

Package: templateflow (via r-universe)

June 3, 2026

Type Package

Title Client for the 'TemplateFlow' Archive

Version 0.1.0

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

Description R client for querying and fetching resources from the TemplateFlow archive using a local cache backed by the public TemplateFlow S3 bucket. Provides functions to list template assets, download missing files, read template metadata, and retrieve citations.

Depends R (>= 4.1.0)

License Apache License (>= 2)

URL <https://github.com/bbuchsbaum/templateflow>

BugReports <https://github.com/bbuchsbaum/templateflow/issues>

Encoding UTF-8

Imports cli, httr2 (>= 1.0.0), jsonlite, rappdirs, rlang

Suggests aws.s3, bidser, gifti, knitr, oro.nifti, rmarkdown, RNifti, testthat (>= 3.0.0), tibble, withr

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.3

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-03-05 18:09:16 UTC

RemoteUrl <https://github.com/bbuchsbaum/templateflow>

RemoteRef HEAD

RemoteSha 4aff5651c9035b1bde6f1a7c2297ef67ffae590a

Contents

print.TemplateFlowClient	2
TemplateFlowClient	3
tf_cache_refresh	3
tf_cache_scan	4
tf_cache_stats	5
tf_cache_update	5
tf_cache_wipe	6
tf_cli	7
tf_client	7
tf_datalad_available	8
tf_datalad_get	8
tf_datalad_install	9
tf_datalad_update	9
tf_entities	10
tf_file_extension	10
tf_get	11
tf_get_citations	12
tf_get_metadata	12
tf_ls	13
tf_prefetch	14
tf_read_file	14
tf_set_client	15
tf_templates	15
Index	17

```
print.TemplateFlowClient
    Print a TemplateFlowClient
```

Description

Print a TemplateFlowClient

Usage

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

Arguments

x	A 'TemplateFlowClient' object.
...	Additional arguments (ignored).

Value

'x', invisibly.

TemplateFlowClient *TemplateFlow R client*

Description

Create a client for querying the TemplateFlow archive.

Usage

```
TemplateFlowClient(root = NULL, cache = NULL, ...)
```

Arguments

root	Optional cache root (defaults to ‘~/cache/templateflow’ or ‘TEMPLATEFLOW_HOME’).
cache	Optional pre-built ‘TemplateFlowCache’.
...	Additional config passed to ‘tf_default_config()’.

Value

A ‘TemplateFlowClient’ object.

Examples

```
## Not run:
client <- TemplateFlowClient()
client <- TemplateFlowClient(root = "/tmp/tf-cache")

## End(Not run)
```

tf_cache_refresh *Refresh cache layout*

Description

Rebuild the layout index without downloading any files.

Usage

```
tf_cache_refresh(cache = NULL)
```

Arguments

cache	A ‘TemplateFlowCache’ object or a character path to the cache root (optional; uses default client cache).
-------	---

Value

The updated cache (invisibly), or a ‘TemplateFlowLayout‘ if a character path was supplied.

Examples

```
## Not run:
tf_cache_refresh()

## End(Not run)
```

tf_cache_scan	<i>Scan cache for problematic files</i>
---------------	---

Description

Find zero-byte stubs and S3 XML error responses in the cache.

Usage

```
tf_cache_scan(cache = NULL)
```

Arguments

cache A ‘TemplateFlowCache‘ object or a character path to the cache root (optional; uses default client cache).

Value

A list with ‘zero‘ (zero-byte paths) and ‘xml‘ (XML error paths).

Examples

```
## Not run:
tf_cache_scan()

## End(Not run)
```

tf_cache_stats	<i>Cache statistics</i>
----------------	-------------------------

Description

Report the number and total size of cached files.

Usage

```
tf_cache_stats(cache = NULL)
```

Arguments

cache	A 'TemplateFlowCache' object or a character path to the cache root (optional; uses default client cache).
-------	---

Value

A list with 'cache_files', 'cache_bytes', and 'cache_size_human'.

Examples

```
## Not run:
tf_cache_stats()

## End(Not run)
```

tf_cache_update	<i>Update template cache</i>
-----------------	------------------------------

Description

Re-extract the skeleton archive, optionally fetching the latest version from the remote repository.

Usage

```
tf_cache_update(cache = NULL, local = FALSE, overwrite = TRUE, silent = FALSE)
```

Arguments

cache	A 'TemplateFlowCache' object (optional; uses default client cache).
local	If 'TRUE', only use the bundled skeleton (no network).
overwrite	If 'TRUE', overwrite existing stub files.
silent	If 'TRUE', suppress progress messages.

Value

The updated cache (invisibly).

Examples

```
## Not run:  
tf_cache_update()  
tf_cache_update(local = TRUE)  
  
## End(Not run)
```

tf_cache_wipe	<i>Wipe the template cache</i>
---------------	--------------------------------

Description

Delete the entire local template cache directory. Not supported when using the DataLad backend (use ‘datalad drop’ instead).

Usage

```
tf_cache_wipe(cache = NULL)
```

Arguments

cache A ‘TemplateFlowCache’ object (optional; uses default client cache).

Value

The cache object (invisibly).

Examples

```
## Not run:  
tf_cache_wipe()  
  
## End(Not run)
```

tf_cli	<i>TemplateFlow CLI</i>
--------	-------------------------

Description

Command-line interface for the TemplateFlow archive. Supports subcommands: 'config', 'ls', 'get', 'wipe', 'update', 'meta', 'cite', 'refresh', 'doctor'.

Usage

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

Arguments

args Character vector of CLI-style arguments (defaults to commandArgs).

Value

Invisible result of the requested command.

Examples

```
## Not run:
tf_cli(c("ls", "MNI152Lin", "--suffix", "T1w"))
tf_cli(c("config"))
tf_cli(c("meta", "MNI152Lin"))
tf_cli(c("doctor", "--fix"))

## End(Not run)
```

tf_client	<i>Get or create the default global TemplateFlow client</i>
-----------	---

Description

Get or create the default global TemplateFlow client

Usage

```
tf_client()
```

Value

A 'TemplateFlowClient'.

tf_datalad_available *Check if DataLad is available*

Description

Tests whether the 'datalad' command-line tool is installed and accessible.

Usage

```
tf_datalad_available()
```

Value

TRUE if datalad is available, FALSE otherwise.

tf_datalad_get *Get files from a DataLad dataset*

Description

Retrieve (download) specific files from a DataLad dataset.

Usage

```
tf_datalad_get(files, dataset)
```

Arguments

files	Character vector of file paths to retrieve.
dataset	Path to the DataLad dataset root.

Value

Invisible files vector.

Examples

```
## Not run:  
tf_datalad_get(c("tpl-MNI152Lin/tpl-MNI152Lin_res-01_T1w.nii.gz"),  
              dataset = "~/cache/templateflow")  
  
## End(Not run)
```

tf_datalad_install *Install a DataLad dataset*

Description

Clone/install a DataLad dataset to a local path.

Usage

```
tf_datalad_install(path, source, recursive = TRUE)
```

Arguments

path	Local destination path.
source	Dataset source URL or path.
recursive	If TRUE, install subdatasets recursively.

Value

Invisible path.

Examples

```
## Not run:  
tf_datalad_install("~/cache/templateflow",  
  source = "https://github.com/templateflow/templateflow.git")  
  
## End(Not run)
```

tf_datalad_update *Update a DataLad dataset*

Description

Fetch updates from the remote and optionally merge.

Usage

```
tf_datalad_update(dataset, recursive = TRUE, merge = TRUE)
```

Arguments

dataset	Path to the DataLad dataset root.
recursive	If TRUE, update subdatasets recursively.
merge	If TRUE, merge fetched changes.

Value

Invisible TRUE.

Examples

```
## Not run:
tf_data_lad_update("~/cache/templateflow", recursive = TRUE, merge = TRUE)

## End(Not run)
```

tf_entities	<i>List available BIDS entity names</i>
-------------	---

Description

Returns the entity names recognised by TemplateFlow for filtering template files (e.g. ‘resolution’, ‘atlas’, ‘suffix’).

Usage

```
tf_entities()
```

Value

A character vector of entity names sorted alphabetically.

Examples

```
tf_entities()
```

tf_file_extension	<i>Extract compound file extension</i>
-------------------	--

Description

Handles compound extensions like ‘.nii.gz’, ‘.surf.gii’, etc.

Usage

```
tf_file_extension(path)
```

Arguments

path File path.

Value

The file extension including leading dot.

tf_get	<i>Fetch template assets (downloading if needed)</i>
--------	--

Description

Download template files from the TemplateFlow archive and optionally read them into R objects. Like [tf_ls()], entity filters support 'NULL' values to exclude files containing that entity.

Usage

```
tf_get(client = NULL, template, raise_empty = FALSE, read = FALSE, ...)
```

Arguments

client	A 'TemplateFlowClient' (optional; default global client).
template	Template identifier (e.g., "MNI152Lin").
raise_empty	If 'TRUE', error when no files match the query.
read	If 'TRUE', read downloaded files into R objects using appropriate readers (NIfTI via RNifti, GIFTI via gifti, JSON, TSV). Requires optional packages for neuroimaging formats.
...	Entity filters (resolution, suffix, atlas, desc, hemi, space, density, label, segmentation, cohort, scale, roi, extension, etc.). Pass 'NULL' to exclude files containing that entity.

Value

Path string, character vector, or R object(s) if 'read = TRUE'. When a single file matches, a scalar path (or single R object) is returned; when multiple files match, a character vector (or list of R objects).

Examples

```
## Not run:  
path <- tf_get(template = "MNI152Lin", resolution = 1, suffix = "T1w")  
img <- tf_get(template = "MNI152Lin", resolution = 1, suffix = "T1w", read = TRUE)  
  
## End(Not run)
```

tf_get_citations	<i>Retrieve template citations</i>
------------------	------------------------------------

Description

Retrieve template citations

Usage

```
tf_get_citations(client = NULL, template, bibtex = FALSE)
```

Arguments

client	A 'TemplateFlowClient' (optional; default global client).
template	Template identifier (e.g., "MNI152Lin").
bibtex	If TRUE, fetch citations in BibTeX format.

Value

Character vector of citation URLs or BibTeX entries.

Examples

```
## Not run:
tf_get_citations(template = "MNI152Nlin2009cAsym")
tf_get_citations(template = "MNI152Nlin2009cAsym", bibtex = TRUE)

## End(Not run)
```

tf_get_metadata	<i>Read template metadata</i>
-----------------	-------------------------------

Description

Read template metadata

Usage

```
tf_get_metadata(client = NULL, template)
```

Arguments

client	A 'TemplateFlowClient' (optional; default global client).
template	Template identifier (e.g., "MNI152Lin").

Value

List parsed from ‘template_description.json’.

Examples

```
## Not run:
meta <- tf_get_metadata(template = "MNI152Lin")
meta$Name

## End(Not run)
```

tf_ls	<i>List template assets</i>
-------	-----------------------------

Description

Query the TemplateFlow cache for files matching a template and optional entity filters.

Usage

```
tf_ls(client = NULL, template, as_df = FALSE, ...)
```

Arguments

client	A ‘TemplateFlowClient’ (optional; default global client).
template	Template identifier (e.g., “MNI152Lin”).
as_df	If ‘TRUE’, return a data frame instead of file paths.
...	Entity filters (resolution, suffix, atlas, desc, hemi, space, density, label, segmentation, cohort, scale, roi, extension, etc.). Pass ‘NULL’ to exclude files containing that entity.

Details

Entity filters are passed via ‘...’. Use ‘NULL’ to *exclude* an entity — for example, ‘desc = NULL’ returns only files that do **not** have a ‘desc-’ key in their filename. This matches the Python client’s behaviour where ‘desc=None’ filters out files with a description tag.

Value

Character vector of file paths, or a data frame if ‘as_df = TRUE’.

Examples

```
## Not run:
tf_ls(template = "MNI152Lin", resolution = 1, suffix = "T1w")
tf_ls(template = "MNI152Lin", extension = ".nii.gz")
tf_ls(template = "MNI152Lin", as_df = TRUE)

## End(Not run)
```

tf_prefetch	<i>Batch prefetch template files</i>
-------------	--------------------------------------

Description

Downloads all zero-byte skeleton stubs matching the given filters.

Usage

```
tf_prefetch(client = NULL, templates = NULL, ...)
```

Arguments

client	A ‘TemplateFlowClient‘ (optional; default global client).
templates	Character vector of template names to prefetch. If NULL, prefetches all templates.
...	Entity filters passed to ‘tf_ls()‘.

Value

Invisible count of files downloaded.

Examples

```
## Not run:
tf_prefetch(templates = "MNI152Lin")
tf_prefetch()

## End(Not run)
```

tf_read_file	<i>Read a template file into an R object</i>
--------------	--

Description

Dispatches by file extension to the appropriate reader. NIfTI files use ‘RNifti‘ or ‘oro.nifti‘ (soft deps), GIFTI files use ‘gifti‘, JSON files use ‘jsonlite‘, and TSV files use ‘utils::read.delim()‘.

Usage

```
tf_read_file(path)
```

Arguments

path	Path to the file.
------	-------------------

Value

An R object appropriate for the file type, or the path if no reader is available.

tf_set_client	<i>Set global client</i>
---------------	--------------------------

Description

Set global client

Usage

```
tf_set_client(client)
```

Arguments

client A ‘TemplateFlowClient’.

Value

The client (invisibly).

Examples

```
## Not run:
client <- TemplateFlowClient()
tf_set_client(client)

## End(Not run)
```

tf_templates	<i>List available templates</i>
--------------	---------------------------------

Description

List available templates

Usage

```
tf_templates(client = NULL, ...)
```

Arguments

client A ‘TemplateFlowClient’ (optional; default global client).
 ... Entity filters (resolution, suffix, atlas, desc, hemi, space, density, label, segmentation, cohort, scale, roi, extension, etc.). Pass ‘NULL’ to exclude files containing that entity.

Value

Character vector of template IDs.

Examples

```
## Not run:  
tf_templates()  
  
## End(Not run)
```

Index

`print.TemplateFlowClient`, 2

`TemplateFlowClient`, 3

`tf_cache_refresh`, 3

`tf_cache_scan`, 4

`tf_cache_stats`, 5

`tf_cache_update`, 5

`tf_cache_wipe`, 6

`tf_cli`, 7

`tf_client`, 7

`tf_datalad_available`, 8

`tf_datalad_get`, 8

`tf_datalad_install`, 9

`tf_datalad_update`, 9

`tf_entities`, 10

`tf_file_extension`, 10

`tf_get`, 11

`tf_get_citations`, 12

`tf_get_metadata`, 12

`tf_ls`, 13

`tf_prefetch`, 14

`tf_read_file`, 14

`tf_set_client`, 15

`tf_templates`, 15