

NFC Based Attendance System

Description:

Developing an app to track attendance using the NFC tag present in student ID cards. Creating API end points to successfully update the attendance based on the unique serial ID generated for each ID-card.

Testing Restful endpoints in postman:

List of Students:

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

- URL Bar:** `http://localhost:5000/attendance`
- Method:** `GET`
- Send Button:** A blue button labeled "Send".
- Params Tab:** Shows "Query Params" with a table:

Key	Value	Description	Bulk Edit
Key	Value	Description	

- Body Tab:** Shows the response in JSON format:

```
1 {
2   {
3     "id": 1,
4     "name": "Alice",
5     "serial_id": "123456"
6   },
7   {
8     "id": 2,
9     "name": "Bob",
10    "serial_id": "234567"
11  },
12  {
13    "id": 3,
14    "name": "Charlie",
15    "serial_id": "345678"
16  }
17 }
```

Additional interface details include tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings. The status bar at the bottom indicates "Status: 200 OK", "Time: 10 ms", and "Size: 375 B".

Posting the attendance of the student with the serial id: 123456

The screenshot shows a REST client interface with the URL `http://localhost:5000/attendance`. The method is set to **POST**. The body is a JSON object: `{ "serial_id": "123456" }`. The response status is **201 CREATED** with a time of 31 ms and a size of 221 B. The response body is `{ "success": "Attendance recorded successfully" }`.

The attendance is recorder successfully

Getting the attendance details of the student with id=1:

The screenshot shows a REST client interface with the URL `http://localhost:5000/attendance/1`. The method is set to **GET**. The response status is **200 OK** with a time of 39 ms and a size of 2.43 KB. The response body is a JSON array of three attendance records for student id=1:

```
[{"date": "2023-07-09", "id": 1, "student_id": 1, "time": "22:40:52"}, {"date": "2023-07-09", "id": 3, "student_id": 1, "time": "22:40:52"}, {"date": "2023-07-09", "id": 4, "student_id": 1, "time": "22:40:52"}]
```

Here, the student with id=1 scanned his tag many number of times.

So, we got many responses

Getting the attendance details of the student with id=2:

The screenshot shows a REST client interface with the URL `http://localhost:5000/attendance/3` and the method `GET`. The response status is `200 OK` with a time of `13 ms` and a size of `259 B`. The response body is displayed in JSON format:

```
{  "date": "2023-07-09",  "id": 2,  "student_id": 3,  "time": "22:40:52"}
```

Here, the student with ID=3 scanned his tag only once, that's why we got only one response.

If we enter serial id of the student which is not in the database of list of students, we will get an error like this:

The screenshot shows a REST client interface with the URL `http://localhost:5000/attendance` and the method `POST`. The response status is `400 BAD REQUEST` with a time of `10 ms` and a size of `208 B`. The response body is displayed in JSON format:

```
{  "error": "Invalid Serial ID"}
```