

# Package ‘kntnr’

April 8, 2020

**Type** Package

**Title** R Client for 'kintone' API

**Version** 0.4.4

**Description** Retrieve data from 'kintone' (<<https://www.kintone.com/>>) via its API. 'kintone' is an enterprise application platform.

**License** MIT + file LICENSE

**URL** <https://yutanihilation.github.io/kntnr/>

**BugReports** <https://github.com/yutanihilation/kntnr/issues>

**Imports** base64enc, dplyr (>= 0.7.0), httr, jsonlite, lubridate, methods, purrr (>= 0.2.3), rlang, rstudioapi, stringr, tibble, tidy

**Suggests** testthat

**LazyData** TRUE

**Encoding** UTF-8

**RoxygenNote** 7.1.0

**NeedsCompilation** no

**Author** Hiroaki Yutani [aut, cre] (<<https://orcid.org/0000-0002-3385-7233>>)

**Maintainer** Hiroaki Yutani <yutani.ini@gmail.com>

**Repository** CRAN

**Date/Publication** 2020-04-08 13:10:02 UTC

## R topics documented:

kntnr . . . . .	2
kntn_auth . . . . .	2
kntn_file . . . . .	3
kntn_record . . . . .	4
kntn_unnest . . . . .	6
<b>Index</b>	<b>7</b>

---

kntnr	<i>R Client for 'kintone' API</i>
-------	-----------------------------------

---

### Description

Retrieve data from kintone.

---

kntn_auth	<i>Authorization Header for 'kintone'</i>
-----------	---

---

### Description

kntn\_get\_authorization\_header() constructs a header for authorization using [add\\_headers](#). kntn\_set\_auth() interactively asks for the type of authentication and the credential and store them as environmental variables KNTN\_URL, KNTN\_AUTH\_TYPE and KNTN\_AUTH. If you want to avoid interaction, please manually set these. kntn\_unset\_auth() unsets these environmental variables.

### Usage

```
kntn_get_authorization_header()

kntn_set_auth(auth_type = c("password", "token"), overwrite = FALSE)

kntn_unset_auth()
```

### Arguments

auth_type	Type of Authentication. password uses username and password. token uses per-app token.
overwrite	If TRUE, overwrite the existing environmental variables.

### See Also

<https://developer.kintone.io/hc/en-us/articles/212495188/#userauthentication>

### Examples

```
## Not run:
# set KNTN_URL, KNTN_AUTH and KNTN_AUTH_TYPE interactively.
# By default, auth_type is password.
kntn_set_auth()

Sys.getenv(c("KNTN_URL", "KNTN_AUTH", "KNTN_AUTH_TYPE"))

# This will return `X-Cybozu-Authorization` header.
kntn_get_authorization_header()
```

```
# Clear environmental variables before trying to use another set of authorization info.
ktn_unset_auth()
ktn_set_auth(auth_type = "token")

# This will return `X-Cybozu-API-Token` header.
ktn_get_authorization_header()

# To avoid interaction, set these environmental variables manually.
Sys.setenv("KNTN_URL" = "https://example.cybozu.com/")
Sys.setenv("KNTN_AUTH" = "abcdefg")
Sys.setenv("KNTN_AUTH_TYPE" = "token")

## End(Not run)
```

---

ktn_file	<i>'kintone' File API</i>
----------	---------------------------

---

## Description

Get a file from kintone API and parse it with [content](#). If you want to parse it by yourself, specify as = "raw" or as = "text".

## Usage

```
ktn_file(fileKey, verbose = FALSE, as = NULL, type = NULL, encoding = NULL)
```

## Arguments

fileKey	File key.
verbose	If TRUE, output logs verbosely.
as	as parameter passed to <a href="#">content</a> .
type	type parameter passed to <a href="#">content</a> .
encoding	encoding parameter passed to <a href="#">content</a> .

## See Also

<https://developer.kintone.io/hc/en-us/articles/212494468/>

## Examples

```
## Not run:
ktn_set_auth()
app <- 10

# get a single record with a file attachment field
d <- ktn_record(app, id = 1)
```

```
f <- kntn_file(fileKey = x$Attachment[[1]]$fileKey[1])

## End(Not run)
```

---

kntn_record	<i>'kintone' Record API</i>
-------------	-----------------------------

---

### Description

`kntn_record()` gets a single record from the specified kintone application. `kntn_records()` retrieves multiple records at once. If the number of records is more than `records_per_request` (the default is 100), `kntn_records()` automatically splits the request into smaller subrequests.

### Usage

```
kntn_record(app, id, as = c("data.frame", "list", "text"), verbose = FALSE)
```

```
kntn_records(
  app,
  fields = NULL,
  query = "",
  max_records = 1000L,
  offset = 0L,
  records_per_request = 100L,
  as = c("data.frame", "list", "text"),
  verbose = FALSE
)
```

### Arguments

<code>app</code>	App ID.
<code>id</code>	Record ID.
<code>as</code>	Desired type of output: <code>data.frame</code> , <code>list</code> or <code>text</code> ( <code>data.frame</code> means <a href="#">tbl_df</a> ).
<code>verbose</code>	If TRUE, output logs verbosely.
<code>fields</code>	Names of fields.
<code>query</code>	Query (e.g. <code>"updated_time &gt; \"2012-02-03T09:00:00+0900\"</code> and <code>updated_time &lt; \"2012-02-03T10:00:00+0900\"</code> ).
<code>max_records</code>	Max number of records to get.
<code>offset</code>	Offset of records.
<code>records_per_request</code>	Number of records per request (max: 100).

## Details

A field will be converted to the correspondent object by the type:

- RECORD\_NUMBER: character
- \_\_ID\_\_: integer
- \_\_REVISION\_\_: integer
- CREATOR: character (code)
- CREATED\_TIME: POSIXct
- MODIFIER: character (code)
- UPDATED\_TIME: POSIXct
- SINGLE\_LINE\_TEXT: character
- NUMBER: numeric
- CALC: character
- MULTI\_LINE\_TEXT: character
- RICH\_TEXT: character
- CHECK\_BOX: nested character
- RADIO\_BUTTON: character
- DROP\_DOWN: character
- MULTI\_SELECT: nested character
- FILE: nested [tbl\\_df](#)
- LINK: character
- DATE: Date
- TIME: character (R has no correspondent class for this)
- DATETIME: POSIXct
- USER\_SELECT: nested character (code)
- ORGANIZATION\_SELECT: nested character (code)
- GROUP\_SELECT: nested character (code)
- CATEGORY: nested character
- STATUS: character
- STATUS\_ASSIGNEE: character
- SUBTABLE: nested tbl

Some types will be converted to nested objects. You can unnest these fields by [kntn\\_unnest](#).

## See Also

<https://developer.kintone.io/hc/en-us/articles/213149287/>

## Examples

```
## Not run:
kntn_set_auth()

app <- 10

# get a single record
d <- kntn_record(app, id = 1)

# get records up to 1000 (default)
d <- kntn_records(app)

# get records up to 5000 records at the latency of 500 records/request.
d <- kntn_records(app, max_records = 5000, records_per_request = 500L)

# get records as list
d <- kntn_records(app, as = "list")

# get records matched with the specified query and fields.
# See https://developer.kintone.io/hc/en-us/articles/213149287/ for the query syntax
d <- kntn_records(app, fields = c("timestamp", "value"),
                 query = "updated_time > \"2016-10-03T09:00:00+0900\"")

# Some types like SUBTABLE are converted as nested data.frame.
# You can unnest them by using kntn_unnest.
kntn_unnest(d)

## End(Not run)
```

---

kntn\_unnest

*Unnest 'kintone' Records*

---

## Description

The data retrieved by [kntn\\_records](#) may contain nested data.frames. `kntn_unnest()` unnests them by using [unnest](#). Note that this function is very experimental and may not work well for all data.

## Usage

```
kntn_unnest(records)
```

## Arguments

records            Data.frame retrieved by `kntn_records()`

# Index

`add_headers`, 2

`content`, 3

`kntn_auth`, 2

`kntn_file`, 3

`kntn_get_authorization_header`  
(`kntn_auth`), 2

`kntn_record`, 4

`kntn_records`, 6

`kntn_records` (`kntn_record`), 4

`kntn_set_auth` (`kntn_auth`), 2

`kntn_unnest`, 5, 6

`kntn_unset_auth` (`kntn_auth`), 2

`kntnr`, 2

`tbl_df`, 4, 5

`unnest`, 6