

Para saber mais: Introspecção

Quando desenvolvemos uma API, é sempre bom disponibilizar documentação sobre ela, para que clientes (por exemplo, quem for fazer o front-end do produto) saibam quais dados podem ser trabalhados e de que forma.

O GraphQL tem uma ferramenta que nos permite visualizar os tipos e queries disponíveis em uma API, o que facilita muito a vida de quem vai trabalhar com ela. É chamada de introspecção, ou *introspection*.

Podemos fazer um teste com a API do GitHub que vimos anteriormente: o [GitHub's GraphQL Explorer](https://developer.github.com/v4/explorer/) (<https://developer.github.com/v4/explorer/>). No playground, digite:

```
{
  __schema {
    types {
      name
    }
  }
}
```

E você terá acesso a todos os types definidos na API do GitHub. Ou, como se trata de uma API pública, a todos os tipos que estão disponíveis:

- Scalars, como “Boolean”,
- Tipos definidos na construção da API, como “Users” e tipos relacionados a “User”;
- Tipos iniciados com `__`, que são parte do sistema de introspecção.

Outro teste interessante que podemos fazer é verificar qual é o ponto de entrada da API:

```
{
  __schema {
    queryType {
      name
    }
  }
}
```

Fazendo o teste no playground do GitHub, o retorno é `"name": "Query"`. O que significa que a API tem um type Query, onde estão definidos os pontos de entrada para consulta à API... Vale notar aqui que *poderia ser adotado qualquer nome para o tipo, mas utilizar Query é uma convenção*, então vamos utilizá-lo.

Podemos passar mais subcampos para ter mais informações ainda sobre os tipos disponíveis nessa API:

```
query {
  __schema {
    types {
      name
      kind
      fields {
```

```

      name
    }
  }
}

```

Essa query vai nos retornar informações mais completas sobre cada tipo: nome, se é objeto, scalar, input; no caso de objetos, quais campos estão associados, e muito mais.

Confira, por exemplo, os types `Boolean` e `User` e verifique as informações de cada um (é possível fazer buscas com “ctrl + f” / “cmd + f”).

Um último teste: Como saber de que se trata certo tipo, por exemplo “Actor”, que aparece na lista de tipos disponíveis na API:

```

{
  __type(name: "Actor") {
    name
    kind
  }
}

```

O retorno aqui é:

```

{
  "data": {
    "__type": {
      "name": "Actor",
      "kind": "INTERFACE"
    }
  }
}

```

Ou seja, `Actor` é um tipo Interface. Vamos ver o que é esse tipo mais para a frente neste curso.

Por definição, tanto o playground quanto a introspecção não devem ficar disponíveis na versão da API que está em produção, como uma boa prática. Porém, caso queira disponibilizar — por exemplo, se for desenvolver uma API pública, é possível declarar isso explicitamente na instância de `ApolloServer` :

```

const server = new ApolloServer({
  typeDefs,
  resolvers,
  introspection: true,
  playground: true,
});

```

Há muito mais coisa que podemos fazer com a introspecção! À medida em que desenvolvemos nossa API durante o curso, você pode testar com ela tudo o que foi feito aqui com a API do GitHub.

