Package 'rKolada'

May 9, 2026

Type Package

Title Access Data from the 'Kolada' Database

Version 0.3.1

Maintainer Love Hansson <love.hansson@gmail.com>

Description Methods for downloading and processing data and metadata from 'Kolada', the official Swedish regions and municipalities database <https://www.kolada.se/>.

License AGPL-3

URL https://lchansson.github.io/rKolada/,
https://github.com/lchansson/rKolada

BugReports https://github.com/lchansson/rKolada/issues

Encoding UTF-8

Suggests testthat (>= 3.0.0), knitr, rmarkdown, stringi, ggplot2, scales, progress

Config/testthat/edition 3

RoxygenNote 7.3.3

Imports cli, dplyr, httr2, jsonlite, purrr, rlang, stringr, tibble, tidyr

VignetteBuilder knitr

Depends R (>= 4.1.0)

NeedsCompilation no

Author Love Hansson [aut, cre, cph]

Repository CRAN

Date/Publication 2026-03-23 11:00:02 UTC

Contents

`allowed_entities` *Allowed entities: Kolada metadata classes*

Description

Allowed entities: Kolada metadata classes

Usage

```
allowed_entities()
```

Value

A vector of names of allowed metadata entities, i.e. the correct spelling of all allowed values of the `entity` parameter in `get_metadata()`.

`compose_data_query` *Compose a query to fetch metadata from the Kolada API. Its use is mainly*

Description

Mainly used as a supporting function for `get_values()` but can also be used to create a working URL to paste in your web browser.

Usage

```
compose_data_query(
  kpi = NULL,
  municipality = NULL,
  period = NULL,
  ou = NULL,
  unit_type = "municipality",
  from_date = NULL,
  page = NA,
  per_page = NA,
  version = "v3"
)
```

Arguments

`kpi`	What kpis should be fetched? Can be a single name or a vector of names.
`municipality`	For which municipalities should data be fetched? Can be a single name or a vector of names.
`period`	For what years should data be fetched? Can be one or more four-digit integers or character strings.
`ou`	(Optional) for what Operating Units should data be fetched? Only available for certain KPIs. Only used if `unit_type` is set to `"ou"`.
`unit_type`	One of `"municipality"` or `"ou"`. Whether to fetch data for Municipalities or Organizational Units. Units. Defaults to `"municipality"`.
`from_date`	(Optional) Only return data updated after this date. Format: `"YYYY-MM-DD"`.
`page`	What page to fetch. Used mainly in large queries. Fetches a page using the value of `"per_page"` as pagination delimiter.
`per_page`	Number of results per page.
`version`	Version of the API. Defaults to `"v3"`.

Value

A string containing a URL to the Kolada REST API.

compose_metadata_query *Compose a query to fetch metadata from the Kolada API.*

Description

Mainly used as a supporting function for `get_metadata()` but can also be used to create a working URL to paste in your web browser.

Usage

```
compose_metadata_query(
  entity = "kpi",
  title = NULL,
  id = NULL,
  municipality = NULL,
  region_type = NULL,
  page = NA,
  per_page = NA,
  version = "v3"
)
```

Arguments

`entity`	Any allowed metadata entity. Check `allowed_entities()` to see an updated list.
`title`	A free-form search term or the exact title of any entry in the current entity. Case insensitive.
`id`	The ID of any entry in the current entity.
`municipality`	If entity is `"ou"`, the municipality parameter can be added to narrow the search.
`region_type`	(Optional) Filter municipalities by region type. Only used when `entity` is `"municipality"`. Common values: `"K"` (municipality), `"L"` (region).
`page`	What page to fetch. Used mainly in large queries. Fetches a page using the value of `"per_page"` as pagination delimiter.
`per_page`	Number of results per page.
`version`	Version of the API. Defaults to `"v3"`.

Value

A string containing a URL to the Kolada REST API.

get_kpi *Download metadata for a specific entity from the Kolada API*

Description

There are five different types of metadata entities in the Kolada database: "kpi", "kpi_groups", "municipality", "municipality_groups", and "ou". For every entity there is a corresponding function `get_ENTITY` which retrieves a table with the metadata for that entity. The `get_ENTITY` functions are thin wrappers around `get_metadata()`.

Usage

```
get_kpi(
  id = NULL,
  max_results = NULL,
  cache = FALSE,
  cache_location = tempdir,
  verbose = FALSE
)

get_kpi_groups(
  id = NULL,
  cache = FALSE,
  max_results = NULL,
  cache_location = tempdir,
  verbose = FALSE
)

get_ou(
  id = NULL,
  municipality = NULL,
  max_results = NULL,
  cache = FALSE,
  cache_location = tempdir,
  verbose = FALSE
)

get_municipality(
  id = NULL,
  region_type = NULL,
  cache = FALSE,
  max_results = NULL,
  cache_location = tempdir,
  verbose = FALSE
)

get_municipality_groups(
```

```
  id = NULL,
  cache = FALSE,
  max_results = NULL,
  cache_location = tempdir,
  verbose = FALSE
)
```

Arguments

id	(Optional) One or several KPI IDs
max_results	(Optional) Specify the maximum number of results returned by the query.
cache	Logical. If TRUE, downloaded data are stored to the local disk in the place specified by cache_location. If data is already present on the local disk, this data is returned instead of downloading data from the API.
cache_location	Where to store and search for cached data. Can be a path to a directory or the name of any function that returns the path to a directory when called, like getwd(). Defaults to tempdir().
verbose	Whether to print the call to the Kolada API as a message to the R console.
municipality	(Optional) A string or vector of strings containing municipality codes. If getting OU data, you can use this parameter to narrow the search.
region_type	(Optional) Filter municipalities by region type. Common values: "K" (municipality), "L" (region).

Value

Returns a tibble with metadata for the specified entity. In rKolada terminology, a table returned by e.g. get_kpi() is referred to as a kpi_df and can be passed to functions starting with "kpi" such as kpi_bind_keywords().

Examples

```
# Download KPI table and store a cache copy of the results in a temporary folder
# (to actually download all available data, don't specify max_results)
if (kolada_available()) {
kpi_df <- get_kpi(cache = TRUE, max_results = 100)
}
```

---

get_metadata *Download metadata from the Kolada API*

Description

This is a generalized function for downloading metadata from the Kolada API. The function parameters closely mask the names specified in the original API. For further information about the Kolada API specification, please see the official documentation on GitHub.

Usage

```
get_metadata(
  entity = "kpi",
  title = NULL,
  id = NULL,
  municipality = NULL,
  region_type = NULL,
  max_results = NULL,
  cache = FALSE,
  cache_location = tempdir,
  verbose = FALSE
)
```

Arguments

entity	Any allowed metadata entity. Check allowed_entities() to see an updated list.
title	A free-form search term or the exact title of any entry in the current entity. Case insensitive.
id	The ID of any entry in the current entity.
municipality	If entity is "ou", the municipality parameter can be added to narrow the search.
region_type	(Optional) Filter municipalities by region type. Only used when entity is "municipality". Common values: "K" (municipality), "L" (region).
max_results	(Optional) Specify the maximum number of results returned by the query.
cache	Logical. If TRUE, downloaded data are stored to the local disk in the place specified by cache_location. If data is already present on the local disk, this data is returned instead of downloading data from the API.
cache_location	Where to store and search for cached data. Can be a path to a directory or the name of any function that returns the path to a directory when called, like getwd(). Defaults to tempdir().
verbose	Whether to print the call to the Kolada API as a message to the R console.

Value

Returns a tibble with metadata for the specified entity. In rKolada terminology, a table returned by e.g. entity = "kpi" is referred to as a kpi_df and can be passed to functions starting with "kpi" such as kpi_bind_keywords().

See Also

get_kpi(), get_kpi_groups(), get_municipality(), get_municipality_groups(), get_ou()

`get_values` *Get data from Kolada*

Description

Download a table of data from Kolada. Data is selected based on three metadata dimensions: KPI (ID), municipality (ID) and period (years). You must supply arguments for at least two of these three dimensions. If a dimension is omitted, all available data for that dimension will be downloaded.

Usage

```
get_values(
  kpi = NULL,
  municipality = NULL,
  period = NULL,
  ou = NULL,
  unit_type = "municipality",
  max_results = NULL,
  from_date = NULL,
  keep_deleted = FALSE,
  simplify = TRUE,
  verbose = FALSE
)
```

Arguments

`kpi`	What kpis should be fetched? Can be a single name or a vector of names.
`municipality`	For which municipalities should data be fetched? Can be a single name or a vector of names.
`period`	For what years should data be fetched? Can be one or more four-digit integers or character strings.
`ou`	(Optional) for what Operating Units should data be fetched? Only available for certain KPIs.
`unit_type`	One of `"municipality"` or `"ou"`. Whether to fetch data for Municipalities or Organizational Units.
`max_results`	(Optional) Specify the maximum number of results returned by the query.
`from_date`	(Optional) Only return data updated after this date. Format: `"YYYY-MM-DD"`.
`keep_deleted`	Logical. If `FALSE` (default), rows where `isdeleted` is `TRUE` are removed from the result.
`simplify`	Whether to make results more human readable.
`verbose`	Whether to print the call to the Kolada API as a message to the R console.

Value

A tibble containing Kolada values and metadata.

Examples

```
# Download data for KPIs for Gross Regional Product ("bruttoregionprodukt" in Swedish)
# for three municipalities
if (kolada_available()) {

# If you already know the ID numbers you are looking for,
# you can use these directly as argments.
# Otherwise, use the get_\*() functions in combination with \*_search()
# functions to find good parameter values.
grp_data <- get_values(
  kpi = c("N03700", "N03701"),
  municipality = c("0180", "1480", "1280")
)

# To download OU data instead of Municipality data, set the parameter
# "unit_type" to "ou".
ou_data <- get_values(
 kpi = "N15033",
 ou = "V15E144001101",
 unit_type = "ou"
)
}
```

`kolada_available` *Check if the Kolada API is available*

Description

Performs a lightweight HTTP check to verify that the Kolada API is reachable. This is primarily useful for guarding examples and tests.

Usage

```
kolada_available()
```

Value

`TRUE` if the API responds within 5 seconds, `FALSE` otherwise.

kolada_cache_dir *Get the rKolada cache directory*

Description

Returns the path to the persistent cache directory used by rKolada. The directory is created if it does not exist.

Usage

```
kolada_cache_dir()
```

Value

A string containing the path to the cache directory.

kolada_clear_cache *Clear the rKolada cache*

Description

Removes all cached files from the rKolada cache directory.

Usage

```
kolada_clear_cache(cache_dir = kolada_cache_dir())
```

Arguments

cache_dir	The cache directory to clear. Defaults to kolada_cache_dir().

Value

Invisibly returns the cache directory path.

kpi_bind_keywords *Add keyword columns to a Kolada KPI table*

Description

Identify n keywords describing the KPI and add them as new columns. Keywords are inferred from the `title` field of the table.

Usage

```
kpi_bind_keywords(kpi_df, n = 2, form = c("wide", "long"))
```

Arguments

kpi_df	A Kolada KPI metadata table, e.g. as created by `get_kpi()`.
n	How many keyword columns should be added?
form	Can be either "wide" (default) or "long". Whether to return keywords as separate columns ("wide") or as separate rows, duplicating all other data ("long").

Value

A Kolada KPI metadata table

Examples

```
if (kolada_available()) {
kpi_df <- get_kpi(id = c("N45933", "U28563")) |>
  kpi_bind_keywords(n = 3)
}
```

kpi_describe *Describe the KPIs in a Kolada KPI metadata table*

Description

Print a human-readable description of each entity of a KPI metadata table (up to a maximum number of rows). Can be printed either directly to the R console or used to populate a R markdown document, which can be useful for documentation purposes.

Usage

```
kpi_describe(
  kpi_df,
  max_n = 5,
  format = "inline",
  heading_level = 2,
  sub_heading_level = heading_level + 1
)
```

Arguments

kpi_df	A Kolada KPI metadata table
max_n	The maximum number of KPIs to describe.
format	Output format. Can be one of "inline" (default) or "md", i.e. markdown.
heading_level	The top heading level output format is "md".
sub_heading_level	The sub heading level output format is "md".

Value

Returns the object passed to the function, invisibly, to be re-used in a pipe.

---

kpi_extract_ids *Extract a vector of KPI ID strings from a Kolada KPI metadata table*

Description

This function is primarily intended as a convenient way to pass a (filtered) Kolada KPI metadata table to `get_values()`.

Usage

```
kpi_extract_ids(kpi_df)
```

Arguments

kpi_df	A Kolada KPI metadata table, e.g. as created by `get_kpi()`.

Value

A vector of KPI IDs.

Examples

```
if (kolada_available()) {
# Download Kolada data for the first KPI in the list matching the term "Grundskola" (primary school)
# for the years 2010-2019
# (omit the parameter "max_results" to actually download all data)
kpi_filter <- get_kpi(max_results = 500) |>
  kpi_search("Grundskola")

# Only keep the top row
kpi_filter <- kpi_filter[1, ]

# Only download 100 observations
# (omit the parameter "max_results" to actually download all data)
kld_data <- get_values(
  kpi = kpi_extract_ids(kpi_filter),
  period = 2010:2019,
  max_results = 100
)
}
```

---

kpi_grp_describe *Describe the KPIs in a Kolada KPI Group metadata table*

Description

Print a human-readable description of each row of a KPI Group metadata table, including member KPIs (up to a maximum number of rows). Can be printed either directly to the R console or used to populate a R markdown document, which can be useful for documentation purposes.

Usage

```
kpi_grp_describe(
  kpi_grp_df,
  max_n = 5,
  format = "inline",
  heading_level = 2,
  sub_heading_level = heading_level + 1
)
```

Arguments

kpi_grp_df	A Kolada KPI Group metadata table, as created by e.g. get_kpi_groups.
max_n	The maximum number of KPI groups to describe.
format	Output format. Can be one of "inline" or "md" (markdown).
heading_level	The top heading level output format is "md".
sub_heading_level	The sub heading level output format is "md".

Value

Returns the object passed to the function, invisibly, to be re-used in a pipe.

---

kpi_grp_extract_ids *Extract KPI ID strings from a Kolada KPI Group metadata table*

Description

This function is primarily intended as a convenient way to pass a (filtered) Kolada KPI Group metadata table to `get_values()`. All IDs of the KPIs contained in each group in the table are extracted.

Usage

```
kpi_grp_extract_ids(kpi_grp_df)
```

Arguments

`kpi_grp_df`	A Kolada KPI Group metadata table, as created by e.g. `get_kpi_groups`.

Value

A vector of KPI IDs.

---

kpi_grp_search *Search a Kolada KPI Group metadata table for group names*

Description

Search a Kolada KPI Group metadata table. Only keep rows that contain the search query. Searches group titles and group IDs. Note that this function does not search for individual KPIs contained within KPI groups! To search for KPIs within a KPI group, see examples below for an example using `kpi_grp_unnest`.

Usage

```
kpi_grp_search(kpi_grp_df, query)
```

Arguments

`kpi_grp_df`	A Kolada KPI Group metadata table, as created by e.g. `get_kpi_groups`.
`query`	A search term or a vector of search terms to filter by. Case insensitive.

Value

A Kolada KPI Group metadata table

Examples

```
if (kolada_available()) {
kpi_grp_df <- get_kpi_groups()

# Which KPI groups match the keyword "ekonomi" (economy)?
kpi_grp_df |> kpi_grp_search("ekonomi")

# Which KPI groups contain KPIs matching the keyword "arbete" (work/labour)?
kpi_grp_df |>
  kpi_grp_unnest() |>
  kpi_search("arbete") |>
  dplyr::count(group_title, sort = TRUE)
}
```

---

kpi_grp_unnest *Create a KPI table from a Kolada KPI Group metadata table*

Description

KPI groups are a convenient way to discover sets of KPIs that can be used to highlight different aspects of a policy area. A practical workflow for discovering such sets can be to search through KPI Group metadata using `kpi_grp_search()` to search for keywords and `kpi_grp_describe()` to inspect contents of KPI groups. Once you have created a KPI group table that has been narrowed down to the group/s you are looking for, `kpi_grp_unnest()` is used to create a KPI metadata table for further processing.

Usage

```
kpi_grp_unnest(kpi_grp_df)
```

Arguments

`kpi_grp_df`	A Kolada KPI Group metadata table, as created by e.g. `get_kpi_groups`.

Value

A Kolada KPI metadata table

Examples

```
if (kolada_available()) {
# Download KPI Group metadata
kpi_grp_df <- get_kpi_groups()

# Create a KPI metadata table from KPI groups matching the term
# "utbidning" (education)
kpi_grp_df |>
  kpi_grp_search("utbildning") |>
```

```
  kpi_grp_unnest()
}
```

`kpi_minimize` *Simplify a KPI table*

Description

Remove all columns from a Kolada KPI metadata table that are monotonous across the table, i.e. columns that contain only one single value. Also remove undocumented columns, i.e. columns that contain unintelligible and undocumented information.

Usage

```
kpi_minimize(kpi_df, remove_monotonous_data = TRUE)
```

Arguments

`kpi_df`	A Kolada KPI metadata table, e.g. as created by `get_kpi()`.
`remove_monotonous_data`	Remove columns from the KPI table which contain exactly the same information for all entries in the table?

Value

A Kolada KPI metadata table

`kpi_search` *Search for Kolada KPIs using a Kolada KPI table*

Description

Search a Kolada KPI metadata table. Only keep rows that contain the search query. Matches against all columns or columns named with the `column` parameter. For more precise matching, please use dplyr::filter.

Usage

```
kpi_search(kpi_df, query, column = NULL)
```

Arguments

`kpi_df`	A Kolada KPI metadata table, e.g. as created by `get_kpi()`.
`query`	A search term or a vector of search terms to filter by. Case insensitive.
`column`	(Optional) A string or character vector with the names of columns in which to search for `query`.

Value

A Kolada KPI metadata table

Examples

```
if (kolada_available()) {
# Search for a single search term in a KPI table
kpis <- get_kpi(id = c("N11002", "N11003", "N11004", "N11005"))
kpi_filter <- kpi_search(kpis, "kostnad")

# Add keywords to a KPI table and search for multiple terms among
# the keywords
kpi_filter <- get_kpi(id = c("N11002", "N11003", "N11004", "N11005")) |>
  kpi_bind_keywords(n = 3) |>
  kpi_search(
    query = c("nettokostnad", "öppen"),
    column = c("keyword_1", "keyword_2", "keyword_3")
  )
}
```

`municipality_extract_ids` *Extract a vector of municipality ID strings from a Kolada municipality table*

Description

This function is primarily intended as a convenient way to pass a (filtered) Kolada municipality metadata table to `get_values()`.

Usage

```
municipality_extract_ids(munic_df)
```

Arguments

`munic_df`	A Kolada Municipality metadata table, as created by e.g. `get_municipality`.

Examples

```
if (kolada_available()) {
# Download Kolada data for all municipalities of type "L"
# (regions and national total) for KPI "N45933"
munic_filter <- get_municipality() |>
  municipality_search("L", column = "type")

kld_data <- get_values(
  kpi = "N45933",
```

```
  municipality = municipality_extract_ids(munic_filter),
  period = 2022
)
}
```

`municipality_grp_describe` *Describe the municipalitie in a Kolada Municipality Group metadata table*

Description

Print a human-readable description of each row of a Municipality Group metadata table, including member municipalities (up to a maximum number of rows). Can be printed either directly to the R console or used to populate a R markdown document, which can be useful for documentation purposes.

Usage

```
municipality_grp_describe(
  munic_grp_df,
  max_n = 5,
  format = "inline",
  heading_level = 2,
  sub_heading_level = heading_level + 1
)
```

Arguments

`munic_grp_df`	A Kolada Municipality Group metadata table, as created by e.g. `get_municipality_groups`.
`max_n`	The maximum number of KPI groups to describe.
`format`	Output format. Can be one of "inline" or "md" (markdown).
`heading_level`	The top heading level output format is "md".
`sub_heading_level`	The sub heading level output format is "md".

Value

Returns the object passed to the function, invisibly, to be re-used in a pipe.

municipality_grp_extract_ids

Extract municipality ID strings from a Kolada municipality group table

Description

This function is primarily intended as a convenient way to pass a (filtered) Kolada municipality group metadata table to `get_values()`. All IDs of the municipalities contained in each group in the table are extracted.

Usage

```
municipality_grp_extract_ids(munic_grp_df)
```

Arguments

`munic_grp_df`	A Kolada municipality group table, as created by e.g. `get_municipality_groups`.

Value

A vector of Municipality IDs.

municipality_grp_search

Search a Kolada Municipality Group metadata table for group names

Description

Search a Kolada Municipality Group metadata table. Only keep rows that contain the search query. Searches group titles and group IDs. Note that this function does not search for individual municipalities contained within municipality groups! To search for KPIs within a KPI group, see examples below for an example using `municipality_grp_unnest`.

Usage

```
municipality_grp_search(munic_grp_df, query)
```

Arguments

`munic_grp_df`	A Kolada Municipality Group metadata table, as created by e.g. `get_municipality_groups`.
`query`	A search term or a vector of search terms to filter by. Case insensitive.

Value

A Kolada Municipality Group metadata table

`municipality_grp_unnest` *Create a municipality table from a Kolada Municipality Group metadata table*

Description

Municipality groups are a convenient way to discover pre-rendered sets of municipalities. A practical workflow for discovering such sets can be to search through Municipality Group metadata using `municipality_grp_search()` to search for keywords and `municipality_grp_describe()` to inspect contents of KPI groups. Once you have created a Municipality Group metadata table that has been narrowed down to the group/s you are looking for, `municipality_grp_unnest()` is used to create a municipality metadata table for further processing.

Usage

```
municipality_grp_unnest(munic_grp_df)
```

Arguments

`munic_grp_df`	A Kolada Municipality Group metadata table, as created by e.g. `get_municipality_groups`.

Value

A Kolada Municipality metadata table

Examples

```
if (kolada_available()) {
# Download Municipality Group metadata
# (skip the parameter "max_results" to actually download all available data)
munic_grp_df <- get_municipality_groups(max_results = 100)

# Create a Municipality metadata table from municipality groups matching the
# term "Arboga"
munic_grp_df |>
  municipality_grp_search("arboga") |>
  municipality_grp_unnest()
}
```

`municipality_id_to_name` *Convert a vector of municipality ids to municipality names*

Description

Given a vector of municipality IDs/codes, return a named vector of names of municipalities or regions. Codes of municipalities and regions follow the Swedish standard for municipality codes. The codes extracted can be used e.g. to pass as a parameter to `get_values()`. This function is the inverse to `municipality_name_to_id()`.

Usage

```
municipality_id_to_name(munic_df, id, remove_na = FALSE)
```

Arguments

`munic_df`	A Kolada Municipality metadata table, as created by e.g. `get_municipality`.
`id`	ID ids of one or several municipalities. Allows repeats.
`remove_na`	Should NA return values be removed?

Value

A vector of Municipality names.

See Also

`municipality_extract_ids()`, `municipality_name_to_id()`

Examples

```
if (kolada_available()) {
munic_df <- get_municipality()
municipality_id_to_name(munic_df, c("1280", "1281", "0180", "1280"))
}
```

`municipality_name_to_id` *Convert a vector of municipality names to municipality ids*

Description

Given a vector of names of municipalities or regions, return a named vector of municipality IDs/codes. Codes of municipalities and regions follow the Swedish standard for municipality codes. The codes extracted can be used e.g. to pass as a parameter to `get_values()`. This function is the inverse to `municipality_id_to_name()`.

Usage

```
municipality_name_to_id(munic_df, municipality, remove_na = FALSE)
```

Arguments

`munic_df`	A Kolada Municipality metadata table, as created by e.g. `get_municipality`.
`municipality`	Name of one or several municipalities. Case insensitive. Allows repeats.
`remove_na`	Should NA return values be removed?

Value

A vector of Municipality IDs.

See Also

`municipality_extract_ids()`, `municipality_id_to_name()`

Examples

```
if (kolada_available()) {
munic_df <- get_municipality()
munic_df |>
  municipality_name_to_id(c("Arboga", "Lund", "Stockholm", "Arboga"))
}
```

`municipality_search` *Search a Kolada municipality metadata table*

Description

Search a Kolada municipality metadata table. Only keep rows that contain the search query. Note that some a quer might be both the name, or part of a name, of a municipality and part of the name of a region. Thus, a search might return rows for both municipalities and regions. To avoid this you can use dplyr::filter to filter the `type` column to keep only "K" (municipalities) or "L" (regions) rows. See also examples below for an alternative approach avoiding any direct calls to `filter`.

Usage

```
municipality_search(munic_df, query, column = NULL)
```

Arguments

`munic_df`	A Kolada Municipality metadata table, as created by e.g. `get_municipality`.
`query`	A search term or a vector of search terms to filter by. Case insensitive.
`column`	(Optional) A string or character vector with the names of columns in which to search for `query`.

Value

A Kolada Municipality metadata table

Examples

```
if (kolada_available()) {
# Search for a single search term in a municipality table
munic_df <- get_municipality()
municipality_search(munic_df, "Arboga")

# Only keep columns with type == "K" (keep municipalities, drop regions)
munic_filter <- get_municipality(cache = TRUE) |>
  municipality_search("K", column = "type")
}
```

`ou_search` *Search a Kolada Organizational Unit metadata table*

Description

Search a Kolada Organizational Unit metadata table. Only keep rows that contain the search query. Matches against all columns or columns named with the `column` parameter. For more precise matching, please use dplyr::filter.

Usage

```
ou_search(ou_df, query, column = NULL)
```

Arguments

`ou_df`	A Kolada Organizational Unit metadata table, as created by e.g. `get_municipality`.
`query`	A search term or a vector of search terms to filter by. Case insensitive.
`column`	(Optional) A string or character vector with the names of columns in which to search for `query`.

Value

A Kolada Organizational Unit metadata table

Examples

```
if (kolada_available()) {
# Search for all OUs matching the search term "skola" (school)
# (skip the parameter "max_results" to actually download all data)
ou_df <- get_ou(max_results = 100)
ou_search(ou_df, "skola")

# Only keep OU entities matching "skola" but not "förskola" (preschool)
# located in Gothenburg municipality and starting with an "A" using
# regex matching
ou_filter <- get_ou(municipality = "1480") |>
  ou_search("^A", column = "title") |>
  ou_search("[^(för)]skola")
}
```

values_legend *Create KPI long-form descriptions to add to a plot*

Description

In a Kolada values table, only KPI ID names are preserved. But in plots you often want to add a legend to explain what each KPI ID represents. But since KPI explanations are mostly relatively wordy, ggplot2 legends are under-dimensioned for this task. `values_legend` returns a string which can conveniently be used as caption to a plot instead.

Usage

```
values_legend(values_df, kpi_df)
```

Arguments

`values_df`	A Kolada value table, as created by `get_values()`.
`kpi_df`	A KPI table, e.g. as created by `get_kpi()`.

Value

A string which should be used as caption in a plot.

values_minimize *Simplify a Kolada values table*

Description

Simplify a Kolada values table, i.e as created by `get_values()`, by removing columns that contain monotonous data, i.e. that contain only one value for all observations.

Usage

```
values_minimize(values_df)
```

Arguments

`values_df`	A Kolada value table, as created by `get_values()`.

Value

A Kolada values table

Examples

```
# Download values for all available years of a given KPI for
# Malmö municipality (code 1280)
if (kolada_available()) {
vals <- get_values(kpi = "N45933", municipality = "1280", simplify = TRUE)
# (Returns a table with 5 rows and 8 columns)

# Remove columns with no information to differentiate between rows
values_minimize(vals)
# (Returns a table with 5 rows and 4 columns)
}
```

Index