

About BE Project

안녕하세요.

버즈앤비 지원자 김주혜입니다.

GraphQL + MongoDB CRUD 구현과제 과정을 담은 자료입니다.

What is node.js / GraphQL/ MongoDB?

Node.js

: 자바 스크립트를 웹 브라우저로 독립시켜 서버 구현을 가능하게 함

〈특징〉

1. 비동기 이벤트 기반 환경 -> 여러가지 작업 수행 용이
2. 단일 언어 사용 -> javascript 기반
3. 경량화 -> 병렬작업, 대용량 데이터를 처리하는데 곳
4. 확장성 -> 새로운 모듈을 추가하거나 업데이트 하는 과정이 간단
5. 커뮤니티와 생태계의 다양성

Node.js

- 기존의 웹서버 : 스레드 기반 입출력처리
(스레드 : 프로세스내에서 실제로 작업을 수행하는 주체)
- Node.js : 이벤트 기반 (대규모 네트워크 서비스 개발 위해),
병렬처리, JSON형식 데이터 쉽게 처리할 수 있음

GraphQL

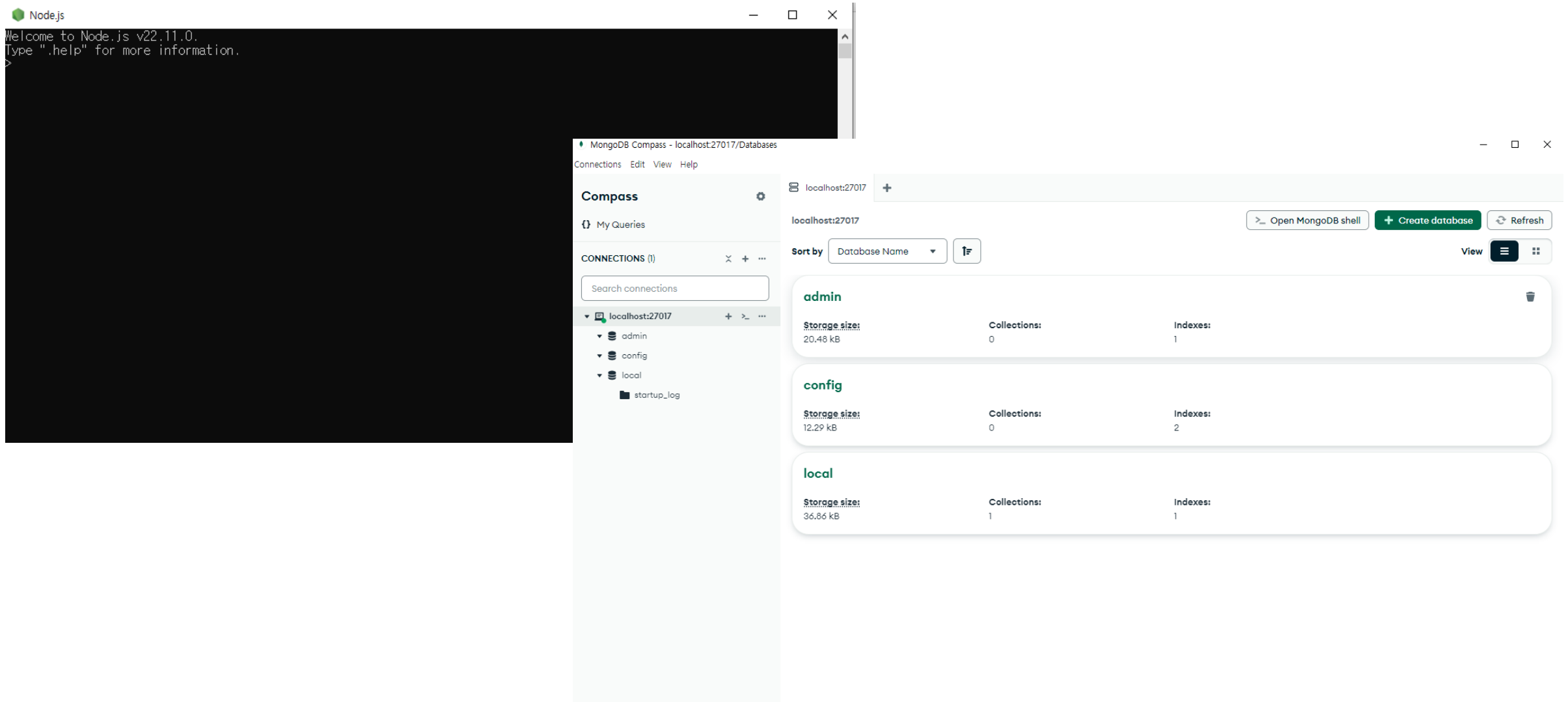
- Facebook에서 만듦 2015년 오픈소스 공개
- API를 위한 쿼리 언어
- 클라이언트가 데이터를 서버로부터 가져오는 것
- 원하는 데이터만 받을 수 있다.
- 하나의 endpoint

MongoDB

- 문서형태로 데이터 저장
- NoSQL 데이터 베이스(SQL만을 사용하는 것이 아니다.)
- 비관계형 데이터베이스 관리 시스템
- 오픈 소스

Project Process

1. Install Node.js & MongoDB



The image displays two application windows side-by-side. On the left is a Node.js terminal window with a black background and white text. It shows the Node.js welcome message and a prompt. On the right is the MongoDB Compass application window, which has a light gray interface. It shows a connection to 'localhost:27017' and a list of databases: 'admin', 'config', and 'local'. Each database entry shows its storage size, the number of collections, and the number of indexes.

Node.js Terminal Window:

```
Node.js
Welcome to Node.js v22.11.0.
Type ".help" for more information.
>
```

MongoDB Compass Window:

Connections Edit View Help

localhost:27017

Sort by Database Name

admin

Storage size: 20.48 kB Collections: 0 Indexes: 1

config

Storage size: 12.29 kB Collections: 0 Indexes: 2

local

Storage size: 36.86 kB Collections: 1 Indexes: 1

2. Install packages

npm install express express-graphql graphql

npm install mongoose

```
C:\Users\User\Desktop> npm install express express-graphql graphql
up to date, audited 103 packages in 2s
15 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

C:\Users\User\Desktop> npm install mongoose
up to date, audited 103 packages in 2s
15 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

C:\Users\User\Desktop>
```

3. Creating MongoDB Schema

```
JS User.js  JS index.js
C: > Users > user > Desktop > BE_proejct > models > JS User.js > ...
1  const mongoose = require('mongoose');
2
3  const userSchema = new mongoose.Schema({
4    src: {
5      type: String,
6      required: true,
7      uppercase: true,
8    },
9    tgt: {
10     type: String,
11     required: true,
12     uppercase: true,
13   },
14   rate: {
15     type: Number,
16     required: true,
17     min: 0,
18   },
19   date: {
20     type: Date,
21     required: true,
22     default: Date.now,
23   },
24   createdAt: {
25     type: Date,
26     default: Date.now,
27   },
28 });
29
30 module.exports = mongoose.model('User', userSchema);
31
```

4. Creating GraphQL Schema

```
JS User.js JS schema.js JS Resolvers.js JS index.js
C: > Users > user > Desktop > BE_proejct > graphql > JS schema.js > ...
1  const { buildSchema } = require("graphql");
2
3  const schema = buildSchema(`
4    type Query {
5      | getExchangeRate(src: String!, tgt: String!): ExchangeInfo
6    }
7
8    type Mutation {
9      | postExchangeRate(info: InputUpdateExchangeInfo): ExchangeInfo
10     | deleteExchangeRate(info: InputDeleteExchangeInfo): ExchangeInfo
11   }
12
13   input InputUpdateExchangeInfo {
14     | src: String!
15     | tgt: String!
16     | rate: Float!
17     | date: String
18   }
19
20   input InputDeleteExchangeInfo {
21     | src: String!
22     | tgt: String!
23     | date: String!
24   }
25
26   type ExchangeInfo {
27     | src: String!
28     | tgt: String!
29     | rate: Float!
30     | date: String!
31   }
32 `);
33
34 module.exports = schema;
35
```

5. Creating GraphQL resolvers

```
● JS schema.js ● JS Resolvers.js X JS index.js ●
> user > Desktop > BE_proejct > graphql > JS Resolvers.js > ...
const User = require("../models/User");
const axios = require("axios");

const resolvers = {
  getExchangeRate: async ({ src, tgt }) => {
    try {
      const response = await axios.get(`https://api.exchangerate-api.com/v4/latest/${src}`);
      const rate = response.data.rates[tgt]; // response -> response로 수정
      const date = response.data.date;

      return { // return 문이 try 블록 안에 있어야 함
        src: src,
        tgt: tgt,
        rate: rate,
        date: date
      };
    } catch (error) {
      console.error(`Error fetching exchange rate:`, error);
      return {
        src: src,
        tgt: tgt,
        rate: 0,
        date: 'Unknown'
      };
    }
  },
  postExchangeRate: ({ info }) => {
    return {
      src: info.src,
      tgt: info.tgt,
      rate: info.rate,
      date: info.date || '2024-12-01' // 날짜 기본값 설정
    };
  },
  deleteExchangeRate: ({ info }) => {
    return {
      src: info.src,
      tgt: info.tgt,
      rate: 0,
      date: info.date
    };
  }
};

module.exports = resolvers;
```

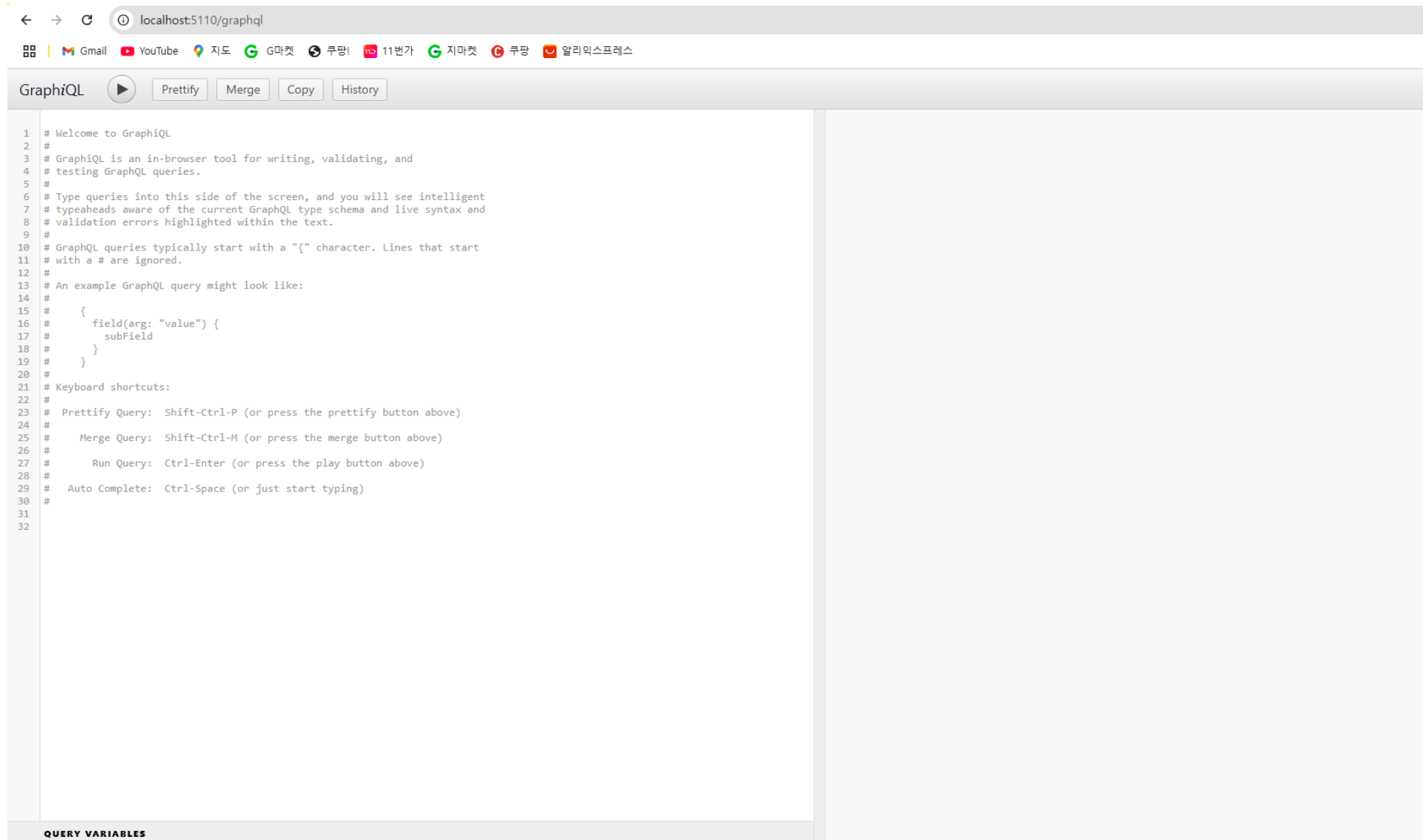
6. Creating Server

```
JS User.js  JS schema.js  JS Resolvers.js  JS index.js  JS index.js > ...
C: > Users > user > Desktop > BE_proejct > JS index.js > ...
1  const express = require("express");
2  const { graphqlHTTP } = require("express-graphql");
3  const mongoose = require("mongoose");
4  const schema = require("../graphql/schema");
5  const resolvers = require("../graphql/Resolvers");
6
7  const app=express();
8
9  mongoose.connect("mongodb://localhost:27017/graphql", {
10    useNewUrlParser: true,
11    useUnifiedTopology: true,
12  });
13  mongoose.connection.once("open", () => {
14    console.log("Connected to MongoDB");
15  });
16  app.use(
17    "/",
18    graphqlHTTP({
19      schema,
20      rootValue: resolvers,
21      graphiql: true,
22    })
23  );
24
25  app.listen(5110, () => {
26    console.log("Server is running on http://localhost:5110/graphql");
27  });
28
```

7. Connect to Server

```
C:\Users\User\Desktop> node index.js
(node:21744) [MONGODB DRIVER] Warning: useUrlParser is a deprecated option: useUrlParser has no effect since
Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:21744) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect
since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:5110/graphql
Connected to MongoDB
```

8. Verify access to <http://localhost:5110/graphql>



9 Example query in GraphQL

```
JS User.js JS schema.js JS Resolvers.js C:\...\graphql JS Resolvers.js C:\...\D:\
C: > Users > user > Desktop > JS Resolvers.js > ...
1  const User=require('./models/User');
2
3  const resolvers={
4    getExchangeRate: ({ src, tgt }) => {
5      return {
6        src: src,
7        tgt: tgt,
8        rate: 1396.50,
9        date: '2024-12-01'
10     };
11   },
12
13   postExchangeRate: ({ info }) => {
14     return {
15       src: info.src,
16       tgt: info.tgt,
17       rate: info.rate,
18       date: info.date || '2024-12-01'
19     };
20   },
21
22   deleteExchangeRate: ({ info }) => {
23     return {
24       src: info.src,
25       tgt: info.tgt,
26       rate: 0,
27       date: info.date
28     };
29   }
30 };
31
32 module.exports = resolvers;
```

GraphQL

Prettify Merge Copy History

```
1 query {
2   getExchangeRate(src: "krw", tgt: "usd") {
3     src
4     tgt
5     rate
6     date
7   }
8 }
9
```

```
{
  "data": {
    "getExchangeRate": {
      "src": "krw",
      "tgt": "usd",
      "rate": 1396.5,
      "date": "2024-12-01"
    }
  }
}
```

Resolvers.js 수정 전) 환율 정보 실시간 반영 x

**-> axios 사용하여 외부 API에서 환율 정보를 가져오도록
Resolvers.js 파일 수정**

← → ↻ localhost:5110/graphql?query=query%20%7B%0A%20%20getExchangeRate(src%3A%20"USD"%2C%20tgt%3A%20"USD")%20%7B%0A%20%20%20src%0A%20...

☰ | Gmail | YouTube | 지도 | G마켓 | 쿠팡! | 11번가 | 지마켓 | 쿠팡 | 알리익스프레스 |

GraphQL ▶ Prettify Merge Copy History

```
1 query {
2   getExchangeRate(src: "USD", tgt: "USD") {
3     src
4     tgt
5     rate
6     date
7   }
8 }
9
```

```
{
  "data": {
    "getExchangeRate": {
      "src": "USD",
      "tgt": "USD",
      "rate": 1,
      "date": "2024-12-02"
    }
  }
}
```

← → ↻ localhost:5110/graphql?query=query%20%7B%0A%20%20getExchangeRate(src%3A%20"USD"%2C%20tgt%3A%20"KRW")%20%7B%0A%20%20%20src%0A%20...

☰ | Gmail | YouTube | 지도 | G마켓 | 쿠팡! | 11번가 | 지마켓 | 쿠팡 | 알리익스프레스 |

GraphQL ▶ Prettify Merge Copy History

```
1 query {
2   getExchangeRate(src: "USD", tgt: "KRW") {
3     src
4     tgt
5     rate
6     date
7   }
8 }
9
```

```
{
  "data": {
    "getExchangeRate": {
      "src": "USD",
      "tgt": "KRW",
      "rate": 1394.73,
      "date": "2024-12-02"
    }
  }
}
```

Resolvers.js 수정 후) 환율 정보 실시간 반영하는 응답 출력

참고 자료

- <https://dev.to/uddeshjain/building-a-graphql-api-for-user-crud-with-nodejs-and-mongodb-9lj>
- https://www.tutorialspoint.com/graphql/graphql_example.htm
- <https://graphql.org/learn/schema/>
- <https://www.youtube.com/watch?v=EkWI6Ru8lFQ&t=319s>