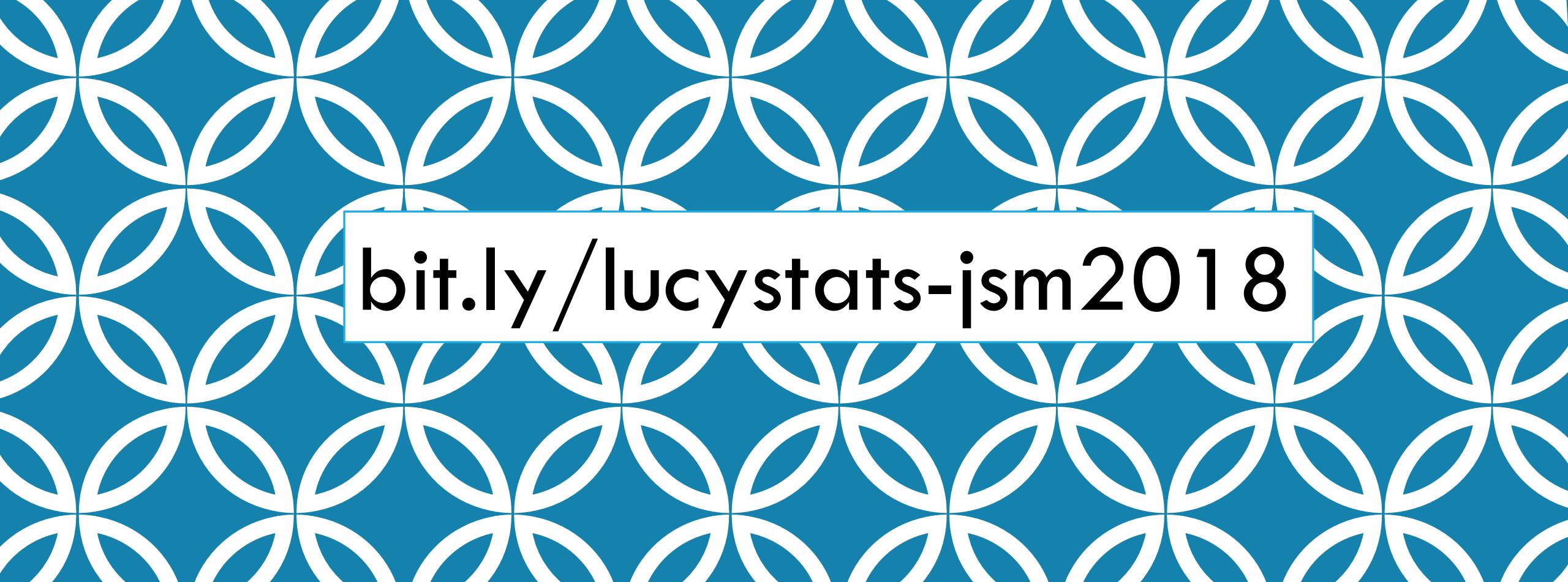


HARNESSING THE POWER OF THE WEB VIA R CLIENTS FOR WEB APIs

Lucy D'Agostino McGowan, PhD
*Johns Hopkins Bloomberg School
of Public Health*
[@LucyStats](https://twitter.com/LucyStats)



bit.ly/lucystats-jsm2018

HARNESSING THE POWER OF THE WEB VIA R CLIENTS FOR WEB APIs

Lucy D'Agostino McGowan, PhD
*Johns Hopkins Bloomberg School
of Public Health*
[@LucyStats](https://twitter.com/LucyStats)

OUTLINE

👉 What is an API?

OUTLINE

- 👉 What is an API?
- 👉 How are APIs accessed from R

OUTLINE

- 👉 What is an API?
- 👉 How are APIs accessed from R
- 👉 Case studies

OUTLINE

- 👉 What is an API?
- 👉 How are APIs accessed from R
- 👉 Case studies

API

API

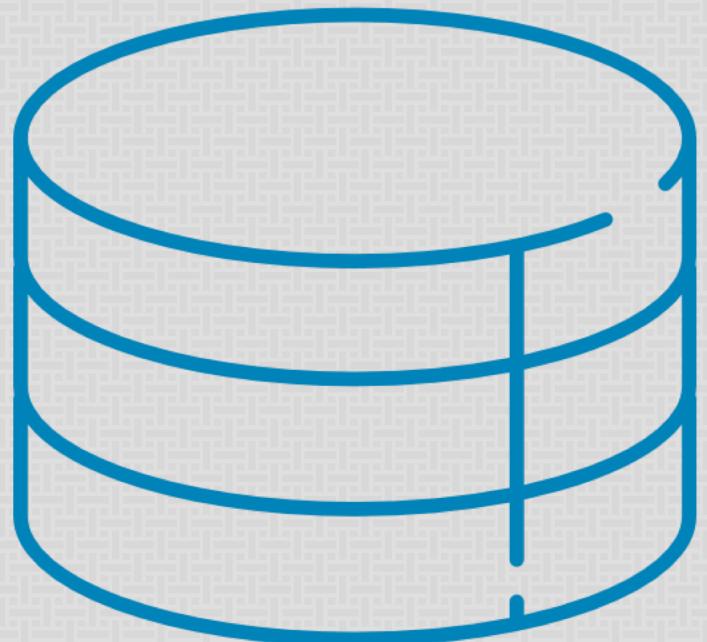
application

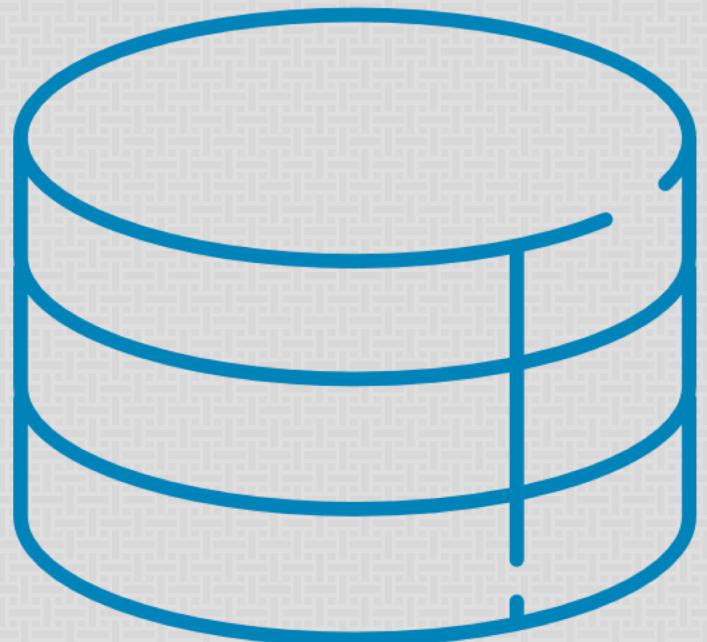
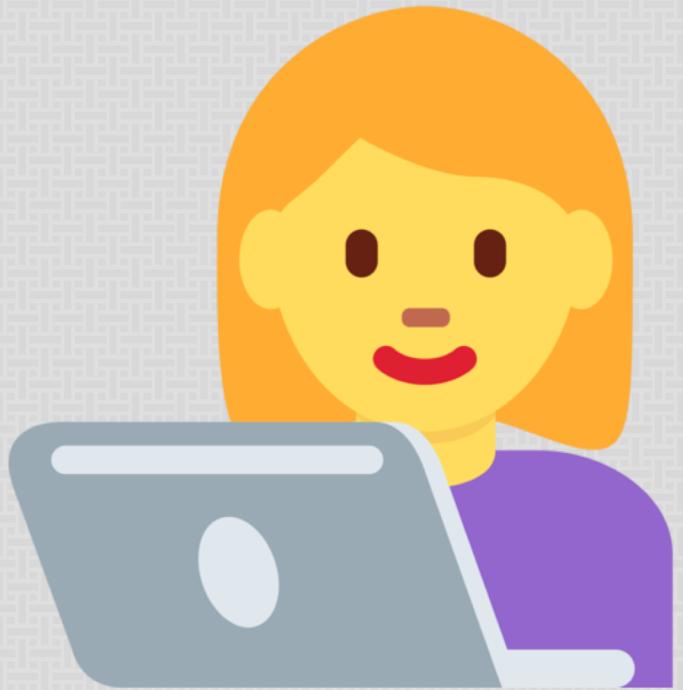
API

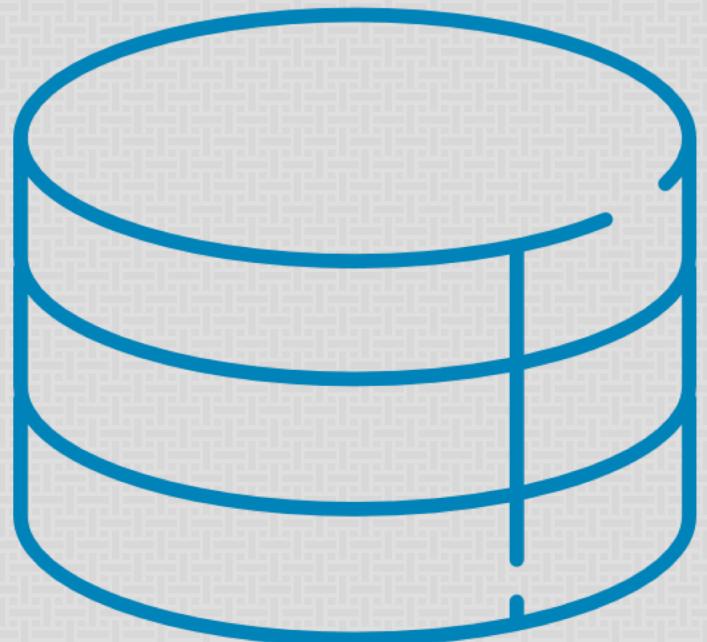
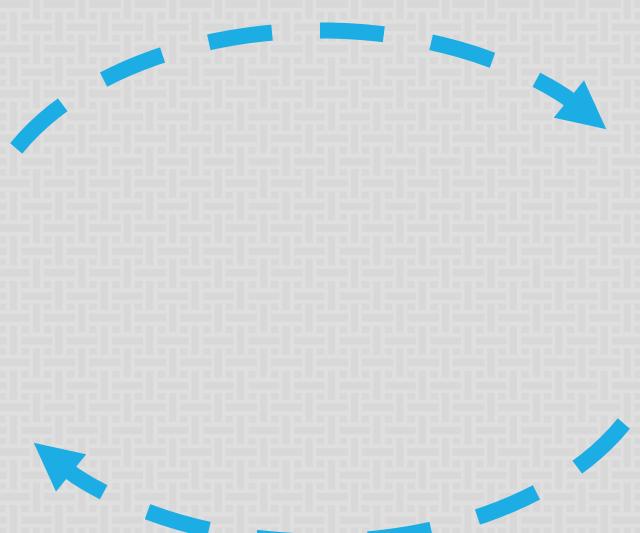
application programming

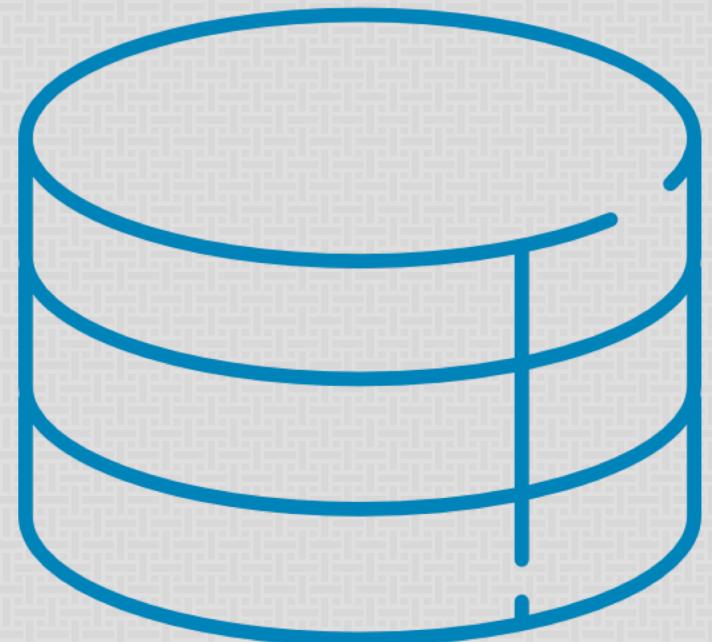
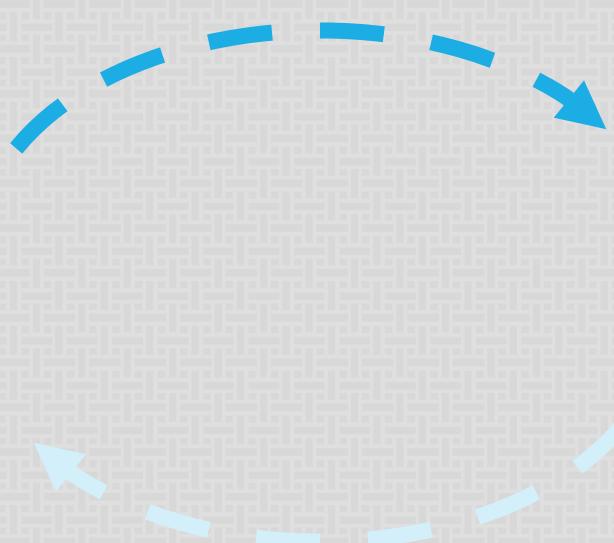
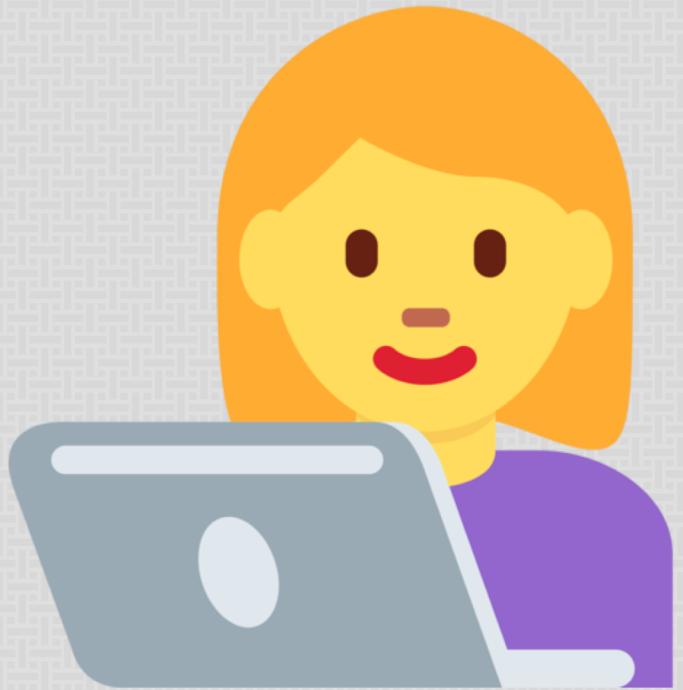
API

application programming interface









REQUEST VIA HTTP METHODS

HyperText Transfer Protocol

REQUEST VIA HTTP METHODS

GET

POST

PUT

PATCH

DELETE

HyperText Transfer Protocol

REQUEST VIA HTTP METHODS

GET



POST

PUT

PATCH

DELETE

REQUEST VIA HTTP METHODS

GET



POST



PUT

PATCH

DELETE

REQUEST VIA HTTP METHODS

GET



POST



PUT



PATCH

DELETE

REQUEST VIA HTTP METHODS

GET
POST
PUT
PATCH
DELETE



REQUEST VIA HTTP METHODS

GET



POST



PUT



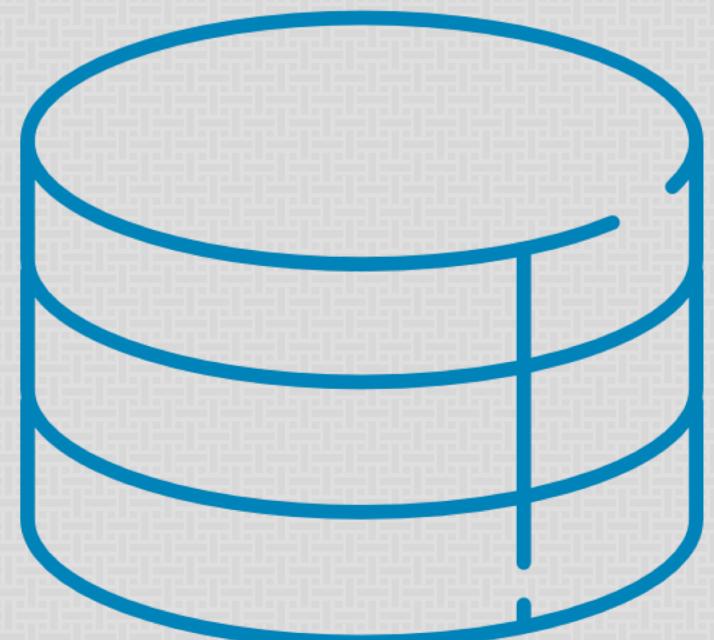
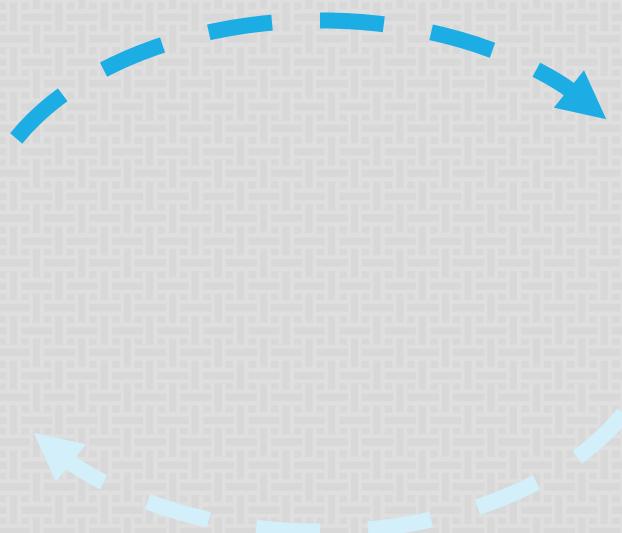
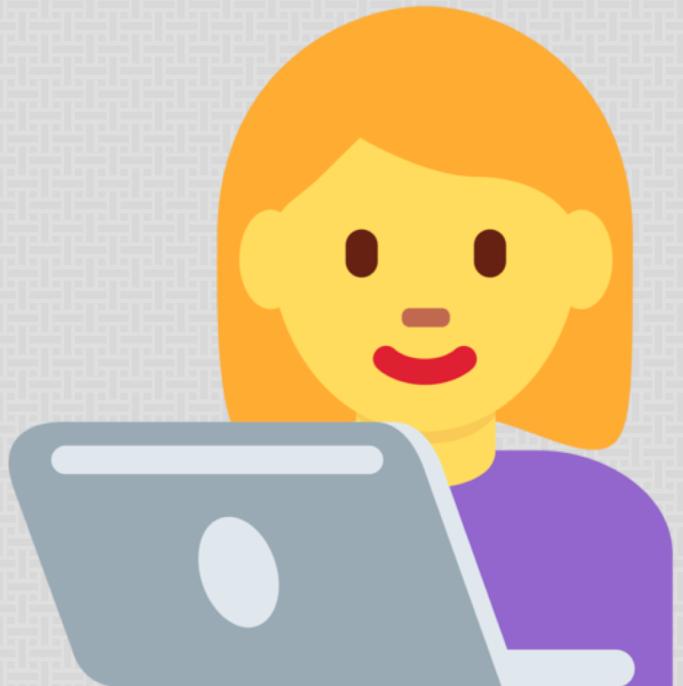
PATCH



DELETE

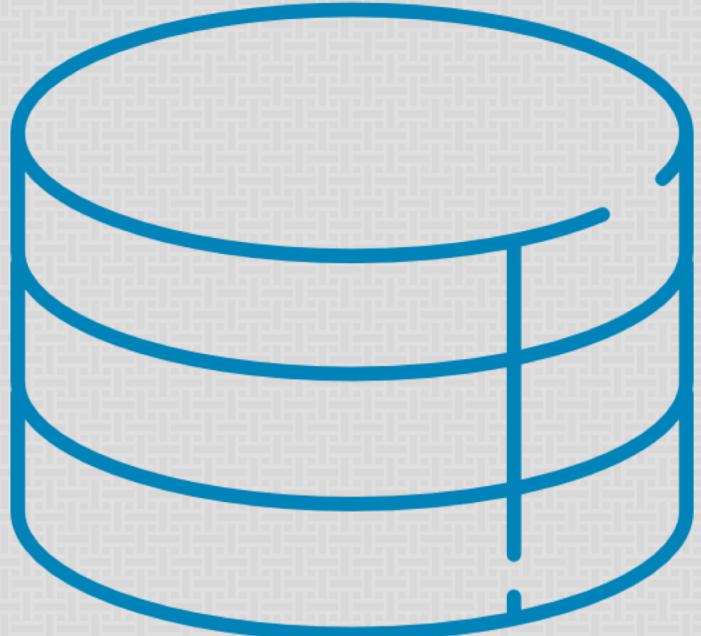


HTTP Methods

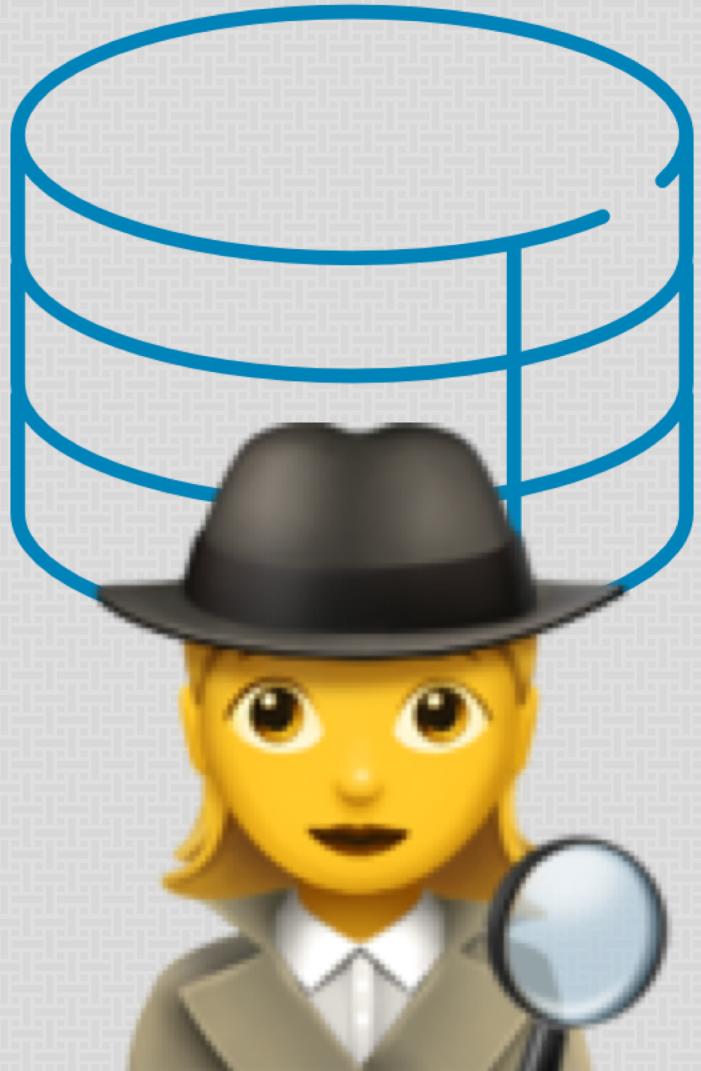




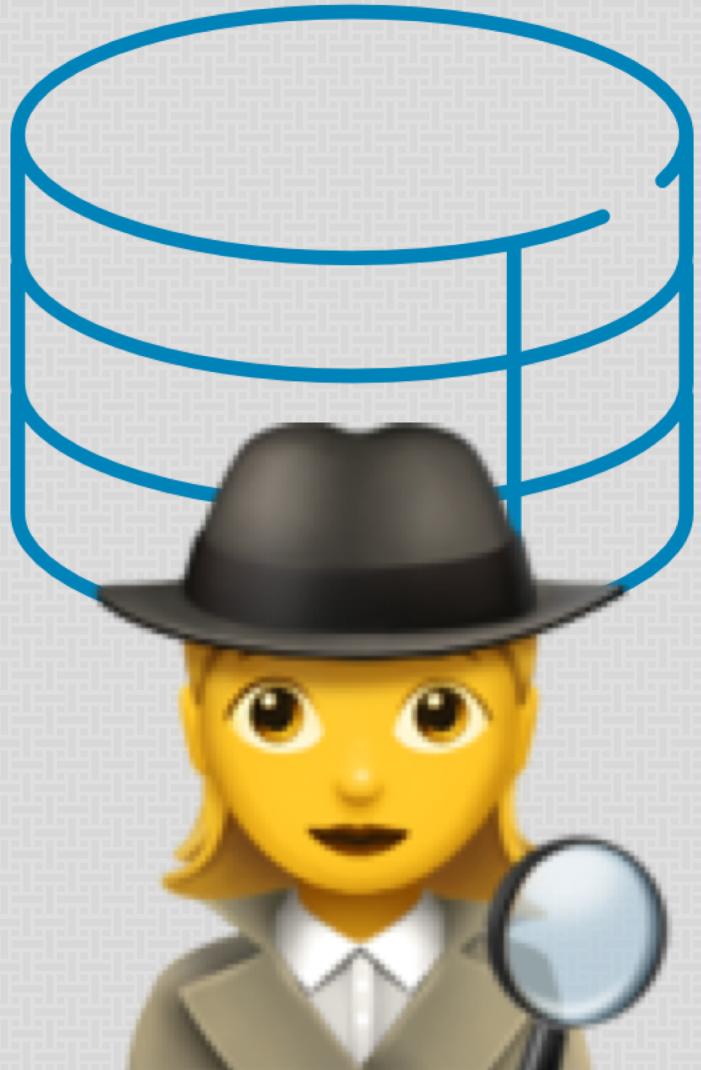
Choose Method



Built a set of urls
that return **data**

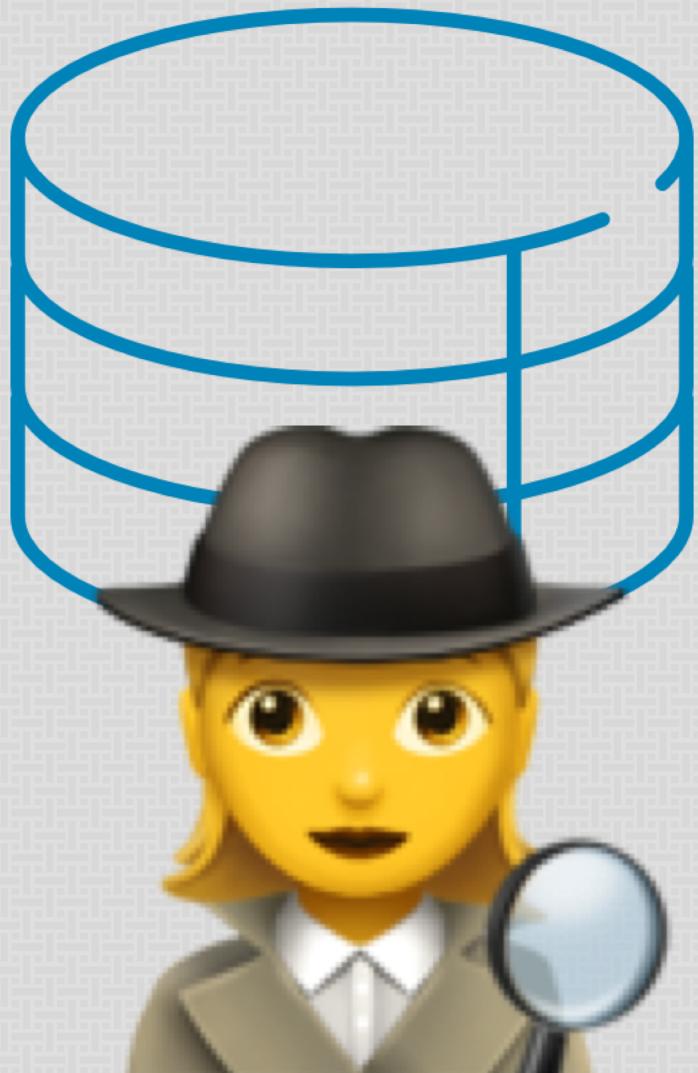


Built a set of urls
that return data



Built a set of urls
that return data

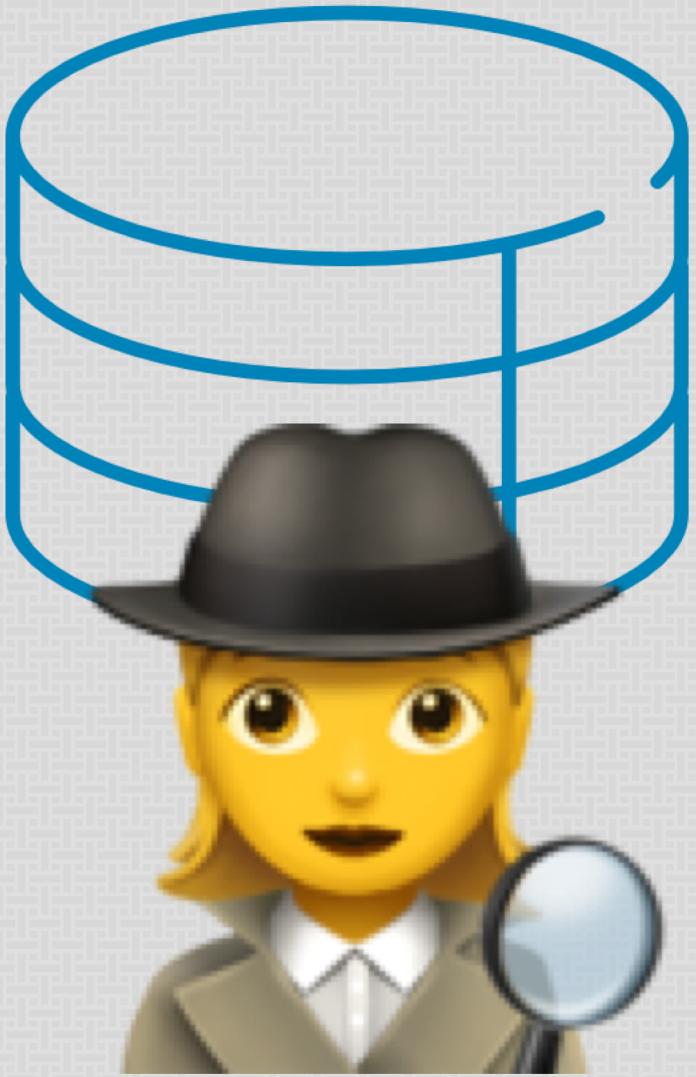
The screenshot shows the Twitter Developer Documentation website. At the top, there is a purple navigation bar with links for 'Developer', 'Use cases', 'Products', 'Docs', and 'More'. On the far right of the bar are 'Apply', a search icon, and a user profile icon. Below the bar, there is a search bar with the placeholder 'Search all documentation...'. The main title 'Docs' is displayed in large, bold, black font. Underneath it, the word 'Basics' is followed by a list of items: 'Accounts and users', 'Tweets', 'Direct Messages', 'Media', 'Trends', 'Geo', and 'Ads'. To the right of these items, there is a section titled 'Stay informed' with the URL <https://developer.twitter.com/en/docs.html>. Below this, there is a paragraph of text: 'Staying informed about changes to our APIs is important for those developing on the platform and can be critical to maintaining your applications. We have a number of channels to help you stay in-the-loop.' A 'Learn how >' button is located below this text. At the bottom of the page, there are two buttons: 'Search Tweets' and 'Filter realtime Tweets'.



Built a set of urls

that return data

The screenshot shows the Google Drive APIs REST API Reference page. The top navigation bar includes links for Guides, Reference, Samples, Support, and Switch to V2, along with a search bar and user profile options. The main content area features a sidebar with a table of contents for various resources like Changes, Channels, Files, Permissions, Replies, Revisions, Teamdrives, and Standard Features. The central area displays the 'API Reference' section with a sub-section for 'Resource types'. Under 'Resource types', there is a table for the 'About' resource, which includes a 'Method' column (containing 'get'), an 'HTTP request' column (containing 'GET /about'), and a 'Description' column (containing 'Gets information about the user, the user's Drive, and system capabilities.'). The URL <https://developers.google.com/drive/api/v3/reference/> is prominently displayed at the bottom of the main content area.



Built a set of urls

The screenshot shows the GitHub Developer REST API v3 documentation. The top navigation bar includes links for API Docs, Blog, Forum, and Versions, along with a search bar. The main content area is titled "REST API v3" and features a prominent blue header with the URL <https://developer.github.com/v3/>. Below the header, a paragraph describes the resources of the API. A sidebar on the right lists various API-related topics such as OAuth Authorizations API, Other Authentication Methods, Troubleshooting, Pre-release Program, API Previews, Versions, Activity, Checks, and Gists. At the bottom of the page, there is a detailed description of the "/about" endpoint.

GitHub Developer

API Docs ▾ Blog Forum Versions ▾ Search...

REST API v3

Reference Guides Libraries

<https://developer.github.com/v3/>

This describes the resources that make up the official GitHub REST API v3. If you have any problems or requests please contact [GitHub support](#).

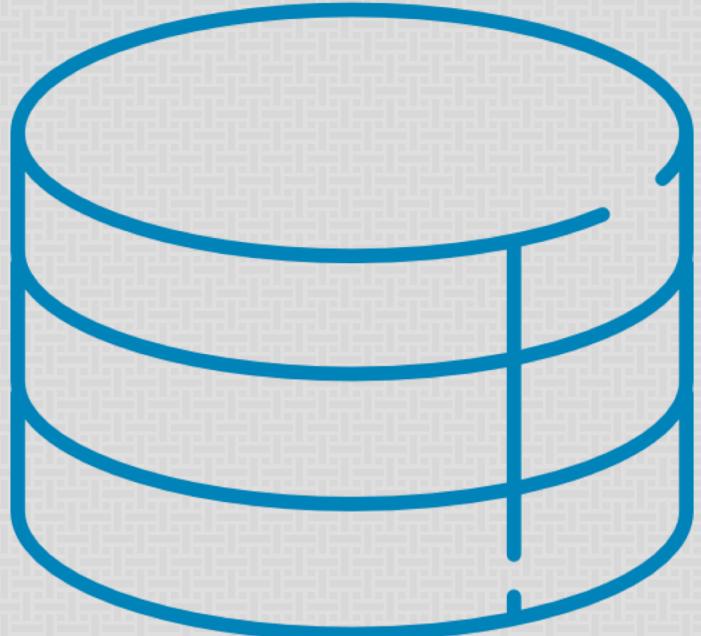
- i. [Current version](#)
- ii. [Schema](#)
- iii. [Authentication](#)
- iv. [Parameters](#)
- v. [Root endpoint](#)
- vi. [GraphQL global node IDs](#)
- vii. [Client errors](#)
- viii. [HTTP redirects](#)
- ix. [HTTP verbs](#)
- x. [Hypermedia](#)

URIs relative to <https://www.googleapis.com/drive/v3>, unless otherwise noted

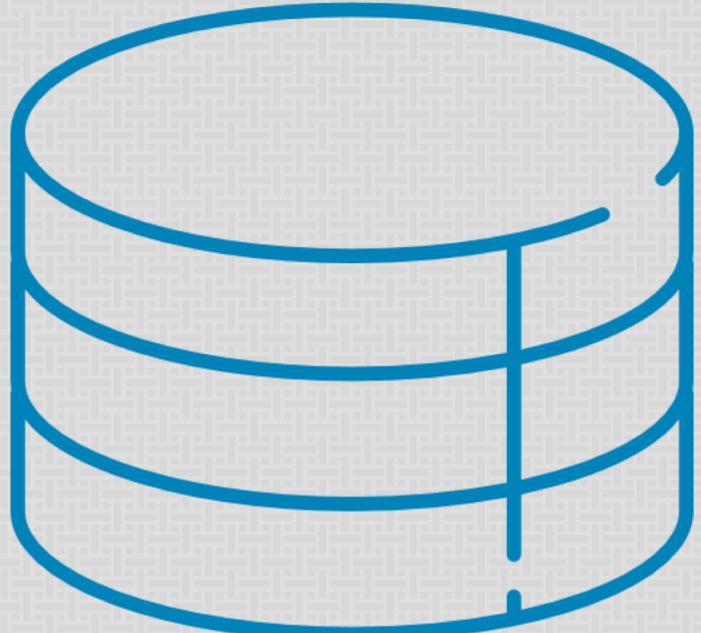
get GET /about Gets information about the user, the user's Drive, and system capabilities.

Media Types

- OAuth Authorizations API
- Other Authentication Methods
- Troubleshooting
- Pre-release Program
- API Previews
- Versions
- Activity
- Checks
- Gists



Built a set of urls
that return **data**



Built a set of urls
that return **data**,
often **JSON**

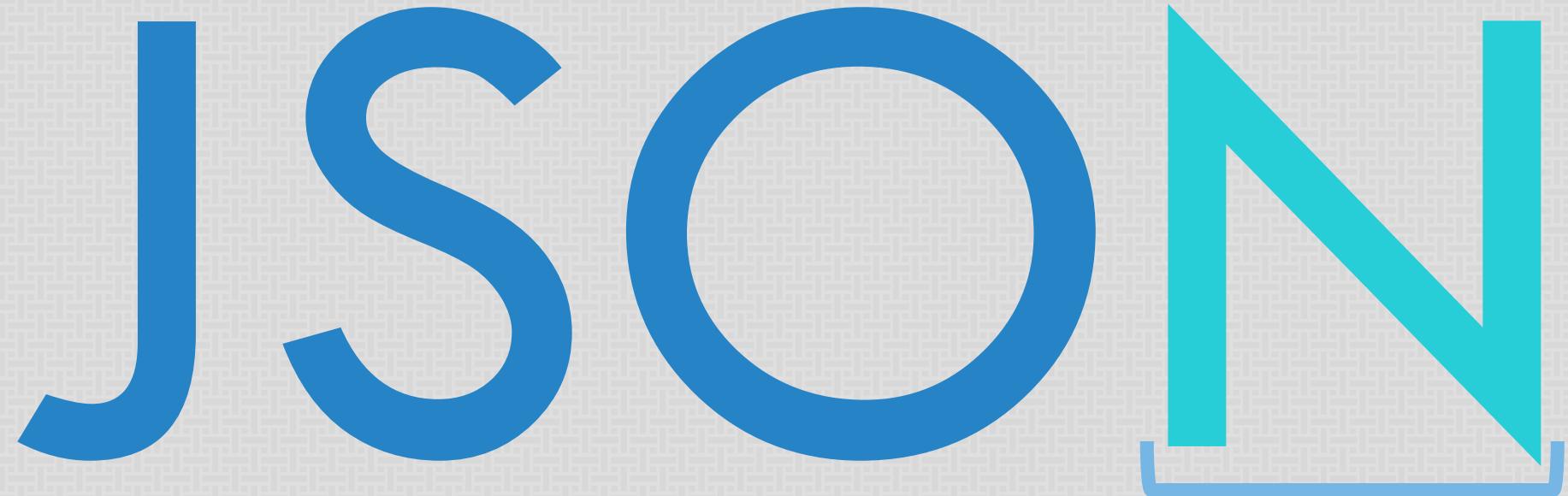
JSON

JavaScript

JSON

JavaScript Object

JSON



JavaScript Object Notation

Z
O
S
U

according to json.org

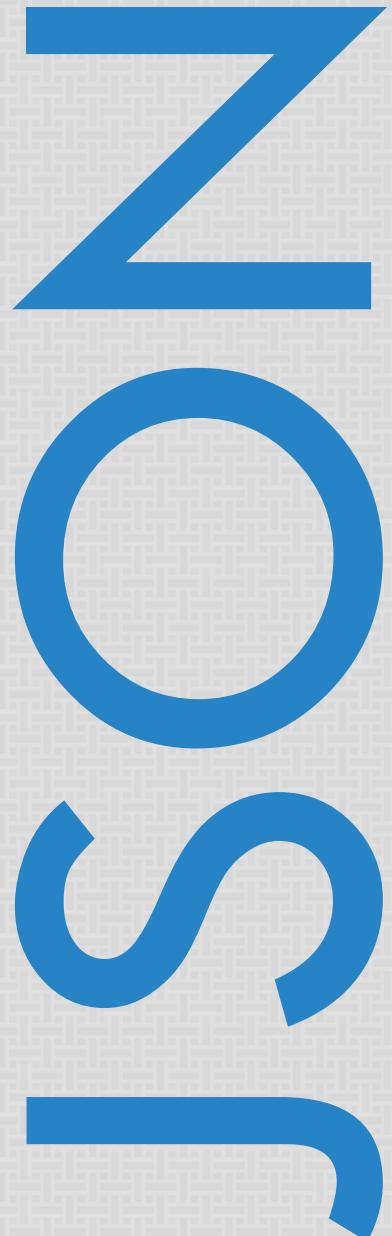
+ light weight data-interchange
format



Z
O
S
U

according to json.org

- + light weight data-interchange format
- easy for humans to read / write



according to json.org

-  light weight data-interchange
format
-  easy for humans to read /
write
-  easy for computers to parse /
generate



```
[  
 {  
   "statuses": {  
     "created_at": "Thu Jul 19 21:45:08 +0000 2018",  
     "id": 1020062005892470000,  
     "id_str": "1020062005892468738",  
     "full_text": "Our team are looking forward to  
     meeting all the @AmstatNews attendees in  
     Vancouver next week #JSM2018 #Vancouver  
     #Canada https://t.co/XiOrCtuUOk",  
     "..."  
   }  
 }  
 ]
```





```
[  
 {  
   "statuses": {  
     "created_at": "Thu Jul 19 21:45:08 +0000 2018",  
     "id": 1020062005892468738,  
     "id_str": "1020062005892468738",  
     "full_text": "Our team are looking forward to  
     meeting all the @AmstatNews attendees in  
     Vancouver next week #JSM2018 #Vancouver  
     #Canada https://t.co/XiOrCtuUOk",  
     "..."  
   }  
 }]
```





```
[  
 {  
   "statuses": {  
     "created_at": "Thu Jul 19 21:45:08 +0000 2018",  
     "id": 1020062005892468738,  
     "id_str": "1020062005892468738",  
     "full_text": "Our team are looking forward to  
     meeting all the @AmstatNews attendees in  
     Vancouver next week #JSM2018 #Vancouver  
     #Canada https://t.co/XiOrCtuUOk",  
     "..."  
   }  
 }]
```



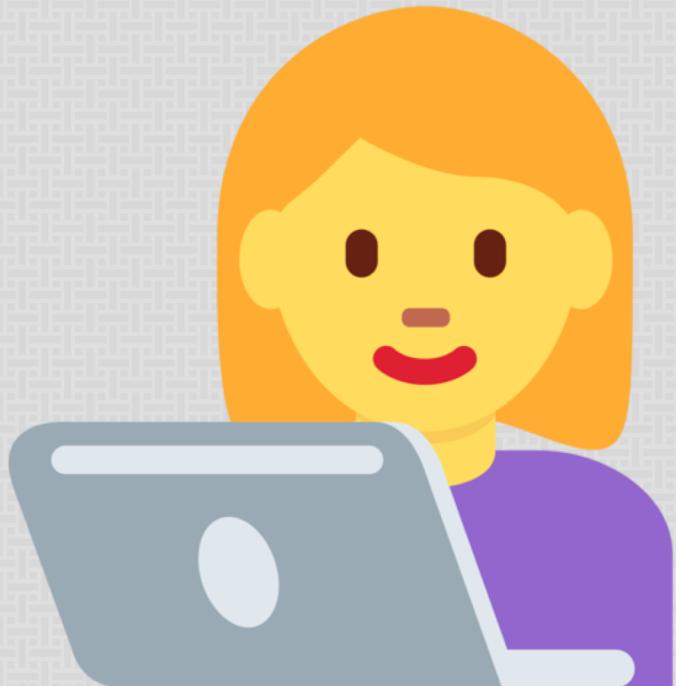


Choose Method

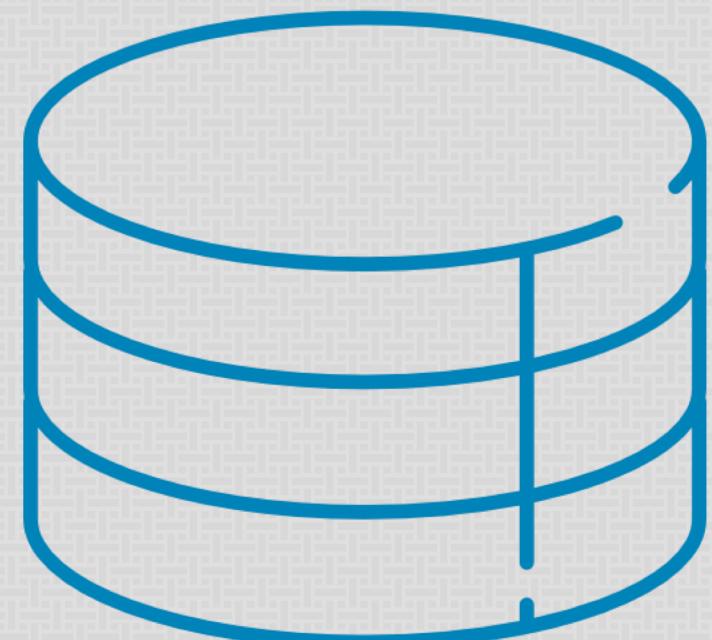


Build URL

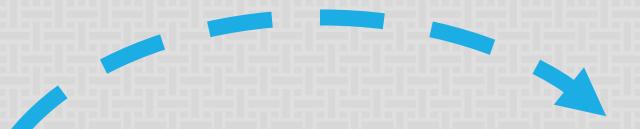
HTTP Methods



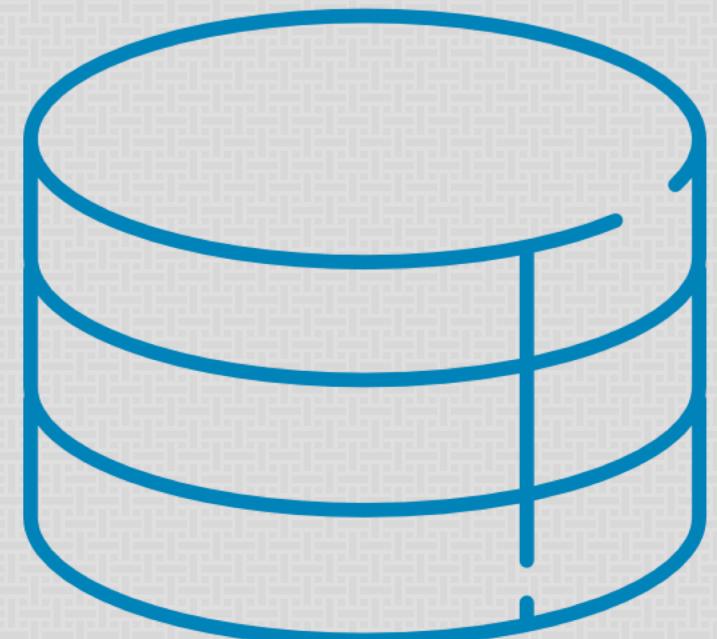
JSON



HTTP Methods



JSON



OAuth



The logo features a large teal circle on the left, with the word "Open" written in teal below it. To the right of the circle, the word "OAuth" is written in large, bold, blue capital letters. A blue bracket is positioned under the "O" of "OAuth", extending from the bottom of the "O" to the top of the "A".

Open

OAuth



Open Authorization



Choose Method



Build URL



Get Authorization

OUTLINE

- 👉 What is an API?
- 👉 How are APIs accessed from R
- 👉 Case studies



httr

- Choose Method
- Build URL
- Get Authorization
- Make Request
- Process Response

- Choose Method
- Build URL
- Get Authorization
- Make Request
- Process Response

[Code](#)[Issues 29](#)[Pull requests 2](#)[Projects 0](#)[Insights](#)[Settings](#)

Label issues and pull requests for new contributors

[Dismiss](#)

Now, GitHub will help potential first-time contributors discover issues

labeled with [help wanted](#) or [good first issue](#)

[Go to Labels](#)[Filters](#) is:open label:docs[Labels](#)[Milestones](#)[New issue](#)[X](#) Clear current search query, filters, and sorts **3 Open** ✓ 1 Closed[Author](#) ▾[Labels](#) ▾[Projects](#) ▾[Milestones](#) ▾[Assignee](#) ▾[Sort](#) ▾ **get comments from docs** [docs](#) [future](#) [sheets](#)

5

#150 opened on Jul 20, 2017 by joelgombin

 Provide workflows for the "missing functions" [docs](#)

3

#123 opened on Jul 1, 2017 by jennybc

 Unable to authorize from Rstudio Server [docs](#)

3

#79 opened on Jun 7, 2017 by tiernanmartin

REQUEST VIA HTTP METHODS

GET
POST
PUT
PATCH
DELETE



```
httr::GET()  
httr::POST()  
httr::PUT()  
httr::PATCH()  
httr::DELETE()
```

REQUEST VIA HTTP METHODS

GET
POST
PUT
PATCH
DELETE



```
httr::GET()  
httr::POST()  
httr::PUT()  
httr::PATCH()  
httr::DELETE()
```

- Choose Method
- Build URL
- Get Authorization
- Make Request
- Process Response

<https://api.github.com/repos/tidyverse/googledrive/issues?labels=docs>

```
library(httr)
```

```
url <- modify_url(  
  url = "https://api.github.com/",  
  path = "repos/tidyverse/googledrive/issues",  
  query = list(labels = "docs"))  
)
```

BUILD URL

base URL

`https://api.github.com/repos/tidyverse/googledrive/issues?labels=docs`

```
library(httr)
```

```
url <- modify_url(  
  url = "https://api.github.com/",  
  path = "repos/tidyverse/googledrive/issues",  
  query = list(labels = "docs"))  
)
```

BUILD URL

base URL

endpoint

`https://api.github.com/repos/tidyverse/googledrive/issues?labels=docs`

```
library(httr)
```

```
url <- modify_url(  
  url = "https://api.github.com/",  
  path = "repos/tidyverse/googledrive/issues",  
  query = list(labels = "docs"))  
)
```

BUILD URL

base URL

endpoint

parameters

https://api.github.com/repos/tidyverse/googledrive/issues?labels=docs

```
library(httr)
```

```
url <- modify_url(  
  url = "https://api.github.com/",  
  path = "repos/tidyverse/googledrive/issues",  
  query = list(labels = "docs"))  
)
```

BUILD URL

- ✓ Choose Method
- ✓ Build URL
- Get Authorization
- Make Request
- Process Response

AUTHENTICATION

1

Register an application

The screenshot shows the 'Register a new OAuth application' page on GitHub. The page has a dark header with navigation links: 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, there's a search bar and a user icon. The main form area has the following fields:

- Application name:** A text input field with a placeholder 'Something users will recognize and trust'.
- Homepage URL:** An empty text input field with a placeholder 'The full URL to your application homepage'.
- Application description:** A text input field containing the placeholder 'Application description is optional'.
- Authorization callback URL:** An empty text input field with a placeholder 'Your application's callback URL. Read our OAuth documentation for more information.'

At the bottom of the form are two buttons: a green 'Register application' button and a 'Cancel' button.

<https://github.com/settings/developers>

JSM 2018

AUTHENTICATION

1

Register an application

A screenshot of the GitHub OAuth application registration form. The page title is "Register a new OAuth application". It has fields for "Application name" (with placeholder "Something users will recognize and trust"), "Homepage URL" (with placeholder "The full URL to your application homepage"), "Application description" (with placeholder "Application description is optional" and note "This is displayed to all users of your application"), and "Authorization callback URL" (with placeholder "Your application's callback URL. Read our OAuth documentation for more information"). At the bottom are "Register application" and "Cancel" buttons.

Search or jump to... Pull requests Issues Marketplace Explore

Register a new OAuth application

Application name

Something users will recognize and trust

Homepage URL

The full URL to your application homepage

Application description

Application description is optional

This is displayed to all users of your application

Authorization callback URL

Your application's callback URL. Read our OAuth documentation for more information.

Register application Cancel

2

Create an OAuth application

library(httr)

```
app <- oauth_app(  
  appname = "NAME",  
  key = "KEY",  
  secret = "SECRET"  
)
```

<https://github.com/settings/developers>

JSM 2018

AUTHENTICATION

1

Register an application

The screenshot shows the 'Register a new OAuth application' form on GitHub. It includes fields for Application name, Homepage URL, Application description, and Authorization callback URL. At the bottom are 'Register application' and 'Cancel' buttons.

Search or jump to... Pull requests Issues Marketplace Explore

Register a new OAuth application

Application name

Something users will recognize and trust

Homepage URL

The full URL to your application homepage

Application description

Application description is optional

This is displayed to all users of your application

Authorization callback URL

Your application's callback URL. Read our [OAuth documentation](#) for more information.

Register application Cancel

2

Create an OAuth application

library(httr)

```
app <- oauth_app(  
  appname = "NAME",  
  key = "KEY",  
  secret = "SECRET"  
)
```

3

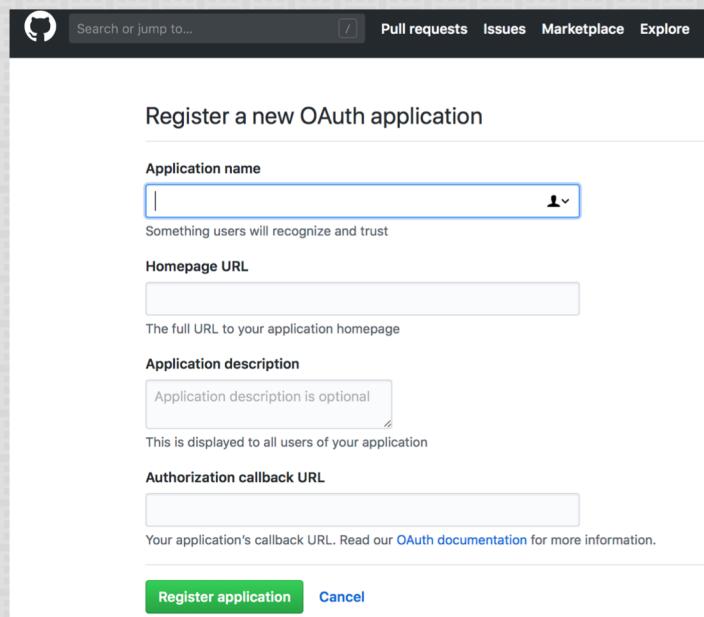
Generate a token

```
token <-  
  oauth2.0_token(  
    oauth_endpoints("github"),  
    app  
)
```

AUTHENTICATION

1

Register an application



A screenshot of the GitHub developer settings page showing the "Register a new OAuth application" form. The form fields include:

- Application name: A text input field containing the placeholder "Something users will recognize and trust".
- Homepage URL: An empty text input field.
- Application description: An empty text input field with the note "Application description is optional".
- Authorization callback URL: An empty text input field with the note "This is displayed to all users of your application".
- Register application button: A green button at the bottom left.
- Cancel button: A blue button at the bottom right.

2

Create an OAuth application

library(httr)

```
app <- oauth_app(  
  appname = "NAME",  
  key = "KEY",  
  secret = "SECRET"  
)
```

3

Generate a token

```
token <-  
  oauth2.0_token(  
    oauth_endpoints("github"),  
    app  
)
```



Code demonstrations:

<u>httr::connection-sharing</u>	Demonstration of how connection sharing saves time	(Run demo in console)
<u>httr::oauth1-twitter</u>	Using twitter api with OAuth 1.0	(Run demo in console)
<u>httr::oauth1-vimeo</u>	Using vimeo api with OAuth 1.0	(Run demo in console)
<u>httr::oauth1-withings</u>	Using withings api with OAuth 1.0	(Run demo in console)
<u>httr::oauth1-yahoo</u>	Using yahoo api with OAuth 1.0	(Run demo in console)
<u>httr::oauth2-azure</u>	Using Azure apis with OAuth 2.0	(Run demo in console)
<u>httr::oauth2-facebook</u>	Using the facebook api with OAuth 2.0	(Run demo in console)
<u>httr::oauth2-github</u>	Using the github api with OAuth 2.0	(Run demo in console)
<u>httr::oauth2-google</u>	Using the google api with OAuth 2.0	(Run demo in console)
<u>httr::oauth2-linkedin</u>	Using linkedin api with OAuth 1.0	(Run demo in console)
<u>httr::oauth2-reddit</u>	Using the reddit api with OAuth 2.0	(Run demo in console)
<u>httr::oauth2-yelp</u>	Using yelp api with OAuth 2.0 and Client Credentials Grant	(Run demo in console)
<u>httr::service-account</u>	Using Google service account	(Run demo in console)

- ✓ Choose Method
- ✓ Build URL
- ✓ Get Authorization
- Make Request
- Process Response

HTTP Requests

```
library(httr)  
  
req <- GET(url, config = token)
```

HTTP
REQUEST
RESPONSE

```
library(httr)

req <- GET(url, config = token)

req
#> Response
#> [https://api.github.com/repos/tidyverse/googledrive/issues?labels=docs]
#>   Date: 2018-07-20 14:35
#>   Status: 200
#>   Content-Type: application/json; charset=utf-8
#>   Size: 86.6 kB
#>   [
#>     {
#>       "url": "https://api.github.com/repos/tidyverse/googledrive/issues/150",
#>       "repository_url": "https://api.github.com/repos/tidyverse/googledrive",
#>     ...
#>   ]
```

HTTP Requests

```
library(httr)

req <- GET(url, config = token)

req
#> Response
#> [https://api.github.com/repos/tidyverse/googledrive/issues?label=docs]
#>   Date: 2018-07-20 14:35
#>   Status: 200
#>   Content-Type: application/json; charset=utf-8
#>   Size: 86.6 kB
#>   [
#>     {
#>       "url": "https://api.github.com/repos/tidyverse/googledrive/issues/150",
#>       "repository_url": "https://api.github.com/repos/tidyverse/googledrive",
#>     ...
#>   ]
```

- ✓ Choose Method
- ✓ Build URL
- ✓ Get Authorization
- ✓ Make Request
- Process Response

PROCESS

```
library(httr)

req <- GET(url, config = token)

res <- content(req)
```

```
library(httr)

req <- GET(url, config = token)

res <- content(req)

res
#> [[1]]
#> [[1]]$url
#> [1] "https://api.github.com/repos/tidyverse/googledrive/issues/150"
#>
#> [[1]]$repository_url
#> [1] "https://api.github.com/repos/tidyverse/googledrive"
```

PROCESS

```
library(httr)
req <- GET("https://api.github.com/repos/tidyverse/googledrive/issues/150")
res <- content(req, "text")
res
#> [[1]]
#> [[1]]$url
#> [1] "https://api.github.com/repos/tidyverse/googledrive/issues/150"
#>
#> [[1]]$repository_url
#> [1] "https://api.github.com/repos/tidyverse/googledrive"
```

A hexagonal sticker with a dark green border. Inside is a black and white illustration of a cat's head and upper body, looking slightly to the left. The word "purrr" is written in a bold, sans-serif font above the cat. At the bottom right corner of the hexagon, the URL "www.rstudio.com" is printed vertically.

PROCESS

```
library(httr)

req <- GET(url, config = token)

res <- content(req)

purrr::map_chr(res, "url")
```

```
library(httr)

req <- GET(url, config = token)

res <- content(req)

purrr::map_chr(res, "url")
#> [1] "https://api.github.com/repos/tidyverse/googledrive/issues/150"
#> [2] "https://api.github.com/repos/tidyverse/googledrive/issues/123"
#> [3] "https://api.github.com/repos/tidyverse/googledrive/issues/79"
```

- ✓ Choose Method
- ✓ Build URL
- ✓ Get Authorization
- ✓ Make Request
- ✓ Process Response



gh
rtweet
googledrive
googlesheets
yelpr
meetupr
imugR
aws.*

```
library(googledrive)
```

```
drive_ls()
```

```
library(googledrive)

drive_ls()
#> Waiting for authentication in browser...
#> Press Esc/Ctrl + C to abort
```



Sign in with Google

tidyverse api packages wants to access your Google Account



lucydagostino@gmail.com

This will allow tidyverse api packages to:



View and manage the files in your Google Drive



Allow tidyverse api packages to do this?

By clicking Allow, you allow this app to use your information in accordance to their terms of service and privacy policies. You can view or remove app access in your [Google Account](#)

CANCEL

ALLOW

```
library(googledrive)
```

```
drive_ls()
```

```
#> Waiting for authentication in browser...
#> Press Esc/Ctrl + C to abort
#> Authentication complete.
#> Items so far:
#> 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1670
#> # A tibble: 1,670 x 3
#>   name          id      drive_resource
#>   * <chr>        <chr>    <list>
#> 1 data.zip     1ET81D_Uku6Tm_Z4U04jB4qTNrC... <list [37]>
#> 2 science_manuscript 10X07LsQI5GL_Tdrq3-3PtgcDtW... <list [31]>
#> 3 talks        12bBwImIu4Yzg5UaqcvxIZ0rbyG... <list [31]>
#> 4 presentations 1-PItelqpv0Sb_LdiEDqb803D_R... <list [32]>
#> 5 science_supplementary_materials 1JNogHk0Zn-H5je6a-DBh0muwBW... <list [31]>
#> 6 iris-collaboration-analysis-plan 10DrdHfxyPFTuACK05YqcpFrcPM... <list [32]>
#> 7 2018-07-16_meeting-minutes 1oUAPeFSvVfo5BBoeCip-2ku0up... <list [32]>
#> 8 p-hack-athon.Rda 1cwgv9GkrgcVEImxcKSni-Tdf-B... <list [32]>
#> 9 rladies-pres 1-cSy2GqNumuD2YpeI0G_Tv2aQ-... <list [33]>
#> 10 Women in Statistics 1ZRK0Bonakg_qkg5McU7Lr9fpDx... <list [30]>
#> # ... with 1,660 more rows
```

OUTLINE

- 👉 What is an API?
- 👉 How are APIs accessed from R
- 👉 Case studies

UPLOAD MY SLIDES TO DRIVE

```
library(googledrive)  
  
drive_upload("2018-07_jsm/slides.pptx",  
             path = "postdoc/talks/",  
             name = "2018-07_jsm_slides",  
             type = "presentation")
```

ORGANIZE MY DRIVE FROM THE COMFORT OF MY CONSOLE

```
library(googledrive)

leek_files <- drive_find(q = "'jtleek@gmail.com' in writers")
postdoc_folder <- drive_mkdir("postdoc")
purrr::walk(leek_files$id,
            ~drive_mv(as_id(.x), path = postdoc_folder)
        )
```

ORGANIZE MY DRIVE FROM THE COMFORT OF MY CONSOLE

```
library(googledrive)

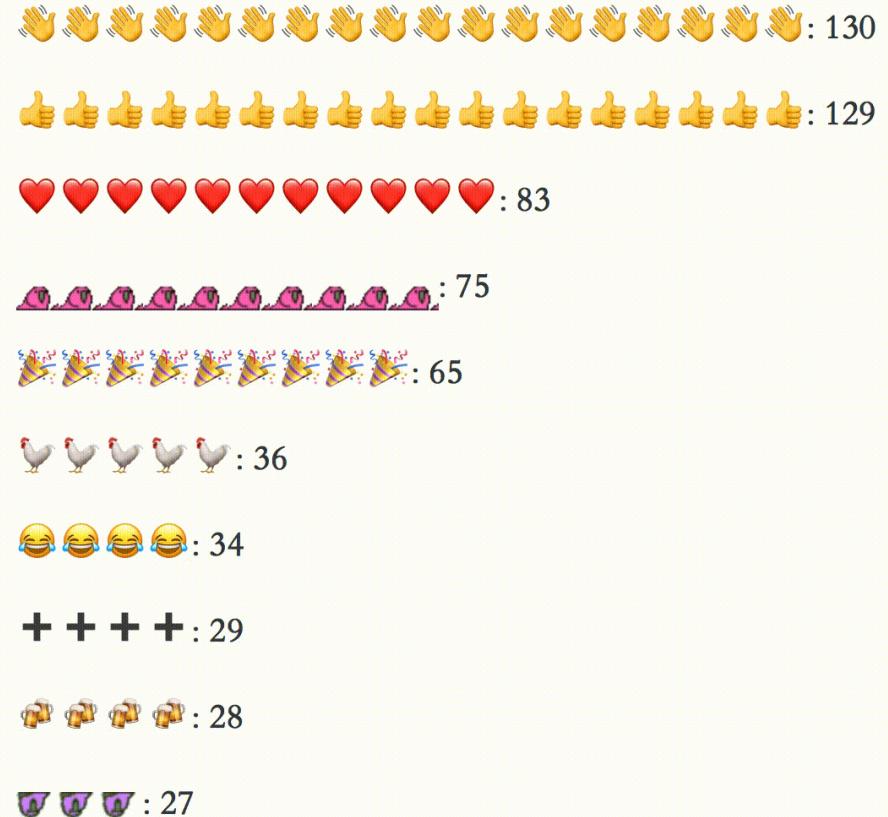
leek_files <- drive_find(q = "'jtleek@gmail.com' in writers")
postdoc_folder <- drive_mkdir("postdoc")
purrr::walk(leek_files$id,
            ~drive_mv(as_id(.x), path = postdoc_folder)
        )
```

ORGANIZE MY DRIVE FROM THE COMFORT OF MY CONSOLE

```
library(googledrive)

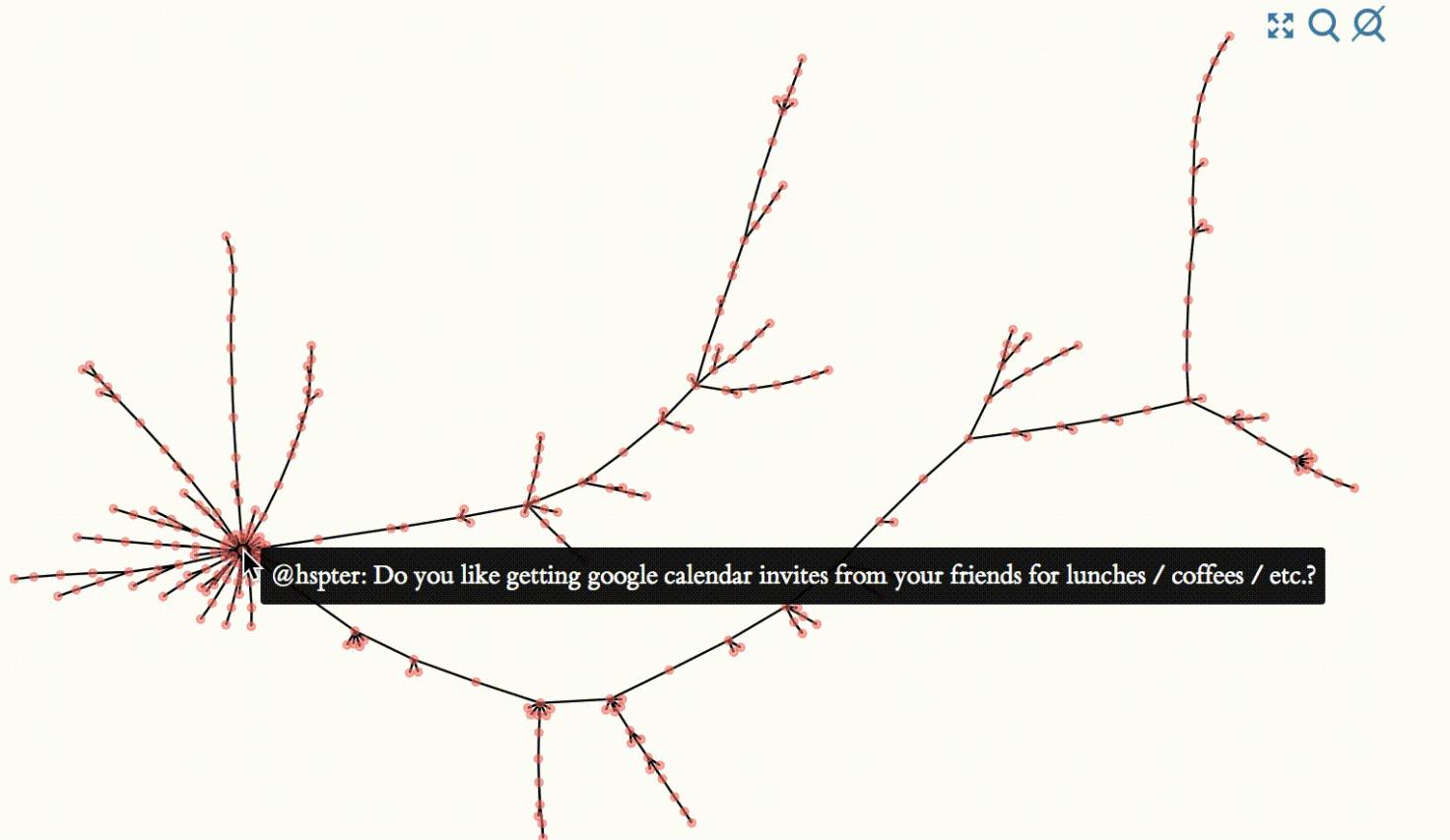
leek_files <- drive_find(q = "'jtleek@gmail.com' in writers")
postdoc_folder <- drive_mkdir("postdoc")
purrr::walk(leek_files$id,
            ~drive_mv(as_id(.x), path = postdoc_folder)
        )
```

PULLING MOST FREQUENT SLACK EMOJI



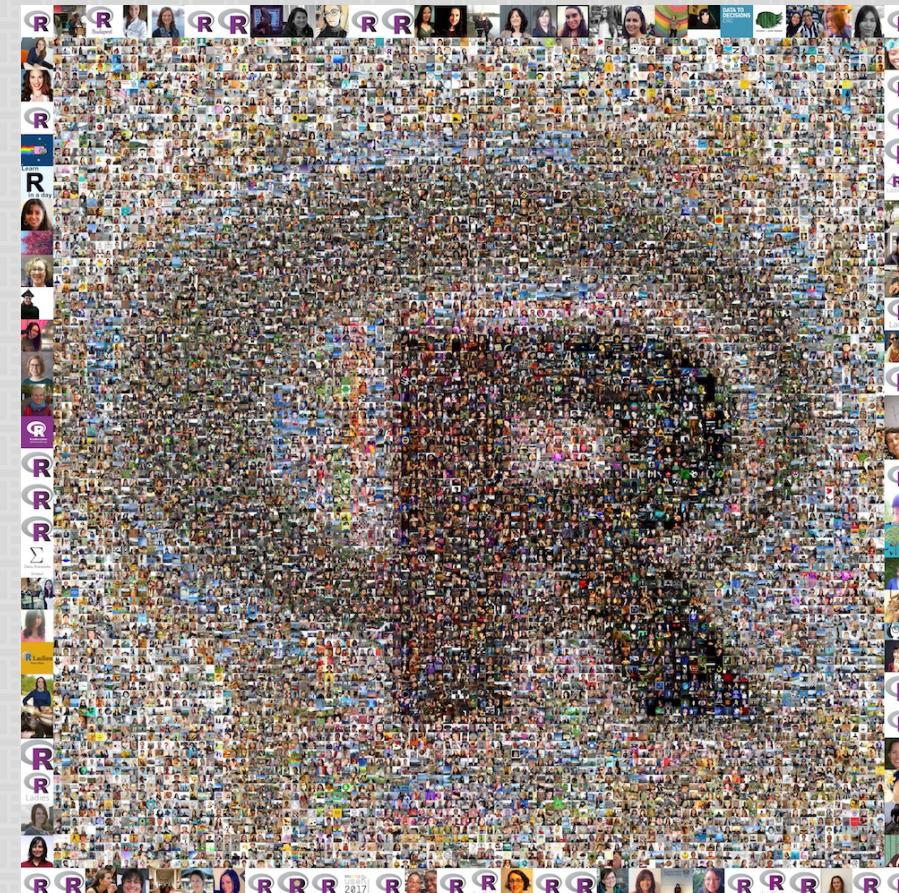
<https://livefreeordichotomize.com/2017/07/17/ropensci-slack-emojis/>

NAVIGATING TWITTER RESPONSES



<https://livefreeordichotomize.com/2017/07/24/twitter-trees/>
JSM 2018

WE R LADIES



<https://livefreeordichotomize.com/2017/07/18/the-making-of-we-r-ladies/>
JSM 2018



@LUCYSTATS

bit.ly/lucystats-jsm2018

LUCYDAGOSTINO@GMAIL.COM

Thank you!

HARNESSING THE POWER OF THE WEB VIA R CLIENTS FOR WEB APIs

Lucy D'Agostino
McGowan, PhD
*Johns Hopkins Bloomberg
School of Public Health*