Package: serpstatr (via r-universe)

September 16, 2024

Type Package

Title 'Serpstat' API Wrapper

Version 0.3.0

URL https://serpstat.com/api/

Description The primary goal of 'Serpstat' API <https://serpstat.com/api/> is to reduce manual SEO (search engine optimization) and PPC (pay-per-click) tasks. You can automate your keywords research or competitors analysis with this API wrapper.

License MIT + file LICENSE

Encoding UTF-8

Imports httr (>= 1.4.2)

RoxygenNote 7.3.2

Suggests testthat

NeedsCompilation no

Author Alex Danilin [aut, cre]

Maintainer Alex Danilin <alexnikdanilin@gmail.com>

Date/Publication 2024-08-16 14:40:05 UTC

Repository https://alex-danilin.r-universe.dev

RemoteUrl https://github.com/cran/serpstatr

RemoteRef HEAD

RemoteSha afaff107709e1248a5ef15dab3e6a38c1c2dae09

Contents

`sst_au_get_summary` *Website audit summary*

Description

Returns the basic stats for the finished audit returns, including number of checked pages, issues by priority, domain optimization score.

Usage

```
sst_au_get_summary(api_token, report_id = NULL)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`report_id`	(required) ID of the audit report to get data from.

Value

Returns the basic metrics for audited website.

API docs

Check all the values for request and response fields here.

API credits consumption

0.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_au_get_summary(
  api_token = api_key,
  report_id = report_id
  )$data

## End(Not run)
```

---

sst_au_start *Start a website audit*

Description

Starts a project website audit with the current audit settings. Uses default audit settings if no settings were changed.

Usage

```
sst_au_start(api_token, project_id = NULL)
```

Arguments

api_token	(required) Serpstat API token from your profile.
project_id	(required) ID of the project in Serpstat.

Value

Returns the ID of the audit report.

API docs

Check all the values for request and response fields here.

API credits consumption

1 audit credit for each checked page.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_au_start(
  api_token  = api_token,
  project_id = 12345
  )$data$reportId

## End(Not run)
```

`sst_bl_domain_summary` *Backlinks summary*

Description

Returns the overview of the backlinks profile for the domain.

Usage

```
sst_bl_domain_summary(
  api_token,
  domain,
  search_type = "domain",
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`domain`	(required) A domain name to analyze.
`search_type`	(optional) Default value is 'domain' for domain only (site.com). See API docs for more details.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns aggregated backlinks data for the domain.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per request.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_bl_domain_summary(
  api_token     = api_token,
  domain        = 'serpstat.com',
  search_type   = 'domain',
  return_method = 'list'
)$data

## End(Not run)
```

`sst_bl_referring_domains` *Referring domains*

Description

Returns the list of referring domains with main backlinks metrics for each domain.

Usage

```
sst_bl_referring_domains(
  api_token,
  domain,
  search_type = "domain",
  page = 1,
  size = 100,
  sort = "domain_rank",
  order = "desc",
  filter = NULL,
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`domain`	(required) A domain name to analyze.
`search_type`	(required) Default value is 'domain' for domain data only (site.com). See API docs for more details.
`page`	(optional) Response page number if there are many pages in response. Default is 1.
`size`	(optional) Response page size. Default is 100.
`sort`	(optional) A field to sort the response. Default is 'domain_rank'. See API docs for more details.

`order`	(optional) The order of sorting. Default is 'desc' for descending order. See API docs for more details.
`filter`	(optional) The nested list of filtering options. See API docs for more details.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns aggregated backlinks data for each referring domain.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per each domain in response.

Sorting

You can sort the response using `sort` argument. The sorting order is defined using `order` argument.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_bl_referring_domains(
  api_token        = api_token,
  domain           = 'serpstat.com',
  page             = 1,
  size             = 100,
  sort             = 'domain_rank',
  order            = 'desc',
  filter           = NULL,
  return_method    = 'list'
)$data

## End(Not run)
```

`sst_call_api_method` *Make a request to Serpstat API endpoint*

Description

Make a request to Serpstat API endpoint

Usage

```
sst_call_api_method(api_token, api_method, api_params = NULL)
```

Arguments

api_token	Serpstat API token from the profile page.
api_method	Internal name of API method.
api_params	A list of API parameters used by api_method. More information about parameters in the official docs.

Value

The list with a response data.

Examples

```
api_params <- list(
  query = 'serpstat.com',
  page  = 1,
  size  = 5
  )
tryCatch({
  serpstatr:::sst_call_api_method(
    api_token  = Sys.getenv('SERPSTAT_API_TOKEN'),
    api_method = 'SerpstatLimitsProcedure.getStats',
    api_params = api_params
    )
})
```

---

sst_lists_to_df *Convert list of lists to data.frame*

Description

API response might contain nested lists with different number of elements. This function fills missing elements and combine lists to a data.frame.

Usage

```
sst_lists_to_df(lists, fill = NA)
```

Arguments

lists	- a list of nested lists with different number of elements
fill	- a value to fill missing values in lists

Value

A data.frame with all missing values filed with specified value.

Examples

```
sst_lists_to_df(
  lists = list(
    first_list  = list(a = 1, b = 2),
    second_list = list(a = 2, c = 3)
  ),
  fill  = 'empty'
)
```

`sst_pm_create_project` *Create a new project*

Description

Creates a new project in Serpstat.

Usage

```
sst_pm_create_project(api_token, domain, name, groups = NULL)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`domain`	(required) Domain to get data for.
`name`	(required) The name of the project. Can be different from the domain.
`groups`	(optional) A list of project groups the project should be added to.

Value

Returns the project ID for the created project.

API docs

Check all the values for request and response fields here.

API credits consumption

1 project credit.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_pm_create_project(
  api_token = api_token,
  domain    = 'serpstat.com',
  name      = 'Serpstat'
  )$data$project_id
```

```
## End(Not run)
```

`sst_pm_delete_project` *Delete the existing project*

Description

Deletes the existing project in Serpstat by project ID.

Usage

```
sst_pm_delete_project(api_token, project_id = NULL)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`project_id`	(required) ID of the project in Serpstat.

Value

Returns the state of the deletion operation.

API docs

Check all the values for request and response fields here.

API credits consumption

returns 1 project credit.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_pm_delete_project(
  api_token  = api_token,
  project_id = 12345
  )

## End(Not run)
```

`sst_pm_list_projects` *List existing projects*

Description

Gets a list of existing projects available for the users with their basic information.

Usage

```
sst_pm_list_projects(api_token, page = 1, size = 100, return_method = "list")
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`page`	(optional) Response page number if there are many pages in response.
`size`	(optional) Response page size.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns basic information on all the projects.

API docs

Check all the values for request and response fields here.

API credits consumption

0.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_pm_list_projects(
  api_token  = api_token,
  page       = 2,
  size       = 10
  )

## End(Not run)
```

sst_return_check *Preprocess the API response*

Description

Every API call returns a JSON object. This object is transformed to a list. Depending on return_method parameter the data element of this list will be a list or data.frame.

Usage

```
sst_return_check(response_content, return_method)
```

Arguments

response_content	The result of sst_call_api_method call.
return_method	Accepted values are 'list' to return data object as list or 'df' to return data object as data.frame.

Value

response_content with a data object as list or data.frame.

sst_rt_competitors *Get the data on competitors in search results*

Description

This method method returns the competing in top20 search results domains that rank for at least two keywords that are added the project.

Usage

```
sst_rt_competitors(
  api_token,
  project_id,
  region_id,
  date_from = Sys.Date() - 8,
  date_to = Sys.Date() - 1,
  domains = NULL,
  sort = "sum_traffic",
  sort_range = "top1",
  order = "desc",
  page = 1,
  size = 100,
  return_method = "list"
)
```

Arguments

api_token	(required) Serpstat API token from your profile.
project_id	(required) The ID of your project in Serpstat. You can find this ID in the URL of any rank tracker report. As an example, in https://serpstat.com/rank-tracker/keywords/12345/positions?get_params the ID would be 12345.
region_id	(required) The ID of a region returned by sst_rt_project_regions.
date_from	(optional) The date string in 'YYYY-MM-DD' format to specify the initial date of retrieved data. Default value is current date minus 8 days.
date_to	(optional) The date string in 'YYYY-MM-DD' format to specify the final date of retrieved data. Must not exceed date_from + 30 days. Default value is yesterday.
domains	(optional) A vector of domain names for which the data should be retrieved. By default the data is retrieved for all domains that rank for at least two keywords that are added to the project.
sort	(optional) Must be one of 'sum_traffic' (default, domain search traffic distribution), 'keywords_count' (number of keywords), 'avg_position' (average domain position), 'position_ranges' (ranges of positions), 'ads_count' (number of ads in search engine results).
sort_range	(optional) The subcategory of ranges of positions to sort by. Must be one of 'top1', 'top3', 'top5', 'top10', 'top20', 'top101' to sort by the number of keywords in specific positions range; or 'keywords_count_top', 'keywords_count_bottom' to sort by the number of keywords in search ads blocks; or 'avg_position_top', 'avg_position_bottom' to sort by the average position in search ads blocks.
order	(optional) The sorting order. Must be one of string 'desc' (default) for descending sorting or 'asc' for ascending sorting.
page	(optional) Response page number if there are many pages in response. The default value is 1.
size	(optional) Response page size. Must be one of 20, 50, 100, 200, 500.The default value is 100.
return_method	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns traffic and keywords distributions, average positions of the domains by date.

API docs

Check all the values for request and response fields here.

API credits consumption

0

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
project_id <- 12345
region_id  <- sst_rt_competitors(
  api_token  = api_token,
  project_id = project_id
  )$data$regions[[1]]$id
sst_rt_competitors(
  api_token     = api_token,
  project_id    = project_id,
  region_id     = region_id,
  date_from     = '2020-12-01',
  date_to       = '2020-12-30',
  domains       = c('serpstat.com', 'serpstatbot.com'),
  sort          = 'sum_traffic',
  sort_range    = 'top1',
  order         = 'desc',
  page          = 1,
  size          = 20,
  return_method = 'list'
)

## End(Not run)
```

`sst_rt_positions_history` *Get ranking history for the domain or URL in selected search region*

Description

This method method returns the rankings for specified domain is selected search region.

Usage

```
sst_rt_positions_history(
  api_token,
  project_id,
  region_id,
  date_from = Sys.Date() - 8,
  date_to = Sys.Date() - 1,
  keywords = NULL,
  url = NULL,
  sort = "keyword",
  order = "desc",
  page = 1,
  size = 100,
  return_method = "list"
)
```

Arguments

api_token	(required) Serpstat API token from your profile.
project_id	(required) The ID of your project in Serpstat. You can find this ID in the URL of any rank tracker report. As an example, in https://serpstat.com/rank-tracker/keywords/12345/positions?get_params the ID would be 12345.
region_id	(required) The ID of a region returned by sst_rt_project_regions.
date_from	(optional) The date string in 'YYYY-MM-DD' format to specify the initial date of retrieved data. Default value is current date minus 8 days.
date_to	(optional) The date string in 'YYYY-MM-DD' format to specify the final date of retrieved data. Must not exceed date_from + 30 days. Default is yesterday.
keywords	(optional) A vector of keywords for witch the data should be retrieved. Maximum 1000 keywords per request. By default all the data for all keywords in the project is returned.
url	(optional) The domain name (e.g. domain.com) or web page address (e.g. https://domain.com/page) to get the data for. By default the results are returned for the projects' domain name.
sort	(optional) Must be one of 'keyword' (default) to sort the results alphabetically or 'date' to sort the results by date.
order	(optional) The sorting order. Must be one of string 'desc' (default) for descending sorting or 'asc' for ascending sorting.
page	(optional) Response page number if there are many pages in response. The default value is 1.
size	(optional) Response page size. Must be one of 20, 50, 100, 200, 500.The default value is 100.
return_method	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns positions of selected domain in search engine results in selected region with corresponding URLs for these positions.

API docs

Check all the values for request and response fields here.

API credits consumption

0

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
project_id <- 12345
region_id  <- sst_rt_project_regions(
```

```
  api_token  = api_token,
  project_id = project_id
  )$data$regions[[1]]$id
sst_rt_positions_history(
  api_token     = api_token,
  project_id    = project_id,
  region_id     = region_id,
  date_from     = '2020-12-01',
  date_to       = '2020-12-30',
  keywords      = c('seo', 'ppc', 'serpstat'),
  url           = 'serpstat.com',
  sort          = 'keyword',
  order         = 'desc',
  page          = 1,
  size          = 100,
  return_method = 'list'
)

## End(Not run)
```

`sst_rt_project_regions` *Get all regions for the project*

Description

In Serpstat you are able to track ranking of your website in multiple regions. This method returns all the regions in your Serpstat project. You will need the results of this method to get rankings in selected region.

Usage

```
sst_rt_project_regions(api_token, project_id, return_method = "list")
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`project_id`	(required) The ID of your project in Serpstat. You can find this ID in the URL of any rank tracker report. As an example, in https://serpstat.com/rank-tracker/keywords/12345/positions?get_params the ID would be 12345.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns the regions of the project with their ID, state (active or not) and other region attributes.

API docs

Check all the values for request and response fields here.

API credits consumption

0

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
project_id <- 12345
sst_rt_project_regions(api_token = api_token, project_id = project_id)

## End(Not run)
```

---

sst_rt_serp_history *Get search results history in search region by keywords*

Description

This method returns top 100 search results in Google.

Usage

```
sst_rt_serp_history(
  api_token,
  project_id,
  region_id,
  date_from = Sys.Date() - 8,
  date_to = Sys.Date() - 1,
  keywords = NULL,
  sort = "keyword",
  order = "desc",
  page = 1,
  size = 100,
  return_method = "list"
)
```

Arguments

api_token	(required) Serpstat API token from your profile.
project_id	(required) The ID of your project in Serpstat. You can find this ID in the URL of any rank tracker report. As an example, in https://serpstat.com/rank-tracker/keywords/12345/positions?get_params the ID would be 12345.
region_id	(required) The ID of a region returned by sst_rt_project_regions.

date_from	(optional) The date string in 'YYYY-MM-DD' format to specify the initial date of retrieved data. Default value is current date minus 8 days.
date_to	(optional) The date string in 'YYYY-MM-DD' format to specify the final date of retrieved data. Must not exceed date_from + 30 days. Default is yesterday.
keywords	(optional) A vector of keywords for witch the data should be retrieved. Maximum 1000 keywords per request. By default all the data for all keywords in the project is returned.
sort	(optional) Must be one of 'keyword' (default) to sort the results alphabetically or 'date' to sort the results by date.
order	(optional) The sorting order. Must be one of string 'desc' (default) for descending sorting or 'asc' for ascending sorting.
page	(optional) Response page number if there are many pages in response. The default value is 1.
size	(optional) Response page size. Must be one of 20, 50, 100, 200, 500.The default value is 100.
return_method	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns the search engine results for specific dates and region including positions and URLs.

API docs

Check all the values for request and response fields here.

API credits consumption

0

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
project_id <- 12345
region_id  <- sst_rt_project_regions(
  api_token  = api_token,
  project_id = project_id
  )$data$regions[[1]]$id
sst_rt_serp_history(
  api_token     = api_token,
  project_id    = project_id,
  region_id     = region_id,
  date_from     = '2020-12-01',
  date_to       = '2020-12-30',
  keywords      = c('seo', 'ppc', 'serpstat'),
  sort          = 'keyword',
  order         = 'desc',
  page          = 1,
```

```
  size          = 100,
  return_method = 'list'
)

## End(Not run)
```

`sst_sa_database_info` *List all Serpstat databases*

Description

In every request to get data from search analytics API you must set se parameter to specify from what country do you want to get the data. This method returns all acceptable values for se parameter with corresponding country names.

Usage

```
sst_sa_database_info(api_token, return_method = "list")
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns country name, se parameter value and local search engine domain name for each country.

API docs

Check all the values for request and response fields here.

API credits consumption

0

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_database_info(api_token)$data

## End(Not run)
```

`sst_sa_domains_info` *Domains summary*

Description

Returns the number of keywords for each domain in SEO and PPC, online visibility and other metrics.

Usage

```
sst_sa_domains_info(
  api_token,
  domains,
  se,
  sort = NULL,
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`domains`	(required) A vector of domain names to analyze.
`se`	(required) Search engine alias (db_name) returned by `sst_sa_database_info`
`sort`	(optional) A field to sort the response. See Sorting for more details.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns aggregated stats for each domain.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per domain in request.

Sorting

You can sort the response using `sort` argument. It must be a list with a single named element. The name of the element must match one of parameters in response. The value of the element must be `asc` for ascending order and `desc` for descending order. For example, `sort = list(ads = 'desc')` would sort the response by `ads` parameter in descending order.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_domains_info(
  api_token     = api_token,
  domains       = c('amazon.com', 'ebay.com'),
  se            = 'g_us',
  return_method = 'df'
)$data

## End(Not run)
```

`sst_sa_domain_history` *Domain history*

Description

Returns historical metrics for the domain with about two weeks between measurements.

Usage

```
sst_sa_domain_history(
  api_token,
  domain,
  se,
  sort = NULL,
  filters = NULL,
  page = 1,
  size = 100,
  during_all_time = TRUE,
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`domain`	(required) Domain to get data for.
`se`	(required) Search engine alias (db_name) returned by `sst_sa_database_info`.
`sort`	(optional) A field to sort the response. See Sorting for more details.
`filters`	(optional) A list of filtering options. See Filtering for more details.
`page`	(optional) Response page number if there are many pages in response.
`size`	(optional) Response page size.
`during_all_time`	(optional) TRUE (default) for all the history, FALSE for year-to-date data.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns a number of metrics for each date for the domain.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per date in the response.

Sorting

You can sort the response using `sort` argument. It must be a list with a single named element. The name of the element must match one of parameters in response. The value of the element must be `asc` for ascending order and `desc` for descending order. For example, `sort = list(ads = 'desc')` would sort the response by `ads` parameter in descending order.

Filtering

To filter the results you can use `filters` argument. It must be a list of named elements. The name of the element must match one of the filtering parameters. See API docs for more details. For example, `filters = list(queries_from = 0, queries_to = 10)` would narrow the results to include only the keywords that have a search volume between 0 and 10.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_domain_history(
  api_token     = api_token,
  domain        = 'serpstat.com',
  se            = 'g_us',
  sort          = list(date = 'desc'),
  filters       = list(traff_from = 20000),
  page          = 2,
  size          = 10,
  return_method = 'df'
)$data

## End(Not run)
```

`sst_sa_domain_keywords` *Domain organic keywords*

Description

Returns up to 60 000 organic keywords from selected region for the domain with a number of metrics for each keyword.

Usage

```
sst_sa_domain_keywords(
  api_token,
  domain,
  se,
  url = NULL,
  keywords = NULL,
  minusKeywords = NULL,
  sort = NULL,
  filters = NULL,
  page = 1,
  size = 100,
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`domain`	(required) Domain to get data for.
`se`	(required) Search engine alias (db_name) returned by `sst_sa_database_info`.
`url`	(optional) Get the results for this URL only.
`keywords`	(optional) A vector of words. Keywords in response will contain these words
`minusKeywords`	(optional) A vector of words. Keywords in response will not contain these words.
`sort`	(optional) A field to sort the response. See Sorting for more details.
`filters`	(optional) A list of filtering options. See Filtering for more details.
`page`	(optional) Response page number if there are many pages in response.
`size`	(optional) Response page size.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns a number of metrics for each keyword.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per keyword in response.

Sorting

You can sort the response using `sort` argument. It must be a list with a single named element. The name of the element must match one of parameters in response. The value of the element must be `asc` for ascending order and `desc` for descending order. For example, `sort = list(ads = 'desc')` would sort the response by `ads` parameter in descending order.

Filtering

To filter the results you can use `filters` argument. It must be a list of named elements. The name of the element must match one of the filtering parameters. See API docs for more details. For example, `filters = list(queries_from = 0, queries_to = 10)` would narrow the results to include only the keywords that have a search volume between 0 and 10.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_domain_keywords(
  api_token     = api_token,
  domain        = 'serpstat.com',
  se            = 'g_us',
  sort          = list(keyword_length = 'desc'),
  url           = 'https://serpstat.com/',
  keywords      = list('google'),
  minusKeywords = list('download'),
  filters       = list(queries_from = 0,
                       queries_to   = 10),
  page          = 2,
  size          = 10,
  return_method = 'df'
)$data

## End(Not run)
```

---

sst_sa_domain_organic_competitors

Domain competitors in organic search

Description

Returns organic competitors for the domain with their key metrics.

Usage

```
sst_sa_domain_organic_competitors(
  api_token,
  domain,
  se,
  sort = NULL,
  page = 1,
  size = 100,
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`domain`	(required) Domain to get data for.
`se`	(required) Search engine alias (db_name) returned by `sst_sa_database_info`.
`sort`	(optional) A field to sort the response. See Sorting for more details.
`page`	(optional) Response page number if there are many pages in response.
`size`	(optional) Response page size.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns a number of metrics for each organic competitor.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per returned page.

Sorting

You can sort the response using `sort` argument. It must be a list with a single named element. The name of the element must match one of parameters in response. The value of the element must be `asc` for ascending order and `desc` for descending order. For example, `sort = list(ads = 'desc')` would sort the response by `ads` parameter in descending order.

Filtering

To filter the results you can use `filters` argument. It must be a list of named elements. The name of the element must match one of the filtering parameters. See API docs for more details. For example, `filters = list(queries_from = 0, queries_to = 10)` would narrow the results to include only the keywords that have a search volume between 0 and 10.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_domain_organic_competitors(
  api_token     = api_token,
  domain        = 'serpstat.com',
  se            = 'g_us',
  sort          = list(relevance = 'desc'),
  page          = 2,
  size          = 20,
  return_method = 'df'
)$data

## End(Not run)
```

`sst_sa_domain_top_pages`

Domain top pages

Description

Returns the number of domain pages with the biggest potential traffic, number of keywords, and Facebook shares.

Usage

```
sst_sa_domain_top_pages(
  api_token,
  domain,
  se,
  sort = NULL,
  filters = NULL,
  page = 1,
  size = 100,
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`domain`	(required) Domain to get data for.
`se`	(required) Search engine alias (db_name) returned by `sst_sa_database_info`.
`sort`	(optional) A field to sort the response. See Sorting for more details.
`filters`	(optional) A list of filtering options. See Filtering for more details.
`page`	(optional) Response page number if there are many pages in response.
`size`	(optional) Response page size.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns domain top pages with their metrics.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per returned page.

Sorting

You can sort the response using `sort` argument. It must be a list with a single named element. The name of the element must match one of parameters in response. The value of the element must be `asc` for ascending order and `desc` for descending order. For example, `sort = list(ads = 'desc')` would sort the response by `ads` parameter in descending order.

Filtering

To filter the results you can use `filters` argument. It must be a list of named elements. The name of the element must match one of the filtering parameters. See API docs for more details. For example, `filters = list(queries_from = 0, queries_to = 10)` would narrow the results to include only the keywords that have a search volume between 0 and 10.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_domain_top_pages(
  api_token     = api_token,
  domain        = 'serpstat.com',
  se            = 'g_us',
  sort          = list(organic_keywords = 'desc'),
  filters       = list(url_contain = 'blog'),
  page          = 2,
  size          = 50,
  return_method = 'df'
)$data

## End(Not run)
```

`sst_sa_keywords` *Phrase match keywords*

Description

A full-text search to find all the keywords that match the queried term with a number of metrics for each keyword like search volume, CPC and competition level.

Usage

```
sst_sa_keywords(
  api_token,
  keyword,
  se,
  minusKeywords = NULL,
  sort = NULL,
  filters = NULL,
  page = 1,
  size = 100,
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`keyword`	(required) A keyword to search for.
`se`	(required) Search engine alias (db_name) returned by `sst_sa_database_info`.
`minusKeywords`	(optional) A vector of words. Keywords in response will not contain these words.
`sort`	(optional) A field to sort the response. See Sorting for more details.
`filters`	(optional) A list of filtering options. See Filtering for more details.
`page`	(optional) Response page number if there are many pages in response.
`size`	(optional) Response page size.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns a number of metrics for each keyword.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per keyword in response.

Sorting

You can sort the response using `sort` argument. It must be a list with a single named element. The name of the element must match one of parameters in response. The value of the element must be `asc` for ascending order and `desc` for descending order. For example, `sort = list(ads = 'desc')` would sort the response by `ads` parameter in descending order.

Filtering

To filter the results you can use `filters` argument. It must be a list of named elements. The name of the element must match one of the filtering parameters. See API docs for more details. For example, `filters = list(queries_from = 0, queries_to = 10)` would narrow the results to include only the keywords that have a search volume between 0 and 10.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_keywords(
  api_token     = api_token,
  keyword       = 'serpstat',
  se            = 'g_us',
  minusKeywords = c('free'),
  sort          = list(keyword_length = 'asc'),
  page          = 2,
  size          = 10,
  return_method = 'df'
)$data

## End(Not run)
```

---

sst_sa_keywords_info *Keywords summary*

Description

Returns a number of metrics for each keyword like search volume, CPC and competition level.

Usage

```
sst_sa_keywords_info(
  api_token,
  keywords,
  se,
  sort = NULL,
  return_method = "list"
)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`keywords`	(required) A vector of keywords to analyze.
`se`	(required) Search engine alias (db_name) returned by `sst_sa_database_info`.
`sort`	(optional) A field to sort the response. See Sorting for more details.
`return_method`	(optional) Accepted values are 'list' (default) to return data object as list or 'df' to return data object as data.frame.

Value

Returns a number of metrics for each keyword.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per keyword in request.

Sorting

You can sort the response using `sort` argument. It must be a list with a single named element. The name of the element must match one of parameters in response. The value of the element must be `asc` for ascending order and `desc` for descending order. For example, `sort = list(ads = 'desc')` would sort the response by `ads` parameter in descending order.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_keywords_info(
  api_token     = api_token,
  keywords      = c('seo', 'ppc', 'serpstat'),
  se            = 'g_us',
  sort          = list(cost = 'asc'),
  return_method = 'df'
)$data

## End(Not run)
```

`sst_sa_keyword_top` *Top for a keyword*

Description

Returns a list of results (URLs) from search engine results page (SERP) including organic results, paid results and different types of SERP features.

Usage

```
sst_sa_keyword_top(api_token, keyword, se, top_size = 100)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.
`keyword`	(required) A keyword to search for.
`se`	(required) Search engine alias (db_name) returned by `sst_sa_database_info`.
`top_size`	(optional) Set the number of URLs to get in response.

Value

Returns a list with the data about search engine results page for the keyword.

API docs

Check all the values for request and response fields here.

API credits consumption

1 per URL in response.

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_keyword_top(
  api_token = api_token,
  keyword   = 'serpstat',
  se        = 'g_us',
  top_size  = 10
)

## End(Not run)
```

`sst_sa_stats` *Get the number of API rows left*

Description

With most API request you spend some amount of API rows. The total amount of API rows available for you is based on your plan. Use this method to control the amount of API rows left.

Usage

```
sst_sa_stats(api_token)
```

Arguments

`api_token`	(required) Serpstat API token from your profile.

Value

Returns a number of API rows left. Also returns some additional information about user and Serpstat plugin limits.

API credits consumption

0

Examples

```
## Not run:
api_token <- Sys.getenv('SERPSTAT_API_TOKEN')
sst_sa_stats(api_token)$summary_info$left_lines

## End(Not run)
```

Index