

Package ‘enigma’

July 8, 2015

Title Client for the 'Enigma' 'API'

Description The company 'Enigma' (<https://enigma.io>) holds many public 'datasets' from governments, companies, universities, and organizations. 'Enigma' provides an 'API' for data, 'metadata', and statistics on each of the 'datasets'. 'enigma' is a client to interact with the 'Enigma' 'API', including getting the data and 'metadata' for 'datasets' in 'Enigma', as well as collecting statistics on 'datasets'. In addition, you can download a 'gzipped' 'csv' file of a 'dataset' if you want the whole 'dataset'. An 'API' key from 'Enigma' is required to use 'enigma'.

Version 0.2.0

License MIT + file LICENSE

LazyData true

URL <https://github.com/ropengov/enigma>

BugReports <https://github.com/ropengov/enigma/issues>

VignetteBuilder knitr

Imports methods, utils, httr (>= 1.0.0), jsonlite, plyr

Suggests knitr, testthat, dplyr, ggplot2, ggthemes

NeedsCompilation no

Author Scott Chamberlain [aut, cre]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2015-07-08 09:50:14

R topics documented:

enigma	2
enigma_data	2
enigma_fetch	4
enigma_metadata	6
enigma_stats	7
rate_limit	9

Index**10**

enigma	<i>enigma, an R client for Enigma.io</i>
--------	--

Description

Enigma holds government data and provides a really nice set of APIs for data, metadata, and stats on each of the datasets. That is, you can request a dataset itself, metadata on the dataset, and summary statistics on the columns of each dataset.

Details

The functions:

- [enigma_data](#) - Fetch and dataset, and filter on columns or rows.
- [enigma_metadata](#) - Get metadata on datasets.
- [enigma_stats](#) - Get columnwise statistics on datasets.
- [enigma_fetch](#) - Get gzipped csv of a dataset. Goes along with [enigma_read](#)
- [rate_limit](#) - Get columnwise statistics on datasets.

An API key is required to use this package. You can supply your key in each function call, or store in your key in your `.Rprofile` file, or execute `options(enigmaKey = "<your key>")` in your R console. Obtain an API key by creating an account with Enigma at <http://enigma.io>, then get an API key from your Enigma account page.

enigma_data	<i>Fetch a dataset from Enigma.</i>
-------------	-------------------------------------

Description

Fetch a dataset from Enigma.

Usage

```
enigma_data(dataset = NULL, limit = 500, select = NULL,
  conjunction = NULL, sort = NULL, page = NULL, where = NULL,
  search = NULL, key = NULL, ...)
```

Arguments

dataset	Dataset name. Required.
limit	(numeric) Number of rows of the dataset to return. Default (and max): 500
select	(character) Vector of columns to be returned with each row. Default is to return all columns.
conjunction	one of "and" or "or". Only applicable when more than one search or where parameter is provided. Default: "and"
sort	(character) Sort rows by a particular column in a given direction. + denotes ascending order, - denotes descending. See examples.
page	(numeric) Paginate row results and return the nth page of results. Pages are calculated based on the current limit, which defaults to 50.
where	(character) Filter results with a SQL-style "where" clause. Only applies to numerical columns - use the search parameter for strings. Valid operators are >, < and =. Only one where clause per request is currently supported.
search	(character) Filter results by only returning rows that match a search query. By default this searches the entire table for matching text. To search particular fields only, use the query format "@fieldname query". To match multiple queries, the (or) operator can be used eg. "query1 query2".
key	(character) Required. An Enigma API key. Supply in the function call, or store in your .Rprofile file, or do options(enigmaKey = "<your key>"). Obtain an API key by creating an account with Enigma at http://enigma.io , then obtain an API key from your account page.
...	Named options passed on to GET

Examples

```
## Not run:
# After obtaining an API key from Enigma's website, pass in your key to the function call
# or set in your options (see above instructions for the key parameter)
# If you pass in your key to the function call use the key parameter

# White house visitor list
enigma_data(dataset='us.gov.whitehouse.visitor-list')

# White house visitor list - selecting three columns
enigma_data(dataset='us.gov.whitehouse.visitor-list', select=c('namelast', 'visitee_namelast',
'last_updatedby'))

# White house visitor list - sort by last name
enigma_data(dataset='us.gov.whitehouse.visitor-list', sort='+namelast')

# White house visitor list - get rows where total_people > 5
enigma_data(dataset='us.gov.whitehouse.visitor-list', where='total_people > 5')

# White house visitor list - search for Vitale in full name field
## remove the 2nd at symbol before running
# enigma_data(dataset='us.gov.whitehouse.visitor-list', search='@namefull=Vitale')
```

```

# White house visitor list - search for SOPHIA in first name field
## remove the 2nd at symbol before running
# enigma_data(dataset='us.gov.whitehouse.visitor-list', search='@namefirst=SOPHIA')

# Domestic Market Flight Statistics (Final Destination)
dataset='us.gov.dot.rita.trans-stats.air-carrier-statistics.t100d-market-all-carrier'
enigma_data(dataset=dataset)

# Search for 'apple' in the Crunchbase dataset, and get two columns back
enigma_data(dataset='com.crunchbase.info.companies.acquisition', search='apple',
select=c('acquisition', 'price_amount'))

# conjunction parameter, compare these two queries
enigma_data(dataset = 'us.gov.dol.ogesdw.msha.msha-accident',
  where = c('degree_injury_cd > 2', 'no_injuries > 1'), conjunction = "and")
enigma_data(dataset = 'us.gov.dol.ogesdw.msha.msha-accident',
  where = c('degree_injury_cd > 2', 'no_injuries > 1'), conjunction = "or")

## End(Not run)

```

enigma_fetch

Download a gzipped csv file of a dataset.

Description

Download a gzipped csv file of a dataset.

Usage

```

enigma_fetch(dataset = NULL, select = NULL, search = NULL, where = NULL,
  conjunction = NULL, sort = NULL, path = NULL, overwrite = TRUE,
  key = NULL, ...)

```

```

enigma_read(input)

```

Arguments

dataset	Dataset name. Required.
select	(character) Vector of columns to be returned with each row. Default is to return all columns.
search	(character) Filter results by only returning rows that match a search query. By default this searches the entire table for matching text. To search particular fields only, use the query format "@fieldname query". To match multiple queries, the (or) operator can be used eg. "query1 query2".
where	(character) Filter results with a SQL-style "where" clause. Only applies to numerical columns - use the search parameter for strings. Valid operators are >, < and =. Only one where clause per request is currently supported.

conjunction	one of "and" or "or". Only applicable when more than one search or where parameter is provided. Default: "and"
sort	(character) Sort rows by a particular column in a given direction. + denotes ascending order, - denotes descending. See examples.
path	File name and path of output zip file. Defaults to write a zip file to your home directory with name of the dataset, and file extension .csv.gz.
overwrite	Will only overwrite existing path if TRUE.
key	(character) Required. An Enigma API key. Supply in the function call, or store in your .Rprofile file, or do options(enigmaKey = "<your key>"). Obtain an API key by creating an account with Enigma at http://enigma.io , then obtain an API key from your account page.
...	Named options passed on to GET
input	The output from enigma_fetch or a path to a file downloaded from Enigma.io

Details

Note that [enigma_fetch](#) downloads the file, and gives back a path to the file. In a separate function, [enigma_read](#), you can read in the data. [enigma_fetch](#) doesn't read in data in case the file is very large which may make your R session crash or slow down significantly.

Examples

```
## Not run:
# After obtaining an API key from Enigma's website, pass in your key to the function call
# or set in your options (see above instructions for the key parameter)
# If you pass in your key to the function call use the key parameter

# Fetch the Crunchbase companies info dataset
res <- enigma_fetch(dataset='com.crunchbase.info.companies.acquisition')
head(enigma_read(res))

# Use the select parameter to limit fields returned
res <- enigma_fetch(dataset='com.crunchbase.info.companies.acquisition',
  select = c("acquisition", "price_amount", "acquired_year"))
head(enigma_read(res))

# Use the search parameter to query entire table or particular fields
res <- enigma_fetch(dataset='com.crunchbase.info.companies.acquisition',
  search = "holdings")
head(enigma_read(res))

# Use the search parameter to query entire table or particular fields
res <- enigma_fetch(dataset='com.crunchbase.info.companies.acquisition',
  where = "price_amount > 0", select = c("acquisition", "price_amount"))
df <- enigma_read(res)
options(scipen = 99)
head(df)

# Use the sort parameter to sort results
```

```

res <- enigma_fetch(dataset='com.crunchbase.info.companies.acquisition',
  sort = "-price_amount", select = c("acquisition", "price_amount"))
df <- enigma_read(res)
options(scipen = 99)
head(df)

# Piping workflow
library('dplyr')
enigma_fetch(dataset='com.crunchbase.info.companies.acquisition') %>%
  enigma_read %>%
  glimpse

# Curl debugging
library('httr')
enigma_fetch(dataset='com.crunchbase.info.companies.acquisition', config=verbose())

## End(Not run)

```

enigma_metadata	<i>Search for metadata on a dataset from Enigma.</i>
-----------------	--

Description

Search for metadata on a dataset from Enigma.

Usage

```
enigma_metadata(dataset = NULL, key = NULL, ...)
```

Arguments

dataset	Dataset name. Required.
key	(character) Required. An Enigma API key. Supply in the function call, or store in your .Rprofile file, or do <code>options(enigmaKey = "<your key>")</code> . Obtain an API key by creating an account with Enigma at http://enigma.io , then obtain an API key from your account page.
...	Named options passed on to GET

Details

Notice when you run the examples that the format of output is different for the "parent nodes" vs. the "table nodes". Where the parent nodes have `ouput$meta` slots for paths, immediate nodes and children tables, while the table nodes have `ouput$meta` slots for info, table, ancestor datapaths, database boundary datapath, database boundary label, database boundary tables, and paths, and an additional slot for description of table column attributes.

Examples

```

## Not run:
# After obtaining an API key from Enigma's website, pass in your key to the function call
# or set in your options (see above instructions for the key parameter)
# If you pass in your key to the function call use the key parameter

# Parent node response attributes from Whitehouse dataset
## US white house
enigma_metadata(dataset='us.gov.whitehouse')

## UCLA Ethnic power relations dataset
enigma_metadata(dataset='edu.ucla.epr')

# Table node response attributes
## US white house visitor list table
enigma_metadata(dataset='us.gov.whitehouse.visitor-list')

## UCLA Ethnic power relations dataset - armed conflict table
enigma_metadata('edu.ucla.epr.ethnic-armed-conflict')

## End(Not run)

```

enigma_stats

Get statistics on columns of a dataset from Enigma.

Description

Get statistics on columns of a dataset from Enigma.

Usage

```

enigma_stats(dataset = NULL, select, conjunction = NULL, operation = NULL,
  by = NULL, of = NULL, limit = 500, search = NULL, where = NULL,
  sort = NULL, page = NULL, key = NULL, ...)

```

Arguments

dataset	Dataset name. Required.
select	(character) Column to get statistics on. Required.
conjunction	one of "and" or "or". Only applicable when more than one search or where parameter is provided. Default: "and"
operation	(character) Operation to run on a given column. For a numerical column, valid operations are sum, avg, stddev, variance, max, min and frequency. For a date column, valid operations are max, min and frequency. For all other columns, the only valid operation is frequency. Defaults to all available operations based on the column's type.

by	(character) Compound operation to run on a given pair of columns. Valid compound operations are sum and avg. When running a compound operation query, the of parameter is required (see below).
of	(character) Numerical column to compare against when running a compound operation. Required when using the by parameter. Must be a numerical column.
limit	(numeric) Limit the number of frequency, compound sum, or compound average results returned. Max: 500; Default: 500.
search	(character) Filter results by only returning rows that match a search query. By default this searches the entire table for matching text. To search particular fields only, use the query format "@fieldname query". To match multiple queries, the (or) operator can be used eg. "query1 query2".
where	(character) Filter results with a SQL-style "where" clause. Only applies to numerical columns - use the search parameter for strings. Valid operators are >, < and =. Only one where clause per request is currently supported.
sort	(character) Sort frequency, compound sum, or compound average results in a given direction. + denotes ascending order, - denotes descending
page	(numeric) Paginate frequency, compound sum, or compound average results and return the nth page of results. Pages are calculated based on the current limit, which defaults to 500.
key	(character) Required. An Enigma API key. Supply in the function call, or store in your .Rprofile file, or do options(enigmaKey = "<your key>"). Obtain an API key by creating an account with Enigma at http://enigma.io , then obtain an API key from your account page.
...	Named options passed on to GET

Examples

```
## Not run:
# After obtaining an API key from Enigma's website, pass in your key to the function call
# or set in your options (see above instructions for the key parameter)
# If you pass in your key to the function call use the key parameter

# stats on a varchar column
cbase <- 'com.crunchbase.info.companies.acquisition'
enigma_stats(dataset=cbase, select='acquired_month')

# stats on a numeric column
enigma_stats(dataset=cbase, select='price_amount')

# stats on a date column
pakistan <- 'gov.pk.secp.business-registry.all-entities'
enigma_metadata(dataset=pakistan)
enigma_stats(dataset=pakistan, select='registration_date')

# stats on a date column, by the average of a numeric column
aust <- 'gov.au.government-spending.federal-contracts'
enigma_metadata(dataset=aust)
enigma_stats(dataset=aust, select='contractstart', by='avg', of='value')
```



```

# Get frequency of distances traveled, and plot
## get columns for the air carrier dataset
dset <- 'us.gov.dot.rita.trans-stats.air-carrier-statistics.t100d-market-all-carrier'
enigma_metadata(dset)$columns$table[,c(1:4)]
out <- enigma_stats(dset, select='distance')
library("ggplot2")
library("ggthemes")
df <- out$result$frequency
df <- data.frame(distance=as.numeric(df$distance), count=as.numeric(df$count))
ggplot(df, aes(distance, count)) +
  geom_bar(stat="identity") +
  geom_point() +
  theme_grey(base_size = 18) +
  labs(y="flights", x="distance (miles)")

# conjunction parameter, compare these two queries
enigma_stats(dataset = 'us.gov.dol.ogesdw.msha.msha-accident', select = "no_injuries",
  where = c('degree_injury_cd > 2', 'no_injuries > 1'), conjunction = "and")
enigma_stats(dataset = 'us.gov.dol.ogesdw.msha.msha-accident', select = "no_injuries",
  where = c('degree_injury_cd > 2', 'no_injuries > 1'), conjunction = "or")

## End(Not run)

```

rate_limit

Get rate limit data.

Description

Get rate limit data.

Usage

```
rate_limit(key = NULL, ...)
```

Arguments

key	(character) Required. An Enigma API key. Supply in the function call, or store in your .Rprofile file, or do <code>options(enigmaKey = "<your key>")</code> . Obtain an API key by creating an account with Enigma at http://enigma.io , then obtain an API key from your account page.
...	Named options passed on to GET

Examples

```

## Not run:
rate_limit()

## End(Not run)

```

Index

enigma, [2](#)
enigma-package (enigma), [2](#)
enigma_data, [2](#), [2](#)
enigma_fetch, [2](#), [4](#), [5](#)
enigma_metadata, [2](#), [6](#)
enigma_read, [2](#), [5](#)
enigma_read(enigma_fetch), [4](#)
enigma_stats, [2](#), [7](#)

GET, [3](#), [5](#), [6](#), [8](#), [9](#)

rate_limit, [2](#), [9](#)