Package 'OhdsiSharing'

May 6, 2016

Type Package

Title Package for sharing of the results of the OHDSI tools

Version 0.1.0

Date 2016-05-06

Author Martijn Schuemie [aut, cre],
Marc Suchard [aut]

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 FTP and S3 to a central site.

Imports rJava,
aws.signature,
httr,
XML,
uuid,
xml2,
base64enc,
digest,
curl

Suggests OhdsiRTools

License Apache License

RoxygenNote 5.0.1

R topics documented:

compressAndEncryptFolder 2
compressFolder 2
decompressFolder 3
decryptAndDecompressFolder 4
decryptFile 5
encryptFile 5
generateKeyPair 6
putS3File 7

Index **8**

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 *Decompress a folder*

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 *Decrypt a data file*

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 *Encrypt a data file*

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)
```

putS3File *Put a local file into a remote S3 bucket*

Description

Put a local file into a remote S3 bucket

Usage

```
putS3File(file, bucket, region = "us-east-1", key, secret,
  appendUUID = TRUE)
```

Arguments

file	The path to the file in the local filesystem.
bucket	The name of the bucket to put the file in.
region	The region of the S3.
key	Your AWS access key.
secret	Your AWS secret access key.
appendUUID	Should append an universally unique identifier to file name when uploading?

Index

compressAndEncryptFolder, 2
compressFolder, 2

decompressFolder, 3
decryptAndDecompressFolder, 4
decryptFile, 5

encryptFile, 5

generateKeyPair, 6

putS3File, 7