

LUMA Technical Interview

Prepared By: Pranjali Patel patelpranzp25@gmail.com

- Stored the static data for two doctors with fields: name and hours
In hours, two time slots have defined, where true represent availability and false represent hours are booked

```
'10AM to 12PM': true,  
'2PM to 5PM': false
```

- Every appointment is stored as one entry in object named 'appointments'

1) Find a doctor's working hours:

Get request based on appropriate doctor, doctor name is passed in request body
call: GET: <http://localhost:3001/doctors/getAvailableHours>

The image shows a terminal window and the Postman application. The terminal window, titled "REST_API_Node_Project — node index.js — 94x32", displays the output of running `node index.js`. It shows the server listening on port 3001 and the method to get doctor's hours for "Dr Thanos". The response is a JSON object with working hours for Monday, Tuesday, and Friday.

```
(base) Pranjals-MBP:REST_API_Node_Project pranzsmac$ node index.js
Listening on port #3001
Method to get doctor's hours
name of doctor : Dr Thanos
{ Monday: '10AM to 12PM',
  Tuesday: '10AM to 12PM , 2PM to 5PM',
  Friday: '10AM to 12PM , 2PM to 5PM' }
```

The Postman application shows a GET request to `http://localhost:3001/doctors/getAvailableHours`. The request body is a JSON object with the key "doctor" and the value "Dr Thanos". The request is configured with the "Body" tab selected, and the content type is set to "JSON (application/json)".

Usage alert
Your team is about to hit the limit on shared usage for free plans. Your oldest history will be archived once you reach the limit. [Learn more.](#)
[Check usage](#) [Upgrade](#)

response:

The image shows the Postman application window. At the top, there's a toolbar with buttons for 'New', 'Import', 'Runner', 'Linkedin', 'Invite', and 'Upgrade'. Below this, the request details are shown: a GET request to 'http://localhost:3001/doctors/getAvailableHours' in the 'No Environment' context. The 'Headers' tab is selected, showing two headers: 'doct' with value 'Dr Thanos' and 'Content-Type' with value 'application/json'. The 'Body' tab is also visible, showing a JSON response. The status bar at the bottom indicates a successful '200 OK' response with a time of '51 ms' and a size of '412 B'. The response body is displayed in the 'Body' tab, showing a JSON object with available hours for Monday, Tuesday, and Friday.

GET http://localhost:3001/doctors/ge + ... No Environment

http://localhost:3001/doctors/getAvailableHours

GET http://localhost:3001/doctors/getAvailableHours Send Save

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input type="checkbox"/> doct	Dr Thanos				
<input checked="" type="checkbox"/> Content-Type	application/json				
Key	Value	Description			

Body Cookies Headers (9) Test Results Status: 200 OK Time: 51 ms Size: 412 B Download

Pretty Raw Preview HTML

```
i 1 [{"Monday": "10AM to 12PM", "Tuesday": "10AM to 12PM , 2PM to 5PM", "Friday": "10AM to 12PM , 2PM to 5PM"}]
```

Learn Build Browse

2) Book a doctor opening

The doctor and patient name, day and timing are passed in request body

Call: POST: <http://localhost:3001/appointment/book>

The screenshot displays a Node.js terminal window and the Postman API client. The terminal window, titled 'REST_API_Node_Project — node index.js — 94x32', shows the following output:

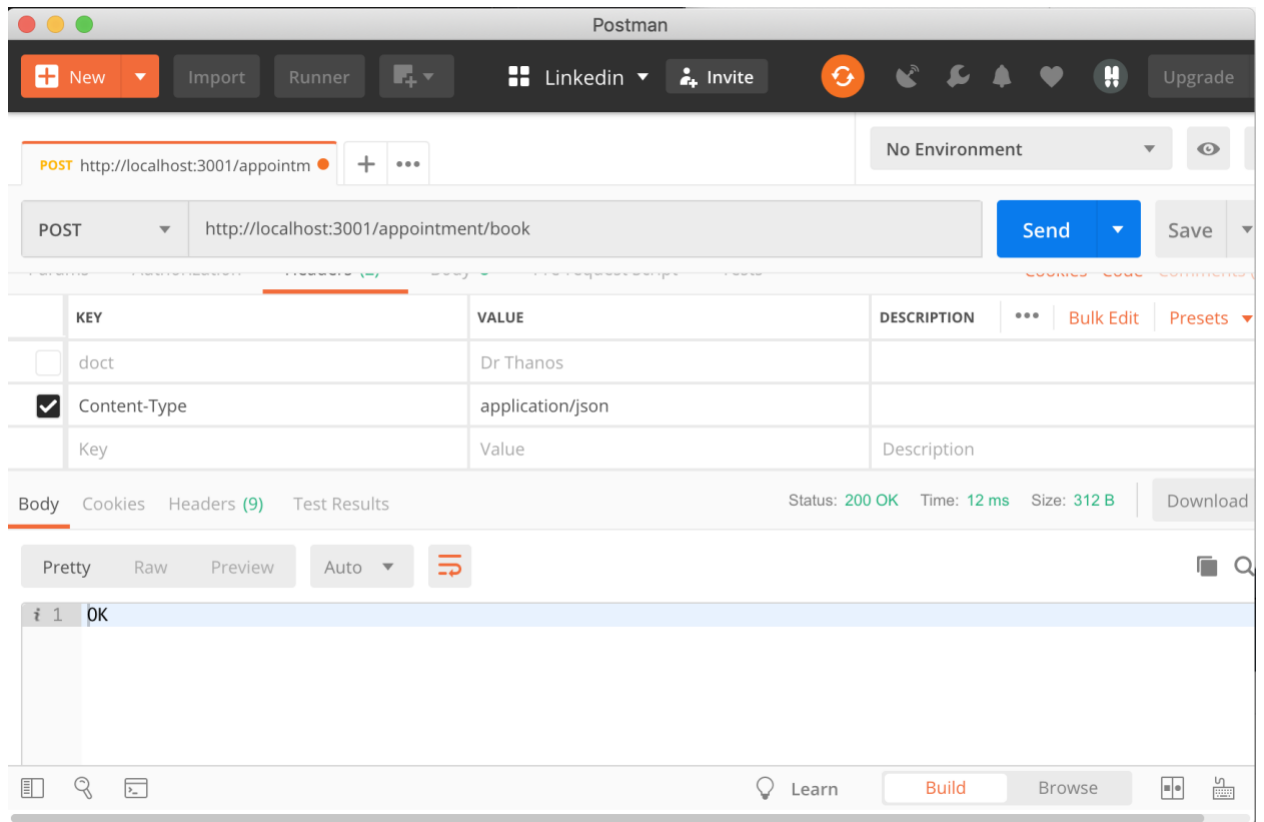
```
(base) Pranjals-MBP:REST_API_Node_Project pranzsmac$ node index.js
Listening on port #3001
Method to get doctor's hours
name of doctor : Dr Thanos
{ Monday: '10AM to 12PM',
  Tuesday: '10AM to 12PM , 2PM to 5PM',
  Friday: '10AM to 12PM , 2PM to 5PM' }
appointment [object Object]
before change : {"10AM to 12PM":true,"2PM to 5PM":false}
before change : {"10AM to 12PM":false,"2PM to 5PM":false}
Appointments : {"doctor_name":"Dr Thanos","patient_name":"Iron Man","day":"Friday","timing":"10AM to 12PM"}
[]
```

The Postman window shows a POST request to <http://localhost:3001/appointment/book>. The request body is a JSON object:

```
{
  "doctor" : "Dr Thanos",
  "day" : "Friday",
  "patient" : "Iron Man",
  "timing" : "10AM to 12PM"
}
```

The Postman interface includes tabs for Params, Authorization, Headers (2), Body, Pre-request Script, Tests, and Cookies. The Body tab is selected, and the request is set to raw JSON (application/json). The bottom of the Postman window shows a 'Learn' button and a 'Build' button.

response:



- 3) Create and update the list of doctor's working hours
Create hours call:

Call: POST: <http://localhost:3001/doctors/addHours>

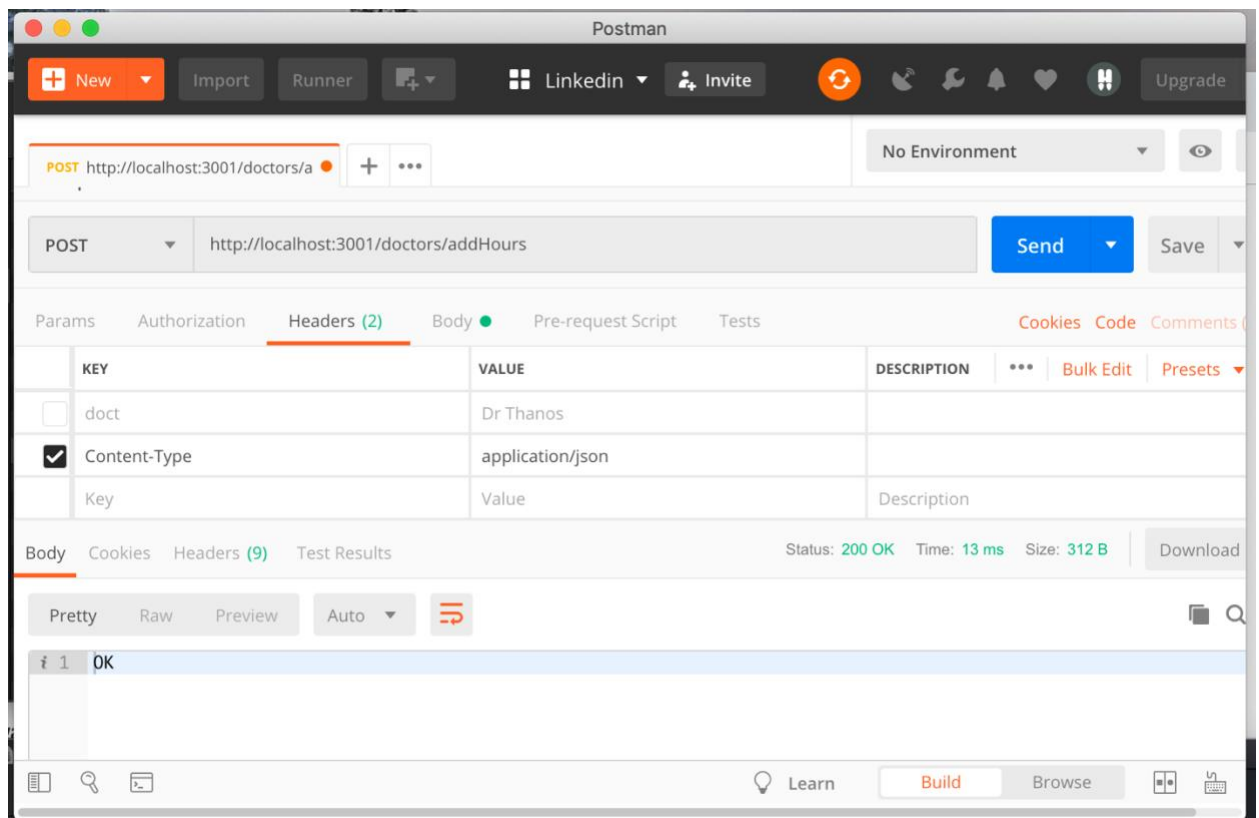
The image shows a terminal window and the Postman application. The terminal window, titled 'REST_API_Node_Project — node index.js — 94x32', displays the output of running 'node index.js'. It shows the server listening on port 3001 and the current state of Dr. Thanos's hours: Monday (10AM to 12PM), Tuesday (10AM to 12PM, 2PM to 5PM), and Friday (10AM to 12PM, 2PM to 5PM). It also shows the process of adding a new hour (5PM to 6PM) on Monday, updating the 'addHour' field in the JSON body.

The Postman application is configured for a POST request to <http://localhost:3001/doctors/addHours>. The request body is a JSON object:

```
{
  "doctor": "Dr Thanos",
  "day": "Monday",
  "addHour": "5PM to 6PM"
}
```

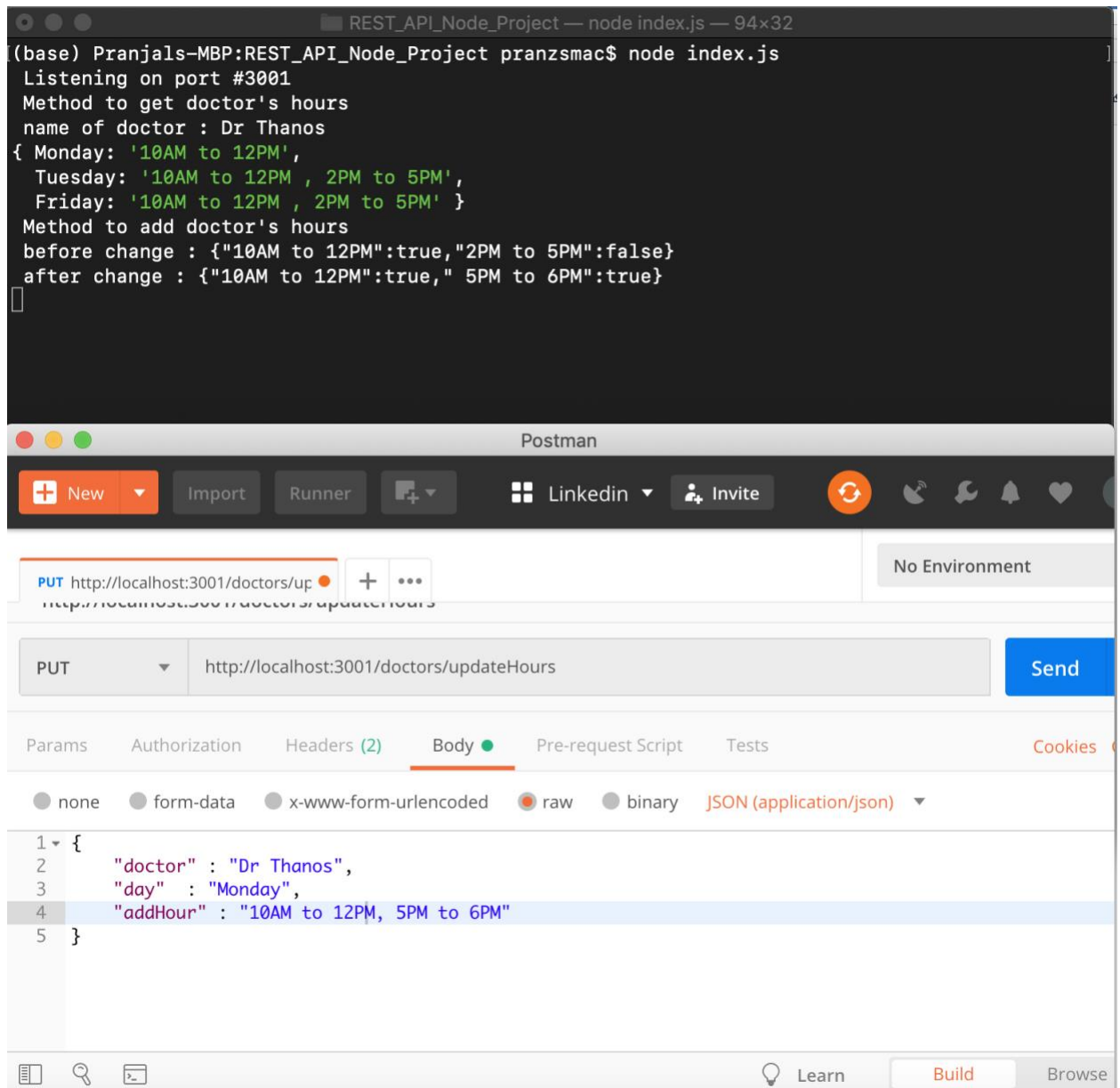
The 'Body' tab is selected, and the content type is set to 'JSON (application/json)'.

response:



Update doctor's hours:

Call: PUT: <http://localhost:3001/doctors/updateHours>



response:

