

# Package: aws.alexas (via r-universe)

August 21, 2024

**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 <gs00d07@gmail.com>

**Date/Publication** 2020-11-10 06:10:02 UTC

**Repository** https://soodoku.r-universe.dev

**RemoteUrl** https://github.com/cran/aws.alexas

**RemoteRef** HEAD

**RemoteSha** 44f796039e5a1737697c95a900fda9d9f916e0bb

## Contents

aws.alexas-package . . . . .	2
alexas_check . . . . .	2
alexas_GET . . . . .	3
alexas_PROCESS . . . . .	4
browse_categories . . . . .	4

category_listing . . . . .	5
in_links . . . . .	6
set_secret_key . . . . .	7
traffic_history . . . . .	7
url_info . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

aws.alexapackage	<i>aws.alexapackage: R Client for the Alexa Web Information Services API</i>
------------------	--

---

### 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.alexapackage, see this vignette: <https://CRAN.R-project.org/package=aws.alexapackage/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

---

alex_check	<i>Request Response Verification</i>
------------	--------------------------------------

---

### Description

Request Response Verification

### Usage

```
alex_check(req)
```

### Arguments

req	request
-----	---------

### Value

in case of failure, a message

alexa\_GET

*Base POST AND GET functions. Not exported.***Description**

GET

**Usage**

```

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 <code>Sys.getenv("AWS_ACCESS_KEY_ID")</code> .
secret	A character string containing an AWS Secret Access Key. The default is retrieved from <code>Sys.getenv("AWS_SECRET_ACCESS_KEY")</code> .
verbose	A logical indicating whether to be verbose. Default is given by <code>options("verbose")</code> .
session_token	Optionally, a character string containing an AWS temporary Session Token. If missing, defaults to value stored in environment variable <code>AWS_SESSION_TOKEN</code> .
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 <a href="#">GET</a> .

**Value**

list

---

alexa_PROCESS	<i>Postprocess the results a bit</i>
---------------	--------------------------------------

---

**Description**

Postprocess the results a bit

**Usage**

```
alexa_PROCESS(res)
```

**Arguments**

res	result
-----	--------

**Value**

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

---

browse_categories	<i>Browse Categories</i>
-------------------	--------------------------

---

**Description**

Uses data from dmoz.org, which is no longer updated.

**Usage**

```
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 <a href="#">alexa_GET</a> .

**Value**

data.frame with 5 columns: path, title, sub\_category\_count, total\_listing\_count, description

**References**

[https://docs.aws.amazon.com/AlexaWebInfoService/latest/index.html?ApiReference\\_CategoryBrowseAction.html](https://docs.aws.amazon.com/AlexaWebInfoService/latest/index.html?ApiReference_CategoryBrowseAction.html)

**Examples**

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

## End(Not run)
```

---

category_listing	<i>Category Listing</i>
------------------	-------------------------

---

**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**

```
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 <a href="#">alexa_GET</a> .

**Value**

data.frame

**References**

[https://docs.aws.amazon.com/AlexaWebInfoService/latest/ApiReference\\_CategoryListingsAction.html](https://docs.aws.amazon.com/AlexaWebInfoService/latest/ApiReference_CategoryListingsAction.html)

**Examples**

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

## End(Not run)
```

---

in_links	<i>Sites linking to the site</i>
----------	----------------------------------

---

**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 <a href="#">alexa_GET</a> .

**Value**

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

**References**

[https://docs.aws.amazon.com/AlexaWebInfoService/latest/ApiReference\\_SitesLinkingInAction.html](https://docs.aws.amazon.com/AlexaWebInfoService/latest/ApiReference_SitesLinkingInAction.html)

**Examples**

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

## End(Not run)
```

---

set_secret_key	<i>Set up Key and Secret</i>
----------------	------------------------------

---

**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 <code>AWS_ACCESS_KEY_ID</code> and <code>AWS_SECRET_ACCESS_KEY</code> stored in the environment

**References**

<https://aws.amazon.com/>

**Examples**

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

---

traffic_history	<i>Get Traffic History of a URL</i>
-----------------	-------------------------------------

---

**Description**

Get Traffic History of a URL

**Usage**

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

**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 <a href="#">alexa_GET</a> .

**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	<i>Get Information about a URL</i>
----------	------------------------------------

---

**Description**

Get Information about a URL

**Usage**

```
url_info(url = NULL, response_group = "SiteData", ...)
```

**Arguments**

url	String; Required; valid url
response_group	String; Required; One of the following: RelatedLinks, Categories, Rank, RankBy-Country, UsageStats, AdultContent, Speed, Language, OwnedDomains, LinksIn-Count, SiteData Default is 'SiteData'. Multiple fields can be passed. They must be separated by comma.
...	Additional arguments passed to <a href="#">alexa_GET</a> .

**Value**

named list



## **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