

# Package ‘OhdsiSharing’

March 25, 2020

**Type** Package

**Title** Package for sharing of the results of the OHDSI tools

**Version** 0.2.1

**Date** 2020-03-25

**Maintainer** Martijn Schuemie <schuemie@ohdsi.org>

**Description** Package for sharing of the results of the OHDSI tools, with functions for encrypting results and sending results through SFTP to a central site.

**Imports** rJava,  
ParallelLogger

**Suggests** testthat

**License** Apache License

**RoxygenNote** 7.0.2

## R topics documented:

compressAndEncryptFolder . . . . .	2
compressFolder . . . . .	2
decompressFolder . . . . .	3
decryptAndDecompressFolder . . . . .	4
decryptFile . . . . .	5
encryptFile . . . . .	5
generateKeyPair . . . . .	6
OhdsiSharing . . . . .	6
sftpCd . . . . .	7
sftpConnect . . . . .	7
sftpDisconnect . . . . .	7
sftpGetFiles . . . . .	8
sftpLs . . . . .	8
sftpMkdir . . . . .	9
sftpPutFile . . . . .	9
sftpRename . . . . .	10
sftpRm . . . . .	10
sftpRmdir . . . . .	10
sftpUploadFile . . . . .	11
sftpPwd . . . . .	11
<b>Index</b>	<b>12</b>

---

`compressAndEncryptFolder`*Compress and encrypt a folder*

---

**Description**

Compress and encrypt a folder

**Usage**

```
compressAndEncryptFolder(sourceFolder, targetFileName, publicKeyFileName)
```

**Arguments**

`sourceFolder`    Name of the folder that must be encrypted.  
`targetFileName`   Name of the file that will hold the encrypted data.  
`publicKeyFileName`  
                    Name of the file where the public key is stored.

**Details**

Compresses all files in a folder and its subfolders, and encrypts using the provided public key.

**Examples**

```
## Not run:  
generateKeyPair("public.key", "private.key")  
  
# Create a folder with some data  
dir.create("test")  
data <- data.frame(x = runif(1000), y = 1:1000)  
saveRDS(data, "test/data1.rds")  
saveRDS(data, "test/data2.rds")  
  
compressAndEncryptFolder("test", "data.zip.enc", "public.key")  
decryptAndDecompressFolder("data.zip.enc", "test2", "private.key")  
  
## End(Not run)
```

---

`compressFolder`*Compress a folder*

---

**Description**

Compress a folder

**Usage**

```
compressFolder(sourceFolder, targetFileName)
```

**Arguments**

sourceFolder    Name of the folder that must be compressed.  
targetFileName    Name of the file that will hold the compressed data.

**Details**

Compresses all files in a folder and its subfolders, and stores it in a single zip file.

**Examples**

```
## Not run:  
# Create a folder with some data  
dir.create("test")  
data <- data.frame(x = runif(1000), y = 1:1000)  
saveRDS(data, "test/data1.rds")  
saveRDS(data, "test/data2.rds")  
  
compressFolder("test", "data.zip")  
decompressFolder("data.zip", "test2")  
  
## End(Not run)
```

---

decompressFolder	<i>Decompress a folder</i>
------------------	----------------------------

---

**Description**

Decompress a folder

**Usage**

```
decompressFolder(sourceFileName, targetFolder)
```

**Arguments**

sourceFileName    Name of the file that must be decompressed.  
targetFolder    Name of the folder that will hold the extracted data.

**Details**

Extracts all compressed files to a folder.

**Examples**

```
## Not run:  
# Create a folder with some data  
dir.create("test")  
data <- data.frame(x = runif(1000), y = 1:1000)  
saveRDS(data, "test/data1.rds")  
saveRDS(data, "test/data2.rds")  
  
compressFolder("test", "data.zip")
```

```
decompressFolder("data.zip", "test2")  
  
## End(Not run)
```

---

decryptAndDecompressFolder

*Decrypt and decompress a folder*

---

## Description

Decrypt and decompress a folder

## Usage

```
decryptAndDecompressFolder(sourceFileName, targetFolder, privateKeyFileName)
```

## Arguments

sourceFileName    Name of the file that must be decrypted.  
targetFolder      Name of the folder that will hold the unencrypted data.  
privateKeyFileName  
                    Name of the file where the private key is stored.

## Details

Decrypts the data using the provided private key and extracts all files to a folder.

## Examples

```
## Not run:  
generateKeyPair("public.key", "private.key")  
  
# Create a folder with some data  
dir.create("test")  
data <- data.frame(x = runif(1000), y = 1:1000)  
saveRDS(data, "test/data1.rds")  
saveRDS(data, "test/data2.rds")  
  
compressAndEncryptFolder("test", "data.zip.enc", "public.key")  
decryptAndDecompressFolder("data.zip.enc", "test2", "private.key")  
  
## End(Not run)
```

---

decryptFile	<i>Decrypt a data file</i>
-------------	----------------------------

---

**Description**

Decrypt a data file

**Usage**

```
decryptFile(sourceFileName, targetFileName, privateKeyFileName)
```

**Arguments**

sourceFileName Name of the file that must be decrypted.  
targetFileName Name of the file that will hold the unencrypted data.  
privateKeyFileName  
Name of the file where the private key is stored.

**Details**

Decrypts the data using the provided private key.

**Examples**

```
## Not run:  
generateKeyPair("public.key", "private.key")  
data <- data.frame(x = runif(1000), y = 1:1000)  
saveRDS(data, "data.rds")  
encryptFile("data.rds", "data.rds.enc", "public.key")  
decryptFile("data.rds.enc", "data2.rds", "private.key")  
  
## End(Not run)
```

---

encryptFile	<i>Encrypt a data file</i>
-------------	----------------------------

---

**Description**

Encrypt a data file

**Usage**

```
encryptFile(sourceFileName, targetFileName, publicKeyFileName)
```

**Arguments**

sourceFileName Name of the file that must be encrypted.  
targetFileName Name of the file that will hold the encrypted data.  
publicKeyFileName  
Name of the file where the public key is stored.

**Details**

Encrypts the data using the provided public key.

**Examples**

```
## Not run:
generateKeyPair("public.key", "private.key")
data <- data.frame(x = runif(1000), y = 1:1000)
saveRDS(data, "data.rds")
encryptFile("data.rds", "data.rds.enc", "public.key")

## End(Not run)
```

---

generateKeyPair	Create a public-private key pair
-----------------	----------------------------------

---

**Description**

Create a public-private key pair

**Usage**

```
generateKeyPair(publicKeyFileName, privateKeyFileName)
```

**Arguments**

```
publicKeyFileName
      Name of the file where the public key should be stored.
privateKeyFileName
      Name of the file where the private key should be stored.
```

**Details**

Creates an RSA 4096-bit public-private key pair. The public key can be used to encrypt data, and only with the private key can the data be decrypted.

**Examples**

```
## Not run:
generateKeyPair("public.key", "private.key")

## End(Not run)
```

---

OhdsiSharing	OhdsiSharing
--------------	--------------

---

**Description**

OhdsiSharing

---

sftpCd	<i>Change the current working director</i>
--------	--

---

**Description**

Change the current working director

**Usage**

```
sftpCd(sftpConnection, remoteFolder)
```

**Arguments**

sftpConnection	An SftpConnection object as created by the <a href="#">sftpConnect</a> function.
remoteFolder	The folder on the server to change to.

---

---

sftpConnect	<i>Connect to the OHDSI SFTP server</i>
-------------	---

---

**Description**

Connect to the OHDSI SFTP server

**Usage**

```
sftpConnect(privateKeyFileName, userName)
```

**Arguments**

privateKeyFileName	
	A character string denoting the path to an RSA private key.
userName	
	A character string containing the user name.

**Value**

An SftpConnection object

---

---

sftpDisconnect	<i>Disconnect from the OHDSI SFTP server.</i>
----------------	---

---

**Description**

Disconnect from the OHDSI SFTP server.

**Usage**

```
sftpDisconnect(sftpConnection)
```

**Arguments**

sftpConnection	An SftpConnection object as created by the <a href="#">sftpConnect</a> function.
----------------	--

---

sftpGetFiles	<i>Get one or more files from the SFTP server</i>
--------------	---

---

### Description

Get one or more files from the SFTP server

### Usage

```
sftpGetFiles(
  sftpConnection,
  remoteFileNames,
  localFolder = getwd(),
  localFileNames = file.path(localFolder, remoteFileNames)
)
```

### Arguments

sftpConnection	An SftpConnection object as created by the <a href="#">sftpConnect</a> function.
remoteFileNames	The name of the file(s) to get from the server.
localFolder	The path of a local folder where all files will be stored. Is ignored if localFileNames is provided.
localFileNames	The name the file(s) should have locally. If not provided, the files will be given the same names as on the server.

---

sftpLs	<i>List the files in folder on the server.</i>
--------	--

---

### Description

List the files in folder on the server.

### Usage

```
sftpLs(sftpConnection, remoteFolder = "./*")
```

### Arguments

sftpConnection	An SftpConnection object as created by the <a href="#">sftpConnect</a> function.
remoteFolder	The folder on the server. Defaults to the current folder.

### Value

A data frame with two columns: the file names, and the file types (directory, link, or file).



---

sftpMkdir	<i>Make a directory</i>
-----------	-------------------------

---

**Description**

Make a directory

**Usage**

```
sftpMkdir(sftpConnection, remoteFolder)
```

**Arguments**

`sftpConnection` An SftpConnection object as created by the [sftpConnect](#) function.  
`remoteFolder` The folder on the server to create.

---

sftpPutFile	<i>Put a file on the SFTP server</i>
-------------	--------------------------------------

---

**Description**

Put a file on the SFTP server

**Usage**

```
sftpPutFile(  
  sftpConnection,  
  localFileName,  
  remoteFileName = basename(localFileName)  
)
```

**Arguments**

`sftpConnection` An SftpConnection object as created by the [sftpConnect](#) function.  
`localFileName` The path to the local file to upload.  
`remoteFileName` The name the file should have on the server.

---

sftpRename	<i>Rename a file or folder</i>
------------	--------------------------------

---

**Description**

Rename a file or folder

**Usage**

```
sftpRename(sftpConnection, oldRemoteFilename, newRemoteFilename)
```

**Arguments**

sftpConnection An SftpConnection object as created by the [sftpConnect](#) function.

oldRemoteFilename

The file on the server to rename.

newRemoteFilename

The new file name.

---

sftpRm	<i>Remove one or more files</i>
--------	---------------------------------

---

**Description**

Remove one or more files

**Usage**

```
sftpRm(sftpConnection, remoteFiles)
```

**Arguments**

sftpConnection An SftpConnection object as created by the [sftpConnect](#) function.

remoteFiles The file(s) on the server to remove.

---

sftpRmdir	<i>Remove a directory</i>
-----------	---------------------------

---

**Description**

Remove a directory

**Usage**

```
sftpRmdir(sftpConnection, remoteFolder)
```

**Arguments**

sftpConnection An SftpConnection object as created by the [sftpConnect](#) function.

remoteFolder The folder on the server to remove.

---

sftpUploadFile	<i>Upload a single file to the OHDSI SFTP server</i>
----------------	--

---

**Description**

This function combines calls to the [sftpConnect](#), [sftpPutFile](#), and [sftpDisconnect](#) functions. A random string will be prefixed to the file name to prevent overwriting existing files on the server.

**Usage**

```
sftpUploadFile(privateKeyFileName, userName, remoteFolder = ".", fileName)
```

**Arguments**

privateKeyFileName	A character string denoting the path to an RSA private key.
userName	A character string containing the user name.
remoteFolder	The remote folder to upload the file to.
fileName	A character string denoting the path to file to upload.

---

sftpPwd	<i>Get the present working directory</i>
---------	--

---

**Description**

Get the present working directory

**Usage**

```
sftpPwd(sftpConnection)
```

**Arguments**

`sftpConnection` An SftpConnection object as created by the [sftpConnect](#) function.

**Value**

A character string representing the current remote folder name.

# Index

compressAndEncryptFolder, [2](#)  
compressFolder, [2](#)  
  
decompressFolder, [3](#)  
decryptAndDecompressFolder, [4](#)  
decryptFile, [5](#)  
  
encryptFile, [5](#)  
  
generateKeyPair, [6](#)  
  
OhdsiSharing, [6](#)  
  
sftpCd, [7](#)  
sftpConnect, [7](#), [7](#), [8–11](#)  
sftpDisconnect, [7](#), [11](#)  
sftpGetFiles, [8](#)  
sftpLs, [8](#)  
sftpMkdir, [9](#)  
sftpPutFile, [9](#), [11](#)  
sftpRename, [10](#)  
sftpRm, [10](#)  
sftpRmdir, [10](#)  
sftpUploadFile, [11](#)  
sftpPwd, [11](#)