Package ‘seekr’

July 19, 2021

Type Package

Title seekr: 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

VignetteBuilder knitr

Suggests rmarkdown

NeedsCompilation no

Author Andrej Blejec [aut, cre]

Maintainer Andrej Blejec <andrej.blejec@nib.com>

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$test)
options("sk.myid")
r <- skGet("people",options("sk.myid"))
# Print contents
print( r, TRUE)
# Short version, default
r
r$response$status_code
status_code(r$response)
r$status

## End(Not run)
```

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

Description

Create pISA layer or *sk* 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. Call through API does not set the 'member

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skGet

Examples

```
## Not run:
if(FALSE)
{
options(.sk$test)
options(sk.prid = 26)
skOptions("id")
meta= list(
      Title=paste("Test project", Sys.time())
    , Description="Testing upload")
meta
rm(sp)
 sp <- skCreate( type = "projects"
    , meta = meta
    )
    (r <- skFindId("projects",meta$Title))
 sp
 str(skData(sp))
# 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$id)
 skOptions("id")
 meta= list(
      Title=paste("Test investigation", Sys.time())
    , Description="Testing upload")
    meta
    rm(si)
 si <- skCreate( type = "investigations"
  , meta=meta
    )
 si
 skData(si)$id
 skFindId("investigations",meta$Title)
 options(sk.iid=si$id)
 skOptions("id")
#'
 iid=skData(si)$id
 options(sk.prid=26, sk.pid=sp$id, sk.iid=si$id)
 skOptions("id")
 ss <- skCreate( type = "studies"
  , meta= list(
      Title=paste("Test study", Sys.time())
    , Description="Testing upload")
    )
 ss
 skData(ss)$id
#'
 options(sk.prid = 26
       , sk.pid=skData(sp)$id
       , sk.iid=skData(si)$id
       , sk.sid=skData(ss)$id
       )
 skOptions()
 sa <- skCreate( type = "assays"
  , meta= list(
      Title=paste("Test assay", Sys.time())
    , Description="Testing upload")
    , class="EXP"
    )
 str(sa)
 skData(sa)$relationships$submitter
 skData(sa)$links
}

# Type: data_file
astring <- "_p_Demo/_I_Test/_S_Show/_A_Work-R/"
oldwd <- setwd(system.file("extdata",astring,package="seekr"))
oldwd
.aname <- getLayer("A")
.aroot <- getRoot("A")
.ameta  <- readMeta()
 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 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)

## End(Not run)
```

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

Description

Get content from an *sk* object.

Usage

```
skData(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$test)
options("sk.myid")
r <- skGet("people",options("sk.myid"))
d <- skData(r,"attributes")
names(d)
d$last_name
skData(r)$attributes$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
```

```
## 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.

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.

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

skFindTitle

Examples

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

## End(Not run)
```

skGet *Get inormation from repository.*

Description

Get inormation from repository.

Usage

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

Arguments

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

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

```
## 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)
```

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)
```

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).

Author(s)

Andrej Blejec <andrej.blejec@nib.si>

See Also

startsWith,endsWith

Examples

```
## Not run:
options(sk.myid=368)
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$test)
options("sk.myid")
r <- skGet("people",options("sk.myid"))
names(r)
r$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

## End(Not run)
```

skSkeleton *Create *sk* skeleton.*

Description

Creates *sk* object with required structure.

Usage

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

Arguments

type	component name (e.g. 'people', 'projects', ...).
meta	a 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

Examples

```
## Not run:
require(jsonlite)
meta <- list(Title = "Test layer", Description = "Some description")
type <- "projects"
sp <- skSkeleton( type = type
  , meta= meta
  )
str(sp)
type = "investigations"
si <- skSkeleton( type = type
  , meta= meta
  )
str(si)
type = "studies"
ss <- skSkeleton( type = type
  , meta= meta
  )
str(ss)
type = "assays"
sa <- skSkeleton( type = type
  , meta= meta
  )
```

```
str(sa)

type = "data_files"
file =
sdata <- skSkeleton( type = type
  , meta= meta
  )
str(sdata)

## End(Not run)
```

skUpload *Upload file.*

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)
```

Index