Package 'seekr'

April 7, 2020

Type Package

Title seekr: R interface for SEEK API (connection to FAIRDONHub)

Version 0.1.0.9000

Description The package provides several functions for support and use of pISA-tree.

License GPL-3

URL https://github.com/NIB-SI/seekr

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports knitr,
rio,
tools,
RCurl,
httr,
jsonlite

VignetteBuilder knitr

Suggests knitr,
rmarkdown

R topics documented:

`contentType` *Determine MIME type for file.*

Description

Determine MIME type for file.

Usage

```
contentType(x)
```

Arguments

`File`	name.

Value

MIME type string.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
contentType("bla.txt")
contentType("bla.pdf")
contentType("bla.tar.gz")
contentType("bla.bla")
```

fhCreate *Create pISA layer or *fh* component.*

Description

Create pISA layer or *fh* component.

Usage

```
fhCreate(type = "assays", meta)
```

Arguments

type	Component name (e.g. 'people', 'projets', ...).
meta	Data frame with pISA metadata or a list with minimal information (Title, Description, *ToDo: add fields*).

Value

FAIRDOMhub created component.

Note

Upon success (status code 200) details of newly created component can be used. Check status code.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

fhGet

Examples

```
\donotrun{
fhLog("Start testing log", append=FALSE)
fhTime()
if(FALSE)
{
fhIni(prid = 26, test=TRUE)
options("fhub")
 sp <- fhCreate( type = "projects"
  , meta= list(
      Title=paste("Test project", Sys.time())
    , Description="Testing of upload")
    )
 str(sp)
 str(fhData(sp))
# Add member manually
 fhIni(prid = 26, pid=104, test=TRUE)
 options("fhub")
 si <- fhCreate( type = "investigations"
```

```
 , meta= list(
     Title=paste("Test investigation", Sys.time())
   , Description="Testing of upload")
   )
si
fhData(si)$id
}
iid=fhData(si)$id
iid <- 115
fhIni(prid = 26, pid=104, iid=iid, test=TRUE)
options("fhub")
ss <- fhCreate( type = "studies"
 , meta= list(
     Title=paste("Test study", Sys.time())
   , Description="Testing of upload")
   )
ss
fhData(ss)$id

fhIni(prid = 26
      , pid=104
      , iid=fhData(si)$id
      , sid=fhData(ss)$id
      , test=TRUE)
options("fhub")
sa <- fhCreate( type = "assays"
 , meta= list(
     Title=paste("Test assay", Sys.time())
   , Description="Testing of upload")
   )
#str(sa)
sa
fhData(sa)$id
}
```

`fhData` *Get content from an *fh* object.*

Description

Get content from an *fh* object.

Usage

```
fhData(r, node, ...)
```

Arguments

`r`	Object retrieved by fhGet.
`type`	Name of the required element. If missing, a list with all relevant objects is returned.

Value

File name (string).

Note

Parameter ... is ignored at this time.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

get ...

Examples

```
\donotrun{
fhIni()
options()$fhub$myid
r <- fhGet("people",options()$fhub$myid)
d <- fhData(r,"attributes")
names(d)
d$last_name
fhData(r)$tools
# Get list of people
r <- fhGet("people")
d <- fhData(r)
length(d)
names(d)
names(d[[1]])
titles <- sapply(d,function(x) x$attributes$title)
head(titles)
# Get FAIRDOMhub user id
myname <- titles[1]
myname
d[[pmatch(myname,titles)]]
id <- d[[pmatch(myname,titles)]]$id
id
}
```

---

fhFindId — *Get details of component with id from an *fh* object.*

Description

Get details of component with id from an *fh* object.

Usage

```
fhFindId(type, title)
```

Arguments

type	Components name (e.g. 'people', 'projets', ...).
title	Character string with the identifier of the component (title part).

Value

FAIRDOMhub component identifier: id, type and title. If argument title is missing, a data frame with identifiers for all items is returned.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

fhFindTitle

Examples

```
\donotrun{
fhIni()
id <- fhFindId("people","Guest")
id
fhFindTitle("people",id)
# List of projects
projects <- fhFindId("projects")
head(projects)
}
```

`fhFindTitle` *Get details of component with id from an *fh* object.*

Description

Get details of component with id from an *fh* object.

Usage

```
fhFindTitle(type, id)
```

Arguments

`type`	Components name (e.g. 'people', 'projets', ...).
`id`	Character string with the identifier of the component (id part).

Value

FAIRDOMhub component identifier: id, type and title. If argument title is missing, a data frame with identifiers for all items is returned.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

fhFindTitle

Examples

```
\donotrun{
fhIni()
id <- fhFindId("people","Guest")
id
idNew <- fhFindTitle("people", id[1])
idNew
all.equal(id, idNew)
}
```

---

fhGet *Get inormation from repository.*

Description

Get inormation from repository.

Usage

```
fhGet(type, id, uri = options()$fhub$baseurl, ...)
```

Arguments

type	Type of information (e.g. "person").
id	Repository id of an item.
uri	Repository base address (URI)..

Value

An object (list) of class seek_api.

Note

Parameter ... is ignored at this time.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

get ...

Examples

```
\donotrun{
fhIni()
options()$fhub$myid
r <- fhGet("people",options()$fhub$myid)
names(r)
r$response$status_code
status_code(r$response)
r
}
```

fhIni *Initialize FAIRDOMhub information*

Description

Define FAIRDOMhub URL and user data

Usage

```
fhIni(prid = NULL, pid = NULL, iid = NULL, sid = NULL,
  aid = NULL, test = TRUE)
```

Arguments

test	If TRUE, test server will be used..

Value

A list with URL and user information. For side effect see Notes.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
fhIni()
options("fhub")
#
fhIni(test=FALSE)
options("fhub")

## End(Not run)
```

fhLog *Writes a note to a log file.*

Description

Writes a note to a log file.

Usage

```
fhLog(..., file = "FAIRDOM.log", append = TRUE)

fhTime(..., file = "FAIRDOM.log")
```

Arguments

...	Objects to form a line.
file	Log file name.
append	Control append/rewrite mode.

Value

System time of invoking..

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

fhGet

Examples

```
tst <- function(){
   t0 <- fhLog("Test", "writing to logfile", file="")
   Sys.sleep(1)
   fhTime(123,file="")
}
tst()
rm(tst)
```

---

fhParse *Parse the response from SEEK API*

Description

Parse the response from SEEK API

Usage

```
fhParse(resp, ...)
```

Arguments

resp	Response from SEEK API.

Value

An object (list) of class seek_api.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

fhGget ...

Examples

```
\donotrun{
fhIni()
options()$fhub$myid
r <- fhGet("people",options()$fhub$myid)
names(r)
r$response$status_code
status_code(r$response)
names(r$content)
r
r <- fhGet("people")
length(r$content)
names(r$content)
r$content[[1]]
r
}
```

---

fhSkeleton *Create *fh* skeleton.*

Description

Creates *fh* object with required structure.

Usage

```
fhSkeleton(type = "assay", meta)
```

Arguments

type	Component name (e.g. 'people', 'projets', ...).
meta	Data frame with pISA metadata or a list with minimal information (Title, Description, *ToDo: add fields*).

Value

A list with the minimal information structure.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

fhCreate

`fileName` *Extract file name*

Description

Extract file name from a file path.

Usage

```
fileName(x, ...)
```

Arguments

`x`	Complete file path or file.type name.
`...`	Any other arguments.

Value

File name (string).

Note

Parameter ... is ignored at this time.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

`fileType`

Examples

```
fileName(".\\validation.Rnw")
fileName("./bla/validation.Rnw")
fileName("./validation.")
fileName("./validation")
```

`fileType` *Extract file type*

Description

Extract file type from a file path.

Usage

```
fileType(x, ...)
```

Arguments

x	Complete file path or file name.
...	Any other arguments.

Value

File type (string).

Note

Parameter ... is ignored at this time.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
fileName(".\\validation.Rnw")
fileName("./bla/validation.Rnw")
fileName("./validation.")
fileName("./validation")
```

---

fsummary *As factor summary of a data frame*

Description

As factor summary of a data frame

Usage

```
fsummary(x, ...)
```

Arguments

x	Data frame.
...	Any other arguments.

Value

Summary object.

Note

Argument ... not used

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
fsummary(data.frame(x=rnorm(20),txt=sample(letters,20,rep=TRUE)))
```

getLayer *Get pISA layer name*

Description

Get pISA layer name

Usage

```
getLayer(x, path = getwd())
```

Arguments

x	Layer character (one of p, I, S, or A).
path	Directory path, defaults to working directory.

Value

Character string with layer name.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
astring <- "_p_Demo/_I_Test/_S_Show/_A_Work-R/other"
oldwd <- setwd(system.file("extdata",astring,package="pisar"))
oldwd
.pname <- getLayer("p")
.pname
getLayer("I")
getLayer("S")
getLayer("A")

## End(Not run)
```

getMeta *Get metadata value*

Description

Get metadata value

Usage

```
getMeta(x, ...)

## Default S3 method:
getMeta(x, item, nl = TRUE)

## S3 method for class 'list'
getMeta(x, ...)
```

Arguments

x	Two column character data frame with Key / Value pairs.
...	Any other arguments (not used at the moment).
item	String, item name.
nl	Logical, expand backslash character for new lines.

Value

Character string with key value.

Note

Parameter item is matched exactly to the item names.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
astring <- "_p_Demo/_I_Test/_S_Show/_A_Work-R/other"
oldwd <- setwd(system.file("extdata",astring,package="pisar"))
oldwd
.iroot <- getRoot("I")
.imeta <- readMeta(.iroot)
getMeta(.imeta, "Description")
# list
listmeta <- list(Title = "My title"
   , Description = "A longer description")
getMeta( listmeta, "Title")
getMeta( listmeta, "Description")
setwd(oldwd)

## End(Not run)
```

getRoot *Get root directory for pISA layer*

Description

Get root directory for pISA layer

Usage

```
getRoot(x = "p", path = getwd(), ...)
```

Arguments

x	Character characteristic for pISA layer (one of p, I, S, or A).
path	Path within the pISA-tree.
...	Any other arguments.

Value

Relative path to the layer directory (from working directory).

Note

The path should be compliant with the pISA-tree structure. Path defaults to the working directory, which is usually in or below an assay. Argument ... is not used.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
getRoot("p", path="d:/_p_prj/_I_inv/_S_st/_A_asy/other/doc")
```

out.path *Create output directory*

Description

Create output directory, name it with appended arguments.

Usage

```
out.path(out.dir = "../out", args = "", which = 1:length(args))
```

Arguments

out.dir	Character string, base output directory.
args	Character vector, arguments used for sub-analysis. or from a batch call
which	Numeric vector, which arguments to use.

Value

Directory name.

Note

Directory is created

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
out.path()
out.path(args="")
dir.exists("../out/xx-ena-dva")
out.path(args=c("xx.txt","ena","dva"))
dir.exists("../out/xx-ena-dva")
unlink("../out/xx-ena-dva",recursive=TRUE)
dir.exists("../out/xx-ena-dva")
```

---

pisa *Extract pISA-tree details*

Description

Extract pISA-tree details: name, root and metadata for all layers above the current directory (below or in the Assay).

Usage

```
pisa(path = getwd(), addArgs = NULL, global = TRUE)
```

Arguments

path	Path to part of pISA-tree, defaults to working directory.
addArgs	Character vector, additional arguments.
global	If TRUE (default) auxiliary objects will be created in the global environment (see note).

Value

A list with layer information components, possibly changing objects in the global environment (Se Note).

Note

If argument global is TRUE (default), auxiliary objects with pISA related information will be created in the global environment. Sometimes it is more convenient to use such object instead of the elements of the (invisibly) returned pisa list. The object are hidden (names start with dot); use `ls(pattern="^\.",all.names=TRUE)` to get a full list of hiddent objects. The created objects are

.[pisa name] layer name

.[pisa root] layer path (relative to the working directory)

.[pisa meta] data frame with corresponding layer metadata

.oroot output directory path

.inroot input (data) directory path

.reproot report directory path

.reproot report directory path

.outputFile output file path and name

.args a vector of additional arguments (possibly from a batch call)

.pfn phenodata file path and name (relative to Investigation)

.ffn featuredata fiel path and name

.outfn output file basename (no type)

.rnwfn knitr source file name (*.Rnw or *.Rmd)

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
astring <- "_p_Demo/_I_Test/_S_Show/_A_Work-R/other"
oldwd <- setwd(system.file("extdata",astring,package="pisar"))
oldwd
pisa <- pisa(global=FALSE)
str(pisa)
names(pisa)
dir(pisa$p$root)
# Set pisa options
options(pisa=pisa(global=FALSE))
# Access details with $
options()$pisa$p$root
setwd(oldwd)

## End(Not run)
```

`pisar` *pisar: pISA-tree support functions*

Description

The package provides several functions for support and use of pISA-tree.

print.pISAmeta *Print metadata object as Dlist*

Description

Print metadata object as Dlist

Usage

```
## S3 method for class 'pISAmeta'
print(x, width = max(nchar(x[, 1])) * 3.5, ...)
```

Arguments

x	Metadata object, data.frame with two columns.
width	Estimated text width.
...	Any other arguments.

Note

Metadata table is printed in convenient Dlist form.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
.pISAloc <- system.file("extdata","_p_Demo",package="pisar")
readMeta(.pISAloc)
```

print.seek_api *Print method for seek_api object*

Description

Print method for seek_api object

Print method for seek_api object

Usage

```
## S3 method for class 'seek_api'
print(x, content = FALSE)

## S3 method for class 'seek_api'
print(x, content = FALSE)
```

Arguments

x	Object of class seek_api.
content	If FALSE (default), content is no printed.
x	Object of class seek_api.
content	If FALSE (default), content is no printed.

Value

An object (list) of class seek_api.

An object (list) of class seek_api.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Andrej Blejec <andrej.blejec@nib.si>

See Also

fhParse ...

skParse ...

Examples

```
\donotrun{
fhIni()
options()$fhub$myid
r <- fhGet("people",options()$fhub$myid)
# Print contents
print( r, TRUE)
# Short version, default
r
r$response$status_code
status_code(r$response)
}

\donotrun{
skIni()
options()$fhub$myid
r <- skGet("people",options()$fhub$myid)
# Print contents
print( r, TRUE)
# Short version, default
r
r$response$status_code
status_code(r$response)
}
```

readMeta *Read metadata file from the given directory*

Description

Read metadata file from the given directory

Usage

```
readMeta(x = ".", ...)
```

Arguments

x	File path to the pISA layer.
...	Any other arguments.

Value

Data frame with Key/value pairs with class 'pISAmeta'.

Note

Metadata table gets the class 'Dlist' to inherit a convenient print.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
astring <- "_p_Demo/_I_Test/_S_Show/_A_Work-R/other"
oldwd <- setwd(system.file("extdata",astring,package="pisar"))
oldwd
.aroot <- getRoot("A")
.ameta <- readMeta(.aroot)
.ameta
setwd(oldwd)

## End(Not run)
```

skCreate *Create pISA layer or *fh* component.*

Description

Create pISA layer or *fh* component.

Usage

```
skCreate(type = "assays", meta = list(), class = "EXP",
  file = "NA.TXT")
```

Arguments

type	Component name (e.g. 'people', 'projets', ...).
meta	Data frame with pISA metadata or a list with minimal information (Title, Description, *ToDo: add fields*).
class	Assay class key string. Possible values are 'EXP' and 'MODEL'.
file	File name with path, relative to layer.

Value

FAIRDOMhub created component.

Note

Upon success (status code 200) details of newly created component can be used. Check status code.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skGet

Examples

```
\donotrun{
if(FALSE)
{
skIni(prid = 26, test=TRUE)
options("fhub")
 sp <- skCreate( type = "projects"
  , meta= list(
      Title=paste("Test project", Sys.time())
    , Description="Testing of upload")
    )
 str(sp)
 str(skData(sp))
# Add member manually
 skIni(prid = 26, pid=104, test=TRUE)
```

```
options("fhub")
si <- skCreate( type = "investigations"
 , meta= list(
     Title=paste("Test investigation", Sys.time())
   , Description="Testing of upload")
   )
si
skData(si)$id

iid=skData(si)$id
iid <- 115
skIni(prid = 26, pid=104, iid=iid, test=TRUE)
options("fhub")
ss <- skCreate( type = "studies"
 , meta= list(
     Title=paste("Test study", Sys.time())
   , Description="Testing of upload")
   )
ss
skData(ss)$id

skIni(prid = 26
      , pid=104
      , iid=skData(si)$id
      , sid=skData(ss)$id
      , test=TRUE)
options("fhub")
sa <- skCreate( type = "assays"
 , meta= list(
     Title=paste("Test assay", Sys.time())
   , Description="Testing of upload")
   , class="EXP"
   )
#str(sa)
sa
}

# Type: data_file
astring <- "_p_Demo/_I_Test/_S_Show/_A_Work-R/"
oldwd <- setwd(system.file("extdata",astring,package="pisar"))
oldwd
.aname <- getLayer("A")
.aroot <- getRoot("A")
.ameta  <- readMeta()
file <- "input/README.MD"
skIni(prid = 26, pid=104, iid=115 , sid=117 , aid=401, test=TRUE)
type <- "data_files"
type <- "documents"
sdat <- skCreate( type = type
 , meta= list(
     Title=paste("Test assay", Sys.time())
   , Description="Testing of upload")
 , file=file
   )
#str(sdat)
sdat
skData(sdat)$id
```

```
if(interactive()) setwd(oldwd)
getwd()
res <- sdat$content
item_link <- file.path(res$meta$base_url,res$links$self)
if(interactive()) shell.exec(item_link)
}
```

---

skData *Get content from an *fh* object.*

Description

Get content from an *fh* object.

Usage

```
skData(r, node, ...)
```

Arguments

r	Object retrieved by skGet.
type	Name of the required element. If missing, a list with all relevant objects is returned.

Value

File name (string).

Note

Parameter ... is ignored at this time.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

get ...

Examples

```
\donotrun{
skIni()
options()$fhub$myid
r <- skGet("people",options()$fhub$myid)
d <- skData(r,"attributes")
names(d)
d$last_name
skData(r)$tools
# Get list of people
r <- skGet("people")
d <- skData(r)
```

```
length(d)
names(d)
names(d[[1]])
titles <- sapply(d,function(x) x$attributes$title)
head(titles)
# Get FAIRDOMhub user id
myname <- titles[1]
myname
d[[pmatch(myname,titles)]]
id <- d[[pmatch(myname,titles)]]$id
id
}
```

---

skFindId *Get details of component with id from an *fh* object.*

Description

Get details of component with id from an *fh* object.

Usage

```
skFindId(type, title)
```

Arguments

type	Components name (e.g. 'people', 'projets', ...).
title	Character string with the identifier of the component (title part).

Value

FAIRDOMhub component identifier: id, type and title. If argument title is missing, a data frame with identifiers for all items is returned.

Note

If item is not found, value 0 is returned as id.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skFindTitle

Examples

```
\donotrun{
skIni()
id <- skFindId("people","Guest")
id
skFindTitle("people",id)
# does not exist
skFindId("people","No User")
# List of projects
projects <- skFindId("projects")
head(projects)
}
```

`skFindTitle` *Get details of component with id from an *fh* object.*

Description

Get details of component with id from an *fh* object.

Usage

```
skFindTitle(type, id)
```

Arguments

`type`	Components name (e.g. 'people', 'projets', ...).
`id`	Character string with the identifier of the component (id part).

Value

FAIRDOMhub component identifier: id, type and title. If argument title is missing, a data frame with identifiers for all items is returned. See note.

Note

If item is not found, empty string is returned as title.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

`skFindTitle`

Examples

```
\donotrun{
skIni()
id <- skFindId("people","Guest")
id
idNew <- skFindTitle("people", id[1])
idNew
all.equal(id, idNew)
skFindTitle("people",0)
}
```

---

skGet *Get inormation from repository.*

Description

Get inormation from repository.

Usage

```
skGet(type, id, uri = options()$fhub$baseurl, ...)
```

Arguments

type	Type of information (e.g. "person").
id	Repository id of an item.
uri	Repository base address (URI)..

Value

An object (list) of class seek_api.

Note

Parameter ... is ignored at this time.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

get ...

Examples

```
\donotrun{
skIni()
options()$fhub$myid
r <- skGet("people",options()$fhub$myid)
names(r)
r$response$status_code
status_code(r$response)
r
Non existent user
skGet("people",0)
}
```

---

skIni *Initialize FAIRDOMhub information*

Description

Define FAIRDOMhub URL and user data

Usage

```
skIni(prid = NULL, pid = NULL, iid = NULL, sid = NULL,
  aid = NULL, test = TRUE)
```

Arguments

test	If TRUE, test server will be used..

Value

A list with URL and user information. For side effect see Notes.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
skIni()
options("fhub")
#
skIni(test=FALSE)
options("fhub")

## End(Not run)
```

skLog *Writes a note to a log file.*

Description

Writes a note to a log file.

Usage

```
skLog(..., file = "FAIRDOM.log", append = TRUE)
```

Arguments

...	Objects to form a line.
file	Log file name.
append	Control append/rewrite mode.

Value

System time of invoking..

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skGet

Examples

```
tst <- function(){
  skLog("Test", "writing to logfile", file="")
  fht <- system.time(Sys.sleep(1))
  skLog( "Time:", round(fht["elapsed"],2))
}
tst()
rm(tst)
```

skParse *Parse the response from SEEK API*

Description

Parse the response from SEEK API

Usage

```
skParse(resp, ...)
```

Arguments

resp	Response from SEEK API.

Value

An object (list) of class seek_api.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skGget ...

Examples

```
\donotrun{
skIni()
options()$fhub$myid
r <- skGet("people",options()$fhub$myid)
names(r)
r$response$status_code
status_code(r$response)
names(r$content)
r
r <- skGet("people")
length(r$content)
names(r$content)
r$content[[1]]
names(r)
r
}
```

skSkeleton *Create *fh* skeleton.*

Description

Creates *fh* object with required structure.

Usage

```
skSkeleton(type = "assay", meta, file)
```

Arguments

type	Component name (e.g. 'people', 'projets', ...).
meta	Data frame with pISA metadata or a list with minimal information (Title, Description, *ToDo: add fields*).

Value

A list with the minimal information structure.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skCreate

skUpload *Upload file.*

Description

Upload file to a crerated object of type 'documents' or 'data_files'.

Usage

```
skUpload(object, file)
```

Arguments

object	A previously created SEEK component. Must be one of 'documents' or 'data_files'.
file	File name with path, relative to layer.

Value

FAIRDOMhub created component.

Note

Upon success (status code 200) details of newly created component can be used. Check status code.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skCreate

Examples

```
file <- dir()[1]
file
fhd <- skCreate("documents",meta,file=file)
fhd
sd <- skUpload(fhd, file)
```

Index