



Para saber mais: Parâmetro info

Já conferimos quatro parâmetros do `resolver` e qual tipo de dado cada um deles carrega:

- `root` (ou `parent`): o resultado da chamada no “nível” anterior da query;
- `args`: os argumentos que o resolver pode receber da query, por exemplo os dados para um novo User ou um `ID`;
- `context`: um objeto com o contexto para o GraphQL, como dados sobre a conexão, permissões de usuário, etc;
- `info`: a representação em árvore da query ou da mutation.

Na maior parte dos casos podemos trabalhar sem prestar muita atenção ao parâmetro `info` ou ao que ele faz. Mas é importante sabermos um pouco mais sobre o que ele faz e como funciona.

Podemos alterar um pouco o resolver `users` e incluir uma chamada no console que nos mostre o que o resolver está recebendo em `info`:

```
users: (root, args, { dataSources }, info) => {  
  console.log(info)  
  return dataSources.usersAPI.getUsers()  
},
```

Fazendo a chamada da query `users` no playground:

```
query {  
  users {  
    nome  
  }  
}
```

Além da lista de nomes no playground, agora deve mostrar no console um objeto com várias propriedades, algumas com nomes que a gente reconhece, por exemplo `fieldName: 'users'`, `returnType: [User!]!` e algumas propriedades iniciando com `__`, como `__Schema: __Schema`.

Então, podemos afirmar que `info` é um objeto que contém a estrutura em árvore da query solicitada. Com essas informações, o resolver sabe quais campos deve retornar, os tipos de dados destes campos, quais são os níveis acima na query e etc.

Para informações detalhadas sobre cada uma das propriedades recebidas através do parâmetro `info`, você pode consultar este [artigo no blog da Prisma \(https://www.prisma.io/blog/graphql-server-basics-demystifying-the-info-argument-in-graphql-resolvers-6f26249f613a\)](https://www.prisma.io/blog/graphql-server-basics-demystifying-the-info-argument-in-graphql-resolvers-6f26249f613a) que está bem completo.