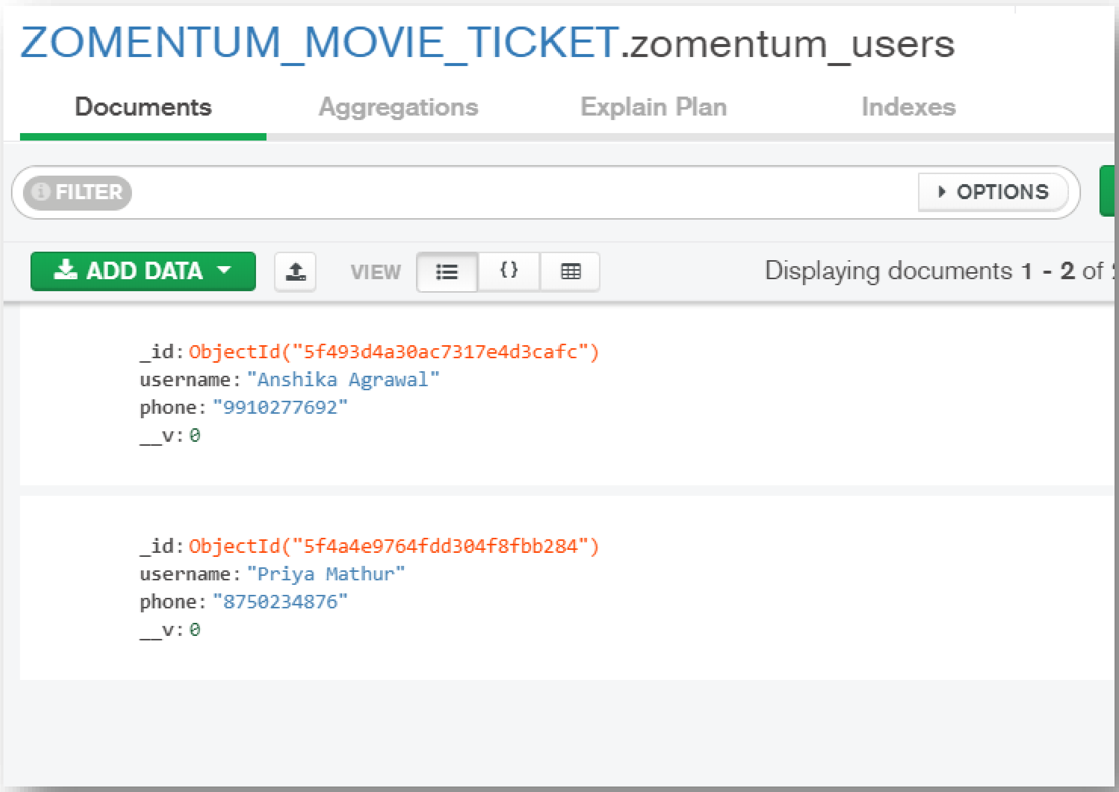
BACKEND PROBLEM STATEMENT

REST interface for a movie theatre ticket booking system in Nodejs and Mongodb

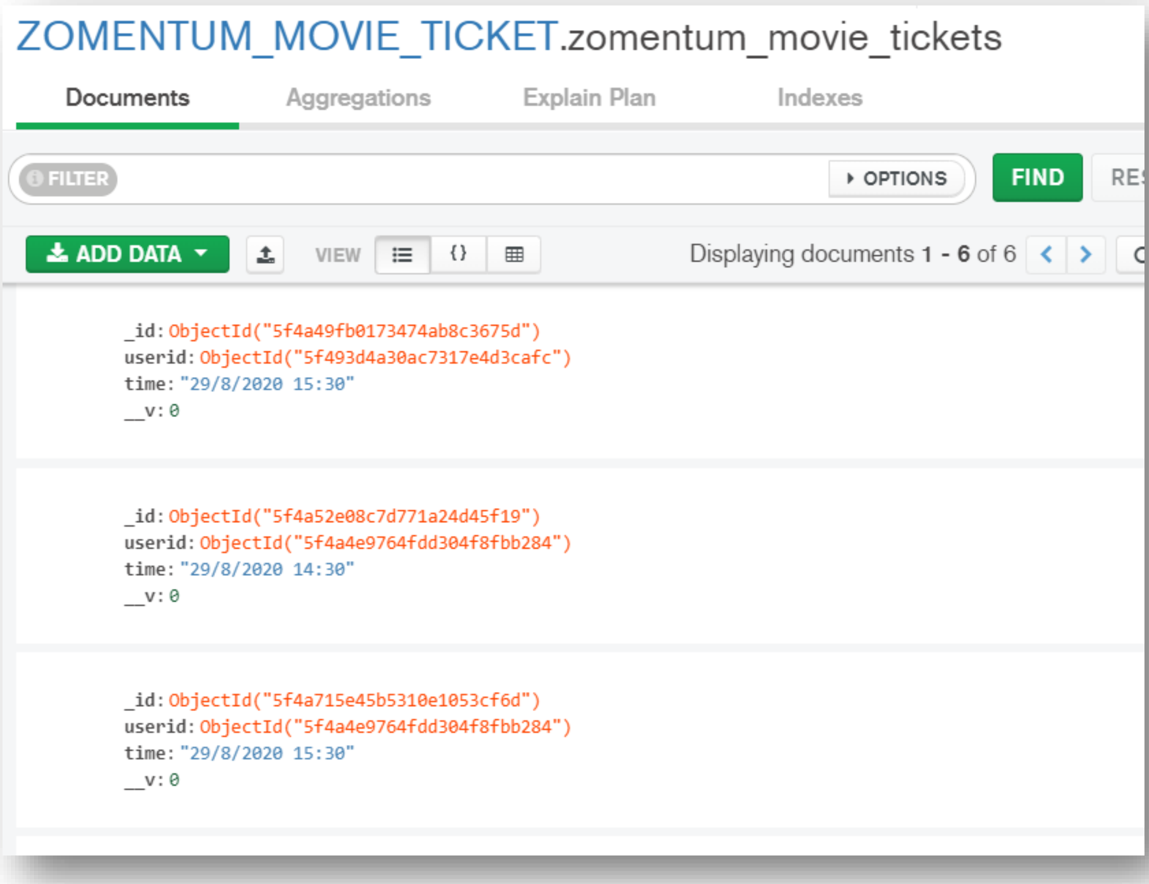
- Created 2 mongodb collections: **1) zomentum users 2) zomentum_movie_tickets**



Displaying documents 1 - 2 of

```
_id: ObjectId("5f493d4a30ac7317e4d3cafc")
username: "Anshika Agrawal"
phone: "9910277692"
__v: 0

_id: ObjectId("5f4a4e9764fdd304f8fbb284")
username: "Priya Mathur"
phone: "8750234876"
__v: 0
```



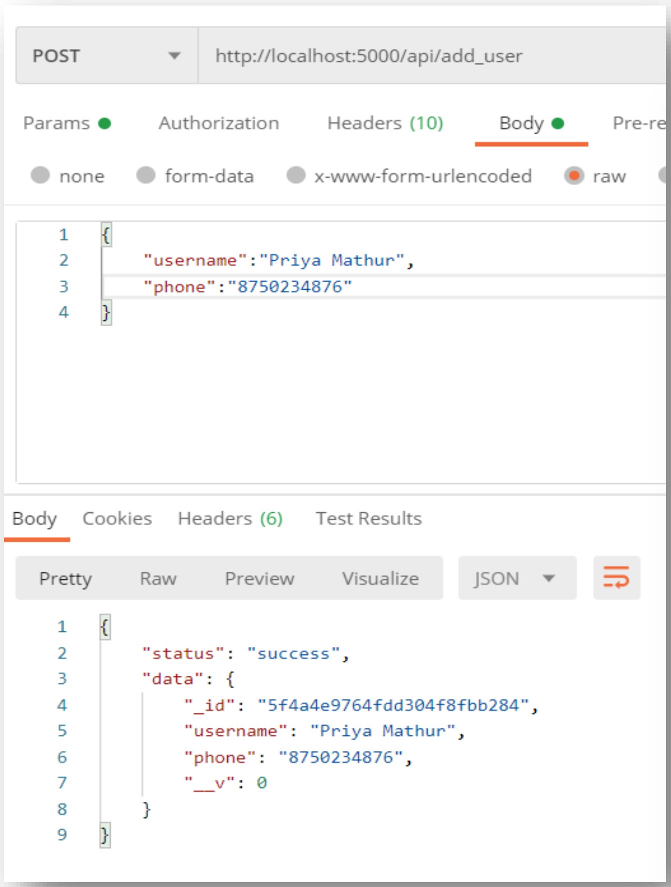
Displaying documents 1 - 6 of 6

```
_id: ObjectId("5f4a49fb0173474ab8c3675d")
userid: ObjectId("5f493d4a30ac7317e4d3cafc")
time: "29/8/2020 15:30"
__v: 0

_id: ObjectId("5f4a52e08c7d771a24d45f19")
userid: ObjectId("5f4a4e9764fdd304f8fbb284")
time: "29/8/2020 14:30"
__v: 0

_id: ObjectId("5f4a715e45b5310e1053cf6d")
userid: ObjectId("5f4a4e9764fdd304f8fbb284")
time: "29/8/2020 15:30"
__v: 0
```

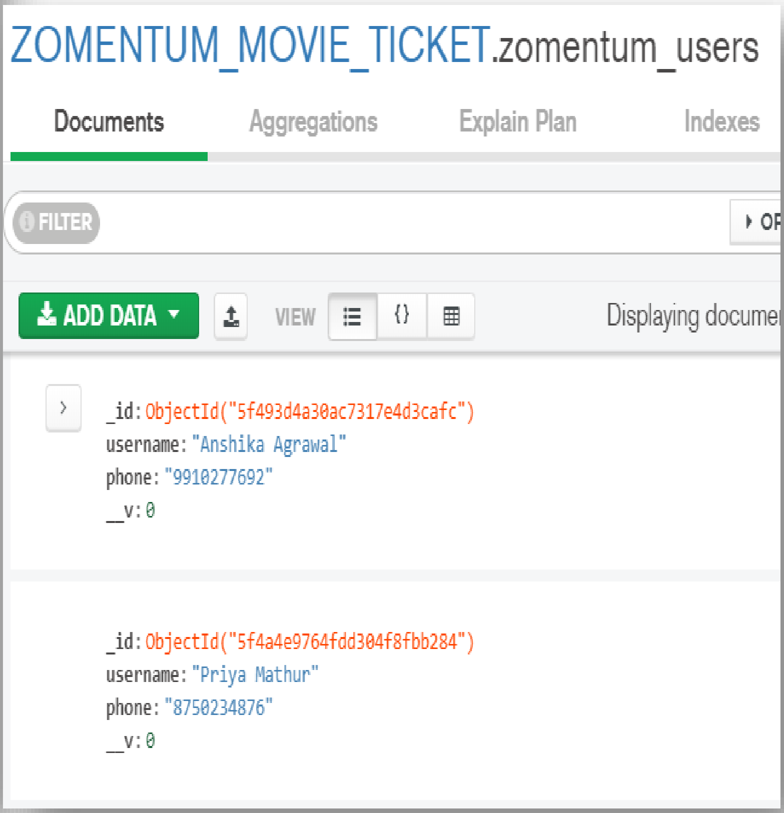
- **An endpoint to add a user**



POST http://localhost:5000/api/add_user

```
{
    "username":"Priya Mathur",
    "phone":"8750234876"
}
```

```
{
    "status": "success",
    "data": {
        "_id": "5f4a4e9764fdd304f8fbb284",
        "username": "Priya Mathur",
        "phone": "8750234876",
        "__v": 0
    }
}
```

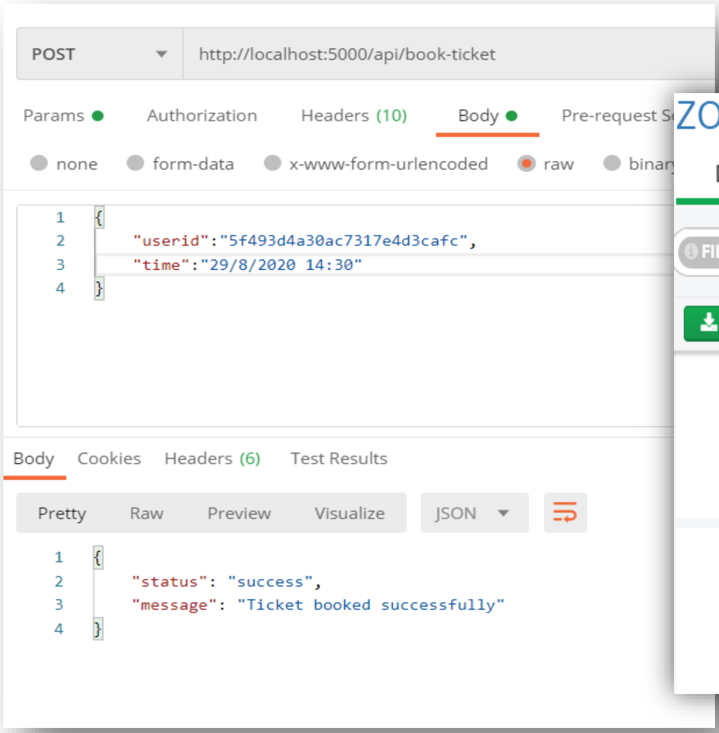


ZOMENTUM_MOVIE_TICKET.zomentum_users

```
_id: ObjectId("5f493d4a30ac7317e4d3cafc")
username: "Anshika Agrawal"
phone: "9910277692"
__v: 0
```

```
_id: ObjectId("5f4a4e9764fdd304f8fbb284")
username: "Priya Mathur"
phone: "8750234876"
__v: 0
```

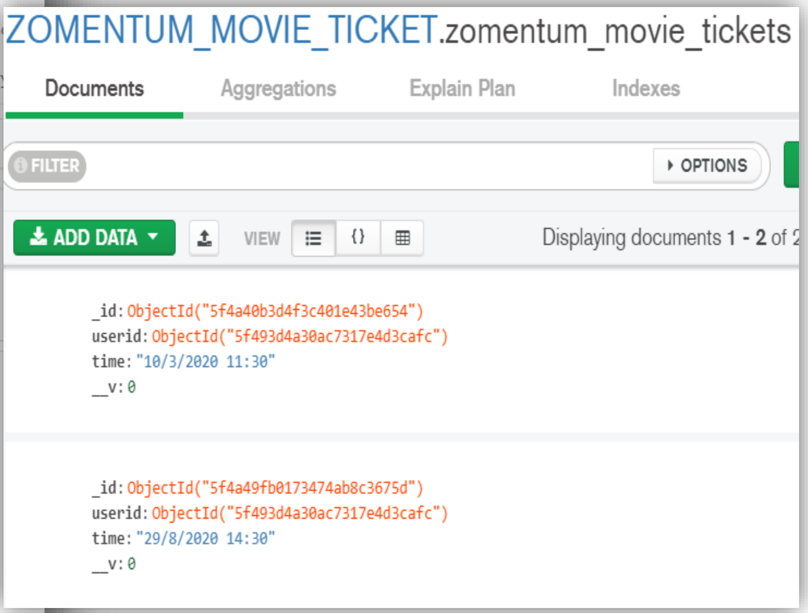
- **An endpoint to book a ticket using a user's id, phone number, and timings**



POST http://localhost:5000/api/book-ticket

```
{
    "userid":"5f493d4a30ac7317e4d3cafc",
    "time":"29/8/2020 14:30"
}
```

```
{
    "status": "success",
    "message": "Ticket booked successfully"
}
```

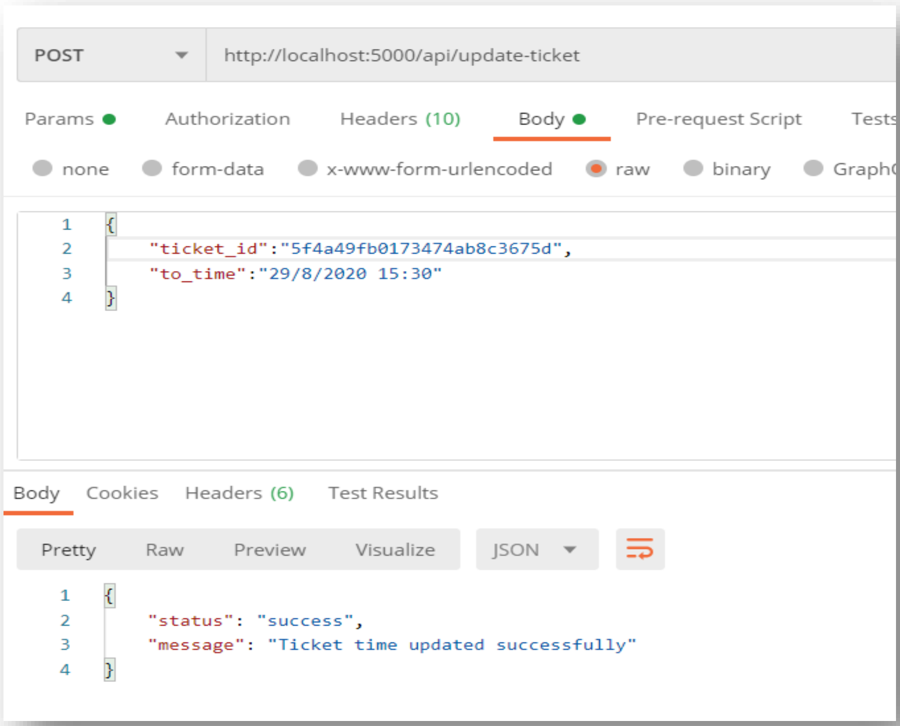


ZOMENTUM_MOVIE_TICKET.zomentum_movie_tickets

```
_id: ObjectId("5f4a40b3d4f3c401e43be654")
userid: ObjectId("5f493d4a30ac7317e4d3cafc")
time: "10/3/2020 11:30"
__v: 0
```

```
_id: ObjectId("5f4a49fb0173474ab8c3675d")
userid: ObjectId("5f493d4a30ac7317e4d3cafc")
time: "29/8/2020 14:30"
__v: 0
```

- **An endpoint to update a ticket timing**



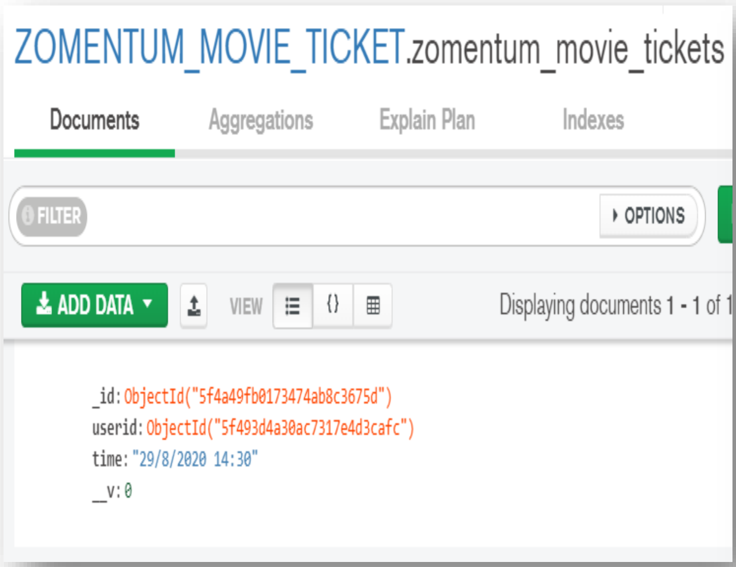
http://localhost:5000/api/update-ticket

```
{
    "ticket_id":"5f4a49fb0173474ab8c3675d",
    "to_time":"29/8/2020 15:30"
}
```

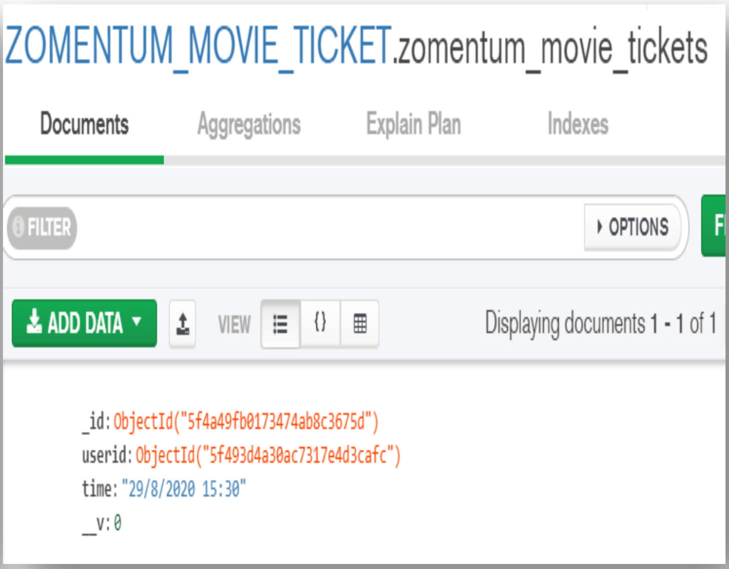
```
{
    "status": "success",
    "message": "Ticket time updated successfully"
}
```

Before update:

After Update

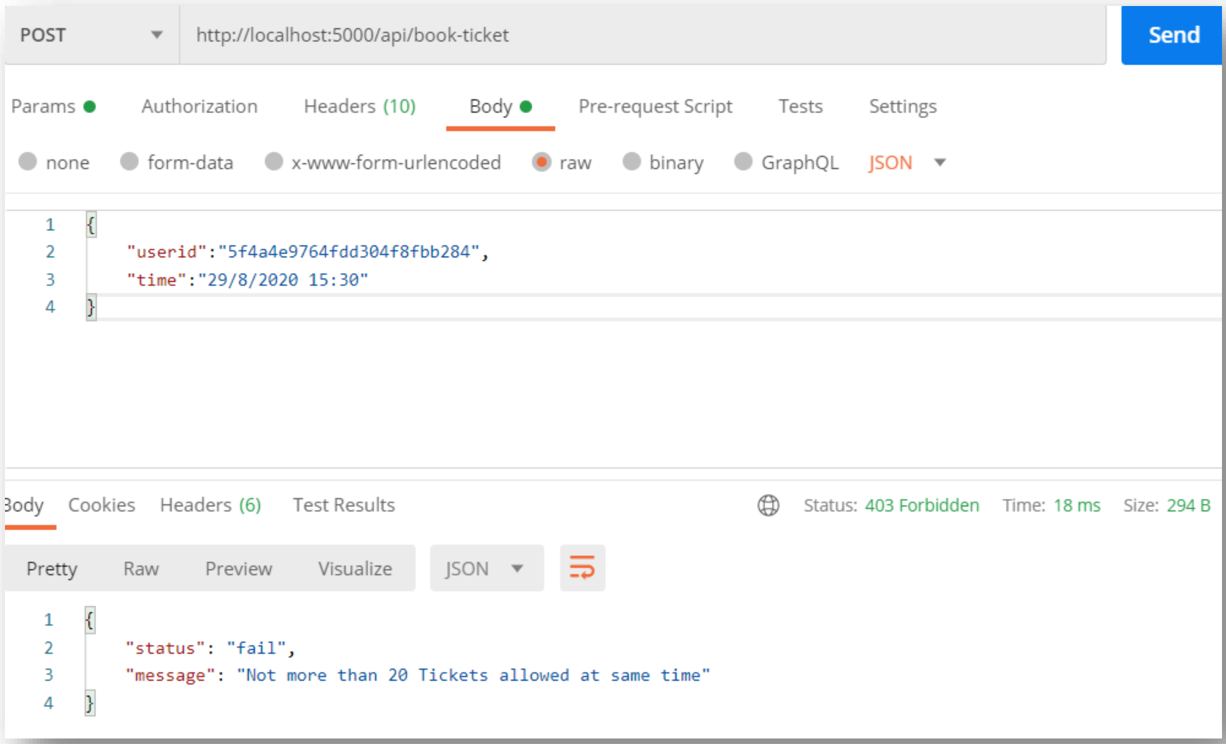

ZOMENTUM_MOVIE_TICKET.zomentum_movie_tickets
Documents
Aggregations
Explain Plan
Indexes
FILTER
OPTIONS
ADD DATA
VIEW
Displaying documents 1 - 1 of 1

```
_id: ObjectId("5f4a49fb0173474ab8c3675d")
userid: ObjectId("5f493d4a30ac7317e4d3cafc")
time: "29/8/2020 14:30"
__v: 0
```


ZOMENTUM_MOVIE_TICKET.zomentum_movie_tickets
Documents
Aggregations
Explain Plan
Indexes
FILTER
OPTIONS
ADD DATA
VIEW
Displaying documents 1 - 1 of 1

```
_id: ObjectId("5f4a49fb0173474ab8c3675d")
userid: ObjectId("5f493d4a30ac7317e4d3cafc")
time: "29/8/2020 15:30"
__v: 0
```

- For a particular timing, a maximum of 20 tickets can be booked.
  1) **Cannot book a ticket when already 20 tickets are booked**



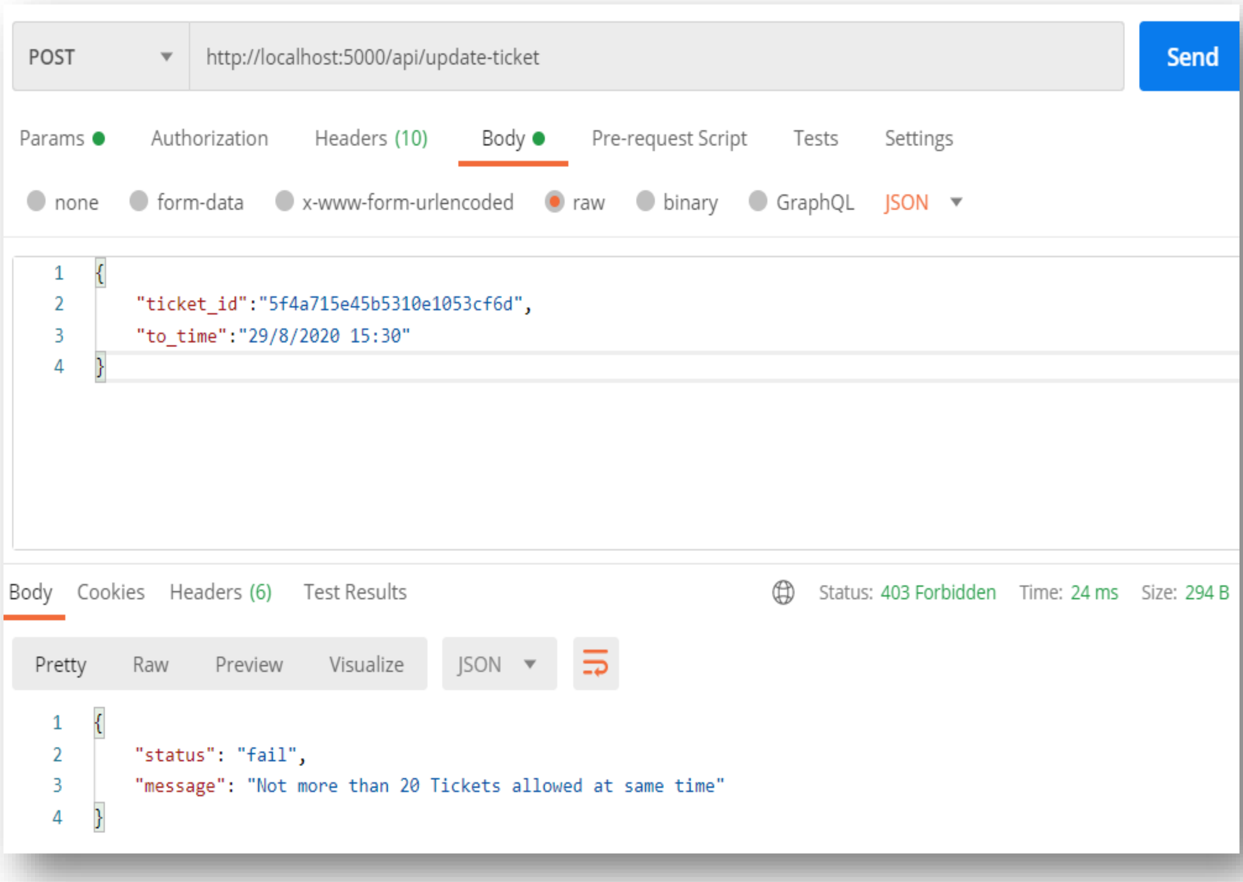
POST http://localhost:5000/api/book-ticket

```
{
    "userid":"5f4a4e9764fdd304f8fbb284",
    "time":"29/8/2020 15:30"
}
```

Status: 403 Forbidden Time: 18 ms Size: 294 B

```
{
    "status": "fail",
    "message": "Not more than 20 Tickets allowed at same time"
}
```

  2) **Cannot update ticket's timing to the new time at which already 20 tickets are booked**



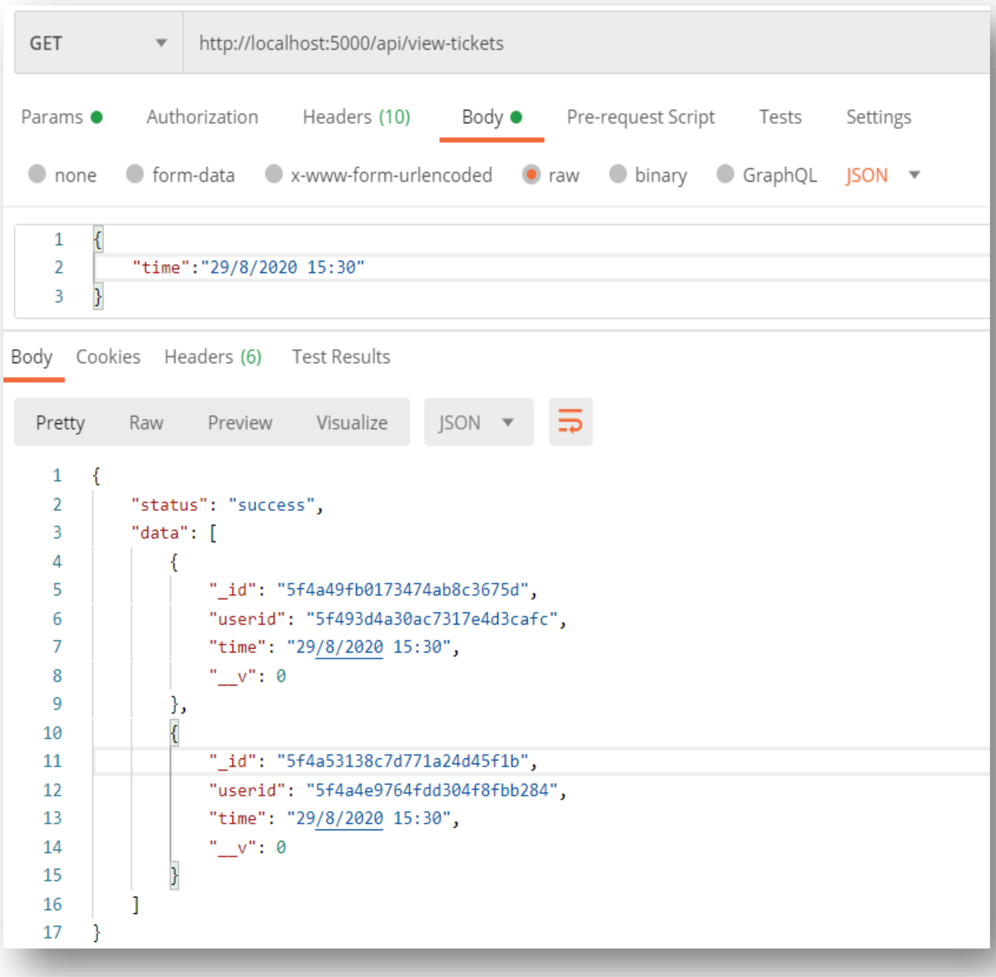
POST http://localhost:5000/api/update-ticket

```
{
    "ticket_id":"5f4a715e45b5310e1053cf6d",
    "to_time":"29/8/2020 15:30"
}
```

Status: 403 Forbidden Time: 24 ms Size: 294 B

```
{
    "status": "fail",
    "message": "Not more than 20 Tickets allowed at same time"
}
```

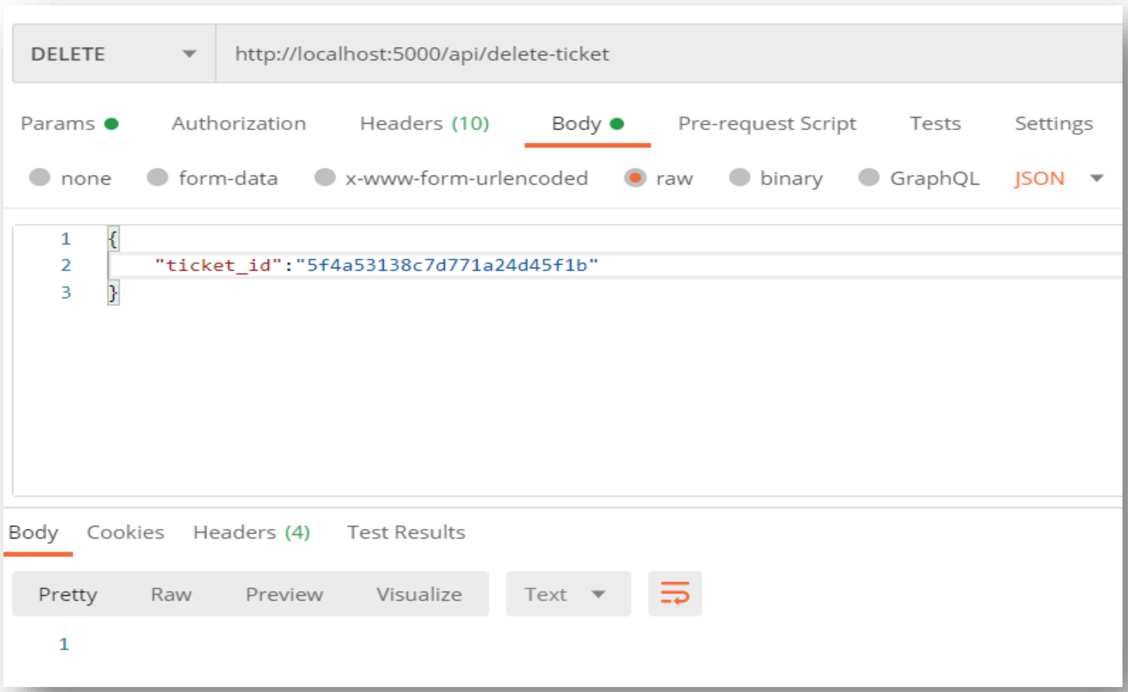
- **An endpoint to view all the tickets for a particular time**



```
{
    "time":"29/8/2020 15:30"
}
```

```
{
    "status": "success",
    "data": [
        {
            "_id": "5f4a49fb0173474ab8c3675d",
            "userid": "5f493d4a30ac7317e4d3cafc",
            "time": "29/8/2020 15:30",
            "__v": 0
        },
        {
            "_id": "5f4a53138c7d771a24d45f1b",
            "userid": "5f4a4e9764fdd304f8fbb284",
            "time": "29/8/2020 15:30",
            "__v": 0
        }
    ]
}
```

- **An endpoint to delete a particular ticket.**

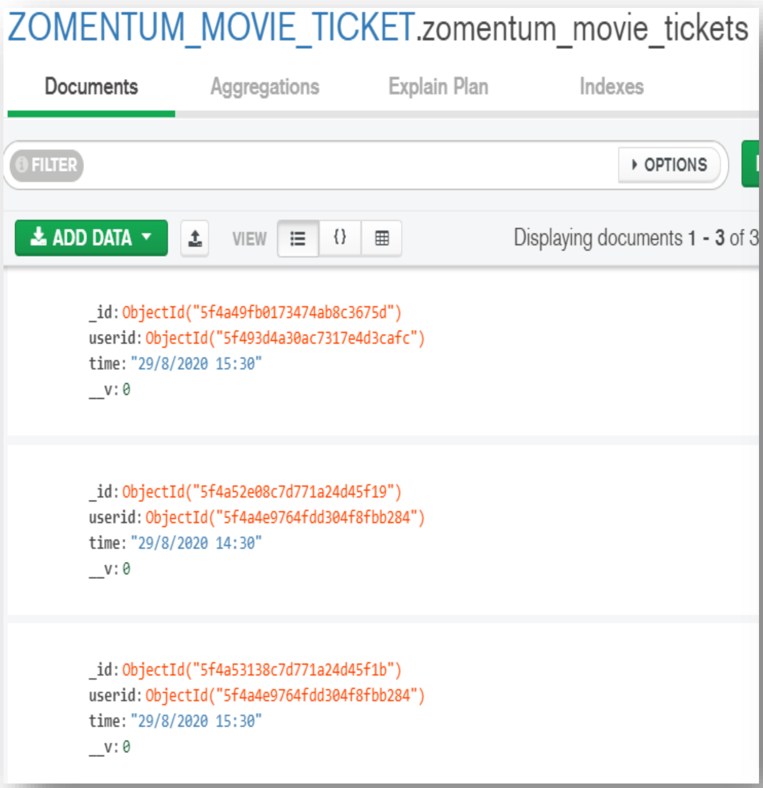


http://localhost:5000/api/delete-ticket

```
{
    "ticket_id":"5f4a53138c7d771a24d45f1b"
}
```

Before Delete:

After Delete:

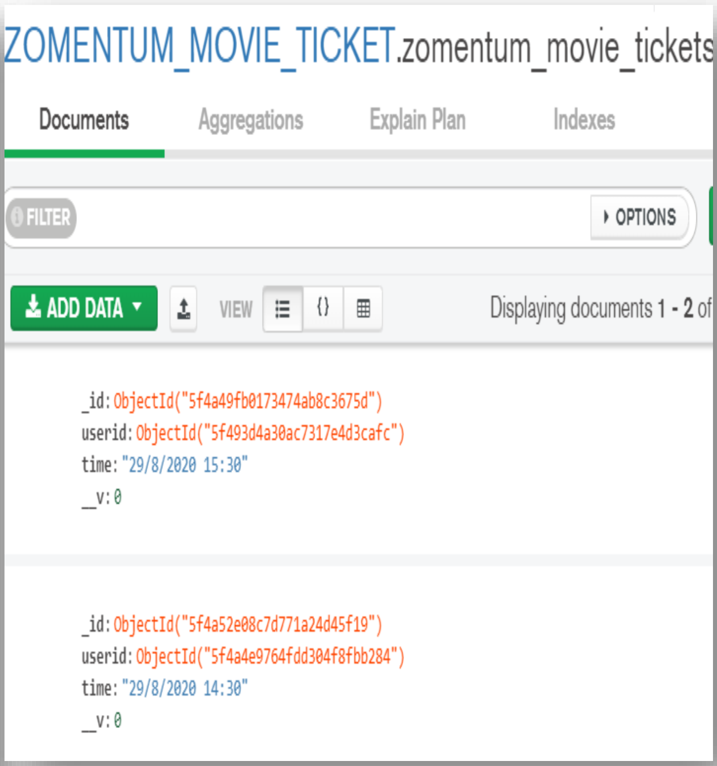


Displaying documents 1 - 3 of 3

```
_id: ObjectId("5f4a49fb0173474ab8c3675d")
userid: ObjectId("5f493d4a30ac7317e4d3cafc")
time: "29/8/2020 15:30"
__v: 0

_id: ObjectId("5f4a52e08c7d771a24d45f19")
userid: ObjectId("5f4a4e9764fdd304f8fbb284")
time: "29/8/2020 14:30"
__v: 0

_id: ObjectId("5f4a53138c7d771a24d45f1b")
userid: ObjectId("5f4a4e9764fdd304f8fbb284")
time: "29/8/2020 15:30"
__v: 0
```

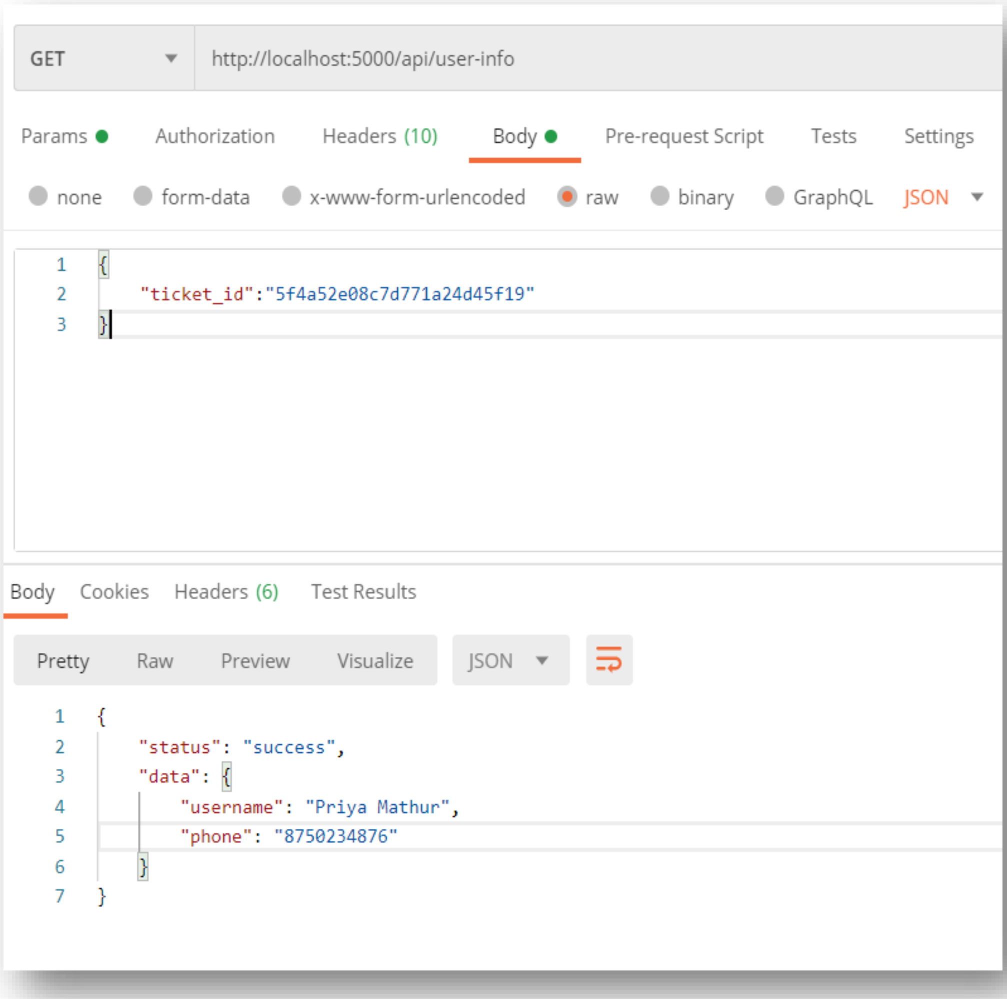


Displaying documents 1 - 2 of

```
_id: ObjectId("5f4a49fb0173474ab8c3675d")
userid: ObjectId("5f493d4a30ac7317e4d3cafc")
time: "29/8/2020 15:30"
__v: 0

_id: ObjectId("5f4a52e08c7d771a24d45f19")
userid: ObjectId("5f4a4e9764fdd304f8fbb284")
time: "29/8/2020 14:30"
__v: 0
```

- **An endpoint to view the user's details based on the ticket id.**



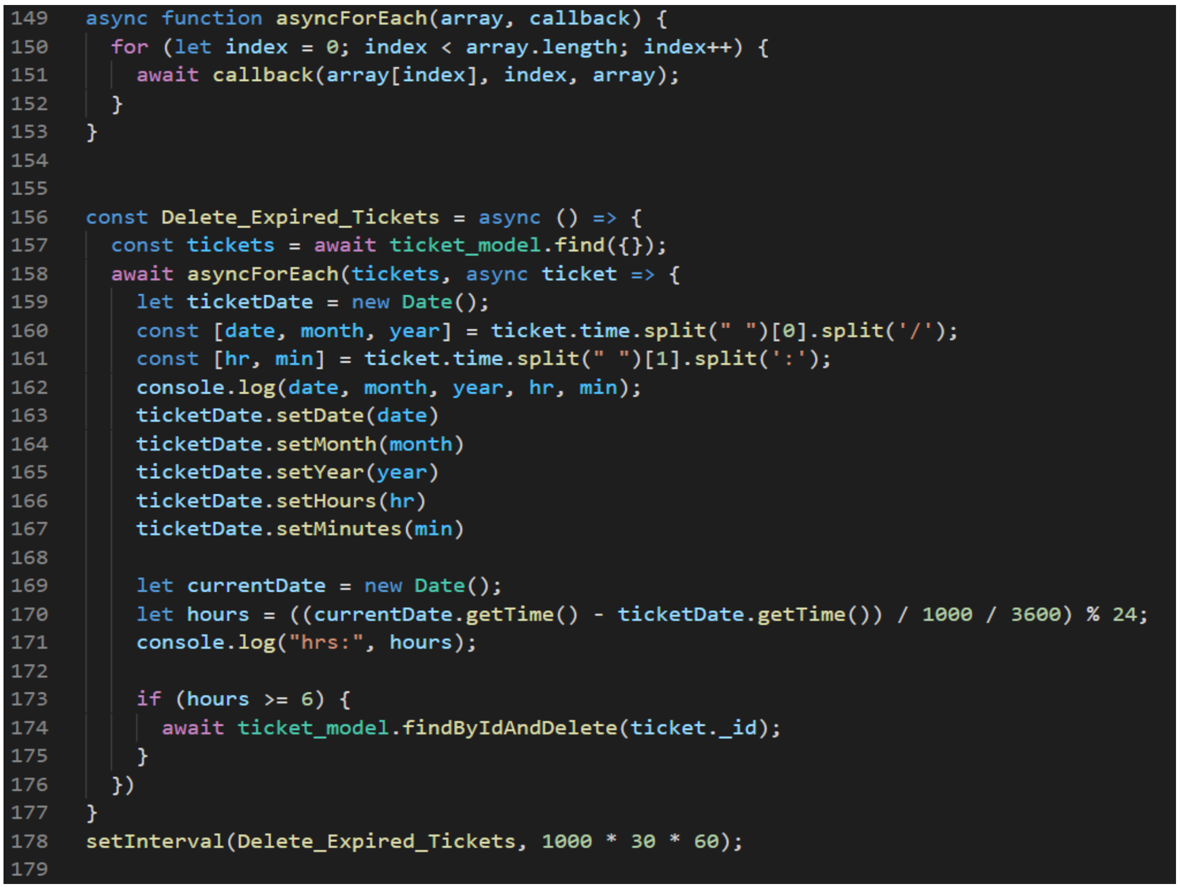
GET http://localhost:5000/api/user-info

```
{
    "ticket_id":"5f4a52e08c7d771a24d45f19"
}
```

Body (JSON):

```
{
    "status": "success",
    "data": {
        "username": "Priya Mathur",
        "phone": "8750234876"
    }
}
```

- **Delete all the tickets which are expired automatically (The following code deletes expired every 6hrs)**



```javascript
async function asyncForEach(array, callback) {
  for (let index = 0; index < array.length; index++) {
    await callback(array[index], index, array);
  }
}


const Delete_Expired_Tickets = async () => {
  const tickets = await ticket_model.find({});
  await asyncForEach(tickets, async ticket => {
    let ticketDate = new Date();
    const [date, month, year] = ticket.time.split(" ")[0].split('/');
    const [hr, min] = ticket.time.split(" ")[1].split(':');
    console.log(date, month, year, hr, min);
    ticketDate.setDate(date)
    ticketDate.setMonth(month)
    ticketDate.setYear(year)
    ticketDate.setHours(hr)
    ticketDate.setMinutes(min)

    let currentDate = new Date();
    let hours = ((currentDate.getTime() - ticketDate.getTime()) / 1000 / 3600) % 24;
    console.log("hrs:", hours);

    if (hours >= 6) {
      await ticket_model.findByIdAndDelete(ticket._id);
    }
  })
}
setInterval(Delete_Expired_Tickets, 1000 * 30 * 60);
```