

Desenvolvimento Web II

Aula 06 - GraphQL

Prof. Fabricio Bizotto

Instituto Federal Catarinense
fabricao.bizotto@ifc.edu.br

Ciência da Computação
22 de março de 2024

- 1 GraphQL
 - Experimentos

Web Service

GraphQL

Definição

GraphQL - *Graph Query Language*

- Linguagem de consulta para APIs.
- Foi criada pelo Facebook em 2012 e tornou-se open-source em 2015.
- É uma alternativa ao REST. Permite que os clientes solicitem dados de forma flexível.

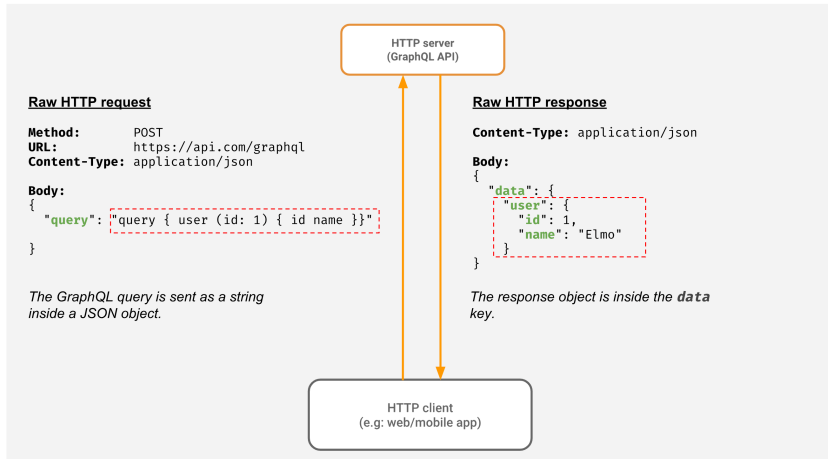


Figura: Estrutura GraphQL

REST	GraphQL
É uma arquitetura	É uma linguagem de consulta
Os dados são expostos como recursos	Os dados são expostos como um grafo
Overfetching ou underfetching são comuns.	Os clientes solicitam apenas os dados necessários, evitando overfetching e underfetching.
Múltiplos endpoints para diferentes recursos.	Um único ponto de entrada para todas as operações.
Operações de escrita (POST, PUT, DELETE) têm endpoints separados.	Mutações são tratadas no mesmo sistema de tipos e no mesmo endpoint que as consultas.
Utiliza diversos métodos HTTP (GET, POST, PUT, DELETE).	Usa o método POST para todas as operações. Pode ser usado com qualquer protocolo de transporte.
Pode exigir versionamento da API para adicionar ou modificar funcionalidades.	Não requer versionamento devido à flexibilidade nas consultas.

Tabela: REST vs GraphQL

Mutation

- É um tipo de operação que permite **criar, atualizar ou excluir dados.**
- É semelhante a uma operação de escrita no REST.
- As mutações são tratadas no mesmo sistema de tipos e no mesmo endpoint que as consultas.

```
1 mutation {  
2   createTask(  
3     title: "Estudar",  
4     description: "Estudar Web Service") {  
5     task {  
6       id  
7       title  
8       description  
9     }  
10  }  
11 }
```

```
{  
  "data": {  
    "createTask": {  
      "task": {  
        "id": "3",  
        "title": "Estudar",  
        "description": "Estudar Web Service"  
      }  
    }  
  }  
}
```

Figura: Exemplo de Mutation

Query

- É um tipo de operação que permite recuperar dados.
- É semelhante a uma operação de leitura no REST.
- As consultas são tratadas no mesmo sistema de tipos e no mesmo endpoint que as mutações.

```
1 query {  
2   tasks {  
3     title  
4   }  
5 }
```

```
{  
  "data": {  
    "tasks": [  
      {  
        "title": "Estudar"  
      }  
    ]  
  }  
}
```

Figura: Exemplo de Query

Experimento 1

- Consumir a API GraphQL <https://graphqlpokemon.favware.tech/v8>.
- Use uma ferramenta como o Thunder Client para testar a API.
- Navegue pela documentação da API e teste os endpoints.

Experimento 2

Consumir a API GraphQL Countries.

Repositório da API: GitHub.

- Utilizar uma biblioteca/framework de sua escolha para o desenvolvimento frontend (por exemplo, React, Vue, Angular, etc.).
- Fazer solicitações HTTP para a API Countries (<https://countries.trevorblades.com/>) para obter informações sobre os países.