformation of the result value. This is mostly useful for top- level command invocations that need to provide the results in a particular format. Instead of a callable, a label for a pre-crafted result transformation can be given. [Default: None]
- **return_type** (*{'generator', 'list', 'item-or-list'}, optional*) -- return value behavior switch. If 'item-or-list' a single value is returned instead of a one-item return value list, or a list in case of multiple return values. *None* is return in case of an empty list. [Default: 'list']

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

INDEX