GraphQL 기본 3

1 Apollo

스키마 & 타입

(https://graphql.org/learn/schema/)

(https://graphql-kr.github.io/learn/schema/)

타입 시스템 / 타입언어

GraphQL schema language

객체 타입과 필드

```
type Character {
  name: String!
  appearsIn: [Episode]!
}
```

객체 타입의 모든 필드는 0개 이상의 인수를 가질 수 있음

```
type Starshipt {
      id: ID!
   name: String!
   length(unit: LengthUnit = METER): Float
}
```

스키마 대부분의 타입은 일반 객체 타입

특별한 타입 2가지

쿼리 타입 & 뮤테이션 타입
type Query 에 우리가 조회하고자 하는 필드가 있어야 함

swapi schema 체크 / spaceX schema 체크

스칼라 타입

객체 타입은 이름과 필드를 가짐

타입을 타고타고 가다가

결국 쿼리의 끝에선 구체적인 데이터로 해석되어야함

기본 제공 Int / Float / String / Boolean / ID

선언도 가능 scalar Date

(https://studio.apollographql.com/sandbox/schema/reference/scalars)

스칼라 타입

객체 타입은 이름과 필드를 가짐

타입을 타고타고 가다가

결국 쿼리의 끝에선 구체적인 데이터로 해석되어야함

기본 타입

Int / Float / String / Boolean / ID

특별한 종류의 스칼라 타입

열거형 타입(Enumeration types)

```
enum Episode {
    NEWHOPE
    EMPIRE
    JEDI
}
```

(https://studio.apollographql.com/sandbox/schema/reference/enums)

리스트 / Non-Null

일종의 유효성 검사

리스트는 [String]

Non-Null은 String!

조합 Quiz

[String]: null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

[String!] : null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

[String]! : null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

[String!]!: null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

조합 Quiz

[String]: null: O / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: O

[String!] : null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

[String]! : null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

[String!]!: null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

조합 Quiz

[String]: null: O / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: O

[String!] : null: O / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: X

[String]! : null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

[String!]!: null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

조합 Quiz

[String]: null: O / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: O

[String!] : null: O / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: X

[String]! : null: X / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: O

[String!]!: null: / []: / [‘a’, ‘b’]: [‘a’, null, ‘b’]:

조합

[String]: null: O / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: O

[String!] : null: O / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: X

[String]! : null: X / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: O

[String!]!: null: X / []: O / [‘a’, ‘b’]: O [‘a’, null, ‘b’]: X

인터페이스(interface)

특정 필드들을 포함하는 추상 타입

해당 인터페이스를 구현(implements)한 타입들이 가져야 할 필드 정의

Inline Fragments 가 Interface나 Union 타입에서 쓰인다고 했었음

swapi Node 보기

유니온 타입(Union types)

인터페이스와 유사하지만 타입 간의 공통 필드를 특정하지 않음

```
union SearchResult = Human | Droid | Starship
```

입력 타입(Input types)

뮤테이션으로 생성될 복잡한 객체를 쉽게 전달 가능

```
input ReviewInput {
    stars: Int!
    commentary: String
}
```

(https://studio.apollographql.com/sandbox/schema/reference/inputs/users_insert_input)

정리

GraphQL 기본 3

타입 시스템	객체 타입과 필드
특별한 타입	쿼리 타입 / 뮤테이션 타입
스칼라 타입	구체적 데이터
기타	인터페이스 / 유니온 / 인풋