

Package ‘webdav’

May 8, 2026

Type Package

Title A Simple Interface for Interacting with 'WebDAV' Servers

Version 0.2.0

Date 2026-04-07

Description An easy-to-use interface for interacting with 'WebDAV' servers, including 'OwnCloud'. It simplifies the use of 'WebDAV' methods such as COPY, MOVE, DELETE and others.

With built-in authentication and request handling, it allows for easy management of files and directories over the 'WebDAV' protocol.

Depends R (>= 4.1.0)

Imports htr2, httpuv, xml2, dplyr, glue, stringr, purrr, tidyr, tibble, curl, cli

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), roxygen2, knitr, fortunes, rmarkdown,

URL <<https://github.com/StrategicProjects/webdav>>,
<<https://strategicprojects.github.io/webdav/>>

Config/Needs/website tidyverse/tidytemplate

NeedsCompilation no

Author Andre Leite [aut, cre],
Marcos Wasilew [aut],
Hugo Vasconcelos [aut],
Carlos Amorin [aut],
Diogo Bezerra [aut]

Maintainer Andre Leite <leite@castlab.org>

Repository CRAN

Date/Publication 2026-04-07 20:20:02 UTC

Contents

handle_response	2
webdav_copy_file	2
webdav_create_directory	3
webdav_create_request	4
webdav_delete_resource	5
webdav_download_file	6
webdav_list_files	7
webdav_upload_file	9

Index	11
--------------	-----------

handle_response	<i>Handle HTTP response from Server</i>
-----------------	---

Description

This function processes the response from the WebDAV server, checking for errors.

Usage

```
handle_response(response)
```

Arguments

response	The response object from an 'httr2' request.
----------	--

Value

The processed response object if successful, or an error if the request failed.

webdav_copy_file	<i>Copy a resource on the WebDAV server</i>
------------------	---

Description

This function copies a resource from one URI to another on the WebDAV server using the COPY method. It validates the provided parameters and handles errors during the copy process.

Usage

```
webdav_copy_file(  
    base_url,  
    from_path,  
    to_path,  
    username = Sys.getenv("WEBDAV_USERNAME"),  
    password = Sys.getenv("WEBDAV_PASSWORD"),  
    verbose = FALSE  
)
```

Arguments

base_url	The base URL of the WebDAV server.
from_path	The source path of the resource to copy.
to_path	The destination path where the resource will be copied.
username	The username for WebDAV authentication. Defaults to the "WEBDAV_USERNAME" environment variable.
password	The password for WebDAV authentication. Defaults to the "WEBDAV_PASSWORD" environment variable.
verbose	Logical. If TRUE, prints detailed messages during the copy process.

Value

Logical value indicating whether the resource was copied successfully.

webdav_create_directory

Create a collection (directory) on a WebDAV server

Description

This function creates a collection (directory/folder) on the WebDAV server using the MKCOL method. It validates parameters and handles errors during the process.

Usage

```
webdav_create_directory(  
    base_url,  
    folder_path,  
    username = Sys.getenv("WEBDAV_USERNAME"),  
    password = Sys.getenv("WEBDAV_PASSWORD"),  
    verbose = FALSE  
)
```

Arguments

base_url	The base URL of the WebDAV server (e.g., "https://example.com/remote.php/dav/files/").
folder_path	The path of the directory to create.
username	The username for WebDAV authentication. Defaults to the "WEBDAV_USERNAME" environment variable.
password	The password for WebDAV authentication. Defaults to the "WEBDAV_PASSWORD" environment variable.
verbose	Logical. If TRUE, prints detailed messages during the directory creation process.

Value

Logical value indicating whether the collection was created successfully.

Examples

```
# Example usage with a public WebDAV server.
# Visit test_server$url link to view the results of the operation.
library(httr2)
test_server <- "http://webdavserver.net/" |>
  request() |>
  req_retry(max_tries = 1, max_seconds = 2, backoff = ~ 1) |>
  req_perform() |>
  try(silent = TRUE)

# Create a directory on the WebDAV server
if (class(test_server) != "try-error")
  webdav_create_directory(base_url = test_server$url, folder_path = "Test_Folder", verbose = TRUE)
```

webdav_create_request *Create a request for the WebDAV server*

Description

This function creates a base request for the WebDAV server with proper authentication. It validates the provided parameters and handles errors during the connection setup.

Usage

```
webdav_create_request(
  base_url,
  username = Sys.getenv("WEBDAV_USERNAME"),
  password = Sys.getenv("WEBDAV_PASSWORD"),
  verbose = FALSE
)
```

Arguments

base_url	The base URL of the WebDAV server (e.g., "https://example.com/remote.php/dav/files/").
username	The username for WebDAV authentication. Defaults to the "WEBDAV_USERNAME" environment variable.
password	The password for WebDAV authentication. Defaults to the "WEBDAV_PASSWORD" environment variable.
verbose	Logical. If TRUE, prints detailed messages during the request creation process.

Value

An 'httr2_request' object with authentication and base URL configured, or an error message if the connection fails.

Examples

```
# Example usage with a public WebDAV server.
# Visit test_server$url link to view the results of the operation.
library(httr2)
test_server <- "http://webdavserver.net/" |>
  request() |>
  req_retry(max_tries = 1, max_seconds = 2, backoff = ~ 1) |>
  req_perform() |>
  try(silent = TRUE)

# Create a request
if (class(test_server) != "try-error")
  req <- webdav_create_request(base_url = test_server$url, verbose = TRUE)
```

webdav_delete_resource

Delete a file or directory from the WebDAV server

Description

This function deletes a file or directory on the WebDAV server using the DELETE method. It validates the provided parameters and handles errors during the process.

Usage

```
webdav_delete_resource(
  base_url,
  resource_path,
  username = Sys.getenv("WEBDAV_USERNAME"),
  password = Sys.getenv("WEBDAV_PASSWORD"),
  verbose = FALSE
)
```

Arguments

base_url	The base URL of the WebDAV server.
resource_path	The path of the file or directory to delete on the WebDAV server.
username	The username for WebDAV authentication. Defaults to the "WEBDAV_USERNAME" environment variable.
password	The password for WebDAV authentication. Defaults to the "WEBDAV_PASSWORD" environment variable.
verbose	Logical value indicating whether to print detailed debug messages. When TRUE, the function outputs additional information about its progress and actions.

Value

Logical value indicating whether the file or directory was deleted successfully.

Examples

```
# Example usage with a public WebDAV server.
# Visit test_server$url link to view the results of the operation.
library(httr2)
test_server <- "http://webdavserver.net/" |>
  request() |>
  req_retry(max_tries = 1, max_seconds = 2, backoff = ~ 1) |>
  req_perform() |>
  try(silent = TRUE)

# Delete a file or directory
if (class(test_server) != "try-error")
  webdav_delete_resource(base_url = test_server$url, resource_path = "Notes.txt", verbose = TRUE)
```

webdav_download_file *Download a file from the WebDAV server*

Description

This function downloads a file from the WebDAV server and saves it to a local directory. It validates the provided parameters, handles errors, and optionally prints detailed logs if requested.

Usage

```
webdav_download_file(
  base_url,
  file_path,
  destination_path = ".",
  username = Sys.getenv("WEBDAV_USERNAME"),
  password = Sys.getenv("WEBDAV_PASSWORD"),
  verbose = FALSE
)
```

Arguments

base_url	The base URL of the WebDAV server (e.g., "https://example.com/remote.php/dav/files/").
file_path	The path of the file on the WebDAV server to download (relative to the 'base_url').
destination_path	The local directory where the downloaded file will be saved. Defaults to the current directory.
username	The username for WebDAV authentication. Defaults to the "WEBDAV_USERNAME" environment variable.
password	The password for WebDAV authentication. Defaults to the "WEBDAV_PASSWORD" environment variable.
verbose	Logical. If TRUE, prints detailed messages during the download process.

Value

Logical value indicating whether the file was downloaded successfully.

Examples

```
# Example usage with a public WebDAV server.
library(httr2)
test_server <- "http://webdavserver.net/" |>
  request() |>
  req_retry(max_tries = 1, max_seconds = 2, backoff = ~ 1) |>
  req_perform() |>
  try(silent = TRUE)

# Download a file from the WebDAV server
if (class(test_server) != "try-error")
  webdav_download_file(base_url = test_server$url,
    file_path = "Project.pdf",
    destination_path = tempdir(),
    verbose = TRUE)
# Visit test_server$url to view the results of the operation.
```

webdav_list_files *List files from a specific folder on WebDAV server*

Description

This function lists the files in a specific folder on the WebDAV server. If no folder path is provided, it lists files from the root directory. The function validates the provided parameters and handles errors during the process.

Usage

```
webdav_list_files(
  base_url,
  folder_path = NULL,
  username = Sys.getenv("WEBDAV_USERNAME"),
  password = Sys.getenv("WEBDAV_PASSWORD"),
  depth = 1,
  verbose = FALSE
)
```

Arguments

<code>base_url</code>	The base URL of the WebDAV server.
<code>folder_path</code>	The path inside WebDAV where the files are located. If not provided or empty, the root folder will be listed.
<code>username</code>	The username for WebDAV authentication. Defaults to the "WEBDAV_USERNAME" environment variable.
<code>password</code>	The password for WebDAV authentication. Defaults to the "WEBDAV_PASSWORD" environment variable.
<code>depth</code>	The depth of the PROPFIND request (default is 1).
<code>verbose</code>	Logical value indicating whether to print detailed debug messages. When TRUE, the function outputs additional information about its progress and actions.

Value

A tibble containing:

display_name The name of the file or directory.
full_path The full URL (href) of the resource.
creation_date The date the resource was created.
last_modified The date the resource was last modified.
content_length The size of the resource in bytes (NA for directories).
is_folder Logical indicating whether the resource is a directory.

Returns 'NULL' if an error occurs during the execution of the function.

Examples

```
# Example usage with a public WebDAV server.
# Visit test_server$url link to view the results of the operation.
library(httr2)
test_server <- "http://webdavserver.net/" |>
  request() |>
  req_retry(max_tries = 1, max_seconds = 2, backoff = ~ 1) |>
  req_perform() |>
  try(silent = TRUE)
```

```
# List files in a directory
if (class(test_server) != "try-error")
  webdav_list_files(base_url = test_server$url, folder_path = "Sales/", verbose = TRUE)
```

webdav_upload_file *Upload a file to the WebDAV server*

Description

This function uploads a file to a specific folder on the WebDAV server. It validates the provided parameters and handles errors during the process.

Usage

```
webdav_upload_file(
  base_url,
  local_path,
  server_path = "",
  username = Sys.getenv("WEBDAV_USERNAME"),
  password = Sys.getenv("WEBDAV_PASSWORD"),
  timeout = 300,
  verbose = FALSE
)
```

Arguments

base_url	The base URL of the WebDAV server.
local_path	The local path of the file to be uploaded.
server_path	The folder path on the WebDAV server where the file will be uploaded.
username	The username for WebDAV authentication. Defaults to the "WEBDAV_USERNAME" environment variable.
password	The password for WebDAV authentication. Defaults to the "WEBDAV_PASSWORD" environment variable.
timeout	The timeout for the upload request in seconds (default is 300 seconds).
verbose	Logical value indicating whether to print detailed debug messages. When TRUE, the function outputs additional information about its progress and actions.

Value

Logical value indicating whether the file was uploaded successfully.

Examples

```
# Example usage with a public WebDAV server.
# Visit test_server$url link to view the results of the operation.
library(httr2)
test_server <- "http://webdavserver.net/" |>
  request() |>
  req_retry(max_tries = 1, max_seconds = 2, backoff = ~ 1) |>
  req_perform() |>
  try(silent = TRUE)

# Upload a file
file_test <- tempfile(pattern = "teste_", fileext = ".txt")
cat("Text file content", file = file_test)
if (class(test_server) != "try-error")
  webdav_upload_file(base_url = test_server$url, local_path = file_test, verbose = TRUE)
```

Index

`handle_response`, [2](#)

`webdav_copy_file`, [2](#)

`webdav_create_directory`, [3](#)

`webdav_create_request`, [4](#)

`webdav_delete_resource`, [5](#)

`webdav_download_file`, [6](#)

`webdav_list_files`, [7](#)

`webdav_upload_file`, [9](#)