Package: xnatR (via r-universe)

June 2, 2026

Type Package

Title Interface with XNAT Systems

Version 0.2.0

Author Bradley Buchsbaum [aut, cre]

Maintainer Bradley Buchsbaum <brad.buchsbaum@gmail.com>

Description An R package for interacting with XNAT systems via the REST API. Provides functions to authenticate, list projects/subjects/experiments/scans, search, and download data from XNAT neuroimaging archives.

License MIT + file LICENSE

URL https://github.com/bbuchsbaum/xnatR

BugReports https://github.com/bbuchsbaum/xnatR/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports httr2 (>= 1.0.0), jsonlite, tibble, yaml, cli, rlang

Suggests testthat (>= 3.0.0), httptest2, withr, pkgload, knitr, rmarkdown

SystemRequirements fzf (optional, for the xnat_explore() terminal browser)

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs libssl-dev

Repository https://bbuchsbaum.r-universe.dev

Date/Publication 2026-06-02 13:09:45 UTC

RemoteUrl https://github.com/bbuchsbaum/xnatR

RemoteRef HEAD

RemoteSha 6e59d77b7af8503ebb9f33778ab07e124273233f

Contents

`authenticate_xnat` *Authenticate with an XNAT Server*

Description

Establishes authentication credentials for communicating with an XNAT server. Credentials are stored in the package environment for use by other functions.

Usage

```
authenticate_xnat(
  base_url = NULL,
  username = NULL,
  password = NULL,
  ssl_verify = TRUE,
  verify = TRUE,
  use_jsession = FALSE
)
```

Arguments

`base_url`	The base URL of the XNAT server (e.g., "https://central.xnat.org"). If NULL, checks XNATR_HOST environment variable, then config file, then .netrc.
`username`	XNAT username. If NULL, checks XNATR_USER environment variable, then config file, then .netrc. For XNAT 1.8+ with auth providers, use "provider/username" format.
`password`	XNAT password. If NULL, checks XNATR_PASS environment variable, then config file, then .netrc.
`ssl_verify`	Whether to verify SSL certificates. Default TRUE.
`verify`	Whether to verify credentials by making a test request. Default TRUE.
`use_jsession`	Whether to establish a JSESSION cookie-based session instead of using Basic Auth for each request. Can improve performance. Default FALSE.

Value

Invisibly returns TRUE if authentication is successful.

Examples

```
## Not run:
# Direct credentials
authenticate_xnat(
  base_url = "https://central.xnat.org",
  username = "guest",
  password = "guest"
)

# Using environment variables (XNATR_HOST, XNATR_USER, XNATR_PASS)
authenticate_xnat()

# Using config file (~/.xnatR_config.yml)
authenticate_xnat()

# Using .netrc file
authenticate_xnat(base_url = "https://central.xnat.org")

# With JSESSION for better performance
authenticate_xnat(
  base_url = "https://central.xnat.org",
  username = "myuser",
  password = "mypass",
  use_jsession = TRUE
)

# XNAT 1.8+ with auth provider
authenticate_xnat(
  base_url = "https://myxnat.org",
  username = "ldap/myuser",
  password = "mypass"
)

## End(Not run)
```

`dicom_dump` *Dump DICOM header fields for a session or scan*

Description

Reads DICOM header values directly from the files stored in the XNAT archive using the `/data/services/dicomdump` service, **without downloading the image files**. This is useful for inspecting acquisition parameters (repetition time, echo time, sequence name, and so on) or for filtering scans on real header values before deciding what to download.

Usage

```
dicom_dump(experiment_id, scan_id = NULL, fields = NULL, client = NULL)
```

Arguments

experiment_id	The experiment (session) identifier (required).
scan_id	Optional scan identifier. When supplied, the dump is limited to that scan; otherwise the whole session is dumped.
fields	Optional character vector of DICOM tags to return. Tags may be given as 8-digit hex ("00100010") or in "(0010,0010)" / "0010,0010" form; recognised tags are normalised to the 8-digit hex the service expects. When NULL, all available header fields are returned.
client	Optional xnat_client. If NULL, uses the global session.

Details

Only DICOM-backed scans are supported. The service samples header values from the archived files, and the exact columns returned depend on your XNAT version.

Value

A tibble of class xnat_dicom_dump. Columns follow the XNAT DICOM dump ResultSet and typically include the DICOM tag, value representation (vr), description, and value.

Examples

```
## Not run:
xnat_connect(base_url = "https://central.xnat.org",
             username = "guest", password = "guest")

# Dump every header field for a session
dicom_dump("OAS1_0001_MR1")

# A single scan, only Modality (0008,0060) and Repetition Time (0018,0080)
dicom_dump("OAS1_0001_MR1", scan_id = "1",
           fields = c("00080060", "00180080"))

## End(Not run)
```

---

download_experiment *Download an Experiment Archive*

Description

Downloads an experiment scan selection (ALL by default) as an archive.

Usage

```
download_experiment(
  experiment_id,
  scan_id = "ALL",
  format = "zip",
  dest_dir = tempdir(),
  dest_file = NULL,
  extract = FALSE,
  progress = TRUE,
  strict = TRUE,
  client = NULL
)
```

Arguments

experiment_id	Experiment identifier.
scan_id	Scan ID/type to download; defaults to "ALL".
format	Archive format: "zip" (default) or "tar.gz".
dest_dir	Destination directory. Defaults to tempdir().
dest_file	Optional destination filename. If NULL, auto-generated.
extract	If TRUE, extract archive contents and return extracted paths.
progress	Show download progress bar. Default TRUE.
strict	If TRUE (default), raise errors on failed download. If FALSE, return NULL on failure.
client	Optional xnat_client. If NULL, uses the global session.

Value

Archive path, extracted file paths (when extract = TRUE), or NULL on failure when strict = FALSE.

---

download_files *Download Files from XNAT*

Description

Downloads files from an XNAT experiment, with options to select specific scans and resources.

Usage

```
download_files(
  project_id,
  subject_id,
  experiment_id,
  scan_id = "ALL",
```

```
  resource = NULL,
  format = "zip",
  dest_dir = getwd(),
  dest_file = NULL,
  progress = TRUE,
  client = NULL
)
```

Arguments

project_id	The project identifier.
subject_id	The subject identifier.
experiment_id	The experiment identifier.
scan_id	Scan ID(s) to download. Can be: • A single scan ID (e.g., "1") • Multiple IDs (e.g., c("1", "2", "3") or "1,2,3") • "ALL" to download all scans (default)
resource	Resource name (e.g., "DICOM", "NIFTI"). NULL for all resources.
format	Download format: "zip" (default) or "tar.gz".
dest_dir	Destination directory. Defaults to current working directory.
dest_file	Custom destination filename. If NULL, auto-generated.
progress	Show download progress bar. Default TRUE.
client	Optional xnat_client. If NULL, uses the global session.

Value

Invisibly returns the path to the downloaded file.

Examples

```
## Not run:
authenticate_xnat(base_url = "https://central.xnat.org",
                  username = "guest", password = "guest")

# Download all scans as ZIP
download_files(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001"
)

# Download specific scans
download_files(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001",
  scan_id = c("1", "2")
```

```
)

# Download DICOM only
download_files(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001",
  resource = "DICOM"
)

## End(Not run)
```

`download_subject` *Download All Data for a Subject*

Description

Downloads all experiments and scans for a subject.

Usage

```
download_subject(
  project_id,
  subject_id,
  format = "zip",
  dest_dir = getwd(),
  progress = TRUE,
  client = NULL
)
```

Arguments

`project_id`	The project identifier.
`subject_id`	The subject identifier.
`format`	Download format: "zip" (default) or "tar.gz".
`dest_dir`	Destination directory.
`progress`	Show download progress bar.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

Invisibly returns a character vector of downloaded file paths.

Examples

```
## Not run:
paths <- download_subject(
  project_id = "MyProject",
  subject_id = "Subject001",
  dest_dir = "~/Downloads"
)

## End(Not run)
```

`download_xnat_file` *Download a Single File from XNAT*

Description

Downloads a file using its full URL or a pre-constructed path.

Usage

```
download_xnat_file(url, dest_file, progress = TRUE, client = NULL)
```

Arguments

`url`	Full URL to the file, or a path relative to the XNAT base URL.
`dest_file`	Destination file path.
`progress`	Show download progress bar. Default TRUE.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

Invisibly returns the destination path.

Examples

```
## Not run:
download_xnat_file(
 url = "/data/projects/P1/subjects/S1/experiments/E1/scans/1/resources/DICOM/files/file.dcm",
  dest_file = "file.dcm"
)

## End(Not run)
```

initialize_config *Initialize Configuration File*

Description

Creates a template configuration file at ~/.xnatR_config.yml

Usage

```
initialize_config()
```

Value

Invisibly returns the path to the config file.

Examples

```
## Not run:
initialize_config()
# Edit ~/.xnatR_config.yml with your credentials

## End(Not run)
```

install_cli *Install the xnatR command-line wrapper*

Description

Copies the packaged `xnatR` command wrapper to a directory on your shell `PATH`, or to another destination directory you choose.

Usage

```
install_cli(dest_dir = "~/.local/bin", overwrite = FALSE, commands = NULL)
```

Arguments

`dest_dir`	Destination directory for command wrappers.
`overwrite`	If `TRUE`, replace an existing command wrapper.
`commands`	Optional command names to install. Currently only `"xnatR"` is supported.

Value

Invisibly returns the installed wrapper path.

is_authenticated *Check if authenticated*

Description

Check if authenticated

Usage

```
is_authenticated(client = NULL)
```

Arguments

client	Optional xnat_client. If NULL, uses the global session.

Value

TRUE if credentials are stored, FALSE otherwise.

Examples

```
## Not run:
if (is_authenticated()) {
  list_projects()
}

## End(Not run)
```

list_assessors *List Assessors for an Experiment*

Description

Retrieves assessors (derived data like FreeSurfer outputs, QC results) for a specific experiment.

Usage

```
list_assessors(
  project_id,
  subject_id,
  experiment_id,
  columns = NULL,
  limit = NULL,
  offset = NULL,
  client = NULL
)
```

Arguments

project_id	The project identifier.
subject_id	The subject identifier.
experiment_id	The experiment identifier.
columns	Character vector of column names to include.
limit	Maximum number of results to return.
offset	Number of results to skip for pagination.
client	Optional xnat_client. If NULL, uses the global session.

Value

A tibble of class xnat_assessors containing assessor details. Common columns include:

- ID: Assessor identifier
- label: Assessor label
- xsiType: Assessor data type
- project: Project ID

Examples

```
## Not run:
assessors <- list_assessors(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001"
)

## End(Not run)
```

---

list_data_types *List Available Data Types*

Description

Retrieves all data types (XSI types) available on the XNAT server. These are the schema elements that define different types of data (e.g., xnat:mrSessionData, xnat:subjectData).

Usage

```
list_data_types(client = NULL)
```

Arguments

client	Optional xnat_client. If NULL, uses the global session.

Value

A tibble of class xnat_datatypes containing data type information. Columns include:

- ELEMENT_NAME: The XSI type name (e.g., "xnat:mrSessionData")
- Other schema-related columns

Examples

```
## Not run:
datatypes <- list_data_types()
print(datatypes)

# Find all MR-related types
mr_types <- datatypes[grep("mr", datatypes$ELEMENT_NAME, ignore.case = TRUE), ]

## End(Not run)
```

---

list_experiment_files *List Files in an Experiment Resource*

Description

Retrieves files from an experiment-level resource.

Usage

```
list_experiment_files(
  project_id,
  subject_id,
  experiment_id,
  resource,
  client = NULL
)
```

Arguments

project_id	The project identifier.
subject_id	The subject identifier.
experiment_id	The experiment identifier.
resource	The resource label.
client	Optional xnat_client. If NULL, uses the global session.

Value

A tibble of class xnat_files.

Examples

```
## Not run:
files <- list_experiment_files(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001",
  resource = "SNAPSHOTS"
)

## End(Not run)
```

`list_experiment_files_all` *List All Files for an Experiment*

Description

Returns experiment-level files and, optionally, falls back to scan-level files (`scans/ALL/files`) if none are present at the experiment level.

Usage

```
list_experiment_files_all(
  experiment_id,
  include_scan_level = TRUE,
  client = NULL
)
```

Arguments

`experiment_id`	Experiment identifier.
`include_scan_level`	If `TRUE` (default), fall back to `scans/ALL/files` when experiment-level files are empty.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_files`.

list_experiment_resources *List Resources for an Experiment*

Description

Retrieves resources available at the experiment level.

Usage

```
list_experiment_resources(project_id, subject_id, experiment_id, client = NULL)
```

Arguments

project_id	The project identifier.
subject_id	The subject identifier.
experiment_id	The experiment identifier.
client	Optional xnat_client. If NULL, uses the global session.

Value

A tibble of class xnat_resources.

Examples

```
## Not run:
resources <- list_experiment_resources(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001"
)

## End(Not run)
```

list_experiments *List Experiments (Sessions) for a Project or Subject*

Description

Retrieves a list of experiments (imaging sessions). When only project_id is given, returns all experiments across every subject in that project. When subject_id is also supplied, results are scoped to that single subject.

Usage

```
list_experiments(
  project_id,
  subject_id = NULL,
  columns = NULL,
  limit = NULL,
  offset = NULL,
  client = NULL
)
```

Arguments

project_id	The project identifier (required).
subject_id	Optional subject identifier. When NULL, all experiments in the project are returned.
columns	Character vector of column names to include.
limit	Maximum number of results to return.
offset	Number of results to skip for pagination.
client	Optional xnat_client. If NULL, uses the global session.

Value

A tibble of class xnat_experiments containing experiment details. Common columns include:

- ID: Experiment identifier
- label: Experiment label
- project: Project ID
- subject_ID: Subject ID
- xsiType: Experiment type (e.g., "xnat:mrSessionData")
- date: Session date

Examples

```
## Not run:
authenticate_xnat(base_url = "https://central.xnat.org",
                  username = "guest", password = "guest")

# All experiments in a project
all_exps <- list_experiments(project_id = "MyProject")

# Experiments for one subject
subj_exps <- list_experiments(
  project_id = "MyProject",
  subject_id = "Subject001"
)

## End(Not run)
```

`list_files` *List Files in a Resource*

Description

Retrieves a list of files within a specific resource.

Usage

```
list_files(
  project_id,
  subject_id,
  experiment_id,
  scan_id,
  resource,
  client = NULL
)
```

Arguments

`project_id`	The project identifier.
`subject_id`	The subject identifier.
`experiment_id`	The experiment identifier.
`scan_id`	The scan identifier.
`resource`	The resource label (e.g., "DICOM", "NIFTI").
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_files` containing file details. Common columns include:

- `Name`: File name
- `Size`: File size in bytes
- `URI`: File URI for downloading
- `collection`: Resource collection name
- `file_format`: File format
- `file_content`: Content type

Examples

```
## Not run:
files <- list_files(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001",
```

```
  scan_id = "1",
  resource = "DICOM"
)

## End(Not run)
```

`list_projects` *List XNAT Projects*

Description

Retrieves a list of projects from the authenticated XNAT server.

Usage

```
list_projects(columns = NULL, limit = NULL, offset = NULL, client = NULL)
```

Arguments

`columns`	Character vector of column names to include in the result. Use NULL (default) to return all available columns.
`limit`	Maximum number of results to return. NULL for no limit.
`offset`	Number of results to skip. Used for pagination.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_projects` containing project details. Common columns include:

- `ID`: Project identifier
- `name`: Project display name
- `secondary_ID`: Secondary identifier
- `description`: Project description
- `pi_firstname`, `pi_lastname`: Principal investigator

Examples

```
## Not run:
authenticate_xnat(base_url = "https://central.xnat.org",
                  username = "guest", password = "guest")

# List all projects
projects <- list_projects()

# Get specific columns only
projects <- list_projects(columns = c("ID", "name", "description"))
```

```
# Pagination
first_10 <- list_projects(limit = 10)
next_10 <- list_projects(limit = 10, offset = 10)

## End(Not run)
```

`list_queryable_fields` *List Queryable Fields for a Data Type*

Description

Retrieves the fields that can be queried/searched for a specific XSI type. Useful for building search queries.

Usage

```
list_queryable_fields(xsi_type, client = NULL)
```

Arguments

`xsi_type`	The XSI type to query (e.g., "xnat:mrSessionData", "xnat:subjectData").
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_fields` containing field information. Common columns include:

- `FIELD_ID`: Field identifier for queries
- `TYPE`: Data type of the field
- `HEADER`: Display name
- `DESC`: Field description

Examples

```
## Not run:
# Get queryable fields for MR sessions
fields <- list_queryable_fields("xnat:mrSessionData")

# Get fields for subjects
subject_fields <- list_queryable_fields("xnat:subjectData")

## End(Not run)
```

`list_recent_sessions` *List Recent Sessions for a Project*

Description

Convenience wrapper around `list_experiments()` that returns experiments sorted by date (most recent first) with an optional cap on the number of rows returned.

Usage

```
list_recent_sessions(
  project_id,
  n = NULL,
  subject_id = NULL,
  columns = NULL,
  client = NULL
)
```

Arguments

`project_id`	The project identifier (required).
`n`	Maximum number of sessions to return. `NULL` (the default) returns all sessions.
`subject_id`	Optional subject identifier to scope results.
`columns`	Character vector of column names to include.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_experiments`, ordered most-recent first.

Examples

```
## Not run:
# 10 most recent sessions in a project
recent <- list_recent_sessions("MyProject", n = 10)

# All sessions for a subject, newest first
recent_subj <- list_recent_sessions("MyProject", subject_id = "S001")

## End(Not run)
```

`list_reconstructions` *List Reconstructions for an Experiment*

Description

Retrieves reconstructions (processed image data) for a specific experiment.

Usage

```
list_reconstructions(
  project_id,
  subject_id,
  experiment_id,
  columns = NULL,
  limit = NULL,
  offset = NULL,
  client = NULL
)
```

Arguments

`project_id`	The project identifier.
`subject_id`	The subject identifier.
`experiment_id`	The experiment identifier.
`columns`	Character vector of column names to include.
`limit`	Maximum number of results to return.
`offset`	Number of results to skip for pagination.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_reconstructions` containing reconstruction details. Common columns include:

- `ID`: Reconstruction identifier
- `type`: Reconstruction type
- `xsiType`: Data type

Examples

```
## Not run:
recons <- list_reconstructions(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001"
)
```

```
## End(Not run)
```

`list_resources` *List Resources for a Scan*

Description

Retrieves resources (e.g., DICOM, NIFTI) available for a specific scan.

Usage

```
list_resources(project_id, subject_id, experiment_id, scan_id, client = NULL)
```

Arguments

`project_id`	The project identifier.
`subject_id`	The subject identifier.
`experiment_id`	The experiment identifier.
`scan_id`	The scan identifier.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_resources` containing resource details. Common columns include:

- `xnat_abstractresource_id`: Resource ID
- `label`: Resource label (e.g., "DICOM", "NIFTI")
- `file_count`: Number of files
- `file_size`: Total size in bytes
- `format`: Resource format

Examples

```
## Not run:
resources <- list_resources(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Exp001",
  scan_id = "1"
)

## End(Not run)
```

`list_scans` *List Scans for an Experiment*

Description

Retrieves a list of scans from a specific experiment.

Usage

```
list_scans(
  project_id,
  subject_id,
  experiment_id,
  columns = NULL,
  limit = NULL,
  offset = NULL,
  client = NULL
)
```

Arguments

`project_id`	The project identifier (required).
`subject_id`	The subject identifier (required).
`experiment_id`	The experiment identifier (required).
`columns`	Character vector of column names to include.
`limit`	Maximum number of results to return.
`offset`	Number of results to skip for pagination.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_scans` containing scan details. Common columns include:

- `ID`: Scan number
- `type`: Scan type (e.g., "T1", "T2", "BOLD")
- `series_description`: Series description
- `quality`: Scan quality rating
- `xsiType`: Scan data type
- `frames`: Number of frames

Examples

```
## Not run:
authenticate_xnat(base_url = "https://central.xnat.org",
                  username = "guest", password = "guest")

scans <- list_scans(
  project_id = "MyProject",
  subject_id = "Subject001",
  experiment_id = "Experiment001"
)

## End(Not run)
```

`list_subjects` *List Subjects in a Project*

Description

Retrieves a list of subjects from a specific project on the XNAT server.

Usage

```
list_subjects(
  project_id,
  columns = NULL,
  limit = NULL,
  offset = NULL,
  client = NULL
)
```

Arguments

`project_id`	The project identifier (required).
`columns`	Character vector of column names to include. Use NULL (default) to return all available columns.
`limit`	Maximum number of results to return.
`offset`	Number of results to skip for pagination.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble of class `xnat_subjects` containing subject details. Common columns include:

- `ID`: Subject identifier
- `label`: Subject label

- project: Project ID
- gender: Subject gender
- handedness: Subject handedness
- age: Subject age

Examples

```
## Not run:
authenticate_xnat(base_url = "https://central.xnat.org",
                  username = "guest", password = "guest")

subjects <- list_subjects(project_id = "MyProject")

# With pagination
subjects <- list_subjects(project_id = "MyProject", limit = 100)

## End(Not run)
```

---

search_criterion *Create a Search Criterion*

Description

Helper function to create a search criterion for use with xnat_search().

Usage

```
search_criterion(field, comparison, value)
```

Arguments

field	The field ID to search.
comparison	Comparison type: "EQUALS", "LIKE", "GREATER_THAN", "LESS_THAN", "GREATER_THAN_EQUAL", "LESS_THAN_EQUAL".
value	The value to compare against.

Value

A list representing the criterion.

Examples

```
## Not run:
# Create criteria for a search
project_criterion <- search_criterion(
  field = "xnat:mrSessionData/project",
  comparison = "EQUALS",
  value = "MyProject"
)

date_criterion <- search_criterion(
  field = "xnat:mrSessionData/date",
  comparison = "GREATER_THAN",
  value = "2023-01-01"
)

# Use in search
results <- xnat_search(
  root_type = "xnat:mrSessionData",
  fields = c("xnat:mrSessionData/ID"),
  criteria = list(project_criterion, date_criterion)
)

## End(Not run)
```

`search_execute` *Execute Search Builder Query*

Description

Execute Search Builder Query

Usage

```
search_execute(builder, format = "json", client = NULL)
```

Arguments

`builder`	A search builder object.
`format`	Output format: "json" (default) or "csv".
`client`	Optional `xnat_client`. If `NULL`, uses the builder client or global session.

Value

A tibble containing the search results.

search_projects *Search XNAT Projects*

Description

Searches for projects containing a substring in their ID or name.

Usage

```
search_projects(pattern, columns = NULL, client = NULL)
```

Arguments

pattern	A character string to search for (case-insensitive).
columns	Character vector of column names to include.
client	Optional xnat_client. If NULL, uses the global session.

Value

A tibble of class xnat_projects containing matching projects.

Examples

```
## Not run:
authenticate_xnat(base_url = "https://central.xnat.org",
                  username = "guest", password = "guest")

# Search for projects containing "test"
results <- search_projects("test")

# Search for projects with "MRI" in name
mri_projects <- search_projects("MRI")

## End(Not run)
```

search_scans *Search Scans Across XNAT*

Description

Performs a scan-parameter search across sessions/scans using XNAT's XML search endpoint with a tidy, xnatR-style interface.

Usage

```
search_scans(
  subject_id = NULL,
  project_id = NULL,
  age = NULL,
  experiment_id = NULL,
  scan_type = NULL,
  tr = NULL,
  te = NULL,
  ti = NULL,
  flip = NULL,
  voxel_res_units = NULL,
  voxel_res_x = NULL,
  voxel_res_y = NULL,
  voxel_res_z = NULL,
  orientation = NULL,
  client = NULL
)
```

Arguments

`subject_id`	Optional subject ID filter.
`project_id`	Optional project ID filter.
`age`	Optional age filter.
`experiment_id`	Optional experiment ID filter.
`scan_type`	Optional scan type filter.
`tr`	Optional repetition time filter.
`te`	Optional echo time filter.
`ti`	Optional inversion time filter.
`flip`	Optional flip angle filter.
`voxel_res_units`	Optional voxel-resolution units filter.
`voxel_res_x`	Optional X voxel-resolution filter.
`voxel_res_y`	Optional Y voxel-resolution filter.
`voxel_res_z`	Optional Z voxel-resolution filter.
`orientation`	Optional orientation filter.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble containing matching scan rows.

search_select *Add Fields to Search Builder*

Description

Add Fields to Search Builder

Usage

```
search_select(builder, ...)
```

Arguments

builder	A search builder object.
...	Field IDs to include in the results.

Value

The modified search builder (for chaining).

search_where *Add Criterion to Search Builder*

Description

Add Criterion to Search Builder

Usage

```
search_where(builder, field, comparison, value)
```

Arguments

builder	A search builder object.
field	Field ID to filter on.
comparison	Comparison type.
value	Value to compare.

Value

The modified search builder (for chaining).

with_xnat_client *Temporarily use a client as the global session*

Description

This helper makes existing code written for the global-session workflow work with an explicit `xnat_client`.

Usage

```
with_xnat_client(client, expr)
```

Arguments

`client`	An `xnat_client`.
`expr`	Expression to evaluate.

Value

The value of `expr`.

xnat_browse *Interactive browsing helpers*

Description

These helpers provide an interactive console UI for exploring XNAT resources. They are designed to be safe in non-interactive contexts: when `.interactive = FALSE`, the functions return the corresponding listing tibble without prompting.

xnat_browse_projects *Browse projects interactively*

Description

Browse projects interactively

Usage

```
xnat_browse_projects(
  client = NULL,
  pattern = NULL,
  columns = c("ID", "name", "description"),
  page_size = 20,
  .interactive = interactive()
)
```

Arguments

client	Optional xnat_client. If NULL, uses the global session.
pattern	Optional substring filter applied to the current page.
columns	Columns to display (defaults to common project fields).
page_size	Rows per page (default 20).
.interactive	If FALSE, returns the listing tibble without prompting.

Value

A tibble row selection (or NULL if quit). In non-interactive mode, returns a tibble of projects.

---

xnat_browse_subjects *Browse subjects interactively*

Description

If project_id is omitted in interactive mode, the function first prompts for a project.

Usage

```
xnat_browse_subjects(
  project_id = NULL,
  client = NULL,
  pattern = NULL,
  columns = c("ID", "label", "gender", "age"),
  page_size = 20,
  .interactive = interactive()
)
```

Arguments

project_id	Project identifier (optional in interactive mode).
client	Optional xnat_client. If NULL, uses the global session.
pattern	Optional substring filter applied to the current page.
columns	Columns to display (defaults to common subject fields).
page_size	Rows per page (default 20).
.interactive	If FALSE, returns the listing tibble without prompting.

Value

A tibble row selection (or NULL if quit). In non-interactive mode, returns a tibble of subjects.

xnat_client *Create an* xnat_client

Description

Low-level constructor for client objects. Prefer `xnat_connect()` for most use cases.

Usage

```
xnat_client(
  base_url,
  username = NULL,
  password = NULL,
  ssl_verify = TRUE,
  jsession = NULL
)
```

Arguments

`base_url`	Base URL of the XNAT server.
`username`	Username.
`password`	Password.
`ssl_verify`	Whether to verify SSL certificates. Default `TRUE`.
`jsession`	Optional JSESSIONID value.

Value

An `xnat_client`.

xnat_connect *XNAT client objects*

Description

`xnat_connect()` creates an explicit client object that can be passed to all xnatR functions via the `client` argument. This avoids relying on global session state stored in the package environment.

Usage

```
xnat_connect(
  base_url = NULL,
  username = NULL,
  password = NULL,
  ssl_verify = TRUE,
  verify = TRUE,
  use_jsession = FALSE,
  jsession = NULL,
  xnat_name = NULL
)
```

Arguments

base_url	Base URL of the XNAT server (e.g., "https://central.xnat.org").
username	Username, or NULL if using jsession only.
password	Password, or NULL if using jsession only.
ssl_verify	Whether to verify SSL certificates. Default TRUE.
verify	Whether to verify credentials by making a test request. Default TRUE.
use_jsession	Whether to establish a JSESSION cookie-based session instead of using Basic Auth for each request. Default FALSE.
jsession	Optional existing JSESSIONID value. When provided, requests use cookie auth and do not send Basic Auth.
xnat_name	Optional server nickname used for compatibility with Rxnat-style environment variables (<XNAT_NAME>_RXNAT_USER, <XNAT_NAME>_RXNAT_PASS).

Details

authenticate_xnat() remains available for a global-session workflow.

Value

xnat_connect() returns an xnat_client.

Examples

```
## Not run:
client <- xnat_connect(
  base_url = "https://central.xnat.org",
  username = "guest",
  password = "guest",
  use_jsession = TRUE
)

projects <- list_projects(client = client)

## End(Not run)
```

xnat_current_client *Get the current global-session client*

Description

Returns an xnat_client populated from the package environment (set by authenticate_xnat()), or NULL if not authenticated.

Usage

```
xnat_current_client()
```

Value

An xnat_client or NULL.

xnat_explore *Explore XNAT interactively with fzf*

Description

Drills down the project -> subject -> experiment -> scan hierarchy using fzf for fuzzy search and multi-select, with a preview pane showing the highlighted row's detail (DICOM headers at the scan level).

Usage

```
xnat_explore(
  client = NULL,
  project_id = NULL,
  subject_id = NULL,
  experiment_id = NULL,
  dest_dir = ".",
  .interactive = interactive(),
  run_fn = .fzf_run
)
```

Arguments

client	Optional xnat_client. If NULL, uses the global session.
project_id, subject_id, experiment_id	Optional starting point; supplying them skips the corresponding levels.
dest_dir	Destination directory for the Ctrl-D download action.
.interactive	If FALSE, the function errors rather than prompting.
run_fn	Internal seam for invoking fzf; overridden in tests.

Details

Keys: Enter opens the highlighted item; Tab marks rows; Ctrl-S returns the marked rows (e.g. to feed download_experiment()); Ctrl-D downloads the marked rows immediately and stays in the explorer; Ctrl-E (at the subject level) jumps to all sessions in the project; Ctrl-O opens the highlighted item in the XNAT web UI; Esc goes back a level (or exits at the top).

Requires the fzf binary on your PATH (e.g. brew install fzf) and an interactive session.

Value

A tibble of the rows selected with Ctrl-S (or Enter on a leaf scan), or NULL if you exit without selecting.

Examples

```
## Not run:
xnat_connect(base_url = "https://central.xnat.org",
             username = "guest", password = "guest")

chosen <- xnat_explore()
# Jump straight to a known session's scans:
xnat_explore(project_id = "CENTRAL_OASIS", subject_id = "OAS1_0001",
             experiment_id = "OAS1_0001_MR1")

## End(Not run)
```

---

xnat_logout *Log out from XNAT*

Description

Clears stored credentials and optionally invalidates the server session.

Usage

```
xnat_logout(invalidate_session = FALSE)
```

Arguments

invalidate_session	Whether to send a logout request to the server. Default FALSE (just clears local credentials).

Value

Invisibly returns TRUE.

Examples

```
## Not run:
xnat_logout()

# Also invalidate server session
xnat_logout(invalidate_session = TRUE)

## End(Not run)
```

`xnat_search` *Search XNAT Data*

Description

Performs an advanced search across XNAT data using the XML search API. This function builds the XML query for you based on simple R syntax.

Usage

```
xnat_search(root_type, fields, criteria = NULL, format = "json", client = NULL)
```

Arguments

`root_type`	The XSI type to search (e.g., "xnat:mrSessionData", "xnat:subjectData"). Use `list_data_types()` to see available types.
`fields`	Character vector of field IDs to return. Use `list_queryable_fields(root_type)` to see available fields.
`criteria`	A list of search criteria. Each criterion is a list with: • `field`: Field ID to search • `comparison`: One of "EQUALS", "LIKE", "GREATER_THAN", "LESS_THAN" • `value`: Value to compare
`format`	Output format: "json" (default) or "csv".
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A tibble containing the search results.

Examples

```
## Not run:
# Search for all MR sessions in a project
results <- xnat_search(
  root_type = "xnat:mrSessionData",
  fields = c("xnat:mrSessionData/ID", "xnat:mrSessionData/label",
```

```
            "xnat:mrSessionData/date"),
  criteria = list(
    list(field = "xnat:mrSessionData/project",
         comparison = "EQUALS",
         value = "MyProject")
  )
)

# Search for subjects with specific gender
subjects <- xnat_search(
  root_type = "xnat:subjectData",
  fields = c("xnat:subjectData/ID", "xnat:subjectData/label"),
  criteria = list(
    list(field = "xnat:subjectData/gender", comparison = "EQUALS", value = "M")
  )
)

## End(Not run)
```

`xnat_search_builder` *Search Builder - Fluent Interface*

Description

Creates a search builder object for constructing complex searches with a fluent API.

Usage

```
xnat_search_builder(root_type, client = NULL)
```

Arguments

`root_type`	The XSI type to search.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A search builder object.

Examples

```
## Not run:
# Build and execute a search using fluent API
results <- xnat_search_builder("xnat:mrSessionData") |>
  search_select("xnat:mrSessionData/ID", "xnat:mrSessionData/label") |>
  search_where("xnat:mrSessionData/project", "EQUALS", "MyProject") |>
  search_where("xnat:mrSessionData/date", "GREATER_THAN", "2023-01-01") |>
  search_execute()
```

```
## End(Not run)
```

`xnat_server` *Get current XNAT server URL*

Description

Get current XNAT server URL

Usage

```
xnat_server(client = NULL)
```

Arguments

`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

The current server URL or NULL if not authenticated.

Examples

```
## Not run:
xnat_server()

## End(Not run)
```

`xnat_token_invalidate` *Invalidate an Alias Token*

Description

Revokes an alias token so it can no longer be used for authentication.

Usage

```
xnat_token_invalidate(alias, secret, client = NULL)
```

Arguments

`alias`	The alias portion of the token to invalidate.
`secret`	The secret portion of the token to invalidate.
`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

Invisibly returns TRUE if successful.

Examples

```
## Not run:
# Invalidate a token
xnat_token_invalidate(alias = "my_alias", secret = "my_secret")

## End(Not run)
```

`xnat_token_issue` *Issue an Alias Token*

Description

Creates a new alias token for the authenticated user. Alias tokens can be used instead of passwords for authentication and can be invalidated independently.

Usage

```
xnat_token_issue(client = NULL)
```

Arguments

`client`	Optional `xnat_client`. If `NULL`, uses the global session.

Value

A list containing the token details:

- `alias`: The alias (username-like identifier)
- `secret`: The secret (password-like value)
- `estimatedExpirationTime`: When the token expires

Examples

```
## Not run:
authenticate_xnat(base_url = "https://central.xnat.org",
                  username = "myuser", password = "mypass")

token <- xnat_token_issue()
# Use token$alias and token$secret for subsequent authentication

## End(Not run)
```

xnat_token_list *List User's Alias Tokens*

Description

Retrieves all active alias tokens for the authenticated user.

Usage

```
xnat_token_list(client = NULL)
```

Arguments

client	Optional xnat_client. If NULL, uses the global session.

Value

A tibble of active tokens with columns:

- alias: Token alias
- xdatUserId: Associated user ID
- estimatedExpirationTime: Expiration timestamp

Examples

```
## Not run:
tokens <- xnat_token_list()
print(tokens)

## End(Not run)
```

xnat_token_validate *Validate an Alias Token*

Description

Checks if an alias token is still valid.

Usage

```
xnat_token_validate(alias, secret, client = NULL)
```

Arguments

alias	The alias portion of the token.
secret	The secret portion of the token.
client	Optional xnat_client. If NULL, uses the global session.

Value

A list with validation details:

- valid: Logical, whether the token is valid
- alias: The alias
- estimatedExpirationTime: When the token expires (if valid

Examples

```
## Not run:
# Check if a token is still valid
result <- xnat_token_validate(alias = "my_alias", secret = "my_secret")
if (result$valid) {
  message("Token is valid until ", result$estimatedExpirationTime)
}

## End(Not run)
```

---

xnat_username *Get current username*

Description

Get current username

Usage

```
xnat_username(client = NULL)
```

Arguments

client	Optional xnat_client. If NULL, uses the global session.

Value

The current username or NULL if not authenticated.

Examples

```
## Not run:
xnat_username()

## End(Not run)
```

`xnatR_cli` *Command-line interface for xnatR*

Description

Parses xnatR command-line arguments, dispatches to package functions, writes results to stdout, and returns a process-style integer status code.

Usage

```
xnatR_cli(args = commandArgs(trailingOnly = TRUE))
```

Arguments

`args`	Character vector of command-line arguments.

Value

Integer status code: `0` for success and `2` for usage/runtime errors.

Index