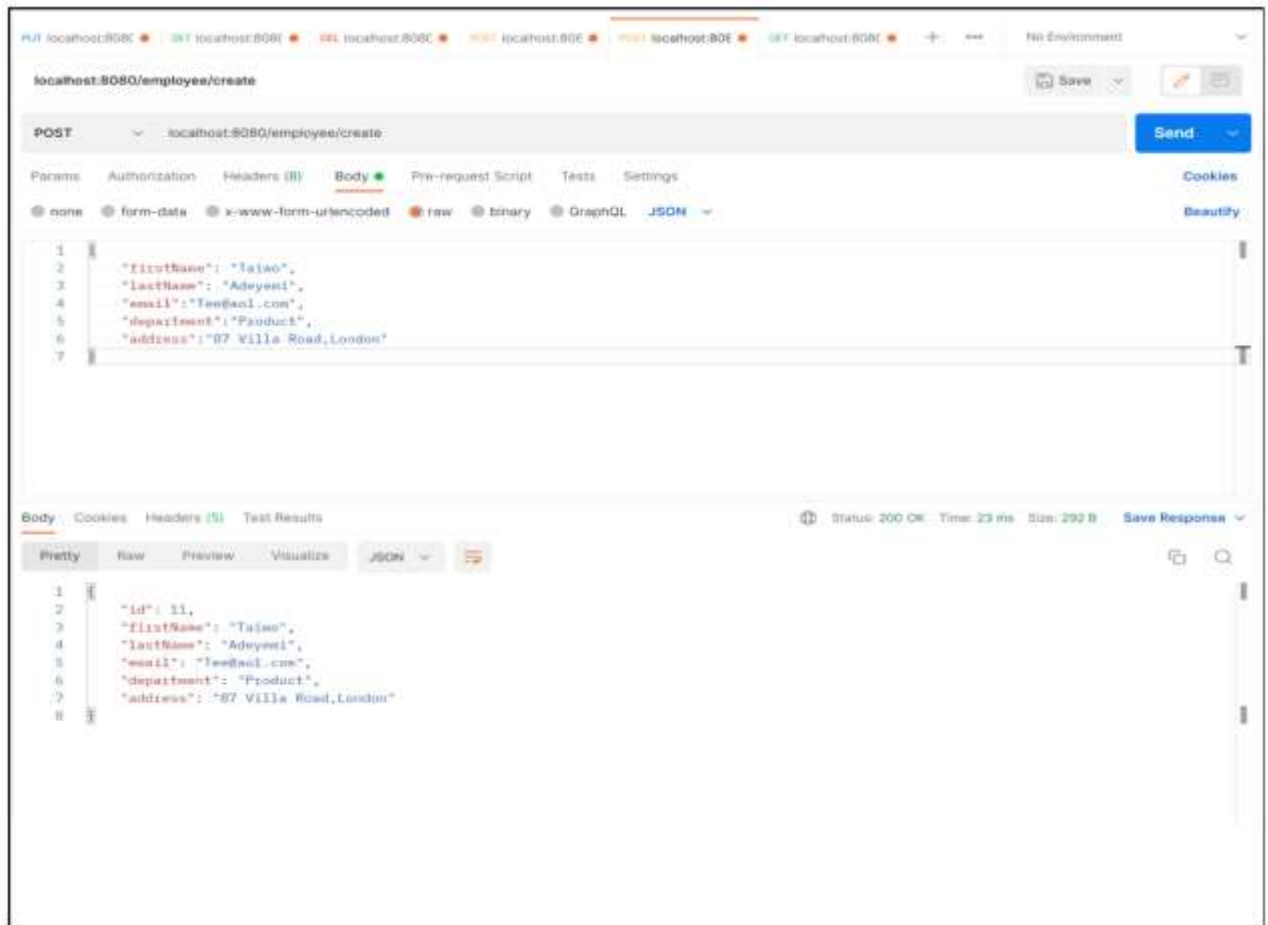
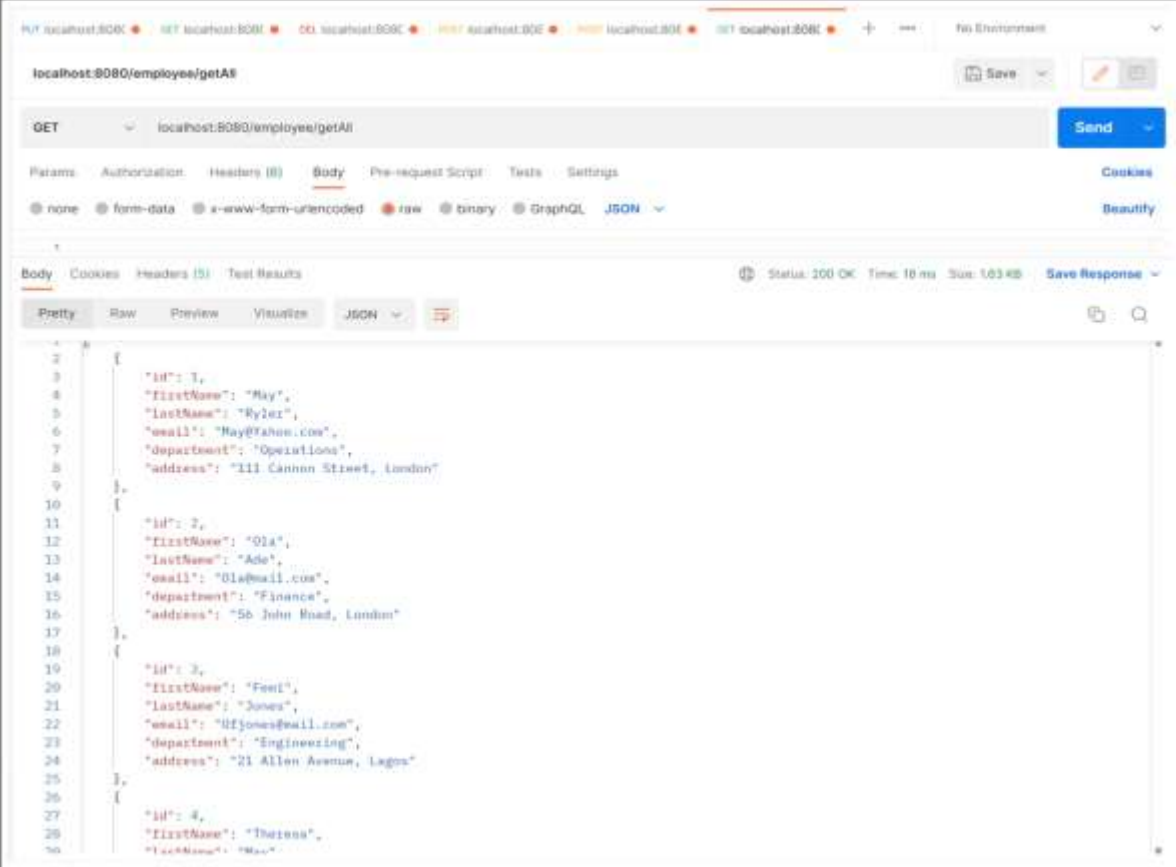


POSTMAN

1. CREATE



2. READ



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

- URL:** localhost:8080/employee/getAll
- Method:** GET
- Status:** 200 OK
- Time:** 10 ms
- Size:** 1.03 KB

The response body is a JSON array of four employee objects, displayed in the 'Body' tab with the 'Pretty' format selected. The JSON is as follows:

```
[
  {
    "id": 1,
    "firstName": "May",
    "lastName": "Ryder",
    "email": "May@yahoo.com",
    "department": "Operations",
    "address": "111 Cannon Street, London"
  },
  {
    "id": 2,
    "firstName": "Ola",
    "lastName": "Ade",
    "email": "Ola@mail.com",
    "department": "Finance",
    "address": "56 John Road, London"
  },
  {
    "id": 3,
    "firstName": "Femi",
    "lastName": "Jones",
    "email": "Hfjones@mail.com",
    "department": "Engineering",
    "address": "21 Allen Avenue, Lagos"
  },
  {
    "id": 4,
    "firstName": "Theiana",
    "lastName": "Musa"
  }
]
```

3. FIRST UPDATE

The screenshot displays a REST client interface with a PUT request to `localhost:8080/employee/update/11`. The request body is a JSON object with the following fields: `id` (11), `firstName` (Olatunke), `lastName` (Adeyemi), `email` (fen@aol.com), `department` (Product), and `address` (87 Villa Road, London). The response status is 200 OK, with a time of 148 ms and a size of 294 B. The response body is shown in a pretty-printed JSON format, matching the request body.

PUT localhost:8080/employee/update/11

PUT localhost:8080/employee/update/11

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

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

```
1 {
2   "id": 11,
3   "firstName": "Olatunke",
4   "lastName": "Adeyemi",
5   "email": "fen@aol.com",
6   "department": "Product",
7   "address": "87 Villa Road, London"
8 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 148 ms Size: 294 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 11,
3   "firstName": "Olatunke",
4   "lastName": "Adeyemi",
5   "email": "fen@aol.com",
6   "department": "Product",
7   "address": "87 Villa Road, London"
8 }
```

4. Second Update

The screenshot shows a REST client interface with a PUT request to `localhost:8080/employee/update/1`. The request body is a JSON object with the following fields:

```
{  "firstName": "Emily",  "lastName": "Bright",  "email": "Ebright@yahoo.com",  "department": "Sales",  "address": "54 Ray Street, Manchester"}
```

The response status is 200 OK, and the response body is a JSON object with the following fields:

```
{  "id": 1,  "firstName": "Emily",  "lastName": "Bright",  "email": "Ebright@yahoo.com",  "department": "Sales",  "address": "54 Ray Street, Manchester"}
```

5. DELETE

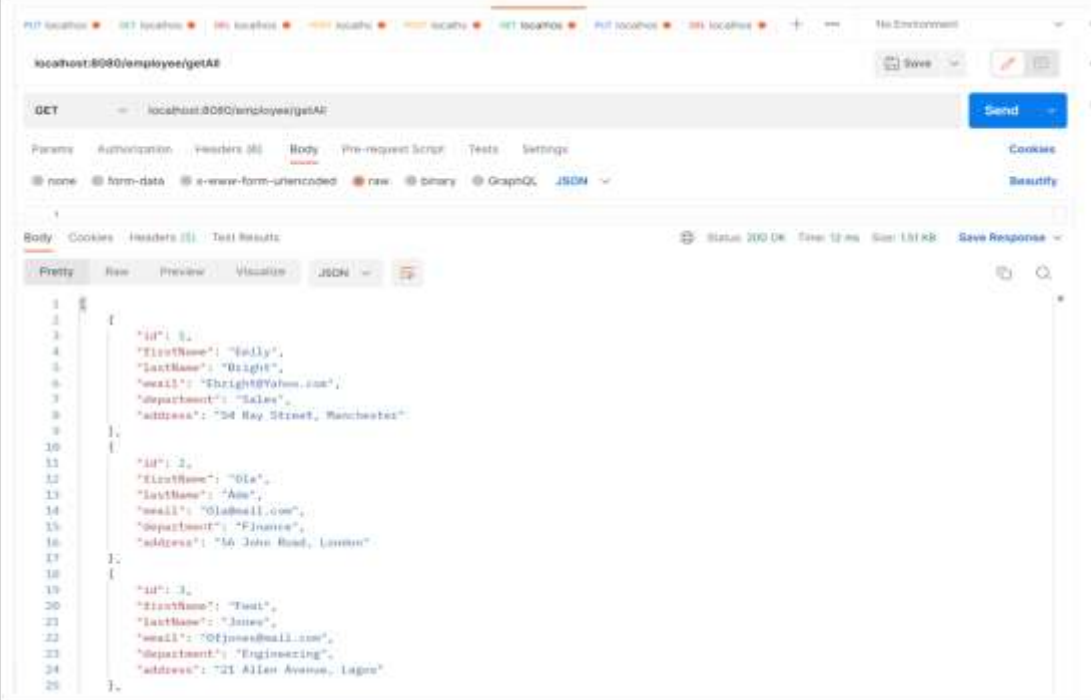
The screenshot shows a REST client interface with a DELETE request to `localhost:8080/employee/delete/11`. The request is configured with the following details:

- Method:** DELETE
- URL:** localhost:8080/employee/delete/11
- Params:** A table with 3 columns: KEY, VALUE, and DESCRIPTION. The first row contains 'Key', 'Value', and 'Description'.
- Body:** The response body is displayed in JSON format, showing `{ "true" }`.
- Status:** 200 OK, Time: 95 ms, Size: 108 B

KEY	VALUE	DESCRIPTION
Key	Value	Description


```
{ "true" }
```

6. GetAll after deleting id:11



The screenshot shows a REST client interface with a GET request to `localhost:8080/employee/getAll`. The response is a JSON array of 3 employee objects. The status is 200 OK, and the size is 1.51 KB.

```
1 {
2   {
3     "id": 1,
4     "firstName": "Emily",
5     "lastName": "Bright",
6     "email": "Thright@Yahoo.com",
7     "department": "Sales",
8     "address": "54 Ray Street, Manchester"
9   },
10  {
11    "id": 2,
12    "firstName": "Dla",
13    "lastName": "Am",
14    "email": "Dla@mail.com",
15    "department": "Finance",
16    "address": "16 John Road, London"
17  },
18  {
19    "id": 3,
20    "firstName": "Fael",
21    "lastName": "Jones",
22    "email": "Ofjones@mail.com",
23    "department": "Engineering",
24    "address": "21 Allen Avenue, Lagos"
25  }
26 }
```



The screenshot shows the continuation of the JSON response from the previous screenshot, showing employee objects with IDs 4, 5, and 6.

```
26 {
27   "id": 4,
28   "firstName": "Theresa",
29   "lastName": "May",
30   "email": "Tmay@mail.com",
31   "department": "Accounting",
32   "address": "235 Long Lane, Bexleyheath"
33 },
34 {
35   "id": 5,
36   "firstName": "Boris",
37   "lastName": "Johnson",
38   "email": "Bj@ukpremier.com",
39   "department": "Management",
40   "address": "10 Downing Street, London"
41 },
42 {
43   "id": 6,
44   "firstName": "Rishi",
45   "lastName": "Sunak",
46   "email": "Rishi@chancellor.com",
47   "department": "Finance",
48   "address": "11 Downing Street, London"
49 },
50 }
```

```
50 {
51   "id": 7,
52   "firstName": "Liz",
53   "lastName": "Truss",
54   "email": "Lizzy@secretary.com",
55   "department": "Administration",
56   "address": "12 Downing Street,London"
57 },
58 {
59   "id": 8,
60   "firstName": "Michael",
61   "lastName": "Gove",
62   "email": "Mickey@traitor.com",
63   "department": "Compliance",
64   "address": "15 Goner Street,London"
65 },
66 {
67   "id": 9,
68   "firstName": "Vin",
69   "lastName": "Diesel",
70   "email": "Vin@fastandfurious.com",
71   "department": "Marketing",
72   "address": "53 Mellanby Road,London"
73 },
```

```
74 {
75   "id": 10,
76   "firstName": "Nicky",
77   "lastName": "Olson",
78   "email": "Nicky@wunny.com",
79   "department": "Human Resources",
80   "address": "32 Battery Road,London"
81 },
82 }
```