Package 'plusser'

August 29, 2016

Date 2014-04-27

Version 0.4-0

Title A Google+ Interface for R

Description plusser provides an API interface to Google+ so that posts, profiles and pages can be automatically retrieved.

URL http://kdss.at

Depends R (>= 3.0.2)

Imports RCurl, RJSONIO, lubridate, plyr

License GPL-3 | file LICENSE

LazyData true

Author Christoph Waldhauser [aut, cre]

Maintainer Christoph Waldhauser <chw@kdss.at>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-04-27 14:56:34

R topics documented:

harvestActivity *Retrieve the users that acted on a G+ post*

Description

This function retrieves the users that either +1ed or reshared a post. Google calls this 'list by activity'.

Usage

```
harvestActivity(activity, kind = c("plusoners", "resharers"),
  nextToken = NULL)
```

Arguments

activity	The post ID for which the users should be retrieved.
kind	Denoting the kind of person to be retrieved. Either plusoners or resharers.
nextToken	This is used internally to retrieve additional pages of answers from the Google+ API. Users won't need to set this argument.

Value

Returns a (character) vector of Google+ user IDs.

See Also

Google+ API documentation: https://developers.google.com/+/api/latest/people/listByActivity.

Examples

```
## Not run:
## User IDs of people that +1ed this post
users.p <- harvestActivity("z131ihvycs30ivrxm04cjbiwjkbqujka0sk0k", "plusoners")

## End(Not run)
```

harvestPage *Retrieve the posts of a user's G+ page*

Description

This function retrieves the most recent posts that a user put on her page. Google calls this 'listing activities'.

Usage

```
harvestPage(user, parseFun = parsePost, results = 1, nextToken = NULL,
  cr = 1)
```

Arguments

user	A user identification string: either user ID or +Name.
parseFun	A function for parsing the results, e.g. the supplied parsePost function. If set to NULL, then the raw list of the retrieved posts is being returned. Defaults to parsePost.
results	The approximate number of results that will be retrieved from Google+.
nextToken,cr	used internally to retrieve additional pages of answers from the Google+ API. Users won't need to set these arguments.

Details

The result is either a simple list of items from the page that can be parsed using e.g. parsePost or a data frame with that (or another user-supplied) function already applied.

When writing your own parsing functions, make sure that the function takes a single list item from the raw list as its argument and returns a vector of values or a one-row data frame. The return values of the function are then fed into plyr's ldply to turn it into a data frame. See parsePost for an example.

The length of the list or the number of rows of the data frame are somewhat ambiguous. Specifying the results argument will try to get that many results. But there may be less (because Google could not find more) or more (because Google is organizing results on pages and it would be a waste to discard them automatically). If you really depend on getting not more rows than you expected, use standard selection (i.e. [) to trim the results.

Value

The function returns either the raw list of retrieved posts or whatever the supplied parsing function does with the retrieved list.

See Also

Google+ API documentation: https://developers.google.com/+/api/latest/activities/list.

Examples

```
## Not run:
myPosts.df <- harvestPage("115046504166916768425")
gPosts.df <- harvestPage("+google", results=200)

## End(Not run)
```

harvestProfile *Retrieve the profile of Google+ users*

Description

This function retrieves the profile of one or more Google+ user(s). Google calls this 'get people'. The results are either returned as a raw list with one element per profile or parsed using a parsing function, either the prepackaged one parseProfile or a user-supplied one.

Usage

```
harvestProfile(id, parseFun = parseProfile)
```

Arguments

id	A character vector of the Google+ user ID(s).
parseFun	the function used to parse the results. If NULL the raw list of results is returned.

Details

When using your own parsing function, be sure that it takes a single element from the returned list and returns either a vector of values or a single row data frame.

Value

The function returns either a raw list or a parsed version. See Details.

See Also

Google+ API documentation: https://developers.google.com/+/api/latest/people/get

Examples

```
## Not run:
gProfile <- harvestProfile("+google")

## End(Not run)
```

parseAttachments *Parsing of post attachments*

Description

This function takes a raw list of posts (as retrieved by harvestPage) and extracts any attachments it might find. It uses private (i.e. not exported) parsing functions for some known attachment types. At present, these are articles, albums, photos and videos. Other attachment types will just be cast generically to data.frames. The rownames of all these data frames are the ids of the posts that attachment belongs to. The columns of the returned data frames should be pretty much self-explanatory. If in doubt, check the Google+ API documentation https://developers.google.com/+/api/latest/activities#object.attachments.

Usage

```
parseAttachments(pl)
```

Arguments

pl	a posting list as retrieved e.g. by harvestPage.

Value

A list containing one data frame per identified attachment type.

Examples

```
## Not run:
myPosts <- harvestPage("115046504166916768425", ret="list")
myPosts.att <- parseAttachments(myPosts)

## End(Not run)
```

parsePost *Parsing a Google+ post*

Description

This function turns a Google+ post into a (1 row) data frame extracting or computing a number of fields. See Details.

Usage

```
parsePost(p)
```

Arguments

p	A raw post as returned from e.g. the harvestPage function.

Details

This function extracts or computes the following fields:

- ti Date and time the post was published.
- age The age of the post as difference between now and ti in (floating point) days.
- id The post's unique Google+ post ID.
- au The post's author's Google+ user ID.
- ve The action describing the post.
- nC The number of comments the post has attracted so far.
- nP The number of +1s the post has attracted so far.
- nR The number of times the post has been reshared so far.
- at The type of attachment (article, photo, ...) as reported by the API
- msg The post's content.

Value

A 1 row data frame filled with the information from the post parsed.

Examples

```
## Not run:
myPosts <- harvestPage("115046504166916768425", ret="list")
myPosts.df <- ldply(myPosts, parsePost)

## End(Not run)
```

---

parseProfile *Parse profile*

Description

This function takes a raw list of profiles (as retrieved by harvestProfile) and parses the contained information into a one row data frame.

Usage

```
parseProfile(p)
```

Arguments

p	a raw profile as retrieved e.g. by harvestProfile.

Details

The following fields will be filled with data (if available) or NA otherwise:

`id` The Google+ user ID.

`sex` The user's gender: `male`, `female`, or `other`.

`ln` The user's last name.

`fn` The user's first name.

`verified` Logical. `TRUE` if it is a verified Google+ profile.

`website` A URL listed in the profile.

`ageMin, ageMax` Google+ provides only age ranges for some profiles. This will contain the lower and upper bound of the age range of the user.

`bday` The birthday of the user (YYYY-MM-DD).

`nCircled` The number of Persons the user circled by.

`currentLoc` The user's current location.

`lang` The primary language the user reported.

`p1count` The number of +1s the user awarded.

`relationship` The user's relationship status.

`bio` The 'About Me' short autobiography.

`tagline` The tagline of a profile.

`type` The type of a profile: `person` or `page`.

`brag` The 'bragging rights' section of the profile.

`occ` The person's occupation.

`skills` The person's skills.

Value

a one row data frame with a number of fields. See Details.

See Also

Google+ API documentation: https://developers.google.com/+/api/latest/people/get

Examples

```
## Not run:
gProfile <- parseProfile(harvestProfile("+google", parseFun=NULL))

## End(Not run)
```

plusser *plusser: A Google+ API Interface for R*

Description

The idea of this package is to provide a high level interface to Google's Google+ social network. As of now, functions related to data retrieval are available. The authentication (beyond using an API key) and posting messages parts of the API are not yet implemented.

Details

As a high-level interface between R and Google+, this package is aimed at social media and researchers. Google does not provide an API to retrieve the following relation ships without full authentication, this package is not yet suitable for social network analysis.

On the social media side, this package can be used to retrieve posts, a post's popularity (comments, +1s, reshares) and the profiles of entities. Espescially interesting might be the retrieval of profiles that interacted with certain posts.

This package is designed as modular as possible with separating harvest and parsing functions. Users can use their own parsing functions if required.

Google+ API Key

This section describes briefly how to obtain a Google+ API key.

1. Go to the Google Developers Console at `https://console.developers.google.com/project`.
2. Create a new project and open it.
3. In the menu, choose `APIs & auth`.
4. Choose Google+ API on the next screen.
5. Activate it by clicking `On`.
6. Choose credentials from the submenu.
7. Click `Create new key` and write it down. This is your API key.

Google+ Quotas

Currently, Google permits 5 requests per second up to a maximum total of 10,000 requests per day.

References

See the official Google Google+ API documentation: `https://developers.google.com/+/api/latest/`.

searchPost *Searching for Google+ Posts*

Description

This function uses the Google+ API to search for a text string in posts. Optionally, search results can be restricted to a certain language.

Usage

```
searchPost(q, ret = "data.frame", language = NULL, results = 1,
  nextToken = NULL, cr = 0)
```

Arguments

q	The query string to search. The string is URL encoded automatically.
ret	A string specifying the kind of return value. Either a `list` of the retrieved items on the page, or that list parsed into a `data.frame`.
language	A language code. See https://developers.google.com/+/api/search#available-languages.
results	The approximate number of results that will be retrieved from Google+.
nextToken,cr	used internally to retrieve additional pages of answers from the Google+ API. Users won't need to set this argument.

Details

The result is either a simple list of items from the page that can be parsed using `parsePost` or a data frame with that function already applied.

The length of the list or the number of rows of the data frame are somewhat ambiguous. Specifying the `results` argument will try to get that many results. But there may be less (because Google could not find more) or more (because Google is organizing results on pages and it would be a waste to discard them automatically). If you really depend on getting not more rows than you expected, use standard selection (i.e. `[`) to trim the results.

Value

The function returns a list or a data frame containing all available data on the posts that met the search criteria. See `Details` for more on its content.

See Also

Google+ API documentation: https://developers.google.com/+/api/latest/activities/search.

Examples

```
## Not run:
searchPost("#cats")

## End(Not run)
```

`searchProfile` *Searching in Google+ Profiles*

Description

This function uses the Google+ API to search for a text string in profiles. Optionally, profiles can be restricted to a certain language.

Usage

```
searchProfile(q, language = NULL, results = 1, nextToken = NULL, cr = 1)
```

Arguments

`q`	The query string to search. The string is URL encoded automatically.
`language`	A language code. See https://developers.google.com/+/api/search#available-languages.
`results`	The approximate number of results that will be retrieved from Google+.
`nextToken,cr`	These are used internally to retrieve additional pages of answers from the Google+ API. Users won't need to set these arguments.

Details

The number of rows of the data frame returned is somewhat ambiguous. Specifying the `results` argument will try to get that many results. But there may be less (because Google could not find more) or more (because Google is organizing results on pages and it would be a waste to discard them automatically). If you really depend on getting not more rows than you expected, use standard selection (i.e. `[`) to trim the results.

Value

A data frame with the user ID, display names and profile type of the profiles that met the search criteria.

See Also

Google+ API documentation: https://developers.google.com/+/api/latest/people/search.

Examples

```
## Not run:
searchProfile("cats")

## End(Not run)
```

`setAPIkey` *Sets an API key for the Google+ API*

Description

This function sets an API key that is then stored invisibly for `plusser` to use when accessing the Google+ API. A warning is issued if URL escaping the api key alters it, as Google should provide you with a HTML-safe API key in the first place.

Usage

```
setAPIkey(apikey)
```

Arguments

`apikey`	The API key as a character string.

Value

Returns `TRUE` if the key was stored successfully.

Examples

```
setAPIkey("thisIsInvalid")
```

Index