

Package ‘calpassapi’

August 27, 2018

Title R Interface to Access CalPASS API

Version 0.0.2

Description Implements methods for querying data from CalPASS using its API.
CalPASS Plus. MMAP API V1. <<https://mmap.calpassplus.org/docs/index.html>>.

Depends R (>= 3.4.0)

Imports httr, dplyr, digest, jsonlite, stringr

License GPL-3

URL <https://github.com/vinhdizzo/calpassapi>

BugReports <https://github.com/vinhdizzo/calpassapi/issues>

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Vinh Nguyen [aut, cre]

Maintainer Vinh Nguyen <nguyenvq714@gmail.com>

Repository CRAN

Date/Publication 2018-08-27 19:24:32 UTC

R topics documented:

calpass_create_isk	2
calpass_get_token	3
calpass_query	4

Index	7
--------------	----------

calpass_create_isk *Create interSegmentKey's for students*

Description

Create interSegmentKey's from students' first names, last names, genders, and birthdates

Usage

```
calpass_create_isk(first_name, last_name, gender, birthdate)
```

Arguments

first_name	a character vector of students' first names.
last_name	a character vector of students' last names.
gender	a character vector of students' genders. The first character will be used (upper-case'd automatically), and should take on values 'M', 'F', or 'X' (use 'X' for unknown or did not disclosed).
birthdate	a character or numeric vector of birthdates of the form 'yyyymmdd'.

Value

a vector of interSegmentKey's

Author(s)

Vinh Nguyen

Examples

```
## single
calpass_create_isk(first_name='Jane', last_name='Doe'
, gender='F', birthdate=20001231)
## data frame
## Not run:
firstname <- c('Tom', 'Jane', 'Jo')
lastname <- c('Ng', 'Doe', 'Smith')
gender <- c('Male', 'Female', 'X')
birthdate <- c(2001231, 19990101, 19981111)
df <- data.frame(firstname, lastname
, gender, birthdate, stringsAsFactors=FALSE)
library(dplyr)
df %>%
  mutate(isk=calpass_create_isk(first_name=firstname
, last_name=lastname
, gender=gender
, birthdate
))
```

```
## End(Not run)
```

calpass_get_token	<i>Obtain CalPASS API token</i>
-------------------	---------------------------------

Description

Obtain a token from CalPASS using your API credentials, which should allow access for 60 minutes.

Usage

```
calpass_get_token(username = Sys.getenv("cp_api_uid"),  
  password = Sys.getenv("cp_api_pwd"), client_id, scope,  
  auth_endpoint = "https://oauth.calpassplus.org/connect/token",  
  verbose = FALSE)
```

Arguments

username	API username. For security reasons, the user could specify cp_api_uid in the user's .Renviron file in the user's home directory (execute Sys.getenv('HOME') in R to check path to home directory). That way, the user does not have to hard code the username in their R script. The function uses for the username here by default.
password	API password. The user could specify cp_api_pwd as above.
client_id	parameter needed in the http body in order to obtain a token (unique to username)
scope	parameter needed in the http body in order to obtain a token (unique to username)
auth_endpoint	Authentication endpoint/url, defaults to 'https://oauth.calpassplus.org/connect/token'.
verbose	If TRUE, then print http exchanges (to assist with debugging). Defaults to FALSE.

Value

CalPASS token string

Author(s)

Vinh Nguyen

References

[MMAP API V1: Getting An Access Token Using User Credentials](#)

Examples

```
## Not run:
cp_token <- calpass_get_token(username='my_cp_api_uid', password='my_cp_api_pwd'
  , client_id='my_client_id'
  , scope='my_scope'
  )

## End(Not run)
```

calpass_query

Query data from CalPASS API endpoints

Description

Query data from CalPASS API endpoints for a single interSegmentKey

Usage

```
calpass_query(interSegmentKey, token,
  api_url = "https://mmap.calpassplus.org/api", endpoint = c("transcript",
  "placement"), verbose = FALSE)
```

```
calpass_query_many(interSegmentKey, token,
  api_url = "https://mmap.calpassplus.org/api", endpoint = c("transcript",
  "placement"), verbose = FALSE, api_call_limit = 150,
  limit_per_n_sec = 60, wait = FALSE)
```

Arguments

interSegmentKey	for calpass_query, a single interSegmentKey; for calpass_query_many, a vector of interSgementKey's. The interSegmentKey's can be created from calpass_create_isk .
token	a token string created from calpass_get_token .
api_url	defaults to 'https://mmap.calpassplus.org/api', but can be overrode if CalPASS changes the url.
endpoint	the api endpoint to use; defaults to 'transcript'.
verbose	If TRUE, then print http exchanges (to assist with debugging). Defaults to FALSE.
api_call_limit	the number of api calls allowed per limit_per_n_sec; defaults to 150 calls per 60 seconds.
limit_per_n_sec	time frame where api_call_limit is applicable to; defaults to 60 seconds.
wait	indicates whether the user is willing to wait limit_per_n_sec seconds per batch if the number of unique values in interSegmentKey is greater than api_call_limit; defaults to FALSE. The user should set to TRUE if there are more than api_call_limit number of calls to be executed.

Value

a data frame with columns `interSegmentKey`, `status_code` (the http response code: 200 means student was found, 204 means student was not found, 429 means the api limit was reached and student was not processed, and anything else in the 400's correspond to http errors.)

Functions

- `calpass_query_many`: Query data from CalPASS API endpoints with a vector of `interSegmentKey`'s. The number of rows returned corresponds to the number of unique `interSegmentKey`'s.

Author(s)

Vinh Nguyen

References

[MMAP API V1](#)

Examples

```
## Not run:
## get access token
cp_token <- calpass_get_token(username='my_cp_api_uid', password='my_cp_api_pwd')

## single run
isk <- calpass_create_isk(first_name='Jane', last_name='Doe'
  , gender='F', birthdate=20001231)
calpass_query(interSegmentKey=isk
  , token=cp_token, endpoint='transcript')
calpass_query(interSegmentKey=isk
  , token=cp_token, endpoint='placement')

## multiple
firstname <- c('Tom', 'Jane', 'Jo')
lastname <- c('Ng', 'Doe', 'Smith')
gender <- c('Male', 'Female', 'X')
birthdate <- c(20001231, 19990101, 19981111)
df <- data.frame(firstname, lastname
  , gender, birthdate, stringsAsFactors=FALSE)
library(dplyr)
df %>%
  mutate(isk=calpass_create_isk(first_name=firstname
    , last_name=lastname
    , gender=gender
    , birthdate
  ))
dfResults <- calpass_query_many(interSegmentKey=df$isk
  , token=cp_token
  , endpoint='transcript'
)
```

```
## End(Not run)
```

Index

`calpass_create_isk`, [2](#), [4](#)
`calpass_get_token`, [3](#), [4](#)
`calpass_query`, [4](#)
`calpass_query_many` (`calpass_query`), [4](#)