User Manual for Online Movie Booking Application

By:

Soumosir Dutta
sdutta7@umd.edu

Table of Contents

Project Introduction

Setup guide

API guide

Security Guide

Logging Guide

Validation for apis

Secure Login

Test guide

Project Introduction

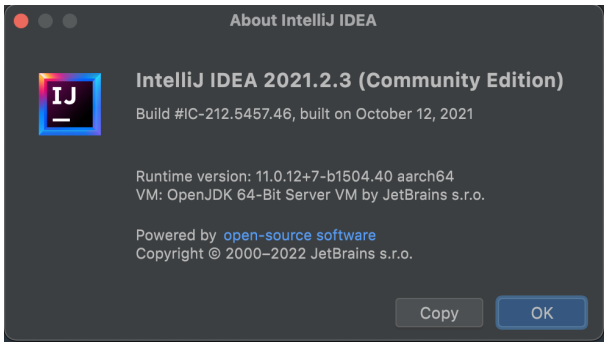
Movie Book is an online platform for booking movies. It will be a common platform for both theater owners and customers. We aim to develop a system where theater owners can manage their theaters and add new movies etc. It also offers information about upcoming movies, their timings, cast, and other movie details.

Project Setup :-

1. Java version 11

java --version
openjdk 11.0.12 2021-07-20
OpenJDK Runtime Environment Microsoft-25199 (build 11.0.12+7)
OpenJDK 64-Bit Server VM Microsoft-25199 (build 11.0.12+7, mixed mode)

2. Development and running test case environment
Intellij Idea 2021.2.3



3. https://dev.mysql.com/downloads/mysql/

Download mysql server
Run DB in 3306 port

Make a root user and set password as “password”

url: jdbc:mysql://localhost:3306/bookmyDB
    username: root
    password: password

If using brew

brew services start mysql
mysql_secure_installation

mysql -u root -p

Reference : https://flaviocopes.com/mysql-how-to-install/

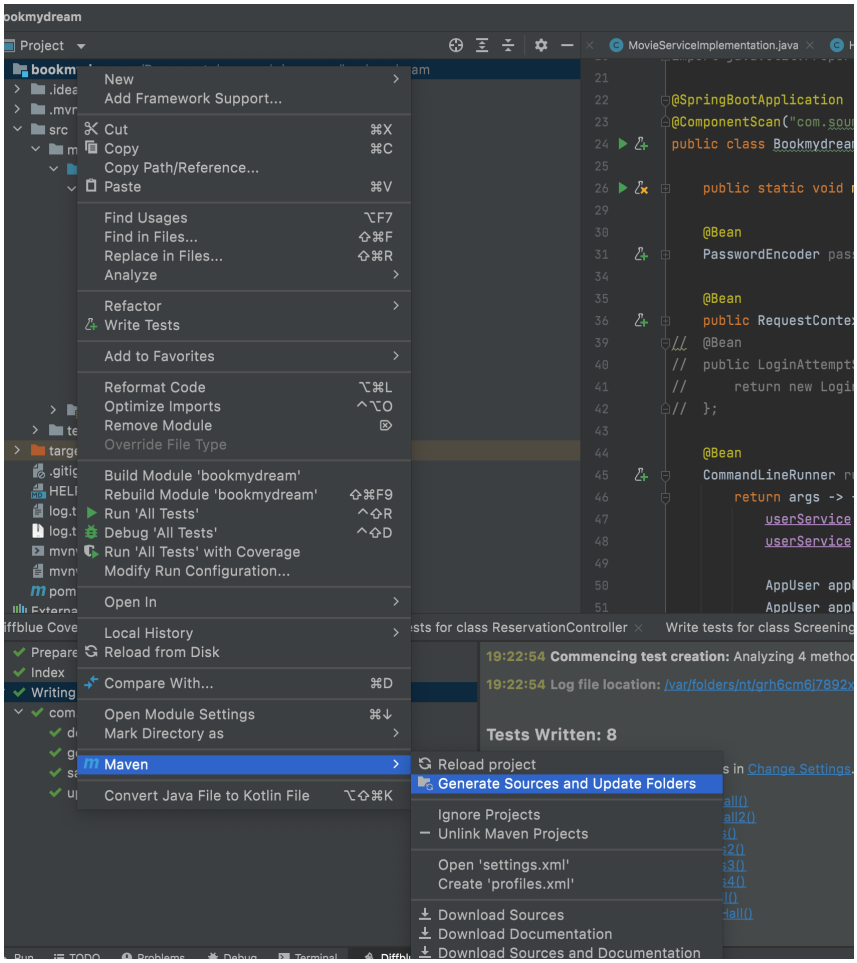
4. Clone the repo in any directory

git clone git@code.umd.edu:sdutta7/ENPM809WFall2022Project-sdutta7.git

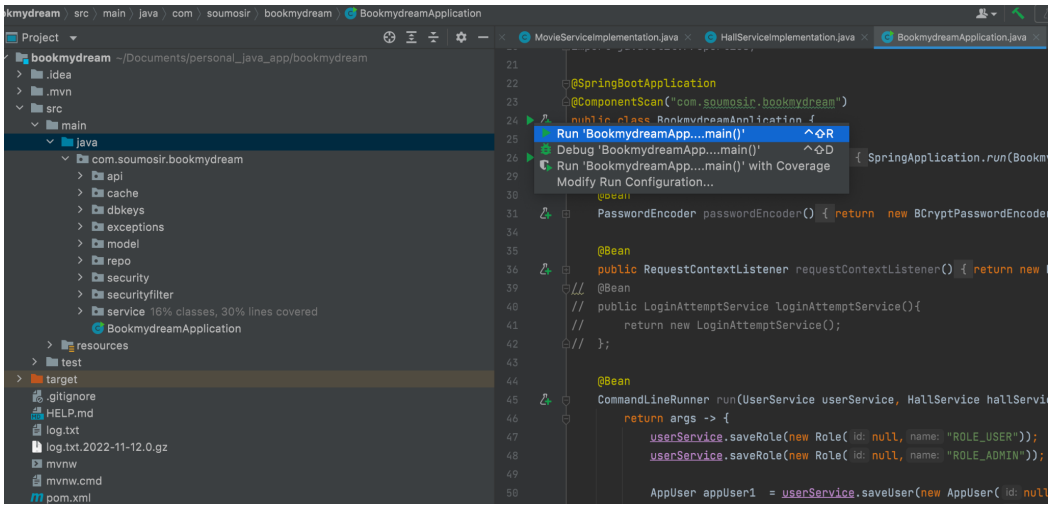
Git checkout Phase2

5. Open the folder in intellij
And wait for some time for the project to load

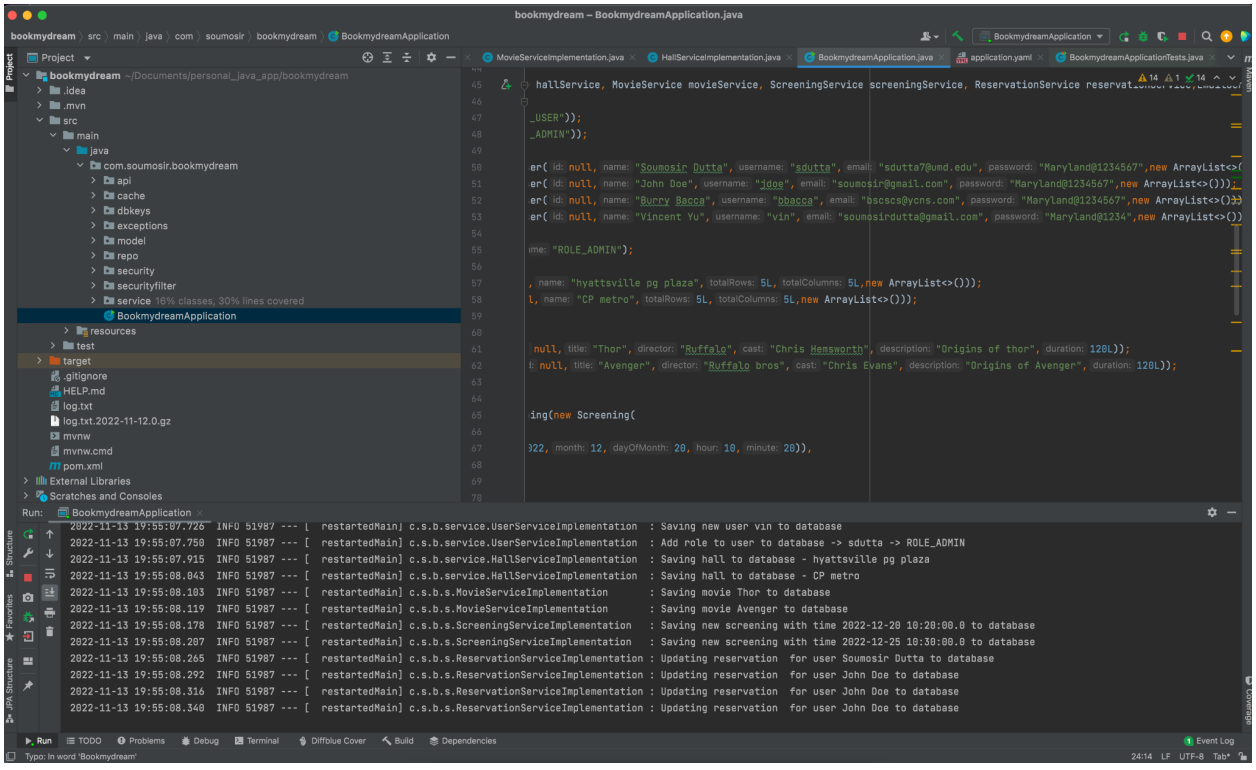
From IntelliJ, you can install maven dependency by installing by clicking pom.xml and selecting reload the project. The screenshot is below for reference.



6.
Navigat to src/main/java/com/soumosir/bookmydream/BookmydreamApplication.java
and
Run BookMyDreamAplication.class by the green button



One the app is up and running, you should be able to see this in console



API - Login - POST - https://localhost:8443/api/login
--data-urlencode 'username=<string>' \
--data-urlencode 'password=<string>'

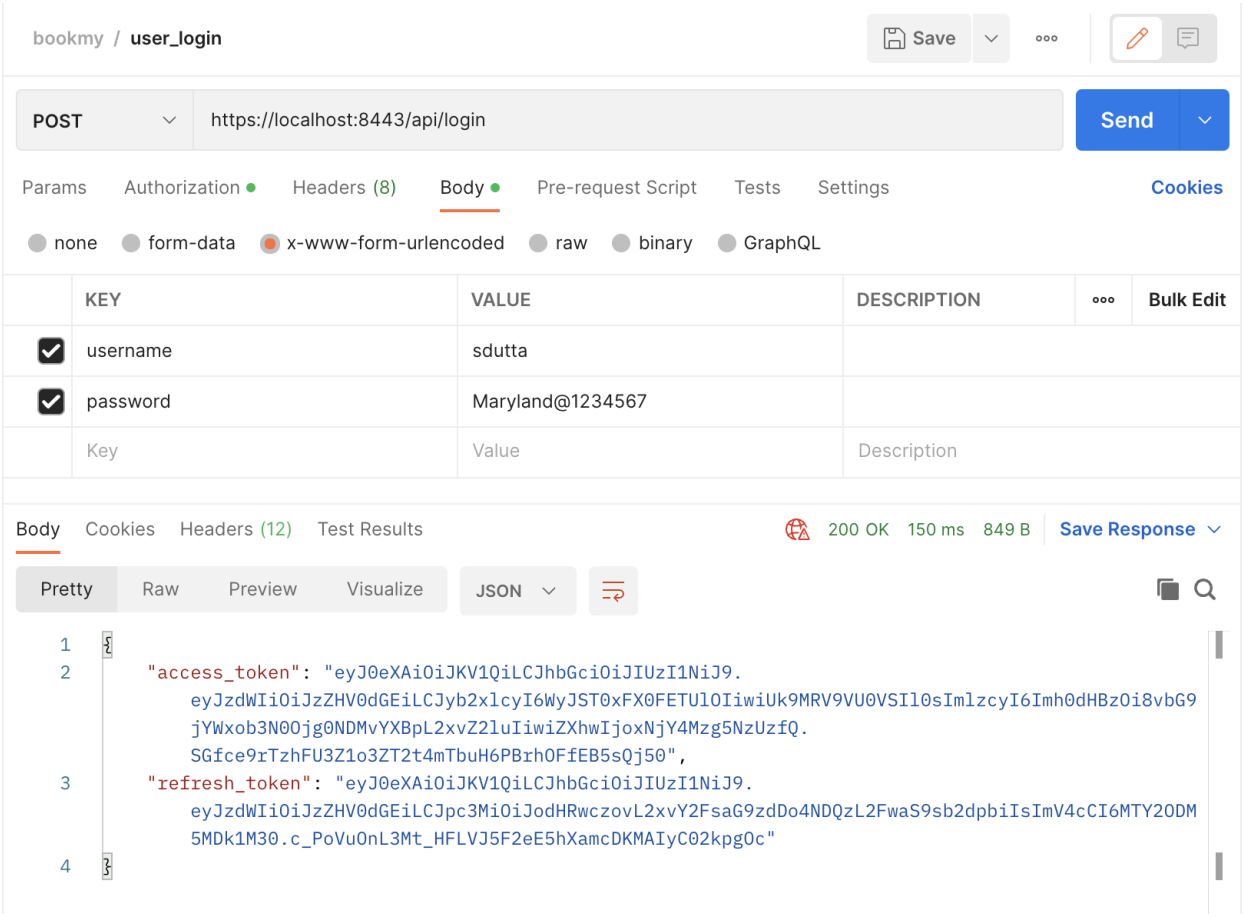
Example :-
Run the curl command or by post man to login as admin

curl -k --location --request POST 'https://localhost:8443/api/login' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'username=sdutta' \
--data-urlencode 'password=Maryland@1234567'

Sample Response

{"access_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSIl0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzk0MTU2fQ.oopZiI9RUET9opQTes50g_IBbncncb5cXR3fGnTaFVo","refresh_token":"eyJ0eXAiOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJpc3MiOiJodHRwczovL2xvY2FsaG9zdDo4NDQzL2FwaS9sb2dpbiIsImV4cCI6MTY2ODM5MTM1Nn0.I7ONqcixAqrGbcDhDBey7MBxZORZLnLi-hUpunzu3jM"}

OR Postman for



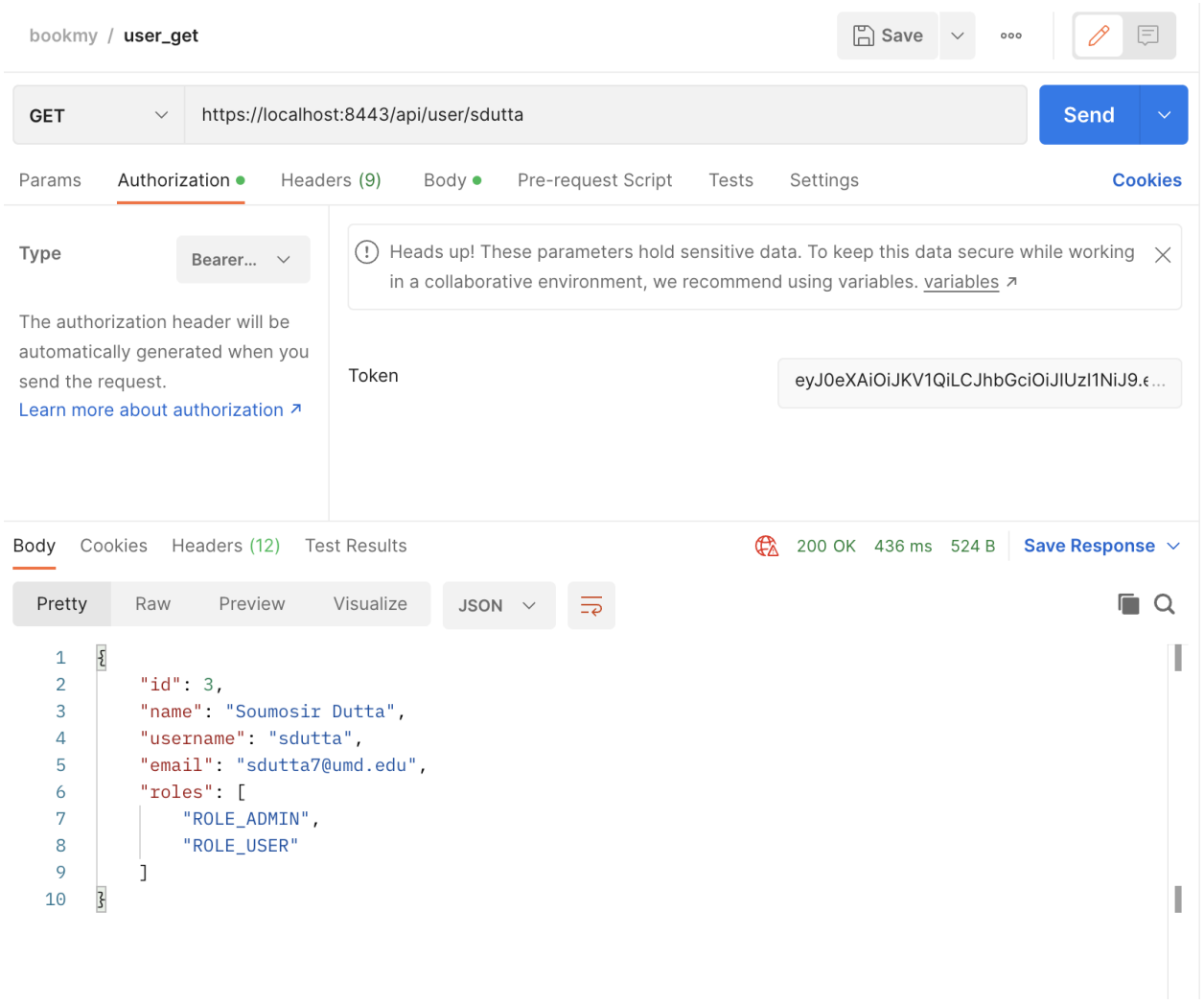
Copy the access token to Authorization bearer For subsequent request

API - Get User by username - GET - 'https://localhost:8443/api/user/sdutta' {Auth_token}

Example

curl -k --location --request GET 'https://localhost:8443/api/user/sdutta' \
--header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSII0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \
--data-raw ''

OR

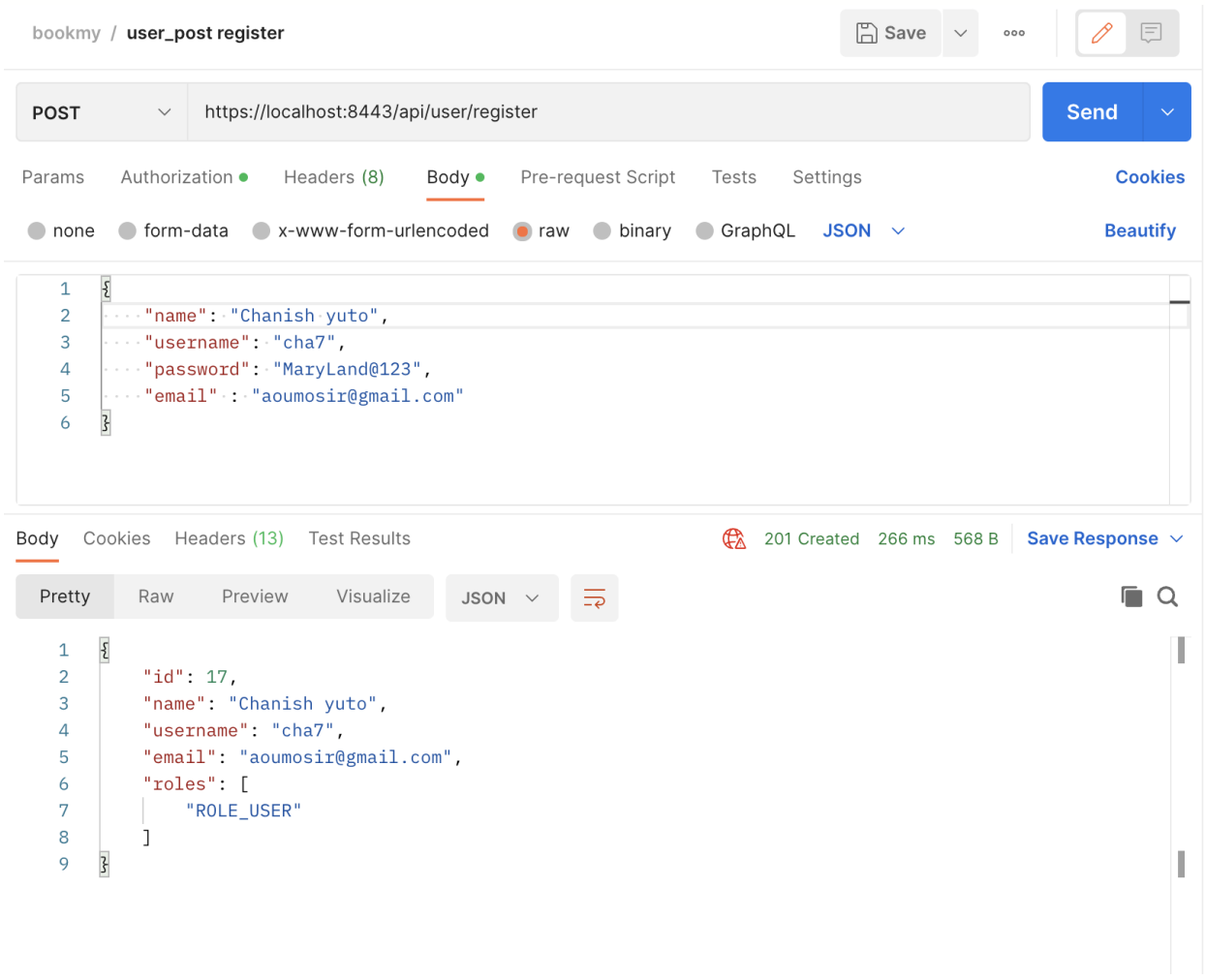


API - registering a new user - POST - 'https://localhost:8443/api/user/register'

```
{
  "name": "<string>",
  "username": "<string>",
  "password": "<string>",
  "email" : "<string>"
}
```

Example

```
curl --location --request POST 'https://localhost:8443/api/user/register' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name": "Chanish yuto",
  "username": "cha7",
  "password": "MaryLand@123",
  "email" : "aoumosir@gmail.com"
}'
```



Note there is validation matcher too

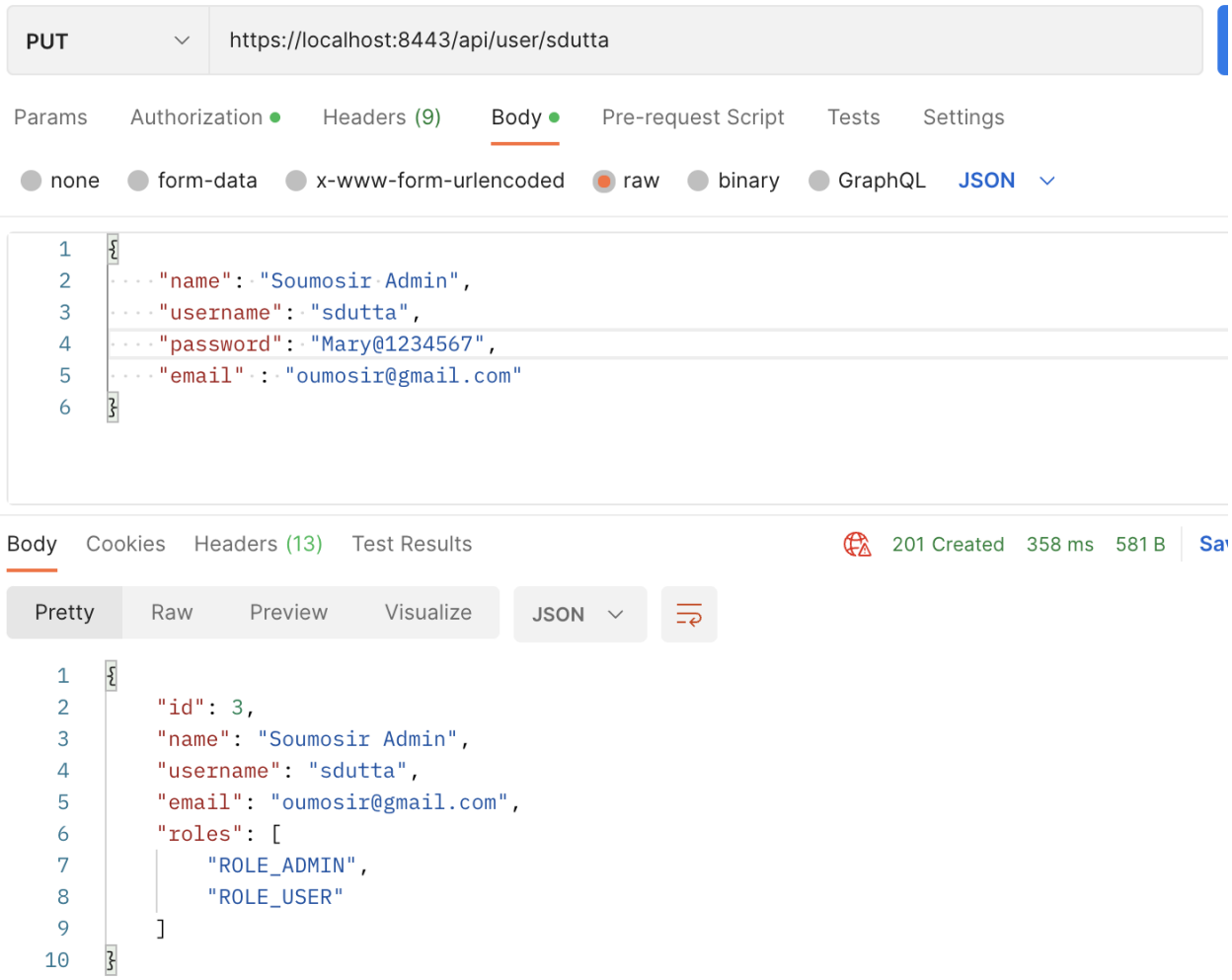
——————————————————

API - **Modify a user** - PUT - https://localhost:8443/api/user/<username> {Auth_token}

```
{
  "name": "<string>",
  "username": "<string>",
  "password": "<string>",
  "email" : "<string>"
}
```

Example :

```
curl -k --location --request PUT 'https://localhost:8443/api/user/sdutta' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSIl0sImlz
cyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PB
rhOFfEB5sQj50' \
--header 'Content-Type: application/json' \
--data-raw '{
   "name": "Soumosir Admin",
   "username": "sdutta",
   "password": "Mary@1234567",
   "email" : "oumosir@gmail.com"
}'
```



Deleting a user

API - Deleting a user - DELETE - https://localhost:8443/api/user/<username> {Auth_token}

Example
curl -k --location --request DELETE 'https://localhost:8443/api/user/sdutta' \

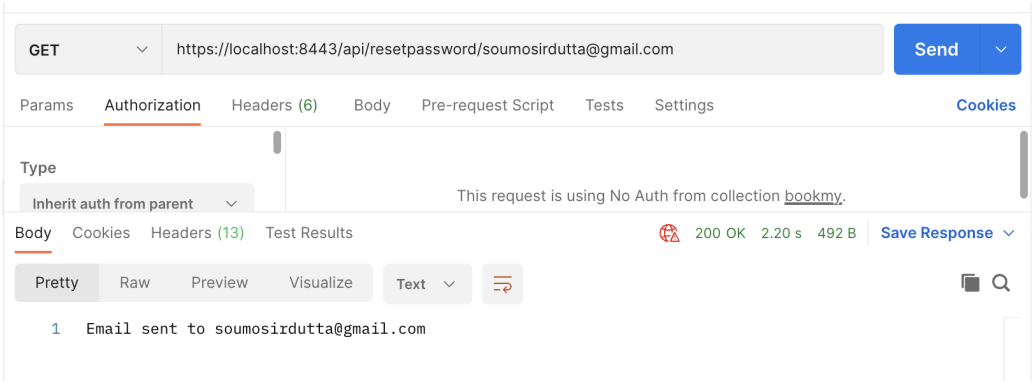
--header 'Authorization: Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSIl0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \
--header 'Content-Type: application/json' \
--data-raw ''

RESET a user password (REQEST)

API - RESET user password - GET - 'https://localhost:8443/api/resetpassword/<email>'

Example
curl --location --request GET 'https://localhost:8443/api/resetpassword/soumosirdutta@gmail.com'



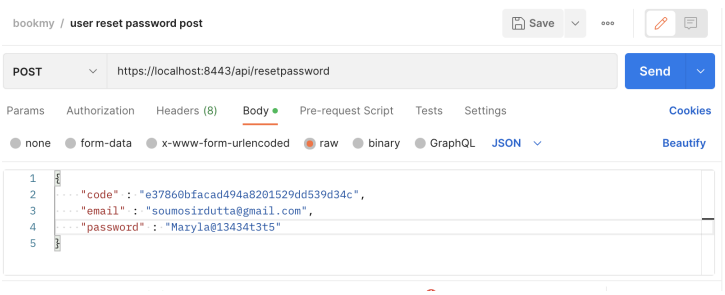
RESET a userpassword (POST)

API - RESET user password - POST - https://localhost:8443/api/resetpassword/

```
{
  "code" : "string",
  "email" : "string",
  "password" : "string"
}
```

Example

```
curl --location --request POST 'https://localhost:8443/api/resetpassword' \
--header 'Content-Type: application/json' \
--data-raw '{
  "code" : "e37860bfacad494a8201529dd539d34c",
  "email" : "soumosirdutta@gmail.com",
  "password" : "Maryla@13434t3t5"
}'
```

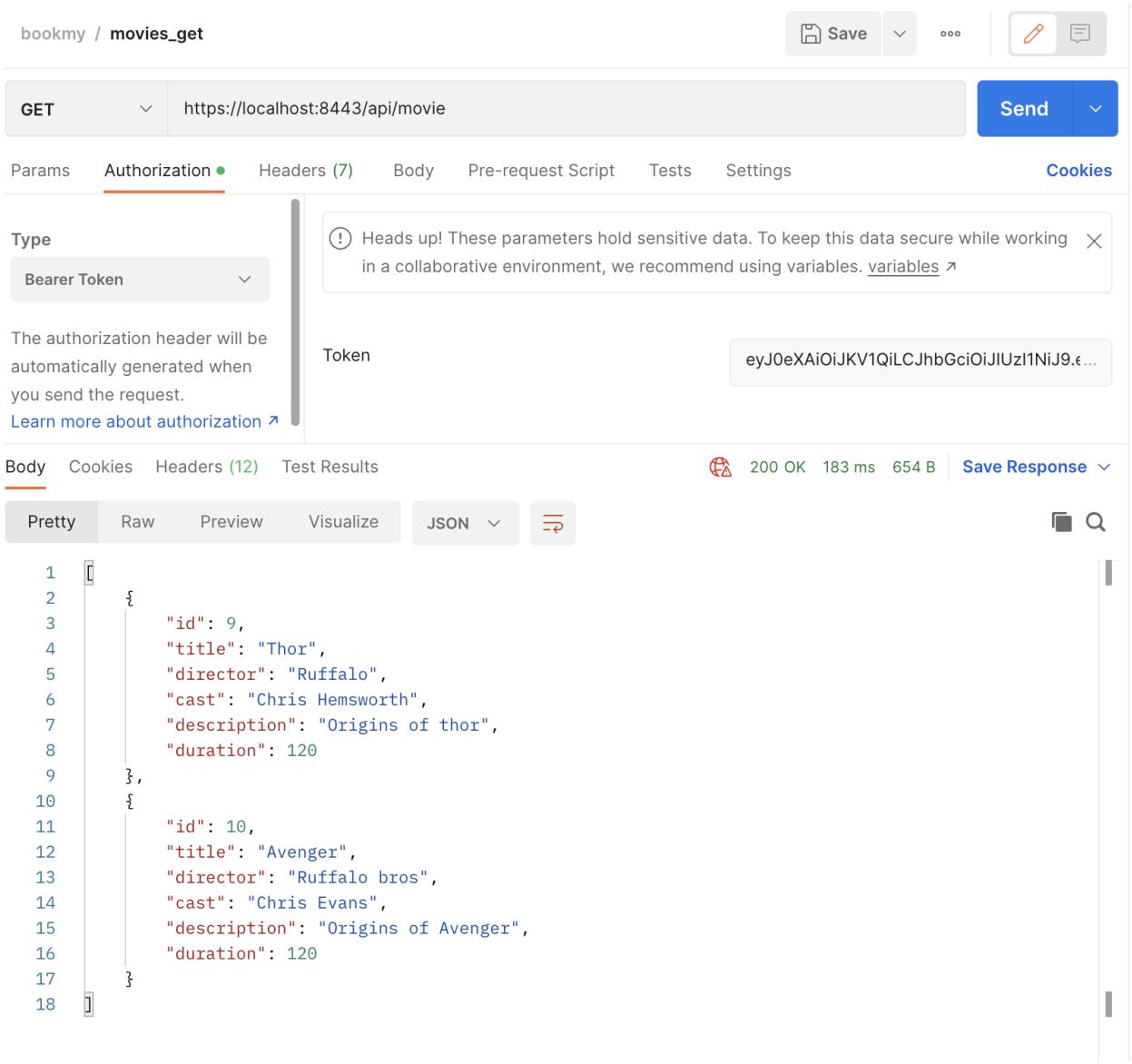


GET all the movies (USER and ADMIN allowed)

API - GET all the movies - GET - https://localhost:8443/api/movie {Auth_token}

Example :-

curl --location --request GET 'https://localhost:8443/api/movie' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSII0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50'



Add a movie (ADMIN Only)

API - add the movies - POST - https://localhost:8443/api/movie {Auth_token}

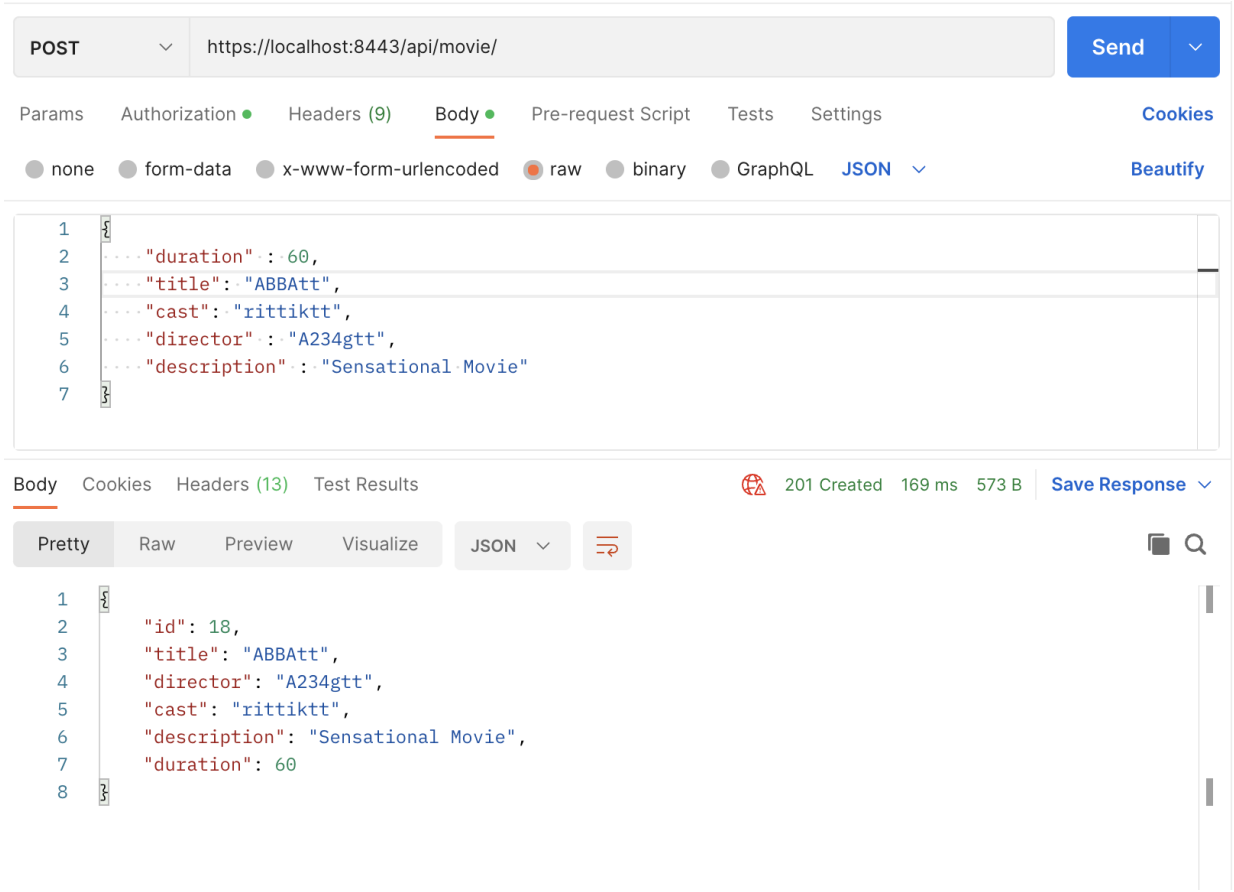
'{

"duration" : number, "title": string, "cast": string "director" : string "description" : string

}'

Example :-

```
curl -k --location --request POST 'https://localhost:8443/api/movie/' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSIl0sImlzcyI6Imh0dHBzOi8v
bG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \
--header 'Content-Type: application/json' \
--data-raw '{
  "duration" : 60,
  "title": "ABBAtt",
  "cast": "rittiktt",
  "director" : "A234gtt",
  "description" : "Sensational Movie"
}'
```



Edit a movie(Admin)

API - Edit a movie - POST - https://localhost:8443/api/movie/<movie_id> {Auth_token}

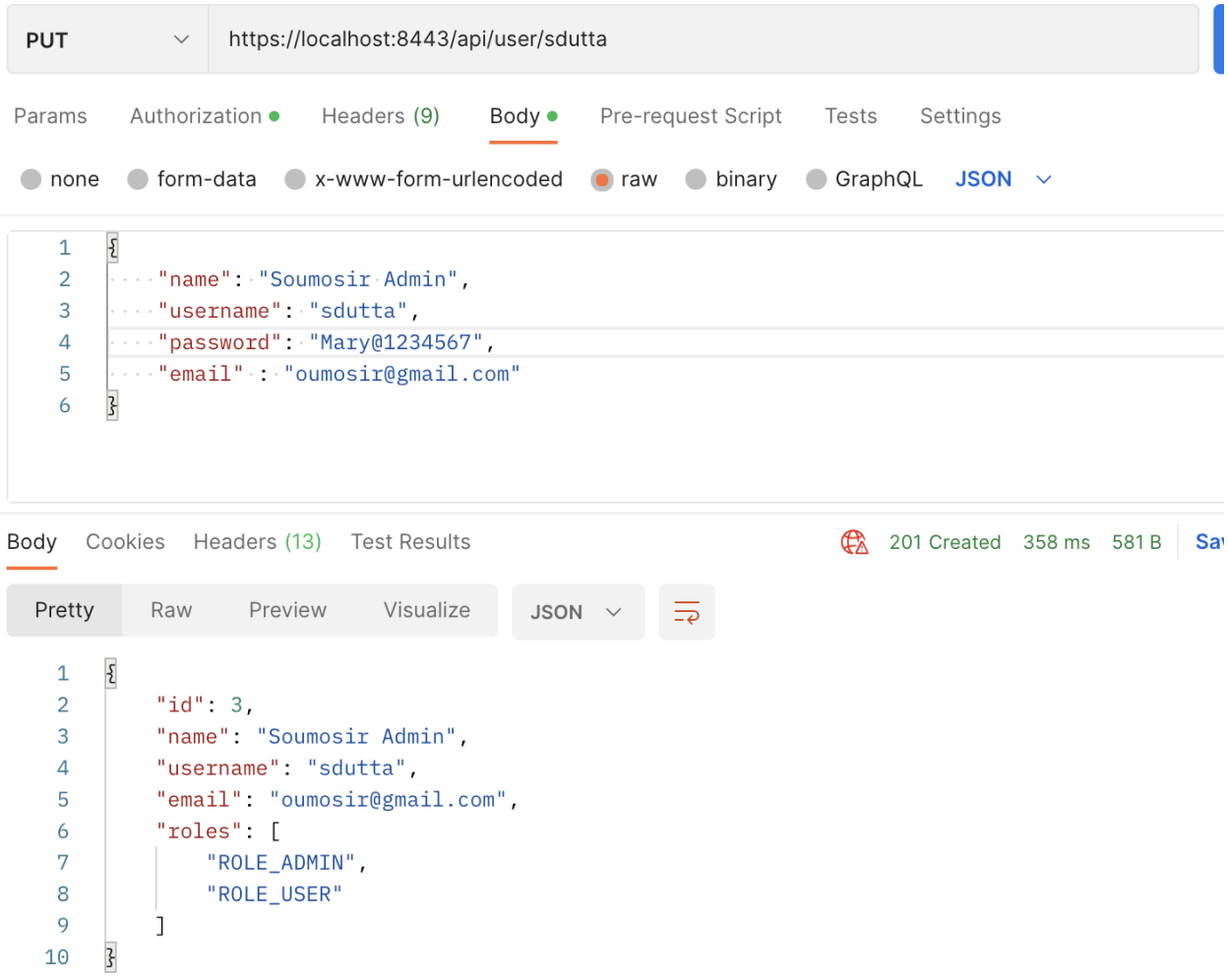
'{

"duration" : number, "title": string, "cast": string "director" : string "description" : string

}'

Example :-

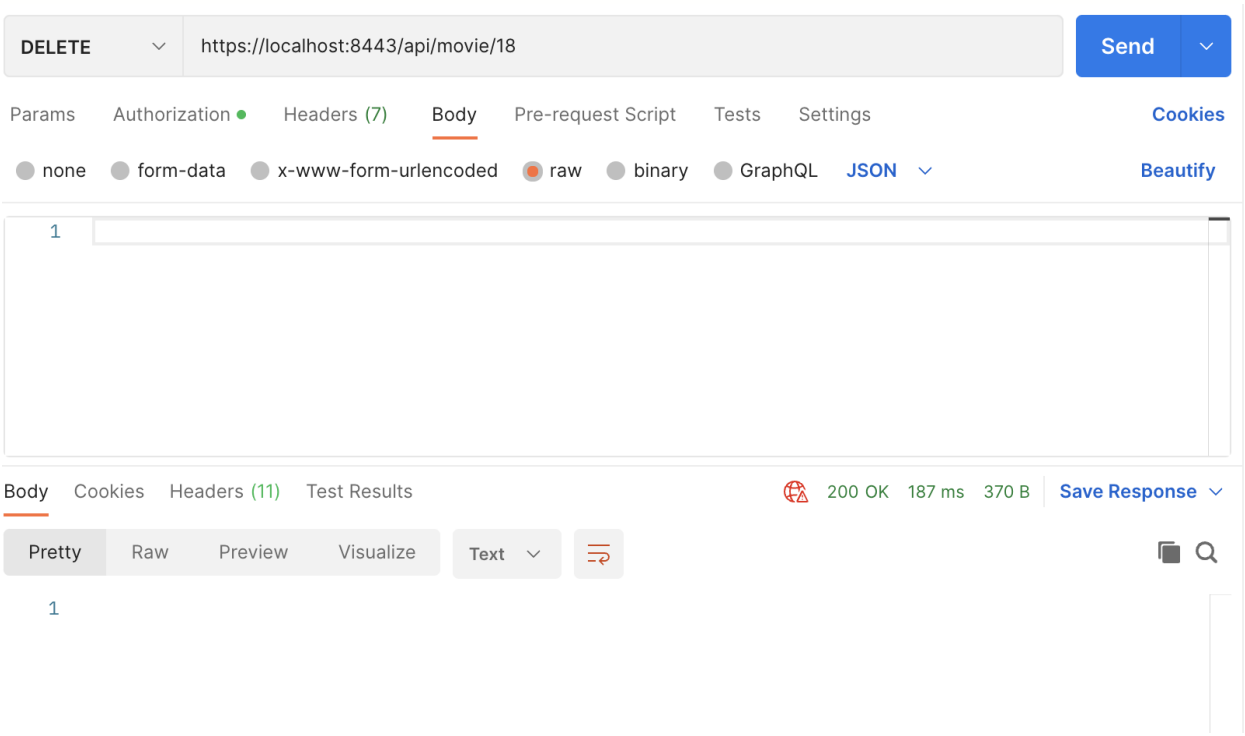
```
curl -k --location --request PUT 'https://localhost:8443/api/movie/18' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSIl0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \
--header 'Content-Type: application/json' \
--data-raw '{
  "duration" : 60,
  "title": "ABBA",
  "cast": "rittik Gus",
  "director" : "Nolan",
  "description" : "Sensational Movie"
}'
```



Delete a movie(Admin)

API - Delete a movie - DELETE - https://localhost:8443/api/movie/<movie_id> {Auth_token}

curl --location --request DELETE 'https://localhost:8443/api/movie/18' \
--header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSII0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \
--data-raw ''

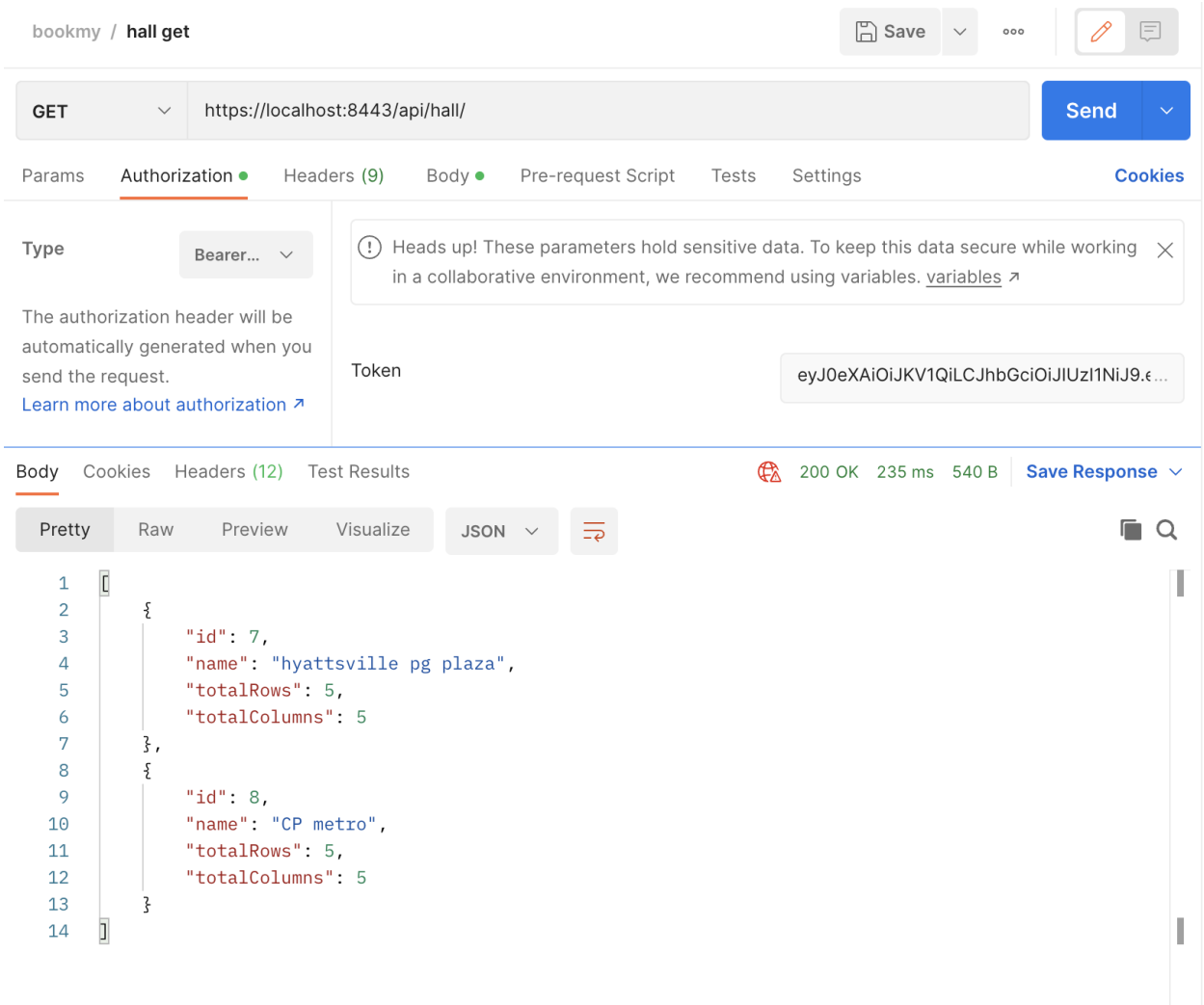


Get all halls(admin and user allowe)

API - Get all halls - GET - https://localhost:8443/api/hall/ {Auth_token}

Example :-

curl -k --location --request GET 'https://localhost:8443/api/hall/' \
--header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSII0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \
--header 'Content-Type: application/json' \
--data-raw ''



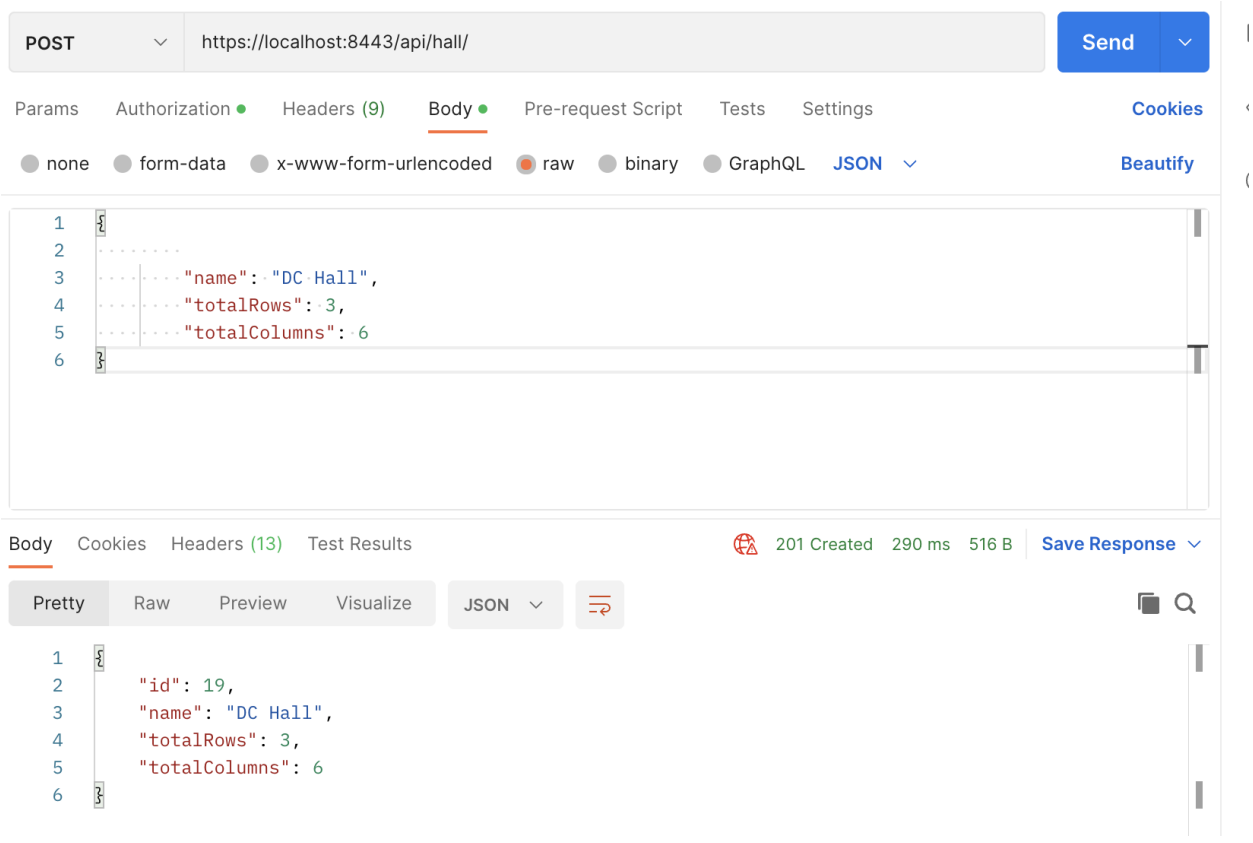
Add a new hall (admin only)

API - Add new hall - POST - https://localhost:8443/api/hall/ {Auth_token}

{ "name": "string", "totalRows": number, "totalColumns": number }

```
curl -k --location --request POST 'https://localhost:8443/api/hall/' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9
VU0VSIl0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9r
TzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \
--header 'Content-Type: application/json' \
--data-raw '{

    "name": "DC Hall",
    "totalRows": 3,
    "totalColumns": 6
}'
```



Edit a hall by id (admin only)

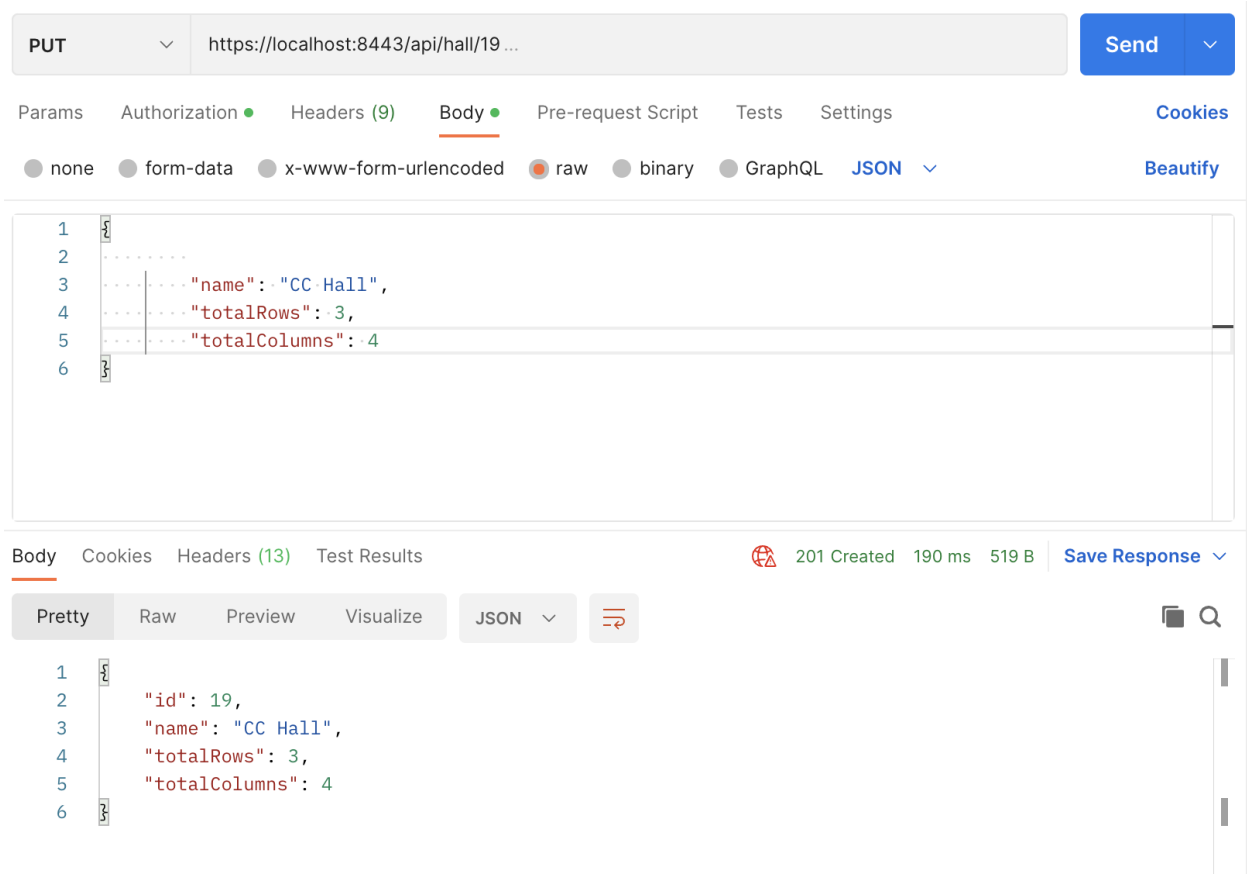
API - edit hall - PUT - https://localhost:8443/api/hall/<hall_id> {Auth_token}

{ "name": "string", "totalRows": number, "totalColumns": number }

Example

```
curl -k --location --request PUT 'https://localhost:8443/api/hall/19' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSII0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \
--header 'Content-Type: application/json' \
--data-raw '{

    "name": "CC Hall",
    "totalRows": 3,
    "totalColumns": 4
}'
```



Delete a hall(admin only)

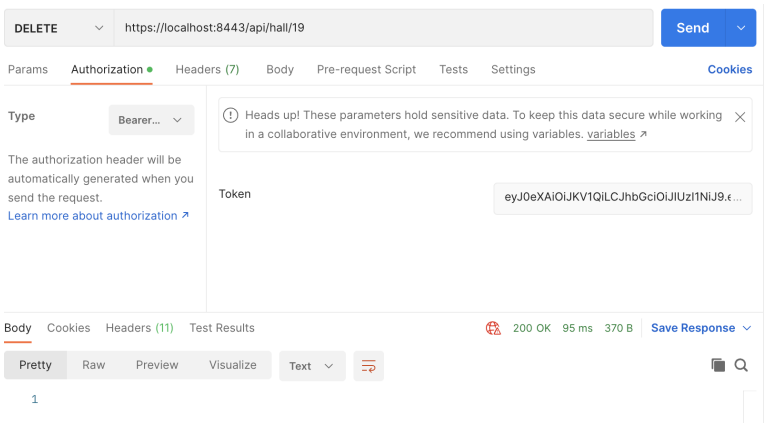
API - delete hall - DELETE - https://localhost:8443/api/hall/<hall_id> {Auth_token}

Example :-

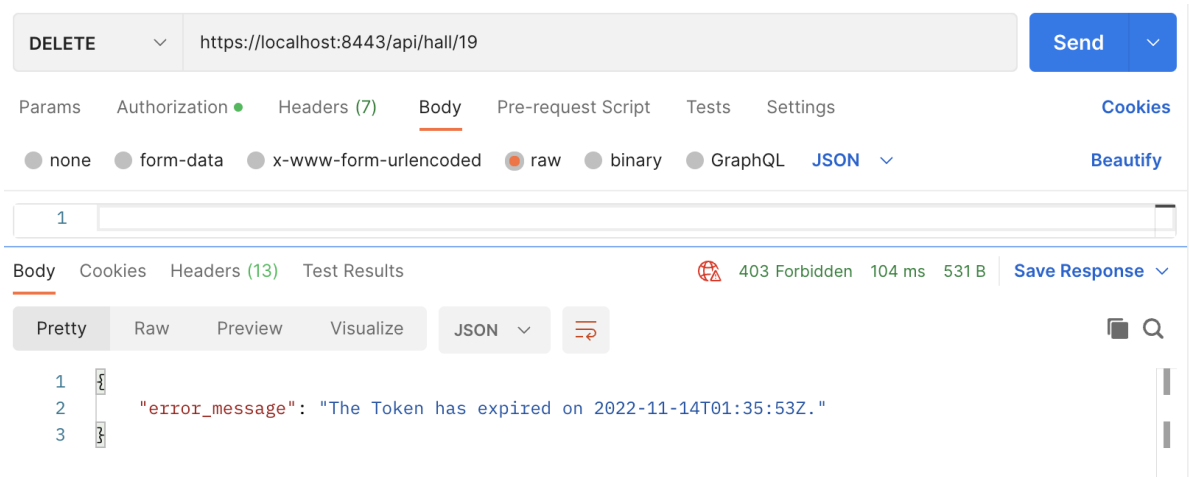
curl -k --location --request DELETE 'https://localhost:8443/api/hall/19' \

--header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSII0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4Mzg5NzUzfQ.SGfce9rTzhFU3Z1o3ZT2t4mTbuH6PBrhOFfEB5sQj50' \

--data-raw ''



Token expires after 5mins

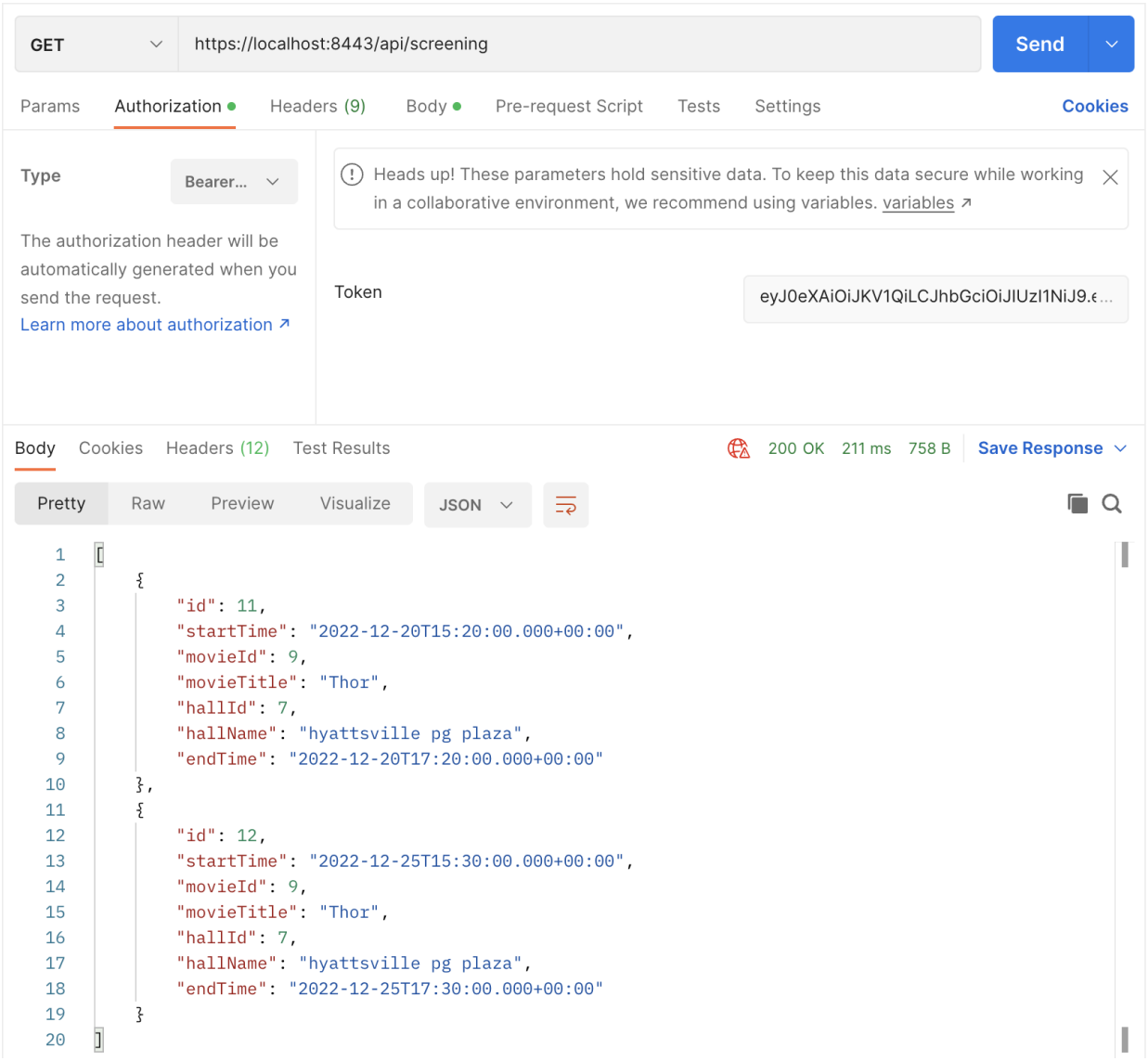


Screenings GET (Admin and user both)
[Screening is like event or the particular movie running on a particular hall at a particular time]

API - get all screenings - GET - https://localhost:8443/api/screeningl/ {Auth_token}

Example:

curl -k --location --request GET 'https://localhost:8443/api/screening' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSII0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4MzkyMjgwfQ.M_ttW_el_tbCN5rEsEq0QSfaxcaQeM6sib0dxuaeqRk' \
--header 'Content-Type: application/json' \
--data-raw ''



Screening Add a new screening (Admin)

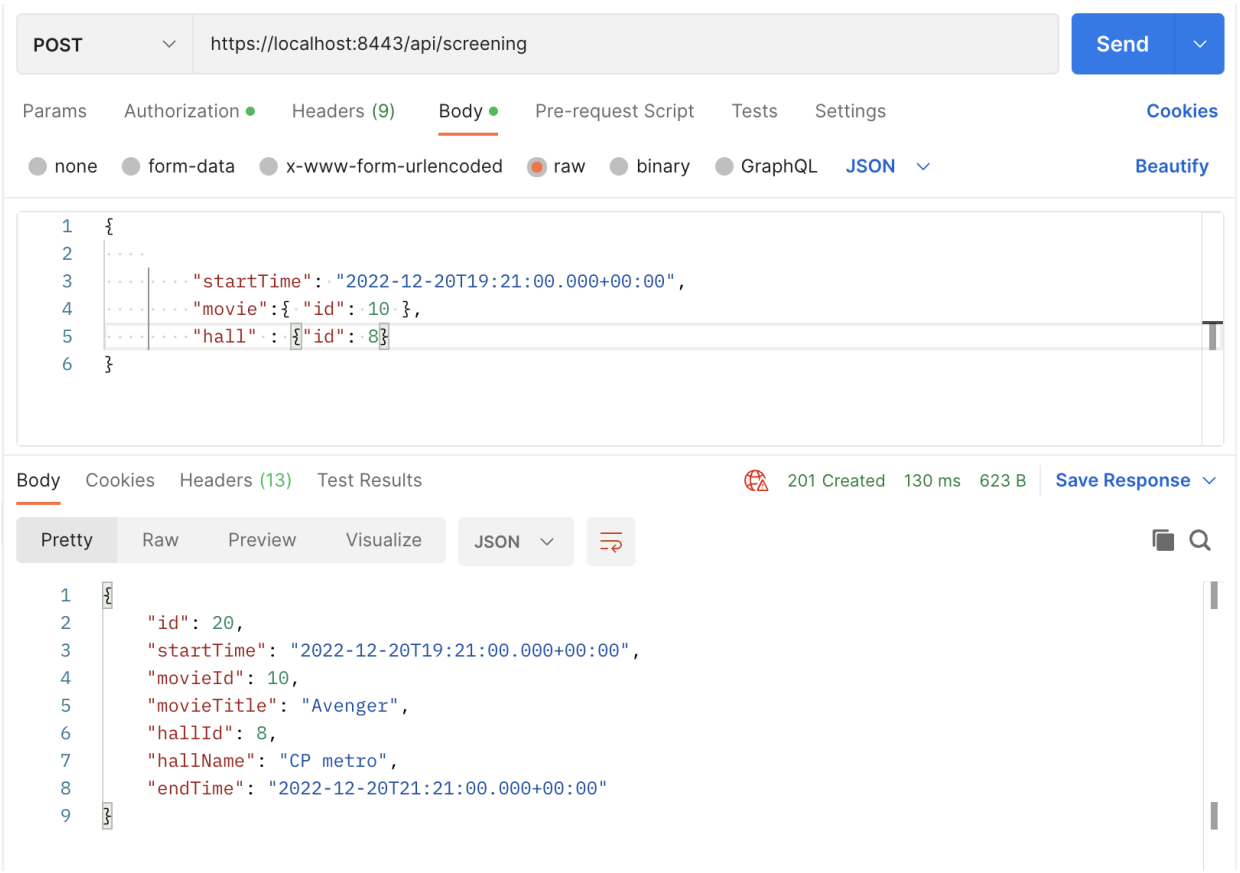
API - Add a new screening - POST - https://localhost:8443/api/screeningl/ {Auth_token}

{ "startTime": "string date", "movie":{ "id": num }, "hall" : {"id": num} }

Example :-

```
curl -k --location --request POST 'https://localhost:8443/api/screening' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9
VU0VSIl0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4MzkyMjgwfQ.M_ttW_el
_tbCN5rEsEq0QSfaxcaQeM6sib0dxuaeqRk' \
--header 'Content-Type: application/json' \
--data-raw '{

    "startTime": "2022-12-20T19:21:00.000+00:00",
    "movie":{ "id": 10 },
    "hall" : {"id": 8}
}'
```



Screening Edit a existing Screening(Admin)

API - edit screening - PUT - https://localhost:8443/api/screening/<id> {Auth_token}

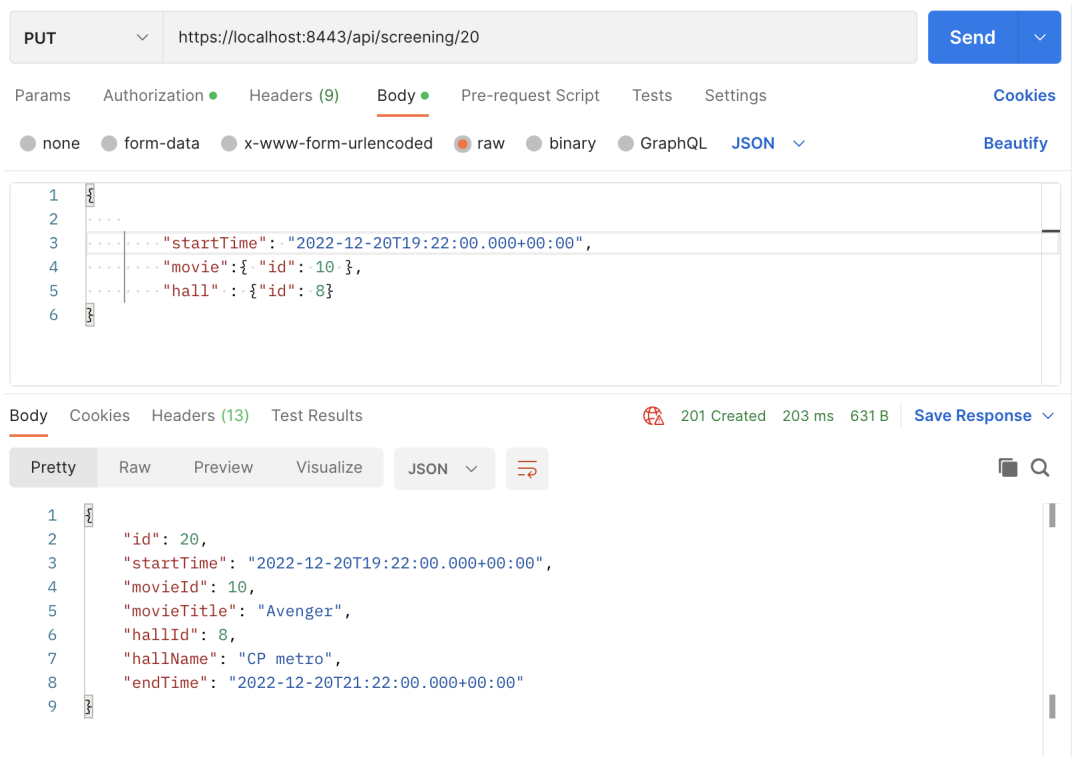
{ "startTime": "string date", "movie":{ "id": num }, "hall" : {"id": num} }

Example -

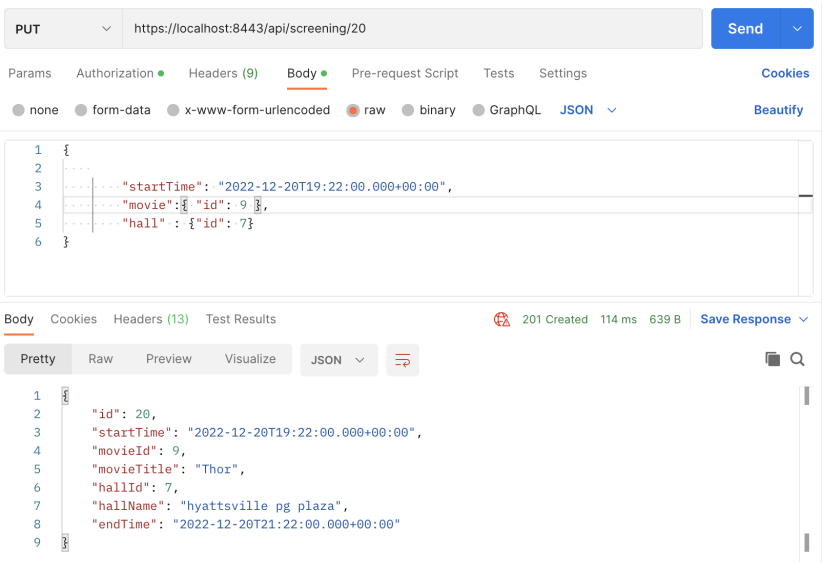
```
curl --location --request PUT 'https://localhost:8443/api/screening/20' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSIl0sImlzcyI6Imh0dHBzOi8v
bG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4MzkyMjgwfQ.M_ttW_el_tbCN5rEsEq0QSfaxcaQeM6sib0dxuaeqRk' \
--header 'Content-Type: application/json' \
--data-raw '{

    "startTime": "2022-12-20T19:22:00.000+00:00",
    "movie":{ "id": 10 },
    "hall" : {"id": 8}
}'
```

Changed start time



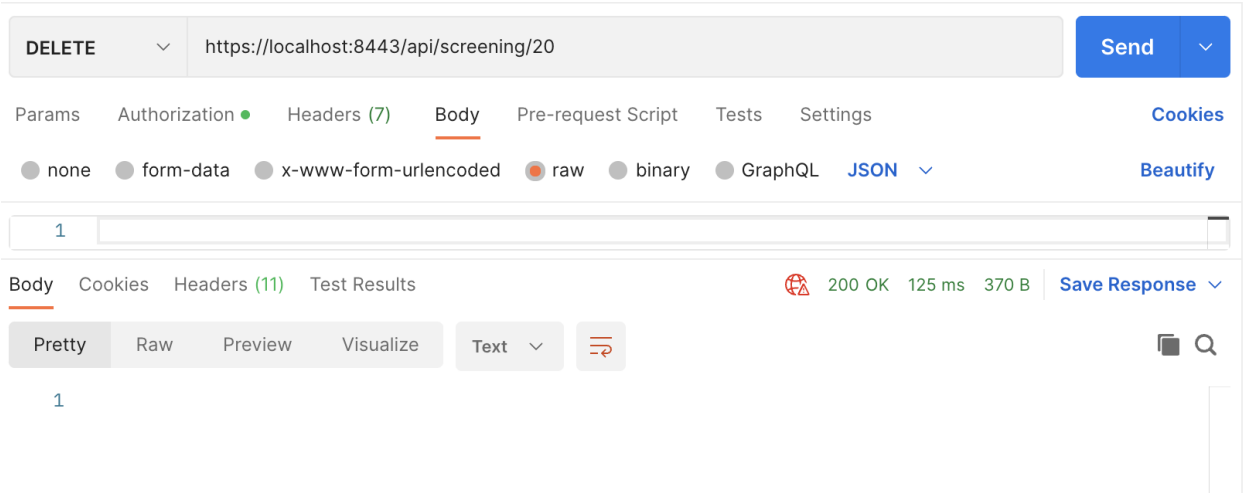
Changed hall , movie and start time -



Screening Delete a existing Screening(Admin)

API - DeleteScreening - DELETE - https://localhost:8443/api/screening/<id> {Auth_token}

curl -k --location --request DELETE 'https://localhost:8443/api/screening/20' \
--header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzZHV0dGEiLCJyb2xlcyI6WyJST0xFX0FETUlOIiwiUk9MRV9VU0VSII0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4MzkyMjgwfQ.M_ttW_el_tbCN5rEsEq0QSfaxcaQeM6sib0dxuaeqRk' \
--data-raw ''

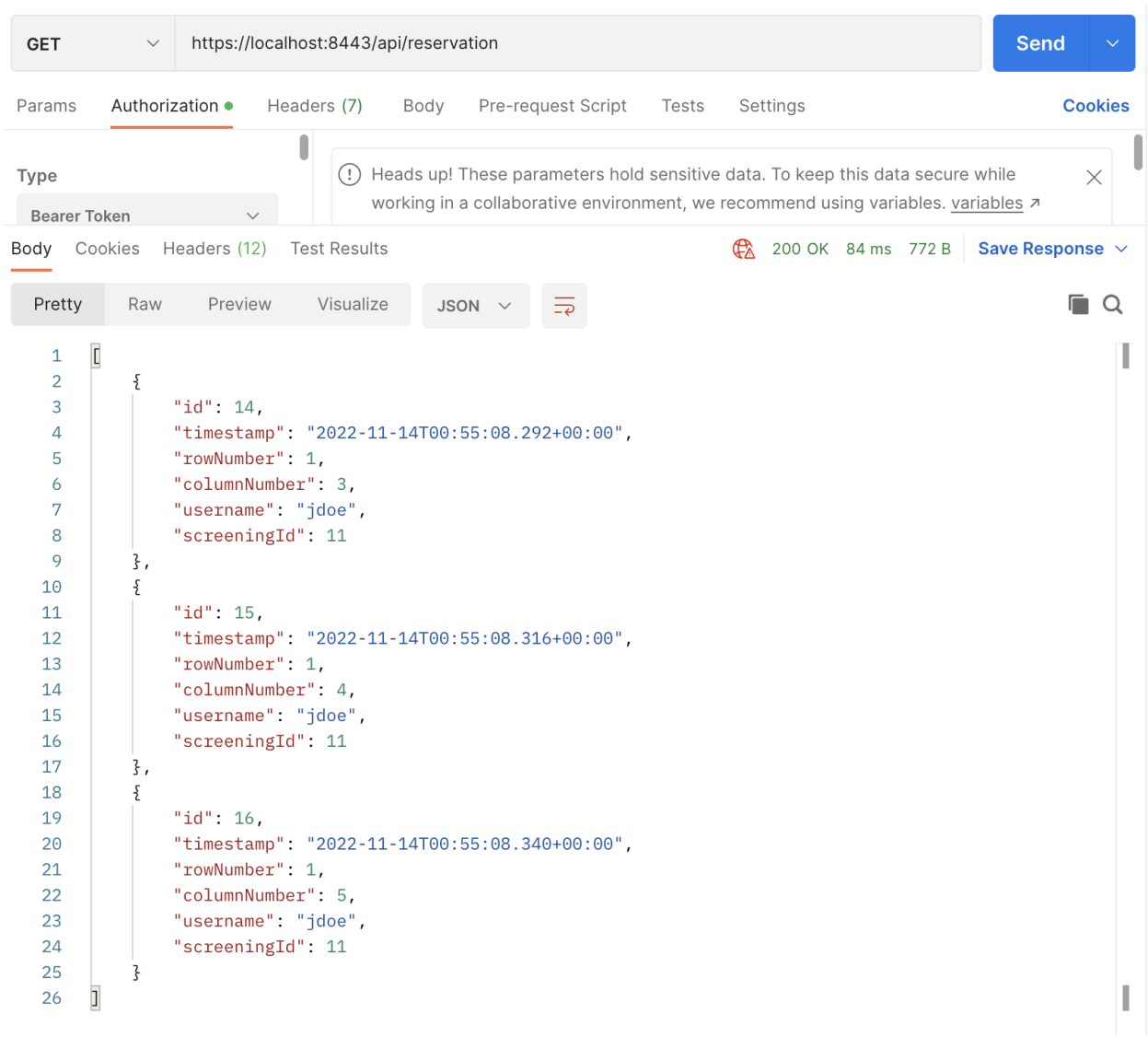


Reservations - User and Admin (Note we are not adding payment functionality as of now)

Get reservation done by that particular user. Admin can see all reservations

API - Get Reservation - DELETE - https://localhost:8443/api/reservation/ {Auth_token}
Example :-

```
curl -k --location --request GET 'https://localhost:8443/api/reservation' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJqZG9lIiwicm9sZXMiOlsiUk9MRV9VU0VSIl0sImlzcyI6Imh0dH
BzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4MzkzMDc5fQ.MJuy2oOLLA0jGvmzbBGhMeiSE5
Z5N_iNcDQ4KB7ar2g' \
--data-raw ''
```

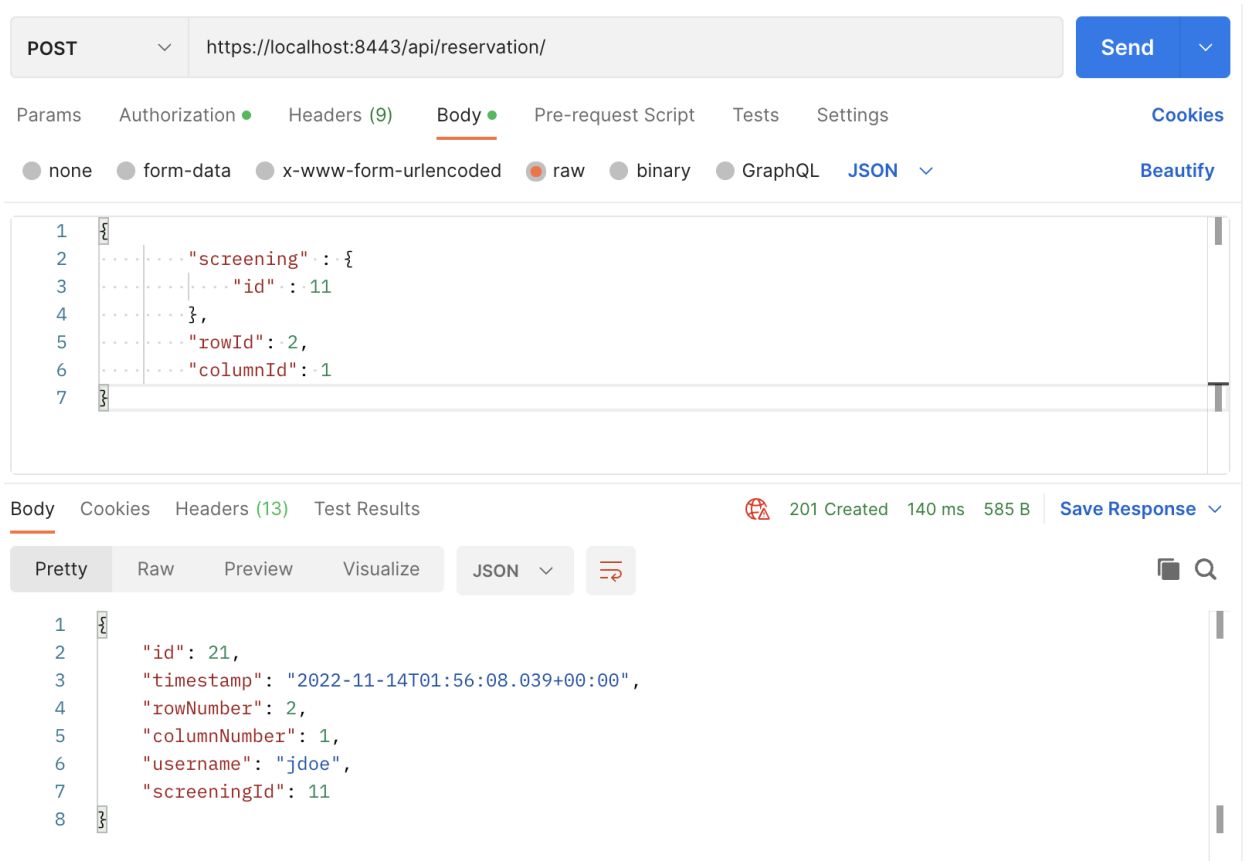


Add Reservations (user - { can only Add his reservation})

API - Add Reservation - POST - https://localhost:8443/api/reservation/ {Auth_token}

{ "screening" : { "id" : num }, "rowId": num, "columnId": num }'

```
curl -k --location --request POST 'https://localhost:8443/api/reservation/' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJqZG9lIiwicm9sZXMiOlsiUk9MRV9VU0VSIl0sImlzcyI6Imh0dH
BzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4MzkzMDc5fQ.MJuy2oOLLA0jGvmzbBGhMeiSE5
Z5N_iNcDQ4KB7ar2g' \
--header 'Content-Type: application/json' \
--data-raw '{
    "screening" : {
      "id" : 11
    },
    "rowId": 2,
    "columnId": 1
}'
```

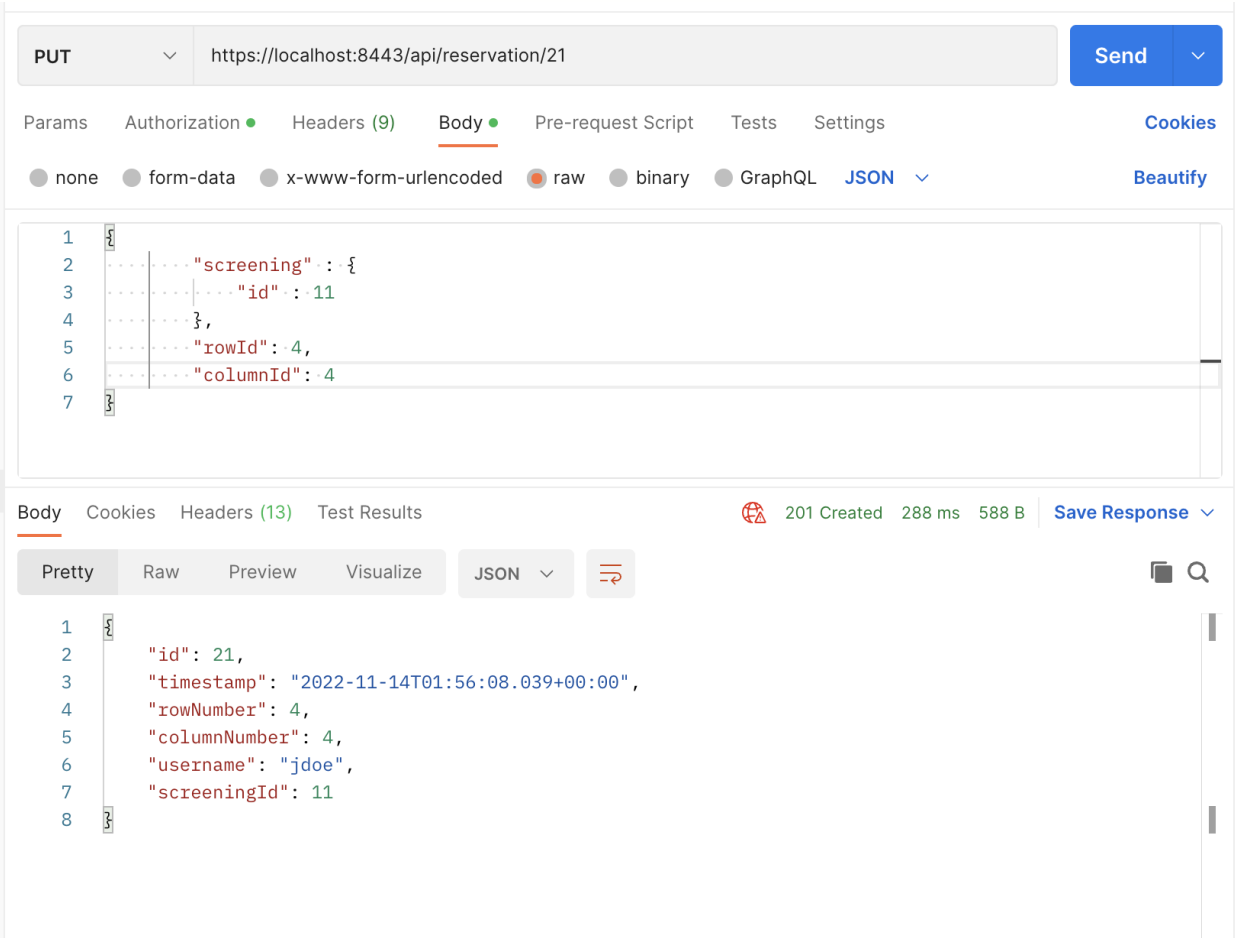


Update Reservations (user - { can only Update his reservation})

API - Edit Reservation - PUT - https://localhost:8443/api/reservation/<id> {Auth_token}

{ "screening" : { "id" : num }, "rowId": num, "columnId": num }'

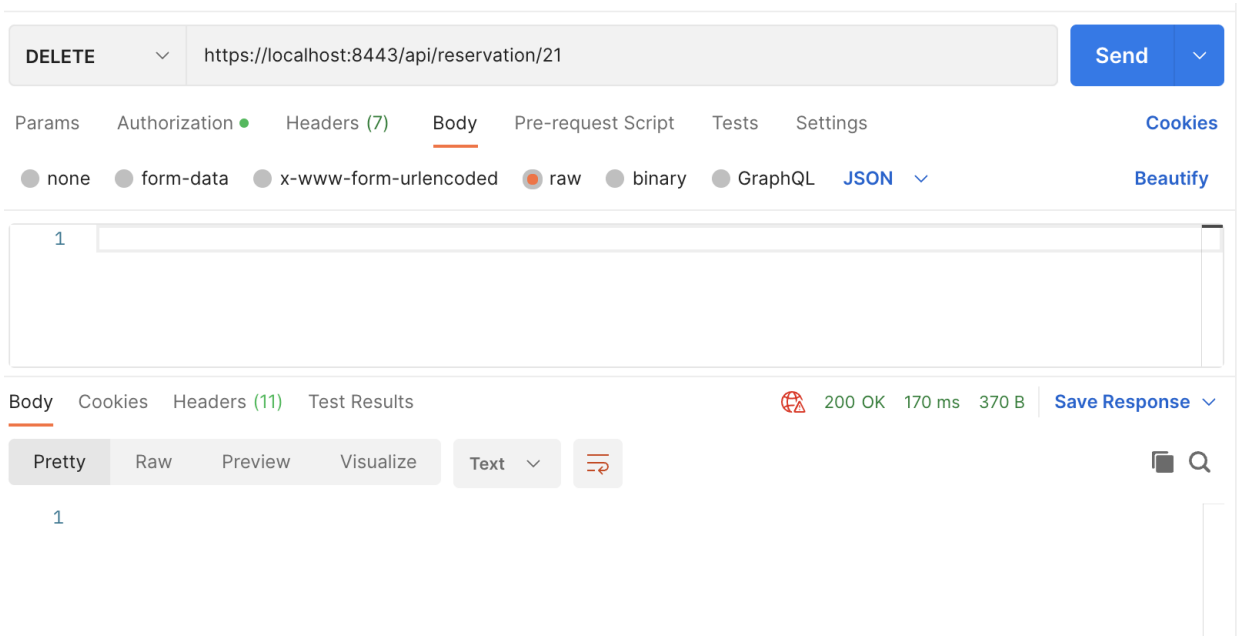
```
curl -k --location --request PUT 'https://localhost:8443/api/reservation/21' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJqZG9lIiwicm9sZXMiOlsiUk9MRV9VU0VSII0sImlzcyI6Imh0dH
BzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4MzkzMDc5fQ.MJuy2oOLLA0jGvmzbBGhMeiSE5
Z5N_iNcDQ4KB7ar2g' \
--header 'Content-Type: application/json' \
--data-raw '{
    "screening" : {
        "id" : 11
    },
    "rowId": 4,
    "columnId": 4
}'
```



Delete Reservations (user - { can only Delete own reservation})

API - Delete Reservation - DELETE - https://localhost:8443/api/reservation/<id> {Auth_token}

curl --location --request DELETE 'https://localhost:8443/api/reservation/21' \
--header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJqZG9lIiwicm9sZXMiOlsiUk9MRV9VU0VSIl0sImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0Ojg0NDMvYXBpL2xvZ2luIiwiZXhwIjoxNjY4MzkzMDc5fQ.MJuy2oOLLA0jGvmzbBGhMeiSE5Z5N_iNcDQ4KB7ar2g' \
--data-raw ''

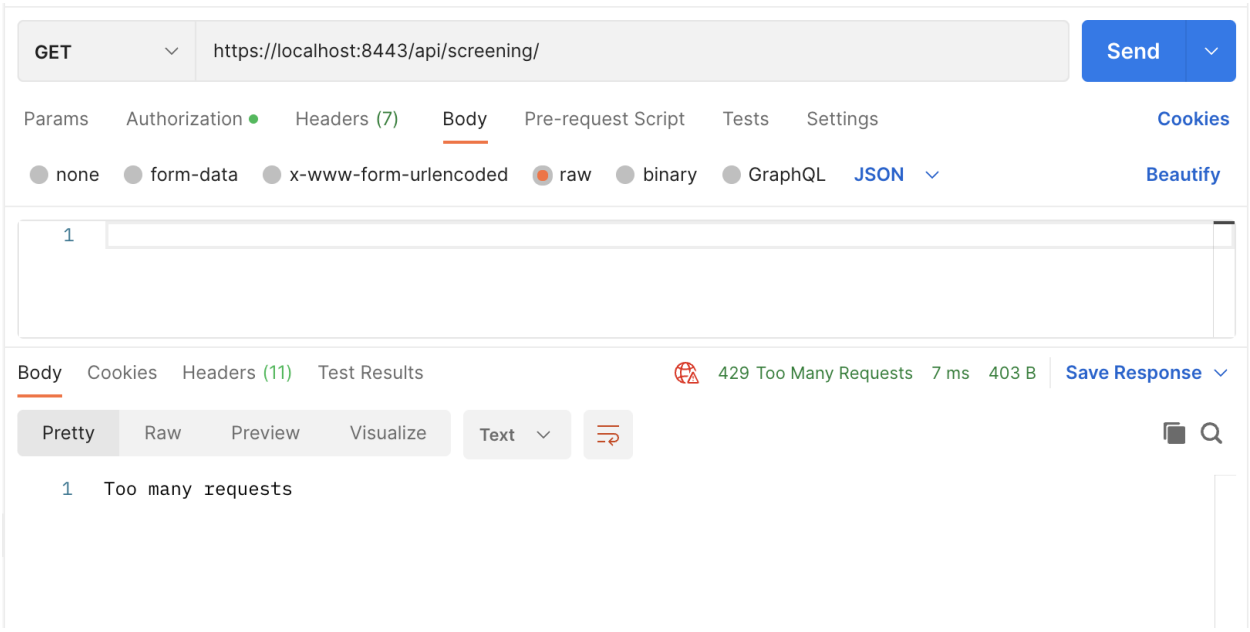


Security features :-

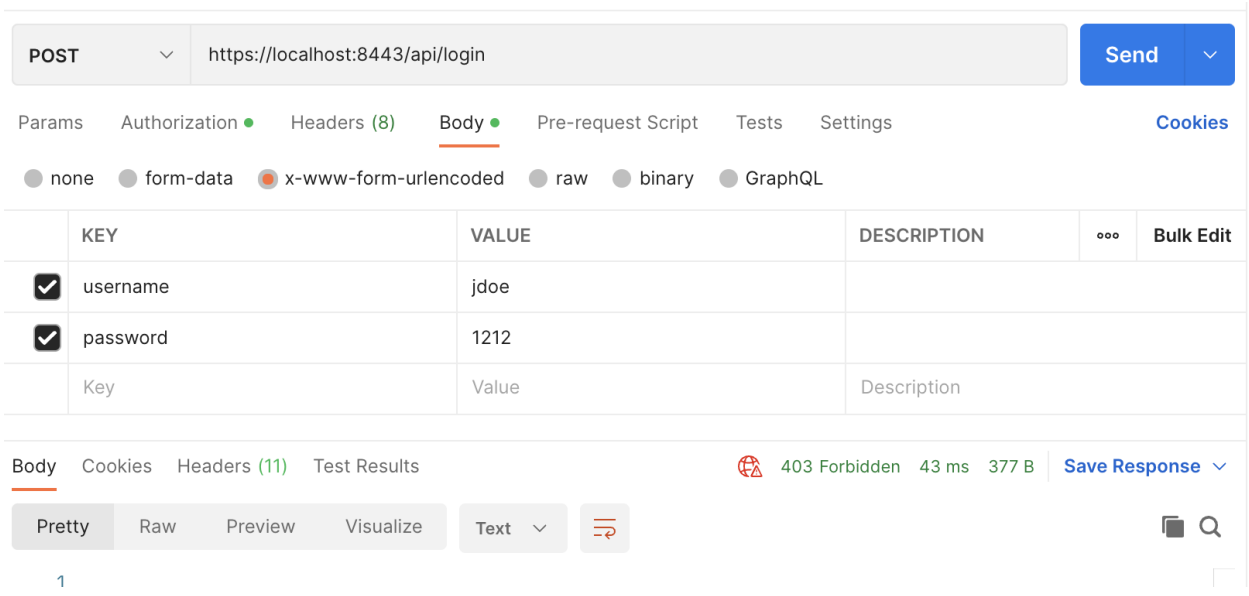
1. Rate limiting

Cannot access api more than 60 times in 1 mins from an IP (Using a cache for this ip:attempts)

Too many request : message **HTTP code 429**



2. **Brute force login attempts limited to 10 from an IP (Using cache ip : attempts)**

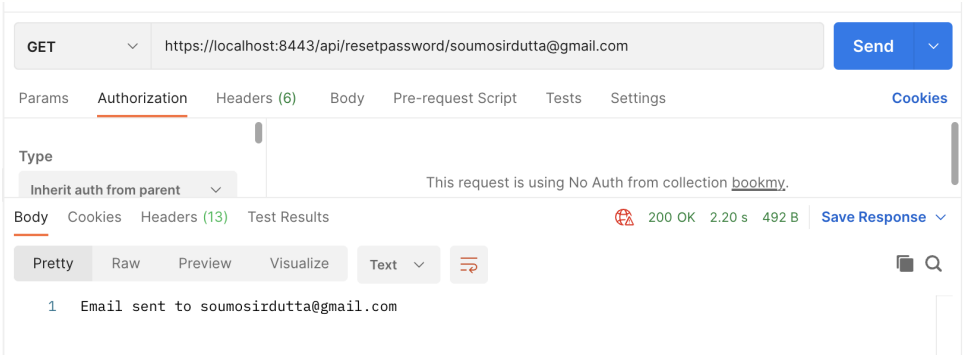


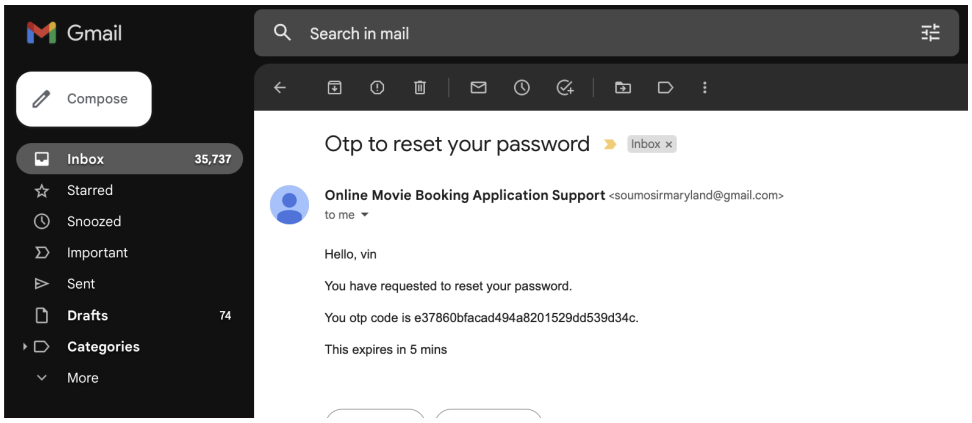
```
2022-11-13 21:14:57.744 ERROR 51987 --- [nio-8443-exec-1] c.s.b.s.CustomAuthenticationFilter      : User on ip 0:0:0:0:0:0:0:1 entered wrong username and password
2022-11-13 21:14:58.369  INFO 51987 --- [nio-8443-exec-2] c.s.b.s.CustomAuthenticationFilter      : Attempting login for Username : jdoe ,
2022-11-13 21:14:58.377  INFO 51987 --- [nio-8443-exec-2] c.s.b.service.UserServiceImplementation  : Ip 0:0:0:0:0:0:0:1 is blocked
2022-11-13 21:14:58.390 ERROR 51987 --- [nio-8443-exec-2] c.s.b.s.CustomAuthenticationFilter      : An internal error occurred while trying to authenticate the user.
```

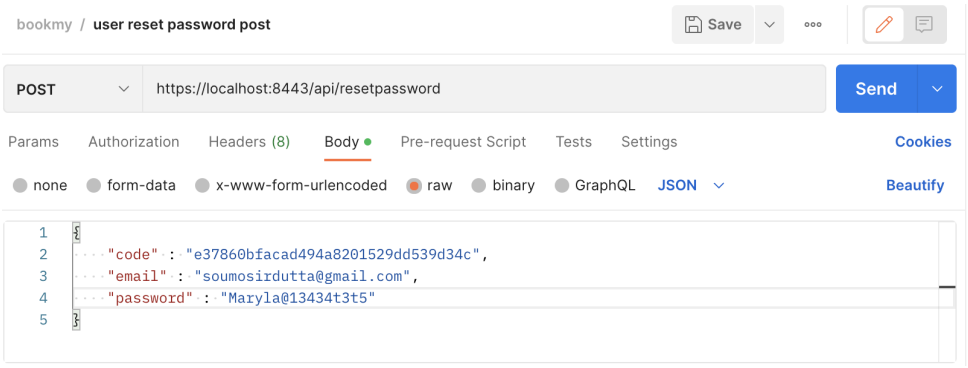
3. **Forgot password uses secret code sent to email address to change password (using cache for this email : code)**

RESET a user password (REQEST)

curl --location --request GET 'https://localhost:8443/api/resetpassword/soumosirdutta@gmail.com'

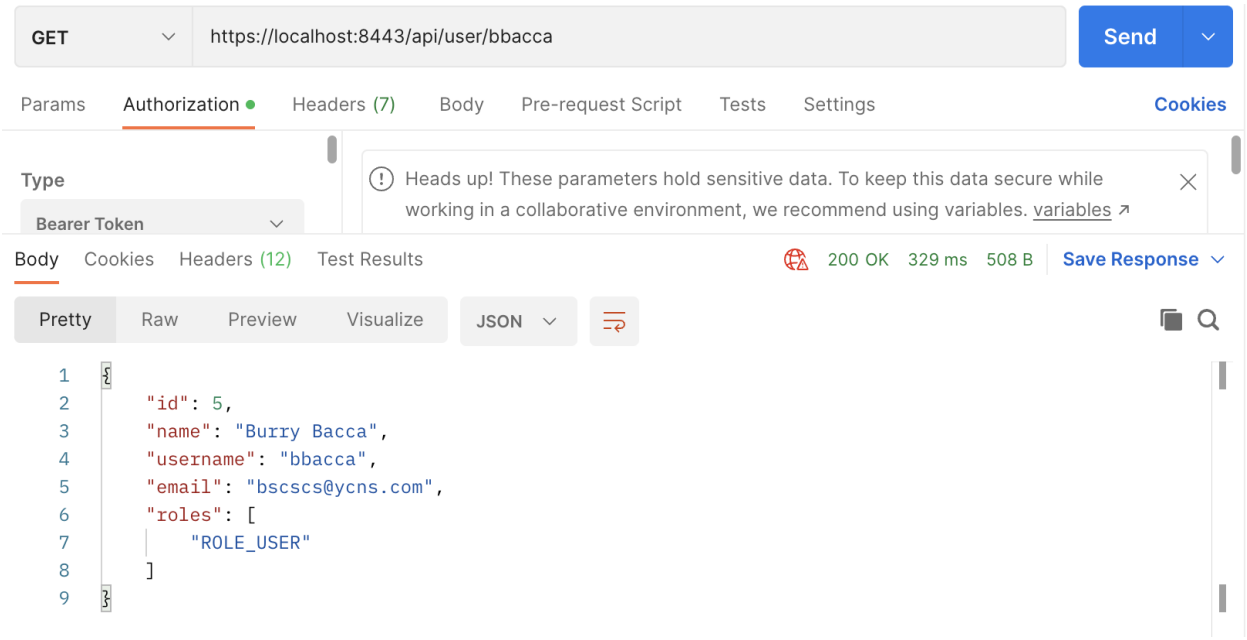


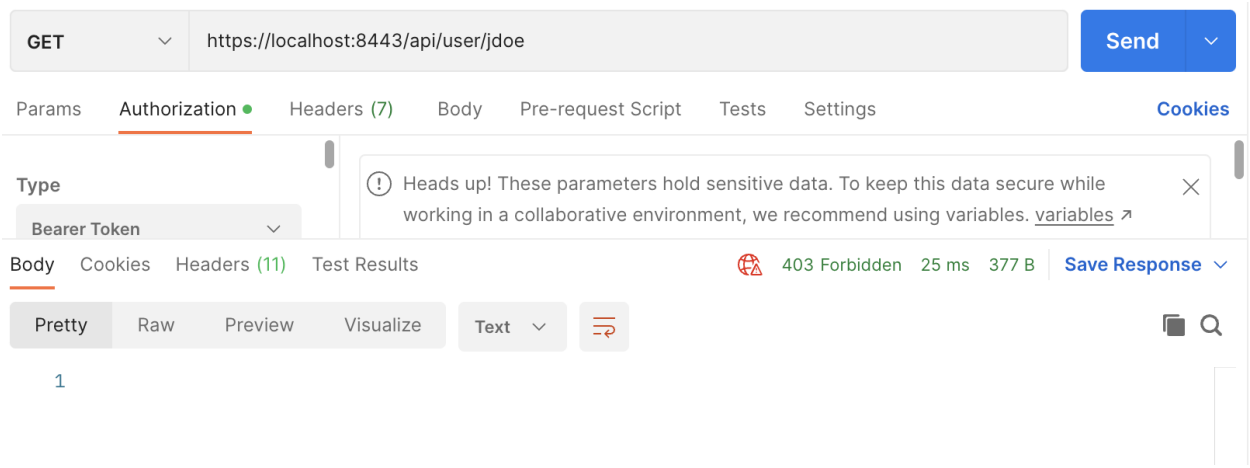




User cannot view other user : -

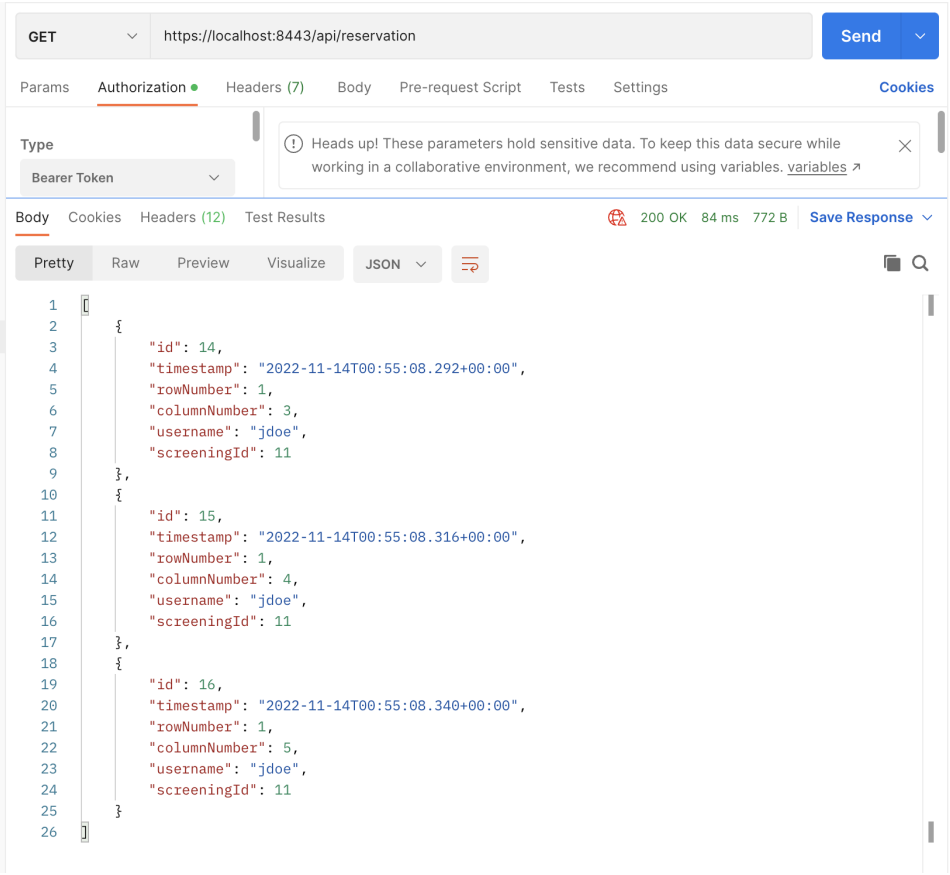
Cannot see/edit/delete user jdoe using bbacca user



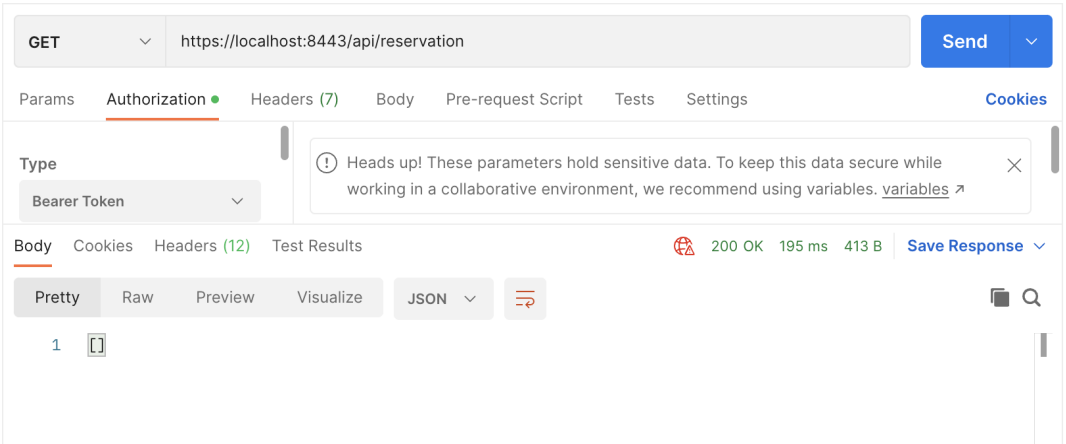


User cannot see/edit/add reservations of other user

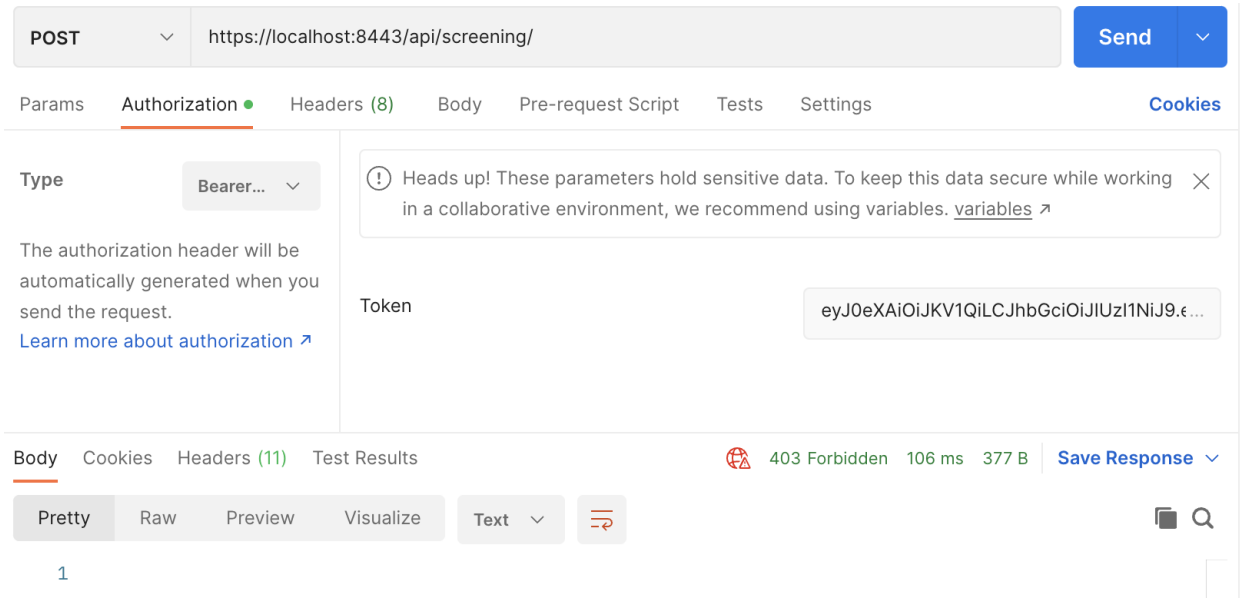
Jdoe ‘s reservation



Bacca’s reservation - same api



Trying to access a Admin route POST api/screening/ using USER_ROLE user



We get forbidden

———————

Secure Communication by SSL certificate

Api listens on https:

References -

https://www.baeldung.com/spring-boot-https-self-signed-certificate

Port given is 8443

Validation

We validation for user email, password, name,
Email - (>3 characters and <30 characters)
Password - (>8 characters and <30 characters)
Atleast 1 upper case character
Atleast 1 lower case character
Atleast 1 special characters
Atleast 1 digit
Name - (>3 characters and <30 characters)
Username - (>3 characters and <30 characters) Alphanumeric character - A-Z, a-z ,0-9

We use model validation for hall name , seat row and seat column
Hall name - >3 characters and <30 characters
Seat row number (1-200)
Seat column number (1-200)

We use string validation for movie title, movie , director and cast (>3 characters and <30 characters)
Title >3 characters and <30 characters
director >3 characters and <30 characters
Cast >3 characters and <30 characters
Description less than 200 charectors

Screening

Starttime - must be a valid time of format YYYY-MM-DDTHH:MM:
2022-12-20T15:20:00.000+00:00
No two movies can screen at the same interval in same hall

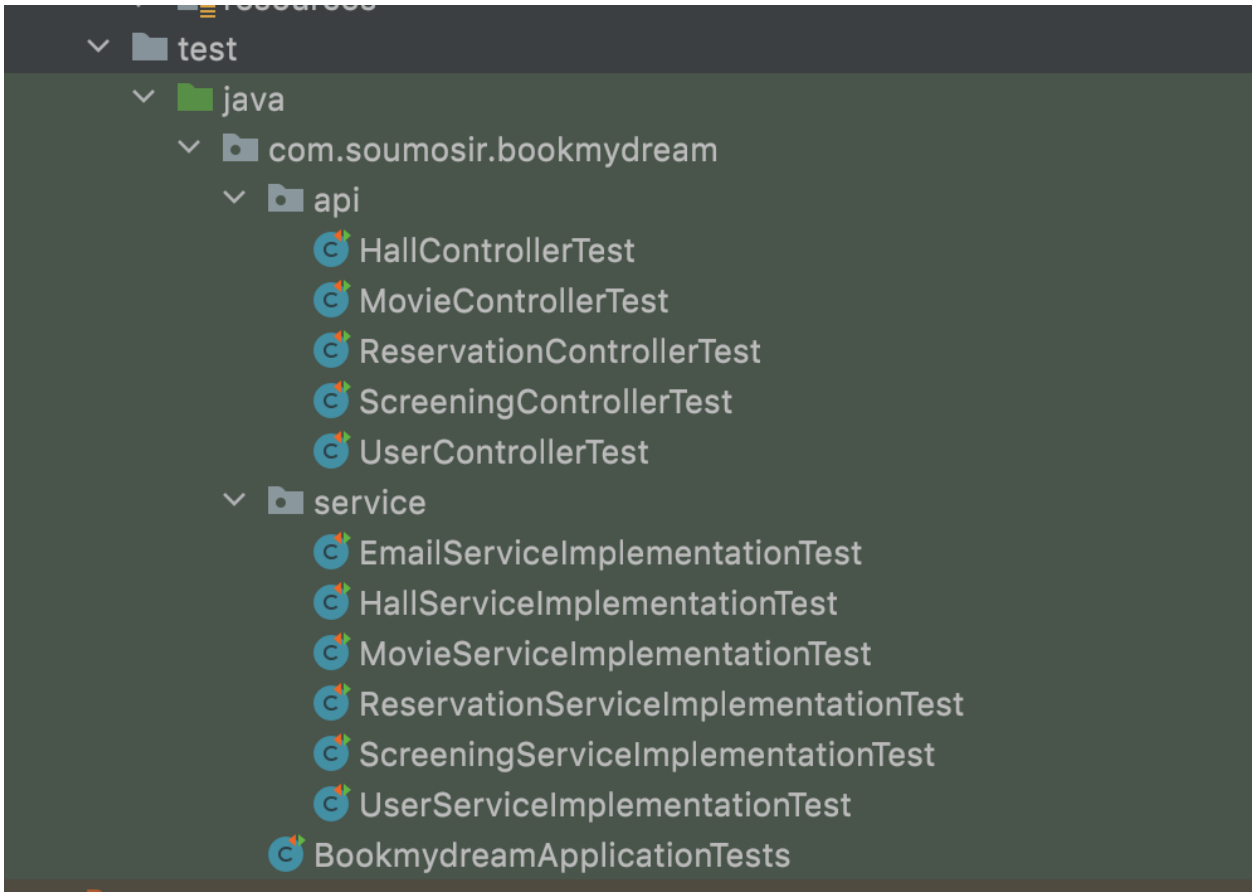
Reservation

No two person can have the same seat for a screening(unique hall and movie for a time)

Running tests:

There are 6 services test classes
5 rest api controller test classes

In src/test/java/com/soumosir/bookmydream



Go to each one and run the file by green button

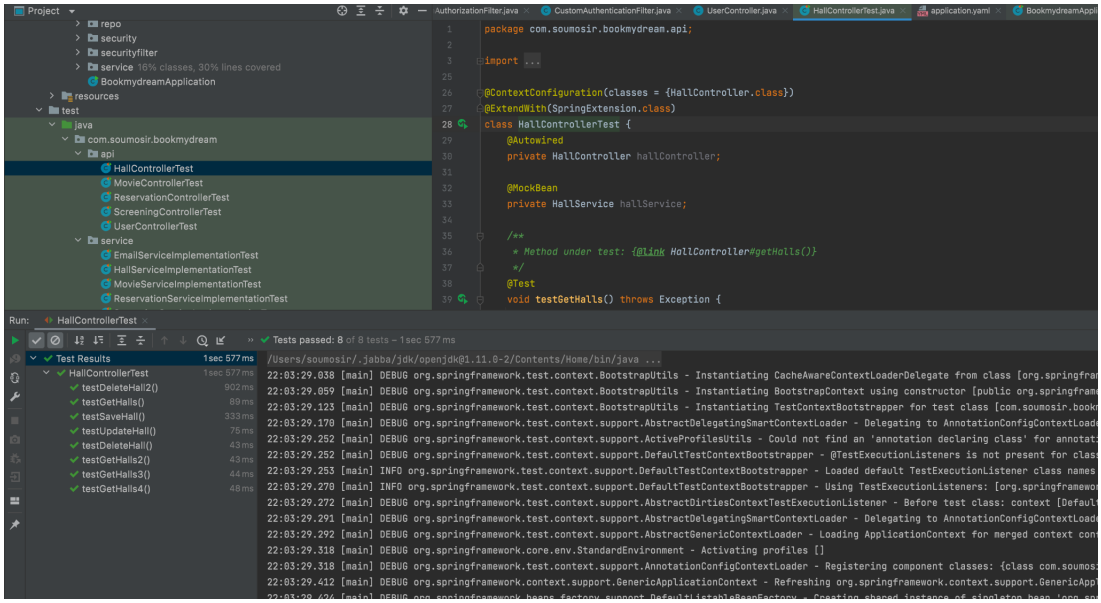
You need to run all the test individually

Example : -

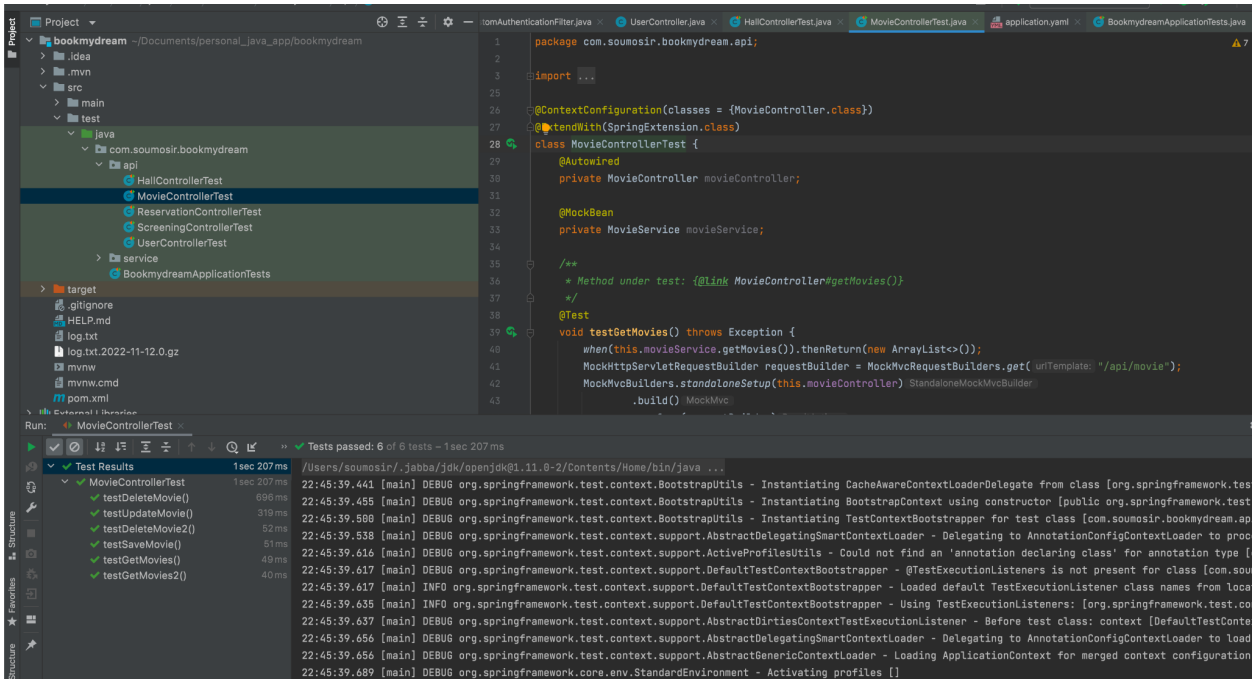
Test result seen

Rest controllers -

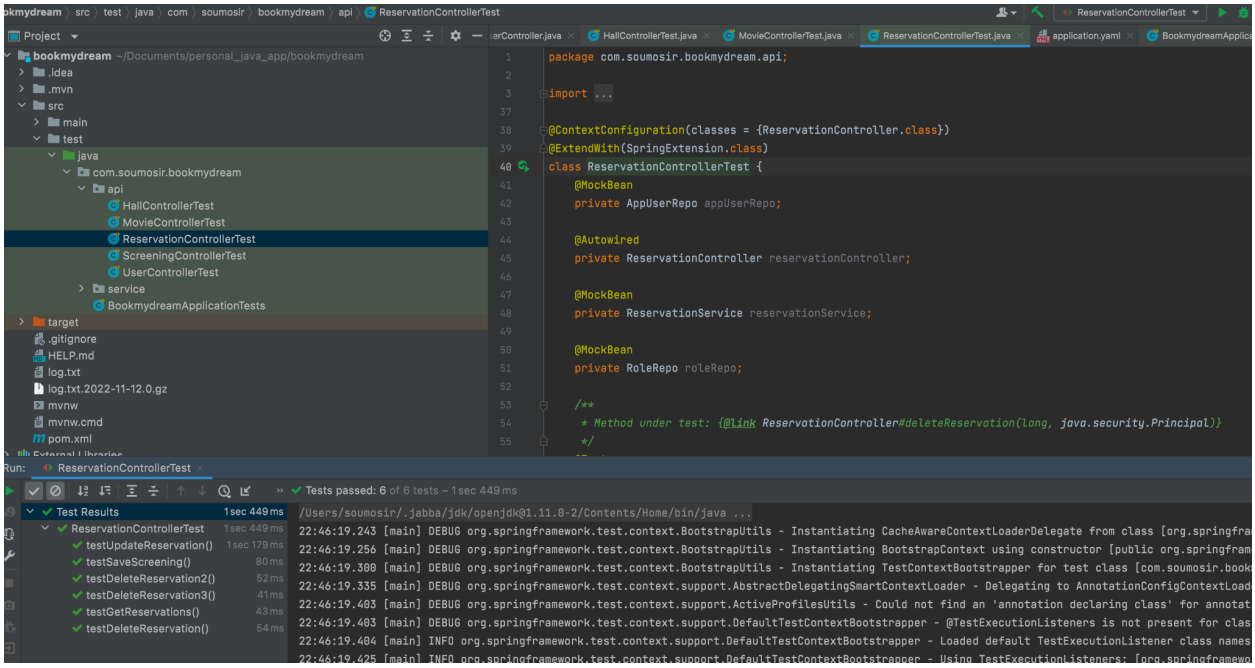
1. **Hall controller 8 tests**



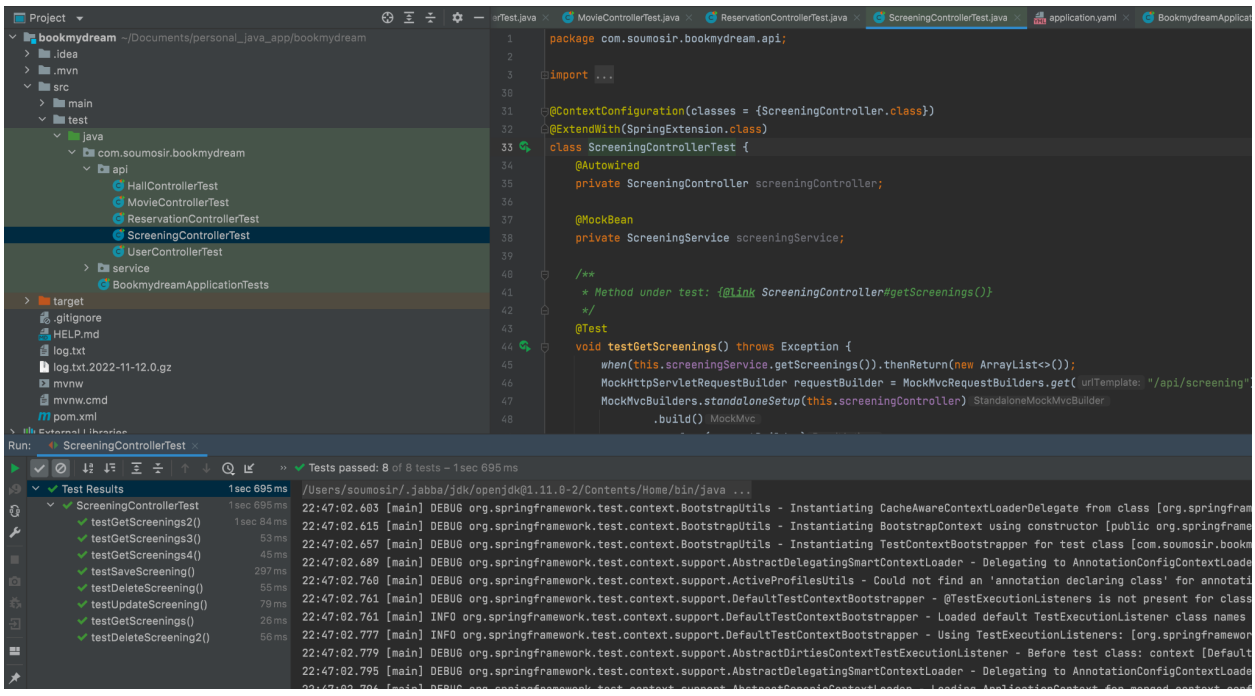
2. **Movie controller 6 tests**



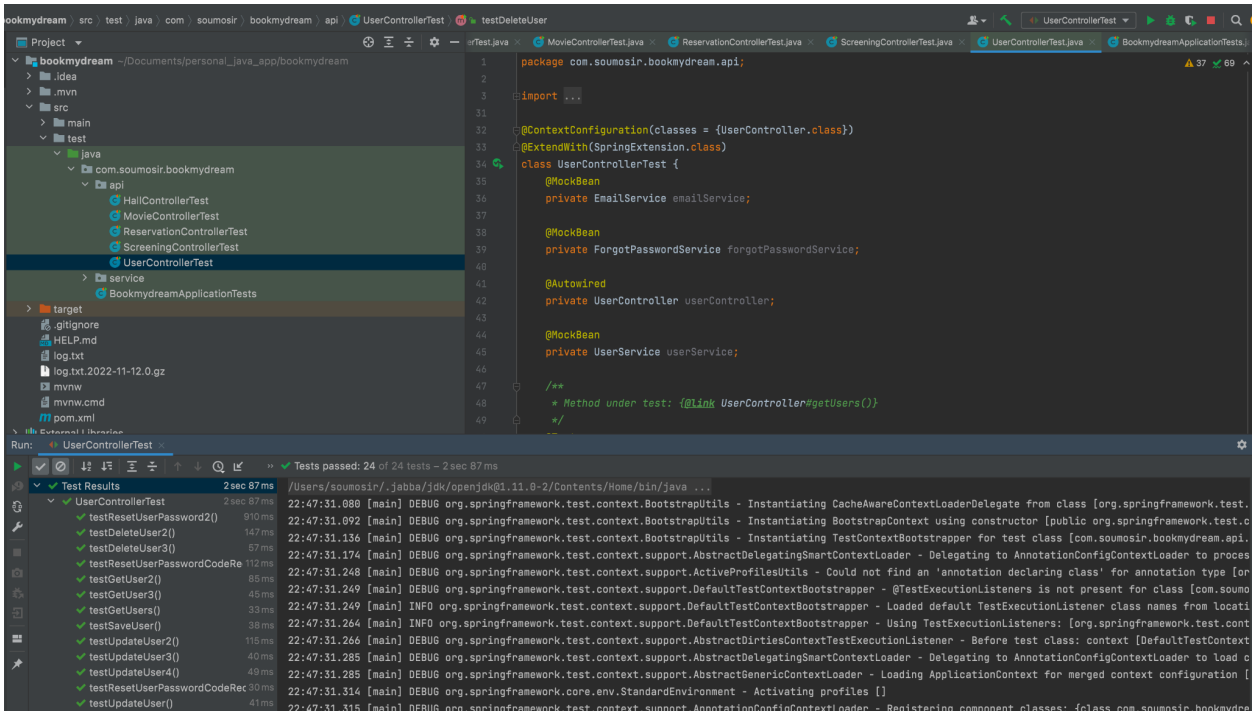
3. reservation controller 8 tests



4. Screening controller 8 test

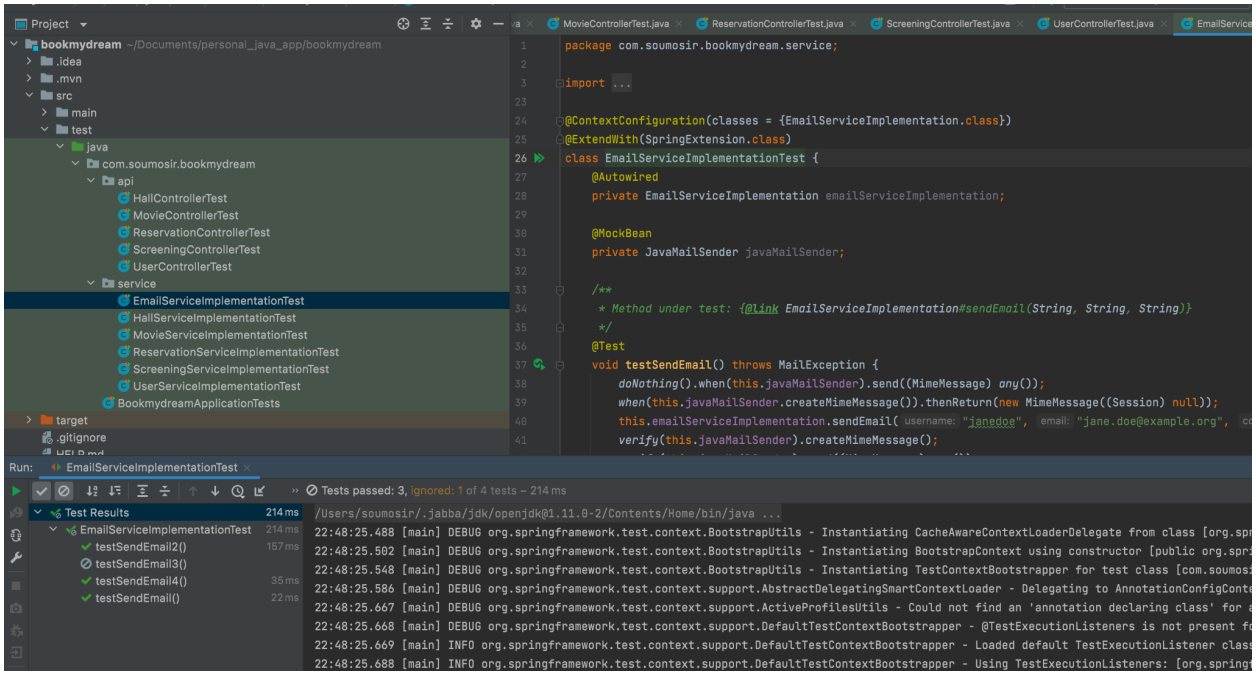


5. User controller - 24 tests

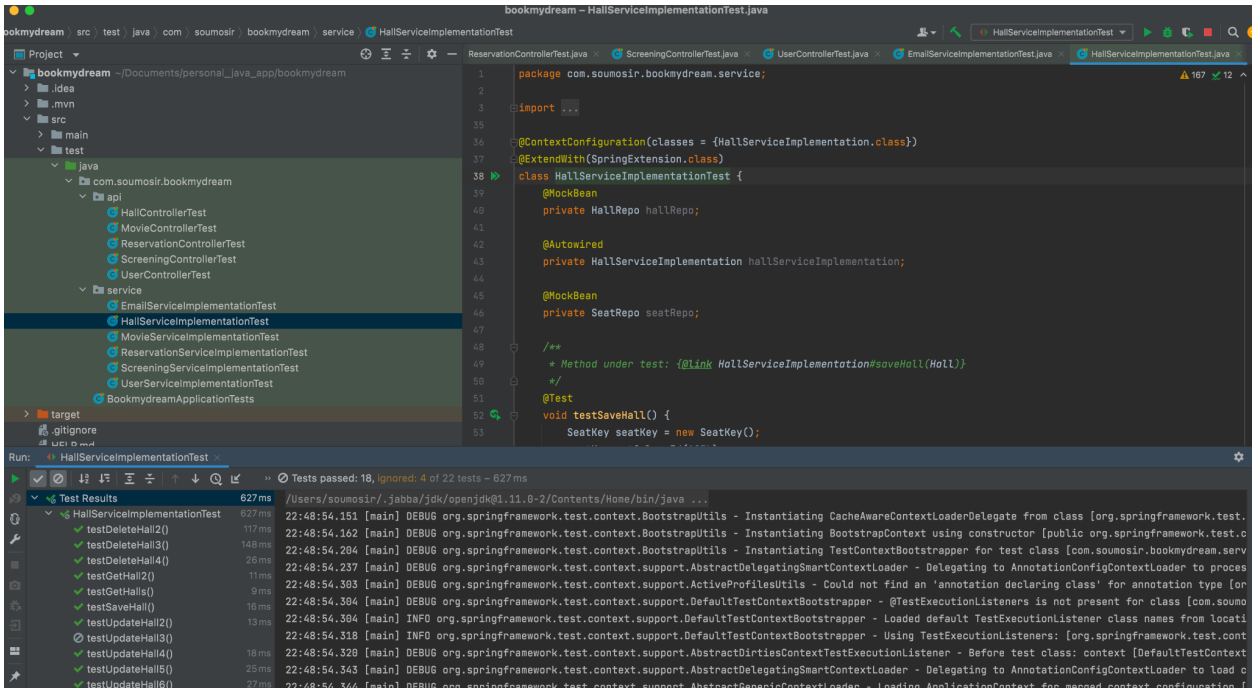


Services

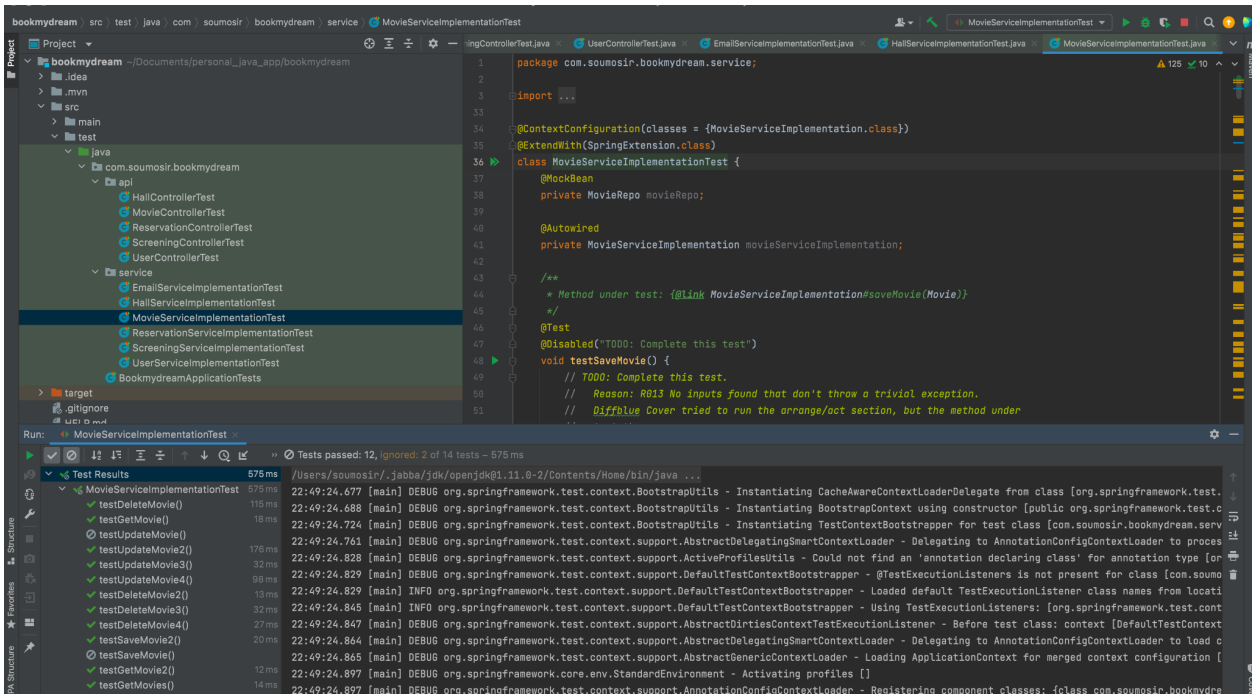
6. Email service 4 tests



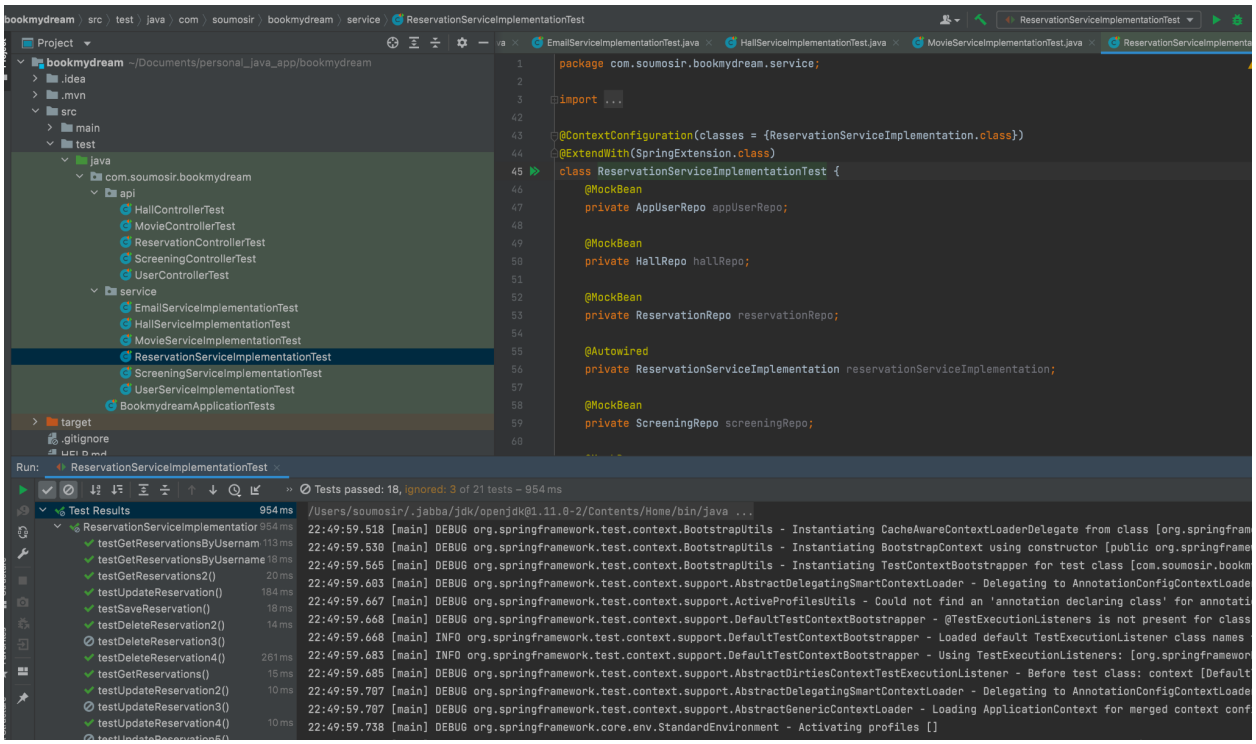
7. Hall service 18 tests



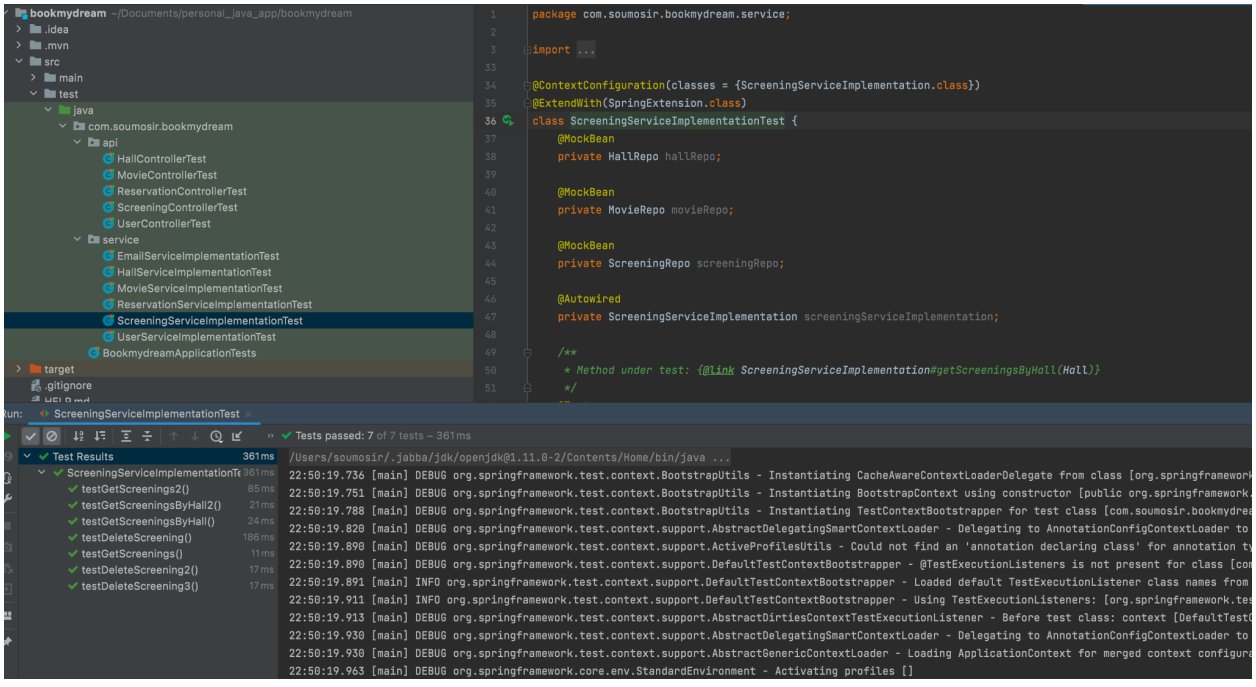
8. Movie Service tests - 12 tests



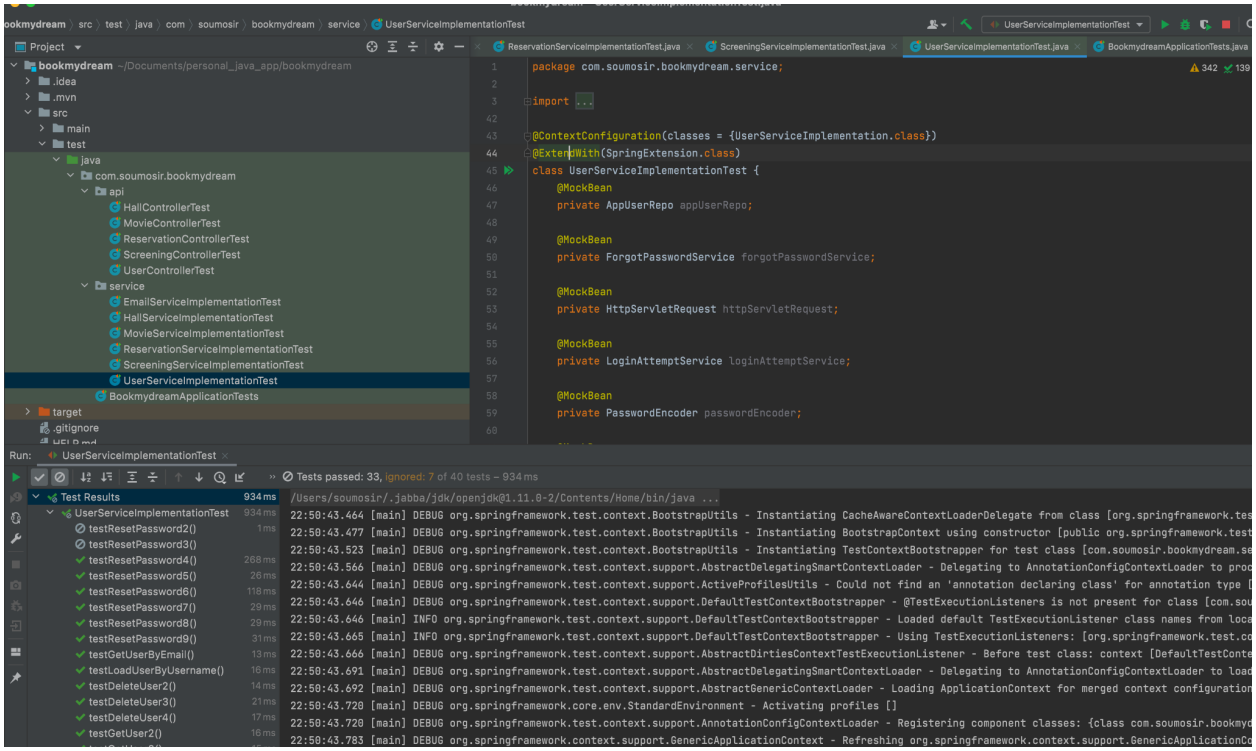
9. Reservation service 18 tests



10. Screening service 7 tests



11. User service tests - 33



References

https://www.baeldung.com/spring-security-block-brute-force-authentication-attempts

https://spring.io/projects/spring-security

https://spring.io/guides/tutorials/rest/