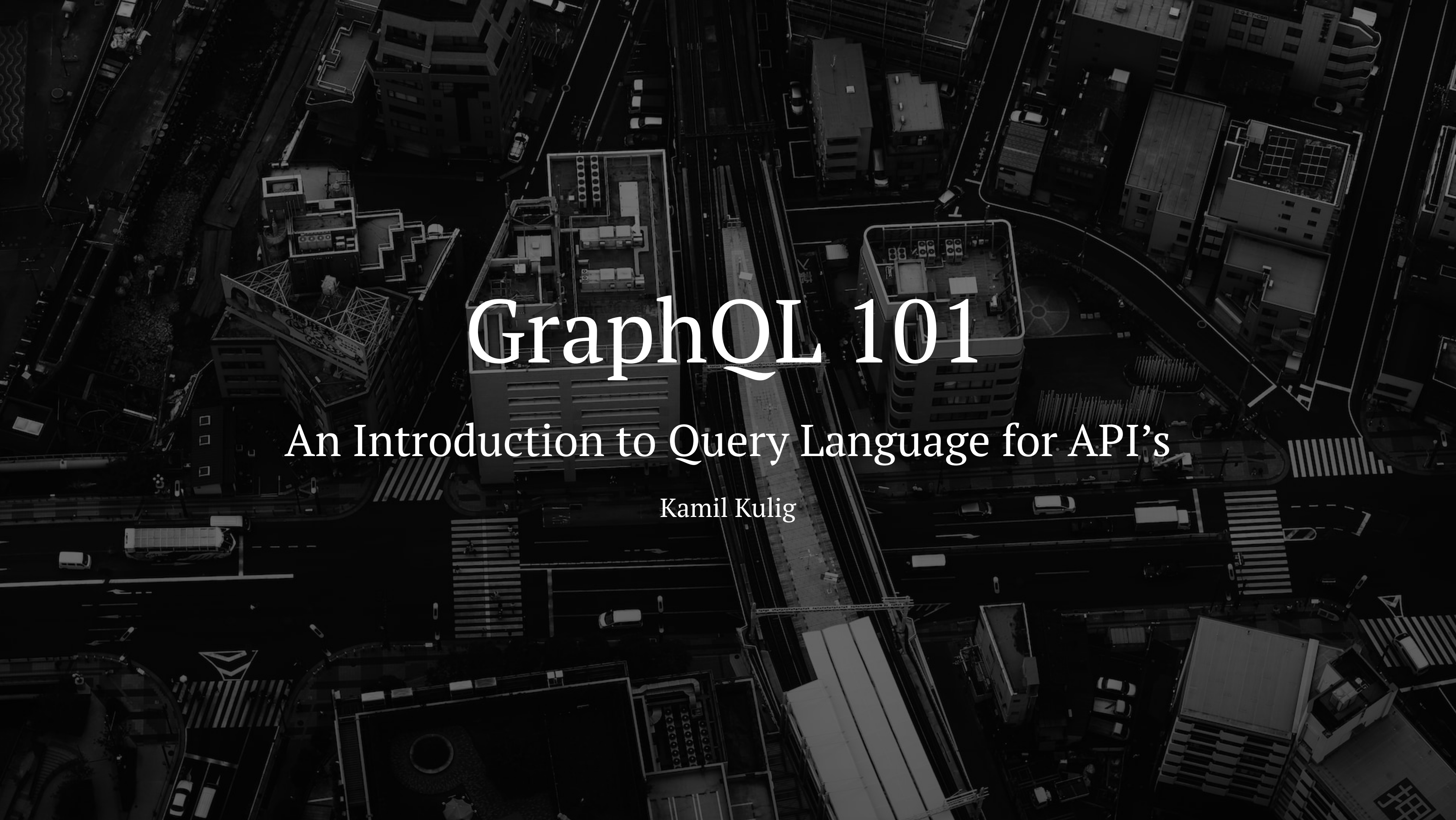


GraphQL 101

An Introduction to Query Language for API’s

Kamil Kulig

agenda

- about me
- motivation
- definition
- compare to other tech
- shines as diamonds
- tools
- challenges

about me

- fullstack developer leader in Schneider Electric 💚
- team leader 🐼 🐯 🐸 🦊 🐰 🐱 🐻
- fullstack developer 👨‍💻
- python & typescript lover 🐍
- eternal optimist 😎
- romantic programmer 🌹
- sport freak ⚽ 🏀 🏈 🥊
- lego fanatic 🧩

motivation

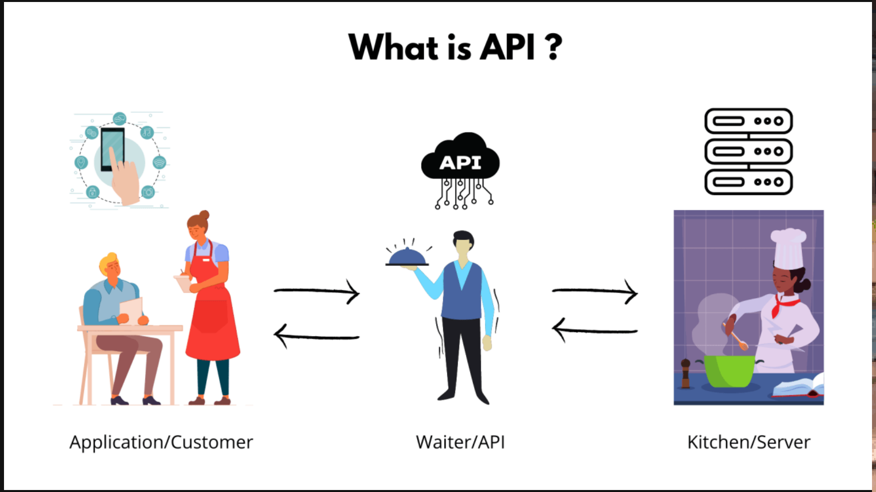
- Maciej Morawski - Speech Recognition with Python
- Sinem Ayyaldaz, ME - (workshop) End-to-end Discord bot development and deployment
- Marcin Brzezinski - (workshop) Introduction to functional programming
- Michal Korzycki - Python the missing bits. Practical application of Metaprogramming

GraphQL

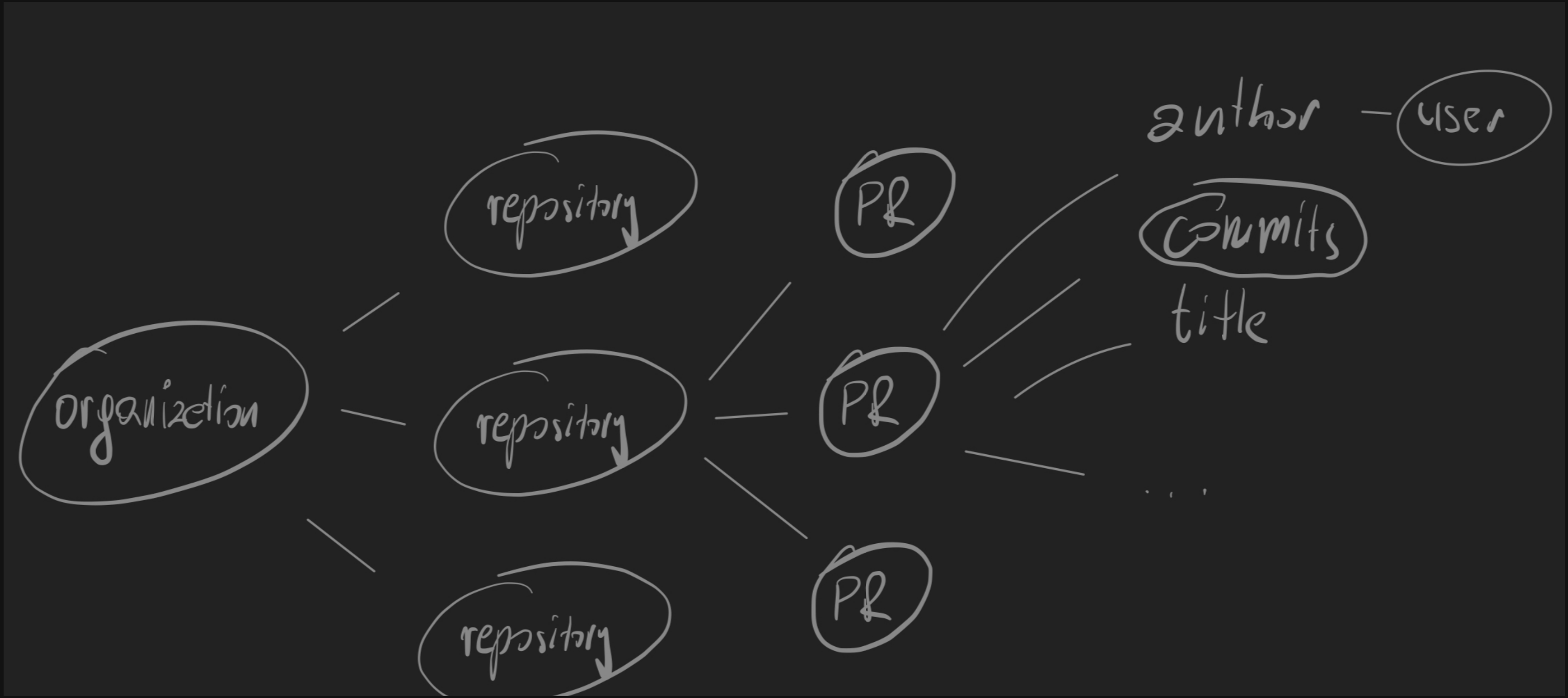
graph query language for APIs

- GQL != GraphQL = Graph Query Language
- created by facebook in 2012
- presented in 2015 at React.js conference
- documentary by HoneyPod about GraphQL https://www.youtube.com/@Honeypotio
- Github started officially use GraphQL in September 14, 2016 https://github.blog/2016-09-14-the-github-graphql-api/

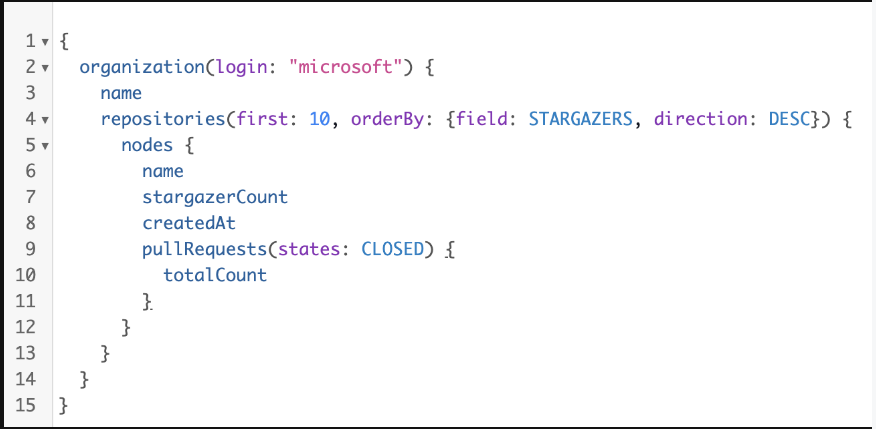
API as restaurant



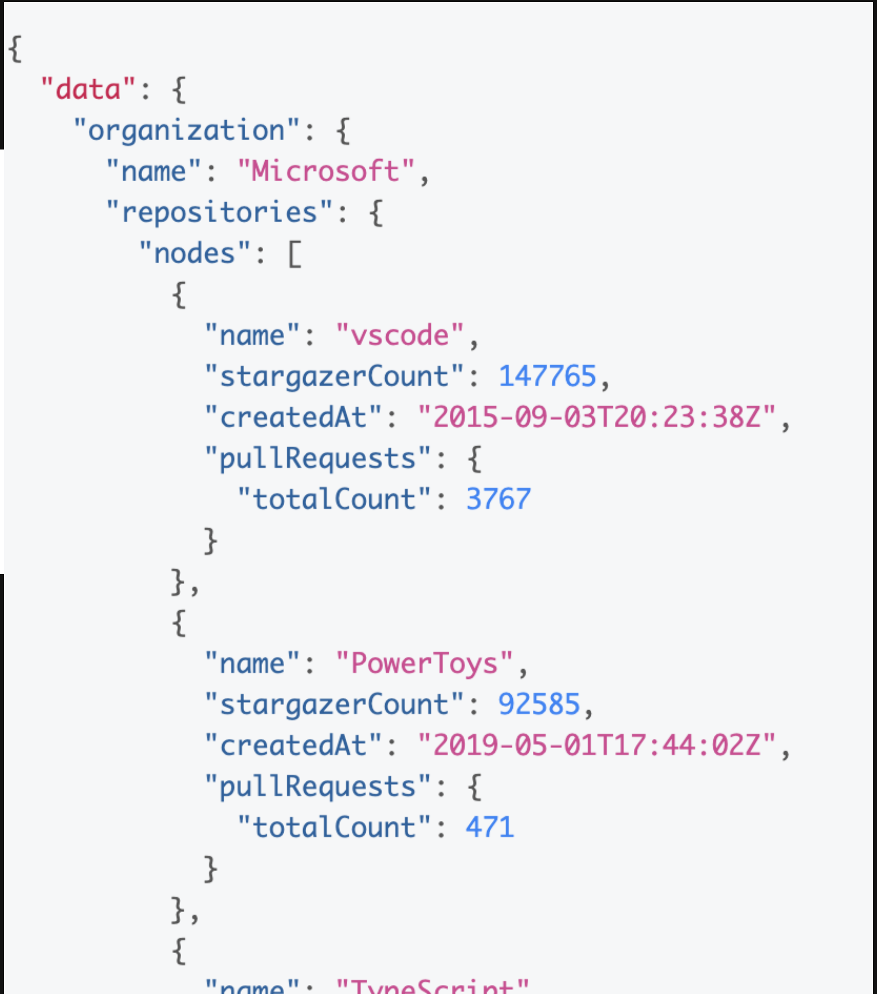
graph


organization
repository
repository
repository
PR
PR
PR
author
user
commits
title
...

query language



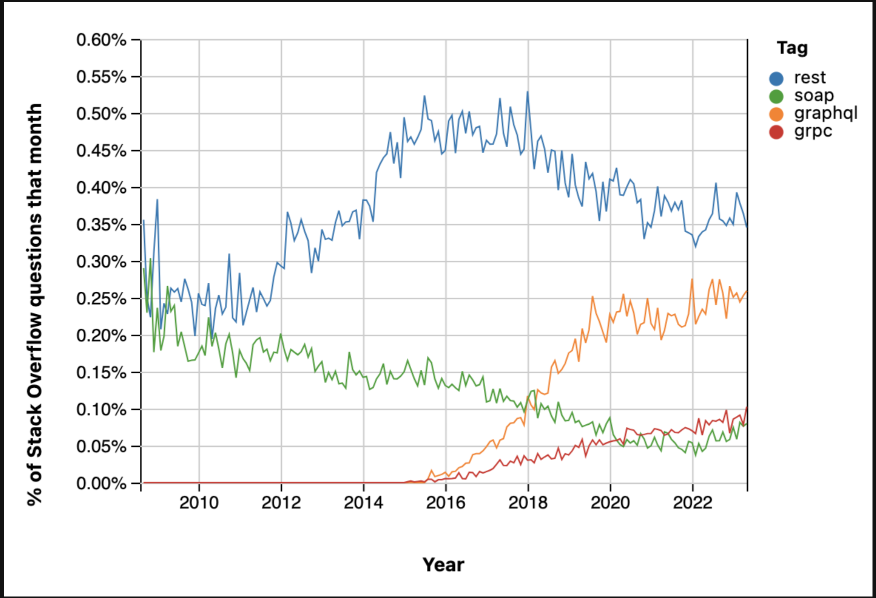
```
{
  organization(login: "microsoft") {
    name
    repositories(first: 10, orderBy: {field: STARGAZERS, direction: DESC}) {
      nodes {
        name
        stargazerCount
        createdAt
        pullRequests(states: CLOSED) {
          totalCount
        }
      }
    }
  }
}
```



```
{
  "data": {
    "organization": {
      "name": "Microsoft",
      "repositories": {
        "nodes": [
          {
            "name": "vscode",
            "stargazerCount": 147765,
            "createdAt": "2015-09-03T20:23:38Z",
            "pullRequests": {
              "totalCount": 3767
            }
          },
          {
            "name": "PowerToys",
            "stargazerCount": 92585,
            "createdAt": "2019-05-01T17:44:02Z",
            "pullRequests": {
              "totalCount": 471
            }
          },
          {
            "name": "TypeScript"
```

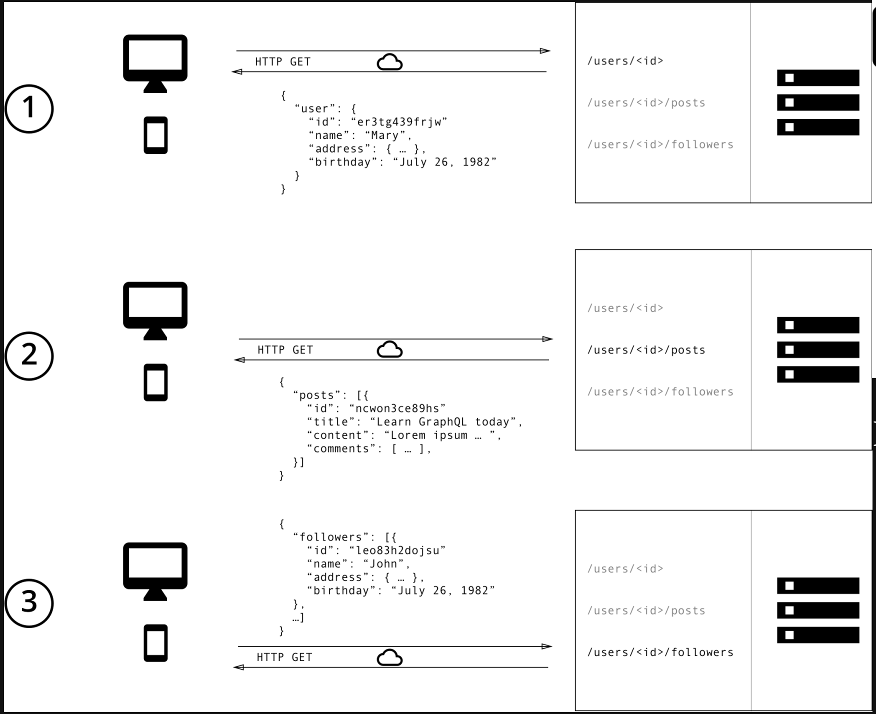
https://graphql.org/learn/

technologies for API’s



Article:
https://wundergraph.com/blog/graphql_rest_openapi_trend_analysis_2023

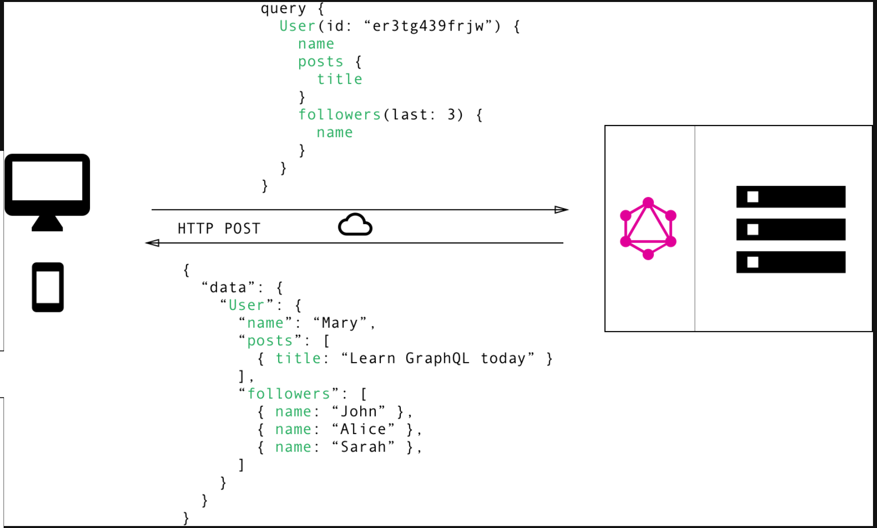
compare with REST



```
{
  “user”: {
    “id”: “er3tg439frjw”
    “name”: “Mary”,
    “address”: { … },
    “birthday”: “July 26, 1982”
  }
}
```

```
{
  “posts”: [{
    “id”: “ncwon3ce89hs”
    “title”: “Learn GraphQL today”,
    “content”: “Lorem ipsum … ”,
    “comments”: [ … ],
  }]
}
```

```
{
  “followers”: [{
    “id”: “leo83h2dojsu”
    “name”: “John”,
    “address”: { … },
    “birthday”: “July 26, 1982”
  },
  …]
}
```



```
query {
  User(id: “er3tg439frjw”) {
    name
    posts {
      title
    }
    followers(last: 3) {
      name
    }
  }
}
```

```
{
  “data”: {
    “User”: {
      “name”: “Mary”,
      “posts”: [
        { title: “Learn GraphQL today” }
      ],
      “followers”: [
        { name: “John” },
        { name: “Alice” },
        { name: “Sarah” },
      ]
    }
  }
}
```

Images from https://www.howtographql.com/

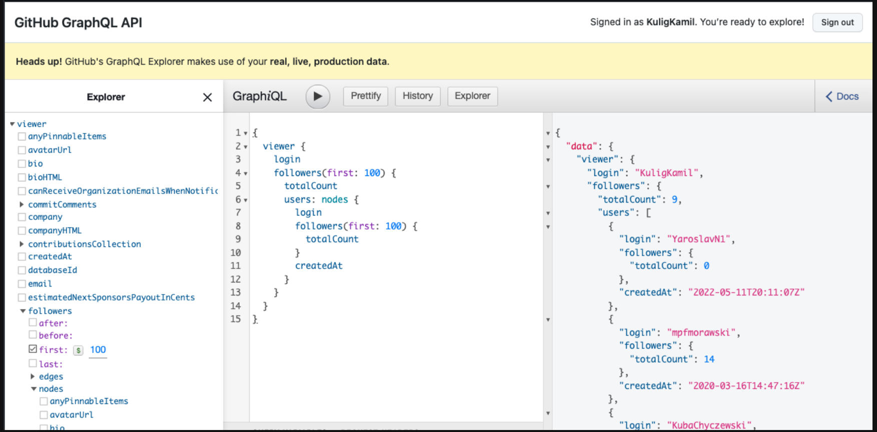
shining 💎

- over-fetching
- under-fetching
- multi clients
- when you have nested data or graph database
- provide performance benefits by reducing the number of network requests

as a client use Github API

https://docs.github.com/en/graphql/overview/explorer

GraphiQL is a graphical interactive in-browser GraphQL IDE.



GitHub GraphQL API

Signed in as **KuligKamil**. You're ready to explore!

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

```
▾ viewer
  ☐ anyPinnableItems
  ☐ avatarUrl
  ☐ bio
  ☐ bioHTML
  ☐ canReceiveOrganizationEmailsWhenNotific
  ▸ commitComments
  ☐ company
  ☐ companyHTML
  ▸ contributionsCollection
  ☐ createdAt
  ☐ databaseId
  ☐ email
  ☐ estimatedNextSponsorsPayoutInCents
  ▾ followers
    ☐ after:
    ☐ before:
    ☑ first: $ 100
    ☐ last:
    ▸ edges
    ▾ nodes
      ☐ anyPinnableItems
      ☐ avatarUrl
```

```
{
  viewer {
    login
    followers(first: 100) {
      totalCount
      users: nodes {
        login
        followers(first: 100) {
          totalCount
        }
        createdAt
      }
    }
  }
}
```

```
{
  "data": {
    "viewer": {
      "login": "KuligKamil",
      "followers": {
        "totalCount": 9,
        "users": [
          {
            "login": "YaroslavN1",
            "followers": {
              "totalCount": 0
            },
            "createdAt": "2022-05-11T20:11:07Z"
          },
          {
            "login": "mpfmorawski",
            "followers": {
              "totalCount": 14
            },
            "createdAt": "2020-03-16T14:47:16Z"
          },
          {
            "login": "KubaChyczewski",
```

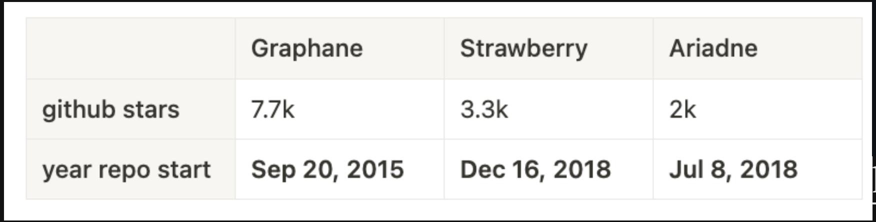
server side

Strawberry vs Graphane vs Ariadne

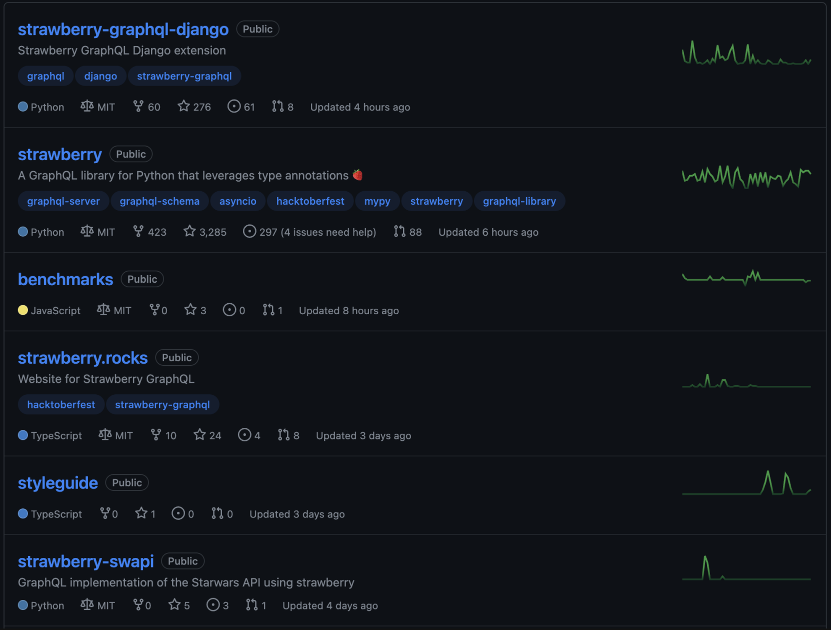
2 approches:

schema-first - describe GraphQL **Schema Definition Language (SDL)** - ariadne

code-first - code describe data: strawberry and graphane



	Graphane	Strawberry	Ariadne
github stars	7.7k	3.3k	2k
year repo start	Sep 20, 2015	Dec 16, 2018	Jul 8, 2018



strawberry-graphql-django Public
Strawberry GraphQL Django extension
graphql django strawberry-graphql
Python MIT 60 276 61 8 Updated 4 hours ago

strawberry Public
A GraphQL library for Python that leverages type annotations 🍓
graphql-server graphql-schema asyncio hacktoberfest mypy strawberry graphql-library
Python MIT 423 3,285 297 (4 issues need help) 88 Updated 6 hours ago

benchmarks Public
JavaScript MIT 0 3 0 1 Updated 8 hours ago

strawberry.rocks Public
Website for Strawberry GraphQL
hacktoberfest strawberry-graphql
TypeScript MIT 10 24 4 8 Updated 3 days ago

styleguide Public
TypeScript 0 1 0 0 Updated 3 days ago

strawberry-swapi Public
GraphQL implementation of the Starwars API using strawberry
Python MIT 0 5 3 1 Updated 4 days ago

https://github.com/strawberry-graphql

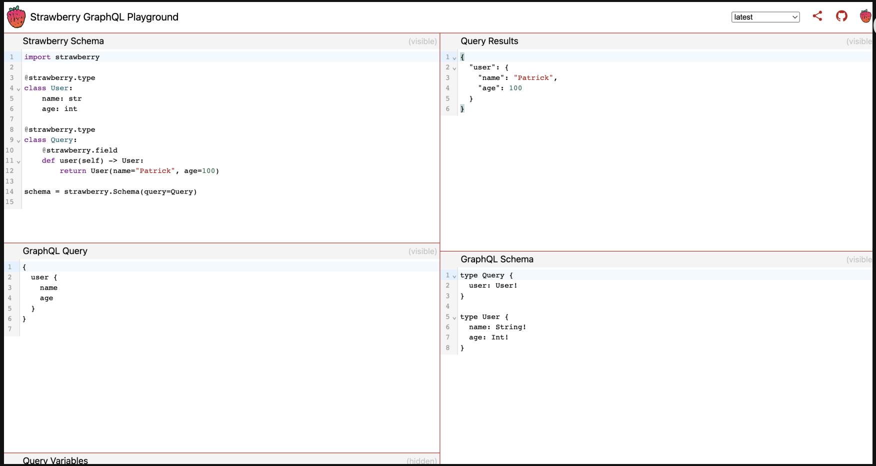
strawberry 🍓

new GraphQL library for Python 3, inspired by dataclasses

examples: https://github.com/strawberry-graphql/examples/

fastapi + sqlalchemy: https://github.com/strawberry-graphql/examples/tree/main/fastapi-sqlalchemy

docs: https://strawberry.rocks/



Strawberry Schema

```python
import strawberry

@strawberry.type
class User:
    name: str
    age: int

@strawberry.type
class Query:
    @strawberry.field
    def user(self) -> User:
        return User(name="Patrick", age=100)

schema = strawberry.Schema(query=Query)
```

Query Results

```json
{
  "user": {
    "name": "Patrick",
    "age": 100
  }
}
```

GraphQL Query

```graphql
{
  user {
    name
    age
  }
}
```

GraphQL Schema

```graphql
type Query {
  user: User!
}

type User {
  name: String!
  age: Int!
}
```

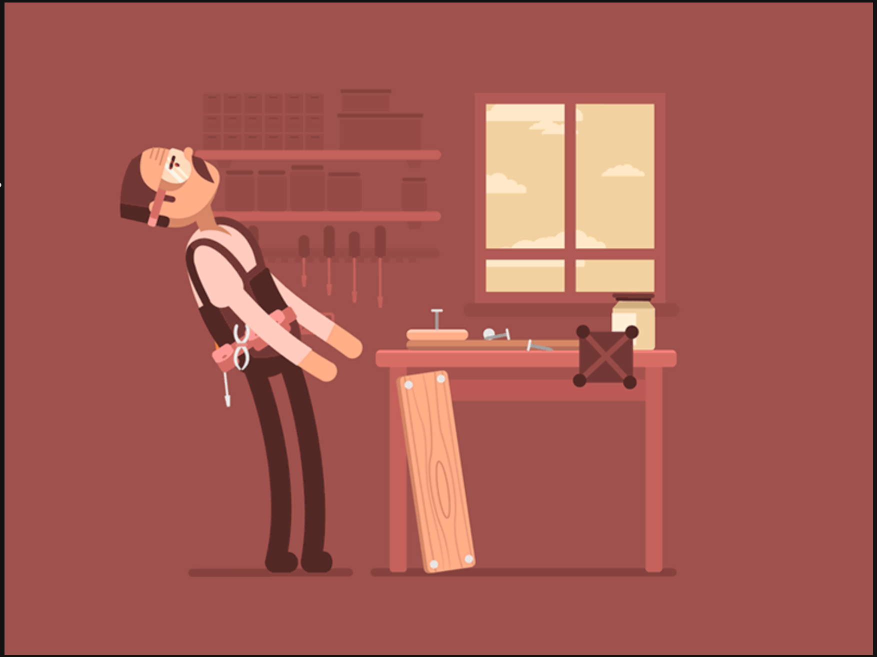
Query Variables

playground: https://play.strawberry.rocks/

hammer time

When you have hammer, each problem looks like a nail.

First be familiar with technology!



challenges

- cost for query
- N+1 problem
- monitoring
- pagination
- depth limiting
- resolver access for data
- recommend presentation by Jakub Bacic "How to create a production ready GraphQL server"

Slides and Links

Will be on repository today

Github

Questions?

Linkedin