Package: opendataes (via r-universe)

July 11, 2024

Version 0.0.1

Title Interact with the datos.gob.es API to download public data from all of Spain

Description Easily interact with the API from http://datos.gob.es to download data over 19,000 files from all different provinces of Spain.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

ByteCompile true

BugReports https://github.com/ropenspain/opendataes/issues

URL https://github.com/ropenspain/opendataes, https://ropenspain.github.io/opendataes/

Imports httr, readr, tibble

RoxygenNote 7.1.1

Suggests testthat, covr, dplyr, knitr, rmarkdown

VignetteBuilder knitr

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

RemoteUrl https://github.com/rOpenSpain/opendataes

RemoteRef HEAD

RemoteSha b6c3848f8d93ad483c800a255519cfa9f041bb05

Contents

`data_list_correct` *Check data_list is in correct formats*

Description

When new checks come up, add them in the same format: logical tests first and then add them to the if statement

Usage

```
data_list_correct(raw_json)
```

Arguments

`raw_json`	Raw JSON response from datos.gob.es

extract_access_url *Extract access URL to the actual data from data_list*

Description

Extract access URL to the actual data from data_list

Usage

```
extract_access_url(data_list)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset

extract_data *Extract data from a data_list.*

Description

Extract data from a data_list.

Usage

```
extract_data(data_list, encoding, guess_encoding, ...)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset
encoding	The encoding passed to read (all) the files. Most cases should be resolved with either 'UTF-8', 'latin1' or 'ASCII'.
guess_encoding	A logical stating whether to guess the encoding. This is set to TRUE by default. Whenever guess_encoding is set to TRUE, the 'encoding' argument is ignored. If guess_encoding fails to guess the encoding, openes_load falls back to the encoding argument.
...	Arguments passed to read_csv and all other read_* functions.

Details

get_data will accept the end path of a data base and it will search for the access url. If the dataset is either a csv, xls, xlsx or xml, then it will attempt to read it. If it succeeds, it will return the data frame. If not, it will return the data frame with only one column containing all available access URL's.

For example, this URL: http://datos.gob.es/es/catalogo/a02002834-numero-de-centros-segun-ancho-de-banda-de-la-conexion-a-internet-que-tiene-el-centro6 says that it has a XML file but once you click on the 'download' XML, it redirects to a JavaScript based website that has the table. This file unfortunately is unreadable to the package.

`extract_dataset_name` *Extract the dataset names from data_list*

Description

For example, elecciones2016.csv and elecciones2014.csv

Usage

```
extract_dataset_name(data_list)
```

Arguments

`data_list`	A data_list similar to resp$result$items[[1]] that contains information on a dataset

`extract_description` *Extract description from data_list*

Description

Extract description from data_list

Usage

```
extract_description(data_list)
```

Arguments

`data_list`	A data_list similar to resp$result$items[[1]] that contains information on a dataset

`extract_endpath` *Extract the end path of the dataset that directs to datos.gob.es from a data_list*

Description

Extract the end path of the dataset that directs to datos.gob.es from a data_list

Usage

```
extract_endpath(data_list)
```

Arguments

`data_list`	A data_list similar to resp$result$items[[1]] that contains information on a dataset

extract_keywords *Extract keywords from data_list*

Description

Extract keywords from data_list

Usage

```
extract_keywords(data_list)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset

extract_language *Extract access languages available from data_list*

Description

Extract access languages available from data_list

Usage

```
extract_language(data_list)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset

extract_metadata *Extract all metadata from a data_list*

Description

Extract all metadata from a data_list

Usage

```
extract_metadata(data_list)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset

`extract_modified_date` *Extract the date at which the data was modified from data_list*

Description

The date is currently exported as a string but should be turned into a a date class

Usage

```
extract_modified_date(data_list)
```

Arguments

`data_list`	A data_list similar to resp$result$items[[1]] that contains information on a dataset

`extract_publisher_code` *Extract the publisher code of the dataset from data_list*

Description

Extract the publisher code of the dataset from data_list

Usage

```
extract_publisher_code(data_list)
```

Arguments

`data_list`	A data_list similar to resp$result$items[[1]] that contains information on a dataset

`extract_publisher_name` *Extract the publisher name of the dataset from data_list*

Description

Extract the publisher name of the dataset from data_list

Usage

```
extract_publisher_name(data_list)
```

Arguments

`data_list`	A data_list similar to resp$result$items[[1]] that contains information on a dataset

extract_release_date *Extract the date at which the data was submitted from data_list*

Description

The date is currently exported as a string but should be turned into a a date class

Usage

```
extract_release_date(data_list)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset

extract_url *Extract URL from datos.gob.es from data_list*

Description

Extract URL from datos.gob.es from data_list

Usage

```
extract_url(data_list)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset

extract_url_format *Extract the format of the dataset from data_list*

Description

For example, csv or xml

Usage

```
extract_url_format(data_list)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset

get_resp *Make GET requests with repeated trials*

Description

Make GET requests with repeated trials

Usage

```
get_resp(ch_url, attempts_left = 5, ...)
```

Arguments

ch_url	A url, preferably from the path_* functions
attempts_left	Number of attempts of trying to request from the website
...	Arguments passed to GET

get_resp_paginated *Make GET requests over several pages of an API*

Description

Make GET requests over several pages of an API

Usage

```
get_resp_paginated(ch_url, num_pages = 1, page = 0, ...)
```

Arguments

ch_url	URL to request from, preferably from the path_* functions
num_pages	Number of pages to request
page	The page at which the request should being. This should rarely be used
...	Arguments passed to GET

Value

the parsed JSON object as a list but inside the items slots it contains all data lists obtained from the pages specified in num_pages

is_publisher_available *Check if publisher is available in opendataes*

Description

Check if publisher is available in opendataes

Usage

```
is_publisher_available(data_list)
```

Arguments

data_list	A data_list similar to resp$result$items[[1]] that contains information on a dataset

make_url *Build a custom url using the httr url class*

Description

Build a custom url using the httr url class

Usage

```
make_url(path, param, ...)
```

Arguments

path	the end path of the dataset of interest
param	arguments for a query
...	any other arguments to building the path correctly. See modify_url

openes_keywords *Explore datasets by keywords and publishers in* https://datos.gob.es/

Description

Explore datasets by keywords and publishers in https://datos.gob.es/

Usage

```
openes_keywords(keyword, publisher)
```

Arguments

keyword	A character string specifying a keyword to identify a data set. For example, 'vivienda'.
publisher	A character string with the **publisher code**. Should be only one publisher code. See publishers_available for the permitted publisher codes.

Details

openes_keywords works only for searching for one keyword for a given publisher. For example, 'viviendas' for the Ayuntamiento of Barcelona. If there are no matches for a keyword-publisher combination, openes_keywords will raise an error stating that there are no matches.

openes_keywords returns a data frame with the following columns:

- description: a short description of each of the matched datasets in Spanish (Spanish is set as default if available, if not, then the first non-spanish language is chosen).
- publisher: the entity that publishes the dataset. See openes_load_publishers for all available publishers.
- is_readable: whether that dataset is currently readable by openes_load. See permitted_formats for currently available formats.
- path_id: the end path that identifies that dataset in the https://datos.gob.es/ API.
- url: the complete url of the dataset in https://datos.gob.es/. Note that this URL is not the access URL to the dataset but to the dataset's homepage in https://datos.gob.es/.

In most cases the user will need to narrow down their search because the result of openes_keywords will have too many datasets. Beware that for passing the result of this function to openes_load the final data frame needs to be narrowed down to only one dataset (that is, 1 row) and the structure needs to be the same as from the original output of openes_keywords (same column names, in the same order). See examples below.

Value

A tibble containing the matched datasets.

See Also

openes_load

Examples

```
## Not run:

library(dplyr)

kw <- openes_keywords("vivienda", "l01080193") # Ayuntamiento de Barcelona

kw

# Notice how we narrow down to only 1 dataset
dts <-
 kw
 filter(grepl("Precios", description))
 openes_load('ASCII')

# Notice that we had to specify the encoding because printing the dataset returns an error.
# If that happens to you, try figuring out the encoding with readr::guess_encoding(dts$data[[1]])
# and specify the most likely encoding in `openes_load`

dts$metadata

dts$data

## End(Not run)
```

---

openes_load *Extract data and metadata from a given data set of* https://datos.gob.es/

Description

Extract data and metadata from a given data set of https://datos.gob.es/

Usage

```
openes_load(x, encoding = "UTF-8", guess_encoding = TRUE, ...)
```

Arguments

x	A tibble given by openes_keywords only containing one dataset (1 row) or the end path of a dataset such as 'l01280148-seguridad-ciudadana-actuaciones-de-seccion-del-menor-en-educacion-vial-20141' from https://datos.gob.es/es/catalogo/l01280148-seguridad-ciudadana-actuaciones-de-seccion-del-menor-en-educacio

encoding	The encoding passed to read (all) the files. Most cases should be resolved with either 'UTF-8', 'latin1' or 'ASCII'.
guess_encoding	A logical stating whether to guess the encoding. This is set to TRUE by default. Whenever guess_encoding is set to TRUE, the 'encoding' argument is ignored. If guess_encoding fails to guess the encoding, openes_load falls back to the encoding argument.
...	Arguments passed to read_csv and the other related read_* functions from readr. Internally, openes_load determines the delimiter of the file being read but the arguments for each of these functions are practically the same, so it doesn't matter how openes_load determines the delimiter, any of the arguments will work on all read_* functions.

Details

openes_load can return two possible outcomes: either an empty list or a list with a slot called metadata and another slot called data. Whenever the path_id argument is an invalid dataset path, it will return an empty list. When path_id is a valid dataset path, openes_load will return an a list with the two slots described above.

For the metadata slot, openes_load returns a tibble with most available metadata of the dataset. The columns are:

- keywords: the available keywords from the dataset in the homepage of the dataset.
- language: the available languages of the dataset's metadata. Note that that this does not mean that the dataset is in different languages but only the metadata.
- description: a short description of the data being read.
- url: the complete url of the dataset in https://datos.gob.es/. Note that this URL is not the access URL to the dataset but to the dataset's homepage in https://datos.gob.es/.
- date_issued: the date at which the dataset was uploaded.
- date_modified: the date at which the last dataset was uploaded. If the dataset has only been uploaded once, this will return 'No modification date available'.
- publisher: the entity that publishes the dataset. See openes_load_publishers for all available publishers.
- publisher_data_url: the homepage of the dataset in the website of the publisher. This is helpful to look at the definitions of the columns in the dataset.

The metadata of the API can sometimes be returned in an incorrect order. For example, there are cases when there are several languages available and the order of the different descriptions are not in the same order of the languages. If you find any of these errors, try raising the issue directly to https://datos.gob.es/ as the package extracts all metadata in the same order as it is.

Whenever the metadata is in different languages, the resulting tibble will have the same numer of rows as there are languages containing the different texts in different languages and repeating the same information whenever it's similar across languages (such as the dates, which are language agnostic).

In case the API returns empty requests, both data and metadata will be empty tibble's with the same column names.

For the data slot, openes_load returns a list containing at least one tibble. If the dataset being request has file formats that openes_load can read (see permitted_formats) it will read those files. If that dataset has several files, then it will return a list of the same length as there are datasets where each slot in that list is a tibble with the data. If for some reason any of the datasets being read cannot be read, openes_load has a fall back mechanism that returns the format that attempted to read together with the URL so that the user can try to read the dataset directly. In any case, the result will always be a list with tibble's where each one could be the requested dataset (success) or a dataset with the format and url that attempted to read but failed (failure).

Inside the data slot, each list slot containing tibble's will be named according to the dataset that was read. When there is more than one dataset, the user can then enter the website in the url column in the metadata slot to see all names of the datasets. This is handy, for example, when the same dataset is repeated across time and we want to figure out which data is which from the slot.

The API of https://datos.gob.es/ is not completely homogenous because it is an aggregator of many different API's from different cities and provinces of Spain. openes_load can only read a limited number of file formats but will keep increasing as the package evolves. You can check the available file formats in permitted_formats. If the file format of the requested path_id is not readable, openes_load will return a list with only one tibble with all available formats with their respective data URL inside the data slot so that users can read the manually.

In a similar line, in order for openes_load to provide the safest behavior, it is very conservative in which publisher it can read from https://datos.gob.es/. Because some publishers do not have standardized datasets, reading many different publishers can become very messy. openes_load currently reads files from selected publishers because they offer standardized datasets which makes it safer to read. As the package evolves and the data quality improves between publishers, the package will include more publishers. See the publishers that the package can read in publishers_available.

Value

if path_id is a valid dataset path, a list with two slots: metadata and data. Each slot contains tibble's that contain either metadata or the data itself. If path_id is not a valid dataset path, it returns an empty list. See the details section for some caveats.

Examples

```
# For a dataset with only one file to read

example_id <- 'l01080193-fecundidad-en-madres-de-15-a-19-anos-de-la-ciudad-de-barcelona1'
some_data <- openes_load(example_id)

# Print the file to get some useful information
some_data

# Access the metadata
some_data$metadata

# Access the data. Note that the name of the dataset is in the list slot. Whenever
# there are different files being read, you might want to enter to homepage
# of the dataset in datos.gob.es with some_data$metadata$url or directly
# to the homepage of dataset at the publisher's website
```

```
# some_data$metadata$publisher_data_url
some_data$data


# For a dataset with many files

## Not run:
example_id <- 'l01080193-domicilios-segun-nacionalidad'
res <- openes_load(example_id)

# Note that you can see how many files were read in '# of files read'
res

# See how all datasets were read but we're not sure what each one means.
# Check the metadata and read the description. If that doesn't do it,
# go to the URL of the dataset from the metadata.
res$data

# Also note that some of the datasets were not read uniformly correct. For example,
# some of these datasets were read with more columns or more rows. This is left
# to the user to fix. We could've added new arguments to the `...` but that would
# apply to ALL datasets and it then becomes too complicated.


# Encoding problems

long <- "l01080193-descripcion-de-la-causalidad-de-los-accidentes"
string <- "-gestionados-por-la-guardia-urbana-en-la-ciudad-de-barcelona"

id <- paste0(long, string)
pl <- openes_load(id)

# The dataset is read successfully but once we print them, there's an error
pl$data
$`2011_ACCIDENTS_CAUSES_GU_BCN_.csv`
Error in nchar(x[is_na], type = "width") :
  invalid multibyte string, element 1

# This error is due to an encoding problem.
# We can use readr::guess_encoding to determine the encoding and reread

# This suggests an ASCII encoding
library(readr)
guess_encoding(pl$data[[1]])

pl <- openes_load(id, 'ASCII')

# Success
pl$data


# For exploring datasets with openes_keywords and piping to openes_load
library(dplyr)
```

```
kw <- openes_keywords("turismo", "l01080193") # Ayuntamiento de Barcelona#'
kw

dts <-
 kw
 filter(is_readable == TRUE,
        grepl("Tipos de propietarios", description))
 openes_load()

dts$metadata

dts$data

## End(Not run)
```

`openes_load_publishers` *Request all available publishers from* https://datos.gob.es/

Description

Request all available publishers from https://datos.gob.es/

Usage

```
openes_load_publishers()
```

Value

a tibble with two columns: publisher_code and publishers

Examples

```
openes_load_publishers()
```

path_begin_end_date *Build a url with a complete begin-end date/ prefix URL*

Description

Build a url with a complete begin-end date/ prefix URL

Usage

```
path_begin_end_date(start_date, end_date, param = NULL, ...)
```

Arguments

start_date	Start date in YYYYMMDD format
end_date	End date in YYYYMMDD format
param	Extra parameters to add to the url. For this function, this is useless because there's not further paths to the distribution end point. Keeping the argument for consistency
...	Extra arguments passed to build_url

path_catalog *Build a url with a complete catalog/ prefix URL*

Description

Build a url with a complete catalog/ prefix URL

Usage

```
path_catalog(path, param = NULL, ...)
```

Arguments

path	the end path of the dataset of interest
param	arguments for a query
...	any other arguments to building the path correctly. See modify_url

path_catalog_dataset *Build a url with a complete catalog/dataset prefix URL*

Description

Build a url with a complete catalog/dataset prefix URL

Usage

```
path_catalog_dataset(path, param = NULL, ...)
```

Arguments

path	the end path of the dataset of interest
param	arguments for a query
...	any other arguments to building the path correctly. See modify_url

path_datasets *Build a url with a complete datasets/ prefix URL*

Description

Build a url with a complete datasets/ prefix URL

Usage

```
path_datasets(param = NULL, ...)
```

Arguments

param	Extra parameters to add to the url. For this function, this is useless because there's not further paths to the dataset end point. Keeping the argument for consistency
...	Extra arguments passed to build_url

path_dataset_id *Build a url with an ID of a dataset*

Description

Build a url with an ID of a dataset

Usage

```
path_dataset_id(id, param = NULL, ...)
```

Arguments

id	dataset id from datos.gob.es such as 'l01080193-numero-total-de-edificios-con-viviendas-segun-numero-de-plantas'
param	Extra parameters to add to the url.
...	Extra arguments passed to build_url

path_distribution *Build a url with a complete distribution/ prefix URL*

Description

Build a url with a complete distribution/ prefix URL

Usage

```
path_distribution(param = NULL, ...)
```

Arguments

param	Extra parameters to add to the url. For this function, this is useless because there's not further paths to the distribution end point. Keeping the argument for consistency
...	Extra arguments passed to build_url

path_explore_keyword *Build a url to search for a given keyword in the datos.gob.es API*

Description

Build a url to search for a given keyword in the datos.gob.es API

Usage

```
path_explore_keyword(keyword)
```

Arguments

keyword	A string with the keyword to build the path with.

path_publishers *Build a url with a complete publishers/ prefix URL*

Description

Build a url with a complete publishers/ prefix URL

Usage

```
path_publishers(param = NULL, ...)
```

Arguments

param	Extra parameters to add to the url. For this function, this is useless because there's not further paths to the publishers end point. Keeping the argument for consistency
...	Extra arguments passed to build_url

permitted_formats *Current readable formats from* https://datos.gob.es/

Description

Current readable formats from https://datos.gob.es/

Usage

```
permitted_formats
```

Examples

```
permitted_formats
```

publishers_available *Available publishers that 'opendataes' can read*

Description

Available publishers that 'opendataes' can read

Usage

```
publishers_available
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 10 rows and 2 columns.

Value

a `tibble` with two columns: publishers and publisher_code

See Also

`openes_load_publishers`

Examples

```
publishers_available
```

translate_publisher *Translate publisher code to publisher name*

Description

Translate publisher code to publisher name

Usage

```
translate_publisher(code)
```

Arguments

code	A publisher code

Index