



# POSTMAN TIMELY

SUNIL YADAV

# TABLE OF CONTENT

Table of Content	2
Assumptions	3
User Registration	4
User Login	5
User Logout	5
Get All Slots	6
Get Available Slots of Others	7
Make Available Slots	8
Book Slots	9
Get All Your BOOKINGS	10
Get All Your Booked Slots	11
Development Sprints	12
POC Images	13
Technology User	22

# ASSUMPTIONS

*No Provision to Views Historical Data -*

*Time Slot is Fixed for 1Hr.*

**Time Slot** - *as Per Data Format*

00:00 - 01:00	0	13:00 - 14:00	13
01:00 - 02:00	1	14:00 - 15:00	14
02:00 - 03:00	2	15:00 - 16:00	15
03:00 - 04:00	3	16:00 - 17:00	16
04:00 - 05:00	4	17:00 - 18:00	17
05:00 - 06:00	5	18:00 - 19:00	18
06:00 - 07:00	6	19:00 - 20:00	19
07:00 - 08:00	7	20:00 - 21:00	20
08:00 - 09:00	8	21:00 - 22:00	21
09:00 - 10:00	9	22:00 - 23:00	22
10:00 - 11:00	10	23:00 - 00:00	23
11:00 - 12:00	11		
12:00 - 13:00	12		

*By Default -No Slot will be Available for a Day to User-*

# USER REGISTRATION

API	http://{IP}/v1/user/register
Request Method	POST
Data_Format	{ "username": "sunil016@yahoo.com", "password": "12345" }
Response Format	{ "timestamp": "2020-04-02T15:36:52.424+0000", "message": "User Created Successfully" }

## Username & Password Criteria -

- 1. Check for Username Criteria. (Should be Email)
- 2. Check for Password Criteria. (Min. 6 Characters)

# USER LOGIN

API	http://{IP}/v1/user/login
Request Method	GET
Response Format	<pre>{   "timestamp": "2020-04-02T15:36:52.424+0000",   "message": "Successful Login" }</pre>

Need to Pass Username and Password in  
Header - "Authorization" - Basic Auth

\*\* API Just verifies that you have entered valid credentials  
and your request is and will be passing through Spring  
Security.

# USER LOGOUT

API	http://{IP}/v1/logout
Request Method	GET
Response Format	<pre>{   "timestamp": "2020-04-02T15:36:52.424+0000",   "message": "Successful Logout" }</pre>

# GET ALL SLOTS

API	http://{IP}/v1/slot/getAllSlots
Request Method	GET
Data_Format	-NA-
Response Format	<pre>{   "timestamp": "2020-04-02T15:36:52.424+0000",   "message": "No Slot Available, Please Make I t Available" }</pre>

*\*\* By Default No Slot will be Available*

API	http://{IP}/v1/slot/getAllSlots
Request Method	GET
Data_Format	-NA-
Response Format	<pre>{   "timestamp": "2020-04-02T15:36:52.424+0000",   "message": [10,12] }</pre>

*\*\* API fond Slot 10 & Slot 12 is available, i.e timestamp from 10:00-11:00 & 12:00-13:00 is available.*

# GET AVAILABLE SLOTS OF OTHERS

API	http://{IP}/v1/slot/getAllSlots/{user}
Request Method	GET
Data_Format	-NA-
Response Format	{ "timestamp": "2020-04-02T15:36:52.424+0000", "message": [1,20] }

\*\*Here user is the username of other person to whom you like to book an appointment with I.e for sanjay01@yahoo.com.

# MAKE AVAILABLE SLOTS

API	http://{IP}/v1/slot/setSlotAvailable/{id}
Request Method	PUT
Data_Format	-NA-
Response Format	<pre>{   "timestamp": "2020-04-02T15:36:52.424+0000",   "message": "Slot 10 is Available Now" }</pre>

**\*\*Here id is the TimeStamp - \_ID I.e for 1200:1300 \_ID is 12.**

API api, mark slot available in your profile, so that other person can book an appointment with you at this slot.

*By Default -No Slot will be Available for a Day to User-*



# BOOK SLOTS

API	http://{IP}/v1/booking/bookSlot
Request Method	POST
Data_Format	{ "to": " <u>sanjay01@yahoo.com</u> ", "slot": 10, "description": "Medical Checkup" }
Response Format	{ "timestamp": "2020-04-02T15:36:52.424+0000", "message": "Slot 10 Booked with <u>sanjay01@yahoo.com</u> " }

**\*\* User X is Making Request to Book an Appointment/Slot to user sanjay01@yahoo.com**

API	http://{IP}/v1/booking/bookSlot
Request Method	POST
Data_Format	{ "to": " <u>sanjay01@yahoo.com</u> ", "slot": 10, "description": "Medical Checkup" }
Response Format	{ "timestamp": "2020-04-02T15:36:52.424+0000", "message": "Slot 10 for <u>sanjay01@yahoo.com</u> Already Book" }

**\*\*Api also check for Availability of Slot.**

# GET ALL YOUR BOOKINGS

API	http://{IP}/v1/booking/getAllBookings
Request Method	GET
Data_Format	-NA-
Response Format	<pre>{   "timestamp": "2020-04-02T16:17:29.408+0000",   "message": [     {       "bookingId": "94704278_10_126",       "to": "sanjay01@yahoo.com",       "from": "sunil016@yahoo.com",       "slot": 10,       "description": "Medical Checkup"     }   ] }</pre>

*\*\* API fetches for all the bookings that you have made to others.*

# GET ALL YOUR BOOKED SLOTS

API	http://{IP}/v1/slot/getAllBookedSlots
Request Method	GET
Data_Format	-NA-
Response Format	<pre>{   "timestamp": "2020-04-02T16:23:02.669+0000",   "message": [     {       "bookingId": "94704278_10_126",       "to": "sanjay01@yahoo.com",       "from": "sunil016@yahoo.com",       "slot": 10,       "description": "Medical Checkup"     }   ] }</pre>

*\*\* API fetches for all the booked appointment for ur profile.. i.e all the booking done for you.*

# DEVELOPMENT SPRINTS

*Day - 1 :: Implement Login and Registration*

*Day - 2 :: Build Database Schema for Slots and Booking.*

*Day - 3 :: Create APIs for Slots and Booking.*

*Day - 4 :: Make Testing and Class Level and Function Level Security.*

*Day - 5 :: Complete APIs Documentation.*

*Day - 6 :: Write Unit Test Cases & See Google Calendar Integration.*

*Day -7 :: Deploy Project on EC\_2.*

*\*\* Go for Distributed/Scalable MicroService Arch. \*\**

# POC IMAGES

GET

localhost/v1/user/login

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a c authorization helper.

Body

Cookies

Headers (9)

Test Results

Status: 202 Accepted

Time: 34ms

Pretty

Raw

Preview

Visualize BETA

JSON

1

2

3

4

5

{

"timestamp": "2020-04-02T15:39:28.694+0000",

"message": "Unauthorized Access Reported -- Access Denied",

"status": "UNAUTHORIZED"

}

*\*\* Trying to login with No Auth Header\*\**

POST localhost/v1/user/register

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

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

```
1 {
2   "username": "sunil016@yahoo.com",
3   "password": "12345"
4 }
```

Body Cookies Headers (9) Test Results Status: 200 OK Time: 181ms

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "timestamp": "2020-04-02T15:34:16.754+0000",
3   "message": "User Created Successfully"
4 }
```

## User Registration

POST localhost/v1/user/register

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

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

```
1 {
2   "username": "sunil016@yahoo.com",
3   "password": "12345"
4 }
```

Body Cookies Headers (9) Test Results Status: 200 OK Time: 127ms

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "timestamp": "2020-04-02T15:34:44.278+0000",
3   "message": "User Already Exists"
4 }
```

GETlocalhost/v1/user/login

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

TYPE

Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Preview Request

Username

sunil016@yahoo.com

Password

12345

☒ Show Password

Body

Cookies (1)

Headers (10)

Test Results

Status: 200 OKTime: 252ms

Pretty

Raw

Preview

Visualize BETA

JSON

1

2

3

4

```
{
  "timestamp": "2020-04-02T15:40:03.883+0000",
  "message": "Successful Login"
}
```

*Users Login*



GET

localhost/v1/slot/getAllSlots

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies (1)

Headers (9)

Test Results

Status: 200 OKTime: 36ms

Pretty

Raw

Preview

Visualize BETA

JSON

```
1 {
2   "timestamp": "2020-04-02T15:44:16.017+0000",
3   "message": "No Slot Available, Please Make It Available"
4 }
```

## Checking for all the Slots Available for the Day (Sanjay01@yahoo.com)

PUT

localhost/v1/slot/setSlotAvailable/10

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

TYPE

Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Preview Request

Username

sanjay01@yahoo.com

Password

\*\*\*\*\*

☐ Show Password

Body

Cookies (1)

Headers (10)

Test Results

Status: 200 OKTime: 203ms

Pretty

Raw

Preview

Visualize BETA

JSON

```
1 {
2   "timestamp": "2020-04-02T16:51:06.874+0000",
3   "message": "Slot 10 is Available Now"
4 }
```

## Making Slot - 10 Available for User (Sanjay01@yahoo.com)

GET localhost/v1/slot/getAllSlots

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (9) Test Results Status: 200 OK Time: 14ms

Pretty Raw Preview Visualize BETA JSON ▼

```
1 {
2   "timestamp": "2020-04-02T16:51:34.106+0000",
3   "message": [
4     10
5   ]
6 }
```

*Checking for all the Slots Available for the Day  
(Sanjay01@yahoo.com)*

GET localhost/v1/booking/getAllBookings

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (10) Test Results Status: 200 OK Time: 245ms

Pretty Raw Preview Visualize BETA JSON ▼

```
1 {
2   "timestamp": "2020-04-02T16:17:29.408+0000",
3   "message": [
4     {
5       "bookingId": "94704278_10_126",
6       "to": "sanjay01@yahoo.com",
7       "from": "sunil016@yahoo.com",
8       "slot": 10,
9       "description": "Medical Checkup"
10    }
11  ]
12 }
```

*Fetching all the Bookings Made by  
sunil016@yahoo.com*

*\*\* All the User Identification is Made by Server via Security Module \*\**

GET

localhost/v1/slot/getAllSlots/sanjay01@yahoo.com

Send

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies (1)

Headers (10)

Test Results

Status: 200 OK

Time: 15ms

Size: 403 B

Save

Pretty

Raw

Preview

Visualize BETA

JSON

1

2

3

4

5

6

7

{

"timestamp": "2020-04-03T06:13:14.003+0000",

"message": [

1,

12

]

}

*Fetching all the Available Slots for sanjay01@yahoo.com, so that I can make booking.*

*\*Result shows 01:00 to 02:00 and 12:00 to 13:00 is available.*

My Workspace Invite

GET L GET L PUT L GET L GET ... POST ... PUT L POST L X

No Environment

Untitled Request Comments (0)

POST localhost/v1/booking/bookSlot/ Send Save

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

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

```
1 {
2   "to": "sanjay01@yahoo.com",
3   "slot": 10,
4   "description": "Medical Checkup"
5 }
```

Body Cookies (1) Headers (9) Test Results Status: 200 OK Time: 157ms Size: 393 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "timestamp": "2020-04-02T16:09:46.730+0000",
3   "message": "Slot 10 Booked with sanjay01@yahoo.com"
4 }
```

*Making a Booking for Slot 10 to  
sanjay01@yahoo.com from sunil016@yahoo.com*

GET

localhost/v1/slot/getAllBookedSlots

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

TYPE

Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Preview Request

Username

sanjay01@yahoo.com

Password

.....

☐ Show Password

Body

Cookies (1)

Headers (9)

Test Results

Status: 200 OK

Time: 109ms

Pretty

Raw

Preview

Visualize BETA

JSON

1

2

3

4

5

6

7

8

9

10

11

12

```
{
  "timestamp": "2020-04-02T16:23:02.669+0000",
  "message": [
    {
      "bookingId": "94704278_10_126",
      "to": "sanjay01@yahoo.com",
      "from": "sunil016@yahoo.com",
      "slot": 10,
      "description": "Medical Checkup"
    }
  ]
}
```

*All the Slots Booked for sanjay01@yahoo.com*

# TECHNOLOGY USER

## ***Language***

Java

## ***Development Framework***

Spring Boot Framework

Spring Security

## ***Database***

Mongodb

## ***Actions***

User Registration

User Login

Spring Security

Slot Status

Make Slot Available

All Booking Status

All Booked Status

Make a Booking for Slot

Custom Exceptions Module

Custom Responses for Exceptions

JUnit-Mokito Testing

