rfslib

Release 1.0.22

Přemysl Šťastný

Aug 03, 2021

CONTENTS:

1 rfslib.abstract_pconnection module 3

2 rfslib.sftp_pconnection module 5

3 rfslib.ftp_pconnection module 7

4 rfslib.smb12_pconnection module 9

5 rfslib.smb23_pconnection module 11

6 rfslib.fs_pconnection module 13

7 rfslib.path_utils module 15

Python Module Index 17

Index 19

This is a documentation of rfslib.

To create a new development enviroment, it is recommended to create python virtual enviroment and install dependencies in requirements.txt

If you want to create a new pdf documentation, you are required to install also texlive on your system.

CHAPTER

ONE

RFSLIB.ABSTRACT_PCONNECTION MODULE

class rfslib.abstract_pconnection.**PConnection**(***args*)

Bases: abc.ABC

abstract _isdir(*remote_path: str*) → bool

Protected method which checks, whether a remote file is a directory.

Parameters **remote_path** – Path of a directory.

Returns True, if remote file is folder. False, if it isn't a folder. Undefined if the file doesns't exist.

abstract _listdir()

Protected method which returns a list of files in the folder.

Parameters **remote_path** – The remote path of a remote folder.

Returns List of files ([str]) in the remote folder

abstract _pull(*remote_path: str*, *local_path: str*)

Protected method which downloads/pulls a file from a remote storage to a local storage in the binary form.

Parameters

- **remote_path** – Path of a remote file to download.
- **local_path** – Path of a local file, where to download/pull a remote file.

abstract _push(*local_path: str*, *remote_path: str*)

Protected method which uploads/pushes a file from a local storage to a remote storage in the binary form.

Parameters

- **local_path** – Path of a local file to upload.
- **remote_path** – Path on the remote storage, where to upload/push a local file.

abstract _rename(*old_name: str*, *new_name: str*)

Protected method which renames/moves a file.

Parameters

- **old_name** – Remote path a file to move.
- **new_name** – Remote path to which move the file.

abstract close()

cp(*old_names*, *new_name*, *recursive=False*)

dcp(*old_names*, *target_dir*, *recursive=False*)

dmv(*old_names*, *target_dir*)

`exists`(*remote_path*)

`fcp`(*old_name*, *new_name*)

`find`(*remote_path*, *child_first=False*)

`fmv`(*old_name*, *new_name*)

`isdir`(*remote_path*)

`lexists`(*remote_path*)

`listdir`(*remote_path*)

`ls`(*remote_path*)

`mkdir`(*remote_path*)

`mv`(*old_names*, *new_name*)

`pmkdir`(*remote_path*)

`pull`(*remote_path*, *local_path*)

`push`(*local_path*, *remote_path*)

`rename`(*old_name*, *new_name*)

`rm`(*remote_path*, *recursive=False*)

`rmdir`(*remote_path*)

`rpull`(*remote_path*, *local_path*)

`rpush`(*local_path*, *remote_path*)

`touch`(*remote_path*)

`unlink`(*remote_path*)

`xls`(*remote_path*)

RFSLIB.SFTP_PCONNECTION MODULE

class rfslib.sftp_pconnection.**SftpPConnection**(***arg*)

Bases: *rfslib.abstract_pconnection.PConnection*

_isdir(*remote_path*)

Protected method which checks, whether a remote file is a directory.

Parameters **remote_path** – Path of a directory.

Returns True, if remote file is folder. False, if it isn't a folder. Undefined if the file doesns't exist.

_listdir(*remote_path*)

Protected method which returns a list of files in the folder.

Parameters **remote_path** – The remote path of a remote folder.

Returns List of files ([str]) in the remote folder

_pull(*remote_path*, *local_path*)

Protected method which downloads/pulls a file from a remote storage to a local storage in the binary form.

Parameters

- **remote_path** – Path of a remote file to download.
- **local_path** – Path of a local file, where to download/pull a remote file.

_push(*local_path*, *remote_path*)

Protected method which uploads/pushes a file from a local storage to a remote storage in the binary form.

Parameters

- **local_path** – Path of a local file to upload.
- **remote_path** – Path on the remote storage, where to upload/push a local file.

_rename(*old_name*, *new_name*)

Protected method which renames/moves a file.

Parameters

- **old_name** – Remote path a file to move.
- **new_name** – Remote path to which move the file.

close()

cp(*old_names*, *new_name*, *recursive=False*)

dcp(*old_names*, *target_dir*, *recursive=False*)

dmv(*old_names*, *target_dir*)

`exists`(*remote_path*)

`fcp`(*old_name*, *new_name*)

`find`(*remote_path*, *child_first=False*)

`fmv`(*old_name*, *new_name*)

`isdir`(*remote_path*)

`lexists`(*remote_path*)

`listdir`(*remote_path*)

`ls`(*remote_path*)

`mkdir`(*remote_path*)

`mv`(*old_names*, *new_name*)

`pmkdir`(*remote_path*)

`pull`(*remote_path*, *local_path*)

`push`(*local_path*, *remote_path*)

`rename`(*old_name*, *new_name*)

`rm`(*remote_path*, *recursive=False*)

`rmdir`(*remote_path*)

`rpull`(*remote_path*, *local_path*)

`rpush`(*local_path*, *remote_path*)

`touch`(*remote_path*)

`unlink`(*remote_path*)

`xls`(*remote_path*)

CHAPTER

THREE

RFSLIB.FTP_PCONNECTION MODULE

class rfslib.ftp_pconnection.**FtpPConnection**(***arg*)

Bases: *rfslib.abstract_pconnection.PConnection*

_isdir(*remote_path*)

Protected method which checks, whether a remote file is a directory.

Parameters **remote_path** – Path of a directory.

Returns True, if remote file is folder. False, if it isn't a folder. Undefined if the file doesns't exist.

_listdir(*remote_path*)

Protected method which returns a list of files in the folder.

Parameters **remote_path** – The remote path of a remote folder.

Returns List of files ([str]) in the remote folder

_pull(*remote_path*, *local_path*)

Protected method which downloads/pulls a file from a remote storage to a local storage in the binary form.

Parameters

- **remote_path** – Path of a remote file to download.
- **local_path** – Path of a local file, where to download/pull a remote file.

_push(*local_path*, *remote_path*)

Protected method which uploads/pushes a file from a local storage to a remote storage in the binary form.

Parameters

- **local_path** – Path of a local file to upload.
- **remote_path** – Path on the remote storage, where to upload/push a local file.

_rename(*old_name*, *new_name*)

Protected method which renames/moves a file.

Parameters

- **old_name** – Remote path a file to move.
- **new_name** – Remote path to which move the file.

close()

cp(*old_names*, *new_name*, *recursive=False*)

dcp(*old_names*, *target_dir*, *recursive=False*)

dmv(*old_names*, *target_dir*)

`exists`(*remote_path*)

`fcp`(*old_name*, *new_name*)

`find`(*remote_path*, *child_first=False*)

`fmv`(*old_name*, *new_name*)

`isdir`(*remote_path*)

`lexists`(*remote_path*)

`listdir`(*remote_path*)

`ls`(*remote_path*)

`mkdir`(*remote_path*)

`mv`(*old_names*, *new_name*)

`pmkdir`(*remote_path*)

`pull`(*remote_path*, *local_path*)

`push`(*local_path*, *remote_path*)

`rename`(*old_name*, *new_name*)

`rm`(*remote_path*, *recursive=False*)

`rmdir`(*remote_path*)

`rpull`(*remote_path*, *local_path*)

`rpush`(*local_path*, *remote_path*)

`touch`(*remote_path*)

`unlink`(*remote_path*)

`xls`(*remote_path*)

CHAPTER

FOUR

RFSLIB.SMB12_PCONNECTION MODULE

class rfslib.smb12_pconnection.**Smb12PConnection**(***arg*)

Bases: `rfslib.abstract_pconnection.PConnection`

_isdir(*remote_path*)

Protected method which checks, whether a remote file is a directory.

Parameters **remote_path** – Path of a directory.

Returns True, if remote file is folder. False, if it isn't a folder. Undefined if the file doesns't exist.

_listdir(*remote_path*)

Protected method which returns a list of files in the folder.

Parameters **remote_path** – The remote path of a remote folder.

Returns List of files ([str]) in the remote folder

_pull(*remote_path*, *local_path*)

Protected method which downloads/pulls a file from a remote storage to a local storage in the binary form.

Parameters

- **remote_path** – Path of a remote file to download.
- **local_path** – Path of a local file, where to download/pull a remote file.

_push(*local_path*, *remote_path*)

Protected method which uploads/pushes a file from a local storage to a remote storage in the binary form.

Parameters

- **local_path** – Path of a local file to upload.
- **remote_path** – Path on the remote storage, where to upload/push a local file.

_rename(*old_name*, *new_name*)

Protected method which renames/moves a file.

Parameters

- **old_name** – Remote path a file to move.
- **new_name** – Remote path to which move the file.

close()

cp(*old_names*, *new_name*, *recursive=False*)

dcp(*old_names*, *target_dir*, *recursive=False*)

dmv(*old_names*, *target_dir*)

exists(*remote_path*)

fcp(*old_name*, *new_name*)

find(*remote_path*, *child_first=False*)

fmv(*old_name*, *new_name*)

isdir(*remote_path*)

lexists(*remote_path*)

listdir(*remote_path*)

ls(*remote_path*)

mkdir(*remote_path*)

mv(*old_names*, *new_name*)

pmkdir(*remote_path*)

pull(*remote_path*, *local_path*)

push(*local_path*, *remote_path*)

rename(*old_name*, *new_name*)

rm(*remote_path*, *recursive=False*)

rmdir(*remote_path*)

rpull(*remote_path*, *local_path*)

rpush(*local_path*, *remote_path*)

touch(*remote_path*)

unlink(*remote_path*)

xls(*remote_path*)

CHAPTER

FIVE

RFSLIB.SMB23_PCONNECTION MODULE

class rfslib.smb23_pconnection.**Smb23PConnection**(***arg*)

Bases: `rfslib.abstract_pconnection.PConnection`

_isdir(*remote_path*)

Protected method which checks, whether a remote file is a directory.

Parameters **remote_path** – Path of a directory.

Returns True, if remote file is folder. False, if it isn't a folder. Undefined if the file doesns't exist.

_listdir(*remote_path*)

Protected method which returns a list of files in the folder.

Parameters **remote_path** – The remote path of a remote folder.

Returns List of files ([str]) in the remote folder

_pull(*remote_path*, *local_path*)

Protected method which downloads/pulls a file from a remote storage to a local storage in the binary form.

Parameters

- **remote_path** – Path of a remote file to download.
- **local_path** – Path of a local file, where to download/pull a remote file.

_push(*local_path*, *remote_path*)

Protected method which uploads/pushes a file from a local storage to a remote storage in the binary form.

Parameters

- **local_path** – Path of a local file to upload.
- **remote_path** – Path on the remote storage, where to upload/push a local file.

_rename(*old_name*, *new_name*)

Protected method which renames/moves a file.

Parameters

- **old_name** – Remote path a file to move.
- **new_name** – Remote path to which move the file.

close()

cp(*old_names*, *new_name*, *recursive=False*)

dcp(*old_names*, *target_dir*, *recursive=False*)

dmv(*old_names*, *target_dir*)

exists(*remote_path*)

fcp(*old_name*, *new_name*)

find(*remote_path*, *child_first=False*)

fmv(*old_name*, *new_name*)

isdir(*remote_path*)

lexists(*remote_path*)

listdir(*remote_path*)

ls(*remote_path*)

mkdir(*remote_path*)

mv(*old_names*, *new_name*)

pmkdir(*remote_path*)

pull(*remote_path*, *local_path*)

push(*local_path*, *remote_path*)

rename(*old_name*, *new_name*)

rm(*remote_path*, *recursive=False*)

rmdir(*remote_path*)

rpull(*remote_path*, *local_path*)

rpush(*local_path*, *remote_path*)

touch(*remote_path*)

unlink(*remote_path*)

xls(*remote_path*)

rfslib.smb23_pconnection.**config_smb23**(***arg*)

CHAPTER

SIX

RFSLIB.FS_PCONNECTION MODULE

class rfslib.fs_pconnection.**FsPConnection**(***arg*)

Bases: `rfslib.abstract_pconnection.PConnection`

_isdir(*remote_path*)

Protected method which checks, whether a remote file is a directory.

Parameters **remote_path** – Path of a directory.

Returns True, if remote file is folder. False, if it isn't a folder. Undefined if the file doesns't exist.

_listdir(*remote_path*)

Protected method which returns a list of files in the folder.

Parameters **remote_path** – The remote path of a remote folder.

Returns List of files ([str]) in the remote folder

_pull(*remote_path*, *local_path*)

Protected method which downloads/pulls a file from a remote storage to a local storage in the binary form.

Parameters

- **remote_path** – Path of a remote file to download.
- **local_path** – Path of a local file, where to download/pull a remote file.

_push(*local_path*, *remote_path*)

Protected method which uploads/pushes a file from a local storage to a remote storage in the binary form.

Parameters

- **local_path** – Path of a local file to upload.
- **remote_path** – Path on the remote storage, where to upload/push a local file.

_rename(*old_name*, *new_name*)

Protected method which renames/moves a file.

Parameters

- **old_name** – Remote path a file to move.
- **new_name** – Remote path to which move the file.

close()

cp(*old_names*, *new_name*, *recursive=False*)

dcp(*old_names*, *target_dir*, *recursive=False*)

dmv(*old_names*, *target_dir*)

exists(*remote_path*)

fcp(*old_name*, *new_name*)

find(*remote_path*, *child_first=False*)

fmv(*old_name*, *new_name*)

isdir(*remote_path*)

lexists(*remote_path*)

listdir(*remote_path*)

ls(*remote_path*)

mkdir(*remote_path*)

mv(*old_names*, *new_name*)

pmkdir(*remote_path*)

pull(*remote_path*, *local_path*)

push(*local_path*, *remote_path*)

rename(*old_name*, *new_name*)

rm(*remote_path*, *recursive=False*)

rmdir(*remote_path*)

rpull(*remote_path*, *local_path*)

rpush(*local_path*, *remote_path*)

touch(*remote_path*)

unlink(*remote_path*)

xls(*remote_path*)

CHAPTER

SEVEN

RFSLIB.PATH_UTILS MODULE

class rfslib.path_utils.**GenericPath**(*path*)

Bases: object

rfslib.path_utils.**add_r_prefix**(*path*)

rfslib.path_utils.**generic_cp**(*conn*, *sources*, *dest*, *recursive=False*)

rfslib.path_utils.**generic_mv**(*conn*, *sources*, *dest*)

rfslib.path_utils.**generic_path_normalize**(*path*)

rfslib.path_utils.**is_remote**(*path*)

rfslib.path_utils.**path_normalize**(*path*)

rfslib.path_utils.**remove_r_prefix**(*path*)

PYTHON MODULE INDEX

r

rfslib.abstract_pconnection, 3
rfslib.fs_pconnection, 13
rfslib.ftp_pconnection, 7
rfslib.path_utils, 15
rfslib.sftp_pconnection, 5
rfslib.smb12_pconnection, 9
rfslib.smb23_pconnection, 11

INDEX

Symbols

_isdir() (*rfslib.abstract_pconnection.PConnection method*), 3
_isdir() (*rfslib.fs_pconnection.FsPConnection method*), 13
_isdir() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
_isdir() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
_isdir() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
_isdir() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11
_listdir() (*rfslib.abstract_pconnection.PConnection method*), 3
_listdir() (*rfslib.fs_pconnection.FsPConnection method*), 13
_listdir() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
_listdir() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
_listdir() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
_listdir() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11
_pull() (*rfslib.abstract_pconnection.PConnection method*), 3
_pull() (*rfslib.fs_pconnection.FsPConnection method*), 13
_pull() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
_pull() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
_pull() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
_pull() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11
_push() (*rfslib.abstract_pconnection.PConnection method*), 3
_push() (*rfslib.fs_pconnection.FsPConnection method*), 13
_push() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
_push() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
_push() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
_push() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11
_rename() (*rfslib.abstract_pconnection.PConnection method*), 3
_rename() (*rfslib.fs_pconnection.FsPConnection method*), 13
_rename() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
_rename() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
_rename() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
_rename() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11

A

add_r_prefix() (*in module rfslib.path_utils*), 15

C

close() (*rfslib.abstract_pconnection.PConnection method*), 3
close() (*rfslib.fs_pconnection.FsPConnection method*), 13
close() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
close() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
close() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
close() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11
config_smb23() (*in module rfslib.smb23_pconnection*), 12
cp() (*rfslib.abstract_pconnection.PConnection method*), 3
cp() (*rfslib.fs_pconnection.FsPConnection method*), 13
cp() (*rfslib.ftp_pconnection.FtpPConnection method*), 7

cp() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
cp() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
cp() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11

D

dcp() (*rfslib.abstract_pconnection.PConnection method*), 3
dcp() (*rfslib.fs_pconnection.FsPConnection method*), 13
dcp() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
dcp() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
dcp() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
dcp() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11
dmv() (*rfslib.abstract_pconnection.PConnection method*), 3
dmv() (*rfslib.fs_pconnection.FsPConnection method*), 13
dmv() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
dmv() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
dmv() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
dmv() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11

E

exists() (*rfslib.abstract_pconnection.PConnection method*), 3
exists() (*rfslib.fs_pconnection.FsPConnection method*), 13
exists() (*rfslib.ftp_pconnection.FtpPConnection method*), 7
exists() (*rfslib.sftp_pconnection.SftpPConnection method*), 5
exists() (*rfslib.smb12_pconnection.Smb12PConnection method*), 9
exists() (*rfslib.smb23_pconnection.Smb23PConnection method*), 11

F

fcp() (*rfslib.abstract_pconnection.PConnection method*), 4
fcp() (*rfslib.fs_pconnection.FsPConnection method*), 14
fcp() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
fcp() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
fcp() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
fcp() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
find() (*rfslib.abstract_pconnection.PConnection method*), 4
find() (*rfslib.fs_pconnection.FsPConnection method*), 14
find() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
find() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
find() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
find() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
fmv() (*rfslib.abstract_pconnection.PConnection method*), 4
fmv() (*rfslib.fs_pconnection.FsPConnection method*), 14
fmv() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
fmv() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
fmv() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
fmv() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
FsPConnection (*class in rfslib.fs_pconnection*), 13
FtpPConnection (*class in rfslib.ftp_pconnection*), 7

G

generic_cp() (*in module rfslib.path_utils*), 15
generic_mv() (*in module rfslib.path_utils*), 15
generic_path_normalize() (*in module rfslib.path_utils*), 15
GenericPath (*class in rfslib.path_utils*), 15

I

is_remote() (*in module rfslib.path_utils*), 15
isdir() (*rfslib.abstract_pconnection.PConnection method*), 4
isdir() (*rfslib.fs_pconnection.FsPConnection method*), 14
isdir() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
isdir() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
isdir() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
isdir() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12

L

lexists() (*rfslib.abstract_pconnection.PConnection*

method), 4
lexists() (*rfslib.fs_pconnection.FsPConnection method*), 14
lexists() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
lexists() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
lexists() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
lexists() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
listdir() (*rfslib.abstract_pconnection.PConnection method*), 4
listdir() (*rfslib.fs_pconnection.FsPConnection method*), 14
listdir() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
listdir() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
listdir() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
listdir() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
ls() (*rfslib.abstract_pconnection.PConnection method*), 4
ls() (*rfslib.fs_pconnection.FsPConnection method*), 14
ls() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
ls() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
ls() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
ls() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12

M

mkdir() (*rfslib.abstract_pconnection.PConnection method*), 4
mkdir() (*rfslib.fs_pconnection.FsPConnection method*), 14
mkdir() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
mkdir() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
mkdir() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
mkdir() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
module
    rfslib.abstract_pconnection, 3
    rfslib.fs_pconnection, 13
    rfslib.ftp_pconnection, 7
    rfslib.path_utils, 15
    rfslib.sftp_pconnection, 5
    rfslib.smb12_pconnection, 9
    rfslib.smb23_pconnection, 11
mv() (*rfslib.abstract_pconnection.PConnection method*), 4
mv() (*rfslib.fs_pconnection.FsPConnection method*), 14
mv() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
mv() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
mv() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
mv() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12

P

path_normalize() (*in module rfslib.path_utils*), 15
PConnection (*class in rfslib.abstract_pconnection*), 3
pmkdir() (*rfslib.abstract_pconnection.PConnection method*), 4
pmkdir() (*rfslib.fs_pconnection.FsPConnection method*), 14
pmkdir() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
pmkdir() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
pmkdir() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
pmkdir() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
pull() (*rfslib.abstract_pconnection.PConnection method*), 4
pull() (*rfslib.fs_pconnection.FsPConnection method*), 14
pull() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
pull() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
pull() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
pull() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
push() (*rfslib.abstract_pconnection.PConnection method*), 4
push() (*rfslib.fs_pconnection.FsPConnection method*), 14
push() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
push() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
push() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
push() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12

R

remove_r_prefix() (*in module rfslib.path_utils*), 15

rename() (*rfslib.abstract_pconnection.PConnection method*), 4
rename() (*rfslib.fs_pconnection.FsPConnection method*), 14
rename() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
rename() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
rename() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
rename() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
rfslib.abstract_pconnection
    module, 3
rfslib.fs_pconnection
    module, 13
rfslib.ftp_pconnection
    module, 7
rfslib.path_utils
    module, 15
rfslib.sftp_pconnection
    module, 5
rfslib.smb12_pconnection
    module, 9
rfslib.smb23_pconnection
    module, 11
rm() (*rfslib.abstract_pconnection.PConnection method*), 4
rm() (*rfslib.fs_pconnection.FsPConnection method*), 14
rm() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
rm() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
rm() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
rm() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
rmdir() (*rfslib.abstract_pconnection.PConnection method*), 4
rmdir() (*rfslib.fs_pconnection.FsPConnection method*), 14
rmdir() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
rmdir() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
rmdir() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
rmdir() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
rpull() (*rfslib.abstract_pconnection.PConnection method*), 4
rpull() (*rfslib.fs_pconnection.FsPConnection method*), 14
rpull() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
rpull() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
rpull() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
rpull() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12
rpush() (*rfslib.abstract_pconnection.PConnection method*), 4
rpush() (*rfslib.fs_pconnection.FsPConnection method*), 14
rpush() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
rpush() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
rpush() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
rpush() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12

S

SftpPConnection (*class in rfslib.sftp_pconnection*), 5
Smb12PConnection (*class in rfslib.smb12_pconnection*), 9
Smb23PConnection (*class in rfslib.smb23_pconnection*), 11

T

touch() (*rfslib.abstract_pconnection.PConnection method*), 4
touch() (*rfslib.fs_pconnection.FsPConnection method*), 14
touch() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
touch() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
touch() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
touch() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12

U

unlink() (*rfslib.abstract_pconnection.PConnection method*), 4
unlink() (*rfslib.fs_pconnection.FsPConnection method*), 14
unlink() (*rfslib.ftp_pconnection.FtpPConnection method*), 8
unlink() (*rfslib.sftp_pconnection.SftpPConnection method*), 6
unlink() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10
unlink() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12

X

xls() (*rfslib.abstract_pconnection.PConnection method*), 4

xls() (*rfslib.fs_pconnection.FsPConnection method*), 14

xls() (*rfslib.ftp_pconnection.FtpPConnection method*), 8

xls() (*rfslib.sftp_pconnection.SftpPConnection method*), 6

xls() (*rfslib.smb12_pconnection.Smb12PConnection method*), 10

xls() (*rfslib.smb23_pconnection.Smb23PConnection method*), 12