**GraphQL 기반 대규모 웹 서비스 구축**

팀원 : 2015104215 장재혁

**요 약**

1. **서론** 
   1. **연구 배경**

기존 웹 서비스는 백엔드-프론트로 나누어진 웹 개발환경에서 RESTful API라는 규칙 아래 네트워크 통신을 주고받았다. RESTful이란 자원의 행위를 표현하는 방법으로 HTTP 통신을 개발자가 이해하기 쉽도록 설계하는 규칙으로 많은 웹 서비스에서 소통 비용을 줄이고 좋은 협업을 이끌어낼 수 있었다. 그러나 모바일 환경이 전 세계 네트워크 시장에서 대세가 됨에 따라서 각종 단말기들의 화면 규격이 파편화되면서, 기존 RESTful API로 대응하기 어려운 상황들이 많이 생기게 되었다. 기존 REST규칙에 따라서 설계된 API들은 PC,모바일 두 환경에서 동일하게 사용되기 어려웠고 따라서 여러 개의 End Point를 추가하여 관리해야 하는 어려움이 증가했다.

* 1. **연구목표**

본 연구는 GraphQL 기반으로 대량의 사용자를 대상으로 하는 웹 커뮤니티 서비스를 구축함으로써 기존 REST 환경에서 발생하던 문제점을 해결한다. 특히 대규모 웹 서비스인 만큼 많은 개발자가 협업하는 상황을 고려하여 구조화된 개발 방식을 제안하여 협업과 유지보수라는 측면에서 서비스를 설계하도록 한다. 이를 통해 프론트엔드와 백엔드 모두 구축하여 가장 기본적인 웹 커뮤니티 기능인 글 작성, 수정, 입력, 댓글, 추천 기능을 구현하고 AWS에 연동시키는 것을 목표로 한다.

1. **기존 연구**
   1. **GraphQL**

GrahpQL이란 Graph Query Language의 줄임말로, Facebook에서 개발한 데이터 질의어(Query)이다. 기존 SQL과 다르게 웹 클라이언트에서 서버로부터 자원을 가져오는 방식을 목적으로 하며 다음과 같은 특징을 가진다.

- 하나의 End point에서 모든 API 요청을 처리한다.

- API 요청의 처리는 Schema, Resolver 두 부분으로 나눠진다.

- Schema를 통해서 서버에 저장된 자원을 미리 정의하여 이를 클라이언트와 공유할 수 있다. 이 요청하는 방식에는 Query와 Mutation이 있다.

- 데이터의 타입을 미리 정의함으로써 형태의 따라 요청한다.

- 원하는 데이터만을 요청할 수 있으므로 over-fetching / under-fetching 문제를 해결한다.

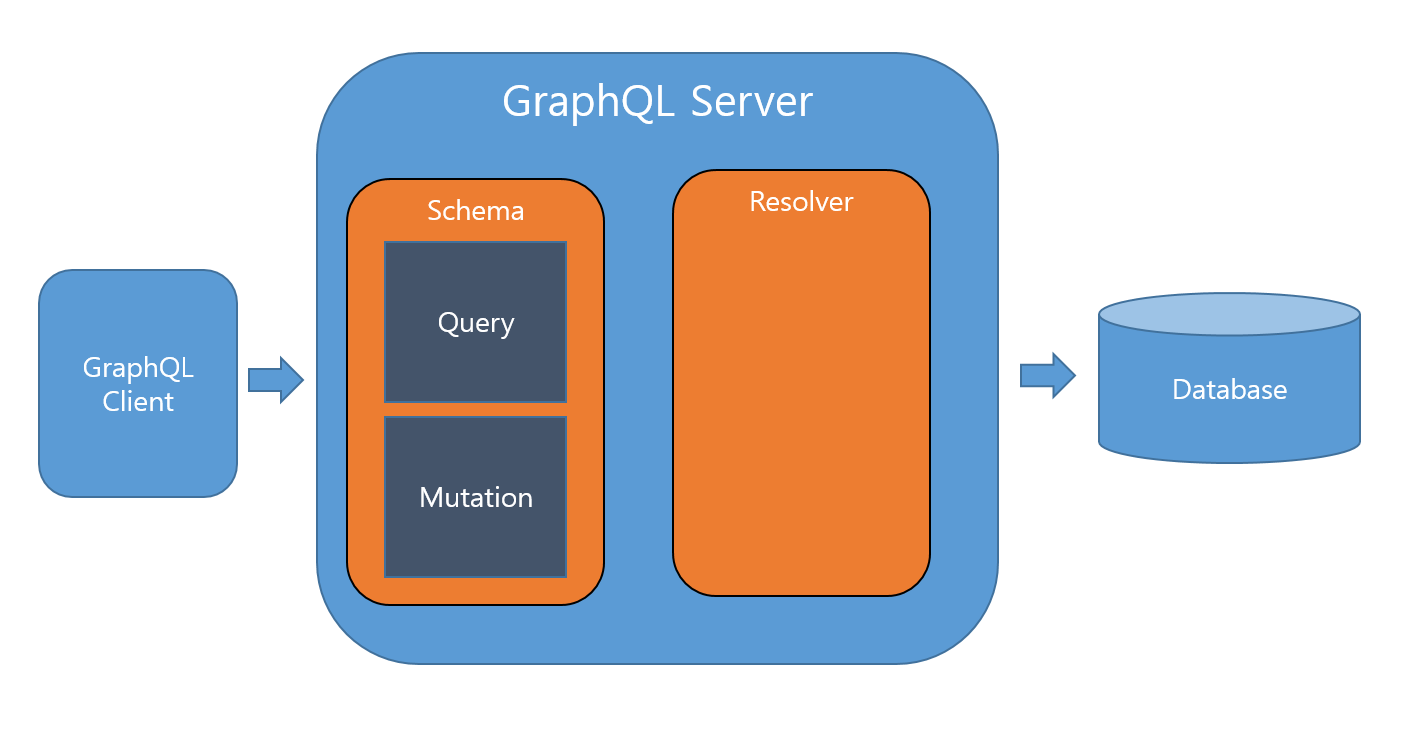


Figure 1 GraphQL Architecture

* 1. **Nest.Js**

Nest.js란 node.js를 기반으로 한 웹 백엔드 오픈소스 프레임워크이다. 웹 프론트엔드/ 백엔드 환경에서 javascript가 공용언어로써 많은 부분을 차지하게 됨에 따라 많은 프레임워크가 공존하게 되었다. 그 중 Nest.js는 MVC기반 아키텍처를 프레임워크 단에서 제공함으로 개발자가 미리 정의된 방식으로 개발하기 때문에 대규모 협업이 이루어지는 서비스에 유리하다. Nest.js는 여러 개의 Module로 계층화 시켜 개발자로 하여금 전체 프로젝트를 파악하기 유리하기 때문이다.

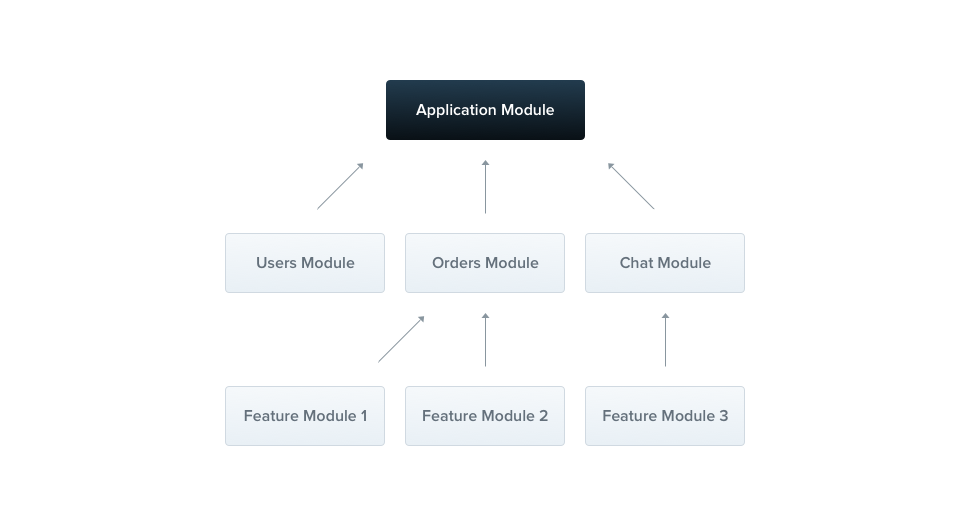


Figure 2 Nest.js Architecture

* 1. **기존 연구의 문제점**
     1. **범용성의 문제**

기존 논의된 GraphQL 기반 API관련 연구들은 모두 특정 상황에 맞는 API 서버의 설계였다. 즉 대부분의 웹 서비스에서 활용되는 CRUD 기능과 게시판 기능에서의 GraphQL의 장점에 대해서 깊게 다루지 못하였다. 또한 GraphQL의 설계 구조와 기존 REST와의 구조적 차이에서 발생되는 효과에 대해서도 논의가 되지 못하였다.

* + 1. **대규모 기반 설계의 부재**

기존 논의는 모두 웹 서비스의 단순 구축에 대해서만 다루었다. 많은 사용자의 사용성이나 개발자들간의 협업을 고려하지 않은 서비스였기 때문에 확장성과 유지보수에 대한 논의도 부족한 실정이다. 또한 사용자의 단말기에 따라 변화되는 API 요청에 대해서도 논의가 부족하였으며, 이에 따라 규모 있는 서비스를 설계하기 위한 많은 요구사항들을 다루지 못하였다.

1. **프로젝트** 
   1. **기존 연구와 차이점 및 해결방안’**

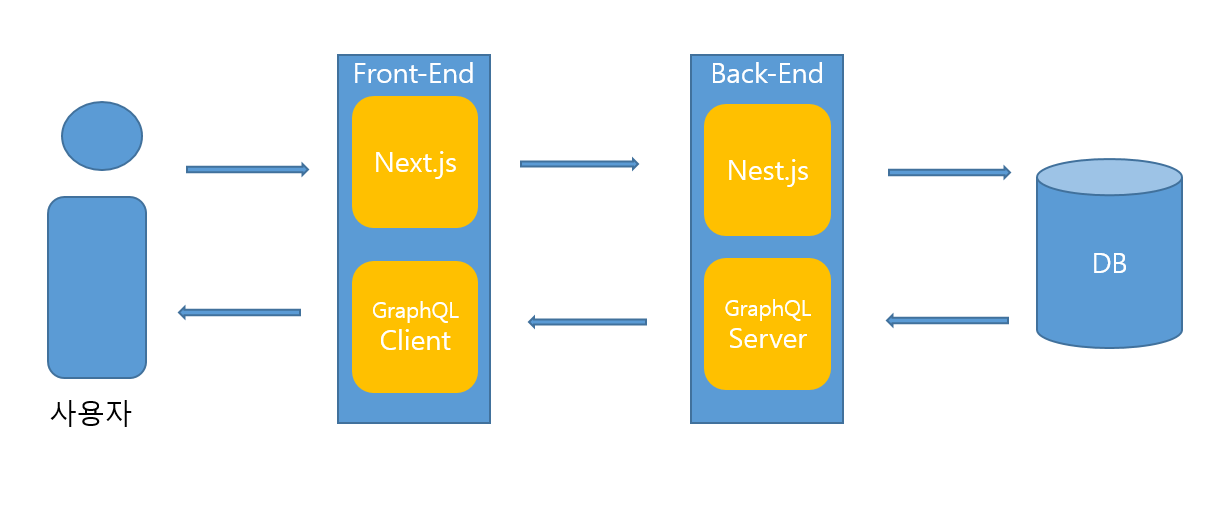
본 프로젝트는 웹 커뮤니티 사이트를 GraphQL을 활용하여 구축함으로써 앞서 언급한 기존 연구에서 논의되지 못한 주제에 대해서 논의하도록 한다. 게시판의 핵심 기능들을 GraphQL로 구현함으로써 기존 REST에서 겪는 문제들을 비교해보고 그 이점에 대해서 파악하도록 한다.

특히 많은 사용자의 트래픽을 감당하기 위한 구조나 서버 관리를 효율적으로 하기 위한 구조를 제시함으로써 대규모 웹 서비스에 대한 논의하는 것을 차이점으로 둔다. 이를 위해 개발 및 협업 방식에 대해서도 연구하여 해결방안으로 제시한다.

* 1. **프로젝트 내용**

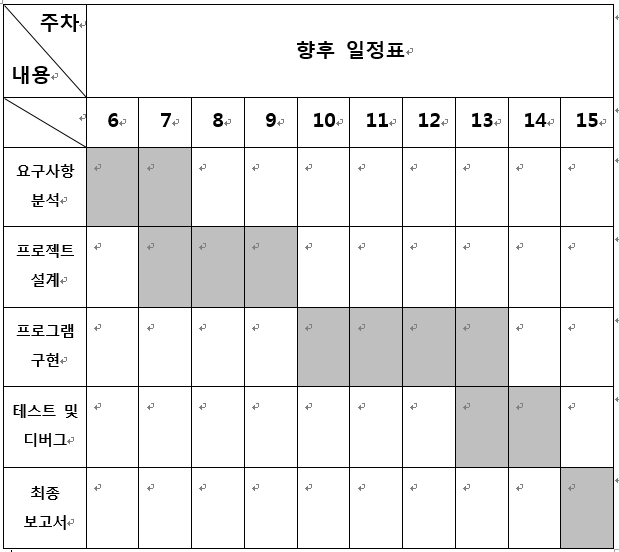
게시판을 활용하는 커뮤니티 사이트 구축을 목표로 한다. 웹 프론트엔드 – 백엔드로 이루어진 구조를 따르면서 아래와 같은 사이트 요구사항을 만족하도록 한다

* + - 1. 게시판 글 등록, 읽기, 수정, 삭제 기본적인 CRUD 기능
      2. 추천, 댓글, 신고 등 커뮤니티 기능
      3. 사용자의 작성 글, 댓글 조회 기능

이를 달성하기 위한 API 서버 구조는 GraphQL을 활용하면서 코드의 재사용성과 유지보수성, 확장성을 고려하도록 한다. 즉, 프론트엔드-백엔드 구조에서 동일한 개발언어를 통해서 개발하여 공통부분을 최대한 재사용함과 동시에 API 명세도 공유하여 협업을 손쉽게 만들도록 한다. 이는 GraphQL의 Schema를 GraphQL client와 Server 측면에서 공유하여 데이터 명세를 획일화 시킴으로써 구현한다. 또한 각각 프론트엔드와 백엔드는 개발 구조를 획일화 시킴으로써 새로운 개발자가 유입이 되더라도 전체 프로젝트 구조를 쉽게 파악하도록 하여 유지보수성을 향상 시키는 것을 목표로 한다. 이를 위한 전체 프로젝트 구상도는 다음과 같이 그려진다. 

프론트엔드 서버에서 요구하는 API요청을 GraphQL Client를 통해서 호출하면 해당 호출은 Nest.js 기반 GraphQL Server에서 Resolve 되어 미리 정의된 Query에 따라 DB에 해당 데이터를 fetch한다. 이때 GraphQL client 와 server간에는 Schema를 공유하며 이 Schema를 이용해 데이터 호출을 관리한다.

1. **진행 일정**

****

1. **결론 및 기대효과**

가장 웹 사이트에서 보편적인 형태인 게시판 기능을 GraphQL로 전환해봄으로써 기존에 여러가지 요구사항을 만족하는 다른 서비스에도 GraphQL이라는 또 다른 선택지를 제시한다는 점에서 의의를 가진다. 또한 협업과 유지보수라는 관점에서 대규모 기반 설계를 제시함으로써 신규 서비스를 구축하는 기업에게 좋은 Practice Case를 제공한다. 이를 통해 서비스의 안정성, 유연성 증대를 기대한다.

**참고 문헌**

**[1] 이응기, 권기웅, 윤정미. (2020). GraphQL을 이용한 가정용 에너지저장시스템 상태 API 서버 구현. 한국통신학회 학술대회논문집, (), 913-914.**

**[2] 임도영, 마이크로서비스 아키텍처와 GraphQL을 활용한 웹 서비스, 중앙대학교 대학원 석사논문**

**[3] GraphQL , https://graphql.org/**

**[4] Nest.js, https://nestjs.com/**