

Graphql

<https://cybersoft.edu.vn/>



Mục Lục

01 Kết quả đạt được

02 Giới thiệu chung

03 Cài đặt

04 Thực hành

RESTful API

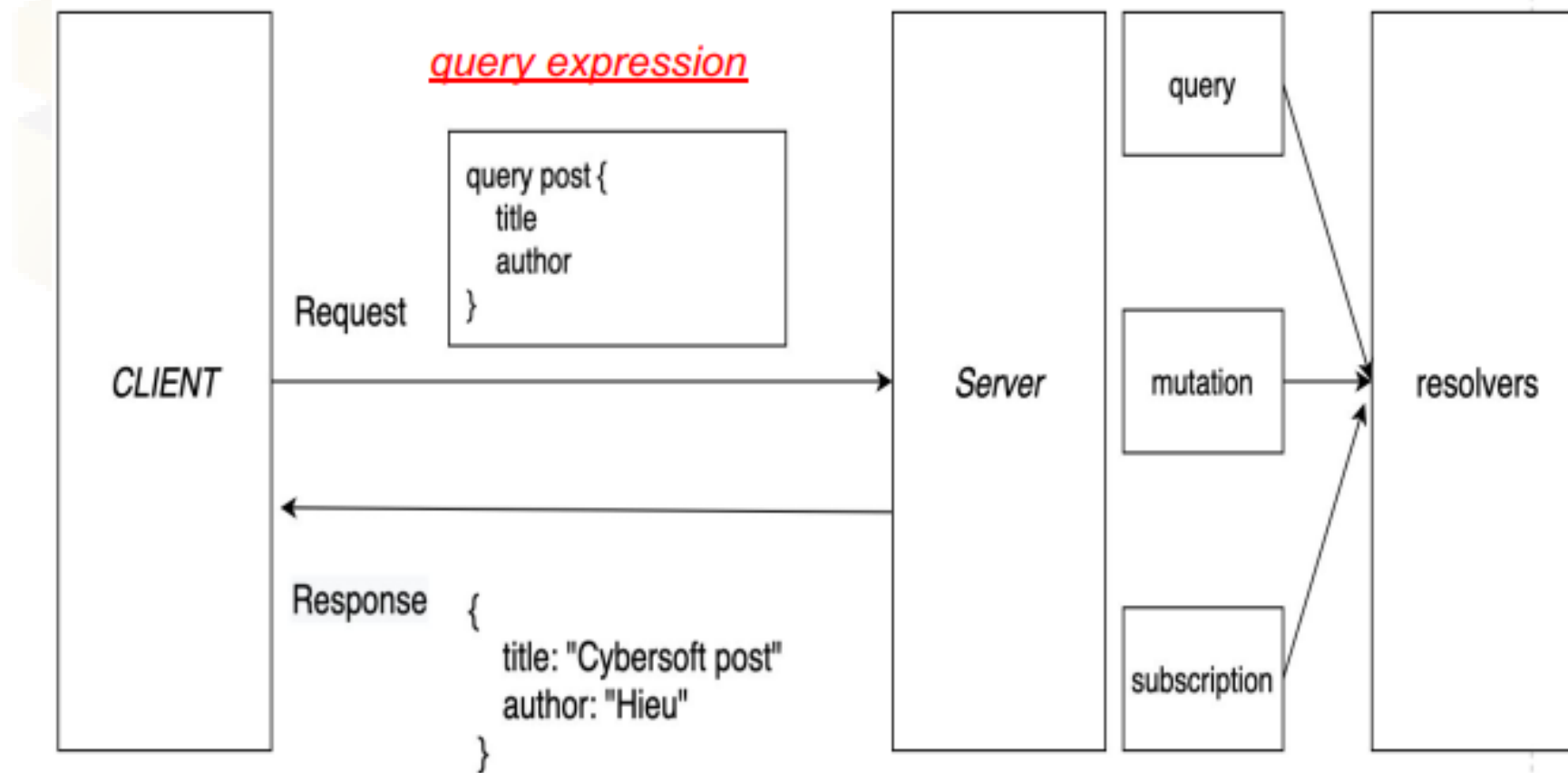


Graphql



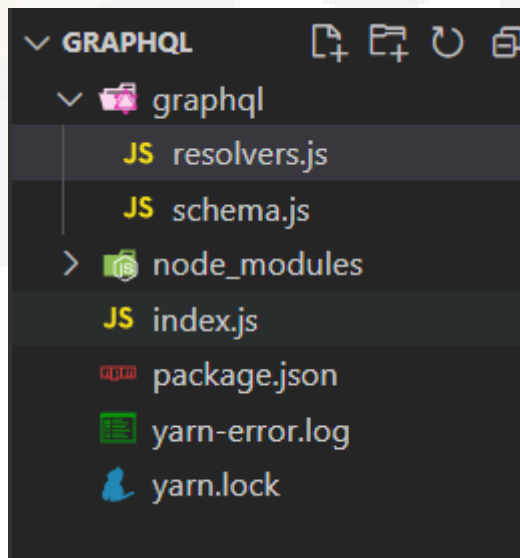
Kết quả đạt được

Front-End có thể tự chủ, linh động truy xuất bất kỳ dữ liệu mà không cần Back-End can thiệp



Cài đặt

- Trang chủ **GraphQL**: <https://graphql.org/>
- Thêm thư viện: **yarn add graphql express-graphql**
- Structure



Cài đặt:

index.js

```
JS index.js X JS resolvers.js JS schema.js package.json
JS index.js > ...
1  const express = require('express');
2
3  const app = express();
4
5  app.use(express.json());
6
7  const {graphqlHTTP} = require('express-graphql');
8  const graphqlSchema = require('./graphql/schema');
9  const graphqlResolver = require('./graphql/resolvers');
10
11 app.use('/graphql', graphqlHTTP({
12   schema: graphqlSchema, // noi dinh nghia cac cau query, gia tri tra ra cua cau query
13   rootValue: graphqlResolver, // noi xu ly logic code tuong ung voi schema
14   graphiql: true
15 }));
16
17 app.listen(8080)
18
```


schema.js

Datatype (kiểu dữ liệu):

- String
- Int
- Float
- Boolean
- ID

Không được trống (null) => String!

Trả về danh sách (list) => [String]



JS index.js JS resolvers.js JS schema.js X package.json

```
graphql > JS schema.js > graphqlSchema
1  const { buildSchema } = require('graphql');
2
3  const graphqlSchema = buildSchema(`
4    type Post {
5      title: String !
6      content: String !
7      imageUrl: String !
8    }
9
10   type User {
11     id: ID
12     userName: String
13     age: Int
14   }
15
16   type RootQuery {
17     post:[Post]
18     user (id: Int , name: String):[User]
19   }
20
21   type RootMutation {
22     createUser(name: String): [User]
23   }
24
25   schema {
26     query: RootQuery
27     mutation: RootMutation
28   }
29
30  });
31
32  module.exports = graphqlSchema;
```

resolvers.js

- Dữ liệu trả về phải khớp với các thuộc tính và định dạng dữ liệu (object or list object) đã khai báo bên hàm schema.js
- Khai báo tên đối tượng phải trùng với bên hàm schema.js
- Hàm nhận tham số được chứa trong đối tượng Arguments

```
JS index.js JS resolvers.js X JS schema.js
graphql > JS resolvers.js > graphqlResolver > user
12
13 const graphqlResolver = {
14   post: () => {
15
16     return [{
17       title: "test",
18       content: "abc",
19       imageUrl: "123"
20     },
21     {
22       title: "test 2",
23       content: "abc 2",
24       imageUrl: "123 2"
25     },
26   ],
27 },
28
29   user: (args) => {
30     return data
31   },
32   createUser: (args) => {
33
34     return [...data, {
35       id: 2,
36       userName: args.name,
37       age: 2
38     }]
39   }
40 }
41
42 module.exports = graphqlResolver;
```


Thực hành

Áp dụng graphql và prisma làm bài tập

- Cài đặt prisma và pull db_order về project
- Viết query **getUser** và **getProduct** từ database
- Viết mutations **createUser**, **updateUser** và **deleteUser** từ database

Authentication and Middleware

Tham khảo => <https://graphql.org/graphql-js/authentication-and-express-middleware/>