

Package: blscrapeR (via r-universe)

September 14, 2024

Type Package

Title An API Wrapper for the United States Bureau of Labor Statistics

Version 4.0.1

Description Scrapes various data from <https://www.bls.gov/>. The Bureau of Labor Statistics is the statistical branch of the United States Department of Labor. The package has additional functions to help parse, analyze and visualize the data.

Depends R (>= 3.5.0)

Imports httr, jsonlite, magrittr, utils, stats, dplyr, purrr, tibble, stringr

Suggests testthat, knitr, rmarkdown

License MIT + file LICENSE

URL <https://github.com/keberwein/blscrapeR>

BugReports <https://github.com/keberwein/blscrapeR/issues>

LazyData true

VignetteBuilder knitr

RoxygenNote 7.2.3

Repository <https://keberwein.r-universe.dev>

RemoteUrl <https://github.com/keberwein/blscrapeR>

RemoteRef HEAD

RemoteSha cc2d0c403d911d5eb7ae74c45f29dc08a55ae960

Contents

bls_api	2
dateCast	3
inflation_adjust	4
quick_employed_level	5
quick_employed_rate	5
quick_laborForce_level	6

quick_laborForce_rate	6
quick_nonfarm_employed	7
quick_unemp_level	7
quick_unemp_rate	8
search_ids	8
urlExists	9

Index	10
--------------	-----------

bls_api	<i>Basic Request Mechanism for BLS Tables</i>
---------	---

Description

Return data frame from one or more requests via the US Bureau of Labor Statistics API. Provided arguments are in the form of BLS series ids.

Usage

```
bls_api(
  seriesid,
  startyear = NULL,
  endyear = NULL,
  registrationKey = NULL,
  catalog = FALSE,
  calculations = FALSE,
  annualaverage = FALSE,
  ...
)
```

Arguments

seriesid	The BLS id of the series your trying to load. A common format would be 'LAUCN040010000000005'. WARNING: All seriesIDs must contain the same time resolution. For example, monthly data sets can not be combined with annual or semi-annual data. If you need help finding seriesIDs, check the BLS website or the BLS Data Finder—links below.
startyear	The first year in your data set.
endyear	The last year in your data set.
registrationKey	The API key issued to you from the BLS website.
catalog	Series description information available only for certain data sets.
calculations	Returns year-over-year calculations if set to TRUE.
annualaverage	Returns an annual average if set to TRUE.
...	additional arguments

Value

A tibble from the BLS API.

Examples

```
## API Version 1.0 R Script Sample Code
## Single Series request
df <- bls_api("LAUCN04001000000005")

## Not run:
## API Version 1.0 R Script Sample Code
## Multiple Series request with date params.
df <- bls_api(c("LAUCN04001000000005", "LAUCN04001000000006"),
  startyear = "2010", endyear = "2012")

## API Version 1.0 R Script Sample Code
## Multiple Series request with date params.
df <- bls_api(c("LAUCN04001000000005", "LAUCN04001000000006"),
  startyear = "2010", endyear = "2012")

## API Version 2.0 R Script Sample Code
## Multiple Series request with full params allowed by v2.
df <- bls_api(c("LAUCN04001000000005", "LAUCN04001000000006"),
  startyear = "2010", endyear = "2012",
  registrationKey = "BLS_KEY",
  calculations = TRUE, annualaverage = TRUE, catalog = TRUE)

## End(Not run)
```

dateCast

Cast a date column to data frame returned by the bls_api() function

Description

A helper function to create a continuous date from Year and Period columns.

Usage

```
dateCast(api_df = NULL, dt_format = NULL)
```

Arguments

api_df	The data frame you wish to cast a date column to. Be sure the data frame contains 'year' and 'period' columns as returned by the bls_api() function.
dt_format	A character string containing a valid date format. The default will return the ISO 8601 date format.

Value

A tibble from the source `api_df` with an additional date column based on the date format given in `dt_format`.

Examples

```
## Cast a date column to data frame returned by the bls_api() function.
df <- bls_api("LAUCN04001000000005") %>%
  dateCast()
```

<code>inflation_adjust</code>	<i>Convert the Value of a US Dollar to a Given month on or after 1947.</i>
-------------------------------	--

Description

Returns a data frame that uses data from the Consumer Price Index (All Goods) to convert the value of a US Dollar [\$1.00 USD] over time.

Usage

```
inflation_adjust(base_date = NA, ...)
```

Arguments

<code>base_date</code>	= A string argument to represent the base month that you would like dollar values converted to. For example, if you want to see the value of a Jan. 2015 dollar in Jan. 2023, you would select "2015-01-01" as a base date and find Jan 2023 in the table.
<code>...</code>	additional arguments

Value

A tibble from the BLS API.

Examples

```
## Not run:
## Get historical USD values based on a dollar from Jan 2015.
values <- inflation_adjust(base_year = "2015-01-01")

## End(Not run)
```

quick_employed_level *Quick employed level*

Description

Returns the employment level. SeriesID: LNS12000000 If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

```
quick_employed_level()
```

Value

A tibble from the BLS API.

Examples

```
## Not run:  
df <- quick_employed_level()  
  
## End(Not run)
```

quick_employed_rate *Quick employed rate*

Description

Returns the "employment to population ratio." SeriesID: LNS12300000 If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

```
quick_employed_rate()
```

Value

A tibble from the BLS API.

Examples

```
## Not run:  
df <- quick_employed_rate()  
  
## End(Not run)
```

quick_laborForce_level

Quick Civilian Labor Force Level

Description

Returns the civilian labor force level. SeriesID: LNS11000000. If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

```
quick_laborForce_level()
```

Value

A tibble from the BLS API.

Examples

```
## Not run:  
df <- quick_laborForce_level()  
  
## End(Not run)
```

quick_laborForce_rate *Quick Civilian Labor Force Rate*

Description

Returns the civilian labor force participation rate. SeriesID: LNS11300000. If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

```
quick_laborForce_rate()
```

Value

A tibble from the BLS API.

Examples

```
## Not run:  
df <- quick_laborForce_rate()  
  
## End(Not run)
```

quick_nonfarm_employed

Quick total nonfarm employment

Description

Returns the Total Nonfarm Payroll Employment, seasonally adjusted. BLS id CES0000000001. If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

```
quick_nonfarm_employed()
```

Value

A tibble from the BLS API.

Examples

```
## Not run:  
df <- quick_nonfarm_employed()  
  
## End(Not run)
```

quick_unemp_level

Quick unemployment level function

Description

Returns the unemployment level. SeriesID: LNS13000000. If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

```
quick_unemp_level()
```

Value

A tibble from the BLS API.

Examples

```
## Not run:  
df <- quick_unemp_level()  
  
## End(Not run)
```

quick_unemp_rate	<i>Quick unemployment rate function</i>
------------------	---

Description

Returns the "official" unemployment rate. That is, seasonally adjusted, 16 year and over, or the "U-3" rate. SeriesID: LNS14000000. If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

```
quick_unemp_rate()
```

Value

A tibble from the BLS API.

Examples

```
## Not run:  
df <- quick_unemp_rate()  
  
## End(Not run)
```

search_ids	<i>Search the internal series_id data set.</i>
------------	--

Description

Search the internal series_id data set.

Usage

```
search_ids(keyword = NULL, periodicity_code = NULL, ...)
```


Arguments

keyword The keyword you want to search. This can be a fuzzy search of multiple keywords. For example "unemployment women".

periodicity_code The period of time of the surveys you are interested in. This is usually monthly, quarterly or annually. You can type full words or beginning letters. For example, periodicity_code = "m" or periodicity_code = "monthly".

... additional arguments

Value

A tibble from the the internal data set bls_ids

Examples

```
## Not run:
# Search for monthly Unemployment Rates for Women
ids <- search_ids(keyword = c("Unemployment Rate", "Women"), periodicity_code = "M")

## End(Not run)
```

urlExists

urlExists

Description

A utility function to run a tryCatch on a URL.

Usage

```
urlExists(target)
```

Arguments

target url

Value

A logical of TRUE or FALSE.

Index

- * **api**
 - bls_api, [2](#)
 - inflation_adjust, [4](#)
- * **bls**
 - bls_api, [2](#)
 - inflation_adjust, [4](#)
- * **cpi**
 - bls_api, [2](#)
 - inflation_adjust, [4](#)
- * **economics**
 - bls_api, [2](#)
 - inflation_adjust, [4](#)
- * **inflation**
 - bls_api, [2](#)
 - inflation_adjust, [4](#)
- * **nonfarm_employed**
 - quick_nonfarm_employed, [7](#)
- * **quick**
 - quick_employed_level, [5](#)
 - quick_employed_rate, [5](#)
 - quick_laborForce_level, [6](#)
 - quick_laborForce_rate, [6](#)
 - quick_nonfarm_employed, [7](#)
 - quick_unemp_level, [7](#)
 - quick_unemp_rate, [8](#)
- * **rate**
 - quick_employed_level, [5](#)
 - quick_employed_rate, [5](#)
 - quick_laborForce_level, [6](#)
 - quick_laborForce_rate, [6](#)
 - quick_unemp_level, [7](#)
 - quick_unemp_rate, [8](#)
- * **unemployment**
 - bls_api, [2](#)
 - inflation_adjust, [4](#)
 - quick_employed_level, [5](#)
 - quick_employed_rate, [5](#)
 - quick_laborForce_level, [6](#)
 - quick_laborForce_rate, [6](#)
- quick_unemp_level, [7](#)
- quick_unemp_rate, [8](#)
- bls_api, [2](#)
- dateCast, [3](#)
- inflation_adjust, [4](#)
- quick_employed_level, [5](#)
- quick_employed_rate, [5](#)
- quick_laborForce_level, [6](#)
- quick_laborForce_rate, [6](#)
- quick_nonfarm_employed, [7](#)
- quick_unemp_level, [7](#)
- quick_unemp_rate, [8](#)
- search_ids, [8](#)
- urlExists, [9](#)