Package ‘aws.alexa’

October 12, 2022

Title  Client for the Amazon Alexa Web Information Services API

Version  0.1.8

Description  Use the Amazon Alexa Web Information Services API to find information about domains, including the kind of content that they carry, how popular are they---rank and traffic history, sites linking to them, among other things. See <https://aws.amazon.com/awis/> for more information.

Imports  httr, aws.signature, xml2, dplyr
Suggests  testthat, rmarkdown, knitr (>= 1.11), lintr
VignetteBuilder  knitr
License  MIT + file LICENSE
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.1
NeedsCompilation  no
Author  Gaurav Sood [aut, cre], Thomas Leeper [ctb]
Maintainer  Gaurav Sood <gsood07@gmail.com>
Repository  CRAN
Date/Publication  2020-11-10 06:10:02 UTC

R topics documented:

aws.alexa-package  .......................................................... 2
alexa_check  ................................................................. 2
alexa_GET  ................................................................. 3
alexa_PROCESS  ................................................................ 4
browse_categories  ................................................................ 4
category_listing  .............................................................. 5
in_links  ........................................................................... 6
set_secret_key  ................................................................. 7
traffic_history  ................................................................. 7
url_info  ............................................................................ 8
aws.alexa-package  

aws.alexa: R Client for the Alexa Web Information Services API

Description

Find information about domains, including the kind of content that they carry, how popular are they, sites linking to them, among other things. The package provides access to the Alexa Web Information Service API: https://docs.aws.amazon.com/AlexaWebInfoService/latest/.

To learn how to use aws.alexa, see this vignette: https://CRAN.R-project.org/package=aws.alexa/vignettes/overview.html.

You need to get credentials (Access Key ID and Secret Access Key) to use this application. If you haven’t already, get these at https://aws.amazon.com/. And set these using `set_secret_key`.

Author(s)

Gaurav Sood

---

alexa_check

Request Response Verification

Description

Request Response Verification

Usage

alexa_check(req)

Arguments

req  request

Value

in case of failure, a message
Description

GET

Usage

```r
alexa_GET(
  query,
  key = Sys.getenv("AWS_ACCESS_KEY_ID"),
  secret = Sys.getenv("AWS_SECRET_ACCESS_KEY"),
  verbose =getOption("verbose", FALSE),
  session_token = NULL,
  region = "us-west-1",
  headers = list(),
  ...
)
```

Arguments

- **query**: query list
- **key**: A character string containing an AWS Access Key ID. The default is retrieved from `Sys.getenv("AWS_ACCESS_KEY_ID")`.
- **secret**: A character string containing an AWS Secret Access Key. The default is retrieved from `Sys.getenv("AWS_SECRET_ACCESS_KEY")`.
- **verbose**: A logical indicating whether to be verbose. Default is given by `options("verbose")`.
- **session_token**: Optionally, a character string containing an AWS temporary Session Token. If missing, defaults to value stored in environment variable `AWS_SESSION_TOKEN`.
- **region**: A character string containing the AWS region. If missing, defaults to "us-west-1".
- **headers**: A list of request headers for the REST call.
- **...**: Additional arguments passed to `GET`.

Value

- **list**
browse_categories  

**Postprocess the results a bit**

**Description**

Postprocess the results a bit

**Usage**

```r
alexa_PROCESS(res)
```

**Arguments**

- `res` : result

**Value**

display request ID and Response Status and the first member of the list

---

browse_categories  

**Browse Categories**

**Description**

Uses data from dmoz.org, which is no longer updated.

**Usage**

```r
browse_categories(
    path = NULL,
    response_group = "Categories",
    description = TRUE,
    ...
)
```

**Arguments**

- `path` : String; Required; valid category path
- `response_group` : String; Required; One of the following: Categories, RelatedCategories, LanguageCategories, LetterBars
- `description` : Boolean; Optional; Whether or not to return descriptions of categories; Default is TRUE
- `...` : Additional arguments passed to `alexa_GET`. 
**category_listing**

### Value

data.frame with 5 columns: path, title, sub_category_count, total_listing_count, description

### References


### Examples

```r
## Not run:
browse_categories(path="Top/Arts")

## End(Not run)
```

---

**category_listing**

**Category Listing**

### Description

Uses data from dmoz.org, which is no longer updated. For any given category, it returns a list of site listings contained within that category.

### Usage

```r
category_listing(
  path = NULL,
  sort_by = "Popularity",
  recursive = TRUE,
  start = 0,
  count = 20,
  description = TRUE,
  ...
)
```

### Arguments

- **path**
  - String; Required; valid category path
- **sort_by**
  - sort results by Popularity, Title, or AverageReview
- **recursive**
  - Boolean; Whether to return listings for the current category only, or for the current category plus all subcategories; Default is TRUE
- **start**
  - index of result at which to start; default is 0
- **count**
  - Number of results to return for this request; Max = 20; Default = 20
- **description**
  - Boolean; Optional; Whether or not to return descriptions of categories; Default is TRUE
- **...**
  - Additional arguments passed to `alexa_GET`. 
Value

data.frame

References

https://docs.aws.amazon.com/AlexaWebInfoService/latest/ApiReference_CategoryListingsAction.html

Examples

## Not run:
category_listing(path="Top/Arts")

## End(Not run)

in_links

Sites linking to the site

Description

Sites linking to the site

Usage

in_links(url = NULL, start = 0, count = 20, ...)

Arguments

url String; Required; valid url
start index of result at which to start; default = 0
count Number of results to return for this request; Max = 20; Default = 20
... Additional arguments passed to alexa_GET.

Value

data.frame with two columns: title (site hostname) and url (specific url)

References

https://docs.aws.amazon.com/AlexaWebInfoService/latest/ApiReference_SitesLinkingInAction.html

Examples

## Not run:
in_links(url = "google.com")

## End(Not run)
set_secret_key

Set up Key and Secret

Description

Get the Access Key ID and Secret Access Key by logging into https://console.aws.amazon.com/, clicking on the username followed by security credentials. The function sets two environmental variables AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY. These environment variables persist within a R session. The function looks for these variables.

Usage

set_secret_key(key = NULL, secret = NULL, force = FALSE)

Arguments

key String; Required; Access Key ID
secret String; Required; Secret Access Key
force String; Required; Force change the AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY stored in the environment

References

https://aws.amazon.com/

Examples

## Not run:
set_secret_key(key = "key", secret = "secret")
## End(Not run)

traffic_history

Get Traffic History of a URL

Description

Get Traffic History of a URL

Usage

traffic_history(url = NULL, range = 31, start = NULL, ...)
url_info

Get Information about a URL

Arguments

url  String; Required; valid url
range  Integer; Required; Default is 31, Maximum is 31. Pick an integer between 1 and 31.
start  String; Optional; A date within the last 4 years in format YYYYMMDD.
...  Additional arguments passed to alexa_GET.

Value

data.frame with the following columns: site, start, range, date, page_views_per_million, page_views_per_user, rank, reach_per_million

References

https://docs.aws.amazon.com/AlexaWebInfoService/latest/ApiReference_TrafficHistoryAction.html

Examples

## Not run:
traffic_history(url = "http://www.google.com", start = "20160505")
## End(Not run)
url_info

References

https://docs.aws.amazon.com/AlexaWebInfoService/latest/ApiReference_UrlInfoAction.html

Examples

## Not run:
url_info(url = "http://www.google.com")

## End(Not run)
Index

alexa_check, 2
alexa_GET, 3, 4–6, 8
alexa_PROCESS, 4
aws.alexa (aws.alexa-package), 2
aws.alexa-package, 2
browse_categories, 4
category_listing, 5
GET, 3
in_links, 6
set_secret_key, 2, 7
traffic_history, 7
url_info, 8