

Package ‘getlandsat’

May 1, 2018

Type Package

Title Get Landsat 8 Data from Amazon Public Data Sets

Description Get Landsat 8 Data from Amazon Web Services ('AWS') public data sets (<<https://registry.opendata.aws/landsat-8/>>). Includes functions for listing images and fetching them, and handles caching to prevent unnecessary additional requests.

Version 0.2.0

License MIT + file LICENSE

URL <https://github.com/ropensci/getlandsat>

BugReports <https://github.com/ropensci/getlandsat/issues>

Imports methods, readr (>= 1.1.1), crul (>= 0.5.2), xml2 (>= 1.2.0), data.table, tibble, rappdirs

Suggests roxygen2 (>= 6.0.1), testthat, knitr, raster

VignetteBuilder knitr

RoxygenNote 6.0.1

X-schema.org-applicationCategory Geospatial

X-schema.org-keywords landsat, nasa, aws, image, imaging, usgs, earth

X-schema.org-isPartOf <https://ropensci.org>

NeedsCompilation no

Author Scott Chamberlain [aut, cre] (<<https://orcid.org/0000-0003-1444-9135>>)

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2018-04-30 22:30:01 UTC

R topics documented:

getlandsat-package	2
lsat_cache	2
lsat_image	4

lsat_list	5
lsat_scenes	6
lsat_scene_files	7

Index	8
--------------	----------

getlandsat-package	<i>getlandsat - get Landsat 8 data from AWS public data sets</i>
--------------------	--

Description

getlandsat provides access to Landsat <https://landsat.usgs.gov> 8 metadata and images hosted on AWS S3 at <https://registry.opendata.aws/landsat-8/>. The package only fetches data. It does not attempt to aid users in downstream usage.

Examples

```
## Not run:
## List scenes
(res <- lsat_scenes(n_max = 10))

## List scene files
lsat_scene_files(x = res$download_url[1])

## Get an image
### Returns path to the image
lsat_image(x = "LC80101172015002LGN00_B5.TIF")

## Visualize
if (requireNamespace("raster")) {
  library("raster")
  x <- lsat_cache_details()[[1]]
  img <- raster(x$file)
  plot(img)
}

## End(Not run)
```

lsat_cache	<i>Manage cached files</i>
------------	----------------------------

Description

Manage cached files

Usage

```
lsat_cache_list()

lsat_cache_delete(files, force = TRUE)

lsat_cache_delete_all(force = TRUE)

lsat_cache_details(files = NULL)
```

Arguments

files	(character) one or more complete file names
force	(logical) Should files be force deleted? Default: TRUE

Details

cache_delete only accepts 1 file name, while cache_delete_all doesn't accept any names, but deletes all files. For deleting many specific files, use cache_delete in a `lapply()` type call

We cache using `rappdirs::user_cache_dir()`, find your cache folder by executing `rappdirs::user_cache_dir("landsa`

Functions

- `lsat_cache_list()` returns a character vector of full path file names
- `lsat_cache_delete()` deletes one or more files, returns nothing
- `lsat_cache_delete_all()` delete all files, returns nothing
- `lsat_cache_details()` prints file name and file size for each file, supply with one or more files, or no files (and get details for all available)

Examples

```
## Not run:
# list files in cache
lsat_cache_list()

# List info for single files
lsat_cache_details(files = lsat_cache_list()[1])
lsat_cache_details(files = lsat_cache_list()[2])

# List info for all files
lsat_cache_details()

# delete files by name in cache
# lsat_cache_delete(files = lsat_cache_list()[1])

# delete all files in cache
# lsat_cache_delete_all()

## End(Not run)
```

lsat_image	<i>Get Landsat image(s)</i>
------------	-----------------------------

Description

Get Landsat image(s)

Usage

```
lsat_image(x, overwrite = FALSE, ...)
```

Arguments

x	(character) A file name for a geotif file, will be more general soon.
overwrite	(logical) Will only overwrite existing path if TRUE. Deprecated, will be removed in the next version. If file exists we return that path so there's no chance of overwriting
...	Curl args passed on to <code>curl::HttpClient()</code>

Value

Path to the file, whether found in cache or new file requested.

See Also

[lsat_cache\(\)](#)

Examples

```
## Not run:
# pass an image name
(res <- lsat_list(max = 40))
tifs <- grep("\\.TIF$", res$Key, value = TRUE)
lsat_image(tifs[5])
lsat_image(tifs[6])
lsat_image(tifs[9])

# caching
## requesting an image you already have will return path if found
lsat_image(tifs[5])

## End(Not run)
```

lsat_list	<i>List Landsat images</i>
-----------	----------------------------

Description

List Landsat images

Usage

```
lsat_list(max = NULL, marker = NULL, prefix = NULL, delimiter = NULL,  
...)
```

Arguments

max	(integer) number indicating the maximum number of keys to return (max 1000, default 1000).
marker	(character) string that specifies the key to start with when listing objects in a AWS bucket. Amazon S3 returns object keys in alphabetical order, starting with key after the marker in order
prefix	(character) string that limits the response to keys that begin with the specified prefix
delimiter	(character) string used to group keys. Read the AWS doc for more detail.
...	curl args passed on to curl::HttpClient()

Details

This is an alternative to using [lsat_scenes\(\)](#). This function uses the AWS S3 API, while [lsat_scenes\(\)](#) downloads the up to date compressed csv file.

Examples

```
## Not run:  
lsat_list(max = 10)  
  
# paging, start a specific key string  
res <- lsat_list(max = 10)  
lsat_list(marker = res$Key[10], max = 10)  
  
# curl options  
lsat_list(max = 3, verbose = TRUE)  
  
## End(Not run)
```

lsat_scenes	<i>List Landsat scenes</i>
-------------	----------------------------

Description

List Landsat scenes

Usage

```
lsat_scenes(...)
```

Arguments

... Further args passed on to [readr::read_csv\(\)](#)

Details

We use [readr::read_csv\(\)](#) to read the scene list file from http://landsat-pds.s3.amazonaws.com/scene_list.gz. See the help file for [readr::read_csv\(\)](#) for what parameter you can pass to modify it's behavior.

This is an alternative to using [lsat_list\(\)](#). This function downloads the up to date compressed csv file, while [lsat_list\(\)](#) uses the AWS S3 API.

See Also

[lsat_scene_files\(\)](#)

Examples

```
## Not run:  
res <- lsat_scenes()  
res  
  
# read only N rows  
lsat_scenes(n_max = 10)  
  
## End(Not run)
```

lsat_scene_files	<i>List files for a Landsat scene</i>
------------------	---------------------------------------

Description

List files for a Landsat scene

Usage

```
lsat_scene_files(x, ...)
```

Arguments

x	(character) A URL to a scene html file
...	Curl options passed on to <code>curl::HttpClient()</code>

Details

This function fetches files available in a scene, while [lsat_scenes\(\)](#) lists the scenes, but not their files

Value

A data.frame with two columns:

- file - file name
- size - file size

See Also

[lsat_scenes\(\)](#)

Examples

```
## Not run:  
res <- lsat_scenes(n_max = 10)  
lsat_scene_files(x = res$download_url[1])  
lsat_scene_files(x = res$download_url[2])  
  
## End(Not run)
```

Index

*Topic **package**

- getlandsat-package, 2

- crul::HttpClient(), 4, 5, 7

- getlandsat (getlandsat-package), 2
- getlandsat-package, 2

- lapply(), 3
- lsat_cache, 2
- lsat_cache(), 4
- lsat_cache_delete (lsat_cache), 2
- lsat_cache_delete_all (lsat_cache), 2
- lsat_cache_details (lsat_cache), 2
- lsat_cache_list (lsat_cache), 2
- lsat_image, 4
- lsat_list, 5
- lsat_list(), 6
- lsat_scene_files, 7
- lsat_scene_files(), 6
- lsat_scenes, 6
- lsat_scenes(), 5, 7

- rappdirs::user_cache_dir(), 3
- readr::read_csv(), 6