

GraphQL Introspection

Angenommen es liegt ein GraphQL Backend vor, in dem ein Schema mit Typen und deren Feldern festgelegt wurde. Ein Frontend kann nun durch die GraphQL query language die gewollten Objekte durch Queries anfordern und über Mutations diese ändern. Für das Frontend wäre es sehr nützlich, wenn es möglichst unkompliziert das vom Backend festgelegte Schema herausfinden könnte und damit weiß, welche Queries und Mutations an das Backend überhaupt gestellt werden können.

Um dies zu implementieren kann das GraphQL Introspection System genutzt werden. Der Vorteil dabei ist, dass die Introspection ein Teil von GraphQL selbst ist und damit direkt und bequem mit GraphQL Queries alle verfügbaren `queryTypes` und `mutationTypes` erfragt werden können. Für diese "Introspection" Queries gibt es bestimmte Felder, die immer im Root Type einer Query vorhanden sind und die gewünschten Informationen enthalten:

`__schema`, `__type`, `__typeKind`, `__field`, `__inputValue`, `__enumValue`, `__directive`. Alle diese Felder beginnen mit einem doppelten Unterstrich. Das Feld `__schema` enthält unter anderem die Felder `queryType` und `mutationType`. Somit können mithilfe des `__schema` Feldes schon alle möglichen Queries und Mutations, die das Schema im Backend definiert hat erfragt werden:

```
{
  __schema {
    queryType {
      name
      fields {
        name
      }
    }
    mutationType {
      name
      fields {
        name
      }
    }
  }
}
```

Diese Query fragt also nach dem `__schema` Feld und in diesem nach den Feldern `queryType` und `mutationType`, welche jeweils einen Namen haben. Außer diesem Namen werden noch die Namen der Felder, die sie beinhalten erfragt. Die Antwort auf diese Query (bei meinem Beispiel Backend) ist folgender JSON-Code:

```
{
  "data": {
    "__schema": {
      "queryType": {
        "name": "RootQueryType",
        "fields": [
          {
            "name": "book"
          },
          {
            "name": "author"
          },
          {
            "name": "books"
          },
          {
            "name": "authors"
          }
        ]
      },
      "mutationType": {
        "name": "Mutation",
        "fields": [
          {
            "name": "addAuthor"
          },
          {
            "name": "addBook"
          }
        ]
      }
    }
  }
}
```

Es werden die Namen aller möglichen Queries (book, author, books, authors) und Mutations (addAuthor, addBook) zurückgegeben!