Package 'rdrop2'

October 14, 2022

Title Programmatic Interface to the 'Dropbox' API

Version 0.8.2.1

Description Provides full programmatic access to the 'Dropbox' file hosting platform <https://dropbox.com>, including support for all standard file operations.

Depends R (>= 3.1.1)

License MIT + file LICENSE

BugReports https://github.com/karthik/rdrop2/issues

LazyData true

Imports assertive, digest, dplyr, httr, jsonlite, magrittr, purrr

Suggests testthat, uuid

RoxygenNote 7.1.1

NeedsCompilation no

Author Karthik Ram [aut, cre],
    Clayton Yochum [aut],
    Caleb Scheidel [ctb],
    Akhil Bhel [cph]

Maintainer Karthik Ram <karthik.ram@gmail.com>

Repository CRAN

Date/Publication 2020-08-05 12:00:03 UTC

R topics documented:

---

drop_acc   *Get information about current Dropbox account.*

Description

Fields returned will vary by account;

Usage

```
drop_acc(dtoken = get_dropbox_token())
```

Arguments

dtoken   The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically
                locate your local credential cache and use them. However, if the credentials are
                not found, the function will initiate a new authentication request. You can over-
                ride this in [drop_auth]() by pointing to a different location where your credentials
                are stored.

Value

Nested list with elements account_id, name (list), email, email_verified, disabled, locale,
referral_link, is_paired, account_type (list).

If available, may also return profile_photo_url, country, team (list), team_member_id.

References

[API documentation]()

Examples

```
## Not run:

  acc_info <- drop_acc()

  # extract display name
  acc_info$name$display_name

## End(Not run)
```

---

drop_auth   *Authentication for Dropbox*

Description

This function authenticates you into Dropbox. The documentation for the core Dropbox API provides more details including alternate methods if you desire to reimplement your own.

Usage

```
drop_auth(
  new_user = FALSE,
  key = "mmhfsybffdom42w",
  secret = "l8zeqqqgm1ne5z0",
  cache = TRUE,
  rdstoken = NA
)
```

Arguments

new_user	Set to TRUE if you need to switch to a new user account or just flush existing token. Default is FALSE.
key	Your application key. rdrop2 already comes with a key/secret but you are welcome to swap out with our own. Since these keys are shipped with the package, there is a small chance they could be voided if someone abuses the key. If you plan to use this in production, or for an internal tool, the recommended practice is to create a new application on Dropbox and use those keys for your purposes.
secret	Your application secret. Like key, rdrop2 comes with a secret but you are welcome to swap out with our own.
cache	By default your credentials are locally cached in a file called .httr-oauth. Set to FALSE if you need to authenticate separately each time.
rdstoken	File path to stored RDS token. In server environments where interactive OAuth is not possible, a token can be created on a desktop client and used in production. See examples.

Value

A Token2.0 object, invisibly

References

API documentation

Examples

```
## Not run:

  # To either read token from .httr-oauth in the working directory or open a
  # web browser to authenticate (and cache a token)
  drop_auth()

  # If you want to overwrite an existing local token and switch to a new
  # user, set new_user to TRUE.
  drop_auth(new_user = TRUE)

  # To store a token for re-use (more flexible than .httr-oauth), save the
  # output of drop_auth and save it to an RDS file
  token <- drop_auth()
  saveRDS(token, "/path/to/tokenfile.RDS")

  # To use a stored token provide token location
  drop_auth(rdstoken = "/path/to/tokenfile.RDS")

## End(Not run)
```

---

drop_content_hash	*Compute Dropbox's content hash for one or more files*

Description

Compute a "content hash" using the same algorithm as dropbox. This can be used to verify the content against the content_hash field returned in drop_dir.

Usage

```
drop_content_hash(file)
```

Arguments

file   A vector of filenames

Details

Dropbox returns a hash of file contents in `drop_dir`. However, this is not a straightforward file hash. Instead the file is divided into 4MB chunks, each of those is hashed and then the concatenation of the hashes is itself hashed (see this page in the dropbox developer documentation for the details). It's entirely unclear why it does not compute a hash of the file itself, but here we are.

Value

A character vector the same length as `file`. Each element is 64 character string which is the unique hash. Two files that have the same hash have the same contents. Compare this hash of a local file with the `content_hash` field from `drop_dir` to see if you have the same file as dropbox.

Examples

```
## Not run:
write.csv(mtcars, file = "mtt.csv")
drop_upload("mtt.csv")
files <- drop_dir()
# Dropbox's reported hash
files$content_hash[files$name == "mtt.csv"]
# Our computed hash:
drop_content_hash("mtt.csv")

## End(Not run)
```

---

drop_copy   *Copies a file or folder to a new location.*

Description

Copies a file or folder to a new location.

Usage

```
drop_copy(
  from_path = NULL,
  to_path = NULL,
  allow_shared_folder = FALSE,
  autorename = FALSE,
  allow_ownership_transfer = FALSE,
  verbose = FALSE,
  dtoken = get_dropbox_token()
)
```

Arguments

`from_path`	Source file or folder
`to_path`	destination file or folder
`allow_shared_folder`	
	If `TRUE`, copy will copy contents in shared folder
`autorename`	If there's a conflict, have the Dropbox server try to autorename the file to avoid the conflict.
`allow_ownership_transfer`	
	Allow moves by owner even if it would result in an ownership transfer for the content being moved. This does not apply to copies. The default for this field is False.
`verbose`	By default verbose output is `FALSE`. Set to `TRUE` if you need to troubleshoot any output or grab additional parameters.
`dtoken`	The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth]() by pointing to a different location where your credentials are stored.

References

[API documentation]()

Examples

```
## Not run:
write.csv(mtcars, file = "mt.csv")
drop_upload("mt.csv")
drop_create("drop_test2")
drop_copy("mt.csv", "drop_test2/mt2.csv")

## End(Not run)
```

`drop_create`	*Creates a folder on Dropbox*

Description

Returns a list containing the following fields: "size", "rev", "thumb_exists", "bytes", "modified", "path", "is_dir", "icon", "root", "revision"

Usage

```
drop_create(
  path = NULL,
  autorename = FALSE,
  verbose = FALSE,
  dtoken = get_dropbox_token()
)
```

Arguments

path	Path in the user's Dropbox, relative to root
autorename	Set to TRUE to automatically rename. Default is FALSE.
verbose	By default verbose output is FALSE. Set to TRUE if you need to troubleshoot any output or grab additional parameters.
dtoken	The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth]() by pointing to a different location where your credentials are stored.

References

[API documentation]()

Examples

```
## Not run:
drop_create(path = "foobar")

## End(Not run)
```

---

drop_delete	*Deletes a file or folder.*

Description

Deletes a file or folder.

Usage

```
drop_delete(path = NULL, verbose = FALSE, dtoken = get_dropbox_token())
```

Arguments

path	Path in the user's Dropbox, relative to root
verbose	By default verbose output is FALSE. Set to TRUE if you need to troubleshoot any output or grab additional parameters.
dtoken	The Dropbox token generated by [drop_auth](#). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth](#) by pointing to a different location where your credentials are stored.

References

[API documentation](#)

---

drop_dir   *List folder contents and associated metadata.*

Description

Can be used to either see all items in a folder, or only items that have changed since a previous request was made.

Usage

```
drop_dir(
  path = "",
  recursive = FALSE,
  include_media_info = FALSE,
  include_deleted = FALSE,
  include_has_explicit_shared_members = FALSE,
  include_mounted_folders = TRUE,
  limit = NULL,
  cursor = FALSE,
  dtoken = get_dropbox_token()
)
```

Arguments

path	path to folder in Dropbox to list contents of. Defaults to the root directory.
recursive	If TRUE, the list folder operation will be applied recursively to all subfolders and the response will contain contents of all subfolders. Defaults to FALSE.
include_media_info	
	If TRUE, FileMetadata.media_info is set for photo and video. Defaults to FALSE.
include_deleted	
	If TRUE, the results will include entries for files and folders that used to exist but were deleted. Defaults to FALSE.

include_has_explicit_shared_members

> If TRUE, the results will include a flag for each file indicating whether or not that file has any explicit members. Defaults to FALSE.

include_mounted_folders

> If TRUE, the results will include entries under mounted folders which includes app folder, shared folder and team folder. Defaults to TRUE.

limit   The maximum number of results to return per request. Note: This is an approximate number and there can be slightly more entries returned in some cases. Defaults to NULL, no limit.

cursor   string or boolean:

> - If FALSE, return metadata of items in `path`
> - If TRUE, return a cursor to be used for detecting changed later
> - If a string, return metadata of changed items since the cursor was fetched

dtoken   The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth]() by pointing to a different location where your credentials are stored.

Value

Either a `tbl_df` of items in folder, one row per file or folder, with metadata values as columns, or a character string giving a cursor to be used later for change detection (see `cursor`).

Examples

```
## Not run:

  # list files in root directory
  drop_dir()

  # get a cursor from root directory,
  # upload a new file,
  # return only information about new file
  cursor <- drop_dir(cursor = TRUE)
  drop_upload("some_new_file")
  drop_dir(cursor = cursor)

## End(Not run)
```

---

drop_download   *Download a file from Dropbox to disk.*

Description

Download a file from Dropbox to disk.

Usage

```
drop_download(
  path,
  local_path = NULL,
  overwrite = FALSE,
  progress = interactive(),
  verbose = interactive(),
  dtoken = get_dropbox_token()
)
```

Arguments

path	path to a file in Dropbox
local_path	path to save file to. If NULL (the default), saves file to working directory with same name. If not null, but a valid folder, file will be saved in this folder with same basename as path. If not null and not a folder, file will be saved to this path exactly.
overwrite	If TRUE, overwrite local file. Defaults to FALSE
progress	If TRUE, show a progress bar for large file downloads. Defaults to TRUE in interactive sessions, otherwise FALSE.
verbose	if TRUE, emit message giving location and size of the newly downloaded file. Defaults to TRUE in interactive sessions, otherwise FALSE.
dtoken	The Dropbox token generated by drop_auth. rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in drop_auth by pointing to a different location where your credentials are stored.

Value

TRUE if successful; error thrown otherwise.

References

API documentation

Examples

```
## Not run:

  # download a file to the current working directory
  drop_get("dataset.zip")

  # download again, overwriting previous result
  drop_get("dataset.zip", overwrite = TRUE)

  # download to a different path, keeping file name constant
  # will download to "some/other/place/dataset.zip"
```

```
    drop_get("dataset.zip", local_path = "some/other/place/")

    # download to to a different path, changing filename
    drop_get("dataset.zip", local_path = "some/other/place/not_a_dataset.zip")

  ## End(Not run)
```

---

drop_exists   *Checks to see if a file/folder exists on Dropbox*

Description

Since many file operations such as move, copy, delete and history can only act on files that currently exist on a Dropbox store, checking to see if the path is valid before operating prevents bad API calls from being sent to the server. This functions returns a logical response after checking if a file path is valid on Dropbox.

Usage

```
drop_exists(path = NULL, dtoken = get_dropbox_token())
```

Arguments

path   The full path to a Dropbox file

dtoken   The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth]() by pointing to a different location where your credentials are stored.

Value

boolean; TRUE is the file or folder exists, FALSE if it does not.

Examples

```
## Not run:
  drop_create("existential_test")
  drop_exists("existential_test")
  drop_delete("existential_test")

## End(Not run)
```

drop_get   *Downloads a file from Dropbox*

Description

Downloads a file from Dropbox

Usage

```
drop_get(
  path = NULL,
  local_file = NULL,
  overwrite = FALSE,
  verbose = FALSE,
  progress = FALSE,
  dtoken = get_dropbox_token()
)
```

Arguments

path	Path in the user's Dropbox, relative to root
local_file	The name of the local copy. Leave this blank if you're fine with the original name.
overwrite	Default is FALSE but can be set to TRUE.
verbose	By default verbose output is FALSE. Set to TRUE if you need to troubleshoot any output or grab additional parameters.
progress	Progress bars are turned off by default. Set to TRUE ot turn this on. Progress is only reported when file sizes are known. Otherwise just bytes downloaded.
dtoken	The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth]() by pointing to a different location where your credentials are stored.

Examples

```
## Not run:
  drop_get(path = 'dataset.zip', local_file = "~/Desktop")
  # To overwrite the existing file
  drop_get(path = 'dataset.zip', overwrite = TRUE)

## End(Not run)
```

drop_get_metadata *Retrieve metadata for a file or folder.*

Description

Details vary by input and args.

Usage

```
drop_get_metadata(
  path,
  include_media_info = FALSE,
  include_deleted = FALSE,
  include_has_explicit_shared_members = FALSE,
  dtoken = get_dropbox_token()
)
```

Arguments

path   Path to a file or folder on Dropbox. Can also be an ID ("id:...") or revision
                ("rev:...").

include_media_info

                If TRUE, additional metadata for photo or video is returns. Defaults to FALSE.

include_deleted

                If TRUE, metadata will be returned for a deleted file, otherwise error. Defaults
                to FALSE.

include_has_explicit_shared_members

                If TRUE, the results will include a flag for each file indicating whether or not
                that file has any explicit members. Defaults to FALSE.

dtoken   The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically
                locate your local credential cache and use them. However, if the credentials are
                not found, the function will initiate a new authentication request. You can over-
                ride this in [drop_auth]() by pointing to a different location where your credentials
                are stored.

Value

possibly-nested list of all available metadata for specified file/folder/id/revision.

References

[API Documentation]()

---

drop_history *Obtains metadata for all available revisions of a file, including the current revision.*

Description

Does not include deleted revisions.

Usage

```
drop_history(path, limit = 10, dtoken = get_dropbox_token())
```

Arguments

path	path to a file in dropbox.
limit	maximum number of revisions to return; defaults to 10.
dtoken	The Dropbox token generated by [drop_auth](#). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth](#) by pointing to a different location where your credentials are stored.

Value

`tbl_df` of metadata, one row per revision.

Examples

```
## Not run:
  write.csv(iris, file = "iris.csv")
  drop_upload("iris.csv")
  write.csv(iris[iris$Species == "setosa", ], file = "iris.csv")
  drop_upload("iris.csv")
  drop_history("iris.csv")

## End(Not run)
```

---

drop_list_shared_links

List all shared links

Description

This function returns a list of all links that are currently being shared

Usage

```
drop_list_shared_links(verbose = TRUE, dtoken = get_dropbox_token())
```

Arguments

verbose   Print verbose output

dtoken   The Dropbox token generated by [drop_auth](#). rdrop2 will try to automatically
                 locate your local credential cache and use them. However, if the credentials are
                 not found, the function will initiate a new authentication request. You can over-
                 ride this in [drop_auth](#) by pointing to a different location where your credentials
                 are stored.

References

[API documentation](#)

Examples

```
## Not run:
drop_list_shared_links()

## End(Not run)
```

---

drop_media   *Returns a link directly to a file.*

Description

Similar to `drop_shared`. The difference is that this bypasses the Dropbox webserver, used to pro-
vide a preview of the file, so that you can effectively stream the contents of your media. This URL
should not be used to display content directly in the browser. IMPORTANT: The media link will
expire after 4 hours. So you'll need to cache the content with knitr cache OR re-run the function
call after expiry.

Usage

```
drop_media(path = NULL, dtoken = get_dropbox_token())
```

Arguments

path   Path in the user's Dropbox, relative to root

dtoken   The Dropbox token generated by [drop_auth](#). rdrop2 will try to automatically
                 locate your local credential cache and use them. However, if the credentials are
                 not found, the function will initiate a new authentication request. You can over-
                 ride this in [drop_auth](#) by pointing to a different location where your credentials
                 are stored.

References

[API documentation](#)

Examples

```
## Not run:
drop_media('Public/gifs/duck_rabbit.gif')

## End(Not run)
```

---

drop_move   *Moves a file or folder to a new location.*

Description

Moves a file or folder to a new location.

Usage

```
drop_move(
  from_path = NULL,
  to_path = NULL,
  allow_shared_folder = FALSE,
  autorename = FALSE,
  allow_ownership_transfer = FALSE,
  verbose = FALSE,
  dtoken = get_dropbox_token()
)
```

Arguments

from_path	Source file or folder
to_path	destination file or folder
allow_shared_folder	
	If TRUE, copy will copy contents in shared folder
autorename	If there's a conflict, have the Dropbox server try to autorename the file to avoid the conflict.
allow_ownership_transfer	
	Allow moves by owner even if it would result in an ownership transfer for the content being moved. This does not apply to copies. The default for this field is False.
verbose	By default verbose output is FALSE. Set to TRUE if you need to troubleshoot any output or grab additional parameters.
dtoken	The Dropbox token generated by [drop_auth](#). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth](#) by pointing to a different location where your credentials are stored.

References

[API documentation](#)

Examples

```
## Not run:
write.csv(mtcars, file = "mt.csv")
drop_upload("mt.csv")
drop_create("drop_test2")
drop_move("mt.csv", "drop_test2/mt.csv")

## End(Not run)
```

---

drop_read_csv   *drop_read_csv*

Description

A lightweight wrapper around `read.csv` to read csv files from Dropbox into memory

Usage

```
drop_read_csv(file, dest = tempdir(), dtoken = get_dropbox_token(), ...)
```

Arguments

`file`	Name of file with full path relative to Dropbox root
`dest`	A temporary directory where a csv file is downloaded before being read into memory
`dtoken`	The Dropbox token generated by [drop_auth](#). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth](#) by pointing to a different location where your credentials are stored.
`...`	Additional arguments into `read.csv`

Examples

```
## Not run:
write.csv(iris, file = "iris.csv")
drop_upload("iris.csv")
# Now let's read this back into an R session
new_iris <- drop_read_csv("iris.csv")

## End(Not run)
```

---

drop_search   *Returns metadata for all files and folders whose filename contains the*
                             given search string as a substring.

Description

Returns metadata for all files and folders whose filename contains the given search string as a
substring.

Usage

```
drop_search(
  query,
  path = "",
  start = 0,
  max_results = 100,
  mode = "filename",
  dtoken = get_dropbox_token()
)
```

Arguments

query	The search string. This string is split (on spaces) into individual words. Files and folders will be returned if they contain all words in the search string.
path	Path in the user's Dropbox, relative to root
start	The starting index within the search results (used for paging). The default for this field is 0
max_results	The maximum number of search results to return. The default for this field is 100.
mode	Mode can take the option of filename, filename_and_content, or search deleted files with deleted_filename
dtoken	The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth]() by pointing to a different location where your credentials are stored.

References

[API documentation]()

Examples

```
## Not run:
# If you know me, you know why this query exists
drop_search('gif') %>% select(path, is_dir, mime_type)
```

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

---

drop_share   *Creates and returns a shared link to a file or folder.*

Description

Creates and returns a shared link to a file or folder.

Usage

```
drop_share(
  path = NULL,
  requested_visibility = "public",
  link_password = NULL,
  expires = NULL,
  dtoken = get_dropbox_token()
)
```

Arguments

path	Path in the user's Dropbox, relative to root
requested_visibility	
	Can be 'public', 'team_only', or 'password'. If the password option is chosen one must specify the 'link_password'. Note that for basic (i.e. free) Dropbox accounts, the only option is to publicly share. Private sharing requires a pro account.
link_password	The password needed to access the document if 'request_visibility' is set to password.
expires	Set the expiry time. The timestamp format is "%Y-%m-%dT%H:%M:%SZ"). If no timestamp is specified, link never expires
dtoken	The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth]() by pointing to a different location where your credentials are stored.

References

[API documentation]()

Examples

```
## Not run:
write.csv(mtcars, file = "mt.csv")
drop_upload("mt.csv")
drop_share("mt.csv")
# If you have a pro account, you can share files privately
drop_share("mt.csv", requested_visibility = "password", link_password = "test")

## End(Not run)
```

---

drop_upload   *Uploads a file to Dropbox.*

Description

This function will allow you to write files of any size to Dropbox(even ones that cannot be read into memory) by uploading them in chunks.

Usage

```
drop_upload(
  file,
  path = NULL,
  mode = "overwrite",
  autorename = TRUE,
  mute = FALSE,
  verbose = FALSE,
  dtoken = get_dropbox_token()
)
```

Arguments

file	Relative path to local file.
path	The relative path on Dropbox where the file should get uploaded.
mode	- "add" - will not overwrite an existing file in case of a conflict. With this mode, when a a duplicate file.txt is uploaded, it will become file (2).txt. - "overwrite" will always overwrite a file -
autorename	This logical determines what happens when there is a conflict. If true, the file being uploaded will be automatically renamed to avoid the conflict. (For example, test.txt might be automatically renamed to test (1).txt.) The new name can be obtained from the returned metadata. If false, the call will fail with a 409 (Conflict) response code. The default is 'TRUE'
mute	Set to FALSE to prevent a notification trigger on the desktop and mobile apps
verbose	By default verbose output is FALSE. Set to TRUE if you need to troubleshoot any output or grab additional parameters.

dtoken   The Dropbox token generated by [drop_auth](). rdrop2 will try to automatically locate your local credential cache and use them. However, if the credentials are not found, the function will initiate a new authentication request. You can override this in [drop_auth]() by pointing to a different location where your credentials are stored.

References

[API documentation]()

Examples

```
## Not run:
write.csv(mtcars, file = "mtt.csv")
drop_upload("mtt.csv")

## End(Not run)
```

Index