

Setting up the Scopus API Key

John Muschelli

2019-09-17

rscopus

The goal of rscopus is to provide an R Scopus Database API Interface.

Steps to get API key

In order to use this package, you need an API key from https://dev.elsevier.com/sc_apis.html. You should login from your institution and go to Create API Key. You need to provide a website URL and a label, but the website can be your personal website, and agree to the terms of service.

1. Go to <https://dev.elsevier.com/user/signin>. Login or create a free account.
2. Click "Create API Key". Put in a label, such as `rscopus key`. Add a website. <http://example.com> is fine if you do not have a site.
3. **Read** and agree to the TOS if you do indeed agree.
4. Add `Elsevier_API = "API KEY GOES HERE"` to `~/.Renviron` file, or add `export Elsevier_API=API KEY GOES HERE` to your `~/.bash_profile`.

Alternatively, you can either set the API key using `rscopus::set_api_key` or by `options("elsevier_api_key" = api_key)`. You can access the API key using `rscopus::get_api_key`.

You should be able to test out the API key using the [interactive Scopus APIs](#).

A note about API keys and IP addresses

The API Key is bound to a set of IP addresses, usually bound to your institution. Therefore, if you are using this for a Shiny application, you must host the Shiny application from your institution servers in some way. Also, you cannot access the Scopus API with this key if you are offsite and must VPN into the server or use a computing cluster with an institution IP.

See https://dev.elsevier.com/tecdoc_api_authentication.html

Example

This is a basic example which shows you how to solve a common problem:

```
library(rscopus)
library(dplyr)
if (have_api_key()) {
```

```

res = author_df(last_name = "Muschelli", first_name = "John", verbose = FALSE,
general = FALSE)
names(res)
head(res[, c("title", "journal", "description")])
unique(res$au_id)
unique(as.character(res$affilname_1))

all_dat = author_data(last_name = "Muschelli",
                      first_name = "John", verbose = FALSE, general = TRUE)

res2 = all_dat$df
res2 = res2 %>%
  rename(journal = `prism:publicationName`,
         title = `dc:title`,
         description = `dc:description`)
head(res[, c("title", "journal", "description")])
}
#> Warning: 'entries_to_df' is deprecated.
#> Use 'gen_entries_to_df' instead.
#> See help("Deprecated")
#>
#> title
#> 1
#> Neuroconductor: An R platform for medical imaging analysis
#> 2
#> MIMoSA: An Approach to automatically
#> segment T2 hyperintense and T1 hypointense lesions in multiple sclerosis
#> 3
#> Objective Evaluation of Multiple Sclerosis Lesion
#> Segmentation using a Data Management and Processing Infrastructure
#> 4
#> MIMoSA: An Automated Method for
#> Intermodal Segmentation Analysis of Multiple Sclerosis Brain Lesions
#> 5 Radiomic subtyping improves disease stratification beyond key molecular, clinical,
#> and standard imaging characteristics in patients with glioblastoma
#> 6
#> Feasibility of Coping Effectiveness Training for Caregivers of
#> Children with Autism Spectrum Disorder: a Genetic Counseling Intervention
#>
#> journal
#> 1
#> Biostatistics
#> 2 Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial
#> Intelligence and Lecture Notes in Bioinformatics)
#> 3
#> Scientific Reports
#> 4
#> Journal of Neuroimaging
#> 5
#> Neuro-Oncology
#> 6
#> Journal of Genetic Counseling
#>
#> description
#> 1
#> Article
#> 2 Conference Paper
#> 3
#> Article
#> 4
#> Article
#> 5
#> Article
#> 6
#> Article

```