Package 'searchConsoleR'

October 14, 2022

Title Google Search Console R Client

Version 0.4.0

Description Provides an interface with the Google Search Console, formally called Google Webmaster Tools.

URL http://code.markedmondson.me/searchConsoleR/

BugReports https://github.com/MarkEdmondson1234/searchConsoleR/issues

Depends R (>= 3.2.0)

License MIT + file LICENSE

LazyData true

Imports googleAuthR (>= 1.0.0), stringr (>= 1.0.0)

Suggests shiny (>= 0.12.1), knitr, rmarkdown, testthat

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Mark Edmondson [aut, cre], Jennifer Bryan [ctb], Parzakonis Manos [ctb]

Maintainer Mark Edmondson <r@sunholo.com>

Repository CRAN

Date/Publication 2019-09-06 10:50:02 UTC

R topics documented:

`add_sitemap` *Submit a sitemap.*

Description

See here for details: https://developers.google.com/webmaster-tools/v3/sitemaps/submit

Usage

```
add_sitemap(siteURL, feedpath)
```

Arguments

`siteURL`	The URL of the website to delete. Must include protocol (http://).
`feedpath`	The URL of the sitemap to submit. Must include protocol (http://).

Value

TRUE if successful, raises an error if not.

See Also

Other sitemap admin functions: `delete_sitemap`, `list_sitemaps`

`add_website` *Adds website to Search Console*

Description

Adds website to Search Console

Usage

```
add_website(siteURL)
```

Arguments

`siteURL`	The URL of the website to add.

Value

TRUE if successful, raises an error if not.

See Also

Other search console website functions: delete_website, list_websites

---

`crawl_errors` *Fetch a time-series of Googlebot crawl errors.*

Description

Get a list of errors detected by Googlebot over time. See here for details: https://developers.google.com/webmaster-tools/v3/urlcrawlerrorscounts/query

Usage

```
crawl_errors(siteURL, category = "all", platform = c("all", "mobile",
  "smartphoneOnly", "web"), latestCountsOnly = FALSE)
```

Arguments

`siteURL`	The URL of the website to delete. Must include protocol (http://).
`category`	Crawl error category. Defaults to 'all'
`platform`	The user agent type. 'all', 'mobile', 'smartphoneOnly' or 'web'.
`latestCountsOnly`	Default FALSE. Only the latest crawl error counts returned if TRUE.

Details

The timestamp is converted to a date as they are only available daily.

Category is one of: authPermissions, manyToOneRedirect, notFollowed, notFound, other, roboted, serverError, soft404.

Platform is one of: mobile, smartphoneOnly or web.

Value

dataframe of errors with $platform $category $count and $timecount.

See Also

Other working with search console errors: error_sample_url, fix_sample_url, list_crawl_error_samples

delete_sitemap *Delete a sitemap.*

Description

See here for details: https://developers.google.com/webmaster-tools/v3/sitemaps/delete

Usage

```
delete_sitemap(siteURL, feedpath)
```

Arguments

siteURL	The URL of the website you are deleting the sitemap from. Must include protocol (http://).
feedpath	The URL of the sitemap to delete. Must include protocol (http://).

Value

TRUE if successful, raises an error if not.

See Also

Other sitemap admin functions: add_sitemap, list_sitemaps

delete_website *Deletes website in Search Console*

Description

Deletes website in Search Console

Usage

```
delete_website(siteURL)
```

Arguments

siteURL	The URL of the website to delete.

Value

TRUE if successful, raises an error if not.

See Also

Other data fetching functions: list_sitemaps

Other search console website functions: add_website, list_websites

`error_sample_url` *Shows details of errors for individual sample URLs*

Description

pageURL is the relative path (without the site) of the sample URL. It must be one of the URLs returned by list_crawl_error_samples. For example, for the URL https://www.example.com/pagename on the site https://www.example.com/, the url value is pagename (string)

Category is one of: authPermissions, manyToOneRedirect, notFollowed, notFound, other, roboted, serverError, soft404.

Platform is one of: mobile, smartphoneOnly or web.

Usage

```
error_sample_url(siteURL, pageURL, category = "notFound",
  platform = "web")
```

Arguments

`siteURL`	The URL of the website to delete. Must include protocol (http://).
`pageURL`	A PageUrl taken from list_crawl_error_samples.
`category`	Crawl error category. Default 'notFound'.
`platform`	User agent type. Default 'web'.

Details

See here for details: https://developers.google.com/webmaster-tools/v3/urlcrawlerrorssamples/get

Value

Dataframe of $linkedFrom, with the calling URLs $last_crawled, $first_detected and a $exampleURL

See Also

Other working with search console errors: `crawl_errors`, `fix_sample_url`, `list_crawl_error_samples`

`fix_sample_url` *Mark As Fixed the individual sample URLs*

Description

pageURL is the relative path (without the site) of the sample URL. It must be one of the URLs returned by list_crawl_error_samples. For example, for the URL https://www.example.com/pagename on the site https://www.example.com/, the url value is pagename (string)

Category is one of: authPermissions, manyToOneRedirect, notFollowed, notFound, other, roboted, serverError, soft404.

Platform is one of: mobile, smartphoneOnly or web.

Usage

```
fix_sample_url(siteURL, pageURL, category = "notFound",
  platform = "web")
```

Arguments

`siteURL`	The URL of the website to delete. Must include protocol (http://).
`pageURL`	A PageUrl taken from list_crawl_error_samples.
`category`	Crawl error category. Default 'notFound'.
`platform`	User agent type. Default 'web'.

Details

See here for details: https://developers.google.com/webmaster-tools/v3/urlcrawlerrorssamples/markAsFixed

Value

TRUE if successful, raises an error if not.

See Also

Other working with search console errors: `crawl_errors`, `error_sample_url`, `list_crawl_error_samples`

list_crawl_error_samples

Lists a site's sample URLs for crawl errors.

Description

Category is one of: authPermissions, manyToOneRedirect, notFollowed, notFound, other, roboted, serverError, soft404.

Platform is one of: mobile, smartphoneOnly or web.

Usage

```
list_crawl_error_samples(siteURL, category = "notFound",
  platform = "web")
```

Arguments

siteURL	The URL of the website to delete. Must include protocol (http://).
category	Crawl error category. Default 'notFound'.
platform	User agent type. Default 'web'.

Details

See here for details: https://developers.google.com/webmaster-tools/v3/urlcrawlerrorssamples

Value

A dataframe of $pageUrl, $last_crawled, $first_detected, $response

See Also

Other working with search console errors: crawl_errors, error_sample_url, fix_sample_url

list_sitemaps

Gets sitemap information for the URL supplied.

Description

See here for details: https://developers.google.com/webmaster-tools/v3/sitemaps

Usage

```
list_sitemaps(siteURL)
```

Arguments

siteURL	The URL of the website to get sitemap information from. Must include protocol (http://).

Value

A list of two dataframes: $sitemap with general info and $contents with sitemap info.

See Also

Other data fetching functions: delete_website

Other sitemap admin functions: add_sitemap, delete_sitemap

---

list_websites *Retrieves dataframe of websites user has in Search Console*

Description

Retrieves dataframe of websites user has in Search Console

Usage

```
list_websites()
```

Value

a dataframe of siteUrl and permissionLevel

See Also

Other search console website functions: add_website, delete_website

---

scr_auth *Do OAuth2 authentication*

Description

Do OAuth2 authentication

Usage

```
scr_auth(token = NULL, email = NULL)
```

Arguments

token	Where you want to save the auth file, or an existing token or file location of a token to authenticate with
email	An email you have authenticated with previously

Details

Run this function first time to authenticate with Google in your browser.

After initial authentication, a `sc.oauth` will be saved to your working directory, where your authentication details are kept. Keep this file safe.

If you want to reauthenticate, delete this file from your directory or run `scr_auth(new_user = TRUE)`

Multiple accounts

You can authenticate with a new auth file for each account. Supply argument `token` with the name of the cache file you want to use e.g. `scr_auth(token = "one.httr-oauth")` for one account, `scr_auth(token = "another.httr-oauth")` for a different account.

Auto-authentication

You can choose to auto-authenticate by moving your `sc.httr-oauth` or by creating a Google OAuth service account JSON file.

Specify an environment variable in R via a `.Renviron` file or using Sys.setenv which point to the file location of your chosen authentication file. See Startup

Once you have set the environment variable `SC_AUTH_FILE` to a valid file location, the function will look there for authentication details upon loading the library meaning you will not need to call `scr_auth()` yourself as you would normally.

An example `.Renviron` file is below:

`SC_AUTH_FILE = "/Users/bob/auth/sc.oauth"`

`SC_AUTH_FILE` can be either a auth file for a token generated by gar_auth or service account JSON ending with file extension `.json`

Your own Google Project

Be default the Google Project used is shared by all users, this is usually sufficient, but you could choose to create your own Google Project and turn on the Webmaster APIs.

You can then download your own client JSON, and set by placing in the `GAR_CLIENT_JSON` environment argument. See gar_set_client for details.

Service accounts

If you use the service account JSON, you will need to add the service account email to your Search Console users to see data e.g. `xxxx@yyyyyy.iam.gserviceaccount.com`

See Also

gar_auth

---

searchConsoleR *searchConsoleR*

Description

Provides an interface with the Google Search Console API v3, formally called Google Webmaster Tools.

To get started, use `googleAuthR::gar_auth()` to authenticate.

Search analytics

search_analytics - download Google SEO data into an R dataframe.

Website admin

list_websites - list websites in your Google Search Console.

add_website - add a website to your Google Search Console.

delete_website - delete a website from your Google Search Console.

Sitemaps

list_sitemaps - list sitemaps recognised in Google Search Console.

add_sitemap - add sitemap URL location to Google Search Console.

delete_sitemap - remove sitemap URL location in Google Search Console.

Error listings

crawl_errors - list types of crawl errors googlebot has found.

list_crawl_error_samples - lists example URLs with errors.

error_sample_url - details about an example URL error.

fix_sample_url - mark a URL as fixed.

search_analytics *Query search traffic keyword data*

Description

Download your Google SEO data.

Usage

```
search_analytics(siteURL, startDate = Sys.Date() - 93,
  endDate = Sys.Date() - 3, dimensions = NULL, searchType = c("web",
  "video", "image"), dimensionFilterExp = NULL,
  aggregationType = c("auto", "byPage", "byProperty"), rowLimit = 1000,
  prettyNames = TRUE, walk_data = c("byBatch", "byDate", "none"))
```

Arguments

siteURL	The URL of the website you have auth access to.
startDate	Start date of requested range, in YYYY-MM-DD.
endDate	End date of the requested date range, in YYYY-MM-DD.
dimensions	Zero or more dimensions to group results by: "date", "country", "device", "page" , "query" or "searchAppearance"
searchType	Search type filter, default 'web'.
dimensionFilterExp	A character vector of expressions to filter. e.g. ("device==TABLET", "country~~GBR")
aggregationType	How data is aggregated.
rowLimit	How many rows to fetch. Ignored if walk_data is "byDate"
prettyNames	If TRUE, converts SO 3166-1 alpha-3 country code to full name and creates new column called countryName.
walk_data	Make multiple API calls. One of ("byBatch","byDate","none")

Details

startDate: Start date of the requested date range, in YYYY-MM-DD format, in PST time (UTC -8:00). Must be less than or equal to the end date. This value is included in the range.

endDate: End date of the requested date range, in YYYY-MM-DD format, in PST time (UTC -8:00). Must be greater than or equal to the start date. This value is included in the range.

dimensions: [Optional] Zero or more dimensions to group results by.

- 'date'
- 'country'
- 'device'

- 'page'
- 'query'
- 'searchAppearance' (can only appear on its own)

The grouping dimension values are combined to create a unique key for each result row. If no dimensions are specified, all values will be combined into a single row. There is no limit to the number of dimensions that you can group by apart from `searchAppearance` can only be grouped alone. You cannot group by the same dimension twice.

Example: `c('country', 'device')`

dimensionFilterExp: Results are grouped in the order that you supply these dimensions. dimensionFilterExp expects a character vector of expressions in the form: ("device==TABLET", "country~~GBR", "dimension operator expression")

- dimension
  - 'country'
  - 'device'
  - 'page'
  - 'query'
  - 'searchAppearance'
- operator
  - '~~' meaning 'contains'
  - '==' meaning 'equals'
  - '!~' meaning 'notContains'
  - '!=' meaning 'notEquals
- expression
  - country: an ISO 3166-1 alpha-3 country code.
  - device: 'DESKTOP','MOBILE','TABLET'.
  - page: not checked, a string in page URLs without hostname.
  - query: not checked, a string in keywords.
  - searchAppearance: 'AMP_BLUE_LINK', 'RICHCARD'

searchType: [Optional] The search type to filter for. Acceptable values are:

- "web": [Default] Web search results
- "image": Image search results
- "video": Video search results

aggregationType: [Optional] How data is aggregated.

- If aggregated by property, all data for the same property is aggregated;
- If aggregated by page, all data is aggregated by canonical URI.
- If you filter or group by page, choose auto; otherwise you can aggregate either by property or by page, depending on how you want your data calculated;

See the API documentation to learn how data is calculated differently by site versus by page. Note: If you group or filter by page, you cannot aggregate by property. If you specify any value other than auto, the aggregation type in the result will match the requested type, or if you request an invalid type, you will get an error. The API will never change your aggregation type if the requested type is invalid. Acceptable values are:

- "auto": [Default] Let the service decide the appropriate aggregation type.
- "byPage": Aggregate values by URI.
- "byProperty": Aggregate values by property.

batchType: [Optional] Batching data into multiple API calls

- byBatch Use the API call to batch
- byData Runs a call over each day in the date range.
- none No batching

Value

A dataframe with columns in order of dimensions plus metrics, with attribute "aggregationType"

See Also

Guide to Search Analytics: https://support.google.com/webmasters/answer/6155685 API docs: https://developers.google.com/webmaster-tools/v3/searchanalytics/query

Examples

```
## Not run:

   library(searchConsoleR)
   scr_auth()
   sc_websites <- list_websites()

   default_fetch <- search_analytics("http://www.example.com")

   gbr_desktop_queries <-
       search_analytics("http://www.example.com",
                        start = "2016-01-01", end = "2016-03-01",
                        dimensions = c("query", "page"),
                        dimensionFilterExp = c("device==DESKTOP", "country==GBR"),
                        searchType = "web", rowLimit = 100)

   batching <-
        search_analytics("http://www.example.com",
                        start = "2016-01-01", end = "2016-03-01",
                        dimensions = c("query", "page", "date"),
                        searchType = "web", rowLimit = 100000,
                        walk_data = "byBatch")


## End(Not run)
```

Index