

Package ‘Microsoft365R’

February 26, 2021

Title Interface to the 'Microsoft 365' Suite of Cloud Services

Version 2.0.0

Description An interface to the 'Microsoft 365' (formerly known as 'Office 365') suite of cloud services, building on the framework supplied by the 'AzureGraph' package. Enables access from R to data stored in 'Teams', 'SharePoint Online' and 'OneDrive', including the ability to list drive folder contents, upload and download files, send messages, and retrieve data lists.

URL <https://github.com/Azure/Microsoft365R>
<https://github.com/Azure/AzureR>

BugReports <https://github.com/Azure/Microsoft365R/issues>

License MIT + file LICENSE

VignetteBuilder knitr

Depends R (>= 3.3)

Imports AzureAuth, AzureGraph (>= 1.2.1), utils, curl, httr, jsonlite, R6, vctrs, mime

Suggests openssl, knitr, rmarkdown, testthat

RoxygenNote 7.1.1

NeedsCompilation no

Author Hong Ooi [aut, cre],
Microsoft [cph]

Maintainer Hong Ooi <hongooi73@gmail.com>

Repository CRAN

Date/Publication 2021-02-26 13:10:02 UTC

R topics documented:

add_methods	2
get_personal_onedrive	4
ms_channel	7
ms_chat_message	9
ms_drive	11

ms_drive_item	13
ms_list	16
ms_list_item	18
ms_site	19
ms_team	21
personal_onedrive	22

Index	24
--------------	-----------

add_methods	<i>Microsoft 365 object accessor methods</i>
-------------	--

Description

Methods for the [AzureGraph::ms_graph](#), [AzureGraph::az_user](#) and [AzureGraph::az_group](#) classes.

Usage

```
## R6 method for class 'ms_graph'
get_drive(drive_id)
```

```
## R6 method for class 'az_user'
get_drive(drive_id = NULL)
```

```
## R6 method for class 'az_group'
get_drive(drive_id = NULL)
```

```
## R6 method for class 'ms_graph'
get_sharepoint_site(site_url = NULL, site_id = NULL)
```

```
## R6 method for class 'az_group'
get_sharepoint_site()
```

```
## R6 method for class 'ms_graph'
get_team(team_id = NULL)
```

```
## R6 method for class 'az_group'
get_team()
```

```
## R6 method for class 'az_user'
list_drives()
```

```
## R6 method for class 'az_group'
list_drives()
```

```
## R6 method for class 'az_user'
list_sharepoint_sites(filter = NULL)
```

```
## R6 method for class 'az_user'
list_teams(filter = NULL)
```

Arguments

- `drive_id`: For `get_drive`, the ID of the drive or shared document library. For the `az_user` and `az_group` methods, if this is `NULL` the default drive/document library is returned.
- `site_url,site_id`: For `ms_graph$get_sharepoint_site()`, the URL and ID of the site. Provide one or the other, but not both.
- `team_name,team_id`: For `az_user$get_team()`, the name and ID of the site. Provide one or the other, but not both. For `ms_graph$get_team`, you must provide the team ID.
- `filter`: For `az_user$list_sharepoint_sites()` and `az_user$list_teams()`, an optional OData expression to filter the list.

Details

`get_sharepoint_site` retrieves a SharePoint site object. The method for the top-level Graph client class requires that you provide either the site URL or ID. The method for the `az_group` class will retrieve the site associated with that group, if applicable.

`get_drive` retrieves a OneDrive or shared document library, and `list_drives` retrieves all such drives/libraries that the user or group has access to. Whether these are personal or business drives depends on the tenant that was specified in `AzureGraph::get_graph_login()/create_graph_login()`: if this was "consumers" or "9188040d-6c67-4c5b-b112-36a304b66dad" (the equivalent GUID), it will be the personal OneDrive. See the examples below.

`get_team` retrieves a team. The method for the Graph client class requires the team ID. The method for the `az_user` class requires either the team name or ID. The method for the `az_group` class retrieves the team associated with the group, if it exists.

Note that Teams, SharePoint and OneDrive for Business require a Microsoft 365 Business license, and are available for organisational tenants only.

Value

For `get_sharepoint_site`, an object of class `ms_site`.

For `get_drive`, an object of class `ms_drive`. For `list_drives`, a list of `ms_drive` objects.

For `get_team`, an object of class `ms_team`. For `list_teams`, a list of `ms_team` objects.

See Also

[ms_site](#), [ms_drive](#), [az_user](#), [az_group](#)

Examples

```
## Not run:

# 'consumers' tenant -> personal OneDrive for a user
gr <- AzureGraph::get_graph_login("consumers", app="myapp")
```

```

me <- gr$get_user()
me$get_drive()

# organisational tenant -> business OneDrive for a user
gr2 <- AzureGraph::get_graph_login("mycompany", app="myapp")
myuser <- gr2$get_user("username@mycompany.onmicrosoft.com")
myuser$get_drive()

# get a site/drive directly from a URL/ID
gr2$get_sharepoint_site("My site")
gr2$get_drive("drive-id")

# site/drive(s) for a group
grp <- gr2$get_group("group-id")
grp$get_sharepoint_site()
grp$list_drives()
grp$get_drive()

## End(Not run)

```

get_personal_onedrive *Login clients for Microsoft 365*

Description

Microsoft365R provides functions for logging into each Microsoft 365 service.

Usage

```

get_personal_onedrive(
  app = .microsoft365r_app_id,
  scopes = c("Files.ReadWrite.All", "User.Read"),
  ...
)

get_business_onedrive(
  tenant = Sys.getenv("CLIMICROSOFT365_TENANT", "common"),
  app = Sys.getenv("CLIMICROSOFT365_AADAPPID"),
  scopes = ".default",
  ...
)

get_sharepoint_site(
  site_name = NULL,
  site_url = NULL,
  site_id = NULL,
  tenant = Sys.getenv("CLIMICROSOFT365_TENANT", "common"),
  app = Sys.getenv("CLIMICROSOFT365_AADAPPID"),

```

```

    scopes = ".default",
    ...
)

list_sharepoint_sites(
    tenant = Sys.getenv("CLIMICROSOFT365_TENANT", "common"),
    app = Sys.getenv("CLIMICROSOFT365_AADAPPID"),
    scopes = ".default",
    ...
)

get_team(
    team_name = NULL,
    team_id = NULL,
    tenant = Sys.getenv("CLIMICROSOFT365_TENANT", "common"),
    app = Sys.getenv("CLIMICROSOFT365_AADAPPID"),
    scopes = ".default",
    ...
)

list_teams(
    tenant = Sys.getenv("CLIMICROSOFT365_TENANT", "common"),
    app = Sys.getenv("CLIMICROSOFT365_AADAPPID"),
    scopes = ".default",
    ...
)

```

Arguments

app	A custom app registration ID to use for authentication. See below.
scopes	The Microsoft Graph scopes (permissions) to obtain. It should never be necessary to change these.
...	Optional arguments that will ultimately be passed to AzureAuth::get_azure_token .
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.
site_name, site_url, site_id	For <code>get_sharepoint_site</code> , either the name, web URL or ID of the SharePoint site to retrieve. Supply exactly one of these.
team_name, team_id	For <code>get_team</code> , either the name or ID of the team to retrieve. Supply exactly one of these.

Details

These functions provide easy access to the various collaboration services that are part of Microsoft 365. On first use, they will call your web browser to authenticate with Azure Active Directory, in a

similar manner to other web apps. You will get a dialog box asking for permission to access your information. You only have to authenticate once; your credentials will be saved and reloaded in subsequent sessions.

When authenticating, you can pass optional arguments in `...` which will ultimately be received by `AzureAuth::get_azure_token`. In particular, if your machine doesn't have a web browser available to authenticate with (for example if you are in a remote RStudio Server session), pass `auth_type="device_code"` which is intended for such scenarios.

Value

For `get_personal_onedrive` and `get_business_onedrive`, an R6 object of class `ms_drive`.

For `get_sharepoint_site`, an R6 object of class `ms_site`; for `list_sharepoint_sites`, a list of such objects.

For `get_team`, an R6 object of class `ms_team`; for `list_teams`, a list of such objects.

Authenticating to Microsoft 365 Business services

Authenticating to Microsoft 365 Business services (Teams, SharePoint and OneDrive for Business) has some specific complexities.

The default "common" tenant for `get_team`, `get_business_onedrive` and `get_sharepoint_site` attempts to detect your actual tenant from your saved credentials in your browser. This may not always succeed, for example if you have a personal account that is also a guest account in a tenant. In this case, supply the actual tenant name, either in the `tenant` argument or in the `CLIMICROSOFT365_TENANT` environment variable. The latter allows sharing authentication details with the [CLI for Microsoft 365](#).

The default when authenticating to these services is for Microsoft365R to use its own internal app ID. As an alternative, you (or your admin) can create your own app registration in Azure: it should have a native redirect URI of `http://localhost:1410`, and the "public client" option should be enabled if you want to use the device code authentication flow. You can supply your app ID either via the `app` argument, or in the environment variable `CLIMICROSOFT365_AADAPPID`.

See Also

[ms_drive](#), [ms_site](#), [ms_team](#)

[add_methods](#) for the associated methods that this package adds to the base AzureGraph classes.

The "Authentication" vignette has more details on the authentication process, including troubleshooting and fixes for common problems.

[CLI for Microsoft 365](#) – a commandline tool for managing Microsoft 365

Examples

```
## Not run:

get_personal_onedrive()

# authenticating without a browser
get_personal_onedrive(auth_type="device_code")
```

```
odb <- get_business_onedrive("mycompany")
odb$list_items()

mysite <- get_sharepoint_site("My site", tenant="mycompany")
mysite <- get_sharepoint_site(site_url="https://mycompany.sharepoint.com/sites/my-site-url")
mysite$get_drive()$list_items()

myteam <- get_team("My team", tenant="mycompany")
myteam$list_channels()
myteam$get_drive()$list_items()

# you can also use your own app registration ID:
get_business_onedrive(app="app_id")
get_sharepoint_site("My site", app="app_id")

# using the app ID for the CLI for Microsoft 365: set a global option
options(microsoft365r_use_cli_app_id=TRUE)
get_business_onedrive()
get_sharepoint_site("My site")
get_team("My team")

## End(Not run)
```

ms_channel

Teams channel

Description

Class representing a Microsoft Teams channel.

Format

An R6 object of class `ms_channel`, inheriting from `ms_object`.

Fields

- `token`: The token used to authenticate with the Graph host.
- `tenant`: The Azure Active Directory tenant for the parent drive.
- `type`: Always "channel" for a channel object
- `team_id`: The ID of the parent team.
- `properties`: The item properties (metadata).

Methods

- `new(...)`: Initialize a new object. Do not call this directly; see 'Initialization' below.
- `delete(confirm=TRUE)`: Delete this channel. By default, ask for confirmation first.
- `update(...)`: Update the channel's properties (metadata) in Microsoft Graph.
- `do_operation(...)`: Carry out an arbitrary operation on the channel.
- `sync_fields()`: Synchronise the R object with the channel metadata in Microsoft Graph.
- `send_message(body, content_type, attachments)`: Sends a new message to the channel. See below.
- `list_messages(n=50)`: Retrieves the messages in the channel. By default, this is limited to the 50 most recent messages; set the `n` argument to change this.
- `get_message(message_id)`: Retrieves a specific message in the channel.
- `delete_message(message_id, confirm=TRUE)`: Deletes a message. Currently the Graph API does not support deleting Teams messages, so this method is disabled.
- `list_files()`: List the files for the channel. See [ms_drive](#) for the arguments available for this and the file upload/download methods.
- `upload_file()`: Uploads a file to the channel.
- `download_file()`: Downloads a file from the channel.
- `get_folder()`: Retrieves the files folder for the channel, as a [ms_drive_item](#) object.

Initialization

Creating new objects of this class should be done via the `get_channel` and `list_channels` methods of the [ms_team](#) class. Calling the `new()` method for this class only constructs the R object; it does not call the Microsoft Graph API to retrieve or create the actual channel.

Messaging

To send a message to a channel, use the `send_message()` method. This has arguments:

- `body`: The body of the message. This should be a character vector, which will be concatenated into a single string with newline separators. The body can be either plain text or HTML formatted.
- `content_type`: Either "text" (the default) or "html".
- `attachments`: Optional vector of filenames.
- `inline`: Optional vector of image filenames that will be inserted into the body of the message. The images must be PNG or JPEG, and the `content_type` argument must be "html" to include inline content.

Note that message attachments are actually uploaded to the channel's file listing (a directory in the team's primary shared document folder). Support for attachments is somewhat experimental, so if you want to be sure that it works, upload the file separately using the `upload_file()` method.

See Also

[ms_team](#), [ms_drive](#), [ms_chat_message](#)

[Microsoft Graph overview](#), [Microsoft Teams API reference](#)

Examples

```
## Not run:

myteam <- get_team("my team")
myteam$list_channels()

chan <- myteam$get_channel()
chan$list_messages()
chan$send_message("hello from R")

# a multi-line message with an attachment
msg_text <- c(
  "message line 1",
  "message line 2",
  "message line 3"
)
chan$send_message(msg_text, attachments="myfile.csv")

# sending an inline image
chan$send_message("", content_type="html", inline="graph.png")

chan$upload_file("mydocument.docx")

chan$list_files()

## End(Not run)
```

ms_chat_message

Teams chat message

Description

Class representing a message in a Teams channel. Currently Microsoft365R only supports channels, not chats between individuals.

Format

An R6 object of class `ms_chat_message`, inheriting from `ms_object`.

Fields

- `token`: The token used to authenticate with the Graph host.
- `tenant`: The Azure Active Directory tenant for the parent drive.
- `type`: Always "Teams message" for a chat message object.
- `properties`: The item properties (metadata).

Methods

- `new(...)`: Initialize a new object. Do not call this directly; see 'Initialization' below.
- `delete(confirm=TRUE)`: Delete this message. Currently the Graph API does not support deleting Teams messages, so this method is disabled.
- `update(...)`: Update the message's properties (metadata) in Microsoft Graph.
- `do_operation(...)`: Carry out an arbitrary operation on the message.
- `sync_fields()`: Synchronise the R object with the message metadata in Microsoft Graph.
- `send_reply(body, content_type, attachments)`: Sends a reply to the message. See below.
- `list_replies(n=50)`: List the replies to this message. By default, this is limited to the 50 most recent replies; set the `n` argument to change this.
- `get_reply(message_id)`: Retrieves a specific reply to the message.
- `delete_reply(message_id, confirm=TRUE)`: Deletes a reply to the message. Currently the Graph API does not support deleting Teams messages, so this method is disabled.

Initialization

Creating new objects of this class should be done via the `get_message` and `list_messages` method of the `ms_team` class. Calling the `new()` method for this class only constructs the R object; it does not call the Microsoft Graph API to retrieve or create the actual message.

Replying to a message

To reply to a message, use the `send_reply()` method. This has arguments:

- `body`: The body of the message. This should be a character vector, which will be concatenated into a single string with newline separators. The body can be either plain text or HTML formatted.
- `content_type`: Either "text" (the default) or "html".
- `attachments`: Optional vector of filenames.
- `inline`: Optional vector of image filenames that will be inserted into the body of the message. The images must be PNG or JPEG, and the `content_type` argument must be "html" to include inline content.

Teams channels don't support nested replies, so any methods dealing with replies will fail if the message object is itself a reply.

Note that message attachments are actually uploaded to the channel's file listing (a directory in the team's primary shared document folder). Support for attachments is somewhat experimental, so if you want to be sure that it works, upload the file separately using the channel's `upload_file()` method.

See Also

[ms_team](#), [ms_channel](#)

[Microsoft Graph overview](#), [Microsoft Teams API reference](#)

Examples

```
## Not run:

myteam <- get_team("my team")

chan <- myteam$get_channel()
msg <- chan$list_messages()[[1]]
msg$list_replies()
msg$send_reply("Reply from R")

## End(Not run)
```

ms_drive

Personal OneDrive or SharePoint document library

Description

Class representing a personal OneDrive or SharePoint document library.

Format

An R6 object of class `ms_drive`, inheriting from `ms_object`.

Fields

- `token`: The token used to authenticate with the Graph host.
- `tenant`: The Azure Active Directory tenant for this drive.
- `type`: always "drive" for a drive object.
- `properties`: The drive properties.

Methods

- `new(...)`: Initialize a new drive object. Do not call this directly; see 'Initialization' below.
- `delete(confirm=TRUE)`: Delete a drive. By default, ask for confirmation first.
- `update(...)`: Update the drive metadata in Microsoft Graph.
- `do_operation(...)`: Carry out an arbitrary operation on the drive.
- `sync_fields()`: Synchronise the R object with the drive metadata in Microsoft Graph.
- `list_items(...)`, `list_files(...)`: List the files and folders under the specified path. See 'File and folder operations' below.
- `download_file(src,dest,overwrite)`: Download a file.
- `upload_file(src,dest,blocksize)`: Upload a file.
- `create_folder(path)`: Create a folder.
- `open_item(path)`: Open a file or folder.

- `create_share_link(...)`: Create a shareable link for a file or folder.
- `delete_item(path, confirm)`: Delete a file or folder.
- `get_item(path)`: Get an item representing a file or folder.
- `get_item_properties(path)`: Get the properties (metadata) for a file or folder.
- `set_item_properties(path, ...)`: Set the properties for a file or folder.

Initialization

Creating new objects of this class should be done via the `get_drive` methods of the `ms_graph`, `az_user` or `ms_site` classes. Calling the `new()` method for this class only constructs the R object; it does not call the Microsoft Graph API to retrieve or create the actual drive.

File and folder operations

This class exposes methods for carrying out common operations on files and folders. They call down to the corresponding methods for the `ms_drive_item` class. In this context, any paths to child items are relative to the root folder of the drive.

`open_item` opens the given file or folder in your browser. If the file has an unrecognised type, most browsers will attempt to download it.

`list_items(path, info, full_names, pagesize)` lists the items under the specified path.

`list_files` is a synonym for `list_items`.

`download_file` and `upload_file` transfer files between the local machine and the drive. For `download_file`, the default destination folder is the current (working) directory of your R session. For `upload_file`, there is no default destination folder; make sure you specify the destination explicitly.

`create_folder` creates a folder with the specified path. Trying to create an already existing folder is an error.

`create_share_link(path, type, expiry, password, scope)` returns a shareable link to the item.

`delete_item` deletes a file or folder. By default, it will ask for confirmation first.

`get_item` retrieves the file or folder with the given path, as an object of class `ms_drive_item`.

`get_item_properties` is a convenience function that returns the properties of a file or folder as a list.

`set_item_properties` sets the properties of a file or folder. The new properties should be specified as individual named arguments to the method. Any existing properties that aren't listed as arguments will retain their previous values or be recalculated based on changes to other properties, as appropriate. You can also call the `update` method on the corresponding `ms_drive_item` object.

See Also

[get_personal_onedrive](#), [get_business_onedrive](#), [ms_site](#), [ms_drive_item](#)

[Microsoft Graph overview](#), [OneDrive API reference](#)

Examples

```
## Not run:

# personal OneDrive
mydrv <- get_personal_onedrive()

# OneDrive for Business
busdrv <- get_business_onedrive("mycompany")

# shared document library for a SharePoint site
site <- get_sharepoint_site("My site")
drv <- site$get_drive()

## file/folder operations
drv$list_files()
drv$list_files("path/to/folder", full_names=TRUE)

# download a file -- default destination filename is taken from the source
drv$download_file("path/to/folder/data.csv")

# shareable links
drv$create_share_link("myfile")
drv$create_share_link("myfile", type="edit", expiry="24 hours")
drv$create_share_link("myfile", password="Use-strong-passwords!")

# file metadata (name, date created, etc)
drv$get_item_properties("myfile")

# rename a file
drv$set_item_properties("myfile", name="newname")

## End(Not run)
```

ms_drive_item	<i>File or folder in a drive</i>
---------------	----------------------------------

Description

Class representing an item (file or folder) in a OneDrive or SharePoint document library.

Format

An R6 object of class `ms_drive_item`, inheriting from `ms_object`.

Fields

- `token`: The token used to authenticate with the Graph host.
- `tenant`: The Azure Active Directory tenant for the parent drive.

- `type`: always "drive item" for a drive item object.
- `properties`: The item properties (metadata).

Methods

- `new(...)`: Initialize a new object. Do not call this directly; see 'Initialization' below.
- `delete(confirm=TRUE)`: Delete this item. By default, ask for confirmation first. For personal OneDrives, deleting a folder will also automatically delete its contents; for business OneDrives or SharePoint document libraries, you must delete the folder contents first before deleting the folder.
- `update(...)`: Update the item's properties (metadata) in Microsoft Graph.
- `do_operation(...)`: Carry out an arbitrary operation on the item.
- `sync_fields()`: Synchronise the R object with the item metadata in Microsoft Graph.
- `open()`: Open the item in your browser.
- `list_items(...)`, `list_files(...)`: List the files and folders under the specified path.
- `download(dest, overwrite)`: Download the file. Only applicable for a file item.
- `create_share_link(type, expiry, password, scope)`: Create a shareable link to the file or folder.
- `upload(src, dest, blocksize)`: Upload a file. Only applicable for a folder item.
- `create_folder(path)`: Create a folder. Only applicable for a folder item.
- `get_item(path)`: Get a child item (file or folder) under this folder.
- `is_folder()`: Information function, returns TRUE if this item is a folder.

Initialization

Creating new objects of this class should be done via the `get_item` method of the `ms_drive` class. Calling the `new()` method for this class only constructs the R object; it does not call the Microsoft Graph API to retrieve or create the actual item.

File and folder operations

This class exposes methods for carrying out common operations on files and folders. Note that for the methods below, any paths to child items are relative to the folder's own path.

`open` opens this file or folder in your browser. If the file has an unrecognised type, most browsers will attempt to download it.

`list_items(path, info, full_names, pagesize)` lists the items under the specified path. It is the analogue of base R's `dir/list.files`. Its arguments are

- `path`: The path.
- `info`: The information to return: either "partial", "name" or "all". If "partial", a data frame is returned containing the name, size and whether the item is a file or folder. If "name", a vector of file/folder names is returned. If "all", a data frame is returned containing *all* the properties for each item (this can be large).
- `full_names`: Whether to prefix the folder path to the names of the items.

- `pagesize`: The number of results to return for each call to the REST endpoint. You can try reducing this argument below the default of 1000 if you are experiencing timeouts.

`list_files` is a synonym for `list_items`.

`download` downloads the file item to the local machine. It is an error to try to download a folder item.

`upload` uploads a file from the local machine into the folder item, and returns another `ms_drive_item` object representing the uploaded file. The uploading is done in blocks of 32MB by default; you can change this by setting the `blocksize` argument. For technical reasons, the block size **must be a multiple of 320KB**. This returns an `ms_drive_item` object, invisibly.

It is an error to try to upload to a file item, or to upload a source directory.

`get_item` retrieves the file or folder with the given path, as another object of class `ms_drive_item`.

`create_folder` creates a folder with the specified path. Trying to create an already existing folder is an error. This returns an `ms_drive_item` object, invisibly.

`create_share_link(path, type, expiry, password, scope)` returns a shareable link to the item. Its arguments are

- `path`: The path.
- `type`: Either "view" for a read-only link, "edit" for a read-write link, or "embed" for a link that can be embedded in a web page. The last one is only available for personal OneDrive.
- `expiry`: How long the link is valid for. The default is 7 days; you can set an alternative like "15 minutes", "24 hours", "2 weeks", "3 months", etc. To leave out the expiry date, set this to NULL.
- `password`: An optional password to protect the link.
- `scope`: Optionally the scope of the link, either "anonymous" or "organization". The latter allows only users in your AAD tenant to access the link, and is only available for OneDrive for Business or SharePoint.

This method returns a URL to access the item, for `type="view"` or `type="edit"`. For `type="embed"`, it returns a list with components `webUrl` containing the URL, and `webHtml` containing a HTML fragment to embed the link in an IFRAME. The default is a viewable link, expiring in 7 days.

See Also

[ms_graph](#), [ms_site](#), [ms_drive](#)

[Microsoft Graph overview](#), [OneDrive API reference](#)

Examples

```
## Not run:

# personal OneDrive
mydrv <- get_personal_onedrive()

docs <- mydrv$get_item("Documents")
docs$list_files()
docs$list_items()
```

```
# this is the file 'Documents/myfile.docx'
myfile <- docs$get_item("myfile.docx")
myfile$properties

# rename a file
myfile$update(name="newname.docx")

# open the file in the browser
myfile$open()

# download the file to the working directory
myfile$download()

# shareable links
myfile$create_share_link()
myfile$create_share_link(type="edit", expiry="24 hours")
myfile$create_share_link(password="Use-strong-passwords!")

# delete the file (will ask for confirmation first)
myfile$delete()

## End(Not run)
```

ms_list

Sharepoint list

Description

Class representing a list in a SharePoint site.

Format

An R6 object of class `ms_list`, inheriting from `ms_object`.

Fields

- `token`: The token used to authenticate with the Graph host.
- `tenant`: The Azure Active Directory tenant for the parent drive.
- `type`: always "list" for a SharePoint list object.
- `properties`: The item properties (metadata).

Methods

- `new(...)`: Initialize a new object. Do not call this directly; see 'Initialization' below.
- `delete(confirm=TRUE)`: Delete this list. By default, ask for confirmation first.
- `update(...)`: Update the list's properties in Microsoft Graph.

- `do_operation(...)`: Carry out an arbitrary operation on the list.
- `sync_fields()`: Synchronise the R object with the list metadata in Microsoft Graph.
- `list_items(filter, select, all_metadata, as_data_frame, pagesize)`: Queries the list and returns items as a data frame. See 'List querying' below.
- `get_column_info()`: Return a data frame containing metadata on the columns (fields) in the list.
- `get_item(id)`: Get an individual list item.
- `create_item(...)`: Create a new list item, using the named arguments as fields.
- `update_item(id, ...)`: Update the *data* fields in the given item, using the named arguments. To update the item's metadata, use `get_item()` to retrieve the item object, then call its `update()` method.
- `delete_item(confirm=TRUE)`: Delete a list item. By default, ask for confirmation first.
- `bulk_import(data)`: Imports a data frame into the list.

Initialization

Creating new objects of this class should be done via the `get_list` method of the `ms_site` class. Calling the `new()` method for this class only constructs the R object; it does not call the Microsoft Graph API to retrieve or create the actual item.

List querying

`list_items` supports the following arguments to customise results returned by the query.

- `filter`: A string giving a logical expression to filter the rows to return. Note that column names used in the expression must be prefixed with `fields/` to distinguish them from item metadata.
- `select`: A string containing comma-separated column names to include in the returned data frame. If not supplied, includes all columns.
- `all_metadata`: If `TRUE`, the returned data frame will contain extended metadata as separate columns, while the data fields will be in a nested data frame named `fields`.
- `as_data_frame`: If `FALSE`, return the result as a list of individual `ms_list_item` objects, rather than a data frame. The `all_metadata` argument is ignored if `as_data_frame=FALSE`.
- `pagesize`: The number of results to return for each call to the REST endpoint. You can try reducing this argument below the default of 5000 if you are experiencing timeouts.

For more information, see [Use query parameters](#) at the Graph API reference.

See Also

[get_sharepoint_site](#), [ms_site](#), [ms_list_item](#)

[Microsoft Graph overview](#), [SharePoint sites API reference](#)

Examples

```
## Not run:

site <- get_sharepoint_site("My site")
lst <- site$get_list("mylist")

lst$get_column_info()

lst$list_items()
lst$list_items(filter="startswith(fields/firstname, 'John')", select="firstname,lastname")

lst$create_item(firstname="Mary", lastname="Smith")
lst$get_item("item-id")
lst$update_item("item_id", firstname="Eliza")
lst$delete_item("item_id")

df <- data.frame(
  firstname=c("Satya", "Mark", "Tim", "Jeff", "Sundar"),
  lastname=c("Nadella", "Zuckerberg", "Cook", "Bezos", "Pichai")
)
lst$bulk_import(df)

## End(Not run)
```

ms_list_item

SharePoint list item

Description

Class representing an item in a SharePoint list.

Format

An R6 object of class `ms_list_item`, inheriting from `ms_object`.

Fields

- `token`: The token used to authenticate with the Graph host.
- `tenant`: The Azure Active Directory tenant for the parent drive.
- `type`: always "drive item" for a drive item object.
- `properties`: The item properties (metadata).

Methods

- `new(...)`: Initialize a new object. Do not call this directly; see 'Initialization' below.
- `delete(confirm=TRUE)`: Delete this item. By default, ask for confirmation first.

- `update(...)`: Update the item's properties (metadata) in Microsoft Graph. To update the list *data*, update the `fields` property. See the examples below.
- `do_operation(...)`: Carry out an arbitrary operation on the item.
- `sync_fields()`: Synchronise the R object with the item metadata in Microsoft Graph.

Initialization

Creating new objects of this class should be done via the `get_item` method of the `ms_list` class. Calling the `new()` method for this class only constructs the R object; it does not call the Microsoft Graph API to retrieve or create the actual item.

See Also

[ms_graph](#), [ms_site](#), [ms_list](#)

[Microsoft Graph overview](#), [SharePoint sites API reference](#)

Examples

```
## Not run:  
  
site <- get_sharepoint_site("My site")  
lst <- site$get_list("mylist")  
  
lst_items <- lst$list_items(as_data_frame=FALSE)  
  
item <- lst_items[[1]]  
  
item$update(fields=list(firstname="Mary"))  
  
item$delete()  
  
## End(Not run)
```

ms_site

Office 365 SharePoint site

Description

Class representing a SharePoint site.

Format

An R6 object of class `ms_site`, inheriting from `ms_object`.

Fields

- token: The token used to authenticate with the Graph host.
- tenant: The Azure Active Directory tenant for this site.
- type: always "site" for a site object.
- properties: The site properties.

Methods

- new(...): Initialize a new site object. Do not call this directly; see 'Initialization' below.
- delete(confirm=TRUE): Delete a site. By default, ask for confirmation first.
- update(...): Update the site metadata in Microsoft Graph.
- do_operation(...): Carry out an arbitrary operation on the site.
- sync_fields(): Synchronise the R object with the site metadata in Microsoft Graph.
- list_drives(): List the drives (shared document libraries) associated with this site.
- get_drive(drive_id): Retrieve a shared document library for this site. If the ID is not specified, this returns the default document library.
- list_subsites(): List the subsites of this site.
- get_lists(): Returns the lists that are part of this site.
- get_list(list_name, list_id): Returns a specific list, either by name or ID.
- get_group(): Retrieve the Microsoft 365 group associated with the site, if it exists. A site that backs a private Teams channel will not have a group associated with it.

Initialization

Creating new objects of this class should be done via the `get_sharepoint_site` method of the `ms_graph` or `az_group` classes. Calling the `new()` method for this class only constructs the R object; it does not call the Microsoft Graph API to retrieve or create the actual site.

See Also

[ms_graph](#), [ms_drive](#), [az_user](#)

[Microsoft Graph overview](#), [SharePoint sites API reference](#)

Examples

```
## Not run:  
  
site <- get_sharepoint_site("My site")  
site$list_drives()  
site$get_drive()  
  
## End(Not run)
```

ms_team	<i>Microsoft Teams team</i>
---------	-----------------------------

Description

Class representing a team in Microsoft Teams.

Format

An R6 object of class `ms_team`, inheriting from `ms_object`.

Fields

- `token`: The token used to authenticate with the Graph host.
- `tenant`: The Azure Active Directory tenant for this team.
- `type`: Always "team" for a team object.
- `properties`: The team properties.

Methods

- `new(...)`: Initialize a new team object. Do not call this directly; see 'Initialization' below.
- `delete(confirm=TRUE)`: Delete a team. By default, ask for confirmation first.
- `update(...)`: Update the team metadata in Microsoft Graph.
- `do_operation(...)`: Carry out an arbitrary operation on the team.
- `sync_fields()`: Synchronise the R object with the team metadata in Microsoft Graph.
- `list_channels(filter=NULL)`: List the channels for this team. Optionally, supply an OData expression to filter the list.
- `get_channel(channel_name, channel_id)`: Retrieve a channel. If the name and ID are not specified, returns the primary channel.
- `create_channel(channel_name, description, membership)`: Create a new channel. Optionally, you can specify a short text description of the channel, and the type of membership: either standard or private (invitation-only).
- `delete_channel(channel_name, channel_id, confirm=TRUE)`: Delete a channel; by default, ask for confirmation first. You cannot delete the primary channel of a team. Note that Teams keeps track of all channels ever created, even if you delete them (you can see the deleted channels by going to the "Manage team" pane for a team, then the "Channels" tab, and expanding the "Deleted" entry); therefore, try not to create and delete channels unnecessarily.
- `list_drives()`: List the drives (shared document libraries) associated with this team.
- `get_drive(drive_id)`: Retrieve a shared document library for this team. If the ID is not specified, this returns the default document library.
- `get_sharepoint_site()`: Get the SharePoint site associated with the team.
- `get_group()`: Retrieve the Microsoft 365 group associated with the team.

Initialization

Creating new objects of this class should be done via the `get_team` and `list_teams` methods of the `ms_graph`, `az_user` or `az_group` classes. Calling the `new()` method for this class only constructs the R object; it does not call the Microsoft Graph API to retrieve or create the actual team.

See Also

[ms_graph](#), [az_group](#), [ms_channel](#), [ms_site](#), [ms_drive](#)

[Microsoft Graph overview](#), [Microsoft Teams API reference](#)

Examples

```
## Not run:

myteam <- get_team("my team")
myteam$list_channels()
myteam$get_channel()
myteam$get_drive()

myteam$create_channel("Test channel", description="A channel for testing")
myteam$delete_channel("Test channel")

## End(Not run)
```

personal_onedrive *Deprecated client functions*

Description

Deprecated client functions

Usage

```
personal_onedrive(
  app = .microsoft365r_app_id,
  scopes = c("Files.ReadWrite.All", "User.Read"),
  ...
)

business_onedrive(
  tenant = Sys.getenv("CLIMICROSOFT365_TENANT", "common"),
  app = Sys.getenv("CLIMICROSOFT365_AADAPPID"),
  scopes = ".default",
  ...
)
```

```
sharepoint_site(  
    site_url = NULL,  
    site_id = NULL,  
    tenant = Sys.getenv("CLIMICROSOFT365_TENANT", "common"),  
    app = Sys.getenv("CLIMICROSOFT365_AADAPPID"),  
    scopes = ".default",  
    ...  
)
```

Arguments

app	A custom app registration ID to use for authentication. For <code>personal_onedrive</code> , the default is to use Microsoft365R's internal app ID. For <code>business_onedrive</code> and <code>sharepoint_site</code> , see below.
scopes	The Microsoft Graph scopes (permissions) to obtain.
...	Optional arguments to be passed to <code>AzureGraph::create_graph_login</code> .
tenant	For <code>business_onedrive</code> and <code>sharepoint_site</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.
site_url, site_id	For <code>sharepoint_site</code> , the web URL and ID of the SharePoint site to retrieve. Supply one or the other, but not both.

Details

These functions have been replaced by [get_personal_onedrive](#), [get_business_onedrive](#) and [get_sharepoint_site](#). They will be removed in a later version of the package.

Index

add_methods, [2](#), [6](#)
az_group, [3](#), [20](#), [22](#)
az_user, [3](#), [12](#), [20](#), [22](#)
AzureAuth::get_azure_token, [5](#)
AzureGraph::az_group, [2](#)
AzureGraph::az_user, [2](#)
AzureGraph::ms_graph, [2](#)

business_onedrive (personal_onedrive),
[22](#)

client-deprecated (personal_onedrive),
[22](#)

get_business_onedrive, [12](#), [23](#)
get_business_onedrive
 (get_personal_onedrive), [4](#)
get_personal_onedrive, [4](#), [12](#), [23](#)
get_sharepoint_site, [17](#), [23](#)
get_sharepoint_site
 (get_personal_onedrive), [4](#)
get_team (get_personal_onedrive), [4](#)

list_sharepoint_sites
 (get_personal_onedrive), [4](#)
list_teams (get_personal_onedrive), [4](#)

ms_channel, [7](#), [10](#), [22](#)
ms_chat_message, [8](#), [9](#)
ms_drive, [3](#), [6](#), [8](#), [11](#), [14](#), [15](#), [20](#), [22](#)
ms_drive_item, [8](#), [12](#), [13](#)
ms_graph, [12](#), [15](#), [19](#), [20](#), [22](#)
ms_list, [16](#), [19](#)
ms_list_item, [17](#), [18](#)
ms_site, [3](#), [6](#), [12](#), [15](#), [17](#), [19](#), [19](#), [22](#)
ms_team, [6](#), [8](#), [10](#), [21](#)

personal_onedrive, [22](#)

sharepoint_site (personal_onedrive), [22](#)