Package 'Rlinkedin'

January 20, 2025

Version 0.2

Title Access to the LinkedIn API via R

Description A series of functions that allow users
to access the 'LinkedIn' API to get information about connections,
search for people and jobs, share updates with their network,
and create group discussions. For more information about using
the API please visit <https://developer.linkedin.com/>.

Author Michael Piccirilli <michael.r.piccirilli@gmail.com>

Maintainer Michael Piccirilli <michael.r.piccirilli@gmail.com>

Depends R (>= 2.12.0), httr, XML, httpuv

Imports methods

License GPL-2

LazyData true

URL https://github.com/mpiccirilli/Rlinkedin

BugReports https://github.com/mpiccirilli/Rlinkedin/issues

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-30 08:58:23

Contents

`Rlinkedin-package` *Access to LinkedIn API via R*

Description

This is an R package that provides a series of functions that allow users to access the LinkedIn API to get information about connections, search for people, search for jobs, share updates with your network, and create group discussions.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

`inOAuth`, `getProfile`, `getMyConnections`, `getGroupPosts`, `getGroups`, `getJobs`, `searchJobs`, `searchPeople`, `submitGroupPost`, `submitShare`, `getCompany`, `searchCompanies`

`getCompany` *Retrieve Company Profile Information*

Description

`getCompany` retrieves company profiles using a company Id, a universal name, or an email domain.

Usage

```
getCompany(token, universal_name = NULL, email_domain = NULL,
  company_id = NULL, partner = 0)
```

Arguments

`token`	Authorization token.
`universal_name`	LinkedIn universal company name. This is the exact name seen at the end of the URL on the company page on linkedin.com.
`email_domain`	The email domain used by the company.
`company_id`	LinkedIn company ID.
`partner`	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Details

The 'universal name' needs to be the exact name seen at the end of the URL on the company page on linkedin.com.

Value

Returns company profile data, such as LinkedIn ID, name, universal-name, email-domains, company-type, ticker, website-url, industries, status, twitter handle, employee-count-range, specialties, locations, description, founded-year, and number of followers.

The output when specifying the `universal_name` or `company_id` will be in a list, whereas the output when specifying the `email_domain` will be a dataframe.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

searchCompanies

Examples

```
## Not run:

company.name <- getCompany(token=in.auth, universal_name="Facebook")

company.email <- getCompany(token=in.auth, email_domain = "columbia.edu")

# Main Columbia Name:
company.id <- getCompany(token=in.auth, company_id = company.email$company_id[14])


## End(Not run)
```

---

getGroupPosts *Extract Posts from your LinkedIn Groups*

Description

`getGroupPosts` will retrieve posts from each LinkedIn group you belong to.

Usage

```
getGroupPosts(token, partner = 0)
```

Arguments

`token`	Authorization token.
`partner`	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Value

Returns a dataframe with the 10 most recent posts from each group you belong to.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

getGroups submitGroupPost

Examples

```
## Not run:

my.groups <- getGroupPosts(in.auth)

## End(Not run)
```

---

getGroups *Retrieve LinkedIn Group Information*

Description

getGroups retrieves information and settings about the LinkedIn groups you belong to.

Usage

```
getGroups(token, details = FALSE, partner = 0)
```

Arguments

token	Authorization token.
details	TRUE or FALSE. If TRUE, it will return group details. FALSE is default.
partner	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Details

This function returns information about what groups you belong to, either with or without group details. Group details can be called by setting the option details = TRUE.

Value

Returns a dataframe including group profile information.

When details = FALSE (default), the function will return information about each group's settings such as whether it allows messages from members, email frequency, and manager announcements.

When details = TRUE, the function will return both a short and long description of the group.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

getGroupPosts submitGroupPost

Examples

```
## Not run:

my.groups <- getGroups(token = in.auth, details=TRUE)

## End(Not run)
```

---

getJobs *Bookmarked and Recommended Jobs on LinkedIn*

Description

getJobs can be used to retrieve your bookmarked and suggested jobs.

Usage

```
getJobs(token, suggestions = NULL, bookmarks = NULL, partner = 0)
```

Arguments

token	Authorization token.
suggestions	TRUE or FALSE. If TRUE, it will return LinkedIn's job recommendations.
bookmarks	TRUE or FALSE. If TRUE, it will return jobs you've bookmarked on LinkedIn.
partner	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Details

This function can return either jobs you've bookmarked on LinkedIn, or jobs LinkedIn is recommending for you, but not both at the same time.

Value

Returns a dataframe of recommended or bookmarked jobs.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

searchJobs

Examples

```
## Not run:

job.suggestions <- getJobs(in.auth, suggestions=TRUE)
job.bookmarks <- getJobs(in.auth, bookmarks=TRUE)

## Will return NULL
job.fail <- getJobs(in.auth)


## End(Not run)
```

---

getMyConnections *Retrieve 1st Degree LinkedIn Connections*

Description

getMyConnections returns information about your 1st degree connections who do not have their profile set to private.

You cannot "browse connections." That is, you cannot get connections of your connections (2nd degree connections).

Usage

```
getMyConnections(token, partner = 0)
```

Arguments

token	Authorization token.
partner	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Value

Returns a dataframe of your 1st degree LinkedIn connections.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

getProfile, searchPeople

Examples

```
## Not run:

my.connections <- getMyConnections(in.auth)

## End(Not run)
```

`getProfile` *Extract LinkedIn Profile Information*

Description

`getProfile` retrieve's proflie information about to yourself, your connections, or another individual.

Usage

```
getProfile(token, connections = FALSE, id = NULL, partner = 0)
```

Arguments

`token`	Authorization token.
`connections`	TRUE or FALSE. If TRUE, will return profile information of your connections. FALSE is default.
`id`	Numeric ID number of a LinkedIn member.
`partner`	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Details

There are three separate calls in `getProfile`.

The first is to return profile information about yourself. The only input into the function under this scenario is the `token`.

The second is to return profile information about all your 1st degree connections. You need to supply the `token` and set the `connections` = TRUE.

The third is to return profile information about an individual based on their id number. This can be found if you search your connections using the `getMyConnections` function.

Value

Returns a list of profile information.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

searchPeople, getMyConnections

Examples

```
## Not run:

profiles <- getProfile(in.auth, connections=TRUE)

## End(Not run)
```

---

inOAuth *Create OAuth token to LinkedIn R session*

Description

inOAuth creates a long-lived OAuth access token that enables R to make authenticated calls to the LinkedIn API. This function relies on the httr package to create the OAuth token.

Usage

```
inOAuth(application_name = NULL, consumer_key = NULL,
  consumer_secret = NULL)
```

Arguments

application_name	Name of your application.
consumer_key	Consumer API Key of your application.
consumer_secret	Consumer Secret Key of your application.

Details

There are two ways to create an authenticated connection. One is to use the default credentials supplied in the package. The second is to obtain your own credentials and using them as inputs into the function. Examples of both are shown below.

Create your own application here: https://developer.linkedin.com/

Value

Authorization token to be used in other functions.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

getProfile, getMyConnections

Examples

```
## Not run:

## Default Consumer and Secret Key for the Rlinkedin package:
in.auth <- inOAuth()

## Use your own Consumer and Secret Key:
in.auth <- inOAuth("your_app_name", "your_consumer_key", "your_consumer_secret")

## End(Not run)
```

---

searchCompanies *Search for Companies on LinkedIn*

Description

searchCompanies searches across LinkedIn's companies pages based on keywords, location, and industry.

Usage

```
searchCompanies(token, keywords, location = NULL, industry = NULL)
```

Arguments

token	Authorization token.
keywords	A keyword used anywhere in a company's listing. Multiple words should be separated by a space.
location	LinkedIn geography code, found here: https://developer.linkedin.com/docs/reference/geography-codes.
industry	LinkedIn industry code, found here: https://developer.linkedin.com/docs/reference/industry-codes.

Details

In order to narrow the search down by location or industry, you must look up the proper input codes on the linkedin website. The geography codes can be found here: https://developer.linkedin.com/docs/reference/geography-codes, and the industry codes can be found here: https://developer.linkedin.com/docs/reference/industry-codes.

Value

Returns a list, information includes company id, company name, universal name, website, twitter handle, employee count, founded date, number of followers, and company description.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

getCompany searchJobs

Examples

```
## Not run:

search.comp <- searchCompanies(in.auth, keywords = "LinkedIn")


## End(Not run)
```

---

searchJobs *Search for Jobs on LinkedIn*

Description

searchJobs searches across LinkedIn's job postings.

There are several parameters that allow you to conduct either a broad or focused search.

In order to use this function, you must create your own appliction and apply for the Vetted API Access here: https://help.linkedin.com/app/ask/path/api-dvr. You cannot use the default credentials supplied in the Rlinkedin package.

Usage

```
searchJobs(token, keywords = NULL, company_name = NULL, job_title = NULL,
  country_code = NULL, postal_code = NULL, distance = NULL, partner = 0)
```

Arguments

token	Authorization token.
keywords	A keyword used in the job title or description. Multiple words should be separated by a space.
company_name	Company posting the job.
job_title	Title of the job.
country_code	Specify the country in which to search. This is the ISO3166 country code, and must be in lower case.
postal_code	Must be combined with the country_code parameter.
distance	Distance matches jobs within a distance from a central point. This is measured in miles and is best used in conjunction with both country_code and postal-code.
partner	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Details

There are many different search parameters that allow you to make a focused search of a particular job within a certain company some area of the country. Or you can search for all jobs posted based on general keywords.

Value

Returns a dataframe of jobs based input parameters

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

getJobs searchCcompanies

Examples

```
## Not run:

search.results <- searchJobs(token = in.auth, keywords = "data scientist")

## End(Not run)
```

---

searchPeople *Search for People on LinkedIn*

Description

searchPeople allows you to search for connections on LinkedIn. It returns most of what shows up when you do a search for people in the box at the top of the page on linkedin.com.

There are a number of parameters that allow you to conduct either a broad or focused search.

In order to use this function, you must create your own appliction and apply for the Vetted API Access here: https://help.linkedin.com/app/ask/path/api-dvr.

Usage

```
searchPeople(token, keywords = NULL, first_name = NULL, last_name = NULL,
  company_name = NULL, current_company = NULL, title = NULL,
  current_title = NULL, school_name = NULL, current_school = NULL,
  country_code = NULL, postal_code = NULL, distance = NULL, partner = 0)
```

Arguments

token	Authorization token.
keywords	A keyword used in a person's profile. Multiple words should be separated by a space.
first_name	Search by a user's first name.
last_name	Search by a user's last name.
company_name	The name of a company where someone has as worked.
current_company	TRUE or FALSE, can only be used in conjunction with company_name.
title	A job title someone has held
current_title	TRUE or FALSE, can only be used in conjunction with title.
school_name	The name of a school someone has attended.
current_school	TRUE or FALSE, can only be used in conjuntion with current_school.
country_code	Specify the country in which to search. This is the ISO3166 country code, and must be in lower case.
postal_code	Must be combined with the country_code parameter.
distance	Distance matches members within a distance from a central point. This is measured in miles and and is best used in conjunction with both country_code and postal-code.
partner	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Value

Returns a dataframe of people based input parameters

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

getProfile

Examples

```
## Not run:

search.results <- searchPeople(token=in.auth, first_name="Michael", last_name="Piccirilli")

## End(Not run)
```

submitGroupPost *Create a Group Discussion Post*

Description

submitGroupPost will create a group discussion post in one of the groups you belong to, specified by a Group Id.

Usage

```
submitGroupPost(token, group_id, disc_title = NULL, disc_summary = NULL,
  content_title = NULL, content_url = NULL, content_img = NULL,
  content_desc = NULL, partner = 0)
```

Arguments

token	Authorization token.
group_id	Numeric Group ID.
disc_title	Group discussion title, required.
disc_summary	Group discussion summary, required.
content_title	Title for content, required.
content_url	Url for content, optional.
content_img	Url for an image, optional.
content_desc	Description of content, optional.
partner	Indicate whether you belong to the Partnership Program. Values: 0 or 1

Details

You must include a minimum of a discussion title, discussion summary, and content title.

Value

There are two possible responses to a successful submittal.

One, your post has been created and is visibile immediately. In this case you have most likely posted to an unmoderated group.

Two, your post has been accepted by the API but is pending approval by the group moderator, in which case you will not see your post until it has bene approved.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

getGroups getGroupPosts

Examples

```
## Not run:

my.groups <- getGroups(in.auth)

id <- my.groups$group_id[1]
disc.title <- "Test connecting to the LinkedIn API via R"
disc.summary <- "Im creating an R package to connect to the LinkedIn API,
                + this is a test post from R!"
url <- "https://github.com/mpiccirilli"
content.desc <- "Dev version of access to LinkedIn API via R.
                              + Collaboration is welcomed!"

submitGroupPost(in.auth, group_id=id, disc_title=disc.title,
disc_summary=disc.summary, content_url=url, content_desc=content.desc)

## End(Not run)
```

---

submitShare *Share an update to your network's activity feed*

Description

submitShare will post a network update to the newsfeed of your connections. You can select the visibility of your post to be seen either by 'anyone' or 'connections-only'.

Usage

```
submitShare(token, comment = NULL, content_title = NULL,
  content_desc = NULL, content_url = NULL, content_img = NULL,
  visibility = "anyone")
```

Arguments

token	Authorization token.
comment	Headline of your post.
content_title	Title of your post.
content_desc	Description of your post.
content_url	Url to content you'd like to share. This is required if you specify either content_title or content_desc.
content_img	Url to an image you would like to include in your post, optional.
visibility	Choose the visibility of the post. The choices are 'anyone' or 'connections-only'.

Details

If either content_title or content_desc is specified, you must also include a content_url for the post.

Value

Shares an update to your networks's activity feed.

Author(s)

Michael Piccirilli <michael.r.piccirilli@gmail.com>

See Also

submitGroupPost

Examples

```
## Not run:

comment <- "Test connecting to the LinkedIn API via R"
title <- "Im creating an R package to connect to the LinkedIn API, this is a test post from R!"
url <- "https://github.com/mpiccirilli"
desc <- "Dev version of access to LinkedIn API via R. Collaboration is welcomed!"

submitShare(token = in.auth, comment=comment, content_tile=title,
            content_url=url, content_desc=desc)

## End(Not run)
```

Index