

Package ‘comtradr’

October 24, 2017

Title Interface with the United Nations Comtrade API

Version 0.1.0

Description Interface with and extract data from the United Nations Comtrade API <<https://comtrade.un.org/data/>>. Comtrade provides country level shipping data for a variety of commodities, these functions allow for easy API query and data returned as a tidy data frame.

Depends R (>= 3.0.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports httr, jsonlite, magrittr (>= 1.5), methods, purrr

RoxygenNote 6.0.1

URL <https://github.com/ropensci/comtradr>

BugReports <https://github.com/ropensci/comtradr/issues>

NeedsCompilation no

Author Chris Muir [aut, cre]

Maintainer Chris Muir <chrismuirRVA@gmail.com>

Suggests testthat, knitr, rmarkdown, dplyr, ggplot2

VignetteBuilder knitr

Repository CRAN

Date/Publication 2017-10-24 03:30:16 UTC

R topics documented:

comtradr	2
ct_commodity_db_type	3
ct_commodity_lookup	4
ct_country_lookup	5
ct_get_remaining_hourly_queries	6

ct_get_reset_time	6
ct_pretty_cols	7
ct_register_token	7
ct_search	8
ct_update_databases	10
ct_use_pretty_cols	12

Index	13
--------------	-----------

comtradr *Interface to the United Nations Comtrade API*

Description

Interface with and extract data from the United Nations Comtrade API. Comtrade provides country level shipping data for a variety of commodities, these functions allow for easy API query and data returned as a tidy data frame.

Package Vignette

- [../doc/comtradr-vignette.html](#)

Documentation for the Comtrade API

- Main Comtrade Site <https://comtrade.un.org/>
- Comtrade Data Query Web GUI <https://comtrade.un.org/data/>
- Full API Documentation <https://comtrade.un.org/data/doc/api/>

Development links

- <https://github.com/ChrisMuir/comtradr>
- Report bugs at <https://github.com/ChrisMuir/comtradr/issues>

comtradr features the following functions

- [ct_commodity_db_type](#)
- [ct_commodity_lookup](#)
- [ct_country_lookup](#)
- [ct_get_remaining_hourly_queries](#)
- [ct_get_reset_time](#)
- [ct_register_token](#)
- [ct_search](#)
- [ct_update_databases](#)
- [ct_use_pretty_cols](#)

ct_commodity_db_type *Get current commodity database type*

Description

Return the "type" of the current commodity database being used by comtrade. For a complete list of the different commodity DB types, see "details".

Usage

```
ct_commodity_db_type()
```

Details

Below is a list of all of the commodity database "types", with a very brief description for each. For more information on each of these types, see <https://comtrade.un.org/data/doc/api/#DataAvailabilityRequests>

- HS: Harmonized System (HS), as reported
- HS1992: HS 1992
- HS1996: HS 1996
- HS2002: HS 2002
- HS2007: HS 2007
- HS2012: HS 2012
- SITC: Standard International Trade Classification (SITC), as reported
- SITCrev1: SITC Revision 1
- SITCrev2: SITC Revision 2
- SITCrev3: SITC Revision 3
- SITCrev4: SITC Revision 4
- BEC: Broad Economic Categories
- EB02: Extended Balance of Payments Services Classification

Value

character vector of the "type" of the current commodity database.

Examples

```
ct_commodity_db_type()
```

ct_commodity_lookup *UN Comtrade commodities database query*

Description

The Comtrade API requires that searches for specific commodities be done using commodity codes. This is a helper function for querying the Comtrade commodity database. It takes as input a vector of commodities or commodity codes. Output is a list or vector of commodity descriptions or codes associated with the input search_terms. For use with the UN Comtrade API, full API docs can be found at <https://comtrade.un.org/data/doc/api/>

Usage

```
ct_commodity_lookup(search_terms, return_code = FALSE, return_char = FALSE,  
  verbose = TRUE, ignore.case = TRUE, ...)
```

Arguments

search_terms	Commodity names or commodity codes, as a char or numeric vector.
return_code	Logical, if set to FALSE, the function will return a set of commodity descriptions along with commodity codes (as a single string for each match found), if set to TRUE it will return only the commodity codes. Default value is FALSE.
return_char	Logical, if set to FALSE, the function will return the matches as a named list, if set to TRUE it will return them as a character vector. Default value is FALSE.
verbose	Logical, if set to TRUE, a warning message will print to console if any of the elements of input "search_terms" returned no matches (message will indicate which elements returned no data). Default is TRUE.
ignore.case	logical, to be passed along to arg ignore.case within grepl . Default value is TRUE.
...	additional args to be passed along to grepl .

Details

This function uses regular expressions (regex) to find matches within the commodity DB. This means it will treat as a match any commodity description that contains the input search term. For more on using regex within R, see this great tutorial by Gloria Li and Jenny Bryan http://stat545.com/block022_regular-expression.html

Value

A list or character vector of commodity descriptions and/or commodity codes that are matches with the elements of "search_terms".

See Also

[grepl](#)

Examples

```
# Look up commodity descriptions related to "halibut"
ct_commodity_lookup("halibut",
                    return_code = FALSE,
                    return_char = FALSE,
                    verbose = TRUE)

# Look up commodity codes related to "shrimp".
ct_commodity_lookup("shrimp",
                    return_code = TRUE,
                    return_char = FALSE,
                    verbose = TRUE)
```

ct_country_lookup *UN Comtrade country database query*

Description

Country names passed to the Comtrade API must have precise spelling/capitalization. This is a helper function for querying the country names/spelling used by Comtrade.. It takes as input a vector of country names, output is any country names that contain any of the input strings, using regex via the base function `grepl`. For use with the UN Comtrade API, full API docs can be found at <https://comtrade.un.org/data/doc/api/>

Usage

```
ct_country_lookup(search_terms, type = c("reporter", "partner"),
                  ignore.case = TRUE, ...)
```

Arguments

<code>search_terms</code>	Char vector of country names.
<code>type</code>	str, the country list to use for the search, valid inputs are "reporter" and "partner".
<code>ignore.case</code>	logical, to be passed along to arg <code>ignore.case</code> within <code>grepl</code> . Default value is TRUE.
<code>...</code>	additional args to be passed along to <code>grepl</code> .

Details

This function uses regular expressions (regex) to find matches within the country DB. This means it will treat as a match any country string that contains the input search term. For more on using regex within R, see this great tutorial by Gloria Li and Jenny Bryan http://stat545.com/block022_regular-expression.html

Value

A character vector of country names that are complete or partial matches with any of the input country names.

See Also

[grepl](#)

Examples

```
# Look up all reporters that contain the terms "korea" and "vietnam"
ct_country_lookup(c("korea", "vietnam"), "reporter")
```

```
ct_get_remaining_hourly_queries
      Comtradr rate limit check
```

Description

Get the remaining number of queries left in the current hour.

Usage

```
ct_get_remaining_hourly_queries()
```

Value

numeric value, number of current queries left in the hour.

Examples

```
ct_get_remaining_hourly_queries()
```

```
ct_get_reset_time      Comtradr rate limit time check
```

Description

Get the time in which the hourly limit will reset.

Usage

```
ct_get_reset_time()
```

Value

date and time in which the hourly query limit will reset. Return is a "POSIXct" object (see [DateTimeClasses](#)).

Examples

```
ct_get_reset_time()

# Get minutes remaining until limit reset, as numeric value.
as.double(ct_get_reset_time() - Sys.time())
```

ct_pretty_cols	<i>"pretty" column headers for Comtrade API data.</i>
----------------	---

Description

Named vector of polished column headers, intended for use with plots, publication tables, etc.

Format

Named vector, with the polished column headers as the names, and the machine-readable column headers as the values. Each element is meant to be treated as a key-value pair. The function [ct_search](#) returns data with the machine-readable column headers by default.

Examples

```
data(ct_pretty_cols)
```

ct_register_token	<i>Comtradr set API token</i>
-------------------	-------------------------------

Description

Function to set an API token for the UN Comtrade API. Details on tokens and rate limits can be found <https://comtrade.un.org/data/doc/api/#Authentication>

Usage

```
ct_register_token(token)
```

Arguments

token char string, valid API token.

Value

Set comtradr API token and update rate limits.

Examples

```
## Not run:
ct_register_token("some_valid_token_str")

## End(Not run)
```

ct_search

*Get UN Comtrade data***Description**

Make queries to the UN Comtrade API, data is returned as a tidy data frame. Comtrade is a DB hosted by the United Nations that houses country-level shipping data. Full API docs can be found here: <https://comtrade.un.org/data/doc/api/>

Usage

```
ct_search(reporters, partners, trade_direction = c("all", "imports",
  "exports", "re_imports", "re_exports"), freq = c("annual", "monthly"),
  start_date = "all", end_date = "all", commod_codes = "TOTAL",
  max_rec = NULL, type = c("goods", "services"),
  url = "https://comtrade.un.org/api/get?")
```

Arguments

reporters	Country(s) of interest, as a character vector. Can either be a vector of country names, or "All" to represent all countries.
partners	Country(s) that have interacted with the reporter country(s), as a character vector. Can either be a vector of country names, or "All" to represent all countries.
trade_direction	Indication of which trade directions on which to focus, as a character vector. Must either be "all", or a vector containing any combination of the following: "imports", "exports", "re_imports", "re_exports". Default value is "all".
freq	Time frequency of the returned results, as a character string. Must be either "annual" or "monthly". Default value is "annual".
start_date	Start date of a time period, or "all". Default value is "all". If inputting a date, must be string w/ structure "yyyy-mm-dd".
end_date	End date of a time period, or "all". Default value is "all". If inputting a date, must be string w/ structure "yyyy-mm-dd".
commod_codes	Character vector of commodity codes, or "TOTAL". Valid commodity codes as input will restrict the query to only look for trade related to those commodities, "TOTAL" as input will return all trade between the indicated reporter country(s) and partner country(s). Default value is "TOTAL".
max_rec	Max number of records returned from each API call, as an integer. If max_rec is set to NULL, then value is determined by whether or not an API token has been registered. API cap without a token is 50000, cap with a valid token is 250000. Default value is NULL. For details on how to register a valid token, see ct_register_token .
type	Type of trade, as a character string. Must be either "goods" or "services". Default value is "goods".
url	Base of the Comtrade url string, as a character string. Default value is "https://comtrade.un.org/api/get?" and should not be changed unless Comtrade changes their endpoint url.


```
# Make API call
ex_2 <- ct_search(reporters = "Canada",
                  partners = "All",
                  trade_direction = "all",
                  start_date = "2011-01-01",
                  end_date = "2015-01-01",
                  commod_codes = shrimp_codes)

nrow(ex_2)

# Access metadata
attributes(ex_2)$url
attributes(ex_2)$time_stamp
attributes(ex_2)$req_duration

## End(Not run)
```

ct_update_databases *Check for updates to country/commodity databases*

Description

Use of the Comtrade API requires access to the Comtrade countries database and commodities database. The `comtradr` package keeps each DB saved as a data frame in the package directory, as Comtrade makes updates to these DB's infrequently (roughly once per year).

Usage

```
ct_update_databases(force = FALSE, verbose = TRUE,
                   commodity_type = c("HS", "HS1992", "HS1996", "HS2002", "HS2007", "HS2012",
                                       "SITC", "SITCrev1", "SITCrev2", "SITCrev3", "SITCrev4", "BEC", "EB02"),
                   commodity_url = NULL, reporter_url = NULL, partner_url = NULL)
```

Arguments

<code>force</code>	logical, if TRUE, both the country and commodity databases will be downloaded, regardless of the status of the DB's on file. Default value is FALSE.
<code>verbose</code>	logical, if TRUE, an update status message will be printed to console. Default value is TRUE.
<code>commodity_type</code>	Trade data classification scheme to use, see "details" for a list of the valid inputs. Default value is "HS", which is the default "type" of the commodity database on file upon install of <code>comtradr</code> . Please note that if the value passed to this arg doesn't match the values in variable "type" of the current commodity DB, then this function will replace the current commodity DB with that of the type specified by this arg. If you don't intend to change the type of the current commodity DB, then no input for this arg is required. To see the "type" of the current commodity DB, use <code>ct_commodity_db_type</code> .

commodity_url	Default value NULL, otherwise this should be the base url of the Comtrade json data directory. Only necessary if the Comtrade site changes from "https://comtrade.un.org/data/cache/". This partial url string will have a commodity extension appended to it to create a valid url. The commodity extension will be chosen based on the input to arg commodity_type.
reporter_url	Default value NULL, otherwise this should be a url as a char string that points to the reporter areas JSON dataset on the Comtrade website. Only necessary if the Comtrade site changes from https://comtrade.un.org/data/cache/reporterAreas.json
partner_url	Default value NULL, otherwise this should be a url as a char string that points to the reporter areas JSON dataset on the Comtrade website. Only necessary if the Comtrade site changes from https://comtrade.un.org/data/cache/partnerAreas.json

Details

This function will check to see if Comtrade has made any updates to either database. If an update is found, it will download the updated DB and save it to the comtradr package directory, and update the DB for use within the current R session.

The default for arg commodity_type is HS. Below is a list of all valid inputs with a very brief description for each, for more information on each of these types, see <https://comtrade.un.org/data/doc/api/#DataAvailabilityRequests>

- HS: Harmonized System (HS), as reported
- HS1992: HS 1992
- HS1996: HS 1996
- HS2002: HS 2002
- HS2007: HS 2007
- HS2012: HS 2012
- SITC: Standard International Trade Classification (SITC), as reported
- SITCrev1: SITC Revision 1
- SITCrev2: SITC Revision 2
- SITCrev3: SITC Revision 3
- SITCrev4: SITC Revision 4
- BEC: Broad Economic Categories
- EB02: Extended Balance of Payments Services Classification

Value

Updated database of commodities and countries.

Examples

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

ct_use_pretty_cols *Use Pretty Column Headers*

Description

Transform the column headers of return data from function `ct_search` into a more "polished" set of column headers. Intended for use with plots, publication tables, etc.

Usage

```
ct_use_pretty_cols(df)
```

Arguments

df data frame, Comtrade API data frame, returned from function `ct_search`.

Value

data frame, input df with polish column headers.

Examples

```
## Not run:  
# Pull API data  
df <- ct_search("Germany", "Canada")  
  
# Use polished column names  
df <- ct_use_pretty_cols(df)  
  
## End(Not run)
```

Index

attributes, [9](#)

comtradr, [2](#)

comtradr-package (comtradr), [2](#)

ct_commodity_db_type, [2](#), [3](#), [10](#)

ct_commodity_lookup, [2](#), [4](#)

ct_country_lookup, [2](#), [5](#)

ct_get_remaining_hourly_queries, [2](#), [6](#)

ct_get_reset_time, [2](#), [6](#)

ct_pretty_cols, [7](#)

ct_register_token, [2](#), [7](#), [8](#), [9](#)

ct_search, [2](#), [7](#), [8](#), [12](#)

ct_update_databases, [2](#), [10](#)

ct_use_pretty_cols, [2](#), [12](#)

DateTimeClasses, [6](#)

grepl, [4-6](#)