Package 'INEbaseR'

July 1, 2019

Type Package

Title R package for obtaining and analyzing open data from INE
(Instituto Nacional de Estadistica) API

Version 0.1.0

Date 2018-01-01

Maintainer Andres Nacimiento <andresnacimiento@gmail.com>

Description R package for obtaining and analyzing open data from INE (Instituto Nacional de Estadistica) API.

License GPL (>= 3)

Encoding UTF-8

LazyData false

BuildResaveData false

Imports jsonlite, highcharter

RoxygenNote 6.1.1

NeedsCompilation no

Author Andres Nacimiento [aut, prg, cre],
Mariano Sanz [ctg],
Carlos J. Perez [aut, drt, pdr]

Depends R (>= 3.5.0)

R topics documented:

---

get_geographical   *Get geographical*

Description

This function returns geographical information

Usage

```
get_geographical(code = NULL, resource = "variable", all = FALSE,
  verbose = FALSE, natcode = NULL, geocode = NULL,
  exponential_notation = FALSE, help = FALSE)
```

Arguments

code	(string) serie id
resource	(string) resource to access, by default `resource = "variable"` to get serie metadata. Possible values are `variable`, `natcode` or `natcode_to_geocode`
all	(boolean) if `all = TRUE` you will get all natcodes
verbose	(boolean) show more information during the process
natcode	(int) geographical code from INE
geocode	(string) geographical code from Eurostat
exponential_notation	
	(boolean) to show or not exponential notation. `e.g. e+10`
help	(boolean) type any value for `resource` param and type `help = TRUE` to see params available for this `resource`.

Examples

```
get_geographical("IPC251522")
get_geographical(resource = "variable", help = TRUE)
get_geographical("IPC251522", resource = "natcode")
get_geographical(natcode = 34050000000, resource = "natcode_to_geocode")
```

---

get_metadata_crossing   *Get metadata crossing*

Description

This function returns data or metadata by metadata crossing

Usage

```
get_metadata_crossing(code = NULL, resource = "series", help = FALSE,
  query = NULL, p = 1, det = 0, tip = NULL, nlast = 1,
  lang = "ES")
```

Arguments

code	(string) operation identificator
resource	(string) resource to access, by default `resource = "metadata"` to get serie meta-data. Possible values are `series` or `data`
help	(boolean) type any value for `resource` param and type `help = TRUE` to see params available for this `resource`.
query	(string) string separated by `AND` with syntax `variable = value` using natural language
p	(int) periodicity, `p = 1` by default
det	(int) `det = 2` to see two levels of depth, specifically to access the `PubFechaAct` object, `det = 0` by default
tip	(string) `tip = M` to obtain the metadata (crossing variables-values) of the series
nlast	last n values
lang	(string) language used to obtain information

Examples

```
get_metadata_crossing(resource = "series", help = TRUE)
get_metadata_crossing("IPC", resource = "series", query = "Provincias = Madrid AND Tipo de dato = Variacion mens
get_metadata_crossing("IPC", resource = "data", query = "Provincias = Madrid AND Tipo de dato = Variacion mensua
```

---

get_operations	*Get operations*

Description

This function returns information about operations

Usage

```
get_operations(code = NULL, resource = "all", help = FALSE,
  ioe = FALSE, geographical_granularity = NULL,
  temporal_granularity = NULL, verbose = TRUE, lang = "ES")
```

Arguments

code	(string) operation identificator
resource	(string) resource to access, by default `resource = "metadata"` to get serie meta-data. Possible values are `all`, `metadata`, `variables_values` or `by_granularity`
help	(boolean) type any value for `resource` param and type `help = TRUE` to see params available for this `resource`.
ioe	(boolean) `TRUE` if code is in format `IO30138`, and `FALSE` by default
geographical_granularity	
	(string) geographical granularity
temporal_granularity	
	(string) temporal granularity
verbose	(boolean) show more information during the process
lang	(string) language used to obtain information

Examples

```
get_operations()
get_operations(resource = "all")
get_operations(resource = "all", help = TRUE)
get_operations("IPC", resource = "metadata")
get_operations("IPC", resource = "variables_values")
get_operations(resource = "by_granularity", geographical_granularity = "PROV")
```

---

get_publications   *Get publications*

Description

This function returns all available publications from an operation or a date

Usage

```
get_publications(code = NULL, resource = "all", help = FALSE,
  det = 0, ioe = FALSE, lang = "ES")
```

Arguments

code	(string) operation or publication indentificator
resource	(string) resource to access, by default `resource = "all"` to get serie metadata. Possible values are `all`, `operation` or `date`
help	(boolean) type any value for `resource` param and type `help = TRUE` to see params available for this `resource`.
det	(int) `det = 2` to see two levels of depth, specifically to access the `PubFechaAct` object, `det = 0` by default
ioe	(boolean) `TRUE` if code is in format `IO30138`, and `FALSE` by default
lang	(string) language used to obtain information

Examples

```
get_publications()
get_publications(resource = "all", help = TRUE)
get_publications("IPC", resource = "operation")
get_publications(8, resource = "date")
```

---

get_series *Get series*

Description

This function returns data or metadata from an operation, table or a serie

Usage

```
get_series(code = NULL, resource = "metadata", help = FALSE,
  ioe = FALSE, det = 0, tip = NULL, lang = "ES",
  date_start = NULL, date_end = NULL, nlast = NULL,
  classification = NULL, verbose = FALSE, benchmark = FALSE,
  geographical_granularity = NULL, temporal_granularity = NULL)
```

Arguments

code	(string) serie, operation or table identificator
resource	(string) resource to access, by default `resource = "metadata"` to get serie metadata. Possible values are `metadata`, `operation`, `values`, `table`, `metadataoperation`, `data`, `by_gra` or `nlast`
help	(boolean) type any value for `resource` param and type `help = TRUE` to see params available for this `resource`.
ioe	(boolean) `TRUE` if code is in format `IO30138`, and `FALSE` by default
det	(int) `det = 2` to see two levels of depth, specifically to access the `PubFechaAct` object, `det = 0` by default
tip	(string) `tip = M` to obtain the metadata (crossing variables-values) of the series
lang	(string) language used to obtain information
date_start	(string) start date in format `YYYY-MM-DD`
date_end	(string) end date in format `YYYY-MM-DD`
nlast	(int) last n serie values
classification	(string) serie classification, if `classification = NULL` this value will be auto-detected
verbose	(boolean) to show more information about this process, `verbose = FALSE` by default
benchmark	(boolean) used to measure the performance of the system, `benchmark = FALSE` by default.
geographical_granularity	
	(string) geographical granularity
temporal_granularity	
	(string) temporal granularity

Examples

```
get_series("IPC206449")
get_series(resource = "metadata", help = TRUE)
get_series("IPC", resource = "operation")
get_series("IPC206449", resource = "values")
get_series(22350, resource = "table")
get_series("IPC251541", resource = "nlast")
get_series("IPC206449", resource = "data", nlast = 5)
get_series("IPC", resource = "by_granularity", geographical_granularity = "CCAA", verbose = TRUE)
get_series("IPC251539", resource = "by_common_parameters", verbose = TRUE)
```

---

get_tables   *Get tables*

Description

This function returns data / metadata from tables

Usage

```
get_tables(code = NULL, resource = "operation", help = FALSE,
  grp = NULL, geo = 0, nlast = 0, det = 0, tip = NULL,
  ioe = FALSE, lang = "ES")
```

Arguments

code	operation (string/int) or table (int) identificator
resource	(string) resource to access, by default resource = "operation" to get tables of an operation. Possible values are operation, group, group_values or data
help	(boolean) type any value for resource param and type help = TRUE to see params available for this resource
grp	(int) group identification
geo	(int) use geo = 1 to access only tables with geographic content, geo = 0 by default
nlast	last n values
det	(int) use det = 2 to see two levels of depth, specifically to access the PubFechaAct object, det = 0 by default
tip	(string) use tip = "A" to view as friendly, specifically the view the field ultima_modificacion or use tip = "AM" to obtain the metadata (crossing variables-values) of the series and a friendly output.
ioe	(boolean) use ioe = TRUE if code is in format IO30138, and FALSE by default
lang	(string) language used to obtain information

Examples

```
get_tables("IPC")
get_tables(resource = "operation", help = TRUE)
get_tables(25, resource = "operation")
get_tables(22350, resource = "group")
get_tables(22350, grp = 81497, resource = "group_values")
get_tables(22350, nlast = 5, resource = "data")
```

---

get_values *Get values*

Description

This function returns values from a variable or variable operation

Usage

```
get_values(code = NULL, resource = "all", operation = NULL,
  help = FALSE, det = 0, ioe = FALSE, lang = "ES")
```

Arguments

code	(int) variable indentificator
resource	(string) resource to access, by default `resource = "all"` to get serie metadata. Possible values are `all` or `variable_operation`
operation	(string) operation indentificator
help	(boolean) type any value for `resource` param and type `help = TRUE` to see params available for this `resource`.
det	(int) `det = 1` to see the detail of the variable to which it belongs, `det = 0` by default
ioe	(boolean) `TRUE` if code is in format `IO30138`, and `FALSE` by default
lang	(string) language used to obtain information

Examples

```
get_values(115)
get_values(resource = "all", help = TRUE)
get_values(762, operation = "IPC", resource = "variable_operation")
```

---

get_variables *Get variables*

Description

This function returns all or operations variables

Usage

```
get_variables(code = NULL, resource = "all", help = FALSE,
  ioe = FALSE, lang = "ES")
```

Arguments

`code`	(string) operation identifier
`resource`	(string) resource to access, by default `resource = "metadata"` to get serie metadata. Possible values are `all` or `operation`
`help`	(boolean) type any value for `resource` param and type `help = TRUE` to see params available for this `resource`.
`ioe`	(boolean) `TRUE` if code is in format `IO30138`, and `FALSE` by default
`lang`	(string) language used to obtain information

Examples

```
get_variables()
get_variables(resource = "all", help = TRUE)
get_variables("IPC", resource = "operation")
```

`represent_series`	*Represent series*

Description

This function allows representing series data in maps or charts

Usage

```
represent_series(code = NULL, resource = "maps", help = FALSE,
  nlast = 0, date_start = NA, date_end = NA, classification = NULL,
  map_scale = 60, verbose = FALSE, benchmark = FALSE, det = 0,
  type = NA, lang = "ES")
```

Arguments

`code`	(string) serie identificator
`resource`	(string) resource to access, by default `resource = "all"` to get serie metadata. Possible values are `maps`, `plot` or `highcharts`
`help`	(boolean) type any value for `resource` param and type `help = TRUE` to see params available for this `resource`.
`nlast`	(int) last n serie data, if `nult = 0` this value will be auto-calculated
`date_start`	(date) start date in format (string) `YYYY-MM-DD`
`date_end`	(date) end date in format (string) `YYYY-MM-DD`
`classification`	(string) serie classification, if `classification = NULL` this value will be auto-detected
`map_scale`	(int) refers to the relationship or ratio between distance on a map and the corresponding distance on the ground. For example, on a `1:1000000` scale map, 1cm on the map equals 1km on the ground. Possible values are: `1,3,10,20` or `60`, and it's only for PROV or CCAA geographical granularity, `map_scale = 60` by default and `map_scale = NULL` for high detailed map.
`verbose`	(boolean) show more information during the process

benchmark	(boolean) used to measure the performance of the system, benchmark = FALSE by default.
det	(int) det = 2 to see two levels of depth, specifically to access the PubFechaAct object, det = 0 by default
type	(string) what type of plot should be drawn, type = "p" (for points) by default. See type in [plot](#)
lang	(string) language used to obtain information

Examples

```
represent_series("IPC251521", resource = "maps")
represent_series(resource = "maps", help = TRUE)
represent_series("IPC251521", resource = "maps", map_scale = NULL)
represent_series("IPC206449", resource = "plot", nlast = 5)
represent_series("IPC206449", resource = "highcharts", nlast = 5)
```

---

update_cache   *Update cache*

Description

This function allows to update cache data

Usage

```
update_cache(code = NULL, resource = "series", help = FALSE, n = 0,
  page = 1, pagination = TRUE, page_start = NA, page_end = NA,
  benchmark = FALSE, force = FALSE, ignore_series = NULL,
  tip = "M", det = 2, lang = "ES")
```

Arguments

code	(string) operation identificator
resource	(string) resource to access, by default resource = "all" to get serie metadata. Possible values are all or series
help	(boolean) type any value for resource param and type help = TRUE to see params available for this resource.
n	(int) number of operation to update starting from first operation getted from get_operations_all() function.
page	(int) page = 1 to obtain data of an specific page (to use this, pagination = FALSE).
pagination	(boolean) TRUE to obtain data page by page and FALSE by default.
page_start	(int) page_start = 1 start page range to obtain data (to use this, pagination = TRUE).
page_end	(int) page_end = 2 end page range to obtain data (to use this, pagination = TRUE).
benchmark	(boolean) used to measure the performance of the system, benchmark = FALSE by default.

force	(boolean) to force to update all cache data, `force = FALSE` by default.
ignore_series	(list) list of operation identificators to ignore. More slow series to cache are: 16, 49, 330 and 334
tip	(string) `tip = M` to obtain the metadata (crossing variables-values) of the series.
det	(int) `det = 2` to see two levels of depth, specifically to access the `PubFechaAct` object, `det = 0` by default
lang	(string) language used to obtain information

Examples

```
update_cache(resource = "series")
update_cache(resource = "series", help = TRUE)
```

Index