

School Application backend web service

School Entity Crud Operations

1. Add new School

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

- URL:** `http://localhost:8080/addNewSchool`
- Method:** `POST`
- Status:** `200 OK`, `12 ms`, `554 B`
- Response Body (JSON):**

```
1 {
2   "id": 16,
3   "schoolName": "Green Valley Public School",
4   "schoolAddress": "45 Park Street, Delhi",
5   "teachers": [
6     {
7       "id": 1,
8       "teacherName": "Amit Sharma",
9       "teacherAddress": "Delhi",
10      "qualification": "M.Tech",
11      "age": 28,
12      "dob": "1996-06-30",
13      "experience": 4,
14      "schoolId": 11
15    },
16    {
17      "id": 2,
18      "teacherName": "Priya Singh",
19      "teacherAddress": "Delhi",
20      "qualification": "M.sc",
21      "age": 28,
22      "dob": "1996-06-21",
23      "experience": 3
```

2. Get School By Id

HTTP <http://localhost:8080/addNewSchool> Save

GET <http://localhost:8080/getSchoolById/2> Send

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

Body Cookies Headers (5) Test Results 200 OK 44 ms 510 B Save Response

Pretty Raw Preview Visualize JSON ≡

```
1 {
2   "id": 2,
3   "schoolName": "De Nobili ",
4   "schoolAddress": "Bokaro",
5   "teachers": [
6     {
7       "id": 2,
8       "teacherName": "Ram",
9       "teacherAddress": "Telangana",
10      "qualification": "M.tech",
11      "age": 30,
12      "dob": "1900-06-06",
13      "experience": 6,
14      "schoolId": 2
15    },
16    {
17      "id": 3,
18      "teacherName": "Rani",
19      "teacherAddress": "Delhi",
20      "qualification": "M.sc",
21      "age": 40,
22      "dob": "1985-06-16",
23      "experience": 15
```

3 Update

HTTP <http://localhost:8080/addNewSchool> Save

PATCH <http://localhost:8080/patchUpdateSchoolById/5> Send

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON ▼ Beautify

```
1 {
2   "schoolName": "Green Valley Public School",
```

Body Cookies Headers (5) Test Results 200 OK 36 ms 268 B Save Response

Pretty Raw Preview Visualize JSON ≡

```
1 {
2   "id": 5,
3   "schoolName": "Green Valley Public School",
4   "schoolAddress": "45 Park Street, Delhi",
5   "teachers": []
6 }
```

4. Delete

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

- URL:** `http://localhost:8080/deleteSchoolById/5`
- Method:** `DELETE`
- Body:**

```
{
  "schoolName": "Green Valley Public School",
}
```
- Response:** `200 OK`, `104 ms`, `123 B`

School With Id 5 is delete from the database

The screenshot shows a database table with the following data:

id	school_address	school_name
1	Noida	DPS
2	Bokaro	De Nobili
3	Bokaro	GGPS
4	Gurugram	Central School
6	Dehradun	The Doon School
7	New-Delhi	Amity
8	New-Delhi	Amity
9	Delhi	Modern School
16	45 Park Street, Delhi	Green Valley Public School
*	(Auto)	(NULL)

5. Get School By Name

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

- URL:** `http://localhost:8080/checkSchoolByName/dps`
- Method:** `GET`
- Body:**

```
{
  "schoolName": "Green Valley Public School",
}
```
- Response:** `200 OK`, `57 ms`, `231 B`

The response body is shown in JSON format:

```
{
  "id": 1,
  "schoolName": "DPS",
  "schoolAddress": "Noida",
  ...
}
```

Teacher Entity Crud Operations

1. Add new Teacher

The screenshot shows a Postman client interface for a POST request. The URL is `http://localhost:8080/addNewTeacher`. The request body is a JSON object with the following fields: `teacherName` (Ramesh), `teacherAddress` (Delhi), `qualification` (M.sc), `age` (36), and `dob` (1984-02-16). The response is a 200 OK status with a 16 ms latency and 300 B of data. The response body is a JSON object with the following fields: `id` (7), `teacherName` (Ramesh), `teacherAddress` (Delhi), `qualification` (M.sc), `age` (36), `dob` (1984-02-16), `experience` (12), and `schoolId` (4).

```
POST http://localhost:8080/addNewTeacher
```

Body

```
{
  "teacherName": "Ramesh",
  "teacherAddress": "Delhi",
  "qualification": "M.sc",
  "age": 36,
  "dob": "1984-02-16"
}
```

200 OK 16 ms 300 B

```
{
  "id": 7,
  "teacherName": "Ramesh",
  "teacherAddress": "Delhi",
  "qualification": "M.sc",
  "age": 36,
  "dob": "1984-02-16",
  "experience": 12,
  "schoolId": 4
}
```

2. GET Teacher By Id

The screenshot shows a Postman client interface for a GET request. The URL is `http://localhost:8080/getTeacherById/3`. The response is a 200 OK status with a 9 ms latency and 298 B of data. The response body is a JSON object with the following fields: `id` (3), `teacherName` (Rani), `teacherAddress` (Delhi), `qualification` (M.sc), `age` (40), `dob` (1985-06-16), `experience` (15), and `schoolId` (2).

```
GET http://localhost:8080/getTeacherById/3
```

200 OK 9 ms 298 B

```
{
  "id": 3,
  "teacherName": "Rani",
  "teacherAddress": "Delhi",
  "qualification": "M.sc",
  "age": 40,
  "dob": "1985-06-16",
  "experience": 15,
  "schoolId": 2
}
```

3. Update Teacher By Id(Updated experience from 14 to 12)

HTTP http://localhost:8080/addNewTeacher

PATCH http://localhost:8080/patchUpdateTeacherById/3

Send

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

none form-data x-www-form-urlencoded raw binary JSON

1 2 3 4

```
1 {
2   "experience": 12
3 }
4
```

Body Cookies Headers (5) Test Results 200 OK 17 ms 298 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 3,
3   "teacherName": "Rani",
4   "teacherAddress": "Delhi",
5   "qualification": "M.sc",
6   "age": 40,
7   "dob": "1985-06-16",
8   "experience": 12,
9   "schoolId": 2
10 }
```

4. Delete Teacher By ID Before

[illegible]

After

[illegible]

PostMan Playlod

Postman interface showing a DELETE request to `http://localhost:8080/deleteTeacherById/7`. The response is 200 OK with 82 ms latency and 123 B body. The interface includes tabs for Params, Authorization, Headers (8), Body, Pre-request Script, Tests, and Settings. The Body tab is selected, showing a single line of text: "1".

5. Check Teacher By Name

Postman interface showing a GET request to `http://localhost:8080/checkTeachersByName/Shobhit`. The response is 200 OK with 32 ms latency and 305 B body. The interface includes tabs for Params, Authorization, Headers (8), Body, Pre-request Script, Tests, and Settings. The Body tab is selected, showing the response in JSON format:

```
1 {
2   "id": 1,
3   "teacherName": "Shobhit",
4   "teacherAddress": "Bokaro",
5   "qualification": "B.tech",
6   "age": 24,
7   "dob": "2000-05-21",
8   "experience": 2,
9   "schoolId": 1
10 }
11
12
```