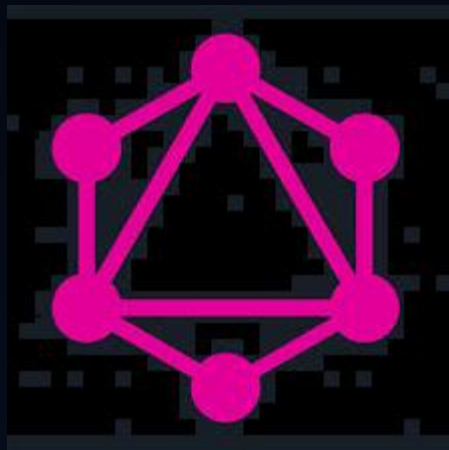


GraphQL

UMA ALTERNATIVA A ARQUITETURA REST



RODRIGO RAFAEL

Agenda

- Problemas da arquitetura REST;
- GraphQL – Resolvendo os problemas;
- Exemplos;
- Dúvidas comuns;
- Implementação;
- Referencial.

Problemas da arquitetura REST

POR QUE CRIAR O GRAPHQL?

REST: *Representational State Transfer*

- Modelo de Arquitetura;
- Vários pontos de acesso (URL), um por recurso;
- Cada operação (GET, POST, PUT...) retorna um conjunto de dados fixo;
- Adicionar ou remover dados ou parâmetros quebra a API.

- API bem normalizada resulta em muitas conexões HTTP;
- Tráfego de dados inúteis ao aplicativo;
- *Endpoints* otimizados para aplicativos - Mudanças de requisitos no *front-end* costumam precisar de adaptações no *backend*;
- Falta Documentação;
- Falta Validação e Garantias;
- Falta ambiente de testes/playground.

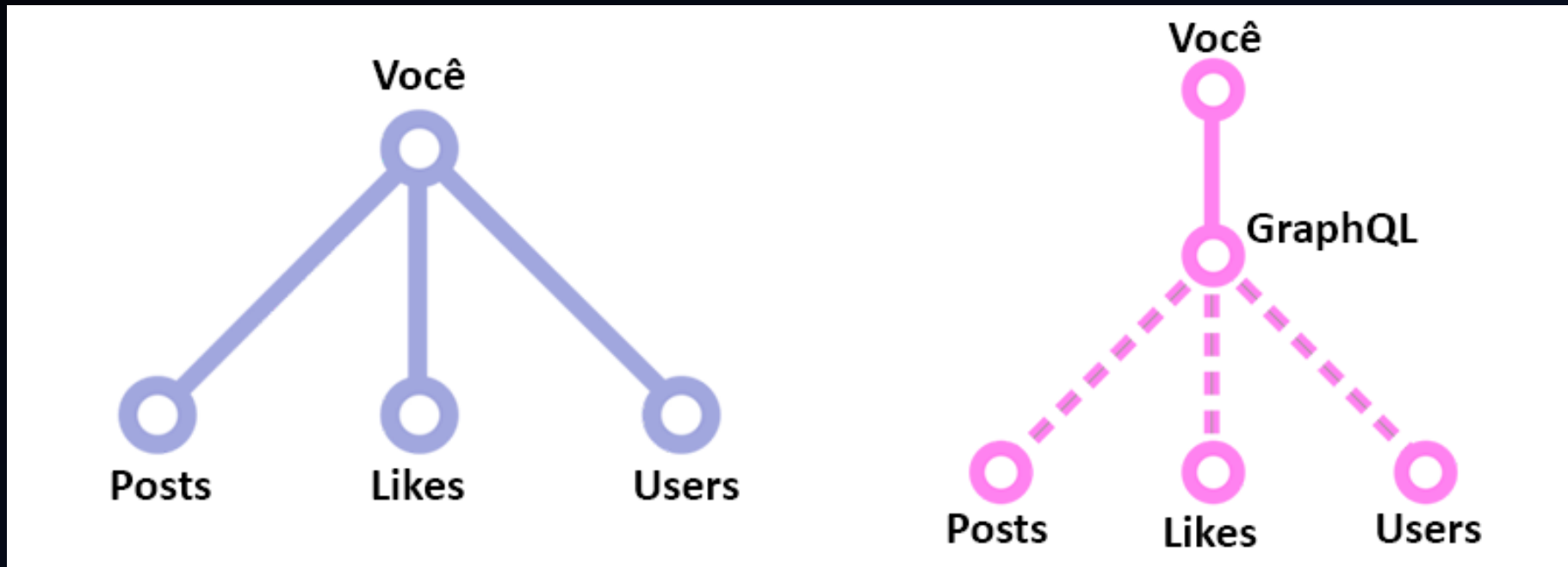


GraphQL

RESOLVENDO OS PROBLEMAS

- GraphQL é uma linguagem de consulta criada pelo Facebook em 2012 e lançada publicamente em 2015.
- É considerada uma alternativa para arquiteturas REST, além de oferecer um serviço *runtime* para rodar comandos e consumir uma API.

Post no Facebook, um problema



Na prática:

```
query {  
  posts  
}
```

```
query {  
  posts {  
    likes  
    comments  
    shares  
  }  
}
```

```
query {  
  posts (userId: 157321){  
    likes  
    comments  
    shares  
  }  
}
```

```
type Query {  
  hero: Character  
}
```

```
type Character {  
  name: String  
  friends: [Character]  
  homeWorld: Planet  
  species: Species  
}
```

```
type Planet {  
  name: String  
  climate: String  
}
```

```
type Species {  
  name: String  
  lifespan: Int  
  origin: Planet  
}
```

```
{  
  hero{  
    name  
    friends{  
      name  
      homeWorld{  
        name  
        climate  
      }  
      species{  
        name  
        lifespan  
        origin{  
          name  
        }  
      }  
    }  
  }  
}
```

Resolvers

- São funções responsáveis por, como o nome sugere, revolver um pedido e devolver o dado solicitado. Abaixo segue o exemplo de um possível resolver para ***Query.User***.

```
type User {  
  uuid: ID!  
  name: String!  
  cpf: String  
  zipcode: String  
  address: Address  
}  
  
type Address {  
  street_address: [String!]!  
  city: String!  
  state: String!  
}
```

```
const singleUserResolver = (root, args, context) => {  
  const { uuid } = args  
  const { knex } = context  
  
  return knex('users')  
    .where({ uuid })  
    .first()  
}
```

Métodos padrões

API REST

- GET /users
- POST /users
- PUT /users/:id
- DELETE /users/:id

GraphQL

- POST /graphql

```
{  
  query allUsers{  
    users()  
  }  
  query allOrders{  
    orders()  
  }  
}
```

Métodos do GraphQL

- Query – consulta;
- Mutation – criar, atualizar ou deletar informação;
- Subscription – monitorar em *real-time*.


```
{  
  mutation updateUser{  
    updateUser(id: 1){  
      name: "Rodrigo"  
    }  
  }  
  
  query users{  
    allUsers()  
  }  
}
```

Vantagens

- Flexibilidade;
- Linguagem de Domínio Específico (DSL) fácil e bem documentada;
- Tipagem forte e garantida pelo motor.

GraphQL não...

- Transporte, em geral HTTP;
- Segurança, em geral TLS (HTTPS);
- Serialização de dados, em geral JSON;
- Autenticação e Autorização, em geral OAuth v2 via HTTP + Headers

- 
- GraphQL não é um banco de dados;
 - GraphQL não é um framework;
 - GraphQL não é exclusivo para HTTP/APIs.

Dúvidas comuns

- Se eu for trabalhar com GraphQL, terei que usar React?
- A ideia do cliente dizer o que quer não me parece segura...
- Ainda acho que não preciso disso. Minha aplicação já está muito boa com as APIs existentes.

Implementação

[HTTPS://GITHUB.COM/RODRIGORAFEAELDAMACENO/GRAPHQL](https://github.com/rodrigorafaeldamaceno/graphql)

Referências

- InfoQ - GraphQL: O que, como e quando;
- CodeCasts - Entendendo o que é GraphQL através de exemplos;
- TreinaWeb - Você conhece o GraphQL e como ele pode te ajudar?
- Diego Fernandes - GraphQL aplicado no Node.js
- <https://graphql.org/>

