

BookMyConsultation Application

1. Register: <http://localhost:8080/users/register>

The screenshot shows the Postman interface for a POST request to `http://localhost:8080/users/register`. The request body is a JSON object with the following fields:

```
1 {
2   "firstName": "fname",
3   "lastName": "lname",
4   "dob": "1903-08-06",
5   "mobile": "1234567890",
6   "password": "test",
7   "emailId": "test@gmail.com"
8 }
```

The response status is 200 OK, with a time of 49 ms and a size of 843 B. The response body is a JSON object with the following fields:

```
1 {
2   "firstName": "fname",
3   "lastName": "lname",
4   "dob": "1903-08-06",
5   "mobile": "1234567890",
6   "emailId": "test@gmail.com",
7   "password": "*****",
8   "created": "2020-08-06T10:00:00.000Z",
9   "salt": "BfE3xVem9G9k6KXLjpy5LuI8sQz5JH28TjhQuS0RI="
}
```

2. Login: <http://localhost:8080/auth/login>

The screenshot shows the Postman interface for a POST request to `http://localhost:8080/auth/login`. The request is configured with Basic Authentication, with the username `test@gmail.com` and password `test`. The response status is 200 OK, with a time of 90 ms and a size of 1.07 KB. The response body is a JSON object with the following fields:

```
1 {
2   "id": "test@gmail.com",
3   "firstName": "fname",
4   "lastName": "lname",
5   "emailAddress": "test@gmail.com",
6   "mobilePhoneNumber": "1234567890",
7   "lastLoginTime": null,
8   "accessToken": "eyJraWQiOiI2MmQ1YzFiMC0xZGZjLTlmOGItOWVlYy1kZWZ0M0ZDA2ODIiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJhdWQiOiJ0ZXN0QGdtYXNpbC5jb20iLCJpc3MiOiJodHRwczovL2Jvb2tteWVbnN1bHRhdGlvbi5jb20iLCJleHAiOiJlZD0iE200IAsIm1hdC"
}
```

Login Failed

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists collections, including "bookmyconsultation". The main panel displays a POST request to `http://localhost:8080/auth/login` using Basic Auth. The username is `test@gmail.com` and the password is `test1`. The "Show Password" checkbox is checked. The response status is `401 Unauthorized` with a time of `20 ms` and size of `703 B`. The response body is shown in JSON format:

```
1 {
2   "code": "USR-003",
3   "message": "Password match failed",
4   "root_cause": null
5 }
```

3. Logout: <http://localhost:8080/auth/logout>

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists collections, including "bookmyconsultation". The main panel displays a POST request to `http://localhost:8080/auth/logout`. The response status is `200 OK` with a time of `41 ms` and size of `582 B`. The response body is shown in Text format:

```
1
```

4. Get User: <http://localhost:8080/users/test@gmasil.com>

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists collections, including "bookmyconsultation". The main panel shows a GET request to `http://localhost:8080/users/test@gmasil.com`. The response is a JSON object with the following details:

```
{
  "firstName": "fname",
  "lastName": "lname",
  "dob": "1903-08-06",
  "mobile": "1234567890",
  "emailId": "test@gmasil.com",
  "password": "80D946D5652D9021",
  "createdDate": "2023-05-01",
  "salt": " //BfE3xVem9G9k6KXLjpy5LuI8sQz5JH28TjhQuS0RI="
}
```

The status is 200 OK, Time: 50 ms, Size: 887 B.

Get User Failed when user not present

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists collections, including "bookmyconsultation". The main panel shows a GET request to `http://localhost:8080/users/test1@gmasil.com`. The response is a 404 Not Found error with the following details:

```
{
  "code": "GEN-001",
  "message": "Resource not available",
  "root_cause": "com.upgrad.bookmyconsultation.exception.ResourceUnavailableException: Resource not available\r\n\tat com.upgrad.bookmyconsultation.service.UserService.lambda$getUser$(UserService.java:38)\r\n\tat java.base/java.util.Optional.orElseThrow(Optional.java:403)\r\n\tat com.upgrad.bookmyconsultation.service.UserService.getUser(UserService.java:38)\r\n\tat com.upgrad.bookmyconsultation.controller.UserAdminController.getUser(UserAdminController.java:27)\r\n\tat java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(Native Method)\r\n\tat java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)\r\n\tat java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)\r\n\tat java.base/java.lang.reflect.Method.invoke(Method.java:568)\r\n\tat org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:197)\r\n\tat org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:141)\r\n\tat org.springframework.web.servlet.mvc.method."
}
```

The status is 404 Not Found, Time: 17 ms, Size: 7.51 KB.

5. Get User Appointments: <http://localhost:8080/users/test@gmail.com/appointments>

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists collections, including "bookmyconsultation" which contains several endpoints. The "GET get User appointments" endpoint is selected. The main panel shows the request details for the URL "http://localhost:8080/users/test@gmail.com/appointments". The request is a GET method. The response is displayed in the "Body" tab, showing a JSON object with appointment details. The status is 200 OK, and the response time is 15 ms.

bookmyconsultation / get User appointments

GET http://localhost:8080/users/test@gmail.com/appointments

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

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (12) Test Results Status: 200 OK Time: 15 ms Size: 908 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "appointmentId": "7d4597f1-3486-4a2d-8089-f3b61ce3fad0",
3   "doctorId": "UUID-34",
4   "doctorName": "Meghan Jordan",
5   "userId": "test@gmail.com",
6   "userName": "fname",
7   "userEmailId": "test@gmail.com",
8   "timeSlot": "05PM-06PM",
9   "appointmentDate": "2021-05-26",
10  "symptoms": "TEST",
11  "priorMedicalHistory": "NA"
12 }
```

Get User Appointment when no appointment is present

The screenshot shows the Postman interface with the same workspace and collection as the previous image. The "GET get User appointments" endpoint is selected. The main panel shows the request details for the URL "http://localhost:8080/users/test@gmail.com/appointments". The request is a GET method. The response is displayed in the "Body" tab, showing an empty JSON object. The status is 200 OK, and the response time is 14 ms.

bookmyconsultation / get User appointments

GET http://localhost:8080/users/test@gmail.com/appointments

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

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (12) Test Results Status: 200 OK Time: 14 ms Size: 625 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {}
```


6. Add Doctor: <http://localhost:8080/doctors>

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists collections, including "bookmyconsultation". The main panel shows a POST request to "http://localhost:8080/doctors". The request body is a JSON object with the following fields: firstName, lastName, dob, mobile, password, emailId, pan, and address. The response is a JSON object with an id, firstName, lastName, speciality, dob, and address. The status is 200 OK, and the time is 81 ms.

```
POST http://localhost:8080/doctors
```

```
{
  "firstName": "Sandeep",
  "lastName": "Chouhan",
  "dob": "1903-08-06",
  "mobile": "1234567890",
  "password": "test",
  "emailId": "test@gmail.com",
  "pan": "EDLSS5080L",
  "address": {}
}
```

```
{
  "id": "57e9ac01-4bc9-4c0b-845b-624254cd22a4",
  "firstName": "Sandeep",
  "lastName": "Chouhan",
  "speciality": "GENERAL_PHYSICIAN",
  "dob": "1903-08-06",
  "address": {
    "id": "57e9ac01-4bc9-4c0b-845b-624254cd22a4",
    "addressLine1": "add1",
  }
}
```

7. Get a Doctor: <http://localhost:8080/doctors/UUID-2>

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists collections, including "bookmyconsultation". The main panel shows a GET request to "http://localhost:8080/doctors/UUID-2". The response is a JSON object with an id, firstName, lastName, speciality, dob, and address. The status is 200 OK, and the time is 15 ms.

```
GET http://localhost:8080/doctors/UUID-2
```

```
{
  "id": "UUID-2",
  "firstName": "Blossom",
  "lastName": "Valentine",
  "speciality": "PULMONOLOGIST",
  "dob": "2021-09-16",
  "address": {
    "id": "UUID-2",
    "addressLine1": "103-4867 Nullam Ave",
    "addressLine2": "-45.86534, 146.33318",
    "city": "Patalillo",
    "state": "San José",
    "postcode": "L2S 2T0"
  }
}
```

Get Doctor when doctor not present with id

The screenshot shows a Postman workspace with a collection named 'bookmyconsultation'. The selected endpoint is 'GET get a doctor' with the URL 'http://localhost:8080/doctors/UUID-1014'. The request is a GET method. The response is a 404 Not Found status with a time of 13 ms and a size of 7.5 KB. The response body is in JSON format and contains the following error message:

```
{
  "code": "GEN-001",
  "message": "Doctor not found with id UUID-1014",
  "root_cause": "com.upgrad.bookmyconsultation.exception.ResourceUnavailableException: Doctor not found with id UUID-1014\r\n\tat com.upgrad.bookmyconsultation.service.DoctorService.lambda$getDoctor$0(DoctorService.java:65)\r\n\tat java.base/java.util.Optional.orElseThrow(Optional.java:403)\r\n\tat com.upgrad.bookmyconsultation.service.DoctorService.getDoctor(DoctorService.java:65)\r\n\tat com.upgrad."
}
```

8. Book an Appointment: <http://localhost:8080/appointments>

The screenshot shows a Postman workspace with a collection named 'bookmyconsultation'. The selected endpoint is 'POST book an appointment' with the URL 'http://localhost:8080/appointments'. The request is a POST method. The request body is in JSON format and contains the following appointment details:

```
{
  "doctorId": "UUID-34",
  "doctorName": "Meghan Jordan",
  "userId": "test@gmail.com",
  "userName": "fname",
  "userEmailId": "test@gmail.com",
  "timeSlot": "05PM-06PM",
  "appointmentDate": "2021-05-26",
  "createdDate": "",
  "symptoms": "TEST",
  "priorMedicalHistory": "NA"
}
```

The response is a 201 Created status with a time of 35 ms and a size of 664 B. The response body is in JSON format and contains the following appointment ID:

```
{
  "appointmentId": "7d4597f1-3485-4a2d-8089-f3b61ce3fad0"
}
```

9. Get an Appointment: <http://localhost:8080/appointments/UUID-1>

The screenshot shows the Postman interface with a workspace named "My Workspace". On the left, a collection named "bookmyconsultation" is expanded, showing various endpoints. The "GET get an appointment" endpoint is selected. The main panel shows the request details for the URL `http://localhost:8080/appointments/7d4597f1-3485-4a2d-8089-f3b61ce3fad0`. The request is a GET method with no parameters, headers, or body. The response is a JSON object with the following structure:

```
1 {
2   "firstName": "fname",
3   "lastName": "lname",
4   "dob": "1903-08-06",
5   "mobile": "1234567890",
6   "password": "test",
7   "emailId": "test@gmail.com",
8 }
```

The status bar at the bottom indicates a 200 OK response with a time of 24 ms and a size of 909 B.

Get an Appointment when appointment not found

The screenshot shows the Postman interface with the same workspace and collection. The "GET get an appointment" endpoint is selected, but the URL is `http://localhost:8080/appointments/1d456bdf-5555-4b36-a1aa`. The response is a 404 Not Found status with a time of 19 ms and a size of 7.52 KB. The response body is a JSON object with the following structure:

```
1 {
2   "code": "GEN-001",
3   "message": "Appointment not found.",
4   "root_cause": "com.upgrad.bookmyconsultation.exception.ResourceUnavailableException: Appointment not found.
  \r\n\tat com.upgrad.bookmyconsultation.service.AppointmentService.lambda$getAppointment$0(AppointmentService.
  java:69)\r\n\tat java.base/java.util.Optional.orElseThrow(Optional.java:483)\r\n\tat com.upgrad."
```

10. Post a Rating: <http://localhost:8080/ratings>

bookmyconsultation / post a rating

POST <http://localhost:8080/ratings> Send

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

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

```
1 {
2   "appointmentId": "f646f379-bc4b-47e3-a335-0fe4704931cb",
3   "doctorId": "UUID-34",
4   "rating": "1",
5   "comments": "asdad"
6 }
```

Body Cookies Headers (11) Test Results Status: 200 OK Time: 36 ms Size: 582 B Save as Example

Pretty Raw Preview Visualize Text

1

11. Get All Doctor: <http://localhost:8080/doctors?speciality=CARDIOLOGIST>

bookmyconsultation / get all doctor

GET <http://localhost:8080/doctors?speciality=CARDIOLOGIST> Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

Query Params

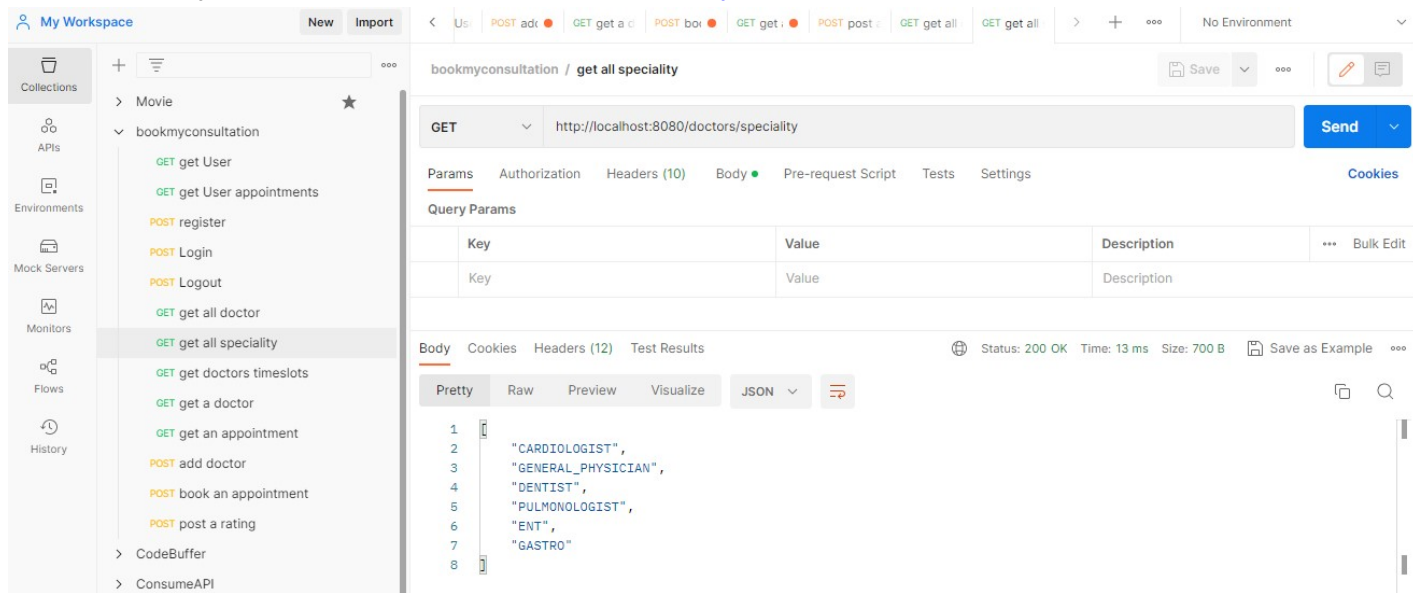
Key	Value	Description
speciality	CARDIOLOGIST	

Body Cookies Headers (12) Test Results Status: 200 OK Time: 56 ms Size: 8.48 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "UUID-42",
3   "firstName": "Alexis",
4   "lastName": "Rodriguez",
5   "speciality": "CARDIOLOGIST",
6   "dob": "2020-11-07",
7   "address": {
8     "id": "UUID-42",
9     "addressLine1": "P.O. Box 120, 1291 Non St.",
10    "addressLine2": "46.75055, 83.74275",
11    "city": "Avignon",
12    "state": "PR",
13    "postcode": "49003"
14  }
```


12. Get All Specialty: <http://localhost:8080/doctors/speciality>



bookmyconsultation / get all speciality

GET <http://localhost:8080/doctors/speciality> Send

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

Query Params

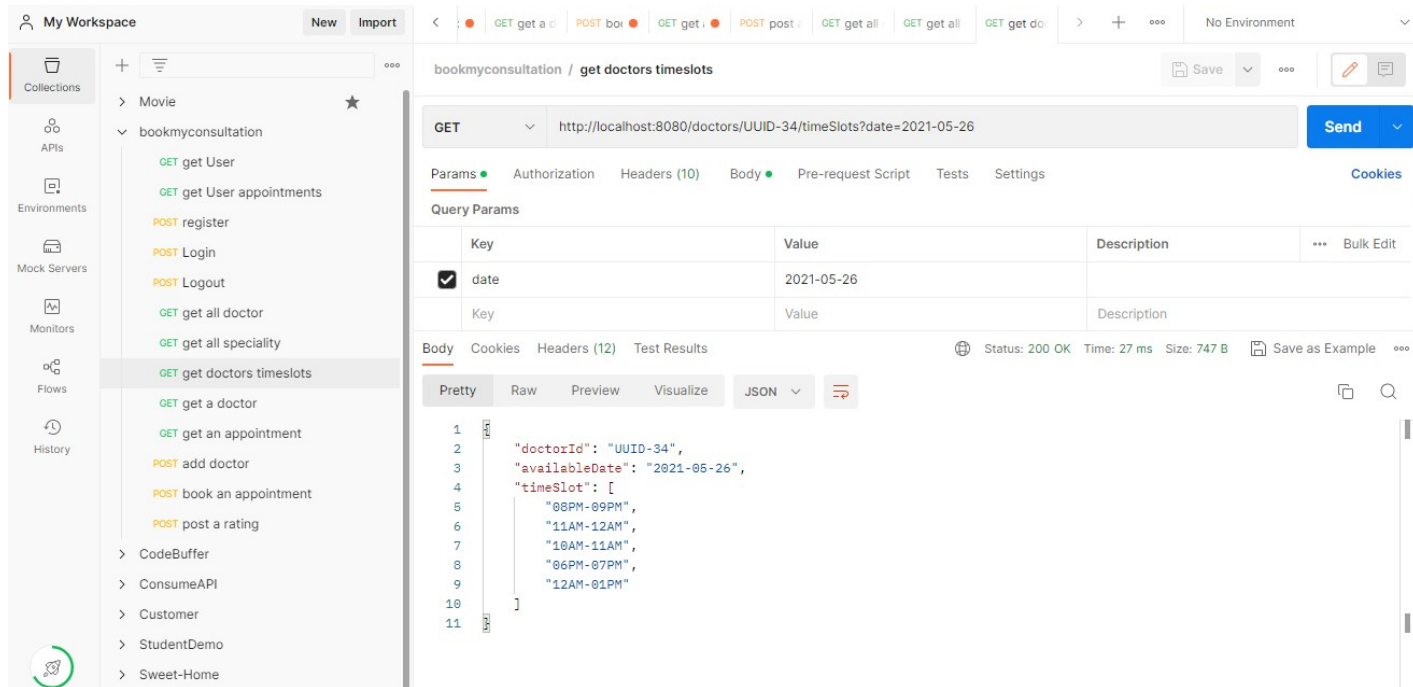
Key	Value	Description
Key	Value	Description

Body Cookies Headers (12) Test Results Status: 200 OK Time: 13 ms Size: 700 B Save as Example

Pretty Raw Preview Visualize JSON Copy

```
1 [
2   "CARDIOLOGIST",
3   "GENERAL_PHYSICIAN",
4   "DENTIST",
5   "PULMONOLOGIST",
6   "ENT",
7   "GASTRO"
8 ]
```

13. Get Doctor Timeslots: <http://localhost:8080/doctors/UUID-34/timeSlots?date=2021-05-26>



bookmyconsultation / get doctors timeslots

GET <http://localhost:8080/doctors/UUID-34/timeSlots?date=2021-05-26> Send

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

Query Params

Key	Value	Description
<input checked="" type="checkbox"/> date	2021-05-26	
Key	Value	Description

Body Cookies Headers (12) Test Results Status: 200 OK Time: 27 ms Size: 747 B Save as Example

Pretty Raw Preview Visualize JSON Copy

```
1 {
2   "doctorId": "UUID-34",
3   "availableDate": "2021-05-26",
4   "timeSlot": [
5     "08PM-09PM",
6     "11AM-12AM",
7     "10AM-11AM",
8     "06PM-07PM",
9     "12AM-01PM"
10  ]
11 }
```

Database Schema and Tables

Navigator: SCHEMAS

Filter objects

- census
- consultation
 - Tables
 - address
 - appointment
 - doctor
 - rating
 - user
 - user_auth_token
 - Views
 - Stored Procedures
 - Functions
- fooddelivery
- org
- sakila
- springboot
- sys
- world

Administration Schemas

Information

address appointment doctor rating user user_auth_token

Limit to 1000 rows

1 • SELECT * FROM consultation.address;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

id	address_line1	address_line2	city	postcode	state
57e9ac01-4bc9-4c0b-845b-624254cd22a4	add1	add1	BHN	80066	TN
UUID-1	P.O. Box 494, 5698 Felis. Ave	-12.24821, 86.15196	Vienna	80066	Wie
UUID-10	3578 Nisl Street	80.86218, 49.58877	Cork	67297	Munster
UUID-100	8068 Pulvinar Rd.	4.15097, -72.68161	Murree	88570-68315	PU
UUID-11	P.O. Box 771, 4467 Id Avenue	-79.27567, 36.24963	Bello	33457	Antioquia
UUID-12	5865 Purus St.	-58.41035, -68.5989	Gresham	817559	Oregon
UUID-13	665-7419 Sit Street	-52.90714, 61.48712	Alhué	RB4 7PG	RM
UUID-14	P.O. Box 860, 9270 Lacus. Street	16.44007, 25.97876	Cork	29966	M
UUID-15	512-2479 Sit Rd.	52.92, -1.72489	Cartagena	45763	Bolívar
UUID-16	Ap #675-9844 Ac St.	35.98661, 132.71938	Muzaffarpur	99585	Bihar
UUID-17	5900 Neque Street	18.22648, -156.96882	Sagamu	208985	Ogun
UUID-18	4089 Consectetur Rd.	29.17934, 67.56259	Wroclaw	9448	Dolnoślą...