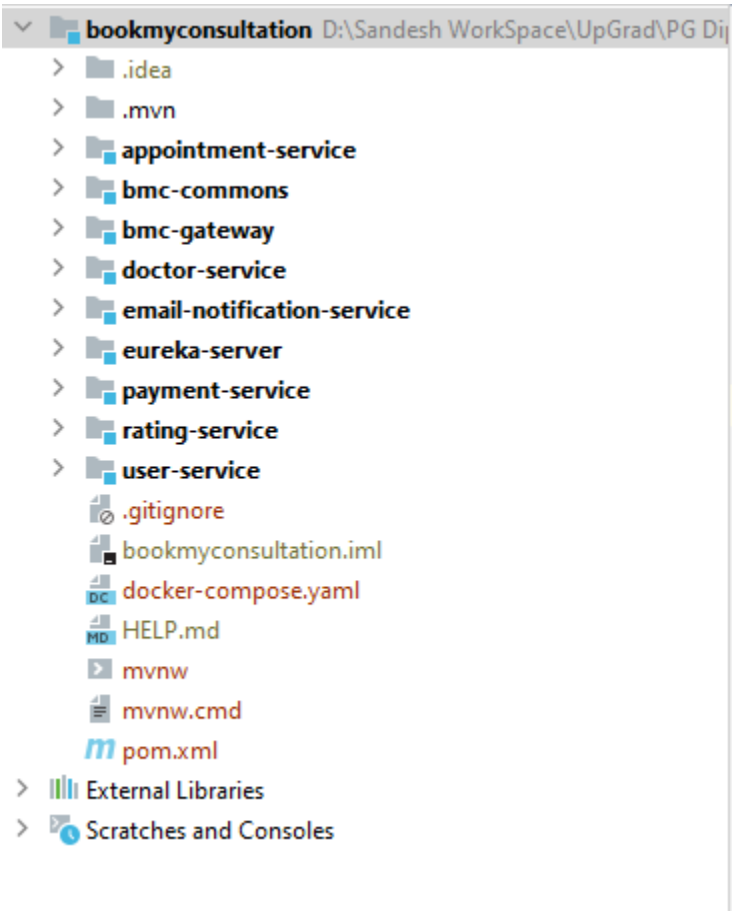
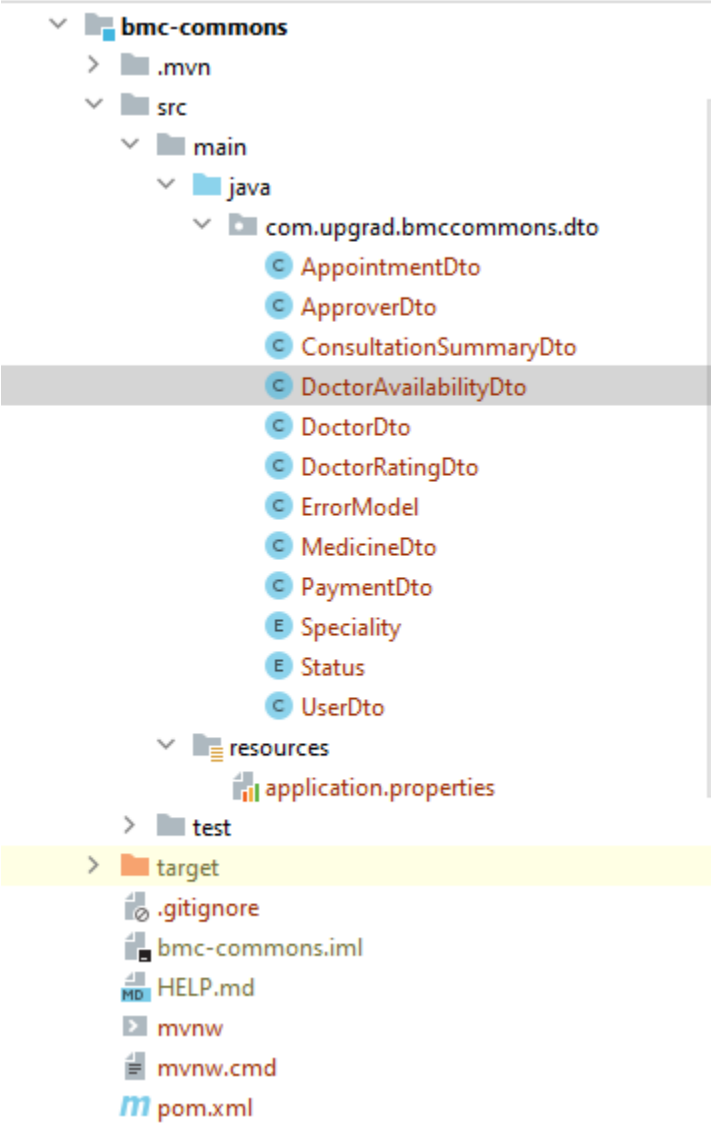
24. Project Structure



A parent project BookMyConsultation consisting for child modules for each of the microservice
The parent module has the docker-compose file

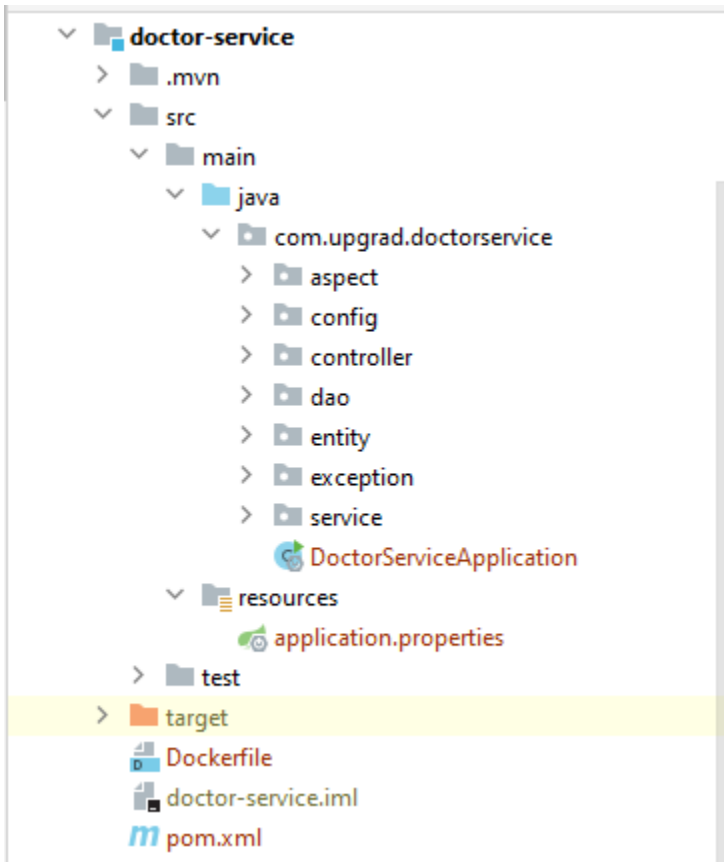
25. bmc-commons:



This module contains all the dto's that are used commonly across the project. By externalizing this we can reduce the duplication and also maintain the same version of dto throughout the project.

This artifact will be used by all the modules to use the dtos

26. doctor-service



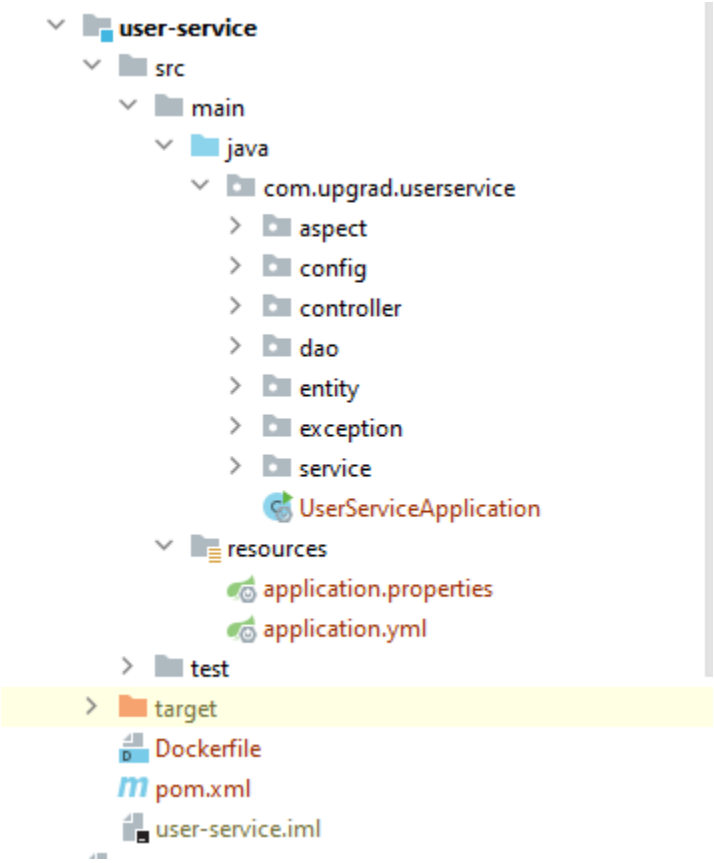
This module is a microservice responsible for doctor onboarding and related functions.
The service is registered as doctor-service in eureka and hosted on 8080

- docker-compose:

```
doctor-service:
 build: doctor-service
 container_name: doctor-service
 image: bookmyconsultation/doctor-service:1.0.0
 ports:
   - "8080:8080"
 networks:
   - bookmyconsulationnet
 environment:
   EUREKA_HOST_NAME: eureka-server
   MONGODB_HOST_NAME: ec2-44-201-170-28.compute-1.amazonaws.com
   KAFKA_HOST_NAME: kafka
   S3_ACCESS_KEY: AKIAUPFTNDWLKA5S34X4
   S3_SECRET_KEY: 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
```

```
depends_on:
  - eureka-server
  - kafka
```

27. user-service



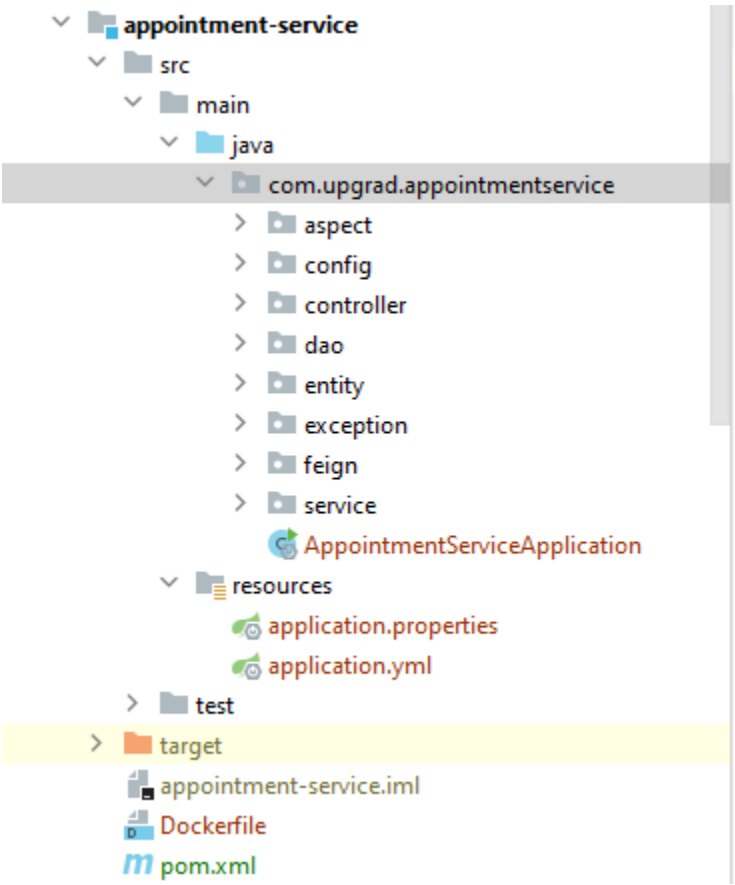
This module is a microservice responsible for user onboarding and related functions.
The service is registered as user-service in eureka and hosted on 8083

- docker-compose

```
user-service:
 build: user-service
 container_name: user-service
 image: bookmyconsultation/user-service:1.0.0
 ports:
   - "8083:8083"
 networks:
   - bookmyconsulationnet
 environment:
```

```
    EUREKA_HOST_NAME: eureka-server
    MONGODB_HOST_NAME: ec2-44-201-170-28.compute-1.amazonaws.com
    KAFKA_HOST_NAME: kafka
    S3_ACCESS_KEY: AKIAUPFTNDWLKA5S34X4
    S3_SECRET_KEY: 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
  depends_on:
    - eureka-server
    - kafka
```

28. appointment-service



This module is a microservice responsible for appointment related functions.
The service is registered as appointment-service in eureka and hosted on 8083

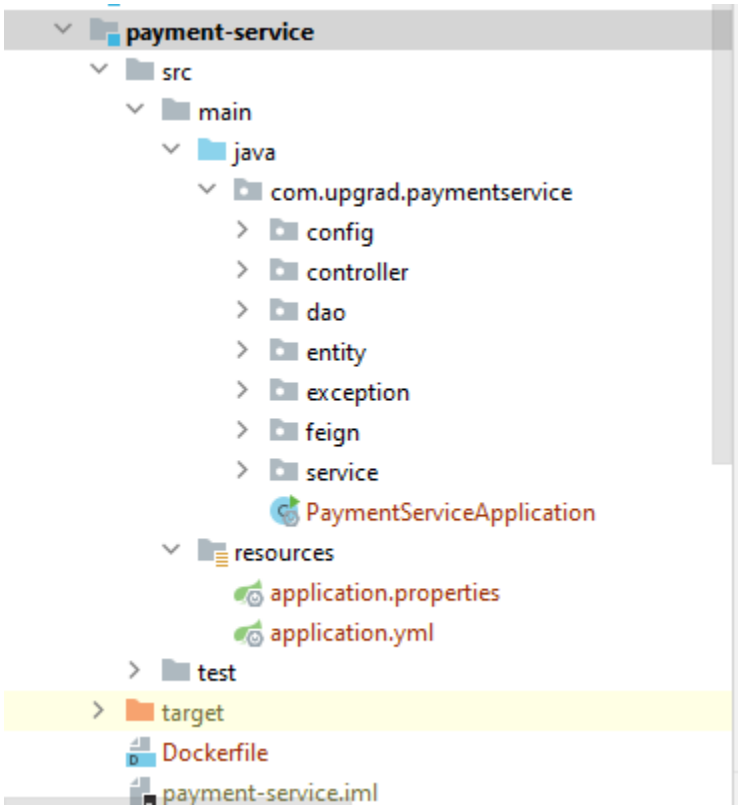
APPOINTMENT-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.10:8082

- docker-compose

```
appointment-service:
 build: appointment-service
 container_name: appointment-service
```

```
    image: bookmyconsultation/appointment-service:1.0.0
    ports:
      - "8082:8082"
    networks:
      - bookmyconsulationnet
    environment:
      EUREKA_HOST_NAME: eureka-server
      MONGODB_HOST_NAME: ec2-44-201-170-28.compute-1.amazonaws.com
      KAFKA_HOST_NAME: kafka
    depends_on:
      - eureka-server
      - user-service
      - doctor-service
      - kafka
```

29. payment-service



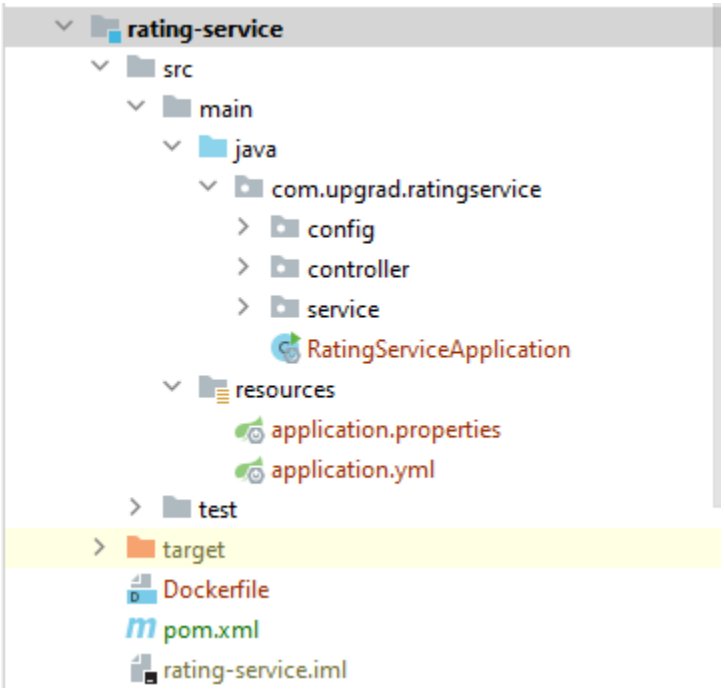
This module is a microservice responsible for appointment related functions.
The service is registered as payment-service in eureka and hosted on 8086

PAYMENT-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.11:8086

- docker-compose

```
payment-service:
 build: payment-service
 container_name: payment-service
 image: bookmyconsultation/payment-service:1.0.0
 ports:
   - "8086:8086"
 networks:
   - bookmyconsulationnet
 environment:
   EUREKA_HOST_NAME: eureka-server
   MONGODB_HOST_NAME: ec2-44-201-170-28.compute-1.amazonaws.com
   KAFKA_HOST_NAME: kafka
   APPOINTMENT_SERVICE_HOST_NAME: BMC-GATEWAY
 depends_on:
   - eureka-server
   - appointment-service
   - kafka
```

30. rating-service



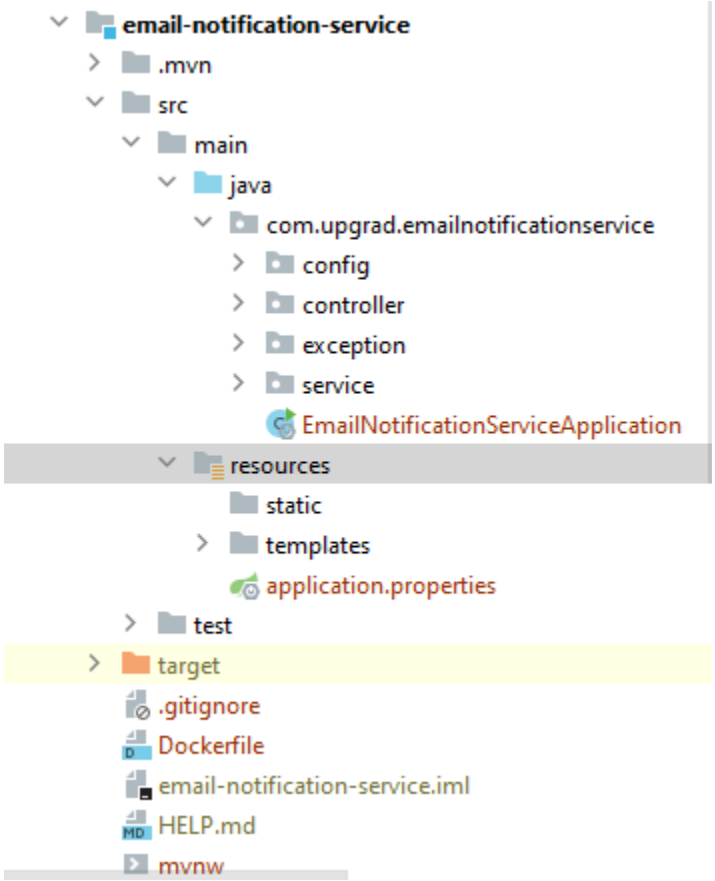
This module is a microservice responsible for rating doctor.
The service is registered as rating-service in eureka and hosted on 8084

RATING-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.5:8084

- docker-compose

```
rating-service:
build: rating-service
container_name: rating-service
image: bookmyconsultation/rating-service:1.0.0
ports:
  - "8084:8084"
networks:
  - bookmyconsulationnet
environment:
  EUREKA_HOST_NAME: eureka-server
  MONGODB_HOST_NAME: ec2-44-201-170-28.compute-1.amazonaws.com
  KAFKA_HOST_NAME: kafka
depends_on:
  - eureka-server
  - kafka
```

31. Email-notification-service



This module is a microservice responsible for send notification emails using AWS SES

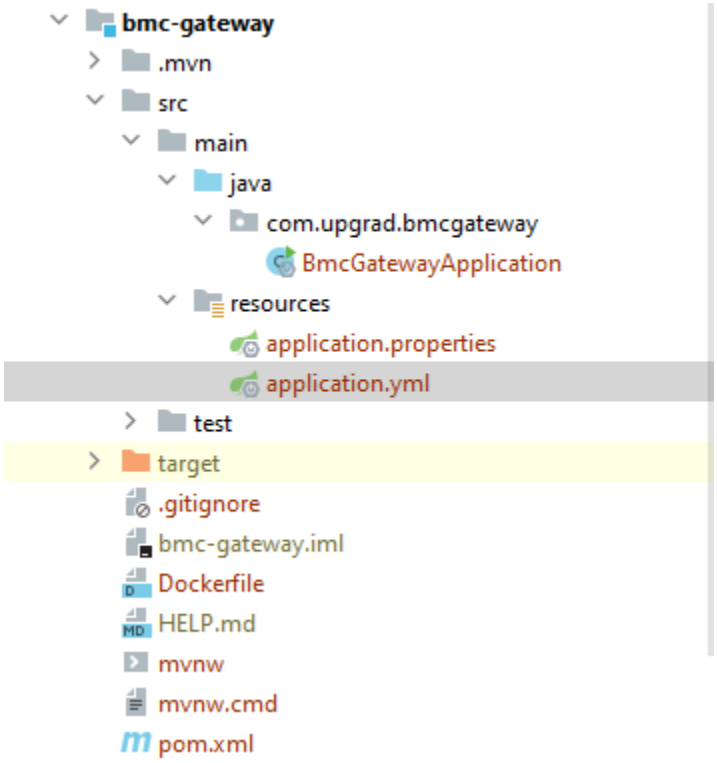
The service is registered as email-notification-service in eureka and hosted on 8087

EMAIL-NOTIFICATION-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.8:8087

- docker-compose

```
email-notification-service:
 build: email-notification-service
 container_name: email-notification-service
 image: bookmyconsultation/email-notification-service:1.0.0
 ports:
   - "8087:8087"
 networks:
   - bookmyconsulationnet
 environment:
   EUREKA_HOST_NAME: eureka-server
   KAFKA_HOST_NAME: kafka
   AWS_SES_ACCESS_KEY: AKIAUPFTNDWLKA5S34X4
   AWS_SES_SECRET_KEY: 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
   SMTP_AWS_ENDPOINT: email-smtp.us-east-1.amazonaws.com
   SMTP_AWS_USERNAME: AKIAUPFTNDWLKHKTK4NZ
   SMTP_AWS_PASSWORD: BBziyBqXJG/k0pbF1qU543YTihTB2+nWOeb2kn/XIO4b
   SMTP_AWS_FROMEMAIL: sandesh.ayyod89@gmail.com
 depends_on:
   - eureka-server
   - kafka
```

32. bmc-gateway



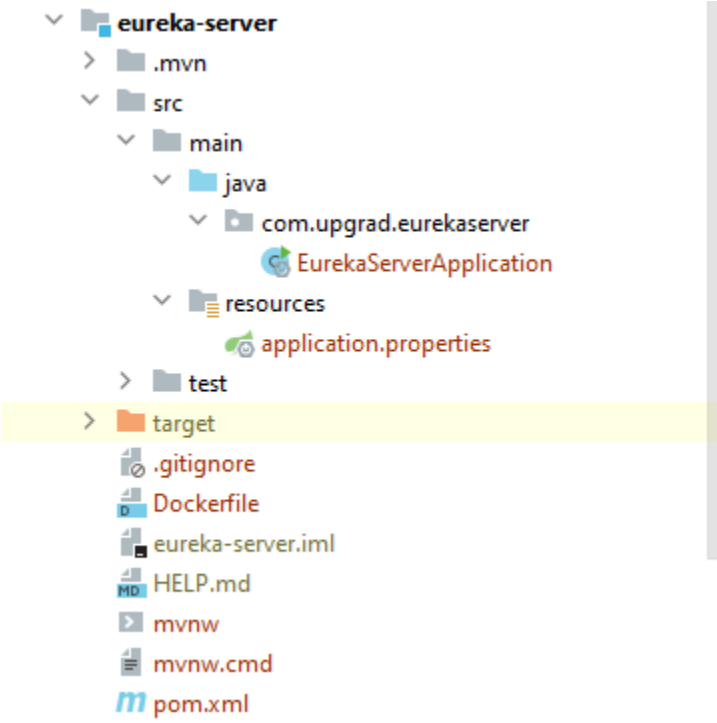
This module is a microservice acts as a gateway for UI to communicate between microsevices
The service is registered as bmc-gateway in eureka and hosted on 9191

BMC-GATEWAY	n/a (1)	(1)	UP (1) - 172.18.0.9:9191

- docker-compose

```
bmc-gateway:
 build: bmc-gateway
 container_name: bmc-gateway
 image: bookmyconsultation/bmc-gateway:1.0.0
 ports:
   - "9191:9191"
 networks:
   - bookmyconsulationnet
 environment:
   EUREKA_HOST_NAME: eureka-server
 depends_on:
   - eureka-server
   - doctor-service
```

33. eureka-server



This module is the service registry and hosted on port 8761

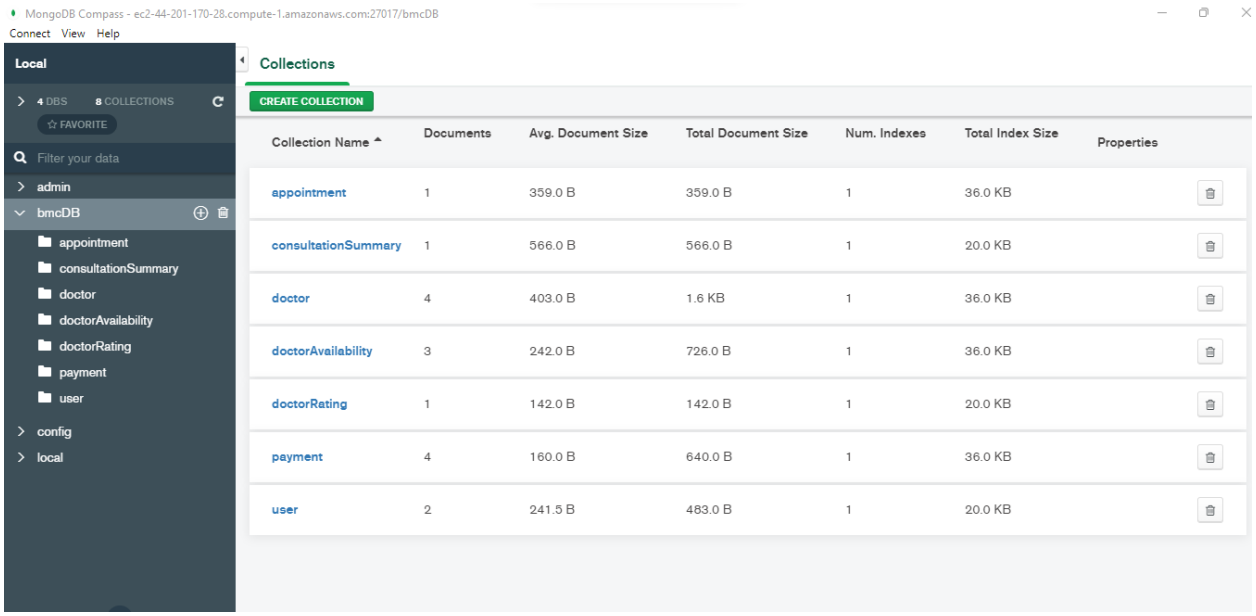
Application	AMIs	Availability Zones	Status
APPOINTMENT-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.10:8082
BMC-GATEWAY	n/a (1)	(1)	UP (1) - 172.18.0.9:9191
DOCTOR-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.6:8080
EMAIL-NOTIFICATION-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.8:8087
PAYMENT-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.11:8086
RATING-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.5:8084
USER-SERVICE	n/a (1)	(1)	UP (1) - 172.18.0.7:8083

http://localhost:8761/

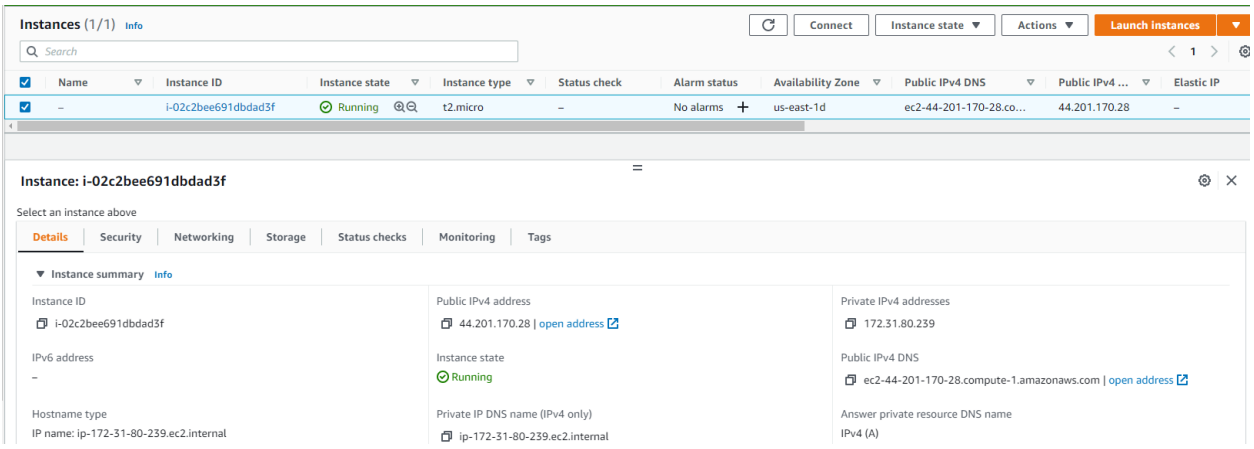
- docker-compose

```
eureka-server:
 build: eureka-server
 container_name: eureka-server
 image: bookmyconsultation/eureka-server:1.0.0
 ports:
   - "8761:8761" # Map the exposed port 3000 on container to port 3000
on the host machine
 networks:
   - bookmyconsulationnet
 environment:
   EUREKA_HOST_NAME: eureka-server
```

34. Database

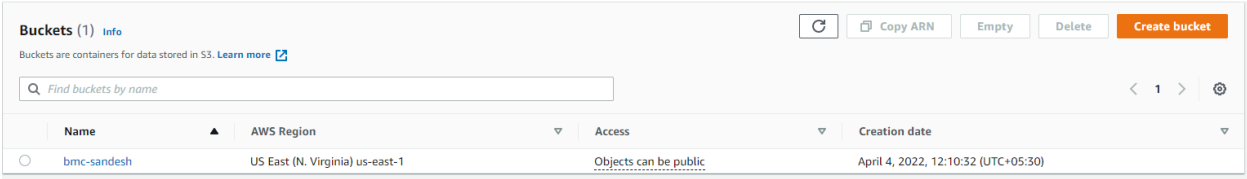


MongoDB created on EC2 server.
And the Public IPv4 DNS is updated in the docker-compose file



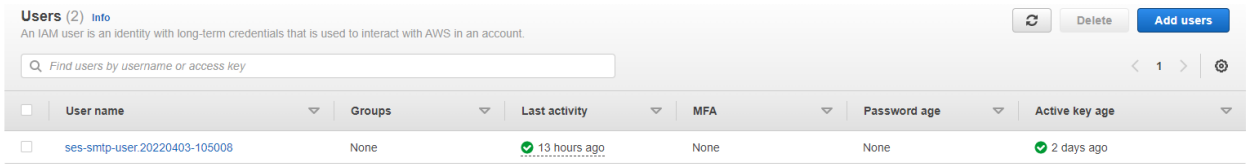

```
MONGODB_HOST_NAME: ec2-44-201-170-28.compute-1.amazonaws.com
```

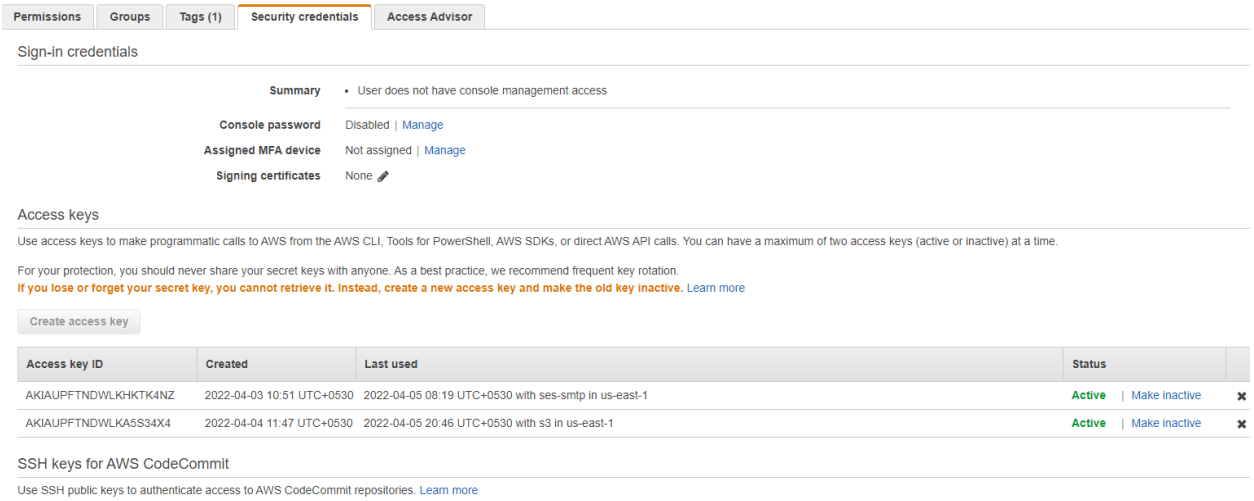
35. S3 Bucket & SES



Access Key and Secret key needs to be generated with IAM and they need to be updated in the docker file

A new IAM user created & Permissions given for s3 and ses





For doctor-service and user-service in docker-compose update the env variables

```
S3_ACCESS_KEY: AKIAUPFTNDWLKA5S34X4
S3_SECRET_KEY: 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
```

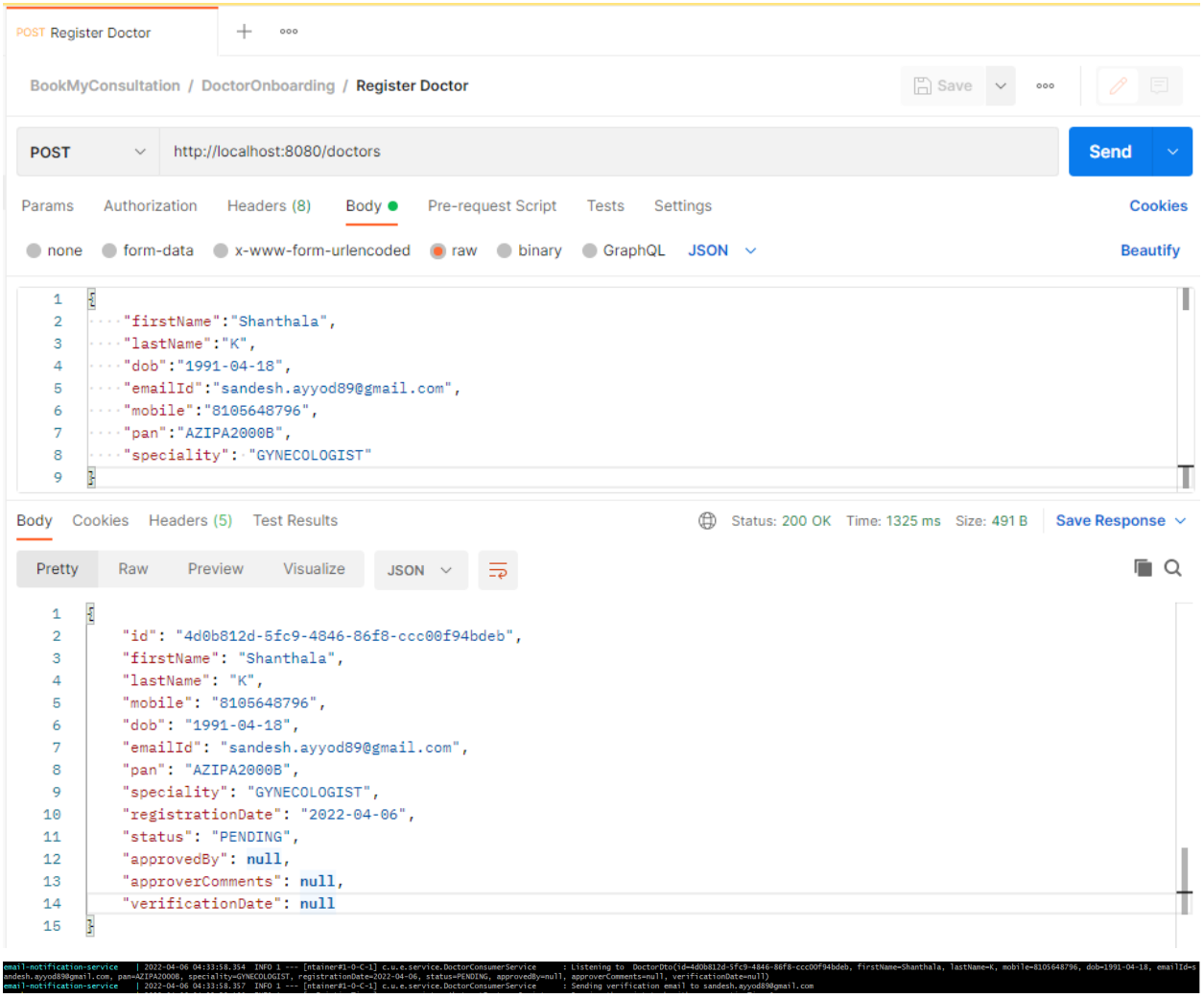
For email-notification service

```
SMTP_AWS_ENDPOINT: email-smtp.us-east-1.amazonaws.com
SMTP_AWS_USERNAME: AKIAUPFTNDWLKHKTK4NZ
SMTP_AWS_PASSWORD: BBziyBqXJG/k0pbF1qU543YTihTB2+nWOeb2kn/XIO4b
SMTP_AWS_FROMEMAIL: sandesh.ayyod89@gmail.com
```

36. Postman screen shots

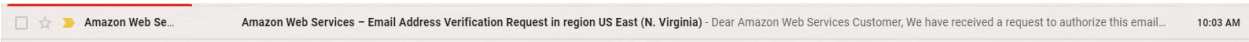
- Register doctor

  Doctor-service persists doctor into mongoDB
  Publishes the dto onto kafka
  Email notification listens to kafka and sends verification email to the email id



```
_id: "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb"
firstName: "Shanthala"
lastName: "K"
mobile: "8105648796"
dob: "1991-04-18"
emailId: "sandesh.ayyod89@gmail.com"
pan: "AZIPA2000B"
speciality: "GYNECOLOGIST"
registrationDate: 2022-04-06T00:00:00.000+00:00
status: "PENDING"
_class: "com.upgrad.doctorservice.entity.Doctor"
```


Email received



- Fetch doctor

GET fetch

BookMyConsultation / DoctorOnboarding / **fetch**

GET http://localhost:8080/doctors/4d0b812d-5fc9-4846-86f8-ccc00f94bdeb

Params Authorization Headers (6) 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 Headers (5) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "id": "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb",
    "firstName": "Shanthala",
    "lastName": "K",
    "mobile": "8105648796",
    "dob": "1991-04-18",
    "emailId": "sandesh.ayyod89@gmail.com",
    "pan": "AZIPA2000B",
    "speciality": "GYNECOLOGIST",
    "registrationDate": "2022-04-06",
    "status": "PENDING",
    "approvedBy": null,
    "approverComments": null,
    "verificationDate": null
}
```

- Upload document

POST upload doctor docum

BookMyConsultation / DoctorOnboarding / upload doctor document

Save

POST http://localhost:8080/doctors/4d0b812d-5fc9-4846-86f8-ccc00f94bdeb/document Send

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

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

KEY	VALUE	DESCRIPTION	Bulk Edit
files	doc_certificate.pdf		
Key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 5.46 s Size: 193 B Save Response

Pretty Raw Preview Visualize Text

```
1  File(s) uploaded Successfully
```

Amazon S3 > Buckets > bmc-sandesh > 4d0b812d-5fc9-4846-86f8-ccc00f94bdeb/

4d0b812d-5fc9-4846-86f8-ccc00f94bdeb/

Copy S3 URI

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

1

Name	Type	Last modified	Size	Storage class
doc_certificate.pdf	pdf	April 6, 2022, 10:13:48 (UTC+05:30)	80.5 KB	Standard

- Approve doctor

PUT approve doctor

BookMyConsultation / DoctorOnboarding / **approve doctor**

PUT http://localhost:8080/doctors/4d0b812d-5fc9-4846-86f8-ccc00f94bdeb/approve

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

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

```
{
    "approvedBy": "sandesh",
    "approverComments" : "Approved doctor"
}
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "id": "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb",
    "firstName": "Shanthala",
    "lastName": "K",
    "mobile": "8105648796",
    "dob": "1991-04-18",
    "emailId": "sandesh.ayyod89@gmail.com",
    "pan": "AZIPA2000B",
    "speciality": "GYNECOLOGIST",
    "registrationDate": "2022-04-06",
    "status": "ACTIVE",
    "approvedBy": "sandesh",
    "approverComments": "Approved doctor",
    "verificationDate": "2022-04-06"
}
```


Email notification received

me **Registration Approved** - Hi Dr sandesh.ayyod89@gmail.com Congratulations !!! Your Registration is Approved in BMC app Regards, BMC

- Reject doctor

PUT reject doctor

BookMyConsultation / DoctorOnboarding / **reject doctor**

PUT http://localhost:8080/doctors/d0bc9c03-9d07-4688-b8f0-edb4fe47d9d5/reject

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

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

```
{
    "approvedBy": "sandesh",
    "approverComments" : "rejected doctor"
}
```

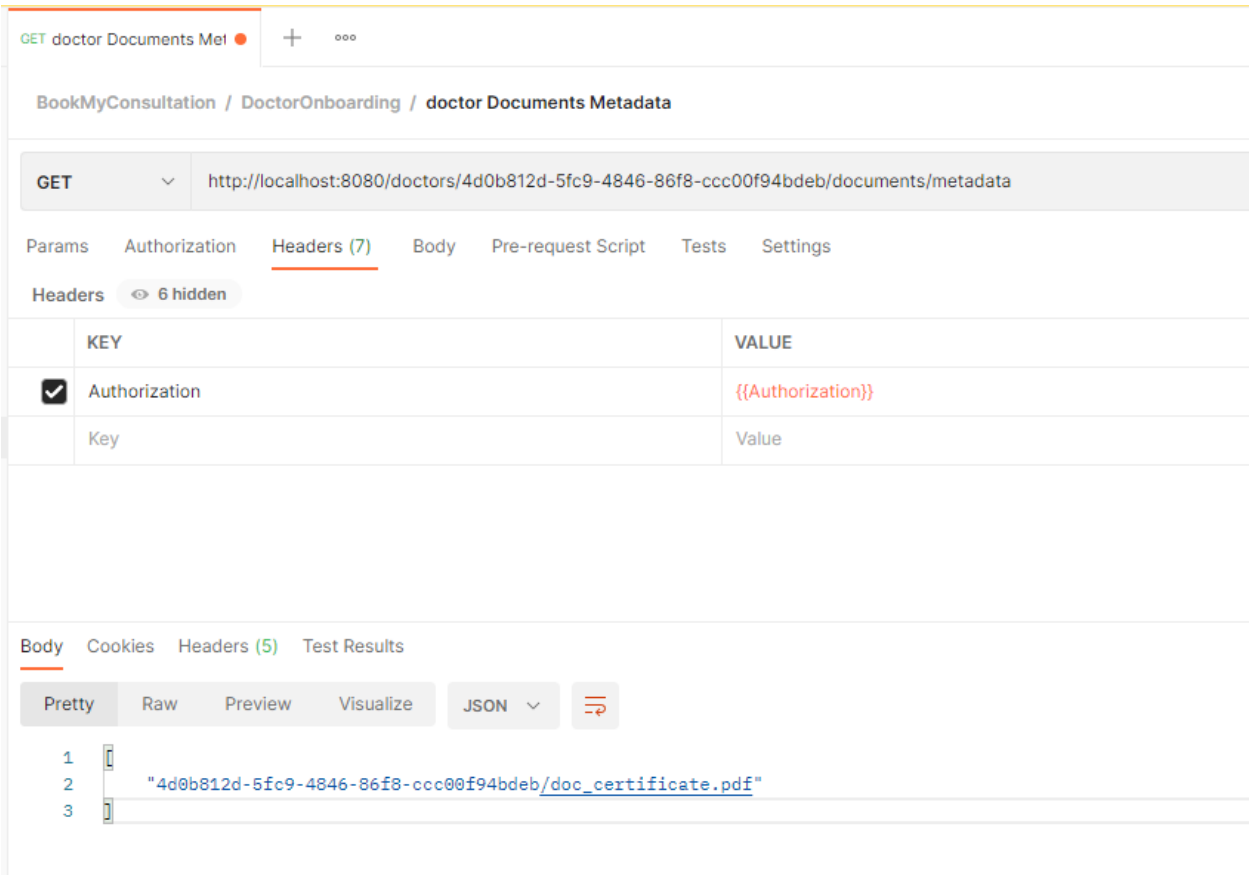
Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

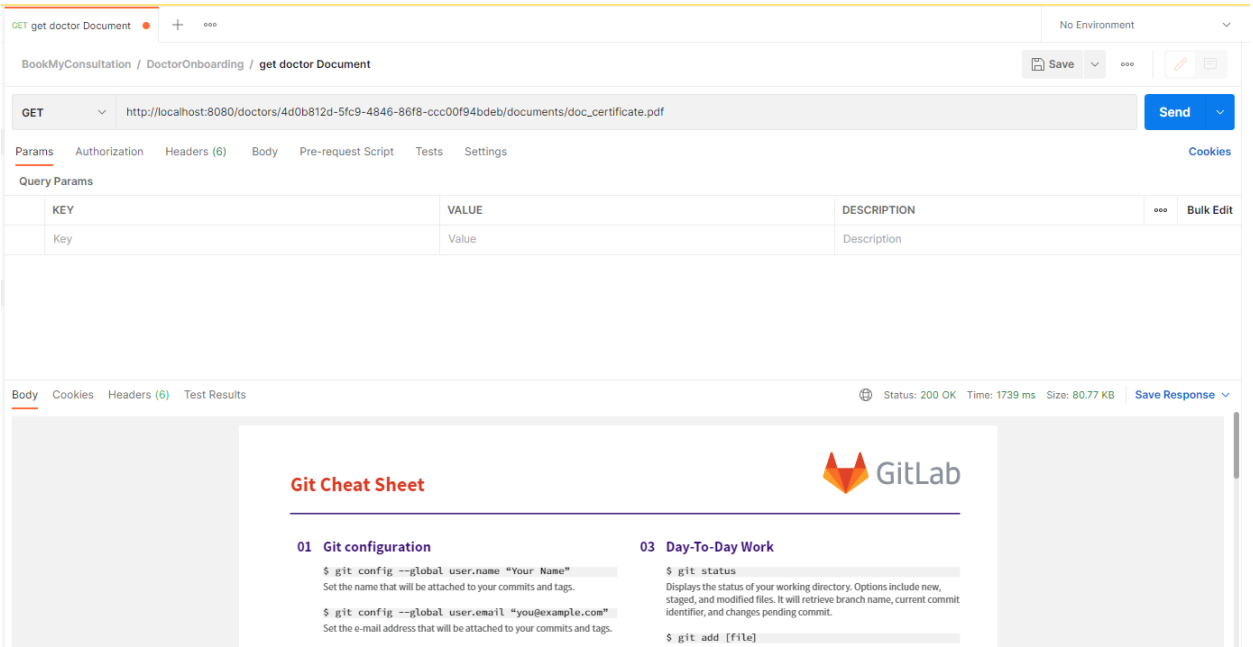
```
{
    "id": "d0bc9c03-9d07-4688-b8f0-edb4fe47d9d5",
    "firstName": "Suman",
    "lastName": "K",
    "mobile": "8105648796",
    "dob": "1991-04-18",
    "emailId": "sandesh.ayyod89@gmail.com",
    "pan": "AZIPA2000B",
    "speciality": "DENTIST",
    "registrationDate": "2022-04-06",
    "status": "REJECTED",
    "approvedBy": "sandesh",
    "approverComments": "rejected doctor",
    "verificationDate": "2022-04-06"
}
```

me **Registration Rejected** - Hi Dr sandesh.ayyod89@gmail.com Unfortunately your Registration is Rejected. Please check with BMC customer-support Regards, BMC

- Get document metadata



- Download uploaded document



- User Registration

BookMyConsultation / UserOnboarding / **enroll user**

POST http://localhost:8083/users

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

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

```
{
    "firstName": "Avish",
    "lastName": "K",
    "dob": "1956-07-01",
    "emailId": "sandesh.ayyod89@gmail.com",
    "mobile":"8105232606"
}
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "id": "644c84c3-e8a7-4ba2-887b-490b6603c228",
    "firstName": "Avish",
    "lastName": "K",
    "dob": "1956-07-01",
    "emailId": "sandesh.ayyod89@gmail.com",
    "mobile": "8105232606",
    "createdDate": "2022-04-06"
}
```

me **Welcome User** - Hi sandesh.ayyod89@gmail.com Congratulations !!! You are registered as a user in BMC app Regards, BMC

- User retrieval

GET fetch user

BookMyConsultation / UserOnboarding / **fetch user**

GET http://localhost:8083/users/644c84c3-e8a7-4ba2-887b-490b6603c228

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

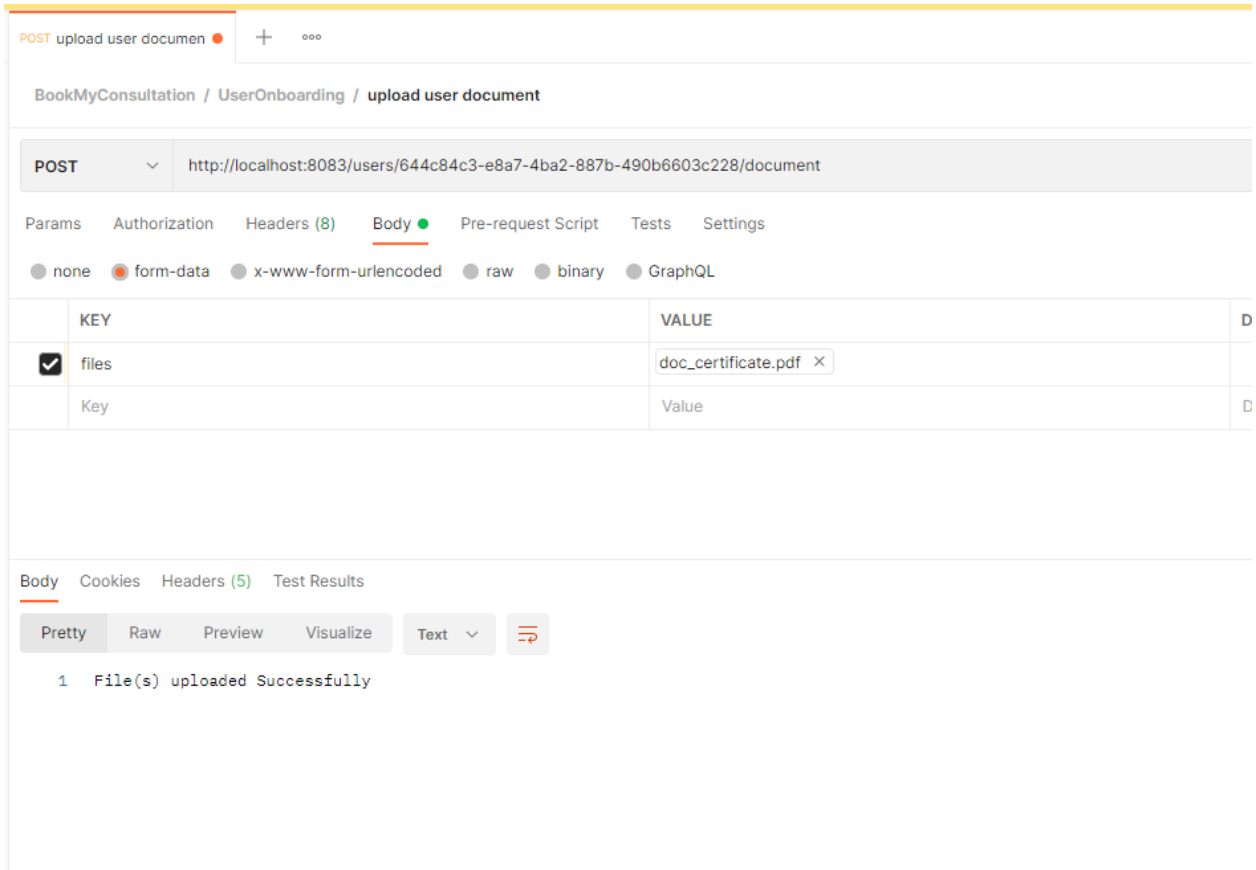
KEY	VALUE
Key	Value

Body Cookies Headers (5) Test Results

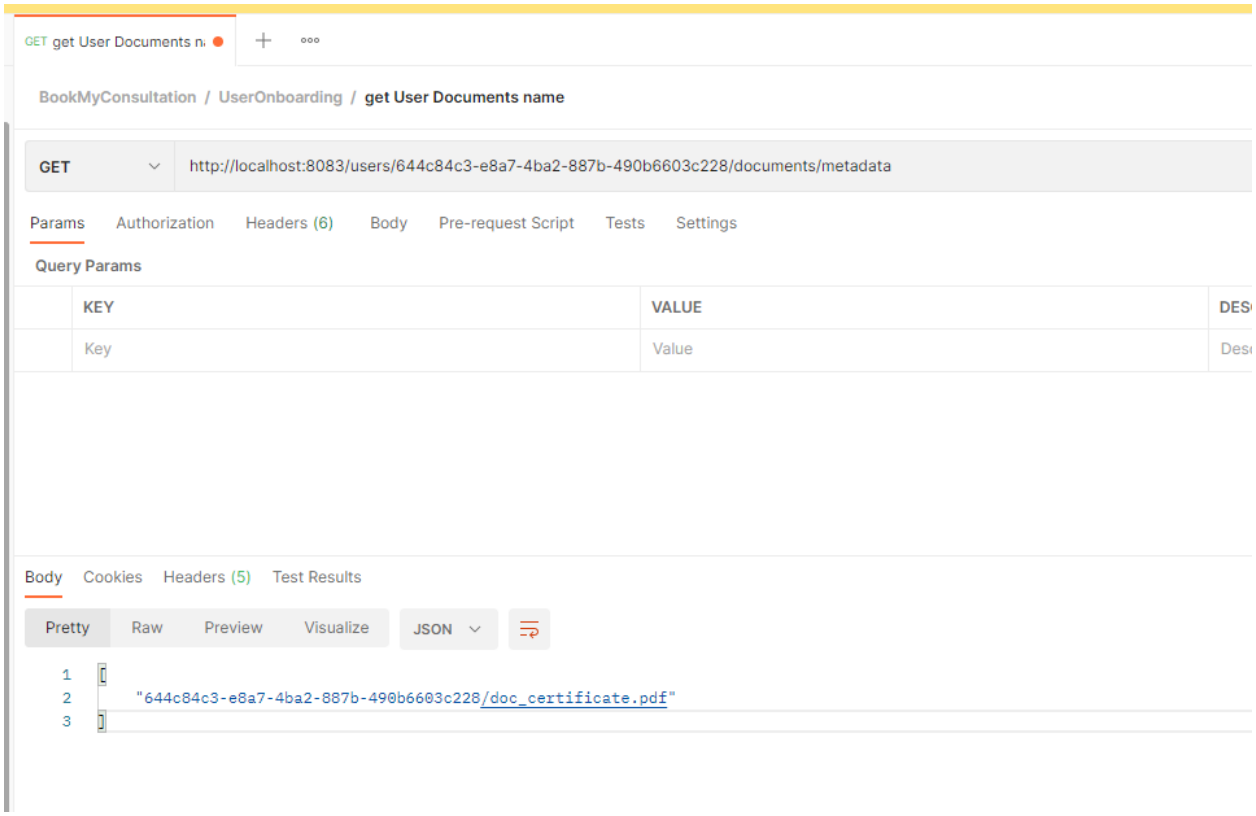
Pretty Raw Preview Visualize JSON


```
{
    "id": "644c84c3-e8a7-4ba2-887b-490b6603c228",
    "firstName": "Avish",
    "lastName": "K",
    "dob": "1956-07-01",
    "emailId": "sandesh.ayyod89@gmail.com",
    "mobile": "8105232606",
    "createdDate": "2022-04-06"
}
```

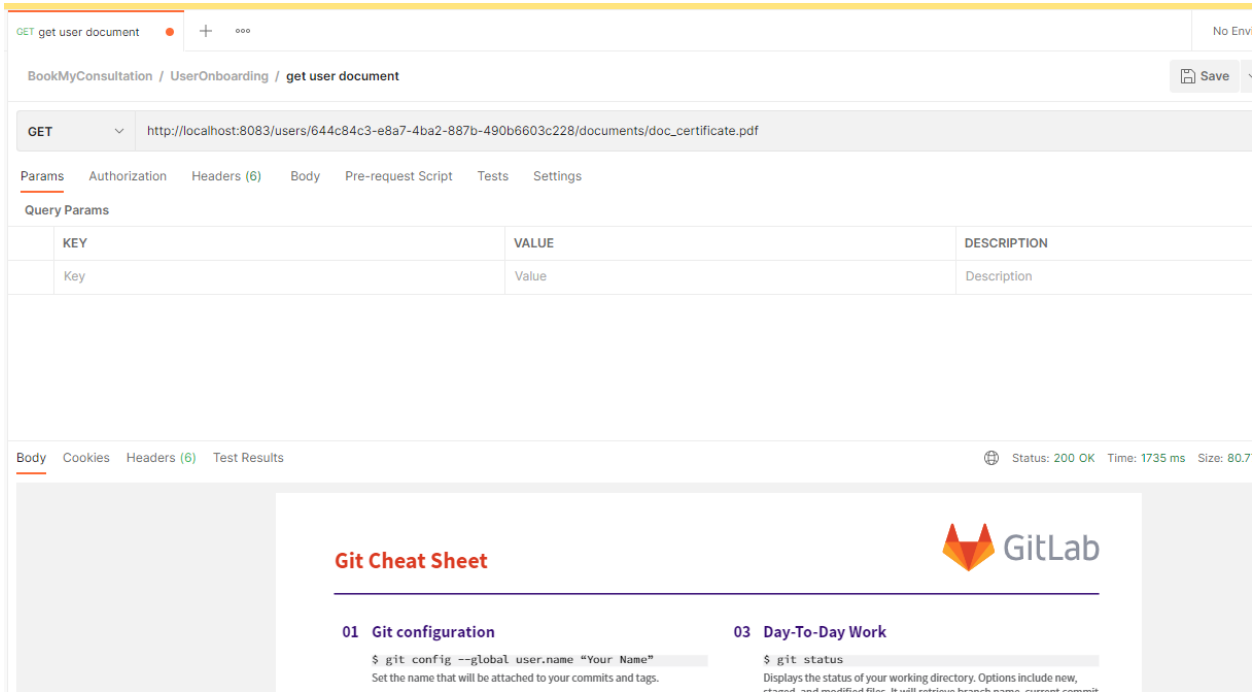

- User document upload



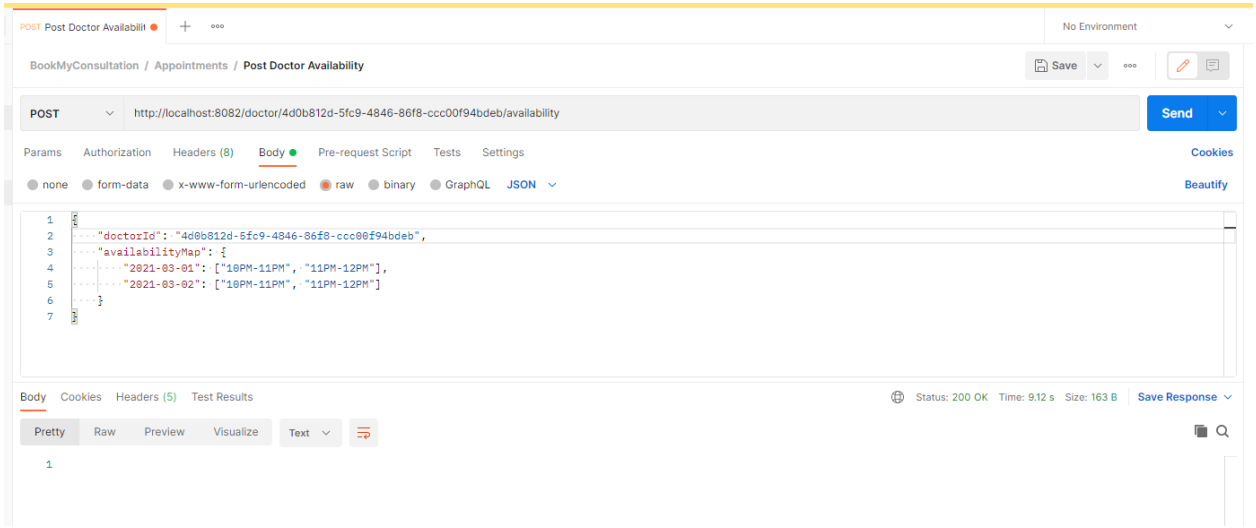
- User document get names



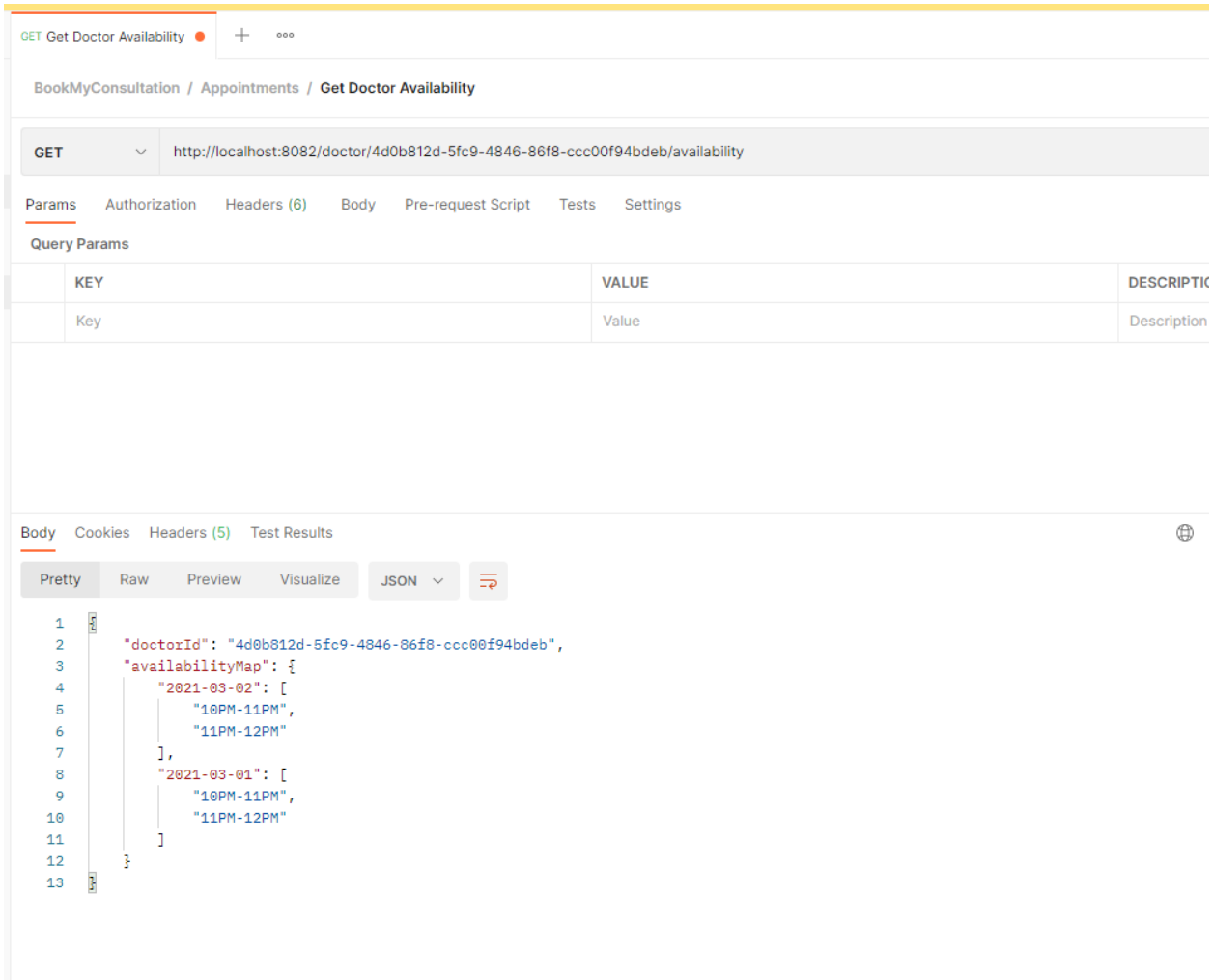
- User document download



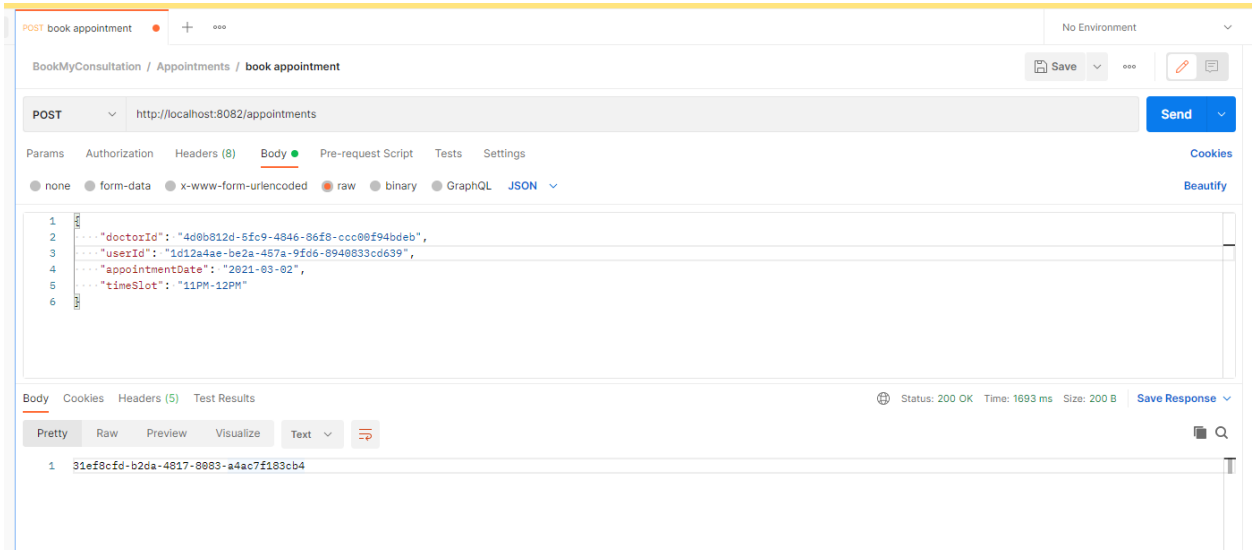
- Update doctor availability

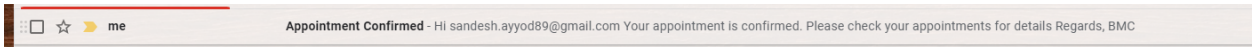


- Get doctor availability



- Book appointment





- Get appointment detail

GET get appointment

BookMyConsultation / Appointments / **get appointment**

GET http://localhost:8082/appointments/31ef8cfd-b2da-4817-8083-a4ac7f183cb4

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
	Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "appointmentId": "31ef8cfd-b2da-4817-8083-a4ac7f183cb4",
    "doctorId": "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb",
    "userId": "1d12a4ae-be2a-457a-9fd6-8940833cd639",
    "userEmailId": "sandesh.ayyod89@gmail.com",
    "appointmentDate": "2021-03-02",
    "timeSlot": "11PM-12PM"
}
```


- Get all the user appointments

GET get user appointments

BookMyConsultation / Appointments / get user appointments

GET http://localhost:8082/appointments/1d12a4ae-be2a-457a-9fd6-8940833cd639/appointments

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DE
Key	Value	De

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON


```
[
    {
        "appointmentId": "31ef8cfd-b2da-4817-8083-a4ac7f183cb4",
        "doctorId": "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb",
        "userId": "1d12a4ae-be2a-457a-9fd6-8940833cd639",
        "userEmailId": "sandesh.ayyod89@gmail.com",
        "appointmentDate": "2021-03-02",
        "timeSlot": "11PM-12PM"
    }
]
```


- Save prescriptions

POST save prescriptions

BookMyConsultation / Appointments / **save prescriptions**

POST http://localhost:8082/prescriptions

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

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


```
{
    "appointmentId": "31ef8cfd-b2da-4817-8083-a4ac7f183cb4",
    "doctorId": "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb",
    "userId": "1d12a4ae-be2a-457a-9fd6-8940833cd639",
    "diagnosis": "head-ache",
    "medicineList": [
        {
            "name": "Calpol",
            "type": "Tablet",
            "dosage": "1 week",
            "duration": "1 week",
            "frequency": "3 times a day",
            "remarks": "after food"
        },
        {
            "name": "PainKill",
            "type": "Syrup",
            "dosage": "1 week",
            "duration": "1 week",
            "frequency": "3 times a day",
```


Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "errorCode": "NOT_WORKING",
    "errorMessage": "Service not working. Please try after sometime",
    "errorFields": null
}
```


Exception because pending payment:

```
appointment-service | 2022-04-06 05:06:15.958  INFO [appointment-service,321e40b1a99151cd,321e40b1a99151cd] 1 --- [nio-8082-exec-5] c.u.a.s.impl.ConsultationServiceImpl     : Uploading consultation summary for Appointment:31ef8cfd-b2da-4817-8083-a4ac7f183cb4
appointment-service | 2022-04-06 05:06:16.191 ERROR [appointment-service,321e40b1a99151cd,321e40b1a99151cd] 1 --- [nio-8082-exec-5] c.u.a.s.impl.ConsultationServiceImpl     : Payment pending for 31ef8cfd-b2da-4817-8083-a4ac7f183cb4
appointment-service | 2022-04-06 05:06:16.197  INFO [appointment-service,321e40b1a99151cd,321e40b1a99151cd] 1 --- [nio-8082-exec-5] c.u.a.aspect.LoggingAspect               : In GlobalControllerAdvice, entering handleServiceException at time: Wed Apr 06 05:06:1
6 GMT 2022
appointment-service | 2022-04-06 05:06:16.213  INFO [appointment-service,321e40b1a99151cd,321e40b1a99151cd] 1 --- [nio-8082-exec-5] c.u.a.aspect.LoggingAspect               : In GlobalControllerAdvice, exiting handleServiceException at time: Wed Apr 06 05:06:16
 GMT 2022, total time taken: 16ms
```

- Payment-service

POST save prescriptions | POST Make Payment

BookMyConsultation / Payment / Make Payment

POST http://localhost:8086/payments?appointmentId=31ef8cfd-b2da-4817-8083-a4ac7f183cb4

Params Authorization Headers (7) 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 Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON


```
{
    "paymentId": "d41cf638-38d4-45f5-8fa2-772457e2ad3d",
    "appointmentId": "31ef8cfd-b2da-4817-8083-a4ac7f183cb4",
    "createDate": "2022-04-06T05:08:45.235777"
}
```


Payment service will take the payment and generate a payment id. And sends the payment dto to kafka.
Appointment service will update the status of appointment to payment confirmed

```
appointment-service     | 2022-04-06 05:08:49.642  INFO [appointment-service,8158d4a60b4e0c5f,0aa802434f19f9c5] 1 --- [ntainer#0-0-C-1] c.u.a.s.impl.PaymentConsumerService      : Listening to  PaymentDto(paymentId=d41cf638-38d4-45f5-8fa2-772457e2ad3d, appointmentId
=31ef8cfd-b2da-4817-8083-a4ac7f183cb4, createDate=2022-04-06T05:08:45.235777)
appointment-service     | 2022-04-06 05:08:49.873  INFO [appointment-service,8158d4a60b4e0c5f,0aa802434f19f9c5] 1 --- [ntainer#0-0-C-1] c.u.a.s.impl.PaymentConsumerService      : Updating the PaymentStatus to CONFIRMED for appointmentId:31ef8cfd-b2da-4817-8083-a4ac
7f183cb4

eureka-server           | 2022-04-06 05:08:59.119  INFO 1 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
eureka-server           | 2022-04-06 05:09:59.120  INFO 1 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
bmc-gateway             | 2022-04-06 05:10:03.487  INFO 1 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka endpoints via configuration
```


```
_id: ObjectId("624d20629c123e452bebe69e")
appointmentId: "31ef8cfd-b2da-4817-8083-a4ac7f183cb4"
doctorId: "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb"
userId: "1d12a4ae-be2a-457a-9fd6-8940833cd639"
userEmailId: "sandesh.ayyod89@gmail.com"
appointmentDate: 2021-03-02T00:00:00.000+00:00
timeSlot: "11PM-12PM"
paymentStatus: "CONFIRMED"
_class: "com.upgrad.appointmentservice.entity.Appointment"
```


- Retry consultation summary after payment

POST http://localhost:8082/prescriptions

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

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

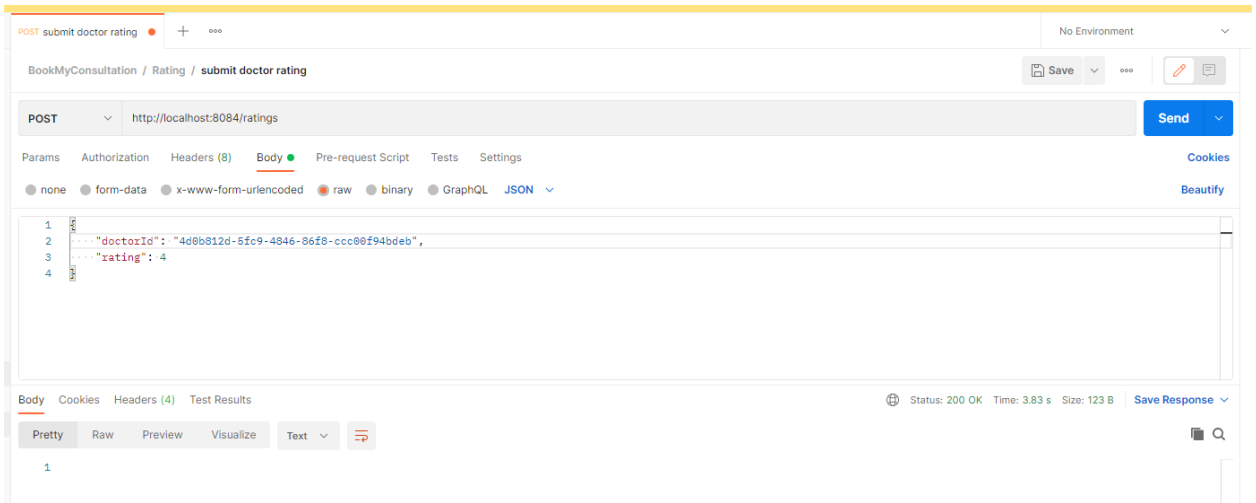
```
1  {
2      "appointmentId": "31ef8cfd-b2da-4817-8083-a4ac7f183cb4",
3      "doctorId": "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb",
4      "userId": "1d12a4ae-be2a-457a-9fd6-8940833cd639",
5      "diagnosis": "head-ache",
6      "medicineList": [
7          {
8              "name": "Calpol",
9              "type": "Tablet",
10             "dosage": "1 week",
11             "duration": "1 week",
12             "frequency": "3 times a day",
13             "remarks": "after food"
14         },
15         {
16             "name": "PainKill",
17             "type": "Syrup",
18             "dosage": "1 week",
19             "duration": "1 week",
20             "frequency": "3 times a day",
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

```
1  Prescriptions Uploaded
```

- Rating service



.

Rating service will send the rating dto to kafka, doctor-service will calculate the average and update in db

```
_id: ObjectId("624bd67821616e032bb4886c")
doctorId: "961177f3-7670-4cf1-8e6a-603d23511d5c"
rating: 4
_class: "com.upgrad.doctorservice.entity.DoctorRating"
```

```
_id: ObjectId("624d21540a3a76481d7a713e")
doctorId: "4d0b812d-5fc9-4846-86f8-ccc00f94bdeb"
rating: 4
_class: "com.upgrad.doctorservice.entity.DoctorRating"
```

```
0f94bdeb, rating=4)
eureka-server | 2022-04-06 05:12:53.055 INFO [doctor-service,093b0fdca1d19053,b6b835e30a5c5884] 1 --- [ntainer#0-0-C-1] c.u.d.service.impl.ConsumerService : Successfully updated doctor rating for DoctorId: 4d0b812d-5fc9-4846-86f8-ccc00f94bdeb
eureka-server | 2022-04-06 05:12:59.121 INFO 1 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 0ms
eureka-server | 2022-04-06 05:13:59.121 INFO 1 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 0ms
```

37. Security

Security has not been implemented. This needs OAuth2 and is not covered as part of the course

38. Things to update to run the project

In docker compose file update the following

```
S3_ACCESS_KEY: <<update iam access key for s3>>
S3_SECRET_KEY: <<update iam secret key for s3>>
MONGODB_HOST_NAME: <<update ec3 instance public ipv4 >>
SMTP_AWS_ENDPOINT: email-smtp.us-east-1.amazonaws.com
```

```
SMTP_AWS_USERNAME: <<update smtp aws user name generated >>
SMTP_AWS_PASSWORD: <<update smtp aws password generated >>
SMTP_AWS_FROMEMAIL: <<update verified email id in ses >>
```