

Sampling GraphQL

FROM BAZ LUHRMANN



@bdougieYO

sampled disco tracks



Grand Master Flash & The Furious Five

@bdougieYO

“the message”

rap became
social commentary

gansta rap



N.W.A



Public Enemy

@bdougieYO

2PAC



all eyez on me



@bdougieYO

“ain’t nothin’ but
a gangsta party”

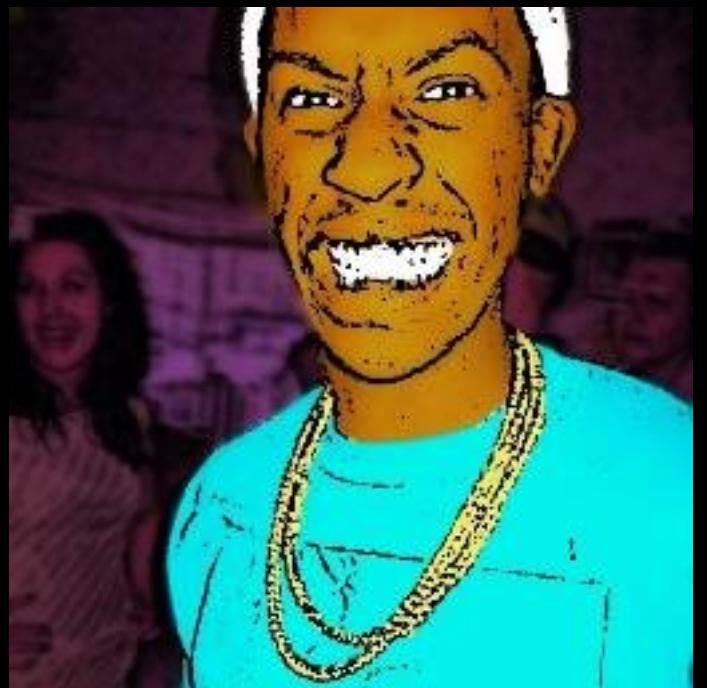
(2 of Amerikaz most wanted)

1996

give it a REST!

GitHub





Developer Advocate

Brian Douglas [Money\\$ign](#)

 [@bdougieYO](#)

 [@bdougie](#)



[@bdougieYO](#)



Queen Bey



What
Would
Bey
Do?

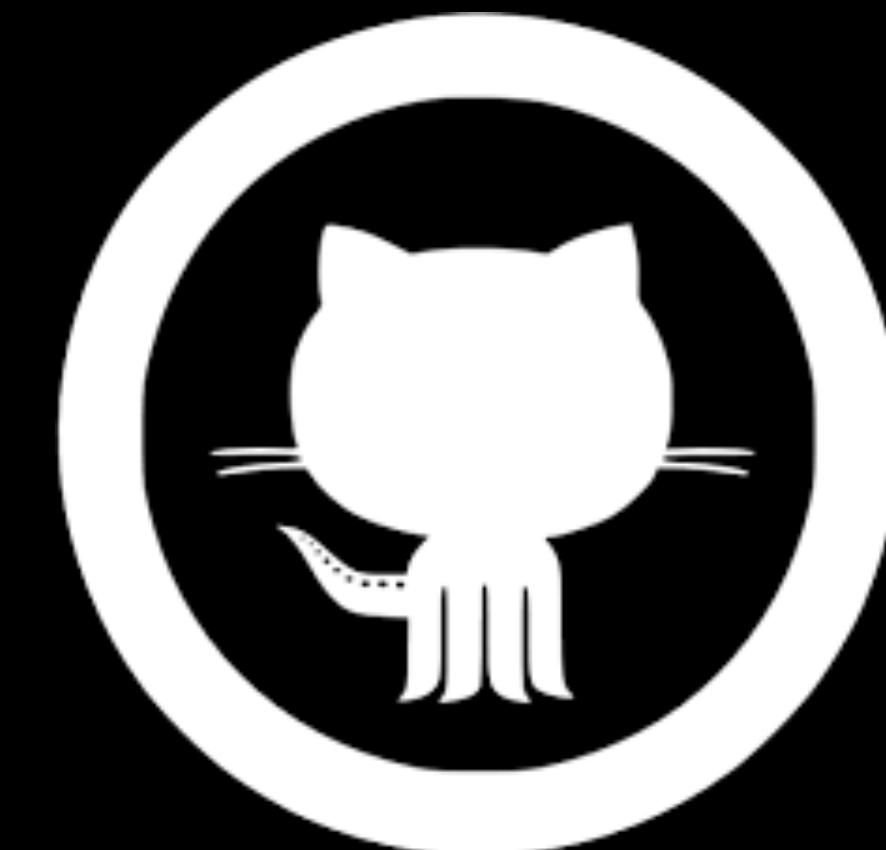


@bdougieYO

A close-up photograph of a woman with long, curly brown hair, smiling warmly at the camera. She has dark eyes and is wearing dark eyeliner and mascara. Her right arm is wrapped around a large, light-colored beehive, which is covered in small, dark spots. The background is a dark, textured surface, possibly a wall or a piece of furniture.

go to bat for the hive

GitHub GraphQL API



**GitHub chose GraphQL
to provide more flexibility
for our integrators.**

improved experience

- Integrators
- Flexibility



**GitHub is a
collaboration platform.**

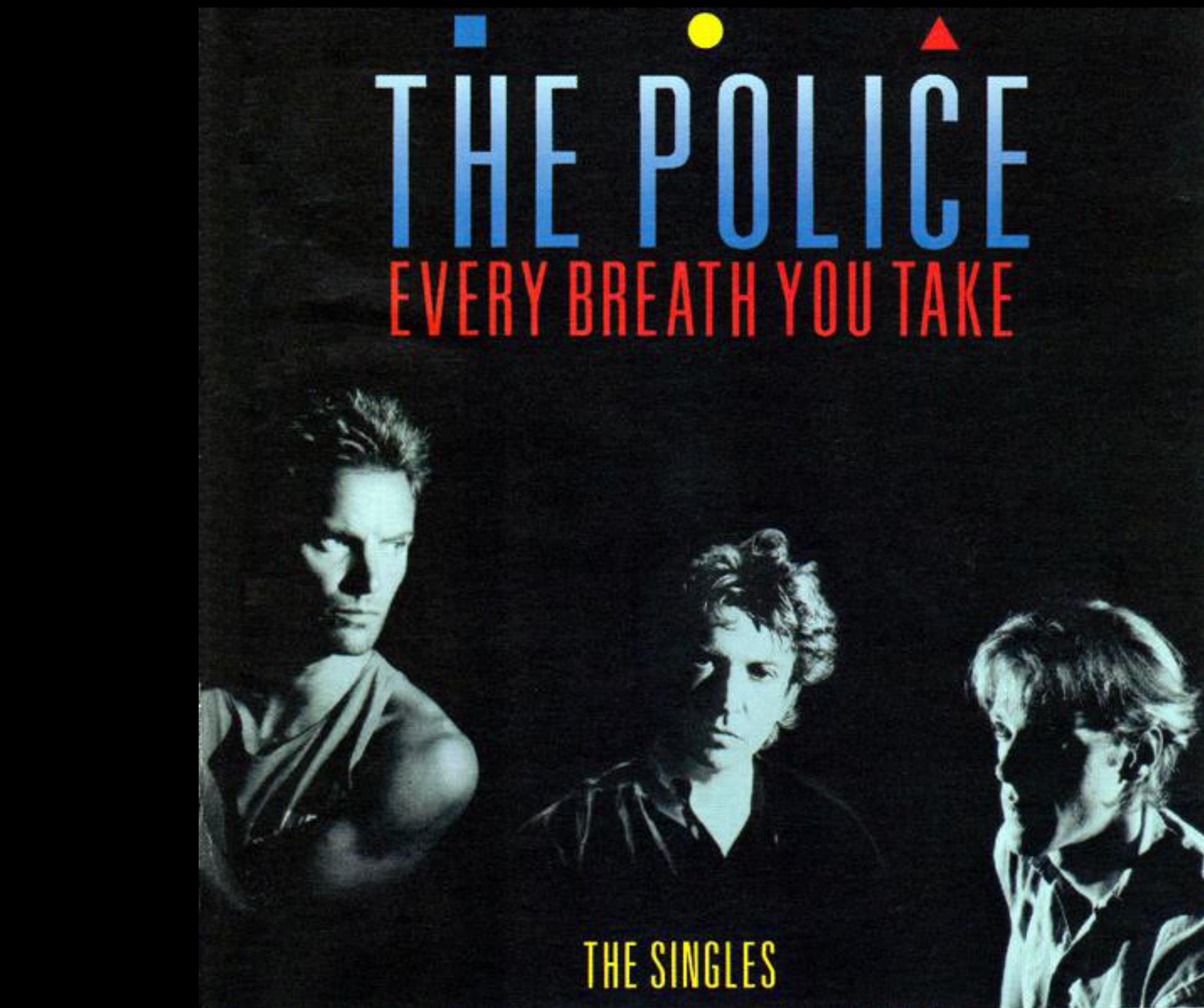
**We provide enough
features and get out of
the way of the
developer.**

better hooks

@bdougieYO



Puff Daddy *(P Diddy)*



@bdougieYO

1997



(wrap changed forever)



(even will came back to rap)

2016

GitHub sampled GraphQL

JAMstack Radio Podcast #5

A screenshot of a web browser displaying the Heavybit Industries website. The page shows the "JAMstack Radio" episode #5, titled "Ep. #5, GraphQL At GitHub". The episode summary mentions Kyle Daigle as a guest. On the left sidebar, there's information about the show, including its logo, a description of it being a show about the JAMstack, and links to subscribe via Apple Podcasts and Google Play.

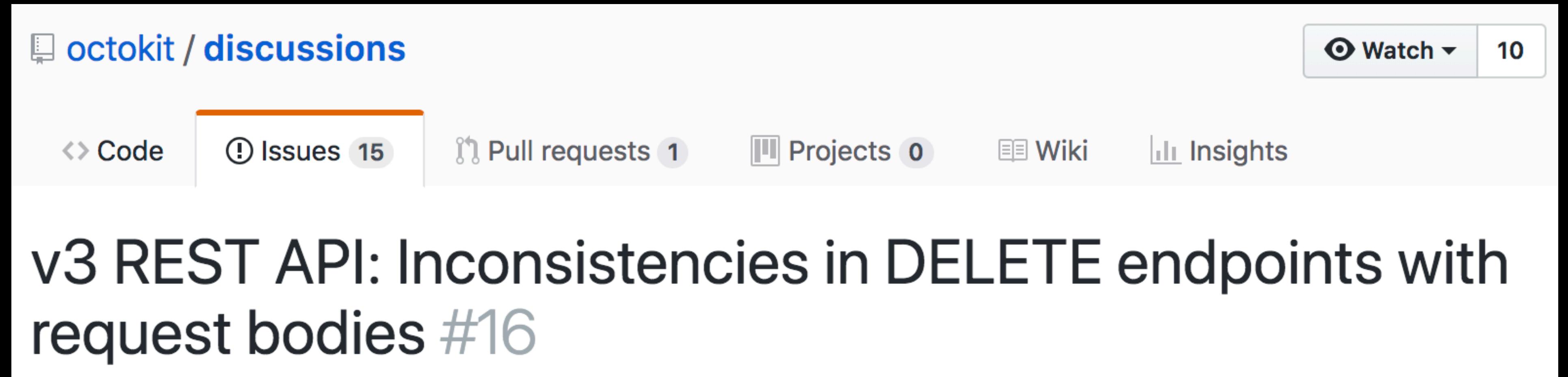
The browser window has a title bar "JAMstack Radio | Ep. #5, GraphQL At GitHub" and a URL "https://www.heavybit.com/library/podcasts/jamstack-radio/ep-5-graphql-at-github/". The top navigation bar includes links for "library", "events", "spotlights", "about", and "apply". The main content area features a play button, a progress bar at 0:00 / 41:01, volume controls, and download options. Below the player, there's a section for "ABOUT THE GUESTS" featuring a bio for Kyle Daigle.

@bdougieYO

2017

**GitHub launched their
GraphQL-first API**

hooks: inconsistency



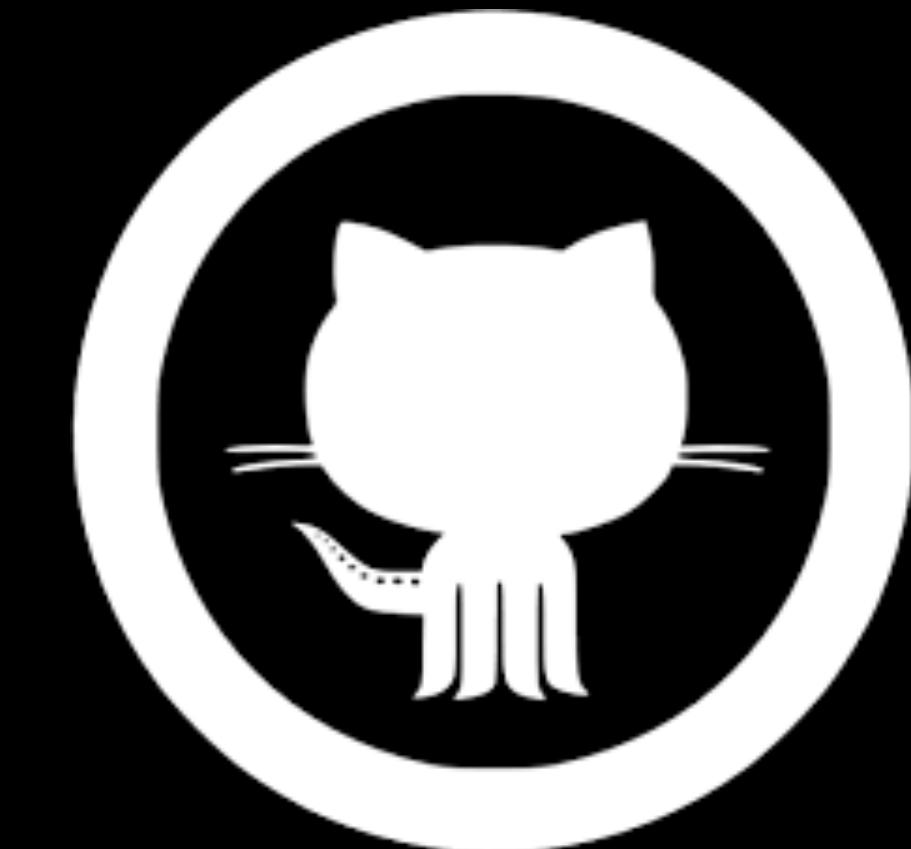
hooks: documentation



@bdougieYO

I wanted to try this new
GraphQL API by building
a new project

GitHub is the place to
discover **open source**



I want to get involved in
Open Source

but **every-time** I find a project
I don't know where to start

Does GraphQL kill REST?

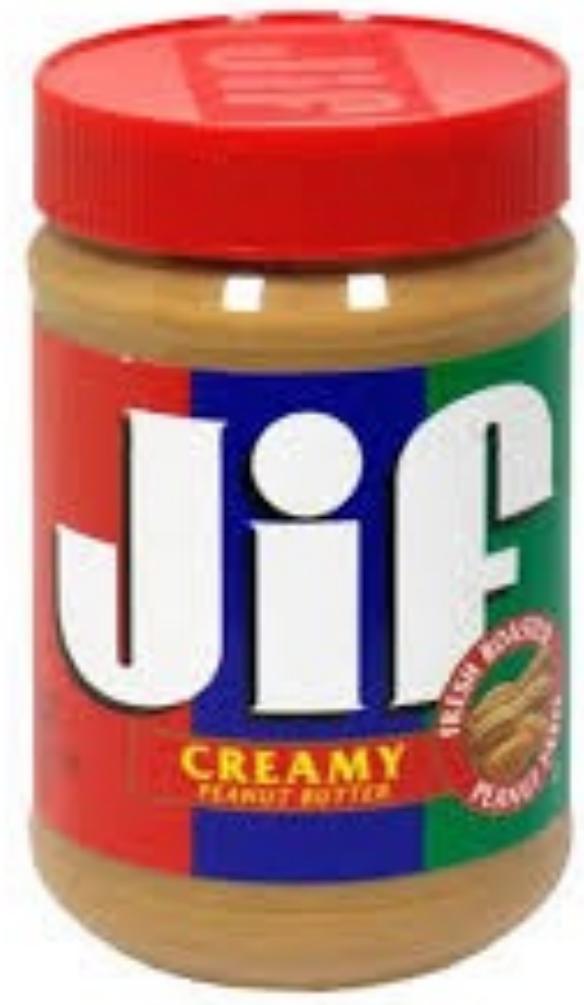
JAM^{STAC}K

@bdougieYO

Monolithic Architecture



@bdougieYO



API



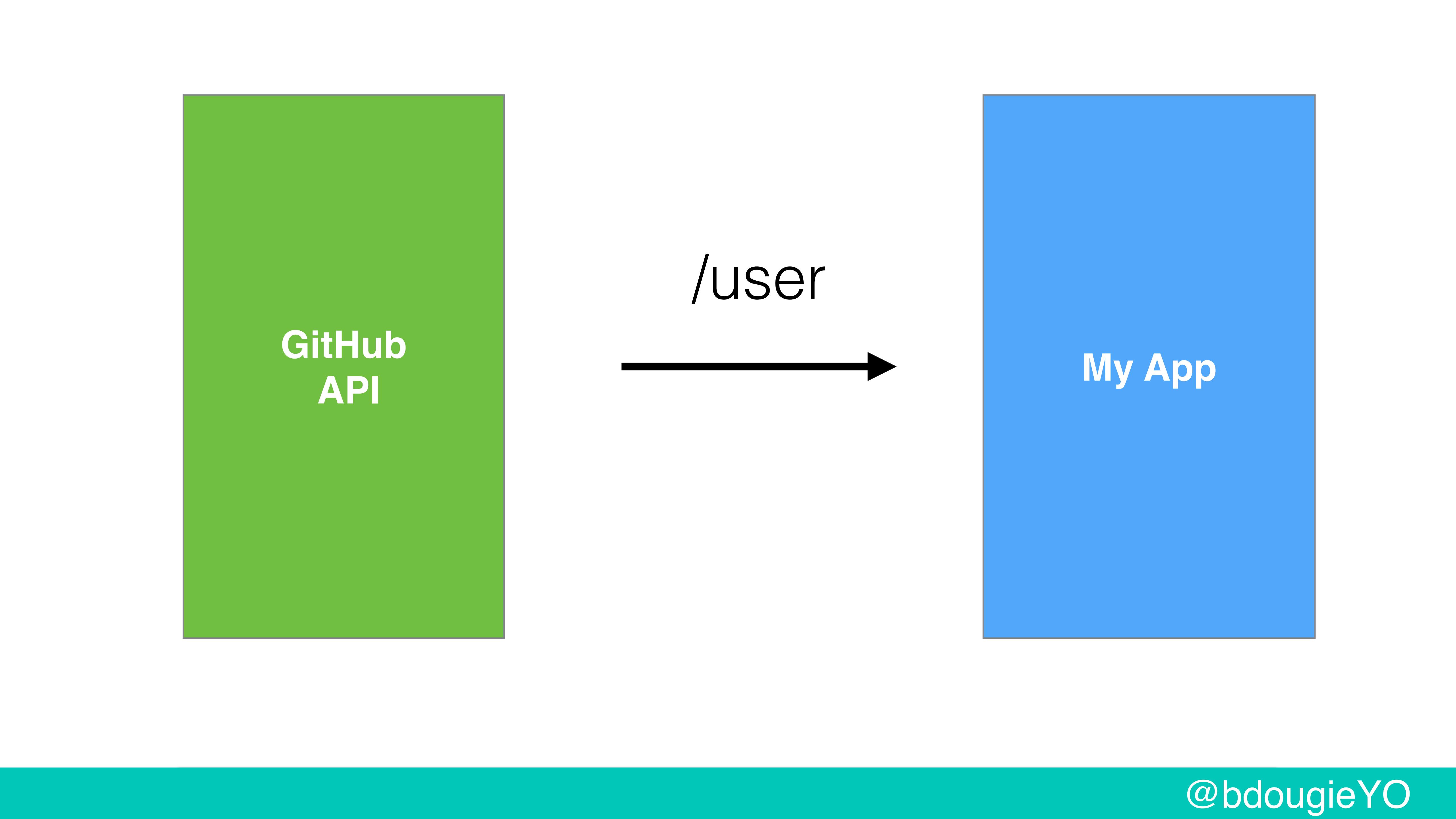
Client

The JAM Architecture

REST endpoints

/user
/repositories
/notifications

lost in docs



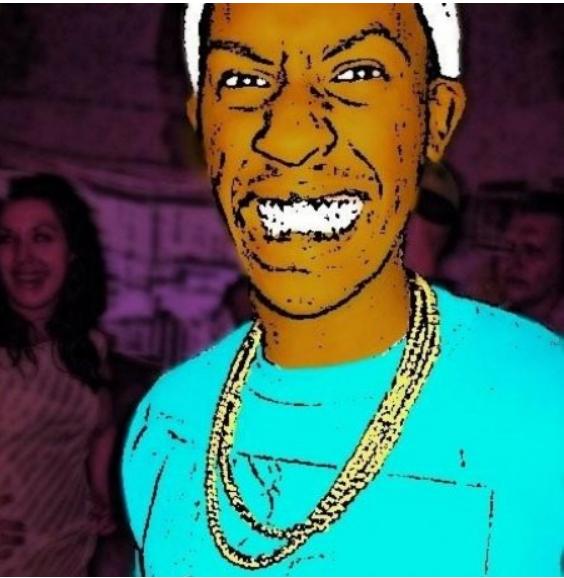
GitHub
API

/user



My App

**GitHub
API**



My App

A diagram illustrating a network request. On the left is a green rectangle labeled "GitHub API". On the right is a blue rectangle labeled "My App". A black horizontal arrow points from "My App" to "GitHub API". Above the arrow, the text "/repositories" is written, indicating the endpoint being requested.

GitHub
API

My App

/repositories



**GitHub
API**



My App

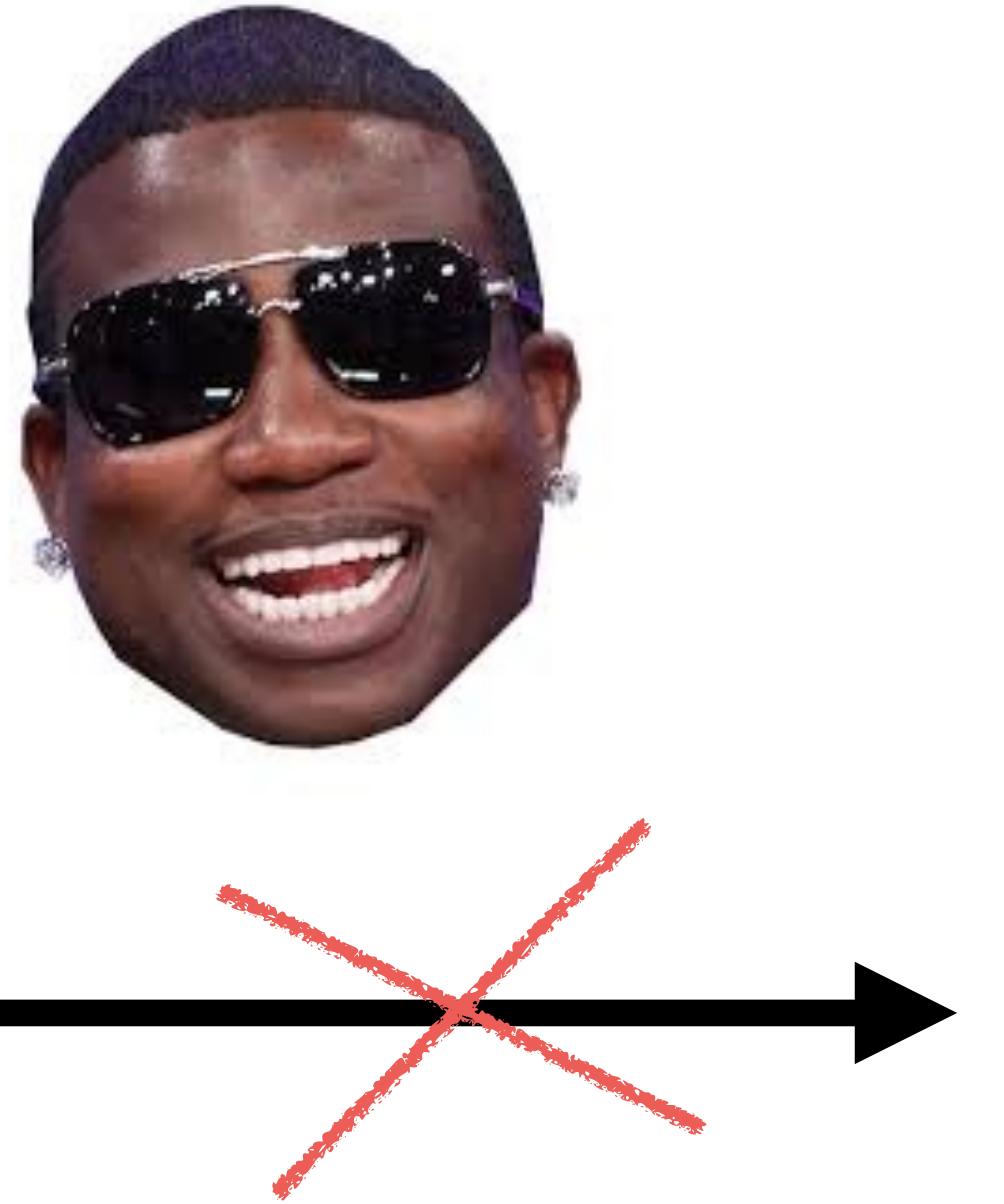
**GitHub
API**

/notifications



My App

**GitHub
API**



My App

REST endpoints

/user
/repositories
/user/notifications

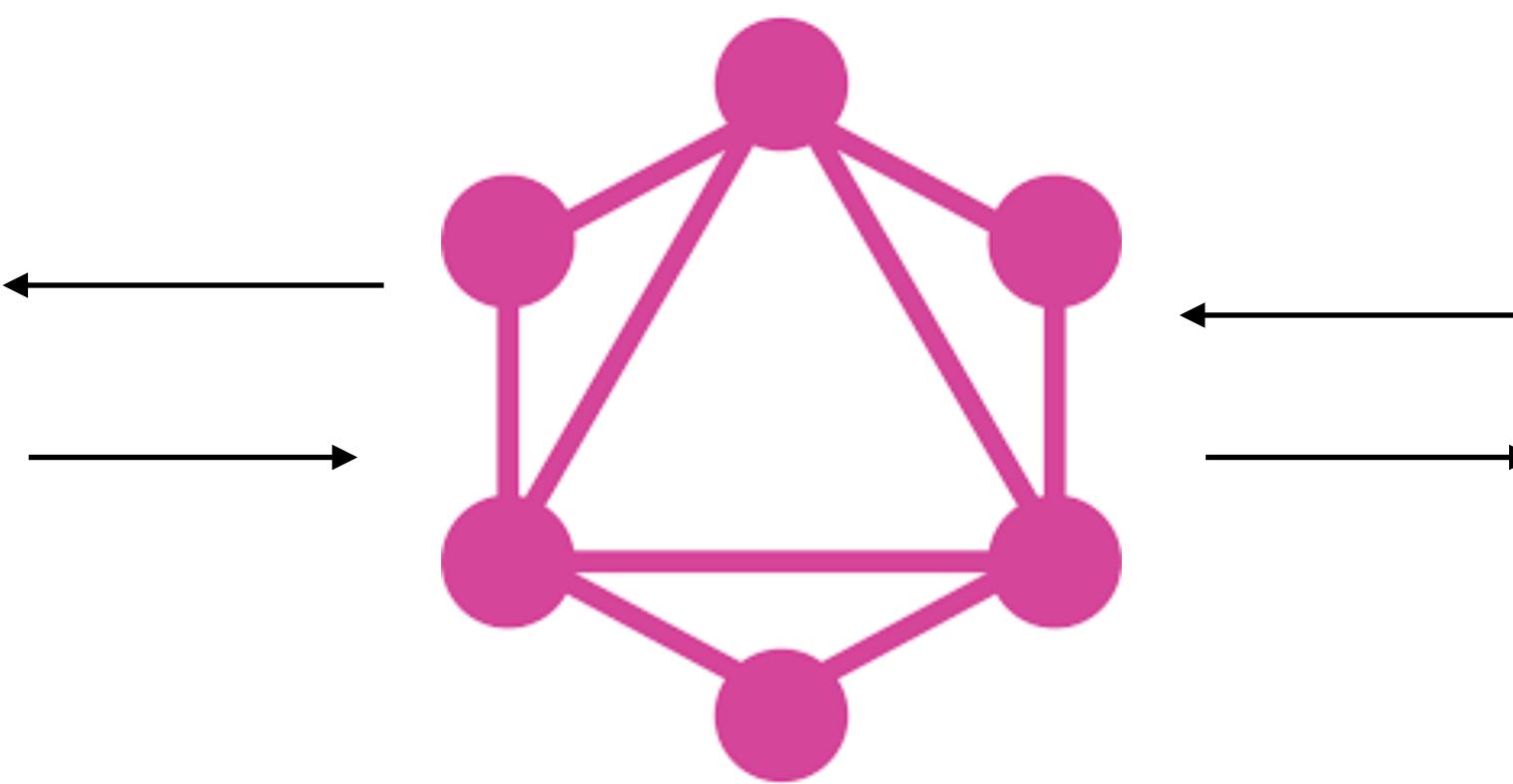
lost in docs

How is GraphQL different from the REST?

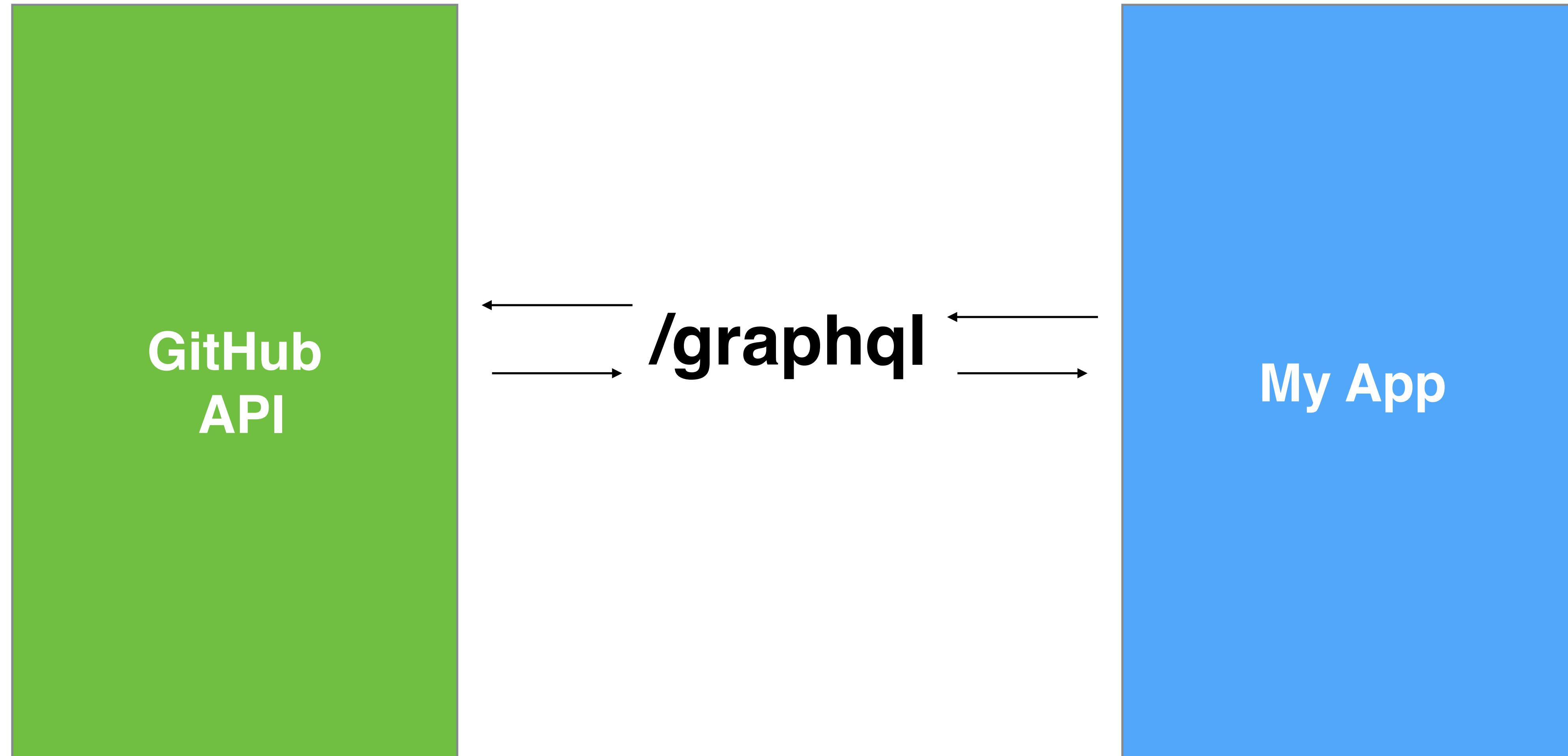


**not a replacement for REST, but
alternative for querying data**

**GitHub
API**

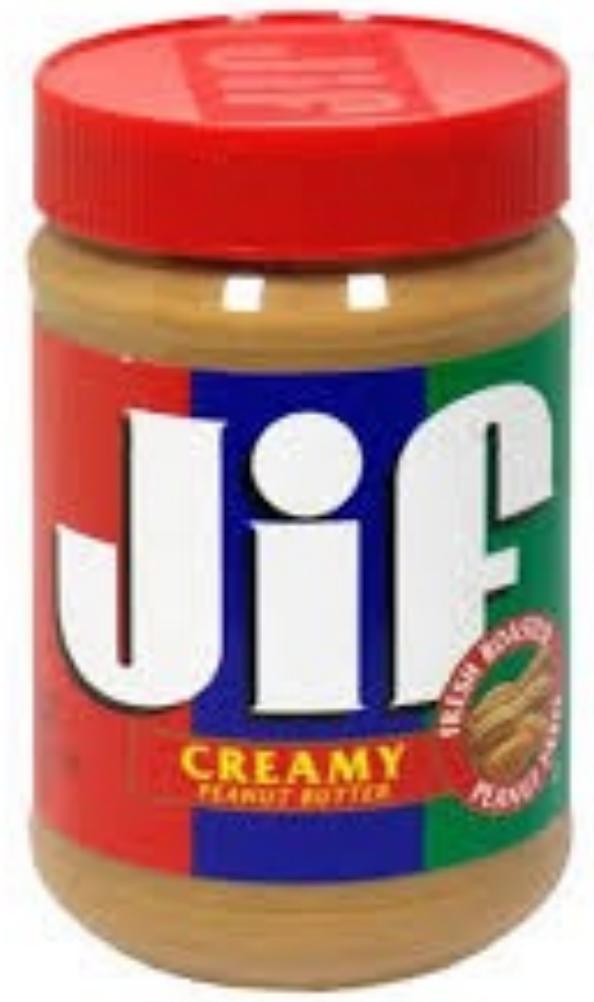


My App

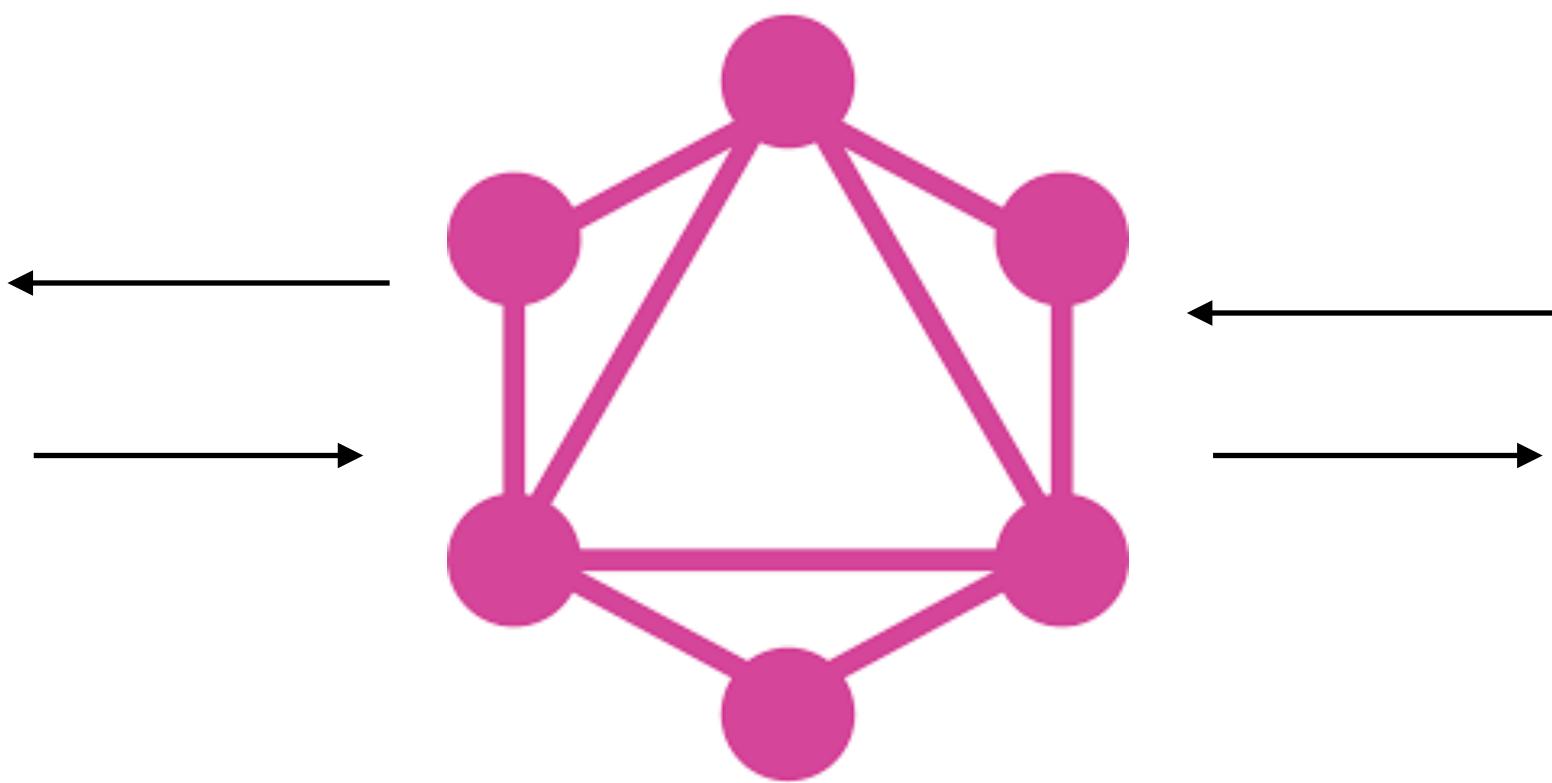




**not a replacement for REST, but
alternative for querying data**



API



Client

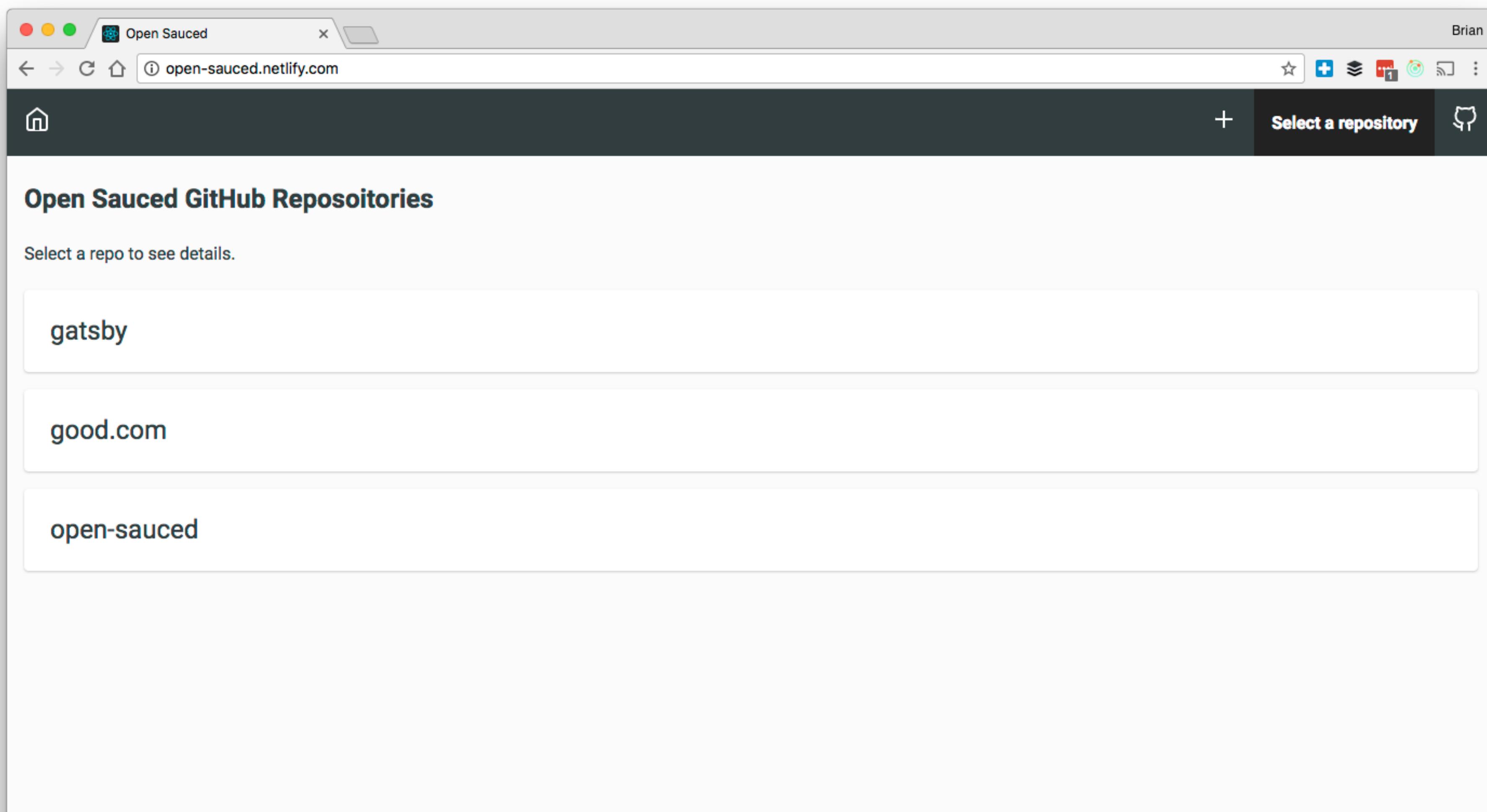
an **open source** project to
manage **open source** projects



If you don't got sauce, then
you lost

~Gucci Mane

opensauced.pizza



@bdougieYO

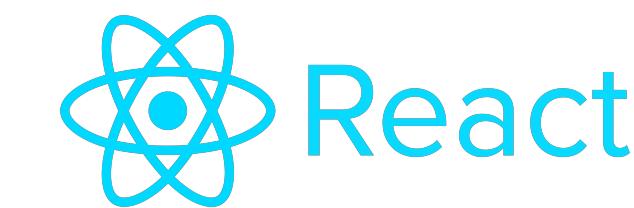
**GitHub
GraphQL
API**

Fetching repo data

Prisma

Fetching saved notes

Saving notes



**Open
Sauced**



a way to **save** and **take notes**
on projects to rediscover later

GraphQL Interactions

Query & Mutation

Query

```
query {  
  repositoryOwner(login: "netlify") {  
    repository(name: "gotrue-js") {  
      name  
      issues {  
        totalCount  
      }  
    }  
  }  
}
```

```
{  
  "data": {  
    "repositoryOwner": {  
      "repository": {  
        "name": "gotrue-js",  
        "issues": {  
          "totalCount": 13  
        }  
      }  
    }  
  }  
}
```

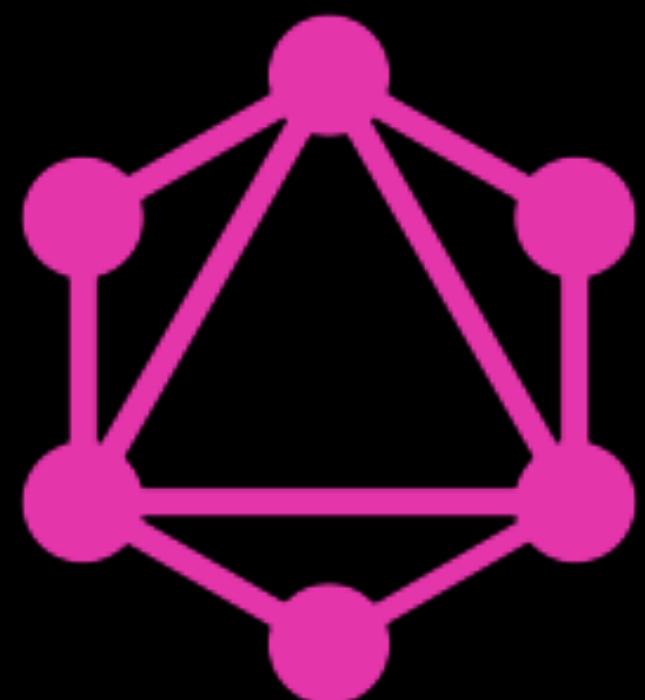
Mutation

```
mutation AddReactionToIssue {  
  addReaction(input: {subjectId: "XXXXXX=", content:HEART}) {  
    clientMutationId  
  }  
}
```



I was hungry to learn the
GitHub GraphQL API

How do I get started with the GraphQL API?





The Godfather of Trap Music

~Gucci Mane

@bdougieYO

Gucci Mane sampled “Lovin’ You”



@bdougieYO



GraphQL Explorer

The screenshot shows the GitHub GraphQL API Explorer interface. The main area displays a GraphQL query for repository issues:

```
1 query repoIssues {  
2   repositoryOwner(login: "bdougie") {  
3     repository(name: "open-sauced") {  
4       name  
5       url  
6       issues(first:1) {  
7         totalCount  
8         edges {  
9           node {  
10             id  
11             title  
12           }  
13         }  
14       }  
15     }  
16   }  
17 }
```

The results pane shows the response structure:

```
{  
  "data": {  
    "repositoryOwner": {  
      "repository": {  
        "name": "open-sauced",  
        "url": "https://github.com/bdougie/open-sauced",  
        "issues": {  
          "totalCount": 13,  
          "edges": [  
            {  
              "node": {  
                "id": "MDU6SXNzdWUyMDE5NDE5NTQ=",  
                "title": "Fix broken form - Eject the app and  
app ENV variables"  
              }  
            }  
          ]  
        }  
      }  
    }  
  }  
}
```

A large red box highlights the query text. The results pane also includes a sidebar for the `AddReactionInput` type.

developer.github.com/v4/explorer/

@bdougieYO

The screenshot shows the GitHub GraphQL API Explorer interface. At the top, it says "Signed in as bdougie. You're ready to explore! Sign out". In the main area, there's a "GraphiQL" section with a "Prettify" button. A red box highlights the "data" field in the JSON response. The response data includes information about a repository owner, repository, issues, and edges. To the right, there's a sidebar titled "AddReactionInput" with fields for "clientMutationId", "subjectId", and "content".

```
1 query repoIssues {  
2   repositoryOwner(login: "bdougie") {  
3     repository(name: "open-sauced") {  
4       name  
5       url  
6       issues(first:1) {  
7         totalCount  
8         edges {  
9           node {  
10             id  
11             title  
12           }  
13         }  
14       }  
15     }  
16   }  
17 }
```

```
{  
  "data": {  
    "repositoryOwner": {  
      "repository": {  
        "name": "open-sauced",  
        "url": "https://github.com/bdougie/open-sauced",  
        "issues": {  
          "totalCount": 13,  
          "edges": [  
            {  
              "node": {  
                "id": "MjAxMS0wMS0xNDE5NTQz",  
                "title": "Fix broken links - Eject the app and  
app ENV variables"  
              }  
            }  
          ]  
        }  
      }  
    }  
  }  
}
```

Autogenerated input type of AddReaction

FIELDS

clientMutationId: String

subjectId: ID!

content: ReactionContent!

developer.github.com/v4/explorer/

@bdougieYO

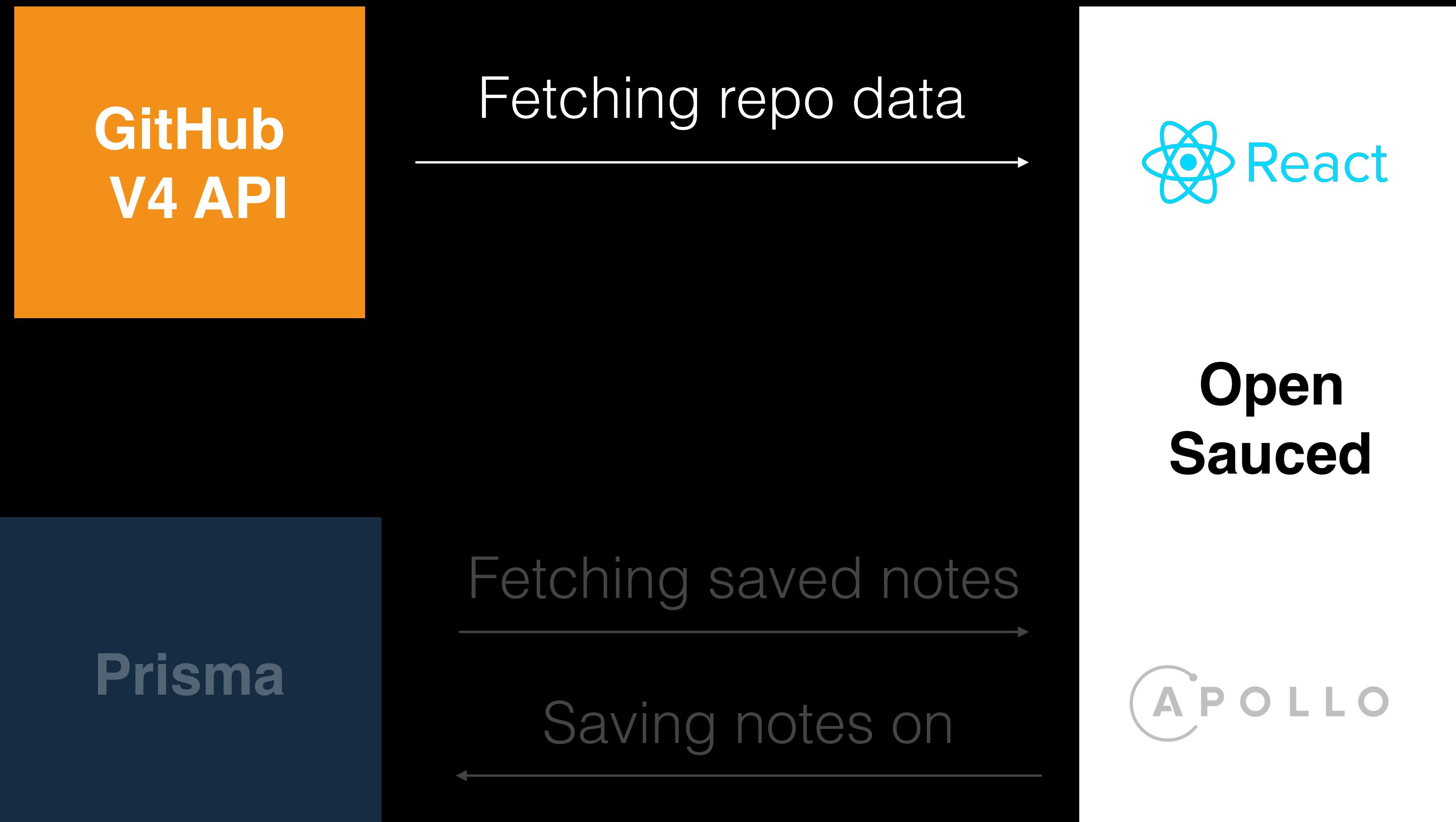
The screenshot shows the GitHub GraphQL API Explorer interface. On the left, there is a code editor window containing a GraphQL query and its variables. The query includes a mutation to add a reaction to an issue and a query for repository issues. The variables section is currently empty. On the right, a modal window titled "AddReactionInput" displays the documentation for the autogenerated input type. It includes fields for clientMutationId (String), subjectId (ID!), and content (ReactionContent!). A red box highlights this modal window. Below the modal, the word "Documentation" is displayed.

```
1 mutation AddReactionToIssue {  
2   addReaction(input: {subjectId: "MDU6SXNzdWUyMDE5NDE5NTQ=",  
3     clientMutationId  
4   }  
5 }  
6  
7 query repoIssues {  
8   repositoryOwner(login: "bdougie") {  
9     repository(name: "open-sauced") {  
10       issues(first:1) {  
11         totalCount  
12         edges {  
13           node {  
14             id  
15             title  
16           }  
17         }  
18       }  
19     }  
20   }  
21 }  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
5510  
5511  
5512  
5513  
5514  
5515  
5516  
5517  
5518  
5519  
5520  
5521  
5522  
5523  
5524  
5525  
5526  
5527  
5528  
5529  
5530  
5531  
5532  
5533  
5534  
5535  
5536  
5537  
5538  
5539  
55310  
55311  
55312  
55313  
55314  
55315  
55316  
55317  
55318  
55319  
55320  
55321  
55322  
55323  
55324  
55325  
55326  
55327  
55328  
55329  
55330  
55331  
55332  
55333  
55334  
55335  
55336  
55337  
55338  
55339  
55340  
55341  
55342  
55343  
55344  
55345  
55346  
55347  
55348  
55349  
55350  
55351  
55352  
55353  
55354  
55355  
55356  
55357  
55358  
55359  
55360  
55361  
55362  
55363  
55364  
55365  
55366  
55367  
55368  
55369  
55370  
55371  
55372  
55373  
55374  
55375  
55376  
55377  
55378  
55379  
55380  
55381  
55382  
55383  
55384  
55385  
55386  
55387  
55388  
55389  
55390  
55391  
55392  
55393  
55394  
55395  
55396  
55397  
55398  
55399  
553100  
553101  
553102  
553103  
553104  
553105  
553106  
553107  
553108  
553109  
553110  
553111  
553112  
553113  
553114  
553115  
553116  
553117  
553118  
553119  
553120  
553121  
553122  
553123  
553124  
553125  
553126  
553127  
553128  
553129  
553130  
553131  
553132  
553133  
553134  
553135  
553136  
553137  
553138  
553139  
553140  
553141  
553142  
553143  
553144  
553145  
553146  
553147  
553148  
553149  
553150  
553151  
553152  
553153  
553154  
553155  
553156  
553157  
553158  
553159  
553160  
553161  
553162  
553163  
553164  
553165  
553166  
553167  
553168  
553169  
553170  
553171  
553172  
553173  
553174  
553175  
553176  
553177  
553178  
553179  
553180  
553181  
553182  
553183  
553184  
553185  
553186  
553187  
553188  
553189  
553190  
553191  
553192  
553193  
553194  
553195  
553196  
553197  
553198  
553199  
553200  
553201  
553202  
553203  
553204  
553205  
553206  
553207  
553208  
553209  
553210  
553211  
553212  
553213  
553214  
553215  
553216  
553217  
553218  
553219  
553220  
553221  
553222  
553223  
553224  
553225  
553226  
553227  
553228  
553229  
553230  
553231  
553232  
553233  
553234  
553235  
553236  
553237  
553238  
553239  
553240  
553241  
553242  
553243  
553244  
553245  
553246  
553247  
553248  
553249  
553250  
553251  
553252  
553253  
553254  
553255  
553256  
553257  
553258  
553259  
553260  
553261  
553262  
553263  
553264  
553265  
553266  
553267  
553268  
553269  
553270  
553271  
553272  
553273  
553274  
553275  
553276  
553277  
553278  
553279  
553280  
553281  
553282  
553283  
553284  
553285  
553286  
553287  
553288  
553289  
553290  
553291  
553292  
553293  
553294  
553295  
553296  
553297  
553298  
553299  
553300  
553301  
553302  
553303  
553304  
553305  
553306  
553307  
553308  
553309  
553310  
553311  
553312  
553313  
553314  
553315  
553316  
553317  
553318  
553319  
553320  
553321  
553322  
553323  
553324  
553325  
553326  
553327  
553328  
553329  
553330  
553331  
553332  
553333  
553334  
553335  
553336  
553337  
553338  
553339  
5533310  
5533311  
5533312  
5533313  
5533314  
5533315  
5533316  
5533317  
5533318  
5533319  
55333110  
55333111  
55333112  
55333113  
55333114  
55333115  
55333116  
55333117  
55333118  
55333119  
553331110  
553331111  
553331112  
553331113  
553331114  
553331115  
553331116  
553331117  
553331118  
553331119  
5533311110  
5533311111  
5533311112  
5533311113  
5533311114  
5533311115  
5533311116  
5533311117  
5533311118  
5533311119  
55333111110  
55333111111  
55333111112  
55333111113  
55333111114  
55333111115  
55333111116  
55333111117  
55333111118  
55333111119  
553331111110  
553331111111  
553331111112  
553331111113  
553331111114  
553331111115  
553331111116  
553331111117  
553331111118  
553331111119  
5533311111110  
5533311111111  
5533311111112  
5533311111113  
5533311111114  
5533311111115  
5533311111116  
5533311111117  
5533311111118  
5533311111119  
55333111111110  
55333111111111  
55333111111112  
55333111111113  
55333111111114  
55333111111115  
55333111111116  
55333111111117  
55333111111118  
55333111111119  
553331111111110  
553331111111111  
553331111111112  
553331111111113  
553331111111114  
553331111111115  
553331111111116  
553331111111117  
553331111111118  
553331111111119  
5533311111111110  
5533311111111111  
5533311111111112  
5533311111111113  
5533311111111114  
5533311111111115  
5533311111111116  
5533311111111117  
5533311111111118  
5533311111111119  
55333111111111110  
55333111111111111  
55333111111111112  
55333111111111113  
55333111111111114  
55333111111111115  
55333111111111116  
55333111111111117  
55333111111111118  
55333111111111119  
553331111111111110  
553331111111111111  
553331111111111112  
553331111111111113  
553331111111111114  
553331111111111115  
553331111111111116  
553331111111111117  
553331111111111118  
553331111111111119  
5533311111111111110  
5533311111111111111  
5533311111111111112  
5533311111111111113  
5533311111111111114  
5533311111111111115  
5533311111111111116  
5533311111111111117  
5533311111111111118  
5533311111111111119  
55333111111111111110  
55333111111111111111  
55333111111111111112  
55333111111111111113  
55333111111111111114  
5533311111
```

I sampled GitHub to
build my integration

stitching my app with the GitHub API

Fetching Data



A screenshot of a web browser window titled "Open Sauced". The URL in the address bar is "open-sauced.netlify.com/new". The page content is a form for entering a GitHub URL to track repository data. The input field contains "https://github.com/netlify/netlify-cms". A teal button labeled "Fetch repository data" is visible. Below the input field, a list of repository details is shown in separate boxes: "netlify-cms", "https://github.com/netlify/netlify-cms", "netlify", "Contributors", "3050", "382", "365", and "A CMS for Static Site Generators". At the bottom left, a message says "Currently 3 repositories in database". A teal button at the bottom left says "Add repository to your list". The top right corner of the browser window shows the name "Brian".

Fetching Repo Data

@bdougieYO

GraphQL != JavaScript

Fetching Repo Data

```
import axios from "axios";

const auth = {Authorization: `bearer ${process.env.githubToken}`};
const githubUrl = "https://api.github.com/graphql";

fetchRepositoryData(name, repo) {
  return axios.post(githubUrl, {
    query: repoQuery(name, repo)
  }, {headers: auth});
}
```

GitHub GraphQL Endpoint

```
import axios from "axios";

const auth = {Authorization: `bearer ${process.env.githubToken}`};
const githubUrl = "https://api.github.com/graphql";

fetchRepositoryData(name, repo) {
  return axios.post(githubUrl, {
    query: repoQuery(name, repo)
  }, {headers: auth});
}
```

GraphQL repoQuery

```
import axios from "axios";

const auth = {Authorization: `bearer ${process.env.githubToken}`};
const githubUrl = "https://api.github.com/graphql";

fetchRepositoryData(name, repo) {
  return axios.post(githubUrl, {
    query: repoQuery(name, repo)
  }, {headers: auth});
}
```

```
function repoQuery(name, repo) {
  return `

    repositoryOwner(login: "${name}") {
      repository(name: "${repo}") {
        name
        url
        owner {
          login
          repositories {
            totalCount
          }
        }
        description
        forks {
          totalCount
        }
        issues {
          totalCount
        }
        stargazers {
          totalCount
        }
      }
    }
  `;
}
```

repoQuery

the exact shape from the GitHub explorer

```
function repoQuery(name, repo) {
  return `

    repositoryOwner(login: "${name}") {
      repository(name: "${repo}") {
        name
        url
        owner {
          login
          repositories {
            totalCount
          }
        }
        description
        forks {
          totalCount
        }
        issues {
          totalCount
        }
        stargazers {
          totalCount
        }
      }
    }
  `;
}
```

repoQuery

GitHub GraphQL API

Heads up! GitHub's GraphQL Explorer makes use of your real, live, production data.

GraphQL  Prettify

```
1 query repoQuery {
2   repositoryOwner(login: "bdougie") {
3     repository(name: "open-sauced") {
4       name
5       url
6       owner {
7         login
8         repositories {
9           totalCount
10      }
11    }
12    description
13    forks {
14      totalCount
15    }
16    issues {
17      totalCount
18    }
19    stargazers {
```

QUERY VARIABLES

```
1 {}
```

Pagination on Edges

< issues

IssueConnection

X

The connection type for Issue.

FIELDS

edges: [IssueEdge]

nodes: [Issue]

pageInfo: PageInfo!

totalCount: Int!

[Issue]

GraphQL API Explorer | GitHub x Brian

Secure | https://developer.github.com/v4/explorer/

GitHub GraphQL API

Heads up! GitHub's GraphQL Explorer makes use of your real, live, production data.

GraphiQL

Prettify

```
1 query repoIssues {
2   repositoryOwner(login: "bdougie") {
3     repository(name: "open-sauced") {
4       name
5       url
6       issues(first:1) {
7         totalCount
8         edges {
9           node {
10             id
11             title
12           }
13         }
14       }
15     }
16   }
17 }
```

QUERY VARIABLES

issues

A list of issues that have been opened in the repository.

TYPE

IssueConnection!

ARGUMENTS

first: Int
Returns the first *n* elements from the list.

last: Int
Returns the last *n* elements from the list.

before: String
Returns the elements in the list that come before the specified global ID.

after: String
Returns the elements in the list that come after the specified global ID.

Show All

@bdougieYO

The screenshot shows the GitHub GraphQL API Explorer interface. In the top navigation bar, it says "GraphQL API Explorer | GitHub". Below that, the URL is "Secure | https://developer.github.com/v4/explorer/". The main area is titled "GitHub GraphQL API" and contains a message: "Heads up! GitHub's GraphQL Explorer makes use of your real, live, production data.". Below this is a "GraphiQL" section with a "Prettify" button. A code editor window displays the following GraphQL query:

```
query repoIssues {
  repositoryOwner(login: "bdougie") {
    repository(name: "open-sauced") {
      name
      url
      issues(first:1) {
        totalCount
        edges {
          node {
            id
            title
          }
        }
      }
    }
  }
}
```

At the bottom of the GraphiQL window, there are two dropdown menus labeled "ezgif.com-video-to-gif (...gif)".

A modal window titled "Search Results" is open, with the search term "issues" in the center. The window contains the following text: "A list of issues that have been opened in the repository." Below this, under the heading "TYPE", is the text "IssueConnection!". Under "ARGUMENTS", there are four entries: "first: Int", "after: String", "last: Int", and "before: String". Each argument is described with its purpose: "first: Int" returns the first *n* elements from the list, "after: String" returns the elements in the list that come after the specified global ID, "last: Int" returns the last *n* elements from the list, and "before: String" returns the elements in the list that come before the specified global ID.

Node Limit

```
query {
  viewer {
    repositories(first: 50) {
      edges {
        repository:node {
          name

          issues(first: 10) {
            totalCount
            edges {
              node {
                title
                bodyHTML
              }
            }
          }
        }
      }
    }
  }
}
```

To pass [schema validation](#), all GraphQL API v4 [calls](#) must meet these standards:

- Clients must supply a `first` or `last` argument on any [connection](#).
- Values of `first` and `last` must be within 1-100.
- Individual calls cannot request more than 500,000 total [nodes](#).

Fetching Issues

```
query {
  repositoryOwner(login: "${name}") {
    repository(name: "${repo}") {
      issues(first: 5, states: OPEN, orderBy: {
        field: CREATED_AT, direction: DESC
      }) {
        totalCount
      }
    }
  }
}
```

Limit is 100 records

```
query {
  repositoryOwner(login: "${name}") {
    repository(name: "${repo}") {
      issues(first: 5, states: OPEN, orderBy: {
        field: CREATED_AT, direction: DESC
      }) {
        totalCount
      }
    }
  }
}
```

Issues

Add prefix links option during development

Bypass react router path matching

Add API to allow plugins to add/modify CLI commands

Add Flow support

Add markdown-doctest to tests

Ordering and Filter happens in the Query

```
query {  
  repositoryOwner(login: "${name}") {  
    repository(name: "${repo}") {  
      issues(first: 5, states: OPEN, orderBy: {  
        field: CREATED_AT, direction: DESC  
      }) {  
        totalCount  
      }  
    }  
  }  
}
```

Filter by only open issues

```
query {
  repositoryOwner(login: "${name}") {
    repository(name: "${repo}") {
      issues(first: 5, states: OPEN, orderBy: {
        field: CREATED_AT, direction: DESC
      }) {
        totalCount
      }
    }
  }
}
```

Order by latest issue

```
query {
  repositoryOwner(login: "${name}") {
    repository(name: "${repo}") {
      issues(first: 5, states: OPEN, orderBy: {
        field: CREATED_AT, direction: DESC
      }) {
        totalCount
      }
    }
  }
}
```

Paginated Issues

Add prefix links option during development

Bypass react router path matching

Add API to allow plugins to add/modify CLI commands

Add Flow support

Add markdown-doctest to tests

< 4/68 >

Paginating Issue Data

```
query {
  repositoryOwner(login: "bdougie") {
    repository(name: "open-sauced") {
      name
      issues(first: 5, states: OPEN, orderBy: {
        field: CREATED_AT, direction: DESC
      }) {
        totalCount
        data: edges {
          cursor
          node {
            id
            title
            url
            state
            author {
              login
            }
          }
        }
      }
    }
  }
}
```

Alias edges to data

```
query {
  repositoryOwner(login: "bdougie") {
    repository(name: "open-sauced") {
      name
      issues(first: 5, states: OPEN, orderBy: {
        field: CREATED_AT, direction: DESC
      }) {
        totalCount
        data: edges {
          cursor
          node {
            id
            title
            url
            state
            author {
              login
            }
          }
        }
      }
    }
  }
}
```

All edge content has a cursor

```
query {
  repositoryOwner(login: "bdougie") {
    repository(name: "open-sauced") {
      name
      issues(first: 5, states: OPEN, orderBy: {
        field: CREATED_AT, direction: DESC
      }) {
        totalCount
        data: edges {
          cursor
          node {
            id
            title
            url
            state
            author {
              login
            }
          }
        }
      }
    }
  }
}
```

Offsetting edge content with cursor

```
issues(first: 5, states: OPEN, after: "${cursor}")
```

Saving Data

updateRepository

```
mutation updateRepository($notes: String!, $id: ID!) {  
  updateRepository(id: $id, notes: $notes) {  
    id  
    notes  
  }  
}
```

Who is using the GitHub GraphQL API?

GitHub

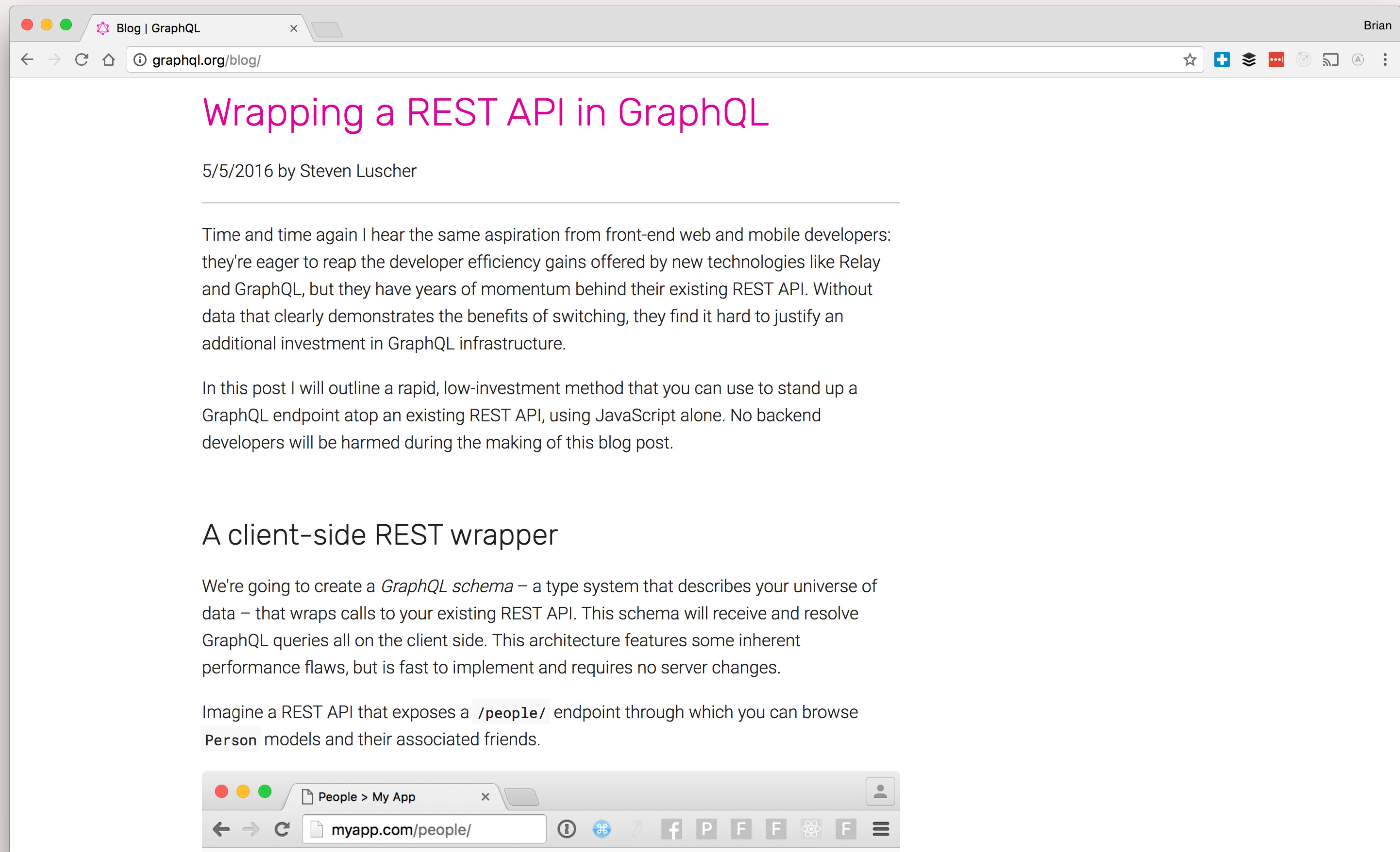
(200+ contributors)



wrappers?

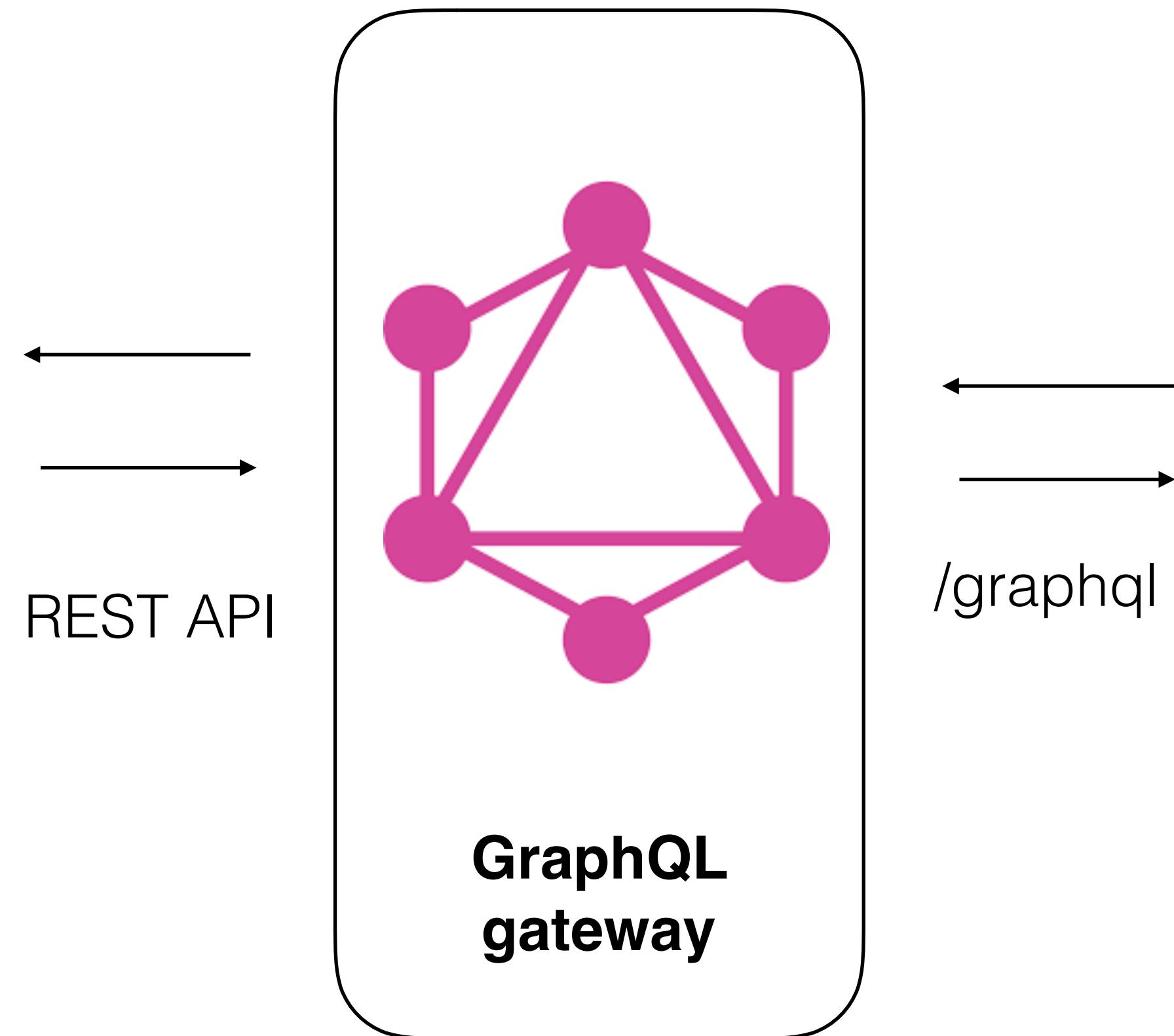
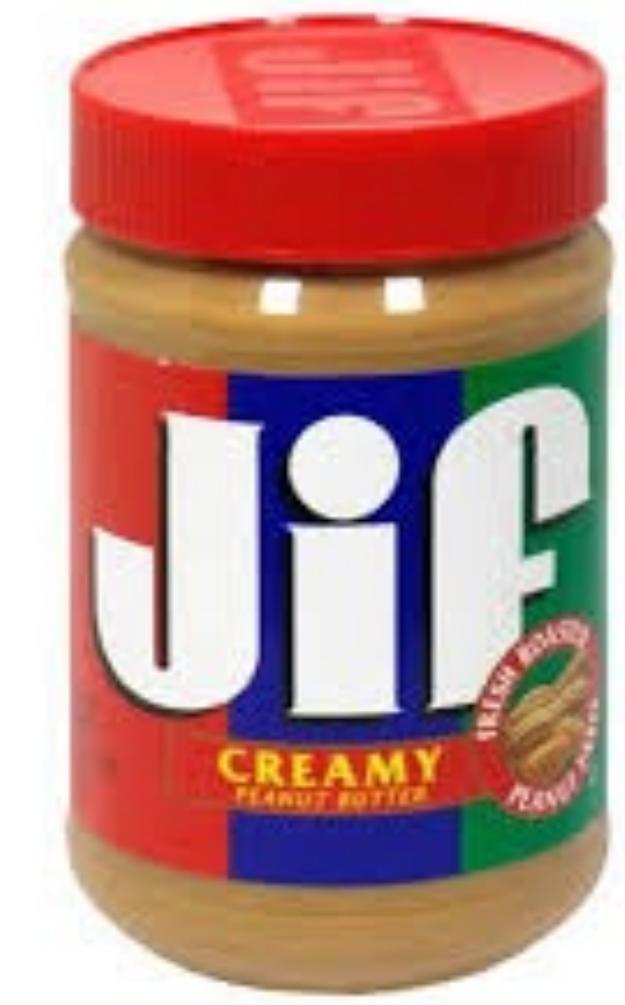


@bdougieYO



The screenshot shows a web browser window with the following details:

- Title Bar:** "Blog | GraphQL" and "Brian".
- Address Bar:** "graphql.org/blog/".
- Content Area:**
 - # Wrapping a REST API in GraphQL
 - 5/5/2016 by Steven Luscher
 - Time and time again I hear the same aspiration from front-end web and mobile developers: they're eager to reap the developer efficiency gains offered by new technologies like Relay and GraphQL, but they have years of momentum behind their existing REST API. Without data that clearly demonstrates the benefits of switching, they find it hard to justify an additional investment in GraphQL infrastructure.
 - In this post I will outline a rapid, low-investment method that you can use to stand up a GraphQL endpoint atop an existing REST API, using JavaScript alone. No backend developers will be harmed during the making of this blog post.
- Section Header:** "A client-side REST wrapper"
- Text:** "We're going to create a *GraphQL schema* – a type system that describes your universe of data – that wraps calls to your existing REST API. This schema will receive and resolve GraphQL queries all on the client side. This architecture features some inherent performance flaws, but is fast to implement and requires no server changes."
- Text:** "Imagine a REST API that exposes a `/people/` endpoint through which you can browse `Person` models and their associated friends."
- Image:** A screenshot of a browser window titled "People > My App" showing the URL "myapp.com/people/" in the address bar. The browser interface includes standard navigation buttons, a search bar, and various icons for different applications and services.



I sampled my
REST API to power my
GraphQL API

**GitHub is powering their
REST API with the
GraphQL API**

REST v GraphQL

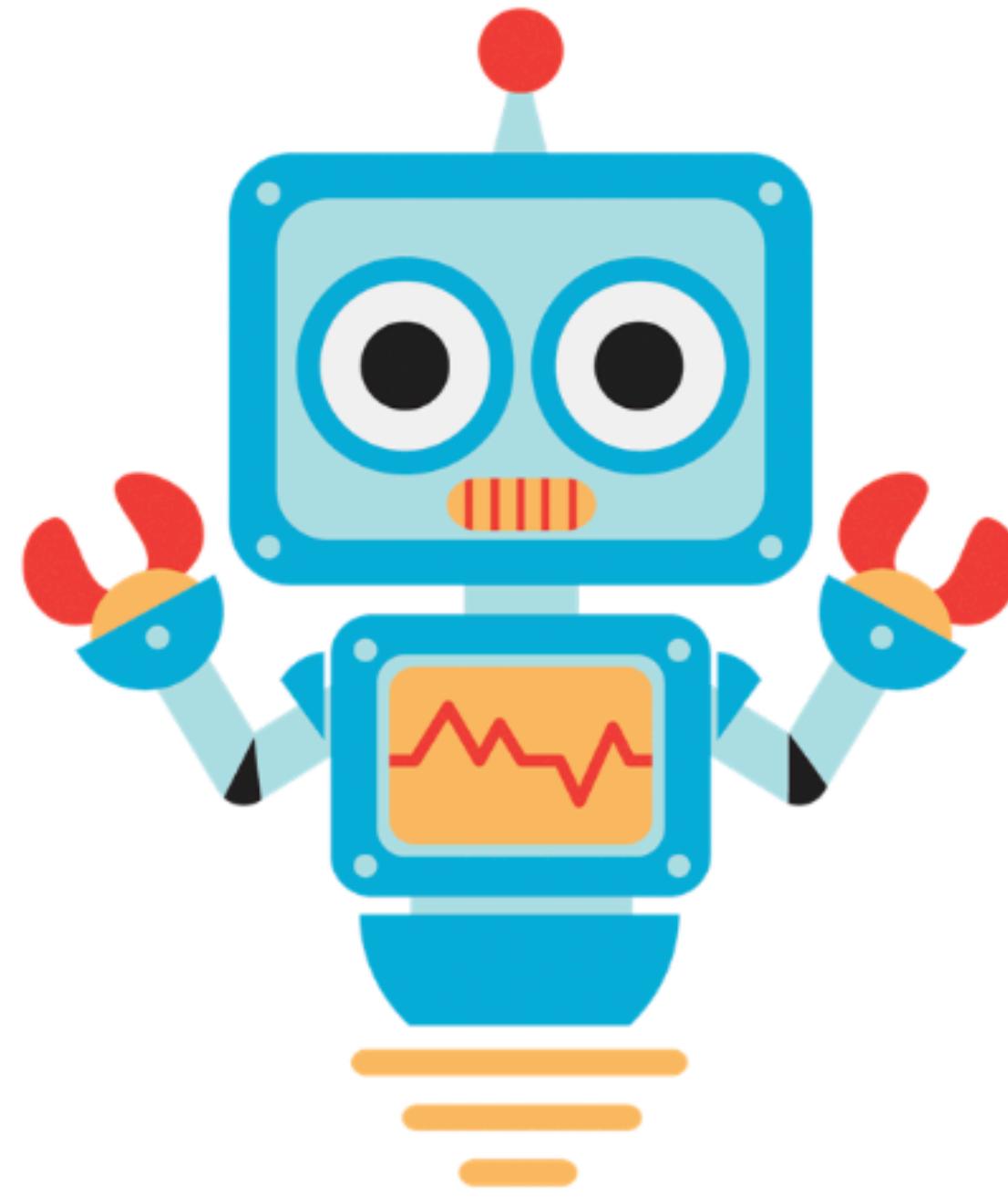
REST + GraphQL

The GitHub REST API is
not going anywhere

Improve you're experience
with GitHub Apps

GitHub Apps

allow integrators the flexibility to request specific access to resources and act as a first class actor without taking up a paid seat within an organization.



PROBOT

GitHub Apps to automate and improve your workflow

Use pre-built apps to extend GitHub,
and easily build and share your own.

probot.github.io



@bdougieYO

automate workflows

By leveraging the GitHub API



@bdougieYO

identify repeated tasks

“We *{action}*
every *{period of time}*.”



pull reminders

Receive DMs for reviews & requests



Pull Reminders 2:00 PM

@ben requested your review on [UI Hotfix](#) (ios)

@isaac approved [DB Indexes](#) (api)



Pull Reminders 4:00 PM

@jane requested changes on [DB Indexes](#) (api)



@bdougieYO

identify repeated tasks

*“We review **pull request**
once a day.”*



listen to hooks

Take action via the GitHub API



@bdougieYO

Brand new issue! #54

[Edit](#)[New issue](#)[Open](#)

hiimbex opened this issue 27 seconds ago · 0 comments



hiimbex commented 27 seconds ago



hello world!



bex-is-the-best bot

added the **needs-response** label 26 seconds ago[Write](#)[Preview](#)

Leave a comment

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

M Styling with Markdown is supported

[Close issue](#)[Comment](#)**Assignees**

No one—assign yourself

Labels**needs-response****Projects**

None yet

Milestone

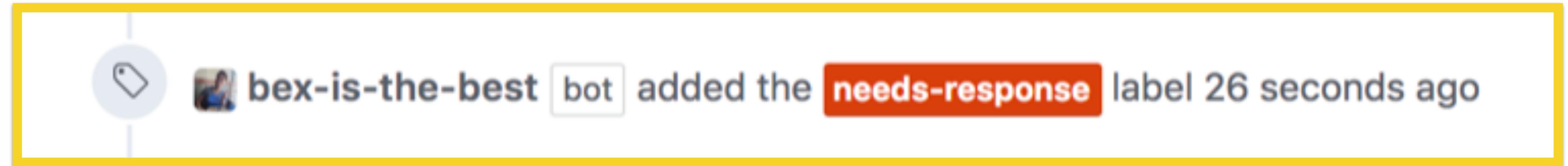
No milestone

Notifications[Unsubscribe](#)

You're receiving notifications because you authored the thread.



@bdougieYO



hiimbex commented 27 seconds ago

hello world!

bex-is-the-best bot added the **needs-response** label 26 seconds ago

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

M Styling with Markdown is supported

[Close issue](#) [Comment](#)

Assignees

No one—assign yourself

Labels

needs-response

Projects

None yet

Milestone

No milestone

Notifications

[Unsubscribe](#)

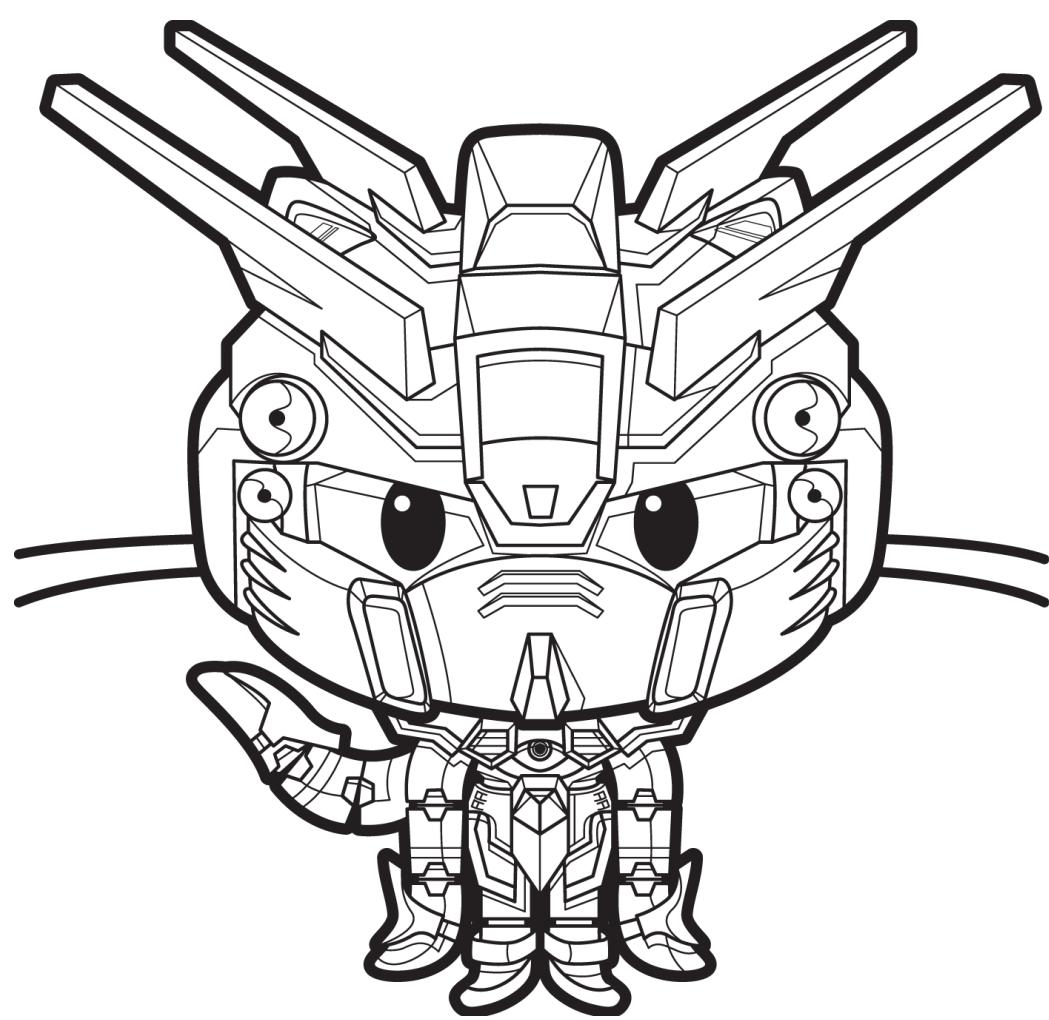
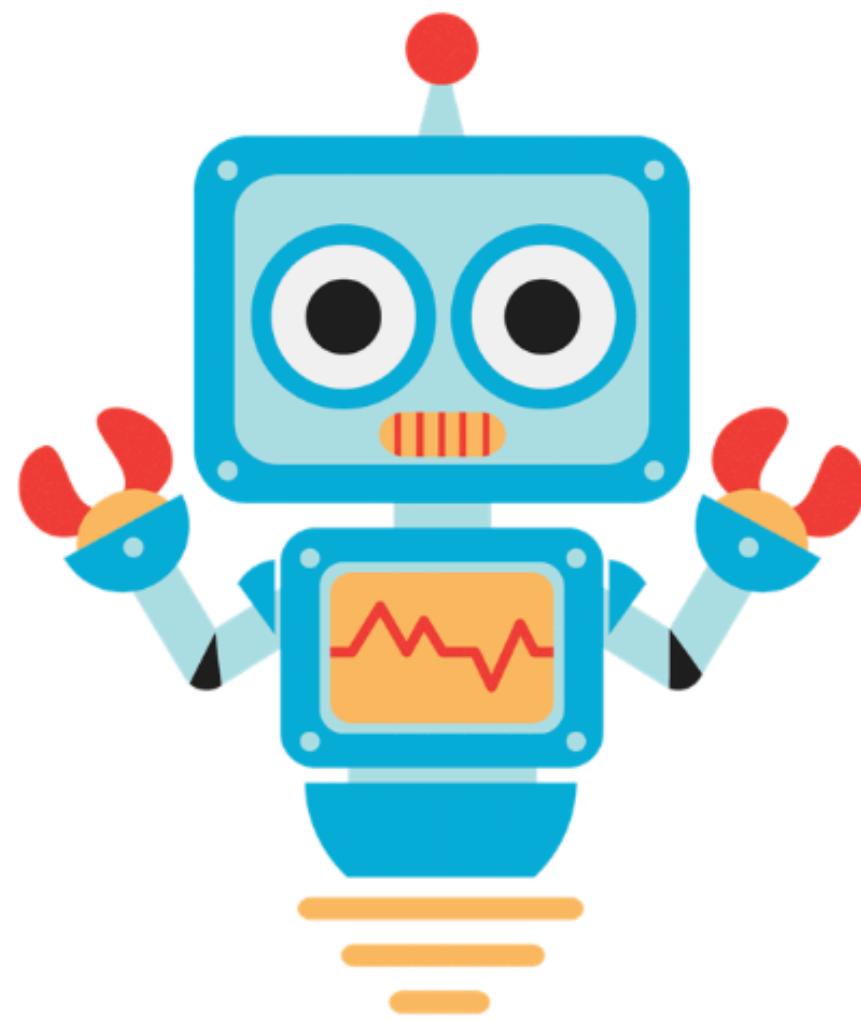
You're receiving notifications because you authored the thread.



@bdougieYO

Probot + Octokit

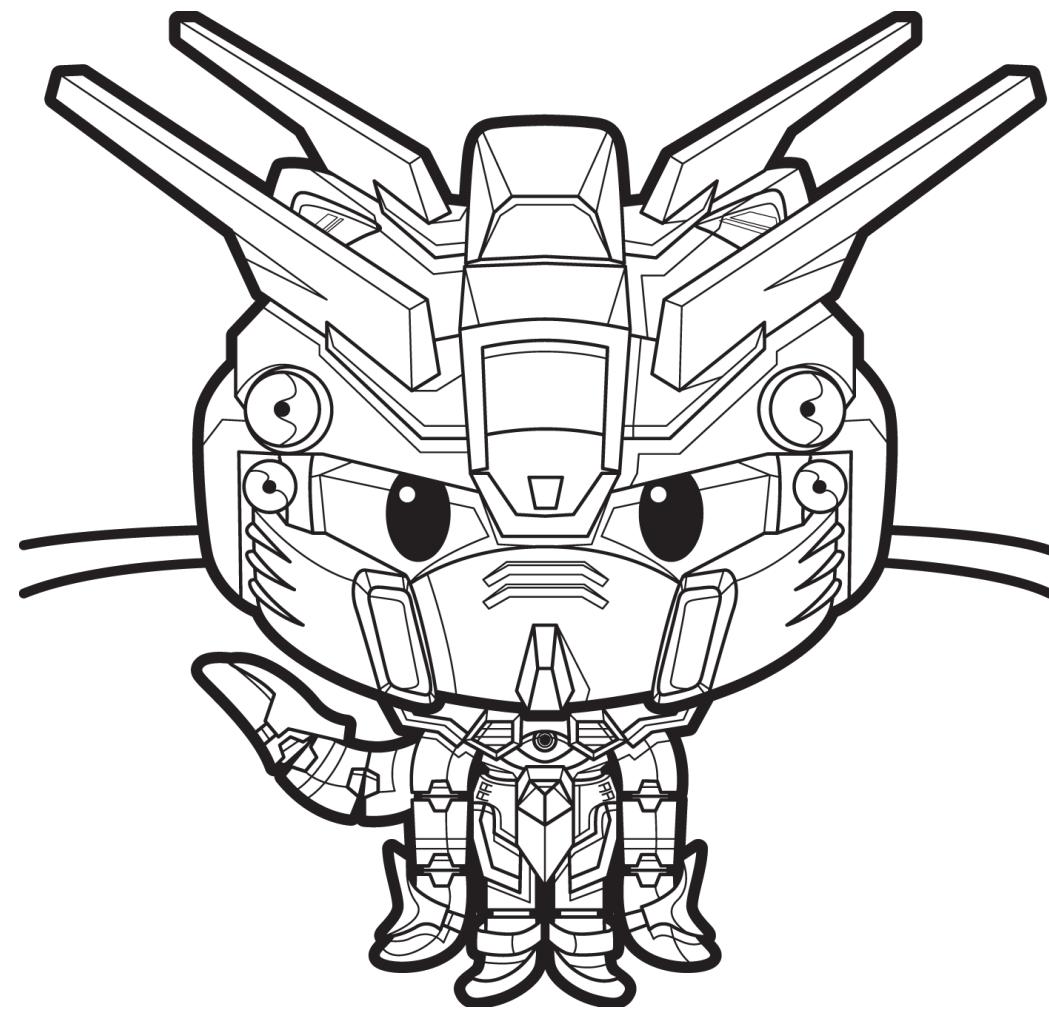
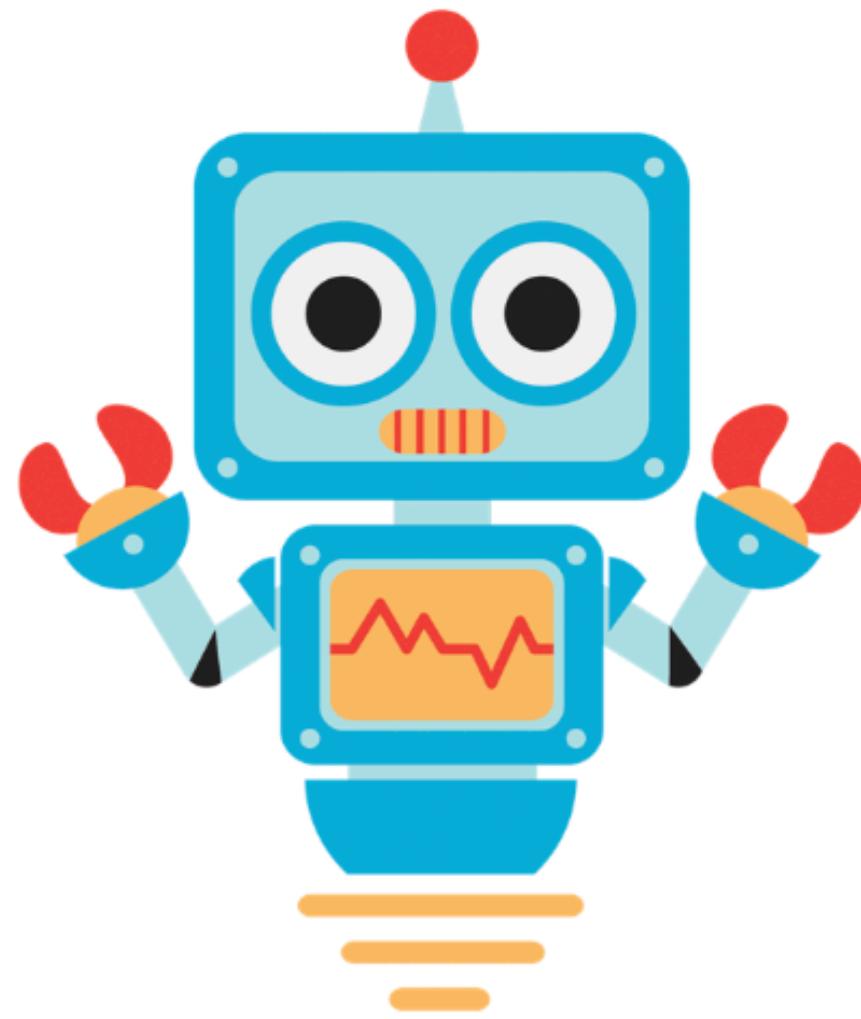
```
  . . .
  app.on('issues.opened', async context => {
    context.github.issues.addLabels(
      context.issue({ labels: ['needs-response'] })
    )
  }
}
```



@bdougieYO

GraphQL + Octokit

```
const addLabel = `mutation label($id: ID!, $body: String!) {  
  addLabel(input: {subjectId: $id, body: $body}) {  
    clientMutationId  
  }  
}  
  
app.on('issues.labeled', async context => {  
  context.github.query(addLabel, {  
    id: context.payload.issue.node_id,  
    body: 'needs-reponse'  
  })  
})
```



@bdouieYO

**the GraphQL API provides opportunity
to define interactions based on
solutions and not endpoints**

all contributors in x repos



```
const allContributorsInXRepos = (count) => `

organization(login: "github") {
  name
  repositories(first: ${count}, orderBy: {field: NAME, direction: ASC}) {
    totalCount
    edges {
      node {
        id
        name
        description
        collaborators(first: 10, affiliation: ALL) {
          totalCount
          edges {
            permission
            node {
              id
              login
              name
            }
          }
        }
      }
    }
  }
}
```

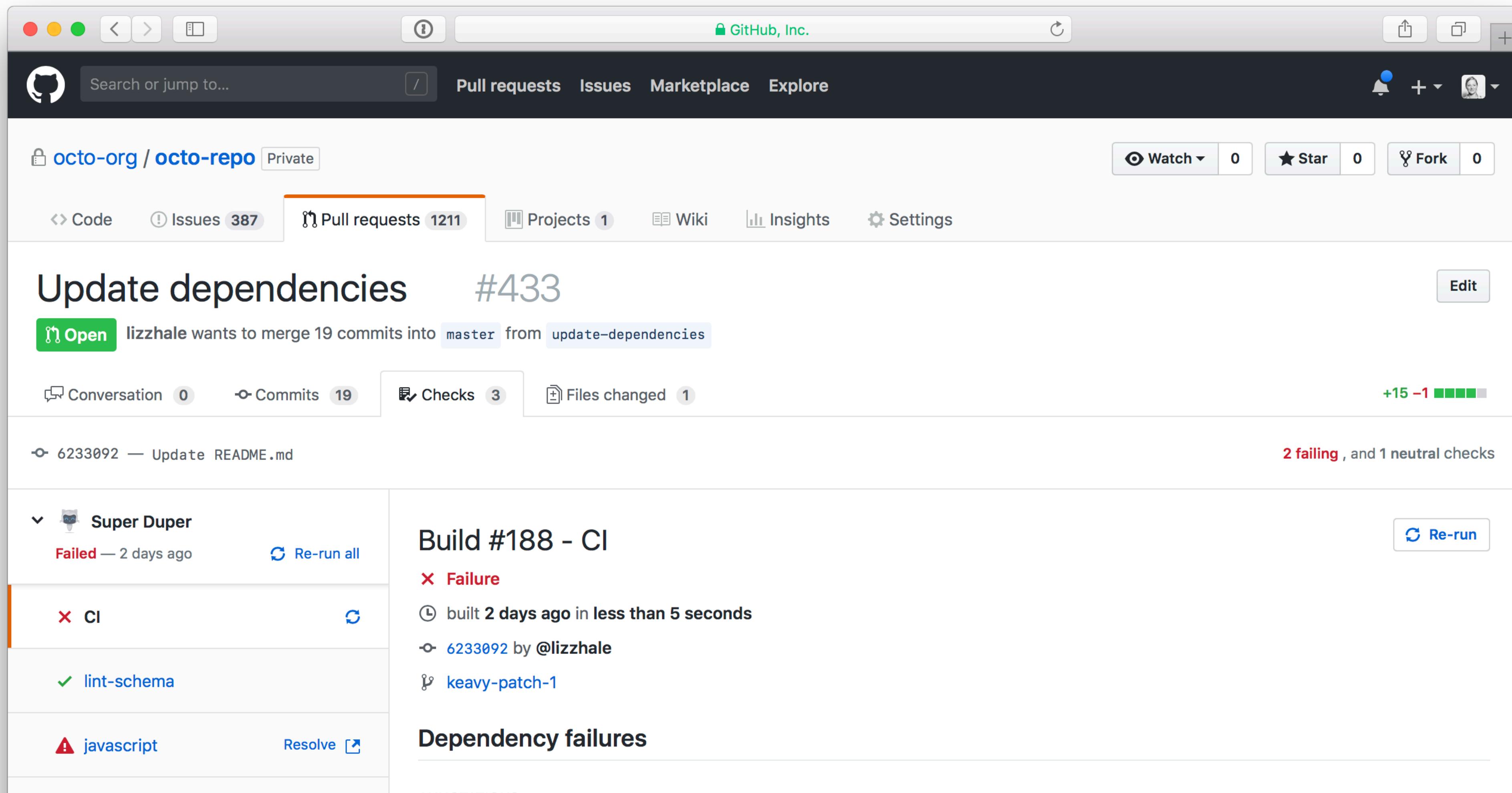


@bdougieYO

the GitHub API is constantly
getting new interactions

Checks API

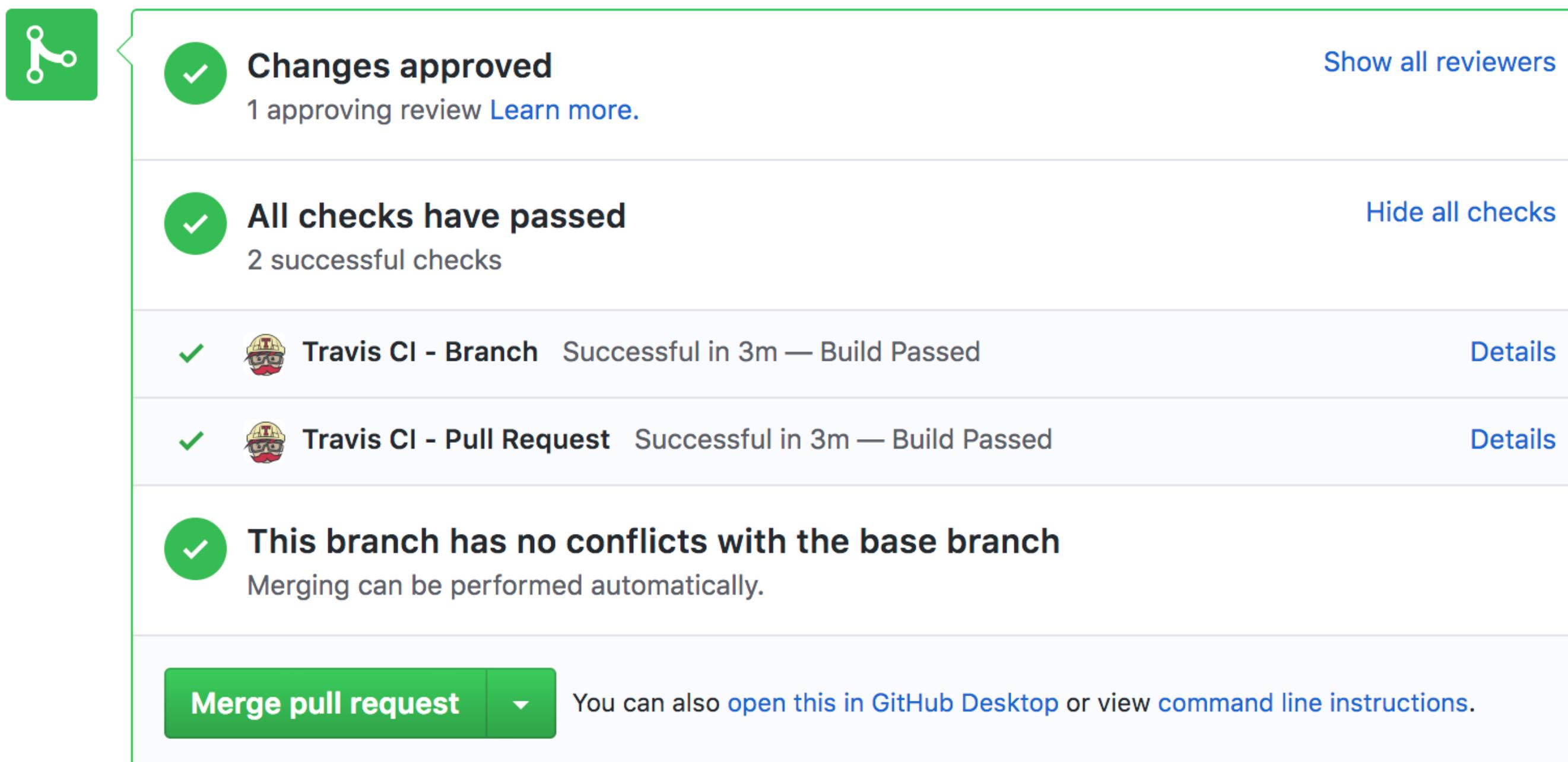
Checks API: Sophisticated feedback



@bdougieYO

GitHub App: TravisCI

- Test and Deploy with Confidence.



The screenshot shows a GitHub Checks interface for a pull request. At the top, there are tabs for Conversation (14), Commits (36), Checks (2), and Files changed (15). The Checks tab is active, showing 2 successful checks. A summary bar indicates +159 green, -18 red, and 5 yellow. Below the tabs, a list of checks is shown:

- d356155 — Remove rollout check
- Travis CI - Pull Request (Succeeded — an hour ago)
- Travis CI - Branch
- Travis CI - Pull Request

A "Build Passed" message is displayed with the note: "The build passed. This is a change from the previous build, which failed." Below this, a "DETAILS" section states: "This is a pull request build. It is running a build against the merge commit, after merging #1 [WIP] Add GitHub Apps logic. Any changes that have been made to the master branch before the build ran are also included." The "Jobs and Stages" section shows two stages: "Stage 1: Testing time" (passed) and "Stage 2: 🚀 it to quay.io".

Checks share more information

Add Beyonce lyrics #1



bougie wants to merge 1 commit into `master` from `add-yonce-lyrics`

Conversation 0

Commits 1

Checks 1

Files changed 1



bdougie commented a minute ago

Owner



...

"Check on It" is a song recorded by American singer Beyoncé, featuring American rappers Bun B and Slim Thug. It was composed by Beyoncé, Swizz Beatz, Sean Garrett, Angela Beyincé, and Slim Thug. Initially recorded by Beyoncé only, the song was supposed to be featured on the soundtrack album for 2006 film The Pink Panther, in which she co-stars as Xania. As it was ultimately not included on its soundtrack album, the song with additional vocals from Slim Thug was placed on the 2005 Destiny's Child's greatest hits album #1's. Columbia Records released "Check on It" in the United States on December 13, 2005. The official single version of the song includes vocals from American rapper Bun B.



Update README.md

Verified

7bae082

Add more commits by pushing to the `add-yonce-lyrics` branch on `bougie/check-on-it`.



All checks have passed

1 neutral check

Hide all checks



alex feedback Completed in 4s — analysis

Details



This branch has no conflicts with the base branch

Merging can be performed automatically.

“Check on it” pull request

@bougieYO

github.com/apps/linter-alex



linter-alex

Configure
Manage your installation settings.

Description

Ensure sensitive, considerate writing before you merge your Pull Requests.

Details

swinton
[linter-alex's website](#)

linter-alex is provided by a third-party and is governed by separate terms, privacy, and support documentation.

[Report abuse](#)

Add Beyonce lyrics #1

Open bdougie wants to merge 1 commit into master from add-yonce-lyrics

Conversation 0 Commits 1 Checks 1 Files changed 1

bdougie commented a minute ago

"Check on It" is a song recorded by American singer Beyoncé, featuring American rappers Bun B and Slim Thug. It was composed by Beyoncé, Swizz Beatz, Sean Garrett, Angela Beyincé, and Slim Thug. Initially recorded by Beyoncé only, the song was supposed to be featured on the soundtrack album for 2006 film The Pink Panther, in which she co-stars as Xania. As it was ultimately not included on its soundtrack album, the song with additional vocals from Slim Thug was placed on the 2005 Destiny's Child's greatest hits album #1's. Columbia Records released "Check on It" in the United States on December 13, 2005. The official single version of the song includes vocals from American rapper Bun B.

Update README.md Verified 7bae082

Add more commits by pushing to the add-yonce-lyrics branch on bdougie/check-on-it.

All checks have passed 1 neutral check

alex feedback Completed in 4s — analysis Details

This branch has no conflicts with the base branch Merging can be performed automatically.

New Checks tab in the UI

@bdougieYO

Add Beyonce lyrics #1

Open bdougie wants to merge 1 commit into master from add-yonce-lyrics

Conversation 0 Commits 1 Checks 1 Files changed 1 +53 -2

- 7bae082 — Update README.md 1 neutral check

alex linter-alex Neutral a minute ago

feedback

- Neutral

ran a minute ago in less than 5 seconds with 0 failures

7bae082 by @bdougie

add-yonce-lyrics

analysis

Alex found 15 issues

ANNOTATIONS

Check notice on line 4 in README.md
alex linter-alex / feedback
boy-girl
'girl' may be insensitive, use 'kid', 'child' instead

Check notice on line 4 in README.md
alex linter-alex / feedback
girl
Be careful with "girl", it's profane in some cases

Check notice on line 6 in README.md
alex linter-alex / feedback
girl
Be careful with "girl", it's profane in some cases

Checks dashboard provides more sophisticated feedback for the developer

Add Beyonce lyrics #1

Open bdougie wants to merge 1 commit into master from add-yonce-lyrics

Conversation 0 Commits 1 Checks 1 Files changed 1 +53 -2

- 7bae082 — Update README.md 1 neutral check

alex linter-alex Neutral a minute ago

feedback

- Neutral
- ran a minute ago in less than 5 seconds with 0 failures
- 7bae082 by @bdougie
- add-yonce-lyrics

analysis

ANNOTATIONS

Check notice on line 4 in README.md

alex linter-alex / feedback

boy-girl

`girl` may be insensitive, use `kid`, `child` instead

Check notice on line 4 in README.md

alex linter-alex / feedback

girl

Be careful with "girl", it's profane in some cases

Check notice on line 6 in README.md

alex linter-alex / feedback

girl

Be careful with "girl", it's profane in some cases

Annotations:

- Check notice on line 4 in README.md
alex linter-alex / feedback
boy-girl
`girl` may be insensitive, use `kid`, `child` instead
- Check notice on line 4 in README.md
alex linter-alex / feedback
girl
Be careful with "girl", it's profane in some cases
- Check notice on line 6 in README.md
alex linter-alex / feedback
girl
Be careful with "girl", it's profane in some cases

Checks dashboard provides more sophisticated feedback for the developer

Add Beyonce lyrics #1

Edit

Open bdougie wants to merge 1 commit into `master` from `add-yonce-lyrics`

Conversation 0 Commits 1 Checks 1 Files changed 1 +53 -2

-o 7bae082 — Update README.md 1 neutral check

alex linter-alex Neutral a minute ago

■ feedback

■ Neutral
🕒 ran a minute ago in less than 5 seconds with 0 failures
-o 7bae082 by @bdougie
↳ add-yonce-lyrics

analysis

13 + What's practical is logical, what the hell, who cares?

ⓘ Check notice on line 13 in README.md

alex linter-alex / feedback

hell

Be careful with “hell”, it’s profane in some cases

14 + All I know is I'm so happy when you're dancing there

15 + I'm a slave for you, I cannot hold it, I cannot control it

alex linter-alex / feedback

girl

Be careful with “girl”, it’s profane in some cases

@bdougieYO

You can build this for
your GraphQL workflow

graphql-doctor



graphql-doctor-dev bot reviewed 7 days ago

[View changes](#)

It looks like your pull request includes some risky changes to our GraphQL schema.

I've commented on those lines to let you know the impact of those changes. If you have questions, please ping [@github/graphql-api-code-review](#).

Our [GraphQL Style Guide](#) also includes information that might be helpful.

Happy shipping!

schema.graphql

```
...     ... @@ -14,7 +14,7 @@ type Query {  
 14      14    type Repository implements Node {  
 15      15      homepageUri: URI  
 16      16      id: ID!
```

@bdougieYO

the Checks API is currently
in developer preview

the Checks API will not work in the explorer

This part of the schema is currently available for developers to preview. During this preview period, the API may change without any advance notice. Please see the [Checks preview](#) for more details.

Note: The GraphQL resources under preview cannot be accessed via the Explorer at this time.

GraphQL Playground

The screenshot shows the GraphQL Playground interface in a browser window. The URL in the address bar is <https://www.graphqlbin.com/v2/zKQ9iO>. The playground has tabs for "allResults" and "New Tab". The main area shows a query being run against the URL <https://api.github.com/graphql>. A large red error message "Server cannot be reached" is displayed. Below the error, the response JSON is shown:

```
1 # Write your query or mutation here
2 mutation {
3   createCheckSuite
4 }
```

```
{
  "data": null,
  "errors": [
    {
      "message": "Field 'createCheckSuite' doesn't exist
on type 'Mutation'",
      "locations": [
        {
          "line": 2,
          "column": 3
        }
      ]
    }
  ]
}
```

At the bottom left, there are buttons for "QUERY VARIABLES" and "HTTP HEADERS (2)". On the right side, there is a vertical sidebar with a "SCHEMA" tab.

You can sample
GraphQL today by using
your own **GitHub** data

Use the GraphQL
Explorer to get
sampln'

developer.github.com/v4

bdougie/open-sauced



@bdougieYO