

## Listing All of the RESTful API's

### Booking API's

#### 1. ✓ Create a Booking

- ↪ Endpoint: /book-slot
- ↪ Method: POST
- ↪ Purpose: Book a time slot with a tutor.
- ↪ Request Body:

Python

```
{  
    "student_id": 2,  
    "tutor_id": 1,  
    "date": "2025-01-20",  
    "start_time": "10:00",  
    "end_time": "11:00",  
    "time_zone": "UTC"  
}
```

- ↪ Response:

Python

```
{  
    "message": "Booking confirmed"  
}
```

#### 2. ✓ Fetch Bookings for a Student

- ↪ Endpoint: /student-bookings/{student\_id}
- ↪ Method: GET
- ↪ Purpose: Fetch all bookings for a specific student.
- ↪ Response:

Python

```
[
```

```
{
  "tutor_name": "John Doe",
  "date": "2025-01-20",
  "start_time": "10:00",
  "end_time": "11:00",
  "status": "confirmed"
}
```

### 3. ✓ Fetch Bookings for a Tutor

- ↪ Endpoint: /tutor-bookings/{tutor\_id}
- ↪ Method: GET
- ↪ Purpose: Fetch all bookings for a specific tutor.
- ↪ Response:

Python

```
[
  {
    "student_name": "Jane Doe",
    "date": "2025-01-20",
    "start_time": "10:00",
    "end_time": "11:00",
    "status": "confirmed"
  }
]
```

### 4. Delete a Booking

- ↪ Endpoint: /delete-booking/{booking\_id}
- ↪ Method: DELETE
- ↪ Purpose: Delete a specific booking by ID.
- ↪ Response:

Python

```
{  
  "message": "Booking deleted successfully"  
}
```

## 5. Update a Booking

- ↪ Endpoint: /update-booking/{booking\_id}
- ↪ Method: PUT
- ↪ Purpose: Update the details of a specific booking.
- ↪ Request:

Python

```
{  
  "student_id": 2,  
  "tutor_id": 1,  
  "date": "2025-01-20",  
  "start_time": "10:30",  
  "end_time": "11:30",  
  "time_zone": "UTC"  
}
```

- ↪ Response:

Python

```
{  
  "message": "Booking updated successfully"  
}
```

## Availability API's

### 6. ✓ Add Tutor Availability

- ↪ Endpoint: /availability/add
- ↪ Method: POST
- ↪ Purpose: Add tutor availability.
- ↪ Request Body:

Python

```
{
    "tutor_id": 1,
    "date": "2025-01-20",
    "start_time": "09:00",
    "end_time": "12:00",
    "time_zone": "UTC",
    "recurrence": "daily", // "daily", "weekly", or ""
    "occurrences": 5
}
```

- ↪ Response:

Python

```
{
    "message": "Availabilities added successfully"
}
```

### 7. ✓ Retrieve All Availabilities

- ↪ Endpoint: /availability/view
- ↪ Method: GET
- ↪ Purpose: View all availabilities, converted to a specific timezone. This is for students to see all of the tutors availability.
- ↪ Query Parameter:
  - timezone (optional): Target timezone (e.g., UTC, Europe/London).
- ↪ Response:

Python

```
[
  {
    "tutor": {
      "id": 1,
      "name": "John Doe",
      "subject": "Math"
    },
    "availability": {
      "date": "2025-01-20",
      "start_time": "10:00",
      "end_time": "12:00",
      "time_zone": "UTC"
    },
    "start_time_converted": "10:00",
    "end_time_converted": "12:00",
    "converted_timezone": "UTC"
  }
]
```

## 8. ✓ Retrieve a Tutor's Availability

- ↪ Endpoint: /availability/{tutor\_id}
- ↪ Method: GET
- ↪ Purpose: Retrieving all of the availabilities for a specific tutor.
- ↪ Response:

Python

```
[
  {
    "date": "2025-01-20",
    "start_time": "09:00",
    "end_time": "12:00",
    "time_zone": "UTC",
    "status": "available" // "available" or "booked"
  }
]
```

```
}  
]
```

## 9. Update Tutor Availability

- Endpoint: /update-availability/{availability\_id}
- Method: PUT
- Purpose: Update the details of a specific availability.
- Request:

Python

```
{  
    "tutor_id": 1,  
    "date": "2025-01-21",  
    "start_time": "10:00",  
    "end_time": "13:00",  
    "time_zone": "UTC"  
}
```

- Response:

Python

```
{  
    "message": "Availability updated successfully"  
}
```

## 10. Delete Tutor Availability

- Endpoint: /delete-availability/{availability\_id}
- Method: DELETE
- Purpose: Delete a specific availability by ID.
- Response:

Python

```
{  
  "message": "Availability deleted successfully"  
}
```

## Notifications API's

### 11. Notifications for Updates:

- ↪ Endpoint: /notifications
- ↪ Method: POST
- ↪ Purpose: Log or send notifications for bookings or availability updates.
- ↪ Request Body:

Python

```
{  
    "type": "booking",  
    "message": "New booking confirmed for tutor 1."  
}
```

- ↪ Response:

Python

```
{  
    "message": "Notification sent successfully"  
}
```



## Student API's

### 12. Create a Student

- ↳ Endpoint: /students
- ↳ Method: POST
- ↳ Purpose: Add a new student.
- ↳ Request Body:

```
Python
{
  "name": "John Doe",
  "email": "johndoe@example.com",
  "timezone": "UTC"
}
```

- ↳ Response:

```
Python
{
  "message": "Student added successfully"
}
```

### 13. Retrieve a Student

- ↳ Endpoint: /students/{student\_id}
- ↳ Method: GET
- ↳ Purpose: Fetch details of a specific student by ID.
- ↳ Response:

```
Python
{
  "id": 1,
  "name": "John Doe",
  "email": "johndoe@example.com",
  "timezone": "UTC"
}
```

## 14. Update a Student

- ↪ Endpoint: /students/{student\_id}
- ↪ Method: PUT
- ↪ Purpose: Update the details of a specific student.
- ↪ Request Body:

Python

```
{  
    "name": "John Doe Updated",  
    "email": "johndoe_updated@example.com",  
    "timezone": "America/New_York"  
}
```

- ↪ Response:

Python

```
{  
    "message": "Student updated successfully"  
}
```

## 15. Update a Student

- ↪ Endpoint: /students/{student\_id}
- ↪ Method: DELETE
- ↪ Purpose: Delete a student by ID.
- ↪ Response:

Python

```
{  
    "message": "Student deleted successfully"  
}
```

## Tutor API's

### 16. Create a Tutor

- ↳ Endpoint: /tutors
- ↳ Method: POST
- ↳ Purpose: Add a new tutor.
- ↳ Request Body:

Python

```
{
  "name": "Jane Smith",
  "email": "janesmith@example.com",
  "subject": "Math"
}
```

- ↳ Response:

Python

```
{
  "message": "Tutor added successfully"
}
```

### 17. Retrieve a Tutor

- ↳ Endpoint: /tutors/{tutor\_id}
- ↳ Method: GET
- ↳ Purpose: Fetch details of a specific tutor by ID.
- ↳ Response:

Python

```
{
  "id": 1,
  "name": "Jane Smith",
  "email": "janesmith@example.com",
  "subject": "Math"
}
```

## 18. Update a Tutor

- ↪ Endpoint: /tutors/{tutor\_id}
- ↪ Method: PUT
- ↪ Purpose: Update the details of a specific tutor.
- ↪ Request Body:

Python

```
{  
    "name": "Jane Smith Updated",  
    "email": "janesmith_updated@example.com",  
    "subject": "Science"  
}
```

- ↪ Response:

Python

```
{  
    "message": "Tutor updated successfully"  
}
```

## 19. Delete a Tutor

- ↪ Endpoint: /tutors/{tutor\_id}
- ↪ Method: DELETE
- ↪ Purpose: Delete a tutor by ID.
- ↪ Response:

Python

```
{  
    "message": "Tutor deleted successfully"  
}
```