

# Package ‘seekr’

October 12, 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

## R topics documented:

asString . . . . .	2
contentType . . . . .	3
print.seek_api . . . . .	3
skContent . . . . .	4
skCreate . . . . .	5
skDelete . . . . .	5
skFindId . . . . .	6
skFindTitle . . . . .	7
skGet . . . . .	8
skList . . . . .	9
skLog . . . . .	10
skOptions . . . . .	11
skParse . . . . .	12
skRead . . . . .	13

skSearch . . . . .	14
skSetOption . . . . .	15
skSkeleton . . . . .	15
skUpload . . . . .	17

<b>Index</b>	<b>18</b>
--------------	-----------

---

astring	<i>Create pISA layer or *sk* component.</i>
---------	---

---

## Description

Create pISA layer or \*sk\* component.

## Usage

astring

## Arguments

type	Component name (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.

## Format

An object of class character of length 1.

## 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](#)

---

contentType	<i>Determine MIME type for file.</i>
-------------	--------------------------------------

---

**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	<i>Print method for seek_api object</i>
----------------	---

---

**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)
```

---

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

---

**Usage**

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

---

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)
```

---

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, 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.  
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)
```

---

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, uri = options("sk.url"), ...)
```

**Arguments**

type	Type of information (e.g. "person").
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](mailto: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)

## End(Not run)
```

---

skLog	<i>Writes a note to a log file.</i>
-------	-------------------------------------

---

**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	<i>Show all seekr related options</i>
-----------	---------------------------------------

---

**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 suffix 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](mailto: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$test)
options("sk.myid")
r <- skRead("people",options("sk.myid"))
names(r)
r$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"), ...)
```

**Arguments**

<code>type</code>	Type of information (e.g. "person").
<code>id</code>	Repository id of an item.
<code>uri</code>	Repository base address (URI).
<code>...</code>	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$test)
options("sk.myid")
r <- skRead("people",options("sk.myid"))
names(r)
r$response$status_code
status_code(r$response)
r
# Non existent user
skRead("people",0)

## End(Not run)
```

---

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"), ...)
```

## Arguments

type	Type of information (e.g. "person", default is "data_files").
q	Text to search for, spaces acceptable.
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>

## 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)
```

---

skSetOption	<i>Sets seekr option according to the type.</i>
-------------	---

---

**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)

## End(Not run)
```

---

skSkeleton	<i>Create *sk* skeleton.</i>
------------	------------------------------

---

**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**

<code>object</code>	A previously created SEEK component. Must be one of 'documents' or 'data_files'.
<code>file</code>	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

## \* **file**

skContent, [4](#)

skGet, [8](#)

## \* **pisa**

astring, [2](#)

contentType, [3](#)

skFindId, [6](#)

skFindTitle, [7](#)

skLog, [10](#)

skSkeleton, [15](#)

skUpload, [17](#)

astring, [2](#)

contentType, [3](#)

endsWith, [11](#)

get, [5](#), [9](#)

print.seek\_api, [3](#)

skContent, [4](#)

skCreate, [5](#), [6](#), [16](#), [17](#)

skDelete, [5](#)

skFindId, [6](#)

skFindTitle, [7](#), [7](#), [8](#), [15](#)

skGet, [2](#), [8](#), [9](#), [10](#), [12](#), [13](#)

skList, [9](#), [9](#)

skLog, [10](#)

skOptions, [8](#), [9](#), [11](#), [13](#), [14](#)

skParse, [4](#), [12](#)

skRead, [9](#), [13](#), [13](#)

skSearch, [14](#)

skSetOption, [15](#)

skSkeleton, [15](#)

skUpload, [17](#)

startsWith, [11](#)