Package: ogdindiar (via r-universe)

July 5, 2024

Type Package

Title API Access to Datasets on Open Government Data - India Portal

Version 0.0.0.9005

Date 2015-05-02

Description Provides API access to selected datasets on Open Government Data - India Portal.

License MIT + file LICENSE

URL https://github.com/steadyfish/ogdindiar,
https://ropengov.github.io/ogdindiar/,
https://github.com/rOpenGov/ogdindiar

BugReports https://github.com/rOpenGov/ogdindiar/issues

Depends R (>= 3.1.1)

Imports curl, magrittr, plyr (>= 1.8.2), RCurl (>= 1.95-4.6), RJSONIO (>= 1.3-0), rvest, stringr, xml2

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.2

X-schema.org-isPartOf http://ropengov.org/

X-schema.org-keywords ropengov

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

RemoteUrl https://github.com/rOpenGov/ogdindiar

RemoteRef HEAD

RemoteSha b07b44b0dc934869d8b15d0dcf0c115072785a21

Contents

`download_dataset` *Download dataset*

Description

Given a download link, obtained by using either ‘search_for_datasets‘ or ‘get_datasets_from_a_catalog‘, this function will download the file.

Usage

```
download_dataset(urllink, filepath = NULL)
```

Arguments

`urllink`	Download link/url
`filepath`	If specified, the file will be downloaded to the specified location. If unspecified, it will be saved in the tmp directory

`fetch_data` *Load data from the Government of India API.*

Description

`fetch_data` is the main function from this package to load the entire data set from the Government of India API.

Usage

```
fetch_data(
  res_id,
  filter = NULL,
  select = NULL,
  sort = NULL,
  field_type_correction = TRUE,
  max_obs = 500
)
```

Arguments

`res_id` a string, JSON data resource id

`filter` a named vector, specifying equality constrainsts of the form "variable" = "condition"

`select` a vector, specifying variables/fields to be selected

`sort` a named vector, specifying sort order in the form "variable" = "order"

`field_type_correction` boolean, whether to apply field type correction. All data fields are downloaded as character and then corrected (if at all) based on accompanying metadata

`max_obs` an integer, specifying maximum no of observations to fetch (will be rounded UP to the nearest 100)

Value

list a list of 2 elements - data from the Government of India API, and metadata, additional information about the fields

Examples

```
## Not run:
### fetch a dataset using it's resource id and your personal API key
# Basic Use:
fetch_data(res_id = "60a68cec-7d1a-4e0e-a7eb-73ee1c7f29b7")

# Advanced Use, specifying additional parameters
fetch_data(res_id = "60a68cec-7d1a-4e0e-a7eb-73ee1c7f29b7"
           filter = c("state" = "Maharashtra"),
           select = c("s_no_","constituency","state"),
           sort = c("s_no_" = "asc","constituency" = "desc"))

## End(Not run)
```

get_count *Get count of elements that were returned from JSON data query*

Description

This will return the no of elements that were returned from JSON data query.

Usage

```
get_count(x)
```

Arguments

x	a list, i.e. a JSON data object

Value

no_elements an integer, no of elements to download a value between 1 to 100

Examples

```
## Not run:
###Return no of elements from a JSON data object (obtained using get_JSON_doc())
get_count(x = JSON_doc)

## End(Not run)
```

get_data *Get data from the JSON data object*

Description

This will return the data from the JSON data object.

Usage

```
get_data(x)
```

Arguments

x	a list, i.e. a JSON data object

Value

data a list, data from the JSON data object

Examples

```
## Not run:
###Return data from a JSON data object (obtained using get_JSON_doc())
get_data(x = JSON_doc)

## End(Not run)
```

`get_datasets_from_a_catalog`

get data sets for a catalog

Description

Get the list of data sets and related info for a catalog

Usage

```
get_datasets_from_a_catalog(
  catalog_link,
  limit_dataset_pages = 5L,
  limit_datasets = 10L
)
```

Arguments

`catalog_link`	Link to the catalog
`limit_dataset_pages`	Limit the number of pages that should be requested and parsed, to acquire the datasets. Default is 5. Set to Inf to request all.
`limit_datasets`	Request more pages until the number of datasets obtained reaches this limit. Default is 10. Set to Inf to request all.

See Also

search_for_datasets

Examples

```
## Not run:
get_datasets_from_a_catalog(
'https://data.gov.in/catalog/session-wise-statistical-information-relating-questions-rajya-sabha',
limit_dataset_pages = 7, limit_datasets = 10)

## End(Not run)
```

get_field_names *Get field/variable names from the JSON data object*

Description

This will return field names from the JSON data object.

Usage

```
get_field_names(x)
```

Arguments

x	a list, i.e. a JSON data object

Value

field_names a vector/list, of field names for JSON data object

Examples

```
## Not run:
###Return field names from a JSON data object (obtained using get_JSON_doc())
get_field_names(x = JSON_doc)

## End(Not run)
```

get_field_type *Get field/variable types from the JSON data object*

Description

This will return field types from the JSON data object.

Usage

```
get_field_type(x)
```

Arguments

x	a list, i.e. a JSON data object

Value

field_types a list/vector, field type of each of the fields

Examples

```
## Not run:
###Return field types from a JSON data object (obtained using get_JSON_doc())
get_field_names(x = JSON_doc)

## End(Not run)
```

---

get_JSON_doc *Get JSON data for requested data resource*

Description

get_JSON_doc will return infomation about the requested resource. Ideally, will be just used internally.

Usage

```
get_JSON_doc(
  link = "https://data.gov.in/api/datastore/resource.json?",
  res_id,
  offset,
  no_elements,
  filter,
  select,
  sort,
  verbose = FALSE
)
```

Arguments

link	a string, general JSON data link
res_id	a string, JSON data resource id
offset	an integer, offset of 1 corresponds to 100 elements
no_elements	an integer, no of elements to download a value between 1 to 100
filter	a named vector, specifying equality constrainsts of the form "variable" = "condition"
select	a vector, specifying variables/fields to be selected
sort	a named vector, specifying sort order in the form "variable" = "asc"
verbose	a boolean, specifying whether to print verbose messages

Value

JSON data object i.e. a list

Examples

```
## Not run:
library(RCurl)
library(RJSONIO)
# Return 100 elements from a hotels data resource
JSON_doc = get_JSON_doc(link="http://data.gov.in/api/datastore/resource.json?",
   res_id="0749068c-a590-4a07-a571-e9df5dddcc8a",
   offset=0,
   no_elements=100)

## End(Not run)
```

ogdindiar — *ogdindiar: Provides API access to selected datasets on Open Government Data - India Portal.*

Description

The ogdindiar package provides three categories of important functions: Downloading entire datasets, Downloading specific elelments based on certain conditions, and Search for data sets.

ogdindiar functions

fetch_data search_datasets

ogdindia_api_key — *Get or set OGDINDIA_API_KEY value*

Description

The API wrapper functions in this package all rely on a Open Government Data India API key residing in the environment variable `OGDINDIA_API_KEY`. The easiest way to accomplish this is to set it in the '.Renviron' file in your home directory.

Usage

```
ogdindia_api_key(force = FALSE)
```

Arguments

`force`	Force setting a new PassiveTotal API key for the current environment?

Value

atomic character vector containing the Open Government Data India API key

rectify_field_type *Apply field type correction based on accompanied metadata*

Description

`rectify_field_type` will convert select fields to numeric based on accompanied metadata

Usage

```
rectify_field_type(d_in, d_fields)
```

Arguments

`d_in`	a data.frame on which the correction is to be applied.
`d_fields`	a data.frame containing fields metadata

Value

data corrected data.frame

Examples

```
## Not run:
rectify_field_type(data_stage2, data_field_type)

## End(Not run)
```

search_for_datasets *Search for data sets*

Description

This function scrapes the data.gov.in search results and returns most of the information available for the datasets. As this function doesn't use API and just parses the web pages, there needs to delay between successive requests, and there should be limits to the number of pages that the function downloads from the web. For a particular search input, there may be multiple pages of search results. Each result page contains a list of catalogs. And each catalog contains multiple pages, with each page containing a list of data sets. There are default limits at each one of these stages. Make them 'Inf' if you need to get all the results or if you don't expect a large number of results. Please refer to vignette for a detailed overview.

Usage

```
search_for_datasets(
  search_terms,
  limit_catalog_pages = 5L,
  limit_catalogs = 10L,
  return_catalog_list = FALSE,
  limit_dataset_pages = 5L,
  limit_datasets = 10L
)
```

Arguments

`search_terms` Either one string with multiple words separated by space, or a character vector with all the search terms

`limit_catalog_pages` Number of pages of search results to request. Default is 5. Set to Inf to get all.

`limit_catalogs` Number of catalogs that the function should parse to get the data sets. Default is 5. Set to Inf to get all.

`return_catalog_list` Default is FALSE. If TRUE, the function will not look for data sets, and will only return the list of catalogs found.

`limit_dataset_pages` Limit the number of pages that should be requested and parsed, to acquire the datasets. Default is 5. Set to Inf to request all.

`limit_datasets` Request more pages until the number of datasets obtained reaches this limit. Default is 10. Set to Inf to request all.

See Also

get_datasets_from_a_catalog

Examples

```
## Not run:
# Basic Use:
search_for_datasets('train usage')

# Advanced Use, specifying additional parameters
search_for_datasets(search_terms = c('state', 'gdp'),
                    limit_catalog_pages = 1,
                    limit_catalogs = 3,
                    limit_dataset_pages = 2)
search_for_datasets(search_terms = c('state', 'gdp'),
                    limit_catalog_pages = 2,
                    return_catalog_list = TRUE)

## End(Not run)
```

`to_data_frame` *Convert data from list to a data.frame*

Description

`to_data_frame` will convert data from 'list' to a 'data.frame'.

Usage

```
to_data_frame(lst_elmnt)
```

Arguments

`lst_elmnt`	a list of data from a JSON data object

Value

data a data.frame, data from the JSON data object

Examples

```
## Not run:
###Convert a list to data.frame
to_data_frame(x = get_data(JSON_list))

## End(Not run)
```

Index