

GET http://localhost:8888/api/employees

The screenshot shows the Postman interface with a failed API call. The URL is `http://localhost:8888/api/employees`. The response status is 401 Unauthorized, with a message "You don't have permission to access this resource". The response body is empty.

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8888/api/employees`. The response status is 200 OK, with a message "Success". The response body is a JSON array containing two employee records:

```
[{"id": 1, "firstName": "Sri", "lastName": "Vinayak", "email": "srivinayak@gmail.com"}, {"id": 2, "firstName": "Sri", "lastName": "Vinayak", "email": "srivinayak@gmail.com"}, {"id": 3, "firstName": "Sri", "lastName": "JaiVinayaka", "email": "srijaivinayaka@gmail.com"}]
```

POST http://localhost:8888/api/employees/createEmployee

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for Home, Workspaces, Explore, and a search bar labeled "Search Postman". On the right side of the header, there are buttons for "Sign In" and "Create Account". The main workspace displays a history of API requests on the left, including various GET and POST requests to the "/api/employees" endpoint. The central area shows a POST request to "http://localhost:8888/api/employees/createEmployee". The "Body" tab is selected, showing a JSON payload:

```
1 {
2   "firstName": "JaiSri",
3   "lastName": "Vinayaka",
4   "email": "jaisrivinayaka@gmail.com"
5 }
```

Below the body, the status bar indicates "Status: 200 OK", "Time: 325 ms", and "Size: 412 B". The bottom of the screen features a toolbar with various icons and a status bar showing "USD/INR -0.20%", "1:43 PM", and "5/19/2024".

This screenshot of Postman shows the same setup as the first one, but the response tab is now active. The response body is displayed in "Pretty" format:

```
1 {
2   "id": 4,
3   "firstName": "JaiSri",
4   "lastName": "Vinayaka",
5   "email": "jaisrivinayaka@gmail.com"
6 }
```

The status bar at the bottom indicates "Status: 200 OK", "Time: 867 ms", and "Size: 250 B". The rest of the interface is identical to the first screenshot.

GET http://localhost:8888/api/employees/2

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, Explore, Sign In, and Create Account. A search bar labeled "Search Postman" is also present. The main area displays a history of API requests. A specific request is selected: "GET http://localhost:8888/api/employees/2". The request details show the method (GET), URL (http://localhost:8888/api/employees/2), and various configuration options like Params, Authorization, Headers, Body, Pre-request Script, Tests, and Settings. The Headers section is currently active, showing "(none)". The Body section notes "This request does not have a body". Below the request details, there is a "Response" section which is currently collapsed. At the bottom of the interface, there is a toolbar with icons for file operations and a status bar indicating "1:46 PM 5/19/2024".

This screenshot shows the same Postman interface as the previous one, but the "Response" section is now expanded. It displays the JSON response from the API call. The response body is as follows:

```
1 {"id": 2,
2   "firstName": "Str",
3   "lastName": "Vinyaka",
4   "email": "stivinaya@gmail.com"
5 }
```

The "Body" tab is selected in the response panel. Other tabs available include Cookies, Headers, and Test Results. The status bar at the bottom indicates "Status: 200 OK Time: 22 ms Size: 244 B Save Response".

PATCH <http://localhost:8888/api/employees/2>

The screenshot shows the Postman application interface. On the left, there's a sidebar titled "History" with a tree view of API requests categorized by date: "Today", "Yesterday", and "May 17". The "Today" section has several entries under "employees". The main workspace shows a "PATCH" request to "http://localhost:8888/api/employees/2". The "Body" tab displays a JSON payload:

```
1
2   {
3     "id": 2,
4     "firstname": "Saikarthika",
5     "lastname": "Shanmuga",
6     "email": "saikarthika@gmail.com"
7 }
```

The "Body" tab also includes tabs for "Pretty", "Raw", "Preview", "Visualize", and "JSON". Below the body, the status bar shows "Status: 200 OK Time: 22 ms Size: 244 B Save Response". At the bottom, there are navigation icons for "Console", "Search", and "Home".

The screenshot shows the Postman application interface. On the left, there's a sidebar titled 'History' with a tree view of API requests categorized by date: Today, Yesterday, and May 17. The main workspace shows a 'PATCH' request to 'http://localhost:8888/api/employees/2'. The 'Body' tab is selected, displaying a JSON payload:

```
1 {  
2   "id": 2,  
3   "firstname": "JaiKarthikeya",  
4   "lastname": "Shannuga",  
5   "email": "jaikarthikeya@gmail.com"  
6 }
```

Below the body, the 'Test Results' section shows the response status as 200 OK with a time of 165 ms and a size of 256 B. The response body is identical to the JSON sent in the body tab.

## POST http://localhost:8888/api/employees/createEmployee

The screenshot shows the Postman application interface. In the center, there is a request card for a POST method to the URL `http://localhost:8888/api/employees/createEmployee`. The request body is set to JSON and contains the following data:

```
1  {
2     "firstName": "JaiVinay",
3     "lastName": "Srinivasan",
4     "email": "vinaysrinib@gmail.com"
5 }
```

Below the request card, the response section shows a status of 200 OK, a response time of 165 ms, and a response size of 256 B. The response body is displayed in pretty JSON format:

```
1  {
2     "id": 8,
3     "firstName": "JaiVinay",
4     "lastName": "Srinivasan",
5     "email": "vinaysrinib@gmail.com"
6 }
```

The Postman interface includes a sidebar with a history of API calls and a bottom navigation bar with various icons.

This screenshot is nearly identical to the one above, showing the same POST request to `http://localhost:8888/api/employees/createEmployee` with the same JSON payload and a successful response. The response body is identical to the previous one:

```
1  {
2     "id": 8,
3     "firstName": "JaiVinay",
4     "lastName": "Srinivasan",
5     "email": "vinaysrinib@gmail.com"
6 }
```

The interface and environment details at the bottom are also identical, including the weather icon showing 37°C Sunny and the date/time 1:51 PM 5/19/2024.

GET http://localhost:8888/api/employees/5

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, Explore, and a search bar labeled "Search Postman". On the right side of the header, there are "Sign In" and "Create Account" buttons. The main area has a "History" sidebar on the left listing various API requests made today, yesterday, and on May 17. The main workspace shows a "GET" request to "http://localhost:8888/api/employees/5". The "Body" tab is selected, showing the response body as "This request does not have a body". At the bottom of the workspace, there are tabs for Body, Cookies, Headers (5), and Test Results. Below the workspace, there is a toolbar with icons for various operations like Create, Import, Export, and a search bar. The status bar at the bottom right shows the date and time as 5/19/2024 1:53 PM.

This screenshot shows the same Postman interface as the previous one, but the response body is now visible. It displays a JSON object representing an employee record:

```
1 {  
2     "id": 6,  
3     "firstName": "JaiVinay",  
4     "lastName": "Srinivasan",  
5     "email": "vinaysrin@gmail.com"  
6 }
```

The rest of the interface is identical to the first screenshot, including the history sidebar, toolbars, and status bar.

GET http://localhost:8888/api/employees/5

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, Explore, Sign In, and Create Account. The main area displays a history of API requests. A specific GET request to `http://localhost:8888/api/employees/5` is selected. The request details panel shows the method (GET), URL (`http://localhost:8888/api/employees/5`), and various tabs like Params, Authorization, Headers, Body, and Settings. The Body tab is currently active, showing the message "This request does not have a body". At the bottom of the request details panel, there are tabs for Body, Cookies, Headers, and Test Results. The status bar at the bottom right indicates a 200 OK response with a time of 15 ms and a size of 250 B.

This screenshot shows the same Postman interface as the previous one, but the response body now contains JSON data for employee 5. The JSON output is as follows:

```
1
2
3   "id": 5,
4   "firstName": "JaiVinay",
5   "lastName": "Srinivasan",
6   "email": "vinayssini@gmail.com"
```

The rest of the interface and status bar are identical to the first screenshot.

GET http://localhost:8888/api/employees

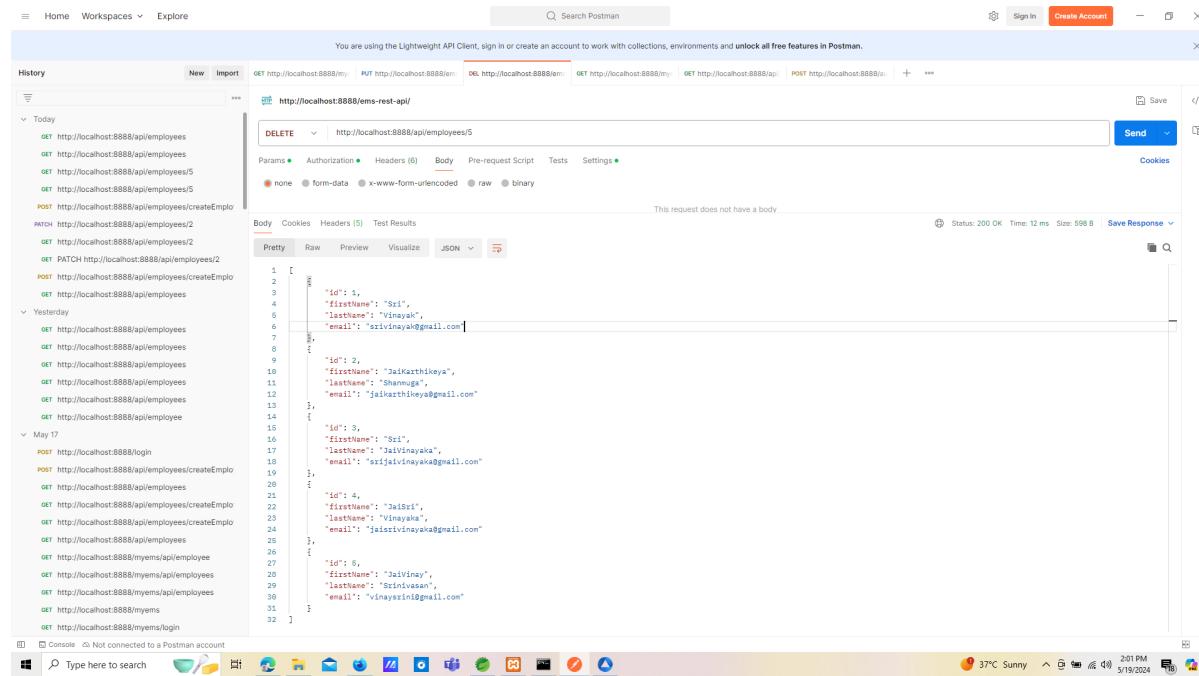
The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'Explore', 'Sign In', and 'Create Account'. A search bar at the top right says 'Search Postman'. Below the header is a message: 'You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.' The main area has a 'History' sidebar on the left listing various API requests. The central workspace shows a 'GET' request to 'http://localhost:8888/api/employees'. The 'Headers' tab is selected, showing '(none)'. The 'Body' tab indicates 'This request does not have a body'. At the bottom, tabs for 'Body', 'Cookies', 'Headers (5)', and 'Test Results' are visible, along with status information: 'Status: 200 OK', 'Time: 18 ms', 'Size: 598 B', and a 'Save Response' button.

This screenshot shows the same Postman interface after executing the GET /api/employees request. The central workspace displays the raw JSON response. The response contains an array of employee objects, each with properties: id, firstName, lastName, and email. The JSON output is as follows:

```
1 [
2   {
3     "id": 1,
4     "firstName": "Sri",
5     "lastName": "Vinayak",
6     "email": "sri.vinayak@gmail.com"
7   },
8   {
9     "id": 2,
10    "firstName": "JaiKarthika",
11    "lastName": "Shanuga",
12    "email": "jai.karthika@gmail.com"
13 },
14 {
15   "id": 3,
16   "firstName": "Sri",
17   "lastName": "JaiVinaya",
18   "email": "sri.jai.vinaya@gmail.com"
19 },
20 {
21   "id": 4,
22   "firstName": "Aishwarya",
23   "lastName": "Vinayak",
24   "email": "aishwarya.vinayak@gmail.com"
25 },
26 {
27   "id": 5,
28   "firstName": "JaiVinay",
29   "lastName": "Srinivasan",
30   "email": "vinay.srinivasan@gmail.com"
31 },
32 ]
```

The bottom status bar shows 'Status: 200 OK', 'Time: 12 ms', 'Size: 598 B', and a 'Save Response' button.

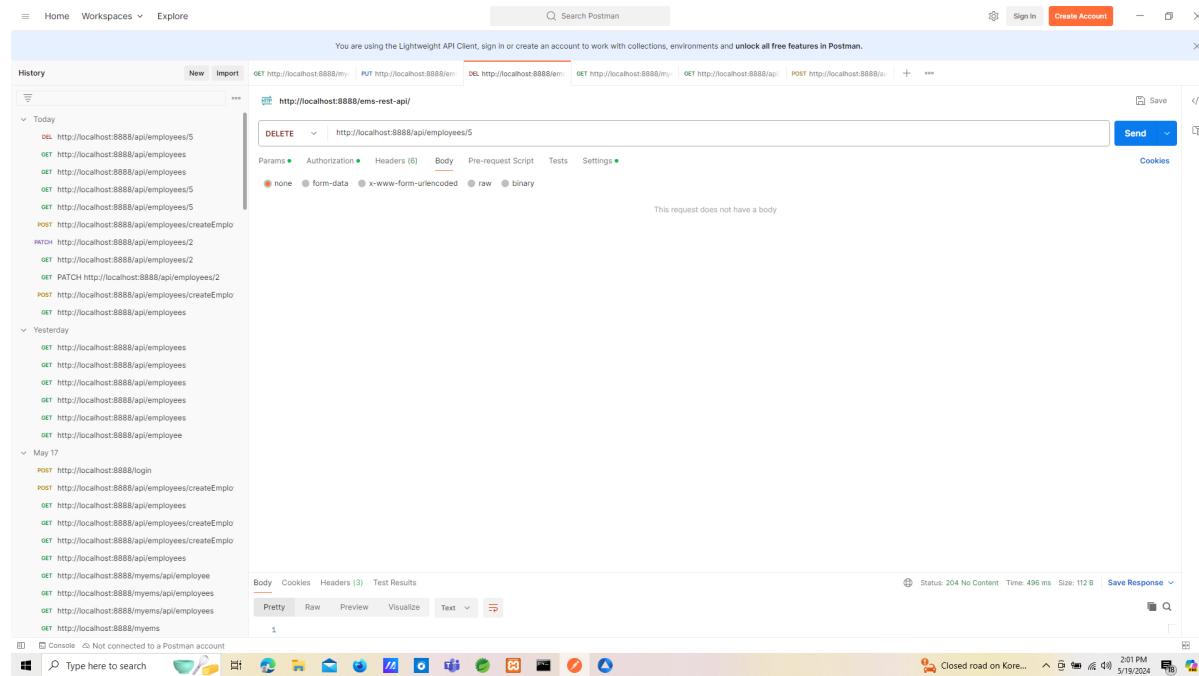
DELETE <http://localhost:8888/api/employees/5>



The screenshot shows the Postman interface with a DELETE request to `http://localhost:8888/api/employees/5`. The request body contains the following JSON:

```
[{"id": 1, "firstName": "Sai", "lastName": "Vinayak", "email": "sai.vinayak@gmail.com"}, {"id": 2, "firstName": "JaiKaarthik", "lastName": "Shanmuga", "email": "jai.jaikarthik@gmail.com"}, {"id": 3, "firstName": "Sai", "lastName": "Vinayaka", "email": "sai.jai.vinayaka@gmail.com"}, {"id": 4, "firstName": "Jaivinay", "lastName": "Srinivasan", "email": "jaivinay.srinivasan@gmail.com"}, {"id": 5, "firstName": "Jaivinay", "lastName": "Srinivasan", "email": "vinay.srinivasan@gmail.com"}]
```

The response status is 200 OK, Time: 12 ms, Size: 598 B.



The screenshot shows the Postman interface with a DELETE request to `http://localhost:8888/api/employees/5`. The response status is 204 No Content, Time: 496 ms, Size: 112 B.

GET http://localhost:8888/api/employees

The screenshot shows the Postman interface with a successful API call. The URL in the main window is `http://localhost:8888/api/employees`. The response status is 204 No Content, time 496 ms, size 112 B. The response body is empty, indicated by the message "This request does not have a body".

The screenshot shows the Postman interface with a successful API call. The URL in the main window is `http://localhost:8888/api/employees`. The response status is 200 OK, time 10 ms, size 511 B. The response body is a JSON array of employee objects:

```
[{"id": 1, "firstName": "Sri", "lastName": "Vinayak", "email": "srivinayak@gmail.com"}, {"id": 2, "firstName": "JaiKaarthikaya", "lastName": "Shannuga", "email": "jaikaarthikaya@gmail.com"}, {"id": 3, "firstName": "Sai", "lastName": "JaiVidhyayaka", "email": "saijavidhyayaka@gmail.com"}, {"id": 4, "firstName": "JaiSiv", "lastName": "Vinayaka", "email": "jaisivvinayaka@gmail.com"}]
```

## POST http://localhost:8888/api/employees/createEmployee

The screenshot shows the Postman interface with a POST request to `http://localhost:8888/api/employees/createEmployee`. The request body contains the following JSON:

```

1 {
2   "firstName": "JayVinay",
3   "lastName": "Srinivasa",
4   "email": "vinaysrinivas@gmail.com"
5 }
6

```

The response status is 200 OK, time 10 ms, size 511 B.

## GET http://localhost:8888/api/employees

The screenshot shows the Postman interface with a GET request to `http://localhost:8888/api/employees`. The response body is a list of employees:

```

1 [
2   {
3     "id": 1,
4     "firstName": "Sai",
5     "lastName": "Vinayak",
6     "email": "sairajvinayak@gmail.com"
7   },
8   {
9     "id": 2,
10    "firstName": "Jaikarthika",
11    "lastName": "Shanmuga",
12    "email": "jaikarthikashanmuga@gmail.com"
13  },
14  {
15    "id": 3,
16    "firstName": "Sri",
17    "lastName": "JaiVinayaka",
18    "email": "srijaivinayaka@gmail.com"
19  },
20  {
21    "id": 4,
22    "firstName": "Jaishi",
23    "lastName": "Vinayaka",
24    "email": "jaishivinayaka@gmail.com"
25  },
26  {
27    "id": 5,
28    "firstName": "Jay",
29    "lastName": "Srinivasa",
30    "email": "vinaysrinivas@gmail.com"
31  }
32 ]

```

The response status is 200 OK, time 14 ms, size 601 B.

## Final Result of Get Employees

```
[  
  {  
    "id": 1,  
    "firstName": "Sri",  
    "lastName": "Vinayak",  
    "email": "srivinayak@gmail.com"  
  },  
  {  
    "id": 2,  
    "firstName": "JaiKarthikeya",  
    "lastName": "Shanmuga",  
    "email": "jaikarthikeya@gmail.com"  
  },  
  {  
    "id": 3,  
    "firstName": "Sri",  
    "lastName": "JaiVinayaka",  
    "email": "srijaivinayaka@gmail.com"  
  },  
  {  
    "id": 4,  
    "firstName": "JaiSri",  
    "lastName": "Vinayaka",  
    "email": "jaisrivenayaka@gmail.com"  
  },  
  {  
    "id": 6,  
    "firstName": "JayVinay",  
    "lastName": "Srinivasa",  
    "email": "vinaysrinivasa@gmail.com"  
  }]  
]
```