

# Package ‘searchConsoleR’

August 21, 2015

**Title** Google Search Console APIv3 R Client

**Version** 0.1.2

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

**URL** <http://www.google.com/webmasters/tools/>,  
<https://developers.google.com/webmaster-tools/>

**Depends** R (>= 3.2.0)

**License** MIT + file LICENSE

**LazyData** true

**Imports** googleAuthR (>= 0.1.1), stringr (>= 1.0.0)

**Suggests** shiny (>= 0.12.1), knitr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Mark Edmondson [aut, cre],  
Jennifer Bryan [ctb]

**Maintainer** Mark Edmondson <r@sunholo.com>

**Repository** CRAN

**Date/Publication** 2015-08-21 16:57:54

## R topics documented:

add_sitemap . . . . .	2
add_website . . . . .	2
crawl_errors . . . . .	3
delete_sitemap . . . . .	4
delete_website . . . . .	4
error_sample_url . . . . .	5
fix_sample_url . . . . .	6
list_crawl_error_samples . . . . .	7
list_sitemaps . . . . .	7

list_websites . . . . .	8
searchConsoleR . . . . .	8
search_analytics . . . . .	9

<b>Index</b>	<b>12</b>
--------------	-----------

---

add_sitemap	<i>Submit a sitemap.</i>
-------------	--------------------------

---

### 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	<i>Adds website to Search Console</i>
-------------	---------------------------------------

---

### 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/urlcrawleerrorscounts/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	<i>Delete a sitemap.</i>
----------------	--------------------------

---

**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	<i>Deletes website in Search Console</i>
----------------	--

---

**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	<i>Shows details of errors for individual sample URLs</i>
------------------	---

---

### 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	<i>Mark As Fixed the individual sample URLs</i>
----------------	---

---

### 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/urlcrawlorssamples/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	<i>Retrieves dataframe of websites user has in Search Console</i>
---------------	---

---

**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](#)

---

searchConsoleR	<i>searchConsoleR</i>
----------------	-----------------------

---

**Description**

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

**Details**

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	<i>Query search traffic keyword data</i>
------------------	--

---

**Description**

Download your Google SEO data.

**Usage**

```
search_analytics(siteURL, startDate, endDate, dimensions = NULL,
  searchType = c("web", "video", "image"), dimensionFilterExp = NULL,
  aggregationType = c("auto", "byPage", "byProperty"), rowLimit = 1000,
  prettyNames = TRUE)
```

**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" or "query"
searchType	Search type filter, default 'web'.

dimensionFilterExp	A character vector of expressions to filter. e.g. <code>c("device==TABLET", "country~~GBR")</code>
aggregationType	How data is aggregated.
rowLimit	How many rows, maximum is 5000.
prettyNames	If TRUE, converts SO 3166-1 alpha-3 country code to full name and creates new column called <code>countryName</code> .

## 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'

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, but 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'
- 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.

**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.

### 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>

# Index

`add_sitemap`, 2, 4, 8, 9

`add_website`, 2, 4, 8, 9

`crawl_errors`, 3, 5–7, 9

`delete_sitemap`, 2, 4, 8, 9

`delete_website`, 3, 4, 8, 9

`error_sample_url`, 3, 5, 6, 7, 9

`fix_sample_url`, 3, 5, 6, 7, 9

`list_crawl_error_samples`, 3, 5, 6, 7, 9

`list_sitemaps`, 2, 4, 7, 9

`list_websites`, 3, 4, 8, 9

`search_analytics`, 9, 9

`searchConsoleR`, 8

`searchConsoleR`-package  
(`searchConsoleR`), 8