<https://www.prisma.io/docs/get-started/03-build-graphql-servers-with-prisma-JAVASCRIPT-e001/>

1. npm install --save graphql-yoga

2. touch schema.graphql

Open schema.graphql and copy in the following

type Query {

publishedPosts: [Post!]!

post(postId: ID!): Post

postsByUser(userId: ID!): [Post!]!

}

type Mutation {

createUser(name: String!): User

createDraft(title: String!, userId: ID!): Post

publish(postId: ID!): Post

}

type User {

id: ID!

email: String

name: String!

posts: [Post!]!

}

type Post {

id: ID!

title: String!

published: Boolean!

author: User

}

3. Replace the index. Js

const { prisma } = require('./generated/prisma-client')

const { GraphQLServer } = require('graphql-yoga')

const resolvers = {

Query: {

publishedPosts(root, args, context) {

return context.prisma.posts({ where: { published: true } })

},

post(root, args, context) {

return context.prisma.post({ id: args.postId })

},

postsByUser(root, args, context) {

return context.prisma.user({

id: args.userId

}).posts()

}

},

Mutation: {

createDraft(root, args, context) {

return context.prisma.createPost(

{

title: args.title,

author: {

connect: { id: args.userId }

}

},

)

},

publish(root, args, context) {

return context.prisma.updatePost(

{

where: { id: args.postId },

data: { published: true },

},

)

},

createUser(root, args, context) {

return context.prisma.createUser(

{ name: args.name },

)

}

},

User: {

posts(root, args, context) {

return context.prisma.user({

id: root.id

}).posts()

}

},

Post: {

author(root, args, context) {

return context.prisma.post({

id: root.id

}).author()

}

}

}

const server = new GraphQLServer({

typeDefs: './schema.graphql',

resolvers,

context: {

prisma

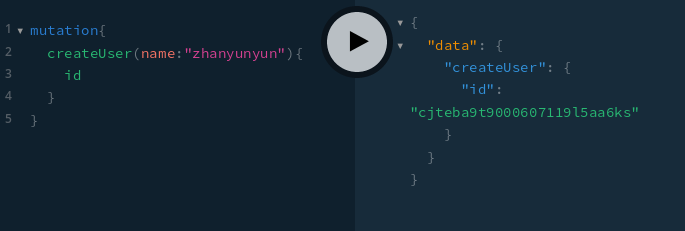
},

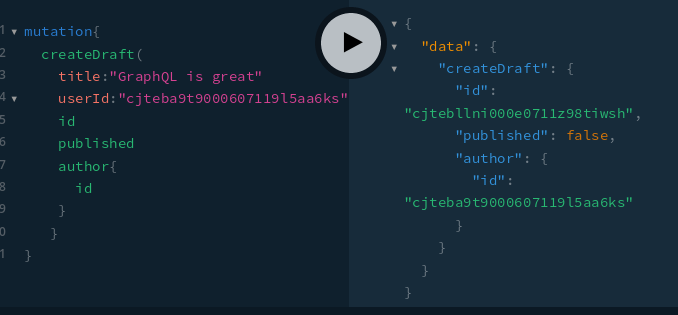
})

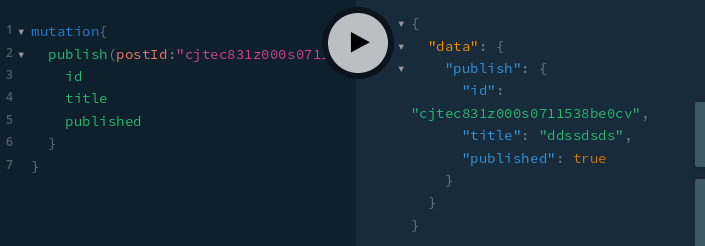
server.start(() => console.log('Server is running on <http://localhost:4000')>)

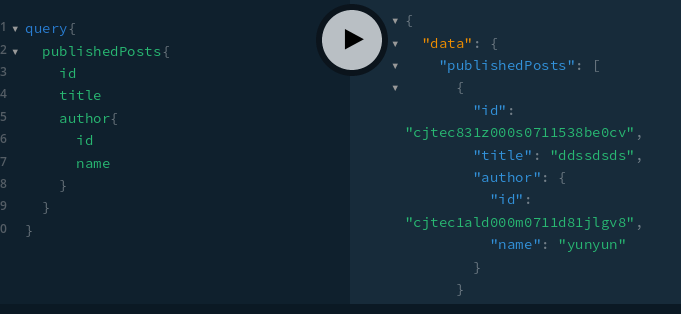
4. node index.js



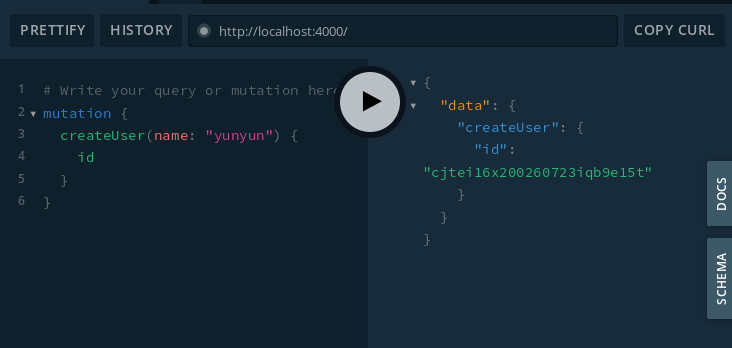




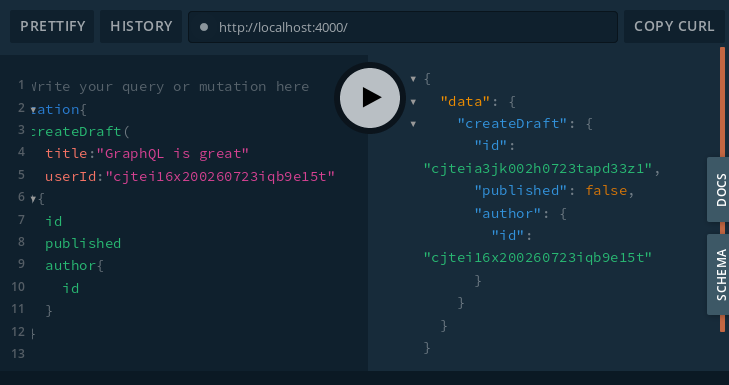




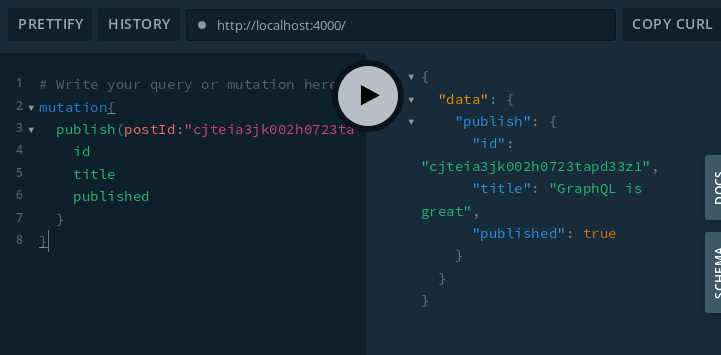
Create a new user



Create a new draft



Publish a post



Fetch published posts

