

Hello!

I'm Jake Dawkins

Open Source Engineer

Apollo GraphQL

@jakedawkins



Hello!

I'm Jake Dawkins

Open Source Engineer

Apollo GraphQL

@jakedawkins



Things to ask me about

- Working at a startup
- Working remote
- My work (GraphQL/dev tools)
- Open Source
- Networking/Social Media
- Conference/Meetup Speaking

About my Work

(trying not to break the internet)



Platform

Pricing

Developers

Enterprise

Company

Launch Studio

The Apollo Data Graph Platform

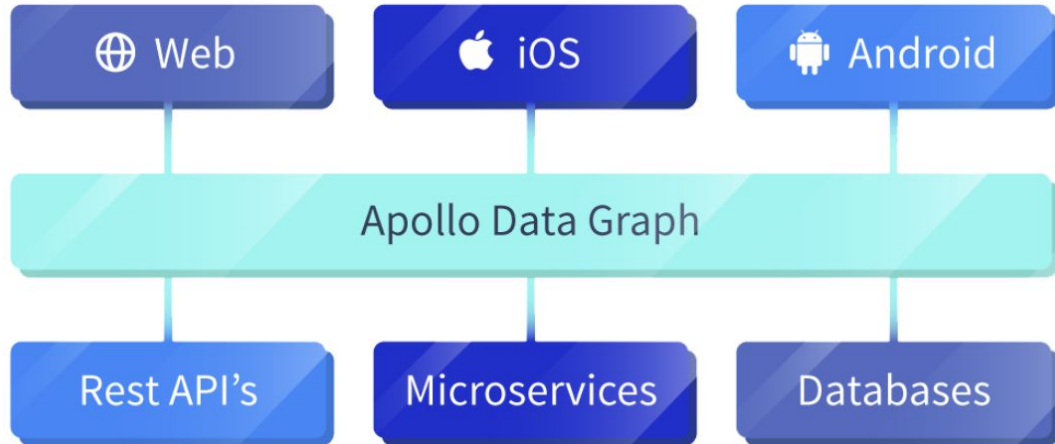
Simplify app development by combining APIs, databases, and microservices into a single data graph that you can query with GraphQL

Start the Tutorial

Talk to an Expert



apollographql.com



```
query Query($productsFirst: Int) { ...  
  products(first: $productsFirst) {  
    edges {  
      product {  
        upc  
        name  
        price  
      }  
    }  
  }  
}
```

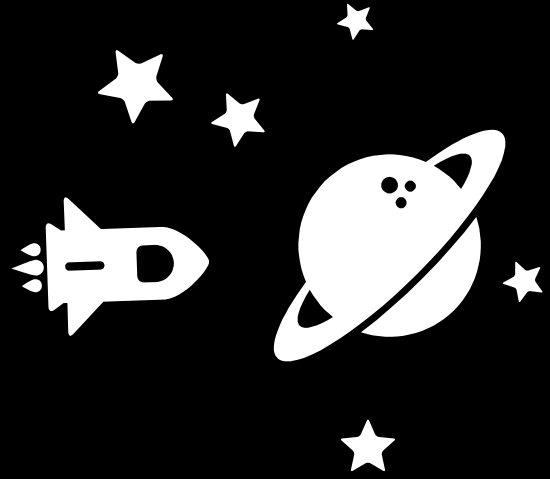
```
{  
  "data": {  
    "products": {  
      "edges": [  
        {  
          "product": {  
            "upc": "1",  
            "name": "Table",  
            "price": 899  
          }  
        },  
        {  
          "product": {  
            "upc": "2",  
            "name": "Couch",  
            "price": 1299  
          }  
        }  
      ],  
    }  
  }  
}
```



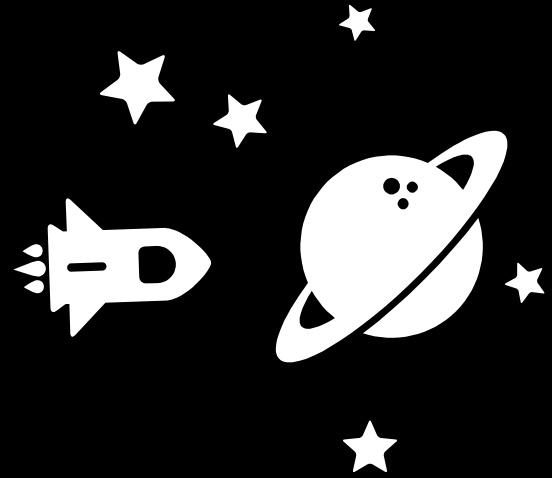
<http://npmjs.com/apollo>

Open Source Projects

(free code, fame, and passive aggressive
comments on the internet)



Projects



Who

Why

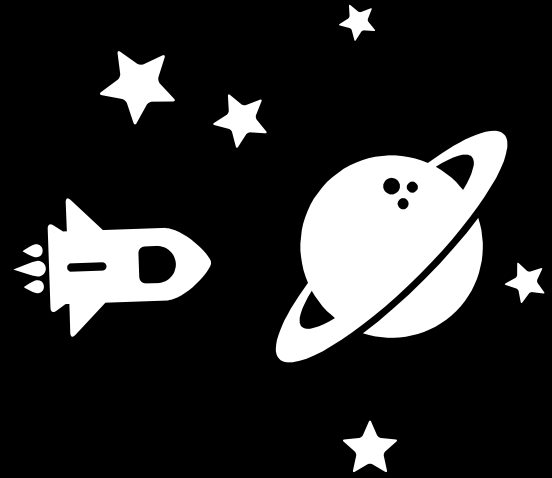
What

Where

How

Who are the stakeholders

(and why would people even think of
working for “free”)



Maintainers

Small group of people with responsibility for the direction and quality of the project

Contributors

Solving problems & committing code, empowered by maintainers

Users

Everyone has incentives

Maintainers

- Solve a problem
- Help people
- Gain notoriety
- Support an ecosystem

Contributors

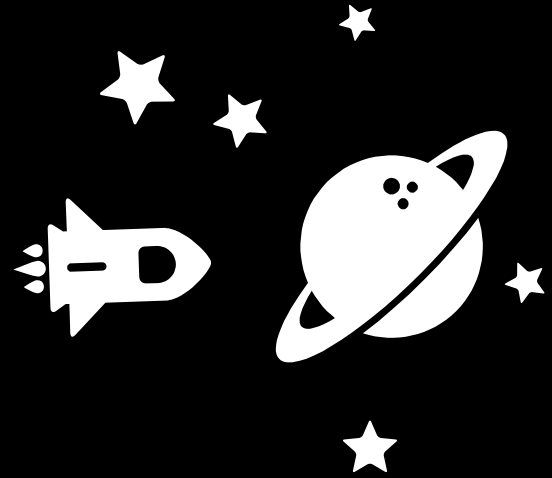
- Learn a technology
- Help advance a project
- Gain notoriety

(active) Users

- Learn a technology
- Stay up-to-date
- Gain credibility (jobs)

What's everyone up to?

(assuming they're not curled up into a ball awaiting the end of the pandemic when they can see their friends and family again... hypothetically of course)



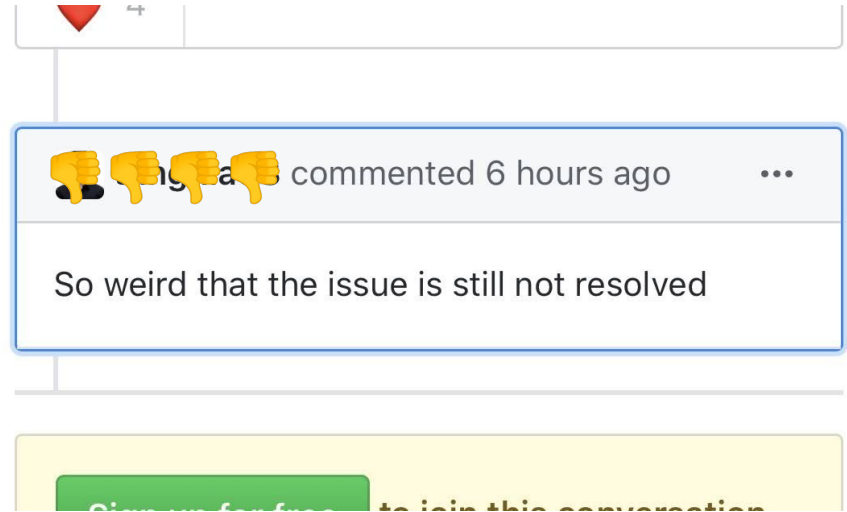
Maintainers

Expectation



Maintainers

Reality



Maintainers

1.

- Q&A
- Coordinating work
- Providing a community space
- Education
- Evangelism
- Write code lol

**Everything takes more and
more time as the project
grows in popularity**

Contributors

2.

- Learn the project
- Solve actionable issues
- Move stalled issues along
- Lift workload from maintainers

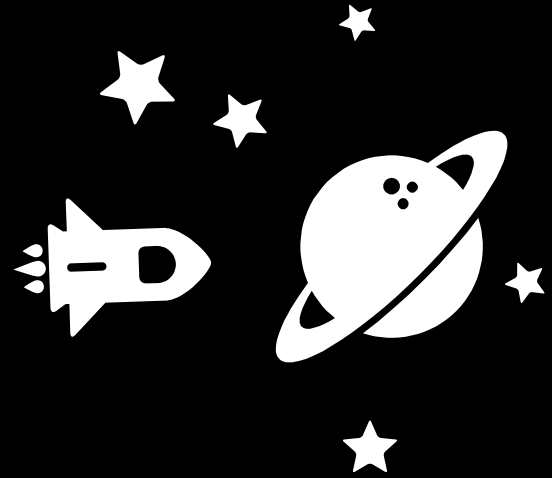
(active) Users

- Education
- Spreading the word
- Answering questions
- Reporting bugs

3.

Where does this all happen?

(the question of the decade)





GitHub



Discourse

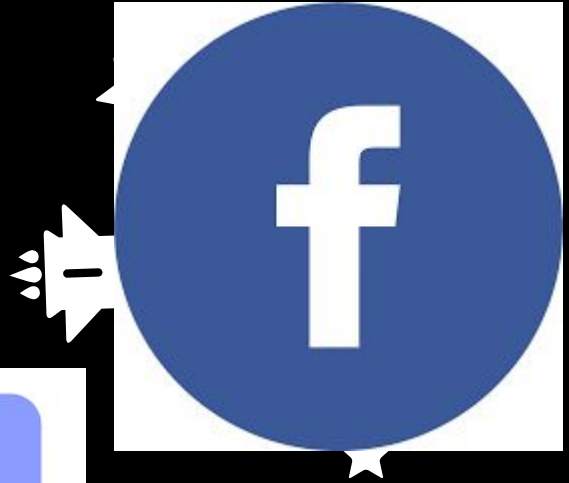
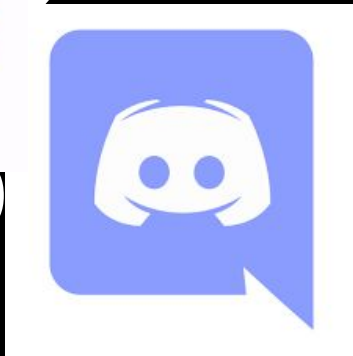


GitLab

ere d
hap



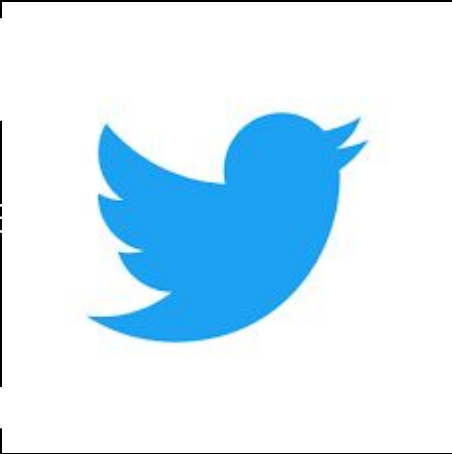
slack



spectrum

t

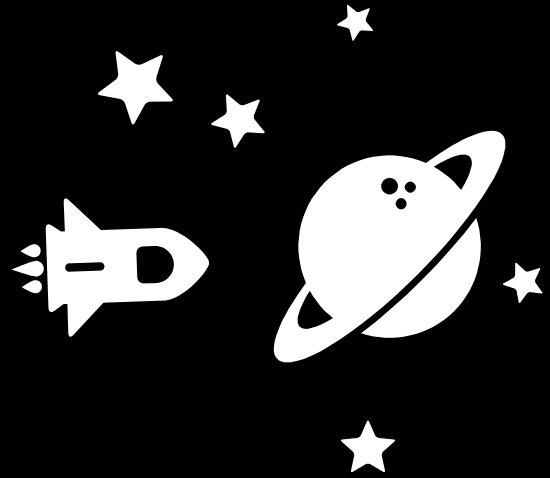
(the



decade)

How are the bills getting paid?

(Wait, I thought this was free?)



**How are these
projects run**

BDFL

Benevolent dictator for life

Corporate Backing

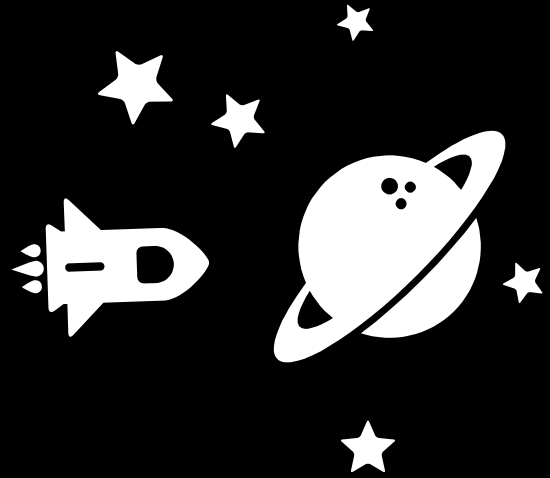
A company builds OSS to back other streams of revenue or hiring

Foundations/TCs

Committees (often with paid membership) formally propose and vote on RFCs

How to be a good citizen?

(empathy, empathy, empathy)



A landscape photograph featuring a double rainbow arching over rolling green hills. In the foreground, a person wearing a red jacket and black pants stands with their arms raised in a celebratory gesture. The sky is a mix of blue and grey, suggesting a recent rain. The overall scene is vibrant and scenic.

**Don't be an
asshole**

Be kind

1.

- Approach the project with kindness
- Assume the best of people
- Respect the time of others
- Understand decisions aren't black/white

Learn the project

2.


- Read any readmes or contribution guidelines
- Does the project even accept outside contributions?
 - If so, what kind & scope?
- Take a look at the Issue/PR templates
- How are issues managed?

Contributing to Rover

Rover is not yet ready for external feature contributors, but [documentation contributions](#) are welcome. At this time, this article is intended primarily for internal contributors and partners.

Open Thorough Issues

3.

- **Search other issues first**
- Write reasonable titles
- Use kind language explaining the issue *and* its importance to you
- Supply workarounds to issues
- Reference other relevant issues
- As much detail about your environment as possible
- Reproduction cases (in a linked repo )

Variables don't seem to be validated #357



JakeDawkins opened this issue 6 days ago · 3 comments



JakeDawkins commented 6 days ago



We noticed recently that when we built a query and accidentally added a query variable, and then passed that variable as an argument to a field in an operation that was invalid, we didn't receive any kind of build errors, which made me wonder if arguments and variables are being properly validated, and I don't think they are.

I intend on looking into this deeper, but as a first pass, to reproduce what I'm talking about, you can add any argument you want in the `keywords_query` test [here](#), and the test will not fail.

The example I used to reproduce this was

```
query searchQuery($criteria: extern!, $invalid_var: Float!) {
  search {
    transactions(criteria: $searchID, invalid_var: $invalid_var) {
      for
      status
    }
  }
}
```

Where `invalid_var` doesn't exist in the schema.

In the meantime, before I have the time to look into this, does anyone know if (and if so, where) these variables are being validated? Or are they not currently part of validation?



JakeDawkins mentioned this issue 6 days ago

investigate why queries that should be invalid can be compiled apollographql/rover#227

🔒 Open



tomhoule commented 6 days ago

Member



There is very little query validation built into the crate — it is something that can be implemented, but it's work that nobody has done yet.



JakeDawkins commented 3 days ago

Author



@tomhoule I think we'd like to take a look at adding this! Would you be able to help review a PR? 😊



tomhoule commented 3 days ago

Member



As you probably noticed I sadly don't have free time to work on graphql-client at the moment, but I can definitely make time to review a PR with tests. So yes, that would be awesome :)



1



1

Open helpful PRs

4.

- **Seek permission first if it's a large change**
- Reasonable titles
- Explain what you fixed in the body, not necessarily how you fixed it
- Reference relevant issues
 - (use the “*Fixes #30*” syntax)
- **Write tests (and run them locally)**
- **Write documentation**
- Be patient



Sindre Sorhus ✓ @sindresorhus · May 21, 2019

Some observations from having merged thousands of pull requests in the past few years:

- Almost no one writes a good pull request title
- More than half don't know about the `Fixes #112` syntax
- ~30% don't run tests locally before submitting a PR
- ~40% don't include docs/tests

33

247

1K



feakury @feakuru · May 21, 2019

how would you determine the rules for writing a good PR title? i'm really interested cause my titles always seem not good enough to me 😞

1



1



Sindre Sorhus ✓
@sindresorhus

Replying to @feakuru

I think this is a good rule to remember:

[twitter.com/mikesharov/sta...](https://twitter.com/mikesharov/status/1130000000000000000)

Some more:

- Describe the problem the PR solves, not the change itself
- Use imperative present tense
- No typos
- Look at existing PRs/commits in the repo and follow their formatting

Share your knowledge

5.

- Be an active user
- If you have thoughtful responses to questions/comments, give them!
- Go where the conversation is
- Don't offer feedback where it's not asked for

Know sometimes things don't go your way

6.

- Software projects are messy and frustrating
 - Conway's law
- **Maintainers have a lot more context for many problems and the future of the project**
- Don't annoy (or back-channel) contributors or maintainers
- Don't be afraid to fork

Thanks!



Any questions?

You can find me at

- @jakedawkins
- jake@apollographql.com