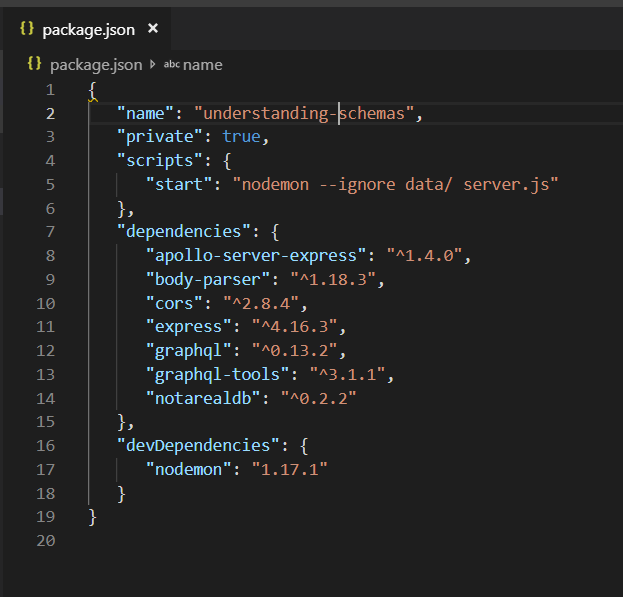
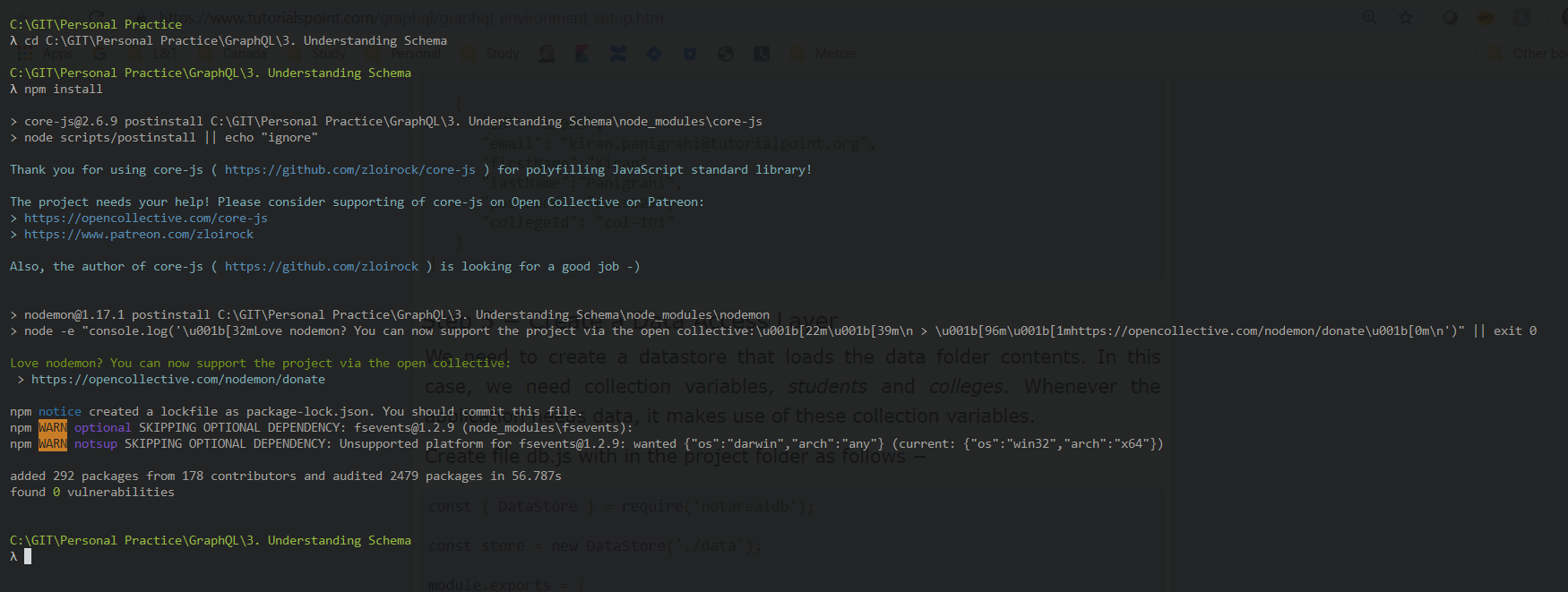
Let us understand how to **add new student record into the datastore** using a mutation query.

## Step 1 − Download and Install Required Dependencies for the Project

Create new package.json file with below code and install all dependencies.

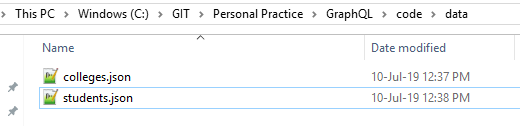


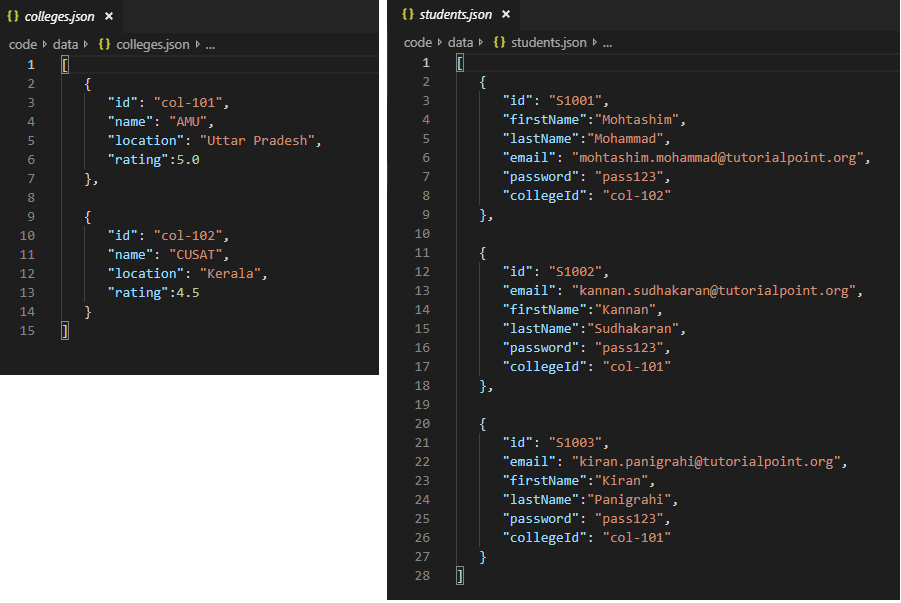


**Create Flat File Database in Data Folder**

we use flat files to store and retrieve data.

Create a folder data and add two files **students.json** and **colleges.json**.





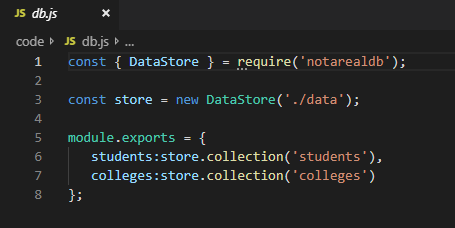
**Create a Data Access Layer**

We need to create a datastore that loads the data folder contents.

In this case, we need collection variables, *students* and *colleges*.

Whenever the application needs data, it makes use of these collection variables.

Create file db.js with in the project folder as follows

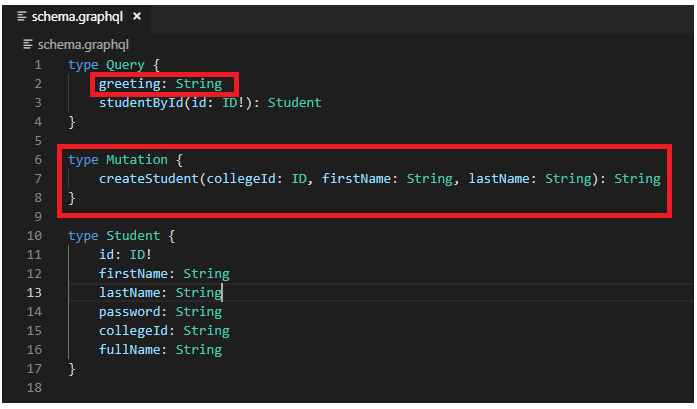


## Step 2 − Create a Schema

Add **schema.graphql** file with following code.

Note that the function createStudent returns a String type.

This is a unique identifier (ID) which is generated after creating a student.



## Step 3 − Create Resolver

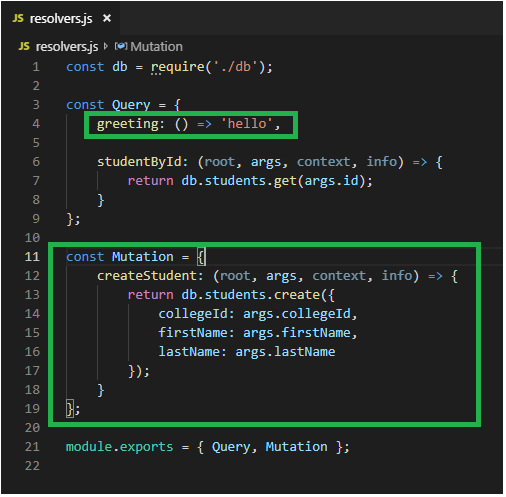
Create a file **resolvers.js** with following code.

The mutation function points to students collection in the datastore.

To add a new student, invoke the create method in students collection.

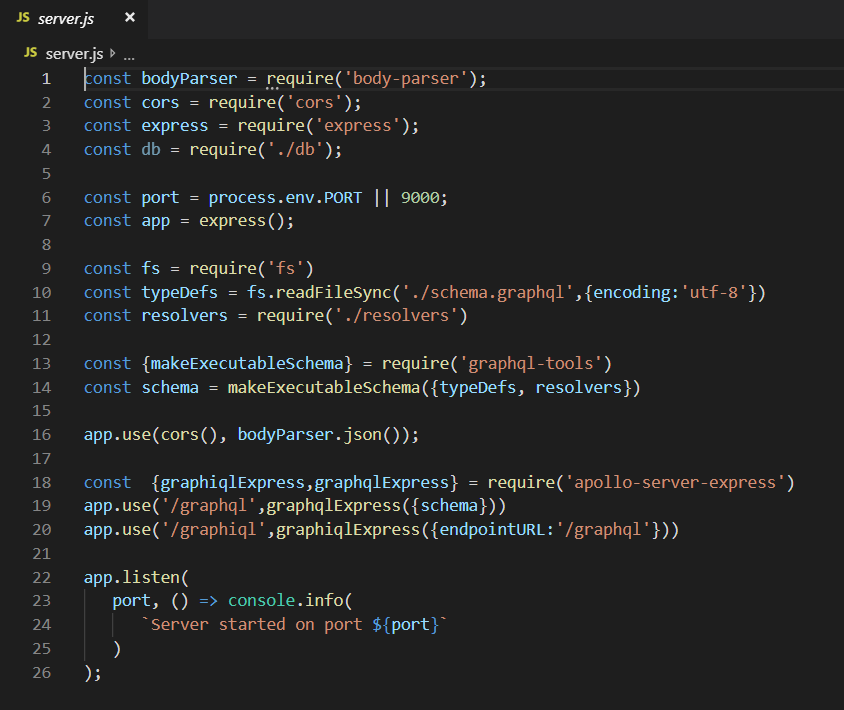
The args object will contain the parameters which are passed in the query.

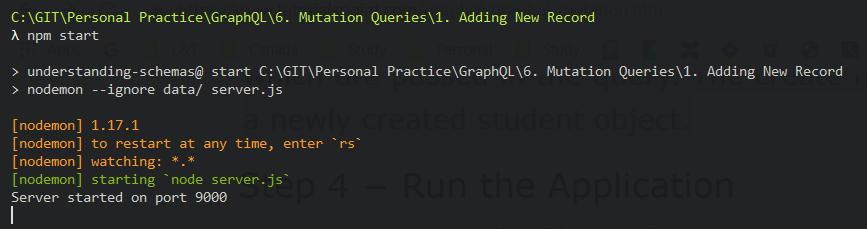
The create method of students collection will return the id of a newly created student object.



## Step 4 − Run the Application

Create a server.js file.





Next step is to open browser and type the URL [**http://localhost:9000/graphiql**](http://localhost:9000/graphiql).

## Testing

Type the following query in the editor –

//college Id should be matched with data from colleges.json for easy retrieval

mutation {

createStudent(collegeId:"col-2",firstName:"Tim",lastName:"George")

}

The above query will create a student object in student.json file.

The query will return a unique identifier. The response of the query is as shown below

{

"data": {

"createStudent": "SkzmtKq-B"

}

}

## On Browser



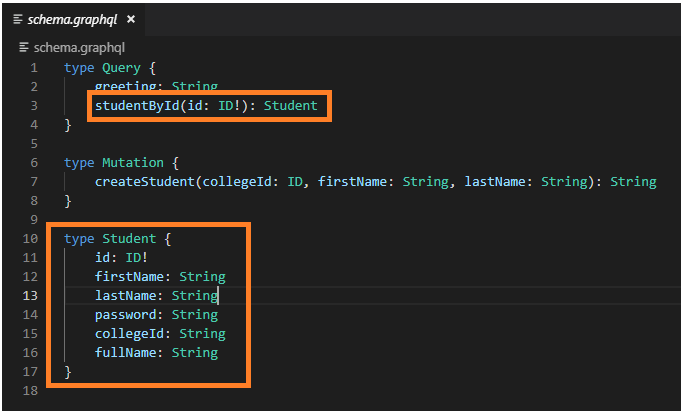
## Updation on Student.json



# verify student object

To verify if the student object is created, we can use the studentById query.

To use studentById query, edit the **schema.graphql** as given below



Edit the **resolver.js** file as given below



## Testing

Given below is the query to get student by unique id returned from the mutation query

{

studentById(id:"SkzmtKq-B") {

id

firstName

lastName

}

}

The response from the server is as follows

{

"data": {

"studentById": {

"id": "SkzmtKq-B",

"firstName": "Tim",

"lastName":"George"

}

}

}

## On Browser

