

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

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, time 21 ms, size 349 B. The response body is a JSON array:

```
[{"id": 1, "firstname": "ShriVinayak", "lastname": "Baba", "email": "s.vinay.888@gmail.com"}, {"id": 2, "firstname": "ShriVinayaka", "lastname": "BabaShivaRam", "email": "s.vinayaka.888@gmail.com"}]
```

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

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, time 267 ms, size 349 B. The response body is a JSON array:

```
[{"id": 1, "firstname": "ShriVinayak", "lastname": "Baba", "email": "s.vinay.888@gmail.com"}, {"id": 2, "firstname": "ShriVinayaka", "lastname": "BabaShivaRam", "email": "s.vinayaka.888@gmail.com"}]
```

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

The screenshot shows the Postman interface with a successful GET request to `http://localhost:8888/api/employees/search/ShriVinayak`. The response body is a JSON array containing one employee object:

```
[{"id": 1, "firstName": "ShriVinayak", "lastName": "Baba", "email": "s.vinayak.888@gmail.com"}]
```

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

The screenshot shows the Postman interface with a successful POST request to `http://localhost:8888/api/employees`. The request body is a JSON object:

```
{"id": 3, "firstName": "JaiSri", "lastName": "Ram", "email": "jaisiran.777@gmail.com"}
```

The response status is 200 OK.

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

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, and Explore, along with a search bar labeled "Search Postman". On the right side of the header, there are buttons for "Sign In" and "Create Account". The main area has tabs for History, New, Import, and a selected item "POST http://localhost:8888/api/employees". Below this, there is a detailed view of the POST request:

- Method:** POST
- URL:** http://localhost:8888/api/employees
- Params:** None
- Authorization:** None
- Headers (1):** Content-Type: application/json
- Body:** (raw) JSON

```
1
2 [
3   {
4     "id": 3,
5     "firstName": "JaiSri",
6     "lastName": "Ram",
7     "email": "jaisiran.777@gmail.com"
8   }
9 ]
```
- Pre-request Script:** None
- Tests:** None
- Settings:** None

On the right side of the request details, there are buttons for "Send", "Save", "Cookies", and "Beautify". Below the request details, the status is shown as "Status: 200 OK Time: 18 ms Size: 262 B Save Response". At the bottom of the interface, there is a toolbar with various icons for file operations and a status bar showing the date and time.

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

The screenshot shows the Postman application interface, identical to the previous one but with a different request type. The top navigation bar, history, and request details are the same. The main difference is in the request method and URL:

- Method:** GET
- URL:** http://localhost:8888/api/employees

The rest of the interface, including headers, body, and results, is identical to the POST request screenshot above.

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

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8888/api/employees`. The response body is a JSON array of two employee objects:

```
[{"id": 1, "firstname": "SriVinayak", "lastname": "Bala", "email": "s.vinay.888@gmail.com"}, {"id": 2, "firstname": "Shrivinaya", "lastname": "Balashivhare", "email": "s.vinayaka.888@gmail.com"}]
```

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

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8888/api/employees/search/JaiSri`. The response body is a JSON array of two employee objects:

```
[{"id": 1, "firstname": "SriVinayak", "lastname": "Bala", "email": "s.vinay.888@gmail.com"}, {"id": 3, "firstname": "Shrivinaya", "lastname": "Balashivhare", "email": "s.vinayaka.888@gmail.com"}]
```

GET <http://localhost:8888/api/employees/search/JaiSri>

The screenshot shows the Postman interface with a successful API call. The URL in the header is `http://localhost:8888/api/employees/search/JaiSri`. The response body is a JSON array containing one employee record:

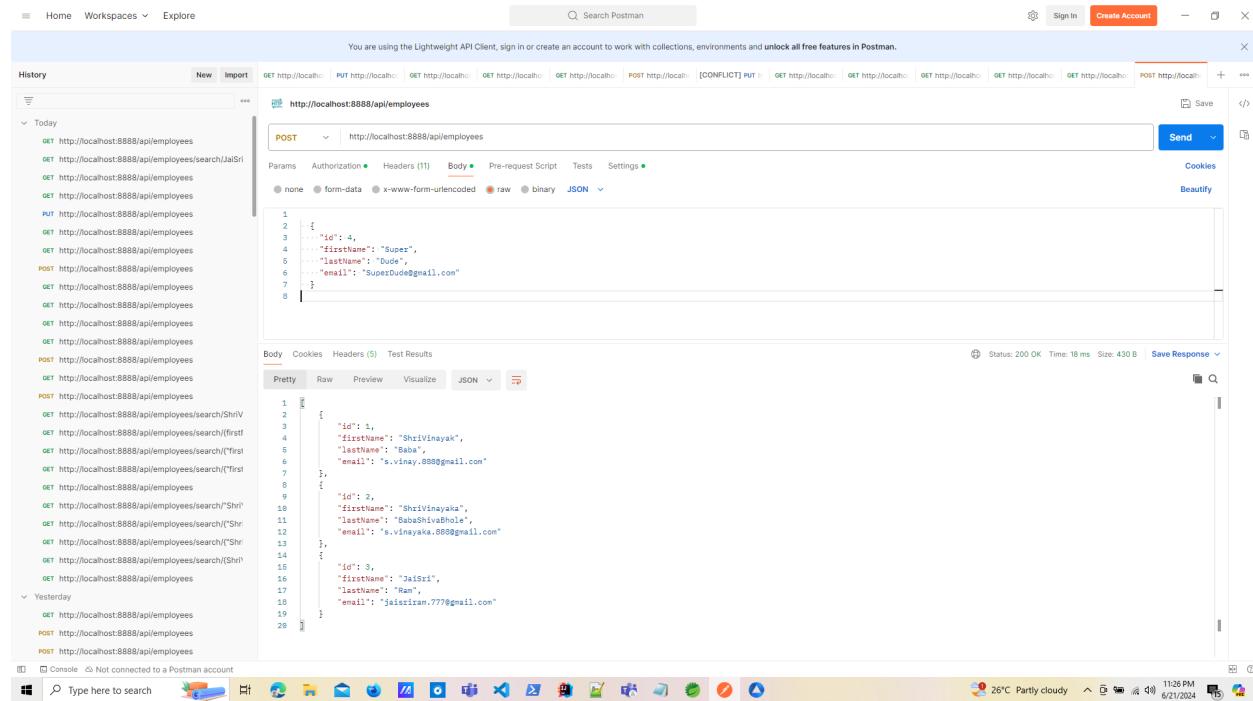
```
[{"id": 3, "firstname": "JaiSri", "lastname": "Ram", "email": "jaisriram.777@gmail.com"}]
```

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

The screenshot shows the Postman interface with a successful API call. The URL in the header is `http://localhost:8888/api/employees`. The response body is a JSON array containing three employee records:

```
[{"id": 1, "firstname": "Shrivinayak", "lastname": "Bala", "email": "s.vinay.888@gmail.com"}, {"id": 2, "firstname": "Shrivinayaka", "lastname": "BabaShivaBhole", "email": "s.vinayaka.888@gmail.com"}, {"id": 3, "firstname": "JaiSri", "lastname": "Ram", "email": "jaisriram.777@gmail.com"}]
```

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

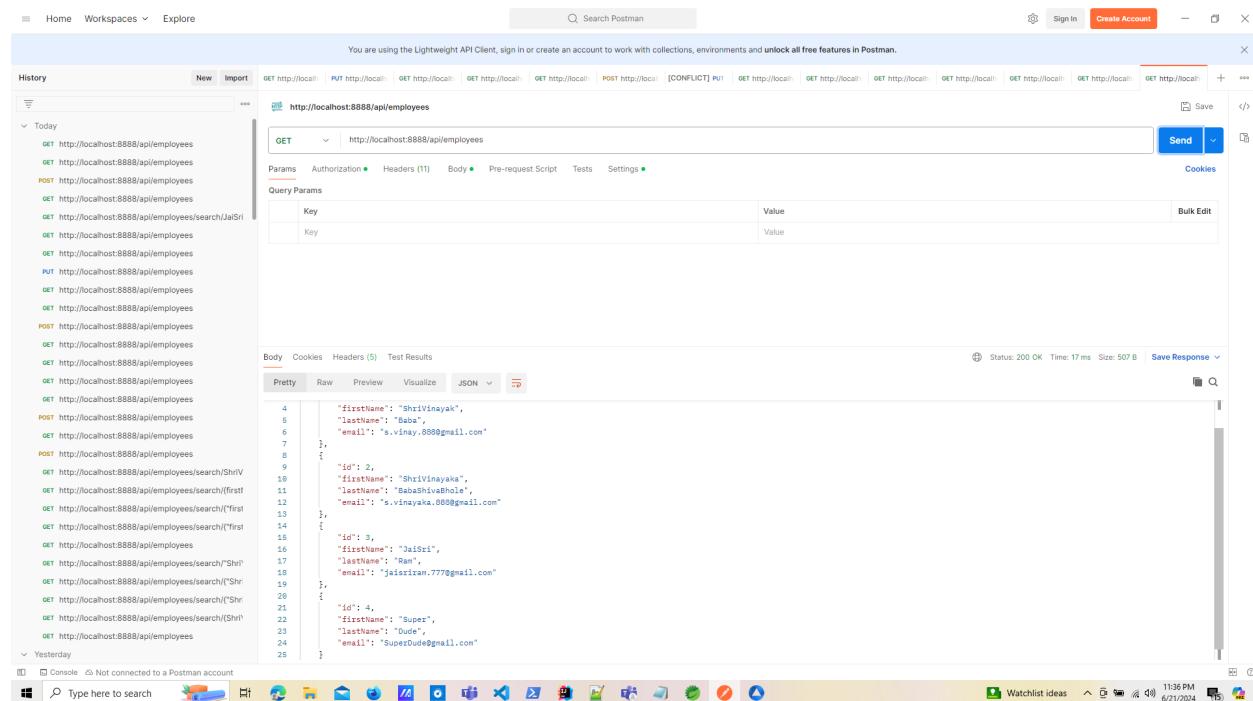


The screenshot shows the Postman interface with a POST request to `http://localhost:8888/api/employees`. The request body is a JSON object:

```
1
2   {
3     "id": 4,
4     "firstname": "Super",
5     "lastname": "Dude",
6     "email": "SuperDude@gmail.com"
7   }
```

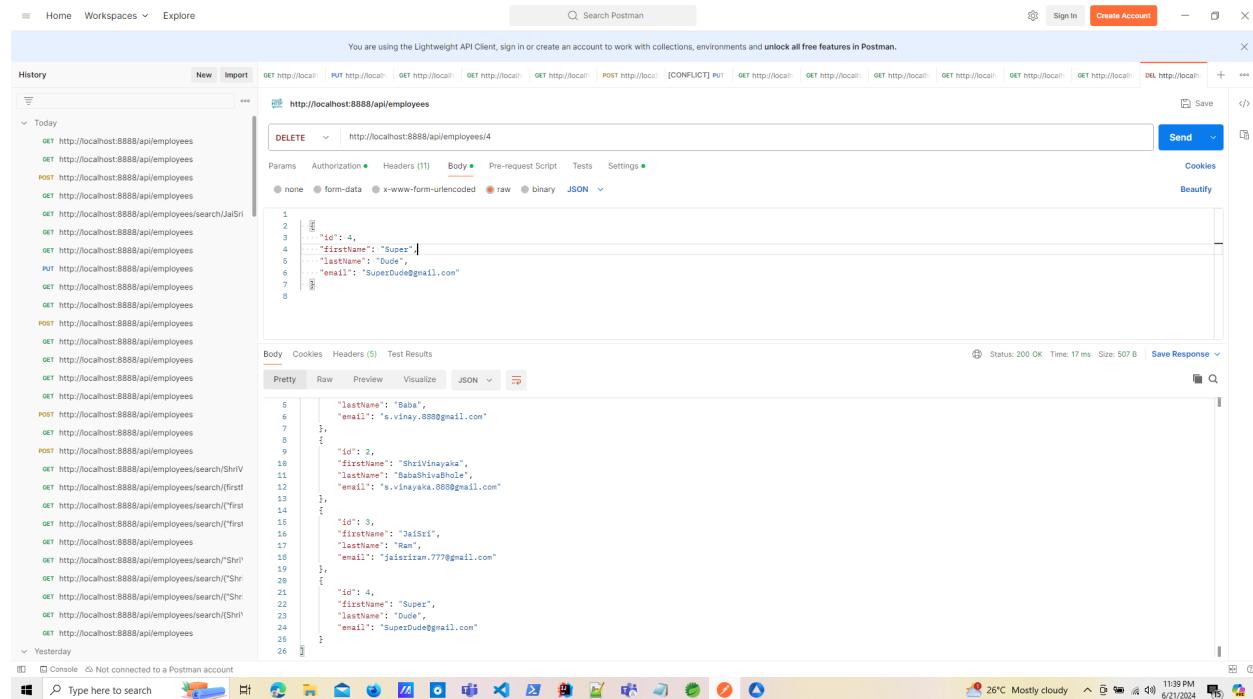
The response status is 200 OK.

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



The screenshot shows the Postman interface with a GET request to `http://localhost:8888/api/employees`. The response status is 200 OK.

DELETE http://localhost:8888/api/employees/4

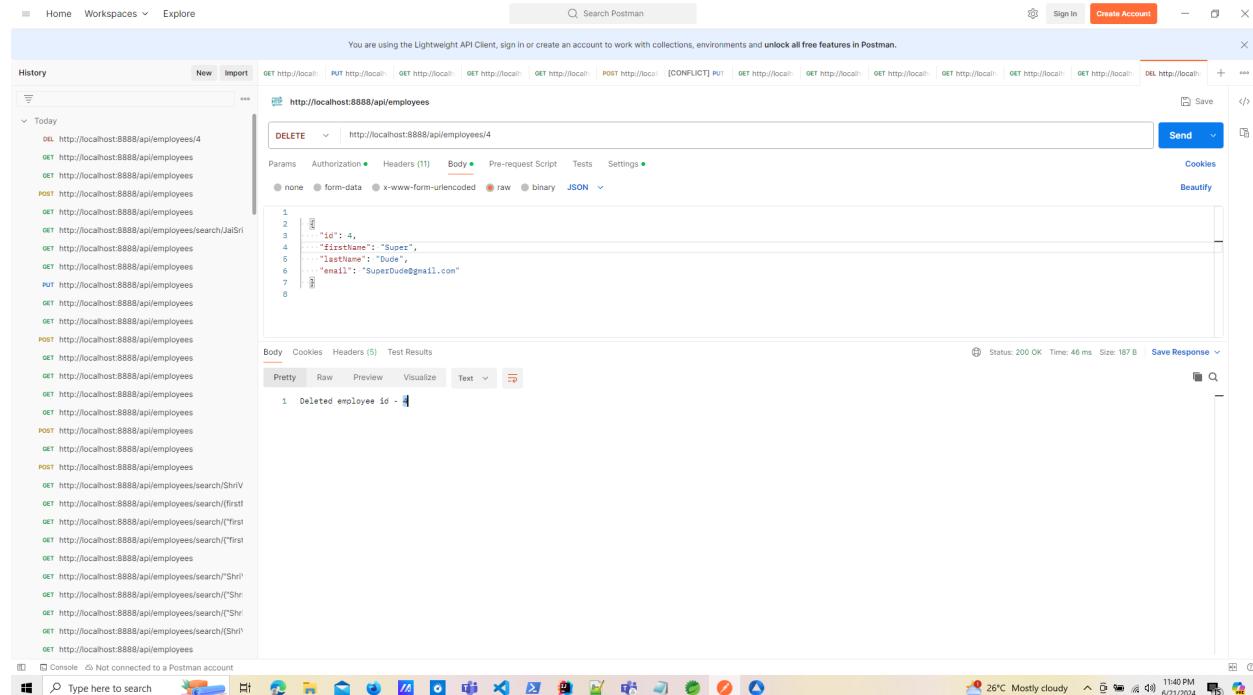


The screenshot shows the Postman interface with a DELETE request to `http://localhost:8888/api/employees/4`. The request body is a JSON object:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
```

The response status is 200 OK.

DELETE http://localhost:8888/api/employees/4

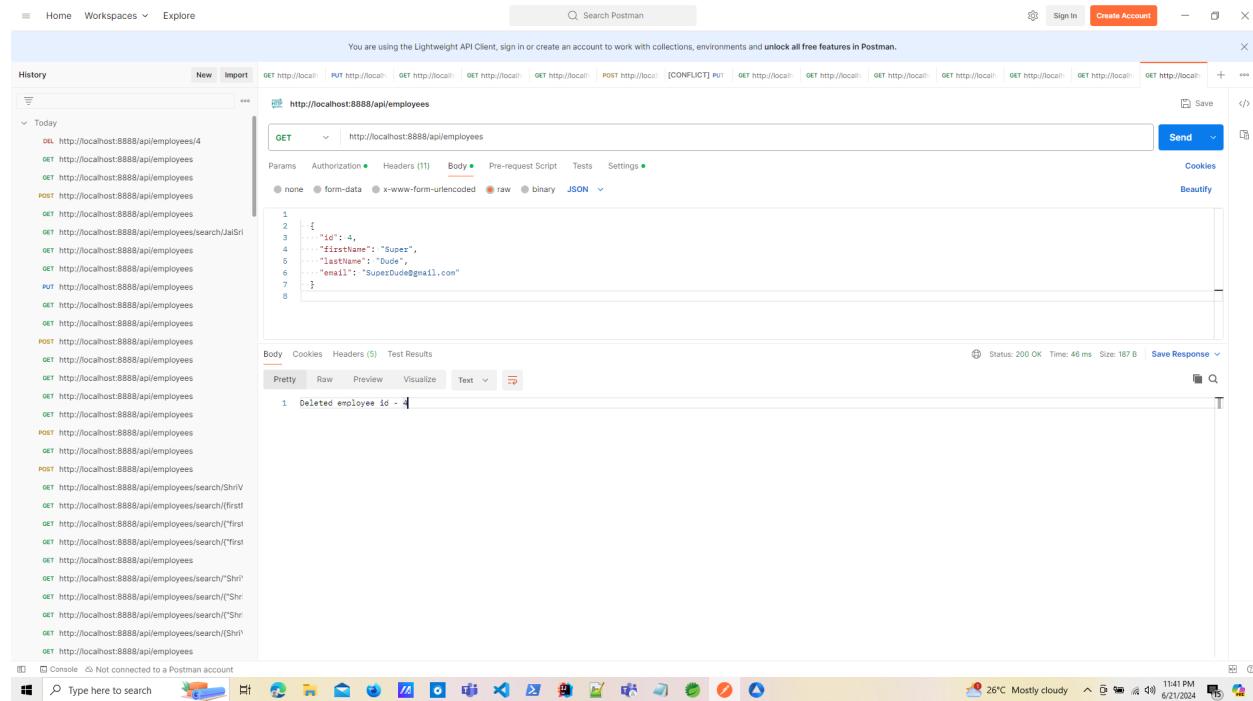


The screenshot shows the Postman interface with a DELETE request to `http://localhost:8888/api/employees/4`. The request body is a JSON object:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
```

The response status is 200 OK, and the response body contains the message: "Deleted employee id - 4".

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



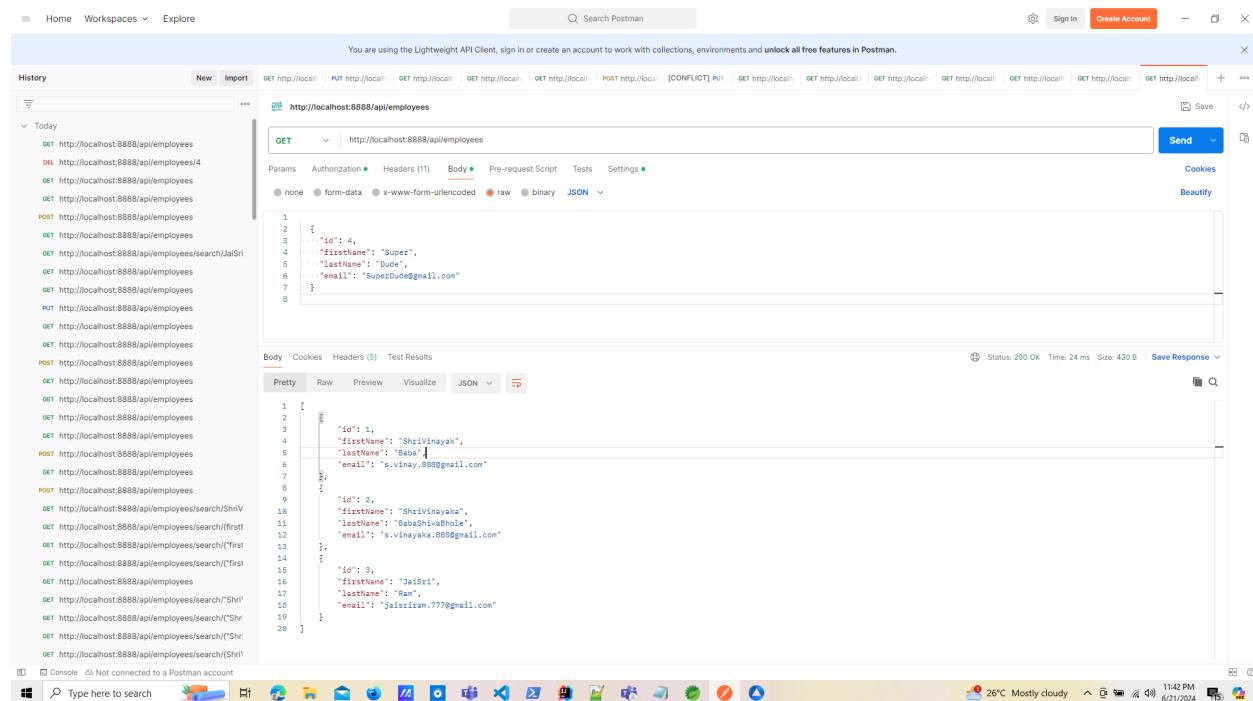
The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for Home, Workspaces, and Explore. A search bar labeled "Search Postman" is located at the top right. The main area has a "History" sidebar on the left listing various API requests. The main workspace shows a "GET" request to "http://localhost:8888/api/employees". The "Body" tab is selected, displaying a JSON object:

```
1
2
3
4
5
6
7
8
```

The response status is 200 OK, with a time of 46 ms and a size of 187 B. The response body is shown in "Pretty" format:

```
1 Deleted employee id : 4
```

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



The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for Home, Workspaces, and Explore. A search bar labeled "Search Postman" is located at the top right. The main area has a "History" sidebar on the left listing various API requests. The main workspace shows a "GET" request to "http://localhost:8888/api/employees". The "Body" tab is selected, displaying a JSON array:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

The response status is 200 OK, with a time of 24 ms and a size of 430 B. The response body is shown in "Pretty" format:

```
1 [
2   {
3     "id": 1,
4     "firstName": "ShriVinayak",
5     "lastName": "Baba",
6     "email": "s.vinay.888@gmail.com"
7   },
8   {
9     "id": 2,
10    "firstName": "ShriVinayaka",
11    "lastName": "BabaShivaBhole",
12    "email": "s.vinayaka.888@gmail.com"
13  },
14  {
15    "id": 3,
16    "firstName": "JaIsri",
17    "lastName": "Ram",
18    "email": "jaIsriram.777@gmail.com"
19  }
20 ]
```

PUT http://localhost:8888/api/employees

The screenshot shows the Postman interface with a PUT request to `http://localhost:8888/api/employees`. The request body is a JSON array with three elements:

```
[{"id": 5, "firstName": "Super", "lastName": "Dude", "email": "SuperDude@gmail.com"}, {"id": 3, "firstName": "ShriVinayak", "lastName": "Rata", "email": "s.vinay.080@gmail.com"}, {"id": 2, "firstName": "ShriVinayaka", "lastName": "BabaShivaRohit", "email": "s.vinayaka.090@gmail.com"}]
```

The response status is 200 OK, Time: 24 ms, Size: 430 B.

PUT http://localhost:8888/api/employees

The screenshot shows the Postman interface with a PUT request to `http://localhost:8888/api/employees`. The request body is a JSON array with three elements:

```
[{"id": 6, "firstName": "Super", "lastName": "Dude", "email": "SuperDude@gmail.com"}]
```

The response status is 200 OK, Time: 28 ms, Size: 241 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 JSON array of employee objects:

```
[{"id": 5, "firstName": "Super", "lastName": "Dude", "email": "SuperDude@gmail.com"}, {"id": 2, "firstName": "Shrivinayaka", "lastName": "BabaShivaBhole", "email": "s.vinayaka.888@gmail.com"}, {"id": 3, "firstName": "JaiSri", "lastName": "Ran", "email": "jaisiran.777@gmail.com"}, {"id": 6, "firstName": "Super", "lastName": "Dude", "email": "SuperDude@gmail.com"}]
```

PUT http://localhost:8888/api/employees

The screenshot shows the Postman interface with a PUT request to `http://localhost:8888/api/employees`. The response body is a JSON array of employee objects, identical to the GET response:

```
[{"id": 5, "firstName": "Super", "lastName": "Dude", "email": "SuperDude@gmail.com"}, {"id": 2, "firstName": "Shrivinayaka", "lastName": "BabaShivaBhole", "email": "s.vinayaka.888@gmail.com"}, {"id": 3, "firstName": "JaiSri", "lastName": "Ran", "email": "jaisiran.777@gmail.com"}, {"id": 6, "firstName": "Super", "lastName": "Dude", "email": "SuperDude@gmail.com"}]
```

PUT http://localhost:8888/api/employees

The screenshot shows the Postman interface with a PUT request to `http://localhost:8888/api/employees`. The request body is set to JSON with the following content:

```
1
2   {
3     "id": 6,
4     "firstname": "SuperRich",
5     "lastname": "Dude",
6     "email": "SuperDude@gmail.com"
7   }
```

The response status is 200 OK with a time of 32 ms and a size of 244 B.

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

The screenshot shows the Postman interface with a GET request to `http://localhost:8888/api/employees`. The request body is set to JSON with the following content:

```
1
2   {
3     "id": 6,
4     "firstname": "SuperRich",
5     "lastname": "Dude",
6     "email": "SuperDude@gmail.com"
7   }
```

The response status is 200 OK with a time of 32 ms and a size of 244 B.

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

The screenshot shows the Postman interface with a successful GET request to `http://localhost:8888/api/employees`. The response body is a JSON array containing four employee objects:

```
[{"id": 1, "firstName": "Shrivinayak", "lastName": "Baba", "email": "s.vinay.888@gmail.com"}, {"id": 2, "firstName": "ShriVinayaka", "lastName": "BabaShivaBhole", "email": "s.vinayaka.088@gmail.com"}, {"id": 3, "firstName": "JaiSri", "lastName": "Ram", "email": "jaisriram.777@gmail.com"}, {"id": 4, "firstName": "SuperRich", "lastName": "Dude", "email": "SuperDude@gmail.com"}]
```

DELETE http://localhost:8888/api/employees/6

The screenshot shows the Postman interface with a successful DELETE request to `http://localhost:8888/api/employees/6`. The response body indicates the deletion of employee with id 6:

```
Deleted employee id ~ 6
```

GET http://localhost:8888/api/employees/6

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8888/api/employees/6`. The response body is a JSON array:

```
[{"id": 1, "firstname": "ShriVinayak", "lastname": "Baba", "email": "s.vinay.888@gmail.com"}, {"id": 2, "firstname": "ShriVinayaka", "lastname": "BabaShivaBhole", "email": "s.vinayaka.888@gmail.com"}, {"id": 3, "firstname": "JaiSri", "lastname": "Ram", "email": "jaisriram.777@gmail.com"}]
```

The screenshot shows the H2 Console interface with the results of a SQL query:

```
SELECT * FROM EMPLOYEE
```

ID	EMAIL	FIRST_NAME	LAST_NAME
1	s.vinay.888@gmail.com	ShriVinayak	Baba
2	s.vinayaka.888@gmail.com	ShriVinayaka	BabaShivaBhole
3	jaisriram.777@gmail.com	JaiSri	Ram

(3 rows, 2 ms)

STS-WorkSpace - employeeservice[src/main/java/com/gl/employeeservice/controller/EmployeesRestController.java - Spring Tool Suite 4

```

public Employee getEmployee(@PathVariable int employeeId) {
    Employee theEmployee = employeeService.findById(employeeId);
    if (theEmployee == null) {
        throw new EmployeeNotFoundException("Employee id not found - " + employeeId);
    }
    return theEmployee;
}

// add mapping for POST /employees - add new employee
@PostMapping("/employees")
public Employee addEmployee(@RequestBody Employee theEmployee) {
    // set id to 0 to force a save of new item
    theEmployee.setId(0);
    employeeService.save(theEmployee);
    return theEmployee;
}

// add mapping for PUT /employees - update existing employee
@PutMapping("/employees")

```

Console > employeeserviceApplication [Java Application] [pid: 15536]



(v3.2.6)

Spring Boot ::

```

2024-06-21T22:46:25.616+05:30 INFO 15536 --- [ restartedMain] c.g.e.EmployeeServiceApplication : Starting EmployeeServiceApplication using Java 17.0.10 with PID 15536 in D:\STS-WorkSpace\employeeservice\target\classes started by asus in D:\STS-WorkSpace\employeeservice
2024-06-21T22:46:25.616+05:30 INFO 15536 --- [ restartedMain] c.g.e.EmployeeServiceApplication : No active profile set, falling back to 1 default profile: "default"
2024-06-21T22:46:25.616+05:30 INFO 15536 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Started Tomcat Web Server in 0.019 seconds (�)
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 72 ms. Found 3 JPA repository interfaces.
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data 3PA repositories in DEFAULT mode.
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 72 ms. Found 3 JPA repository interfaces.
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.apache.catalina.core.StandardService$ServiceLifecycleListener : Starting service [Tomcat]
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Initiating embedded web application context
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] w.s.c.WebXmlServletServerWebApplicationContext : Root WebApplicationContext initialization completed in 1759 ms
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Added connection conn0: url=jdbc:h2://localhost/~/testdb user=SA
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000294: Processing PersistenceUnitInfo [name: default]
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.hibernate.Version : HHH000412: Hibernate ORM core version 6.4.8.Final
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.hibernate.cfg.Environment : HHH000414: SessionFactory configuration loaded from resource [hibernate.cfg.xml]
2024-06-21T22:46:25.676+05:30 INFO 15536 --- [ restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup. Ignoring JDL class transformer
2024-06-21T22:46:28.140+05:30 WARN 15536 --- [ restartedMain] o.hibernate.orm.deprecation : HHH0000025: H2Dialect does not need to be specified explicitly using 'hibernate.dialect' (remove the property setting and it will be selected by default)
2024-06-21T22:46:29.024+05:30 INFO 15536 --- [ restartedMain] o.h.e.t.j.p.EntityManagerInitiator : HHH000049: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2024-06-21T22:46:29.076+05:30 INFO 15536 --- [ restartedMain] o.localContainerEntityManagerFactory : Initialized EntityManagerFactory for persistence unit 'default'

```

12:13 AM 6/22/2024

Select Administrator: WindowsPowerShell

```

Directory: C:\Program Files (x86)\H2\bin

Mode LastWriteTime Length Name
---- ----- ---- 
6/20/2024 10:43 PM 10B h2.bat
6/20/2024 7:38 PM docs
6/20/2024 7:38 PM service
6/20/2024 7:38 PM temp
7/4/2023 11:40 AM 360 h2c.bat
7/4/2023 11:40 AM 676 build.sh
6/20/2024 7:38 PM 168877 Uninstall.exe

PS C:\Program Files (x86)\H2\bin> cd ..\bin\

C:\Program Files (x86)\H2\bin> 12

```

Directory: C:\Program Files (x86)\H2\bin

```

Mode LastWriteTime Length Name
---- ----- ---- 
6/20/2024 7:33 PM org
9/17/2023 3:33 AM 2614932 h2 2.2.224.jar
9/17/2023 3:33 AM 189 h2.bat
9/17/2023 3:33 AM 185 h2c.bat
6/26/2024 10:01 PM 16304 testdb.mv.db

```

PS C:\Program Files (x86)\H2\bin> .\h2.bat

PS C:\Program Files (x86)\H2\bin>

Windows PowerShell

25°C Mostly clear 12:13 AM 6/22/2024