

Arnaud Lauret @apihandyman

apistylebook.com

Collections Of Resources For API Designers

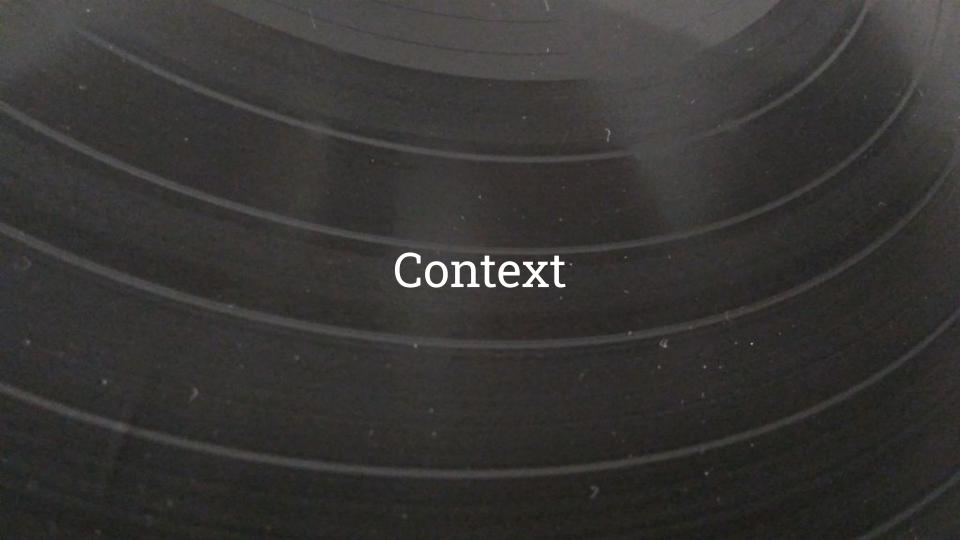


METALLICA

IRAN MAIDEN

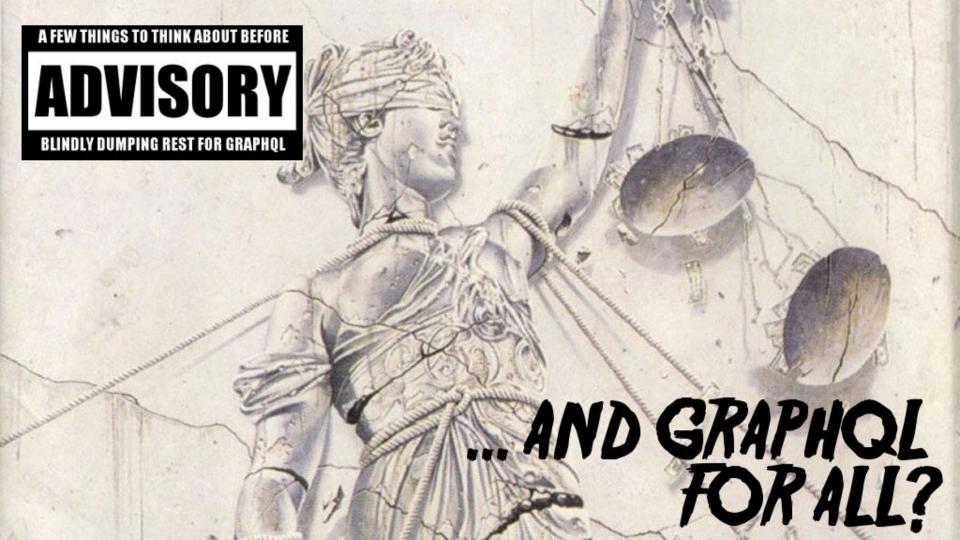






Feelings vs Context

API Style Impacts Whole API Lifecyle





Query Language for Data





```
viewer {
name
                     GET /user
avatarURL
```

```
POST https://api.github.com/graphql
"query": "{ me: viewer { name avatarURL} }"
```

```
"data": {
 "viewer": {
  "name": "Arnaud Lauret",
  "avatarURL": "https://avatars0.githubus..."
```

```
me: viewer {
fullname: name
 picture: avatarURL
```

```
"data": {
 "me": {
  "fullname": "Arnaud Lauret",
  "picture": "https://avatars0.githubus..."
```

GET /user

GET /user/repos?sort=created&direction=desc

```
{ me: viewer {
  fullname: name
  picture: avatarURL
  repositories(
    first: 2,
    orderBy: {field: CREATED_AT, direction: DESC}) {
   edges {
    node {
     name
```

```
"data": {
 "me": {
  "fullname": "Arnaud Lauret",
  "picture": "https://avatars0.githubus...",
  "repositories": {
   "edges": [
    {"node": { "name": "apistylebook-api"} },
    { "node": { "name": "restfest-videos-data"}}
```

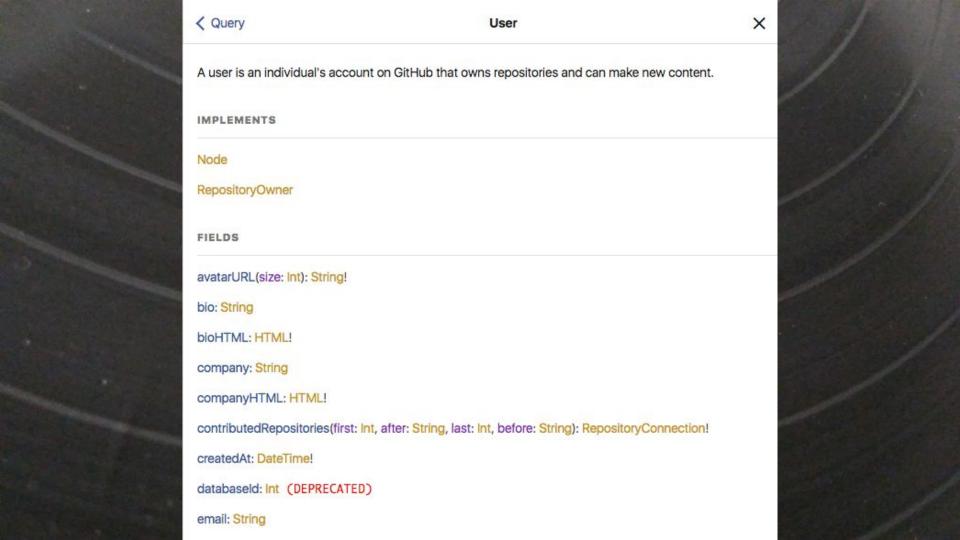
```
me: viewer { name avatarURL }
kin: user(login: "kinlane") { name avatarURL }
mike: user(login: "mamund") { name avatarURL }
graphqlRepos: search(
     first: 5, query: "graphql", type: REPOSITORY) {
  edges { node { ... on Repository {
  name
  description
 }}}
```



```
"data": {
 "me": {
                                 "data": {
  "myFirstTwoRepos":
                                  "me": {
   "edges": |
                                   "myFirstTwoRepos":
      repo": {
                                     {"name": "a-repo"}
      "name": "a-repo"
```



```
__type(name:"User") {
 fields {
  name
  description
```





Consumer Libraries GraphiQL Explorer Server Libraries



- Data you need
- Less data volume
- Less API calls
- Schema
- Providers and consumers tools

learngraphql.com

developer.github.com/early-access/graphql/

Fascinating but let's keep a cool head





GET /users/{id}
PATCH /users/{id}
DELETE /users/{id}

Consistency & Predictability

Query: Read Mutation: Write

Inconsistency & Unpredictability



404

Е

Shared Semantic
Humans And Machines

```
user(login: "unknown-user") {
 name
```

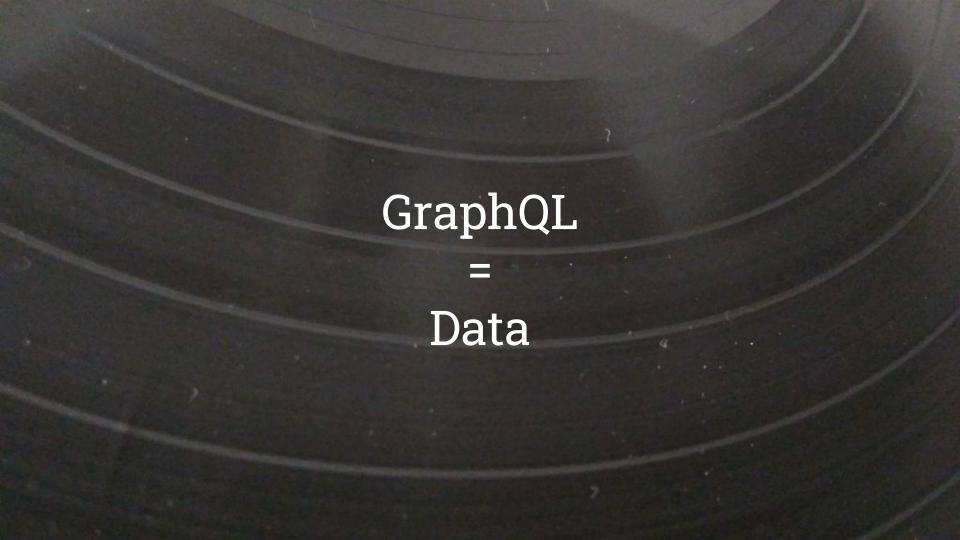
Could not resolve to a User with the login of 'unknown-user'

user { name Field 'user' is missing required arguments: login

Specific Semantic Humans Only



Hypermedia REST = Process





REST

Standard HTTP protocol caching

GraphQL = Custom Client Caching



- More flexibility
- Less predictability and consistency
- Less guidance in process
- More Responsability

Good or Bad? ...it depends!



"If you suck at providing REST API You'll suck at providing GraphQL API"

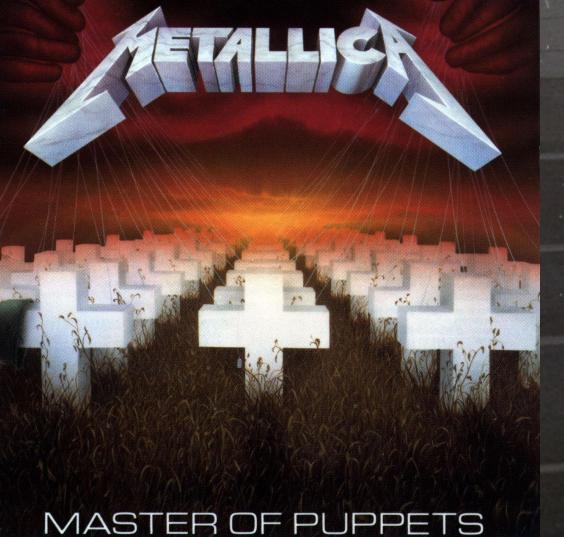
Arnaud Lauret, @apihandyman



Still a well designed consumer friendly abstraction

Need stronger governance





A5 Amsus2 B B5 B7 B7/D# C C5 C#5 Cadd9 D D5 D#5 E5 Em F#m

Interface Description it not Documentation

Less data may be more data and more CPU

Track 3

Bandwidth Usage





Query used + Query depth + Data Volume

Pricing?

API Gateway Impacts?



Same Challenges As REST

But From A New Perspective

- Design Governance
- Documentation
- Architecture And Infrastructure
- Pricing Model
- API Tooling



... And GraphQL for all API toolboxes

FEE FILL

IT DEPENDS!!!

