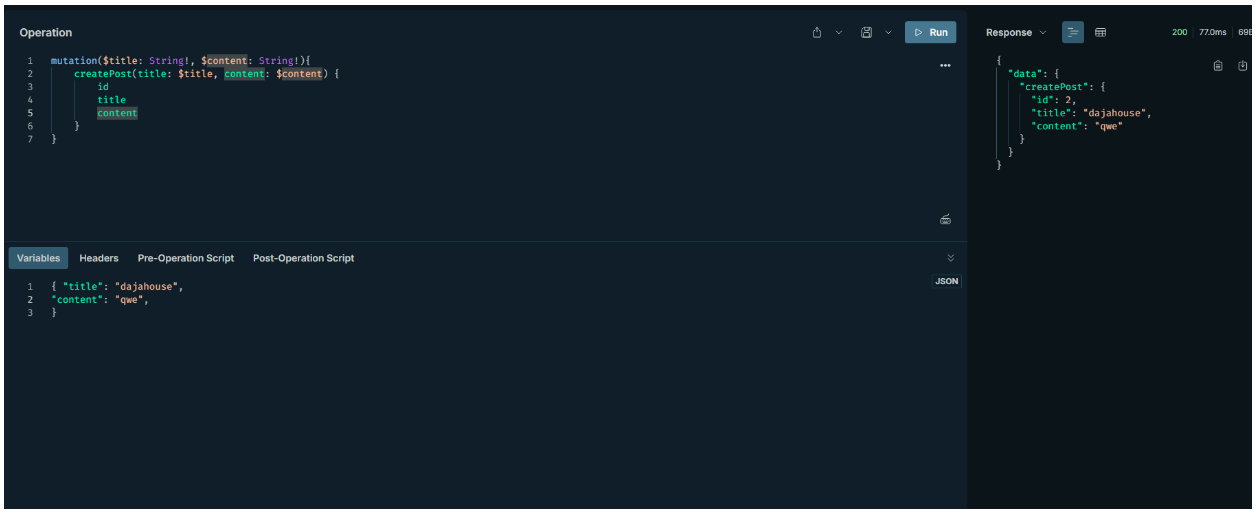
Joshua Dajao

3/16/25

Post-Service

Create:

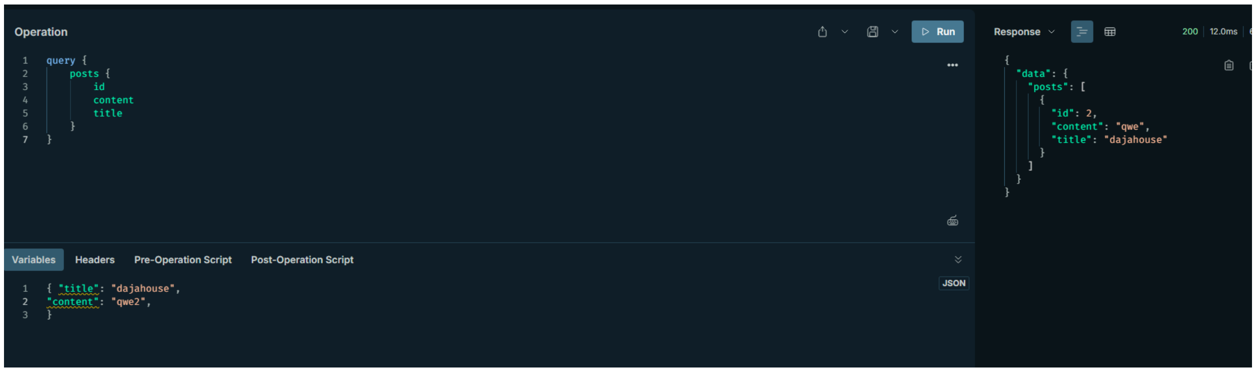


```
mutation($title: String!, $content: String!){
    createPost(title: $title, content: $content) {
        id
        title
        content
    }
}
```

```
{ "title": "dajahouse",
"content": "qwe",
}
```

```
{
  "data": {
    "createPost": {
      "id": 2,
      "title": "dajahouse",
      "content": "qwe"
    }
  }
}
```

Read

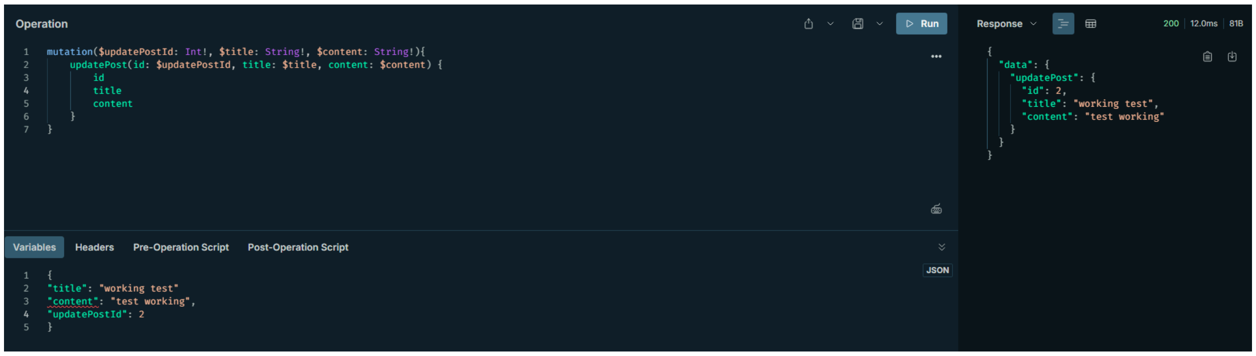


```
query {
    posts {
        id
        content
        title
    }
}
```

```
{ "title": "dajahouse",
"content": "qwe2",
}
```

```
{
  "data": {
    "posts": [
      {
        "id": 2,
        "content": "qwe",
        "title": "dajahouse"
      }
    ]
  }
}
```

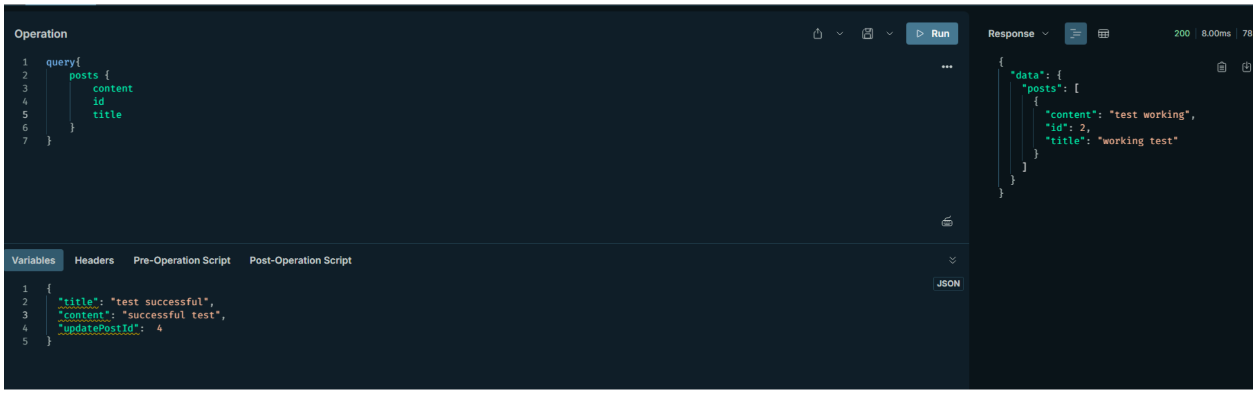
Update



```graphql
mutation($updatePostId: Int!, $title: String!, $content: String!){
    updatePost(id: $updatePostId, title: $title, content: $content) {
        id
        title
        content
    }
}
```

```json
{
"title": "working test",
"content": "test working",
"updatePostId": 2
}
```

```json
{
  "data": {
    "updatePost": {
      "id": 2,
      "title": "working test",
      "content": "test working"
    }
  }
}
```

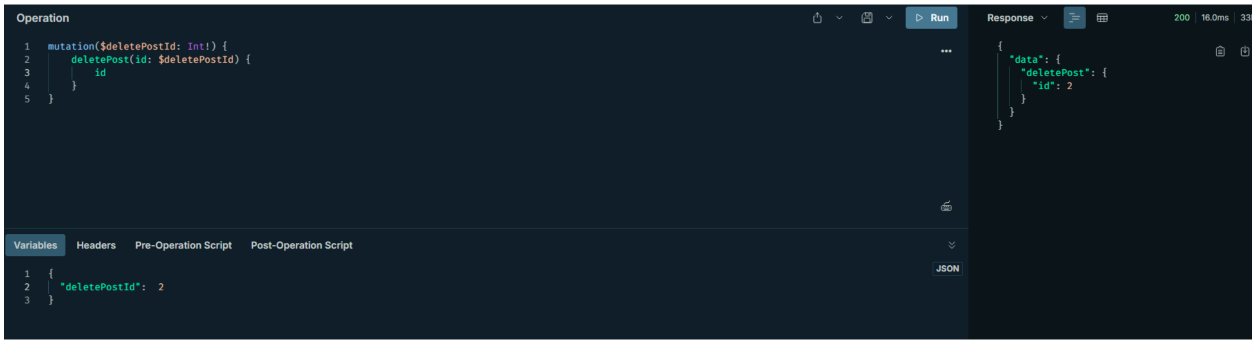


```graphql
query{
    posts {
        content
        id
        title
    }
}
```

```json
{
  "title": "test successful",
  "content": "successful test",
  "updatePostId":  4
}
```

```json
{
  "data": {
    "posts": [
      {
        "content": "test working",
        "id": 2,
        "title": "working test"
      }
    ]
  }
}
```

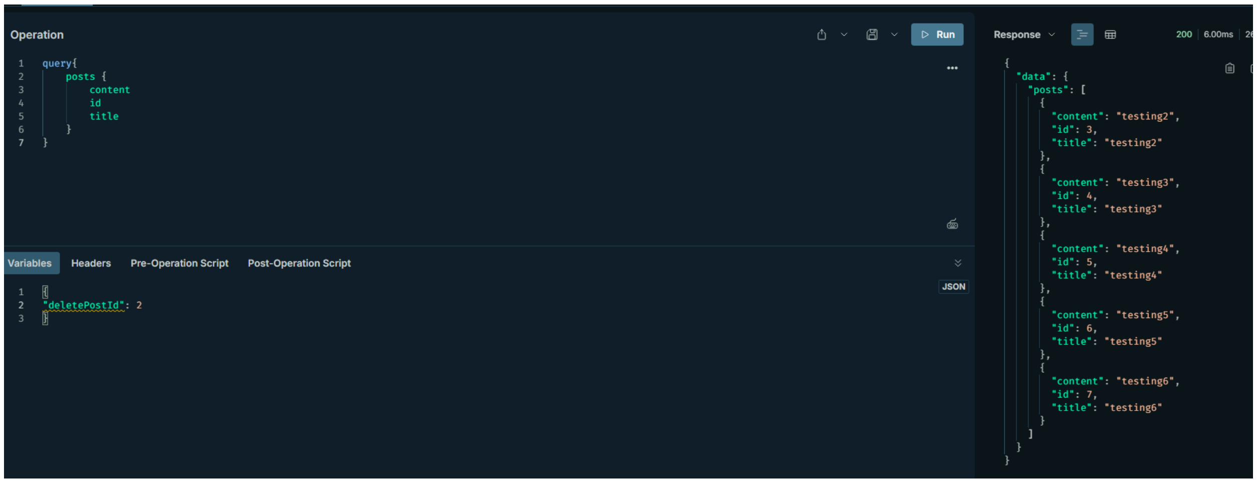
Delete



```graphql
mutation($deletePostId: Int!) {
    deletePost(id: $deletePostId) {
        id
    }
}
```

```json
{
  "deletePostId":  2
}
```

```json
{
  "data": {
    "deletePost": {
      "id": 2
    }
  }
}
```



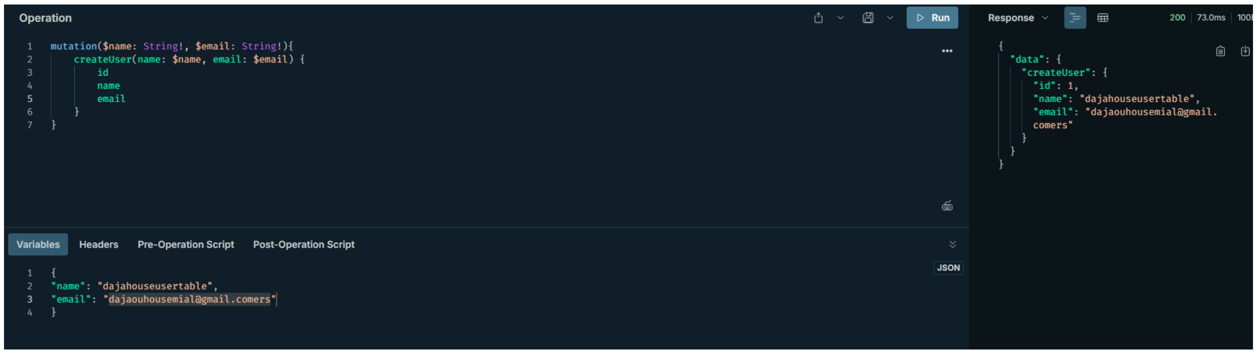
```
query{
    posts {
        content
        id
        title
    }
}
```

```
{
"deletePostId": 2
}
```

```
{
  "data": {
    "posts": [
      {
        "content": "testing2",
        "id": 3,
        "title": "testing2"
      },
      {
        "content": "testing3",
        "id": 4,
        "title": "testing3"
      },
      {
        "content": "testing4",
        "id": 5,
        "title": "testing4"
      },
      {
        "content": "testing5",
        "id": 6,
        "title": "testing5"
      },
      {
        "content": "testing6",
        "id": 7,
        "title": "testing6"
      }
    ]
  }
}
```

Users-table

Create

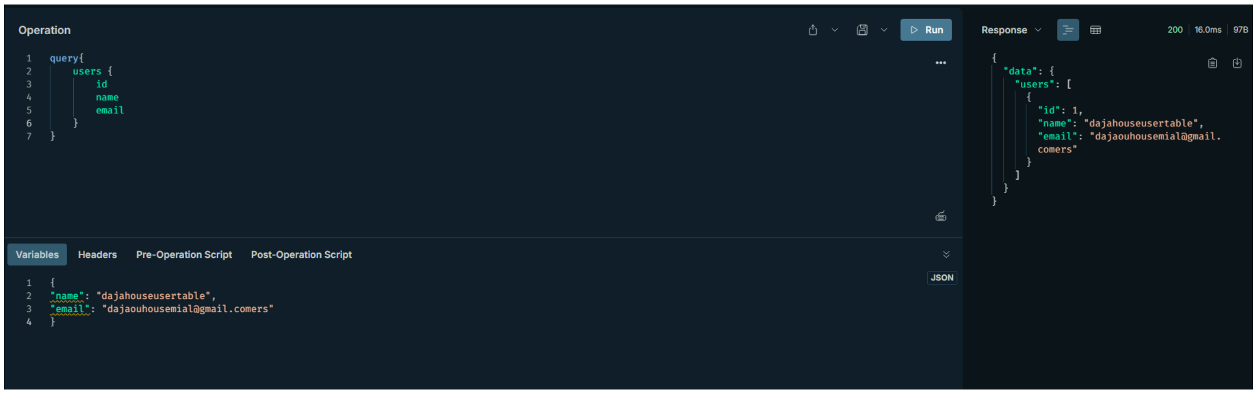


```
mutation($name: String!, $email: String!){
    createUser(name: $name, email: $email) {
        id
        name
        email
    }
}
```

```
{
"name": "dajahouseusertable",
"email": "dajaouhousemial@gmail.comers"
}
```

```
{
  "data": {
    "createUser": {
      "id": 1,
      "name": "dajahouseusertable",
      "email": "dajaouhousemial@gmail.
comers"
    }
  }
}
```

Read

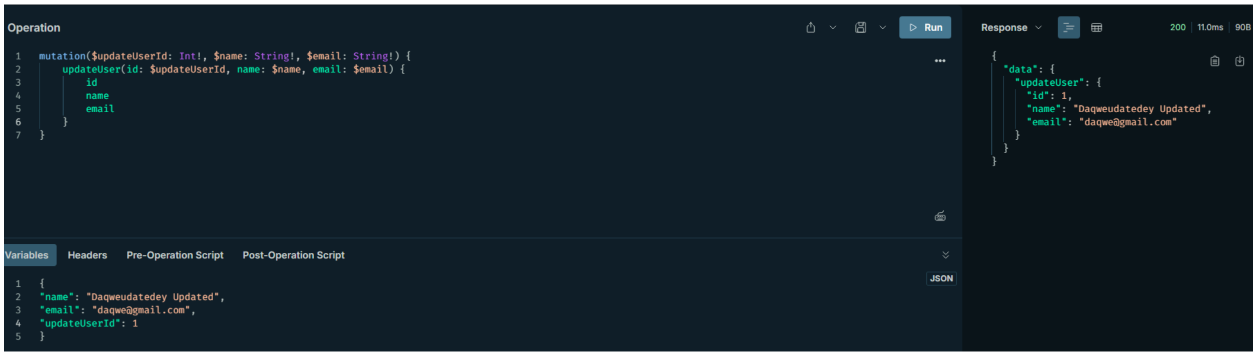


```
query{
    users {
        id
        name
        email
    }
}
```

```
{
"name": "dajahouseusertable",
"email": "dajaouhousemial@gmail.comers"
}
```

```
{
  "data": {
    "users": [
      {
        "id": 1,
        "name": "dajahouseusertable",
        "email": "dajaouhousemial@gmail.
comers"
      }
    ]
  }
}
```

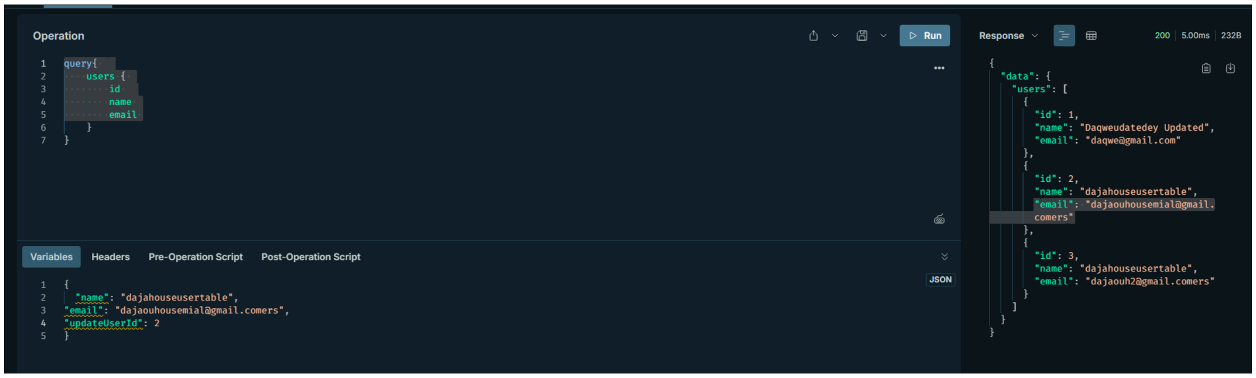
Update



```
mutation($updateUserId: Int!, $name: String!, $email: String!) {
    updateUser(id: $updateUserId, name: $name, email: $email) {
        id
        name
        email
    }
}
```

```
{
"name": "Daqweudatedey Updated",
"email": "daqwe@gmail.com",
"updateUserId": 1
}
```

```
{
  "data": {
    "updateUser": {
      "id": 1,
      "name": "Daqweudatedey Updated",
      "email": "daqwe@gmail.com"
    }
  }
}
```

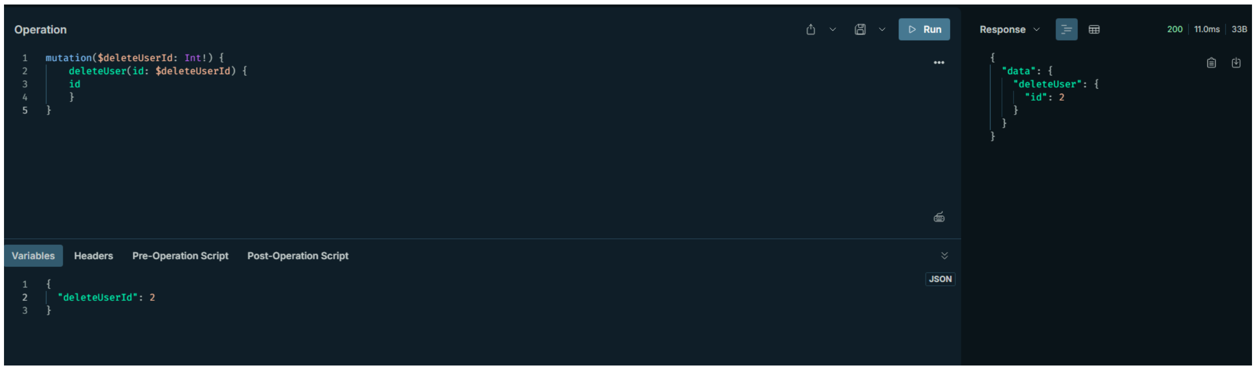


```
query{
    users {
        id
        name
        email
    }
}
```

```
{
  "name": "dajahouseusertable",
"email": "dajaouhousemial@gmail.comers",
"updateUserId": 2
}
```

```
{
  "data": {
    "users": [
      {
        "id": 1,
        "name": "Daqweudatedey Updated",
        "email": "daqwe@gmail.com"
      },
      {
        "id": 2,
        "name": "dajahouseusertable",
        "email": "dajaouhousemial@gmail.
comers"
      },
      {
        "id": 3,
        "name": "dajahouseusertable",
        "email": "dajaouh2@gmail.comers"
      }
    ]
  }
}
```

Delete



```
mutation($deleteUserId: Int!) {
    deleteUser(id: $deleteUserId) {
    id
    }
}
```

```
{
  "deleteUserId": 2
}
```

```
{
  "data": {
    "deleteUser": {
      "id": 2
    }
  }
}
```

What Do Database Migrations Do and Why Are They Useful?

From what I understand, database migrations work like a "snapshot" that captures the state of a database at a certain point in time. Each migration builds on the last one whenever there are changes or additions. They're useful because they keep track of the database's history, making it easy to go back to an earlier version if needed. It's kind of like how I can see MySQL and XAMPP working together—when I update something in MySQL, I can monitor those changes through XAMPP, which helps me manage and test my database more easily.

How Does GraphQL Differ from REST for CRUD Operations?

Based on discussions I've had with my friends while working on this output, GraphQL uses a single endpoint where clients can ask for just the data they need. This helps avoid problems like over-fetching or under-fetching. REST, on the other hand, usually has multiple endpoints with fixed responses, which can sometimes be inefficient. Another thing we talked about was how GraphQL's type system helps keep things clear between the client and server. Furthermore, in diving deeper into the iceberg of IT, in terms of making a modern web app, graphql definitely comes first compared to xampp because it's just a more flexible api, and just gives you what you need nothing more nothing less.