

## README file

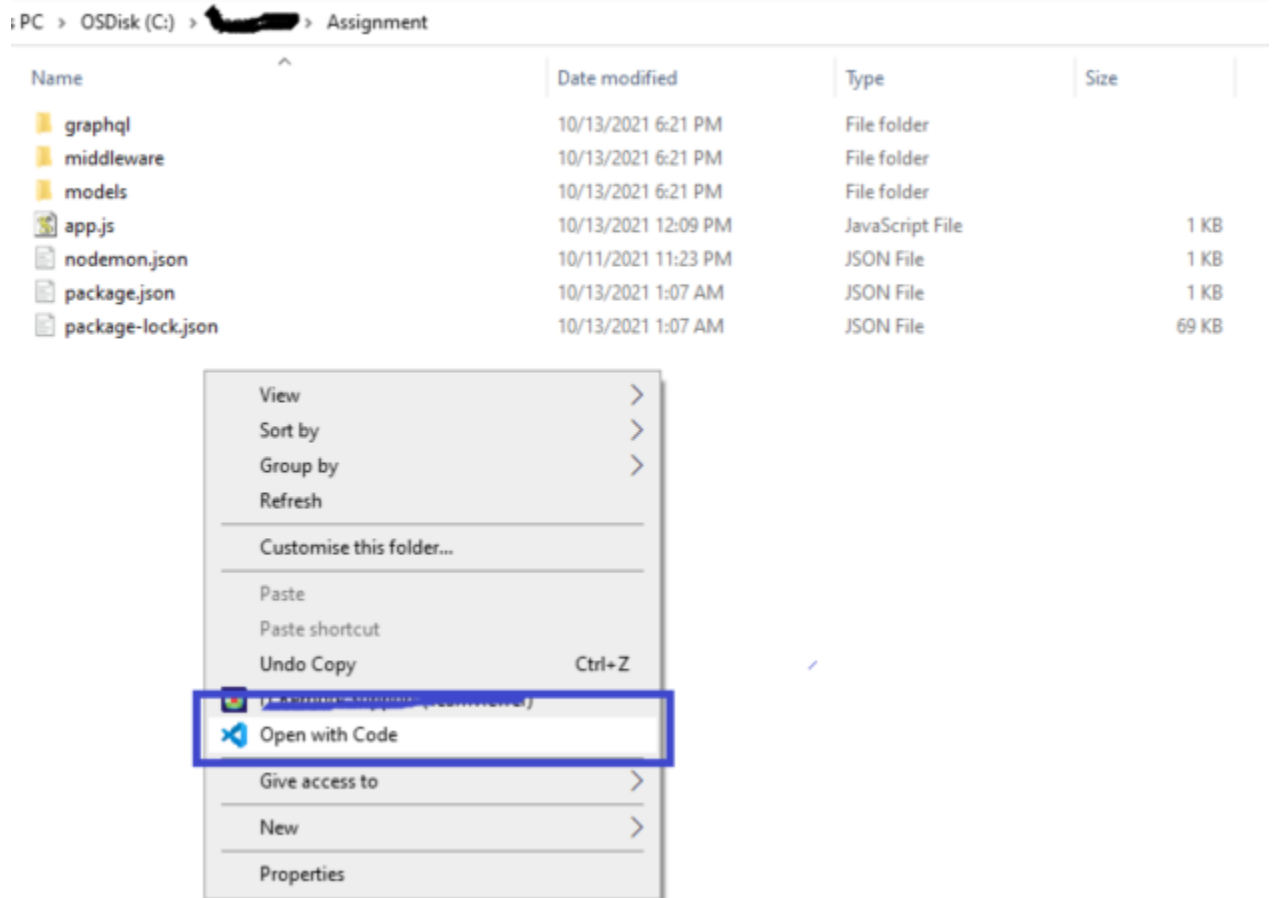
### 1. How to get the project?

Clone/download it from below git repository  
Url –

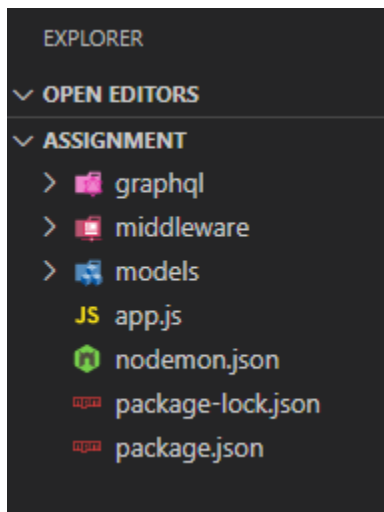
### 2. How to run the project?

Use Visual Studio code to open this project.

- Once the project is cloned or downloaded (zip file will be downloaded in this case. Unzip it.)
- Go inside **Assignment** folder & right click and select – **Open with code**, like below –



### 3. Its folder structure will look like this -

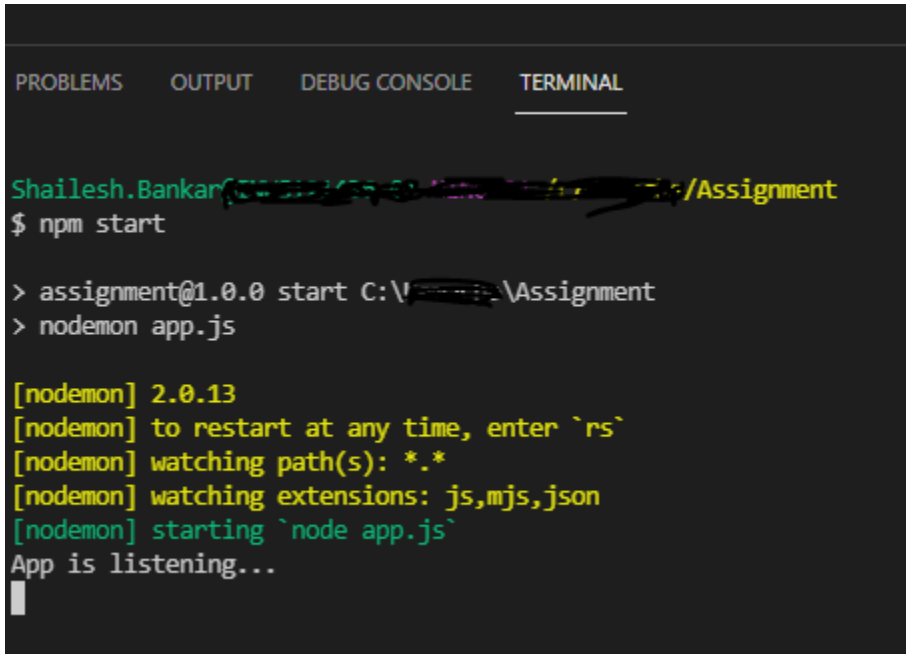


4. Open terminal of Visual studio run command –

**npm install**

It will install all the required packages for this project & will create a **node\_modules** folder

5. Once it is installed, run **npm start** command in terminal, it will start the app & below message will be seen in the terminal –



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Shailesh.Bankar@C:\Users\Shailesh\Documents\Assignment
$ npm start

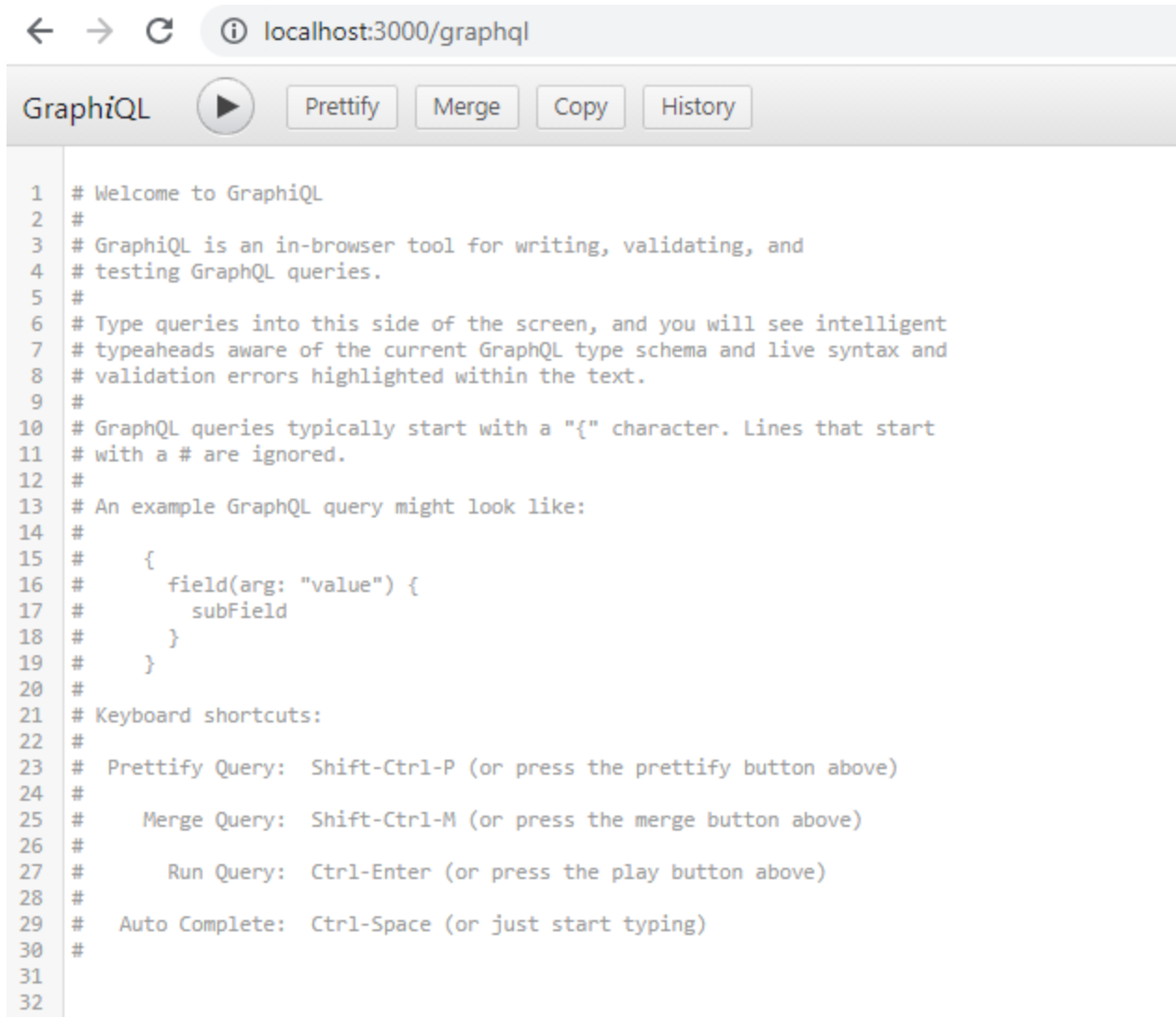
> assignment@1.0.0 start C:\Users\Shailesh\Documents\Assignment
> nodemon app.js

[nodemon] 2.0.13
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
App is listening...
█
```

6. Now, go to chrome browser and hit below url –

<http://localhost:3000/graphql>

It will show the UI provided by GraphQL as below –



7. Now the app is running.

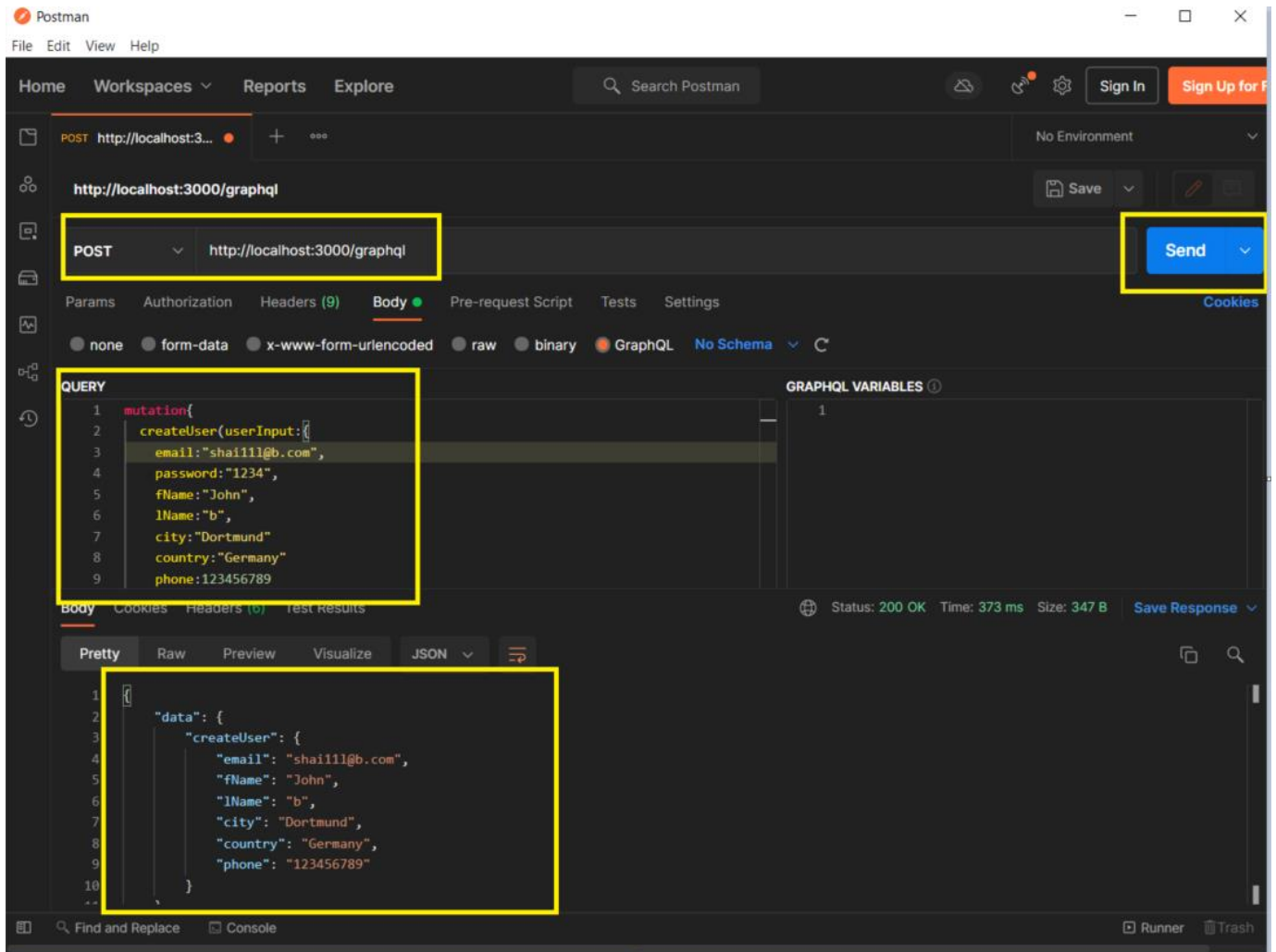
To test it Please use **Postman** tool

8. Pls See belwo image, select request type as **POST**, enter url in the adgesent box as –

<http://localhost:3000/graphql>

Use **QUERY** field to write the query to hit the server as shown image below

Click the **Send** button and the response will be shown in the **Body** section (below QUERY section)



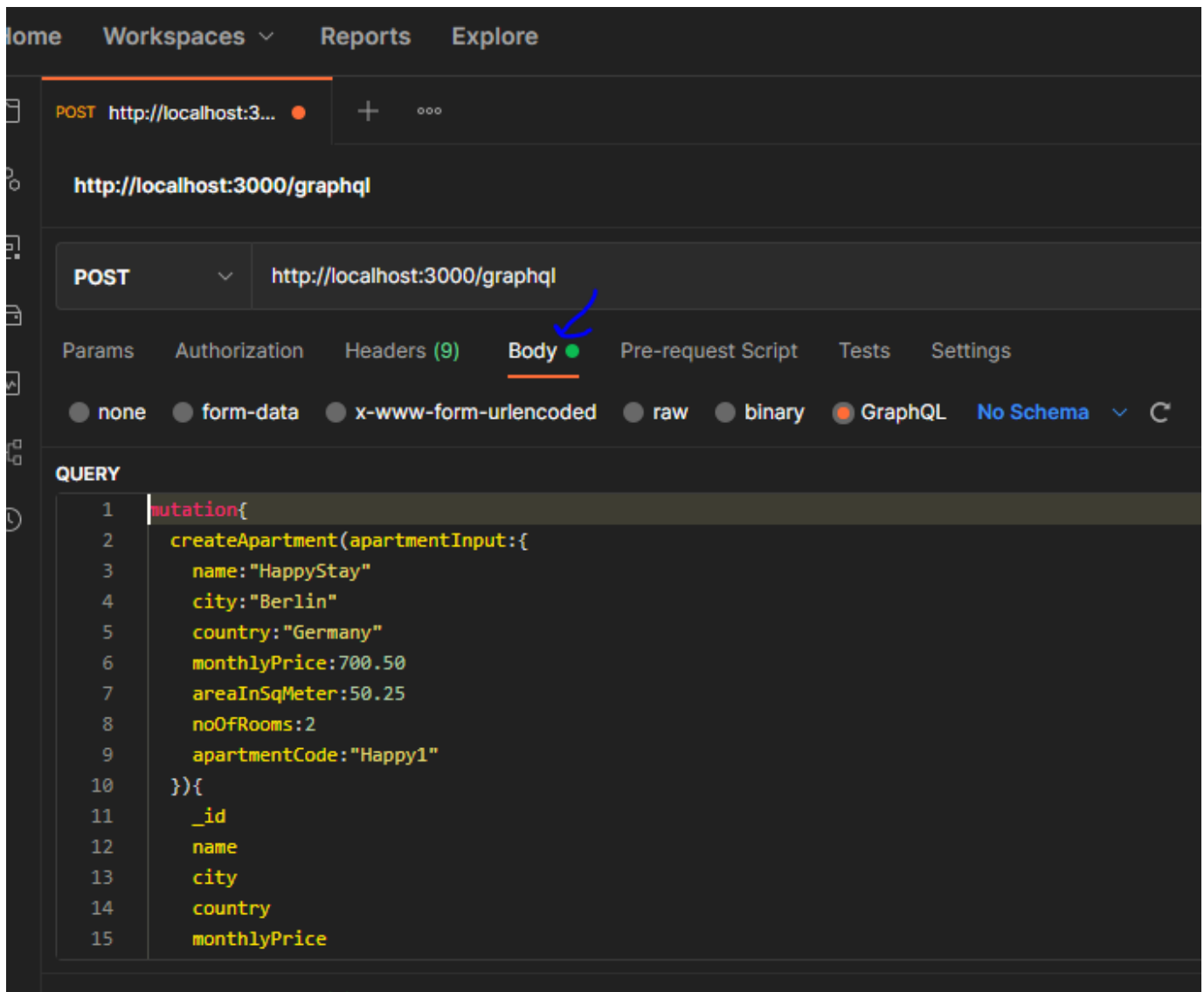
Please use below queries for particular operations –  
(I request you to kindly follow the sequence as I have given below)

Sequence is like –

User registration → User login → Creating apartment (user needs to be logged in) → Listing apartments  
→ Creating favorite apartments (user needs to be logged in) → Listing favorite apartments (user needs to be logged in)

User can see all the apartments, search without login

Use below query for use Body section of Postman to make queries, like -



## User Registration –

```
mutation{
  createUser(userInput:{
    email:"shail@b.com",
    password:"1234",
    fName:"John",
    lName:"b",
    city:"Dortmund"
    country:"Germany"
    phone:123456789
  }){
    email
    fName
    lName
    city
    country
    phone
  }
}
```

## User Login –

```
query{
  login(email:"shail@b.com", password:"1234"){
    token
    userId
  }
}
```

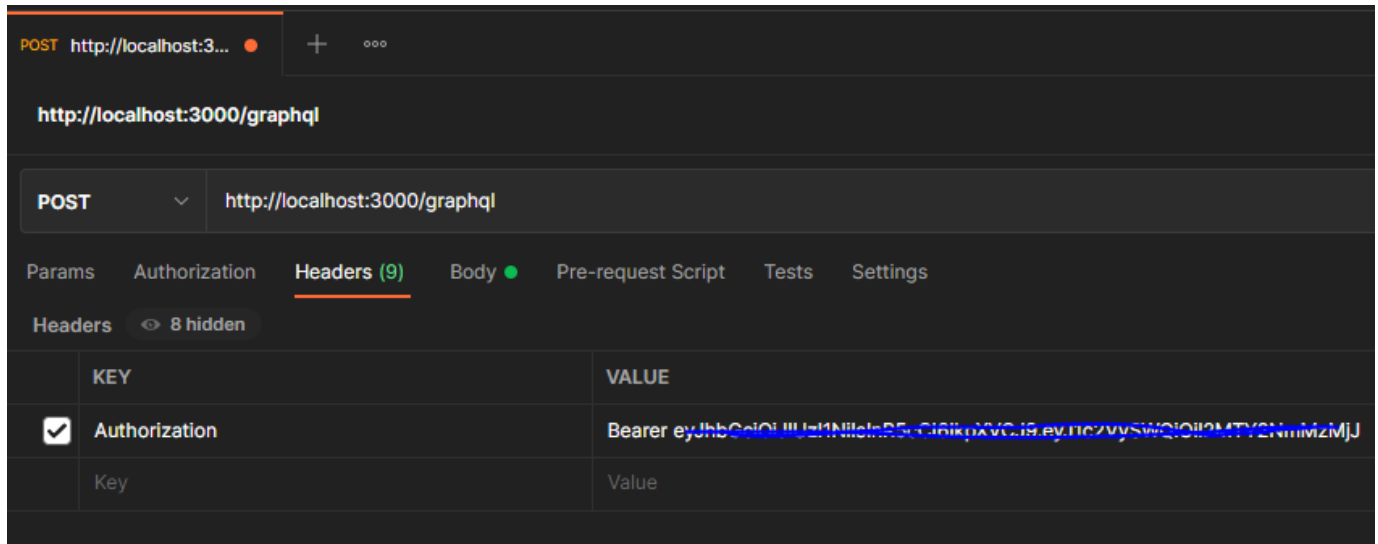
Once the user is logged in, it will return a jwt token – please save it to perform operations such as – create apartment, mark apartment as favorite etc

### Create Apartment-

To create apartment, use the token returned by login api in above step

Use it as below in Postman -

**Bearer<space>token** like -



And then write below query in **Body** section, hit **Send** button.

```
mutation{
  createApartment(apartmentInput:{
    name:"HappyStay"
    city:"Berlin"
    country:"Germany"
    monthlyPrice:700.50
    areaInSqMeter:50.25
    noOfRooms:2
    apartmentCode:"Happy1"
  }){
    _id
    name
    city
    country
    monthlyPrice
    areaInSqMeter
    noOfRooms
    creator{
      email
      password
      fName
      lName
      city
      country
      phone
    }
  }
}
```



## List Apartments

(no login required)

```
query{
  apartment{
    _id
    name
    city
    country
    monthlyPrice
    areaInSqMeter
    noOfRooms
    apartmentCode
    creator{
      _id
      email
      fName
      lName
      city
      country
      phone
    }
  }
}
```

Note - Please note down a couple of **apartments \_id** to create favorite apartments.

## Create Favourite Apartment

(User has to login - pass the Bearer token as Authorization in header)

To mark apartment as favourite, you must pass the created apartment's id which can be retrieved from above (list apartments) query

For example, I passed below Id: 61667f30854c48008d3ffcc8 to apartmentId parameter

```
mutation{
  createFavoriteApartment(apartmentId:"61667f30854c48008d3ffcc8"){
    _id
    apartment{
      _id
      name
      city
      country
      monthlyPrice
      areaInSqMeter
      noOfRooms
      apartmentCode
      creator{
        _id
        email
        fName
        lName
        city
        country
        phone
      }
    }
  }
}
```

## List favourite apartments

(Login needed)

```
query{
  favoriteApartment
  {
    _id
    apartment{
      name
      city
      country
      monthlyPrice
      areaInSqMeter
      noOfRooms
      apartmentCode
    }
    users{
      email
      fName
      lName
      city
      country
      phone
    }
  }
}
```

### Search Apartment:

(no login needed)

It will ask search by value & value to search -

Value of **searchBy** parameter could be one of the following & it is case sensitive -

- city
- country
- noOfRooms

e.g. -

```
query{
  searchApartments(searchBy:"country",searchValue:"Germany"){
    _id
    name
    city
    country
    monthlyPrice
    areaInSqMeter
    noOfRooms
    apartmentCode
  }
}
```

-----End of Document-----