Package ‘seekr’

April 9, 2022

Type Package

Title R interface for SEEK API (connection to FAIRDOMHub)

Version 0.1.0.9000

Description The package provides several functions for upload/download
of data organized in pISA-tree to SEEK based cloud platform (specifically FAIRDOMHub).

License GPL-3

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

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

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

VignetteBuilder knitr

Suggests rmarkdown

R topics documented:

`contentType` *Determine MIME type for file.*

Description

Determine MIME type for file.

Usage

```
contentType(file)
```

Arguments

`file`	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")
contentType("bla")
contentType("bla.")
```

print.seek_api *Print method for seek_api object*

Description

Print method for seek_api object

Usage

```
## 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.
...	further arguments passed to or from other methods.

Value

An object (list) of class seek_api.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skParse

Examples

```
## Not run:
options(.sk$test)
options("sk.myid")
r <- skRead("people",options("sk.myid"))
# Print contents
print( r, TRUE)
# Short version, default
r
r$response$status_code
status_code(r$response)
r$status
```

```
## End(Not run)
```

setPublicAccess *Set public access permission.*

Description

Set public access permission according to the "Sharing permission:" key in metadata.

Usage

```
setPublicAccess(meta, na.access = "no_access")
```

Arguments

meta	data frame or list. pISA metadata object.
na.access	character string. Default access, either 'no_access' for #' Private layer or 'download' for Public layer.

Examples

```
## Not run:
meta = data.frame(
Key=c("Assay:","Sharing permission:"), Value=c("_A_Test","Public"))
setPublicAccess(meta)
meta = data.frame(
Key=c("Assay:","Sharing permission:"), Value=c("_A_Test","Private"))
setPublicAccess(meta)
meta = data.frame(
Key=c("Assay:"), Value=c("_A_Test"))
setPublicAccess(meta)
setPublicAccess(meta, "download")

## End(Not run)
```

skCheck *Check if component exists within the parent layer.*

Description

This is a faster version of skExists, intended mostly to check existence of ISA layers in the parent layer.

Usage

```
skCheck(type, title, ptype, pid = skGetOption(ptype), verbose = FALSE)
```

Arguments

`type`	Component type (possibly shortened e.g. 'pe' for 'people', 'proj' for 'projects', ...).
`title`	Character string, component name.
`ptype`	Parent component type (possibly shortened).
`pid`	Parent component id. If missing, appropriate part skOptions is used.
`verbose`	logical, if TRUE, details are printed.

Value

logical value indicating whether the component named by argument title exists.

Note

Check if named layer (.e.g. assay) exists in the parent layer. Parent layer is usually just one above (e.g. study where the assay is supposed to reside).

Author(s)

Andrej Blejec <`andrej.blejec@nib.si`>

Examples

```
## Not run:
skReset()
skFindId("proj","_p_Demo")
type <- "inv"
title <- "_I_Test"
ptype <- "proj"
#system.time(print(skExists(type,title,ptype))) # 35 seconds
print.simple.list(skOptions("id"))
system.time(print(skCheck(type,title,ptype)))
print.simple.list(skOptions("id"))
type <- "inv"
title <- "Testa"
ptype <- "proj"
(skCheck(type,title,ptype))
       print.simple.list(skOptions("id"))
(skCheck("inv","Test layer","proj"))
       print.simple.list(skOptions("id"))
title <- "Test investigation"
(skCheck(type,title,ptype))
       print.simple.list(skOptions("id"))
type <- "inv"
ptype <- "proj"
title <- "_I_01_LabTrials-test"
       print.simple.list(skOptions("id"))
system.time(print(skCheck(type,title,ptype)))
system.time(skExists(type,title,ptype))
       print.simple.list(skOptions("id"))
type <- "stud"
title <- "_S_03_dsRNAprod"
ptype <- "inv"
(skCheck(type,title,ptype))
```

```
       print.simple.list(skOptions("id"))
#
type <- "assay"
title <- "_A_01_pIsol-wet"
ptype <- "stu"
(skCheck(type,title,ptype))
       print.simple.list(skOptions("id"))
skCheck("proj","_p_Demo","people")

## End(Not run)
```

skCheckAndCreate *Check if layer exists and create it if not.*

Description

Check if layer exists and create it if not.

Usage

```
skCheckAndCreate(
  type,
  layers,
  root = skGetOption("root"),
  create = TRUE,
  verbose = FALSE,
  quiet = FALSE,
  na.access = "no_access"
)
```

Arguments

type	character. ISA layer type (words starting with I, S, or A, can be lowercase)
layers	character vector. Info about layers, usually the result of a call to skSetLayers.
root	character. Root used to create the file list, (usually?) pointing to the investigation above the working directory.
create	logical. Create layer if not existing?
verbose	logical. Print intermediate results?
quiet	logical. Print result?

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="seekr"))
oldwd
require(pisar)
require(httr)
require(jsonlite)
pisa <- pisa()
skReset()
options(.sk$test)
seekignore <- readLines(file.path(.iroot, "seekignore.txt"))
seekignore
dirs <- skFilesToUpload(.iroot)
layers <- skSetLayers(dirs[7], root=.iroot)
skFindId("projects", layers$pname)
(layers$iname <- paste(layers$iname, date()))
layers
print.simple.list(skOptions())
system.time(skCheckAndCreate("I",layers, create=TRUE, verbose=TRUE))
print.simple.list(skOptions())
system.time(skCheckAndCreate("S",layers, create=TRUE, verbose=TRUE))
print.simple.list(skOptions())
system.time(skCheckAndCreate("A",layers, create=TRUE, verbose=TRUE))
print.simple.list(skOptions())
if(interactive()) setwd(oldwd)
getwd()

## End(Not run)
```

skCheckAndCreate2 *Check if layer exists and create if not.*

Description

New function that wouldn't depend on the position of the working directory within the pISA-tree.

Usage

```
skCheckAndCreate2(
  type,
  layers,
  root = skGetOption("root"),
  create = TRUE,
  verbose = FALSE,
  quiet = FALSE
)
```

Arguments

type	character. ISA layer type (words starting with I, S, or A, can be lowercase)
layers	character vector. Info about layers, usually the result of a call to skSetLayers.

root	character. Root used to create the file list, usually pointing to investigation.
create	logical. Create layer if not existing?
verbose	logical. Print intermediate results?
quiet	logical. Print result?

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="seekr"))
oldwd
require(pisar)
require(httr)
require(jsonlite)
pisa <- pisa()
skReset()
options(.sk$test)
seekignore <- readLines(file.path(.sroot, "seekignore.txt"))
seekignore
dirs <- skFilesToUpload(.iroot)
layers <- skSetLayers(dirs[7], root=.iroot)
skFindId("projects", layers$pname)
print.simple.list(skOptions())
(layers$iname <- paste(layers$iname, date()))
skCheckAndCreate("I",layers, create=TRUE, verbose=TRUE)
skCheckAndCreate("S",layers, create=TRUE, verbose=TRUE)
skCheckAndCreate("A",layers, create=TRUE, verbose=TRUE)
if(interactive()) setwd(oldwd)
getwd()

## End(Not run)
```

---

skContent *Get content from an *sk* object.*

Description

Get content from an *sk* object.

Usage

```
skContent(r, node, ...)
```

Arguments

r	object retrieved by skGet.
node	name of the required element. If missing, a list with all relevant objects is returned.
...	further arguments.

Value

File name (string).

Note

Parameter ... is ignored at this time.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

get

Examples

```
## Not run:
options(.sk$test)
options("sk.myid")
r <- skRead("people",options("sk.myid"))
d <- skContent(r,"attributes")
names(d)
d$last_name
skContent(r)$attributes$tools
# Get list of people
r <- skList("people")
d <- skContent(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

## End(Not run)
```

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

Description

Create pISA layer or *sk* component.

Usage

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

Arguments

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

Created component id is set in options.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skGet

Examples

```
## Not run:
if(FALSE)
{
options(.sk$test)
options(sk.prid = 26)
skOptions("id")
meta= list(
      project=paste("Test project", Sys.time())
    , Title="Test project"
    , Description="Testing upload")
meta
rm(sp)
 sp <- skCreate( type = "projects"
    , meta = meta
    )
 skFindId("projects",meta$project)
 sp
 cat(skContent(sp)$attributes$description)
 sp$url
# Call to API does not set the 'member' field
# Add member manually in the project page on the web site:
# /Actions/Administer project members
#
 options(sk.prid=26, sk.pid=sp$id)
 skOptions("id")
 meta= list(
      Investigation=paste("Test investigation", Sys.time())
    , Title = "Test investigation"
    , Description="Testing upload")
    meta
    rm(si)
 si <- skCreate( type = "investigations"
  , meta=meta
    )
 si
 skContent(si)$id
 skFindId("investigations",meta$Investigation)
 skOptions("id")
# Create Study
 iid <- si$id
 options(sk.prid=26, sk.pid=sp$id, sk.iid=si$id)
 skOptions("id")
 meta= list(
      Study=paste("Test study", Sys.time())
    , Title="Test Study"
    , Description="Testing upload")
 ss <- skCreate( type = "studies", meta, access="no_access")
 ss
ss$id
 skFindId("studies",meta$Study)
# Assay
 options(sk.prid = 26
       , sk.pid=skContent(sp)$id
       , sk.iid=skContent(si)$id
       , sk.sid=skContent(ss)$id
       )
 skOptions()
 meta= list(
      Assay=paste("Test Assay", Sys.time())
    , Title="Test Assay"
    , Description="Testing upload"
    , class="EXP"
    )
 sa <- skCreate( type = "assays"  , meta)
 sa
 sa$id
 skContent(sa)$relationships$submitter
 skContent(sa)$links
}

 file <- "input/README.MD"
 options(sk.prid = 26, sk.pid=sp$id, sk.iid=si$id , sk.sid=ss$id , sk.aid=sa$id)
 skOptions()
 type <- "data_files"
 type <- "documents"
 sdat <- skCreate( type = type
  , meta= list(
     Title=paste("Test document", Sys.time())
    , Description="Testing of upload")
  , file=file
    )
 #str(sdat)
 sdat
 skContent(sdat)$id
if(interactive()) setwd(oldwd)
getwd()
res <- sdat$content
item_link <- paste0(res$meta$base_url,res$links$self)
if(interactive()) shell.exec(item_link)

## End(Not run)
```

skDelete *Delete component.*

Description

Delete component.

Usage

```
skDelete(type, id, uri = options("sk.url"), ...)
```

Arguments

`type`	Component type (e.g. "assay").
`id`	Repository id of component.
`uri`	Repository base address (URI).
`...`	further arguments.

Value

Deleted object (list) of class `seek_api`.

Note

Parameter ... is ignored at this time.

Component id in options is set to NULL.

Author(s)

Andrej Blejec <`andrej.blejec@nib.si`>

See Also

`skCreate`

Examples

```
## Not run:
options(.sk$test)
options("sk.myid")
r <- skDelete("assay",options("sk.aid"))
names(r)
r$response$status_code
status_code(r$response)
r

## End(Not run)
```

---

skExists *Check if component exists within the layer.*

Description

Check if component exists within the layer.

Usage

```
skExists(type, title, ptype, pid, verbose = FALSE)
```

Arguments

`type`	Component type (possibly shortened e.g. 'pe' for 'people', 'proj' for 'projects', ...).
`title`	Character string, component name.
`ptype`	Parent component type (possibly shortened).
`pid`	Parent component id. If missing, appropriate part skOptions is used.
`verbose`	logical, if TRUE, details are printed.

Value

logical value indicating whether the component named by argument title exists.

Note

Check if named layer (.e.g. assay) exists in the parent layer. Parent layer is usually just one above (e.g. study where the assay is supposed to reside) but can be any above.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
options(.sk$test)
skFindId("projects","_p_Demo")
skOptions()
skExists("people", "Andrej Blejec", "proj", options("sk.pid"))
skExists("people", "Andrej Blejec", "proj", verbose=TRUE)
skExists("investigations","_I_Test", "proj", options("sk.pid"),TRUE)
skExists("pe", "Guest", "proj", options("sk.pid"))
 skExists("proj","_p_Demo","progr",26,verbose=TRUE)
skOptions()

## End(Not run)
```

`skExistsa` *Check if component exist anywhere in the repository.*

Description

Check if component exist anywhere in the repository.

Usage

```
skExistsa(type, title, verbose = FALSE)
```

Arguments

`type`	Component type (possibly shortened e.g. 'pe' for 'people', 'proj' for 'projects', ...).
`title`	Character string with the identifier of the component (title part).
`verbose`	logical, if TRUE, details are printed.

Value

logical value indicating whether the component named by argument title exists.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
options(.sk$test)
skExistsa("people", "Andrej Blejec")
skExistsa("projects","Demo", verbose=TRUE)
skExistsa("pe", "Guest")

## End(Not run)
```

skExistsp *Check if component exist within the project.*

Description

Check if component exist within the project.

Usage

```
skExistsp(type, title, pid = getOption("sk.pid"), verbose = FALSE)
```

Arguments

`type`	Component type (possibly shortened e.g. 'pe' for 'people', 'proj' for 'projects', ...).
`title`	Character string with the identifier of the component (title part).
`pid`	numeric or character. Parent component id. Note: might be not needed.
`verbose`	logical, if TRUE, details are printed.

Value

logical value indicating whether the component named by argument title exists.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
## Not run:
options(.sk$test)
skFindId("projects","_p_Demo")
skOptions()
skExistsp("people", "Andrej Blejec")
skExistsp("investigations","_I_Test", verbose=TRUE)
skExistsp("pe", "Guest")

## End(Not run)
```

skExpand *Expand shortened type name into full name.*

Description

Expand shortened type name into full name.

Usage

```
skExpand(type)
```

Arguments

`type` Component type (possibly shortened e.g. 'pe' for 'people', 'proj' for 'projects', ...).

Value

Expanded type name.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

Examples

```
skExpand("pe")
skExpand("proj")
skExpand("inv")
skExpand("st")
skExpand("a")
skExpand(c("a","inv"))
# skExpand("p")
# Error in skExpand("p") : No such type or duplicate entry for 'p'
```

`skFilesToUpload` *Prepare the list of files for upload*

Description

Prepare the list of files for upload

Usage

```
skFilesToUpload(
  root,
  exclude.nokey = FALSE,
  exclude.md = TRUE,
  verbose = TRUE,
  quiet = FALSE,
  skignore = file.path(root, "seekignore.txt"),
  recursive = TRUE
)
```

Arguments

`root` character string. Directory, pointing to the branch of pISA-tree to be checked.

`exclude.nokey` logical value. If TRUE, the layer will be ignored if the "Upload to FAIRDOMHub" is not present.

`exclude.md` logical value. If TRUE, the file will be ignored if it contains just the header line.

`verbose` logical. Shold print progress details?

`quiet` logical. Should print final report?

`skignore` sharacter. File name or path to seekignore file.

`recursive` logical. Should the listing recurse into directories?

Note

Function ‘skFilesToUpload‘ will first create a list of files in root directory. Then, it will read the seekignore.txt and extend it with patterns of layers that should not be uploaded according to metadata key "Upload to FAIRDOMHub". In addition, seekignre list will be appended with the list of empty README.MD files that might clutter the FAIRDOMHub list of files. Then it will filter it according to declarations in the combined seekignore pattern list.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skIgnoreBlankReadme, skDoNotUpload

Examples

```
## Not run:
skFilesToUpload(.sroot)
skFilesToUpload(.aroot)
dirs <- skFilesToUpload(.iroot)
dirs
skFilesToUpload(exclude.md=FALSE)
skFilesToUpload( verbose=FALSE, quiet=TRUE)
seekignore <- c("# ignore patterns","*.zip","*.txt")
skFilesToUpload(skignore=seekignore)
skFilesToUpload(file.path(.iroot),skignore=NULL)

## End(Not run)
```

---

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

Description

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

Usage

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

Arguments

type	Components name (e.g. ’people’, ’projects’, ...).
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.

Touched component id is set in options.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skFindTitle

Examples

```
## Not run:
options(.sk$test)
id <- skFindId("people","Guest")
id
skFindTitle("people",id)
# does not exist
skFindId("people","No User")
# List of projects
projects <- skFindId("projects")
head(projects)
tail(projects)

## End(Not run)
```

---

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

Description

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

Usage

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

Arguments

type	Components name (e.g. 'people', 'projects', ...).
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.

Touched component id is set in options.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skFindTitle

Examples

```
## Not run:
skFindTitle("people",0)

## End(Not run)
```

skGet — *Get information from repository.*

Description

Get information from repository.

Usage

```
skGet(type, id, filter, uri = options("sk.url"), verbose = FALSE, ...)
```

Arguments

type	Type of information (e.g. "person").
id	Repository id of an item.
uri	Repository base address (URI).
verbose	logical, print details if TRUE.
...	further arguments.

Value

An object (list) of class seek_api.

Note

Parameter ... is ignored at this time.

Touched component id is set in the options (see: skOptions).

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

get, skRead, skList

Examples

```
## Not run:
options(.sk$test)
options("sk.myid")
r <- skGet("people",options("sk.myid"))
names(r)
r$response$status_code
status_code(r$response)
r
# Non existent user
skGet("people",0)

## End(Not run)
```

---

skGetLayer *Get layer name from path.*

Description

Get layer name from path.

Usage

```
skGetLayer(layer = "p", path = getwd(), verbose = FALSE)
```

Arguments

layer	character. pISA-tree layer prefix (one of characters "p", "I", "S" or "A").
path	Directory path, defaults to working directory.
verbose	logical. Should warnings be printed?

Note

This function should be included in the package pisar.

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="seekr"))
oldwd
.pname <- getLayer("p")
.pname
getLayer("I")
getLayer("S")
getLayer("A")
if(interactive()) setwd(oldwd)
getwd()
```

```
## End(Not run)
```

skGetOption *Get seekr option according to the type.*

Description

Get seekr option according to the type.

Usage

```
skGetOption(type)
```

Arguments

type	Components name (e.g. 'people', 'projects', ...).

Value

Option specified by type.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skSetOption

Examples

```
## Not run:
skGetOption("myid")
skSetOption("myid", 368)
skOptions()
skSetOption("assay",1234)
skGetOption("assay")
# Delete/reset option sk.aid
skSetOption("assay",NULL)
skOptions()
Get options
skGetOption("myid")
skGetOption("inv")

## End(Not run)
```

skList *List operations return a list of all objects of the specified type, to which the authenticated user has access.*

Description

List operations return a list of all objects of the specified type, to which the authenticated user has access.

Usage

```
skList(type, filter = "", uri = options("sk.url"), ...)
```

Arguments

`type`	Type of information (e.g. "person").
`filter`	character. For future use, not implemented
`uri`	Repository base address (URI).
`...`	further arguments.

Value

An object (list) of class `seek_api`.

Note

Parameter ... is ignored at this time.

Touched component id is set in the options (see: `skOptions`).

Author(s)

Andrej Blejec `<andrej.blejec@nib.si>`

See Also

`skRead`, `skGet`

Examples

```
## Not run:
options(.sk$test)
options("sk.myid")
r <- skList("people")
names(r)
r$response$status_code
status_code(r$response)
l <- skContent(r)
length(l)
# Short list of users
head(sort(sapply(l, function(x) x$attributes$title)),10)
# ToDO
# Querries like:
```

```
# https://testing.sysmo-db.org/people?filter[programme]=26
# https://testing.sysmo-db.org/people/368/items#institutions

## End(Not run)
```

---

skListp *List of all objects of the specified type within the project.*

Description

List of all objects of the specified type within the project.

Usage

```
skListp(
  class = projects,
  type,
  pid = getOption("sk.pid"),
  uri = options("sk.url"),
  ...
)
```

Arguments

class	Type of parent layer (e.g. "projects").
type	Type of information (e.g. "person").
pid	Id of a project giving the scope of the list.
uri	Repository base address (URI).
...	further arguments.

Value

A data.frame with ids and titles of related object.

Note

Parameter ... is ignored at this time.

Touched component id is set in the options (see: skOptions).

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skList

Examples

```
## Not run:
options(.sk$test)
options("sk.pid")
r <- skListp("proj","people")
r
skListp("inv","assays")
skListp("proj", "inv")

## 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), file="")
}
tst()
rm(tst)
```

skOptions *Show all seekr related options*

Description

Lists selected seekr related options that are set in via the options function. Selection is based on the start and end of the name. All seekr options start with prefix 'sk.'. Argument `end` give the required sufix for the option to be listed. Typical use would be to list identifiers, when `end = "id"`.

Usage

```
skOptions(end = "", start = "sk.", hide = "sk.pwd")
```

Arguments

`end`	character string defining the end of the string. By default all names are valid (empty string).
`start`	character string defining the start of the name, default is "sk.", which is typical for seekr options.
`hide`	vector of names, that should not be listed. Default is "sk.pwd" which prevents password to be revealed.

Value

A list of seekr related options.

Note

At the moment two locations are available: main FAIRDOMHub (https://www.fairdomhub.org/) and testing site (https://testing.sysmo-db.org).

Function options() is used to store server location, user credentials and pISA layer identifications. Options that can be declared:

sk.url server location

sk.usr username

sk.pwd password

sk.myid user id on server

sk.instid user's institution id

sk.prid program id

sk.pid project id

sk.iid investigation id

sk.sid study id

sk.aid assay id

sk.root root of files for upload

Author(s)

Andrej Blejec `<andrej.blejec@nib.si>`

See Also

startsWith,endsWith

Examples

```
## Not run:
options(sk.myid=368, sk.pid=166, sk.usr="Username")
skOptions()
skOptions("id")
skOptions(hide=NULL)

## End(Not run)
```

skParse *Parse the response from SEEK API and convert to a list.*

Description

Parse the response from SEEK API and convert to a list.

Usage

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

Arguments

resp	response from SEEK API.
...	further arguments.

Value

An object (list) of class seek_api.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skGet

Examples

```
## Not run:
options(.sk$test)
options("sk.myid")
r <- skRead("people",options("sk.myid"))
names(r)
r$response$status_code
status_code(r$response)
names(r$content)
r
r <- skRead("people")
```

```
length(r$content)
names(r$content)
r$content[[1]]
names(r)
r

## End(Not run)
```

skRead *Read operation will return information about the instance identified.*

Description

Read operation will return information about the instance identified.

Usage

```
skRead(type, id, uri = options("sk.url"), content = FALSE, ...)
```

Arguments

`type`	Type of information (e.g. "person").
`id`	Repository id of an item.
`uri`	Repository base address (URI).
`content`	Return complete response (FALSE) or content part (TRUE).
`...`	further arguments.

Value

An object (list) of class `seek_api`.

Note

Parameter ... is ignored at this time.

Touched component id is set in the options (see: `skOptions`).

Author(s)

Andrej Blejec <`andrej.blejec@nib.si`>

See Also

`skList`, `skGet`

Examples

```
## Not run:
options(.sk$test)
options("sk.myid")
r <- skRead("people",options("sk.myid"))
names(r)
r$response$status_code
status_code(r$response)
r
# Non existent user
skRead("people",0)

## End(Not run)
```

skRelated *List of relationships within an object.*

Description

List of relationships within an object.

Usage

```
skRelated(type, id, select = NULL, uri = options("sk.url"), ...)
```

Arguments

type	Object type (e.g. "project").
id	Id of the object giving the scope of the list. If missing, appropriate part of skOptions is used.
select	Vector of character strings with object types of interest.
uri	Repository base address (URI).
...	further arguments.

Value

A data.frame with ids, types, roles and titles of related objects.

Note

Parameter ... is ignored at this time.

Id of each last touched component is set in the options (see: skOptions).

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skListp

Examples

```
## Not run:
skRelated("projects",152)
skRelated("investigations",151)
skRelated("investigations",152)
skRelated("investigations",152, "assays")
skRelated("proj",options("sk.pid"), "inv")
skRelated("inv",152, c("people","studies"))
skRelated("inv",152, c("pe","a"))
skRelated("studies",165)
skRelated("assays",494)
skRelated("people",368,"projects")

## End(Not run)
```

skReset *Reset seekr related options*

Description

Reset seekr related options

Usage

```
skReset(all = FALSE)
```

Arguments

`all`	logical. Reset all options (including user and server related).

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skOptions

Examples

```
## Not run:
options(sk.myid=368, sk.pid=166, sk.usr="Username")
skOptions()
skReset()
skOptions()

## End(Not run)
```

skSearch *Search operation returns a list of references to resources.*

Description

Search operation returns a list of references to resources.

Usage

```
skSearch(type = "data_files", q, uri = options("sk.url"), content = FALSE, ...)
```

Arguments

`type`	Type of information (e.g. "person", default is "data_files").
`q`	Text to search for, spaces acceptable.
`uri`	Repository base address (URI).
`content`	Return complete response (FALSE) or content part (TRUE).
`...`	further arguments.

Value

An object (list) of class `seek_api`.

Note

Parameter ... is ignored at this time.

Touched component id is set in the options (see: `skOptions`).

Author(s)

Andrej Blejec <`andrej.blejec@nib.si`>

Examples

```
## Not run:
options(.sk$test)
options("sk.myid")
r <- skSearch("people","Guest")
names(r)
r$response$status_code
status_code(r$response)
str(skContent(r))
# Non existent user
x <- skSearch("people","XY")
x$content
str(skContent(x))
(skContent(skSearch("institutions","systems biology")))

## End(Not run)
```

skSetLayers *Set seek layer names based on pISA-tree file path.*

Description

Set seek layer names based on pISA-tree file path.

Usage

```
skSetLayers(
  path,
  root = skGetOption("root"),
  studyForFiles = "Investigation files",
  assaySuffix = "-files",
  verbose = FALSE
)
```

Arguments

path	Directory path, defaults to working directory.
verbose	logical. Should warnings be printed?
layer	character. pISA-tree layer prefix (one of characters "p", "I", "S" or "A").

Note

SEEK allows only files to be associated with assays. To upload files in investigation and study levels, a dummy study with assays for each study will be used to hold their files.

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="seekr"))
oldwd
require(pisar)
pisa <- pisa()
options(.sk$test)
seekignore <- readLines(file.path(.iroot, "seekignore.txt"))
seekignore
dirs <- skFilesToUpload(.iroot)
skSetLayers(dirs[1], root=.iroot)
skSetLayers(dirs[2], root=.iroot)
skSetLayers(dirs[length(dirs)], root=.iroot)
skSetLayers(dirs[10], root=.iroot)
sapply(dirs, function(x) skSetLayers(x))
if(interactive()) setwd(oldwd)
getwd()

## End(Not run)
```

skSetOption *Sets seekr option according to the type.*

Description

Sets seekr option according to the type.

Usage

```
skSetOption(type, id)
```

Arguments

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

Value

A list with the set option or NA if the type is not registered.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skFindTitle

Examples

```
## Not run:
skSetOption("people",111)
skSetOption("myid", 368)
skOptions()
skSetOption("bla",1) # can not be set
# Set option sk.aid
skSetOption("assay",1234)
skOptions()
# Delete/reset option sk.aid
skSetOption("assay",NULL)
skOptions()
Get options
skGetOption("myid")
skGetOption("inv")

## End(Not run)
```

skSkeleton *Create *sk* skeleton.*

Description

Creates *sk* object with required structure.

Usage

```
skSkeleton(type = "assay")
```

Arguments

type	Component type (e.g. 'people', 'projects', ...). 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

Examples

```
## Not run:
require(jsonlite)
type <- "projects"
sp <- skSkeleton( type = type )
str(sp)
type = "investigations"
si <- skSkeleton( type = type )
str(si)
type = "studies"
ss <- skSkeleton( type = type )
str(ss)
type = "assays"
sa <- skSkeleton( type = type )
str(sa)
type = "data_files"
file =
sdata <- skSkeleton( type = type )
str(sdata)

## End(Not run)
```

skUpload *Upload file content as a blob.*

Description

Upload file to a created 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

```
## Not run:
file <- dir()[1]
file
fhd <- skCreate("documents",meta,file=file)
fhd
sd <- skUpload(fhd, file)

## End(Not run)
```

skUploadFile *Upload file.*

Description

Create an object of type 'documents' or 'data_files' and upload file content as a blob.

Usage

```
skUploadFile(
  file,
  root = skGetOption("root"),
  verbose = FALSE,
  reportFile = NULL,
  access = "no_access"
)
```

Arguments

file	character. File name and path (from the root directory).
root	character. Root directory.

Note

Argument file is usually a result of a function skFilesToUpload that sets the option 'root' that was used.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skFilesToUpload, skUpload

Examples

```
## Not run:
skUploadFile(file)

## End(Not run)
```

skUploadFiles *Create appropriate layers and upload files.*

Description

Create an object of type 'documents' or 'data_files' and upload file content as a blob.

Usage

```
skUploadFiles(
  files,
  root = skGetOption("root"),
  test = FALSE,
  append = TRUE,
  verbose = FALSE,
  reportFile = NULL,
  invname
)
```

Arguments

files	character. Vector of file names with paths (from the root directory).
root	character. Root directory.
test	logical. If TRUE, original investigation name will be extended with the date-time to ensure unique name.
append	logical. Should log entries be appended to the existing log?
invname	character. Name that overrides automatic investigation name if test is TRUE.

Note

Argument files is usually a result of a function skFilesToUpload that sets the option 'root' that was used.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skFilesToUpload, skUpload

Examples

```
## Not run:
skUploadFiles(files)

## End(Not run)
```

Index