Package 'comtradr'

October 5, 2018

Title Interface with the United Nations Comtrade API

Version 0.2.2

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), purrr

RoxygenNote 6.1.0

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

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

NeedsCompilation no

Maintainer Chris Muir <chrismuirRVA@gmail.com>

Suggests testthat, knitr, rmarkdown, ggplot2, dplyr

VignetteBuilder knitr

Author Chris Muir [aut, cre],
Alicia Schep [rev] (<https://orcid.org/0000-0002-3915-0618>, Alicia reviewed the package for rOpenSci, see https://github.com/ropensci/onboarding/issues/141),
Rafael Hellwig [rev] (<https://orcid.org/0000-0002-3092-3493>, Rafael reviewed the package for rOpenSci, see https://github.com/ropensci/onboarding/issues/141)

Repository CRAN

Date/Publication 2018-10-05 05:30:02 UTC

R topics documented:

`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_commodity_db_type` *Get current commodity database type*

Description

Return the "type" of the current commodity database being used by `comtradr`. 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

`search_terms`	Char vector of country names.
`type`	str, the country list to use for the search, valid inputs are "reporter" and "partner".
`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 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 *"pretty" column headers for Comtrade API data.*

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 *Comtradr set API token*

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". See "details" for more info on valid input formats when not using "all" as input.
end_date	End date of a time period, or "all". Default value is "all". See "details" for more info on valid input formats when not using "all" as input.
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 mot be changed unless Comtrade changes their endpoint url.

Details

Basic rate limit restrictions listed below. For details on how to register a valid token, see ct_register_token. For API docs on rate limits, see https://comtrade.un.org/data/doc/api/#Limits

- Without authentication token: 1 request per second, 100 requests per hour (each per IP address).
- With valid authentication token: 1 request per second, 10,000 requests per hour (each per IP address or authenticated user).

In addition to these rate limits, the API imposes some limits on parameter combinations, they are listed below:

- Between params "reporters", "partners", and the query date range (as dictated by the two params "start_date" and "end_date"), only one of these three may use the catch-all input "All".
- For the same group of three ("reporters", "partners", date range), if the input is not "All", then the maximum number of input values for each is five. For date range, if not using "All", then the "start_date" and "end_date" must not span more than five months or five years. There is one exception to this rule, if arg "freq" is "monthly", then a single year can be passed to "start_date" and "end_date" and the API will return all of the monthly data for that year.
- For param "commod_codes", if not using input "All", then the maximum number of input values is 20 (although "All" is always a valid input).

This function returns objects with metadata related to the API call that can be accessed via attributes. The metadata accessible is:

- url: url of the API call.
- time_stamp: date-time of the API call.
- req_duration: total duration of the API call, in seconds.

For args start_date and end_date, if inputting a date (as opposed to the catch-all input "all"), valid input format is dependent on the input passed to arg freq. If freq is "annual", start_date and end_date must be either a string w/ format "yyyy" or "yyyy-mm-dd", or a year as an integer (so "2016", "2016-01-01", and 2016 would all be valid). If freq is "monhtly", start_date and end_date must be a string with format "yyyy-mm" or "yyyy-mm-dd" (so "2016-02" and "2016-02-01" would both be valid).

Value

Data frame of Comtrade shipping data.

Examples

```
## Not run:
## Example API call number 1:
# All exports from China to South Korea, United States and Mexico over all
# years.
ex_1 <- ct_search(reporters = "China",
                  partners = c("Rep. of Korea", "USA", "Mexico"),
                  trade_direction = "exports")
nrow(ex_1)
```

```
## Example API call number 2:
# All shipments related to shrimp between Canada and all other countries,
# between 2011 and 2015.
# Perform "shrimp" query
shrimp_codes <- ct_commodity_lookup("shrimp",
                                    return_code = TRUE,
                                    return_char = TRUE)

# Make API call
ex_2 <- ct_search(reporters = "Canada",
                  partners = "All",
                  trade_direction = "all",
                  start_date = 2011,
                  end_date = 2015,
                  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", "HS2017", "SITC", "SITCrev1", "SITCrev2", "SITCrev3",
  "SITCrev4", "BEC", "EB02"), commodity_url = NULL,
  reporter_url = NULL, partner_url = NULL)
```

Arguments

`force`	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.
`verbose`	logical, if TRUE, an update status message will be printed to console. Default value is TRUE.

commodity_type 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 comtradr. 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 ct_commodity_db_type.

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
- HS2017: HS 2017
- 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