GraphQL Briefing 🌎

Q2 2021

📣 What's New

Rover

New CLI to manage data graphs

- Re-written in Rust
- No longer need NPM
- Replaces Apollo CLI
- Public Preview

Workbench

VS Code extension

- Federated schema design
- Composition errors
- Query plan
- Helpful for type migrations

Subscriptions

Federation support

🗣 Talks & Learnings 🧠

How Your Infrastructure Choices Make (or Break) Your Team Culture

Jason Lengstorf, VP Developer Experience, Netlify

- Issues with complex systems
- "Slowball" effect

Learnings

- Clear boundaries
- Decoupled systems
- Small changes
- Ship often
- Make it easy to ship

Acquiring a Data Graph

Taz Singh, Founder, Guildd.io

- Challenges with big data graphs
- Inconsistency with entities

Learnings

- Align on the naming
- Principled GraphQL
- Entity normalization

Subscriptions with Federated Data

Mandi Wise, Solutions Architect, Apollo GraphQL

- GraphQL subscriptions
- Separate endpoint
- Schema re-use

Considerations

- Use only when needed
- Single service
- Caching
- Data consistency

🚀 GraphQL at Scale

Considerations & Best Practices

Nullability

Field Nullability

```
// schema definition 🤝
type Restaurant {
  name: String!
  rating: Int!
}

type Query {
  topRestaurant(): Restaur
}
```

```
// client query 👌
query {
    topRestaurant {
        name
        rating
    }
}
```

```
// all data present 👍
{
  topRestaurant: {
    name: "OneTaco",
    rating: 5
  }
}
```

```
// no restaurant exist 👎
{ topRestaurant: null }
```

```
// partial data 🙀
{
  "data": {
    "topRestaurant": nul
  },
  "errors": [{
    "message": "Cannot r
  }]
}
```

Argument Nullability

```
// schema definition 🤝
type RestaurantFilter {
  cuisine: String
  rating: Int
}

type Query {
  topRestaurant(filter: RestaurantFilte
}
```

```
// old client query 💔
query {
    topRestaurant {
        name
        rating
    }
}
```

```
// new client query 😐
query {
    topRestaurant ({rating: 5}) {
        name
        rating
    }
}
```

Other Considerations

- Understanding query performance during development
- Alerting on long running queries
- Query design with security in mind
- Do not pass user identifiers with queries
- Governance Team
- Building patterns
- Include people in the process

🛠️ Tooling for Testing

- Unit testing - Jest
- Integration testing - Nock
- E2E testing - K6
- Load Testing - K6

📚 Resources

- Rover CLI - https://www.apollographql.com/docs/rover/
- Apollo Workbench - https://marketplace.visualstudio.com/items?itemName=apollographql.apollo-workbench
- GraphQL Summit recordings: https://www.youtube.com/playlist?list=PLpi1lPB6opQybdg6sPe-jNgYdDcBjSShH
- Optimize for Deletion: https://www.netlify.com/blog/2020/10/28/optimize-for-deletion-speed-up-development-without-adding-risk/
- Principled GraphQL - https://principledgraphql.com/
- GraphQL Subscriptions - https://www.apollographql.com/docs/apollo-server/data/subscriptions/
- Federation subscription example - https://github.com/apollographql/federation-subscription-tools
- Nullability in GraphQL - https://www.apollographql.com/blog/using-nullability-in-graphql-2254f84c4ed7/
- K6 - https://github.com/k6io/k6
- Nock - https://github.com/nock/nock

🙏 Thank You