

> sessions/workshops/



Building automations

with GitHub Apps and GraphQL



Andrew Hoglund

**Senior Engineer
Ecosystem API**



Mark Tareshawty

**Senior Engineer
Ecosystem Apps**



Jovel Crisostomo

@jovel



Juan Pemberthy

@jpemberthy



Marc-Andre Giroux

@xuorig

Today we will...

Learn the fundamentals of GitHub Apps and GraphQL

Create a GitHub App, and set it up for local development.

Install our GitHub App onto repositories.

Receive events from GitHub to our local application.

Use GraphQL to respond to those events in an authenticated manner.

Learn the fundamentals of GitHub Apps and GraphQL

Create a GitHub App, and set it up for local development.

Install our GitHub App onto repositories.

Receive events from GitHub to our local application.

Use GraphQL to respond to those events in an
authenticated manner.

Learn the fundamentals of GitHub Apps and GraphQL

Create a GitHub App, and set it up for local development.

Install our GitHub App onto repositories.

Receive events from GitHub to our local application.

Use GraphQL to respond to those events in an authenticated manner.

Learn the fundamentals of GitHub Apps and GraphQL

Create a GitHub App, and set it up for local development.

Install our GitHub App onto repositories.

Receive events from GitHub to our local application.

Use GraphQL to respond to those events in an
authenticated manner.

Learn the fundamentals of GitHub Apps and GraphQL

Create a GitHub App, and set it up for local development.

Install our GitHub App onto repositories.

Receive events from GitHub to our local application.

Use GraphQL to respond to those events in an
authenticated manner.



Demo

GitHub Apps

Granular permissions

Singular WebHook for all events

Grantable on individual resources

Better insight into user identity

Tokens expire for additional security

More comprehensive rate limiting

Granular permissions

Singular WebHook for all events

Grantable on individual resources

Better insight into user identity

Tokens expire for additional security

More comprehensive rate limiting

Granular permissions

Singular WebHook for all events

Grantable on individual resources

Better insight into user identity

Tokens expire for additional security

More comprehensive rate limiting

Granular permissions

Singular WebHook for all events

Grantable on individual resources

Better insight into user identity

Tokens expire for additional security

More comprehensive rate limiting

Granular permissions

Singular WebHook for all events

Grantable on individual resources

Better insight into user identity

Tokens expire for additional security

More comprehensive rate limiting

Granular permissions

Singular WebHook for all events

Grantable on individual resources

Better insight into user identity

Tokens expire for additional security

More comprehensive rate limiting

Two ways to
authenticate

Server to server (Bots)

User to server (via a User)

Server to server

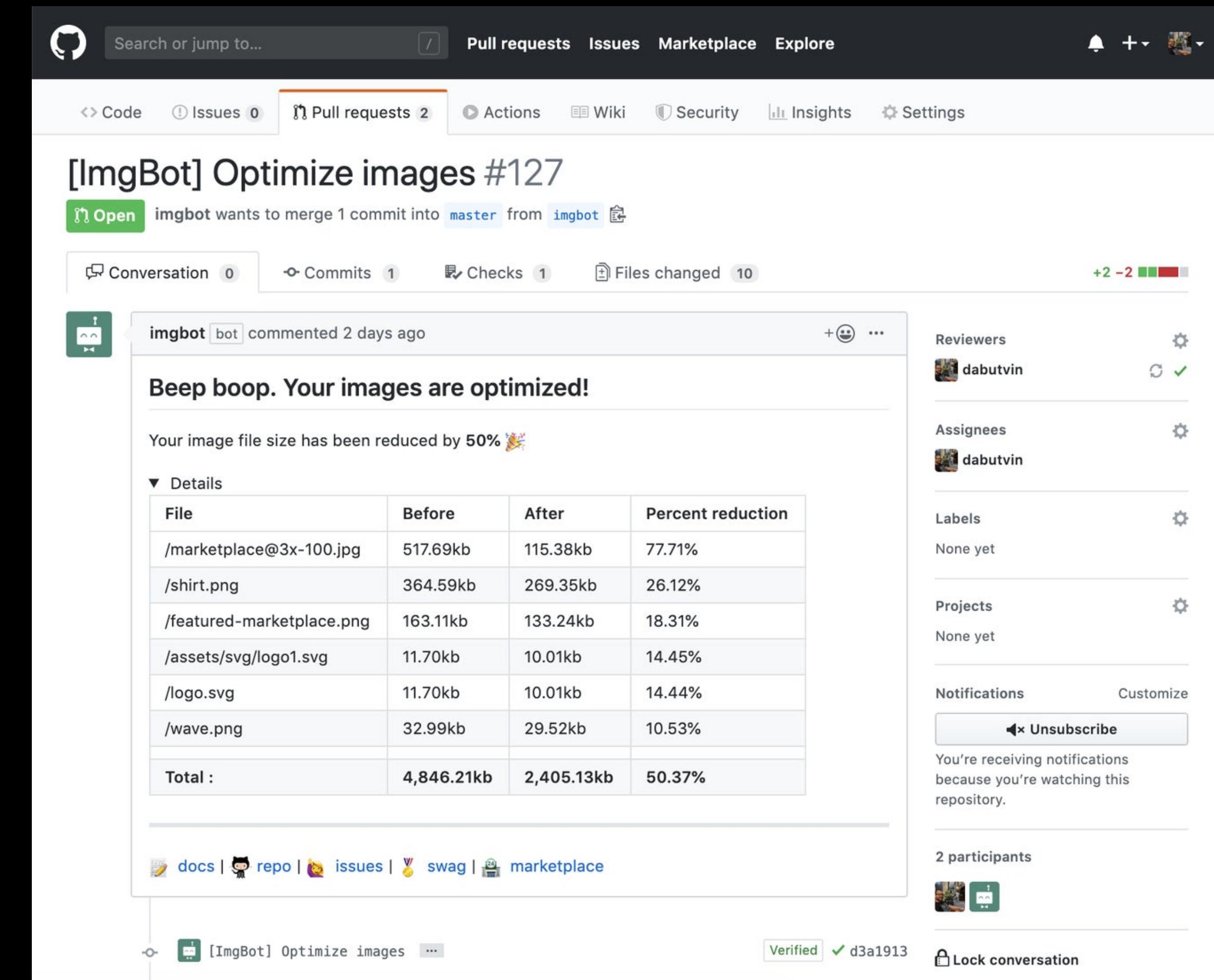
- First-class actors within GitHub.



<https://github.com/apps/stale>

Server to server

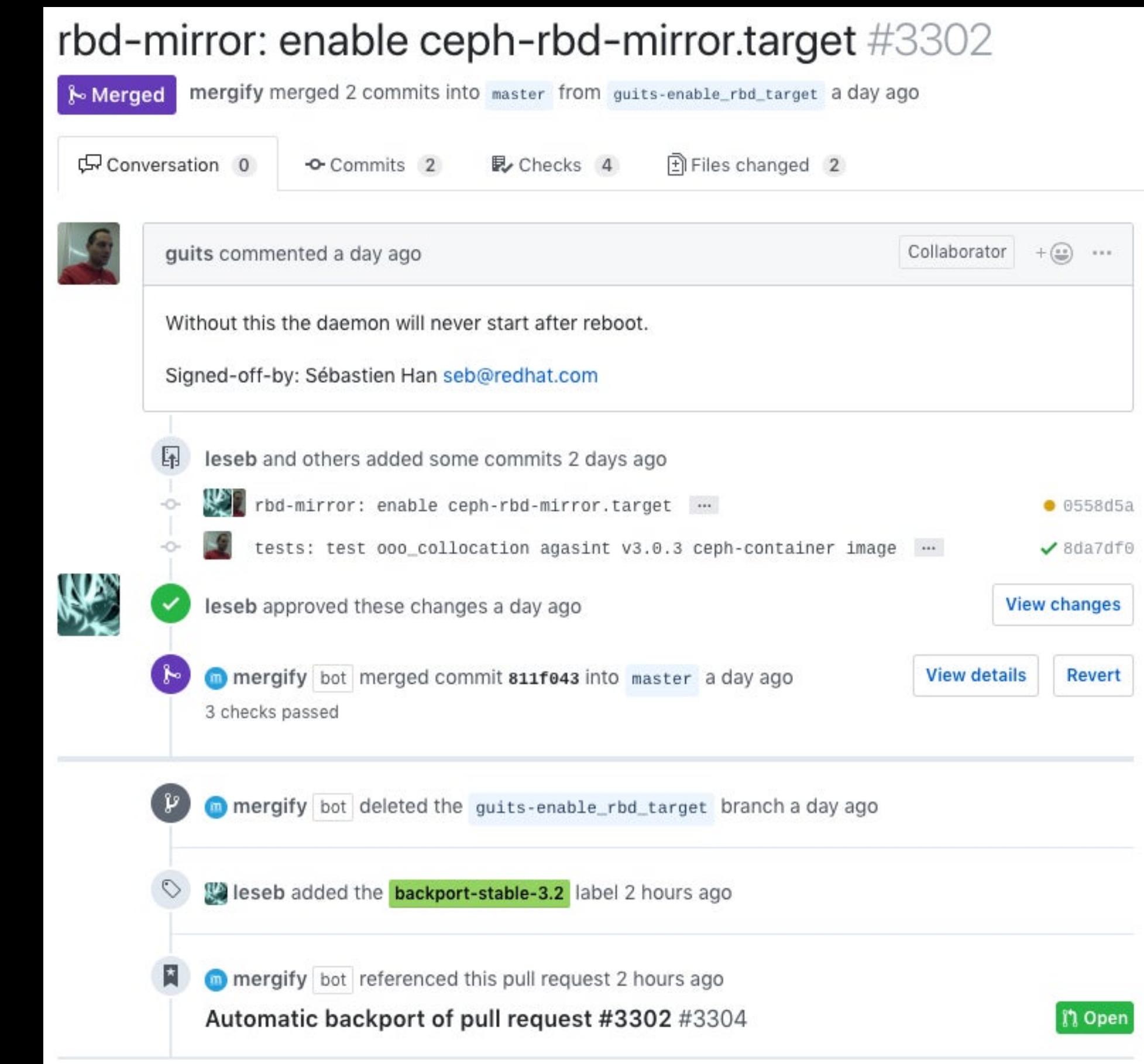
- First-class actors within GitHub.
- Can act on its own behalf, taking actions via the API directly using its own identity.



<https://github.com/marketplace/imgbot>

Server to server

- First-class actors within GitHub.
- Can act on its own behalf, taking actions via the API directly using its own identity.
- Which means you don't need to maintain a bot or service account as a separate user.



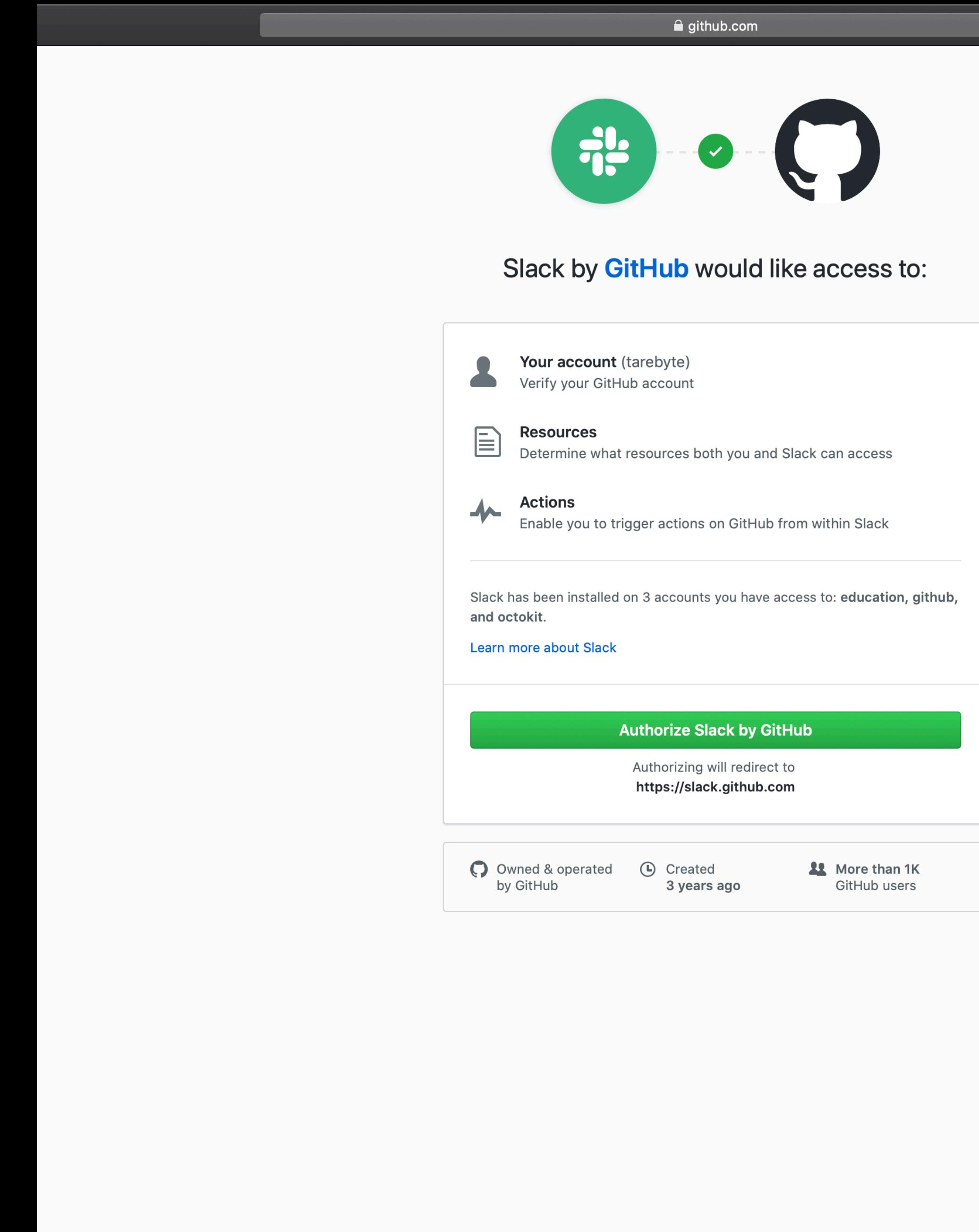
<https://github.com/marketplace/mergify>

Server to server (Bots)

User to server (via a User)

User to server

- Uses the user's identity instead of the bot



User to server

- Uses the user's identity instead of the bot
- Allows you to represent an action the app took as the user

ArgumentError: string contains null byte #3

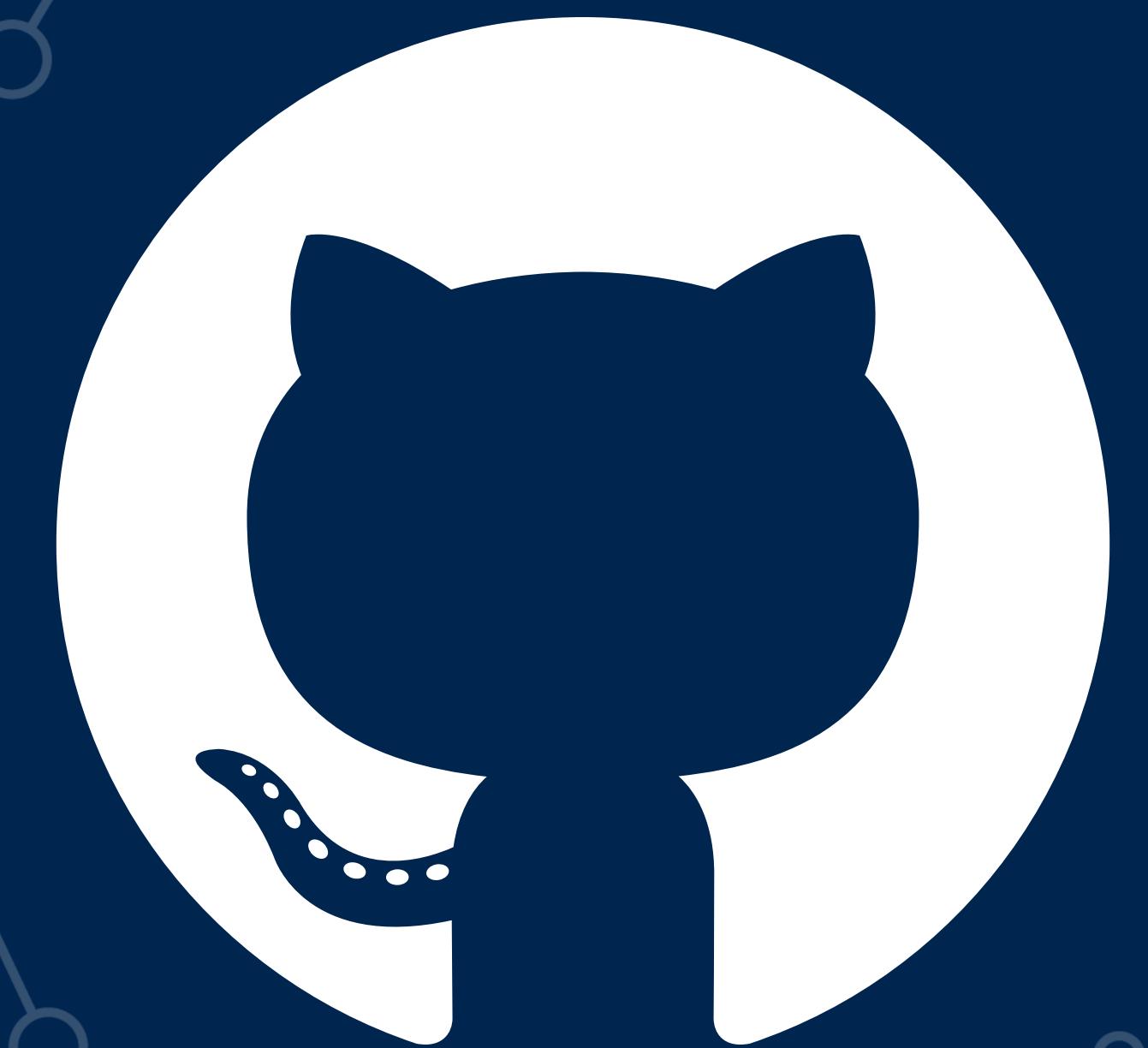
 Open davidcelis opened this issue 19 hours ago — with Sentry · 1 comment

 Conversation 1  Commits 1  Files changed 1

davidcelis commented 19 hours ago Member

<https://sentry.io/goodbrews/goodbrews/issues/42>

```
ArgumentError (string contains null byte):  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/multipart/parser.rb:22:  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/multipart/parser.rb:22:  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/multipart/parser.rb:135:  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/multipart/parser.rb:59:  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/multipart/parser.rb:56:  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/multipart/parser.rb:56:  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/multipart/parser.rb:25:in `par  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/request.rb:375:in `pars  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/request.rb:207:in `POST  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/methodoverride.rb:39:in  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/methodoverride.rb:27:in  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/methodoverride.rb:15:in  
  vendor/gems/2.4.0/ruby/2.4.0/gems/rack-1.6.5/lib/rack/runtime.rb:18:in `call'  
  lib/goodbrews/default_middleware.rb:16:in `call'  
  vendor/gems/2.4.0/ruby/2.4.0/gems/railties-3.2.22.5.github.0a82cf1/lib/rails/  
  vendor/gems/2.4.0/ruby/2.4.0/gems/railties-3.2.22.5.github.0a82cf1/lib/rails/
```

GraphQL



GraphQL @ GitHub

- Introduced in 2016

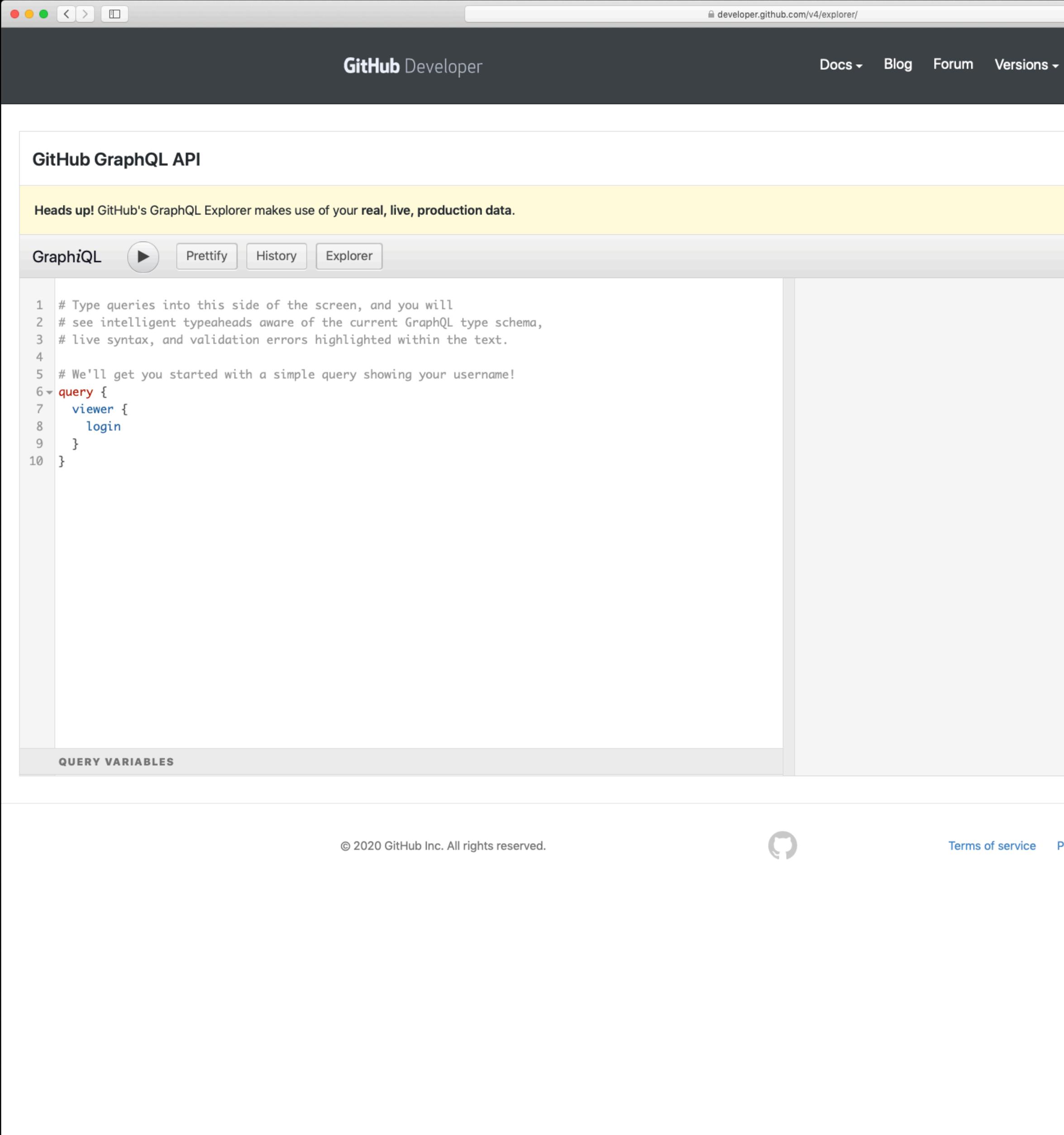
The screenshot shows the GitHub GraphQL API Explorer. At the top, there's a navigation bar with links for 'Docs', 'Blog', 'Forum', and 'Versions'. Below the navigation, the title 'GitHub GraphQL API' is displayed, followed by a note: 'Heads up! GitHub's GraphQL Explorer makes use of your real, live, production data.' The main area is titled 'GraphiQL' and includes buttons for 'Prettify', 'History', and 'Explorer'. A code editor window contains a GraphQL query:

```
1 # Type queries into this side of the screen, and you will
2 # see intelligent typeheads aware of the current GraphQL type schema,
3 # live syntax, and validation errors highlighted within the text.
4
5 # We'll get you started with a simple query showing your username!
6query {
7   viewer {
8     login
9   }
10}
```

Below the code editor is a 'QUERY VARIABLES' section. At the bottom of the page, there's a copyright notice: '© 2020 GitHub Inc. All rights reserved.' and the GitHub logo.

GraphQL @ GitHub

- Introduced in 2016
- One of the first large public GraphQL API's



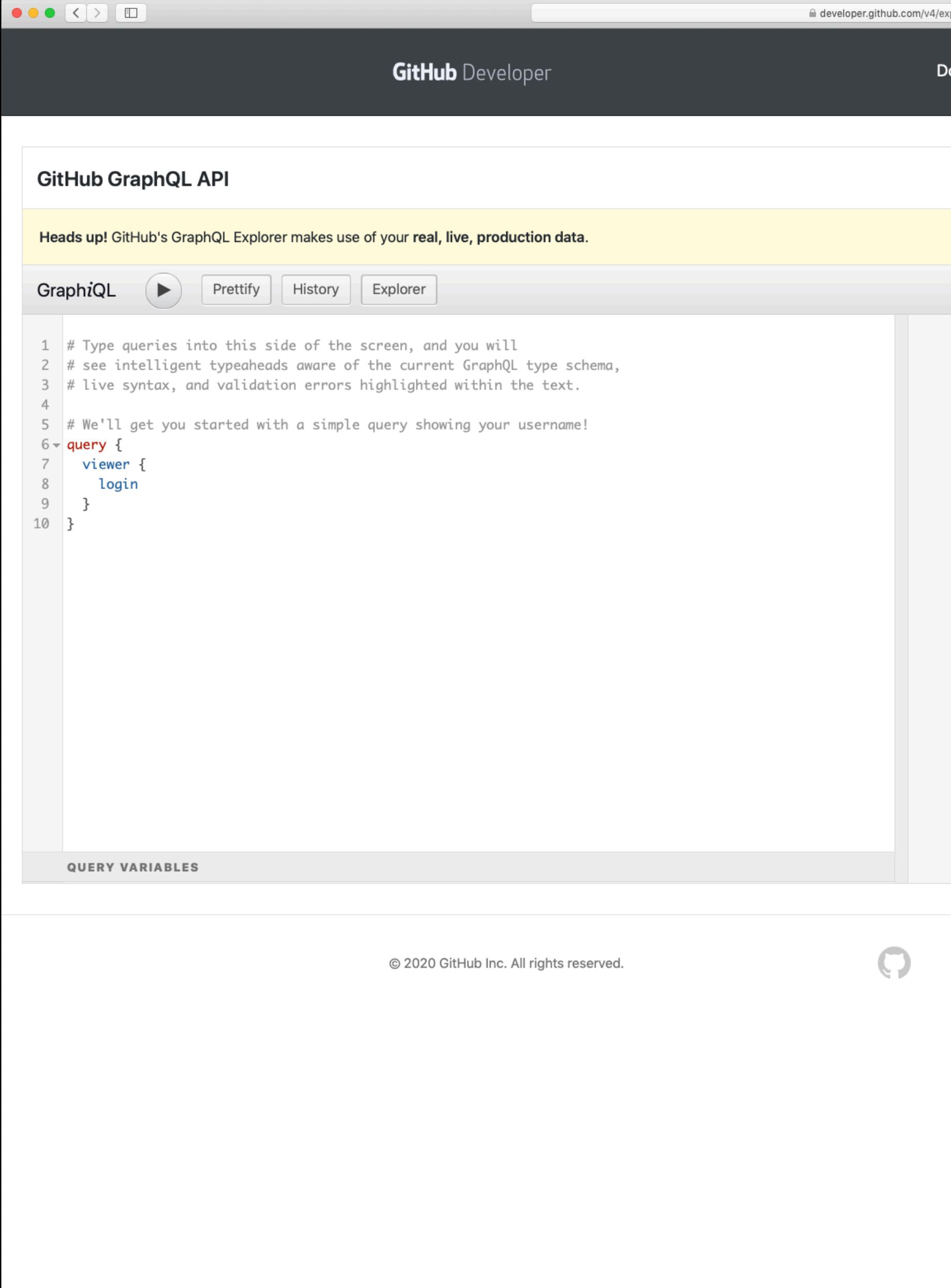
The screenshot shows the GitHub GraphQL Explorer interface. At the top, there's a navigation bar with links for 'Docs', 'Blog', 'Forum', and 'Versions'. Below the navigation is a header for 'GitHub GraphQL API' with a note: 'Heads up! GitHub's GraphQL Explorer makes use of your real, live, production data.' The main area contains a code editor with a GraphQL query:

```
1 # Type queries into this side of the screen, and you will
2 # see intelligent typeheads aware of the current GraphQL type schema,
3 # live syntax, and validation errors highlighted within the text.
4
5 # We'll get you started with a simple query showing your username!
6 query {
7   viewer {
8     login
9   }
10 }
```

Below the code editor is a section labeled 'QUERY VARIABLES' which is currently empty. At the bottom of the page, there's a copyright notice: '© 2020 GitHub Inc. All rights reserved.' and the GitHub logo.

GraphQL @ GitHub

- Introduced in 2016
- One of the first large public GraphQL API's
- New approach to exposing GitHub functionality to users.



The screenshot shows the GitHub GraphQL API Explorer interface. At the top, there's a navigation bar with links for 'Docs', 'Blog', 'Forum', and 'Versions'. Below the navigation is a header for 'GitHub Developer' and the title 'GitHub GraphQL API'. A yellow banner below the title says 'Heads up! GitHub's GraphQL Explorer makes use of your real, live, production data.' The main area contains a code editor with a GraphQL query. The code is numbered from 1 to 10 and includes comments and a simple query for the viewer's login information. Below the code editor is a section labeled 'QUERY VARIABLES' which is currently empty. At the bottom right of the interface, there's a GitHub logo and links for 'Terms of service' and 'API documentation'.

```
1 # Type queries into this side of the screen, and you will
2 # see intelligent typeheads aware of the current GraphQL type schema,
3 # live syntax, and validation errors highlighted within the text.
4
5 # We'll get you started with a simple query showing your username!
6query {
7   viewer {
8     login
9   }
10}
```

© 2020 GitHub Inc. All rights reserved.

Why GraphQL?

- Type safety

```
query {  
  viewer {  
    login  
  }  
}
```

Why GraphQL?

- Type safety
- Custom response data

```
query {  
  organization(login: "npm") {  
    repositories(first: 10) {  
      edges {  
        node {  
          nameWithOwner  
        }  
      }  
    }  
  }  
}
```

Why GraphQL?

- Type safety
- Custom response data
- Reduced network traffic

```
query {  
  repository(name: "linux") {  
    forks(last: 100) {  
      edges {  
        node {  
          nameWithOwner  
        }  
      }  
    }  
  }  
}
```


Thank you!

