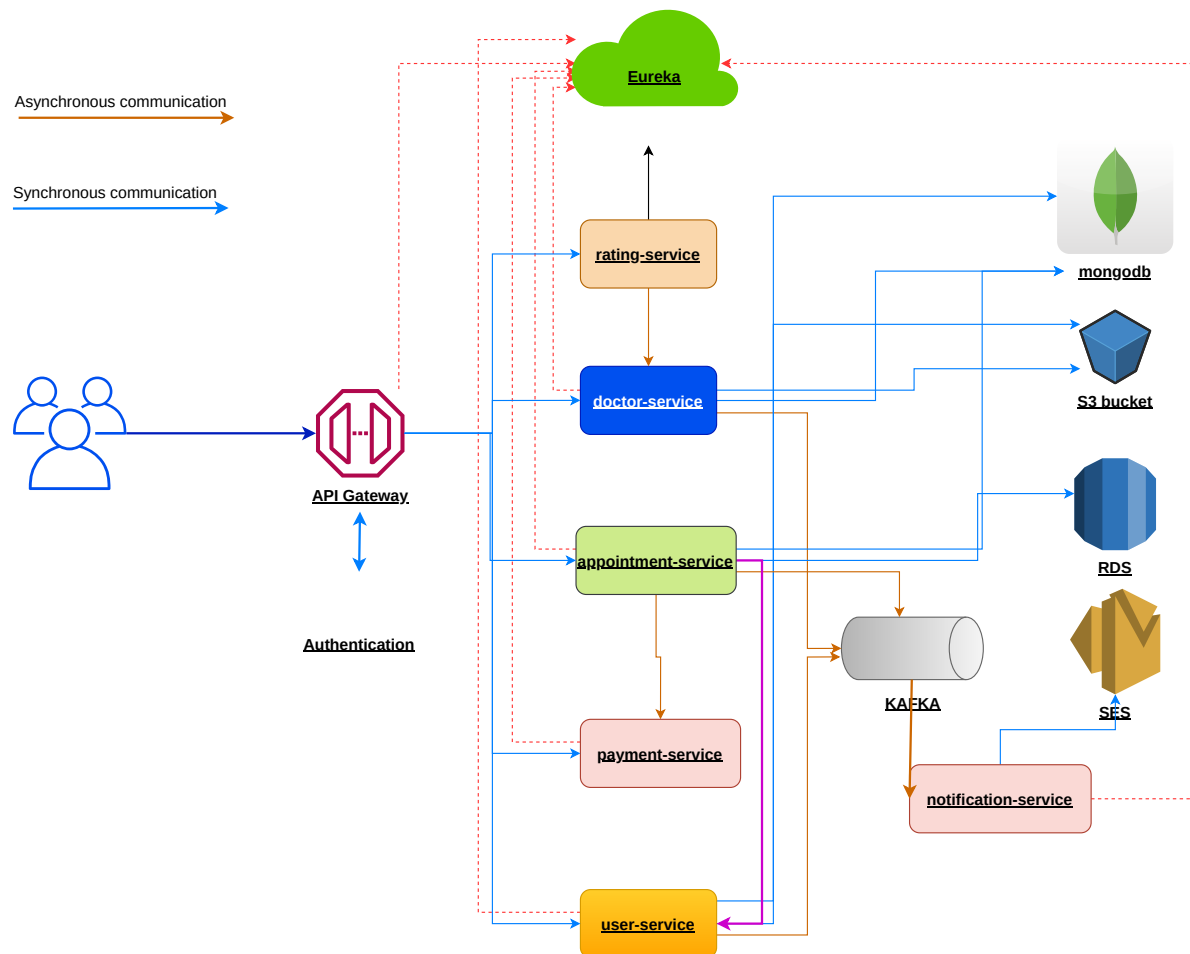


# CodeLogic.pdf

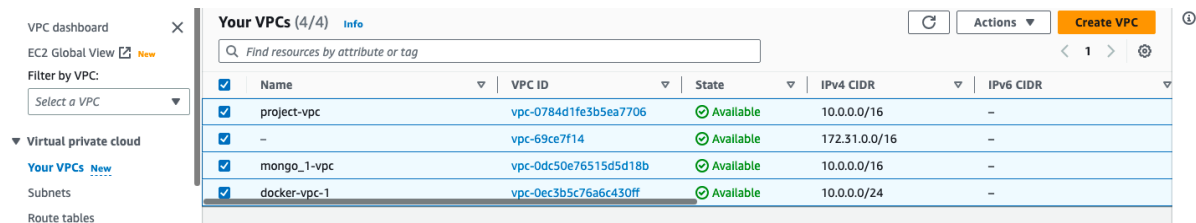
## Architecture



## Setup Required to run application

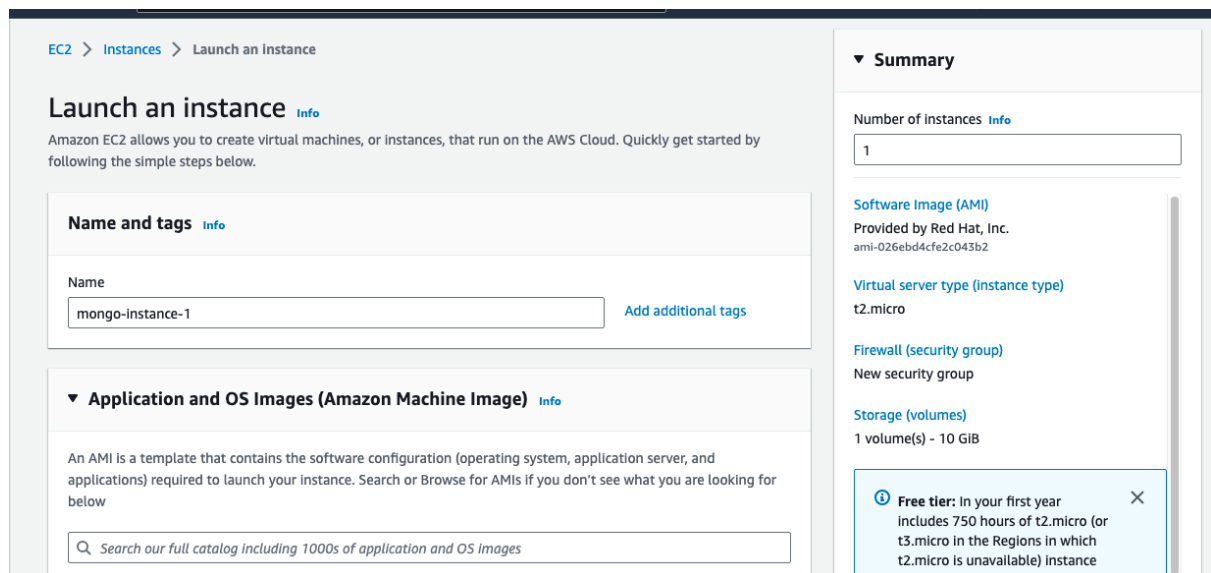
### VPC setup:

- create an `vpc` first before creating the resources in aws so we can use same vpc for all the resources.



## MongoDb setup:

- we create an `ec2` instance of `t2-micro` and install mongodb in that



```
[ec2-user@ip-172-31-33-151 ~]$ systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; preset: disabled)
   Active: active (running) since Wed 2023-08-16 03:56:46 UTC; 2h 28min ago
     Docs: https://docs.mongodb.org/manual
   Process: 927 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
   Process: 930 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
   Process: 931 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
   Process: 935 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Main PID: 965 (mongod)
    Memory: 153.4M
      CPU: 27.907s
    CGroup: /system.slice/mongod.service
           └─965 /usr/bin/mongod -f /etc/mongod.conf

Aug 16 03:56:43 ip-172-31-33-151.ec2.internal systemd[1]: Starting MongoDB Database Server...
Aug 16 03:56:43 ip-172-31-33-151.ec2.internal mongod[935]: about to fork child process, waiting until server is ready for
Aug 16 03:56:43 ip-172-31-33-151.ec2.internal mongod[965]: forked process: 965
Aug 16 03:56:46 ip-172-31-33-151.ec2.internal mongod[935]: child process started successfully, parent exiting
Aug 16 03:56:46 ip-172-31-33-151.ec2.internal systemd[1]: Started MongoDB Database Server.
lines 1-19/19 (END)
```

## Change security group to accept external request for mongo port 27017

VPC > Security Groups > sg-0ef056323e77a64d2 - launch-wizard-14 > Edit inbound rules

### Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

#### Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	
sg-r0741c4f511829afb	Custom TCP	TCP	27017	Custom <input type="text" value="0.0.0.0"/>	<input type="text"/>	Delete
sg-r0f426f6a7dcdf4ba	SSH	TCP	22	Custom <input type="text" value="0.0.0.0"/>	<input type="text"/>	Delete

Add rule

Cancel Preview changes Save rules

## RDS setup:

- we required relational database for running our application so set we setup **DynamoDb** **rds** instance

How was your experience creating an Amazon RDS database? [Provide feedback](#)

RDS > Databases > bmc-db-1

## bmc-db-1

[Modify](#) [Actions](#)

### Summary

DB identifier bmc-db-1	CPU -	Status Available	Class db.t3.micro
Role Instance	Current activity	Engine MySQL Community	Region & AZ us-east-1a

[Connectivity & security](#) [Monitoring](#) [Logs & events](#) [Configuration](#) [Maintenance & backups](#) [Tags](#)

### Connectivity & security

<b>Endpoint &amp; port</b> Endpoint bmc-db-1.cb9zqmw5buxr.us-east-1.rds.amazonaws.com Port	<b>Networking</b> Availability Zone us-east-1a VPC vpc-69ce7f14	<b>Security</b> VPC security groups <a href="#">launch-wizard-14 (sg-0ef056323e77a64d2)</a> Active Publicly accessible
---	---	--

## Change security group to accept external request for **rds** port **3306**

VPC > Security Groups > sg-0ef056323e77a64d2 - launch-wizard-14 > Edit inbound rules

## Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

### Inbound rules

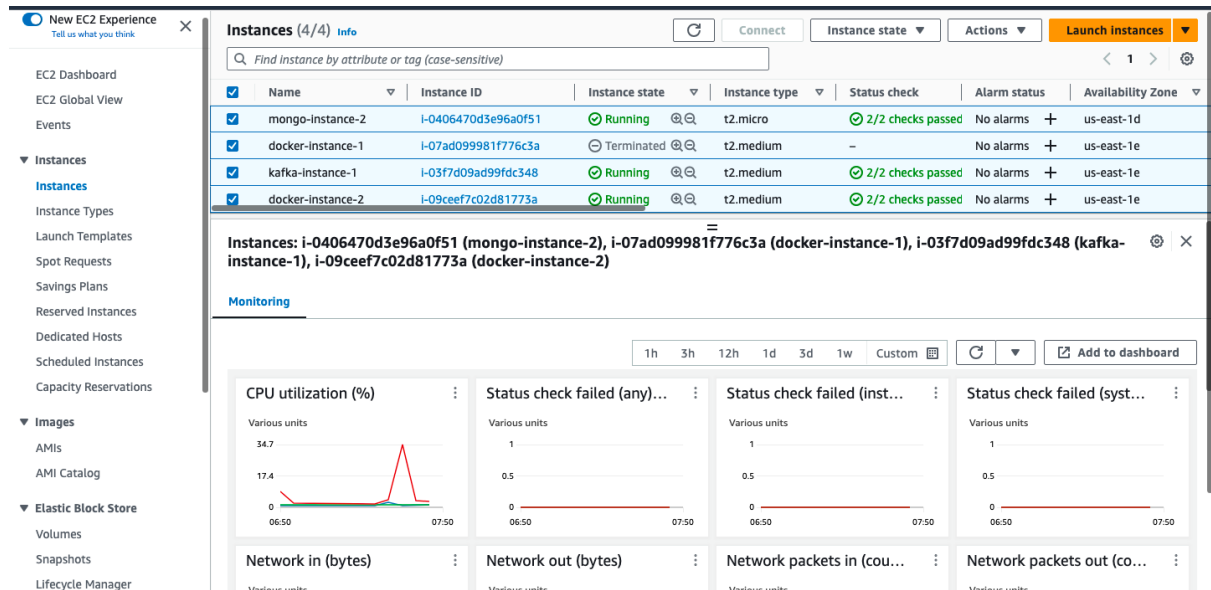
Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sg-0741c4f511829afbf	Custom TCP	TCP	27017	Custom 0.0.0.0/0		Delete
sg-0b97d308ffc96d6fb	MySQL/Aurora	TCP	3306	Custom 0.0.0.0/0		Delete
sg-0f426f6a7dcfd4ba	SSH	TCP	22	Custom 0.0.0.0/0		Delete

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

# Kafka Setup

- we are using external kafka server for our application.
- so created one more `ec2` instance of `t2-medium` and installed kafka there



start kafka from binaries

```
[ec2-user@ip-172-31-48-24 kafka_2.12-3.5.1]$ ./bin/zookeeper-server-start.sh ./config/zookeeper.properties ^C
[ec2-user@ip-172-31-48-24 kafka_2.12-3.5.1]$ ./bin/kafka-server-start.sh ./config/server.properties
```

edit security group to access kafka from external open port `9092`

VPC > Security Groups > sg-0ef056323e77a64d2 - launch-wizard-14 > Edit inbound rules

## Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules** Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	
sg-r-0741c4f511829afb	Custom TCP	TCP	27017	Custom <input type="text" value="0.0.0.0/0"/>		Delete
sg-r-020a685bb83ce6145	Custom TCP	TCP	9092	Custom <input type="text" value="0.0.0.0/0"/>		Delete
sg-r-0b3512c925df27f6e	Custom TCP	TCP	8761	Custom <input type="text" value="0.0.0.0/0"/>		Delete
sg-r-0b97d308ffc96d6fb	MySQL/Aurora	TCP	3306	Custom <input type="text" value="0.0.0.0/0"/>		Delete
sg-r-08a50b92fcb21c0de	Custom TCP	TCP	180 - 8096	Custom <input type="text" value="0.0.0.0/0"/>		Delete
sg-r-0f426f6a7dcfd4ba	SSH	TCP	22	Custom <input type="text" value="0.0.0.0/0"/>		Delete

## Docker setup

- to setup docker we created a one more instance with `t2-medium`
- then installed docker

New EC2 Experience

EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Instances (4/4) Info

Connect

Instance state

Actions

Launch instances

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	mongo-instance-2	i-0406470d3e96a0f51	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d
<input checked="" type="checkbox"/>	docker-instance-1	i-07ad099981f776c3a	Terminated	t2.medium	-	No alarms	us-east-1e
<input checked="" type="checkbox"/>	kafka-instance-1	i-03f7d09ad99fdc348	Running	t2.medium	2/2 checks passed	No alarms	us-east-1e
<input checked="" type="checkbox"/>	docker-instance-2	i-09ceef7c02d81773a	Running	t2.medium	2/2 checks passed	No alarms	us-east-1e

Instances: i-0406470d3e96a0f51 (mongo-instance-2), i-07ad099981f776c3a (docker-instance-1), i-03f7d09ad99fdc348 (kafka-instance-1), i-09ceef7c02d81773a (docker-instance-2)

Monitoring

1h 3h 12h 1d 3d 1w Custom

CPU utilization (%)

Status check failed (any)...

Status check failed (inst...)

Status check failed (syst...)

CodeLogic.pdf

6

```

ubuntu@ip-172-31-58-233:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-08-16 05:02:23 UTC; 2h 55min ago
 TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 10396 (dockerd)
     Tasks: 145
    Memory: 312.6M
   CGroup: /system.slice/docker.service
           └─10396 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
           └─34802 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8761 -container-ip 172.19.0.2 -contain>
           └─34808 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 8761 -container-ip 172.19.0.2 -contain>
           └─34899 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8086 -container-ip 172.19.0.3 -co>
           └─34907 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 8086 -container-ip 172.19.0.3 -contain>
           └─34929 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8085 -container-ip 172.19.0.4 -co>
           └─34940 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 8085 -container-ip 172.19.0.4 -contain>

```

- make sure `rds` , `mongodb` and `kafka` services must be accessible from `docker instance`

## IAM user setup for SES and S3 access

- create an `IAM` user with full access to `SES` and `S3`
- create a `Access Key` for that user and save the `Access Key` and `Password`
- This is required to be passed to the application while running

ses-smtp-user.20230815-134044 [Info](#)

Delete

### Summary

ARN  
 arn:aws:iam::063745262440:user/ses-smtp-user.20230815-134044

Console access  
Disabled

Access key 1  
AKIAQ5V4DQNUG6UCGY4P - Active  
 Never used. 17 hours old.

Created  
August 15, 2023, 13:44 (UTC+05:30)

Last console sign-in  
-

Access key 2  
[Create access key](#)

[Permissions](#) | [Groups](#) | [Tags \(3\)](#) | [Security credentials](#) | [Access Advisor](#)

### Permissions policies (3)

Permissions are defined by policies attached to the user directly or through groups.



Remove

Add permissions ▼

Filter by Type  
 All types ▼ < 1 > ⚙️

<input type="checkbox"/>	Policy name	Type	Attached via
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	Directly
<input type="checkbox"/>	AmazonSESEFullAccess	AWS managed	Directly

## open ports for external access to access the apis

→ ports **8761** for eureka service to run

→ ports **8080** to **8090** for applications to expose the services

[VPC](#) > [Security Groups](#) > [sg-0ef056323e77a64d2 - launch-wizard-14](#) > Edit inbound rules

### Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

#### Inbound rules [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-0741c4f511829afb	Custom TCP ▼	TCP	27017	Custom ▼ <input type="text" value="0.0.0.0"/>	<input type="text" value=""/>	Delete
sgr-020a685bb83ce6145	Custom TCP ▼	TCP	9092	Custom ▼ <input type="text" value="0.0.0.0"/>	<input type="text" value=""/>	Delete
sgr-0b3512c925df27f6e	Custom TCP ▼	TCP	8761	Custom ▼ <input type="text" value="0.0.0.0"/>	<input type="text" value=""/>	Delete
sgr-0b97d308ffc96d6fb	MySQL/Aurora ▼	TCP	3306	Custom ▼ <input type="text" value="0.0.0.0"/>	<input type="text" value=""/>	Delete
sgr-08a50b92fcb21c0de	Custom TCP ▼	TCP	<b>8080 - 8090</b>	Custom ▼ <input type="text" value="0.0.0.0"/>	<input type="text" value=""/>	Delete
sgr-0f426f6a7dcdf4ba	SSH ▼	TCP	22	Custom ▼ <input type="text" value="0.0.0.0"/>	<input type="text" value=""/>	Delete



# Running an application

## *steps to run the application:*

1. upload the `BookMyConsultation.zip` to the `docker instance`
2. unzip the file and navigate to the file
3. before running need to give parameter to the application to start

Only `docker-compose.yml` file need to be changed

1. `mongo host` and `port` changes
  - replace `MONGO_HOST` value with `mongo running instance id`
  - replace `MONGO_PORT` value with port where mongo is running in mongo host
  - replace `MONGO_DATABASE` with mongo db name
2. `MYSQL RDS` changes.
  - replace `MYSQL_HOST` value with `rds public IP`
  - replace `MYSQL_HOST_PORT` value with mysql port set by you
  - replace `MYSQL_USER` value with admin user details set for rds
  - replace `MYSQL_PASSWORD` value with admin user password set for rds
  - replace `MySQL_DATABASE` value with database name
3. `KAFKA` changes
  - replace `KAFKA_HOST` value with kafka instance ip where kafka can be accessed from.
  - replace `KAFKA_HOST_PORT` value with kafka port value set by you
4. `SES` and `S3 access key` and `secret key`
  - replace values of `SES_ACCESSKEY` and `S3_ACCESS_KEY` with the token access token name generated

- replace values of `SES_SECRETKEY` and `S3_SECRET_KEY` with the password generated for the access token
- replace the `SES_FROM_EMAIL` with the email from which emails need to be sent this should be verified email.

4. After properly replacing all the values we are ready to run the application.
5. execute command `sudo docker compose up`
6. it will references the `Dockerfile` present in each services home folder and build the image one-by one and start the containers.

```

ubuntu@ip-172-31-58-233:~/BMC_RUNS/BookMyConsultation$ sudo docker compose up
[+] Running 8/8
! usersvc Warning                                0.1s
! appointmentsvc Warning                         0.2s
! paymentsvc Warning                             0.1s
! ratingsvc Warning                             0.3s
! notificationsvc Warning                       0.1s
! gatewaysvc Warning                            0.2s
! eurekasvc Warning                             0.2s
! bookingsvc Warning                            0.1s
[+] Building 1.7s (70/70) FINISHED
=> [eurekasvc internal] load build definition from Dockerfile           0.0s
=> => transferring dockerfile: 422B                                     0.0s
=> [eurekasvc internal] load .dockerignore                             0.0s
=> => transferring context: 2B                                           0.0s
=> [paymentsvc internal] load metadata for docker.io/library/openjdk:14-jdk-alpine 0.6s
=> [paymentsvc internal] load metadata for docker.io/library/maven:3.6.3-jdk-14    0.6s
=> [paymentsvc builder 1/5] FROM docker.io/library/maven:3.6.3-jdk-14@sha256:aea55d01edac9df38eb17d19f6aabb55263f 0.0s
=> [paymentsvc stage-1 1/3] FROM docker.io/library/openjdk:14-jdk-alpine@sha256:b8082268ef46d44ec70fd5a64c71d4454 0.0s
=> [eurekasvc internal] load build context                               0.0s
=> => transferring context: 3.54kB                                       0.0s
=> CACHED [paymentsvc stage-1 2/3] WORKDIR /usr/app                    0.0s

```

you can see applications coming up one by one



The screenshot shows the Spring Eureka dashboard in a web browser. The browser's address bar shows the URL `35.175.130.200:8761`. The dashboard has a dark header with the 'spring Eureka' logo and navigation links for 'HOME' and 'LAST 1000 SINCE STARTUP'.

**System Status**

Environment	N/A	Current time	2023-08-16T09:32:37 +0000
Data center	N/A	Uptime	01:55
		Lease expiration enabled	true
		Renews threshold	11
		Renews (last min)	12

**DS Replicas**

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - a9daad1be3dc:API-GATEWAY:9191
APPOINTMENT-SERVICE	n/a (1)	(1)	UP (1) - 9ff42c1feadc:APPOINTMENT-SERVICE:8083
DOCTOR-SERVICE	n/a (1)	(1)	UP (1) - 265b69395895:DOCTOR-SERVICE:8081
PAYMENT-SERVICE	n/a (1)	(1)	UP (1) - 69e69e45d825:PAYMENT-SERVICE:8084
RATING-SERVICE	n/a (1)	(1)	UP (1) - 539673c63cfb:RATING-SERVICE:8085
USER-SERVICE	n/a (1)	(1)	UP (1) - 8d2be22dfd61:USER-SERVICE:8082

**General Info**

to bring down the services `sudo docker compose down`

## API test screen shots

### 1. Doctor-Onboarding Service

**Endpoint 1:** URI: `/doctors`

This endpoint is responsible for collecting information about the doctor.

- Endpoint is forbidden, for roles other than User, Admin

BookMyConsultation / doctor-service / Doctor Registration

POST http://54.197.76.252:8081/doctors

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "firstName": "Mitra",
3   ... "lastName": "s",
4   ... "dob": "2020-07-04",
5   ... "emailId": "mitra065@gmail.com",
6   ... "mobile": "1235754281",
7   ... "pan": "GBCPS4135T"
8 }
```

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2023-08-12T16:10:32.170+00:00",
3   "status": 403,
4   "error": "Forbidden",
5   "message": "",
6   "path": "/doctors"
7 }
```

- Endpoint is forbidden, for roles other than User, Admin

BookMyConsultation / doctor-service / Doctor Registration

POST http://54.197.76.252:8081/doctors

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "firstName": "Mitra",
3   ... "lastName": "s",
4   ... "dob": "2020-04",
5   ... "emailId": "mitra065@gmail",
6   ... "mobile": "123575281",
7   ... "pan": "GBCPS135T"
8 }
```

Body Cookies (1) Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "errorMessage": "Invalid input. Parameter Name:",
3   "errorCode": "ERR_INVALID_INPUT",
4   "errorFields": [
5     "Mobile",
6     "PAN",
7     "Date Of Birth"
8   ]
9 }
```

- Success Scenario

The screenshot shows a REST client interface for a service named "BookMyConsultation / doctor-service / Doctor Registration". The request is a POST to the URL "http:// 54.197.76.252:8081/doctors". The request body is in JSON format and contains the following data:

```

1  {
2    "firstName": "Mitra",
3    "lastName": "s",
4    "dob": "2020-07-04",
5    "emailId": "mitra065@gmail.com",
6    "mobile": "1235754281",
7    "pan": "GBCPS4135T"
8  }

```

The response is also in JSON format and contains the following data:

```

1  {
2    "id": "7fbe9cf7-bcee-4408-b9f4-4dcd4850cc64",
3    "firstName": "Mitra",
4    "lastName": "s",
5    "mobile": "1235754281",
6    "dob": "2020-07-04",
7    "emailId": "mitra065@gmail.com",
8    "pan": "GBCPS4135T",
9    "specialization": "GENERAL_PHYSICIAN",
10   "status": "Pending",
11   "registrationDate": "12-08-2023"
12 }

```

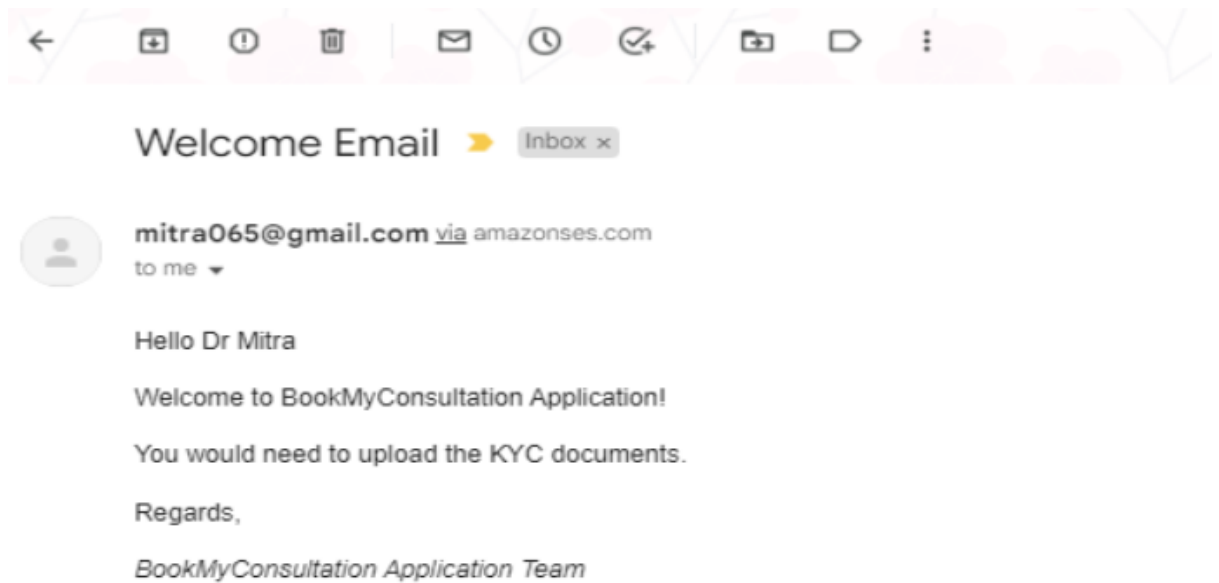
Kafka consumer logs

```

{
  "id" : "7fbe9cf7-bcee-4408-b9f4-4dcd4850cc64",
  "firstName" : "Mitra",
  "lastName" : "s",
  "mobile" : "1235754281",
  "dob" : "2020-07-04",
  "emailId" : "mitra065@gmail.com",
  "pan" : "GBCPS4135T",
  "specialization" : "GENERAL_PHYSICIAN",
  "status" : "Pending",
  "registrationDate" : "12-08-2023",
  "approvedBy" : "Pending",
  "approverComments" : "Pending",
  "verificationDate" : "Pending",
  "rating" : 0
}

```

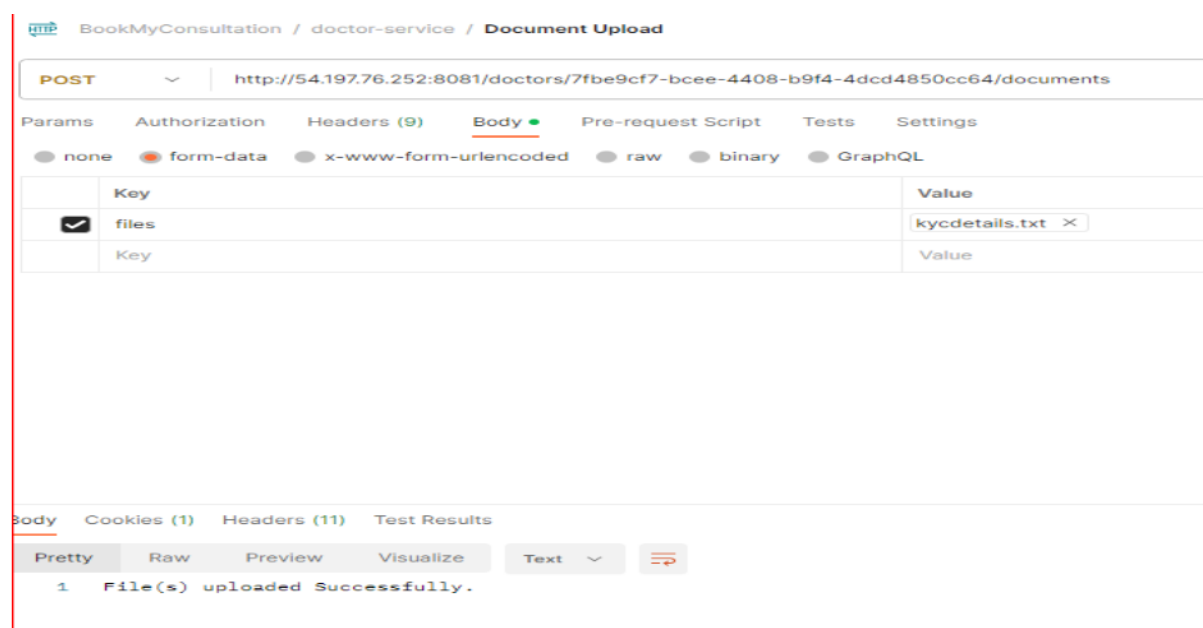
- Verification mail Sent to Doctor



**Endpoint 2: URI:** `/doctors/{doctorId}/document`

This endpoint is responsible for uploading the documents to an S3 bucket by the doctor.

- Success Scenario – Document Uploaded Successfully



- Endpoint Forbidden with authorisation header

BookMyConsultation / doctor-service / Document Upload

POST http://54.197.76.252:8081/doctors/7f9cf7-bcee-4408-b9f4-4dcd4850cc64/documents

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Key	Value	Des
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>	
<input checked="" type="checkbox"/> Content-Type	multipart/form-data; boundary=<calculated when request is sent>	
<input checked="" type="checkbox"/> Content-Length	<calculated when request is sent>	
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.32.3	
<input checked="" type="checkbox"/> Accept	/*/*	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiU9.eyJzdWIiOiJ0aXRyYSIsImF1dG8iOiJhbmVzIiwiaWF0IjoiMTY5MjM0ODUyIn0=	
Key	Value	Des

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize JSON Copy

```

1 {
2   "timestamp": "2023-08-12T16:21:01.217+00:00",
3   "status": 403,
4   "error": "Forbidden",
5   "message": "",
6   "path": "/doctors/7f9cf7-bcee-4408-b9f4-4dcd4850cc64/documents"
7 }
```

- Files Uploaded in S3 bucket

Amazon S3 > Buckets > upgradbmc > 7f9cf7-bcee-4408-b9f4-4dcd4850cc64/

7f9cf7-bcee-4408-b9f4-4dcd4850cc64/ Copy S3

Objects Properties

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
kycdetails.txt	txt	August 12, 2023, 21:46:46 (UTC+05:30)	0 B	Standard

Endpoint 3: URI: /doctors/{doctorId}/approve



## Endpoint Forbidden for unauthorised users

The screenshot shows a REST client interface for a project named "BookMyConsultation / doctor-service / Approve the doctors registration". The request is a PUT to the URL `http://54.197.76.252:8081/doctors/7fbe9cf7-bcee-4408-b9f4-4dcd4850cc64/approve`. The request body is JSON: `{ "approvedBy": "Mittra", "approverComments": "Verified" }`. The response is also JSON, showing a "timestamp", "status": 403, "error": "Forbidden", "message": "", and "path".

BookMyConsultation / doctor-service / Approve the doctors registration

PUT `http://54.197.76.252:8081/doctors/7fbe9cf7-bcee-4408-b9f4-4dcd4850cc64/approve`

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

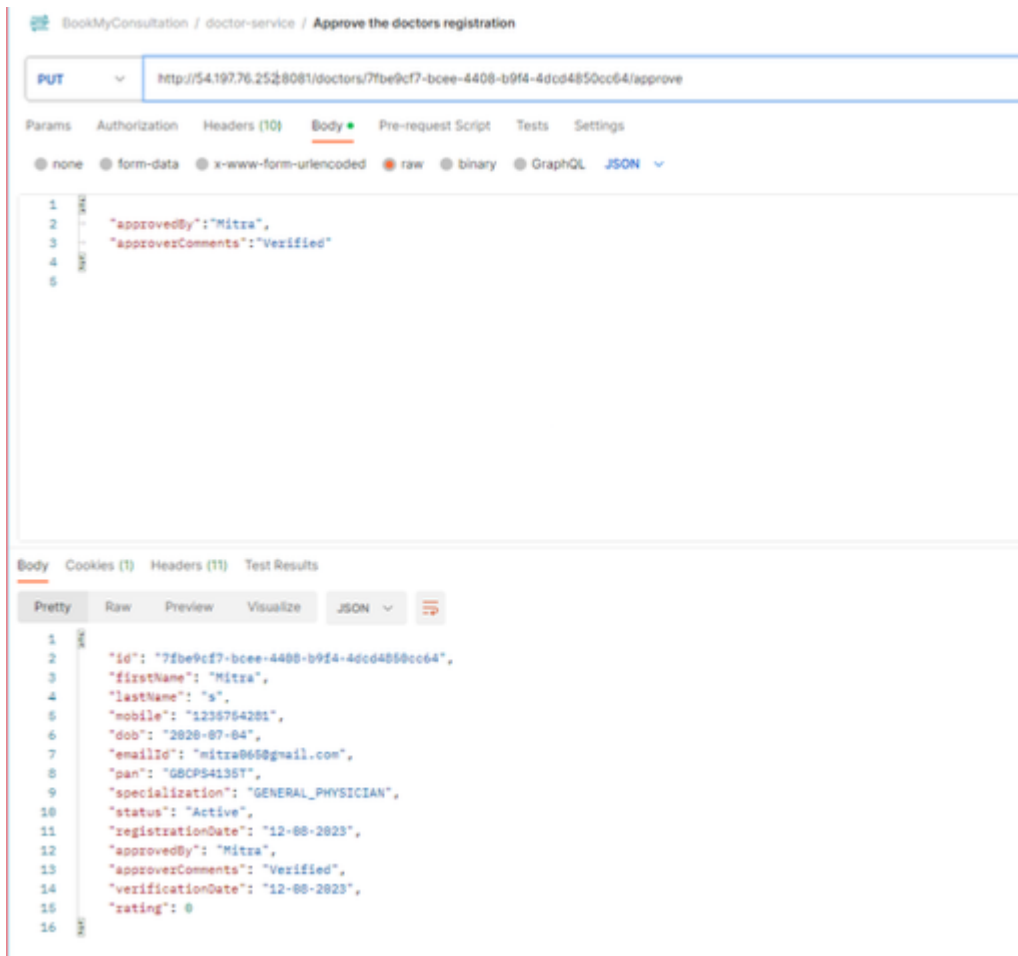
```
1 {
2   "approvedBy": "Mittra",
3   "approverComments": "Verified"
4 }
5
```

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2023-08-12T16:22:04.419+00:00",
3   "status": 403,
4   "error": "Forbidden",
5   "message": "",
6   "path": "/doctors/7fbe9cf7-bcee-4408-b9f4-4dcd4850cc64/approve"
7 }
```

- Successful Scenario – Approve Request



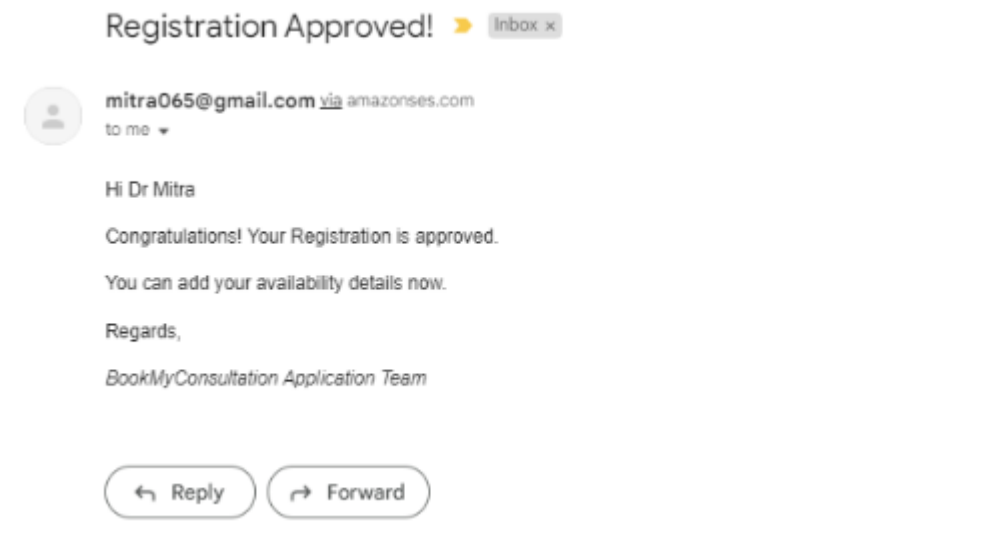
- Kafka Consumer Logs

```

{
  "id" : "7f9e9cf7-bcee-4408-b9f4-4dcd4850cc64",
  "firstName" : "Mitra",
  "lastName" : "s",
  "mobile" : "1235754281",
  "dob" : "2020-07-04",
  "emailId" : "mitra065@gmail.com",
  "pan" : "GBCPS4135T",
  "specialization" : "GENERAL_PHYSICIAN",
  "status" : "Active",
  "registrationDate" : "12-08-2023",
  "approvedBy" : "Mitra",
  "approverComments" : "Verified",
  "verificationDate" : "12-08-2023",
  "rating" : 0
}

```

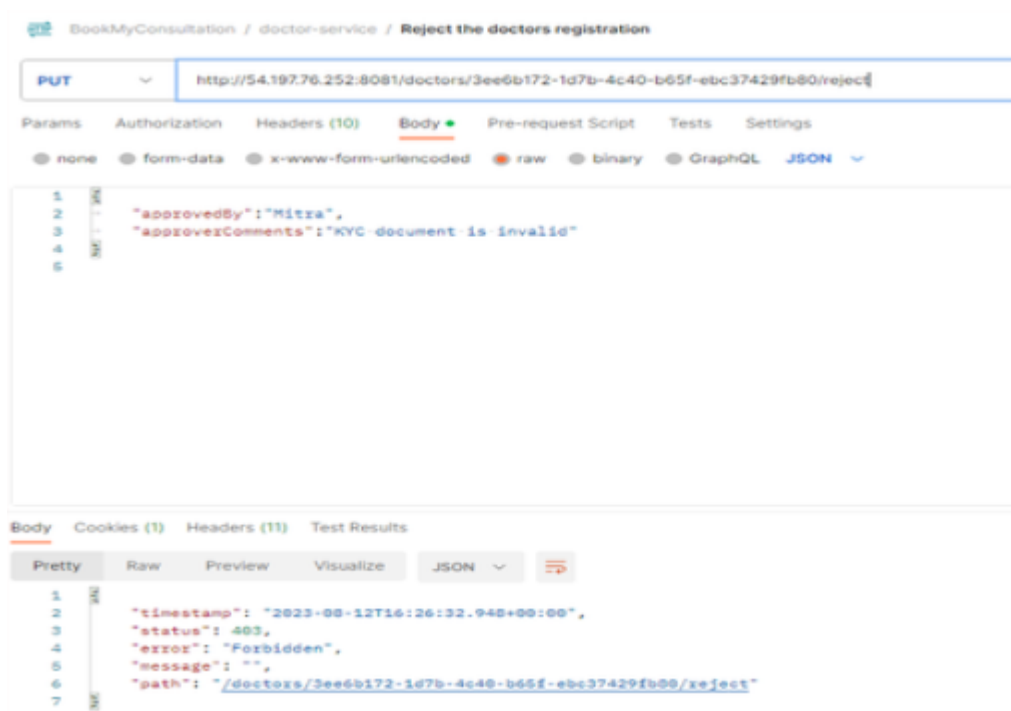
- Registration Approved Mail Screenshot



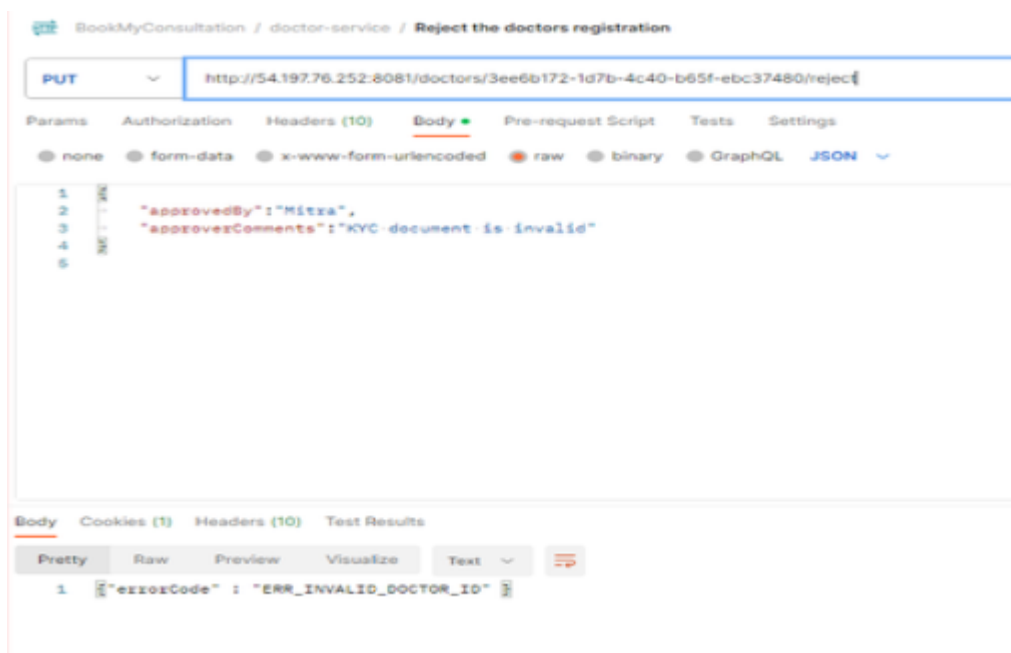
**Endpoint 4: URI:** `/doctors/{doctorId}/reject`

This endpoint is responsible for rejecting the registration of the doctor

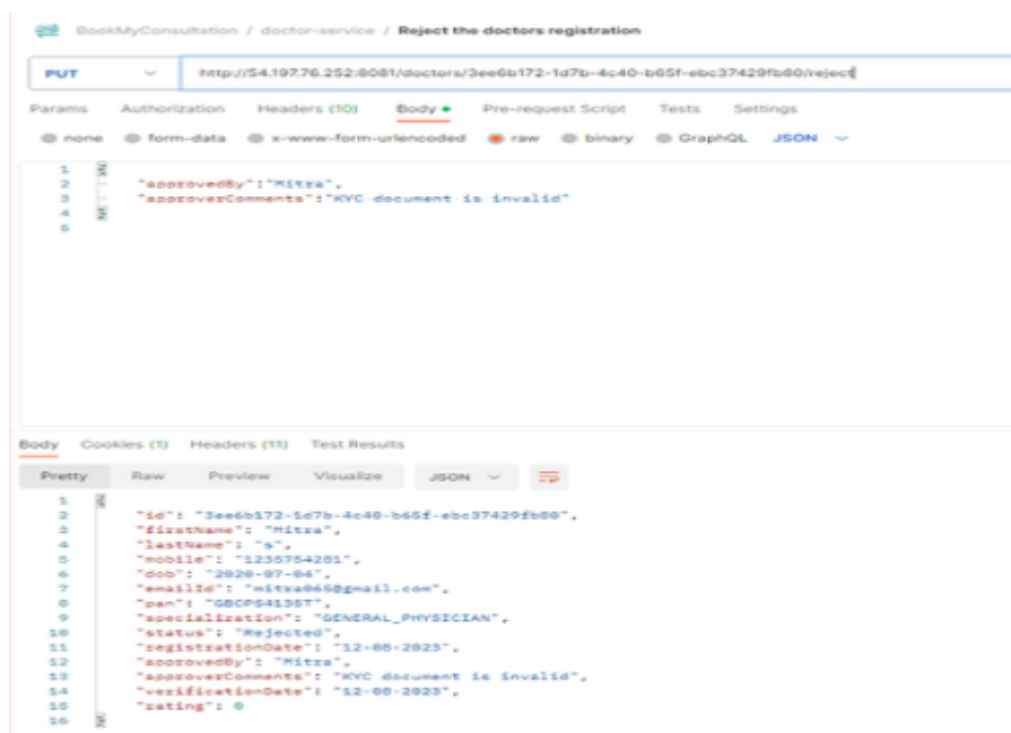
- Endpoint Forbidden for unauthorised users



- Validation Error when Invalid Doctor Id is passed



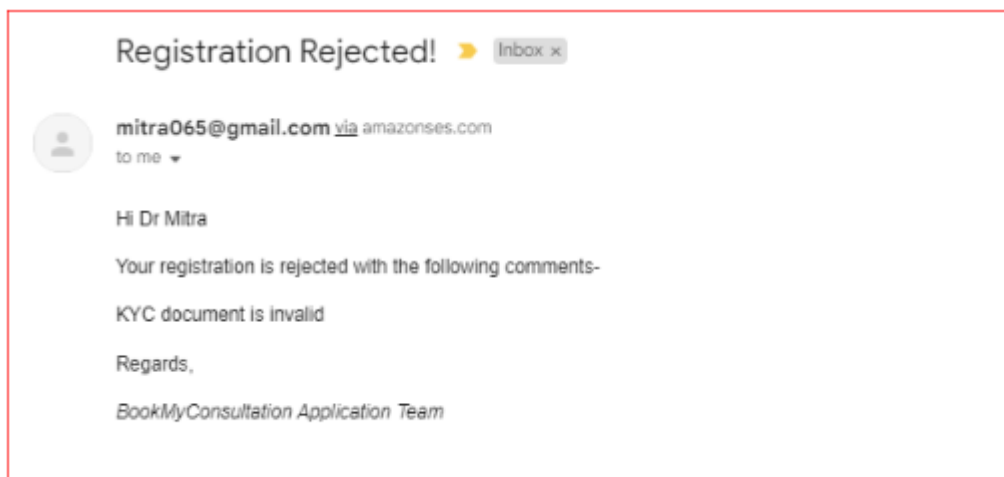
- Success Scenario - Rejection



- Kafka Consumer Logs

```
{
  "id" : "3ee6b172-1d7b-4c40-b65f-ebc37429fb80",
  "firstName" : "Mitra",
  "lastName" : "s",
  "mobile" : "1235754281",
  "dob" : "2020-07-04",
  "emailId" : "mitra065@gmail.com",
  "pan" : "GBCPS4135T",
  "specialization" : "GENERAL_PHYSICIAN",
  "status" : "Rejected",
  "registrationDate" : "12-08-2023",
  "approvedBy" : "Mitra",
  "approverComments" : "KYC document is invalid",
  "verificationDate" : "12-08-2023",
  "rating" : 0
}
```

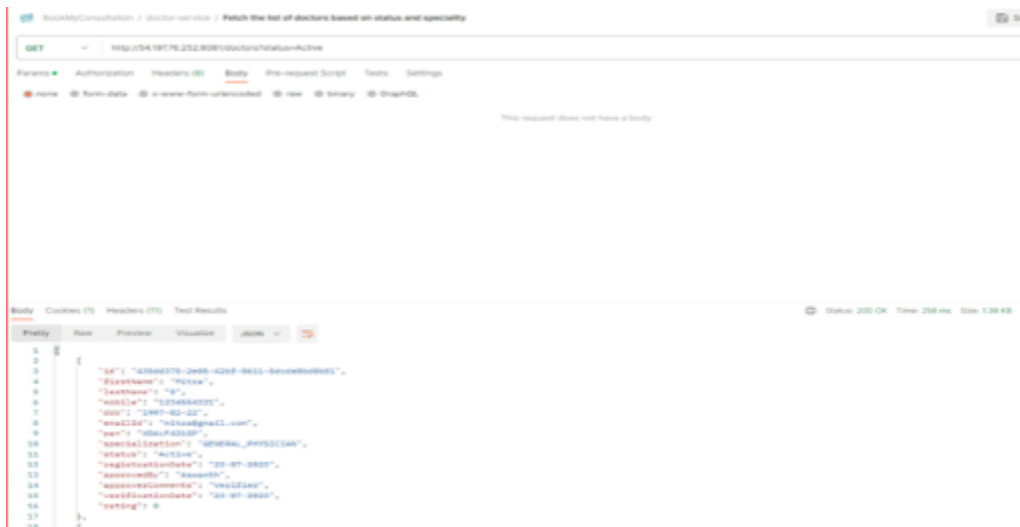
- Registration Rejection Mail Received Screenshot



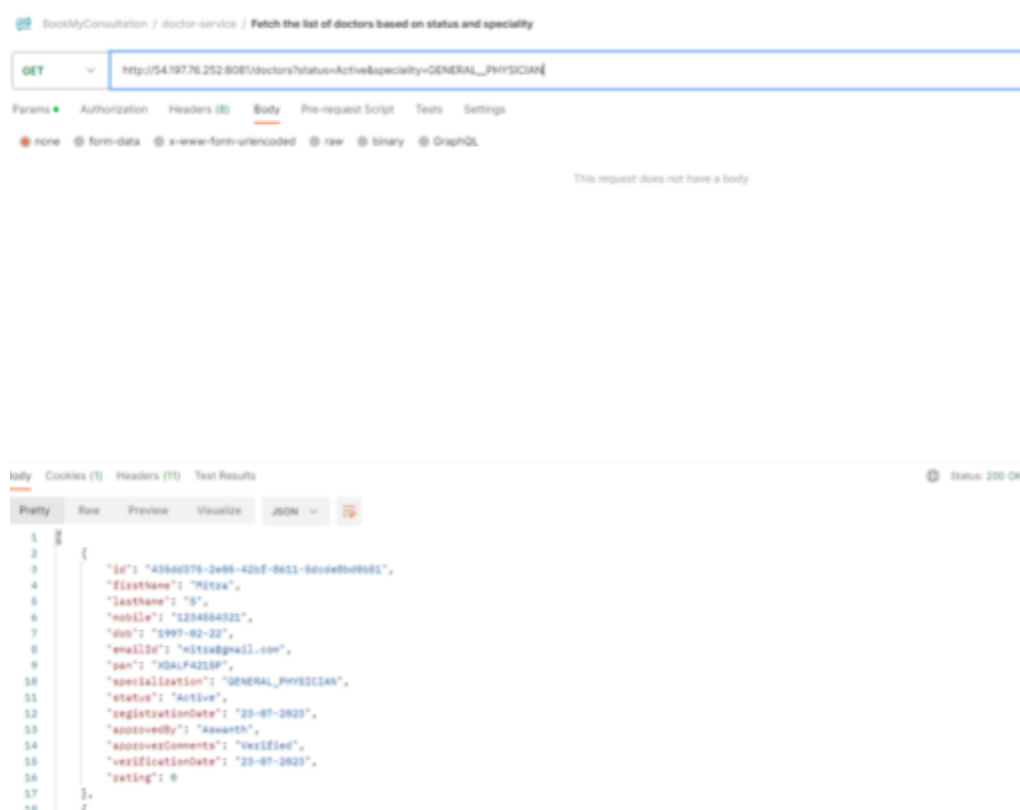
**Endpoint 5: URI:** `/doctors?<Required Status & Speciality Filter Condition>`

This endpoint is responsible for returning the list of 20 doctors sorted by ratings.

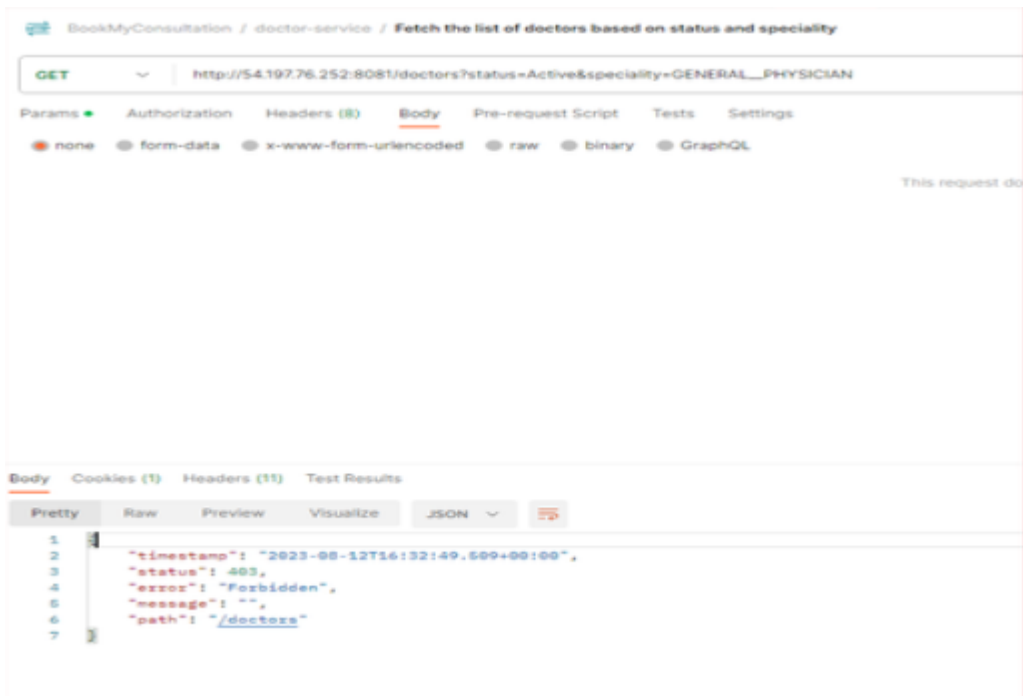
- With Status Filter



- With Status & Specialty Filter



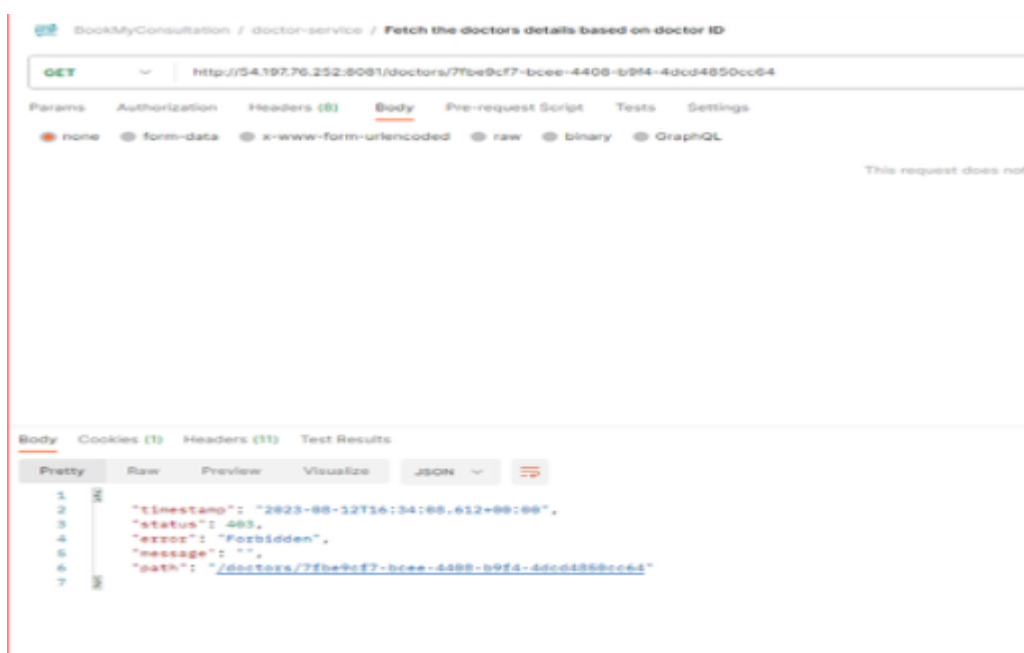
- Forbidden Error for Unauthorised error



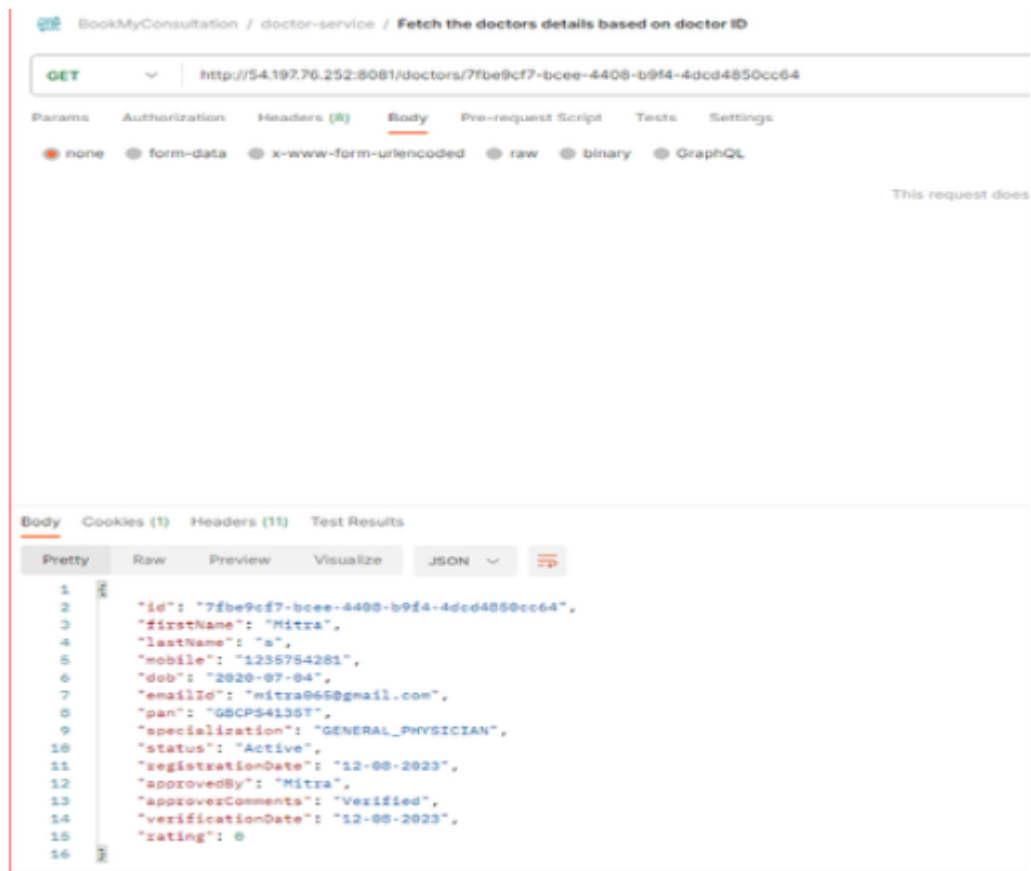
**Endpoint 6:** URI: `/doctors/{doctorId}`

This endpoint is responsible for returning the details of the doctor based on the doctor ID.

- Forbidden Error for Unauthorised error

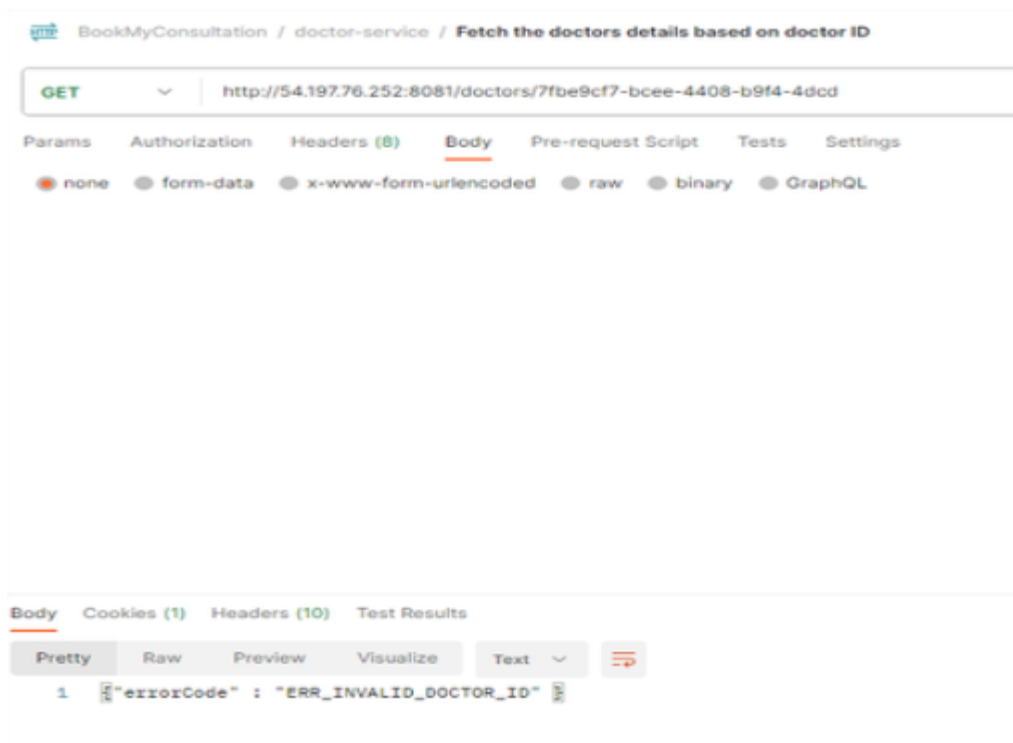


- Success Scenario



- Validation Error – When Invalid Doctor Id passed





## User -Onboarding Service

**Endpoint 1:** URI: `/users`

This endpoint is responsible for collecting information about the user.

- Successful Scenario

BookMyConsultation / user-service / Register the User

POST http://54.197.76.252:9191/users

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   ...
3   "firstName": "Aswanth",
4   "lastName": "V",
5   "dob": "1995-07-19",
6   "emailId": "mitra065@gmail.com",
7   "mobile": "987583903"
8 }
9
10

```

Body Cookies (1) Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": "e40a7c0d-0db0-4bc1-af76-e4c5c9e1c3b7",
3   "firstName": "Aswanth",
4   "lastName": "V",
5   "mobile": "987583903",
6   "dob": "1995-07-19",
7   "emailId": "mitra065@gmail.com",
8   "createdDate": "12-08-2023"
9 }

```

- Forbidden Error for unauthorised error

BookMyConsultation / user-service / Register the User

POST http://localhost:9191/users

Params Authorization Headers (1) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   ...
3   "firstName": "Aswanth",
4   "lastName": "V",
5   "dob": "1995-07-19",
6   "emailId": "mitra065@gmail.com",
7   "mobile": "987583903"
8 }
9
10

```

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize Text

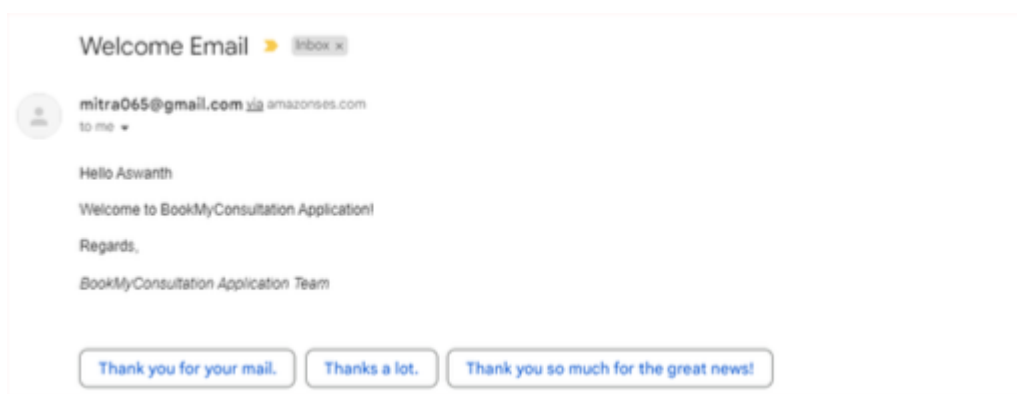
1

Status: 403 Forbidden

- Kafka Consumer logs

```
{
  "id" : "e40a7c0d-0db0-4bc1-af76-e4c5c9e1c3b7",
  "firstName" : "Aswanth",
  "lastName" : "V",
  "mobile" : "987583903",
  "dob" : "1995-07-19",
  "emailId" : "mitra065@gmail.com",
  "createdDate" : "12-08-2023"
}
```

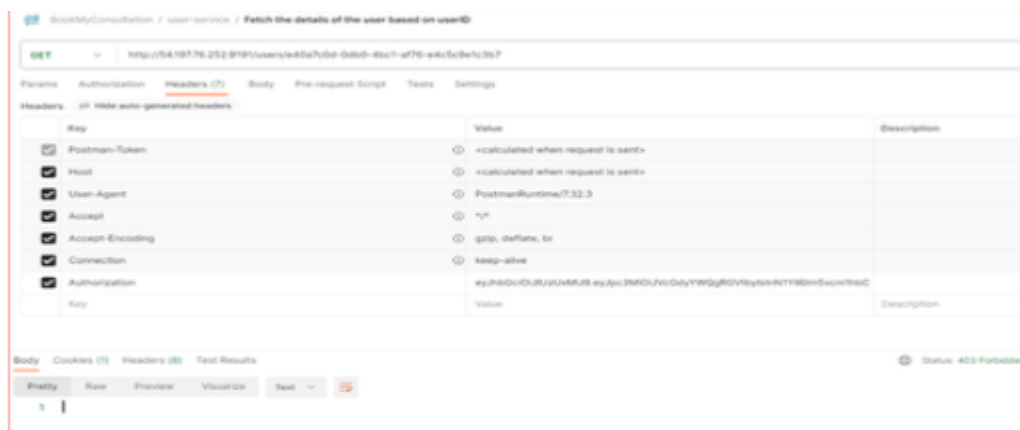
- Verification mail received after user registration



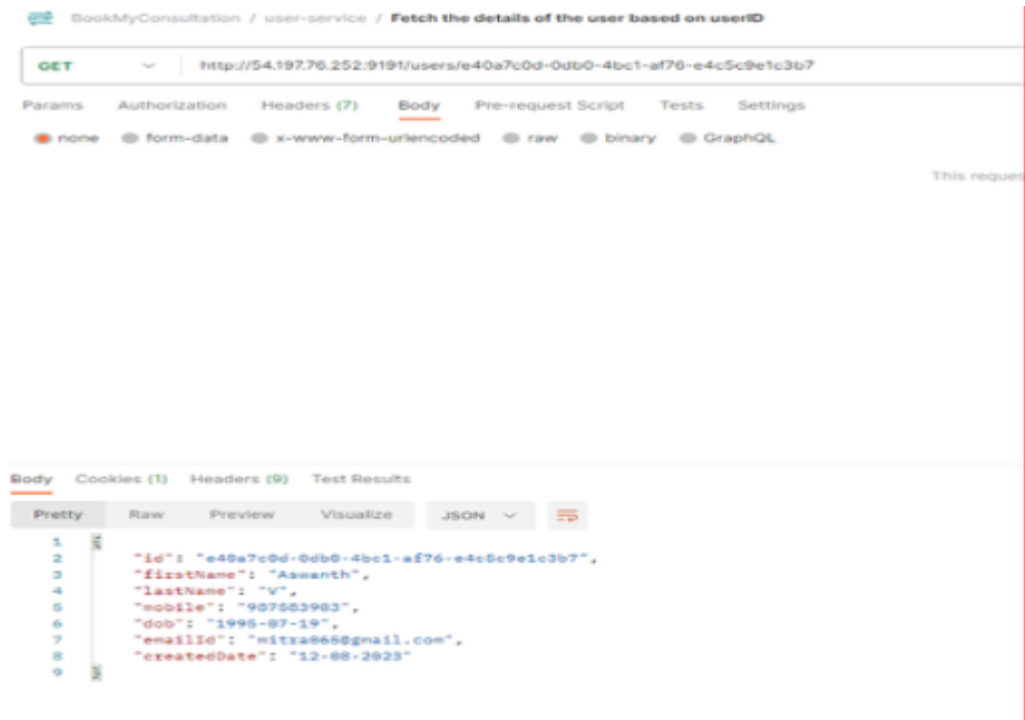
**Endpoint 2: URI:** `/users/{userID}`

This endpoint is responsible for collecting information about the user.

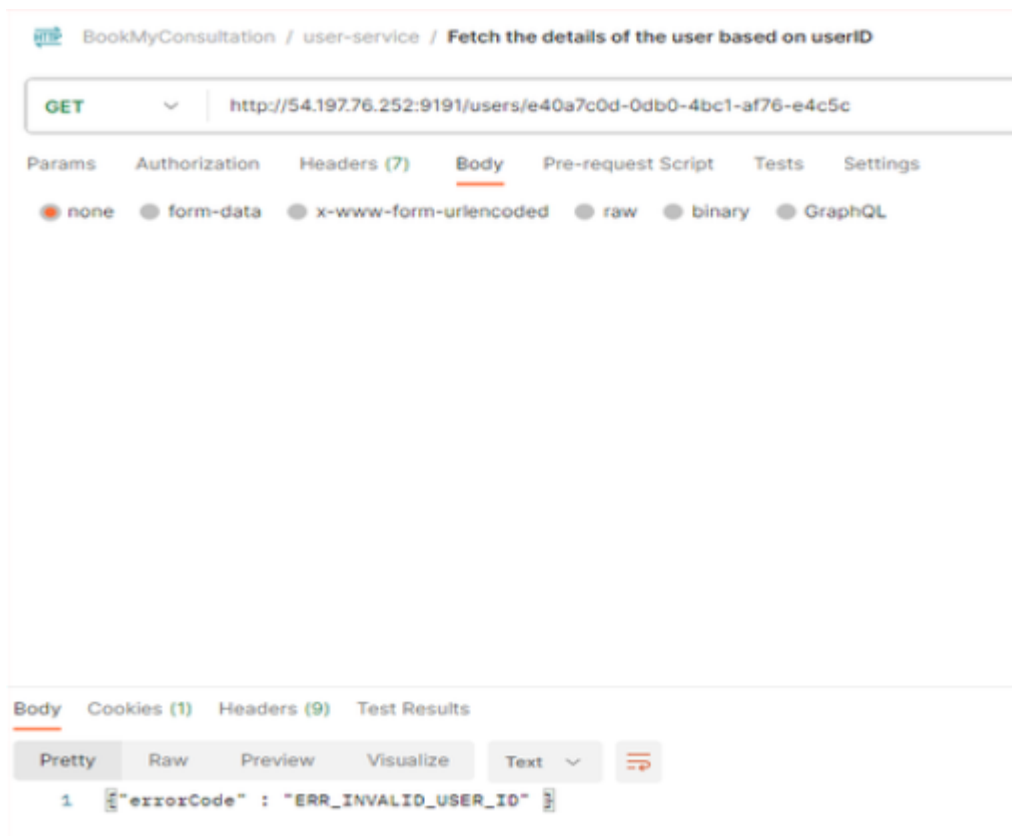
- Forbidden Error for unauthorised error



- Successful Scenario



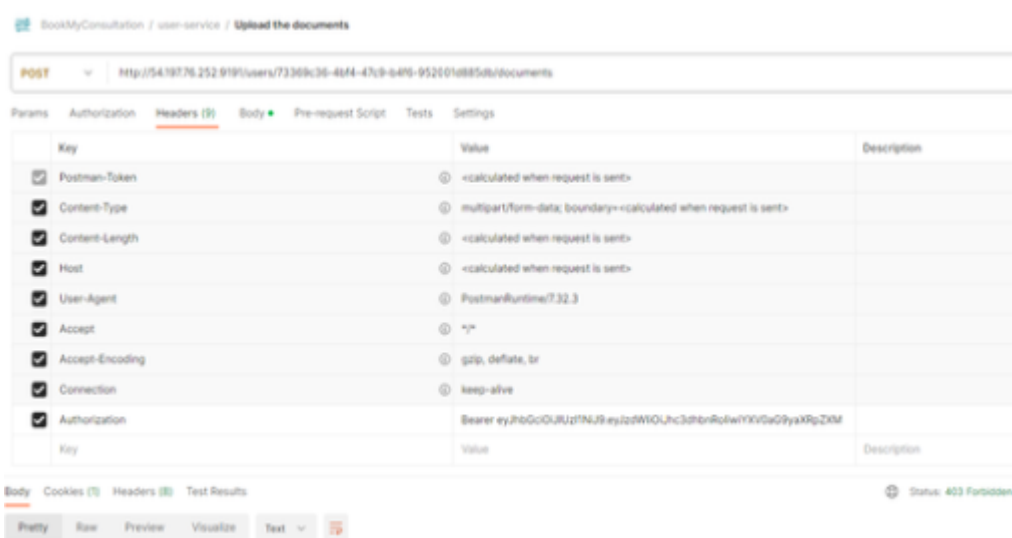
- Validation Error - Invalid User Id Passed



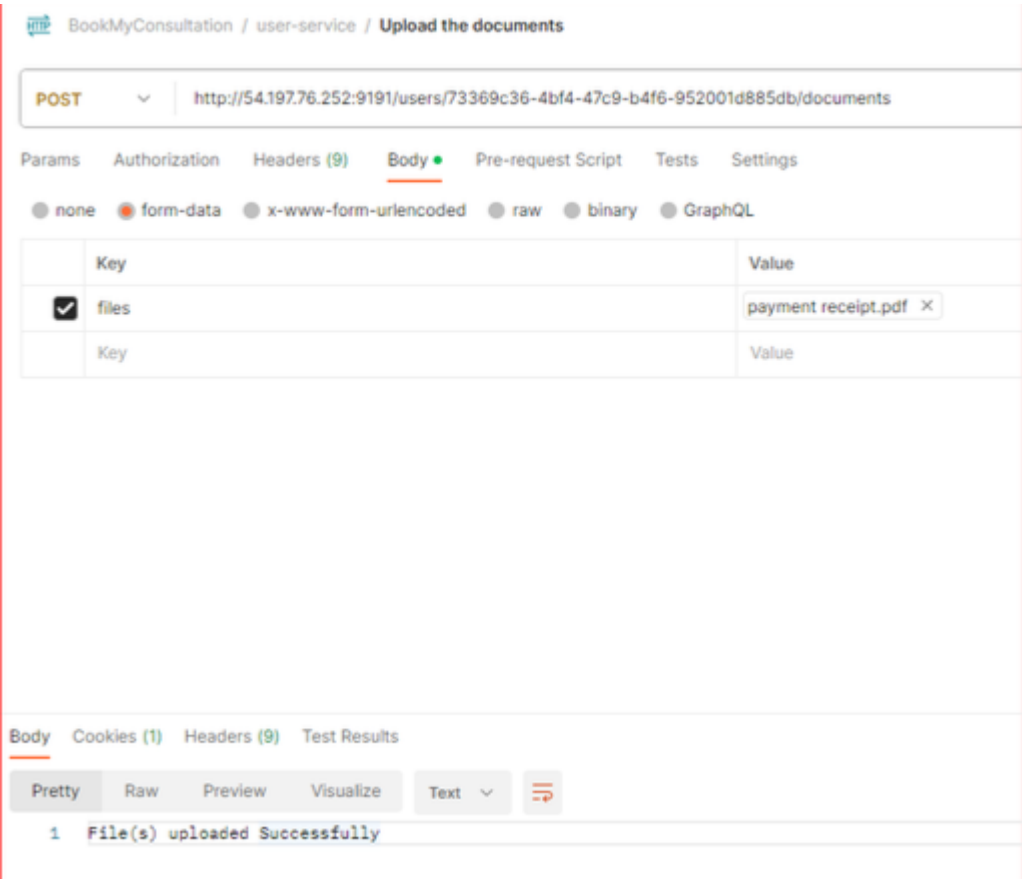
**Endpoint 3:** URI: `/users/{id}/documents`.

A POST request will be sent to this endpoint to upload the files.

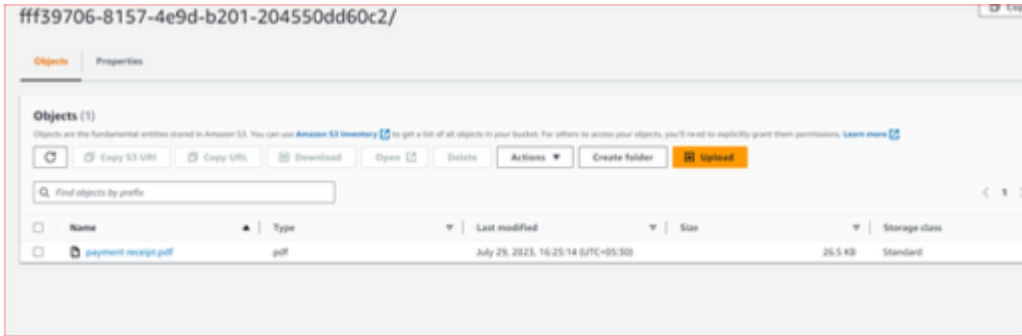
- Forbidden Error for unauthorised error



- File Uploaded Successfully



- File Uploaded in S3 bucket

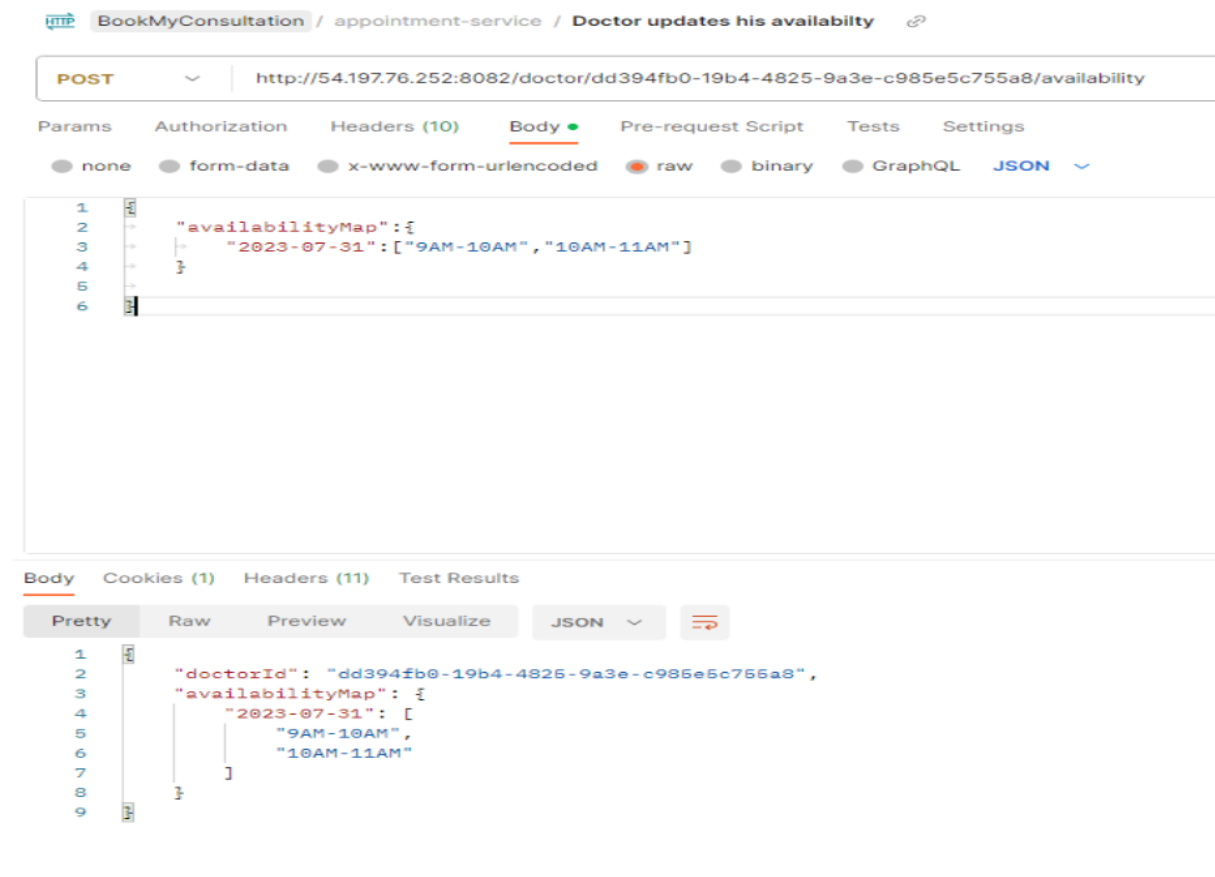


# Appointment Service

**Endpoint 1:** POST URI: `/doctor/{doctorId}/availability`

This endpoint is responsible for updating the availability of the doctors.

- Successful Scenario



**Endpoint 2:** GET URI: `/doctor/{doctorId}/availability`

This endpoint is responsible for returning the availability of the doctors.

- Forbidden Error for Unauthorised Users


GET ▼ http://54.197.76.252:8082/doctor/dd394fb0-19b4-4825-9a3e-c985e5c755a8/availability

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

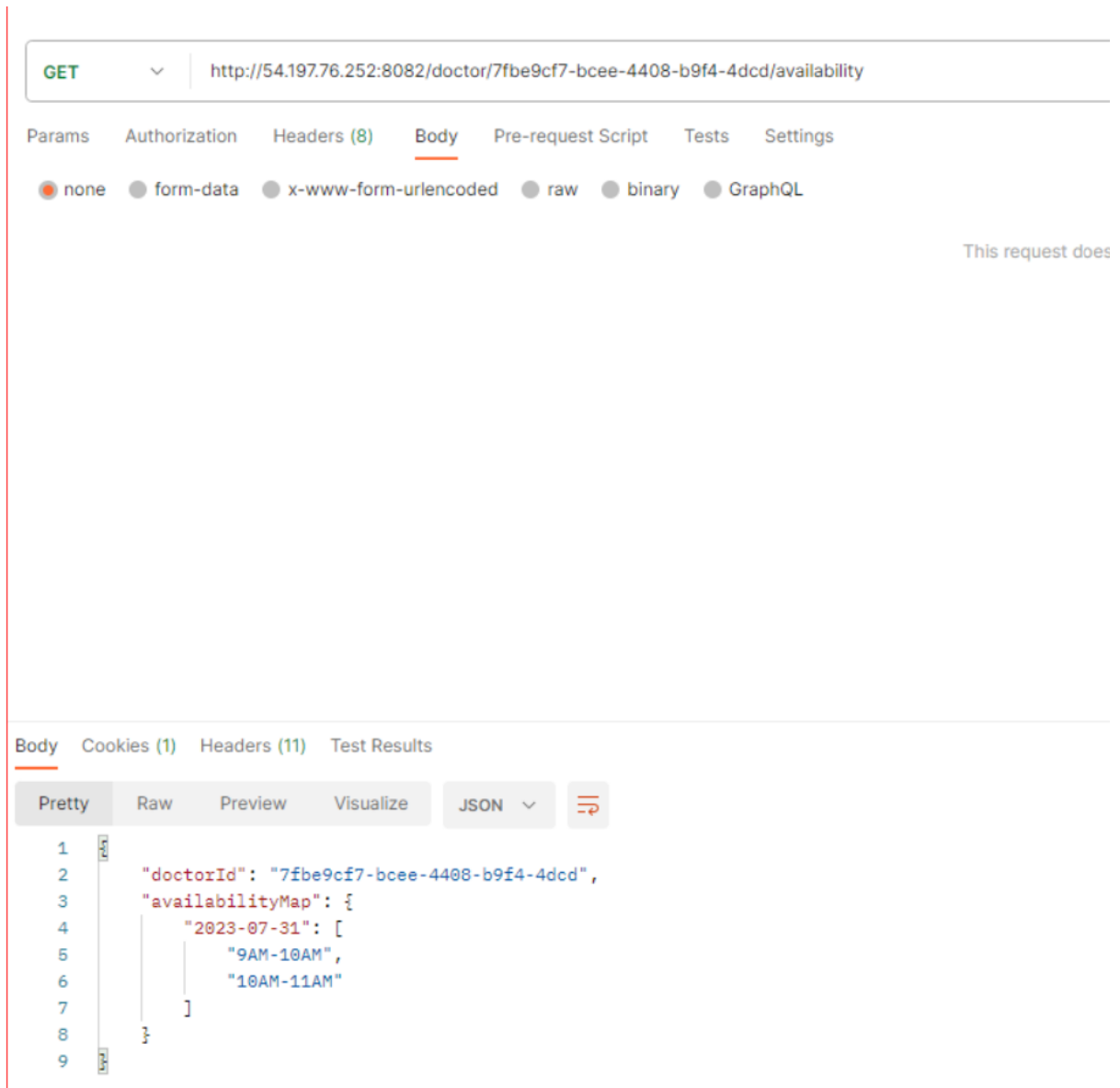
Body Cookies (1) Headers (10) Test Results ⊕ Status: 403 Forbidden

Pretty Raw Preview Visualize Text ▼ 

1

- Successful Scenario





### Endpoint 3: URI: `/appointments`

This endpoint is responsible for booking an appointment.

- Forbidden Error for Unauthorised Users

POST http://54.197.76.252:8082/appointments

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1
2  "doctorId": "7fbe9cf7-bcee-4408-b9f4-4dcd",
3  "doctorName": "Mitra S",
4  "userId": "e40a7c0d-0db0-4bc1-af76-e4c5c9e1c3b7",
5  "timeSlot": "9.30AM-10AM",
6
7  "appointmentDate": "2023-07-31"
8
9
```

Body Cookies (1) Headers (10) Test Results

Status: 403 Forbidden

Pretty Raw Preview Visualize Text

1

- Validation Error When Invalid User Id is Passed

POST http://54.197.76.252:8082/appointments

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

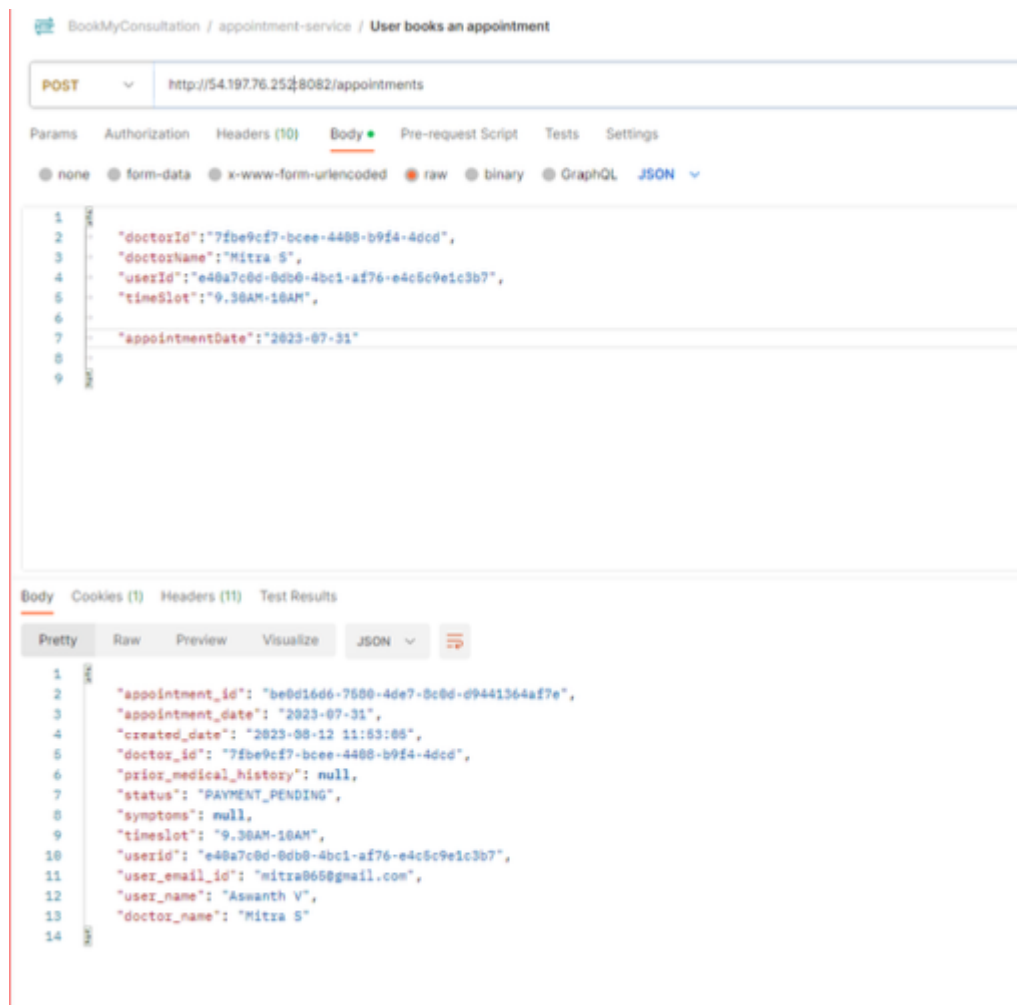
```
1
2  "doctorId": "7fbe9cf7-bcee-4408-b9f4-4dcd",
3  "doctorName": "Mitra S",
4  "userId": "e40a7c0d-0db0-4bc1-af76-e4c5c9e1c3b7",
5  "timeSlot": "9.30AM-10AM",
6
7  "appointmentDate": "2023-07-31"
8
9
```

Body Cookies (1) Headers (10) Test Results

Pretty Raw Preview Visualize Text

```
1 {"statusCode":400, "message":"Invalid User"}
```

- Successful Scenario



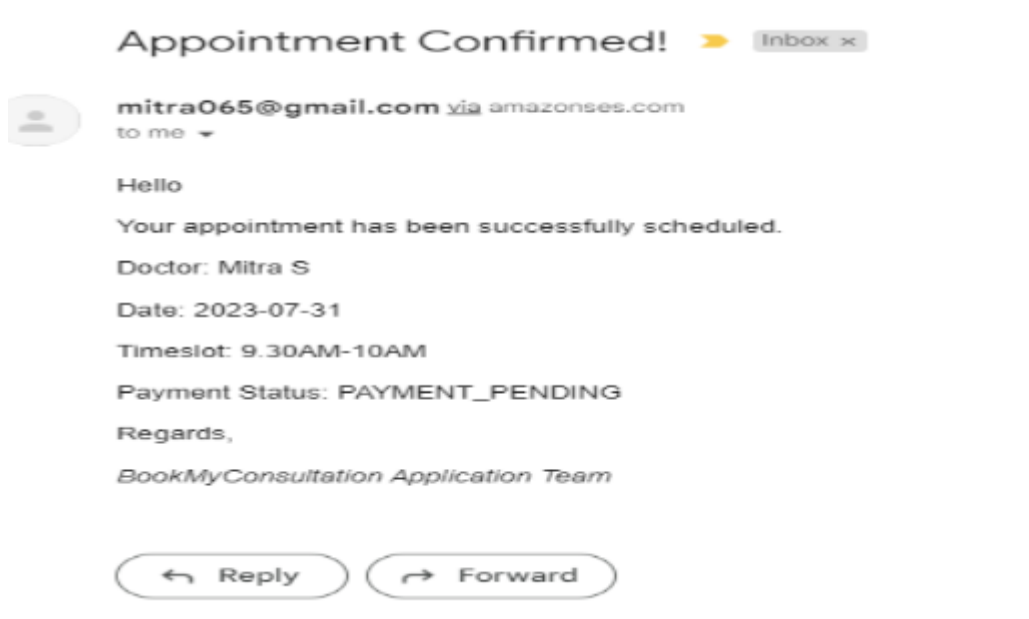
- Kafka Consumer logs

```

{
  "appointment_id" : "be0d16d6-7580-4de7-8c0d-d9441364af7e",
  "appointment_date" : "2023-07-31",
  "created_date" : "2023-08-12 11:53:05",
  "doctor_id" : "7fbe9cf7-bcee-4408-b9f4-4dcd",
  "prior_medical_history" : null,
  "status" : "PAYMENT_PENDING",
  "symptoms" : null,
  "timeslot" : "9.30AM-10AM",
  "userid" : "e40a7c0d-0db0-4bc1-af76-e4c5c9e1c3b7",
  "user_email_id" : "mitra065@gmail.com",
  "user_name" : "Aswanth V",
  "doctor_name" : "Mitra S"
}

```

- Appointment Confirmation Mail



**Endpoint 4:** URI: `/appointments/{appointmentId}`

This endpoint is responsible for retrieving the details of an appointment.

- Forbidden Error for Unauthorised Users

GET <http://54.197.76.252:8082/appointments/be0d16d6-7580-4de7-8c0d-d9441364af7e>


Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

	Key	Value	Description
	Key	Value	Description

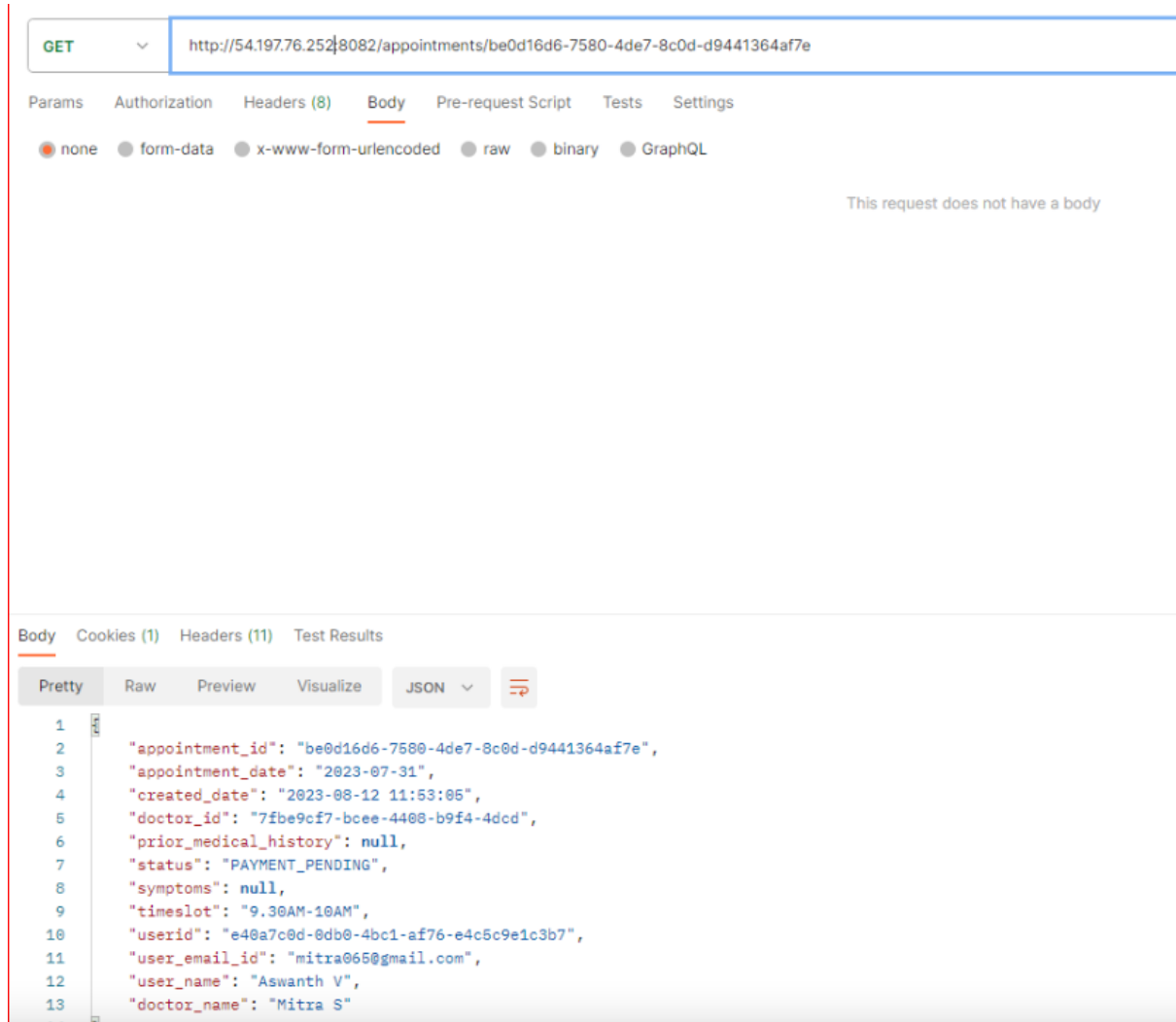
Body Cookies (1) Headers (10) Test Results

Status: 403 Forbidden 1

Pretty Raw Preview Visualize Text 

1

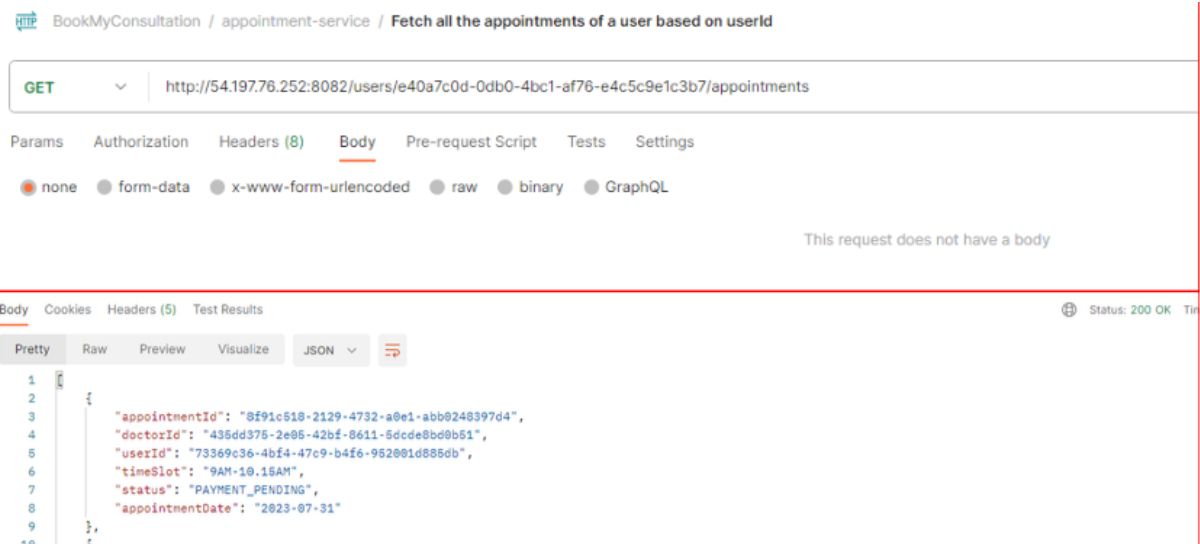
- Successful Scenario



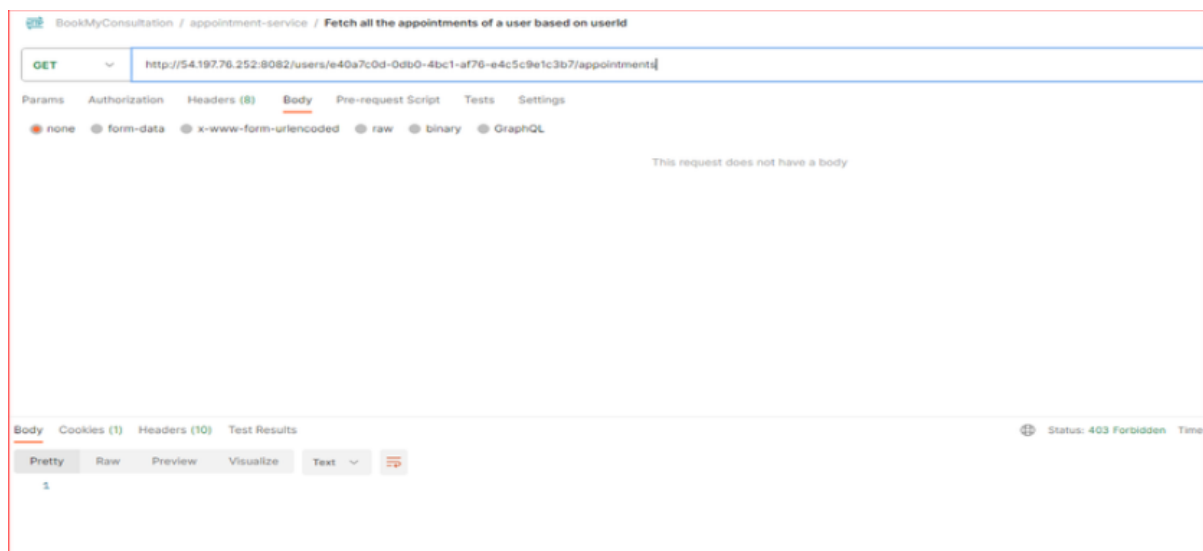
**Endpoint 5: URI:** `/users/{userId}/appointments`

This endpoint is responsible for retrieving the details of all the appointments corresponding to a `userId`.

- Successful Scenario



- Forbidden Error for Unauthorised Users



**Endpoint 6: URI:** `/prescriptions`

This endpoint is responsible for sending the prescriptions for the appointment.

- Forbidden Error for Unauthorised Users

BookMyConsultation / appointment-service / Prescription Save

POST http://54.197.76.252:8082/prescriptions

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "appointmentId": "8f91c518-2129-4732-a0e1-abb0248397d4",
3   "doctorId": "4d5be250-0184-4c55-ba00-df3c36835a4d",
4   "doctorName": "Mitra S",
5   "userId": "73369c36-4bf4-47c9-b4f6-952001d885db",
6
7   "diagnosis": "Teeth Cavity",
8   "medicineList": [
9     {
10      "name": "Calpol",
11      "type": "Tablet",
12      "dosage": "1 week",
13      "duration": "1 week",
14      "frequency": "3 times a day",
15      "remarks": "after food"
16    },
17    {
18      "name": "PainKiller",

```

Body Cookies (1) Headers (10) Test Results Status: 403 Forbidden

Pretty Raw Preview Visualize Text

```

1

```

- Successful Scenario

BookMyConsultation / appointment-service / Prescription Save

POST http://54.197.76.252:8082/prescriptions

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "appointmentId": "8f91c518-2129-4732-a0e1-abb0248397d4",
3   "doctorId": "4d5be250-0184-4c55-ba00-df3c36835a4d",
4   "doctorName": "Mitra S",
5   "userId": "73369c36-4bf4-47c9-b4f6-952001d885db",
6
7   "diagnosis": "Teeth Cavity",
8   "medicineList": [
9     {
10      "name": "Calpol",
11      "type": "Tablet",
12      "dosage": "1 week",
13      "duration": "1 week",
14      "frequency": "3 times a day",
15      "remarks": "after food"
16    },
17    {
18      "name": "PainKiller",

```

Body Cookies Headers (5) Test Results Status: 200 OK Time:

Pretty Raw Preview Visualize JSON

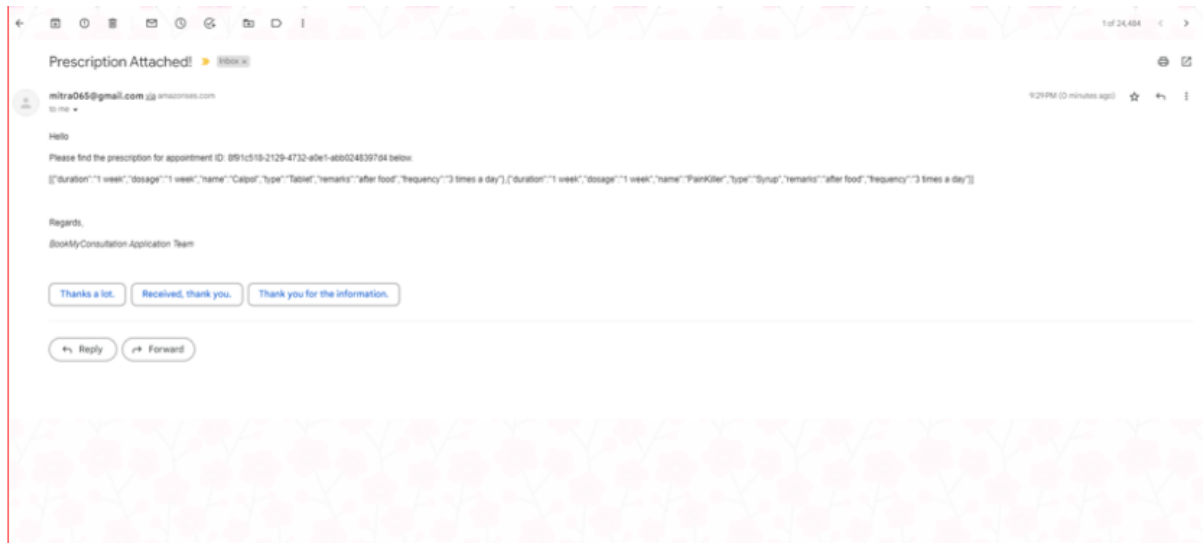
```

1 {
2   "appointmentId": "8f91c518-2129-4732-a0e1-abb0248397d4",
3   "doctorId": "4d5be250-0184-4c55-ba00-df3c36835a4d",
4   "doctorName": "Mitra S",
5   "userId": "73369c36-4bf4-47c9-b4f6-952001d885db",
6   "diagnosis": "Teeth Cavity",
7   "emailId": "mitra066@gmail.com",
8   "medicineList": [
9     {
10      "name": "Calpol",
11      "type": "Tablet",
12      "dosage": "1 week",
13      "duration": "1 week",
14      "frequency": "3 times a day",
15      "remarks": "after food"
16    },
17    {
18      "name": "PainKiller",

```



- Prescription Mail Received



## Payment Service

**Endpoint 1: URI:** `/payments?appointmentId=<the appointmentId for which you want to make apayment>`

This endpoint is responsible for making payments.

- Forbidden Error for Unauthorised Users

BookMyConsultation / payment-service / **Payment endpoint**

**POST** ▼ <http://54.197.76.252:8086/payments?appointmentId=8f91c518-2129-4732-a0e1-abb0248397d4>

Params ● Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

1 |

Body Cookies (1) Headers (10) Test Results ⊞ Status: 403 Forbidden

Pretty Raw Preview Visualize Text ▼ ≡

1

- Successful Scenario

BookMyConsultation / payment-service / **Payment endpoint** Ⓔ

**POST** ▼ <http://54.197.76.252:8086/payments?appointmentId=8f91c518-2129-4732-a0e1-abb0248397d4>

Params ● Authorization **Headers (9)** Body ● Pre-request Script Tests Settings

Key	Value
<input checked="" type="checkbox"/> Postman-Token	① <calculated when request is sent>
<input checked="" type="checkbox"/> Content-Type	① application/json
<input checked="" type="checkbox"/> Content-Length	① <calculated when request is sent>
<input checked="" type="checkbox"/> Host	① <calculated when request is sent>
<input checked="" type="checkbox"/> User-Agent	① PostmanRuntime/7.32.3
<input checked="" type="checkbox"/> Accept	① */*
<input checked="" type="checkbox"/> Accept-Encoding	① gzip, deflate, br
<input checked="" type="checkbox"/> Connection	① keep-alive
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiU9.eyJzdWIiOiJtaXRyYSIsImF1dG8iOiJhbn0248397d4
Key	Value

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize **JSON** ▼ ≡

```

1  {
2    "id": "332de89b-2a61-46f7-9536-3feea78652fc",
3    "appointmentId": "8f91c518-2129-4732-a0e1-abb0248397d4",
4    "createdDate": "12-08-2023 07:48:59"
5  }
```

# Rating Service

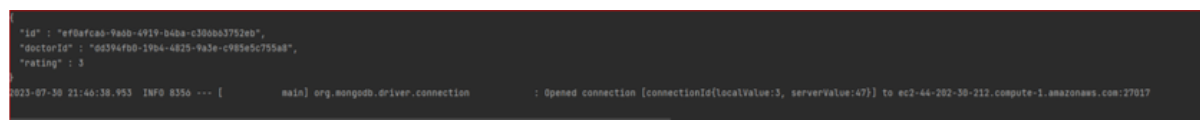
**Endpoint 1: URI:** `/ratings`

This endpoint is used by the users to submit the ratings of their experience with the doctor with whom they had an appointment.

- Successful Scenario



- Kafka Logs Screenshot



# Notification Service

The notification service uses Kafka to receive notifications sent from each service. For particular services, we receive confirmations to send emails using Amazon SES. The functionality has been implemented as shown in each of the required APIs.

The **CreateDoctor** and **CreateUser** API sends verification email using SES to the respective email ID to confirm that the email exists.

The **approveDoctor**, **rejectDoctor**, **setAppointment** and **setPrescription** APIs uses SES to send custom messages to the respective email ID of the Doctor or the User.

Other services send kafka notifications to print by themselves, as shown in each API