

Advanced Data IO

Data Wrangling in R

Google Sheets

Untitled spreadsheet

File Edit View Insert Format Data Tools Add-ons Help

Share

fx

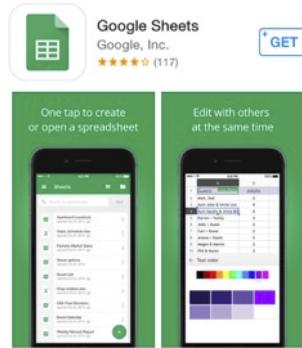
	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

+ Sheet1 ▾ ⋮ star

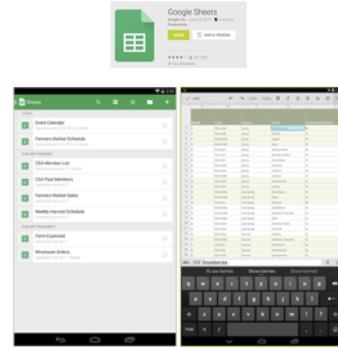
<https://docs.google.com/spreadsheets>

<https://speakerdeck.com/jennybc/googlesheets-talk-at-user2015>

iOS



Android



enter data from a phone
enter data w/o WiFi

Reading data with the `googlesheets` package

```
install.packages("googlesheets")
library(googlesheets)
```

data_wrangling_background_2019 ★ A+ ⚙

File Edit View Insert Format Data Tools Add-ons Help

Share

Git

	A	B	C	D	E	F	G	H	I
1	Git	Github	R	Rstudio	Reproducible Research	R markdown	Data import	Web scraping	Data cleaning
2		9	9	9	9	9	9	10	10
3		1	1	5	5	7	5	4	2
4		0	0	2	3	3	0	3	0
5		0	1	1	1	1	1	1	0
6		0	2	7	7	7	5	8	0
7		2	2	3	2	0	0	1	1
8		0	0	4	5	7	5	5	2
9		1	2	3	3	5	0	0	0
10		1	1	3	3	4	3	3	0
11		1	1	5	5	6	4	7	1
12		1	2	3	3	3	2	2	1
13		1	1	4	4	3	3	5	2
14		0	1	5	6	0	0	1	1
15		0	0	1	1	9	0	0	0
16		1	2	7	8	5	6	5	0
17		1	1	3	3	9	1	3	0
18		1	1	5	5	5	2	6	0
19		1	1	1	2	1	3	3	1
20		0	1	2	3	2	0	3	4

+ ⌂ Sheet1 Explore

<https://docs.google.com/spreadsheets/d/1WBrH655fxqKW1QqvD5hnqvWEWlvRzDJcKEg>

docs.google.com/spreadsheets/d/1WBrH655fxqKW1QqvD5hnqvEWlvRzDJcKEgjjFeYx...

data_wrangling_background_2019

File Edit View Insert Format Data Tools Add-ons Help

Share

- New
- Open
- Import
- Make a copy
- Download
- Email as attachment
- Make available offline
- Version history
- Rename
- Move
- Add shortcut to Drive
- Move to trash
- Publish to the web
- Email collaborators

C	D	E	F	G	H	I	dp
	Rstudio	Reproducible Research	R markdown	Data import	Web scraping	Data cleaning	
9	9	9	9	10	10	10	9
5	5	7	5	4	2	6	
2	3	3	0	3	0	2	
1	1	1	1	1	1	0	0
7	7	7	5	8	0	6	
3	2	0	0	1	1	2	
4	5	7	5	5	2	4	
3	3	5	0	0	0	7	
3	3	4	3	3	0	3	
5	5	6	4	7	1	5	
3	3	3	2	2	1	1	
4	4	3	3	5	2	4	
5	6	0	0	1	1	5	
1	1	9	0	0	0	0	0
7	8	5	6	5	0	6	
3	3	9	1	3	0	3	
5	5	5	2	6	0	3	
1	2	1	3	3	1	5	
2	3	2	0	3	4	4	

Explore

```
sheets_url = paste0("https://docs.google.com/spreadsheets/d/",
                    "1WBrH655fxqKW1QqvD5hnqvWEIvRzDJcKEgjjFeYxeM")
```

```
gsurl1 = gs_url(sheets_url)
```

```
Warning: `as_data_frame()` is deprecated as of tibble 2.0.0.  
Please use `as_tibble()` instead.  
The signature and semantics have changed, see `?as_tibble`.  
This warning is displayed once every 8 hours.  
Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
dat = gs_read(gsurl1)  
date_read = lubridate::today()  
head(dat)
```

```
# A tibble: 6 x 12  
  Git Github     R Rstudio `Reproducible R... `R markdown` `Data import`  
  <dbl> <dbl> <dbl>    <dbl>                      <dbl>          <dbl>          <dbl>  
1     9     9     9      9                         9             9            10  
2     1     1     5      5                         7             5             4  
3     0     0     2      3                         3             0             3  
4     0     1     1      1                         1             1             1  
5     0     2     7      7                         7             5             8  
6     2     2     3      2                         0             0             1  
# ... with 5 more variables: `Web scraping` <dbl>, `Data cleaning` <dbl>,  
#   dplyr <dbl>, Bioconductor <dbl>, `Regular expressions` <dbl>
```

Googlesheets4 - API v4 Google Sheets Package

What if I don't want it public?

```
library(googlesheets4)
# May be necessary on rstudio.cloud
options(httr_oob_default=TRUE)
# Will ask you to log in
out = read_sheet(sheets_url)
```

Can also save and load a token

```
token = readr::read_rds("googledrive_token.rds")
library(googledrive)
drive_auth(token = token) # could also use googlesheets4::gs4_auth
library(googlesheets4)
out = read_sheet(sheets_url)
head(out)
```

```
# A tibble: 6 x 12
  Git Github R Rstudio `Reproducible R... `R markdown` `Data import`
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1     9     9     9     9     9     9     9     10
2     1     1     5     5     7     5     4
3     0     0     2     3     3     0     3
4     0     1     1     1     1     1     1
5     0     2     7     7     7     5     8
6     2     2     3     2     0     0     1
# ... with 5 more variables: `Web scraping` <dbl>, `Data cleaning` <dbl>,
#   dplyr <dbl>, Bioconductor <dbl>, `Regular expressions` <dbl>
```

Google Sheets

<https://SISBIB.github.io/Module1/lab-sheets-lab.Rmd>

JSON: JavaScript Object Notation

Lists of stuff

Why JSON matters

The screenshot shows a browser window displaying the GitHub REST API documentation at docs.github.com/en/rest/reference/search. The left sidebar contains a list of API endpoints, and the main content area shows a "Default response" example.

Default response

Status: 200 OK

```
{  
  "total_count": 7,  
  "incomplete_results": false,  
  "items": [  
    {  
      "name": "classes.js",  
      "path": "src/attributes/classes.js",  
      "sha": "d7212f9dee2dcc18f084d7df8f417b80846ded5a",  
      "url": "https://api.github.com/repositories/167174/contents/src/attrib  
      "git_url": "https://api.github.com/repositories/167174/git/blobs/d7212  
      "html_url": "https://github.com/jquery/jquery/blob/825ac3773694e0cd23e  
      "repository": {  
        "id": 167174,  
        "node_id": "MDEwOlJlcG9zaXRvcnkxNjcxNzQ=",  
        "name": "jquery",  
        "full_name": "jquery/jquery",  
        "owner": {  
          "login": "jquery",  
          "id": 70142,  
          "node_id": "MDQ6VXNlcjcwMTQy",  
          "avatar_url": "https://0.gravatar.com/avatar/6906f317a4733f4379b06  
          "gravatar_id": "",  
          "url": "https://api.github.com/users/jquery",  
          "html_url": "https://github.com/jquery",  
          "followers_url": "https://api.github.com/users/iauerv/followers".  
        }  
      }  
    }  
  ]  
}
```

<https://docs.github.com/en/rest/reference/search>

```
#install.packages("jsonlite")
library(jsonlite)
jsonData <- fromJSON("https://api.github.com/users/jtleek/repos")
head(jsonData)
```

	id	node_id	name		
1	155565363	MDEwOlJlcG9zaXRvcnkxNTU1NjUzNjM=	2018		
2	264786491	MDEwOlJlcG9zaXRvcnkyNjQ3ODY0OTE=	ads2020		
3	101394164	MDEwOlJlcG9zaXRvcnkxMDEzOTQxNjQ=	advdatasci		
4	111447948	MDEwOlJlcG9zaXRvcnkxMTE0NDc5NDg=	advdatasci-project		
5	47568815	MDEwOlJlcG9zaXRvcnk0NzU2ODgxNQ==	advdatasci-swirl		
6	41645119	MDEwOlJlcG9zaXRvcnk0MTY0NTExOQ==	advdatasci15		
	full_name	private	owner.login	owner.id	owner.node_id
1	jtleek/2018	FALSE	jtleek	1571674	MDQ6VXNlcjE1NzE2NzQ=
2	jtleek/ads2020	FALSE	jtleek	1571674	MDQ6VXNlcjE1NzE2NzQ=
3	jtleek/advdatasci	FALSE	jtleek	1571674	MDQ6VXNlcjE1NzE2NzQ=
4	jtleek/advdatasci-project	FALSE	jtleek	1571674	MDQ6VXNlcjE1NzE2NzQ=
5	jtleek/advdatasci-swirl	FALSE	jtleek	1571674	MDQ6VXNlcjE1NzE2NzQ=
6	jtleek/advdatasci15	FALSE	jtleek	1571674	MDQ6VXNlcjE1NzE2NzQ=
	owner.avatar_url	owner.gravatar_id			
1	https://avatars2.githubusercontent.com/u/1571674?v=4				
2	https://avatars2.githubusercontent.com/u/1571674?v=4				
3	https://avatars2.githubusercontent.com/u/1571674?v=4				
4	https://avatars2.githubusercontent.com/u/1571674?v=4				
5	https://avatars2.githubusercontent.com/u/1571674?v=4				
6	https://avatars2.githubusercontent.com/u/1571674?v=4				
	owner.url	owner.html_url			
1	https://api.github.com/users/jtleek	https://github.com/jtleek			
2	https://api.github.com/users/jtleek	https://github.com/jtleek			
3	https://api.github.com/users/jtleek	https://github.com/jtleek			
4	https://api.github.com/users/jtleek	https://github.com/jtleek			

Data frame structure from JSON

```
dim(jsonData)
```

```
[1] 30 73
```

```
head(jsonData$name)
```

```
[1] "2018"                 "ads2020"             "advdatasci"  
[4] "advdatasci-project"  "advdatasci-swirl"   "advdatasci15"
```

```
class(jsonData$owner) #Some of the columns is a data frame!
```

```
[1] "data.frame"
```

```
dim(jsonData$owner); names(jsonData$owner)
```

```
[1] 30 18
```

```
[1] "login"                  "id"                   "node_id"  
[4] "avatar_url"             "gravatar_id"          "url"  
[7] "html_url"               "followers_url"        "following_url"  
[10] "gists_url"              "starred_url"          "subscriptions_url"  
[13] "organizations_url"     "repos_url"            "events_url"  
[16] "received_events_url"   "type"                "site_admin"
```

JSON Lab

<https://SISBIB.github.io/Module1/lab.Rmd>

Web Scraping

This is data

<http://bowtie-bio.sourceforge.net/recount/>

← → ⌂ ⓘ Not Secure | bowtie-bio.sourceforge.net/recount/  R 🍎 🔎 🛡️ ⓘ

Please note that to use the ExpressionSets below, you will need to install Bioconductor and run the command library(Bioconductor)

» The Datasets

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	not published, but publicly available here	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human	41	834,584,950	link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and females
maqc	20167110	human	14 (technical)** 2 (biological)	71,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison
wang	18978772	human	22	223,929,919	link	link	link	tissue comparison
								control vs

View the source

Please note that to use the expressionsets below, you will need to install Bioconductor and run the command library(BioBase)

» The Datasets

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	Expotype	Notes
bodymap	not published, but publicly available here	human	19	2,197,622,796	link	Illumina Human BodyMap 2.0 - - tissue comparison
cheung	20856902	human	41	834,584,950	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	liver; males and females
maqc	20167110	human	14 (technical)** 2 (biological)	71,970,164	original pooled	original pooled original pooled experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link link HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link link HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link link cell type comparison
wang	18978772	human	22	223,929,919	link	link link tissue comparison
katz.mouse	21057496	mouse	4	14,368,471	link	link link control vs. CUG-BP1

What the computer sees

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <script src="sorttable.js" type="text/javascript"></script>
5 <title>ReCount: analysis-ready RNA-seq gene count datasets</title>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 <link rel="stylesheet" type="text/css" href="css/style.css" media="screen" />
8 <script type="text/javascript">
9
10 var _gaq = _gaq || [];
11 _gaq.push(['_setAccount', 'UA-26478269-2']);
12 _gaq.push(['_trackPageview']);
13
14 (function() {
15   var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
16   ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
17   var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
18 })();
19
20 </script>
21
22 </head>
23
24 <body class="c20">
25 <div id="wrap">
26   <div id="top">
27     <div class="lefts">
28       <table width="100%" cellpadding="2">
29         <tr><td>
30           <a href=".index.shtml"><h1>ReCount</h1></a>
31           <h2>A multi-experiment resource of analysis-ready RNA-seq gene count datasets</h2>
32           </td><td align="right" valign="middle">
33             <h1><a href="http://www.biostat.jhsph.edu/"></a>&ampnbsp&ampnbsp</h1>
34           </td></tr>
35         </table>
36       </div>
37     </div>
38
39     <div id="subheader">
40       <p><b>There is now <a href="https://jhbiostatistics.shinyapps.io/recount/">a new version of recount</a> that provides processed and summarized express data for nearly 60,000 human RNA-seq samples from the Sequence Read Archive (SRA). The <a href="https://github.com/leekgroup/recount">associated Bioconductor package</a> provides a convenient API for querying, downloading, and analyzing the data. Each processed study consists of meta- and phenot data, the expression levels of genes and their underlying exons and splice junctions, and corresponding genomic annotation. See <a
```

Ways to see the source

Chrome:

1. right click on page
2. select “view source”

Firefox:

1. right click on page
2. select “view source” Microsoft Edge:
3. right click on page
4. select “view source”

Safari

1. click on “Safari”
2. select “Preferences”
3. go to “Advanced”
4. check “Show Develop menu in menu bar”
5. click on “Develop”
6. select “show page source”
7. alternatively to 5./6., right click on page and select “view source”

<https://github.com/simonmunzert/rscraping-jsm-2016/blob/c04fd91fec711df65c838e07723125155a7f2cda/02-scraping-with-rvest.r>

Inspect element

Not Secure | bowtie-bio.sourceforge.net/recount/

Please note that to use the expressionsets below, you will need to install Bioculator and run the command library (biocore).

✖ The Datasets

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	Expression	Back	Forward	Reload	Notes
maqc	20167110	human	14 (technical)** 2 (biological)	71,970,164	original pooled	Save As...	Print...	Cast...	experiment: MAQC-2
modencodefly	21179090	fly	147 (technical)** 30 (biological)	2,278,788,557	original pooled	Translate to English	View Page Source	View Frame Source	developmental time course
modencodeworm	19181841	worm	46	1,451,119,823	link	Reload Frame	Inspect	Speech	developmental time course
hammer	20452967	rat	8	158,178,477	link	link	link	▶	experimental vs. control at 2 time points
nagalakshmi	18451266	yeast	4	7,688,602	link	link	link		priming technique comparison
bottomly	21455293	mouse	21	343,445,340	link	link	link		2 inbred mouse strains
yang	20363980	mouse	1	27,883,862	link	link	link		hybrid cell line, X always inactive
trapnell	20436464	mouse	4	111,376,152	link	link	link		time course
mortazavi	18516045	mouse	3	61,732,881	link	link	link		tissue comparison

A context menu is open over the "Inspect" button in the "modencodeworm" row. The menu items are: Back, Forward, Reload, Save As..., Print..., Cast..., Translate to English, View Page Source, View Frame Source, Reload Frame, Inspect (which is highlighted in blue), and Speech. Below the menu, there are three links: link, link, and link.

Copy XPath

The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. A context menu is open over a table row with the ID 'recounttab'. The menu options include:

- Copy
- Cut element
- Copy element
- Paste element
- Copy outerHTML
- Copy selector
- Copy JS path
- Copy styles
- Copy XPath** (highlighted in blue)
- Copy full XPath
- Store as global variable
- Speech

The table data is as follows:

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	not published, but publicly available here	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 - tissue comparison
cheung	20856902	human	41	834,584,950	link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
gilad	2000			41,356,738	link	link	link	liver; males and females

The DevTools sidebar shows the DOM tree with the current node being the table with ID 'recounttab'. The right panel displays the element's styles, computed styles, event listeners, and DOM breakpoints.

rvest package

```
recount_url = "http://bowtie-bio.sourceforge.net/recount/"
# install.packages("rvest")
library(rvest)
htmlfile = read_html(recount_url)

nds = html_nodes(htmlfile,
xpath='//*[@id="recounttab"]/table')
dat = html_table(nds)
dat = as.data.frame(dat)
head(dat)
```

	X1	X2	X3
1	Study	PMID	Species
2	bodymap not published, but publicly available here		human
3	cheung	20856902	human
4	core	19056941	human
5	gilad	20009012	human
6	maqc	20167110	human
	X4	X5	
1	Number of biological replicates	Number of uniquely aligned reads	
2	19	2,197,622,796	
3	41	834,584,950	
4	2	8,670,342	
5	6	41,356,738	
6	14 (technical)** 2 (biological)	71,970,164	
	X6	X7	X8
1	ExpressionSet link	Count table link	Phenotype table link
2			

Little cleanup

```
colnames(dat) = as.character(dat[1,])
dat = dat[-1,]
head(dat)
```

	Study	PMID	Species
2	bodymap not published, but publicly available here		human
3	cheung	20856902	human
4	core	19056941	human
5	gilad	20009012	human
6	maqc	20167110	human
7	montgomery	20220756	human
	Number of biological replicates	Number of uniquely aligned reads	
2		19	2,197,622,796
3		41	834,584,950
4		2	8,670,342
5		6	41,356,738
6	14 (technical)** 2 (biological)		71,970,164
7		60	*886,468,054
	ExpressionSet	Count table	Phenotype table
2	link	link	link
3	link	link	link
4	link	link	link
5	link	link	link
6	original pooled	original pooled	original pooled
7	link	link	link
			Notes
2	Illumina Human BodyMap 2.0 -- tissue comparison		
3		HapMap - CEU	

<http://motherboard.vice.com/read/70000-okcupid-users-just-had-their-data-published>

MOTHERBOARD
TECH BY VICE

70,000 OkCupid Users Just Had Their Data Published

Just because data is sort-of public, doesn't mean that it's ethical to collect en masse.

By [Joseph Cox](#)

<https://www.theguardian.com/science/2012/may/23/text-mining-research-tool-forbidden>

theguardian.com/science/2012/may/23/text-mining-research-tool-forbidden

Peer review and scientific publishing

Text mining: what do publishers have against this hi-tech research tool?

Researchers push for end to publishers' default ban on computer scanning of tens of thousands of papers to find links between genes and diseases



Alok Jha, Science correspondent

Wed 23 May 2012 11.27 EDT

[f](#) [t](#) [e](#)

3 62

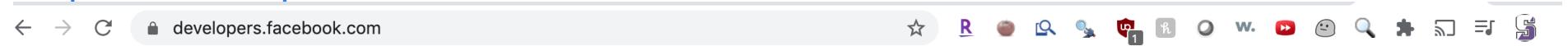


▲ A DNA sequence is examined: text mining allows links to be discovered that would never have been noticed during manual research. Photograph: Simon Dolan/Alamy

APIs

Application Programming Interfaces

<https://developers.facebook.com/>



DEVELOPER TOOLS

Take a closer look at the products we offer.

Messenger

Build lasting customer relationships through conversation.

 Learn more

Instagram

Create tools for businesses, creators, and people to enhance the Instagram experience.

 Learn more

Business Tools

Build and scale your business across the Facebook family of apps.

 Learn more

Open Source

Artificial Intelligence

AR/VR

In biology too!

<http://www.ncbi.nlm.nih.gov/books/NBK25501/>

The screenshot shows a web browser displaying the "Entrez Programming Utilities Help" page. The URL in the address bar is [ncbi.nlm.nih.gov/books/NBK25501/](http://www.ncbi.nlm.nih.gov/books/NBK25501/). The page header includes the NCBI logo and navigation links for "Views", "PubReader", "Print View", "Cite this Page", and "PDF version of this title (1.8M)". The main content area features a sidebar with links to "Entrez Programming Utilities Help", "NCBI Help Manual", and "National Center for Biotechnology Information". Below this is a search bar with the placeholder "Search this book". The main text area is titled "Introduction to the E-utilities" and contains two bullet points: "YouTube E-utilities Introduction" and "Please see the [Release Notes](#) for details and changes.". A detailed description follows, explaining the E-utilities as a set of eight server-side programs providing a stable interface to the Entrez system. The page also includes a "Contents" section with a link to "E-utilities Quick Start" and a note about the last update. At the bottom, there are "Expand All" and "Collapse All" buttons, and a footer with links to "Recent Activity" and "Turn Off Clear".

Entrez Programming Utilities Help

Bethesda (MD): National Center for Biotechnology Information (US); 2010-.
Copyright and Permissions

Search this book

Views
PubReader
Print View
Cite this Page
PDF version of this title (1.8M)

Other titles in this collection
NCBI Help Manual

Related information
NLM Catalog

Recent Activity
Turn Off Clear

Entrez Programming Utilities Help

A fully automated pipeline for brain structure segmentation in multiple sclerosis...

Validation of Accelerometer Wear and Nonwear Time Classification Algorithm

The Evolution of Earned, Transparent, and Quantifiable Faculty Salary Compensation...

Evaluating the Impact of Intensity Normalization on MR Image

Contents

E-utilities Quick Start

Created: December 12, 2008; Last Update: October 24, 2018.

Expand All Collapse All

Step 0: Did someone do this already

<https://ropensci.org/packages/>

The screenshot shows a web browser displaying the ropensci.org/packages/ page. The page features a search bar at the top with the placeholder text "Search by: name, maintainer, or keyword". Below the search bar is a table with columns: NAME, MAINTAINER, DESCRIPTION, and STATUS. The STATUS column includes icons for CRAN and GitHub, along with a speech bubble icon for each package. The packages listed are:

NAME	MAINTAINER	DESCRIPTION	STATUS
gert	Jeroen Ooms	Simple Git Client for R	CRAN
tic	Kirill Müller	CI-Agnostic Workflow Definitions	CRAN
rotl	Francois Michonneau	Interface to the 'Open Tree of Life' API	CRAN
colpluz	Scott Chamberlain	Catalogue of Life Plus Client	CRAN
crul	Scott Chamberlain	HTTP Client	CRAN
c14bazAAR	Clemens Schmid	Download and Prepare C14 Dates from Different Source Databases	CRAN
stplanr	Robin Lovelace	Sustainable Transport Planning	CRAN
webchem	Tamás Stirling	Chemical Information from the Web	CRAN

Do it yourself: read the Docs

<https://docs.github.com/en/rest>

The screenshot shows a web browser displaying the GitHub REST API documentation at <https://docs.github.com/en/rest>. The page has a dark blue sidebar on the left containing navigation links for various GitHub products like REST API, OVERVIEW, and REFERENCE. The main content area features a banner at the top announcing unified product documentation with links to the REST API, GraphQL API, and Developers blog. Below the banner, there's a search bar and language selection (English). The main title is "GitHub REST API", followed by a brief description: "You can use the GitHub REST API to create calls to get the data you need to integrate with GitHub." Under the title, there are two sections: "REST API overview" and "Reference". The "REST API overview" section describes the purpose of the API, and the "Reference" section provides a link to learn about resources available in the GitHub REST API.

We've unified all of GitHub's product documentation in one place! Check out the content for [REST API](#), [GraphQL API](#), and [Developers](#). Learn more on the [GitHub blog](#).

English ▾

Search topics, products...

Article version: GitHub.com ▾

GitHub REST API

You can use the GitHub REST API to create calls to get the data you need to integrate with GitHub.

REST API overview

Learn about resources, libraries, previews and troubleshooting for GitHub's REST API.

Reference

View reference documentation to learn about the resources available in the GitHub REST API.

Read the docs

<https://docs.github.com/en/rest/reference/repos>

The screenshot shows a web browser displaying the GitHub REST API documentation for repositories. The URL in the address bar is <https://docs.github.com/en/rest/reference/repos>. The page title is "Repositories". On the left, there is a sidebar with a dark blue background containing a list of links: Code Scanning, Codes of conduct, Emojis, Gists, Git database, Gitignore, Interactions, Issues, Licenses, Markdown, Meta, Migrations, OAuth Authorizations, Organizations, Projects, Pulls, Rate limit, Reactions, Repositories (which is the current page), SCIM, Search, Teams, and Users. The main content area has a white background. At the top, it shows the REST API / Reference / Repositories path. Below that is a large section titled "Repositories". Under this, there is a sub-section titled "List organization repositories" with the description "Lists repositories for the specified organization." It includes a "GET /orgs/{org}/repos" endpoint. Below this are sections for "Parameters" and "Responses". The "Parameters" table has columns for Name, Type, In, and Description. It lists three parameters: "accept" (string, header, recommended, with a note about preview notices), "org" (string, path), and "type" (string, query, with a description of its function). To the right of the main content, there is a sidebar titled "In this article" which lists various repository-related endpoints. The "Article version" is shown as "GitHub.com".

Code Scanning
Codes of conduct
Emojis
Gists
Git database
Gitignore
Interactions
Issues
Licenses
Markdown
Meta
Migrations
OAuth Authorizations
Organizations
Projects
Pulls
Rate limit
Reactions
Repositories
SCIM
Search
Teams
Users

REST API / Reference / Repositories

Article version: GitHub.com

Repositories

List organization repositories

Lists repositories for the specified organization.

GET /orgs/{org}/repos

Parameters

Name	Type	In	Description
accept	string	header	Setting to <code>application/vnd.github.v3+json</code> is recommended See preview notices.
org	string	path	
type	string	query	Specifies the types of repositories you want returned. Can be one of <code>all</code> , <code>public</code> , <code>private</code> , <code>forks</code> , <code>sources</code> , <code>member</code> , <code>internal</code> . Default: <code>all</code> . If your organization is associated with an enterprise account using GitHub

In this article

- List organization repositories
- Create an organization repository
- Get a repository
- Update a repository
- Delete a repository
- Enable automated security fixes
- Disable automated security fixes
- List repository contributors
- Create a repository dispatch event
- List repository languages
- List repository tags
- List repository teams
- Get all repository topics
- Replace all repository topics
- Transfer a repository
- Check if vulnerability alerts are enabled for a repository
- Enable vulnerability alerts
- Disable vulnerability alerts
- Create a repository using a template

Read the docs

<https://docs.github.com/en/rest/reference/rate-limit>

The screenshot shows a web browser displaying the GitHub REST API Rate Limit documentation at <https://docs.github.com/en/rest/reference/rate-limit>. The page has a dark blue sidebar on the left containing a list of links to other API documentation sections. The main content area has a light gray background. At the top of the content area is a navigation bar with icons for star, R, search, and other functions. The main title 'Rate limit' is displayed in a large, bold, dark font. Below the title, a paragraph of text explains the REST API overview documentation describes the [rate limit rules](#). It states that you can check your current rate limit status at any time using the Rate Limit API described below. A section titled 'Understanding your rate limit status' follows, explaining that the Search API has a [custom rate limit](#), separate from the rate limit governing the rest of the REST API. It also notes that the GraphQL API has a [custom rate limit](#) that is separate from and calculated differently than rate limits in the REST API. Below this, it says that for these reasons, the Rate Limit API response categorizes your rate limit. It then describes the four objects under `resources`: `core` (providing status for non-search-related resources), `search` (providing status for the [Search API](#)), `graphql` (providing status for the [GraphQL API](#)), and `integration_manifest` (providing status for the [GitHub App Manifest code conversion](#) endpoint). At the bottom, there is a note about headers and values in the rate limit response, linking to the ["Rate limiting."](#) page.

Code Scanning
Codes of conduct
Emojis
Gists
Git database
Gitignore
Interactions
Issues
Licenses
Markdown
Meta
Migrations
OAuth Authorizations
Organizations
Projects
Pulls
Rate limit
Reactions
Repositories
SCIM
Search
Teams
Users

Rate limit

The REST API overview documentation describes the [rate limit rules](#). You can check your current rate limit status at any time using the Rate Limit API described below.

Understanding your rate limit status

The Search API has a [custom rate limit](#), separate from the rate limit governing the rest of the REST API. The GraphQL API also has a [custom rate limit](#) that is separate from and calculated differently than rate limits in the REST API.

For these reasons, the Rate Limit API response categorizes your rate limit. Under `resources`, you'll see four objects:

- The `core` object provides your rate limit status for all non-search-related resources in the REST API.
- The `search` object provides your rate limit status for the [Search API](#).
- The `graphql` object provides your rate limit status for the [GraphQL API](#).
- The `integration_manifest` object provides your rate limit status for the [GitHub App Manifest code conversion](#) endpoint.

For more information on the headers and values in the rate limit response, see ["Rate limiting."](#)

In this article

[Understanding your rate limit status](#)
[Get rate limit status for the authenticated user](#)

Read the docs

Example

Suppose you want to find the definition of the `addClass` function inside `jQuery`. Your query would look something like this:

```
https://api.github.com/search/code?q=addClass+in:file+language:js+repo:jquery/jquery
```

Here, we're searching for the keyword `addClass` within a file's contents. We're making sure that we're only looking in files where the language is JavaScript. And we're scoping the search to the `repo:jquery/jquery` repository.

A dissected example

[https://api.github.com/search/repositories?
q=created:2014-08-13+language:r+-user:cran](https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran)

The base URL

[https://api.github.com/search/repositories?
q=created:2014-08-13+language:r+-user:cran](https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran)

The Path: Search repositories

[https://api.github.com/search/repositories?
q=created:2014-08-13+language:r+-user:cran](https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran)

Create a query - pass the `q` parameter

[`https://api.github.com/search/repositories?`](https://api.github.com/search/repositories?)
`q=created:2014-08-13+language:r+-user:cran`

Date repo was created

[https://api.github.com/search/repositories?
q=created:2014-08-13+language:r+-user:cran](https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran)

Language repo is in

[https://api.github.com/search/repositories?
q=created:2014-08-13+language:r+-user:cran](https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran)

Ignore repos from “cran”

[https://api.github.com/search/repositories?
q=created:2014-08-13+language:r+-user:cran](https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran)

```
#install.packages("httr")
library(httr)

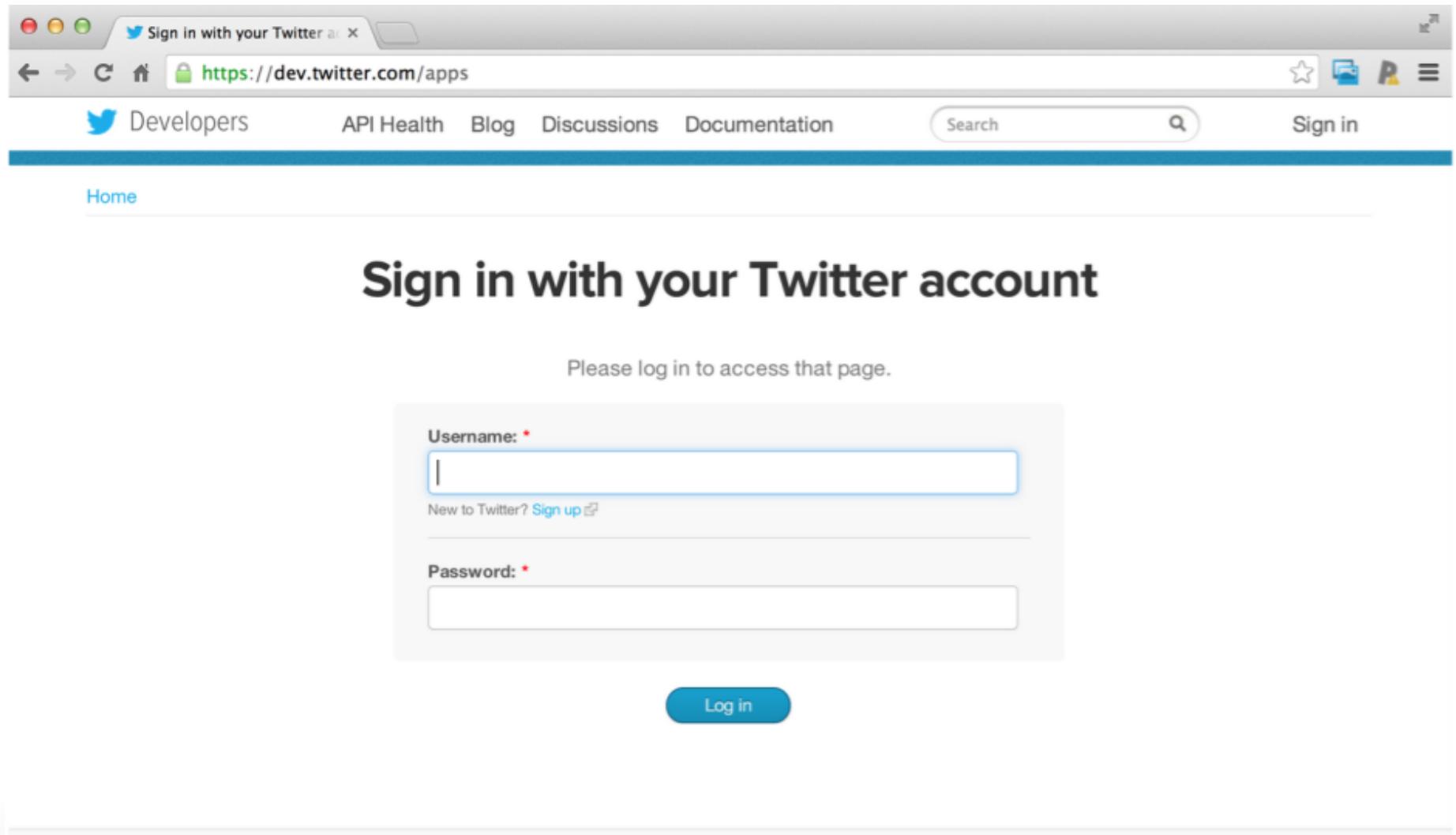
query_url = paste0("https://api.github.com/", "search/repositories",
                  "?q=created:2014-08-13", "+language:r", "+-user:cran")

req = GET(query_url)
names(content(req))
```

```
[1] "total_count"      "incomplete_results" "items"
```

Not all APIs are “open”

<https://apps.twitter.com/>



A screenshot of a web browser window displaying the Twitter API sign-in page. The URL in the address bar is <https://dev.twitter.com/apps>. The page title is "Sign in with your Twitter account". The content area includes a "Username:" field with a placeholder "New to Twitter? [Sign up](#)", a "Password:" field, and a "Log in" button.

Sign in with your Twitter account

Please log in to access that page.

Username: *

New to Twitter? [Sign up](#)

Password: *

Log in

(see also `twitteR` package)

```
myapp = oauth_app("twitter",
                  key="yourConsumerKeyHere", secret="yourConsumerSecretHere")
sig = sign_oauth1.0(myapp,
                     token = "yourTokenHere",
                     token_secret = "yourTokenSecretHere")
homeTL = GET("https://api.twitter.com/1.1/statuses/home_timeline.json", sig)

# But you can get cool data
json1 = content(homeTL)
json2 = jsonlite::fromJSON(toJSON(json1))
json2[1,1:4]
```

	created_at	id	id_str
1	Mon Jan 13 05:18:04 +0000 2014	4.225984e+17	422598398940684288

1 Now that P. Norvig's regex golf IPython notebook hit Slashdot, let's see if