Package: imfr (via r-universe)

November 22, 2024

Type Package

Title Download Data from the International Monetary Fund's Data API

Version 0.2.0.1

Description Explore and download data from the International Monetary Fund's data API <http://data.imf.org/>.

Date 2023-03-27

URL https://github.com/christophergandrud/imfr

BugReports https://github.com/christophergandrud/imfr/issues

License GPL (>= 3)

Imports dplyr, httr (>= 1.2.0), jsonlite, methods, purrr, ratelimitr, tidyr

LazyData TRUE

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat, tidyverse, stringr

Encoding UTF-8

Depends R (>= 3.5.0)

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libssl-dev

Repository https://cjyetman.r-universe.dev

RemoteUrl https://github.com/christophergandrud/imfr

RemoteRef HEAD

RemoteSha ab9a326af366044ccf1f2f1b700849ad168e9b99

Contents

`imf_app_name` *Set the IMF Application Name*

Description

Set a unique application name to be used in requests to the IMF API as a hidden environment variable

Usage

```
imf_app_name(name = "imfr")
```

Arguments

`name`	A string representing the application name. Default is "imfr".

Details

The ‘imf_app_name‘ function sets the application name that will be used in the request header when making API calls to the IMF API. The IMF API has an application-based rate limit of 50 requests per second, with the application identified by the "user_agent" variable in the request header. The function sets the application name by changing the ‘IMF_APP_NAME‘ hidden variable in ‘.Renviron‘. If this variable doesn’t exist, ‘imf_app_name‘ will create it.

Value

Invisible NULL

Examples

```
imf_app_name("my_custom_app_name")
```

`imf_databases` *List IMF database IDs and descriptions*

Description

List IMF database IDs and descriptions

Usage

```
imf_databases(times = 3)
```

Arguments

times	numeric. Maximum number of API requests to attempt.

Value

Returns a data frame with database_id and text description for each database available through the IMF API endpoint.

Examples

```
# Return first 6 IMF database IDs and descriptions
databases <- imf_databases()
```

---

imf_dataset *Download a data series from the IMF*

Description

Function to request data from a database through the IMF API endpoint.

Usage

```
imf_dataset(
  database_id,
  parameters,
  start_year,
  end_year,
  return_raw = FALSE,
  print_url = FALSE,
  times = 3,
  include_metadata = FALSE,
  accounting_entry,
  activity,
  adjustment,
  age,
  classification,
  cofog_function,
  commodity,
  comp_method,
  composite_breakdown,
  counterpart_area,
  counterpart_sector,
  currency_denom,
  cust_breakdown,
  disability_status,
  education_lev,
```

```
  expenditure,
  financial_institution,
  flow_stock_entry,
  freq,
  functional_cat,
  gfs_sto,
  income_wealth_quantile,
  indicator,
  instr_asset,
  instrument_and_assets_classification,
  int_acc_item,
  maturity,
  occupation,
  prices,
  product,
  ref_area,
  ref_sector,
  reporting_type,
  series,
  sex,
  sto,
  summary_statistics,
  survey,
  transformation,
  type,
  unit_measure,
  urbanisation,
  valuation
)
```

Arguments

database_id	character string. Database ID for database from which you would like to request data. Can be found using imf_databases.
parameters	list of data frames providing input parameters for your API request. Retrieve list of all possible input parameters using imf_parameters and filter each data frame in the list to reduce it to the inputs you want.
start_year	integer four-digit year. Earliest year for which you would like to request data.
end_year	integer four-digit year. Latest year for which you would like to request data.
return_raw	logical. Whether to return the raw list returned by the API instead of a cleaned-up data frame. This argument exists strictly for purposes of backward compatibility with earlier versions of **imfr** and will be discontinued in a future version.
print_url	logical. Whether to print the URL used in the API call.
times	numeric. Maximum number of requests to attempt.
include_metadata	logical. Whether to return the database metadata header along with the data series.

accounting_entry, activity, adjustment, age, classification, cofog_function, commodity, comp_method, composite_breakdown, counterpart_area, counterpart_sector, currency_denom, cust_breakdown, disability_status, education_lev, expenditure, financial_institution, flow_stock_entry, freq, functional_cat, gfs_sto, income_wealth_quantile, indicator, instr_asset, instrument_and_assets_classification, int_acc_item, maturity, occupation, prices, product, ref_area, ref_sector, reporting_type, series, sex, sto, summary_statistics, survey, transformation, type, unit_measure, urbanisation, valuation

character vector. Use imf_parameters to identify which parameters to use for requests from a given database and to see all valid input codes for each parameter.

Details

Only the database_id argument is strictly required; all other arguments are optional. If you provide a database_id without any other arguments, the function will attempt to download the entire database. However, many databases available through the API are too large to download in their entirety, and your request will fail. Additional arguments to the function act as filter parameters to reduce the size of the returned dataset. For instance, supplying c("A","M") as the freq argument will return all database observations of annual or monthly frequency, while excluding all observations of quarterly frequency.

There are two ways to supply parameters for your API request. The optimal way is to retrieve a list of data frames using imf_parameters, filter each data frame to retain only the parameters you want, and then supply the modified list object to imf_dataset as its parameters argument. However, users who are not comfortable modifying data frames in a nested list may find it easier to instead supply one or more character vectors as arguments, as in the example in the previous paragraph. (There are a total of 44 possible parameters for making request from various databases through the API, and each parameter uses unique input codes, which is why the parameters list method simplifies things!) These two methods for specifying parameters may not be combined. Only database_id, start_year, end_year, print_url, and times arguments may be used in combination with a parameters list object; any other arguments will be ignored (and a warning thrown).

Value

If return_raw == FALSE and include_metadata == FALSE, returns a tidy data frame with the data series. If return_raw == FALSE but include_metadata == TRUE, returns a list whose first item is the database header, and whose second item is the tidy data frame. If return_raw == TRUE, returns the raw JSON fetched from the API endpoint.

Examples

```
# Retrieve "Current Account, Goods and Services, Services, Travel, Personal,
# Other, Credit, US Dollars" for "United States" from the Balance of Payments
# database using the character vector method
df <- imf_dataset(database_id = 'BOP',freq='A',ref_area = 'US',
                  indicator = 'BXSTVPO_BP6_USD',start_year=2020)
```

`imf_parameters` *List parameters and parameter values for IMF API requests*

Description

List input parameters and available parameter values for use in making API requests from a given IMF database.

Usage

```
imf_parameters(database_id, times = 3)
```

Arguments

`database_id`	character string of a `database_id` from `imf_databases`.
`times`	numeric. Maximum number of API requests to attempt.

Details

Retrieves a list of data frames containing all possible input parameters for requests from a given database available through the IMF API. Each data frame in the returned list has an `input_code` column and a `description` column. Retrieve the list, filter each data frame for the parameters you want, and then supply the modified list object to the `imf_dataset` function as its `parameters` argument. Alternatively, individually supply `input_code` values from each data frame as arguments to `imf_dataset`.

Value

Returns a named list of data frames. Each list item name corresponds to an input parameter for API requests from the database. All list items are data frames, with an `input_code` column and a `description` column. The `input_code` column is a character vector of all possible input codes for that parameter when making requests from the IMF API endpoint. The `descriptions` column is a character vector of text descriptions of what each input code represents.

Examples

```
# Fetch the full list of indicator codes and descriptions for the Primary
# Commodity Price System database
params <- imf_parameters(database_id = 'PCPS')
```

`imf_parameter_defs` *Get definitions of IMF API parameters*

Description

Get text descriptions of input parameters used in making API requests from a given IMF database

Usage

```
imf_parameter_defs(database_id, times = 3, inputs_only = T)
```

Arguments

`database_id`	character string of a `database_id` from `imf_databases`.
`times`	numeric. Maximum number of API requests to attempt.
`inputs_only`	logical. Whether to return only parameters used as inputs in API requests, or also output variables.

Value

Returns a data frame of input parameters used in making API requests from a given IMF database, along with text descriptions or definitions of those parameters. Useful in cases when parameter names returned by `imf_databases` are not self-explanatory. (Note that the usefulness of text descriptions can be uneven, depending on the database design.)

Examples

```
# Get names and text descriptions of parameters used in IMF API calls to the
# Primary Commodity Price System database
param_defs <- imf_parameter_defs(database_id = 'PCPS')
```

Index