

Introdução ao GraphQL

Cleuton Sampaio

GraphQL é baseado em grafos

Você pode solicitar qualquer pedaço do grafo de informação.

Usamos requests **GET** ou **POST** para enviar queries ao servidor GraphQL. É mais comum e seguro enviarmos utilizando POST, que não é **idempotente**.

Esquema

Leia a documentação oficial do [GraphQL](#). A implementação em **Golang** pode ser um pouco diferente, então vamos nos ater ao site oficial.

Imagine um esquema como esse:

```
type Query {  
  url(name: String!) : String  
}
```

Esta é a linguagem **SDL - Schema Definition Language** do GraphQL. Estamos declarando um tipo de dados "Query" que tem um campo "url". Os campos podem ou não ter argumentos. Neste caso, para obter a URL, precisamos passar o NOME do site. O nome não pode ser null (note a exclamação após o tipo de dados).

Se quisermos saber a URL de um site, fazemos uma query assim:

```
{  
  url(name: "github")  
}
```

E podemos enviar esta query utilizando **POST** ou **GET**. No caso de GET, poderíamos enviar algo assim:

```
http://myapi/graphql?query={url(name:"github")}
```

Em um Request POST, podemos enviar um content-type **application/json** no **body** com este formato:

```
{  
  "query" : "{url(name: \"github\")}"  
}
```

O resultado poderia ser algo assim:

```
{  
  "data" : {"url":"http://github.com"}  
}
```

Mutations

Se quisermos alterar algo no servidor, então precisamos definir uma mutação (mutation). Por exemplo:

```
type Mutation {  
  addSite(name: String!, url: String!): Boolean!  
}
```

A mutação "addSite" recebe dois argumentos e retorna um Boolean (True / False), indicando se foi possível ou não adicionar o site.

Para executar, basta enviar isso em um POST:

```
{  
  "query" : "mutation {addSite(name: \"netflix\", url:  
\"netflix.com\")}"  
}
```

A resposta pode ser algo assim:

```
{  
  "data" : {"addSite":true}  
}
```