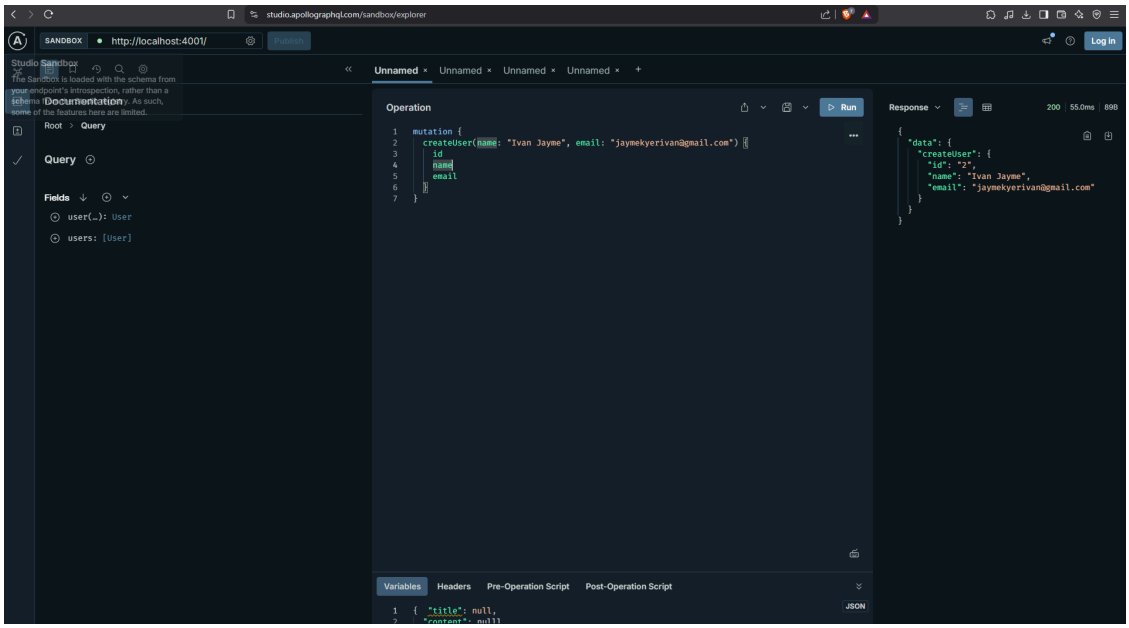
Building Microservices with Database Migrations and GraphQL CRUD Endpoints

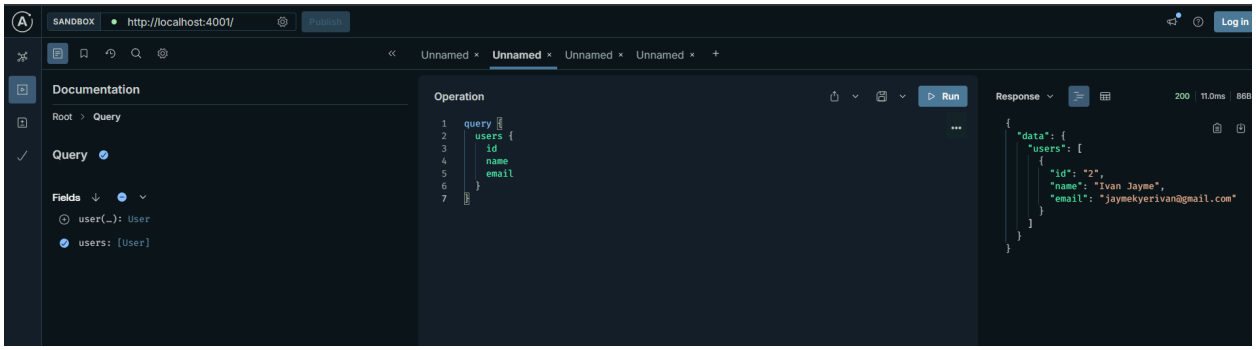
CREATE NEW USER

```
mutation {
  createUser(name: "Ivan Jayme", email: "jaymekyerivan@gmail.com") {
    id
    name
    email
  }
}
```



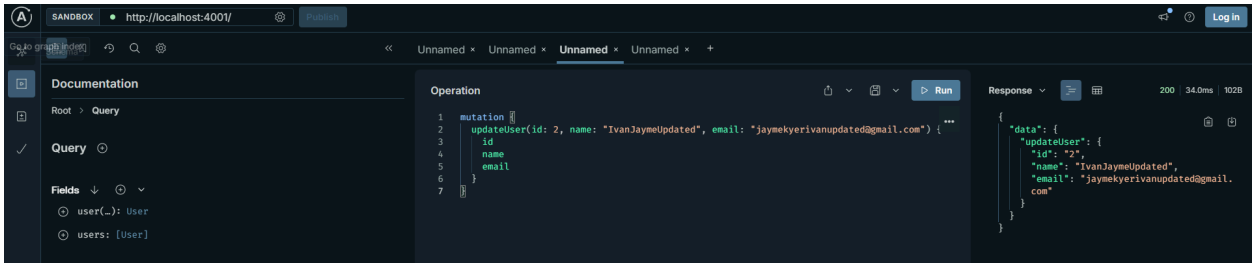
GET ALL USERS

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

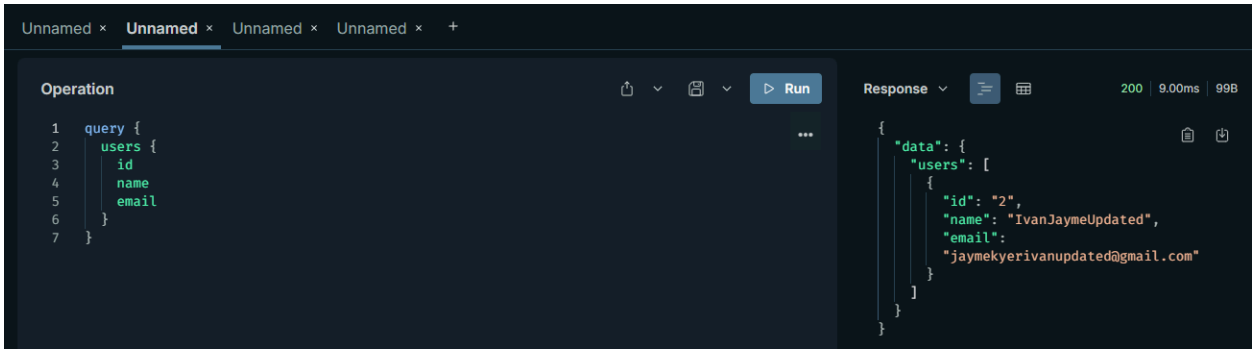


UPDATE USER

```
mutation {
  updateUser(id: 2, name: "IvanJaymeUpdated", email:
"jaymekyerivanupdated@gmail.com") {
    id
    name
    email
  }
}
```

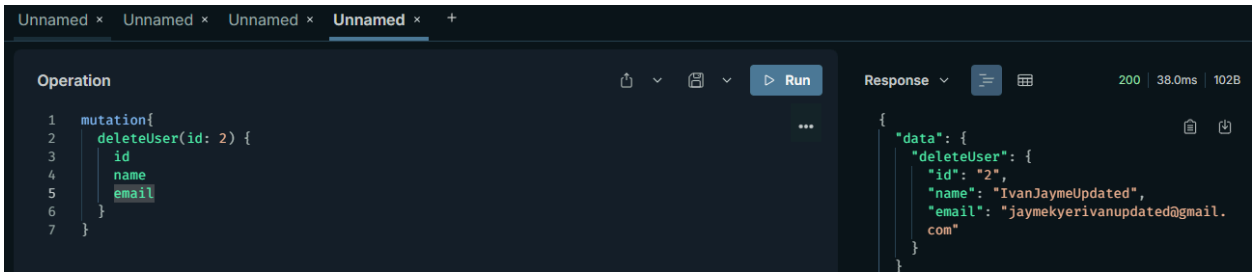


GETTING ALL USERS (UDPATED)

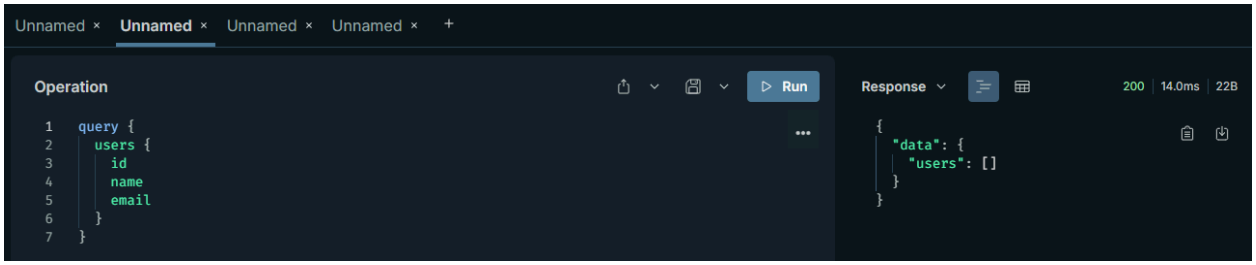


Deletion of user

```
mutation {
  updatePost(id: 1, title: "Jaymer's journey to heaven", content: "Did he make it?") {
    id
    title
    content
  }
}
```

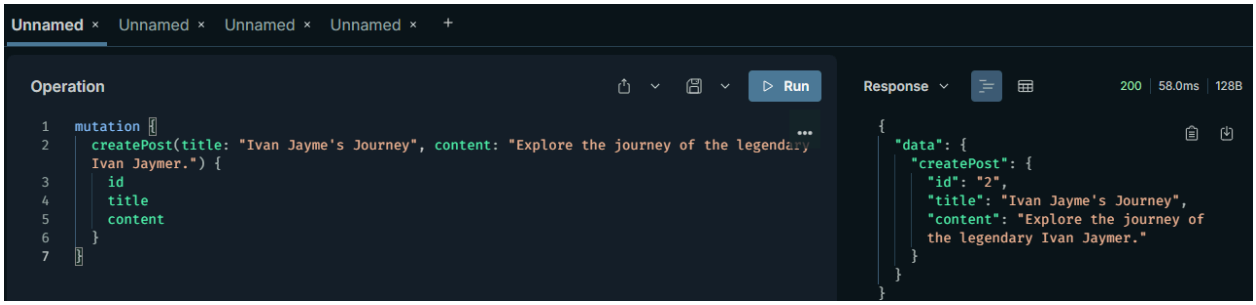


GETTING ALL USERS (DELETED)



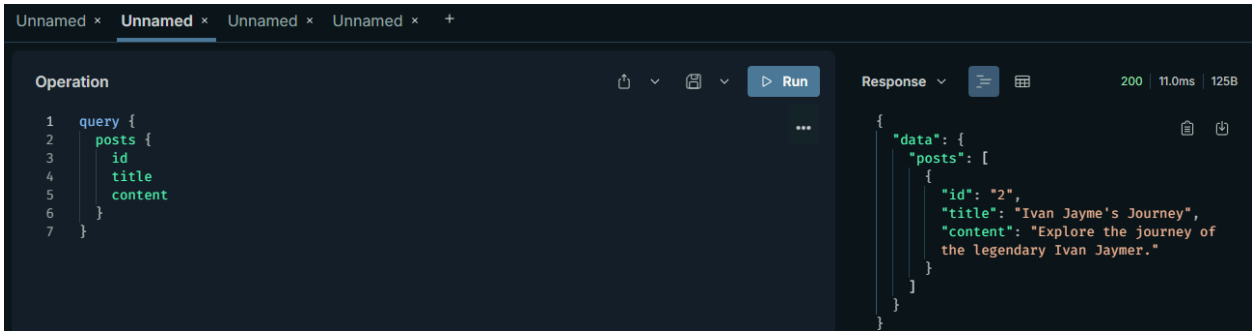
POST
Creating new post

```
mutation {
  createPost(title: "Ivan Jayme's Journey", content: "Explore the journey of the legendary Ivan Jaymer.") {
    id
    title
    content
  }
}
```



Getting all posts

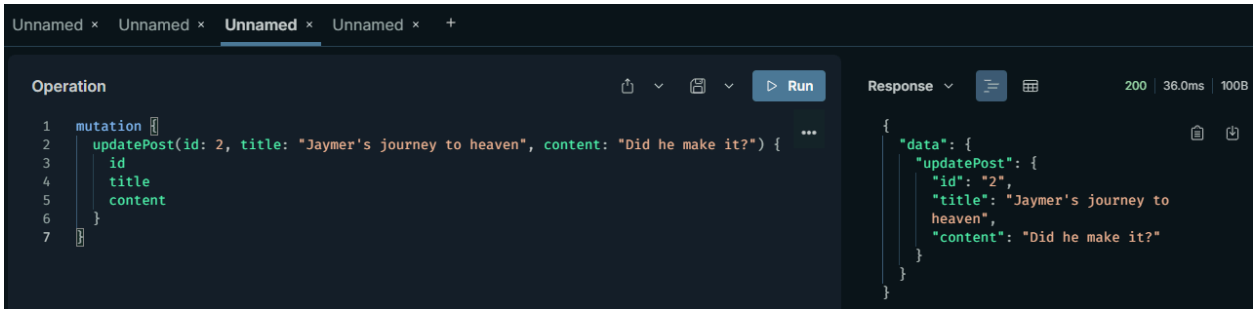
```
query {
  posts {
    id
    title
    content
  }
}
```



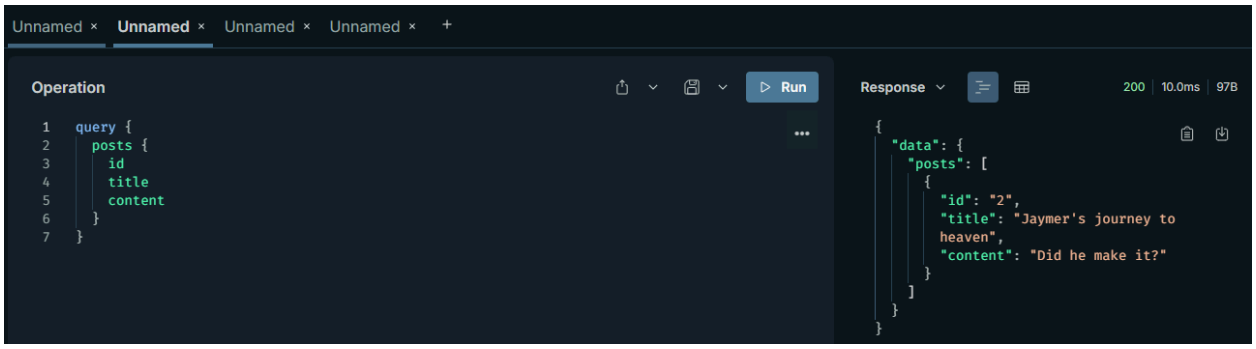
Update Post

```
mutation {
```

```
  updatePost(id: 2, title: "Jaymer's journey to heaven", content: "Did he
make it?") {
    id
    title
    content
  }
}
```

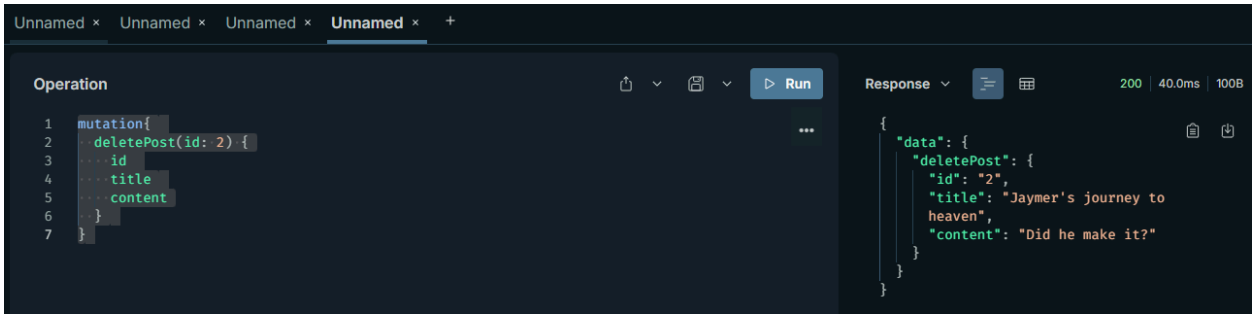


Getting all post (updated)

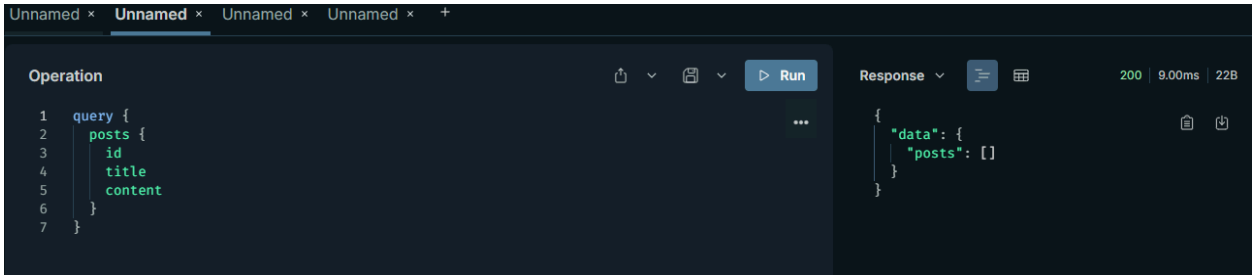


Deleting Post

```
mutation{
  deletePost(id: 2) {
    id
    title
    content
  }
}
```



UPDATED GET ALL POSTS



What do database migrations do and why are they useful?

I believe that database migrations are a way to keep track of changes in the database as time passes. Instead of just manually editing tables, migrations can help us apply, change, or rollback changes in a much more structured and version-controlled way. They're pretty useful because they prevent data loss when updating the database, it also allows teams to collaborate and make sure that they work with the same database.

How does GraphQL differ from REST for CRUD operations?

The rest API's are the get/users, post/users, put and delete. The problem with rest is that it over-fetches or under-fetches data. Like for example if i only need the user's name, the API returns name, email, password hash and all that information that i don't even need. But with graphQL it solves it by only using one endpoint instead of multiple routes we just get what we need.