Package: sftpR (via r-universe)

October 7, 2026

Type Package

Title Robust SFTP Interface Using 'curl'

Version 0.2.0

Description Provides a high-level, object-oriented interface for Secure File Transfer Protocol (SFTP) operations built upon the 'curl' package. The package implements an 'R6' class to manage persistent connections and provides 'tidyverse'-style functions for common file system tasks. Key features include recursive directory creation with idempotency support, ``smart'' local path resolution that distinguishes between files and directories, and the ability to download remote resources directly into memory as raw vectors for seamless integration into data processing pipelines. It is designed to handle common SFTP edge cases gracefully, providing informative error messages and robust path sanitization to ensure compatibility across different server configurations.

License MIT + file LICENSE

URL https://mikuo0628.github.io/sftpR/, https://github.com/mikuo0628/sftpR

BugReports https://github.com/mikuo0628/sftpR/issues

Encoding UTF-8

LazyData true

SystemRequirements libcurl: libcurl (with libssh2 support)

Depends R (>= 4.1.0)

Imports curl (>= 7.0.0), R6 (>= 2.6.1)

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

VignetteBuilder knitr

Collate 'sftp_connect.R' 'sftp_delete.R' 'sftp_download.R' 'sftp_list.R' 'sftp_mkdir.R' 'sftp_rename.R' 'sftp_upload.R' 'utils.R' 'shared_docs.R'

Repository https://mikuo0628.r-universe.dev

Date/Publication 2026-04-09 19:03:58 UTC

RemoteUrl https://github.com/mikuo0628/sftpr

RemoteRef HEAD

RemoteSha 693799cea227767f4e2101bf75aa8a48d4f991bb

Contents

sftp_connect 2
sftp_delete 3
sftp_download 5
sftp_list 6
sftp_mkdir 8
sftp_rename 9
sftp_upload 11

Index **13**

`sftp_connect` *Create an* `SFTPConn` *R6 object that contains important connection information safely*

Description

An R6 class to safely store information needed for SFTP connection, with convenient methods to check connections and existence of files or directories, and create specific handles for `sftp_*` function family of CRUD operations.

Usage

```
sftp_connect(
  protocol = "sftp",
  hostname = "localhost",
  path = NULL,
  port = "22",
  user = NA_character_,
  password = NA_character_,
  timeout = 30L,
  ...,
  .verbose = TRUE
)
```

Arguments

protocol	Character. Protocol string. Defaults to "sftp".
hostname	Character. Server URL or IP. Defaults to "localhost".
path	Character. Sub-path on server.
port	Character. Port number. Defaults to "22".
user	Character. SFTP account name.
password	Character. SFTP password.
timeout	Integer. Connection timeout.
...	Additional arguments passed to curl::handle_setopt().
.verbose	Logical. Defaults to TRUE. Prints helpful messages.

Details

One important goal of this design choice is to keep user credentials safe, as private fields. Credentails are used to create specific handles for sftp_* family, and are reused where approrpiate. SFTPConn This class checks if credential is valid, and has a internal convenience methods such as safe printing for basic information, checking destination existence, and ensuring URL is correctly formatted.

Value

SFTPConn R6 class object, used in sftp_* family.

Examples

```
if (interactive() || Sys.getenv("R_SFTP_TEST_SERVER") == "true") {
  # Create a new SFTP connection
  sftp_conn <- sftp_connect(
    hostname = "127.0.0.1",
    port     = 2222,
    user     = "tester",
    password = "password123"
  )
}
```

---

sftp_delete *Delete Files or Directories from SFTP Server*

Description

Deletes a specific file or directory from the remote server. If the target is a directory, it must be empty unless .recursive = TRUE is specified.

Usage

```
sftp_delete(
  sftp_conn,
  remote_url = NULL,
  .recursive = FALSE,
  .verbose = TRUE,
  .validate = TRUE
)
```

Arguments

sftp_conn	An SFTPConn object containing connection details and authentication. Created by sftp_connect.
remote_url	Character. The full URL or path of the file or directory to be operated on.
.recursive	Logical. Defaults to FALSE. If TRUE, will recursively perform the SFTP operation: • sftp_delete(): deletes the directory and everything within. • sftp_list(): lists all the directories and files. • sftp_mkdir(): creates all the missing parent directories. • sftp_rename(): see sftp_mkdir().
.verbose	Logical. Defaults to TRUE. Prints helpful messages.
.validate	Logical. Whether to validate the remote_url against the connection object. Defaults to TRUE, which will parse remote_url, comapre to that of SFTPConn, and replaces parts incongruent with SFTPConn. Internally set to FALSE when .recursive = TRUE because the URLs produced by the listing operation aren't subjected to human errors, thus do not need further validation. This provides a minor performance boost.

Value

TRUE (invisibly) if the operation was successful.

Safety Warnings

- **Irreversibility:** Deletion on SFTP is permanent. There is no "Trash" or "Recycle Bin" on most SFTP server configurations.
- **Recursive Caution:** Setting .recursive = TRUE on a high-level directory can result in significant data loss. Always verify the remote_url before executing.

Examples

```
if (interactive() || Sys.getenv("R_SFTP_TEST_SERVER") == "true") {
  # Create new SFTP connection
  sftp_conn <- sftp_connect(
    hostname = "127.0.0.1",
    port     = "2222",
    user     = "tester",
```

```
    password = "password123"
  )

  # Delete a single file
  sftp_delete(sftp_conn, "project/old_report.csv")

  # Delete an entire directory and its contents
  sftp_delete(sftp_conn, "project/temp_outputs/", .recursive = TRUE)
}
```

`sftp_download` *Download Files from SFTP Server*

Description

Downloads a file from a remote SFTP server to a local disk location or directly into R's memory as a raw vector.

Usage

```
sftp_download(
  sftp_conn,
  remote_file,
  local_file = NA_character_,
  .create_dir = FALSE,
  .overwrite = FALSE,
  .verbose = TRUE,
  ...
)
```

Arguments

`sftp_conn`	An `SFTPConn` object containing connection details and authentication. Created by `sftp_connect`.
`remote_file`	Character. The path or URL of the file on the SFTP server.
`local_file`	Character or `NULL`. • If `NA` (default) or an `""`: The file is saved to the current working directory while using `remote_file` filename. • If `character`: The local path where the file should be saved. • If `NULL`: The file is downloaded to memory and returned as a `raw` vector.
`.create_dir`	Logical. Defaults to `FALSE`. If `TRUE`, creates the necessary parent directories if needed.
`.overwrite`	Logical. Defaults to `FALSE`. If `TRUE`, will overwrite destination file.
`.verbose`	Logical. Defaults to `TRUE`. Prints helpful messages.
`...`	Additional arguments passed to `curl::curl_download`.

Value

If `local_file` is `NULL`, a `raw` vector of the file contents. Otherwise, the resolved local path to the saved file (invisibly).

Local Path Resolution Caveats

To provide a "smart" user experience, the function guesses if `local_file` is intended to be a directory or a specific filename:

- **Directory Detection:** If the path exists as a directory, ends in a trailing slash, or has no file extension, it is treated as a folder. The `remote_file` filename will be appended to this path.
- **File Detection:** If the path does not exist and contains a file extension (e.g., ".csv"), it is treated as the final destination filename.
- **Ambiguity:** In ambiguous cases (e.g., a non-existent path without a slash or extension), the function defaults to treating the path as a directory.

Examples

```
if (interactive() || Sys.getenv("R_SFTP_TEST_SERVER") == "true") {
  # Create new SFTP connection
  sftp_conn <- sftp_connect(
    hostname = "127.0.0.1",
    port     = "2222",
    user     = "tester",
    password = "password123"
  )

  # Download and use the remote filename
  sftp_download(sftp_conn, "data/raw_logs.zip")

  # Download to a specific local name
  sftp_download(sftp_conn, "remote_file.csv", "local_name.csv")

  # Download to memory for immediate processing
  raw_bytes <- sftp_download(sftp_conn, "data.json", local_file = NULL)
  # Parse with appropriate packages
  # data <- jsonlite::fromJSON(rawToChar(raw_bytes))
}
```

---

sftp_list *List and Crawl SFTP Directory Contents*

Description

Retrieves a directory listing from an SFTP server. If `.recursive = TRUE`, it will perform a depth-first crawl of all subdirectories found, implementing a path-tracking algorithm to detect and skip circular symbolic links, preventing infinite recursion and stack overflow errors.

Usage

```
sftp_list(
  sftp_conn = NULL,
  sftp_url = NULL,
  .verbose = TRUE,
  .recursive = FALSE
)
```

Arguments

sftp_conn	An SFTPConn object containing connection details and authentication. Created by sftp_connect.
sftp_url	A SFTP URL of which the contents will be listed. If NULL, the base URL in SFTPConn will be used: contents of the SFTP home folder will be listed.
.verbose	Logical. Defaults to TRUE. Prints helpful messages.
.recursive	Logical. Defaults to FALSE. If TRUE, will recursively perform the SFTP operation: • sftp_delete(): deletes the directory and everything within. • sftp_list(): lists all the directories and files. • sftp_mkdir(): creates all the missing parent directories. • sftp_rename(): see sftp_mkdir().

Value

A data.frame containing remote file/directory metadata:

- permission: Unix-style permission string (e.g., "drwxr-xr-x").
- nlink: Number of hard links.
- user: Owner username.
- group: Owner group.
- size: File size in bytes.
- month, day, time_year: Timestamp components.
- name: File or directory name.
- type: Categorization as "dir" or "file".
- url: The source URL for that specific object.

Examples

```
if (interactive() || Sys.getenv("R_SFTP_TEST_SERVER") == "true") {
  # Create a new SFTP connection
  sftp_conn <- sftp_connect(
    hostname = "127.0.0.1",
    port     = "2222",
    user     = "tester",
    password = "password123"
```

```
  )

  # List recursively
  sftp_list(sftp_conn, .recursive = TRUE)
}
```

`sftp_mkdir` *Create Remote Directories in SFTP*

Description

This function creates directories on a remote server using the SFTP protocol. It supports recursive directory creation, effectively behaving like `mkdir -p` on a Unix-like system.

Usage

```
sftp_mkdir(
  sftp_conn,
  remote_url = NULL,
  .recursive = TRUE,
  .verbose = TRUE,
  .ignore_error = .recursive
)
```

Arguments

`sftp_conn`	An `SFTPConn` object containing connection details and authentication. Created by `sftp_connect`.
`remote_url`	Character. The full URL or path of the file or directory to be operated on.
`.recursive`	Logical. Defaults to `FALSE`. If `TRUE`, will recursively perform the SFTP operation: • `sftp_delete()`: deletes the directory and everything within. • `sftp_list()`: lists all the directories and files. • `sftp_mkdir()`: creates all the missing parent directories. • `sftp_rename()`: see `sftp_mkdir()`.
`.verbose`	Logical. Defaults to `TRUE`. Prints helpful messages.
`.ignore_error`	Logical. If `TRUE`, the function uses the `*` prefix in the curl quote command to ignore errors (e.g., if the directory already exists). This is useful in `.recursive = TRUE` because attempting to create a directory that already exists will return error. While this can be avoided by checking for directory existence, doing so adds extra step that impacts performance.

Details

When `.recursive = TRUE`, the function splits the path into segments and attempts to create each one sequentially. It uses the `*` prefix for internal calls to ensure that existing directories do not trigger errors.

Value

`invisible(TRUE)` on success.

Examples

```
## Not run:
if (interactive() || Sys.getenv("R_SFTP_TEST_SERVER") == "true") {
  # Create new SFTP connection
  sftp_conn <- sftp_connect(
    hostname = "127.0.0.1",
    port     = "2222",
    user     = "tester",
    password = "password123"
  )

  # Create a nested directory structure
  sftp_mkdir(sftp_conn, "project/data/results/2026", .recursive = TRUE)

  # Create a single directory and fail if parents are missing
  sftp_mkdir(sftp_conn, "simple_dir", .recursive = FALSE)
}

## End(Not run)
```

---

sftp_rename *Rename or Move Remote SFTP Resources*

Description

Renames a file or directory on the SFTP server. This can also be used to move files between directories.

Usage

```
sftp_rename(
  sftp_conn,
  remote_url_from = NULL,
  remote_url_to = NULL,
  .recursive = FALSE,
  .verbose = TRUE
)
```

Arguments

sftp_conn	An SFTPConn object containing connection details and authentication. Created by sftp_connect.
remote_url_from	Character. The current path of the file or directory.
remote_url_to	Character. The new path for the file or directory.
.recursive	Logical. Defaults to FALSE. If TRUE, will recursively perform the SFTP operation: • sftp_delete(): deletes the directory and everything within. • sftp_list(): lists all the directories and files. • sftp_mkdir(): creates all the missing parent directories. • sftp_rename(): see sftp_mkdir().
.verbose	Logical. Defaults to TRUE. Prints helpful messages.

Details

The SFTP protocol's rename command is typically non-overwriting. If remote_url_to already exists, the operation will fail.

When .recursive = TRUE, parent directories of remote_url_to are identified, and existence ensured before attempting the renaming.

Value

invisible(TRUE) on success.

Examples

```
if (interactive() || Sys.getenv("R_SFTP_TEST_SERVER") == "true") {
  # Create new SFTP connection
  sftp_conn <- sftp_connect(
    hostname = "127.0.0.1",
    port     = "2222",
    user     = "tester",
    password = "password123"
  )

  # Simple rename in the same folder
  sftp_rename(sftp_conn, "old_name.csv", "new_name.csv")

  # Move a file to a new, potentially non-existent directory
  sftp_rename(
    sftp_conn,
    "data/raw.csv",
    "archive/2026/processed.csv",
    .recursive = TRUE
  )
}
```

`sftp_upload` *Upload a file to an SFTP server*

Description

A robust wrapper to upload local files or data frames to a remote SFTP server. It manages the libcurl handle lifecycle, ensures connections are closed, and validates URLs against the connection's "Source of Truth".

Usage

```
sftp_upload(
  sftp_conn,
  local_file,
  remote_file = NULL,
  .create_dir = FALSE,
  .verbose = TRUE
)
```

Arguments

`sftp_conn`	An `SFTPConn` object containing connection details and authentication. Created by `sftp_connect`.
`local_file`	Character string (path to a file) or a `data.frame`. Data frames are automatically written to a temp file before upload. The temp file will automatically be cleaned up at the end of function.
`remote_file`	Character string. The destination path on the server. If `NULL`, attempts to use the basename of the `local_file`.
`.create_dir`	Logical. Defaults to `FALSE`. If `TRUE`, creates the necessary parent directories if needed.
`.verbose`	Logical. Defaults to `TRUE`. Prints helpful messages.

Details

The function uses a secure lifecycle:

1. Validates the remote URL to prevent credential leakage.
2. Opens a file connection to the local source.
3. Uses `on.exit` to ensure file handles are released and temporary files are unlinked even if the transfer is interrupted.

Value

Returns `TRUE` (invisibly) on success. Throws an error on failure.

Examples

```
if (interactive() || Sys.getenv("R_SFTP_TEST_SERVER") == "true") {
  # Create new SFTP connection
  sftp_conn <- sftp_connect(
    hostname = "127.0.0.1",
    port     = "2222",
    user     = "tester",
    password = "password123"
  )

  # Upload `my_df` as csv
  sftp_upload(conn, my_df, "uploads/data.csv")
}
```

Index

sftp_connect, 2, *4, 5*, *7, 8*, *10, 11*
sftp_delete, 3
sftp_download, 5
sftp_list, 6
sftp_mkdir, 8
sftp_rename, 9
sftp_upload, 11