

## Details and Screenshots of REST APIs

1. An endpoint to book a ticket using a user's name, phone number, and timings

**Url :** [http://localhost:8080/Ticket\\_Booking/webapi/users/bookTicket](http://localhost:8080/Ticket_Booking/webapi/users/bookTicket)

**Body:**

```
{  
    "showTiming": "2020-08-30 18:00:00",  
    "userName": "arvind",  
    "phoneNumber": "1234567"  
}
```

**Response :** ticket generated successfully

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** [http://localhost:8080/Ticket\\_Booking/webapi/users/bookTicket](http://localhost:8080/Ticket_Booking/webapi/users/bookTicket)
- Body Tab:** Selected, showing the request body as JSON:

```
{  
  "showTiming": "2020-08-30 18:00:00",  
  "userName": "arvind",  
  "phoneNumber": "1234567"  
}
```
- Response Tab:** Selected, showing the response body as JSON:

```
{  
  "message": "ticket generated successfully"  
}
```

## 2. An endpoint to update a ticket timing.

Url : [http://localhost:8080/Ticket\\_Booking/webapi/users/updateTicketTiming](http://localhost:8080/Ticket_Booking/webapi/users/updateTicketTiming)

Body :

```
{  
    "showTiming": "2020-09-01 11:00:00",  
    "ticketId": 7  
}
```

Response : Ticket timing is updated

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** [http://localhost:8080/Ticket\\_Booking/webapi/users/updateTicketTiming](http://localhost:8080/Ticket_Booking/webapi/users/updateTicketTiming)
- Body Tab:** Selected, showing a JSON payload:

```
{  
  "showTiming": "2020-09-01 11:00:00",  
  "ticketId": 7  
}
```
- Response Tab:** Selected, showing the response body:

```
1 ticket timing is updated
```

### 3. An endpoint to view all the tickets for a particular time

Url : [http://localhost:8080/Ticket\\_Booking/webapi/users/viewAllTickets](http://localhost:8080/Ticket_Booking/webapi/users/viewAllTickets)

Body :

```
{  
    "showTiming": "2020-08-30 18:00:00"  
}
```

Response :

```
{  
    "showName": "Dabangg",  
    "showTiming": "2020-09-01 11:00:00.0",  
    "status": "active",  
    "ticketId": 7,  
    "userName": "Manish Garg"  
}
```

The screenshot displays a REST client interface. At the top, a POST request is configured for the URL `http://localhost:8080/Ticket_Booking/webapi/users/viewAllTickets`. The 'Body' tab is selected, showing a JSON payload: `{ "showTiming": "2020-09-01 11:00:00" }`. Below the request, the 'Response' tab is active, showing the received JSON data in 'Pretty' format: `{ "showName": "Dabangg", "showTiming": "2020-09-01 11:00:00.0", "status": "active", "ticketId": 7, "userName": "Manish Garg" }`. The interface includes tabs for Params, Authorization, Headers (8), Body, Pre-request Script, Tests, and Settings. The response section also has tabs for Body, Cookies, Headers (5), and Test Results.

#### 4. An endpoint to delete a particular ticket

**Url :** [http://localhost:8080/Ticket\\_Booking/webapi/users/deleteTicket](http://localhost:8080/Ticket_Booking/webapi/users/deleteTicket)

**Body :**

```
{  
    "ticketId": 9  
}
```

**Response :** Ticket is deleted at this Id

The screenshot displays a REST client interface with the following components:

- Request Bar:** Method **POST** and URL `http://localhost:8080/Ticket_Booking/webapi/users/deleteTicket`.
- Request Body:** The **Body** tab is selected. The format is set to **JSON**. The body content is:

```
{  
    "ticketId": 9  
}
```
- Response Section:** The **Body** tab is selected. The response is displayed in **Pretty** format as **JSON**. The response text is:

```
1 Ticket is deleted at this Id
```

## 5. An endpoint to view the users details based on the ticket id

**Url :** [http://localhost:8080/Ticket\\_Booking/webapi/users/userDetail/12](http://localhost:8080/Ticket_Booking/webapi/users/userDetail/12)

**Body :** null

**Response :**

```
{  
  "id": 4,  
  "name": "arvind",  
  "number": "1234567",  
  "ticketId": 12  
}
```

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** [http://localhost:8080/Ticket\\_Booking/webapi/users/userDetail/12](http://localhost:8080/Ticket_Booking/webapi/users/userDetail/12)
- Body Tab:** Selected, showing a JSON object: 

```
{
```
- Response Tab:** Selected, showing the full JSON response: 

```
{  
  "id": 4,  
  "name": "arvind",  
  "number": "1234567",  
  "ticketId": 12  
}
```
- Status:** 200 OK

6. Mark a ticket as expired if there is a diff of 8 hours between the ticket timing and current time.

Url : [http://localhost:8080/Ticket\\_Booking/webapi/users/updateTicketStatus](http://localhost:8080/Ticket_Booking/webapi/users/updateTicketStatus)

Body : null

Response : All Ticket Status is updated

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** `http://localhost:8080/Ticket_Booking/webapi/users/updateTicketStatus`
- Body Tab:** Selected, showing a JSON object: `{`
- Response Tab:** Selected, showing the response body: `All Ticket Status is updated`

The interface includes tabs for Params, Authorization, Headers (8), Body, Pre-request Script, Tests, and Settings. The response is formatted as JSON, and the status bar indicates the response is successful.

## 7. For a particular timing, a maximum of 20 tickets can be booked

**Url :** [http://localhost:8080/Ticket\\_Booking/webapi/users/bookTicket](http://localhost:8080/Ticket_Booking/webapi/users/bookTicket)

**Body :**

```
{  
    "showTiming": "2020-08-30 18:00:00",  
    "userName": "arvind",  
    "phoneNumber": "1234567"  
}
```

**Response :** Show is full at this time

The screenshot displays a REST client interface for a POST request to `http://localhost:8080/Ticket_Booking/webapi/users/bookTicket`. The 'Body' tab is selected, showing a JSON payload: `{ "userName": "arvind", "phoneNumber": "1234567" }`. The 'JSON' format is chosen. Below the request, the 'body' tab shows the response: `Show is full at this time`. The status bar at the bottom right indicates a `200 OK` status.

POST `http://localhost:8080/Ticket_Booking/webapi/users/bookTicket`

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

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

```
3     "userName": "arvind",  
4     "phoneNumber": "1234567"  
5 }  
6
```

body Cookies Headers (5) Test Results 🌐 Status: 200 OK Tim

Pretty Raw Preview Visualize **JSON**

```
1 Show is full at this time
```

8. Delete all the tickets which are expired automatically.

**Url :** [http://localhost:8080/Ticket\\_Booking/webapi/users/deleteExpiredStatus](http://localhost:8080/Ticket_Booking/webapi/users/deleteExpiredStatus)

**Body :** null

**Response :** Expired Tickets deleted successfully

The screenshot displays a REST client interface with the following components:

- Request Section:**
  - Method: POST
  - URL: `http://localhost:8080/Ticket_Booking/webapi/users/deleteExpiredStatus`
  - Tab: Body (selected)
  - Content Type: raw (selected)
  - Body content: `{`
- Response Section:**
  - Tab: Body (selected)
  - Format: JSON (selected)
  - Response content: `Expired Tickets deleted successfully`