

Package: sharepointr (via r-universe)

October 16, 2026

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>

Depends R (>= 4.1.0)

Imports AzureGraph, cli, glue, httr, httr2 (>= 1.1.0), jsonlite, lifecycle, Microsoft365R, purrr (>= 1.2.0), rlang, stringr, utils, vctrs

Suggests carrier (>= 0.3.0), fs, knitr, mirai (>= 2.5.1), officer, openxlsx2, rappdirs, readr, readxl, rmarkdown, sf, testthat (>= 3.0.0), withr, yaml12

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.1.9000

Config/pak/sysreqs libicu-dev libssl-dev

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

Date/Publication 2026-10-16 14:59:10 UTC

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

RemoteRef HEAD

RemoteSha c79809f565bded946745ea883a84d331c27de106

Contents

as_column_definition	3
AzureR_config	4
column_validation	4
compare_sp_list	5
copy_column_definition_list	9
create_column_definition	10
create_column_definition_list	16
create_sp_drive	17
create_sp_list	19
create_sp_list_column	22
create_sp_list_items	25
create_sp_list_lookup_column	29
data_as_column_definition_list	34
delete_sp_item	36
delete_sp_list_item	37
download_sp_item	39
download_sp_list	42
get_sp_group	43
get_sp_item	44
get_sp_list_column	47
list_sp_drives	48
list_sp_item_versions	49
list_sp_list_views	50
list_sp_pages	54
list_sp_site_user_info	56
read_sharepoint	57
sp_dir_create	58
sp_dir_info	60
sp_drive	63
sp_item_copy	64
sp_item_properties	68
sp_list	70
sp_list_definition	73
sp_list_definition_table	79
sp_list_item	80
sp_plan	83
sp_plan_bucket	84
sp_site	86
sp_tasks	87
update_sp_drive	89
upload_sp_item	91
write_sharepoint	92

as_column_definition	Validate a column definition
----------------------	------------------------------

Description

`as_column_definition()` validates a named list of Microsoft Graph columnDefinition properties, such as a column read from a YAML file with `read_sp_list_yaml()` or returned by `create_column_definition()`.

A column definition must have a name and exactly one column type key (e.g. `text`, `choice`, or `dateTime`) holding the properties for that type. Property names must match the Graph API documentation. Unknown properties and read-only properties (e.g. `type`) are errors. A column `id` is allowed as a reference to an existing column but isn't used to match, create, or update columns. A custom element is allowed and isn't validated, so it can hold metadata used by other applications.

Formulas for calculated columns, default values, and validation have a leading `"=`" added if missing. Unlike `create_calculated_column()`, the formula isn't processed with `glue::glue()`.

Usage

```
as_column_definition(x, ..., allow_custom = TRUE, call = caller_env())
```

Arguments

<code>x</code>	A named list of columnDefinition properties.
<code>...</code>	Must be empty.
<code>allow_custom</code>	If TRUE (default), allow a custom element.
<code>call</code>	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

The validated column definition `x` with choices converted to a character vector and formulas normalized.

See Also

- [columnDefinition resource type](#)
- `read_sp_list_yaml()` for the YAML format that uses these definitions.

Examples

```
as_column_definition(  
  list(  
    name = "Notes",  
    displayName = "Project Notes",  
    text = list(allowMultipleLines = TRUE)  
  )  
)
```

 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

<code>path</code>	Path to configuration directory for AzureR package where the JSON file for graph_logins is stored. If NULL, path is set with the base folder for <code>rappdirs::user_cache_dir("AzureR")</code> (for Windows) or <code>rappdirs::user_config_dir("AzureR")</code> (for non-Windows).
<code>glob</code>	A wildcard aka globbing pattern (e.g. <code>*.csv</code>) passed on to <code>grep()</code> to filter paths.
<code>filename</code>	Filename to delete from configuration directory. Defaults to "graph_logins.json". Set to NULL if path contains a file name.

Value

For `AzureR_config_ls()`, a character vector of file paths matching `glob` in the AzureR configuration directory. For `AzureR_config_delete()`, invisibly returns the path to the deleted file.

 column_validation

Create a column validation definition

Description

`column_validation()` creates a `columnValidation` definition for the validation property of a column definition. A validation formula must evaluate to TRUE for a value to be saved.

Usage

```
column_validation(formula, descriptions = NULL, default_language = "en-US")
```

Arguments

formula	Required. Validation formula. Reference columns with the display name enclosed in square brackets, e.g. "=LEN([Title])<10". A leading "=" is added if missing.
descriptions	Optional message shown when validation fails. Either a string (used for default_language) or a named character vector with language tags as names, e.g. c("en-US" = "Too long"). Graph property: descriptions.
default_language	Default BCP 47 language tag for descriptions. Graph property: defaultLanguage.

Details

The Graph API doesn't support creating or updating a list column with a validation property, so `create_sp_list_column()` and `sync_sp_list()` apply validation with the SharePoint REST API instead. Validation isn't returned when reading list columns with the Graph API, so it can't be compared to an existing column.

Value

A named list formatted as a columnValidation resource.

See Also

[columnValidation resource type](#)

Examples

```
column_validation("=LEN([Project Code])=8", "Use an 8 character code.")
```

compare_sp_list	<i>Compare or sync a list definition with an existing SharePoint list</i>
-----------------	---

Description

[Experimental]

`compare_sp_list()` compares a list definition (e.g. from a YAML file read with `read_sp_list_yaml()`) with an existing SharePoint list and returns the list settings, columns, and views to add, update, or delete, and the folders to add to a document library.

`sync_sp_list()` applies those changes. By default, it only prints the planned changes (`dry_run = TRUE`) and doesn't delete columns or views (`delete = FALSE`).

Usage

```

compare_sp_list(
  definitions,
  sp_list = NULL,
  ...,
  views = TRUE,
  folders = TRUE,
  call = caller_env()
)

sync_sp_list(
  definitions,
  sp_list = NULL,
  ...,
  dry_run = TRUE,
  delete = FALSE,
  allow_data_loss = FALSE,
  views = TRUE,
  folders = TRUE,
  call = caller_env()
)

```

Arguments

definitions	A list definition: a <code>sp_list_definition</code> object (see read_sp_list_yaml()), a path to a YAML file, or a list of column definitions created with create_column_definition() or create_column_definition_list() .
sp_list	A <code>ms_list</code> object. If <code>NULL</code> , the list is retrieved with get_sp_list() using the definition id (with the site from <code>parentReference.siteId</code> or from additional arguments passed to <code>...</code>) or, if the definition has no id, the definition <code>displayName</code> and any additional arguments passed to <code>...</code> . If the definition has an id, it must match the id of <code>sp_list</code> . For compare_sp_list() , <code>sp_list</code> can also be a list of column metadata from get_sp_list_metadata(as_data_frame = FALSE) (only columns are compared).
...	Additional parameters passed to get_sp_list() if <code>sp_list</code> is <code>NULL</code> .
views	If <code>TRUE</code> (default), compare views if the definition has a <code>views</code> element. If <code>FALSE</code> , views are skipped.
folders	If <code>TRUE</code> (default), compare folders if the definition has a <code>folders</code> element. If <code>FALSE</code> , folders are skipped.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of abort() for more information.
dry_run	If <code>TRUE</code> (default), print the planned changes without changing the list.
delete	If <code>TRUE</code> , delete columns and views that aren't in the definitions. Defaults to <code>FALSE</code> .
allow_data_loss	If <code>TRUE</code> , apply changes that may cause data loss (see details). Defaults to <code>FALSE</code> .

Details

Comparing columns

Columns are matched by internal name (name). Only the properties included in a definition are compared, so a definition with `text: {}` matches any text column. A property missing from a definition isn't changed.

SharePoint internal columns and the Title column are excluded unless Title is included in the definitions. The built-in columns of a document library (e.g. FileLeafRef or _ExtendedDescription) are also excluded unless they're included in the definitions. Columns in the list that aren't in the definitions are returned with `action = "delete"`.

Each change has one of these actions:

- "add": the column or view isn't in the list.
- "update": a property is different.
- "delete": the column or view isn't in the definitions.
- "blocked": the change can't be made. This includes changing the column type (e.g. text to number), changing a lookup column's source list, adding a column type the Graph API can't create, adding a list column with a name longer than 32 characters (which SharePoint would cut), changing the list template, and removing the default view without setting another default view.
- "unverified": the current value can't be read with the Graph API. This includes validation (which `sync_sp_list()` applies every time) and columns where the Graph API doesn't return the column type (hyperlink, picture, thumbnail, and term columns).

A column's internal name can't be changed. A renamed column is returned as a column to add and a column to delete.

If `sp_list` is a `ms_list` object and `definitions` is a list definition, the list `displayName`, `description`, and list settings (`hidden` and `contentTypeEnabled`) are also compared. A different template is blocked since the template can't be changed after a list is created.

Comparing views

Views are only compared if the definition has a `views` element and `views = TRUE`. A definition without `views` never adds, changes, or deletes views.

Views are matched by `Id` (if included in the definition) or `Title`, so a view with an `Id` can be renamed. Only the properties included in a view definition are compared. Hidden and personal views are excluded, and views in the list that aren't in the definition are returned with `action = "delete"`.

Setting `DefaultView: true` makes a view the default view (and the current default view is no longer the default). The current default view can't be removed from the default or deleted unless another view is set as the default view. Changes are applied in this order: new views, updated views, the default view, and then deleted views.

Views are read and changed with the SharePoint REST API. Use `views = FALSE` to skip views (e.g. if the SharePoint REST API isn't available).

Comparing folders

Folders are only compared if the definition has a `folders` element, `folders = TRUE`, and the list is a document library. Each folder path that doesn't exist is returned with `action = "add"`. A path

that exists as a file is blocked. Folders that aren't in the definition are never deleted. Each folder path is checked with one request.

Changes that use the SharePoint REST API

SharePoint stores single and multiple lines of text and single and multiple choice columns as different field types, and the Graph API can't change them. These changes use the SharePoint REST API (method = "rest"):

- `text.allowMultipleLines`: switching to multiple lines keeps existing values. Switching to a single line truncates values longer than 255 characters, so it is skipped unless `allow_data_loss = TRUE`.
- `choice.displayAs` to or from "checkboxes": switching to multiple choice keeps existing values. Switching to a single choice may drop values from items with more than one choice, so it is skipped unless `allow_data_loss = TRUE`.
- `validation`: the Graph API can't create or update column validation.
- All view changes.

The SharePoint REST API requires a delegated (user) login with a refresh token, such as the default Microsoft365R login.

Value

`compare_sp_list()` returns a data frame with one row per added or deleted column or view and one row per changed property, with columns:

- `object`: "list", "column", "view", or "folder".
- `name`: list display name, internal column name, view title, or folder path.
- `id`: column or view ID for existing columns and views.
- `action`: one of "add", "update", "delete", "blocked", or "unverified".
- `property`: changed property, e.g. "displayName", "text.allowMultipleLines", "list.hidden", or "RowLimit".
- `current`, `proposed`: list columns with the current and proposed values.
- `method`: "graph" or "rest" for changes that can be applied.
- `data_loss`: TRUE if the change may cause data loss.
- `note`: explanation for blocked, unverified, or data loss changes.

`sync_sp_list()` invisibly returns the same data frame with a status column: "planned" (for a dry run), "applied", "skipped", "blocked", or "failed".

Examples

```
## Not run:
definition <- read_sp_list_yaml("list-fields/capital-project.yaml")

compare_sp_list(definition, site_url = "<SharePoint site url>")

# Print planned changes
sync_sp_list(definition, site_url = "<SharePoint site url>")
```

```
# Apply changes
sync_sp_list(
  definition,
  site_url = "<SharePoint site url>",
  dry_run = FALSE
)

## End(Not run)
```

copy_column_definition_list

Create a new column definition based on an existing list

Description

[Experimental]

Usage

```
copy_column_definition_list(sp_list = NULL, ...)
```

Arguments

sp_list	A ms_list object or a data frame created by get_sp_list_metadata() . Optional if additional parameters are provided to ... that can be used to get a SharePoint list to copy column definitions from.
...	Arguments passed on to get_sp_list_metadata
	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.
	keep One of "all" (default), "editable", "external" (non-internal fields). 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.
	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.
- `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.
- `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.

Details

[copy_column_definition_list\(\)](#) takes an existing SharePoint list and uses the list metadata to create a column definition list that can be used to create a new SharePoint list. Note: lookup columns retain the original lookup list references so self-referencing lookup columns are copied as lookup columns referencing the source list.

Value

A list of named lists, each formatted as a `columnDefinition` (as created by [create_column_definition\(\)](#)) for use as the `fields` argument to [create_sp_list\(\)](#).

`create_column_definition`

Create a column definition for use with the create column method for SharePoint lists

Description

[create_column_definition\(\)](#) builds a named list with the properties of the `columnDefinition` resource type. The `create_*_column()` helpers create a definition for a single column type.

Arguments that set a `columnDefinition` property note the matching Graph property name. Graph property names can also be passed to ... (e.g. `create_text_column("Notes", allowMultipleLines = TRUE)` or `create_column_definition("Notes", displayName = "Project Notes")`). Column-level properties are added to the definition and any other values are added to the properties for the column type. Supplying the same property with both an argument and a Graph name is an error. Definitions are checked with the same rules as [as_column_definition\(\)](#).

Properties left as NULL aren't included in the definition so SharePoint uses the default value when a column is created.

More information: <https://learn.microsoft.com/en-us/graph/api/resources/columndefinition?view=graph-rest-1.0>

Usage

```
create_column_definition(  
  name,  
  ...,  
  .col_type = "text",  
  enforce_unique = NULL,  
  hidden = NULL,  
  deletable = NULL,  
  indexed = NULL,  
  sealed = NULL,  
  propagate_changes = NULL,  
  read_only = NULL,  
  required = NULL,  
  validation = NULL,  
  default = get_column_default(),  
  description = NULL,  
  display_name = NULL,  
  id = NULL,  
  displayname = deprecated(),  
  call = current_env()  
)
```

```
create_text_column(  
  name,  
  ...,  
  multiple_lines = NULL,  
  append_changes = NULL,  
  lines = NULL,  
  max_length = NULL,  
  text_type = NULL  
)
```

```
create_choice_column(  
  name,  
  choices,  
  ...,  
  allow_text = NULL,  
  display_as = NULL,  
  allow_na = TRUE,  
  na_replacement = "NA",  
  split = NULL  
)
```

```
create_number_column(  
  name,  
  ...,  
  decimal_places = NULL,  
  display_as = NULL,  
  ...  
)
```

```
    max = NULL,  
    min = NULL,  
    decimals = deprecated()  
)  
  
create_datetime_column(name, ..., display_as = NULL, format = NULL)  
  
create_boolean_column(name, ...)  
  
create_currency_column(name, ..., locale = NULL)  
  
create_calculated_column(name, ..., formula, format = NULL, output_type = NULL)  
  
create_lookup_column(  
    name,  
    lookup_list_column = NULL,  
    ...,  
    lookup_list_id = NULL,  
    lookup_list = NULL,  
    allow_multiple_values = NULL,  
    allow_unlimited_length = NULL,  
    primary_lookup_column_id = NULL,  
    allow_multiple = deprecated()  
)  
  
create_person_column(  
    name,  
    ...,  
    allow_multiple_selection = NULL,  
    display_as = NULL,  
    from_type = "peopleOnly",  
    allow_multiple = deprecated()  
)  
  
create_group_column(  
    name,  
    ...,  
    allow_multiple_selection = NULL,  
    display_as = NULL,  
    from_type = "peopleAndGroups",  
    allow_multiple = deprecated()  
)  
  
create_hyperlink_column(name, ..., is_picture = FALSE)  
  
create_picture_column(name, ..., is_picture = TRUE)  
  
create_thumbnail_column(name, ...)
```

```

create_geolocation_column(name, ...)

create_term_column(
    name,
    ...,
    allow_multiple_values = NULL,
    show_full_name = NULL,
    allow_multiple = deprecated()
)

```

Arguments

name	Column name. Graph property: name. The name can't be changed after a column is created.
...	Additional arguments passed to create_column_definition() or columnDefinition properties using Graph property names.
.col_type	Column type. Defaults to "text". Must be one of "boolean", "calculated", "choice", "currency", "dateTime", "lookup", "number", "personOrGroup", "text", "term", "hyperlinkOrPicture", "thumbnail", "contentApprovalStatus", or "geolocation".
enforce_unique	Enforce unique values in column. Graph property: enforceUniqueValues.
hidden	If TRUE, column will be hidden by default. Graph property: hidden.
deletable	If TRUE, column can't be deleted separate from the list. Graph property: isDeletable.
indexed, sealed, propagate_changes, read_only, id	Additional arguments used by create_column_definition() . Graph properties: indexed, isSealed, propagateChanges, readOnly, and id.
required	If TRUE, column will be required. Graph property: required.
validation	Column validation created with column_validation() . Graph property: validation.
default	Default value set by helper get_column_default() function. Graph property: defaultValue.
description	Column description. Graph property: description.
display_name	Column display name. Graph property: displayName.
displayname	[Deprecated] Use display_name.
call	The execution environment used in error messages. The create*_column() helpers pass their own environment.
multiple_lines	Logical. If TRUE, allow multiple lines of text. Graph property: allowMultipleLines.
append_changes	Logical. If TRUE, append changes to existing value for column. Graph property: appendChangesToExistingText.
lines	Whole number. Size of the text box. Graph property: linesForEditing.
max_length	Whole number. Max length in number of characters. Graph property: maxLength.
text_type	One of c("plain", "richText"). Graph property: textType.
choices	A character vector of choice options. Graph property: choices.

allow_text	If TRUE, allow text entry in the choice column. Graph property: allowTextEntry.
display_as	Value displayed as option. For create_choice_column one of c("checkboxes", "dropDownMenu", "radioButtons"). For create_number_column, one of c("number", "percentage"). For create_datetime_column, one of c("default", "friendly", "standard"). Graph property: displayAs.
allow_na	If TRUE, allow NA values in choices.
na_replacement	Used as replacement by <code>stringr::str_replace_na()</code> on choices if they contain NA values.
split	character vector (or object which can be coerced to such) containing regular expression (s) (unless fixed = TRUE) to use for splitting. If empty matches occur, in particular if split has length 0, x is split into single characters. If split has length greater than 1, it is re-cycled along x.
decimal_places	One of c("automatic", "none", "one", "two", "three", "four", "five") or a whole number between 0 and 5. Graph property: decimalPlaces.
max, min	Minimum and maximum values allowed in number column. Graph properties: maximum and minimum.
decimals	[Deprecated] Use decimal_places.
format	For create_datetime_column(), "dateOnly" or "dateTime". Graph property: format.
locale	Locale used to set the currency symbol, e.g. "en-us". Graph property: locale.
formula	Required string with formula for calculated column definition. See examples of common formulas in lists . Reference existing columns using the display name enclosed in square brackets. The formula must start with an equals sign "=" which this function appends to the formula text if it is missing. The formula is processed with <code>glue::glue()</code> . Graph property: formula.
output_type	Value type returned by calculated formula. One of c("text", "boolean", "currency", "dateTime", "number"). Defaults to "text". Graph property: outputType.
lookup_list_column	Name of lookup column in the lookup list to use. Graph property: columnName.
lookup_list_id, lookup_list	Lookup list ID string or "ms_list" class object with id value in list properties. Graph property: listId.
allow_multiple_values	If TRUE, allow a lookup or term column to store multiple values. Graph property: allowMultipleValues.
allow_unlimited_length	If TRUE, allow lookup column to return any length value. Graph property: allowUnlimitedLength.
primary_lookup_column_id	If column definition is for a secondary column, the primary lookup column ID must be supplied. Graph property: primaryLookupColumnId.
allow_multiple	[Deprecated] Use allow_multiple_values for <code>create_lookup_column()</code> and <code>create_term_column()</code> or allow_multiple_selection for <code>create_person_column()</code> and <code>create_group_column()</code> .

allow_multiple_selection	If TRUE, allow a person or group column to store multiple values. Graph property: allowMultipleSelection.
from_type	What type of resources to choose from. Defaults to "peopleOnly" for <code>create_person_column()</code> or "peopleAndGroups" for <code>create_group_column()</code> . Graph property: chooseFromType.
is_picture	Logical indicator for display of hyperlink value as link (FALSE, default for <code>create_hyperlink_column()</code>) or image (TRUE, default for <code>create_picture_column()</code>). Graph property: isPicture.
show_full_name	If TRUE, display the entire term path. Graph property: showFullyQualifiedName.

Details

Display as options

Display as options vary by columnDefinition type. See documentation for more details:

- personOrGroupColumn: <https://learn.microsoft.com/en-us/graph/api/resources/personorgroupcolumn?view=graph-rest-1.0#displayas-options>
- choiceColumn: <https://learn.microsoft.com/en-us/graph/api/resources/choicecolumn?view=graph-rest-1.0#properties>
- numberColumn: <https://learn.microsoft.com/en-us/graph/api/resources/numbercolumn?view=graph-rest-1.0#properties>
- dateTimeColumn: <https://learn.microsoft.com/en-us/graph/api/resources/datetimetable?view=graph-rest-1.0>

Column types the Graph API can't create

As of October 2026, the Graph API returns an "Invalid request" error when creating a list column with the hyperlinkOrPicture, thumbnail, geolocation, or term column types. `create_hyperlink_column()`, `create_picture_column()`, `create_thumbnail_column()`, `create_geolocation_column()`, and `create_term_column()` still create valid definitions, but `create_sp_list_column()` can't use them. The Graph API also doesn't return the column type for hyperlinkOrPicture, thumbnail, or term columns when reading list columns.

Term columns also need a term set, which isn't supported.

Value

A named list of columnDefinition properties formatted for use as the columns argument to `create_sp_list()` or as an element of the list returned by `create_column_definition_list()`.

Examples

```
create_text_column("TextColumn")

create_text_column("NotesColumn", multiple_lines = TRUE)

# Graph property names are also supported
create_text_column("NotesColumn", allowMultipleLines = TRUE)

fruit <- c("apple", "banana", "pear", "pineapple")
create_choice_column("ChoiceColumn", fruit)
```

```

create_number_column("NumberColumn")

create_number_column("PercentColumn", display_as = "percentage", max = 1)

create_datetime_column("DatetimeColumn")

create_datetime_column("DateColumn", format = "dateOnly")

create_calculated_column(
  name = "FormulaColumn",
  formula = "[Text Column]"
)

```

```
create_column_definition_list
```

Create a list of column definitions

Description

`create_column_definition_list()` is a vectorized version of `create_column_definition()` that uses a list or data frame input to create a list of column definitions. This list can be used as the columns argument for `create_sp_list()`.

Usage

```
create_column_definition_list(definitions, col_type = "text", ignore_na = TRUE)
```

Arguments

definitions	A list or data frame with arguments to use in creation of column definitions.
col_type	Column type to use if not provided as a "type" column in the input definitions data frame. Allowed values include the Graph column type keys (e.g. "date-Time" or "personOrGroup") and the shorter names date, datetime, person, group, hyperlink, and picture. Not case sensitive. A "date" column uses format = "dateOnly" unless a format is supplied.
ignore_na	If TRUE, drop any parameters with a NA value.

Details

Columns in definitions can use the argument names of the `create_*_column()` helpers (e.g. `multiple_lines`) or Graph property names (e.g. `allowMultipleLines`). See `create_column_definition()` for details.

Value

A list of named lists, one per row of definitions, each formatted as a `columnDefinition` (as created by `create_column_definition()`) for use as the columns argument to `create_sp_list()`.

Examples

```
definition_df <- data.frame(
  name = c("FirstColumn", "SecondColumn"),
  type = c("text", "number"),
  decimal_places = c(NA, 0),
  multiple_lines = c(TRUE, NA)
)

create_column_definition_list(definition_df)
```

create_sp_drive	Create a SharePoint document library
-----------------	--------------------------------------

Description**[Experimental]**

[create_sp_drive\(\)](#) creates a document library on a SharePoint site and returns the library's drive as a `ms_drive` object. A document library is a SharePoint list created with the "documentLibrary" template, so this function wraps [create_sp_list\(\)](#). Columns, views, and folders can be supplied with a list definition (see [read_sp_list_yaml\(\)](#)).

The library's list can be used with the list functions (e.g. [compare_sp_list\(\)](#) or [sync_sp_list\(\)](#)) and the drive can be used with the file and folder functions (e.g. [sp_dir_create\(\)](#) or [upload_sp_item\(\)](#)).

[create_sp_drive\(\)](#) doesn't create a SharePoint Embedded container (a Graph `fileStorageContainer`). Containers aren't part of a SharePoint site and can only be created by an app that owns a registered container type.

Usage

```
create_sp_drive(
  drive_name = NULL,
  description = NULL,
  ...,
  folders = NULL,
  definition = NULL,
  properties = FALSE,
  site_url = NULL,
  site = NULL,
  call = caller_env()
)
```

Arguments

<code>drive_name</code>	Library name used as the <code>displayName</code> property. SharePoint sets the library URL from this name, so the drive name and URL can't be changed separately. Required unless <code>definition</code> is supplied.
-------------------------	---

description	Optional description.
...	Additional parameters passed to create_sp_list() (e.g. columns, content_types, or hidden) and to get_sp_site() if site is NULL.
folders	Optional. A character vector of folder paths to create, relative to the library root (e.g. <code>c("Reports/2026", "Data")</code>). Parent folders are created as needed. Replaces any folders in definition.
definition	Optional. A list definition from read_sp_list_yaml() or a path to a YAML file. The definition list.template must be documentLibrary if it is set. See create_sp_list() for how the definition is used.
properties	If TRUE, return the drive properties instead of the ms_drive object. Defaults to FALSE.
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.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

A ms_drive object for the new library, or the drive properties as a named list if properties = TRUE.

See Also

[update_sp_drive\(\)](#), [delete_sp_drive\(\)](#), [create_sp_list\(\)](#), [get_sp_drive\(\)](#)

Examples

```
## Not run:
create_sp_drive(
  "Project Files",
  folders = c("Contracts", "Reports/2026"),
  site_url = "<SharePoint site url>"
)

create_sp_drive(
  definition = "library-fields/project-files.yaml",
  site_url = "<SharePoint site url>"
)

## End(Not run)
```

create_sp_list

*Create, update, or delete a SharePoint List***Description**

`create_sp_list()` allows the creation of a SharePoint list for a site. See: <https://learn.microsoft.com/en-us/graph/api/list-create?view=graph-rest-1.0&tabs=http>

`update_sp_list()` allows the modification of the list display name and description.

`delete_sp_list()` deletes an existing list and requires user confirmation by default.

Notes on creating a SharePoint list:

- Dashes ("-"") in list names are removed from the list name but retained in the list display name.
- Column names longer than 32 characters (counting each space or special character as 7) are an error for a list (but not a document library), since SharePoint cuts them without an error. A display name longer than 255 characters (the limit on the column settings page) is a warning.
- Calculated columns are added after the list is created since formulas may reference other columns.
- Column validation is applied with the SharePoint REST API after the list is created since the Graph API doesn't support it.
- A "Title" column in columns or definition updates the default "Title" column of a "genericList" list (combined with title_definition) or a "documentLibrary" library instead of adding a new column.
- Views in definition are created with the SharePoint REST API after the columns are created. A view titled "All Items" updates the default view of a "genericList" list and a view titled "All Documents" updates the default view of a "documentLibrary" library.
- Folders in definition are created after the views for a "documentLibrary" library. Use `create_sp_drive()` to create a document library and return the `ms_drive` object.

Notes on updating a SharePoint list:

- The "Title" column type is always a "text" type column and can't be changed.

Usage

```
create_sp_list(
    list_name = NULL,
    ...,
    description = NULL,
    columns = NULL,
    template = NULL,
    content_types = NULL,
    hidden = NULL,
    title_definition = list(required = FALSE),
    definition = NULL,
```

```

    site_url = NULL,
    site = NULL,
    call = caller_env()
)

update_sp_list(
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  display_name = NULL,
  description = NULL,
  ...,
  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 name used as displayName property. Required unless definition is supplied.
...	Additional parameters passed to get_sp_site() or Microsoft365R::get_sharepoint_site() .
description	Optional description.
columns	Optional. Use create_column_definition() to create a single column definition or create_column_definition_list() to create a list of column definitions. custom metadata is dropped. A sp_list_definition object is used as definition.
template	Optional list template (Graph listInfo property template). Defaults to "genericList".
content_types	Optional. Set TRUE for contentTypesEnabled to be enabled.
hidden	Optional. Set TRUE for list to be hidden.

title_definition	Named list used to update the column definition of the default "Title" column created when using the "genericList" template. By default, makes Title column optional.
definition	Optional. A list definition from <code>read_sp_list_yaml()</code> or a path to a YAML file. The definition displayName, description, columns, and list settings (template, hidden, and contentTypesEnabled) are used unless the matching argument is supplied. Read-only list properties (e.g. id) and column ids are ignored. definition can't be combined with columns. Views in the definition are also created (see <code>list_sp_list_views()</code>), and so are folders for a document library.
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 <code>ms_site</code> 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. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.
list_id	List ID for list to update or delete.
sp_list	A <code>Microsoft365R::ms_list</code> object.
display_name	Display name to replace existing display name. Used by <code>update_sp_list()</code> .
drive_name, drive_id	SharePoint Drive name or ID passed to <code>get_drive</code> method for SharePoint site object.
drive	A <code>ms_drive</code> object. If drive is supplied, drive_name and drive_id are ignored.
confirm	If TRUE, confirm deletion of list before proceeding.

Value

For `create_sp_list()`, invisibly returns a `ms_list` object for the newly created list. For `update_sp_list()`, the updated `ms_list` object. For `delete_sp_list()`, invisibly returns NULL.

Examples

```
## Not run:
create_sp_list(
  definition = "list-fields/capital-project.yaml",
  site_url = "<SharePoint site url>"
)

## End(Not run)
```

create_sp_list_column *Create, update, and delete a SharePoint list column*

Description

[create_sp_list_column\(\)](#) adds a column to a SharePoint list and [delete_sp_list_column\(\)](#) removes a column to a SharePoint list. [update_sp_list_column\(\)](#) updates a column definition for an existing column in a SharePoint list. Only properties that differ from the existing column are sent. Use [sync_sp_list\(\)](#) for changes the Graph API can't make, such as switching a text column to multiple lines.

Usage

```
create_sp_list_column(  
    sp_list = NULL,  
    ...,  
    column_name = NULL,  
    column_definition = NULL,  
    list_name = NULL,  
    site_url = NULL,  
    site = NULL,  
    call = caller_env()  
)  
  
update_sp_list_column(  
    sp_list = NULL,  
    column_name = NULL,  
    column_id = NULL,  
    ...,  
    list_name = NULL,  
    site_url = NULL,  
    site = NULL,  
    column_definition = NULL,  
    column_name_type = "name",  
    call = caller_env()  
)  
  
delete_sp_list_column(  
    sp_list = NULL,  
    column_name = NULL,  
    column_id = NULL,  
    list_name = NULL,  
    site_url = NULL,  
    site = NULL,  
    column_name_type = "name",  
    call = caller_env()  
)
```

Arguments

sp_list	A <code>ms_list</code> object. If supplied, <code>list_name</code> , <code>list_id</code> , <code>site_url</code> , <code>site</code> , <code>drive_name</code> , <code>drive_id</code> , <code>drive</code> , and any additional parameters passed to ... are all ignored.
...	Arguments passed on to <code>create_column_definition</code>
name	Column name. Graph property: <code>name</code> . The name can't be changed after a column is created.
display_as	Value displayed as option. For <code>create_choice_column</code> one of <code>c("checkboxes", "dropDownMenu", "radioButtons")</code> . For <code>create_number_column</code> , one of <code>c("number", "percentage")</code> . For <code>create_datetime_column</code> , one of <code>c("default", "friendly", "standard")</code> . Graph property: <code>displayAs</code> .
.col_type	Column type. Defaults to "text". Must be one of "boolean", "calculated", "choice", "currency", "dateTime", "lookup", "number", "person-OrGroup", "text", "term", "hyperlinkOrPicture", "thumbnail", "contentApprovalStatus", or "geolocation".
enforce_unique	Enforce unique values in column. Graph property: <code>enforceUniqueValues</code> .
hidden	If TRUE, column will be hidden by default. Graph property: <code>hidden</code> .
deletable	If TRUE, column can't be deleted separate from the list. Graph property: <code>isDeletable</code> .
required	If TRUE, column will be required. Graph property: <code>required</code> .
default	Default value set by helper <code>get_column_default()</code> function. Graph property: <code>defaultValue</code> .
description	Column description. Graph property: <code>description</code> .
display_name	Column display name. Graph property: <code>displayName</code> .
displayname	[Deprecated] Use <code>display_name</code> .
validation	Column validation created with <code>column_validation()</code> . Graph property: <code>validation</code> .
indexed, sealed, propagate_changes, read_only, id	Additional arguments used by <code>create_column_definition()</code> . Graph properties: <code>indexed</code> , <code>isSealed</code> , <code>propagateChanges</code> , <code>readOnly</code> , and <code>id</code> .
multiple_lines	Logical. If TRUE, allow multiple lines of text. Graph property: <code>allowMultipleLines</code> .
append_changes	Logical. If TRUE, append changes to existing value for column. Graph property: <code>appendChangesToExistingText</code> .
lines	Whole number. Size of the text box. Graph property: <code>linesForEditing</code> .
max_length	Whole number. Max length in number of characters. Graph property: <code>maxLength</code> .
text_type	One of <code>c("plain", "richText")</code> . Graph property: <code>textType</code> .
choices	A character vector of choice options. Graph property: <code>choices</code> .
allow_na	If TRUE, allow NA values in choices.
na_replacement	Used as replacement by <code>stringr::str_replace_na()</code> on choices if they contain NA values.
allow_text	If TRUE, allow text entry in the choice column. Graph property: <code>allowTextEntry</code> .

decimal_places One of c("automatic", "none", "one", "two", "three", "four", "five") or a whole number between 0 and 5. Graph property: decimalPlaces.

max,min Minimum and maximum values allowed in number column. Graph properties: maximum and minimum.

decimals **[Deprecated]** Use decimal_places.

format For create_datetime_column(), "dateOnly" or "dateTime". Graph property: format.

locale Locale used to set the currency symbol, e.g. "en-us". Graph property: locale.

formula Required string with formula for calculated column definition. See [examples of common formulas in lists](#). Reference existing columns using the display name enclosed in square brackets. The formula must start with an equals sign "=" which this function appends to the formula text if it is missing. The formula is processed with [glue::glue\(\)](#). Graph property: formula.

output_type Value type returned by calculated formula. One of c("text", "boolean", "currency", "dateTime", "number"). Defaults to "text". Graph property: outputType.

lookup_list_column Name of lookup column in the lookup list to use. Graph property: columnName.

lookup_list_id,lookup_list Lookup list ID string or "ms_list" class object with id value in list properties. Graph property: listId.

allow_multiple_values If TRUE, allow a lookup or term column to store multiple values. Graph property: allowMultipleValues.

allow_unlimited_length If TRUE, allow lookup column to return any length value. Graph property: allowUnlimitedLength.

primary_lookup_column_id If column definition is for a secondary column, the primary lookup column ID must be supplied. Graph property: primaryLookupColumnId.

allow_multiple **[Deprecated]** Use allow_multiple_values for [create_lookup_column\(\)](#) and [create_term_column\(\)](#) or allow_multiple_selection for [create_person_column\(\)](#) and [create_group_column\(\)](#).

allow_multiple_selection If TRUE, allow a person or group column to store multiple values. Graph property: allowMultipleSelection.

from_type What type of resources to choose from. Defaults to "peopleOnly" for [create_person_column\(\)](#) or "peopleAndGroups" for [create_group_column\(\)](#). Graph property: chooseFromType.

is_picture Logical indicator for display of hyperlink value as link (FALSE, default for [create_hyperlink_column\(\)](#)) or image (TRUE, default for [create_picture_column\(\)](#)). Graph property: isPicture.

show_full_name If TRUE, display the entire term path. Graph property: showFullyQualifiedName.

split character vector (or object which can be coerced to such) containing [regular expression](#)(s) (unless fixed = TRUE) to use for splitting. If empty matches occur, in particular if split has length 0, x is split into single characters. If split has length greater than 1, it is re-cycled along x.

column_name, column_id	Column ID for column to delete.
column_definition	List with column definition created with <code>create_column_definition()</code> or a related function. Optional if column_name and any required additional parameters are provided. A custom element (e.g. from <code>read_sp_list_yaml()</code>) is dropped. A validation element is applied with the SharePoint REST API since the Graph API doesn't support it.
list_name	List name. Required if sp_list is NULL.
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.
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 <code>abort()</code> for more information.
column_name_type	"name" or "displayName". Used to match column ID so column_name must be unique if column_name_type = "displayName".

Details

See documentation: <https://learn.microsoft.com/en-us/graph/api/list-post-columns?view=graph-rest-1.0&tabs=http>

Value

For `create_sp_list_column()`, a named list with the newly created columnDefinition resource.
 For `update_sp_list_column()`, invisibly returns the input sp_list. For `delete_sp_list_column()`, an empty list with a "status" attribute giving the HTTP response status code.

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 = FALSE,
    .id = "id",
    site_url = NULL,
    site = NULL,
    drive_name = NULL,
    drive_id = NULL,
    drive = NULL,
    check_fields = TRUE,
    sync_fields = FALSE,
    create_list = FALSE,
    strict = FALSE,
    .batch = getOption("sharepointtr.batch", TRUE),
    .progress = TRUE,
    call = caller_env()
)

```

```

update_sp_list_items(
  data,
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  ...,
  .id = "id",
  allow_display_nm = FALSE,
  check_fields = TRUE,
  strict = FALSE,
  na_fields = c("drop", "replace"),
  drop_fields = c("ContentType", "Attachments"),
  .batch = getOption("sharepointtr.batch", TRUE),
  .progress = TRUE,
  call = caller_env()
)

```

```

update_sp_list_item(
  ...,
  .data = NULL,
  item_id = NULL,
  sp_list_item = NULL,
  .id = "id",
  check_fields = TRUE,
  na_fields = c("drop", "replace"),
  drop_fields = c("ContentType", "Attachments"),
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  site_url = NULL,
  site = NULL,

```

```

    drive_name = NULL,
    drive_id = NULL,
    drive = NULL,
    .multi_fields = NULL,
    call = caller_env()
  )

```

Arguments

data	Required. A data frame or a list of named lists (one record per item) to import as items to the supplied or identified SharePoint list. If data is an sf object, the geometry column is coerced to text using <code>sf::st_as_text()</code> . For <code>update_sp_list_items()</code> , each record must include an <code>.id</code> element and data can also be a single named list record for one item. Unlike a data frame, any field missing from a record is left unchanged, even when <code>na_fields = "replace"</code> . For <code>create_sp_list_items()</code> , wrap a single record in a list (e.g. <code>list(record)</code>) and data must be a data frame if <code>create_list = TRUE</code> .
list_name, list_id	SharePoint List name or ID string.
sp_list	A <code>ms_list</code> object. If supplied, <code>list_name</code> , <code>list_id</code> , <code>site_url</code> , <code>site</code> , <code>drive_name</code> , <code>drive_id</code> , <code>drive</code> , and any additional parameters passed to ... are all ignored.
...	Additional parameters passed to <code>get_sp_site()</code> or <code>Microsoft365R::get_sharepoint_site()</code> .
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. Default FALSE.
.id	Name of the column in data (or the element in each record if data is a list) with item ID values. Defaults to "id". Item IDs must be whole numbers or non-empty strings and are checked before any items are updated. For <code>create_sp_list_items()</code> , <code>.id</code> is only used to keep the ID column name from being replaced when <code>allow_display_nm = TRUE</code> . For <code>update_sp_list_item()</code> , <code>.id</code> is used to get <code>item_id</code> from <code>.data</code> if <code>item_id</code> isn't 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 <code>ms_site</code> object. If site is supplied, <code>site_url</code> , <code>site_name</code> , and <code>site_id</code> are ignored.
drive_name, drive_id	SharePoint Drive name or ID passed to <code>get_drive</code> method for SharePoint site object.
drive	A <code>ms_drive</code> object. If drive is supplied, <code>drive_name</code> and <code>drive_id</code> are ignored.
check_fields	If TRUE (default), column names (or record field names) for the input data are matched to the fields of the list object. If FALSE, names aren't checked and the Graph API errors for any name that isn't a list field.

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 meta-data.
create_list	If TRUE and list_name is supplied, a new list is created using data_as_column_definition_list() to set the column definitions for the list.
strict	If TRUE, all column names in a data frame (or field names in a list of records) must match field names in the supplied SharePoint list. If FALSE (default), unmatched names are dropped with a message. Only used if check_fields = TRUE.
.batch	If TRUE (default), items are sent with Microsoft Graph \$batch requests (up to 20 items per request), which is much faster than a separate request for each item. Requests throttled by the Graph API or failing with a temporary error (e.g. "Database is read only") are retried up to 5 times after the requested delay (or an increasing delay of up to 60 seconds). If any items fail, the remaining items are still sent and the error lists the failed items with the HTTP status. If FALSE, a separate request is sent for each item and the first failed item is an error. Use options(sharepointtr.batch = FALSE) to change the default for the session. In either case, requests are sent in parallel if mirai::daemons() are set (see purrr::in_parallel()). For create_sp_list_items() , items sent in a \$batch request (or in parallel) may be created in a different order than data, so new item IDs may not follow the order of data. Use .batch = FALSE (without daemons) if item IDs must follow the order of data.
.progress	Whether to show a progress bar. Use TRUE to turn on a basic progress bar, use a string to give it a name, or see progress_bars for more details.
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.
na_fields	How to handle NA fields in input data. One of "drop" (remove NA and empty fields, e.g. a multi-select value of character(0), before updating list items, leaving existing values in place) or "replace" (overwrite existing list values with new replacement NA values or, for multi-value fields, an empty selection).
drop_fields	Column names to drop from data even if they are listed as editable fields. Defaults to c("ContentType", "Attachments")
.data	A list or data frame with fields to update.
item_id	A SharePoint list item id. Either item_id or sp_list_item must be provided but not both.
sp_list_item	Optional. A SharePoint list item object to update.
.multi_fields	Optional. Names of multi-value (Collection) fields, such as multi-select choice columns, that should always be sent as an array with an "@odata.type" annotation. If NULL and sp_list is available, field names are found from the list column definitions. Otherwise, only fields with a length other than 1 are treated as multi-value fields.

Details

Validation of data 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 do not match fields, the function errors.

Value

`create_sp_list_items()` and `update_sp_list_items()` invisibly return the input data, unmodified (even if data is empty).

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
  )
}
```

```
create_sp_list_lookup_column
```

Create SharePoint list lookup column and update lookup column items

Description

[Experimental]

`update_sp_list_person_items()` is a variant of `update_sp_list_lookup_items()` for person or group columns. Person or group columns are a type of lookup column where the lookup list is the hidden "User Information List" for the site. By default, users are matched by email address (ignoring case). Users that have never accessed the site are not included in the "User Information List" and can't be matched.

Usage

```
create_sp_list_lookup_column(
  sp_list = NULL,
  column_name,
  lookup_list,
```

```

    lookup_list_column = column_name,
    ...,
    list_name = NULL,
    site = NULL,
    site_url = NULL,
    call = caller_env()
)

```

```

create_sp_list_person_column(
  sp_list = NULL,
  column_name,
  ...,
  allow_multiple_selection = NULL,
  display_as = NULL,
  from_type = "peopleOnly",
  list_name = NULL,
  site = NULL,
  site_url = NULL,
  allow_multiple = deprecated(),
  call = caller_env()
)

```

```

update_sp_list_lookup_items(
  data = NULL,
  sp_list = NULL,
  column_name,
  lookup_list_data = NULL,
  lookup_list = NULL,
  join_column = column_name,
  lookup_join_column = join_column,
  ...,
  .id = "id",
  ignore_case = FALSE,
  na_fields = c("drop", "replace"),
  .progress = TRUE,
  call = caller_env()
)

```

```

fmt_sp_list_lookup_items(
  data,
  column_name,
  lookup_list_data = NULL,
  lookup_list = NULL,
  lookup_join_column = column_name,
  ...,
  .id = "id",
  ignore_case = FALSE,
  call = caller_env()
)

```

```

)

update_sp_list_person_items(
    data = NULL,
    sp_list = NULL,
    column_name,
    join_column = column_name,
    user_column = "EMail",
    user_info_data = NULL,
    allow_hidden = FALSE,
    ...,
    .id = "id",
    na_fields = c("drop", "replace"),
    .progress = TRUE,
    call = caller_env()
)

```

Arguments

sp_list	A <code>ms_list</code> object. If supplied, <code>list_name</code> , <code>site</code> , and <code>site_url</code> are all ignored.
column_name	Name of the lookup (or person or group) column. For <code>create_sp_list_lookup_column()</code> , the name of the new column. For <code>update_sp_list_lookup_items()</code> and <code>update_sp_list_person_items()</code> , lookup ID values are written to the "{column_name}LookupId" field. For <code>fmt_sp_list_lookup_items()</code> , one or more column (or record element) names in <code>data</code> to format.
lookup_list	The lookup list as a <code>ms_list</code> object, list name, or list URL. The lookup list must be in the same site as <code>sp_list</code> and a list name is retrieved from the same site as <code>sp_list</code> . For <code>update_sp_list_lookup_items()</code> and <code>fmt_sp_list_lookup_items()</code> , <code>lookup_list</code> is only used to retrieve <code>lookup_list_data</code> if not supplied. For <code>fmt_sp_list_lookup_items()</code> , a list name requires site information (e.g. <code>site_url</code>) passed with <code>....</code>
lookup_list_column	Name of lookup column in the lookup list to use. Graph property: <code>columnName</code> .
...	Arguments passed on to <code>create_lookup_column</code>
	<code>.col_type</code> Column type. Defaults to "text". Must be one of "boolean", "calculated", "choice", "currency", "dateTime", "lookup", "number", "person-OrGroup", "text", "term", "hyperlinkOrPicture", "thumbnail", "contentApprovalStatus", or "geolocation".
	<code>enforce_unique</code> Enforce unique values in column. Graph property: <code>enforceUniqueValues</code> .
	<code>hidden</code> If TRUE, column will be hidden by default. Graph property: <code>hidden</code> .
	<code>deletable</code> If TRUE, column can't be deleted separate from the list. Graph property: <code>isDeletable</code> .
	<code>required</code> If TRUE, column will be required. Graph property: <code>required</code> .
	<code>default</code> Default value set by helper <code>get_column_default()</code> function. Graph property: <code>defaultValue</code> .
	<code>description</code> Column description. Graph property: <code>description</code> .
	<code>display_name</code> Column display name. Graph property: <code>displayName</code> .

displayname **[Deprecated]** Use `display_name`.
validation Column validation created with `column_validation()`. Graph property: `validation`.
indexed, sealed, propagate_changes, read_only, id Additional arguments used by `create_column_definition()`. Graph properties: `indexed`, `isSealed`, `propagateChanges`, `readOnly`, and `id`.
multiple_lines Logical. If TRUE, allow multiple lines of text. Graph property: `allowMultipleLines`.
append_changes Logical. If TRUE, append changes to existing value for column. Graph property: `appendChangesToExistingText`.
lines Whole number. Size of the text box. Graph property: `linesForEditing`.
max_length Whole number. Max length in number of characters. Graph property: `maxLength`.
text_type One of `c("plain", "richText")`. Graph property: `textType`.
choices A character vector of choice options. Graph property: `choices`.
allow_na If TRUE, allow NA values in choices.
na_replacement Used as replacement by `stringr::str_replace_na()` on choices if they contain NA values.
allow_text If TRUE, allow text entry in the choice column. Graph property: `allowTextEntry`.
decimal_places One of `c("automatic", "none", "one", "two", "three", "four", "five")` or a whole number between 0 and 5. Graph property: `decimalPlaces`.
max,min Minimum and maximum values allowed in number column. Graph properties: `maximum` and `minimum`.
decimals **[Deprecated]** Use `decimal_places`.
format For `create_datetime_column()`, `"dateOnly"` or `"dateTime"`. Graph property: `format`.
locale Locale used to set the currency symbol, e.g. `"en-us"`. Graph property: `locale`.
formula Required string with formula for calculated column definition. See [examples of common formulas in lists](#). Reference existing columns using the display name enclosed in square brackets. The formula must start with an equals sign `"=`" which this function appends to the formula text if it is missing. The formula is processed with `glue::glue()`. Graph property: `formula`.
output_type Value type returned by calculated formula. One of `c("text", "boolean", "currency", "dateTime", "number")`. Defaults to `"text"`. Graph property: `outputType`.
allow_multiple_values If TRUE, allow a lookup or term column to store multiple values. Graph property: `allowMultipleValues`.
allow_unlimited_length If TRUE, allow lookup column to return any length value. Graph property: `allowUnlimitedLength`.
primary_lookup_column_id If column definition is for a secondary column, the primary lookup column ID must be supplied. Graph property: `primaryLookupColumnId`.

	<code>is_picture</code> Logical indicator for display of hyperlink value as link (FALSE, default for <code>create_hyperlink_column()</code>) or image (TRUE, default for <code>create_picture_column()</code>). Graph property: <code>isPicture</code>. <code>show_full_name</code> If TRUE, display the entire term path. Graph property: <code>showFullyQualifiedName</code>. <code>split</code> character vector (or object which can be coerced to such) containing regular expression(s) (unless <code>fixed = TRUE</code>) to use for splitting. If empty matches occur, in particular if <code>split</code> has length 0, <code>x</code> is split into single characters. If <code>split</code> has length greater than 1, it is re-cycled along <code>x</code>.
<code>list_name</code>	List name. Required if <code>sp_list</code> is NULL.
<code>site</code>	A <code>ms_site</code> object. If <code>site</code> is supplied, <code>site_url</code> , <code>site_name</code> , and <code>site_id</code> are ignored.
<code>site_url</code>	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.
<code>call</code>	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of abort() for more information.
<code>allow_multiple_selection</code>	If TRUE, allow a person or group column to store multiple values. Graph property: <code>allowMultipleSelection</code> .
<code>display_as</code>	Value displayed as option. For <code>create_choice_column</code> one of c("checkboxes", "dropDownMenu", "radioButtons"). For <code>create_number_column</code> , one of c("number", "percentage"). For <code>create_datetime_column</code> , one of c("default", "friendly", "standard"). Graph property: <code>displayAs</code> .
<code>from_type</code>	What type of resources to choose from. Defaults to "peopleOnly" for <code>create_person_column()</code> or "peopleAndGroups" for <code>create_group_column()</code> . Graph property: <code>chooseFromType</code> .
<code>allow_multiple</code>	[Deprecated] Use <code>allow_multiple_values</code> for <code>create_lookup_column()</code> and <code>create_term_column()</code> or <code>allow_multiple_selection</code> for <code>create_person_column()</code> and <code>create_group_column()</code> .
<code>data</code>	Optional. A data frame, a list of named lists (one record per item), or a single named list record with item ID values (<code>.id</code>) and join values (<code>join_column</code>) for the items to update. A single record or a named list of records is returned by fmt_sp_list_lookup_items() as an unnamed list of records. If NULL, items are retrieved from <code>sp_list</code> . Required for fmt_sp_list_lookup_items() where <code>data</code> must include all <code>column_name</code> values as <code>column</code> (or record element) names.
<code>lookup_list_data</code>	Optional. A data frame or a list of named lists with item ID values (<code>.id</code>) and unique join values (<code>lookup_join_column</code>) for the lookup list items. If NULL, items are retrieved from <code>lookup_list</code> .
<code>join_column</code>	Name of column (or record element) in <code>data</code> to match items to lookup list items. Defaults to <code>column_name</code> . Items without a matching lookup list item are not updated when <code>na_fields = "drop"</code> .
<code>lookup_join_column</code>	Name of column (or record element) in <code>lookup_list_data</code> with values to match to <code>join_column</code> values. Defaults to <code>join_column</code> . For fmt_sp_list_lookup_items() ,

	lookup_join_column must be length 1 or the same length as column_name and defaults to column_name.
.id	Name of column (or record element) with item ID values in data and lookup_list_data. Defaults to "id".
ignore_case	If TRUE, ignore case when matching join values. Defaults to FALSE.
na_fields	How to handle NA fields in input data. One of "drop" (remove NA and empty fields, e.g. a multi-select value of character(0), before updating list items, leaving existing values in place) or "replace" (overwrite existing list values with new replacement NA values or, for multi-value fields, an empty selection).
.progress	Whether to show a progress bar. Use TRUE to turn on a basic progress bar, use a string to give it a name, or see progress_bars for more details.
user_column	Name of column (or record element) in user_info_data with values to match to join_column values. Defaults to "EMail".
user_info_data	Optional. A data frame or an unnamed list of named lists from the "User Information List" for the site. If NULL, items are retrieved with list_sp_site_user_info() for the site of sp_list.
allow_hidden	If TRUE, include users with UserInfoHidden = TRUE in the possible matches. Defaults to FALSE.

Details

[create_sp_list_lookup_column\(\)](#) is a wrapper for [create_lookup_column\(\)](#) and [create_sp_list_column\(\)](#) that creates a lookup column in a list (sp_list) using a column from a second list in the same site (the "lookup" list). [create_sp_list_person_column\(\)](#) is a wrapper for [create_person_column\(\)](#) and [create_sp_list_column\(\)](#) that creates a person or group column.

[fmt_sp_list_lookup_items\(\)](#) formats one or more lookup (or person or group) columns in data by replacing the values with matching lookup list item ID values and renaming the columns to "{column_name}LookupId". Use the output with [create_sp_list_items\(\)](#) or [update_sp_list_items\(\)](#).

Values in data that can't be matched to a lookup list item are listed in a message and replaced with NA values. Missing values are never matched.

data_as_column_definition_list

Convert a data frame to a column definition list

Description

[data_as_column_definition_list\(\)](#) is used to create a column definition list based on an existing data frame. This function is used internally by [create_sp_list_items\(\)](#) when create_list = TRUE.

Usage

```
data_as_column_definition_list(
  data,
  ...,
  split = "|",
  ignore_na = TRUE,
  definitions_as = c("definition_list", "table")
)
```

Arguments

<code>data</code>	A data frame input. Column types are used to infer the appropriate Microsoft Lists column definition.
<code>...</code>	Ignored.
<code>split</code>	character vector (or object which can be coerced to such) containing regular expression (s) (unless <code>fixed = TRUE</code>) to use for splitting. If empty matches occur, in particular if <code>split</code> has length 0, <code>x</code> is split into single characters. If <code>split</code> has length greater than 1, it is re-cycled along <code>x</code> .
<code>ignore_na</code>	If <code>TRUE</code> , drop any parameters with a NA value.
<code>definitions_as</code>	If <code>"definition_list"</code> (default) return named list output from <code>create_column_definition_list()</code> . If <code>"table"</code> return a dataframe with the column names and types.

Details

Converting R data types to SharePoint column definitions

The type for each vector in the input data frame is checked with `vctrs::vec_ptype_abbr` and mapped to corresponding SharePoint list column definitions:

- factors are specified as choice columns
- integers are specified as number columns with `decimal_places` set to "none"
- characters with any value exceeding 255 characters have `multiple_lines` set to `TRUE`
- characters composed entirely of URL values are specified as hyperlink columns
- dates are specified as date columns
- dtm values are specified as datetime columns
- logical values are specified as boolean columns if they include no NA values or text columns if they do

All other vectors are specified as text columns. If the levels of any input factor column contain the character specified with `split`, this function errors.

Value

If `definitions_as = "definition_list"` (default), a list of named lists formatted as `columnDefinitions` for use as the `fields` argument to `create_sp_list()`. If `definitions_as = "table"`, a data frame with one row per column of data describing the inferred name, type, and other definition properties.

Examples

```
data_as_column_definition_list(mtcars)
```

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 <i>not</i> 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 <code>default_drive_name</code> argument is used as the <code>drive_name</code> and the <code>item_id</code> 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 <code>by_item = TRUE</code> 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 <code>get_sp_item</code> <code>drive_name, drive_id</code> 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.

Value

Invisibly returns NULL.

delete_sp_list_item	<i>Delete SharePoint list item or items</i>
---------------------	---

Description

[delete_sp_list_item\(\)](#) deletes a single SharePoint list item and [delete_sp_list_items\(\)](#) deletes multiple SharePoint list items. Set confirm = FALSE to use without interactive confirmation.

Usage

```

delete_sp_list_item(
  item_id = NULL,
  sp_list_item = NULL,
  ...,
  .id = "id",
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  site_url = NULL,
  site = NULL,
  confirm = TRUE,
  call = caller_env()
)

delete_sp_list_items(
  item_id = NULL,
  ...,
  .id = "id",
  sp_list = NULL,
  filter = NULL,
  confirm = TRUE,
  .batch = getOption("sharepointtr.batch", TRUE),
  .progress = TRUE,
  call = caller_env()
)

```

Arguments

<code>item_id</code>	ID value for list item or items to delete. <code>item_id</code> can also be a data frame with a column named with the <code>.id</code> value, a single named list record, or (for delete_sp_list_items()) a list of named lists (one per item) where each record includes an element named with the <code>.id</code> value. Item IDs must be whole numbers or non-empty strings and are checked before any items are deleted. delete_sp_list_item() requires a single item ID.
<code>sp_list_item</code>	Optional. A SharePoint list item object to delete.
<code>...</code>	For delete_sp_list_item() , must be empty. For delete_sp_list_items() , additional parameters passed to get_sp_list() if <code>sp_list</code> is <code>NULL</code> .
<code>.id</code>	Name of column (if <code>item_id</code> is a data frame) or element (if <code>item_id</code> is a list of records) to use for item ID values. Defaults to "id".
<code>list_name, list_id</code>	SharePoint List name or ID string.
<code>sp_list</code>	A <code>ms_list</code> object. If supplied, <code>list_name</code> , <code>list_id</code> , <code>site_url</code> , <code>site</code> , <code>drive_name</code> , <code>drive_id</code> , <code>drive</code> , and any additional parameters passed to <code>...</code> are all ignored.
<code>site_url</code>	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 <code>ms_site</code> object. If site is supplied, <code>site_url</code> , <code>site_name</code> , and <code>site_id</code> are ignored.
confirm	If TRUE (default), user confirmation is required to delete items.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.
filter	Optional. A string with an OData filter expression used to find the items to delete if <code>item_id</code> is NULL. Can't be supplied with <code>item_id</code> . See <code>list_sp_list_items()</code> .
.batch	If TRUE (default), items are sent with Microsoft Graph \$batch requests (up to 20 items per request), which is much faster than a separate request for each item. Requests throttled by the Graph API or failing with a temporary error (e.g. "Database is read only") are retried up to 5 times after the requested delay (or an increasing delay of up to 60 seconds). If any items fail, the remaining items are still sent and the error lists the failed items with the HTTP status. If FALSE, a separate request is sent for each item and the first failed item is an error. Use <code>options(sharepointtr.batch = FALSE)</code> to change the default for the session. In either case, requests are sent in parallel if <code>mirai::daemons()</code> are set (see <code>purrr::in_parallel()</code>). For <code>create_sp_list_items()</code> , items sent in a \$batch request (or in parallel) may be created in a different order than data, so new item IDs may not follow the order of data. Use <code>.batch = FALSE</code> (without daemons) if item IDs must follow the order of data.
.progress	Whether to show a progress bar. Use TRUE to turn on a basic progress bar, use a string to give it a name, or see <code>progress_bars</code> for more details.

Value

For `delete_sp_list_item()`, invisibly returns an empty list with a "status" attribute giving the HTTP response status code. For `delete_sp_list_items()`, invisibly returns a list of these responses, one per deleted item.

download_sp_item	<i>Download one or more items from SharePoint to a file or folder</i>
------------------	---

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. If {carrier} and {mirai} are installed and `mirai::daemons()` are set, items are downloaded in parallel. The `parallel` argument is separate: it is passed to the download method to download the contents of a folder in parallel.

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. `keep_list_cols` is intended to allow the preservation of list columns but it is not yet supported.

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
	<code>list_name</code> , <code>list_id</code> SharePoint List name or ID string.
	<code>as_data_frame</code> 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 <code>metadata = TRUE</code> as list metadata is always returned as a data frame.
	<code>metadata</code> If TRUE, get_sp_list() applies the <code>get_column_info</code> method to the returned SharePoint list and returns a data frame with column metadata for the list.
	<code>keep</code> One of "all" (default), "editable", "external" (non-internal fields). Argument determines if the returned list metadata includes read only columns or hidden columns.
	<code>sync_fields</code> If TRUE, use the <code>sync_fields</code> method to sync the fields of the local <code>ms_list</code> object with the fields of the SharePoint List source before retrieving list metadata.

	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.
	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.
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.

Value

Invisibly returns the path to the downloaded file.

get_sp_group	<i>Get SharePoint group for a site or list group members</i>
--------------	--

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,
  ...,
  sp_group = 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.
sp_group	A "az_group" object returned by get_sp_group() or using the get_group method.
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.

Value

For [get_sp_group\(\)](#), an az_group object for the site. For [list_sp_group_members\(\)](#), a data frame of group members if as_data_frame = TRUE (default), or a list of az_user objects if FALSE.

Description

`get_sp_item()` wraps the `get_item` method for `ms_drive` objects and returns a `ms_drive_item` object by default. Additional parameters in ... are passed to `get_sp_drive()`.

Use `get_sp_item_properties()` to get the item properties instead. Setting `properties = TRUE` returns the same properties, but only `get_sp_item_properties()` can include the document library column values (fields) for the item.

If `path` or `item_url` is a SharePoint URL for a file or folder (e.g. a link from "Copy link" in SharePoint, a document URL, or a document library URL) and `item_id`, `drive_name`, `drive_id`, and `drive` aren't supplied, `get_sp_item()` gets the item with the **Microsoft Graph shares API**. This works for items in any document library and for links to items that have been moved. The Microsoft Graph login uses `site`, `site_url`, or the site in the URL. If the shares API request fails, the `site`, `drive`, and `path` or item ID are parsed from the URL instead.

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()
)
```

Arguments

<code>path</code>	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 <i>not</i> 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 <code>default_drive_name</code> argument is used as the <code>drive_name</code> and the <code>item_id</code> is extracted from the URL.
<code>item_id</code>	A SharePoint item ID passed to the <code>itemid</code> parameter of the <code>get_item</code> method for <code>ms_drive</code> objects.
<code>item_url</code>	A SharePoint item URL used to parse the item ID, drive name, and site URL.
...	Arguments passed on to <code>get_sp_drive</code>
<code>drive_name</code> , <code>drive_id</code>	SharePoint Drive name or ID passed to <code>get_drive</code> method for SharePoint site object.
<code>drive_url</code>	A SharePoint Drive URL to parse for a Drive name and other information. If <code>drive_name</code> is a URL, it is used as <code>drive_url</code> .
<code>default_drive_name</code>	Drive name string used only if input is a document URL and drive name is not part of the URL. Defaults to <code>getOption("sharepointtr.default_drive_name", "Documents")</code>

cache	If TRUE, cache drive to a file using <code>cache_sp_drive()</code> .
refresh	If TRUE, get a new drive even if the existing drive is cached as a local option. If FALSE, use the cached <code>ms_drive</code> object if it exists.
cache_file	File name for cached drive or site. Default NULL.
site	A <code>ms_site</code> object. If site is supplied, <code>site_url</code> , <code>site_name</code> , and <code>site_id</code> are ignored.
overwrite	If TRUE, replace the existing cached object named by <code>cache_file</code> with the new object. If FALSE, error if a cached file with the same <code>cache_file</code> name already exists.
drive_name, drive_id	SharePoint drive name or ID.
drive	A <code>ms_drive</code> object. If drive is supplied, <code>drive_name</code> , <code>site_url</code> , and any additional parameters passed to <code>...</code> 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 <code>get_item_properties</code> method and return item properties instead of the item.
as_data_frame	If TRUE, return a data frame. If FALSE (default), return a <code>ms_item</code> or <code>ms_item_properties</code> object.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

A `ms_drive_item` object (or a list of item properties if `properties = TRUE`), or a 1 row data frame if `as_data_frame = TRUE`.

See Also

`get_sp_item_properties()`, `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
  )
}
```

get_sp_list_column	<i>Get SharePoint list column definition for a single column</i>
--------------------	--

Description

`get_sp_list_column()` gets a list column definition for a single column specified by column name or ID.

Usage

```
get_sp_list_column(
  sp_list = NULL,
  column_name = NULL,
  column_id = NULL,
  ...,
  list_name = NULL,
  site_url = NULL,
  site = NULL,
  column_name_type = "name",
  call = caller_env()
)
```

Arguments

<code>sp_list</code>	A <code>ms_list</code> object. If supplied, <code>list_name</code> , <code>list_id</code> , <code>site_url</code> , <code>site</code> , <code>drive_name</code> , <code>drive_id</code> , <code>drive</code> , and any additional parameters passed to ... are all ignored.
<code>column_name</code> , <code>column_id</code>	Column name or ID to get a definition for.
...	Arguments passed on to <code>get_sp_list</code>
<code>list_name</code> , <code>list_id</code>	SharePoint List name or ID string.
<code>keep</code>	One of "all" (default), "editable", "external" (non-internal fields). Argument determines if the returned list metadata includes read only columns or hidden columns.
<code>sync_fields</code>	If TRUE, use the <code>sync_fields</code> method to sync the fields of the local <code>ms_list</code> object with the fields of the SharePoint List source before retrieving list metadata.
<code>drive_name</code> , <code>drive_id</code>	SharePoint Drive name or ID passed to <code>get_drive</code> method for SharePoint site object.
<code>drive</code>	A <code>ms_drive</code> object. If drive is supplied, <code>drive_name</code> and <code>drive_id</code> are ignored.
<code>filter</code>	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.
<code>n</code>	Maximum number of lists, plans, tasks, or other items to return. Defaults to NULL which sets <code>n</code> to Inf.

list_name	SharePoint List name. Optional if sp_list or list_id are provided.
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.
column_name_type	"name" or "displayName". Used to match column ID so column_name must be unique if column_name_type = "displayName".
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

See the [Get columnDefinition Graph API documentation](#) for more information.

Value

A named list with the columnDefinition resource for the matched column.

list_sp_drives	<i>List drives for a SharePoint site</i>
----------------	--

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 drive or site. Default NULL.
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_dir	Cache directory. By default, uses an option named "sharepoint.cache_dir". If "sharepoint.cache_dir" is not set, the cache directory is set to <code>rappdirs::user_cache_dir("sharepoint")</code> .
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. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

A data frame with a list column of ms_drive objects if as_data_frame = TRUE (default), or a list of ms_drive objects if FALSE.

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_list_views	<i>List, get, create, update, or delete SharePoint list views</i>
--------------------	---

Description**[Experimental]**

- [list_sp_list_views\(\)](#) lists the views for a SharePoint list.
- [get_sp_list_view\(\)](#) gets a single view by title or ID (or the default view).

- `create_sp_list_view()` creates a view.
- `update_sp_list_view()` updates a view. Only properties that differ from the existing view are changed.
- `delete_sp_list_view()` deletes a view. The default view can't be deleted.

The Graph API doesn't support list views, so these functions use the SharePoint REST API and require a delegated (user) login with a refresh token, such as the default Microsoft365R login. View properties use the SharePoint REST API **SP.View** property names (e.g. ViewFields or RowLimit) and the arguments use the same names in snake case (e.g. view_fields or row_limit).

Usage

```
list_sp_list_views(  
    sp_list = NULL,  
    ...,  
    hidden = FALSE,  
    as_data_frame = TRUE,  
    list_name = NULL,  
    site_url = NULL,  
    site = NULL,  
    call = caller_env()  
)  
  
get_sp_list_view(  
    sp_list = NULL,  
    view_title = NULL,  
    view_id = NULL,  
    ...,  
    list_name = NULL,  
    site_url = NULL,  
    site = NULL,  
    call = caller_env()  
)  
  
create_sp_list_view(  
    sp_list = NULL,  
    title = NULL,  
    ...,  
    view_fields = NULL,  
    view_query = NULL,  
    row_limit = NULL,  
    paged = NULL,  
    default_view = NULL,  
    hidden = NULL,  
    scope = NULL,  
    custom_formatter = NULL,  
    view_definition = NULL,  
    list_name = NULL,  
    site_url = NULL,
```

```

    site = NULL,
    call = caller_env()
)

update_sp_list_view(
  sp_list = NULL,
  view_title = NULL,
  view_id = NULL,
  ...,
  title = NULL,
  view_fields = NULL,
  view_query = NULL,
  row_limit = NULL,
  paged = NULL,
  default_view = NULL,
  hidden = NULL,
  scope = NULL,
  custom_formatter = NULL,
  view_definition = NULL,
  list_name = NULL,
  site_url = NULL,
  site = NULL,
  call = caller_env()
)

delete_sp_list_view(
  sp_list = NULL,
  view_title = NULL,
  view_id = NULL,
  ...,
  confirm = TRUE,
  list_name = NULL,
  site_url = NULL,
  site = NULL,
  call = caller_env()
)

```

Arguments

<code>sp_list</code>	A <code>ms_list</code> object. If <code>NULL</code> , the list is retrieved with <code>get_sp_list()</code> using <code>list_name</code> and any additional arguments passed to ...
...	Additional arguments passed to <code>get_sp_list()</code> if <code>sp_list</code> is <code>NULL</code> .
<code>hidden</code>	For <code>list_sp_list_views()</code> , if <code>TRUE</code> , include hidden views. For <code>create_sp_list_view()</code> and <code>update_sp_list_view()</code> , if <code>TRUE</code> , hide the view. SP.View property: Hidden.
<code>as_data_frame</code>	If <code>TRUE</code> (default), return a data frame with one row per view. If <code>FALSE</code> , return a list of views.
<code>list_name</code>	List name or URL. Used to get the list if <code>sp_list</code> is <code>NULL</code> .

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.
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.
view_title, view_id	Title or ID of an existing view. If both are NULL, get_sp_list_view() returns the default view.
title	View title. Required for create_sp_list_view() unless supplied with view_definition. For update_sp_list_view() , a new title for the view. SP.View property: Title.
view_fields	Character vector of internal column names to show in the view, in order. Use "LinkTitle" for the title column with a link to the item. For create_sp_list_view() , defaults to the fields of the default view. SP.View property: ViewFields.
view_query	A CAML query used to filter, sort, or group items (see details). SP.View property: ViewQuery.
row_limit	Number of items to show per page. SP.View property: RowLimit.
paged	If TRUE, show items in pages of row_limit items. SP.View property: Paged.
default_view	If TRUE, make the view the default view for the list. The current default view is no longer the default. SP.View property: DefaultView.
scope	For lists with folders, whether to show items in folders. One of 0 (default), 1 (recursive), 2 (recursive all), or 3 (files only). SP.View property: Scope.
custom_formatter	JSON view formatting as a JSON string or a list. See Use view formatting to customize SharePoint . SP.View property: CustomFormatter.
view_definition	Optional. A named list of SP.View properties (e.g. list(Title = "Active", RowLimit = 50)). Used in place of the other view arguments, which can't be supplied with view_definition.
confirm	If TRUE (default), ask for confirmation before deleting a view.

Details

View queries

view_query is a **CAML** query with optional <Where>, <OrderBy>, and <GroupBy> elements (but without an enclosing <Query> element). Reference columns by internal name. For example, to show active items sorted by amount:

```
<Where><Eq><FieldRef Name="Status"/><Value Type="Choice">Active</Value></Eq></Where>
<OrderBy><FieldRef Name="Amount" Ascending="FALSE"/></OrderBy>
```

Limitations

- Board, gallery, and calendar views can be listed but their layout settings can't be changed.
- Personal views can be listed but not created.
- Hidden views include list form settings (e.g. a hidden "Untitled Form" view holds custom form formatting) and are excluded by default.
- Changing view fields removes and re-adds every field. If a request fails partway through, call `update_sp_list_view()` again.
- Renaming a view doesn't change the view URL.

Value

`list_sp_list_views()` returns a data frame or a list of views. The data frame always has the same columns (Id, Title, DefaultView, Hidden, ViewFields, ViewQuery, RowLimit, Paged, Scope, CustomFormatter, MobileView, MobileDefaultView, ServerRelativeUrl, ViewType, and PersonalView), even if the list has no views. ViewFields is a list column and missing values are NA. `get_sp_list_view()` returns a named list of SP.View properties (Id, Title, ViewFields, ViewQuery, RowLimit, DefaultView, and other properties). `create_sp_list_view()` and `update_sp_list_view()` invisibly return the created or updated view in the same format. `delete_sp_list_view()` invisibly returns NULL.

Examples

```
## Not run:
list_sp_list_views(list_name = "Projects", site_url = "<SharePoint site url>")

create_sp_list_view(
  sp_list,
  title = "Active Projects",
  view_fields = c("LinkTitle", "Status", "Amount"),
  view_query = '<Where><Eq><FieldRef Name="Status"/>
    <Value Type="Choice">Active</Value></Eq></Where>',
  row_limit = 50
)

update_sp_list_view(
  sp_list,
  view_title = "Active Projects",
  default_view = TRUE
)

## End(Not run)
```

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 drive or site. Default NULL.
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_dir	Cache directory. By default, uses an option named "sharepoint.cache_dir". If "sharepoint.cache_dir" is not set, the cache directory is set to <code>rappdirs::user_cache_dir("sharepoint")</code> .
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. <code>caller_env()</code> . 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.

Value

For [list_sp_pages\(\)](#), a data frame with a list column of ms_page objects if as_data_frame = TRUE (default), or a list of page properties if FALSE. For [get_sp_page\(\)](#), a named list with the properties of the matched sitePage.

list_sp_site_user_info

List items from the hidden SharePoint "User Information List"

Description

[Experimental]

Usage

```
list_sp_site_user_info(
  ...,
  sp_site = NULL,
  user_type = c("all", "visible"),
  filter = NULL,
  as_data_frame = TRUE,
  call = caller_env()
)
```

Arguments

...	Additional parameters passed to get_sp_site() if <code>sp_site</code> is not supplied.
<code>sp_site</code>	A <code>ms_site</code> object.
<code>user_type</code>	Type of users to return. "all" (default) returns all users and "visible" excludes users where <code>UserInfoHidden</code> is <code>TRUE</code> .
<code>filter</code>	Optional. A filter query string passed to list_sp_list_items() . If <code>user_type = "visible"</code> , <code>filter</code> is combined with a filter for visible users.
<code>as_data_frame</code>	If <code>TRUE</code> , 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 <code>n</code> rows or all lists available for the SharePoint site or drive. Defaults to <code>FALSE</code> . Ignored if <code>metadata = TRUE</code> as list metadata is always returned as a data frame.
<code>call</code>	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of abort() for more information.

Details

[list_sp_site_user_info\(\)](#) lists users from the hidden "User Information List" for a site. Use the results to match users to lookup ID values for person or group columns with [fmt_sp_list_lookup_items\(\)](#) or [update_sp_list_person_items\(\)](#).

read_sharepoint	<i>Read a SharePoint item based on a file URL or file name and site details</i>
-----------------	---

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()`.

The function also serves as a wrapper for `list_sp_list_items()` if the provided file parameter is a SharePoint list URL or a `ms_list` object.

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 <code>item_id</code> is supplied, a file name to use in combination with <code>new_path</code> to set dest with location and filename for downloaded item. If file appears to be a SharePoint list URL, the list items are retrived with <code>list_sp_list_items()</code> .
...	Additional parameters passed to one of the functions identified in the description or supplied to <code>.f</code>

.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 <code>fs::dir_create()</code> .
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.

Value

A data frame of list items if file is a SharePoint list URL or ms_list object. Otherwise, the object returned by .f (if supplied), or by whichever reader function is selected based on the downloaded file's extension (see Description).

sp_dir_create	<i>Create SharePoint folders</i>
---------------	----------------------------------

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 <code>getOption("sharepointtr.default_drive_name")</code> "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.
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.
cache_file	File name for cached drive or site. Default NULL.
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. <code>caller_env()</code> . 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 path (or path combined with relative if supplied).

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 "/". path can be a string or a character vector. If a vector or path of URLs are supplied, provide a drive object to improve performance. If path is a SharePoint folder or document library URL and drive_name, drive_id, and drive aren't supplied, the drive and folder are found with the Microsoft Graph shares API (see get_sp_item()). If the request fails, the drive and folder path are parsed from the URL instead.
...	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 <code>getOption("sharepoint.default_drive_name")</code> "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.
	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.
	cache_file File name for cached drive or site. Default NULL.
	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 <i>not</i> 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.

Value

For [sp_dir_info\(\)](#), a data frame with one row per file or folder at path, with columns determined by info ("partial", "name", or "all"). For [sp_dir_ls\(\)](#), a character vector of file and folder names or paths.

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("sharepointtr.default_drive_name", "Documents"),
  cache = getOption("sharepointtr.cache", FALSE),
  refresh = getOption("sharepointtr.refresh", TRUE),
  overwrite = FALSE,
  cache_file = NULL,
  call = caller_env()
)

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

Arguments

<code>drive_name, drive_id</code>	SharePoint Drive name or ID passed to <code>get_drive</code> method for SharePoint site object.
<code>drive_url</code>	A SharePoint Drive URL to parse for a Drive name and other information. If <code>drive_name</code> is a URL, it is used as <code>drive_url</code> .
<code>properties</code>	If TRUE, return the drive properties instead of the <code>ms_drive</code> object. Defaults to FALSE.
<code>...</code>	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.
	cache_dir Cache directory. By default, uses an option named "sharepoint.cache_dir". If "sharepoint.cache_dir" is not set, the cache directory is set to rappdirs::user_cache_dir("sharepoint")
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("sharepoint.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 or site. Default NULL.
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.

Value

For [get_sp_drive\(\)](#), a ms_drive object, or the drive properties as a named list if properties = TRUE. For [cache_sp_drive\(\)](#), invisibly returns NULL.

See Also

[Microsoft365R::ms_drive](#)

sp_item_copy	<i>Copy or move SharePoint files and folders</i>
--------------	--

Description

These functions copy or move items in a SharePoint document library. Their interface is based on `fs::file_copy()`, `fs::file_move()`, and `fs::dir_copy()`:

- `sp_item_copy()` copies files. Use `sp_dir_copy()` to copy folders.
- `sp_item_move()` moves (or renames) files or folders.
- `sp_dir_copy()` and `sp_dir_move()` copy or move folders and their contents.

As with `{fs}`, if `new_path` is an existing folder, each item is placed inside that folder and keeps its name. Otherwise, `new_path` is the full path for the new item and its parent folder must already exist. A `new_path` that ends in `"/"` must be an existing folder.

`new_path` can be a path relative to the drive that holds the item, or a SharePoint folder or document library URL. Items can be copied to another drive or site, but they can only be moved within the same drive.

Usage

```
sp_item_copy(  
  path,  
  new_path,  
  ...,  
  overwrite = FALSE,  
  wait = TRUE,  
  drive_name = NULL,  
  drive_id = NULL,  
  drive = NULL,  
  site_url = NULL,  
  call = caller_env()  
)
```

```
sp_item_move(  
  path,  
  new_path,  
  ...,  
  overwrite = FALSE,  
  drive_name = NULL,  
  drive_id = NULL,  
  drive = NULL,  
  site_url = NULL,  
  call = caller_env()  
)
```

```
sp_dir_copy(  
  path,  
  new_path,  
  ...,  
  overwrite = FALSE,  
  wait = TRUE,
```

```

    drive_name = NULL,
    drive_id = NULL,
    drive = NULL,
    site_url = NULL,
    call = caller_env()
)

sp_dir_move(
  path,
  new_path,
  ...,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  site_url = NULL,
  call = caller_env()
)

```

Arguments

path	A character vector of file or folder paths, or SharePoint file or folder URLs. A relative path must not include the drive name. path can also be a <code>ms_drive_item</code> object or a list of <code>ms_drive_item</code> objects.
new_path	A character vector of new paths, or SharePoint folder or document library URLs. new_path must be length 1 or the same length as path. A relative new_path is in the same drive as the item it is paired with.
...	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 <code>getOption("sharepointtr.default_drive_name")</code> "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 <code>ms_drive</code> object if it exists.
cache_file	File name for cached drive or site. Default NULL.
site	A <code>ms_site</code> object. If site is supplied, site_url, site_name, and site_id are ignored.
overwrite	If FALSE (default), an error is returned if an item already exists at the new path. If TRUE, an existing file is replaced. SharePoint can't replace a folder, so sp_dir_copy() works like <code>fs::dir_copy()</code> instead: if new_path is an existing folder and overwrite = TRUE, the contents of path are copied into new_path (replacing any files with the same name). sp_dir_move() has no overwrite argument because a folder can't be moved to an existing folder.
wait	If TRUE (default), wait until each copy is complete. See details.
drive_name, drive_id	SharePoint drive name or ID.

drive	A <code>ms_drive</code> object. If drive is supplied, <code>drive_name</code> , <code>site_url</code> , 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. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of abort() for more information.

Details

Copying is asynchronous

Microsoft Graph copies items in the background. If `wait = TRUE` (the default), [sp_item_copy\(\)](#) and [sp_dir_copy\(\)](#) check the status of each copy until it is complete, so they can report an error if a copy fails. Large folders can take a while to copy. If `wait = FALSE`, the copy may still be running when the function returns, and a failed copy isn't reported.

Value

Invisibly returns a character vector with the new path of each item, relative to the drive it was copied or moved to.

See Also

[Microsoft365R::ms_drive_item](#)

Examples

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

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

  # Copy a file into an existing folder
  sp_item_copy("Reports/report.docx", "Archive", drive = drive)

  # Rename a file
  sp_item_move(
    "Reports/report.docx",
    "Reports/report-2026.docx",
    drive = drive
  )

  # Copy a folder to a new folder named "Reports-backup"
  sp_dir_copy("Reports", "Reports-backup", drive = drive)

  # Move a folder into the folder at a SharePoint URL
  sp_dir_move("Reports-backup", dir_url, drive = drive)
}
```

sp_item_properties	<i>Get or update the properties of a SharePoint file or folder</i>
--------------------	--

Description

`get_sp_item_properties()` gets the properties of a file or folder, such as the name, size, and creation date. Set `fields = TRUE` to also get the values of the document library columns (fields) for the item. To get the fields for many items, use `get_sp_list()` to get the document library as a list and use `list_sp_list_items()` or `get_sp_list_item()` instead.

[Experimental]

`update_sp_item_properties()` renames a file or folder or updates the values of document library columns for the item. Field values are updated with `update_sp_list_item()` using the list for the document library, so the same field name checks and formatting (e.g. for multi-choice columns) are used. To move an item to another folder, use `sp_item_move()` instead.

Usage

```
get_sp_item_properties(  
  path = NULL,  
  item_id = NULL,  
  item_url = NULL,  
  ...,  
  drive = NULL,  
  drive_name = NULL,  
  drive_id = NULL,  
  site_url = NULL,  
  fields = FALSE,  
  as_data_frame = FALSE,  
  call = caller_env()  
)  
  
update_sp_item_properties(  
  path = NULL,  
  ...,  
  .data = NULL,  
  name = NULL,  
  item_id = NULL,  
  item_url = NULL,  
  item = NULL,  
  check_fields = TRUE,  
  na_fields = c("drop", "replace"),  
  drive_name = NULL,  
  drive_id = NULL,  
  drive = NULL,  
  site_url = NULL,  
  fields = FALSE,
```

```

    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 <i>not</i> 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.
...	For get_sp_item_properties() , additional parameters passed to get_sp_drive() . For update_sp_item_properties() , named field values to update, e.g. Title = "Annual report". Names can be column names or display names. Ignored if .data is supplied.
drive	A ms_drive object. If drive is supplied, drive_name, site_url, and any additional parameters passed to ... are ignored.
drive_name, drive_id	SharePoint drive name or 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.
fields	If TRUE, add a fields element with the document library column values for the item (a named list). The id field is the list item id (not the drive item id). If as_data_frame = TRUE, fields is a list column with a 1 row data frame of the field values (use <code>tidyr::unnest(data, fields, names_sep = "_")</code> to add the fields as columns). Defaults to FALSE. Fields are only available for items in a SharePoint document library.
as_data_frame	If TRUE, return a 1 row data frame. If FALSE (default), return a list of item properties.
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 named list or a single row data frame with field values to update. Optional.
name	A new name for the item. Optional. name must not include a folder path. An error is returned if an item with the same name already exists in the folder.
item	A ms_drive_item object. If item is supplied, path, item_id, item_url, and the drive arguments are ignored.
check_fields	If TRUE (default), column names (or record field names) for the input data are matched to the fields of the list object. If FALSE, names aren't checked and the Graph API errors for any name that isn't a list field.

na_fields How to handle NA fields in input data. One of "drop" (remove NA and empty fields, e.g. a multi-select value of `character(0)`, before updating list items, leaving existing values in place) or "replace" (overwrite existing list values with new replacement NA values or, for multi-value fields, an empty selection).

Value

A list of item properties, or a 1 row data frame if `as_data_frame = TRUE`. `update_sp_item_properties()` returns the properties after the update.

See Also

`get_sp_item()`, `get_sp_list_item()`

Examples

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

if (is_sp_url(sp_item_url)) {
  get_sp_item_properties(sp_item_url, fields = TRUE)

  update_sp_item_properties(
    sp_item_url,
    Title = "Annual report",
    name = "annual-report-2026.docx"
  )
}
```

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", "external"),
  sync_fields = FALSE,
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  as_data_frame = TRUE,
  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("sharepointtr.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 or site. Default NULL.
	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", "external" (non-internal fields). 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.

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](#)
- [create_sp_list\(\)](#); [delete_sp_list\(\)](#)

sp_list_definition	<i>Read, write, and create SharePoint list definitions</i>
--------------------	--

Description

[Experimental]

A SharePoint list definition describes a list and its columns using Microsoft Graph list and columnDefinition property names. Definitions can be stored as YAML files, used to create a list with `create_sp_list(definition = )`, and compared or synced with an existing list using [compare_sp_list\(\)](#) and [sync_sp_list\(\)](#).

- [read_sp_list_yaml\(\)](#) reads and validates a YAML file.
- [write_sp_list_yaml\(\)](#) writes a definition or an existing list to a YAML file.
- [get_sp_list_definition\(\)](#) creates a definition from an existing list.
- [as_sp_list_definition\(\)](#) validates a list with the same structure as a YAML file.

Usage

```
read_sp_list_yaml(path, call = caller_env())
```

```
as_sp_list_definition(x, ..., display_name = NULL, call = caller_env())
```

```
get_sp_list_definition(
  sp_list = NULL,
  ...,
  keep_defaults = FALSE,
  read_only = "stable",
  include_views = FALSE,
  include_folders = FALSE,
  call = caller_env()
)
```

```
write_sp_list_yaml(
  x,
  path,
  ...,
  merge = TRUE,
  keep_defaults = FALSE,
  read_only = "stable",
  include_views = FALSE,
  include_folders = FALSE,
  doc_start = NULL,
  call = caller_env()
)
```

Arguments

path	Path to a YAML file.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.
x	For <code>write_sp_list_yaml()</code> , a <code>sp_list_definition</code> object or a <code>ms_list</code> object. <code>keep_defaults</code> , <code>read_only</code> , <code>include_views</code> , and <code>include_folders</code> are only used for a <code>ms_list</code> object. For <code>as_sp_list_definition()</code> , a named list with the same structure as a YAML file or an unnamed list of column definitions.
...	Must be empty.
display_name	List display name. Used by <code>as_sp_list_definition()</code> if <code>x</code> is an unnamed list of column definitions.
sp_list	A <code>ms_list</code> object. If supplied, <code>list_name</code> , <code>list_id</code> , <code>site_url</code> , <code>site</code> , <code>drive_name</code> , <code>drive_id</code> , <code>drive</code> , and any additional parameters passed to ... are all ignored.
keep_defaults	If FALSE (default), drop properties with the default value returned by the Graph API (e.g. <code>required: false</code>) to keep definitions short. Properties that aren't included aren't compared by <code>compare_sp_list()</code> .
read_only	Read-only list properties to include as a reference to the existing list. One of "stable" (default) for properties that don't change after a list is created (<code>id</code> , <code>name</code> , <code>webUrl</code> , <code>createdDateTime</code> , <code>parentReference</code> , <code>sharepointIds</code> , and <code>system</code>), "all", "none", or a character vector of property names (e.g. <code>c("id", "createdBy")</code>). Column ids (and view ids) are included if "id" is included. <code>createdBy</code> and <code>lastModifiedBy</code> include the name and email of a person.
include_views	If TRUE, include the list views (excluding hidden and personal views). Defaults to FALSE. Views are read with the SharePoint REST API, which requires a delegated (user) login. See <code>list_sp_list_views()</code> .
include_folders	If TRUE, include the folders of a document library. Defaults to FALSE. Folders are listed recursively, which takes one request for each folder. Only the deepest folders of each path are included.
merge	If TRUE (default) and <code>path</code> exists, keep the comment header and column order from the existing file. custom metadata for the list and each column comes from <code>x</code> if it has any and is otherwise kept from the existing file (a list never has custom metadata, so it's always kept when <code>x</code> is a <code>ms_list</code> object). Read-only list properties and column ids always come from <code>x</code> . Views come from <code>x</code> if it has views (e.g. with <code>include_views = TRUE</code>) and are otherwise kept from the existing file. Folders are kept from the existing file the same way. Columns that are only in the existing file are dropped (unless the Graph API doesn't return their column type). Comments after the header are always lost.
doc_start	If TRUE, write a document start marker (<code>---</code>) after the comment header (or at the start of a file with no header). If FALSE, don't. If NULL (default), write the marker only if the existing file has one (with <code>merge = TRUE</code>).

Details

YAML format

```
# Comments before the first key are kept by write_sp_list_yaml()
---
format_version: 1
displayName: Capital Project
description: Capital projects and their status.
list:
  template: genericList
# Read-only properties of an existing list (optional)
id: 5a8b3c4d-0000-0000-0000-000000000000
webUrl: https://example.sharepoint.com/sites/Planning/Lists/CapitalProject
parentReference:
  siteId: example.sharepoint.com,1111,2222
custom:
  owner: Planning
columns:
  - name: Title
    displayName: Project Name
    required: true
    text: {}
  - name: ProjectID
    displayName: Project ID
    description: Project reference ID
    required: true
    text: {}
  - name: Notes
    displayName: Project Notes
    text:
      allowMultipleLines: true
    custom:
      form_order: 2
  - name: Status
    choice:
      choices:
        - Active
        - Closed
    displayAs: radioButtons
```

Top-level keys use the Graph **list** property names:

- **displayName** (required): the list display name. SharePoint sets the read-only list name (used in the list URL) from the display name when a list is created. Changing the display name later doesn't change the URL.
- **columns** (required): a sequence of column definitions.
- **description** (optional): the list description.

- `list` (optional): list settings using the `listInfo` property names template (e.g. `genericList`, the default, or `documentLibrary`), `hidden`, and `contentTypesEnabled`. The template can't be changed after a list is created.
- `format_version` (optional): the format version. Only 1 is supported.
- `custom` (optional): a mapping that isn't validated (see below).

Comments and blank lines before the first key are a header that `write_sp_list_yaml()` keeps when it updates a file. An optional document start marker (`---`) after the header marks where the header ends. Other comments (including comments between `---` and the first key) are lost when a file is rewritten, so use `custom` for notes that need to be kept. A file can only have one YAML document.

Read-only list properties can also be included as a reference to an existing list: `id`, `name`, `webUrl`, `createdDateTime`, `createdBy`, `lastModifiedDateTime`, `lastModifiedBy`, `eTag`, `parentReference`, `sharepointIds`, and `system`. These properties are never sent to SharePoint. `compare_sp_list()` and `sync_sp_list()` use `id` and `parentReference.siteId` to get the list if `sp_list` isn't supplied, and error if a supplied list has a different `id`. `create_sp_list()` ignores them.

Columns can also include an `id` as a reference to an existing column. Columns are always matched by name.

Column keys use the Graph `columnDefinition` property names:

- `name` (required): the internal column name. The name can't be changed after a column is created and should avoid names that look like spreadsheet cell references (e.g. `V4`). For a list (but not a document library), the name of a new column can't be longer than 32 characters, counting each space or special character as 7 (e.g. a space is stored as `_x0020_`). SharePoint cuts longer names without an error, so `create_sp_list()` and `create_sp_list_column()` error instead. Display names can be longer: the column settings page allows up to 255 characters.
- `displayName`, `description`, `required`, `enforceUniqueValues`, `hidden`, `indexed`, `readOnly`, `defaultValue` (with `value` or `formula`), and `validation` (see `column_validation()`).
- Exactly one column type key holding the properties for that type. Use an empty mapping (`{}`) for a type with no properties.

A `Title` column updates the default title column of a `genericList` list instead of adding a new column.

Column type keys and properties:

Type key	Properties
<code>text</code>	<code>allowMultipleLines</code> , <code>appendChangesToExistingText</code> , <code>linesForEditing</code> , <code>maxLength</code>
<code>choice</code>	<code>allowTextEntry</code> , <code>choices</code> , <code>displayAs</code>
<code>number</code>	<code>decimalPlaces</code> , <code>displayAs</code> , <code>maximum</code> , <code>minimum</code>
<code>dateTime</code>	<code>displayAs</code> , <code>format</code>
<code>currency</code>	<code>locale</code>
<code>calculated</code>	<code>formula</code> , <code>outputType</code> , <code>format</code>
<code>lookup</code>	<code>listId</code> , <code>columnName</code> , <code>allowMultipleValues</code> , <code>allowUnlimitedLength</code> , <code>primaryLookup</code>
<code>personOrGroup</code>	<code>allowMultipleSelection</code> , <code>chooseFromType</code> , <code>displayAs</code>
<code>hyperlinkOrPicture</code>	<code>isPicture</code>
<code>term</code>	<code>allowMultipleValues</code> , <code>showFullyQualifiedNames</code>
<code>boolean</code> , <code>geolocation</code> , <code>thumbnail</code>	<code>none</code>

Validation rules:

- Unknown keys are errors, with a hint for common alternatives (e.g. label for displayName). Read-only properties (e.g. id) are errors.
- Values must match the type and allowed values in the Graph API documentation.
- A calculated column requires formula and outputType. format is only allowed when outputType is dateTime.
- A lookup column requires listId and columnName. The listId is specific to a site.
- Column names must be unique. Duplicate display names are a warning since formulas reference columns by display name.

Keys that aren't columnDefinition properties go under custom, either for the list or for each column. sharepointtr doesn't validate custom and doesn't send it to SharePoint, so it can hold metadata used by other applications (e.g. form order or app-enforced choices).

Properties that aren't included in a column definition aren't compared or changed by `sync_sp_list()`. Removing a property from a file doesn't reset it to the default value.

List views

An optional views key holds a sequence of list views using the SharePoint REST API SP.View property names (see `list_sp_list_views()`):

views:

- Title: Active Projects

DefaultView: true

ViewFields:

- LinkTitle
- ProjectStatus
- Budget

ViewQuery: <Where><Eq><FieldRef Name="ProjectStatus"/><Value Type="Choice">Active</Value></Eq></Where>

RowLimit: 50

CustomFormatter:

additionalRowClass: sp-field-severity--good

- Title is required and must be unique. Only one view can set DefaultView: true.
- Other properties are ViewFields (internal column names), ViewQuery (a CAML query), RowLimit, Paged, Scope, Hidden, MobileView, MobileDefaultView, and CustomFormatter (view formatting as a YAML mapping or a JSON string). Id and ServerRelativeUrl can be included as a reference to an existing view.
- A warning is given if a view shows fields that aren't columns in the definition or built-in fields (e.g. LinkTitle, ID, or Modified).

`create_sp_list(definition = )` creates views after creating the columns. A view titled All Items updates the default view of a new genericList list. `compare_sp_list()` and `sync_sp_list()` compare and update views only if the definition has a views element. Views are only written by `write_sp_list_yaml()` if `include_views = TRUE`.

Document libraries

A definition with `list.template: documentLibrary` describes a document library. Use `create_sp_drive()` (or `create_sp_list()`) to create a library from a definition. The built-in library columns (e.g. `FileLeafRef` for the file name or `_ExtendedDescription`) are left out of a definition from an existing library and aren't compared unless a definition includes them. A view titled `All Documents` updates the default view of a new library.

An optional `folders` key holds a sequence of folder paths relative to the library root. Folders are only allowed for a document library.

```
list:
  template: documentLibrary
folders:
  - Contracts
  - Reports/2026
```

- Separate folders with `/`. Parent folders are created as needed, so `Reports/2026` also creates `Reports`.
- Paths must be unique. A folder name can't be empty, start or end with a space, end with a period, or include any of `" * : < > ? \ |`.
- `compare_sp_list()` and `sync_sp_list()` add folders that are missing from the library. Folders are never deleted or renamed.
- Folders are only written by `write_sp_list_yaml()` if `include_folders = TRUE`. Only the deepest folders of each path are written.

Value

A `sp_list_definition` object: a list with `format_version`, `displayName`, `description`, `list`, any read-only list properties, `custom`, `columns`, `views`, and `folders` elements. `write_sp_list_yaml()` invisibly returns the definition that was written.

See Also

`sp_list_definition_table()` to convert a definition to a data frame.

Examples

```
path <- system.file("extdata", "example-list.yaml", package = "sharepointR")

definition <- read_sp_list_yaml(path)

definition

sp_list_definition_table(definition)
```

`sp_list_definition_table`*Convert a list definition to a data frame*

Description

[Experimental]

`sp_list_definition_table()` converts a `sp_list_definition` object to a data frame with one row per column. The `as.data.frame()` method for `sp_list_definition` objects uses the same function.

The data frame has `list_name`, `name`, `displayName`, `description`, and `type` columns followed by any other column-level properties, the properties for each column type, and the custom metadata for each column. Properties for column types aren't prefixed since each column has a single column type. Properties with more than one value (e.g. choices) are list columns and all other columns are simplified to atomic vectors.

Usage

```
sp_list_definition_table(x, custom = TRUE, custom_prefix = NULL)
```

```
## S3 method for class 'sp_list_definition'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  ...,
  custom = TRUE,
  custom_prefix = NULL
)
```

Arguments

<code>x</code>	A <code>sp_list_definition</code> object or a path to a YAML file.
<code>custom</code>	If TRUE (default), include the custom metadata for each column.
<code>custom_prefix</code>	Optional prefix added to the names of custom metadata columns. A custom key with the same name as a <code>columnDefinition</code> property is an error unless a prefix is supplied.
<code>row.names</code> , <code>optional</code> , ...	Ignored.

Value

A data frame with one row per column.

Examples

```
path <- system.file("extdata", "example-list.yaml", package = "sharepointR")

sp_list_definition_table(path)
```

sp_list_item	<i>Get, list, update, and delete SharePoint list items</i>
--------------	--

Description

`list_sp_list_items()` lists `sp_list` items. This function uses a modified version of the `list_items` method for `Microsoft365R::ms_list` objects that adds the `order_by` and `order_dir` arguments.

Usage

```
list_sp_list_items(
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  ...,
  filter = NULL,
  select = NULL,
  order_by = NULL,
  order_dir = "desc",
  all_metadata = FALSE,
  as_data_frame = TRUE,
  col_formatting = c("asis", "date"),
  display_nm = c("drop", "label", "replace"),
  select_type = c("asis", "editable", "external"),
  n = NULL,
  tz = Sys.timezone(),
  name_repair = "unique",
  pagesize = 5000,
  site_url = NULL,
  site = NULL,
  drive_name = NULL,
  drive_id = NULL,
  drive = NULL,
  call = caller_env()
)

get_sp_list_items(
  list_name = NULL,
  list_id = NULL,
  sp_list = NULL,
  ...,
```

```

    filter = NULL,
    select = NULL,
    order_by = NULL,
    order_dir = "desc",
    all_metadata = FALSE,
    as_data_frame = TRUE,
    col_formatting = c("asis", "date"),
    display_nm = c("drop", "label", "replace"),
    n = NULL,
    tz = Sys.timezone(),
    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

<code>list_name, list_id</code>	SharePoint List name or ID string.
<code>sp_list</code>	A <code>ms_list</code> object. If supplied, <code>list_name</code> , <code>list_id</code> , <code>site_url</code> , <code>site</code> , <code>drive_name</code> , <code>drive_id</code> , <code>drive</code> , and any additional parameters passed to <code>...</code> are all ignored.
<code>...</code>	Arguments passed on to get_sp_list
<code>list_name, list_id</code>	SharePoint List name or ID string.
<code>metadata</code>	If <code>TRUE</code> , get_sp_list() applies the <code>get_column_info</code> method to the returned SharePoint list and returns a data frame with column metadata for the list.
<code>keep</code>	One of "all" (default), "editable", "external" (non-internal fields). Argument determines if the returned list metadata includes read only columns or hidden columns.

	<code>sync_fields</code> If TRUE, use the <code>sync_fields</code> method to sync the fields of the local <code>ms_list</code> object with the fields of the SharePoint List source before retrieving list metadata. <code>drive_name, drive_id</code> SharePoint Drive name or ID passed to <code>get_drive</code> method for SharePoint site object.
<code>filter</code>	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.
<code>select</code>	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.
<code>order_by</code>	Optional. Field name to order by.
<code>order_dir</code>	Direction to order results if <code>order_by</code> is provided.
<code>all_metadata</code>	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 <code>fields</code> . This is always set to FALSE if <code>n</code> = NULL or <code>as_data_frame</code> = FALSE. The " <code>@odata.etag</code> " value the Graph API returns with the fields of each item is only included if <code>all_metadata</code> = TRUE.
<code>as_data_frame</code>	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 <code>n</code> rows or all lists available for the SharePoint site or drive. Defaults to FALSE. Ignored if <code>metadata</code> = TRUE as list metadata is always returned as a data frame.
<code>col_formatting</code>	"asis" (default) or "date". If "date", use the list column metadata and convert date columns to Date class and datetime columns to POSIXct class vectors (latter is not yet tested).
<code>display_nm</code>	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 <code>name_repair</code> argument is required since there is no requirement on SharePoint for lists to use unique display names and invalid data frames can result.
<code>select_type</code>	Type of columns to select. Ignored if <code>select</code> is supplied. "asis" (default) returns all available columns. "editable" returns ID and all non-read-only columns and "external" returns ID and all non-internal columns.
<code>n</code>	Maximum number of lists, plans, tasks, or other items to return. Defaults to NULL which sets <code>n</code> to Inf.
<code>tz</code>	Time zone to use in reformatting date/time columns if <code>col_formatting</code> = "date". Defaults to <code>Sys.timezone()</code> .
<code>name_repair</code>	Passed to repair argument of vctrs::vec_as_names()
<code>pagesize</code>	Number of list items to return. Reduce from default of 5000 if experiencing timeouts.
<code>site_url</code>	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.
<code>site</code>	A <code>ms_site</code> object. If <code>site</code> is supplied, <code>site_url</code> , <code>site_name</code> , and <code>site_id</code> 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.

Value

For [list_sp_list_items\(\)](#) and [get_sp_list_items\(\)](#), a data frame of list items if `as_data_frame = TRUE` (default), or a list of item fields otherwise. For [get_sp_list_item\(\)](#), a `ms_list_item` object.

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	<i>List SharePoint planner buckets</i>
----------------	--

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. Users seeking to access a SharePoint subsite must provide the `site_id` instead of a `site_url` or `site_name` value. You can see available subsite ID values by using the `list_subsites()` method for `Microsoft365R::ms_site` objects.

Usage

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

cache_sp_site(
  ...,
  site = NULL,
  cache_file = NULL,
  cache_dir = NULL,
  overwrite = FALSE,
  call = caller_env()
)
```

Arguments

<code>site_url</code>	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.
<code>site_name, site_id</code>	Site name or ID of the SharePoint site as an alternative to the SharePoint site URL. Exactly one of <code>site_url</code> , <code>site_name</code> , and <code>site_id</code> must be supplied.
<code>...</code>	Arguments passed on to <code>Microsoft365R::get_sharepoint_site</code>
<code>app</code>	A custom app registration ID to use for authentication. See below.
<code>scopes</code>	The Microsoft Graph scopes (permissions) to obtain. It should never be necessary to change these.

	token An AAD OAuth token object, of class <code>AzureAuth::AzureToken</code>. If supplied, the <code>tenant</code>, <code>app</code>, <code>scopes</code> and ... arguments will be ignored. See "Authenticating with a token" below. tenant For <code>get_business_onedrive</code>, <code>get_sharepoint_site</code> and <code>get_team</code>, the name of your Azure Active Directory (AAD) tenant. If not supplied, use the value of the <code>CLIMICROSOFT365_TENANT</code> environment variable, or "common" if that is unset. team_name, team_id For <code>get_team</code>, either the name or ID of the team to retrieve. Supply exactly one of these. shared_mbox_id, shared_mbox_name, shared_mbox_email For <code>get_business_outlook</code>, an ID/principal name/email address. Supply exactly one of these to retrieve a shared mailbox. If all are NULL (the default), retrieve your own mailbox. chat_id For <code>get_chat</code>, the ID of a group, one-on-one or meeting chat in Teams.
cache	If TRUE, cache site to a file using <code>cache_sp_site()</code> .
refresh	If TRUE, get a new site even if the existing site is cached as a local option. If FALSE, use the cached <code>ms_site</code> object.
overwrite	If TRUE, replace the existing cached object named by <code>cache_file</code> with the new object. If FALSE, error if a cached file with the same <code>cache_file</code> name already exists.
cache_file	File name for cached drive or site. Default NULL.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.
site	A <code>ms_site</code> object. If <code>site</code> is supplied, <code>site_url</code> , <code>site_name</code> , and <code>site_id</code> are ignored.
cache_dir	Cache directory. By default, uses an option named "sharepoint.cache_dir". If "sharepoint.cache_dir" is not set, the cache directory is set to <code>rappdirs::user_cache_dir("sharepoint")</code> .

Value

For `get_sp_site()`, a `ms_site` object. For `cache_sp_site()`, invisibly returns NULL.

See Also

[Microsoft365R::ms_site](#)

sp_tasks	<i>Get a SharePoint task or list tasks from a planner</i>
----------	---

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 <code>ms_plan</code> 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 <code>ms_plan_task</code> objects. If FALSE (default get_sp_task()), return a <code>ms_plan_task</code> object or list of <code>ms_plan_task</code> objects.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . 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](#)

update_sp_drive	<i>Update or delete a SharePoint document library</i>
-----------------	---

Description**[Experimental]**

`update_sp_drive()` updates the display name or description of a document library. Changing the display name changes the drive name but not the library URL.

`delete_sp_drive()` deletes a document library and all of its files and folders. It requires confirmation by default and can't delete the default document library of a site (usually named "Documents").

Both functions get the library's list from the drive and use `update_sp_list()` or `delete_sp_list()`.

Usage

```
update_sp_drive(
  drive_name = NULL,
  drive_id = NULL,
  drive_url = NULL,
  drive = NULL,
  display_name = NULL,
  description = NULL,
  ...,
  properties = FALSE,
  site_url = NULL,
  site = NULL,
  call = caller_env()
)
```

```
delete_sp_drive(
  drive_name = NULL,
  drive_id = NULL,
  drive_url = NULL,
  drive = NULL,
  confirm = TRUE,
  ...,
  site_url = NULL,
  site = NULL,
  call = caller_env()
)
```

Arguments

drive_name, drive_id	SharePoint Drive name or ID passed to <code>get_drive</code> 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.
drive	A <code>ms_drive</code> object. If drive is supplied, drive_name, drive_id, and drive_url are ignored.
display_name	New display name for the library. Optional.
description	New description for the library. Optional.
...	Additional parameters passed to <code>get_sp_drive()</code> if drive is NULL.
properties	For <code>update_sp_drive()</code> , if TRUE, return the drive properties instead of the <code>ms_drive</code> object. Defaults to FALSE.
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 <code>ms_site</code> 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. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.
confirm	If TRUE (default), confirm deletion of the library before proceeding.

Value

For `update_sp_drive()`, the updated `ms_drive` object or its properties if `properties = TRUE`. For `delete_sp_drive()`, invisibly returns NULL.

See Also

`create_sp_drive()`

Examples

```
## Not run:
drive <- update_sp_drive(
  "Project Files",
  description = "Files for active projects",
  site_url = "<SharePoint site url>"
)

delete_sp_drive(drive = drive)

## End(Not run)
```

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. `upload_sp_items()` uploads multiple files or folders. If `{carrier}` and `{mirai}` are installed and `mirai::daemons()` are set, `upload_sp_items()` uploads files in parallel (folders are always uploaded one at a time).

Usage

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

Arguments

<code>file</code>	Path for file or directory to upload. Optional if <code>src</code> is supplied.
-------------------	---

dest	Destination on SharePoint for file to upload. Must be a <code>ms_drive_item</code> for a folder, a SharePoint folder URL, or a drive path. If a <code>ms_drive_item</code> is provided and <code>drive_name</code> is <code>NULL</code> , the item URL is used to determine the drive. If <code>NULL</code> , the top level of the provided drive is used as the destination. If file is a directory, dest must be an <i>existing</i> directory. If dest is a SharePoint folder or document library URL and <code>drive_name</code> , <code>drive_id</code> , and <code>drive</code> aren't supplied, the drive and folder are found with the Microsoft Graph shares API (see get_sp_item()). If the request fails, the drive and folder path are parsed from the URL instead. For upload_sp_items() , dest can also be a character vector with one destination for each file. Destinations can be in different drives.
...	Additional parameters passed to get_sp_site() or Microsoft365R::get_sharepoint_site() .
src	Data source path passed to <code>upload_folder</code> or <code>upload_file</code> method. Defaults to <code>NULL</code> and set to use file value by default.
overwrite	If <code>FALSE</code> (default), error if an item with the name specified in file or src already exists at the specified destination. If <code>TRUE</code> , overwrite any existing items with the same name. The latter is the default for the <code>upload_file</code> method.
drive_name, drive_id	SharePoint drive name or ID.
drive	A <code>ms_drive</code> object. If drive is supplied, <code>drive_name</code> , <code>site_url</code> , and any additional parameters passed to ... are ignored.
blocksize, recursive, parallel	Additional parameters passed to <code>upload_folder</code> or <code>upload_file</code> method for <code>ms_drive</code> objects.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of abort() for more information.

Value

Invisibly returns the destination path (`dest`) for the uploaded file or directory.

write_sharepoint	<i>Write an object to file and upload file to SharePoint</i>
------------------	--

Description

[write_sharepoint\(\)](#) uses [upload_sp_item\(\)](#) to upload an object created using:

- `sf::write_sf()` if `x` is an `sf` object (shapefiles are not supported)
- `readr::write_csv()` if `x` is a data.frame and file does not have a `xlsx` extension
- `openxlsx2::write_xlsx()` if `x` is a data.frame and file has 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 = NULL,
  ...,
  .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). A shapefile writes more than one file, so file can't use a "shp" extension for sf objects. Use a single file format like GeoPackage ("gpkg") or GeoJSON ("geojson") instead.
dest	Destination on SharePoint for file to upload. Must be a ms_drive_item for a folder, a SharePoint folder URL, or a drive path. If a ms_drive_item is provided and drive_name is NULL, the item URL is used to determine the drive. If NULL, the top level of the provided drive is used as the destination. If file is a directory, dest must be an <i>existing</i> directory. If dest is a SharePoint folder or document library URL and drive_name, drive_id, and drive aren't supplied, the drive and folder are found with the Microsoft Graph shares API (see get_sp_item()). If the request fails, the drive and folder path are parsed from the URL instead. For upload_sp_items() , dest can also be a character vector with one destination for each file. Destinations can be in different drives.
...	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 <code>ms_drive</code> 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 <code>ms_site</code> 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

- * **drives**
 - create_sp_drive, 17
 - sp_drive, 63
 - update_sp_drive, 89
- * **files**
 - sp_dir_create, 58
 - sp_dir_info, 60
 - sp_item_copy, 64
- * **groups**
 - get_sp_group, 43
- * **items**
 - delete_sp_item, 36
 - download_sp_item, 39
 - get_sp_item, 44
 - list_sp_item_versions, 49
 - sp_item_properties, 68
- * **lists**
 - as_column_definition, 3
 - column_validation, 4
 - compare_sp_list, 5
 - copy_column_definition_list, 9
 - create_column_definition, 10
 - create_column_definition_list, 16
 - create_sp_list, 19
 - create_sp_list_items, 25
 - create_sp_list_lookup_column, 29
 - data_as_column_definition_list, 34
 - delete_sp_list_item, 37
 - list_sp_list_views, 50
 - list_sp_site_user_info, 56
 - sp_list, 70
 - sp_list_definition, 73
 - sp_list_definition_table, 79
 - sp_list_item, 80
- * **pages**
 - list_sp_pages, 54
- * **plans**
 - sp_plan, 83
 - sp_plan_bucket, 84
 - sp_tasks, 87
- * **sites**
 - sp_site, 86
- abort(), 3, 6, 10, 18, 21, 25, 28, 33, 37, 39, 40, 43, 44, 46, 48–50, 53, 55, 56, 59, 62, 64, 67, 69, 72, 74, 83–85, 87, 88, 90, 92, 94
- as.data.frame.sp_list_definition (sp_list_definition_table), 79
- as_column_definition, 3
- as_column_definition(), 3, 10
- as_sp_list_definition (sp_list_definition), 73
- as_sp_list_definition(), 73, 74
- AzureR_config, 4
- AzureR_config_delete (AzureR_config), 4
- AzureR_config_delete(), 4
- AzureR_config_ls (AzureR_config), 4
- AzureR_config_ls(), 4
- cache_sp_drive (sp_drive), 63
- cache_sp_drive(), 46, 59, 61, 63, 64, 66, 71
- cache_sp_site (sp_site), 86
- cache_sp_site(), 49, 55, 86, 87
- column_validation, 4
- column_validation(), 4, 13, 23, 32, 76
- compare_sp_list, 5
- compare_sp_list(), 5, 6, 8, 17, 73, 74, 76–78
- copy_column_definition_list, 9
- copy_column_definition_list(), 10
- create_boolean_column (create_column_definition), 10
- create_calculated_column (create_column_definition), 10
- create_calculated_column(), 3
- create_choice_column (create_column_definition), 10
- create_column_definition, 10, 23

- create_column_definition(), [3](#), [6](#), [10](#), [13](#),
[16](#), [20](#), [23](#), [25](#), [32](#)
- create_column_definition_list, [16](#)
- create_column_definition_list(), [6](#), [15](#),
[16](#), [20](#)
- create_currency_column
(create_column_definition), [10](#)
- create_datetime_column
(create_column_definition), [10](#)
- create_geolocation_column
(create_column_definition), [10](#)
- create_geolocation_column(), [15](#)
- create_group_column
(create_column_definition), [10](#)
- create_group_column(), [14](#), [15](#), [24](#), [33](#)
- create_hyperlink_column
(create_column_definition), [10](#)
- create_hyperlink_column(), [15](#), [24](#), [33](#)
- create_lookup_column, [31](#)
- create_lookup_column
(create_column_definition), [10](#)
- create_lookup_column(), [14](#), [24](#), [33](#), [34](#)
- create_number_column
(create_column_definition), [10](#)
- create_person_column
(create_column_definition), [10](#)
- create_person_column(), [14](#), [15](#), [24](#), [33](#), [34](#)
- create_picture_column
(create_column_definition), [10](#)
- create_picture_column(), [15](#), [24](#), [33](#)
- create_sp_drive, [17](#)
- create_sp_drive(), [17](#), [19](#), [78](#), [90](#)
- create_sp_list, [19](#)
- create_sp_list(), [10](#), [15–19](#), [21](#), [35](#), [73](#), [76](#),
[78](#)
- create_sp_list_column, [22](#)
- create_sp_list_column(), [5](#), [15](#), [22](#), [25](#), [34](#),
[76](#)
- create_sp_list_items, [25](#)
- create_sp_list_items(), [27–29](#), [34](#), [39](#)
- create_sp_list_lookup_column, [29](#)
- create_sp_list_lookup_column(), [31](#), [34](#)
- create_sp_list_person_column
(create_sp_list_lookup_column),
[29](#)
- create_sp_list_person_column(), [34](#)
- create_sp_list_view
(list_sp_list_views), [50](#)
- create_sp_list_view(), [51–54](#)
- create_term_column
(create_column_definition), [10](#)
- create_term_column(), [14](#), [15](#), [24](#), [33](#)
- create_text_column
(create_column_definition), [10](#)
- create_thumbnail_column
(create_column_definition), [10](#)
- create_thumbnail_column(), [15](#)
- data_as_column_definition_list, [34](#)
- data_as_column_definition_list(), [28](#),
[34](#)
- delete_sp_drive (update_sp_drive), [89](#)
- delete_sp_drive(), [18](#), [89](#), [90](#)
- delete_sp_item, [36](#)
- delete_sp_item(), [36](#)
- delete_sp_list (create_sp_list), [19](#)
- delete_sp_list(), [19](#), [21](#), [73](#), [89](#)
- delete_sp_list_column
(create_sp_list_column), [22](#)
- delete_sp_list_column(), [22](#), [25](#)
- delete_sp_list_item, [37](#)
- delete_sp_list_item(), [37–39](#)
- delete_sp_list_items
(delete_sp_list_item), [37](#)
- delete_sp_list_items(), [37–39](#)
- delete_sp_list_view
(list_sp_list_views), [50](#)
- delete_sp_list_view(), [51](#), [54](#)
- download_sp_file (download_sp_item), [39](#)
- download_sp_file(), [41](#)
- download_sp_item, [39](#)
- download_sp_item(), [39](#), [41](#)
- download_sp_list, [42](#)
- download_sp_list(), [42](#)
- fmt_sp_list_lookup_items
(create_sp_list_lookup_column),
[29](#)
- fmt_sp_list_lookup_items(), [31](#), [33](#), [34](#),
[56](#)
- fs::as_fs_bytes(), [60](#)
- fs::dir_copy(), [65](#), [66](#)
- fs::dir_create(), [40](#), [58](#)
- fs::dir_info(), [60](#)
- fs::dir_ls(), [4](#)
- fs::file_copy(), [65](#)
- fs::file_delete(), [4](#)

fs::file_move(), 65

 get_column_default(), 13, 23, 31
 get_sp_drive, 45, 59, 61, 66, 71
 get_sp_drive(sp_drive), 63
 get_sp_drive(), 18, 45, 63, 64, 69, 90
 get_sp_group, 43
 get_sp_group(), 43, 44, 84, 85, 88
 get_sp_item, 36, 40, 44, 50
 get_sp_item(), 45, 61, 70, 92, 93
 get_sp_item_properties
 (sp_item_properties), 68
 get_sp_item_properties(), 45, 46, 68, 69
 get_sp_list, 42, 47, 81
 get_sp_list(sp_list), 70
 get_sp_list(), 6, 9, 38, 42, 52, 56, 68, 70,
 72, 81, 82
 get_sp_list_column, 47
 get_sp_list_column(), 47
 get_sp_list_definition
 (sp_list_definition), 73
 get_sp_list_definition(), 73
 get_sp_list_item(sp_list_item), 80
 get_sp_list_item(), 68, 70, 83
 get_sp_list_items(sp_list_item), 80
 get_sp_list_items(), 83
 get_sp_list_metadata, 9
 get_sp_list_metadata(sp_list), 70
 get_sp_list_metadata(), 9
 get_sp_list_view(list_sp_list_views),
 50
 get_sp_list_view(), 50, 53, 54
 get_sp_page(list_sp_pages), 54
 get_sp_page(), 54, 55
 get_sp_plan(sp_plan), 83
 get_sp_plan(), 83, 84
 get_sp_site, 49, 55, 63
 get_sp_site(sp_site), 86
 get_sp_site(), 18, 20, 27, 44, 55, 56, 86, 87,
 92
 get_sp_site_group(get_sp_group), 43
 get_sp_task(sp_tasks), 87
 get_sp_task(), 87, 88
 glue::glue(), 3, 14, 24, 32
 grep(), 4

 import_sp_list_items
 (create_sp_list_items), 25

 list_sp_drives, 48
 list_sp_drives(), 48
 list_sp_group_members(get_sp_group), 43
 list_sp_group_members(), 43, 44
 list_sp_item_versions, 49
 list_sp_list_items(sp_list_item), 80
 list_sp_list_items(), 39, 56, 57, 68, 80, 83
 list_sp_list_views, 50
 list_sp_list_views(), 21, 50, 52, 54, 74, 77
 list_sp_lists(sp_list), 70
 list_sp_lists(), 9, 42, 56, 70, 72, 82
 list_sp_pages, 54
 list_sp_pages(), 54, 55
 list_sp_plan_buckets(sp_plan_bucket),
 84
 list_sp_plan_buckets(), 84, 85
 list_sp_plans(sp_plan), 83
 list_sp_plans(), 83, 84
 list_sp_site_user_info, 56
 list_sp_site_user_info(), 34, 56
 list_sp_tasks(sp_tasks), 87
 list_sp_tasks(), 87–89

 Microsoft365R::get_sharepoint_site, 86
 Microsoft365R::get_sharepoint_site(),
 20, 27, 44, 86, 92
 Microsoft365R::ms_drive, 64
 Microsoft365R::ms_drive_item, 46, 67
 Microsoft365R::ms_list, 72
 Microsoft365R::ms_list_item, 73
 Microsoft365R::ms_plan, 84
 Microsoft365R::ms_plan_task, 85, 89
 Microsoft365R::ms_site, 49, 87
 mirai::daemons(), 41, 91

 officer::read_docx(), 57
 officer::read_pptx(), 57
 openxlsx2::write_xlsx(), 92

 print(), 92
 progressBars, 28, 34, 39
 purrr::in_parallel(), 28, 39

 rappdirs::user_config_dir(), 4
 read_sharepoint, 57
 read_sharepoint(), 57
 read_sp_list_yaml(sp_list_definition),
 73
 read_sp_list_yaml(), 3, 5, 6, 17, 18, 21, 25,
 73

readr::read_delim(), 57
 readr::read_lines(), 57
 readr::read_rds(), 57
 readr::write_csv(), 92, 93
 readr::write_rds(), 92, 93
 readxl::read_excel(), 57
 regular expression, 14, 24, 33, 35

 sf::read_sf(), 57
 sf::st_as_text(), 27
 sf::write_sf(), 92
 sp_dir_copy(sp_item_copy), 64
 sp_dir_copy(), 65–67
 sp_dir_create, 58
 sp_dir_create(), 17, 58
 sp_dir_info, 60
 sp_dir_info(), 60, 62
 sp_dir_ls(sp_dir_info), 60
 sp_dir_ls(), 60, 62
 sp_dir_move(sp_item_copy), 64
 sp_dir_move(), 65, 66
 sp_drive, 63
 sp_file_download(download_sp_item), 39
 sp_item_copy, 64
 sp_item_copy(), 65, 67
 sp_item_download(download_sp_item), 39
 sp_item_move(sp_item_copy), 64
 sp_item_move(), 65, 68
 sp_item_properties, 68
 sp_list, 70
 sp_list_definition, 73
 sp_list_definition_table, 79
 sp_list_definition_table(), 78, 79
 sp_list_item, 80
 sp_plan, 83
 sp_plan_bucket, 84
 sp_site, 86
 sp_tasks, 87
 stringr::str_replace_na(), 14, 23, 32
 sync_sp_list(compare_sp_list), 5
 sync_sp_list(), 5, 7, 8, 17, 22, 73, 76–78

 tempdir(), 93

 update_sp_drive, 89
 update_sp_drive(), 18, 89, 90
 update_sp_item_properties
 (sp_item_properties), 68
 update_sp_item_properties(), 68–70

 update_sp_list(create_sp_list), 19
 update_sp_list(), 19, 21, 89
 update_sp_list_column
 (create_sp_list_column), 22
 update_sp_list_column(), 22, 25
 update_sp_list_item
 (create_sp_list_items), 25
 update_sp_list_item(), 27, 68
 update_sp_list_items
 (create_sp_list_items), 25
 update_sp_list_items(), 27, 29, 34
 update_sp_list_lookup_items
 (create_sp_list_lookup_column),
 29
 update_sp_list_lookup_items(), 29, 31
 update_sp_list_person_items
 (create_sp_list_lookup_column),
 29
 update_sp_list_person_items(), 29, 31,
 56
 update_sp_list_view
 (list_sp_list_views), 50
 update_sp_list_view(), 51–54
 upload_sp_item, 91
 upload_sp_item(), 17, 91, 92
 upload_sp_items(upload_sp_item), 91
 upload_sp_items(), 91–93

 vctrs::vec_as_names(), 82

 write_sharepoint, 92
 write_sharepoint(), 92
 write_sp_list_yaml
 (sp_list_definition), 73
 write_sp_list_yaml(), 73, 74, 76–78