**Exercise 4: Employee Management System - Implementing CRUD Operations**

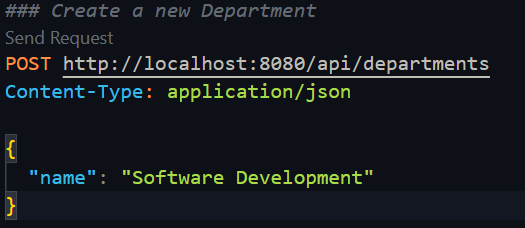
* **Basic CRUD Operations**:
* ***JpaRepository Methods****:* Utilize *JpaRepository's* built-in methods such as save, findAll, findById, deleteById to perform CRUD operations on Employee and Department entities.
* ***RESTful Endpoints****:* Implement RESTful endpoints using *EmployeeController* and *DepartmentController* to expose CRUD functionalities via HTTP methods (GET, POST, PUT, DELETE).
* ***Response Handling****:* Wrap responses in a custom *ApiResponse* object to standardize and include messages, data, and status.
* ***REST Client Interaction***: Test CRUD operations through a REST client by making requests to the defined endpoints and observing the structured responses.

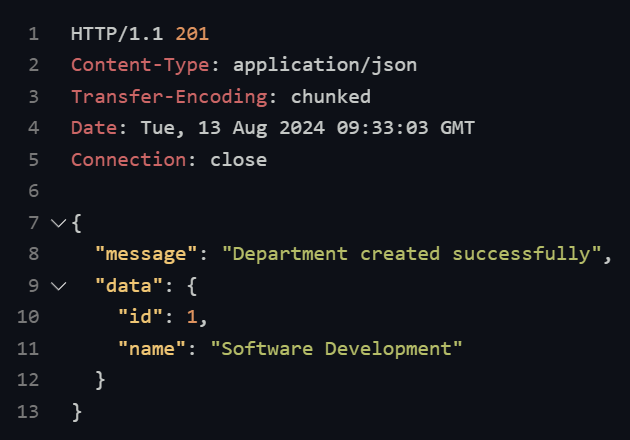
GitHub link of the code – [link](https://github.com/Hyperstrom/Aniket-Pal_5017587/tree/main/WEEK-3/Exercise-4)

* **Output of the program**:

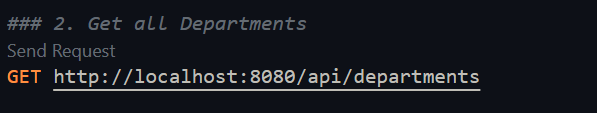
To add and update employees and departments using a **REST client.** A REST client is a tool or application used to interact with RESTful web services by sending HTTP requests and receiving responses. It allows developers to test and debug APIs by performing CRUD operations (Create, Read, Update, Delete) on resources over the web.

1. Department operations:

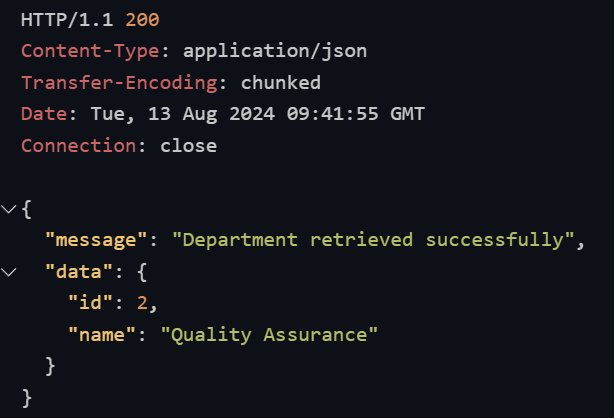
* **Add department:** Use the **POST** method and send a request to the endpoint ***http://localhost:8080/api/departments.***



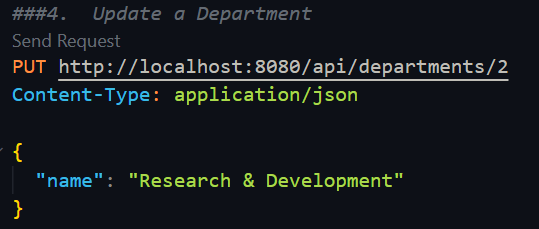
This means the department is successfully created (This is the Response from the REST CLIENT)

* **Read department:** Use the **GET** method and send a request to the endpoint ***http://localhost:8080/api/departments.***

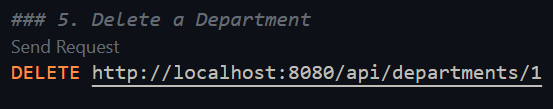
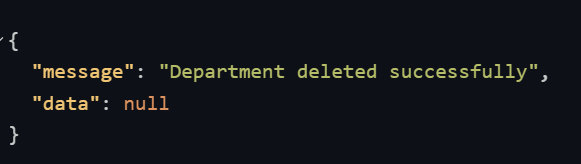
This is the all-department names and ids which it gets from the REST Client response

* **Search department**: Use the **GET** and with the department id number to search that department

This is the Search Response of department

**Update department:** Use the PUT method and send a request to the endpoint along with the department id number which you have to update

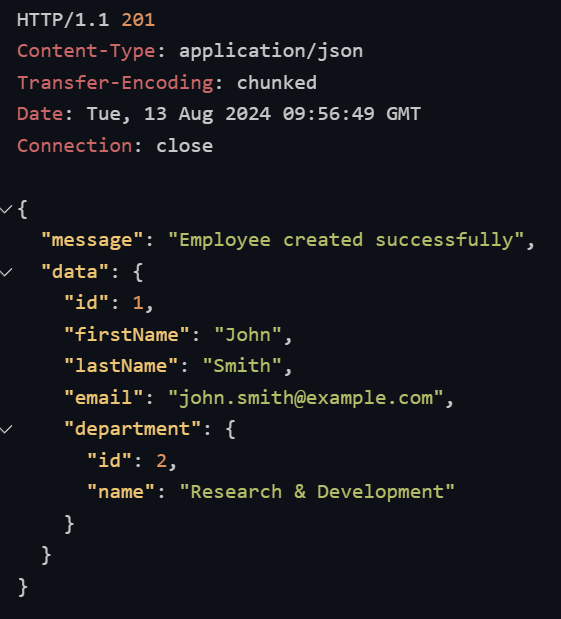
The id no. 2 dept is updated from Quality Assurance to Research & Development.

**Delete department:** Use the **DELETE** method and send a request to the endpoint along with the department id number which you have to delete.

2. Employee operations:

* **Add Employee:** Use the **POST** method to send a request to ***http://localhost:8080/api/employees.***

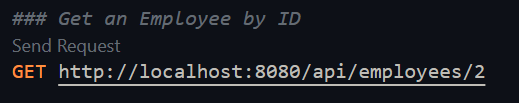
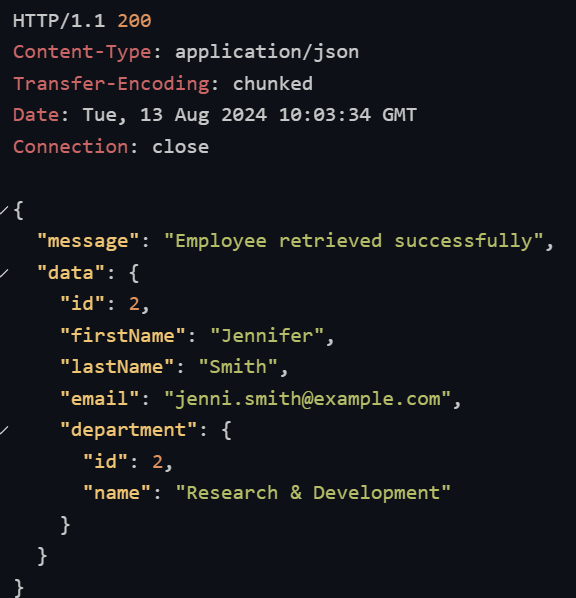
add Employees in this way

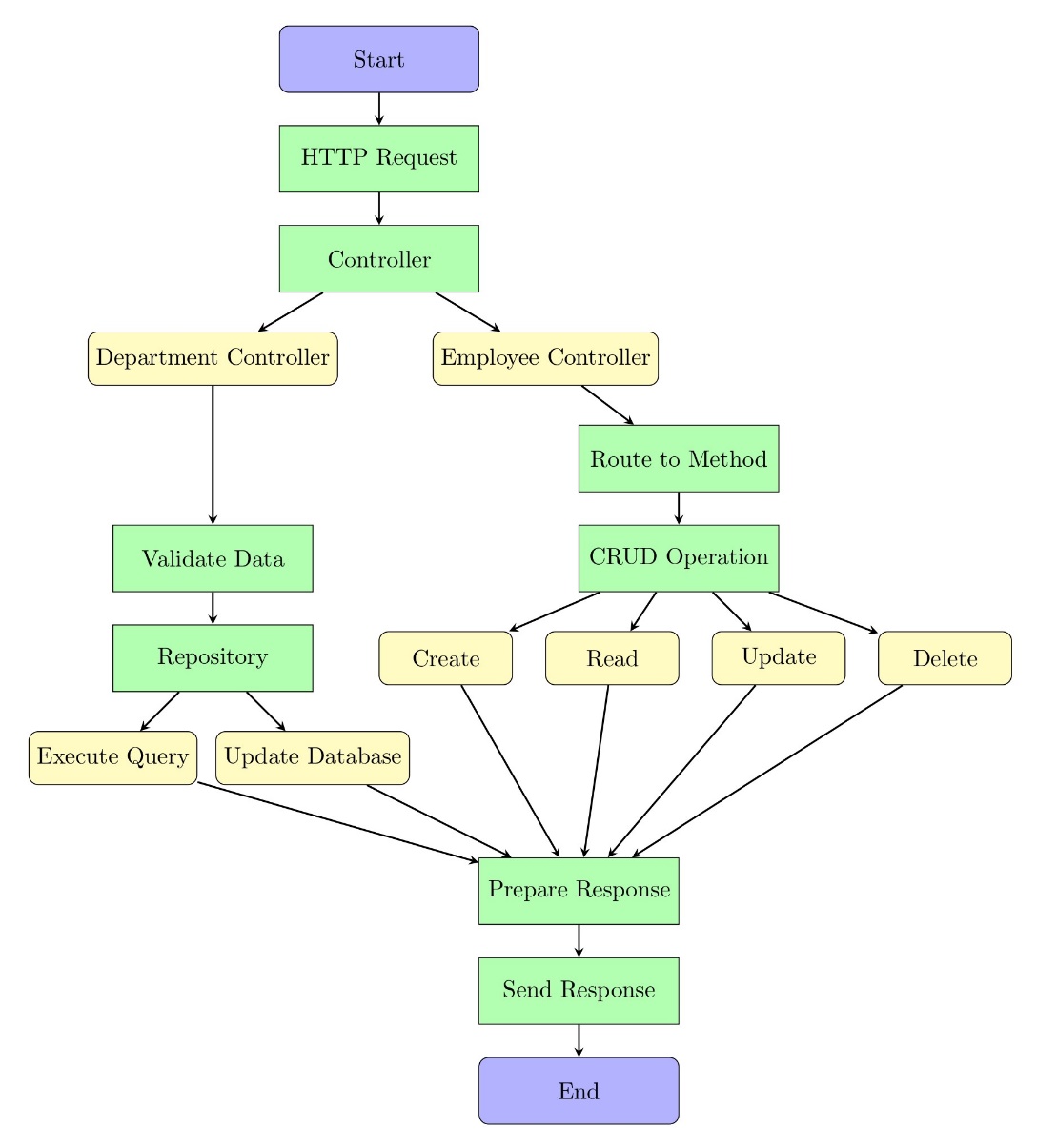
****

This is the response from the REST Client. That means the Employee is successfully added

* **Read Employee:** Use the **GET** method and send a request to the endpoint ***http://localhost:8080/api/employees***

This is the response from the REST Client this is the all-employee list. Which are in the database

* **Search Employee:** Use the **GET** and with the Employee id number to search that Employee
* **Delete Employee:** Use the **DELETE** method and send a request to the endpoint along with the Employee id number which you have to delete
* **Flow Chart:**



* The repository interacts with the database, executes the necessary query, and prepares a response for the client.
* The program begins by receiving an HTTP request routed to the appropriate controller.
* The controller validates the input and routes the request to the corresponding CRUD method.