Package 'censys'

December 31, 2016

Title Tools to Query the 'Censys' API

Version 0.1.0

Description The 'Censys' public search engine enables researchers to quickly ask questions about the hosts and networks that compose the Internet. Details on how 'Censys' was designed and how it is operated are available at <https://www.censys.io/about>. Both basic and extended research access queries are made available. More information on the SQL dialect used by the 'Censys' engine can be found at <https://cloud.google.com/bigquery/docs/reference/legacy-sql>.

URL https://github.com/hrbrmstr/censys

BugReports https://github.com/hrbrmstr/censys/issues

Depends R (>= 3.2.2)

License AGPL + file LICENSE

LazyData true

Suggests testthat

Imports httr, jsonlite, stringi, purrr, utils

RoxygenNote 5.0.1

NeedsCompilation no

Author Bob Rudis [aut, cre]

Maintainer Bob Rudis <brudis@rapid7.com>

Repository CRAN

Date/Publication 2016-12-31 16:37:02

R topics documented:

censys 2
censys_export_download 3
censys_export_job_status 3
censys_get_job_result 4
censys_get_job_status 5
censys_query 6

censys_report 7
censys_search 8
censys_series 9
censys_series_details 9
censys_start_export 10
get_series 11
view_document 12
view_result 13
view_series 13

Index **15**

`censys` *Tools to access the Censys API*

Description

Censys is a search engine that allows computer scientists to ask questions about the devices and networks that compose the Internet. Driven by I nternet-wide scanning, Censys lets researchers find specific hosts and create aggregate reports on how devices, websites, and certificates are configured and deployed.

Details

The Censys API provides programmatic access to the same data accessible through web interface (`https://censys.io/`).

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Censys tutorial: `https://censys.io/tutorial`

Author(s)

Bob Rudis (brudis@rapid7.com)

References

`https://censys.io/about`; `https://censys.io/static/censys.pdf`

censys_export_download *Download export job files to a specified directory*

Description

Download export job files to a specified directory

Usage

```
censys_export_download(job_id, path)
```

Arguments

job_id	Censys export job id (from calling censys_start_export())
path	Location for downloaded data.

Value

API call result (invisibly)

Examples

```
## Not run:
q <- censys_start_export("
SELECT location.country, count(ip) FROM ipv4.20161206 GROUP BY location.country
")
censys_export_job_status(q$job_id)
censys_export_download(q$job_id, "~/Data")

## End(Not run)
```

censys_export_job_status *Get status of a Censys export job*

Description

The Get Job Status endpoint lets you retrieved information about a previously submitted job. The status field will return "pending" until the job has completed at which time status will be "success" or "error". On success, the output will define download_paths, a list of files that can be downloaded for the next 24 hours. After 24 hour, the job status will change to "expired" and the files will no longer be retrievable.

Usage

```
censys_export_job_status(job_id)
```

Arguments

job_id	Censys export job id (from calling censys_start_export())

Details

You must have both CENSYS_API_ID and CENSYS_API_SECRET present in the R environment for the functions in this package to work. It is highly suggested that you place those in ~/.Renviron at least for interactive work.

Value

API call result (invisibly)

References

Censys SQL query syntax: https://censys.io/query; API doc: https://censys.io/api/v1/docs/export

Examples

```
## Not run:
q <- censys_start_export("
SELECT location.country, count(ip) FROM ipv4.20161206 GROUP BY location.country
")
censys_export_job_status(q$job_id)
censys_export_download(q$job_id, "~/Data")

## End(Not run)
```

---

censys_get_job_result *Get results of completed Censys SQL query job*

Description

The Get Results endpoint allows you to retrieve results of a query after it has completed successfully.

Usage

```
censys_get_job_result(job_id, page = 1)
```

Arguments

job_id	Censys job id (from calling censys_query())
page	page number of paged results

Details

You must have both CENSYS_API_ID and CENSYS_API_SECRET present in the R environment for the functions in this package to work. It is highly suggested that you place those in ~/.Renviron at least for interactive work.

References

Censys SQL query syntax: https://censys.io/query; API doc: https://www.censys.io/api/v1/docs/search

Examples

```
## Not run:
q <- censys_query("SELECT p443.https.tls.cipher_suite.name, count(ip) FROM ipv4
                   WHERE p443.https.tls.validation.browser_trusted=true
                   GROUP BY p443.https.tls.cipher_suite.name;")
censys_get_job_status(q$job_id)
censys_get_job_result(q$job_id)

## End(Not run)
```

---

censys_get_job_status *Get status of a Censys SQL query job*

Description

The Get Job Status endpoint allows you to determine whether a job has completed. Once it has successfully finished, you can then retrieved results with the Get Results endpoint. Data should be posted as a JSON request document.

Usage

```
censys_get_job_status(job_id)
```

Arguments

job_id	Censys job id (from calling censys_query())

Details

You must have both CENSYS_API_ID and CENSYS_API_SECRET present in the R environment for the functions in this package to work. It is highly suggested that you place those in ~/.Renviron at least for interactive work.

Value

API call result (invisibly)

References

Censys SQL query syntax: https://censys.io/query; API doc: https://www.censys.io/api/v1/docs/search

Examples

```
## Not run:
q <- censys_query("SELECT p443.https.tls.cipher_suite.name, count(ip) FROM ipv4
                   WHERE p443.https.tls.validation.browser_trusted=true
                   GROUP BY p443.https.tls.cipher_suite.name;")
censys_get_job_status(q$job_id)
censys_get_job_result(q$job_id)

## End(Not run)
```

`censys_query` *Issue SQL Queries against the Censys API*

Description

The Query API allows executing SQL queries against our daily snapshots and raw data analogous to the Query web interface. Queries are executed asynchronously. You must first start a job, then check its status. Once a job has completed, you can view paginated results using the get results endpoint. Jobs typically require 15-30 seconds to execute; results can be viewed for 24 hours after the job completed. Ddefinition endpoints are also exposed where you can list the series and view series details (i.e., list tables and schema).

Usage

```
censys_query(sql)
```

Arguments

`sql`	SQL query string

Details

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Value

API call result (invisibly)

References

Censys SQL query syntax: https://censys.io/query; API doc: https://www.censys.io/api/v1/docs/search

Examples

```
## Not run:
q <- censys_query("SELECT p443.https.tls.cipher_suite.name, count(ip) FROM ipv4
                   WHERE p443.https.tls.validation.browser_trusted=true
                   GROUP BY p443.https.tls.cipher_suite.name;")
censys_get_job_status(q$job_id)
censys_get_job_result(q$job_id)

## End(Not run)
```

`censys_report` *Create aggregate reports on the breakdown of a field in the result set of a query*

Description

The build report endpoint lets you run aggregate reports on the breakdown of a field in a result set analogous to the "Build Report" functionality in the front end. For example, if you wanted to determine the breakdown of cipher suites selected by Top Million Websites.

Usage

```
censys_report(index, query, field, buckets = 50)
```

Arguments

`index`	The search index to be queried. Must be one of either `ipv4`, `websites`, or `certificates`.
`query`	The query to be executed. For example, `80.http.get.headers.server: nginx`.
`field`	The field you are running a breakdown on in "dot notation", e.g. `location.country_code`.
`buckets`	(optional) The maximum number of values to be returned in the report. Maximum: `500`. Default: `50`.

Details

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Value

list of information about the endpoint

References

Censys search syntax: https://www.censys.io/ipv4/help; API doc: https://www.censys.io/api/v1/docs/report

Examples

```
## Not run:
censys_report("ipv4", "80.http.get.headers.server: Apache",
              "location.country", 100)

## End(Not run)
```

`censys_search` *Perform queries against Censys data*

Description

The search endpoint allows searches against the current Censys data in the IPv4, Top Million Websites, and Certificates indexes using the same search syntax as the primary site. The endpoint returns a paginated result set of hosts (or websites or certificates) that match the search.

Usage

```
censys_search(index = c("ipv4", "websites", "certificates"), query,
  page = 1, fields = NULL)
```

Arguments

`index`	The search index to be queried. Must be one of either `ipv4`, `websites`, or `certificates`.
`query`	The query to be executed. For example, `80.http.get.headers.server: nginx`.
`page`	The page of the result set to be returned. The number of pages in the result set is available under metadata in any request. By default, the API will return the first page of results. "1" indexed.
`fields`	(optional) character vector of fields you would like returned in the result set in "dot notation", e.g. `location.country_code`.

Details

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Value

list of information about the endpoint

References

Censys search syntax: https://www.censys.io/ipv4/help; API doc: https://www.censys.io/api/v1/docs/search

Examples

```
## Not run:
censys_search("ipv4", "80.http.get.headers.server: Apache", 2,
              c("ip", "location.country", "autonomous_system.asn"))

## End(Not run)
```

---

censys_series *List all series that can be queried from the SQL interface*

Description

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Usage

```
censys_series()
```

References

Censys SQL query syntax: https://censys.io/query; API doc: https://www.censys.io/api/v1/docs/search

Examples

```
## Not run:
censys_series()

## End(Not run)
```

---

censys_series_details *Get details about a series, including the list of tables and schema for the series*

Description

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Usage

```
censys_series_details(series = "ipv4")
```

Arguments

series	series name (call censys_series() to see them all). Defaults to ipv4.

References

Censys SQL query syntax: https://censys.io/query; API doc: https://www.censys.io/api/v1/docs/search

Examples

```
## Not run:
censys_series_details("ipv4")

## End(Not run)
```

---

censys_start_export *Export large datasets and structured records from Censys to JSON or CSV files*

Description

The Export API allows exporting large datasets and structured records from Censys to JSON or CSV files. Unlike the query endpoint, there are no limits on the type or amount of data returned.

Usage

```
censys_start_export(query, format = c("csv", "json"), flatten = TRUE,
  compress = FALSE, delimiter = ",", headers = TRUE)
```

Arguments

query	the SQL query to be executed
format	the format data should be output in. Must be csv or json. Default: csv.
flatten	should nested and repeated fields in the query results be flattened. Default: true.
compress	should data files be gzipped. Default: false.
delimiter	delimiter to use between fields in the exported data. Default: ",".
headers	should a header row be included in results files. Default: true.

Details

Exports are executed as asynchronous jobs. You must first start a job. If the query is parsed successfully, the call will return a job ID, which is used in subsequent calls to get job. Once a job executes successfully, the get job endpoint will provide a list of 128MB JSON files that are available for download for 24 hours. Jobs typically require 15-30 seconds to execute.

You must have both CENSYS_API_ID and CENSYS_API_SECRET present in the R environment for the functions in this package to work. It is highly suggested that you place those in ~/.Renviron at least for interactive work.

Value

API call result (invisibly)

References

Censys SQL query syntax: https://censys.io/query; API doc: https://censys.io/api/v1/docs/export

Examples

```
## Not run:
q <- censys_start_export("
SELECT location.country, count(ip) FROM ipv4.20161206 GROUP BY location.country
")
censys_export_job_status(q$job_id)
censys_export_download(q$job_id, "~/Data")

## End(Not run)
```

`get_series` *Retrieve data on the types of scans Censys regularly performs ("series").*

Description

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Usage

```
get_series()
```

Value

list of series metadata

Note

Censys ID & Secret must be in the R environment

References

Census API: https://www.censys.io/api/v1/docs/data

Examples

```
## Not run:
scans <- get_series()
names(scans$raw_series)
names(scans$primary_series)

## End(Not run)
```

`view_document` *Retrieve data that Censys has about a specific host, website, or certificate.*

Description

The view endpoint fetches the structured data we have about a specific host, website, or certificate once you know the host's IP address, website's domain, or certificate's SHA-256 fingerprint.

Usage

```
view_document(index, id)
```

Arguments

`index`	The search index the document is in. Must be one of either `ipv4`, `websites`, or `certificates`.
`id`	The ID of the document you are requesting. In the `ipv4` index, this is IP address (e.g., `192.168.1.1`), domain name in the `websites` index (e.g., `google.com`) and SHA-256 fingerprint in the `certificates` index (e.g., `9d3b51a6b80daf76e074730f19dc01e643ca0`

Details

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Value

list of information about the endpoint

Examples

```
## Not run:
view_document("google.com")

## End(Not run)
```

view_result *Retrieve data on a particular scan "result"*

Description

Generally used after a call to either `get_series` or `view_series`.

Usage

```
view_result(series_id, result_id)
```

Arguments

`series_id`	Censys series id (e.g. `"22-ssh-banner-full_ipv4"`)
`result_id`	Censys series result id (e.g. `"20150930T0056"`)

Details

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Value

list of specific series result details

References

Census API: https://www.censys.io/api/v1/docs/data

view_series *Retrieve data that Censys has about a particular series*

Description

A "series" is a scan of the same protocol and destination accross time, including the list of scans.

Usage

```
view_series(series_id)
```

Arguments

`series_id`	Censys series id (e.g. `"22-ssh-banner-full_ipv4"`)

Details

You must have both `CENSYS_API_ID` and `CENSYS_API_SECRET` present in the R environment for the functions in this package to work. It is highly suggested that you place those in `~/.Renviron` at least for interactive work.

Value

list of specific series details

References

Census API: https://www.censys.io/api/v1/docs/data

Examples

```
## Not run:
view_series("443-https-tls-full_ipv4")

## End(Not run)
```

Index

censys, 2
censys-package (censys), 2
censys_export_download, 3
censys_export_job_status, 3
censys_get_job_result, 4
censys_get_job_status, 5
censys_query, 6
censys_report, 7
censys_search, 8
censys_series, 9
censys_series_details, 9
censys_start_export, 10

get_series, 11

view_document, 12
view_result, 13
view_series, 13