Package ‘databraryr’

April 18, 2024

Type Package

Title Interact with the 'Databrary.org' API

Version 0.6.4

Maintainer Rick O. Gilmore <rog1@psu.edu>

Description 'Databrary.org' is a restricted access repository for research data, especially video and audio. This package provides commands to interact with the data stored on 'Databrary.org'.

License MIT + file LICENSE


BugReports https://github.com/databrary/databraryr/issues

Imports assertthat (>= 0.2.1), dplyr (>= 1.1.4), getPass (>= 0.2.4), httr (>= 1.4.7), httr2 (>= 1.0.0), jsonlite (>= 1.8.8), keyring (>= 1.3.2), lifecycle (>= 1.0.4), magick (>= 2.8.2), magrittr, methods (>= 4.3.2), plyr, purrr (>= 1.0.2), rvest, stringr (>= 1.5.1), tibble (>= 3.2.1), utils (>= 4.3.2), xfun (>= 0.41)

Suggests av (>= 0.9.0), ggplot2, knitr, openssl, readr, rmarkdown (>= 2.26), testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Rick O. Gilmore [aut, cre, cph], Jeffrey Spies [aut], National Science Foundation OAC-2032713 [fnd], National Institutes of Health R01HD094830 [fnd]

Repository CRAN

Date/Publication 2024-04-18 13:02:37 UTC
R topics documented:

assign_constants .............................................. 3
check_ssl_certs ................................................. 4
DATABRARY_API ................................................... 4
download_asset .................................................. 5
download_party .................................................. 6
download_party_avatar ........................................... 6
download_session_asset ......................................... 7
download_session_assets_fr_df ................................. 8
download_session_csv ........................................... 9
download_session_zip ........................................... 10
download_single_session_asset_fr_df ......................... 11
download_video ................................................... 13
download_volume_zip ............................................ 14
get_assets_from_session ....................................... 15
get_asset_segment_range ...................................... 15
GET_db_contents ............................................... 16
get_db_stats ..................................................... 17
get_file_duration ............................................... 18
get_party_as_df .................................................. 18
get_party_by_id .................................................. 19
get_permission_levels ........................................ 20
get_release_levels ............................................. 20
get_session_as_df ............................................. 21
get_session_by_id ............................................. 22
get_session_by_name ........................................... 22
get_supported_file_types ...................................... 23
get_video_stats .................................................. 24
get_volume_by_id ............................................... 25
HHMMSSmmm_to_ms ............................................... 25
is_institution .................................................... 26
is_person .......................................................... 27
list_affiliates ................................................... 27
list_assets_by_type ............................................. 28
list_assets_in_session ......................................... 29
list_assets_in_volume .......................................... 29
list_asset_formats .............................................. 30
list_authorized_investigators ................................. 30
list_containers_records ...................................... 31
list_individual_sponsors ...................................... 32
list_institutional_sponsors ................................... 33
list_party ........................................................ 33
list_party_affiliates .......................................... 34
list_party_sponsors ............................................. 35
list_party_volumes ............................................. 35
list_people ....................................................... 36
list_sessions ....................................................... 37
assign_constants

Description

Download Databrary Constants From API.

Usage

assign_constants(vb = FALSE, rq = NULL)

Arguments

- vb: A Boolean value. If TRUE provides verbose output.
- rq: An `httr2` request object. Defaults to NULL.

Value

A data frame with the constants.
Examples

```r
assign_constants()
```

---

check_ssl_certs

*Check SSL Certificates For nyu.databrary.org.*

Description

check_ssl_certs checks the SSL certificates for nyu.databrary.org and returns a data frame with the relevant information.

Usage

```r
check_ssl_certs(host = "nyu.databrary.org")
```

Arguments

- `host`: Target URL. Defaults to ’nyu.databrary.org’.

Value

A data frame with information about the SSL certificates.

Examples

```r
check_ssl_certs()
```

---

**DATABRARY_API**

*Load Package-wide Constants into Local Environment*

Description

Load Package-wide Constants into Local Environment

Usage

```r
DATABRARY_API
```

Format

An object of class character of length 1.
**download_asset**

Download Asset From Databrary.

### Description

**[Superseded]**

`download_asset()` has been superseded in favor of `download_video()` and the more general `download_session_asset()`.

### Usage

```r
download_asset(
  asset_id = 1,
  session_id = 9807,
  file_name = "test.mp4",
  target_dir = tempdir(),
  return_response = FALSE,
  vb = FALSE
)
```

### Arguments

- `asset_id`  
  Asset id for target file.
- `session_id`  
  Slot/session number where target file is stored.
- `file_name`  
  Name for downloaded file.
- `target_dir`  
  Directory to save the downloaded file. Default is a temporary directory given by a call to `tempdir()`.
- `return_response`  
  A Boolean value.
- `vb`  
  A Boolean value. If TRUE provides verbose output.

### Value

A string that is the full file name of the asset.

### Examples

```r
download_asset() # Download's 'numbers' file from volume 1.
download_asset(asset_id = 11643, session_id = 9825, file_name = "rdk.mp4")
# Downloads a display with a random dot kinematogram (RDK).
```
**download_party**  
*Download Information About a Party on Databrary.*

**Description**

[Deprecated]  
This function has been deprecated and may be removed in a future release. See `get_party_by_id()` for similar, and more general functionality.

**Usage**

```
download_party(party_id = 6, convert_JSON = TRUE, vb = FALSE)
```

**Arguments**

- **party_id**: Party number to retrieve information about.
- **convert_JSON**: A Boolean value if TRUE converts the JSON download.
- **vb**: A Boolean value if TRUE returns verbose output.

**Value**

A data frame with information about the party.

**Examples**

```
download_party()
```

**download_party_avatar**  
*Returns the Avatar(s) (images) for Authorized User(s).*

**Description**

Returns the Avatar(s) (images) for Authorized User(s).

**Usage**

```
download_party_avatar(
    party_id = 6,
    show_party_info = TRUE,
    vb = FALSE,
    rq = NULL
)
```
download_session_asset

Arguments

party_id A number or range of numbers. Party number or numbers to retrieve information about. Default is 6 (Rick Gilmore).

show_party_info A logical value. Show the person’s name and affiliation in the output. Default is TRUE.

vb A Boolean value. If TRUE returns verbose output. Default is FALSE.

rq An httr2 request object. If not provided, a new request is generated via make_default_request().

Value

An list with the avatar (image) file and a name_affil string.

Examples

```r
## Not run:
download_party_avatar() # Show Rick Gilmore's (party 6) avatar.

# Download avatars from Databrary's founders (without name/affiliations)
download_party_avatar(5:7, show_party_info = FALSE)

# Download NYU logo
download_party_avatar(party = 8)

## End(Not run)
```

download_session_asset

Download Asset From Databrary.

Description

Databrary stores file types (assets) of many types. This function downloads an asset based on its system-unique integer identifier (asset_id) and system-unique session (slot) identifier (session_id).

Usage

```r
download_session_asset(
  asset_id = 1,
  session_id = 9807,
  file_name = NULL,
  target_dir = paste0("./", session_id),
  vb = FALSE,
  rq = NULL
)
```
Arguments

asset_id  An integer. Asset id for target file. Default is 1.
session_id  An integer. Slot/session number where target file is stored. Default is 9807.
file_name  A character string. Name for downloaded file. Default is NULL.
target_dir  A character string. Directory to save the downloaded file. Default is a temporary directory given by a call to tempdir().
vb  A logical value. If TRUE provides verbose output. Default is FALSE.
rq  A list in the form of an \texttt{httr2} request object. Default is NULL.

Value

Full file name to the asset or NULL.

Examples

```r
## Not run:
download_session_asset() # Download 'numbers' file from volume 1.
download_session_asset(asset_id = 11643, session_id = 9825, file_name = "rdk.mp4")
# Downloads a display with a random dot kinematogram (RDK).
## End(Not run)
```

---

**download_session_assets_fr_df**

*Download Asset From A Databrary Session.*

Description

Databrary stores file types (assets) of many types. This function downloads assets in a data frame generated by list_session_assets().

Usage

```r
download_session_assets_fr_df(
  session_df = list_session_assets(),
  target_dir = tempdir(),
  overwrite = TRUE,
  make_portable_fn = FALSE,
  timeout_secs = 600,
  vb = FALSE,
  rq = NULL
)
```
**Arguments**

- **session_df**: A data frame as generated by `list_session_assets_2()`.
- **target_dir**: A character string. Directory to save the downloaded file. Default is directory named after the session_id.
- **overwrite**: A logical value. Overwrite an existing file. Default is TRUE.
- **make_portable_fn**: A logical value. Replace characters in file names that are not broadly portable across file systems. Default is FALSE.
- **timeout_secs**: An integer. The seconds an `httr2` request will run before timing out. Default is 600 (10 min). This is to handle very large files.
- **vb**: A logical value. If TRUE provides verbose output. Default is FALSE.
- **rq**: A list in the form of an `httr2` request object. Default is NULL.

**Value**

Full file names to the downloaded assets or NULL.

**Examples**

```
## Not run:
download_session_assets_fr_df() # Downloads all of the files from session 9807 in Databrary volume 1.

# Just the CSVs
v1 <- list_session_assets()
v1_csv <- dplyr::filter(v1, format_extension == "csv")
download_session_assets_fr_df(v1_csv, vb = TRUE)
```

**Description**

Databrary generates a CSV-formated spreadsheet that summarizes information about individual sessions. This command downloads that CSV file as a temporary file or with a name specified by the user.
Usage

```r
download_session_csv(
  vol_id = 1,
  file_name = "test.csv",
  target_dir = tempdir(),
  as_df = FALSE,
  vb = FALSE,
  rq = NULL
)
```

Arguments

- `vol_id` An integer. Target volume number. Default is 1.
- `file_name` A character string. Name for the output file. Default is 'test.csv'.
- `as_df` A logical value. Convert the data from a list to a data frame. Default is FALSE.
- `vb` A logical value. Provide verbose feedback. Default is FALSE.
- `rq` An `httr2` request object. Default is NULL.

Value

A character string that is the name of the downloaded file or a data frame if `as_df` is TRUE.

Examples

```r
## Not run:
download_session_csv() # Downloads "session" CSV for volume 1
## End(Not run)
```

Download Zip Archive From Databrary Session.

Description

Download Zip Archive From Databrary Session.
Usage

download_session_zip(
  vol_id = 31,
  session_id = 9803,
  out_dir = tempdir(),
  file_name = "test.zip",
  vb = FALSE,
  rq = NULL
)

Arguments

  vol_id           Volume number.
  session_id       Slot/session number.
  out_dir          Directory to save output file.
  file_name        Name for downloaded file, default is 'test.zip'.
  vb               A Boolean value. If TRUE provides verbose output.
  rq               An httr2 request object. Default is NULL.

Value

  Full filename of the downloaded file.

Examples

  ## Not run:
  download_session_zip() # Downloads Zip Archive from volume 31, session 9803

  ## End(Not run)

**Description**

Databrary stores file types (assets) of many types. This function downloads an asset based on its system-unique integer identifier (asset_id) and system-unique session (slot) identifier (session_id). It is designed to work with `download_session_assets_fr_df()` so that multiple files can be downloaded simultaneously.
download_single_session_asset_fr_df

Usage

```r
download_single_session_asset_fr_df(
i = NULL,
session_df = NULL,
target_dir = tempdir(),
overwrite = TRUE,
make_portable_fn = FALSE,
timeout_secs = 600,
vb = FALSE,
rq = NULL
)
```

Arguments

- `i` An integer. Index into a row of the session asset data frame. Default is NULL.
- `session_df` A row from a data frame from `list_session_assets()` or `list_volume_assets()`. Default is NULL.
- `target_dir` A character string. Directory to save the downloaded file. Default is a temporary directory given by a call to `tempdir()`.
- `overwrite` A logical value. Overwrite an existing file. Default is TRUE.
- `make_portable_fn` A logical value. Replace characters in file names that are not broadly portable across file systems. Default is FALSE.
- `timeout_secs` An integer. The seconds an httr2 request will run before timing out. Default is 600 (10 min). This is to handle very large files.
- `vb` A logical value. If TRUE provides verbose output. Default is FALSE.
- `rq` A list in the form of an httr2 request object. Default is NULL.

Value

Full file name to the asset or NULL.

Examples

```r
## Not run:
vol_1 <- list_session_assets(session_id = 9807)
a_1 <- vol_1[1,]
tmp_dir <- tempdir()
fn <- file.path(tmp_dir, paste0(a_1$asset_name, ",", a_1$format_extension))
download_single_session_asset_fr_df(a_1$asset_id,
fn,
session_id = a_1$session_id,
vb = TRUE)

## End(Not run)
```
Download Video From Databrary.

Description
Download Video From Databrary.

Usage
```
download_video(  
  asset_id = 1,  
  session_id = 9807,  
  file_name = tempfile(paste0(session_id, "_", asset_id, "_"), fileext = ".mp4"),  
  target_dir = tempdir(),  
  vb = FALSE,  
  rq = NULL  
)
```

Arguments

- `asset_id` Asset id for target file.
- `session_id` Slot/session number where target file is stored.
- `file_name` Name for downloaded file.
- `target_dir` Directory to save the downloaded file. Default is a temporary directory given by a call to `tempdir()`.
- `vb` A Boolean value. If TRUE provides verbose output.
- `rq` An `httr2` request object.

Value
Full file name to the asset.

Examples
```
download_asset() # Download's 'numbers' file from volume 1.  
download_asset(asset_id = 11643, session_id = 9825, file_name = "rdk.mp4")  
# ' # Downloads a display with a random dot kinematogram (RDK).
```
Description

Download Zip Archive of All Data in a Volume.

Usage

```
download_volume_zip(
  vol_id = 31,
  out_dir = tempdir(),
  file_name = "test.zip",
  vb = FALSE,
  rq = NULL
)
```

Arguments

- **vol_id**: Volume number.
- **out_dir**: Directory to save output file.
- **file_name**: Name for downloaded file, default is ‘test.mp4’.
- **vb**: A Boolean value. If TRUE provides verbose output.
- **rq**: An `httr2` request object. Default is NULL.

Value

Full filename of the downloaded file.

Examples

```
## Not run:
download_volume_zip() # Zip file of all data from volume 31, the default.

## End(Not run)
```
get_assets_from_session

Helper function for list_volume_assets

Description
Helper function for list_volume_assets

Usage
get_assets_from_session(volume_container, ignore_materials = TRUE)

Arguments

volume_container
The 'container' list from a volume.

ignore_materials
A logical value.

get_asset_segment_range

Get Time Range For An Asset.

Description
Get Time Range For An Asset.

Usage
get_asset_segment_range(
  vol_id = 1,
  session_id = 9807,
  asset_id = 1,
  convert_JSON = TRUE,
  segment_only = TRUE,
  vb = FALSE,
  rq = NULL
)
GET_db_contents

Queries the Databrary API with an HTML GET command

Description

GET_db_contents is a helper function that retrieves information from the 'Databrary.org' API in a structured way.

[Deprecated]
This function has been deprecated and may be removed in a future version.

Usage

```
GET_db_contents(
  base_URL = "https://nyu.databrary.org",
  URL_components = "/api/volume/1",
  convert_JSON = TRUE,
  vb = FALSE
)
```

Arguments

- `URL_components`: Other components of API URL usually assembled using a paste0() command.
- `convert_JSON`: A Boolean value. If TRUE, convert JSON to a data frame. Default is TRUE.
- `vb`: A Boolean value. If TRUE provides verbose output.

Value

The time range (in ms) for an asset, if one is indicated.

Examples

```
geet_asset_segment_range()
```
**get_db_stats**

Value

Output from GET command with the specified parameters, usually a string or a data frame.

Examples

```r
GET_db_contents() # Returns a list with data from volume 1.
```

---

**Description**

`get_db_stats` returns basic summary information about the institutions, people, and data hosted on 'Databrary.org'.

**Usage**

```r
get_db_stats(type = "stats", vb = FALSE, rq = NULL)
```

**Arguments**

- **type**: Type of Databrary report to run "institutions", "people", "data"
- **vb**: A Boolean value. If TRUE provides verbose output.
- **rq**: An *httr2* request object.

**Value**

A data frame with the requested data or NULL if there is no new information.

**Examples**

```r
get_db_stats()
get_db_stats("stats")
get_db_stats("people") # Information about the newest authorized investigators.
get_db_stats("places") # Information about the newest institutions.
```
get_file_duration  Get Duration (In ms) Of A File.

Description
Get Duration (In ms) Of A File.

Usage
get_file_duration(
    asset_id = 1,
    types_w_durations = c("-600", "-800"),
    vb = FALSE,
    rq = NULL
)

Arguments
asset_id  Asset number.
types_w_durations  Asset types that have valid durations.
vb  A Boolean value. If TRUE provides verbose output.
rq  An httr2 request object. Default is NULL.

Value
Duration of a file in ms.

Examples
get_file_duration() # default is the test video from databrary.org/volume/1

get_party_as_df  Retrieve Party (Person/Institution) Info as Data Frame.

Description
[Deprecated]
This function has been deprecated and may be removed in a future release. See get_party_by_id() for similar, and more general functionality.
**get_party_by_id**

**Usage**

```r
get_party_as_df(party_id = 6, convert_JSON = TRUE, vb = FALSE)
```

**Arguments**

- `party_id` Party number(s) to retrieve information about.
- `convert_JSON` A Boolean value if TRUE converts the JSON download.
- `vb` A Boolean value if TRUE returns verbose output.

**Value**

A data frame with information about the selected party or parties.

**Examples**

```r
get_party_as_df() # Info about Rick Gilmore.
get_party_as_df(party_id = 8) # Info about NYU.
get_party_as_df(5:7) # Info about Databrary's founders
```

---

**get_party_by_id**  
*Download Information About a Party on Databrary as JSON*

**Description**

Download Information About a Party on Databrary as JSON

**Usage**

```r
get_party_by_id(party_id = 6, vb = FALSE, rq = NULL)
```

**Arguments**

- `party_id` An integer. The party number to retrieve information about.
- `vb` A Boolean value if TRUE returns verbose output.
- `rq` An httr2-style request object. If NULL, then a new request will be generated using make_default_request().

**Value**

A nested list with information about the party. This can be readily parsed by other functions.
get_permission_levels  

Extract Databrary Permission Levels.

Description
Extract Databrary Permission Levels.

Usage

get_permission_levels(vb = FALSE)

Arguments

vb  
A Boolean value. If TRUE provides verbose output.

Value
An array with the permission levels that can be assigned to data.

Examples

get_permission_levels()

get_release_levels  

Show Databrary Release Levels

Description
Show Databrary Release Levels

Usage

get_release_levels(vb = FALSE)
get_session_as_df

Arguments

vb  A Boolean value. If TRUE provides verbose output.

Value

A data frame with Databrary’s release levels.

Examples

get_release_levels()
get_session_by_id  Get Session (Slot) Data From A Databrary Volume

Description

Get Session (Slot) Data From A Databrary Volume

Usage

get_session_by_id(session_id = 9807, vol_id = 1, vb = FALSE, rq = NULL)

Arguments

- **session_id**: An integer indicating a valid session/slot identifier linked to a volume. Default value is 9807, the materials folder for volume 1.
- **vol_id**: An integer indicating the volume identifier. Default is 1.
- **vb**: A logical value. Show verbose feedback. Default is FALSE.
- **rq**: An httr2 request object.

Value

A JSON blob with the session data. If the user has previously logged in to Databrary via `login_db()`, then session(s) that have restricted access can be downloaded, subject to the sharing release levels on those session(s).

Examples

```r
## Not run:
get_session_by_id() # Default is Volume 1, Session 9807, the Materials folder
## End(Not run)
```

get_session_by_name  Get Session (Slot) Data From A Databrary Volume By Session Name.

Description

Get Session (Slot) Data From A Databrary Volume By Session Name.
get_supported_file_types

Usage

get_session_by_name(
    session_name = "Advisory Board Meeting",
    vol_id = 1,
    vb = FALSE,
    rq = NULL
)

Arguments

  session_name  A string. The name of the target session. Defaults to "Advisory Board Meeting", the name of several of the sessions in the public Volume 1.
  vol_id        An integer indicating the volume identifier. Default is 1.
  vb            A logical value. Show verbose feedback. Default is FALSE.
  rq            An httr2 request, such as that generated by make_default_request().

Value

One or more JSON blobs (as lists) whose session name(s) match name in the given volume.

Examples

## Not run:
# Returns multiple JSON blobs with the name "Advisory Board Meeting" in Volume 1.
get_session_by_name()

get_session_by_name(name = 'Top-level materials')

## End(Not run)

get_supported_file_types

Extracts File Types Supported by Databrary.

Description

Extracts File Types Supported by Databrary.

Usage

get_supported_file_types(vb = FALSE)

Arguments

  vb              A Boolean value. If TRUE provides verbose output.
get_video_stats

Value

A data frame with the file types permitted on Databrary.

Examples

get_supported_file_types()

get_video_stats

Report Stats About Videos in a Volume.

Description

"" @description [Superseded] get_video_stats() has been superseded in favor of summarize_videos_in_volume().

Usage

get_video_stats(this_vol_id = 2, vb = FALSE)

Arguments

this_vol_id Target volume number.

vb A boolean value.

Value

A data frame with summary statistics about the videos in the volume.

Examples

## Not run:
get_video_stats() # Retrieves summary statistics about videos in volume 2.

## End(Not run)
**get_volume_by_id**

Get Data From A Databrary Volume

**Usage**

```r
get_volume_by_id(vol_id = 1, vb = FALSE, rq = NULL)
```

**Arguments**

- `vol_id`  
  Volume ID.
- `vb`  
  A logical value. Show verbose messages.
- `rq`  
  An `httr2` request object. If NULL (the default), a new request is generated using `make_default_request()`. To access restricted data, the user must login with a specific request object using `login_db()`.

**Value**

A JSON blob with the volume data. If the user has previously logged in to Databrary via `login_db()`, then volume(s) that have restricted access can be downloaded, subject to the sharing release levels on those volume(s).

**Examples**

```r
get_volume_by_id() # Default is Volume 1
```

---

**HHMMSSmmm_to_ms**

Convert Timestamp String To ms.

**Description**

Convert Timestamp String To ms.

**Usage**

```r
HHMMSSmmm_to_ms(HHMMSSmmm = "01:01:01:333")
```

**Arguments**

- `HHMMSSmmm`  
  a string in the format "HH:MM:SS:mmm"
**Value**

A numeric value in ms from the input string.

**Examples**

```hmmssmmm_to_ms() # 01:01:01:333 in ms```

**Description**

Is This Party An Institution?

**Usage**

```is_institution(party_id = 8, vb = FALSE, rq = NULL)```

**Arguments**

- **party_id**: Databrary party ID
- **vb**: A Boolean value. If TRUE provides verbose output.
- **rq**: An `httr2` request object.

**Value**

TRUE if the party is an institution, FALSE otherwise.

**Examples**

```is_institution() # Is party 8 (NYU) an institution.```
is_person

Description
Is This Party A Person?

Usage
is_person(party_id = 7, vb = FALSE, rq = NULL)

Arguments
party_id Databrary party ID
vb A boolean value. If TRUE provides verbose output.
rq An httr2 request object.

Value
TRUE if the party is a person, FALSE otherwise.

Examples
is_person()

list_affiliates

Description
List Affiliates for A Databrary Party.

Usage
list_affiliates(party_id = 6, report_target_party = FALSE, vb = FALSE)

Arguments
party_id Target volume number.
report_target_party Report data about the target party. Default is FALSE.
vb A Boolean value. If TRUE provides verbose output.
**list_assets_by_type**  

*List Stored Assets (Files) By Type.*

**Description**

[Deprecated]

This function is deprecated and may be removed from a future version of the package. `list_volume_assets()` is preferred.

**Usage**

```r
list_assets_by_type(vol_id = 2, type = "video", vb = FALSE)
```

**Arguments**

- `vol_id` Volume ID
- `type` Data file type, e.g. "video", "pdf", or "csv"
- `vb` A boolean value. If TRUE provides verbose output.

**Value**

A data frame with information about assets in a volume, sorted by type.

**Examples**

```r
list_assets_by_type() # Lists assets in volume 2.
```
**list_assets_in_session**

*Assets In Databrary Session.*

### Description

[Superseded]

This function has been superseded by `list_session_assets()`.

### Usage

```r
list_assets_in_session(session_id = 9807, vb = FALSE)
```

### Arguments

- **session_id**
  - Slot/session ID.
- **vb**
  - A Boolean value. If TRUE provides verbose output.

### Value

A data frame with information about a selected session.

### Examples

```r
list_assets_in_session() # Assets in session 9807 (from Volume 1)
```

---

**list_assets_in_volume**  
*List Assets in a Databrary Volume.*

### Description

[Deprecated]

This function has been deprecated in favor of `list_volume_assets()`.

### Usage

```r
list_assets_in_volume(vol_id = 2, vb = FALSE)
```

### Arguments

- **vol_id**
  - Target volume number.
- **vb**
  - A boolean value.
list_authorized_investigators

Description
List Authorized Investigators at Institution

Usage
list_authorized_investigators(party_id = 12, vb = FALSE, rq = NULL)

Value
A data frame with information about the data formats Databrary supports.

Examples

list_asset_formats()

list_asset_formats  List Stored Assets (Files) By Type.

Description
List Stored Assets (Files) By Type.

Usage
list_asset_formats(vb = FALSE)

Arguments
vb A boolean value. If TRUE provides verbose output.

Value
A data frame with information about all assets in a volume.

Examples

list_assets_in_volume() # Assets in volume 2

list_assets_in_volume() # Assets in volume 2

#
list_containers_records

List Containers and Records In Databrary Volume.

Description

[Deprecated]
This function has been deprecated and may be removed in a future release. See get_volume_by_id() for related functionality.

Usage

list_containers_records(vol_id = 1, vb = FALSE)

Arguments

vol_id Databrary volume number.
vb A Boolean value. If TRUE provides verbose output.

Value

A list with containers and records from the specified volume.

Examples

list_containers_records() # Containers and records from volume 1.
#
List Individual Sponsors

Description

[Deprecated]

This function has been deprecated and may be removed in a future release. See `list_party_sponsors()` for similar functionality.

A party (person or institution) may have a sponsor that is another person or an institution. This function lists the person(s) who sponsor a party.

Usage

`list_individual_sponsors(party_id = 406, vb = FALSE, rq = NULL)`

Arguments

- `party_id`: Party number. Default is 406 (Kasey Soska)
- `vb`: A Boolean value. If TRUE provides verbose output.
- `rq`: An `httr2` request object. If NULL (the default) a request will be generated, but this will only permit public information to be returned.

Value

A data frame with information about a party.

Examples

```r
## Not run:
list_individual_sponsors() # Default is Kasey Soska (party 406)

## End(Not run)
```
list_institutional_sponsors

List Institutional Sponsors For A Party.

Description

[Deprecated]
This function has been deprecated and may be removed in a future release.
A party (person or institution) may have a sponsor that is another person or an institution. This function lists the institution(s) that sponsor a party.

Usage

list_institutional_sponsors(
    party_id = 6, 
    report_target_party = FALSE, 
    vb = FALSE 
)

Arguments

party_id Target volume number.
report_target_party A Boolean value. Print info about the target party.
vb A Boolean value. If TRUE provides verbose output.

Value

A data frame with information about a party’s institutional sponsor(s).

Examples

list_institutional_sponsors() # Defaults to Rick Gilmore (party 6)

list_party

List Data For A Party (person or institution).

Description

[Deprecated]
This function has been deprecated and may be removed in a future release. See get_party_by_id() for similar functionality.
Usage

`list_party(party_id = 8, component = "all", vb = FALSE)`

Arguments

- **party_id**: Target volume number. Default is volume 8 (NYU).
- **component**: Which data to return ‘children’, ‘parents’, or ‘all’. Default is all.
- **vb**: A Boolean value. If TRUE provides verbose output. Default is FALSE.

Value

A data frame with information about a party (person or institution).

Examples

```r
list_party() # Default is New York University (party 8)
```

---

**list_party_affiliates**  
*List Affiliates For A Party*

Description

List Affiliates For A Party

Usage

`list_party_affiliates(party_id = 6, vb = FALSE, rq = NULL)`

Arguments

- **party_id**: Target party ID.
- **vb**: A Boolean value. If TRUE provides verbose output.
- **rq**: An `httr2` request object. Defaults to NULL.

Value

A data frame with information about a party’s affiliates.

Examples

```r
list_party_affiliates() # Default is Rick Gilmore (party 6)
```
list_party_sponsors  List Sponsors For A Party

Description
List Sponsors For A Party

Usage
list_party_sponsors(party_id = 6, vb = FALSE, rq = NULL)

Arguments
- party_id: Target party ID.
- vb: A Boolean value. If TRUE provides verbose output.
- rq: An httr2-style request object. If NULL, then a new request will be generated using make_default_request().

Value
A data frame with information about a party’s sponsors.

Examples
## Not run:
list_party_sponsors() # Default is Rick Gilmore (party 6)
## End(Not run)

list_party_volumes  List Volumes For A Party

Description
List Volumes For A Party

Usage
list_party_volumes(party_id = 6, vb = FALSE, rq = NULL)
list_people

List Info About People.

Description

[Deprecated]
This function has been deprecated and may be removed in a future release. See get_party_by_id() for similar functionality.

Usage

list_people(people_list = 5:7, vb = FALSE)

Arguments

people_list Party number(s) for people to list.
vb A Boolean value if TRUE returns verbose output.

Value

A data frame with information about the people in the specified range of party ID’s.

Examples

list_people() # Lists people with party IDs 5:7, Databrery’s founders
#
list_sessions

Download Information About A Volume or Volume’s Sessions.

Description
[Deprecated] list_sessions() has been deprecated in favor of list_volume_sessions().

Usage
list_sessions(vol_id = 1, vb = FALSE)

Arguments

vol_id  Target volume ID. Defaults to 1.

vb  A boolean value.

Value
Information about sessions in a volume.

Examples

list_sessions() # Lists sessions in volume 1.

list_sessions_in_volume

List Sessions in a Volume.

Description
[Superseded] list_sessions_in_volume() has been superseded in list_sessions().

Usage
list_sessions_in_volume(vol_id = 1, vb = FALSE)

Arguments

vol_id  Target volume ID. Defaults to 1.

vb  Provide verbose output. Default is FALSE.

Value
A data frame with information about the sessions in a volume.
**Examples**

```r
list_sessions_in_volume() # Lists sessions in volume 1.
```

---

**list_session_activity**  
*List Activity History in Databrary Session.*

**Description**

If a user has access to a volume and session, this function returns the history of modifications to that session.

**Usage**

```r
list_session_activity(session_id = 6256, vb = FALSE, rq = NULL)
```

**Arguments**

- `session_id`  
  Selected session/slot number.

- `vb`  
  A Boolean value. If TRUE provides verbose output.

- `rq`  
  An `httr2` request object. Defaults to NULL. To access the activity history on a volume a user has privileges on. Create a request (`rq <- make_default_request()`) login using `make_login_client(rq = rq)`; then run `list_session_activity(session_id = <YOUR_SESSION_ID>, rq = rq)`

**Value**

A list with the activity history on a session/slot.

**Examples**

```r
## Not run:
# The following will only return output if the user has write privileges
# on the session.
list_session_activity(session_id = 6256, vb = FALSE) # Activity for session 6256.
```

## End(Not run)
list_session_assets  List Assets in a Databrary Session.

Description

#' @description [Experimental]

Usage

list_session_assets(session_id = 9807, vb = FALSE, rq = NULL)

Arguments

session_id  An integer. A Databrary session number. Default is 9807, the "materials" folder from Databrary volume 1.

vb  A logical value. Show verbose feedback. Default is FALSE.

rq  An httr2 request object. If NULL, a default request is generated from databraryr::make_default_request().

Details

As of v0.6.3 list_session_assets() replaces an older function that is now named list_volume_session_assets(). The older function requires both a volume ID and a session ID. The new function requires only a session ID.

Value

A data frame with information about all assets in a volume.

Examples

## Not run:
list_session_assets() # Session 9807 in volume 1

## End(Not run)
list_specified_assets_in_session

Lists assets in a given Databrary session (slot).

Description

Lists assets in a given Databrary session (slot).

Usage

list_specified_assets_in_session(
    session_id = 9807,
    media_type = "MPEG-4 video",
    vb = FALSE
)

Arguments

session_id  Slot/session ID.
media_type  A string indicating what type of file.
vb          A Boolean value. If TRUE provides verbose output.

Value

A data frame with the assets in the selected volume and session.

list_sponsors  List Sponsors For A Party

Description

List Sponsors For A Party

Usage

list_sponsors(party_id = 6, vb = FALSE, rq = NULL)

Arguments

party_id  Target party ID.
vb        A Boolean value. If TRUE provides verbose output.
 rq        An http2-style request object. If NULL, then a new request will be generated using make_default_request().
Value

A data frame with information about a party’s sponsors.

Examples

```r
## Not run:
list_sponsors() # Default is Rick Gilmore (party 6)
## End(Not run)
```

Description

[Deprecated] `list_volume()` has been deprecated in lieu of other, more specific, volume-related functions. It will be dropped from a future release.

Usage

```r
list_volume(vol_id = 1, search_str = "", vb = FALSE)
```

Arguments

- `vol_id` Target volume number.
- `search_str` The specific data to retrieve for the volume
- `vb` A Boolean value. If TRUE provides verbose output.

Value

A data frame with information about a volume.

Examples

```r
## Not run:
list_volume() # Default is volume 1
## End(Not run)
```
**list_volume_activity**  
*List Activity In A Databrary Volume*

**Description**

If a user has access to a volume, this command lists the modification history of the volume as a

**Usage**

```
list_volume_activity(vol_id = 1, vb = FALSE, rq = NULL)
```

**Arguments**

- `vol_id`: Selected volume number.
- `vb`: A Boolean value. If TRUE provides verbose output.
- `rq`: An `httr2` request object. Defaults to NULL.

**Value**

A list with the activity history on a volume.

**Examples**

```r
## Not run:
# The following will only return output if the user has write privileges
# on the volume.
list_volume_activity(vol_id = 1) # Activity on volume 1.
## End(Not run)
```

**list_volume_assets**  
*List Assets in Databrary Volume.*

**Description**

List Assets in Databrary Volume.

**Usage**

```
list_volume_assets(vol_id = 1, vb = FALSE, rq = NULL)
```
Arguments

vol_id  Target volume number. Default is 1.
vb  A logical value. Default is FALSE.
rq  An http2 request object. Default is NULL.

Value

A data frame with information about all assets in a volume.

Examples

```r
## Not run:
list_volume_assets()  # Assets in volume 1
## End(Not run)
```

```r
list_volume_excerpts  List Image or Video Excerpts On A Databrary Volume.
```

Description

List Image or Video Excerpts On A Databrary Volume.

Usage

```r
list_volume_excerpts(vol_id = 1, vb = FALSE, rq = NULL)
```

Arguments

vol_id  Target volume number.
vb  A Boolean value. If TRUE provides verbose output.
rq  An http2 request object. Default is NULL.

Value

A list with information about any available excerpts.

Examples

```r
list_volume_excerpts()
```
**list_volume_funding**  
*Lists Funders Associated With a Databrary Volume.*

**Description**
Lists Funders Associated With a Databrary Volume.

**Usage**
```r
list_volume_funding(vol_id = 1, add_id = TRUE, vb = FALSE, rq = NULL)
```

**Arguments**
- **vol_id**  
  Target volume number.
- **add_id**  
  A logical value. Include the volume ID in the output. Default is TRUE.
- **vb**  
  A Boolean value. If TRUE provides verbose output.
- **rq**  
  An `httr2` request object.

**Value**
A data frame with funder information for the volume.

**Examples**
```r
## Not run:
list_volume_funding() # Funding for volume 1
list_volume_funding(vol_id = c(1:10))
## End(Not run)
```

---

**list_volume_links**  
*Retrieves URL Links From A Databrary Volume.*

**Description**
Retrieves URL Links From A Databrary Volume.

**Usage**
```r
list_volume_links(vol_id = 1, vb = FALSE, rq = NULL)
```
list_volume_metadata

Arguments

vol_id  Target volume number.
vb  A Boolean value. If TRUE provides verbose output.
rq  An httr2 request object.

Value

A data frame with the requested data.

Examples

## Not run:
list_volume_links() # Links from volume 1
## End(Not run)

Description

[Deprecated]
This function has been deprecated and may be removed in a future release. See get_volume_by_id() and other list_volume_* functions for related functionality.

Usage

list_volume_metadata(vol_id = 1, vb = FALSE)

Arguments

vol_id  Selected volume number.
vb  A Boolean value. If TRUE provides verbose output.

Value

A data frame with metadata about a volume.

Examples

list_volume_metadata() # List info about volume 1.
list_volume_owners  List Owners of a Databrary Volume.

Description
List Owners of a Databrary Volume.

Usage
list_volume_owners(vol_id = 1, vb = FALSE, rq = NULL)

Arguments

- **vol_id**: Selected volume number. Default is volume 1.
- **vb**: A boolean value. If TRUE provides verbose output.
- **rq**: An `httr2` request object. If NULL (the default) a request will be generated, but this will only permit public information to be returned.

Value

A data frame with information about a volume’s owner(s).

Examples

    list_volume_owners() # Lists information about the owners of volume 1.

---

list_volume_sessions  List Sessions in Databrary Volume.

Description
List Sessions in Databrary Volume.

Usage

    list_volume_sessions(
        vol_id = 1,
        include_vol_data = FALSE,
        vb = FALSE,
        rq = NULL
    )
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vol_id</code></td>
<td>Target volume number.</td>
</tr>
<tr>
<td><code>include_vol_data</code></td>
<td>A Boolean value. Include volume-level metadata or not. Default is FALSE.</td>
</tr>
<tr>
<td><code>vb</code></td>
<td>A Boolean value. Show verbose output. Default is FALSE.</td>
</tr>
<tr>
<td><code>rq</code></td>
<td>An <code>httr2</code> request object. If NULL (the default) a request will be generated, but this will only permit public information to be returned.</td>
</tr>
</tbody>
</table>

Value

A data frame with information about all assets in a volume.

Examples

```r
## Not run:
list_volume_sessions() # Sessions in Volume 1

## End(Not run)
```

Description

#' @description [Experimental]

Usage

```r
list_volume_session_assets(
  vol_id = 1,
  session_id = 9807,
  vb = FALSE,
  rq = NULL
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vol_id</code></td>
<td>Target volume number.</td>
</tr>
<tr>
<td><code>session_id</code></td>
<td>The session number in the selected volume.</td>
</tr>
<tr>
<td><code>vb</code></td>
<td>A boolean value.</td>
</tr>
<tr>
<td><code>rq</code></td>
<td>An <code>httr2</code> request object.</td>
</tr>
</tbody>
</table>
Details

list_volume_session_assets() is a new name for the <v0.6.2 list_session_assets() as function. There is a new list_session_assets() function that does not require the volume ID. The list_volume_session_assets() requires a volume ID.

Value

A data frame with information about all assets in a volume.

Examples

```r
## Not run:
list_volume_session_assets() # Session 9807 in volume 1
## End(Not run)
```

---

list_volume_tags Lists Keywords And Tags For A Volume.

Description

Lists Keywords And Tags For A Volume.

Usage

```r
list_volume_tags(vol_id = 1, vb = FALSE, rq = NULL)
```

Arguments

- `vol_id` Target volume number.
- `vb` A Boolean value. If TRUE provides verbose output.
- `rq` An `httr2` request object.

Value

A data frame with the requested data.

Examples

```r
list_volume_tags()
```
Description

Log In To Databrary.org.

Usage

```r
login_db(
  email = NULL,
  password = NULL,
  store = FALSE,
  overwrite = FALSE,
  vb = FALSE,
  SERVICE = KEYRING_SERVICE,
  rq = NULL
)
```

Arguments

- `email`: Databrary account email address.
- `password`: Databrary password (not recommended as it will displayed as you type).
- `store`: A boolean value. If TRUE store/retrieve credentials from the system keyring/keychain.
- `overwrite`: A boolean value. If TRUE and store is TRUE, overwrite/update stored credentials in keyring/keychain.
- `vb`: A boolean value. If TRUE provides verbose output.
- `SERVICE`: A character label for stored credentials in the keyring. Default is "databrary".
- `rq`: An http request object. Defaults to NULL.

Value

Logical value indicating whether log in is successful or not.

Examples

```r
login_db() # Queries user for email and password interactively.

## Not run:
# The following shows how to use credentials that have been stored previously.

login_db(email = "you@provider.com", store = TRUE)

## End(Not run)```
logout_db  Log Out of Databrary.org.

Description
Log Out of Databrary.org.

Usage
logout_db(vb = FALSE, rq = NULL)

Arguments
vb A boolean value. If TRUE provides verbose output.
rq An httr2 request object. Defaults to NULL.

Value
TRUE if logging out succeeds, FALSE otherwise.

Examples
logout_db()

make_default_request  Set default httr request parameters.

Description
make_default_request sets default parameters for httr requests.

Usage
make_default_request()

Value
An httr2 request object

Examples
make_default_request()
**make_fn_portable**

**Make Portable File Names**

**Description**

Make Portable File Names

**Usage**

```
make_fn_portable(
  fn,
  vb = FALSE,
  replace_regex = "[ &\"\'\()\(\}\{\[\]\+\@#\$%^\*]",
  replacement_char = "_"
)
```

**Arguments**

- **fn**: Databrary party ID
- **vb**: A boolean value. If TRUE provides verbose output.
- **replace_regex**: A character string. A regular expression to capture the "non-portable" characters in fn.
- **replacement_char**: A character string. The character(s) that will replace the non-portable characters.

**Value**

A "cleaned" portable file name

---

**make_login_client**

**Log In To Databrary.org.**

**Description**

Log In To Databrary.org.

**Usage**

```
make_login_client(
  email = NULL,
  password = NULL,
  store = FALSE,
  overwrite = FALSE,
  vb = FALSE,
  SERVICE = KEYRING_SERVICE,
  rq = NULL
)
```
read_csv_data_as_df

---

**Arguments**

- `email` (`Databrary account email address`).
- `password` (`Databrary password (not recommended as it will displayed as you type)`).
- `store` (`A boolean value. If TRUE store/retrieve credentials from the system keyring/keychain`).
- `overwrite` (`A boolean value. If TRUE and store is TRUE, overwrite/ update stored credentials in keyring/keychain`).
- `vb` (`A boolean value. If TRUE provides verbose output`).
- `SERVICE` (`A character label for stored credentials in the keyring. Default is "databrary"`).
- `rq` (`An httr2 request object. Defaults to NULL`).

**Value**

Logical value indicating whether log in is successful or not.

**Examples**

```r
google_books(make_login_client()) # Queries user for email and password interactively.

## Not run:
# The following shows how to use credentials that have been stored previously.
make_login_client(email = "you@provider.com", store = TRUE)
## End(Not run)
```

---

**read_csv_data_as_df**

Download Stored CSV As Data Frame.

---

**Description**

[Deprecated]

This function has been deprecated and may be removed in a future release. See `download_session_asset()` for more similar, but more general functionality.

**Usage**

```r
read_csv_data_as_df(session_id = 9807, asset_id = 153108, vb = FALSE)
```

**Arguments**

- `session_id` (`Session ID within some volume`).
- `asset_id` (`Asset ID within the session`).
- `vb` (`A Boolean value. If TRUE provides verbose output`).
**Details**

If a given Databrary session has a CSV file, this function will try to download it as a data frame.

**Value**

A data frame with the CSV data.

**Examples**

```r
read_csv_data_as_df() # Downloads a CSV depicting Databrary growth from volume 1
```

---

**search_for_funder**  
*Report Information About A Funder.*

**Description**

Report Information About A Funder.

**Usage**

```r
search_for_funder(
  search_string = "national+science+foundation",
  vb = FALSE,
  rq = NULL
)
```

**Arguments**

- `search_string`  
  String to search.
- `vb`  
  A Boolean value. If TRUE provides verbose output.
- `rq`  
  An `httr2` request object. Default is NULL.

**Value**

A data frame with information about the funder.

**Examples**

```r
search_for_funder("national+science+foundation")
```
search_for_keywords  

*Search For Keywords in Databrary Volumes.*

**Description**

Search For Keywords in Databrary Volumes.

**Usage**

```r
search_for_keywords(search_string = "locomotion", vb = FALSE, rq = NULL)
```

**Arguments**

- **search_string**: String to search.
- **vb**: A Boolean value. If TRUE provides verbose output.
- **rq**: An `httr2` request object. Default is NULL.

**Value**

A list with the volumes that contain the keyword.

**Examples**

```r
## Not run:
search_for_keywords() # searches for volumes with "locomotion" as a keyword.

search_for_keywords("adult") # searches for volumes with "adult" as a keyword.
## End(Not run)
```

search_for_tags  

*Search For Tags on Volumes or Sessions.*

**Description**

Search For Tags on Volumes or Sessions.

**Usage**

```r
search_for_tags(search_string = "ICIS", vb = FALSE, rq = NULL)
```

**Arguments**

- **search_string**: String to search.
- **vb**: A Boolean value. If TRUE provides verbose output.
- **rq**: An `httr2` request object. Default is NULL.
summarize_videos_in_volume

Value

An array of tags that match the tag_string.

Examples

## Not run:
search_for_tags() # Searches for volumes that have the tag "ICIS"

## End(Not run)

summarize_videos_in_volume

Summarize Data About Videos in Databrary Volume.

Description

[Deprecated]

This function is deprecated and may be removed from a future version of the package. list_volume_assets() is preferred.

Usage

summarize_videos_in_volume(vol_id = 2, vb = FALSE)

Arguments

  vol_id  An integer. Selected volume number or numbers.
  vb     A Boolean value. If TRUE provides verbose output.

Value

Data frame with the vol_id, number of videos, and total hours.

Examples

## Not run:
summarize_videos_in_volume(vol_id = 2) # Summary of volume 2
summarize_videos_in_volume(vol_id = c(4, 5)) # Summarize volumes 4 & 5

## End(Not run)
Index

* datasets
  DATABRARY_API, 4
assign_constants, 3
check_ssl_certs, 4
DATABRARY_API, 4
download_asset, 5
download_party, 6
download_party_avatar, 6
download_session_asset, 7
download_session_assets_fr_df, 8
download_session_csv, 9
download_session_zip, 10
download_single_session_asset_fr_df, 11
download_video, 13
download_volume_zip, 14
get_asset_segment_range, 15
get_assets_from_session, 15
GET_db_contents, 16
get_db_stats, 17
get_file_duration, 18
get_party_as_df, 18
get_party_by_id, 19
get_permission_levels, 20
get_release_levels, 20
get_session_as_df, 21
get_session_by_id, 22
get_session_by_name, 22
get_supported_file_types, 23
get_video_stats, 24
get_volume_by_id, 25
HHMMSSmmm_to_ms, 25
is_institution, 26
is_person, 27
list_affiliates, 27
list_asset_formats, 30
list_assets_by_type, 28
list_assets_in_session, 29
list_assets_in_volume, 29
list_authorizedInvestigators, 30
list_containers_records, 31
list_individual_sponsors, 32
list_institutional_sponsors, 33
list_party, 33
list_party_affiliates, 34
list_party_sponsors, 35
list_party_volumes, 35
list_people, 36
list_session_activity, 38
list_session_assets, 39
list_sessions, 37
list_sessions_in_volume, 37
list_specified_assets_in_session, 40
list_sponsors, 40
list_volume, 41
list_volume_activity, 42
list_volume_assets, 42
list_volume_excerpts, 43
list_volume_funding, 44
list_volume_links, 44
list_volume_metadata, 45
list_volume_owners, 46
list_volume_session_assets, 47
list_volume_sessions, 46
list_volume_tags, 48
login_db, 49
logout_db, 50
make_default_request, 50
make_fn_portable, 51
make_login_client, 51
read_csv_data_as_df, 52

56
INDEX

search_for_funder, 53
search_for_keywords, 54
search_for_tags, 54
summarize_videos_in_volume, 55