<https://www.prisma.io/docs/get-started/02-change-data-model-JAVASCRIPT-c001/>

1. Modify the datamodel. Prisma to fully cover with the following

type User {

id: ID! @unique

email: String @unique

name: String!

posts: [Post!]!

}

type Post {

id: ID! @unique

title: String!

published: Boolean! @default(value: "false")

author: User

}

2. Prisma deploy

3. Prisma generate

4. Open the file prisma.yml and append the following code to the end

hooks:

post-deploy:

- prisma generate

5. Modify this file: index.js, copy in the following: comment out the previous file, keep it

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

// A `main` function so that we can use async/await

async function main() {

// Create a new user with a new post

const newUser = await prisma

.createUser({

name: "Bob",

email: "bob@prisma.io",

posts: {

create: [{

title: "Join us for GraphQL Conf in 2019",

}, {

title: "Subscribe to GraphQL Weekly for GraphQL news",

}]

},

})

console.log(`Created new user: ${newUser.name} (ID: ${newUser.id})`)

// Read all users from the database and print them to the console

const allUsers = await prisma.users()

console.log(allUsers)

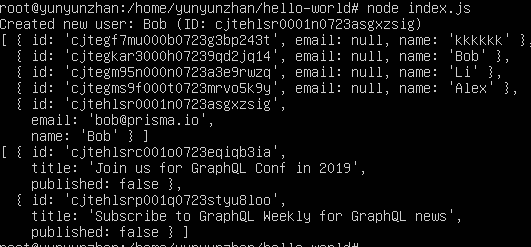
const allPosts = await prisma.posts()

console.log(allPosts)

}

main().catch(e => console.error(e))

6. node index.js



7. Modify the indes. Js file again

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

// A `main` function so that we can use async/await

async function main() {

// Read the previously created user from the database and print their posts to the console

const postsByUser = await prisma

.user({ email: "bob@prisma.io" })

.posts()

console.log(`All posts by that user: ${JSON.stringify(postsByUser)}`)

}

main().catch(e => console.error(e))

