Package: sharepointr (via r-universe)

October 21, 2024

Type Package

Title Read and Write from SharePoint Sites

Version 0.1.0.9000

Maintainer Eli Pousson <eli.pousson@gmail.com>

Description A R package extending Microsoft365R to make it easier to read, write, and work with SharePoint items, lists, and plans.

License MIT + file LICENSE

URL https://github.com/elipousson/sharepointr, https://elipousson.github.io/sharepointr/

BugReports https://github.com/elipousson/sharepointr/issues

Imports cli, glue, httr2, Microsoft365R, rlang (>= 1.1.0), stringr, utils, vctrs

Suggests fs, knitr, officer, openxlsx2, rappdirs, readr, readxl, rmarkdown, sf, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Language en-US

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

RemoteUrl https://github.com/elipousson/sharepointr

RemoteRef HEAD

RemoteSha 8ff3dabecd61c253a4bdc35920dc0decfbf4310a

Contents

`AzureR_config` *List or delete files from the AzureR configuration directory*

Description

`AzureR_config_ls()` uses `rappdirs::user_config_dir()` and `fs::dir_ls()` to list files from the AzureR configuration directory. `AzureR_config_delete()` uses `fs::file_delete()` to remove the "graph_logins.json" configuration file if needed. Use this function with caution but it may be an option to address a "Unable to refresh token" error.

Usage

```
AzureR_config_ls(path = NULL, glob = "*.json")

AzureR_config_delete(path = NULL, filename = "graph_logins.json")
```

Arguments

`path` Path to configuration directory for AzureR package where the JSON file for graph_logins is stored. If `NULL`, path is set with `rappdirs::user_config_dir("AzureR")`.

`glob` A wildcard aka globbing pattern (e.g. `*.csv`) passed on to `grep()` to filter paths.

`filename` Filename to delete from configuration directory. Defaults to "graph_logins.json". Set to `NULL` if path contains a file name.

`create_sp_list_items` *Create or update list items*

Description

Create or update list items

Usage

```
create_sp_list_items(
  data,
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  ...,
  allow_display_nm = TRUE,
  .id = "id",
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  check_fields = TRUE,
  sync_fields = FALSE,
  strict = FALSE,
  call = caller_env()
)

update_sp_list_items(
  data,
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  ...,
  .id = "id",
  allow_display_nm = TRUE,
  call = caller_env()
)
```

```
update_sp_list_item(
  ...,
  .data = NULL,
  id = NULL,
  sp_list_item = NULL,
  na_fields = c("drop", "replace"),
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  call = caller_env()
)
```

Arguments

`data` Required. A data frame to import as items to the supplied or identified SharePoint list.

`list_name, list_id` SharePoint List name or ID string.

`sp_list` A `ms_list` object. If supplied, `list_name`, `list_id`, `site_url`, `site`, `drive_name`, `drive_id`, `drive`, and any additional parameters passed to `...` are all ignored.

`...` Additional parameters passed to `get_sp_site()` or `Microsoft365R::get_sharepoint_site()`.

`allow_display_nm` If `TRUE`, allow data to use list field display names instead of standard names. Note this requires a separate API call so may result in a slower request.

`.id` Name of column in data to use for item ID values. Defaults to "id".

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`site` A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.

`drive_name, drive_id` SharePoint Drive name or ID passed to `get_drive` method for SharePoint site object.

`drive` A `ms_drive` object. If `drive` is supplied, `drive_name` and `drive_id` are ignored.

`check_fields` If `TRUE` (default), column names for the input data are matched to the fields of the list object. If `FALSE`, the function will error if any column names can't be matched to a field in the supplied SharePoint list.

`sync_fields` If `TRUE`, use the `sync_fields` method to sync the fields of the local `ms_list` object with the fields of the SharePoint List source before retrieving list metadata.

`strict` Not yet implemented as of 2024-08-12. If `TRUE`, all column names in data must be matched to field names in the supplied SharePoint list. If `FALSE` (default), unmatched columns will be dropped with a warning.

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

`.data` A list or data frame with fields to update.

`sp_list_item` Optional. A SharePoint list item object to use.

`na_fields` How to handle `NA` fields in input data. One of "drop" (remove NA fields before updating list items, leaving existing values in place) or "replace" (overwrite existing list values with new replacement NA values).

Details

Validation of data with with `create_sp_list_items()`

The handling of item creation when column names in `data` do not match the fields names in the supplied list includes a few options:

- If no names in data match fields in the list, the function errors and lists the field names.
- If all names in data match fields in the list the records are created. Any fields that do not have corresponding names in data remain blank.
- If any names in data do not match fields in the list, by default, those columns are dropped before adding items to the list.
- If `strict = TRUE` and any names in data to not match fields, the function errors.

Examples

```
sp_list_url <- "<SharePoint List URL with a Name field>"

if (is_sp_url(sp_list_url)) {
  create_sp_list_items(
    data = data.frame(
      Name = c("Jim", "Jane", "Jayden")
    ),
    list_name = sp_list_url
  )
}
```

`delete_sp_item` *Delete SharePoint items (files and directories)*

Description

`delete_sp_item()` deletes items including files and directories using the `delete` method for . By default `confirm = TRUE`, which requires the user to respond to a prompt: "Do you really want to delete the drive item ...? (yes/No/cancel)" to continue.

Usage

```
delete_sp_item(
  path = NULL,
  confirm = TRUE,
  by_item = FALSE,
  ...,
  item_id = NULL,
  item_url = NULL,
  item = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  site_url = NULL,
  call = caller_env()
)
```

Arguments

`path` A SharePoint file URL or the relative path to a file located in a SharePoint drive. If input is a relative path, the string should *not* include the drive name. If input is a shared file URL, the text "Shared " is removed from the start of the SharePoint drive name by default. If file is a document URL, the `default_drive_name` argument is used as the `drive_name` and the `item_id` is extracted from the URL.

`confirm` If `TRUE`, confirm before deleting item. If `TRUE` and session is not interactive, the function will error.

`by_item` For business OneDrive or SharePoint document libraries, you may need to set `by_item = TRUE` to delete the contents of a folder depending on the policies set up by your SharePoint administrator policies. Note, that this method can be slow for large folders.

`...` Arguments passed on to `get_sp_item`

`drive_name,drive_id` SharePoint drive name or ID.

`as_data_frame` If `TRUE`, return a data frame. If `FALSE` (default), return a `ms_item` or `ms_item_properties` object.

`item_id` A SharePoint item ID passed to the `itemid` parameter of the `get_item` method for `ms_drive` objects.

`item_url` A SharePoint item URL used to parse the item ID, drive name, and site URL.

`item` A `ms_drive_item` class object. Optional if path or other parameters to get an SharePoint item are supplied.

`drive_name, drive_id` SharePoint drive name or ID.

`drive` A `ms_drive` object. If drive is supplied, `drive_name`, `site_url`, and any additional parameters passed to `...` are ignored.

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

Details

Trouble-shooting errors

If you get the error: "The resource you are attempting to access is locked", you or another user may have the file or a file within the directory open for editing. Close the file and try deleting the item again.

If you get the error: "Request was cancelled by event received. If attempting to delete a non-empty folder, it's possible that it's on hold." set `by_item` = `TRUE` and try again.

download_sp_item *Download one or more items from SharePoint to a file or folder*

Description

`download_sp_item()` wraps the `download` method for SharePoint items making it easier to download items based on a shared file URL or document URL.

Usage

```
download_sp_item(
  path = NULL,
  new_path = "",
  ...,
  item_id = NULL,
  item_url = NULL,
  item = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  site_url = NULL,
  dest = NULL,
  overwrite = FALSE,
  recursive = FALSE,
  parallel = FALSE,
  call = caller_env()
)

download_sp_file(file, new_path = "", ..., call = caller_env())
```

Arguments

path, file
: Required. A SharePoint shared file URL, document URL, or, if item_id is supplied, a file name to use with path to set dest with location and filename for downloaded item.

new_path
: Path to directory for downloaded item. Optional if dest is supplied. If path contains a file name, the item will be downloaded using that file name instead of the file name of the original item. If new_path refers to a nonexistent directory and the item is a file, the directory will be silently created using fs::dir_create().

...
: Arguments passed on to get_sp_item

  drive_name,drive_id SharePoint drive name or ID.

  properties If TRUE, use the get_item_properties method and return item properties instead of the item.

  as_data_frame If TRUE, return a data frame. If FALSE (default), return a ms_item or ms_item_properties object.

item_id
: A SharePoint item ID passed to the itemid parameter of the get_item method for ms_drive objects.

item_url
: A SharePoint item URL used to parse the item ID, drive name, and site URL.

item
: A ms_drive_item class object. Optional if path or other parameters to get an SharePoint item are supplied.

drive_name, drive_id
: SharePoint drive name or ID.

drive
: A ms_drive object. If drive is supplied, drive_name, site_url, and any additional parameters passed to ... are ignored.

site_url
: A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

dest, overwrite, recursive, parallel
: Parameters passed to download method for ms_drive_item object.

call
: The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Details

The default value for path is "" so, by default, SharePoint items are downloaded to the current working directory. Set overwrite = TRUE to allow this function to overwrite an existing file. download_sp_file() is identical except for the name of the path parameter (which is file instead of path).

Note, if the selected item is a folder and recurse = TRUE, it may take some time to download the enclosed items and {Microsoft365R} does not provide a progress bar for that operation.

Value

Invisibly returns the input dest or the dest parsed from the input path or item properties.

Batch downloads for SharePoint items

If provided with a vector of paths or a vector of item ID values, download_sp_item() can execute a batch download on a set of files or folders. Make sure to supply a vector to new_path or dest vector with the directory names or file names to use as a destination for the downloads. With either option, you must supply a drive, a drive_name and site, or a drive_url. You can also pass a bare list of items and the value for the dest can be inferred from the item properties.

Examples

```
# Download a single directory

sp_dir_url <- "<SharePoint directory url>"

new_path <- "local file path"

if (is_sp_url(sp_dir_url)) {
  download_sp_item(
    sp_dir_url,
    new_path = new_path,
    recursive = TRUE
  )
}

# Batch download multiple directories from a SharePoint Drive

sp_drive_url <- "<SharePoint Drive url>"

if (is_sp_url(sp_drive_url)) {
  drive <- get_sp_drive(drive_name = sp_drive_url)

  drive_dir_list <- sp_dir_info(
    drive = drive,
    recurse = TRUE,
    type = "dir"
  )

  download_sp_item(
    item_id = drive_dir_list$id,
    dest = drive_dir_list$name,
    recursive = TRUE,
    drive = drive
  )
}
```

---

download_sp_list *Download a SharePoint List*

Description

download_sp_list() downloads a SharePoint list to a CSV or XLSX file.

Usage

```
download_sp_list(
  ...,
  new_path = "",
  sp_list = NULL,
  fileext = "csv",
  keep_list_cols = c("createdBy", "lastModifiedBy"),
  call = caller_env()
)
```

Arguments

... Arguments passed on to get_sp_list

list_name,list_id SharePoint List name or ID string.

as_data_frame If TRUE, return a data frame with a "ms_list" column. get_sp_list() returns a 1 row data frame and list_sp_lists() returns a data frame with n rows or all lists available for the SharePoint site or drive. Defaults to FALSE. Ignored is metadata = TRUE as list metadata is always returned as a data frame.

metadata If TRUE, get_sp_list() applies the get_column_info method to the returned SharePoint list and returns a data frame with column metadata for the list.

drive_name,drive_id SharePoint Drive name or ID passed to get_drive method for SharePoint site object.

drive A ms_drive object. If drive is supplied, drive_name and drive_id are ignored.

site_url A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

site A ms_site object. If site is supplied, site_url, site_name, and site_id are ignored.

new_path Optional path to new file. If not new_path provided, the file name is pulled from the name of the SharePoint list using the provided fileext. If new_path is provided, fileext is ignored.

sp_list SharePoint list object. If supplied, all parameters supplied to ... are ignored.

fileext File extension to use for output file. Must be "csv" or "xlsx".

keep_list_cols Column names for those columns to maintain in a list format instead of attempting to convert to a character vector.

call The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

get_sp_group *Get SharePoint group for a site or list group members*

Description

get_sp_group() gets the group associated with an individual site using the get_group method. list_sp_group_members() lists members in the group. Note that, as of February 1, 2024, the returned member data frame when as_data_frame = TRUE contains a large number of list columns that could be coerced into character columns. This should be addressed in a later update.

Usage

```
get_sp_group(
  site_url = NULL,
  site_name = NULL,
  site_id = NULL,
  ...,
  site = NULL,
  call = caller_env()
)

list_sp_group_members(
  site_url = NULL,
  site_name = NULL,
  site_id = NULL,
  ...,
  as_data_frame = TRUE,
  call = caller_env()
)
```

Arguments

site_url
: A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

site_name, site_id
: Site name or ID of the SharePoint site as an alternative to the SharePoint site URL. Exactly one of site_url, site_name, and site_id must be supplied.

...
: Additional parameters passed to get_sp_site() or Microsoft365R::get_sharepoint_site().

site
: A ms_site object. If site is supplied, site_url, site_name, and site_id are ignored.

call
: The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

`as_data_frame` If `TRUE` (default), converted list of members into a data frame with a list column named `az_user` that contains the member list and properties converted into columns.

`get_sp_item` *Get a SharePoint item or item properties*

Description

`get_sp_item()` wraps the `get_item` method for `ms_drive` objects and returns a `ms_drive_item` object by default. `get_sp_item_properties()` uses the `get_item_properties` method (also available by setting `properties = TRUE` for `get_sp_item()`).

Additional parameters in ... are passed to `get_sp_drive()` by `get_sp_item()` or to `get_sp_item()` by `get_sp_item_properties()` or `delete_sp_item()`.

Usage

```
get_sp_item(
  path = NULL,
  item_id = NULL,
  item_url = NULL,
  ...,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  site_url = NULL,
  properties = FALSE,
  as_data_frame = FALSE,
  call = caller_env()
)

get_sp_item_properties(
  path = NULL,
  item_id = NULL,
  item_url = NULL,
  ...,
  drive = NULL,
  drive_name = NULL,
  drive_id = NULL,
  site_url = NULL,
  as_data_frame = FALSE,
  call = caller_env()
)
```

Arguments

`path` A SharePoint file URL or the relative path to a file located in a SharePoint drive. If input is a relative path, the string should *not* include the drive name. If input is a shared file URL, the text "Shared " is removed from the start of the SharePoint drive name by default. If file is a document URL, the `default_drive_name` argument is used as the `drive_name` and the `item_id` is extracted from the URL.

`item_id` A SharePoint item ID passed to the `itemid` parameter of the `get_item` method for `ms_drive` objects.

`item_url` A SharePoint item URL used to parse the item ID, drive name, and site URL.

`...` Arguments passed on to `get_sp_drive`

- `drive_name,drive_id` SharePoint Drive name or ID passed to `get_drive` method for SharePoint site object.
- `drive_url` A SharePoint Drive URL to parse for a Drive name and other information. If `drive_name` is a URL, it is used as `drive_url`.
- `default_drive_name` Drive name string used only if input is a document URL and drive name is not part of the URL. Defaults to `getOption("sharepointr.default_drive_name` `"Documents")`
- `cache` If `TRUE`, cache drive to a file using `cache_sp_drive()`.
- `refresh` If `TRUE`, get a new drive even if the existing drive is cached as a local option. If `FALSE`, use the cached `ms_drive` object if it exists.
- `cache_file` File name for cached drive if `cache = TRUE`. Defaults to a option set with `sharepointr.cache_file_drive` (which defaults to `"sp_drive.rds"`).
- `site` A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.
- `overwrite` If `TRUE`, replace the existing cached object named by `cache_file` with the new object. If `FALSE`, error if a cached file with the same `cache_file` name already exists.

`drive_name, drive_id`
SharePoint drive name or ID.

`drive` A `ms_drive` object. If drive is supplied, `drive_name`, `site_url`, and any additional parameters passed to `...` are ignored.

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`properties` If `TRUE`, use the `get_item_properties` method and return item properties instead of the item.

`as_data_frame` If `TRUE`, return a data frame. If `FALSE` (default), return a `ms_item` or `ms_item_properties` object.

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

See Also

Microsoft365R::ms_drive_item

Examples

```
sp_item_url <- "<SharePoint item url>"

if (is_sp_url(sp_item_url)) {
  get_sp_item(
    item_url = sp_item_url
  )
}
```

`list_sp_drives` *List drives for a SharePoint site*

Description

list_sp_drives() loads a SharePoint site uses the list_drives method to returns a data frame with a list column or ms_drive objects or, if as_data_frame = FALSE. This is helpful if a drive has been renamed and can't easily be identified using a Drive URL alone.

Usage

```
list_sp_drives(
  ...,
  site = NULL,
  filter = NULL,
  n = NULL,
  as_data_frame = TRUE,
  call = caller_env()
)
```

Arguments

... Arguments passed on to get_sp_site

site_url A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

site_name,site_id Site name or ID of the SharePoint site as an alternative to the SharePoint site URL. Exactly one of site_url, site_name, and site_id must be supplied.

refresh If TRUE, get a new site even if the existing site is cached as a local option. If FALSE, use the cached ms_site object.

cache If TRUE, cache site to a file using cache_sp_site().

cache_file File name for cached file if cache = TRUE. Defaults to "sp_site.rds" or option set with sharepointr.cache_file_site.

overwrite If TRUE, replace the existing cached object named by cache_file with the new object. If FALSE, error if a cached file with the same cache_file name already exists.

site
: A ms_site object. If site is supplied, site_url, site_name, and site_id are ignored.

filter
: Filter to apply to query

n
: Max number of drives to return

as_data_frame
: If TRUE (default), return list as a data frame.

call
: The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

See Also

Microsoft365R::ms_site

list_sp_item_versions *List versions for a SharePoint item*

Description

This is an implementation of a new method for a ms_item object using the do_operation method directly. The intent is to add a list_versions method back to the Microsoft365R package which may include changes to the functionality and output of this function.

Usage

```
list_sp_item_versions(..., sp_item = NULL, as_data_frame = TRUE)
```

Arguments

...
: Arguments passed on to get_sp_item

path A SharePoint file URL or the relative path to a file located in a SharePoint drive. If input is a relative path, the string should *not* include the drive name. If input is a shared file URL, the text "Shared " is removed from the start of the SharePoint drive name by default. If file is a document URL, the default_drive_name argument is used as the drive_name and the item_id is extracted from the URL.

item_id A SharePoint item ID passed to the itemid parameter of the get_item method for ms_drive objects.

item_url A SharePoint item URL used to parse the item ID, drive name, and site URL.

drive_name,drive_id SharePoint drive name or ID.

drive A ms_drive object. If drive is supplied, drive_name, site_url, and any additional parameters passed to ... are ignored.

properties If TRUE, use the get_item_properties method and return item properties instead of the item.

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

`sp_item` SharePoint item to get versions for. Additional parameters passed to `...` are ignored if sp_item is supplied.

`as_data_frame` If `TRUE`, return a data frame of versions. If `FALSE`, return a list.

Value

A data frame if `as_data_frame` = `TRUE` or a list if `FALSE`.

`list_sp_pages` *List SharePoint pages or get a single SharePoint page*

Description

`list_sp_pages()` returns a list of SharePoint pages associated with a specified SharePoint site. `get_sp_page()` returns a single SharePoint page.

Usage

```
list_sp_pages(
  ...,
  site = NULL,
  page_type = c("sitePage", "page"),
  as_data_frame = TRUE,
  call = caller_env()
)

get_sp_page(page_url = NULL, page_id = NULL, ..., site = NULL)
```

Arguments

`...` Arguments passed on to `get_sp_site`

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`site_name,site_id` Site name or ID of the SharePoint site as an alternative to the SharePoint site URL. Exactly one of `site_url`, `site_name`, and `site_id` must be supplied.

`refresh` If `TRUE`, get a new site even if the existing site is cached as a local option. If `FALSE`, use the cached `ms_site` object.

`cache` If `TRUE`, cache site to a file using `cache_sp_site()`.

cache_file File name for cached file if cache = TRUE. Defaults to "sp_site.rds" or option set with sharepointr.cache_file_site.

overwrite If TRUE, replace the existing cached object named by cache_file with the new object. If FALSE, error if a cached file with the same cache_file name already exists.

site Optional ms_site object to use. If not provided, the ... arguments are passed to get_sp_site().

page_type Page type to request. One of "sitePage" or "page".

as_data_frame If TRUE, return a data frame with details on the SharePoint site pages. If FALSE, return a list.

call The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

page_url, page_id
SharePoint page URL or ID.

read_sharepoint *Read a SharePoint item based on a file URL or file name and site details*

Description

read_sharepoint() is designed to download a SharePoint item to a temporary folder and read the file based on the file extension. If a function is provided to .f, it is used to read the downloaded file. If not, class of the returned object depends on the file extension of the file parameter.

- If the file is a csv, csv2, or tsv file or a Microsoft Excel file (xlsx or xls), readr::read_delim() or readxl::read_excel() is used to return a data frame.
- If the file is a rds file, readr::read_rds() is used to return the saved object.
- If the file is a pptx or docx file, officer::read_docx() or officer::read_pptx() are used to read the file into a rdocx or rpptx object.
- If the file has a "gpkg", "geojson", "kml", "gdb", or "zip" file extension, sf::read_sf() is used to read the file into a sf data frame.

If the file has none of these file extensions, an attempt is made to read the file with readr::read_lines().

Usage

```
read_sharepoint(
  file,
  ...,
  .f = NULL,
  new_path = tempdir(),
  overwrite = TRUE,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  site_url = NULL,
  site_name = NULL,
  site_id = NULL,
  site = NULL
)
```

Arguments

`file` Required. A SharePoint shared file URL, document URL, or, if `item_id` is supplied, a file name to use in combination with `new_path` to set `dest` with location and filename for downloaded item.

`...` Additional parameters passed to one of the functions identified in the description or supplied to `.f`

`.f` Optional function to use to read file downloaded from SharePoint.

`new_path` Path to directory for downloaded item. Optional if `dest` is supplied. If path contains a file name, the item will be downloaded using that file name instead of the file name of the original item. If `new_path` refers to a nonexistent directory and the item is a file, the directory will be silently created using `fs::dir_create()`.

`overwrite` If `TRUE`, replace the existing cached object named by `cache_file` with the new object. If `FALSE`, error if a cached file with the same `cache_file` name already exists.

`drive_name, drive_id` SharePoint drive name or ID.

`drive` A `ms_drive` object. If drive is supplied, `drive_name`, `site_url`, and any additional parameters passed to `...` are ignored.

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`site_name, site_id` Site name or ID of the SharePoint site as an alternative to the SharePoint site URL. Exactly one of `site_url`, `site_name`, and `site_id` must be supplied.

`site` A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.

`sp_dir_create` *Create SharePoint folders*

Description

`sp_dir_create()` is a wrapper for the `create_folder` method that handles character vectors. If `drive_name` is a folder URL and `relative` is `TRUE`, the values for `path` are appended to the file path parsed from the url.

Usage

```
sp_dir_create(
  path,
  ...,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  relative = FALSE,
  call = caller_env()
)
```

Arguments

`path` A character vector of one or more paths.

`...` Arguments passed on to `get_sp_drive`

- `drive_url` A SharePoint Drive URL to parse for a Drive name and other information. If `drive_name` is a URL, it is used as `drive_url`.
- `default_drive_name` Drive name string used only if input is a document URL and drive name is not part of the URL. Defaults to `getOption("sharepointr.default_drive_name` `"Documents")`
- `cache` If `TRUE`, cache drive to a file using `cache_sp_drive()`.
- `refresh` If `TRUE`, get a new drive even if the existing drive is cached as a local option. If `FALSE`, use the cached `ms_drive` object if it exists.
- `cache_file` File name for cached drive if `cache` = `TRUE`. Defaults to a option set with `sharepointr.cache_file_drive` (which defaults to `"sp_drive.rds"`).
- `site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.
- `site` A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.
- `overwrite` If `TRUE`, replace the existing cached object named by `cache_file` with the new object. If `FALSE`, error if a cached file with the same `cache_file` name already exists.

`drive_name, drive_id` SharePoint Drive name or ID passed to `get_drive` method for SharePoint site object.

`drive` A `ms_drive` object. If `drive` is supplied, `drive_name` and `drive_id` are ignored.

`relative` If `TRUE` and `drive_name` is a folder URL, the values for `path` are appended to the file path parsed from the url. If `relative` is a character vector, it must be length 1 or the same length as path and appended to path as a vector of parent directories. The second option takes precedence over any file path parsed from the url.

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

Examples

```
drive_url <- "<link to SharePoint drive>"

if (is_sp_url(drive_url)) {
  sp_dir_create(
    path = "parent_folder/subfolder",
    drive_name = drive_url
  )

  sp_dir_create(
    path = c("subfolder1", "subfolder2", "subfolder3"),
    relative = "parent_folder",
    drive_name = drive_url
  )
}

dir_url <- "<link to SharePoint directory>"

if (is_sp_url(dir_url)) {
  sp_dir_create(
    path = c("subfolder1", "subfolder2", "subfolder3"),
    drive_name = dir_url,
    relative = TRUE
  )
}
```

`sp_dir_info` *List SharePoint files and folders*

Description

`sp_dir_info()` is a wrapper for the `list_files` method with some additional features based on `fs::dir_info()`. `sp_dir_ls()` returns a character vector and does not yet include support for the `recurse` argument. If `{fs}` is installed, the size column is formatted using `fs::as_fs_bytes()` and an additional "type" factor column is added with values for directory and file.

Usage

```
sp_dir_info(
  path = NULL,
  ...,
  info = "partial",
  full_names = TRUE,
  pagesize = 1000,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  recurse = FALSE,
  type = "any",
  regexp = NULL,
  invert = FALSE,
  perl = FALSE,
  call = caller_env()
)

sp_dir_ls(
  path = NULL,
  ...,
  full_names = FALSE,
  pagesize = 1000,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  type = "any",
  regexp = NULL,
  invert = FALSE,
  perl = FALSE,
  call = caller_env()
)
```

Arguments

`path` Path to directory or folder. SharePoint folder URLs are allowed. If `NULL`, path is set to default "/".

`...` Arguments passed on to `get_sp_drive`

- `drive_url` A SharePoint Drive URL to parse for a Drive name and other information. If `drive_name` is a URL, it is used as `drive_url`.
- `default_drive_name` Drive name string used only if input is a document URL and drive name is not part of the URL. Defaults to `getOption("sharepointr.default_drive_name` `"Documents")`
- `cache` If `TRUE`, cache drive to a file using `cache_sp_drive()`.
- `refresh` If `TRUE`, get a new drive even if the existing drive is cached as a local option. If `FALSE`, use the cached `ms_drive` object if it exists.
- `cache_file` File name for cached drive if `cache = TRUE`. Defaults to a option set with `sharepointr.cache_file_drive` (which defaults to `"sp_drive.rds"`).
- `site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.
- `site` A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.
- `overwrite` If `TRUE`, replace the existing cached object named by `cache_file` with the new object. If `FALSE`, error if a cached file with the same `cache_file` name already exists.

`info` The information to return: "partial", "name" or "all". If "partial", a data frame is returned containing the name, size, ID and whether the item is a file or folder. If "all", a data frame is returned containing all the properties for each item (this can be large).

`full_names` If `TRUE` (default), return the full file path as the name for each item.

`pagesize` Maximum number of items to return. Defaults to 1000. Decrease if you are experiencing timeouts.

`drive_name`, `drive_id`
SharePoint Drive name or ID passed to `get_drive` method for SharePoint site object.

`drive` A `ms_drive` object. If `drive` is supplied, `drive_name` and `drive_id` are ignored.

`recurse` If `TRUE`, get info for each directory at the supplied path and combine this info with the item info for the supplied path.

`type` Type of item to return. Can be "any", "file", or "directory". "directory" is not a supported option for `sp_dir_ls()`

`regexp` Regular expression passed to `grep()` and used to filter the paths before they are returned.

`invert` logical. If `TRUE` return indices or values for elements that do *not* match.

`perl` logical. Should Perl-compatible regexps be used?

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

Examples

```
dir_url <- "<link to SharePoint directory or drive>"

if (is_sp_url(dir_url)) {
  sp_dir_info(
    path = dir_url
  )

  sp_dir_ls(
    path = dir_url
  )
}
```

`sp_drive` *Get a SharePoint drive or set a default SharePoint drive*

Description

`get_sp_drive()` wraps the `get_drive` method returns a `ms_drive` object.

`cache_sp_drive()` allows you to cache a default SharePoint drive for use by other functions. Additional parameters in `...` are passed to `get_sp_drive()`.

Usage

```
get_sp_drive(
  drive_name = NULL,
  drive_id = NULL,
  drive_url = NULL,
  properties = FALSE,
  ...,
  site_url = NULL,
  site = NULL,
  default_drive_name = getOption("sharepointr.default_drive_name", "Documents"),
  cache = FALSE,
  refresh = TRUE,
  overwrite = FALSE,
  cache_file = getOption("sharepointr.cache_file_drive", "sp_drive.rds"),
  call = caller_env()
)

cache_sp_drive(
  ...,
  drive = NULL,
  cache_file = getOption("sharepointr.cache_file_drive", "sp_drive.rds"),
  overwrite = FALSE,
  call = caller_env()
)
```

Arguments

`drive_name, drive_id`
: SharePoint Drive name or ID passed to `get_drive` method for SharePoint site object.

`drive_url`
: A SharePoint Drive URL to parse for a Drive name and other information. If `drive_name` is a URL, it is used as `drive_url`.

`properties`
: If `TRUE`, return the drive properties instead of the `ms_drive` object. Defaults to `FALSE`.

`...`
: Arguments passed on to `get_sp_site`

  `site_name,site_id` Site name or ID of the SharePoint site as an alternative to the SharePoint site URL. Exactly one of `site_url`, `site_name`, and `site_id` must be supplied.

`site_url`
: A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`site`
: A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.

`default_drive_name`
: Drive name string used only if input is a document URL and drive name is not part of the URL. Defaults to `getOption("sharepointr.default_drive_name", "Documents")`

cache — If TRUE, cache drive to a file using cache_sp_drive().

refresh — If TRUE, get a new drive even if the existing drive is cached as a local option. If FALSE, use the cached ms_drive object if it exists.

overwrite — If TRUE, replace the existing cached object named by cache_file with the new object. If FALSE, error if a cached file with the same cache_file name already exists.

cache_file — File name for cached drive if cache = TRUE. Defaults to a option set with sharepointr.cache_file_driv (which defaults to "sp_drive.rds").

call — The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

drive — A ms_drive object. If drive is supplied, drive_name and drive_id are ignored.

See Also

Microsoft365R::ms_drive

---

sp_list — *Get a SharePoint list or a list of SharePoint lists*

Description

get_sp_list() is a wrapper for the get_list and list_items methods. This function is still under development and does not support the URL parsing used by get_sp_item(). list_sp_lists() returns all lists for a SharePoint site or drive as a list or data frame. Note, when using filter with get_sp_list(), names used in the expression must be prefixed with "fields/" to distinguish them from item metadata.

Usage

```
get_sp_list(
  list_name = NULL,
  list_id = NULL,
  ...,
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  metadata = FALSE,
  as_data_frame = FALSE,
  call = caller_env()
)

list_sp_lists(
  site_url = NULL,
  filter = NULL,
  n = Inf,
  ...,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  as_data_frame = TRUE,
  call = caller_env()
)

get_sp_list_metadata(
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  ...,
  keep = c("all", "editable", "visible"),
  sync_fields = FALSE,
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  call = caller_env()
)

delete_sp_list(
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  confirm = TRUE,
  ...,
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  call = caller_env()
)
```

Arguments

`list_name`, `list_id`

SharePoint List name or ID string.

`...`

Arguments passed on to `get_sp_drive`

`drive_url` A SharePoint Drive URL to parse for a Drive name and other information. If `drive_name` is a URL, it is used as `drive_url`.

`default_drive_name` Drive name string used only if input is a document URL and drive name is not part of the URL. Defaults to `getOption("sharepointr.default_drive_name` `"Documents")`

`cache` If `TRUE`, cache drive to a file using `cache_sp_drive()`.

`refresh` If `TRUE`, get a new drive even if the existing drive is cached as a local option. If `FALSE`, use the cached `ms_drive` object if it exists.

`cache_file` File name for cached drive if `cache = TRUE`. Defaults to a option set with `sharepointr.cache_file_drive` (which defaults to `"sp_drive.rds"`).

`overwrite` If `TRUE`, replace the existing cached object named by `cache_file` with the new object. If `FALSE`, error if a cached file with the same `cache_file` name already exists.

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`site` A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.

`drive_name`, `drive_id`

SharePoint Drive name or ID passed to `get_drive` method for SharePoint site object.

`drive` A `ms_drive` object. If `drive` is supplied, `drive_name` and `drive_id` are ignored.

`metadata` If `TRUE`, `get_sp_list()` applies the `get_column_info` method to the returned SharePoint list and returns a data frame with column metadata for the list.

`as_data_frame` If `TRUE`, return a data frame with a "ms_list" column. `get_sp_list()` returns a 1 row data frame and `list_sp_lists()` returns a data frame with n rows or all lists available for the SharePoint site or drive. Defaults to `FALSE`. Ignored is `metadata = TRUE` as list metadata is always returned as a data frame.

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

`filter` A string with an OData expression apply as a filter to the results. Learn more in the Microsoft Graph API documentation on using filter query parameters.

`n` Maximum number of lists, plans, tasks, or other items to return. Defaults to `NULL` which sets n to `Inf`.

`sp_list` A `ms_list` object. If supplied, `list_name`, `list_id`, `site_url`, `site`, `drive_name`, `drive_id`, `drive`, and any additional parameters passed to `...` are all ignored.

`keep` One of "all" (default), "editable", "visible" (not yet supported). Argument determines if the returned list metadata includes read only columns or hidden columns.

`sync_fields` If `TRUE`, use the `sync_fields` method to sync the fields of the local `ms_list` object with the fields of the SharePoint List source before retrieving list metadata.

`confirm` If `TRUE`, confirm deletion of list before proceeding.

Value

A data frame as_data_frame = TRUE or a ms_list object (or list of ms_list objects) if FALSE.

See Also

- Microsoft365R::ms_list
- Microsoft365R::ms_list_item

---

sp_list_item *Get, list, update, and delete SharePoint list items*

Description

list_sp_list_items() lists sp_list items. Additional functions should be completed for the get_item, create_item, update_item, and delete_item methods documented in Microsoft365R::ms_list.

Usage

```
list_sp_list_items(
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  ...,
  filter = NULL,
  select = NULL,
  all_metadata = FALSE,
  as_data_frame = TRUE,
  display_nm = c("drop", "label", "replace"),
  name_repair = "unique",
  pagesize = 5000,
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  call = caller_env()
)

get_sp_list_item(
  id,
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  ...,
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  call = caller_env()
)
```

Arguments

list_name, list_id
: SharePoint List name or ID string.

sp_list
: A ms_list object. If supplied, list_name, list_id, site_url, site, drive_name, drive_id, drive, and any additional parameters passed to ... are all ignored.

...
: Arguments passed on to get_sp_list

  list_name,list_id SharePoint List name or ID string.

  metadata If TRUE, get_sp_list() applies the get_column_info method to the returned SharePoint list and returns a data frame with column metadata for the list.

  drive_name,drive_id SharePoint Drive name or ID passed to get_drive method for SharePoint site object.

filter
: A string with an OData expression apply as a filter to the results. Learn more in the Microsoft Graph API documentation on using filter query parameters.

select
: A character vector of column names to include in the returned data frame of list items. If NULL, the data frame includes all columns from the list.

all_metadata
: If TRUE, the returned data frame will contain extended metadata as separate columns, while the data fields will be in a nested data frame named fields. This is always set to FALSE if n = NULL or as_data_frame = FALSE.

as_data_frame
: If TRUE, return a data frame with a "ms_list" column. get_sp_list() returns a 1 row data frame and list_sp_lists() returns a data frame with n rows or all lists available for the SharePoint site or drive. Defaults to FALSE. Ignored is metadata = TRUE as list metadata is always returned as a data frame.

display_nm
: Option of "drop" (default), "label", or "replace". If "drop", display names are not accessed or used. If "label", display names are used to label matching columns in the returned data frame. If "replace", display names replace column names in the returned data frame. When working with the last option, the name_repair argument is required since there is no requirement on SharePoint for lists to use unique display names and invalid data frames can result.

name_repair
: Passed to repair argument of vctrs::vec_as_names()

pagesize
: Number of list items to return. Reduce from default of 5000 is experiencing timeouts.

site_url
: A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

site
: A ms_site object. If site is supplied, site_url, site_name, and site_id are ignored.

`drive_name, drive_id`
: SharePoint Drive name or ID passed to `get_drive` method for SharePoint site object.

`drive`
: A `ms_drive` object. If `drive` is supplied, `drive_name` and `drive_id` are ignored.

`call`
: The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

`id`
: Required. A SharePoint list item ID typically an integer for the record number starting from 1 with the first record.

`sp_plan` *Get a SharePoint plan or list SharePoint plans*

Description

`get_sp_plan()` returns a single `ms_plan` object using the `get_plan` method. `list_sp_plans()` returns a data frame with plan properties or a list of `ms_plan` objects.

Usage

```
get_sp_plan(
  plan_title = NULL,
  plan_id = NULL,
  ...,
  site = NULL,
  site_url = NULL,
  as_data_frame = FALSE,
  call = caller_env()
)

list_sp_plans(
  ...,
  filter = NULL,
  n = NULL,
  site = NULL,
  as_data_frame = TRUE,
  call = caller_env()
)
```

Arguments

`plan_title, plan_id`
: Planner title or ID. Exactly one of the two arguments must be supplied.

`...`
: Additional arguments passed to `get_sp_group()`.

`site` A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`as_data_frame` If `TRUE` (default for `list_sp_plans()`), return a data frame with the plan title, id, creation date/time, and owner ID with a list column named "ms_plan" containing `ms_plan` objects. If `FALSE` (default `get_sp_plan()`), return a `ms_plan` object or list of `ms_plan` objects.

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

`filter` A string with an OData expression apply as a filter to the results. Learn more in the Microsoft Graph API documentation on using filter query parameters.

`n` Maximum number of lists, plans, tasks, or other items to return. Defaults to `NULL` which sets n to `Inf`.

Value

For `get_sp_plan()`, a `ms_plan` class object or a 1 row data frame with a "ms_plan" column.

For `list_sp_plans()`, A list of `ms_plan` class objects or a data frame with a list column named "ms_plan".

See Also

Microsoft365R::ms_plan

`sp_plan_bucket` *List SharePoint planner buckets*

Description

`list_sp_plan_buckets()` lists the buckets for a specified plan using the `list_buckets` method.

Usage

```
list_sp_plan_buckets(
  plan_title = NULL,
  plan_id = NULL,
  ...,
  filter = NULL,
  n = NULL,
  plan = NULL,
  as_data_frame = TRUE,
  call = caller_env()
)
```

Arguments

`plan_title, plan_id`
: Planner title or ID. Exactly one of the two arguments must be supplied.

`...`
: Additional arguments passed to `get_sp_group()`.

`filter`
: A string with an OData expression apply as a filter to the results. Learn more in the Microsoft Graph API documentation on using filter query parameters.

`n`
: Maximum number of lists, plans, tasks, or other items to return. Defaults to `NULL` which sets n to `Inf`.

`plan`
: A `ms_plan` object. If `plan` is supplied, `plan_title`, `plan_id`, and any additional parameters passed to `...` are ignored.

`as_data_frame`
: If `TRUE` (default for `list_sp_plan_buckets()`), return a data frame of object properties with with a list column named "ms_plan_bucket" containing `ms_plan_task` objects. If `FALSE`, return a `ms_plan_bucket` object or list of `ms_plan_bucket` objects.

`call`
: The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

Value

For `list_sp_plan_buckets()`, a list of `ms_plan_bucket` class objects or a data frame with a list column named "ms_plan_task".

See Also

Microsoft365R::ms_plan_task

`sp_site` *Get a SharePoint site or set a default SharePoint site*

Description

`get_sp_site()` is a wrapper for `Microsoft365R::get_sharepoint_site()` and returns a `ms_site` object. `cache_sp_site()` allows you to cache a default SharePoint site for use by other functions.

Usage

```
get_sp_site(
  site_url = NULL,
  site_name = NULL,
  site_id = NULL,
  ...,
  cache = FALSE,
  refresh = TRUE,
  overwrite = FALSE,
  cache_file = getOption("sharepointr.cache_file_site", "sp_site.rds"),
  call = caller_env()
)

cache_sp_site(
  ...,
  site = NULL,
  cache_file = getOption("sharepointr.cache_file_site", "sp_site.rds"),
  cache_dir = "sharepointr.cache_dir",
  overwrite = FALSE,
  call = caller_env()
)
```

Arguments

`site_url` A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`site_name, site_id` Site name or ID of the SharePoint site as an alternative to the SharePoint site URL. Exactly one of `site_url`, `site_name`, and `site_id` must be supplied.

`...` Arguments passed on to `Microsoft365R::get_sharepoint_site`

- `app` A custom app registration ID to use for authentication. See below.
- `scopes` The Microsoft Graph scopes (permissions) to obtain. It should never be necessary to change these.
- `token` An AAD OAuth token object, of class `AzureAuth::AzureToken`. If supplied, the `tenant`, `app`, `scopes` and `...` arguments will be ignored. See "Authenticating with a token" below.
- `tenant` For `get_business_onedrive`, `get_sharepoint_site` and `get_team`, the name of your Azure Active Directory (AAD) tenant. If not supplied, use the value of the `CLIMICROSOFT365_TENANT` environment variable, or "common" if that is unset.

`cache` If `TRUE`, cache site to a file using `cache_sp_site()`.

`refresh` If `TRUE`, get a new site even if the existing site is cached as a local option. If `FALSE`, use the cached `ms_site` object.

`overwrite` If `TRUE`, replace the existing cached object named by `cache_file` with the new object. If `FALSE`, error if a cached file with the same `cache_file` name already exists.

`cache_file` File name for cached file if `cache = TRUE`. Defaults to `"sp_site.rds"` or option set with `sharepointr.cache_file_site`.

`call` The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

`site` A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.

`cache_dir` Cache directory. By default, uses an environmental variable named "sharepointr.cache_dir". If "sharepointr.cache_dir" is not set, the cache directory is set to `rappdirs::user_cache_dir("sharepointr")`.

See Also

Microsoft365R::ms_site

sp_tasks *Get a SharePoint task or list tasks from a planner*

Description

`get_sp_task()` gets an individual planner task using the `get_task` method. `list_sp_tasks()` lists the tasks for a specified plan using the `list_tasks` method.

Usage

```
get_sp_task(
  task_title = NULL,
  task_id = NULL,
  ...,
  plan_title = NULL,
  plan_id = NULL,
  plan = NULL,
  as_data_frame = FALSE,
  call = caller_env()
)

list_sp_tasks(
  plan_title = NULL,
  plan_id = NULL,
  ...,
  filter = NULL,
  n = NULL,
  plan = NULL,
  as_data_frame = TRUE,
  call = caller_env()
)
```

Arguments

`task_title, task_id`
Planner task title and id. Exactly one of `task_title` and `task_id` must be supplied.

`...` Additional arguments passed to `get_sp_group()`.

plan_title, plan_id
Planner title or ID. Exactly one of the two arguments must be supplied.

plan
A ms_plan object. If plan is supplied, plan_title, plan_id, and any additional parameters passed to ... are ignored.

as_data_frame
If TRUE (default for list_sp_tasks()), return a data frame of object properties with with a list column named "ms_plan_task" containing ms_plan_task objects. If FALSE (default get_sp_task()), return a ms_plan_task object or list of ms_plan_task objects.

call
The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

filter
A string with an OData expression apply as a filter to the results. Learn more in the Microsoft Graph API documentation on using filter query parameters.

n
Maximum number of lists, plans, tasks, or other items to return. Defaults to NULL which sets n to Inf.

Value

For list_sp_tasks(), a list of ms_plan_task class objects or a data frame with a list column named "ms_plan_task".

See Also

Microsoft365R::ms_plan_task

upload_sp_item *Upload a file or folder to a SharePoint Drive*

Description

upload_sp_item() wraps the upload_folder and upload_file method for ms_drive objects.

Usage

```
upload_sp_item(
  file = NULL,
  dest,
  ...,
  src = NULL,
  overwrite = FALSE,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  blocksize = 327680000,
  recursive = FALSE,
  parallel = FALSE,
  call = caller_env()
)

upload_sp_items(file = NULL, dest, ..., src = NULL, call = caller_env())
```

Arguments

`file`	Path for file or directory to upload. Optional if `src` is supplied.
`dest`	Destination on SharePoint for file to upload. SharePoint folder URLs are supported.
`...`	Additional parameters passed to `get_sp_site()` or `Microsoft365R::get_sharepoint_site()`.
`src`	Data source path passed to `upload_folder` or `upload_file` method. Defaults to `NULL` and set to use file value by default.
`overwrite`	If `FALSE` (default), error if an item with the name specified in `file` or `src` already exists at the specified destination. If `TRUE`, overwrite any existing items with the same name. The latter is the default for the `upload_file` method.
`drive_name`, `drive_id`	SharePoint drive name or ID.
`drive`	A `ms_drive` object. If drive is supplied, `drive_name`, `site_url`, and any additional parameters passed to `...` are ignored.
`blocksize`, `recursive`, `parallel`	Additional parameters passed to `upload_folder` or `upload_file` method for `ms_drive` objects.
`call`	The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

`write_sharepoint` *Write an object to file and upload file to SharePoint*

Description

`write_sharepoint()` uses `upload_sp_item()` to upload an object created using:

- `sf::write_sf()` if x is an `sf` object
- `readr::write_csv()` if x is a `data.frame` and file does not include a xlsx extension
- `openxlsx2::write_xlsx()` if x is a `data.frame` file does include a xlsx extension
- The `print()` method from `{officer}` if x is a `rdocx`, `rpptx`, or `rxlsx` object
- `readr::write_rds()` if x is any other class

Usage

```
write_sharepoint(
  x,
  file,
  dest,
  ...,
  .f = NULL,
  new_path = tempdir(),
  overwrite = FALSE,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  site_url = NULL,
  site_name = NULL,
  site_id = NULL,
  site = NULL,
  blocksize = 327680000,
  call = caller_env()
)
```

Arguments

`x`
: Object to write to file and upload to SharePoint. `sf`, `rdocx`, `rpptx`, `rxlsx`, and `data.frame` objects are saved with the functions noted in the description. All other object types are saved as `rds` outputs.

`file`
: File to write to. Passed to `file` parameter for `readr::write_csv()` or `readr::write_rds()`, `dsn` for `sf::write_sf()`, or `target` for `print()` (when working with `{officer}` class objects).

`dest`
: Destination on SharePoint for file to upload. SharePoint folder URLs are supported.

`...`
: Additional parameters passed to write function.

`.f`
: Optional function to write the input data to disk before uploading file to SharePoint.

`new_path`
: Path to write file to. Defaults to `tempdir()`

`overwrite`
: If `FALSE` (default), error if an item with the name specified in `file` or `src` already exists at the specified destination. If `TRUE`, overwrite any existing items with the same name. The latter is the default for the `upload_file` method.

`drive_name, drive_id`
: SharePoint drive name or ID.

`drive`
: A `ms_drive` object. If drive is supplied, `drive_name`, `site_url`, and any additional parameters passed to `...` are ignored.

`site_url`
: A SharePoint site URL in the format "https://[tenant name].sharepoint.com/sites/[site name]". Any SharePoint item or document URL can also be parsed to build a site URL using the tenant and site name included in the URL.

`site_name`, `site_id`

Site name or ID of the SharePoint site as an alternative to the SharePoint site URL. Exactly one of `site_url`, `site_name`, and `site_id` must be supplied.

`site`

A `ms_site` object. If `site` is supplied, `site_url`, `site_name`, and `site_id` are ignored.

`blocksize`

Additional parameter passed to `upload_folder` or `upload_file` method for `ms_drive` objects.

`call`

The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

Value

Invisibly returns the input object x.

Index