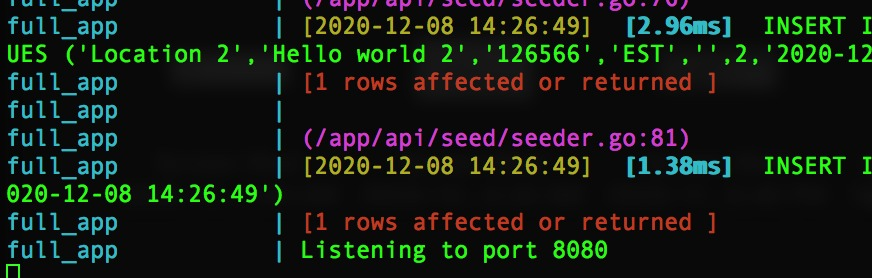
GoMap2020 User Manual

Prerequisites:  
1. Docker  
2. GIT

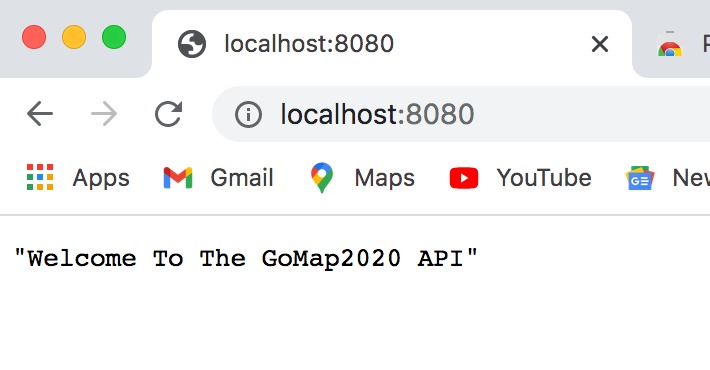
To start the App please follow below steps :

1. Go to command prompt
2. git clone <https://github.com/diptyojha/goLngFirstProject/>
3. go to the project folder and run below command

docker-compose up –build

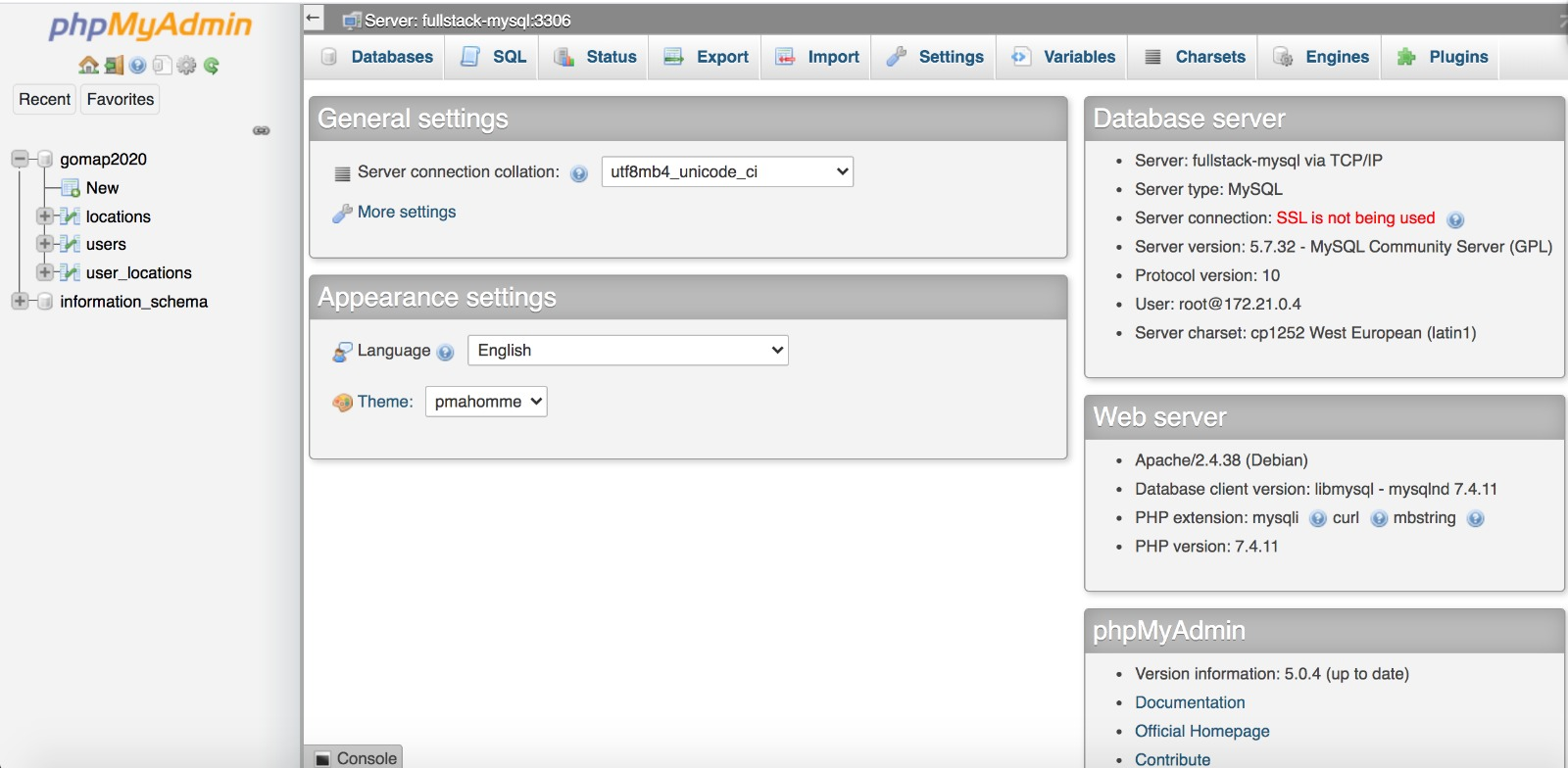


1. Once the containers are up you will get message as shown in above image
2. Now open http:localhost:8080 in your browser, you will get a welcome message



1. Now you can open below URL to view the Mysql browser client and see 3 tables already created. Also dummy data is seeded already.

URL: http://localhost:9090/

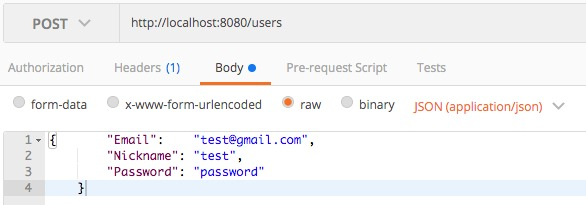


1. Now the app is running Smoothly in your machine

Time to test

Please follow below steps to test the APIs

* **Steps 1 : Create New User (Password here is not hidden , so that can show its hashed and stored in DB)**



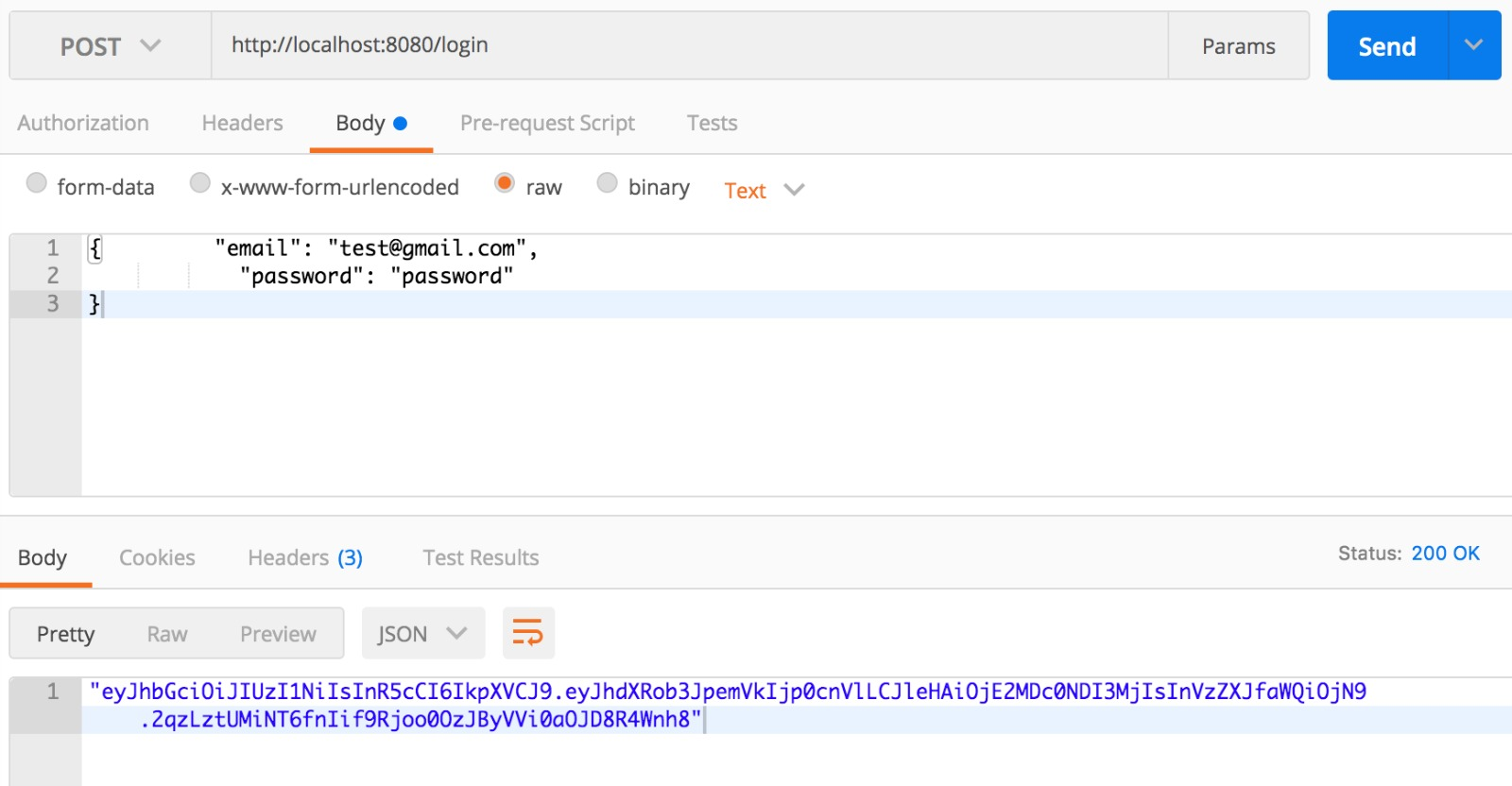
Graphical user interface, text, application

Description automatically generated

**Request**:  
POST /users HTTP/1.1  
Host: localhost:8080  
Content-Type: application/json  
Cache-Control: no-cache

{ "Email": "test@gmail.com",  
 "Nickname": "test",  
 "Password": "password"  
}

**Response**:  
{  
 "id": 3,  
 "nickname": "test",  
 "email": "test@gmail.com",  
 "password": "$2a$10$712CexqpOwM3wtJxxS1ohePgG4tqbUH3FgZXX3Xg/SEURCxPgYs5m",  
 "created\_at": "2020-12-08T16:38:18.7062006Z",  
 "updated\_at": "2020-12-08T16:38:18.7062226Z"  
}

* **Step 2: Login new user and get the token ID**  
  

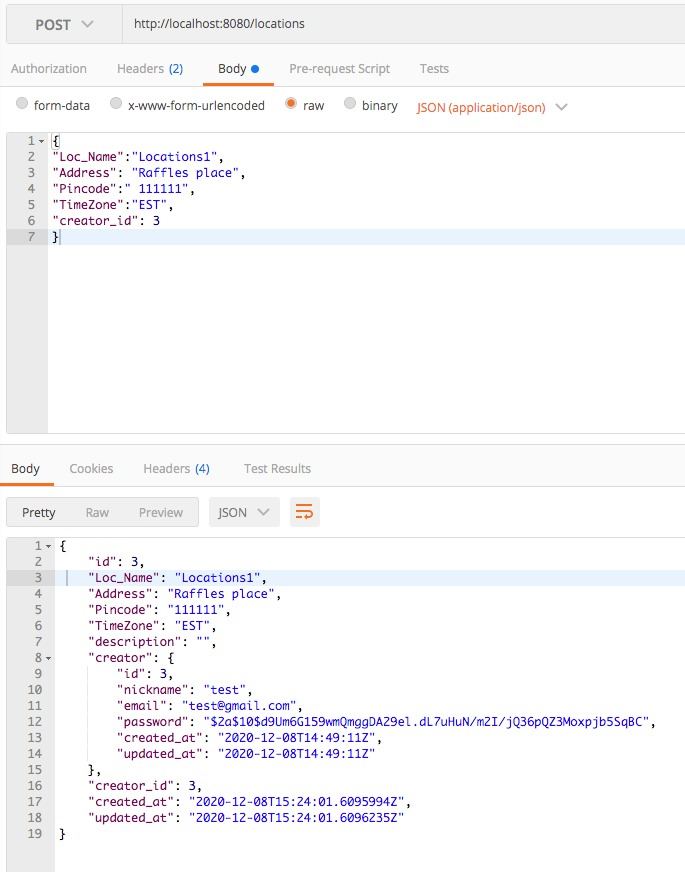
**Request**:  
POST /login HTTP/1.1  
Host: localhost:8080  
Cache-Control: no-cache

{ "email": "test@gmail.com",  
 "password": "password"  
}

**Response**:

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdXRob3JpemVkIjp0cnVlLCJleHAiOjE2MDc0NDQ1NTksInVzZXJfaWQiOjN9.ghW7CQEuA0ds4xY10CTFqOU3v21JwWXF8mxTcrpxnF0"

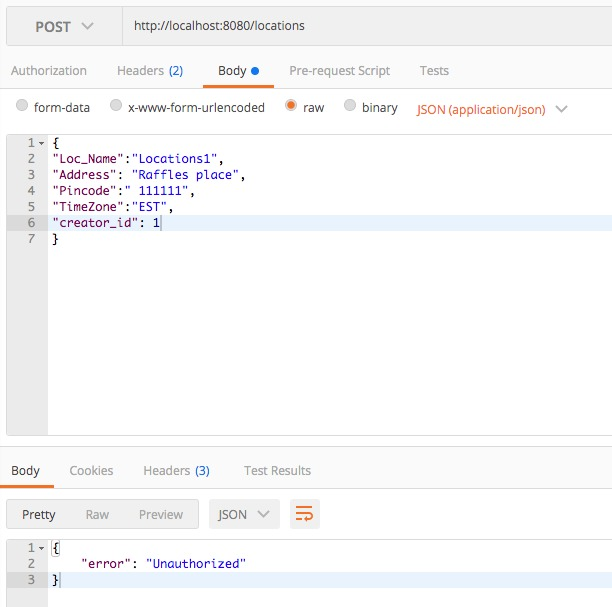
Note: Now onwards we will be using <TOKEN> instead of the actual token. Please replace it with the token you generate.

* **Step 3: Add new location by passing Authorization Token**

**Request**:  
POST /locations HTTP/1.1  
Host: localhost:8080  
Authorization: Bearer <TOKEN>  
Content-Type: application/json  
Cache-Control: no-cache

{  
"Loc\_Name":"Location1",  
"Address": "Raffles place",  
"Pincode":" 111111",  
"TimeZone":"EST",  
"creator\_id": 3  
}

**Response**:  
{  
 "id": 7,  
 "Loc\_Name": "Location1",  
 "Address": "Raffles place",  
 "Pincode": "111111",  
 "TimeZone": "EST",  
 "description": "",  
 "creator": {  
 "id": 3,  
 "nickname": "test",  
 "email": "test@gmail.com",  
 "password": "$2a$10$d9Um6G159wmQmggDA29el.dL7uHuN/m2I/jQ36pQZ3Moxpjb5SqBC",  
 "created\_at": "2020-12-08T14:49:11Z",  
 "updated\_at": "2020-12-08T14:49:11Z"  
 },  
 "creator\_id": 3,  
 "created\_at": "2020-12-08T15:44:57.4728873Z",  
 "updated\_at": "2020-12-08T15:44:57.4730496Z"  
}

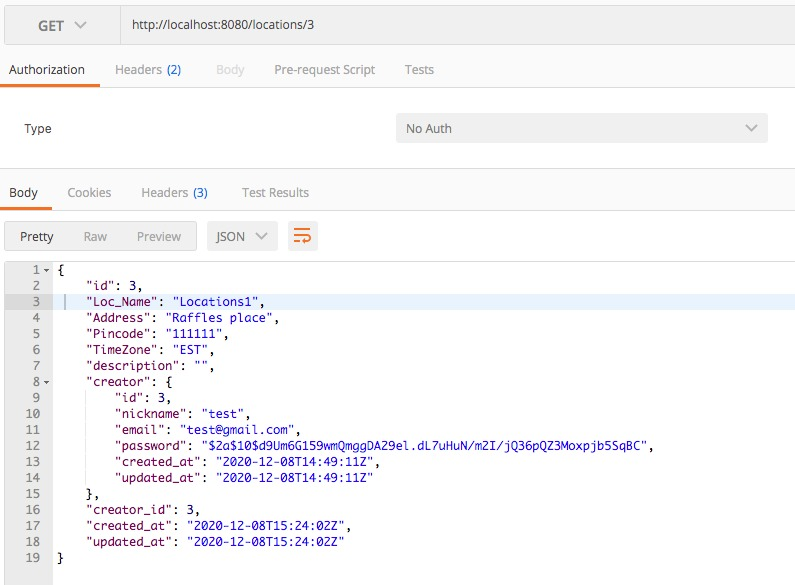
* **Step 4: Create Location without passing Authorization Token**  
  

**Request**:  
POST /locations HTTP/1.1  
Host: localhost:8080  
Content-Type: application/json  
Cache-Control: no-cache

{  
"Loc\_Name":"LocationNo2",  
"Address": "Raffles place",  
"Pincode":" 111111",  
"TimeZone":"EST",  
"creator\_id": 1  
}

**Response**:  
{  
 "error": "Unauthorized"  
}

* **Step 5: View the added location by creator of that location**



**Request**:

GET /locations/3 HTTP/1.1  
Host: localhost:8080  
Authorization: Bearer <TOKEN>  
Content-Type: application/json  
Cache-Control: no-cache

**Response**:  
{  
 "id": 3,  
 "Loc\_Name": "Locations1",  
 "Address": "Raffles place",  
 "Pincode": "111111",  
 "TimeZone": "EST",  
 "description": "",  
 "creator": {  
 "id": 3,  
 "nickname": "test",  
 "email": "test@gmail.com",  
 "password": "$2a$10$d9Um6G159wmQmggDA29el.dL7uHuN/m2I/jQ36pQZ3Moxpjb5SqBC",  
 "created\_at": "2020-12-08T14:49:11Z",  
 "updated\_at": "2020-12-08T14:49:11Z"  
 },  
 "creator\_id": 3,  
 "created\_at": "2020-12-08T15:24:02Z",  
 "updated\_at": "2020-12-08T15:24:02Z"  
}

* **Step 6: Add favorite location for a user**

**Request**:

POST /userlocations HTTP/1.1

Host: localhost:8080

Authorization: Bearer <TOKEN>

Content-Type: application/json

Cache-Control: no-cache

{  
 "user\_location\_id":3,

"creator\_id":3  
}

**Response**:

{

"id": 3,

"user\_location": {

"id": 3,

"Loc\_Name": "Locations1",

"Address": "Raffles place",

"Pincode": "111111",

"TimeZone": "EST",

"description": "",

"creator": {

"id": 0,

"nickname": "",

"email": "",

"password": "",

"created\_at": "0001-01-01T00:00:00Z",

"updated\_at": "0001-01-01T00:00:00Z"

},

"creator\_id": 3,

"created\_at": "2020-12-08T15:24:02Z",

"updated\_at": "2020-12-08T15:24:02Z"

},

"user\_location\_id": 3,

"creator": {

"id": 3,

"nickname": "test",

"email": "test@gmail.com",

"password": "$2a$10$d9Um6G159wmQmggDA29el.dL7uHuN/m2I/jQ36pQZ3Moxpjb5SqBC",

"created\_at": "2020-12-08T14:49:11Z",

"updated\_at": "2020-12-08T14:49:11Z"

},

"creator\_id": 3,

"created\_at": "2020-12-08T16:51:58.6498985Z",

"updated\_at": "2020-12-08T16:51:58.6499221Z"

}

Graphical user interface, text, application, email

Description automatically generated

* **Step 7: View favorite location just added**   
  Graphical user interface, text, application, email

  Description automatically generated

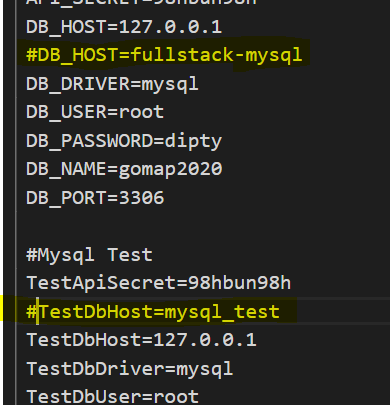
**Request**:  
GET /userlocations/3 HTTP/1.1  
Host: localhost:8080  
Authorization: Bearer <TOKEN>  
Content-Type: application/json  
Cache-Control: no-cache

**Response**:  
{  
 "id": 3,  
 "Loc\_Name": "Locations1",  
 "Address": "Raffles place",  
 "Pincode": "111111",  
 "TimeZone": "EST",  
 "description": "",  
 "creator": {  
 "id": 3,  
 "nickname": "test",  
 "email": "test@gmail.com",  
 "password": "$2a$10$d9Um6G159wmQmggDA29el.dL7uHuN/m2I/jQ36pQZ3Moxpjb5SqBC",  
 "created\_at": "2020-12-08T14:49:11Z",  
 "updated\_at": "2020-12-08T14:49:11Z"  
 },  
 "creator\_id": 3,  
 "created\_at": "2020-12-08T15:24:02Z",  
 "updated\_at": "2020-12-08T15:24:02Z"  
}

Time to run the Test Cases

Please follow below steps to run all the unit tests :

1. You must have Golang in your PATH set
2. You should connect Mysql database also.
3. Comment below yellow lines from .env file and uncomment the DB\_HOST and TestDb\_Host



1. Go to the project root path and run below command all the tests will run at once :-

go test -v ./...

