

# Package ‘csodata’

August 14, 2020

**Type** Package

**Title** Download Data from the CSO 'StatBank' API

**Version** 1.0.2

**Date** 2020-08-12

**Maintainer** Conor Crowley <conor.crowley@csodata.ie>

**Description** Imports 'StatBank' data in JSON-stat format and (optionally) reshapes it into wide format. The Central Statistics Office (CSO) is the national statistical institute of Ireland and 'StatBank' is the CSO's online database of Official Statistics. This database contains current and historical data series compiled from CSO statistical releases and is accessed at <<http://www.cso.ie/px/pxeirestat/statire/SelectTable/Omrade0.asp?Planguage=0>>. A tutorial explaining and illustrating how to navigate to the 'StatBank' and how to search the 'StatBank' by keyword or theme can be found here: <<https://www.cso.ie/en/interactivezone/youtubevideos/statbanktutorial/>>. This tutorial also demonstrates how a user can create, edit and download a table from the 'StatBank'. The CSO 'StatBank' Application Programming Interface (API), which is accessed in this package, provides access to 'StatBank' data in JSON-stat format at <<https://statbank.cso.ie/webserviceclient/>>. This dissemination tool allows developers machine to machine access to CSO 'StatBank' data.

**Imports** dplyr, httr, jsonlite, reshape2, rjstat, R.cache, sf

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.0.2

**Suggests** knitr, rmarkdown, tmap, viridisLite

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Eoin Horgan [aut] (<<https://orcid.org/0000-0002-3446-6154>>),  
Conor Crowley [aut, cre],  
Mervyn O'Luing [aut]

**Repository** CRAN

**Date/Publication** 2020-08-14 13:10:02 UTC

R topics documented:

csodata . . . . .	2
cso_clear_cache . . . . .	3
cso_disp_geo_meta . . . . .	3
cso_disp_meta . . . . .	4
cso_get_content . . . . .	4
cso_get_data . . . . .	5
cso_get_geo . . . . .	6
cso_get_geo_meta . . . . .	7
cso_get_interval . . . . .	8
cso_get_meta . . . . .	8
cso_get_toc . . . . .	9
cso_get_vars . . . . .	10
cso_get_var_values . . . . .	11
cso_search_toc . . . . .	11
<b>Index</b>	<b>13</b>

---

csodata	<i>csodata: A package for downloading CSO data.</i>
---------	---

---

Description

The csodata package allows for easily downloading CSO (Central Statistics Office, the National Statistics Institute of Ireland) StatBank data into R.

Details

A specific table can be downloaded using [cso\\_get\\_data](#), while a list of all tables currently available and their titles can be found using [cso\\_get\\_toc](#) and [cso\\_search\\_toc](#) is used to search their descriptions. Metadata for a specified table can be retrieved with [cso\\_get\\_meta](#), or printed on the console using [cso\\_disp\\_meta](#).

[cso\\_get\\_vars](#), [cso\\_get\\_interval](#), and [cso\\_get\\_content](#) all return a subset of the full metadata of a table. [cso\\_get\\_var\\_values](#) returns all the variables in the tables, as well as the values they have.

These functions provide the option to cache the returned data using the R.cache package. The cache can be deleted using [cso\\_clear\\_cache](#).

ESRI shapefiles covering the country in varying degrees of granularity can be downloaded from [cso.ie](#) and imported as an sf data frame using the [cso\\_get\\_geo](#) function. Metadata about the map data can be retrieved with [cso\\_get\\_geo\\_meta](#), and displayed on the console with [cso\\_disp\\_geo\\_meta](#).

---

cso_clear_cache	<i>Clear csodata cache</i>
-----------------	----------------------------

---

### Description

Deletes all data cached by the csodata package. The cached data from the csodata package is stored in a subdirectory of the default R.cache cache at `R.cache::getCachePath()`. This function provides a quick way to delete those files and free up space.

### Usage

```
cso_clear_cache()
```

### Details

This brings up a [Y/N] prompt with the number of files and directories that will be deleted, asking the user to confirm their intent.

### Examples

```
## Not run:  
cso_clear_cache()  
  
## End(Not run)
```

---

cso_disp_geo_meta	<i>Prints metadata from an ESRI shapefile to console</i>
-------------------	--

---

### Description

Takes the output from `cso_get_geo` or otherwise and prints information about it to the console as formatted text.

### Usage

```
cso_disp_geo_meta(shp)
```

### Arguments

shp	sf data.frame. Geographic data stored as an sf object. StatBank.
-----	--

### Examples

```
## Not run:  
cso_disp_geo_meta(shp)  
  
## End(Not run)
```

---

cso_disp_meta	<i>Prints metadata from a StatBank table to the console</i>
---------------	---

---

### Description

Takes the output from `cso_get_meta` and prints it to the console as formatted text.

### Usage

```
cso_disp_meta(table_code)
```

### Arguments

table\_code      string. A valid code for one table from the StatBank.

### Examples

```
## Not run:
cso_disp_meta("EP001")

## End(Not run)
```

---

cso_get_content	<i>Returns a character vector listing the statistics in a CSO data table</i>
-----------------	--

---

### Description

Uses the ContentsIndicatorsList service of the CSO Statbank API to returns a list the contained statistics for a table. See <https://statbank.cso.ie/StatbankServices/StatbankServices.svc/jsonservice/help/operations/ContentsIndicatorsList> for more information on this.

### Usage

```
cso_get_content(table_code)
```

### Arguments

table\_code      string. A valid code for one table from the StatBank.

### Value

character vector. The names of the statistics included in the table, with one element for each statistic.

### Examples

```
## Not run:
cso_get_content("EP008")

## End(Not run)
```

---

cso\_get\_data

*Return a CSO table as a data frame*


---

### Description

Returns a CSO table from the CSO StatBank Application Programming Interface (API) as a data frame, with the option to give it in wide format (default) very wide or long/tidy format.

### Usage

```
cso_get_data(table_code, wide_format = "wide", include_ids = TRUE,
             id_list = NULL, use_factors = TRUE, cache = TRUE)
```

### Arguments

table_code	string. If the table_code is a filename or a path to a file, e.g. "QNQ22.json", it is imported from that file. Otherwise if it is only a table code e.g. "QNQ22", the file is downloaded from StatBank and checked to see if it is a valid table.
wide_format	string, one of "wide", "very_wide" or "tall". If "wide" (default) the table is returned in wide (human readable) format, with statistic as a column (if it exists). If "very_wide" the table is returned wide format and spreads the statistic column to rows. If "tall" the table is returned in tall/tidy format.
include_ids	logical. The JSON-stat format stores variables as ids i.e. IE11 and labels i.e. Border. While the label is generally preferred, sometimes it is useful to have the ids to match on. If include_ids is TRUE (default) then ids are retrieved and appended to the table to the right of the original column with the name <columnName>.id.
id_list	either NULL (default) or a character vector of columns that should have ids appended if include_ids is TRUE. if NULL then every column that is not included in the vector remove_id will be used.
use_factors	logical. If TRUE (default) factors will be used in strings.
cache	logical. if TRUE (default) csodata will cache the result using R.cache. The raw data downloaded from the statbank is cached, which means that calling cso_get_data with the same table_code but different parameters will result in cached data being used.

### Details

The data is pulled from the ResponseInstance service on the CSO API in JSON-Stat format, using the GET method from the httr package.

### Value

data frame of the requested CSO table.

## Examples

```
## Not run:
tbl1 <- cso_get_data("QNQ22")
tbl2 <- cso_get_data("QLF07.json")

## End(Not run)
```

---

cso_get_geo	<i>Return geographic data as a sf data frame</i>
-------------	--

---

## Description

Retrieves an ESRI shapefile of vector data for Ireland from the cso website [cso.ie](http://cso.ie) and returns it as an sf data frame. The data is returned as a zip file, which is downloaded to and unzipped in a temporary directory.

## Usage

```
cso_get_geo(map_data, cache = TRUE)
```

## Arguments

map_data	<p>string. Indicates which shapefile to download. Options are:</p> <ul style="list-style-type: none"> <li>• "Provinces" OR "p",</li> <li>• "NUTS2",</li> <li>• "NUTS3",</li> <li>• "NUTS2_2011",</li> <li>• "NUTS3_2011",</li> <li>• "Administrative Counties" OR "admin_counties" OR "ac",</li> <li>• "Electoral Divisions" OR "elec_div" OR "ed",</li> <li>• "Small Areas" OR "sa" and</li> <li>• "Gaeltacht" OR "g".</li> </ul> <p>Until v0.1.5 "NUTS2" and "NUTS3" gave access to the 2011 dataset.</p>
cache	<p>logical. Indicates whether to cache the result using R.cache. TRUE by default.</p>

## Details

The map data is from the 2011 census, and is 20m generalised, which offers a good balance of fidelity and low file size. More datasets, as well as 50m generalised, 100m generalised and ungeneralised versions of the map files can also be found on the OSi (Ordnance Survey Ireland) website at <https://data-osi.opendata.arcgis.com/search?tags=boundaries>.

The NUTS2 and NUTS3 map files are the updated versions for 2016, including three NUTS2 regions and the movement of Louth and South Tipperary into new NUTS3 regions. These files are downloaded directly from the OSi website, as they are not available on the CSO website, and do not contain the population and housing data contained in the map files from the CSO website.

**Value**

data frame of the requested CSO table.

**Examples**

```
## Not run:  
shp <- cso_get_geo("NUTS2")  
  
## End(Not run)
```

---

cso_get_geo_meta	Returns a data frame with the metadata of a vector shapefile
------------------	--

---

**Description**

Takes the output from [cso\\_get\\_geo](#) or otherwise and returns information about it in a data frame.

**Usage**

```
cso_get_geo_meta(shp)
```

**Arguments**

shp                      sf data.frame. Geographic data stored as an sf object.

**Value**

list with eight elements:

- The coordinate reference system, itself a list with two elements, the EPSG code (if any, NA value if none), and the proj4string
- The number of polygons in the data
- If all the polygons are simple (not self-intersecting)
- If any polygons are empty
- If all of the polygons are valid
- The average area of the polygons, including units

**Examples**

```
## Not run:  
shp_meta <- cso_get_geo_meta(shp)  
  
## End(Not run)
```

---

cso_get_interval	<i>Returns a the time interval used to record data in a CSO table</i>
------------------	---

---

### Description

Reads the metadata of a table to return an atomic character vector displaying the intervals at which the data included in the table was gathered/calculated.

### Usage

```
cso_get_interval(table_code, cache_data = TRUE)
```

### Arguments

table_code	string. A valid code for one table from the StatBank.
cache_data	logical. Whether to use cached data, if available. Default value is TRUE. Strongly recommended to use caching, as otherwise the entire table could be downloaded only to access a small part of its metadata.

### Value

character vector. The names of the statistics included in the table, with one element for each statistic.

### Examples

```
## Not run:
cso_get_interval("C0636")

## End(Not run)
```

---

cso_get_meta	<i>Returns a data frame with the metadata of a CSO data table</i>
--------------	---

---

### Description

Checks the CSO Statbank API for a metadata on a dataset and returns it as a list of metadata and contained statistics.

### Usage

```
cso_get_meta(table_code, cache_data = TRUE, cache_toc = TRUE)
```



**Arguments**

table_code	string. A valid code for a table from the StatBank.
cache_data	logical. Whether to use cached data, if available. Default value is TRUE.
cache_toc	logical. Whether to use cached table of contents, if available. Default value is TRUE.

**Details**

The statistics are pulled using the [cso\\_get\\_content](#) function.

**Value**

list with eight elements:

- The title of the table.
- The source of the data.
- The date when the data was retrieved from StatBank.
- The time interval used in the data. (Census year, Quarter, Month)
- The units used (the R class of the value column)
- The date the table was last modified.
- The names of the variables included in the table, returned as a character vector with one element for each variable.
- The names of the statistics included in the table, returned as a character vector with one element for each statistic.

**Examples**

```
## Not run:
head(cso_get_meta("VSA16"))
meta1 <- cso_get_meta("HS014")

## End(Not run)
```

---

cso_get_toc	<i>Returns a data frame with all valid CSO Statbank tables listed sequentially by id number, e.g. A0101, A0102, A0103, etc.</i>
-------------	---

---

**Description**

Checks the CSO Statbank API for a list of all the table codes (e.g. A0101, A0102, A0103, etc.), which also includes date last modified and title for each table, and returns this list as an R data frame.

**Usage**

```
cso_get_toc(cache = TRUE, suppress_messages = FALSE)
```

**Arguments**

- `cache` logical. If TRUE (default) the table of contents is cached with the system date as a key.
- `suppress_messages` logical. If FALSE (default) a message is printed when loading a previously cached table of contents.

**Details**

The data is pulled from the TitledetailsList on the CSO API. See <https://statbank.cso.ie/StatbankServices/StatbankServices.svc/jsonservice/help/operations/TitledetailsList> for more information on this.

**Value**

data frame of three character columns:

- `id`. Contains all of the table codes currently available on the CSO API.
- `LastModified`. The date the table was last modified in POSIXct format.
- `title`. The title of the table.

**Examples**

```
head(cso_get_toc())
```

---

<code>cso_get_vars</code>	<i>Returns a character vector listing the contents of a CSO data table</i>
---------------------------	--

---

**Description**

Reads the metadata of a table to return a character vector of the included variables in the table.

**Usage**

```
cso_get_vars(table_code, cache_data = TRUE)
```

**Arguments**

- `table_code` string. A valid code for one table from the StatBank.
- `cache_data` logical. Whether to use cached data, if available. Default value is TRUE. Strongly recommended to use caching, as otherwise the entire table could be downloaded only to access a small part of its metadata.

**Value**

character vector. The names of the statistics included in the table, with one element for each statistic.

**Examples**

```
cso_get_vars("IPA03")
```

---

cso_get_var_values	<i>Returns a list of the values of variables of a CSO data table</i>
--------------------	--

---

**Description**

Reads the table to determine all the unique values taken by the variables in the table and returns them as a list.

**Usage**

```
cso_get_var_values(table_code, cache_data = TRUE)
```

**Arguments**

table_code	string. A valid code for one table from the StatBank.
cache_data	logical. Whether to use cached data, if available. Default value is TRUE.

**Value**

list. It has length equal to the number of variables in the table, and each element is a character vector which has all the values taken by one variable.

**Examples**

```
var_val <- cso_get_var_values("IPA03")
```

---

cso_search_toc	<i>Search list of all table descriptions for given string</i>
----------------	---

---

**Description**

Searches the list of all table descriptions returned by `cso_get_toc()` for a given substring.

**Usage**

```
cso_search_toc(string, toc = cso_get_toc(suppress_messages = TRUE))
```

**Arguments**

string	string. The text to search for. Case insensitive.
toc	data.frame. The table of contents as returned by <code>cso_get_toc</code> . If not given, will be re-downloaded (or retrieved from cache) using <code>cso_get_toc()</code> .

**Value**

data frame of three character columns, with layout identical to that of `cso_get_toc`. A subset of the results of `cso_get_toc`, with only rows where the description field contains the entered string.

**Examples**

```
trv <- cso_search_toc("travel")
```

# Index

`cso_clear_cache`, [2](#), [3](#)  
`cso_disp_geo_meta`, [2](#), [3](#)  
`cso_disp_meta`, [2](#), [4](#)  
`cso_get_content`, [2](#), [4](#), [9](#)  
`cso_get_data`, [2](#), [5](#)  
`cso_get_geo`, [2](#), [3](#), [6](#), [7](#)  
`cso_get_geo_meta`, [2](#), [7](#)  
`cso_get_interval`, [2](#), [8](#)  
`cso_get_meta`, [2](#), [4](#), [8](#)  
`cso_get_toc`, [2](#), [9](#)  
`cso_get_var_values`, [2](#), [11](#)  
`cso_get_vars`, [2](#), [10](#)  
`cso_search_toc`, [2](#), [11](#)  
`csodata`, [2](#)  
`csodata-package` (`csodata`), [2](#)