

# Package ‘rdav’

October 23, 2025

**Type** Package

**Title** Simple 'WebDAV' Client

**Version** 0.1.3

**Date** 2025-10-20

**Maintainer** Gunther Krauss <guntherkrauss@uni-bonn.de>

**Description** A simple 'WebDAV' client that provides functions to fetch and send files or folders to servers using the 'WebDAV' protocol  
(see 'RFC' 4918 <<https://www.rfc-editor.org/rfc/rfc4918>>).  
Only a subset of the protocol is implemented (e.g. file locks are not yet supported).

**License** GPL-2

**URL** <https://github.com/gk-crop/rdav>

**BugReports** <https://github.com/gk-crop/rdav/issues>

**Depends** R (>= 4.1.0)

**Imports** utils, httr2, xml2

**Suggests** keyring, knitr, rmarkdown, testthat (>= 3.0.0)

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**VignetteBuilder** knitr

**Config/testthat.edition** 3

**NeedsCompilation** no

**Author** Gunther Krauss [aut, cre]

**Repository** CRAN

**Date/Publication** 2025-10-23 14:50:02 UTC

## Contents

<i>rdav</i> . . . . .	2
<i>wd_connect</i> . . . . .	3
<i>wd_copy</i> . . . . .	4
<i>wd_delete</i> . . . . .	5
<i>wd_dir</i> . . . . .	6
<i>wd_download</i> . . . . .	7
<i>wd_isdir</i> . . . . .	7
<i>wd_mkdir</i> . . . . .	8
<i>wd_move</i> . . . . .	9
<i>wd_upload</i> . . . . .	9

## Index

11

---

rdav*rdav: Interchange Files With 'WebDAV' Servers*

---

### Description

Provides functions to interchange files with WebDAV servers

### Details

- download a file or a directory (recursively) from a WebDAV server
- upload a file or a directory (recursively) to a WebDAV server
- copy, move, delete files or directories on a WebDAV server
- list directories on the WebDAV server

Notice: when uploading or downloading files, they are overwritten without any warnings.

### Author(s)

Gunther Krauss

### See Also

Useful links:

- <https://github.com/gk-crop/rdav>
- Report bugs at <https://github.com/gk-crop/rdav/issues>

## Examples

```
## Not run:  
# establish a connection, you will be asked for a password  
r <- wd_connect("https://example.com/remote.php/webdav/", "exampleuser")  
  
# show files / directoriess in main directory  
wd_dir(r)  
  
# lists 'subdir', returns a dataframe  
wd_dir(r, "subdir", as_df = TRUE)  
  
# create directory 'mydirectory' on the server  
wd_mkdir(r, "mydirectory")  
  
# upload the local file testfile.R to the subdirectory 'mydirectory'  
wd_upload(r, "testfile.R", "mydirectory/testfile.R")  
  
# download content of 'mydirectory' from the server and  
# store it in 'd:/data/fromserver' on your computer  
wd_download(r, "mydirectory", "d:/data/fromserver")  
  
## End(Not run)
```

---

wd\_connect

*Establishes a connection to a WebDAV server*

---

## Description

Creates and authenticate a request handle to the WebDAV server

## Usage

```
wd_connect(url, username, password = NULL)
```

## Arguments

url	url of the WebDAV directory
username	username
password	password - if not given, you will be asked for it

## Details

Notice: it's not recommended to write the password as plain text. Either omit the parameter (then you will be asked to enter a password interactively) or use for example the system credential store via keyring package.

**Value**

a httr2 request to the WebDAV server location

**Examples**

```
## Not run:
# establish a connection, you will be asked for a password

r <- wd_connect("https://example.com/remote.php/webdav/", "exampleuser")

# establish a connection, use keyring package to retrieve the password

keyring::key_set("mydav", "exampleuser") # call only once

r <- wd_connect("https://example.com/remote.php/webdav/",
               "exampleuser",
               keyring::key_get("mydav", "exampleuser"))

## End(Not run)
```

**wd\_copy***Copies a file or directory on the WebDAV server***Description**

Copies a file or directory on the WebDAV server

**Usage**

```
wd_copy(req, source, target, overwrite = TRUE)
```

**Arguments**

<code>req</code>	request handle obtained from <a href="#">wd_connect</a>
<code>source</code>	path of the source on the server
<code>target</code>	path of the target on the server
<code>overwrite</code>	overwrites files when TRUE (default)

**Value**

TRUE on success, FALSE on failure (invisibly)

## Examples

```
## Not run:  
wd_copy(r, "testfile.R", "testfile_old.R")  
  
## End(Not run)
```

---

wd\_delete

*Deletes a file or directory (collection) on WebDAV server*

---

## Description

Deletes a file or directory (collection) on WebDAV server

## Usage

```
wd_delete(req, file)
```

## Arguments

req	request handle obtained from <a href="#">wd_connect</a>
file	path to file or directory to delete on the server

## Value

TRUE on success, FALSE on failure (invisibly)

## Examples

```
## Not run:  
wd_delete(r, "testfile.R")  
  
## End(Not run)
```

---

wd_dir	<i>Lists the content of a WebDAV directory</i>
--------	--

---

## Description

Lists the content of a WebDAV directory

## Usage

```
wd_dir(req, directory = "", full_names = FALSE, as_df = FALSE)
```

## Arguments

req	request handle obtained from <a href="#">wd_connect</a>
directory	directory path
full_names	if TRUE, the directory path is prepended to the file names to give a relative file path (relevant only if as_df is FALSE)
as_df	if TRUE outputs a data.frame with file information

## Value

a vector of filenames or a dataframe (when as\_df is TRUE) with detailed file information (filename, path, isdir, size, lastmodified)

## Examples

```
## Not run:

# lists names of files and directories in the main directory
wd_dir(r)

# lists names of files and directories in the subdirectory "mydirectory"
wd_dir(r, "mydirectory")

# lists names of files and directories with the relative path
wd_dir(r, "mydirectory", full_names=TRUE)

# returns a data.frame with the columns filename, size and isdir (whether
# it's a directory or file
wd_dir(r, "mydirectory", as_df=TRUE)

## End(Not run)
```

---

wd\_download*Fetches a file or directory (recursively) from the WebDAV server*

---

**Description**

Directories are downloaded recursively. If the source is a file and the target a directory, then the file is downloaded to the target directory. If the target is omitted, then the file or directory name ([basename](#)) will be used.

**Usage**

```
wd_download(req, source, target = "")
```

**Arguments**

req	request handle obtained from <a href="#">wd_connect</a>
source	path to source file or directory on server
target	path to local target file or directory, if omitted the file or directory name will be used, if source is a file and target a directory then the file will be put into the target directory

**Value**

vector of downloaded files (invisibly)

**Examples**

```
## Not run:  
  
wd_download(r, "weatherfiles", "d:/data/weather")  
wd_download(r, "test/xyz.txt", "d:/data/abc.txt")  
  
## End(Not run)
```

---

wd\_isdir

*Checks if the resource on WebDAV is a directory*

---

**Description**

Checks if the resource on WebDAV is a directory

**Usage**

```
wd_isdir(req, directory, silent = FALSE)
```

**Arguments**

req	request handle obtained from <a href="#">wd_connect</a>
directory	path to directory
silent	if FALSE a warning is given if the directory does not exists

**Value**

TRUE if it is a directory, FALSE else

**Examples**

```
## Not run:

wd_isdir(r, "testfile.R") # FALSE
wd_isdir(r, "mydirectory") # TRUE

## End(Not run)
```

**wd\_mkdir**

*Creates a directory (collection) on WebDAV server*

**Description**

When creating a subdirectory, all parent directories have to exist on the server.

**Usage**

```
wd_mkdir(req, directory)
```

**Arguments**

req	request handle obtained from <a href="#">wd_connect</a>
directory	directory path on server

**Value**

TRUE on success, FALSE on failure (invisibly)

**Examples**

```
## Not run:

# creates 'newdir' inside the subdirectory 'existing/directory'
wd_mkdir(r, "existing/directory/newdir")

## End(Not run)
```

---

wd_move	<i>Moves a file or directory on the server</i>
---------	--

---

## Description

Moves a file or directory on the server

## Usage

```
wd_move(req, source, target, overwrite = TRUE)
```

## Arguments

req	request handle obtained from <a href="#">wd_connect</a>
source	path of the source on the server
target	path of the target on the server
overwrite	overwrites files when TRUE (default)

## Value

TRUE on success, FALSE on failure (invisibly)

## Examples

```
## Not run:  
  
wd_move(r, "testfile.R", "testfile_old.R")  
  
## End(Not run)
```

---

wd_upload	<i>Uploads a file or directory to WebDAV</i>
-----------	--

---

## Description

Directories are uploaded recursively. If the source is a file and the target a directory, then the file is uploaded into the directory. If the target is omitted, then the file or directory name ([basename](#)) will be used.

## Usage

```
wd_upload(req, source, target = "")
```

**Arguments**

req	request handle obtained from <a href="#">wd_connect</a>
source	path to local file or directory
target	path to remote file or directory, if omitted the file or directory name will be used, if source is a file and target a directory then the file will be put into the target directory

**Value**

vector of uploaded files (invisibly)

**Examples**

```
## Not run:  
  
wd_upload(r, "d:/data/weather", "weatherfiles")  
wd_upload(r, "d:/data/abc.txt", "test/xyz.txt")  
wd_upload(r, "d:/data/abc.txt", "test") # uploaded file will be test/abc.txt  
  
## End(Not run)
```

# Index

basename, [7](#), [9](#)

rdav, [2](#)

rdav-package (rdav), [2](#)

wd\_connect, [3](#), [4–10](#)

wd\_copy, [4](#)

wd\_delete, [5](#)

wd\_dir, [6](#)

wd\_download, [7](#)

wd\_isdir, [7](#)

wd\_mkdir, [8](#)

wd\_move, [9](#)

wd\_upload, [9](#)