**Technical Documentation for Sales Employee Information System**

**Description**: This project implements an information system used to manage sales employees and determine their commission. It allows the following functions:

1. Create New Sales Employee
2. Update Sales Employee
3. Get Sales Employee
4. Calculate Sales Commission of a Sales Employee

**Scope and Assumptions**:

1. This system will be deployed on a local server. In this case, it can run on a machine for this example.
2. No frontend implementation and deliverable is a backend project that exposes web services that will perform the functions mentioned in the description.
3. Postman will be used to test how the system will perform.
4. Security is not in scope.
5. Searching for Update is using ID only.
6. Validation is not in scope

Audience: The system will be used by Sales Employee Management Team A frontend system may access the web services but for this implementation, we will use postman to check for the service.

**Tech Stack:**

List the technologies used in the project (e.g., Spring Boot, Spring MVC, Spring Data JPA).

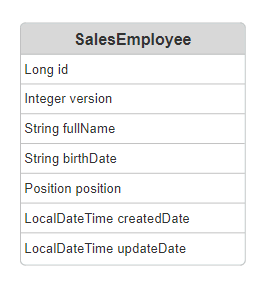
1. Spring Framework
2. Spring Boot
3. H2 Database
4. Maven
5. Lombok

**Component Breakdown**

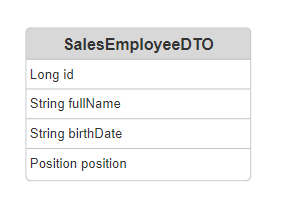
* SalesEmployeeController
* SalesEmployeeService
* SalesEmployeeRepository
* BootstrapData - used to create initial records

**Data Model**

System will use one table - Sales Employee



**DTO Model**



**API Reference:**

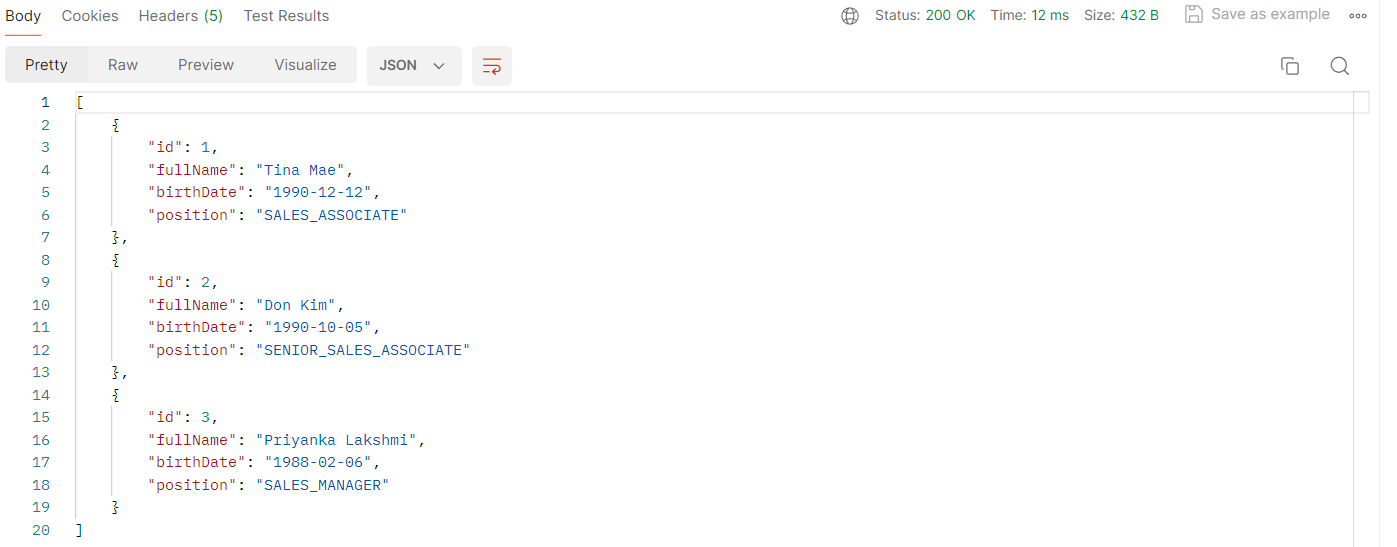
1. List all Sales employees

Description: Allows user to view all Sales Employee Records

Location: /api/v1/salesEmployee

HTTP Method: Get

Positive Response Sample



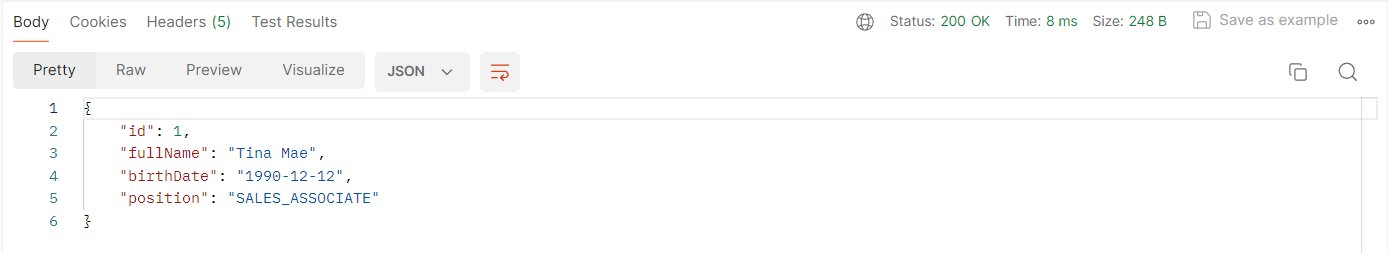
1. Get Sales Employee By Id

Description: Allows user to view a Sales Employee Record using Id

Location: /api/v1/salesEmployee/{salesEmployeeId}

HTTP Method: Get

Positive Result:



1. Create New Sales Employee

Description: Allows user to create New Sales Employee provided with Full Name, BirthDate and Position

Location: /api/v1/reservation/

HTTP Method: Post

Sample JSON:

{

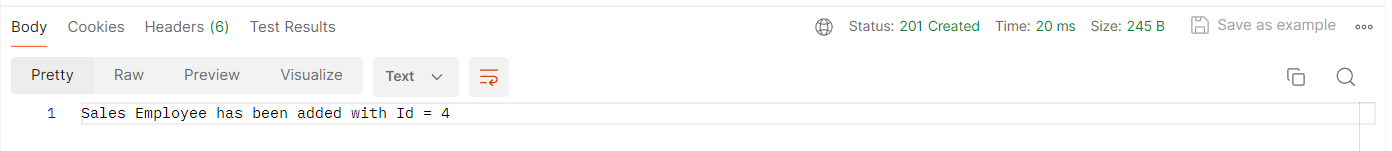
"fullName": "Mary Knoll",

"birthDate": "1990-11-12",

"position": "SALES\_ASSOCIATE"

}

Positive Response:



1. Update Sales Employee

Description: User can update sales Employee Record provided with Full Name, BirthDate and Position using Id

Location: /api/v1/salesEmployee/{salesEmployeeId}

HTTP Method: Put

Sample Json Requests:

{

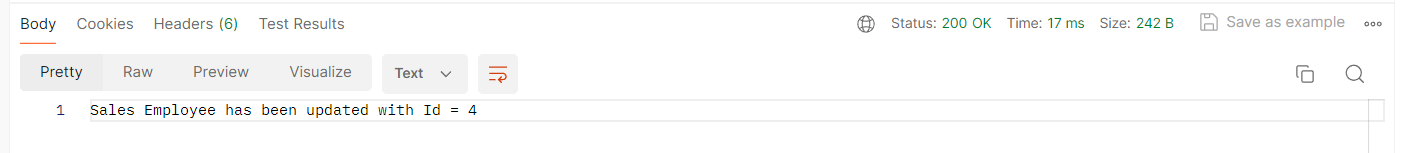
"fullName": "Mary Grey",

"birthDate": "1990-11-12",

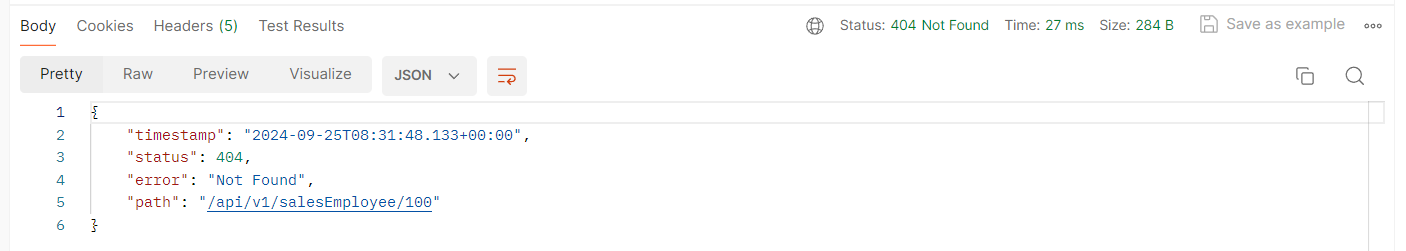
"position": "SALES\_ASSOCIATE"

}

Positive Response



Negative Response



1. Get Commission

Description: This service allows user to get a sales employee’s commission by providing sales amount and will determine the commission based on the following computations:

* Sales Associate: There would be a 5% commission rate on the total sales of a sales associate.
* Senior Sales Associate: There would be a 7% commission rate on total sales of a senior sales associate and an additional 2% on the excess of the target threshold of 50,000. (Sample: 60,000 - total sales. Commission = (7% \* 60,000) + ((60,000 - 50,000) \* 2%)
* Sales Manager: 6% commission rate on team total sales and an additional fixed rate bonus of 1,000.0 if the team target threshold of 200,000 has been achieved (Sample: 210,000 - total Sales. Commission = (6% \* 210,000) + 1,000)

Location: /api/v1/salesEmployee/{salesEmployeeId}/commission

HTTP Method: Put

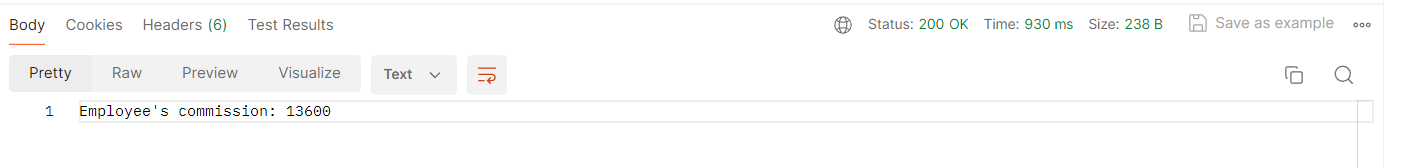
Sample Json Requests:

{

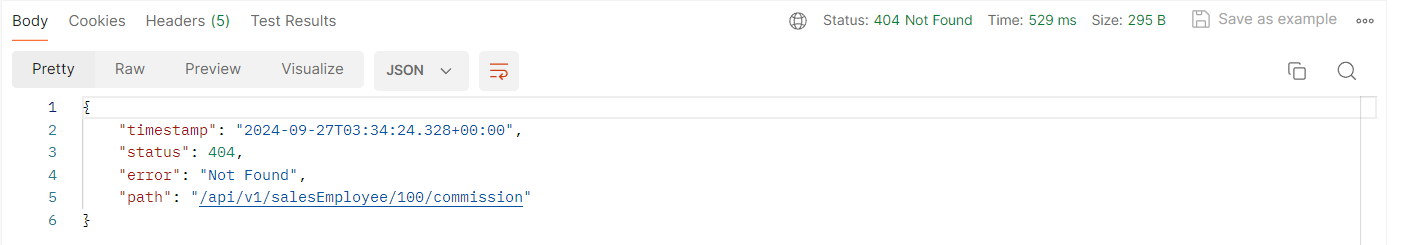
"salesAmount": 210000

}

Positive Response



Negative Response



**Error Handling:**

1. Sales Employee is not found, NotFoundException will be thrown