

Uri Heroku : <https://room-meet.herokuapp.com/>

Uri MySQL : mysql://uqidid6srumulmbu3:H1u3xJPwK7uTnd0asuJU@bv7ui0dhd6fl3fngoyu-mysql.services.clever-cloud.com:3306/bv7ui0dhd6fl3fngoyu

Create DB : sequelize db:create

Running migration : sequelize db:migrate

1. Create User

melakukan upload file photo ke cloudinary, role user default 'guest'

The screenshot shows a Postman interface for a POST request to `http://localhost:4000/api/v1/user/create`. The request body is a form-data payload with the following fields:

Field	Value
file	002.png
email	user@gmail.com
password	user
role	admin

The response is a JSON object with the following structure:

```
{
  "status": 200,
  "message": "success created user",
  "data": {
    "role": "guest",
    "id": 1,
    "email": "user@gmail.com",
    "password": "$2b$10$G%gfZstYMri/rteu01Vkjul08LwbuFrehLyG8n09H3QxbIb1r8H96",
    "photo": "https://res.cloudinary.com/yourname/image/upload/v1608256154/uploader.nyeban/cw6qah2tyzxbp4dqpuls.png",
    "updatedAt": "2020-12-18T01:49:21.332Z",
    "createdAt": "2020-12-18T01:49:21.332Z"
  }
}
```

2. Login

Output response data access token dengan expired token 24h

The screenshot shows a Postman interface for a POST request to `http://localhost:4000/api/v1/user/login`. The request body is a form-data payload with the following fields:

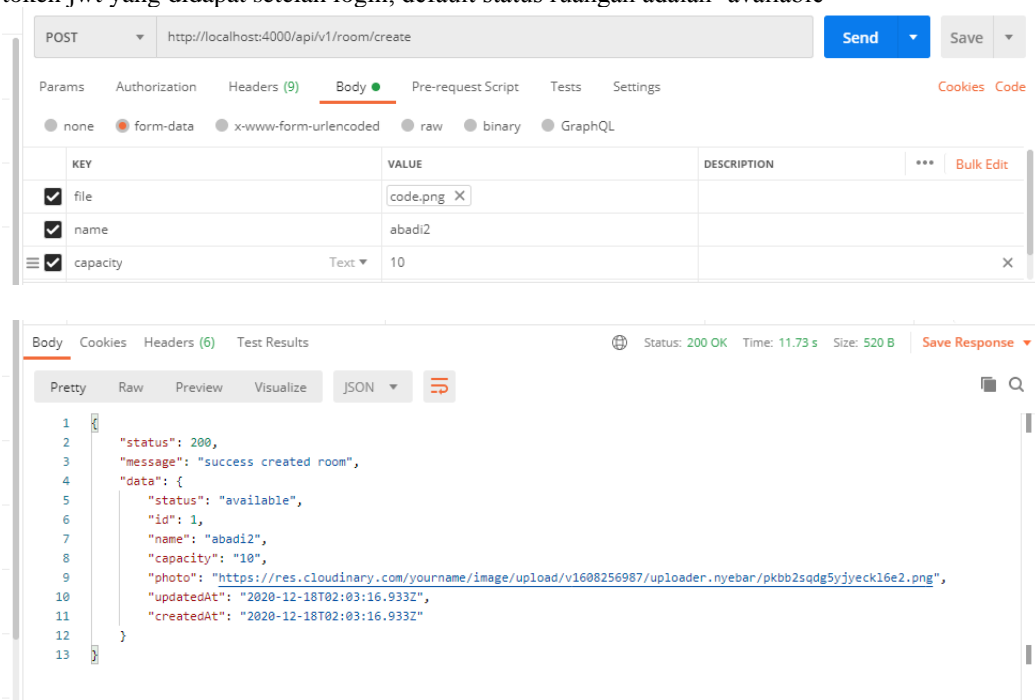
Field	Value
email	user@gmail.com
password	user

The response is a JSON object with the following structure:

```
{
  "status": 200,
  "message": "succes login",
  "data": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidm95ZSI6Imd1ZXN0IiwiaWF0IjoxNjA4MjUzMzI4LCJleHAiOjE2MDgzNDI3MjN9.Y6krZ8VoCyAay9UUsLxA-iIt9v2D1qZhuGfQqFXB1tg"
}
```

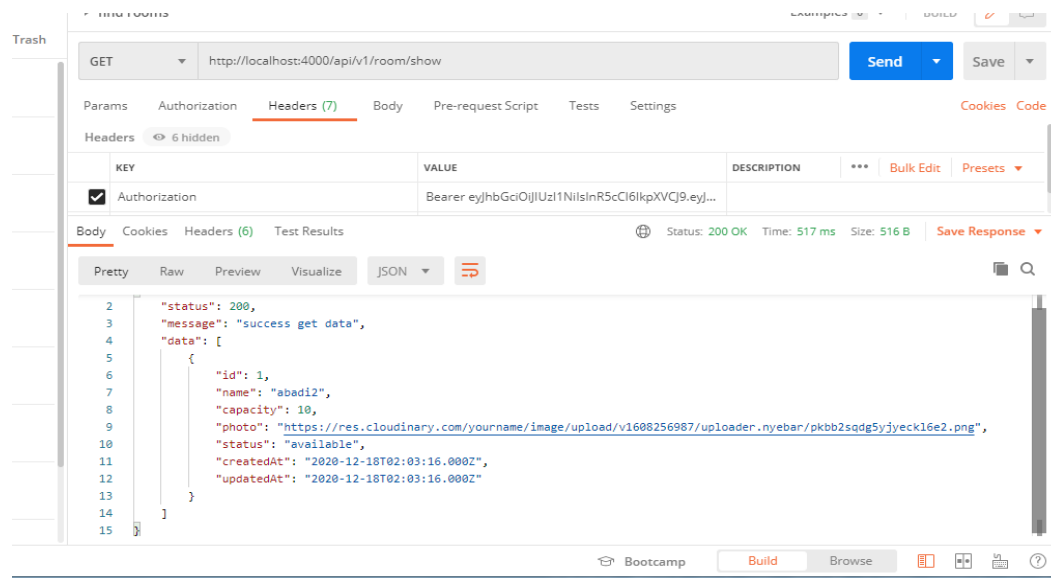
3. Create room

User dengan role 'admin' dapat membuat room, terdapat middleware autentifikasi untuk memvalidasi user dengan token jwt yang didapat setelah login, default status ruangan adalah 'available'



4. Cek ruangan yang tersedia

User 'guest' yang telah login dapat melihat ruangan yang tersedia untuk menentukan memesan ruangan



5. Memesan Ruangan

User 'guest' melakukan pemesanan ruangan dengan memasukkan data ruangan yang dibooking, waktu booking, total orang dalam ruangan, response ketika jumlah orang dalam ruangan melebihi kapasitas ruangan sebagai berikut

h

POST http://localhost:4000/api/v1/booking/create Send Save

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

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

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	room	1			
<input checked="" type="checkbox"/>	time	2020-12-18			
<input checked="" type="checkbox"/>	noted	booking1			
<input checked="" type="checkbox"/>	total	11			

Body Cookies Headers (6) Test Results Status: 401 Unauthorized Time: 440 ms Size: 265 B Save Response

Pretty Raw Preview Visualize JSON ⌵

```

1 {
2   "status": 401,
3   "error": "room over capacity"
4 }

```

Ketika user berhasil memesan ruangan, status ruang tersebut akan terupdate menjadi 'not available', response success memesan ruangan sebagai berikut :

Body Cookies Headers (6) Test Results Status: 200 OK Time: 18.74 s Size: 472 B

Pretty Raw Preview Visualize JSON ⌵

```

1 {
2   "status": 200,
3   "message": "success, please check your email",
4   "data": {
5     "id": 1,
6     "userId": 1,
7     "roomId": "1",
8     "total_person": "10",
9     "booking_time": "2020-12-18T00:00:00.000Z",
10    "noted": "booking1",
11    "updatedAt": "2020-12-18T02:22:19.912Z",
12    "createdAt": "2020-12-18T02:22:19.912Z"
13  }
14 }

```

Bootcamp Build Browse

Response mengirim email ke user yang memesan ruangan sebagai berikut :

mailtrap by followware

Inboxes > My Inbox > testing Upgrade bagus priambudi

Start with...

testing to: <user@gmail.com> 6 minutes ago

testing to: <user@gmail.com> an hour ago

testing to: <user@gmail.com> an hour ago

testing to: <user@gmail.com> 11 hours ago

testing to: <user@gmail.com> 12 hours ago

testing to: <user@gmail.com> 13 hours ago

testing

testing

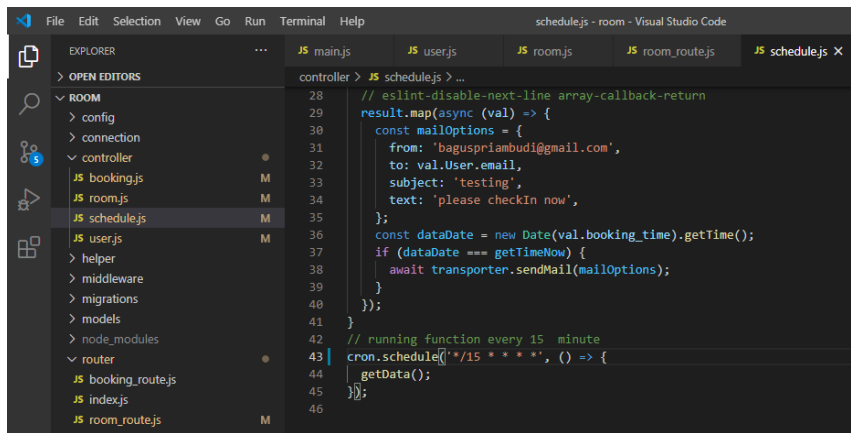
From: <baguspriambudi@gmail.com>
To: <user@gmail.com>
Show Headers

HTML HTML Source **Text** Raw Analysis Tech Info

thank you has been booking

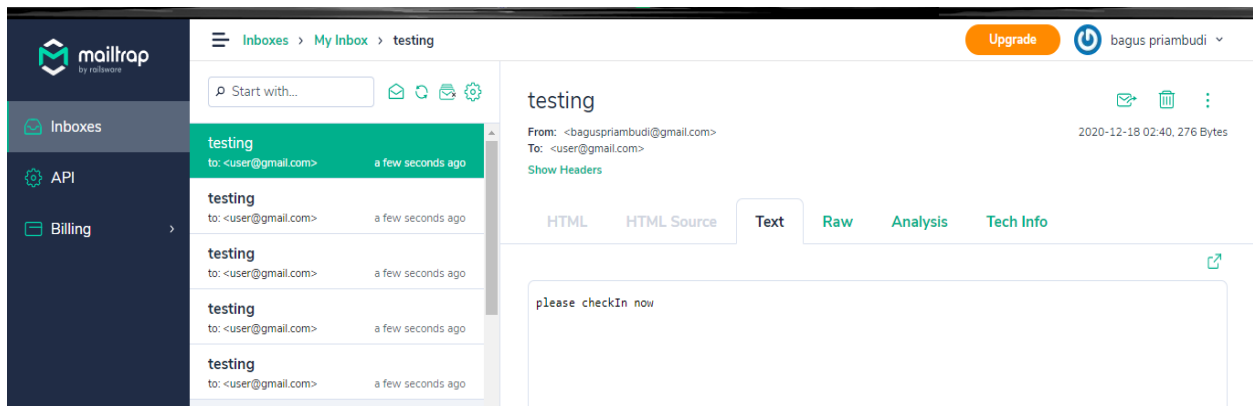
6. Ketika tanggal booking = tanggal hari ini

Saat aplikasi berjalan terdapat script untuk melakukan pengecekan secara terus menerus setiap 15 menit, didalam script tersebut melakukan pengecekan tanggal booking dan tanggal hari ini



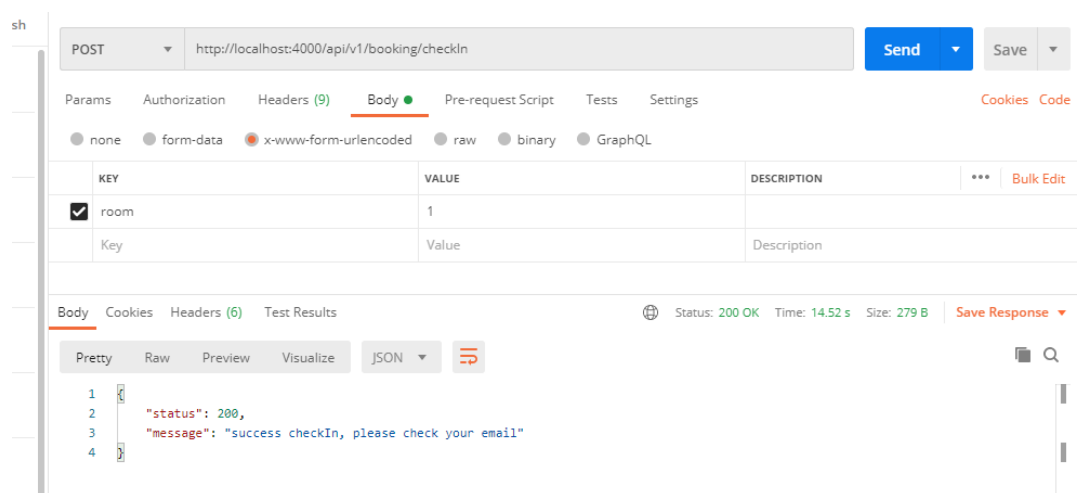
```
28 // eslint-disable-next-line array-callback-return
29 result.map(async (val) => {
30   const mailOptions = {
31     from: 'baguspriambudi@gmail.com',
32     to: val.User.email,
33     subject: 'testing',
34     text: 'please checkIn now',
35   };
36   const dataDate = new Date(val.booking_time).getTime();
37   if (dataDate === getDateNow()) {
38     await transporter.sendMail(mailOptions);
39   }
40 });
41 }
42 // running function every 15 minute
43 cron.schedule('*/*/*15 * * * *', () => {
44   getData();
45 });
46
```

Response mengirim email saat tanggal ini = tanggal booking sebagai berikut :

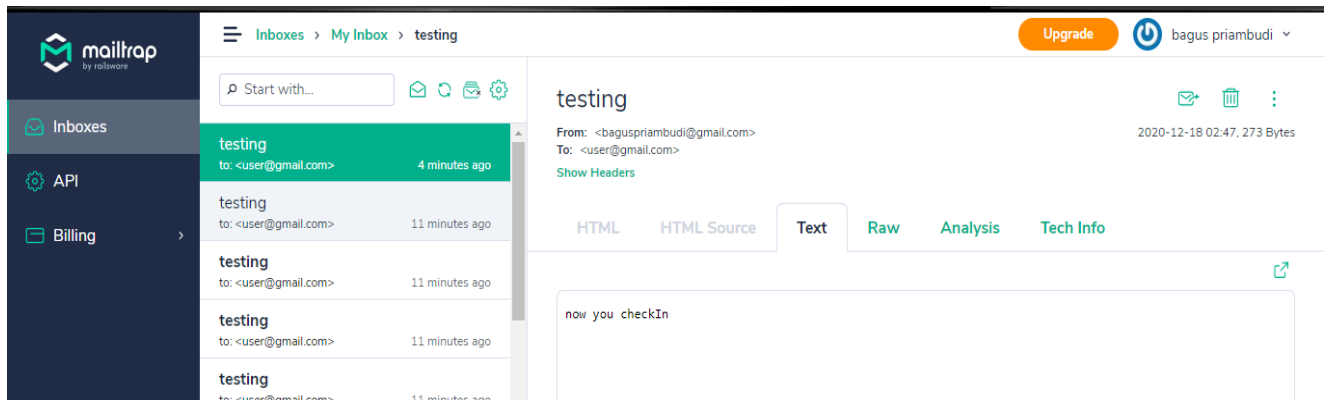


7. Check In

User ketika melakukan checkIn, harus meninputkan data ruangnya, sistem akan mengupdate filed 'check_in_time' di table booking semula null menjadi tanggal saat checkIn, berikut response success

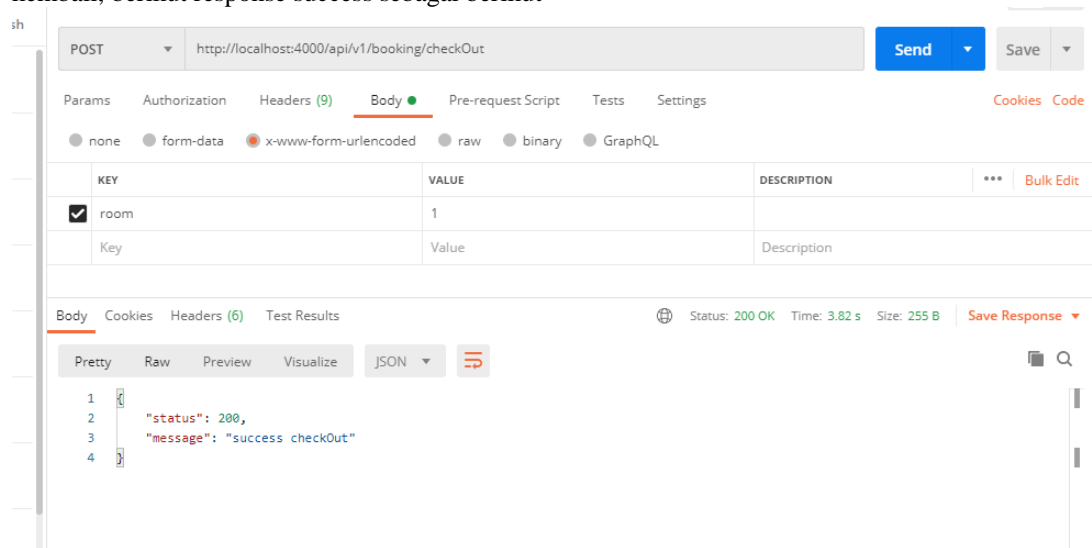


Berikut response email saat checkIn



8. Check Out

Saat user 'guest' check out ruangan sistem akan mengupdate field 'check_out_time' semula null menjadi waktu check out saat itu dan mengupdate status ruangan semula 'not available' menjadi 'available' agar dapat dipesan kembali, berikut response success sebagai berikut



9. Data Booking include User

