

Package ‘urlshorteneR’

December 5, 2016

Type Package

Title R Wrapper for the 'Bit.ly', 'Goo.gl' and 'Is.gd' URL Shortening Services

Description Allows using different URL shortening services, which also provide expanding and analytic functions. Specifically developed for 'Bit.ly', 'Goo.gl' (both OAuth2) and 'is.gd' (no API key). Others can be added by request.

Version 0.9.2

Date 2016-02-14

Maintainer John Malc <cincenko@outlook.com>

Depends R (>= 3.2.3)

Imports httr (>= 1.1.0), jsonlite (>= 0.9.19), stringr (>= 1.0.0)

Suggests roxygen2 (>= 5.0.1), knitr (>= 1.12), testthat (>= 0.11.0)

VignetteBuilder knitr

License Apache License 2.0

URL <https://github.com/dmpe/urlshorteneR>

BugReports <https://github.com/dmpe/urlshorteneR/issues>

RoxygenNote 5.0.1

NeedsCompilation no

Author John Malc [aut, cre] (@dmpe)

Repository CRAN

Date/Publication 2016-12-05 18:28:48

R topics documented:

bitly_IsProDomain	2
bitly_LinksExpand	3
bitly_LinksInfo	4
bitly_LinksLookup	5
bitly_LinksMetricsClicks	6

bitly_LinksMetricsCountries	7
bitly_LinksMetricsEncoders	8
bitly_LinksMetricsEncodersByCount	9
bitly_LinksMetricsEncodersCount	10
bitly_LinksMetricsReferrers	11
bitly_LinksMetricsReferrersByDomain	12
bitly_LinksMetricsReferringDomains	13
bitly_LinksShorten	14
bitly_UserInfo	15
bitly_UserLinkHistory	17
bitly_UserMetricsClicks	18
bitly_UserMetricsCountries	19
bitly_UserMetricsPopularLinks	20
bitly_UserMetricsReferrers	21
bitly_UserMetricsReferringDomains	22
bitly_UserMetricsShortenCounts	23
bitly_UserTrackingDomains	24
googl_auth	25
googl_LinksExpand	26
googl_LinksShorten	27
googl_UserLinkHistory	28
isgd_LinksExpand	29
isgd_LinksShorten	30

Index 31

bitly_IsProDomain	<i>Validate given domain for the PRO features</i>
-------------------	---

Description

Query whether a given domain is a valid bitly pro domain. Keep in mind that bitly custom short domains are restricted to less than 15 characters in length.

Usage

```
bitly_IsProDomain(domain, showRequestURL = FALSE)
```

Arguments

domain - A short domain. ie: nyti.ms.
 showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

bitly_pro_domain - 0 or 1 designating whether this is a current bitly domain.
 domain - an echo back of the request parameter.

See Also

See http://dev.bitly.com/domains.html#v3_bitly_pro_domain

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8", secret = "")
bitly_IsProDomain(domain = "nytidsfds.ms")
bitly_IsProDomain(domain = "nyti.ms", showRequestURL = TRUE)

## End(Not run)
```

bitly_LinksExpand	<i>Given a bitly URL or hash (or multiple), returns the target (long) URL.</i>
-------------------	--

Description

See http://dev.bitly.com/links.html#v3_expand

Usage

```
bitly_LinksExpand(hashIN = NULL, shortUrl = NULL, showRequestURL = FALSE)
```

Arguments

hashIN - refers to one bitly hashes, (e.g.: 2bYgqR or a-custom-name). Required

shortUrl - refers to one Bitlinks e.g.: <http://bit.ly/1RmnUT> or <http://j.mp/1RmnUT>. Optional.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

short_url - this is an echo back of the shortUrl request parameter.

hash - this is an echo back of the hash request parameter.

user_hash - the corresponding bitly user identifier.

global_hash - the corresponding bitly aggregate identifier.

error - indicates there was an error retrieving data for a given shortUrl or hash. An example error is "NOT_FOUND".

long_url - the URL that the requested short_url or hash points to.

TODO

or more URLs Up TO 15

Note

Either shortUrl or hash must be given as a parameter.

The maximum number of shortUrl and hash parameters is 15.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8",
secret = "b7e4abaf8b26ec4daa92b1e64502736f5cd78899")
bitly_LinksExpand(shortUrl = "http://bit.ly/DPetrov")
bitly_LinksExpand(hash = "DPetrov", showRequestURL = TRUE)
bitly_LinksExpand(hash = "DPetrov")
bitly_LinksExpand(shortUrl = "on.natgeo.com/1bEVhwE", hash = "1bEVhwE")

## manyHashes <- list("DPetrov", "1QU8CFm", "1R1LPSE", "1LNqqva")
## for (u in 1:length(manyHashes)) {
##   print(bitly_LinksExpand(hashIN = manyHashes[[u]], showRequestURL = TRUE))
## }

## End(Not run)
```

bitly_LinksInfo	<i>Used to return the page title for a given Bitlink.</i>
-----------------	---

Description

See http://dev.bitly.com/links.html#v3_info

Usage

```
bitly_LinksInfo(hashIN = NULL, shortUrl = NULL, expand_user = "true",
showRequestURL = FALSE)
```

Arguments

- | | |
|----------------|---|
| hashIN | - refers to one bitly hashes, (e.g.: 2bYgqR or a-custom-name). Required |
| shortUrl | - refers to one Bitlinks e.g.: http://bit.ly/1RmnUT or http://j.mp/1RmnUT . Optional. |
| expand_user | - optional true/false (default) - include extra user info in response. |
| showRequestURL | - show URL which has been build and requested from server. For debug purposes. |

Value

short_url - this is an echo back of the shortUrl request parameter.

hash - this is an echo back of the hash request parameter.

user_hash - the corresponding bitly user identifier.

global_hash - the corresponding bitly aggregate identifier.

error - indicates there was an error retrieving data for a given shortUrl or hash. An example error is "NOT_FOUND".

title - the HTML page title for the destination page (when available).

created_by - the bitly username that originally shortened this link, if the link is public. Otherwise, null.

created_at - the epoch timestamp when this Bitlink was created.

Note

Either shortUrl or hash must be given as a parameter (or both).

The maximum number of shortUrl and hash parameters is 15.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8",
secret = "b7e4abaf8b26ec4daa92b1e64502736f5cd78899")
bitly_LinksInfo(shortUrl = "http://bit.ly/DPetrov")
bitly_LinksInfo(hash = "DPetrov", showRequestURL = TRUE)
bitly_LinksInfo(hash = "DPetrov", expand_user = "true")

## hash is the one which is only returned. Dont use
bitly_LinksInfo(shortUrl = "on.natgeo.com/1bEVhwE", hash = "DPetrov")

## manyHashes <- list("DPetrov", "1QU8CFm", "1R1LPSE", "1LNqqva")
## for (u in 1:length(manyHashes)) {
##   print(bitly_LinksInfo(hashIN = manyHashes[[u]], showRequestURL = TRUE))
## }

## End(Not run)
```

bitly_LinksLookup

Query for a Bitlink based on a long URL.

Description

See http://dev.bitly.com/links.html#v3_link_lookup

Usage

```
bitly_LinksLookup(url, showRequestURL = FALSE)
```

Arguments

`url` - one long URLs to lookup.
`showRequestURL` - show URL which has been build and requested from server. For debug purposes.

Value

`url` - an echo back of the url parameter.
`aggregate_link` - the corresponding bitly aggregate link (global hash).

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8",
secret = "b7e4abaf8b26ec4daa92b1e64502736f5cd78899")
bitly_LinksLookup(url = "http://www.seznam.cz/")
bitly_LinksLookup(url = "http://www.seznam.cz/", showRequestURL = TRUE)

manyUrIs <- list("http://www.seznam.cz/", "http://www.seznamasdas.cz/",
"http://www.seznam.cz/asadasd", "http://www.seznam.cz/adqwrewtregt")
for (u in 1:length(manyUrIs)) {
  print(bitly_LinksLookup(url = manyUrIs[[u]], showRequestURL = TRUE))
}

## End(Not run)
```

```
bitly_LinksMetricsClicks
```

Returns numbers of click on a link.

Description

Returns the number of clicks on a single Bitlink of the authenticated user.

Usage

```
bitly_LinksMetricsClicks(link, limit = 1000, unit = c("minute", "hour",
"day", "week", "month"), units = -1, rollup = "true",
showRequestURL = FALSE)
```

Arguments

link	- a Bitlink.
limit	- 1 to 1000 (default=1000).
unit	- minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.
units	- an integer representing the time units to query data for. Pass -1 to return all units of time.
rollup	- true (default) or false. Return data for multiple units rolled up to a single result instead of a separate value for each period of time.
showRequestURL	- show URL which has been build and requested from server. For debug purposes.

Value

clicks - the number of clicks on the specified Bitlink.

dt - time in UTC format (only when rollup = "false")

See Also

http://dev.bitly.com/link_metrics.html#v3_link_clicks

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_LinksMetricsClicks(link = "http://bit.ly/DPetrov", unit = "day", units = -1, limit = 100)

## End(Not run)
```

bitly_LinksMetricsCountries

Returns metrics about the countries from a link.

Description

Returns metrics about the countries referring click traffic to a single Bitlink.

Usage

```
bitly_LinksMetricsCountries(link, limit = 1000, unit = c("minute", "hour",
  "day", "week", "month"), units = -1, showRequestURL = FALSE)
```

Arguments

link	- a Bitlink.
limit	- 1 to 1000 (default=1000).
unit	- minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.
units	- an integer representing the time units to query data for. Pass -1 to return all units of time.
showRequestURL	- show URL which has been build and requested from server. For debug purposes.

Value

clicks - the number of clicks referred from this country.

country - the two-letter code of the referring country.

See Also

http://dev.bitly.com/link_metrics.html#v3_link_countries

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_LinksMetricsCountries(link = "http://bit.ly/DPetrov", unit = "day", units = -1, limit = 100)
bitly_LinksMetricsCountries(link = "http://bit.ly/DPetrov", unit = "day", units = -1, limit = 100,
showRequestURL= TRUE)

## End(Not run)
```

bitly_LinksMetricsEncoders

Returns users who have encoded this long URL.

Description

Returns users who have encoded this long URL (optionally only those in the requesting user's social graph).

Usage

```
bitly_LinksMetricsEncoders(link, my_network = "false", limit = 25,
expand_user = "false", subaccounts = "false", showRequestURL = FALSE)
```


Arguments

link	- a Bitlink.
my_network	- true or false (default) restrict to my network.
limit	- 1 to 1000 (default=1000).
expand_user	- true or false (default) include display names of encoders.
subaccounts	- (only available to enterprise accounts) false (always default) restrict to this enterprise account and its subaccounts
showRequestURL	- show URL which has been build and requested from server. For debug purposes.

Value

entries - a mapping of link, user, and ts (when the Bitlink was created) and possible more depending on input parameters.

Note

Some users may not be returned from this call depending on Bitlink privacy settings.

See Also

http://dev.bitly.com/link_metrics.html#v3_link_encoders

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_LinksMetricsEncoders(link = "http://bit.ly/DPetrov")
bitly_LinksMetricsEncoders("http://bit.ly/DPetrov", expand_user = "true", my_network = "false")

## End(Not run)
```

bitly_LinksMetricsEncodersByCount

Returns users who have encoded this link.

Description

Returns users who have encoded this link (optionally only those in the requesting user's social graph), sorted by the number of clicks on each encoding user's link.

Usage

```
bitly_LinksMetricsEncodersByCount(link, limit = 100, my_network = "false",
  expand_user = "false", subaccounts = "false", showRequestURL = FALSE)
```

Arguments

link - a Bitlink.
 limit - 1 to 1000 (default=1000).
 my_network - true or false (default) restrict to my network
 expand_user - false (always default) include display names of encoders
 subaccounts - (only available to enterprise accounts) false (always default) restrict to this enterprise account and its subaccounts
 showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

entries - a mapping of link, user, and ts (when the Bitlink was created) and possible more depending on input parameters.

Note

The response will only contain users whose links have gotten at least one click, and will not contain any users whose links are private.

See Also

See http://dev.bitly.com/link_metrics.html#v3_link_encoders_by_count

Examples

```

## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_LinksMetricsEncodersByCount("http://bit.ly/DPetrov", my_network = "false", limit = 100)
bitly_LinksMetricsEncodersByCount("http://bit.ly/DPetrov", my_network = "false", limit = 100,
expand_user = "true")

## End(Not run)

```

bitly_LinksMetricsEncodersCount

Returns the number of users who have shortened a link.

Description

Returns the number of users who have shortened (encoded) a single Bitlink.

Usage

```
bitly_LinksMetricsEncodersCount(link, showRequestURL = FALSE)
```

Arguments

link - a Bitlink.
 showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

aggregate_link - the aggregate (global) Bitlink for the provided Bitlink.
 count - the number of bitly users who have shortened (encoded) this link.

See Also

http://dev.bitly.com/link_metrics.html#v3_link_encoders_count

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_LinksMetricsEncodersCount(link = "http://bit.ly/DPetrov")

## End(Not run)
```

bitly_LinksMetricsReferrers

Returns metrics about the pages referring click traffic to a single Bitlink.

Description

http://dev.bitly.com/link_metrics.html#v3_link_referrers

Usage

```
bitly_LinksMetricsReferrers(link, limit = 1000, unit = c("minute", "hour",
  "day", "week", "month"), units = -1, showRequestURL = FALSE)
```

Arguments

link - a Bitlink.
 limit - 1 to 1000 (default=1000).
 unit - minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.
 units - an integer representing the time units to query data for. Pass -1 to return all units of time.
 showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

clicks - the number of clicks referred from this domain.

referrer - the URL referring clicks.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_LinksMetricsReferrers(link = "http://bit.ly/DPetrov", unit = "day", units = -1, limit = 100)

## End(Not run)
```

bitly_LinksMetricsReferrersByDomain

Returns metrics about the pages referring click traffic to a single Bitlink, grouped by referring domain.

Description

http://dev.bitly.com/link_metrics.html#v3_link_referrers_by_domain

Usage

```
bitly_LinksMetricsReferrersByDomain(link, limit = 1000, unit = c("minute",
  "hour", "day", "week", "month"), units = -1, showRequestURL = FALSE)
```

Arguments

link - a Bitlink.

limit - 1 to 1000 (default=1000).

unit - minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.

units - an integer representing the time units to query data for. Pass -1 to return all units of time.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

clicks - the number of clicks referred from this domain.

referrer - the URL referring clicks.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_LinksMetricsReferrersByDomain("http://bit.ly/DPetrov",unit="day",units=-1,limit = 100)

## End(Not run)
```

bitly_LinksMetricsReferringDomains

Returns metrics about the domains referring click traffic to a link.

Description

Returns metrics about the domains referring click traffic to a single Bitlink.

Usage

```
bitly_LinksMetricsReferringDomains(link, limit = 1000, unit = c("minute",
  "hour", "day", "week", "month"), units = -1, showRequestURL = FALSE)
```

Arguments

link - a Bitlink.

limit - 1 to 1000 (default=1000).

unit - minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.

units - an integer representing the time units to query data for. Pass -1 to return all units of time.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

clicks - the number of clicks referred from this domain.

domain - the domain referring clicks.

url - the complete URL of the domain referring clicks.

See Also

http://dev.bitly.com/link_metrics.html#v3_link_referring_domains

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_LinksMetricsReferringDomains("http://bit.ly/DPetrov", unit = "day", units=-1, limit = 100)

## End(Not run)
```

bitly_LinksShorten *Given a long URL, returns a short Bit.ly link.*

Description

See http://dev.bitly.com/rate_limiting.html and http://dev.bitly.com/links.html#v3_shorten

Usage

```
bitly_LinksShorten(longUrl, domain = NULL, showRequestURL = FALSE)
```

Arguments

longUrl - a long URL to be shortened (example: <http://betaworks.com/>).

domain - (optional) the short domain to use; either bit.ly, j.mp, or bitly.com or a custom short domain. The default for this parameter is the short domain selected by each user in their bitly account settings. Passing a specific domain via this parameter will override the default settings.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

new_hash - designates if this is the first time this long_url was shortened by this user. The return value will equal 1 the first time a long_url is shortened. It will also then be added to the user history.

hash - a bitly identifier for long_url which is unique to the given account.

long_url - an echo back of the longUrl request parameter. This may not always be equal to the URL requested, as some URL normalization may occur (e.g., due to encoding differences, or case differences in the domain). This long_url will always be functionally identical the the request parameter.

global_hash - a bitly identifier for long_url which can be used to track aggregate stats across all Bitlinks that point to the same long_url.

url - the actual Bitlink that should be used, and is a unique value for the given Bitly account.

Note

Look in the vignette for bulk shortening of URLs. Each call of this function == 1 API call. Take that into consideration due to limits etc.

The bitly API does not support shortening more than one long URL with a single API call. Meaning 1 Long URL = 1 Function call.

Long URLs should be URL-encoded. You can not include a longUrl in the request that has &, ?, #, or other reserved parameters without first encoding it.

The default value for the domain parameter is selected by each user from within their bitly account settings at <https://bitly.com/a/settings/advanced>.

Long URLs should not contain spaces: any longUrl with spaces will be rejected. All spaces should be either percent encoded spaces are all indications of errors. Please remember to strip leading and trailing whitespace from any user input before shortening.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8",
secret = "b7e4abaf8b26ec4daa92b1e64502736f5cd78899")
bitly_LinksShorten(longUrl = "http://slovník.seznam.cz/")
bitly_LinksShorten(longUrl = "https://travis-ci.org/dmpe/rbitly/builds/68231423", domain = "j.mp")

## End(Not run)
```

bitly_UserInfo*Return or update information about a user.*

Description

See http://dev.bitly.com/user_info.html#v3_user_info

Usage

```
bitly_UserInfo(showRequestURL = FALSE)
```

Arguments

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

login - the specified bitly login or the login of the authenticated user.

profile_url - URL of user's profile page.

profile_image - URL of user's profile image.

member_since - Unix timestamp for the moment the user signed up.

full_name - (optional) the user's full name, if set.

display_name - (optional) the user's display name, if set.

share_accounts - (optional) a list of the share accounts (Twitter or Facebook) linked to the user's account.

NOTICE: Only included in requests for a user's own info.

apiKey - the user's bitly API key.

is_enterprise - 0 or 1 to indicate if this account is signed up for Bitly Brand Tools.

has_master - 0 or 1 to indicate if this account is a customer sub account.

custom_short_domain - A short domain registered with this account that can be used in place of bit.ly for shortening links.

tracking_domains - A list of domains configured for analytics tracking.

default_link_privacy - public or private indicating the default privacy setting for new links.

domain_preference_options - A list of the valid short domains that this account can choose as a default.

NOTICE: Only included for enterprise accounts (is_enterprise == 1 or has_master == 1).

sub_accounts - (optional) list of accounts associated with this account.

e2e_domains - (optional) list of domains associated with this custom_short_domain.

tracking_url_prefixes - A list of owned 3rd party urls such as Facebook tracked for analytics

master_account - (optional) the login of a master account, if this is associated with an enterprise account.

enterprise_permissions - (optional) list of enterprise permissions associated with this account.

btt_start_date - (optional) the date for when this account became a Bitly Brand Tools account.

Note

Both returned columns (!) are character type.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8", secret = "")
uI <- bitly_userInfo()

## End(Not run)
```

bitly_UserLinkHistory *Returns entries from a user's link history in reverse chronological order.*

Description

See http://dev.bitly.com/user_info.html#v3_user_link_history

Usage

```
bitly_UserLinkHistory(limit = 100, private = "off", archived = "both",  
    expand_client_id = "false", showRequestURL = FALSE)
```

Arguments

limit - optional integer in the range 1 to 100; default: 100, specifying the max number of results to return.

private - on, off and both (default) whether to include or exclude private history entries. (on = return only private history entries)

archived - on, off (default) or both whether to include or exclude archived history entries. (on = return only archived history entries)

expand_client_id - true or false (always default) whether to provide additional information about encoding application.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

link - the Bitlink specific to this user and this long_url.

aggregate_link - the global bitly identifier for this long_url.

long_url - the original long URL.

archived - a true/false value indicating whether the user has archived this link.

private - a true/false value indicating whether the user has made this link private.

created_at - an integer unix epoch indicating when this link was shortened/encoded.

user_ts - a user-provided timestamp for when this link was shortened/encoded, used for backfilling data.

modified_at - an integer unix epoch indicating when this link's metadata was last edited.

title - the title for this link.

note - the user-provided note, if set.

shares - a list of share actions (for the authenticated user only)

client_id - the oauth client ID of the app that shortened/saved this link on behalf of the user. If **expand_client_id** is set to false (only currently supported), this will be a string corresponding to the **client_id** of the encoding oauth application.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8", secret = "")
lh <- bitly_UserLinkHistory()

## End(Not run)
```

```
bitly_UserMetricsClicks
```

Returns the aggregate number of clicks on all of the authenticated user's Bitlinks.

Description

See http://dev.bitly.com/user_metrics.html#v3_user_clicks

Usage

```
bitly_UserMetricsClicks(limit = 1000, unit = c("minute", "hour", "day",
  "week", "month"), units = -1, rollup = c("false", "true"),
  showRequestURL = FALSE)
```

Arguments

limit	- 1 to 1000 (default=1000).
unit	- minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.
units	- an integer representing the time units to query data for. Pass -1 to return all units of time.
rollup	- true or false. Return data for multiple units rolled up to a single result instead of a separate value for each period of time.
showRequestURL	- show URL which has been build and requested from server. For debug purposes.

Value

dt - a unix timestamp representing the beginning of this unit.
 day_start - a unix timestamp representing the beginning of the specified day (ONLY returned if unit is not specified).
 clicks - the number of clicks on this user's links in the specified timeframe.

Note

without the parameter unit this endpoint returns a legacy response format which assumes rollup=false, unit=day and units=7.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_UserMetricsClicks(unit = "day", units = -1, limit = 100, rollup = "true")
bitly_UserMetricsClicks(unit = "day", units = -1, limit = 100, rollup = "false")

## End(Not run)
```

bitly_UserMetricsCountries

Returns aggregate metrics about the countries referring click traffic to all of the authenticated user's Bitlinks.

Description

See http://dev.bitly.com/user_metrics.html#v3_user_countries

Usage

```
bitly_UserMetricsCountries(limit = 1000, unit = c("minute", "hour", "day",
  "week", "month"), rollup = "true", units = -1, showRequestURL = FALSE)
```

Arguments

limit - 1 to 1000 (default=1000).

unit - minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.

rollup - true or false. Return data for multiple units rolled up to a single result instead of a separate value for each period of time.

units - an integer representing the time units to query data for. Pass -1 to return all units of time.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

clicks - the number of clicks referred from this country.
country - the two-letter code of the referring country.

Note

When a unit is specified (always the case), rollup is always (!) true.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_UserMetricsCountries(unit = "day", units = -1, limit = 100, rollup = "true")

## End(Not run)
```

bitly_UserMetricsPopularLinks

Returns the authenticated user's most-clicked Bitlinks (ordered by number of clicks) in a given time period.

Description

See http://dev.bitly.com/user_metrics.html#v3_user_popular_links

Usage

```
bitly_UserMetricsPopularLinks(limit = 1000, unit = c("minute", "hour",
  "day", "week", "month"), units = -1, showRequestURL = FALSE)
```

Arguments

limit - 1 to 1000 (default=1000).
unit - minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.
units - an integer representing the time units to query data for. Pass -1 to return all units of time.
showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

link - a Bitlink.
clicks - the number of clicks on that Bitlink in the specified timeframe.

Note

This has replaced the realtime_links endpoint.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_UserMetricsPopularLinks(unit = "day", units = -1, limit = 100)

## End(Not run)
```

bitly_UserMetricsReferrers

Returns aggregate metrics about the pages referring click traffic to all of the authenticated user's Bitlinks.

Description

See http://dev.bitly.com/user_metrics.html#v3_user_referrers

Usage

```
bitly_UserMetricsReferrers(limit = 1000, unit = c("minute", "hour", "day",
  "week", "month"), rollup = c("false", "true"), units = -1,
  showRequestURL = FALSE)
```

Arguments

limit - 1 to 1000 (default=1000).

unit - minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.

rollup - true or false. Return data for multiple units rolled up to a single result instead of a separate value for each period of time.

units - an integer representing the time units to query data for. Pass -1 to return all units of time.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

clicks - the number of clicks referred from this URL.
referrer - the URL referring clicks.

Note

When a unit is specified (always the case), rollup is always (!) true.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
user_Metrics_Referrers(unit = "day", units = -1, limit = 100, rollup = "true")

## End(Not run)
```

bitly_UserMetricsReferringDomains

Returns aggregate metrics about the domains referring click traffic to all of the authenticated user's Bitlinks.

Description

If the user is a master (ent.) account, or is a subaccount with full_reports permission, the user may choose to view the metrics of any account belonging to the master account.

Usage

```
bitly_UserMetricsReferringDomains(limit = 1000, unit = c("minute", "hour",
  "day", "week", "month"), rollup = c("false", "true"), units = -1,
  login = NULL, exclude_social_networks = c("true", "false"),
  showRequestURL = FALSE)
```

Arguments

limit	- 1 to 1000 (default=1000).
unit	- minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.
rollup	- true or false. Return data for multiple units rolled up to a single result instead of a separate value for each period of time.
units	- an integer representing the time units to query data for. Pass -1 to return all units of time.
login	- an optional string consisting of the account name used to report the appropriate statistics; defaults to the current user.
exclude_social_networks	- true (default) or false. If true, exclude domains that are part of a social network that bitly tracks.
showRequestURL	- show URL which has been build and requested from server. For debug purposes.

Value

clicks - the number of clicks referred from this URL.

referrer - the URL referring clicks.

Note

When a unit is specified (always the case), rollup is always (!) true.

See Also

See http://dev.bitly.com/user_metrics.html#v3_user_referring_domains

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_UserMetricsReferringDomains(unit = "day", units = -1, limit = 100, rollup = "true")
bitly_UserMetricsReferringDomains(unit = "day", units = -1, limit = 100, rollup = "false")
bitly_UserMetricsReferringDomains(unit = "day", units = -1, limit = 100,
exclude_social_networks = "false")
bitly_UserMetricsReferringDomains(unit = "day", units = -1, limit = 100,
exclude_social_networks = "true")

## End(Not run)
```

bitly_UserMetricsShortenCounts

Returns the number of Bitlinks created in a given time period by the authenticated user.

Description

See http://dev.bitly.com/user_metrics.html#v3_user_shorten_counts

Usage

```
bitly_UserMetricsShortenCounts(limit = 1000, unit = c("minute", "hour",
"day", "week", "month"), rollup = c("false", "true"), units = -1,
showRequestURL = FALSE)
```

Arguments

limit - 1 to 1000 (default=1000).

unit - minute, hour, day, week or month, default: day; Note: when unit is minute the maximum value for units is 60.

rollup - true or false. Return data for multiple units rolled up to a single result instead of a separate value for each period of time.

units - an integer representing the time units to query data for. Pass -1 to return all units of time.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

dt - datetime when shortens had been made.

shortens - the number of shortens made by the specified user in the specified time.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "", secret = "")
bitly_UserMetricsShortenCounts(unit = "day", units = -1, limit = 100, rollup = "true")
bitly_UserMetricsShortenCounts(unit = "day", units = -1, limit = 100, rollup = "false")
bitly_UserMetricsShortenCounts(unit = "day", units = -1, limit = 100)

## End(Not run)
```

bitly_UserTrackingDomains

Returns a list of tracking domains a user has configured.

Description

See http://dev.bitly.com/user_info.html#v3_user_tracking_domain_list

Usage

```
bitly_UserTrackingDomains(showRequestURL = FALSE)
```

Arguments

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

tracking_domains - a list of tracking domains configured for the authenticated user.

Examples

```
## Not run:
bitly_token <- bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8", secret = "")
bitly_UserTrackingDomains()

## End(Not run)
```

`googl_auth`*Assign API tokens using OAuth2.0*

Description

You should register an application in order to get Client ID and Client Secret code. For Bit.ly, go to https://bitly.com/a/oauth_apps and in the field Redirect URIs: type for example "http://localhost:1410". For Goo.gl API Keys you should go to the <http://console.developers.google.com/project/>, select "APIs & auth", then "Credentials", then "add OAuth2.0 client ID" and lastly you select "Type:Other".

Usage

```
googl_auth(key = "", secret = "")
```

```
bitly_auth(key = "", secret = "")
```

Arguments

<code>key</code>	- Client ID
<code>secret</code>	- Client Secret

See Also

See http://dev.bitly.com/rate_limiting.html

See <http://dev.bitly.com/authentication.html>

See https://developers.google.com/url-shortener/v1/getting_started#APIKey

Examples

```
## Not run:
googl_token <-
  googl_auth(key = "806673580943-78jdskus76fu7r0m21erihqtl1tcka29i.apps.googleusercontent.com",
             secret = "qItL-PZnm8GFxUOYM0zPvr_t")
bitly_token <-
  bitly_auth(key = "be03aead58f23bc1aee6e1d7b7a1d99d62f0ede8",
             secret = "b7e4abaf8b26ec4daa92b1e64502736f5cd78899")

## End(Not run)
```

googl_LinksExpand *Expand a short URL to a longer one*

Description

For the given short URL, the url.get method returns the corresponding long URL and the status.

Usage

```
googl_LinksExpand(shortUrl = "", projection = "FULL",  
  showRequestURL = FALSE)
```

Arguments

shortUrl - The short URL, including the protocol.

projection - "FULL" - returns the creation timestamp and all available analytics (default)
OR "ANALYTICS_CLICKS" - returns only click counts OR "ANALYTICS_TOP_STRINGS"
- returns only top string counts (e.g. referrers, countries, etc)

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

id - is the short URL you passed in.

longUrl - is the long URL to which it expands. Note that longUrl may not be present in the response, for example, if status is "REMOVED".

status - is "OK" for most URLs. If Google believes that the URL is fishy, status may be something else, such as "MALWARE".

Quotas

By default, your registered project gets 1,000,000 requests per day for the URL Shortener API (<https://console.developers.google.com/>)

Note

Returns a dataframe of expanded short URL and a list of its analytics.

See Also

See https://developers.google.com/url-shortener/v1/getting_started#shorten

See <https://developers.google.com/url-shortener/v1/url/get>

Examples

```
## Not run:
googl_token <- googl_auth(key = "", secret = "")
g1 <- googl_LinksExpand(shortUrl = "http://goo.gl/vM0w4", showRequestURL = TRUE)
g4 <- googl_LinksExpand(shortUrl="http://goo.gl/vM0w4", projection = "ANALYTICS_TOP_STRINGS")

## End(Not run)
```

googl_LinksShorten *Given a long URL, returns a short Goo.gl link.*

Description

Given a full URL, returns an goo.gl short URL. The returned resource contains the short URL and the long URL. Note that the returned long URL may be loosely canonicalized, e.g. to convert "google.com" into "http://google.com/". See the Authentication [googl_auth](#) section for more details.

Usage

```
googl_LinksShorten(longUrl = "", showRequestURL = FALSE)
```

Arguments

longUrl - a long URL to be shortened (example: <http://betaworks.com/>).

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

id is the short URL that expands to the long URL you provided. If your request includes an auth token, then this URL will be unique. If not, then it might be reused from a previous request to shorten the same URL.

longURL - longUrl is the long URL to which it expands. In most cases, this will be the same as the URL you provided. In some cases, the server may canonicalize the URL. For instance, if you pass <http://www.google.com>, the server will add a trailing slash.

See Also

See <https://developers.google.com/url-shortener/v1/url/insert>

See https://developers.google.com/url-shortener/v1/getting_started#shorten

See <http://stackoverflow.com/a/13168073>

Examples

```
## Not run:
googl_token <- googl_auth(key = "", secret = "")
g2 <- googl_LinksShorten(longUrl = "https://developers.google.com/url-shortener/v1/url/insert")

## End(Not run)
```

googl_UserLinkHistory *Retrieves a list of URLs shortened by the authenticated user.*

Description

returns a paginated list of information about short URLs created by a user, sorted in reverse chronological order. Each returned resource contains the short URL, long URL, creation timestamp, and status.

Usage

```
googl_UserLinkHistory(projection = "FULL", showRequestURL = FALSE)
```

Arguments

projection - an optional (!) information to return : "ANALYTICS_CLICKS" - Returns short URL click counts. OR "FULL" - Returns full analytics information (default)

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Value

totalItems - is an approximate number of items in the user's entire history.

nextPageToken - is an opaque string you can use to get the next page of history. It looks a lot like an ISO 8601 formatted date right now, but you should not count on that being true. The nextPageToken will be present on all but the last page

long_url - items contains the list of entries for the first "page" of the user's history, in order of descending creation time. The values for each entry are the same as specified in the Analytics section https://developers.google.com/url-shortener/v1/getting_started#url_analytics.

Note

Requires that the user authenticates with his google account through OAUTH 2.0 ! Thus no API key is necessary

See Also

See <https://developers.google.com/url-shortener/v1/url/list>

See https://developers.google.com/url-shortener/v1/getting_started#history

Examples

```
## Not run:
googl_token <- googl_auth(key = "", secret = "")
googl_UserLinkHistory(showRequestURL = TRUE)
googl_UserLinkHistory(projection = "FULL", showRequestURL = TRUE)

## End(Not run)
```

isgd_LinksExpand

Expand a short URL to a longer one

Description

See <https://is.gd/apilookupreference.php>

Usage

```
isgd_LinksExpand(shorturl = "", showRequestURL = FALSE)
```

Arguments

- `shorturl` - (optional character) You can specify the `shorturl` parameter if you'd like to pick a shortened URL instead of having `is.gd` randomly generate one. These must be between 5 and 30 characters long and can only contain alphanumeric characters and underscores. Shortened URLs are case sensitive. Bear in mind that a desired short URL might already be taken (this is very often the case with common words) so if you're using this option be prepared to respond to an error and get an alternative choice from your app's user.
- `showRequestURL` - show URL which has been build and requested from server. For debug purposes.

Examples

```
### isgd_LinksExpand(shorturl = "http://is.gd/4oIAXJ", showRequestURL = TRUE)
```

isgd_LinksShorten *Given a long URL, returns a short is.gd link.*

Description

See <https://is.gd/apishorteningreference.php>

Usage

```
isgd_LinksShorten(longUrl = "", logstats = "0", shorturl = NULL,  
  showRequestURL = FALSE)
```

Arguments

longUrl - The url parameter is the address that you want to shorten.

logstats - (optional) Adding the parameter logstats=1 turns on logging of detailed statistics when the shortened URL you create is accessed. This allows you to see how many times the link was accessed on a given day, what pages referred people to the link, what browser visitors were using etc. You can access these stats via the link preview page for your shortened URL (add a hyphen/dash to the end of the shortened URL to get to it). Creating links with statistics turned on has twice the "cost" towards our rate limit of other shortened links, so leave this parameter out of your API call if you don't require statistics on usage. See our usage limits page for more information on this <https://is.gd/usagelimits.php>.

shorturl - (optional character) You can specify the shorturl parameter if you'd like to pick a shortened URL instead of having is.gd randomly generate one. These must be between 5 and 30 characters long and can only contain alphanumeric characters and underscores. Shortened URLs are case sensitive. Bear in mind that a desired short URL might already be taken (this is very often the case with common words) so if you're using this option be prepared to respond to an error and get an alternative choice from your app's user.

showRequestURL - show URL which has been build and requested from server. For debug purposes.

Examples

```
## asd <- isgd_LinksShorten(longUrl = "http://debil.cz/", showRequestURL = TRUE)
```

Index

bitly_auth (googl_auth), [25](#)
bitly_IsProDomain, [2](#)
bitly_LinksExpand, [3](#)
bitly_LinksInfo, [4](#)
bitly_LinksLookup, [5](#)
bitly_LinksMetricsClicks, [6](#)
bitly_LinksMetricsCountries, [7](#)
bitly_LinksMetricsEncoders, [8](#)
bitly_LinksMetricsEncodersByCount, [9](#)
bitly_LinksMetricsEncodersCount, [10](#)
bitly_LinksMetricsReferrers, [11](#)
bitly_LinksMetricsReferrersByDomain,
[12](#)
bitly_LinksMetricsReferringDomains, [13](#)
bitly_LinksShorten, [14](#)
bitly_UserInfo, [15](#)
bitly_UserLinkHistory, [17](#)
bitly_UserMetricsClicks, [18](#)
bitly_UserMetricsCountries, [19](#)
bitly_UserMetricsPopularLinks, [20](#)
bitly_UserMetricsReferrers, [21](#)
bitly_UserMetricsReferringDomains, [22](#)
bitly_UserMetricsShortenCounts, [23](#)
bitly_UserTrackingDomains, [24](#)

googl_auth, [25](#), [27](#)
googl_LinksExpand, [26](#)
googl_LinksShorten, [27](#)
googl_UserLinkHistory, [28](#)

isgd_LinksExpand, [29](#)
isgd_LinksShorten, [30](#)