Package: openneuroR (via r-universe)

June 27, 2026

Title Programmatic Access to OpenNeuro Datasets

Version 0.1.0

Description Search, explore, and download datasets from 'OpenNeuro' <https://openneuro.org>, the largest open neuroimaging data repository. Queries the 'OpenNeuro' GraphQL API to discover datasets by modality, diagnosis, or keyword; inspect snapshots, files, and subject lists; and download full datasets or selected subsets via HTTPS, Amazon S3, or 'DataLad'. Downloaded data are cached locally so subsequent requests skip already-fetched files.

License MIT + file LICENSE

Depends R (>= 4.1.0)

URL https://bbuchsbaum.github.io/openneuroR/, https://github.com/bbuchsbaum/openneuroR

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

Imports cli (>= 3.6.0), dplyr (>= 1.1.0), fs (>= 1.6.6), httr2 (>= 1.2.1), jsonlite (>= 1.8.0), processx (>= 3.8.0), rlang (>= 1.1.0), tibble (>= 3.2.0)

Suggests knitr, pkgdown, rmarkdown, testthat (>= 3.0.0), httptest2, withr, stringi, bidser

VignetteBuilder knitr

Language en-US

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/Needs/website bbuchsbaum/albersdown

Config/pak/sysreqs cmake make libuv1-dev libssl-dev

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

Date/Publication 2026-06-26 23:43:08 UTC

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

RemoteRef HEAD

RemoteSha 11fdbfd5e6c516a4edae0bb7683a931df3b5b912

Contents

`on_bids` *Create BIDS Project from OpenNeuro Handle*

Description

Converts a fetched OpenNeuro dataset handle into a bidser `bids_project` object, enabling BIDS-aware data access to subjects, sessions, files, and derivatives.

Usage

```
on_bids(handle, fmriprep = FALSE, prep_dir = "derivatives/fmriprep")
```

Arguments

handle	An openneuro_handle object, typically created with on_handle() and fetched with on_fetch(). If the handle is in "pending" state, it will be automatically fetched first.
fmriprep	Logical. If TRUE, include fMRIPrep derivatives from the default derivatives/fmriprep path. Ignored if prep_dir is specified. Default is FALSE.
prep_dir	Character. Path to derivatives directory relative to the dataset root. If specified, takes precedence over fmriprep. Default is "derivatives/fmriprep".

Details

This function provides a bridge between OpenNeuro's download system and bidser's BIDS-aware data structures. The resulting bids_project object exposes:

- Subject and session information
- BIDS file listings by modality
- Derivatives access (if available)

The bidser package is required but listed as an optional dependency (Suggests). If not installed, a helpful message guides installation.

Value

A bids_project object from the bidser package.

Derivatives Handling

When fmriprep = TRUE, the function looks for derivatives at derivatives/fmriprep within the dataset. You can specify a custom derivatives path with prep_dir.

If prep_dir is set to a non-default value, it takes precedence over fmriprep = TRUE. A warning is issued if the requested derivatives path does not exist.

See Also

on_handle() to create a handle, on_fetch() to download data.

Examples

```
## Not run:
# Basic usage
handle <- on_handle("ds000001")
handle <- on_fetch(handle)
bids <- on_bids(handle)

# Auto-fetch if needed
handle <- on_handle("ds000002")
bids <- on_bids(handle)  # Fetches automatically

# Include fMRIPrep derivatives
```

```
bids <- on_bids(handle, fmriprep = TRUE)

# Custom derivatives path
bids <- on_bids(handle, prep_dir = "derivatives/custom-pipeline")

## End(Not run)
```

`on_cache_clear` *Clear Cache*

Description

Removes cached datasets. Can clear a specific dataset or all cached data.

Usage

```
on_cache_clear(dataset_id = NULL, confirm = interactive())
```

Arguments

`dataset_id`	Dataset identifier to clear (e.g., "ds000001"), or NULL to clear all cached datasets.
`confirm`	If TRUE (default in interactive sessions), asks for confirmation before clearing. Set FALSE to skip confirmation.

Value

Invisibly returns the number of datasets cleared.

Examples

```
## Not run:
# Clear specific dataset (with confirmation)
on_cache_clear("ds000001")

# Clear specific dataset without confirmation
on_cache_clear("ds000001", confirm = FALSE)

# Clear all cached datasets
on_cache_clear()

## End(Not run)
```

on_cache_info *Get Cache Information*

Description

Returns information about the openneuroR cache location and total size.

Usage

```
on_cache_info()
```

Value

A list with:

cache_path Path to cache directory

n_datasets Number of cached datasets

total_size Total size in bytes

size_formatted Human-readable total size (e.g., "5.3 GB")

Examples

```
## Not run:
# Get cache info
info <- on_cache_info()
info$cache_path    # Where cache is stored
info$n_datasets    # How many datasets
info$size_formatted  # Human-readable size

## End(Not run)
```

on_cache_list *List Cached Datasets*

Description

Returns a tibble of all datasets currently in the openneuroR cache.

Usage

```
on_cache_list()
```

Value

A tibble with columns:

dataset_id Dataset identifier (e.g., "ds000001")

snapshot_tag Cached snapshot version (may be NA if unknown)

n_files Number of cached files

total_size Total size in bytes

size_formatted Human-readable size (e.g., "1.2 GB")

cached_at When first cached (ISO 8601 timestamp)

type Type of cached data: "raw" for raw dataset files, "derivative" for fMRIPrep/MRIQC outputs, or "raw+derivative" if both are cached

Examples

```
## Not run:
# List all cached datasets
on_cache_list()

# Check total cache usage
cached <- on_cache_list()
sum(cached$total_size)  # total bytes

# Filter to only derivatives
cached[grepl("derivative", cached$type), ]

## End(Not run)
```

`on_client` *Create OpenNeuro API Client*

Description

Creates a client object for accessing the OpenNeuro GraphQL API. The client stores configuration including the API endpoint URL and optional authentication token.

Usage

```
on_client(url = "https://openneuro.org/crn/graphql", token = NULL)
```

Arguments

`url`	API endpoint URL. Defaults to the OpenNeuro GraphQL endpoint.
`token`	API token for authentication. Defaults to the value of the `OPENNEURO_API_KEY` environment variable, or `NULL` if not set. Authentication is optional for read-only access to public datasets.

Value

An openneuro_client object (S3 class) containing:

url The API endpoint URL

token The authentication token (or NULL)

See Also

on_request() for executing queries with the client

Examples

```
# Create client with default settings
client <- on_client()
print(client)

# Create client with custom endpoint
client <- on_client(url = "https://staging.openneuro.org/crn/graphql")
```

---

on_dataset *Get Dataset Metadata*

Description

Retrieves detailed metadata for a single OpenNeuro dataset.

Usage

```
on_dataset(id, client = NULL)
```

Arguments

id	Dataset identifier (e.g., "ds000001").
client	An openneuro_client object. If NULL, creates a default client.

Value

A tibble with one row containing:

id Dataset identifier

name Dataset title

created Timestamp when dataset was created (POSIXct)

public Whether the dataset is publicly accessible (logical)

latest_snapshot Tag of the most recent snapshot (if any)

See Also

on_search() to find datasets, on_snapshots() for version history

Examples

```
## Not run:
# Get metadata for a specific dataset
ds <- on_dataset("ds000001")
print(ds)

# Access fields
ds$name
ds$created

## End(Not run)
```

---

on_derivatives *Discover Derivative Datasets*

Description

Finds derivative datasets (fMRIPrep, MRIQC, etc.) available for an OpenNeuro dataset. Searches both embedded derivatives within the dataset and external derivatives from the OpenNeuroDerivatives GitHub organization.

Usage

```
on_derivatives(
  dataset_id,
  sources = c("embedded", "openneuro-derivatives"),
  refresh = FALSE,
  client = NULL
)
```

Arguments

dataset_id	Dataset identifier (e.g., "ds000102").
sources	Character vector specifying which sources to check. Default is c("embedded", "openneuro-derivatives") to check both. Use "embedded" for derivatives stored within the dataset, or "openneuro-derivatives" for external derivatives from GitHub.
refresh	If TRUE, bypass cache and fetch fresh data from APIs. Default is FALSE to use cached results when available.
client	An openneuro_client object for embedded derivative checks. If NULL (default), creates a default client.

Details

Derivative Sources:

Embedded derivatives are stored directly within the dataset's BIDS structure in a `derivatives/` subdirectory. These are typically provided by the dataset authors.

OpenNeuroDerivatives are externally processed derivatives maintained by the OpenNeuro team, available from the OpenNeuroDerivatives GitHub organization. These are stored on S3 and can be downloaded separately.

Source Preference:

When the same pipeline exists in both sources, embedded derivatives are preferred and the OpenNeuroDerivatives entry is removed from results. This follows the principle that author-provided derivatives should take precedence.

Caching:

Results are cached per-session to minimize API calls. Use `refresh = TRUE` to bypass the cache and fetch fresh data.

Value

A tibble with one row per available derivative, containing:

dataset_id The dataset identifier

pipeline Pipeline name (e.g., "fmriprep", "mriqc")

source Where the derivative is from: "embedded" or "openneuro-derivatives"

version Pipeline version (NA if not available)

n_subjects Number of subjects processed (NA if not available)

n_files Number of derivative files (NA if not available)

total_size Human-readable size (e.g., "2.3 GB", NA if not available)

last_modified Last modification time (POSIXct, NA if not available)

s3_url S3 URL for OpenNeuroDerivatives sources (NA for embedded)

Returns an empty tibble with the same structure if no derivatives are found.

See Also

`on_files()` for listing files within datasets

Examples

```
## Not run:
# Find all derivatives for a dataset
derivs <- on_derivatives("ds000102")
print(derivs)

# Check only OpenNeuroDerivatives (GitHub)
github_derivs <- on_derivatives("ds000102", sources = "openneuro-derivatives")
```

```
# Check only embedded derivatives
embedded_derivs <- on_derivatives("ds000102", sources = "embedded")

# Force refresh of cached data
fresh_derivs <- on_derivatives("ds000102", refresh = TRUE)

# Filter for fMRIPrep derivatives
fmriprep <- derivs[derivs$pipeline == "fmriprep", ]

## End(Not run)
```

---

on_doctor *OpenNeuro Backend Diagnostics*

Description

Reports the status of all available download backends, showing which are installed, their versions, and readiness for use.

Usage

```
on_doctor()
```

Value

Invisibly returns an object of class `openneuro_doctor` containing:

https List with available (always TRUE), version (NA)

s3 List with available (logical), version (character or NA)

datalad List with available (logical), version (character or NA)

Examples

```
on_doctor()
```

---

on_download *Download OpenNeuro Dataset*

Description

Downloads files from an OpenNeuro dataset to local disk. Supports downloading the full dataset, specific files, files matching a regex pattern, or specific subjects.

Usage

```
on_download(
  id,
  tag = NULL,
  files = NULL,
  subjects = NULL,
  include_derivatives = TRUE,
  dest_dir = NULL,
  use_cache = TRUE,
  quiet = FALSE,
  verbose = FALSE,
  force = FALSE,
  backend = NULL,
  client = NULL
)
```

Arguments

id	Dataset identifier (e.g., "ds000001").
tag	Snapshot version tag. If NULL (default), uses latest snapshot.
files	Character vector of specific files to download, or a single regex pattern (detected by presence of regex metacharacters). If NULL (default), downloads all files.
subjects	Character vector of subject IDs (e.g., `c("sub-01", "sub-02")`) or a regex pattern wrapped in `regex()` (e.g., `regex("sub-0[1-5]")`). Subject IDs can be specified with or without the "sub-" prefix. If NULL (default), downloads all subjects.
include_derivatives	If TRUE (default) and `subjects` is specified, also include derivative outputs for matching subjects from the `derivatives/` directory.
dest_dir	Destination directory. If NULL (default) and `use_cache` is TRUE, downloads to cache location. If NULL and `use_cache` is FALSE, creates `./dataset_id/` in the current working directory.
use_cache	If TRUE (default) and dest_dir is NULL, downloads to CRAN-compliant cache location. Set FALSE to use current working directory. Ignored when dest_dir is explicitly provided.
quiet	If TRUE, suppress all progress output. Default FALSE.
verbose	If TRUE, show per-file progress in addition to overall progress. Default FALSE.
force	If TRUE, re-download files even if they exist with correct size. Default FALSE.
backend	Backend to use for downloading: "datalad", "s3", or "https". If NULL (default), auto-selects best available backend with priority: DataLad > S3 > HTTPS. DataLad provides git-annex integrity verification, S3 uses AWS CLI for fast parallel sync, HTTPS is the universal fallback.
client	An openneuro_client object. If NULL, creates default client.

Details

By default, files are downloaded to a CRAN-compliant cache location (platform-specific, see Details). Repeat downloads of the same files are skipped automatically based on manifest tracking.

Cache locations by platform:

- Mac: ~/Library/Caches/R/openneuroR
- Linux: ~/.cache/R/openneuroR
- Windows: ~/AppData/Local/R/cache/openneuroR

Each dataset is stored in a subdirectory by dataset ID. A manifest.json file tracks downloaded files, enabling automatic skip of already-cached files on repeat downloads.

Backend selection:

- **DataLad**: Clones from OpenNeuroDatasets GitHub with git-annex. Provides cryptographic integrity verification. Requires `datalad` and `git-annex` CLI tools.
- **S3**: Uses AWS CLI `s3 sync` for fast parallel downloads. Requires `aws` CLI tool.
- **HTTPS**: Direct file downloads via httr2. Always available, no external dependencies.

Subject filtering:

When `subjects` is specified, only files belonging to those subjects are downloaded, plus root-level files (e.g., `dataset_description.json`, `participants.tsv`). Subject IDs can be provided with or without the "sub-" prefix - both `"01"` and `"sub-01"` work.

For pattern matching, wrap the pattern in `regex()`. Patterns are auto-anchored for full subject ID matching, so `regex("sub-01")` will match "sub-01" but not "sub-010".

Value

Invisibly returns a list with:

downloaded Number of files downloaded

skipped Number of files skipped (already cached or existed)

failed Character vector of failed file names

total_bytes Total bytes downloaded

dest_dir Path to destination directory

backend Backend used for download (if S3 or DataLad)

Examples

```
## Not run:
# Download to cache (default - auto-selects best backend)
on_download("ds000001", files = "participants.tsv")

# Repeat download skips cached files
result <- on_download("ds000001", files = "participants.tsv")
result$skipped  # >= 1 (files already in cache)

# Download to specific directory (bypasses cache)
```

```
on_download("ds000001", dest_dir = "~/data/openneuro")

# Download to current working directory
on_download("ds000001", use_cache = FALSE)

# Force re-download of cached files
on_download("ds000001", force = TRUE)

# Use specific backend
on_download("ds000001", backend = "s3")
on_download("ds000001", backend = "https")  # Force HTTPS

# Download specific subjects
on_download("ds000001", subjects = c("sub-01", "sub-02"))

# Download subjects matching pattern
on_download("ds000001", subjects = regex("sub-0[1-5]"))

# Download subjects without derivatives
on_download("ds000001", subjects = c("01", "02"), include_derivatives = FALSE)

## End(Not run)
```

`on_download_derivatives` *Download Derivative Dataset*

Description

Downloads fMRIPrep, MRIQC, or other derivative outputs from OpenNeuro datasets. Supports filtering by subject, output space, and BIDS suffix. Uses S3 backend (openneuro-derivatives bucket) with HTTPS fallback.

Usage

```
on_download_derivatives(
  dataset_id,
  pipeline,
  subjects = NULL,
  space = NULL,
  suffix = NULL,
  dry_run = FALSE,
  dest_dir = NULL,
  use_cache = TRUE,
  quiet = FALSE,
  verbose = FALSE,
  force = FALSE,
  backend = NULL,
  client = NULL
)
```

Arguments

dataset_id	Dataset identifier (e.g., "ds000001").
pipeline	Pipeline name (e.g., "fmriprep", "mriqc").
subjects	Character vector of subject IDs (e.g., c("sub-01", "sub-02")) or a regex pattern wrapped in regex() (e.g., regex("sub-0[1-5]")). Subject IDs can be specified with or without the "sub-" prefix. If NULL (default), downloads all subjects.
space	Character string: output space to filter by (e.g., "MNI152NLin2009cAsym", "fsaverage", "T1w"). If NULL (default), downloads all spaces. Matching is exact (specify full space name). Files without a _space- entity (native space) are always included.
suffix	Character vector of BIDS suffixes to filter by (e.g., c("bold", "T1w", "mask")). If NULL (default), downloads all suffixes. Files without a clear suffix (metadata files) are always included.
dry_run	If TRUE, returns a tibble of files that would be downloaded without actually downloading them. Default is FALSE.
dest_dir	Destination directory. If NULL (default) and use_cache is TRUE, downloads to BIDS-compliant cache location: {cache}/{dataset_id}/derivatives/{pipeline}/.
use_cache	If TRUE (default) and dest_dir is NULL, downloads to CRAN-compliant cache location. Set FALSE to use current working directory.
quiet	If TRUE, suppress all progress output. Default FALSE.
verbose	If TRUE, show per-file progress in addition to overall progress. Default FALSE.
force	If TRUE, re-download files even if they exist with correct size. Default FALSE.
backend	Backend to use for downloading: "s3" or "https". If NULL (default), auto-selects S3 for openneuro-derivatives bucket.
client	An openneuro_client object. If NULL, creates default client.

Details

Filter Logic:

All filters combine with AND logic - a file must match ALL specified filters to be included. For example, subjects = "sub-01", space = "MNI152NLin2009cAsym" downloads only sub-01's MNI-space files.

Cache Structure:

Derivatives are cached in BIDS-compliant structure: {cache_root}/{dataset_id}/derivatives/{pipeline}/ This keeps derivatives organized alongside raw data while maintaining clear separation by pipeline.

Backend Selection:

S3 backend is preferred for the openneuro-derivatives bucket as it provides fast parallel sync. HTTPS fallback is used if S3 is unavailable.

Space Matching:

Space matching is exact - specify the full space name (e.g., "MNI152NLin2009cAsym", not "MNI"). Files without a _space- entity (native/T1w space per BIDS convention) are always included when filtering by space.

Value

If `dry_run = TRUE`, returns a tibble with columns:

path Relative path within derivative

size File size in bytes

size_formatted Human-readable size (e.g., "1.2 GB")

dest_path Full destination path where file would be downloaded

If `dry_run = FALSE`, invisibly returns a list with:

downloaded Number of files downloaded

skipped Number of files skipped (already cached)

failed Character vector of failed file names

total_bytes Total bytes downloaded

dest_dir Path to destination directory

backend Backend used for download

See Also

`on_derivatives()` to discover available derivatives, `on_spaces()` to discover available output spaces, `on_download()` to download raw datasets

Examples

```
## Not run:
# Download all fMRIPrep derivatives for a dataset
on_download_derivatives("ds000001", "fmriprep")

# Download specific subjects
on_download_derivatives("ds000001", "fmriprep",
                        subjects = c("sub-01", "sub-02"))

# Download only MNI-space outputs
on_download_derivatives("ds000001", "fmriprep",
                        space = "MNI152NLin2009cAsym")

# Download only BOLD and mask files
on_download_derivatives("ds000001", "fmriprep",
                        suffix = c("bold", "mask"))

# Preview files without downloading
files <- on_download_derivatives("ds000001", "fmriprep",
                                 subjects = "sub-01",
                                 space = "MNI152NLin2009cAsym",
                                 dry_run = TRUE)
print(files)

# Combine all filters
on_download_derivatives("ds000001", "fmriprep",
```

```
                        subjects = regex("sub-0[1-5]"),
                        space = "MNI152NLin2009cAsym",
                        suffix = c("bold", "T1w"))

## End(Not run)
```

`on_fetch` *Fetch Handle (Materialize Download)*

Description

Materializes a lazy handle by downloading the referenced dataset. If the handle is already in "ready" state, returns it unchanged unless `force = TRUE`.

Usage

```
on_fetch(handle, ...)

## S3 method for class 'openneuro_handle'
on_fetch(handle, quiet = FALSE, force = FALSE, ...)
```

Arguments

`handle`	An object to fetch. For `openneuro_handle` objects, triggers the download.
`...`	Additional arguments passed to methods.
`quiet`	If TRUE, suppress progress output during download.
`force`	If TRUE, re-download even if handle is already "ready".

Value

The handle with updated state. For `openneuro_handle`, returns the handle with `state = "ready"`, `path` set to the download location, and `fetch_time` set to current time.

Important

You must capture the return value! S3 objects have copy semantics:

```
# CORRECT
handle <- on_fetch(handle)

# WRONG - changes are lost
on_fetch(handle)
```

See Also

`on_handle()` to create a handle, `on_path()` to get path.

Examples

```
## Not run:
handle <- on_handle("ds000001", files = "participants.tsv")
handle <- on_fetch(handle)  # Downloads now
handle$state  # "ready"

## End(Not run)
```

`on_files` *List Files in a Snapshot*

Description

Lists all files in a dataset snapshot. Can list the root directory or drill into subdirectories using the `tree` parameter.

Usage

```
on_files(id, tag = NULL, tree = NULL, client = NULL)
```

Arguments

`id`	Dataset identifier (e.g., "ds000001").
`tag`	Snapshot version tag (e.g., "1.0.0"). If `NULL` (default), uses the most recent snapshot.
`tree`	Subdirectory token for listing nested files. Use the `id` column from a previous call to explore subdirectories. Default `NULL` lists the root directory.
`client`	An `openneuro_client` object. If `NULL`, creates a default client.

Details

OpenNeuro stores datasets using git-annex, where large files are stored separately from the git repository. The `annexed` column indicates which files use this storage method.

To explore a directory structure:

1. Call `on_files()` to get the root listing
2. Filter for `directory == TRUE` entries
3. Use the `id` from a directory to call `on_files(tree = id)`

Value

A tibble with columns:

filename Name of the file or directory

size File size in bytes (numeric), may be NA for directories

directory TRUE if this entry is a directory (logical)

annexed TRUE if file is stored in git-annex (logical). Annexed files are typically larger and require special download handling.

id Unique identifier for this entry. Pass it as the `tree` argument to explore a subdirectory.

urls List column of direct HTTPS download URLs for the entry (character vector, empty for directories).

key Backward-compatible alias of `id` (the directory tree token).

Returns an empty tibble with the same column structure if the snapshot has no files.

See Also

`on_snapshots()` to list available snapshots

Examples

```
## Not run:
# List root files using latest snapshot
files <- on_files("ds000001")
print(files)

# List files in a specific snapshot
files <- on_files("ds000001", tag = "1.0.0")

# Explore a subdirectory
dirs <- files[files$directory, ]
if (nrow(dirs) > 0) {
  subfiles <- on_files("ds000001", tree = dirs$id[1])
  print(subfiles)
}

# Find all annexed (large) files
annexed_files <- files[files$annexed & !files$directory, ]

## End(Not run)
```

---

on_handle *Create Lazy Handle to OpenNeuro Dataset*

Description

Creates a lazy handle that references an OpenNeuro dataset without triggering an immediate download. The handle can be fetched later when the data is actually needed.

Usage

```
on_handle(dataset_id, tag = NULL, files = NULL, backend = NULL)
```

Arguments

dataset_id	Dataset identifier (e.g., "ds000001").
tag	Snapshot version tag. If NULL, uses latest snapshot when fetched.
files	Character vector of specific files to download when fetched, or a regex pattern. If NULL, downloads all files when fetched.
backend	Backend to use when fetching: "datalad", "s3", or "https". If NULL, auto-selects best available backend.

Details

Handles support a lazy evaluation pattern:

1. Create handle with `on_handle()` - no download occurs
2. Fetch data with `on_fetch()` - download happens here
3. Get path with `on_path()` - returns filesystem path

This is useful for pipelines where dataset references need to be defined early but data should only be downloaded when needed.

Value

An S3 object of class `openneuro_handle` with state "pending".

Important

S3 objects have copy semantics. You must capture the return value of `on_fetch()`:

```
# WRONG - handle not updated
on_fetch(handle)
handle$state  # Still "pending"!

# CORRECT - capture returned handle
handle <- on_fetch(handle)
handle$state  # Now "ready"
```

See Also

`on_fetch()` to materialize the download, `on_path()` to get path.

Examples

```
## Not run:
# Create lazy handle - no download yet
handle <- on_handle("ds000001", files = "participants.tsv")
print(handle)  # Shows state: pending

# Fetch when data is needed
handle <- on_fetch(handle)
print(handle)  # Shows state: ready
```

```
# Get filesystem path
path <- on_path(handle)

## End(Not run)
```

---

on_path *Get Path from Handle*

Description

Returns the filesystem path for a fetched handle. Raises an error if the handle has not been fetched yet.

Usage

```
on_path(handle)

## S3 method for class 'openneuro_handle'
on_path(handle)
```

Arguments

handle	An object to get the path from. For openneuro_handle objects, returns the download location.

Value

Character string with the filesystem path.

See Also

on_handle() to create a handle, on_fetch() to materialize.

Examples

```
## Not run:
handle <- on_handle("ds000001")
handle <- on_fetch(handle)
path <- on_path(handle)
list.files(path)

## End(Not run)
```

`on_request` *Execute GraphQL Query*

Description

Executes a GraphQL query against the OpenNeuro API. Handles authentication, retry logic, rate limiting, and error handling.

Usage

```
on_request(query, variables = NULL, client = NULL)
```

Arguments

`query`	A GraphQL query string.
`variables`	A named list of variables to pass to the query.
`client`	An `openneuro_client` object. If `NULL`, creates a default client.

Details

The function implements several reliability features:

- Automatic retry on transient errors (429, 500, 502, 503)
- Rate limiting (10 requests per minute)
- User-Agent header for API identification
- Bearer token authentication when available

GraphQL errors (returned with HTTP 200 status) are detected and raised as R errors with class `openneuro_api_error`.

Value

The `data` field from the GraphQL response.

See Also

`on_client()` for creating client objects

Examples

```
## Not run:
# Execute a simple query
query <- "query { datasets(first: 1) { edges { node { id } } } }"
result <- on_request(query)

## End(Not run)
```

`on_search` *Search OpenNeuro Datasets*

Description

Searches the OpenNeuro database for datasets. When a text query is provided, uses the search endpoint if available. Otherwise lists datasets with optional filtering.

Usage

```
on_search(
  query = NULL,
  modality = NULL,
  limit = 50,
  all = FALSE,
  client = NULL
)
```

Arguments

`query`	Text query to search for. Note: The OpenNeuro search API may have limited availability. If search returns no results, consider using `query = NULL` with `modality` filter instead.
`modality`	Filter by modality (e.g., "MRI", "EEG", "MEG", "iEEG", "PET"). Case-insensitive matching is attempted.
`limit`	Maximum number of results to return per page (default 50).
`all`	If `TRUE`, paginate through all matching results. If `FALSE` (default), return only the first page.
`client`	An `openneuro_client` object. If `NULL`, creates a default client.

Value

A tibble with columns:

id Dataset identifier (e.g., "ds000001")

name Dataset title

created Timestamp when dataset was created (POSIXct)

public Whether the dataset is publicly accessible (logical)

modalities List of modalities in the dataset

n_subjects Number of subjects in the dataset

tasks List of tasks in the dataset

Returns an empty tibble with the same column structure if no matches found.

See Also

on_dataset() for detailed metadata on a single dataset

Examples

```
## Not run:
# List datasets (most reliable)
results <- on_search(limit = 10)

# Filter by modality
mri_datasets <- on_search(modality = "MRI", limit = 25)
eeg_datasets <- on_search(modality = "EEG", limit = 25)

# Text search (may have limited availability)
results <- on_search("visual cortex", limit = 10)

# Get all datasets (may be slow)
all_datasets <- on_search(all = TRUE)

## End(Not run)
```

---

on_snapshots *List Dataset Snapshots*

Description

Retrieves all snapshots (versioned releases) for a dataset. Snapshots are immutable versions of the dataset that can be referenced by tag.

Usage

```
on_snapshots(id, client = NULL)
```

Arguments

id	Dataset identifier (e.g., "ds000001").
client	An openneuro_client object. If NULL, creates a default client.

Value

A tibble with columns:

tag Snapshot version tag (e.g., "1.0.0")

created Timestamp when snapshot was created (POSIXct)

size Total size of the snapshot in bytes (numeric)

Rows are ordered with most recent snapshot first. Returns an empty tibble with the same column structure if the dataset has no snapshots.

See Also

on_files() to list files in a snapshot, on_dataset() for metadata

Examples

```
## Not run:
# List all snapshots for a dataset
snaps <- on_snapshots("ds000001")
print(snaps)

# Get the latest snapshot tag
latest_tag <- snaps$tag[1]

# Calculate total size in GB
snaps$size_gb <- snaps$size / (1024^3)

## End(Not run)
```

---

on_spaces *Discover Available Output Spaces*

Description

Discovers the available output spaces (MNI152NLin2009cAsym, fsaverage, etc.) for a derivative dataset. Parses BIDS _space- entity from filenames.

Usage

```
on_spaces(derivative, refresh = FALSE, client = NULL)
```

Arguments

derivative	A single-row tibble from on_derivatives() output. Must contain columns: dataset_id, pipeline, and source.
refresh	If TRUE, bypass cache and fetch fresh data. Default is FALSE to use cached results.
client	An openneuro_client object for API calls (embedded sources). If NULL (default), creates a default client.

Details

Space Discovery:
This function samples derivative files and extracts the _space-<label> entity from BIDS-formatted filenames. It does NOT infer T1w from files without a space entity (per BIDS convention, native space files may omit the space entity).

Source Handling:

- **embedded**: Uses the OpenNeuro API to list files in the `derivatives/{pipeline}/` directory.
- **openneuro-derivatives**: Uses AWS CLI to list files from the `s3://openneuro-derivatives/` bucket.

Caching:

Results are cached per-session to minimize API/S3 calls. Use `refresh = TRUE` to bypass the cache.

Value

A character vector of space names, sorted alphabetically. Common spaces include:

- Volumetric: MNI152NLin2009cAsym, MNI152NLin6Asym, T1w
- Surface: fsaverage, fsaverage5, fsaverage6, fsnative

Returns `character(0)` with a warning if no spaces are found.

See Also

`on_derivatives()` to discover available derivative datasets

Examples

```
## Not run:
# First, get available derivatives for a dataset
derivs <- on_derivatives("ds000102")
print(derivs)

# Then get spaces for the first derivative
spaces <- on_spaces(derivs[1, ])
print(spaces)
# Example output: c("MNI152NLin2009cAsym", "fsaverage")

# Force refresh of cached spaces
spaces <- on_spaces(derivs[1, ], refresh = TRUE)

## End(Not run)
```

---

on_subjects *List Subjects in a Dataset*

Description

Returns the subject IDs present in a dataset snapshot without downloading any data. This is a metadata-only query using the OpenNeuro GraphQL API.

Usage

```
on_subjects(id, tag = NULL, client = NULL)
```

Arguments

id	Dataset identifier (e.g., "ds000001").
tag	Snapshot version tag (e.g., "1.0.0"). If NULL (default), uses the most recent snapshot.
client	An openneuro_client object. If NULL, creates a default client.

Details

Subject IDs are returned in natural sort order, so "sub-10" comes after "sub-9" rather than after "sub-1".

The n_sessions and n_files columns provide dataset-level context. Per-subject session and file counts are not available from the OpenNeuro API.

Value

A tibble with columns:

dataset_id The dataset identifier

subject_id Subject identifier (e.g., "sub-01")

n_sessions Number of sessions in the dataset (same for all rows)

n_files Estimated files per subject (same for all rows)

Returns an empty tibble with the same column structure if the dataset has no BIDS subjects (e.g., non-BIDS datasets).

See Also

on_files() to list files, on_download() to download data

Examples

```
## Not run:
# List subjects in a dataset
subjects <- on_subjects("ds000001")
print(subjects)

# List subjects in a specific snapshot
subjects <- on_subjects("ds000001", tag = "1.0.0")

# Get subject count
nrow(subjects)

## End(Not run)
```

print.openneuro_doctor

Print Method for OpenNeuro Doctor

Description

Displays styled CLI output showing backend availability and versions.

Usage

```
## S3 method for class 'openneuro_doctor'
print(x, ...)
```

Arguments

x	An openneuro_doctor object.
...	Additional arguments (ignored).

Value

x invisibly.

print.openneuro_handle

Print Method for OpenNeuro Handle

Description

Print Method for OpenNeuro Handle

Usage

```
## S3 method for class 'openneuro_handle'
print(x, ...)
```

Arguments

x	An openneuro_handle object.
...	Additional arguments (ignored).

Value

x invisibly.

regex *Mark String as Regex Pattern for Subject Filtering*

Description

Creates a regex pattern object for use with the `subjects` parameter in `on_download()`. Patterns are auto-anchored to match complete subject IDs.

Usage

```
regex(pattern)
```

Arguments

`pattern`	A single non-empty character string containing a regex pattern.

Value

A character vector with class `c("on_regex", "character")`.

See Also

`on_download()` for downloading with subject filters

Examples

```
# Match subjects sub-01 through sub-05
regex("sub-0[1-5]")

# Match any subject starting with sub-1
regex("sub-1.*")

## Not run:
# Use in on_download()
on_download("ds000001", subjects = regex("sub-0[1-5]"))

## End(Not run)
```

Index