

# GraphQL API

**GitHub**



@wilhelmklopp

1 Answer

active

oldest

votes



You should be able to do this with the [GraphQL API](#)

1

In fact, the point of the GraphQL API is to only specify exactly what you want.



So you can have a query like this:

```
{
  repository(owner: ":owner", name: ":repo") {
    id
    issues(first: 100) {
      edges {
        node {
          id
          number
          title
          url
        }
      }
    }
  }
}
```

That will return the first 100 issues for a specific repository, and only include the `id`, `number`, `title`, and `url` for each issue.

Useful links:

Repository documentation: <https://developer.github.com/early-access/graphql/object/repository/>

Issue documentation: <https://developer.github.com/early-access/graphql/object/issue/>

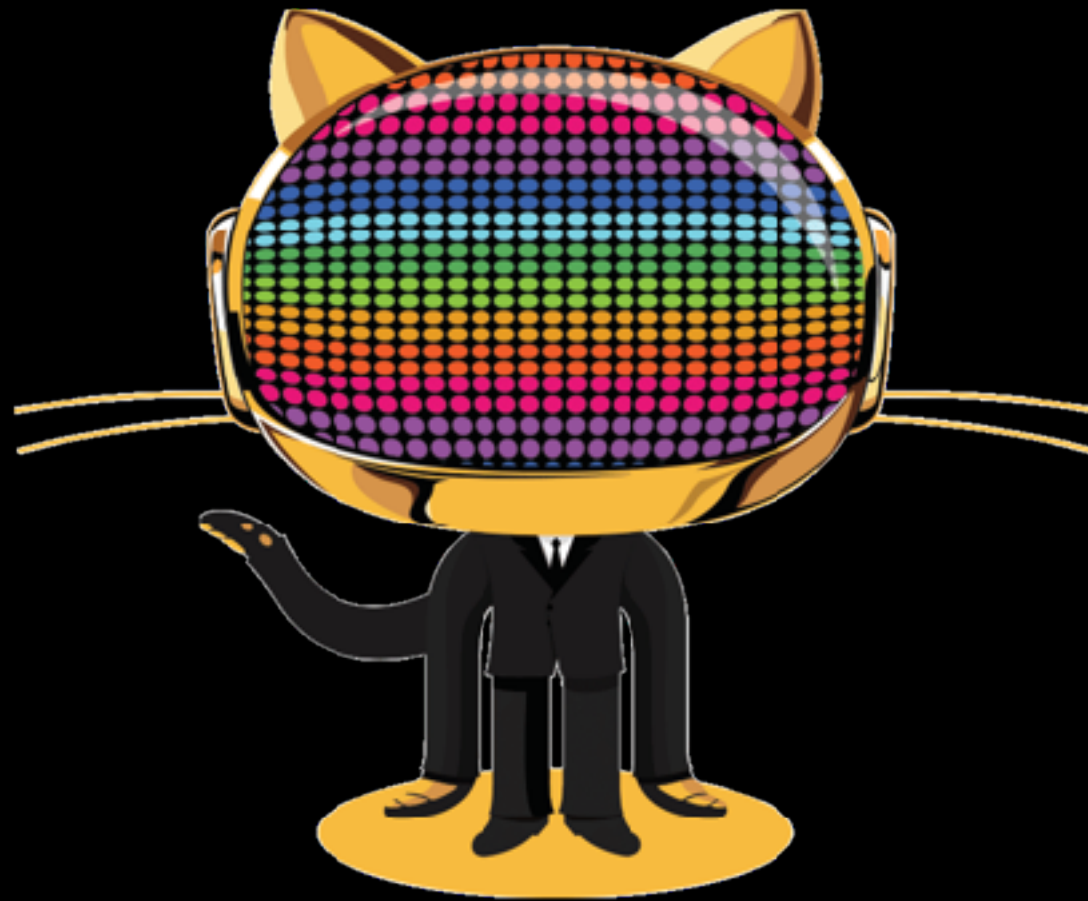
[share](#) [edit](#) [delete](#) [flag](#)

answered Mar 16 at 11:57



[Wilhelm Klopp](#)

1,227 ● 10 ● 22



APIs

# REST

[https://api.themoviedb.org/3/discover/movie?  
api\\_key=deae2f3e8a5928ef1db93f2fb32b7d22](https://api.themoviedb.org/3/discover/movie?api_key=deae2f3e8a5928ef1db93f2fb32b7d22)

```
{
  "page": 1,
  "results": [
    {
      "poster_path": "/tnmL0g604PDRJwGJ5fsUSYKfo9.jpg",
      "adult": false,
      "overview": "A live-action adaptation of Disney's version of the classic 'Beauty and the Beast' tale of a cursed prince and a beautiful young woman who helps him break the spell.",
      "release_date": "2017-03-15",
      "genre_ids": [
        14,
        10402,
        10749
      ],
      "id": 321612,
      "original_title": "Beauty and the Beast",
      "original_language": "en",
      "title": "Beauty and the Beast",
      "backdrop_path": "/6aUWe0GS169wMTSWWexsorMIvwU.jpg",
      "popularity": 191.297003,
      "vote_count": 989,
      "video": false,
      "vote_average": 7.2
    },
  ],
}
```

# GraphQL

## Request

```
{  
  me {  
    name  
  }  
}
```

## Response

```
{  
  "me": {  
    "name": "Luke Skywalker"  
  }  
}
```

# GraphQL

## Request

```
query {  
  user(login: "xoneco") {  
    isCampusExpert  
    isEmployee  
    repositories(last: 3) {  
      totalCount  
      edges {  
        node {  
          name  
        }  
      }  
    }  
  }  
}
```

## Response

```
{  
  "data": {  
    "user": {  
      "isCampusExpert": true,  
      "isEmployee": false,  
      "repositories": {  
        "totalCount": 35,  
        "edges": [  
          {  
            "node": {  
              "name": "inst2005-shop"  
            }  
          },  
          {  
            "node": {  
              "name": "veruto"  
            }  
          },  
          {  
            "node": {  
              "name": "github-graphql-workshop"  
            }  
          }  
        ]  
      }  
    }  
  }  
}
```



# GraphQL



“Power to the clients”

# Features

- Type System
- Introspection
- Efficiency
- Simplicity
- Scalability

# GitHub GraphQL API

- First major public GraphQL API
- Launched in October 2016
- Currently in “Early Access”



Untitled Query 1

Untitled Query 2

Untitled Query 3

Untitled Query 4

GraphQL Endpoint

<https://api.github.com/graphql>

Method

POST

[Edit HTTP Headers](#)

GraphiQL



Prettify

[Docs](#)

```
1 # Welcome to GraphiQL
2 #
3 # GraphiQL is an in-browser IDE for writing, validating, and
4 # testing GraphQL queries.
5 #
6 # Type queries into this side of the screen, and you will
7 # see intelligent typedheads aware of the current GraphQL type schema and
8 # live syntax and validation errors highlighted within the text.
9 #
10 # To bring up the auto-complete at any point, just press Ctrl-Space.
11 #
12 # Press the run button above, or Cmd-Enter to execute the query, and the result
13 # will appear in the pane to the right.
14
15 query {
16   viewer {
17     login
18   }
19 }
20
21 |
```

```
{
  "data": {
    "viewer": {
      "login": "xoneco"
    }
  }
}
```

QUERY VARIABLES

Query

Find full name of any user

Query

List your repositories

# Challenges

Query

List your repositories and  
their languages



Query

Issues on  
facebook/react

Query

Find your own  
databaseID

Query

Find the databaseID of the  
GitHub CEO and check  
whether he is a Campus  
Expert

# Mutations

```
mutation {  
  addReaction(input: {  
    subjectId: "MDEx01B1bGxSZXF1ZXN0OTE3NTkxMTE=",  
    content: THUMBS_UP  
  }) {  
    reaction {  
      content  
      id  
    }  
  }  
}
```

# Mutations Challenge

Mutation

Like the latest issue on  
xoneco's "github-graphql-  
workshop" repository

graphql.org