

# Package ‘kofdata’

July 22, 2025

**Type** Package

**Version** 0.2.1

**Title** Get Data from the 'KOF Datenservice' API

**Description** Read Swiss time series data from the 'KOF Data' API, <<https://datenservice.kof.ethz.ch>>. The API provides macro economic time series data mostly about Switzerland. The package itself is a set of wrappers around the 'KOF Datenservice' API. The 'kofdata' package is able to consume public information as well as data that requires an API token.

**Depends** R (>= 3.0.0), jsonlite (>= 1.1), httr

**Imports** xts, zoo,

**Suggests** ttools, rmarkdown, testthat

**URL** <https://github.com/KOF-ch/kofdata>

**BugReports** <https://github.com/KOF-ch/kofdata/issues>

**Date** 2024-01-05

**License** GPL-2

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Matthias Bannert [aut, cre],  
Oliver Müller [aut],  
Severin Thoeni [aut],  
Diana Diaz [ctb]

**Maintainer** Matthias Bannert <bannert@kof.ethz.ch>

**Repository** CRAN

**Date/Publication** 2024-02-03 13:50:02 UTC

## Contents

cantons . . . . .	2
download_cached_file . . . . .	2
get_collection . . . . .	3
get_dataset . . . . .	4
get_legacy_key . . . . .	4
get_metadata . . . . .	5
get_remaining_quota . . . . .	5
get_time_series . . . . .	6
list_available_collections . . . . .	6
list_available_sets . . . . .	7
list_cached_files . . . . .	7
list_keys_in_collection . . . . .	8
list_keys_in_set . . . . .	8
list_public_keys . . . . .	9
param_defs . . . . .	9
translate_legacy_keys . . . . .	9

## Index 10

---

cantons	<i>Vector of Swiss cantons</i>
---------	--------------------------------

---

### Description

Vector of Swiss cantons

### Usage

```
cantons(full = F)
```

### Arguments

full            logical full names or two letter classification. defaults to FALSE.

---

download_cached_file	<i>get_dataset</i>
----------------------	--------------------

---

### Description

Download a predefined set of time series.

### Usage

```
download_cached_file(username, api_key, file_to_download, target = NULL)
```

**Arguments**

username	Your dataservice user name
api_key	<b>character</b> Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.
file_to_download	The name of the file to retrieve.
target	Path to the location, at which to store the file. If NULL, the file will be saved to the working directory.

**Examples**

```
f <- download_cached_file("kofdatapkg", "313984fcd9f343d3961891319b0ed321",  
"empty.txt", file.path(tempdir(), "empty.txt"))
```

---

get_collection	<i>Download Pre-Defined collection</i>
----------------	--

---

**Description**

Download a predefined collection of time series.

**Usage**

```
get_collection(collection_name, api_key = NULL, show_progress = FALSE)
```

**Arguments**

collection_name	The name of the collection you wish to download. For a list of available collections, go to ["list_available_collections"]
api_key	<b>character</b> Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.
show_progress	If collection to true, shows a progress bar of the data being downloaded.

**Examples**

```
get_collection("ds_kmi_mixed_freq", show_progress = TRUE)
```

---

get_dataset	<i>Download Pre-Defined Dataset</i>
-------------	-------------------------------------

---

### Description

This function is deprecated. Use [get\\_collection](#) instead. Download a predefined set of time series.

### Usage

```
get_dataset(set_name, api_key = NULL, show_progress = FALSE)
```

### Arguments

set_name	The name of the set you wish to download. For a list of available sets, go to ["list_available_sets"]
api_key	<b>character</b> Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.
show_progress	If set to true, shows a progress bar of the data being downloaded.

---

get_legacy_key	<i>Get a Former Time Series Identifier from KOF Database</i>
----------------	--

---

### Description

Get legacy keys given new time series keys. Note that not all records need to have a legacy key because there are more new keys than old keys.

### Usage

```
get_legacy_key(ts_keys)
```

### Arguments

ts_keys	A vector of time series keys
---------	------------------------------

### Value

A named list of legacy keys

---

get_metadata	<i>Get Localized Meta Information</i>
--------------	---------------------------------------

---

**Description**

Download metadata given a time series key.

**Usage**

```
get_metadata(ts_keys, locale = c("en", "de", "fr", "it"))
```

**Arguments**

ts_keys	A vector of time series keys.
locale	character ISO-2-digit language of requested metadata. Meta data are for survey related typically available in English, German, French and Italian. Non survey data are not provided with meta information, yet.

**Value**

A named list of lists containing metadata.

**Examples**

```
get_metadata("kofbarometer", "en")
```

---

get_remaining_quota	<i>Check Remaining Quota</i>
---------------------	------------------------------

---

**Description**

Query the API for the number of time series downloads remaining in the current month.

**Usage**

```
get_remaining_quota(api_key)
```

**Arguments**

api_key	<b>character</b> Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.
---------	---

**Examples**

```
get_remaining_quota("313984fcd9f343d3961891319b0ed321")
```

---

get\_time\_series      *Get Time Series form KOF Datenservice API*

---

### Description

Download time series data from the KOF web API.

### Usage

```
get_time_series(ts_keys, api_key = NULL, show_progress = FALSE)
```

### Arguments

ts_keys	A vector of timeseries keys
api_key	<b>character</b> Your API key. An API key is needed if accessing non-public timeseries. If downloading public timeseries parameter can be omitted.
show_progress	If set to true, shows a progress bar of the data being downloaded. year-month-day, otherwise only year and month.

### Examples

```
get_time_series("kofbarometer")
```

---

list\_available\_collections  
*List Available collection Names and Their Description*

---

### Description

collections are pre-defined lists of time series. collections are the convenient alternative to concatenating many series in an URL.

### Usage

```
list_available_collections(api_key = NULL)
```

### Arguments

api_key	<b>character</b> Your API key. An API key is needed if accessing non-public timeseries. If downloading public timeseries parameter can be omitted.
---------	--

### Value

A data frame with the names, descriptions and public availability of collections. These collections can be downloaded via get\_collection.

**Examples**

```
list_available_collections()
```

---

```
list_available_sets    List Available Set Names and Their Description
```

---

**Description**

This function is deprecated. Use [list\\_available\\_collections](#) instead. Sets are pre-defined lists of time series. Sets are the convenient alternative to concatenating many series in an URL.

**Usage**

```
list_available_sets(api_key = NULL)
```

**Arguments**

`api_key` **character** Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.

**Value**

A data frame with the names, descriptions and public availability of datasets. These sets can be downloaded via `get_dataset`.

---

```
list_cached_files    Show Cached Files of Specific User
```

---

**Description**

Some users get individually composed, cached files. If you ordered such a service you can list all files available to you.

**Usage**

```
list_cached_files(username, api_key)
```

**Arguments**

`username` Your dataservice user name  
`api_key` **character** Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.

**Examples**

```
available_files <- list_cached_files("kofdatapkg",
  "313984fcd9f343d3961891319b0ed321")
available_files
```

---

```
list_keys_in_collection
```

*List All Keys in a Collection*

---

### Description

List the keys of all time series in a collection. To learn more about specific keys, use `get_metadata`.

### Usage

```
list_keys_in_collection(collectionname, api_key = NULL)
```

### Arguments

`collectionname` The name of the collection

`api_key` **character** Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.

### Value

If a single collection name is provided, a vector of time series keys. If multiple collection names are provided, a list of vectors of time series keys.

### Examples

```
list_keys_in_collection("ds_kmi_mixed_freq")
```

---

```
list_keys_in_set
```

*List All Keys in a Set*

---

### Description

List the keys of all time series in a set. To learn more about specific keys, use `get_metadata`.

### Usage

```
list_keys_in_set(setname, api_key = NULL)
```

### Arguments

`setname` The name of the set

`api_key` **character** Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.

### Value

If a single set name is provided, a vector of time series keys. If multiple set names are provided, a list of vectors of time series keys.



---

list_public_keys	<i>List Keys for All Publicly Available Time Series</i>
------------------	---

---

**Description**

This function will be removed from future versions of kofdata.

**Usage**

```
list_public_keys()
```

**Value**

An array of all public time series keys.

---

param_defs	<i>Common parameters</i>
------------	--------------------------

---

**Description**

Common parameters

**Arguments**

api_key	<b>character</b> Your API key. An API key is needed if accessing non-public time-series. If downloading public timeseries parameter can be omitted.
---------	---

---

translate_legacy_keys	<i>translate_legacy_keys</i>
-----------------------	------------------------------

---

**Description**

Gives ts\_keys for the given legacy keys.

**Usage**

```
translate_legacy_keys(legacy_keys, chunksize = 100)
```

**Arguments**

legacy_keys	A vector of legacy keys for which to find new keys
chunksize	integer to set chunksize in order to avoid URLs that are too long.

**Value**

A list of vectors of new ts\_keys

# Index

cantons, [2](#)

download\_cached\_file, [2](#)

get\_collection, [3](#), [4](#)

get\_dataset, [4](#)

get\_legacy\_key, [4](#)

get\_metadata, [5](#)

get\_remaining\_quota, [5](#)

get\_time\_series, [6](#)

list\_available\_collections, [6](#), [7](#)

list\_available\_sets, [7](#)

list\_cached\_files, [7](#)

list\_keys\_in\_collection, [8](#)

list\_keys\_in\_set, [8](#)

list\_public\_keys, [9](#)

param\_defs, [9](#)

translate\_legacy\_keys, [9](#)