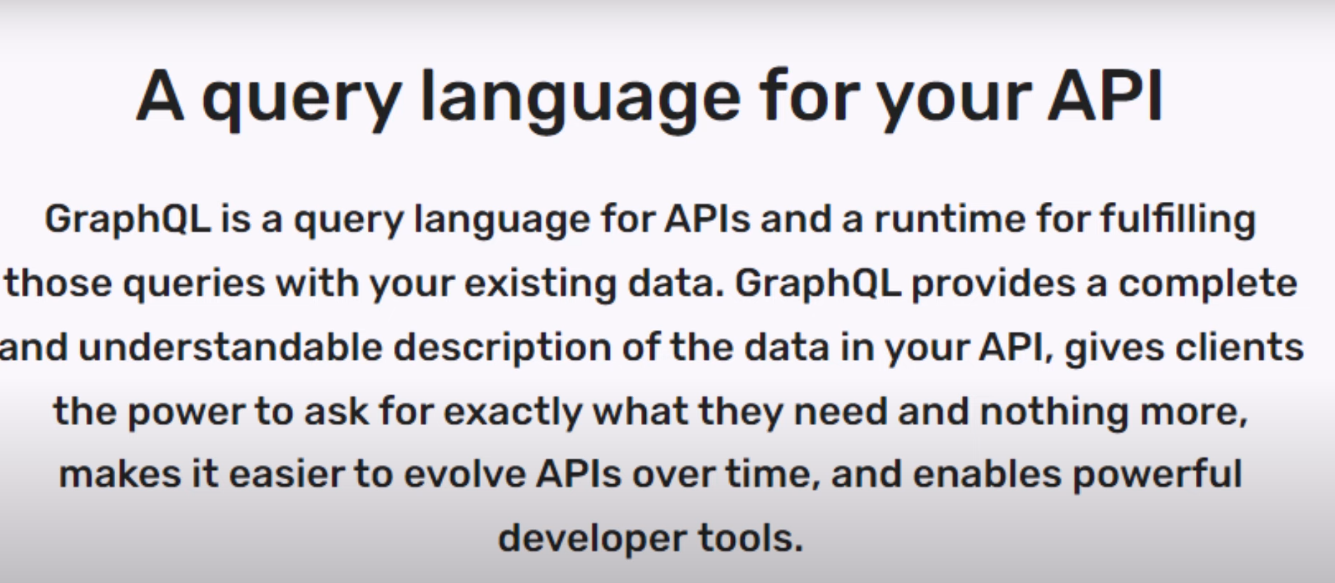
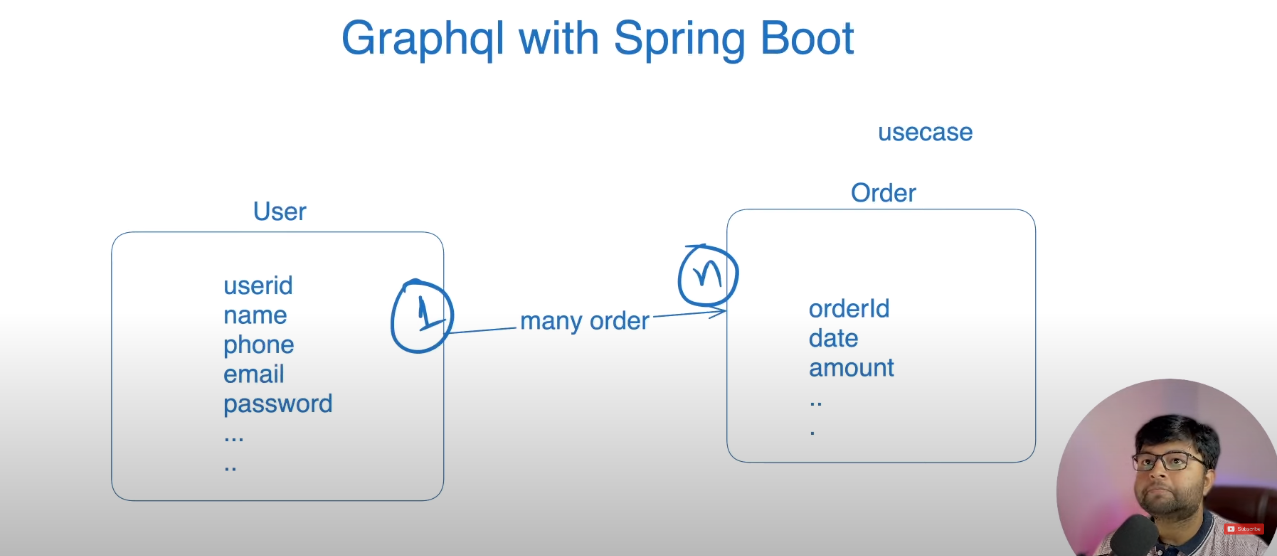
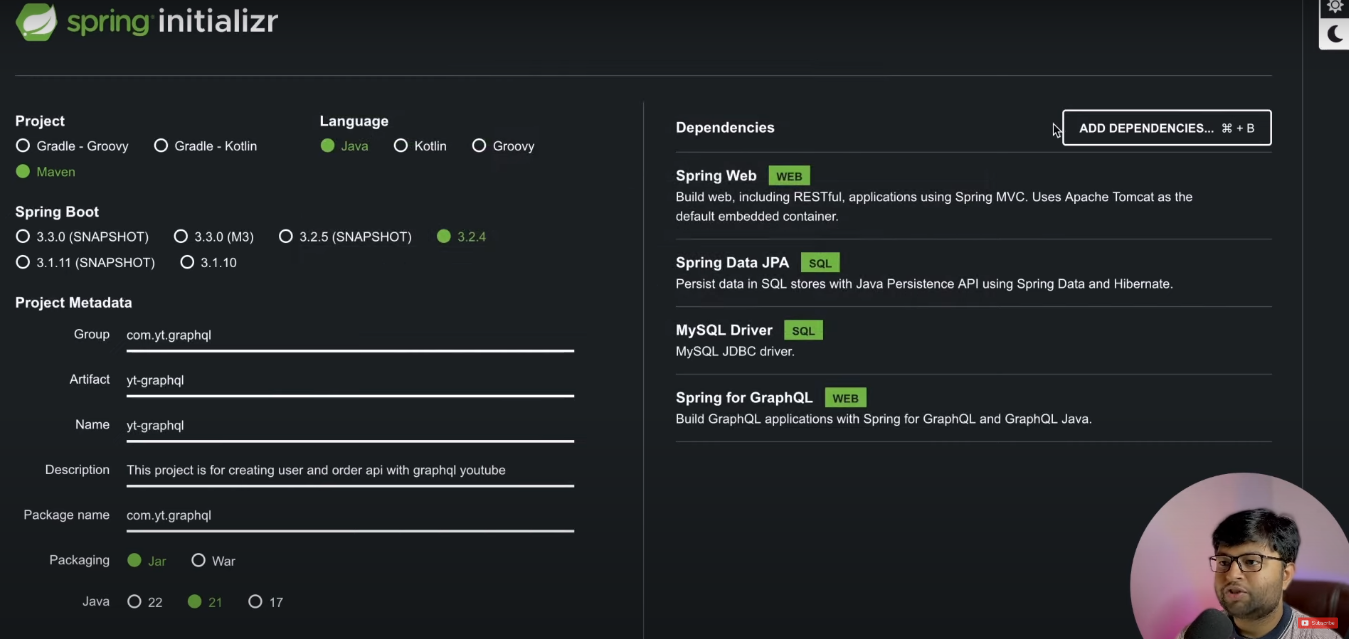
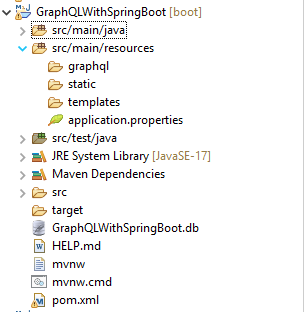
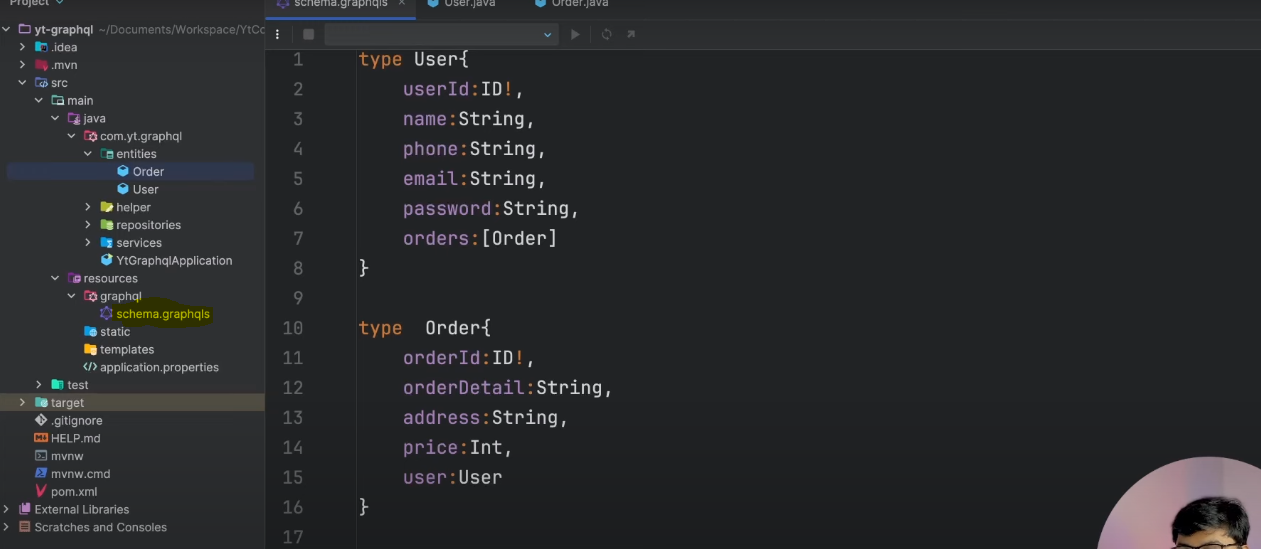
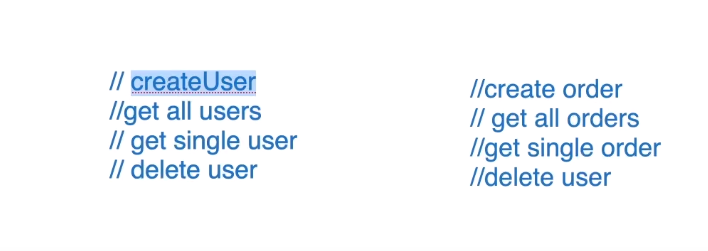
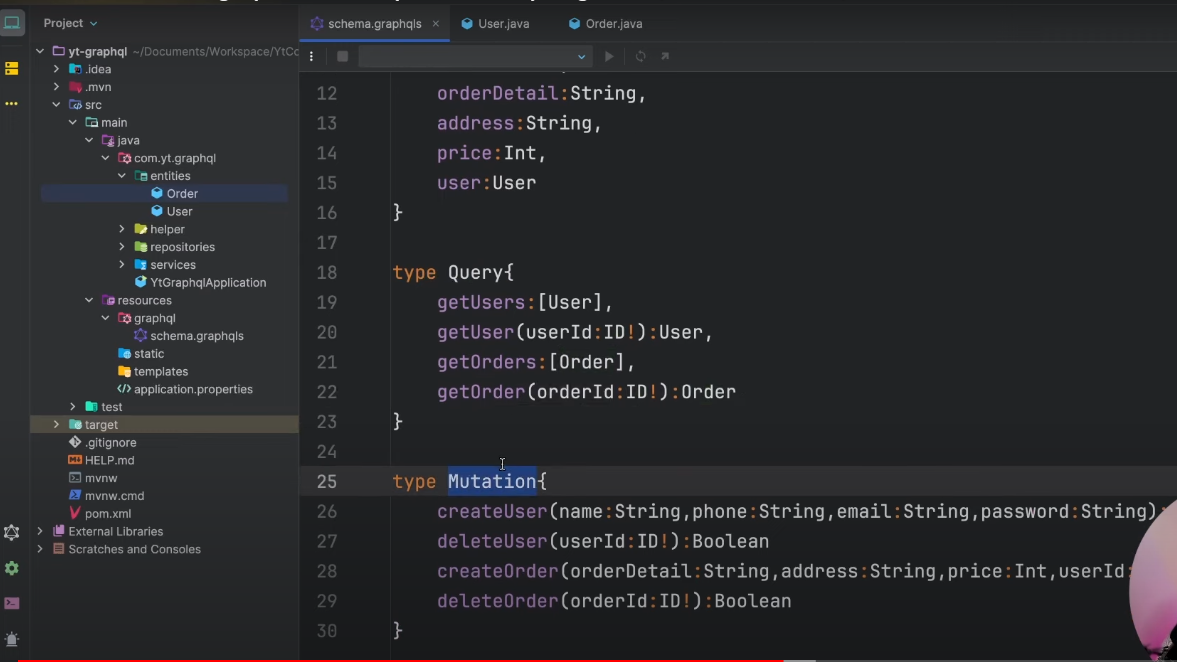
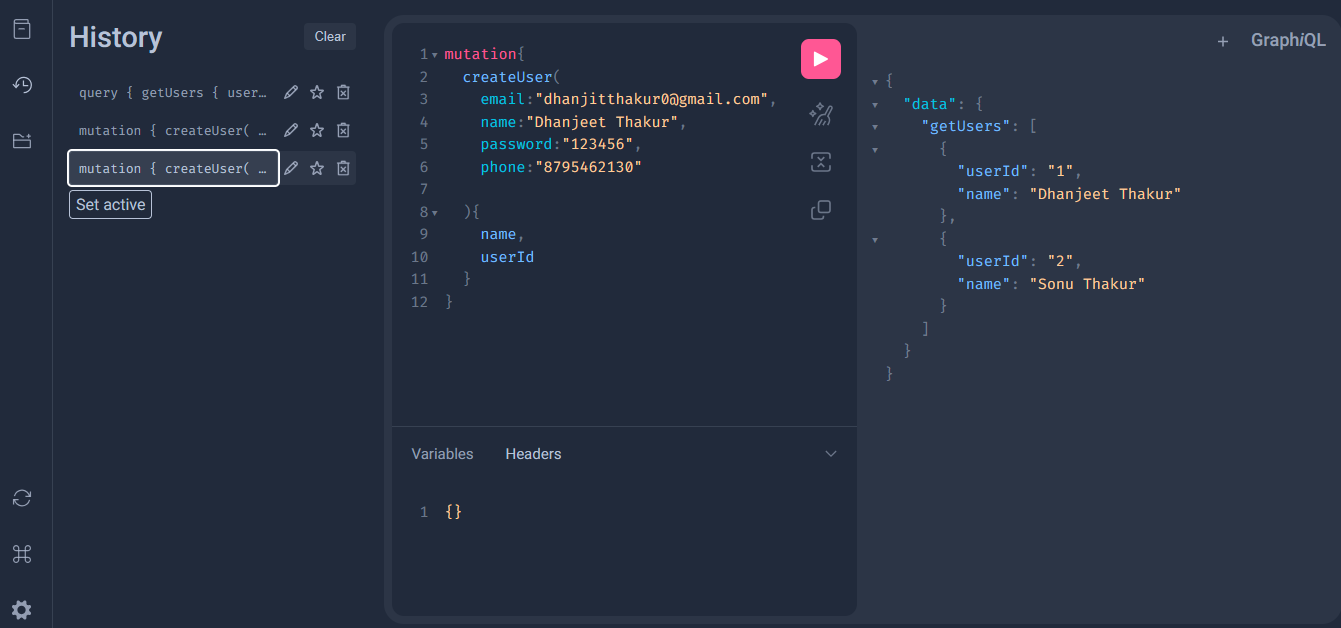
**GraphQL With Spring Boot**  
Q) What is GraphQL?

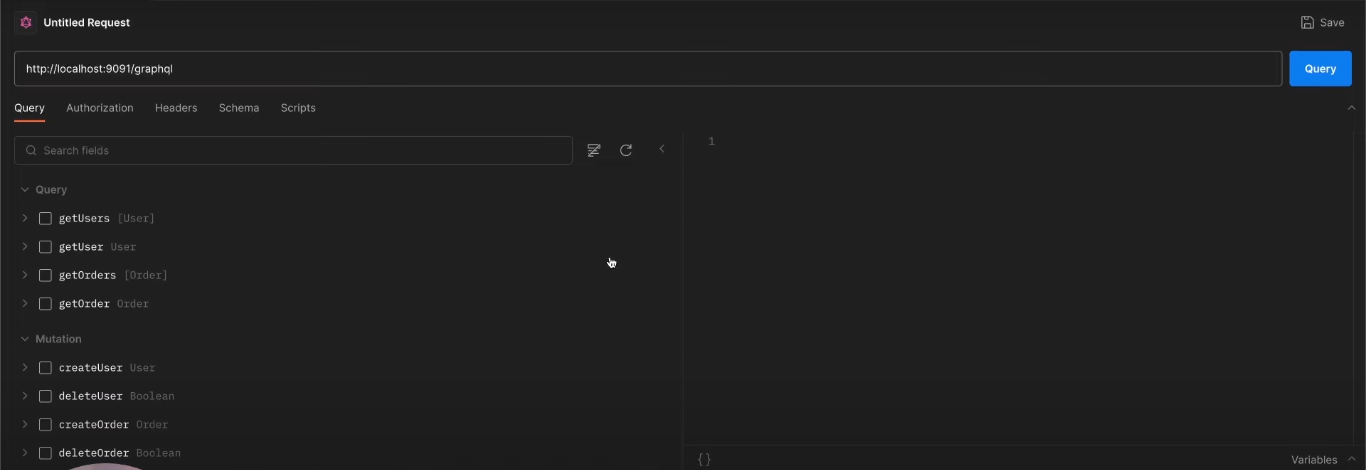
Let’s learn by a use case. In a shopping website one user can order many orders here it is oneToMany relationship between user and order table.  
  
Let’s create a project **“GraphQLWithSpringBoot”** the dependencies which we need is   
1. Spring Web 🡪 To create API.  
2. Spring Data JPA 🡪 For database connectivity.  
3. Spring for GraphQL 🡪 To get only desire data from database without any unwanted data.  
4. Sqlite-jdbc 🡪 To use SQLite3 Database.  
5. Hibernate-community-dialects 🡪 To execute the queries which generated by hibernate.  
  
  
  
**Project Structure:**  
Spring Boot code related to User and Order check in Github Repository.  
<https://github.com/DhanjeetKumarThakur/GraphQLWithSpringBoot>  
  
For now let’s understand graphQL. We created a file name “**schema.graphqls**” in “**src/main/resources/graphql**” Folder.

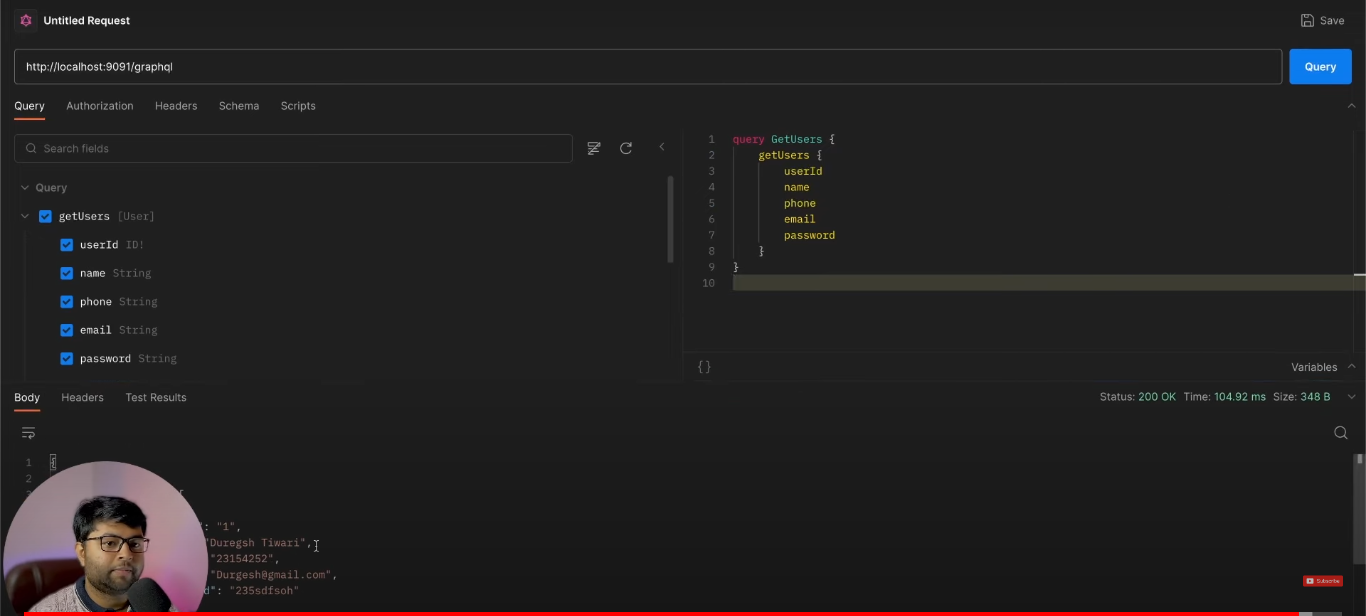
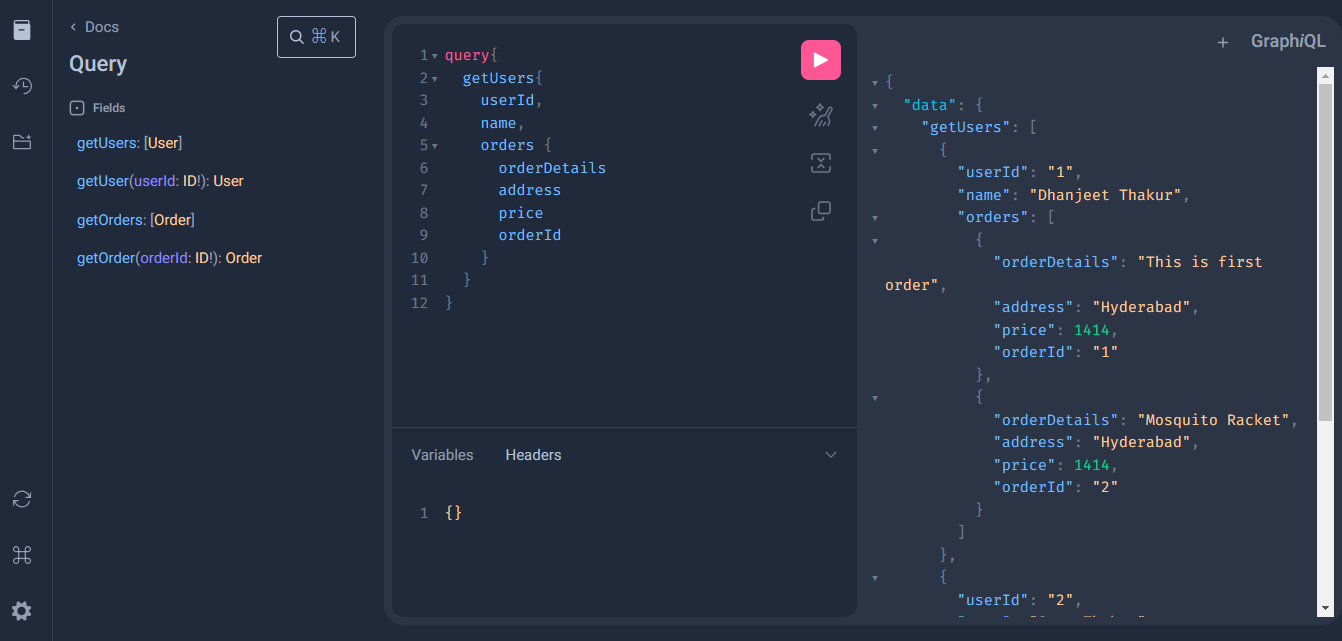
Let’s understand the graphql syntax, in the below pictures type is used for the Table follow by Class Name which represent Table (POJO class). The data types of graphql is

1. For primary Key 🡪 ID! => here ! is used to represent that it can’t be null  
2. String 🡪 For String type, Int 🡪 For Int   
3. [] 🡪 for group of Data and in the bracket the DATA\_TYPE eg. [Order]  
Note: Make sure the attribute name is same as class variable name.

The API’s which we will be creating is   
  
And for this in graphQL file we have to create two more types for fetching data we have to create **type Query** in graphql same file and for writing data we have to create **type Mutation**Now we will be mapping this query and Mutations to our controllers.  
  
Entire logic is written in the code check the code. Run the code and for testing qraphql we have something called graphiql it is a UI for firing graphql queries and mutations.  
First enable it from application.properties file  
  
#graphiql enable

spring.graphql.graphiql.enabled=true  
  
Then run the code and go to the browser and search   
<http://localhost:9091/graphiql>  
  
  
  
If you want to use postman then <http://localhost:9091/graphql>



Select from query section what ever query you want to run.  
  
  
  
  
You can have a customized query with only desired data.  
