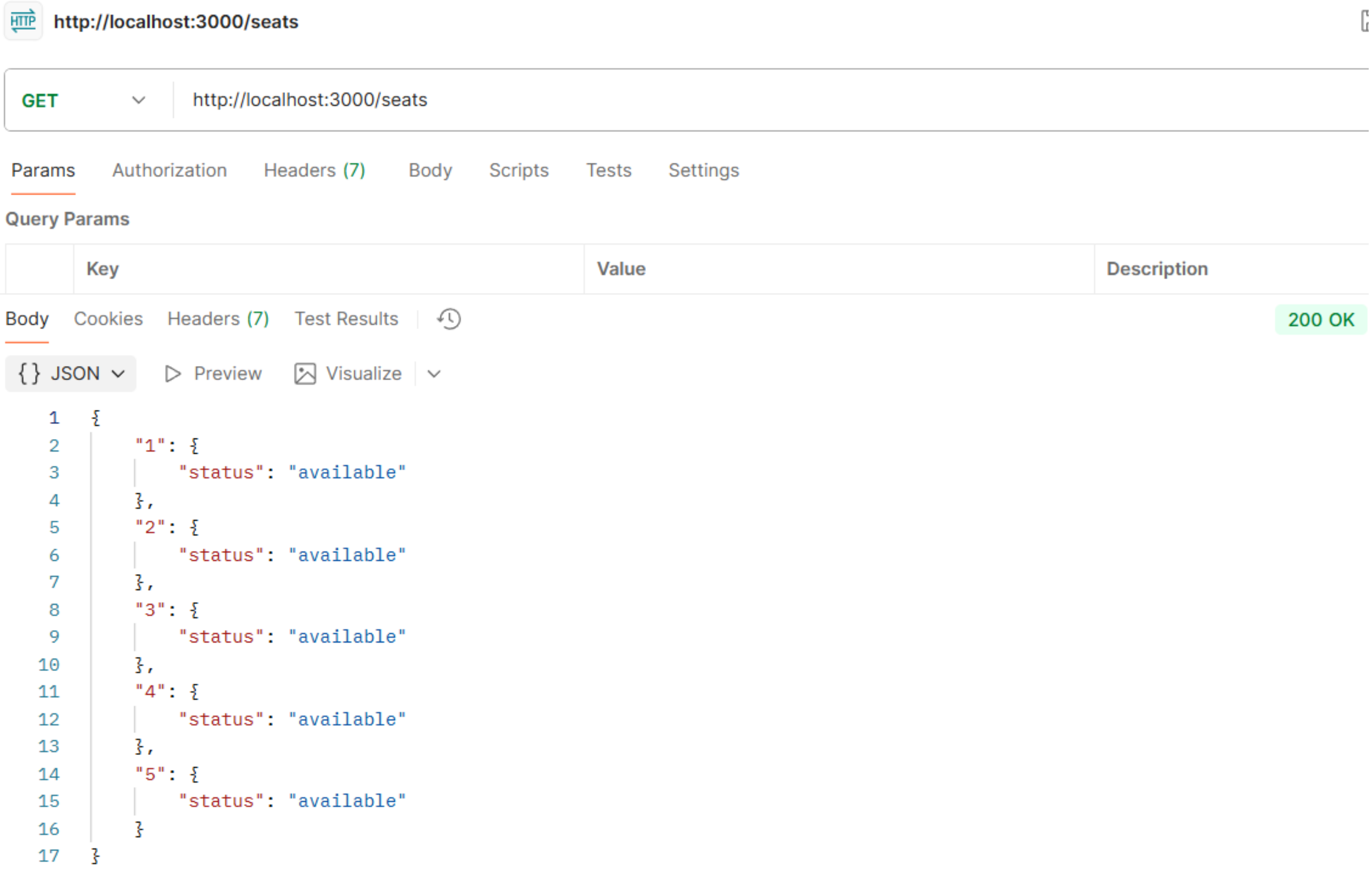
NAME	J k Kunal
UID	23BCS11041
CLASS	622-A

- NodeJs PRACTISE 3
- CODE

```
const express = require('express');
const app = express();
const port = 3000;

app.use(express.json());

const seats = {
    '1': { status: 'available', lockedUntil: null },
    '2': { status: 'available', lockedUntil: null },
    '3': { status: 'available', lockedUntil: null },
    '4': { status: 'available', lockedUntil: null },
    '5': { status: 'available', lockedUntil: null }
};

const LOCK_DURATION_MS = 60 * 1000;

function checkAndClearLock(seatId) {
    const seat = seats[seatId];
    if (seat && seat.status === 'locked' && seat.lockedUntil <=
Date.now()) {
        seat.status = 'available';
        seat.lockedUntil = null;
        console.log(`Lock for seat ${seatId} expired and cleared.`);
    }
```

```
}

app.get('/seats', (req, res) => {
    const currentSeats = {};
    Object.keys(seats).forEach(seatId => {
        checkAndClearLock(seatId);
        const { status } = seats[seatId];
        currentSeats[seatId] = { status };
    });
    res.status(200).json(currentSeats);
});

app.post('/lock/:seatId', (req, res) => {
    const seatId = req.params.seatId;
    const seat = seats[seatId];

    if (!seat) {
        return res.status(404).json({ message: "Seat not found" });
    }

    checkAndClearLock(seatId);

    if (seat.status === 'available') {
        seat.status = 'locked';
        seat.lockedUntil = Date.now() + LOCK_DURATION_MS;

        return res.status(200).json({
            message: `Seat ${seatId} locked successfully. Confirm within
${LOCK_DURATION_MS / 1000} seconds.`,
            lockedUntil: new Date(seat.lockedUntil).toISOString()
        });
    } else if (seat.status === 'locked') {
        return res.status(409).json({ message: `Seat ${seatId} is
currently locked. Try again later.` });
    } else if (seat.status === 'booked') {
        return res.status(409).json({ message: `Seat ${seatId} is already
booked.` });
    }
});

app.post('/confirm/:seatId', (req, res) => {
    const seatId = req.params.seatId;
    const seat = seats[seatId];

    if (!seat) {
        return res.status(404).json({ message: "Seat not found" });
    }

    checkAndClearLock(seatId);

    if (seat.status === 'locked' && seat.lockedUntil > Date.now()) {
        seat.status = 'booked';
        seat.lockedUntil = null;

        return res.status(200).json({ message: `Seat ${seatId} booked
successfully!` });
    } else {
        return res.status(400).json({ message: "Seat is not locked and
cannot be booked" });
    }
});

app.listen(port, () => {
    console.log(`Booking system listening at http://localhost:${port}`);
    console.log(`Initial seats: ${JSON.stringify(Object.keys(seats))}`);
});
```

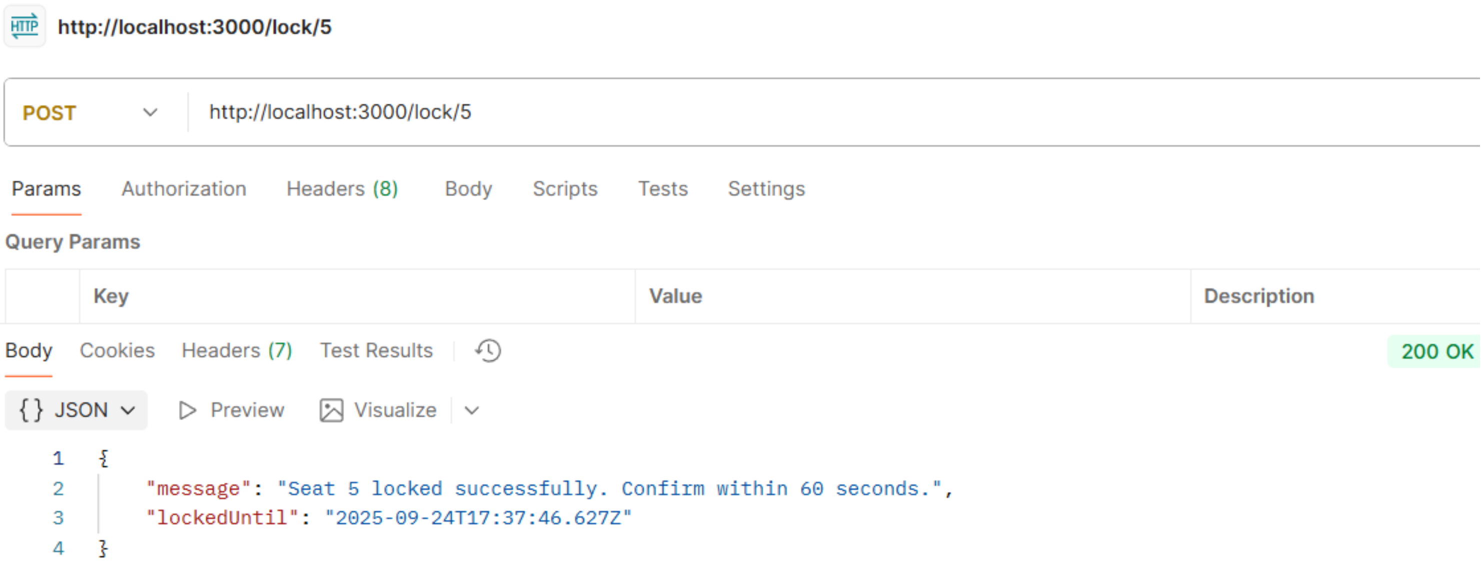
- OUTPUT



GET http://localhost:3000/seats

200 OK

```
{
    "1": {
        "status": "available"
    },
    "2": {
        "status": "available"
    },
    "3": {
        "status": "available"
    },
    "4": {
        "status": "available"
    },
    "5": {
        "status": "available"
    }
}
```

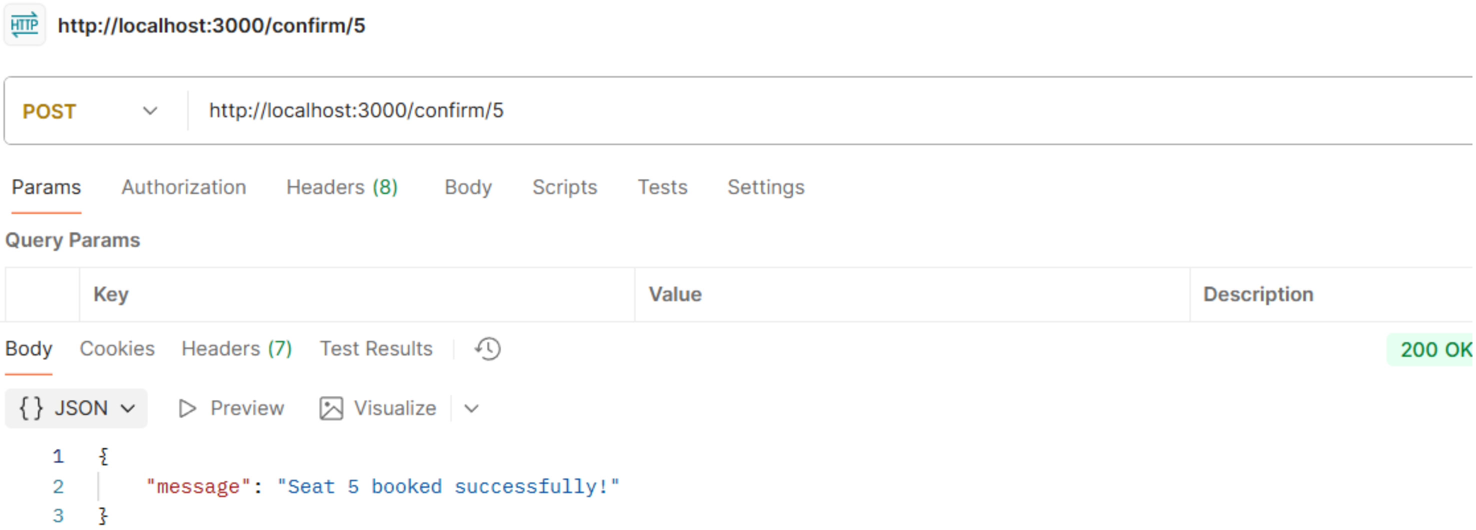


http://localhost:3000/lock/5

POST http://localhost:3000/lock/5

200 OK

```
{
    "message": "Seat 5 locked successfully. Confirm within 60 seconds.",
    "lockedUntil": "2025-09-24T17:37:46.627Z"
}
```

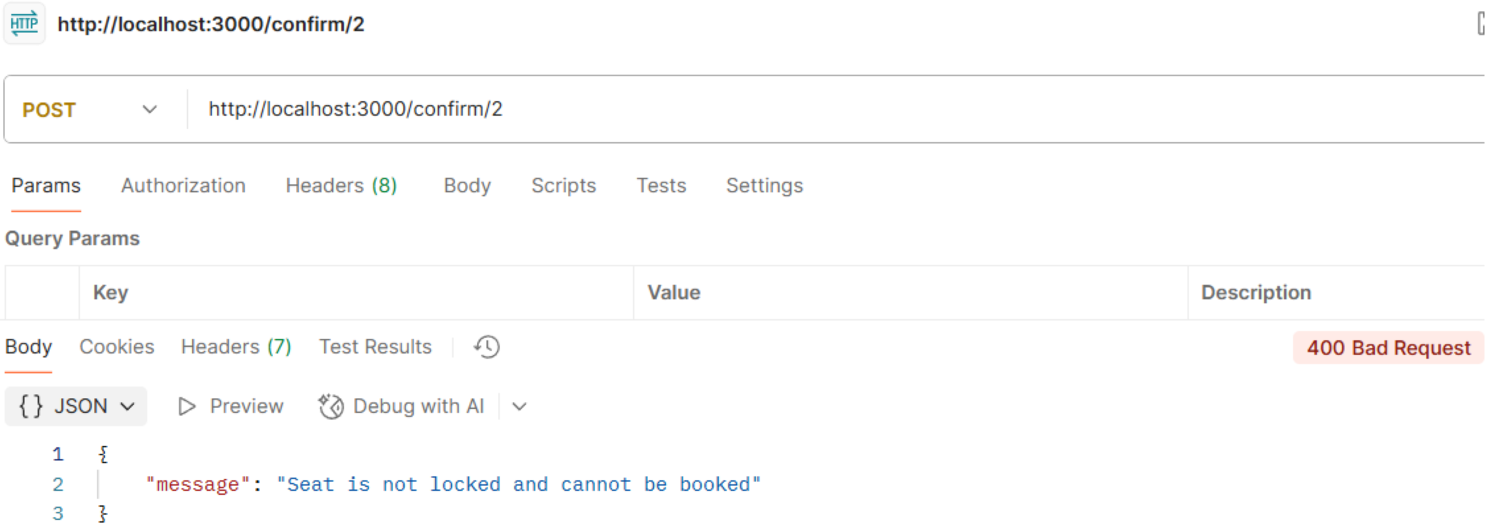


http://localhost:3000/confirm/5

POST http://localhost:3000/confirm/5

200 OK

```
{
    "message": "Seat 5 booked successfully!"
}
```



http://localhost:3000/confirm/2

POST http://localhost:3000/confirm/2

400 Bad Request

```
{
    "message": "Seat is not locked and cannot be booked"
}
```

Output Link : http://localhost:3000/seats

API's: GET http://localhost:3000/seats POST http://localhost:3000/lock/5

POST http://localhost:3000/confirm/5 POST http://localhost:3000/confirm/2