

Package ‘pipr’

September 30, 2026

Title Client for the Poverty and Inequality Platform ('PIP') API

Version 1.5.0

Description An interface to compute poverty and inequality indicators for more than 160 countries and regions from the World Bank's database of household surveys, through the Poverty and Inequality Portal (PIP).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL <https://worldbank.github.io/pipr/>,
<https://github.com/worldbank/pipr>

BugReports <https://github.com/worldbank/pipr/issues>

Suggests covr, testthat (>= 3.0.0), spelling, knitr, rmarkdown, markdown, callr, ggplot2, tidyr, ggthemes, forcats, scales, dplyr, readr

Language en-US

Imports arrow, attempt, curl, jsonlite, tibble, purrr, cli, rlang, utils, httr2, stringr, vroom

Depends R (>= 4.1.0)

Config/testthat/edition 3

Date 2026-09-29

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Tony Fujs [aut],
Aleksander Eilertsen [aut],
Ronak Shah [aut],
R.Andrés Castañeda [aut, cre],
Giorgia Cecchinato [aut],
World Bank [cph]

Maintainer R.Andrés Castañeda <acastaneda@worldbank.org>

Repository CRAN
Date/Publication 2026-09-30 15:00:17 UTC

Contents

args_to_string	2
build_request	3
build_request_old	3
call_aux	4
change_grouped_stats_to_csv	4
check_api	5
datt_rural	5
datt_urban	6
delete_cache	7
display_aux	7
get_aux	8
get_cache_info	14
get_cp	14
get_cp_ki	15
get_gd	16
get_pip_info	18
get_stats	19
get_versions	21
parse_error_body	22
pip_is_transient	23
retry_after	23
unnest_ki	24
Index	25

args_to_string	<i>convert arguments and values of a function to a string to parse into other functions</i>
----------------	---

Description

convert arguments and values of a function to a string to parse into other functions

Usage

args_to_string(il)

Arguments

il list from `as.list(environment())` right after the function is called.

Value

character

build_request	<i>Build request version 2</i>
---------------	--------------------------------

Description

Build request version 2

Usage

build_request(server, api_version, endpoint, ...)

Arguments

server	character: Server. For WB internal use only
api_version	character: API version
endpoint	character: PIP API endpoint
...	other parameters

Value

httr2 request

build_request_old	<i>build_request, OLD version</i>
-------------------	-----------------------------------

Description

build_request, OLD version

Usage

build_request_old(server, api_version, endpoint, ...)

Arguments

server	character: Server. For WB internal use only
api_version	character: API version
endpoint	character: PIP API endpoint
...	other parameters

Value

httr2 request

call_aux	<i>call a table from .pip env</i>
----------	-----------------------------------

Description

call a table from .pip env

Usage

```
call_aux(table = NULL)
```

Arguments

table	character: name of table in .pip env. If NULL, it displays the names of tables available in .pip env
-------	--

Value

The stored auxiliary table when table is given. With table = NULL, invisibly returns the names of stored tables.

Examples

```
# List tables already stored in this R session (no network request).
call_aux()
if (interactive()) {
  get_aux("gdp", assign_tb = TRUE, replace = TRUE)
  call_aux("gdp")
}
```

change_grouped_stats_to_csv	<i>Convert a grouped-statistics result to a data frame</i>
-----------------------------	--

Description

Convert a grouped-statistics result to a data frame

Usage

```
change_grouped_stats_to_csv(out)
```

Arguments

out	A list of grouped statistics with a deciles vector.
-----	---

Value

A data frame with one column per decile and no deciles list column.

Examples

```
change_grouped_stats_to_csv(list(country = "AGO", deciles = c(0.1, 0.2)))
```

check_api	<i>Check internet connection and API status</i>
-----------	---

Description

Check internet connection and API status

Usage

```
check_api(api_version = "v1", server = NULL)
```

Arguments

api_version	API version. Currently only "v1" is supported.
server	One non-empty server name, or NULL for the production server. For World Bank internal use only.

Value

character

Examples

```
if (interactive()) {
  check_api()
}
```

datt_rural	<i>Datt (1998) grouped data for rural india, 1983</i>
------------	---

Description

Dataset from Datt (1998) with grouped data for rural India in 1983.

Usage

```
datt_rural
```

Format

A data frame with 13 observations on the following 6 variables:

monthly_pc_exp Welfare range class

mean_monthly_pc_exp Mean welfare for given welfare range class

percentage_of_persons Percentage of individuals in given welfare class

L Cumulative welfare

p Cumulative population

area rural

Source

Datt, G. (1998). See get_cp vignette.

datt_urban

Grouped data for urban india, 1983

Description

Dataset from Sarvekshana N26 Vol 9 N 4, created by the author following Datt(1998) methodology with grouped data for urban India in 1983.

Usage

datt_urban

Format

A data frame with 13 observations on the following 6 variables:

monthly_pc_exp Welfare range class

mean_monthly_pc_exp Mean welfare for given welfare range class

percentage_of_persons Percentage of individuals in given welfare class

L Cumulative welfare

p Cumulative population

area urban

Source

Sarvekshana N26 Vol 9 N 4, and Datt, G. (1998) for methodology. See get_cp vignette.

delete_cache	<i>Deletes content of the cache folder</i>
--------------	--

Description

Deletes content of the cache folder

Usage

```
delete_cache()
```

Value

Side effect. Deletes files.

Examples

```
if (interactive()) delete_cache()
```

display_aux	<i>Display available auxiliary tables</i>
-------------	---

Description

Display available auxiliary tables

Usage

```
display_aux(  
  version = NULL,  
  ppp_version = NULL,  
  release_version = NULL,  
  api_version = "v1",  
  format = c("rds", "json", "csv"),  
  simplify = TRUE,  
  server = NULL,  
  assign_tb = TRUE  
)
```

Arguments

version	Data version in YYYYMMDD_PPP_XX_YY_IDENTITY format, or NULL. See get_versions().
ppp_version	One integer-valued numeric PPP year, a four-digit year string, or NULL.
release_version	A valid publication date in YYYYMMDD format, or NULL.
api_version	API version. Currently only "v1" is supported.
format	Response format: "rds", "json", or "csv".
simplify	One non-missing logical value. If TRUE (the default), return simplified data.
server	One non-empty server name, or NULL for the production server. For World Bank internal use only.
assign_tb	logical: Whether to assign table to .pip env. Default is TRUE

Value

invisible tibble with names of auxiliary tables

Examples

```
if (interactive()) {
  display_aux()
}
```

get_aux	<i>Get auxiliary data</i>
---------	---------------------------

Description

get_aux() gets an auxiliary dataset. If no table is specified, it lists the available tables.
Use get_aux("countries") for a table of country names, ISO codes, and associated region codes.

Usage

```
get_aux(
  table = NULL,
  version = NULL,
  ppp_version = NULL,
  release_version = NULL,
  api_version = "v1",
  format = c("rds", "json", "csv"),
  simplify = TRUE,
  server = NULL,
  assign_tb = FALSE,
  replace = FALSE
)
```

```
get_countries(  
  version = NULL,  
  ppp_version = NULL,  
  release_version = NULL,  
  api_version = "v1",  
  format = c("rds", "json", "csv"),  
  server = NULL  
)  
  
get_regions(  
  version = NULL,  
  ppp_version = NULL,  
  release_version = NULL,  
  api_version = "v1",  
  format = c("rds", "json", "csv"),  
  server = NULL  
)  
  
get_cpi(  
  version = NULL,  
  ppp_version = NULL,  
  release_version = NULL,  
  api_version = "v1",  
  format = c("rds", "json", "csv"),  
  server = NULL  
)  
  
get_dictionary(  
  version = NULL,  
  ppp_version = NULL,  
  release_version = NULL,  
  api_version = "v1",  
  format = c("rds", "json", "csv"),  
  server = NULL  
)  
  
get_gdp(  
  version = NULL,  
  ppp_version = NULL,  
  release_version = NULL,  
  api_version = "v1",  
  format = c("rds", "json", "csv"),  
  server = NULL  
)  
  
get_incgrp_coverage(  
  version = NULL,  
  ppp_version = NULL,
```

```
    release_version = NULL,  
    api_version = "v1",  
    format = c("rds", "json", "csv"),  
    server = NULL  
)  
  
get_interpolated_means(  
    version = NULL,  
    ppp_version = NULL,  
    release_version = NULL,  
    api_version = "v1",  
    format = c("rds", "json", "csv"),  
    server = NULL  
)  
  
get_hfce(  
    version = NULL,  
    ppp_version = NULL,  
    release_version = NULL,  
    api_version = "v1",  
    format = c("rds", "json", "csv"),  
    server = NULL  
)  
  
get_pop(  
    version = NULL,  
    ppp_version = NULL,  
    release_version = NULL,  
    api_version = "v1",  
    format = c("rds", "json", "csv"),  
    server = NULL  
)  
  
get_pop_region(  
    version = NULL,  
    ppp_version = NULL,  
    release_version = NULL,  
    api_version = "v1",  
    format = c("rds", "json", "csv"),  
    server = NULL  
)  
  
get_ppp(  
    version = NULL,  
    ppp_version = NULL,  
    release_version = NULL,  
    api_version = "v1",  
    format = c("rds", "json", "csv"),
```

```

    server = NULL
  )

  get_region_coverage(
    version = NULL,
    ppp_version = NULL,
    release_version = NULL,
    api_version = "v1",
    format = c("rds", "json", "csv"),
    server = NULL
  )

  get_survey_means(
    version = NULL,
    ppp_version = NULL,
    release_version = NULL,
    api_version = "v1",
    format = c("rds", "json", "csv"),
    server = NULL
  )

```

Arguments

table	Aux table
version	Data version in YYYYMMDD_PPP_XX_YY_IDENTITY format, or NULL. See <code>get_versions()</code> .
ppp_version	Validated as a PPP year, but not sent to the auxiliary endpoint because this endpoint does not accept it.
release_version	A valid publication date in YYYYMMDD format, or NULL.
api_version	API version. Currently only "v1" is supported.
format	Response format: "rds", "json", or "csv".
simplify	One non-missing logical value. If TRUE (the default), return simplified data.
server	One non-empty server name, or NULL for the production server. For World Bank internal use only.
assign_tb	assigns table to specified name to the .pip environment. If FALSE no assignment will be performed. If TRUE, the table will be assigned to exactly the same name as the one of the desired table. If character, the table will be assigned to that name.
replace	logical: force replacement of aux files in .pip env. Default is FALSE.

Value

With no table, invisibly returns the available tables (a tibble with a `tables` column by default). With a selected table, returns its tibble when `simplify = TRUE`, or a `pip_api` list when `simplify = FALSE`. With `assign_tb = TRUE` or a name, invisibly returns TRUE on successful assignment to .pip; unsuccessful assignment raises an error.

Functions

- `get_countries()`: Returns a table countries with their full names, ISO codes, and associated region code
- `get_regions()`: Returns a table regional grouping used for computing aggregate poverty statistics.
- `get_cpi()`: Returns a table of Consumer Price Index (CPI) values used for poverty and inequality computations. statistics
- `get_dictionary()`: Returns a data dictionary with a description of all variables available through the PIP API.
- `get_gdp()`: Returns a table of Growth Domestic Product (GDP) values used for poverty and inequality statistics.
- `get_incgrp_coverage()`: Returns a table of survey coverage for low and lower-middle income countries. If this coverage is less than 50%, World level aggregate statistics will not be computed.
- `get_interpolated_means()`: Returns a table of key information and statistics for all years for which poverty and inequality statistics are either available (household survey exists) or extra- / interpolated. Please see `get_aux("dictionary")` for more information about each variable in this table.
- `get_hfce()`: Returns a table of Household Final Consumption Expenditure (HFCE) values used for poverty and inequality computations.
- `get_pop()`: Returns a table of population values used for poverty and inequality computations.
- `get_pop_region()`: Returns a table of total population by region-year. These values are used for the computation of regional aggregate poverty statistics.
- `get_ppp()`: Returns a table of Purchasing Power Parity (PPP) values used for poverty and inequality computations.
- `get_region_coverage()`: Return a table of regional survey coverage: Percentage of available surveys for a specific region-year.
- `get_survey_means()`: Returns a table of all available surveys and associated key statistics. See `get_aux("dictionary")` for more information about each variable in this table.

Examples

```
if (interactive()) {
# Get list of tables
x <- get_aux()

# Get GDP data
df <- get_aux("gdp")

# Get countries
df <- get_aux("countries")

# Display auxiliary tables
get_aux()
```

```
# Bind gdp table to "gdp" in .pip env
get_aux("gdp", assign_tb = TRUE)

# Bind gdp table to "new_name" in .pip env
get_aux("gdp", assign_tb = "new_name")

}
if (interactive()) {
# Short hand to get countries
get_aux("countries")
}
if (interactive()) {
# Short hand to get regions
get_aux("regions")
}
if (interactive()) {
# Short hand to get cpi
get_aux("cpi")
}
if (interactive()) {
# Short hand to get dictionary
get_aux("dictionary")
}
if (interactive()) {
# Short hand to get gdp
get_aux("gdp")
}
if (interactive()) {
# Short hand to get incgrp_coverage
get_aux("incgrp_coverage")
}
if (interactive()) {
# Short hand to get interpolated_means
get_aux("interpolated_means")
}
if (interactive()) {
# Short hand to get hfce
get_aux("pce")
}
if (interactive()) {
# Short hand to get pop
get_aux("pop")
}
if (interactive()) {
# Short hand to get pop_region
get_aux("pop_region")
}
if (interactive()) {
# Short hand to get ppp
get_aux("ppp")
}
if (interactive()) {
# Short hand to get region_coverage
```

```

get_aux("region_coverage")
}
if (interactive()) {
# Short hand to get survey_means
get_aux("survey_means")
}

```

get_cache_info	<i>Provides some information about cached items</i>
----------------	---

Description

Provides some information about cached items

Usage

```
get_cache_info()
```

Value

The result of the cache status message, invisibly (NULL).

Examples

```
get_cache_info()
```

get_cp	<i>Get Country Profiles</i>
--------	-----------------------------

Description

Get Country Profiles

Usage

```

get_cp(
  country = "all",
  povline = 2.15,
  version = NULL,
  ppp_version = 2017,
  release_version = NULL,
  api_version = "v1",
  format = c("arrow", "rds", "json", "csv"),
  simplify = TRUE,
  server = NULL
)

```

Arguments

country	Uppercase, three-letter country ISO codes or "all".
povline	One finite, non-negative numeric poverty line, or NULL. With ppp_version = 2011, NULL sends 1.9; otherwise the API applies its default poverty line.
version	Data version in YYYYMMDD_PPP_XX_YY_IDENTITY format, or NULL. See get_versions().
ppp_version	One integer-valued numeric PPP year, a four-digit year string, or NULL.
release_version	A valid publication date in YYYYMMDD format, or NULL.
api_version	API version. Currently only "v1" is supported.
format	Response format: "arrow", "rds", "json", or "csv".
simplify	One non-missing logical value. If TRUE (the default), return simplified data.
server	One non-empty server name, or NULL for the production server. For World Bank internal use only.

Value

A tibble of country profile data when simplify = TRUE, or a pip_api list when simplify = FALSE.

Examples

```
if (interactive()) {
  # One country, all years with default ppp_version = 2017
  res <- get_cp(country = "AGO")

  # All countries, povline = 1.9
  res <- get_cp(povline = 1.9)

  # All countries and years with default values
  res <- get_cp()
}
```

get_cp_ki

Get Country Profiles Key Indicators

Description

Get Country Profiles Key Indicators

Usage

```
get_cp_ki(
  country = NULL,
  povline = 2.15,
  version = NULL,
  ppp_version = 2017,
```

```
    release_version = NULL,  
    api_version = "v1",  
    simplify = TRUE,  
    server = NULL  
  )
```

Arguments

country	One uppercase, three-letter country ISO code (required).
povline	One finite, non-negative numeric poverty line, or NULL. With ppp_version = 2011, NULL sends 1.9; otherwise the API applies its default poverty line.
version	Data version in YYYYMMDD_PPP_XX_YY_IDENTITY format, or NULL. See get_versions().
ppp_version	One integer-valued numeric PPP year, a four-digit year string, or NULL.
release_version	A valid publication date in YYYYMMDD format, or NULL.
api_version	API version. Currently only "v1" is supported.
simplify	One non-missing logical value. If TRUE (the default), return simplified data.
server	One non-empty server name, or NULL for the production server. For World Bank internal use only.

Value

An unnested data frame of key indicators when simplify = TRUE, or a pip_api list when simplify = FALSE.

Examples

```
if (interactive()) {  
  # One country, all years with default ppp_version = 2017  
  res <- get_cp_ki(country = "IDN")  
  
  # One country, povline = 1.9  
  res <- get_cp_ki(country = "IDN", povline = 1.9)  
}
```

get_gd	<i>Get grouped stats</i>
--------	--------------------------

Description

Get grouped stats from the PIP API.

Usage

```
get_gd(
  cum_welfare,
  cum_population,
  estimate = c("stats", "lorenz", "params"),
  requested_mean = NULL,
  povline = NULL,
  popshare = NULL,
  lorenz = NULL,
  n_bins = NULL,
  api_version = "v1",
  format = c("rds", "json", "csv"),
  simplify = TRUE,
  server = NULL
)
```

Arguments

cum_welfare	numeric: Cumulative welfare values, expressed in shares. Any length. They should be monotonically increasing, and sum to 1.
cum_population	numeric: Cumulative population values, expressed in shares. Any length. They should be monotonically increasing, and sum to 1.
estimate	character: One of "stats", "lorenz", "params".
requested_mean	numeric: Requested mean.
povline	numeric: Poverty line. Required for estimate = "stats".
popshare	One numeric population share from 0 to 1, or NULL.
lorenz	character: Lorenz curve methodology. Either "lb" or "lq".
n_bins	numeric: Number of bins. Required for estimate = "lorenz".
api_version	API version. Currently only "v1" is supported.
format	Response format: "rds", "json", or "csv".
simplify	One non-missing logical value. If TRUE (the default), return simplified data.
server	One non-empty server name, or NULL for the production server. For World Bank internal use only.

Value

A data frame when simplify = TRUE, or a pip_api list when simplify = FALSE.

Examples

```
if (interactive()) {

  datt_data <- data.frame(p = c(0.0092, 0.0339, 0.0850, 0.160, 0.2609, 0.4133,
                                0.5497, 0.7196, 0.8196, 0.9174, 0.9570, 0.9751,
                                1),
                        L = c(0.00208, 0.01013, 0.03122, 0.07083, 0.12808,
```

```

0.23498, 0.34887, 0.51994, 0.64270, 0.79201,
0.86966, 0.91277, 1))

# estimate = 'stats': retrieve poverty statistics.
stats <- get_gd(cum_welfare = datt_data$L, cum_population = datt_data$p,
               estimate = "stats",
               requested_mean = 19,
               povline = 2.15)

# estimate = 'lorenz': retrieve Lorenz curve data points for a specified number of bins.

## Best lorenz curve methodology selected by default:
lorenz <- get_gd(cum_welfare = datt_data$L,
                cum_population = datt_data$p,
                estimate = "lorenz",
                n_bins = 100) # must be specified, default is NULL.

## Specify lorenz curve methodology:
### Beta Lorenz ("lb")
lorenz_lb <- get_gd(cum_welfare = datt_data$L,
                   cum_population = datt_data$p,
                   estimate = "lorenz",
                   lorenz = "lb",
                   n_bins = 100)

### Quadratic Lorenz ("lq")
lorenz_lq <- get_gd(cum_welfare = datt_data$L,
                   cum_population = datt_data$p,
                   estimate = "lorenz",
                   lorenz = "lq",
                   n_bins = 100)

# estimate = 'params': retrieve regression parameters used for the lorenz curve estimation.
params <- get_gd(cum_welfare = datt_data$L,
                 cum_population = datt_data$p,
                 estimate = "params")
}

```

get_pip_info

Get PIP info

Description

Get information about the API.

Usage

```
get_pip_info(api_version = "v1", server = NULL)
```

Arguments

`api_version` API version. Currently only "v1" is supported.

`server` One non-empty server name, or NULL for the production server. For World Bank internal use only.

Value

list

Examples

```
if (interactive()) {
  get_pip_info()
}
```

<code>get_stats</code>	<i>Get poverty and inequality statistics</i>
------------------------	--

Description

Get poverty and inequality statistics

Usage

```
get_stats(
  country = "all",
  year = "all",
  povline = NULL,
  popshare = NULL,
  fill_gaps = FALSE,
  nowcast = FALSE,
  subgroup = NULL,
  welfare_type = c("all", "income", "consumption"),
  reporting_level = c("all", "national", "urban", "rural"),
  version = NULL,
  ppp_version = NULL,
  release_version = NULL,
  api_version = "v1",
  format = c("arrow", "rds", "json", "csv"),
  simplify = TRUE,
  server = NULL
)

get_wb(
  year = "all",
  povline = NULL,
  version = NULL,
```

```

    ppp_version = NULL,
    release_version = NULL,
    api_version = "v1",
    format = c("rds", "json", "csv"),
    simplify = TRUE,
    server = NULL
)

get_agg(
  year = "all",
  povline = NULL,
  version = NULL,
  ppp_version = NULL,
  release_version = NULL,
  aggregate = NULL,
  api_version = "v1",
  format = c("rds", "json", "csv"),
  simplify = TRUE,
  server = NULL
)

```

Arguments

country	Uppercase, three-letter country ISO codes or "all".
year	Integer-valued numeric years, "all", or "MRV" (most recent value). Character strings containing years are not accepted.
povline	One finite, non-negative numeric poverty line, or NULL. Ignored when popshare is supplied.
popshare	One numeric population share from 0 to 1, or NULL.
fill_gaps	One non-missing logical value. If TRUE, interpolate or extrapolate missing years. Also enabled by nowcast = TRUE.
nowcast	One non-missing logical value. If TRUE, return nowcast estimates and enable fill_gaps.
subgroup	NULL, "wb_regions", or "none". A non-NULL value uses the grouped endpoint and disables gap filling and nowcasting.
welfare_type	character: Welfare type either of c("all", "income", "consumption")
reporting_level	character: Geographical reporting level either of c("all", "national", "urban", "rural")
version	Data version in YYYYMMDD_PPP_XX_YY_IDENTITY format, or NULL. See get_versions().
ppp_version	One integer-valued numeric PPP year, a four-digit year string, or NULL.
release_version	A valid publication date in YYYYMMDD format, or NULL.
api_version	API version. Currently only "v1" is supported.
format	Response format: "arrow", "rds", "json", or "csv" for get_stats(). get_wb() and get_agg() accept "rds", "json", or "csv".

simplify	One non-missing logical value. If TRUE (the default), return simplified data.
server	One non-empty server name, or NULL for the production server. For World Bank internal use only.
aggregate	Aggregate name. See get_aux("country_list") for available options.

Value

A tibble of statistics when simplify = TRUE, or a pip_api list when simplify = FALSE.

Examples

```
if (interactive()) {
  # One country-year
  res <- get_stats(country = "AGO", year = 2000)

  # All years for a specific country
  res <- get_stats(country = "AGO", year = "all")

  # All countries and years
  res <- get_stats(country = "all", year = "all")

  # All countries and years w/ alternative poverty line
  res <- get_stats(country = "all", year = "all", povline = 3.2)

  # Fill gaps for years without available survey data
  res <- get_stats(country = "all", year = "all", fill_gaps = TRUE)

  # Proportion living below the poverty line
  res <- get_stats(country = "all", year = "all", popshare = .4)

  # World Bank global and regional aggregates
  res <- get_stats("all", year = "all", subgroup = "wb_regions")

  # Short hand to get WB global/regional stats
  res <- get_wb()

  # Short hand to get fcv stats
  res <- get_agg(aggregate = "fcv")

  # Custom aggregates
  res <- get_stats(c("ARG", "BRA"), year = "all", subgroup = "none")
}
```

get_versions

Get versions

Description

Get available data versions.

Usage

```
get_versions(api_version = "v1", server = NULL, simplify = TRUE)
```

Arguments

api_version	API version. Currently only "v1" is supported.
server	One non-empty server name, or NULL for the production server. For World Bank internal use only.
simplify	One non-missing logical value. If TRUE (the default), return simplified data.

Value

tibble or list

Examples

```
if (interactive()) {  
  get_versions()  
}
```

parse_error_body	<i>parse_error_body</i>
------------------	-------------------------

Description

Helper function to parse error messages generated by the PIP API

Usage

```
parse_error_body(resp)
```

Arguments

resp	A httr response
------	-----------------

Value

character

pip_is_transient	<i>pip_is_transient</i>
------------------	-------------------------

Description

Helper function to determine if an error is due to the number of requests going over the rate limit

Usage

```
pip_is_transient(resp)
```

Arguments

resp	A httr response
------	-----------------

Value

logical

retry_after	<i>retry_after</i>
-------------	--------------------

Description

Helper function to determine how much time to wait before a new query can be sent

Usage

```
retry_after(resp)
```

Arguments

resp	A httr response
------	-----------------

Value

numeric

unnest_ki	<i>Unnest the key indicators</i>
-----------	----------------------------------

Description

Unnest the key indicators

Usage

```
unnest_ki(out)
```

Arguments

out	parsed and simplified output from cp-key-indicators endpoint
-----	--

Value

data frame, unnested.

Functions

- `unnest_ki()`: takes the simplified output from cp-key-indicators endpoint and unnests it.

Index

- * **datasets**
 - datt_rural, [5](#)
 - datt_urban, [6](#)
- args_to_string, [2](#)
- build_request, [3](#)
- build_request_old, [3](#)
- call_aux, [4](#)
- change_grouped_stats_to_csv, [4](#)
- check_api, [5](#)
- datt_rural, [5](#)
- datt_urban, [6](#)
- delete_cache, [7](#)
- display_aux, [7](#)
- get_agg (get_stats), [19](#)
- get_aux, [8](#)
- get_cache_info, [14](#)
- get_countries (get_aux), [8](#)
- get_cp, [14](#)
- get_cp_ki, [15](#)
- get_cpi (get_aux), [8](#)
- get_dictionary (get_aux), [8](#)
- get_gd, [16](#)
- get_gdp (get_aux), [8](#)
- get_hfce (get_aux), [8](#)
- get_incgrp_coverage (get_aux), [8](#)
- get_interpolated_means (get_aux), [8](#)
- get_pip_info, [18](#)
- get_pop (get_aux), [8](#)
- get_pop_region (get_aux), [8](#)
- get_ppp (get_aux), [8](#)
- get_region_coverage (get_aux), [8](#)
- get_regions (get_aux), [8](#)
- get_stats, [19](#)
- get_survey_means (get_aux), [8](#)
- get_versions, [21](#)
- get_wb (get_stats), [19](#)
- parse_error_body, [22](#)
- pip_is_transient, [23](#)
- retry_after, [23](#)
- unnest_ki, [24](#)