**Employee Management System - Creating Projections**

In this exercise, **projections** implemented to fetch specific data subsets from the **Employee** and **Department** entities. **Projections allow us to retrieve only the necessary fields from the database, improving performance and reducing the amount of data transferred over the network.**

**1. Interface-Based Projection**

**Definition:** Interface-based projections are a way to select specific fields from an entity using an interface. This method is useful when you need to project a subset of data without creating a new class for it.

**Implementation:**

* **Interface Name:** EmployeeNameAndDepartment
* **Purpose:** To fetch the employee name and department name.
* **Method:**
  + String getName();
  + String getDepartmentName();
* **Repository Usage Method:** findEmployeeNameAndDepartment()
* **Query:** Uses JPQL to join Employee and Department and select specific fields.

**2. Class-Based Projection**

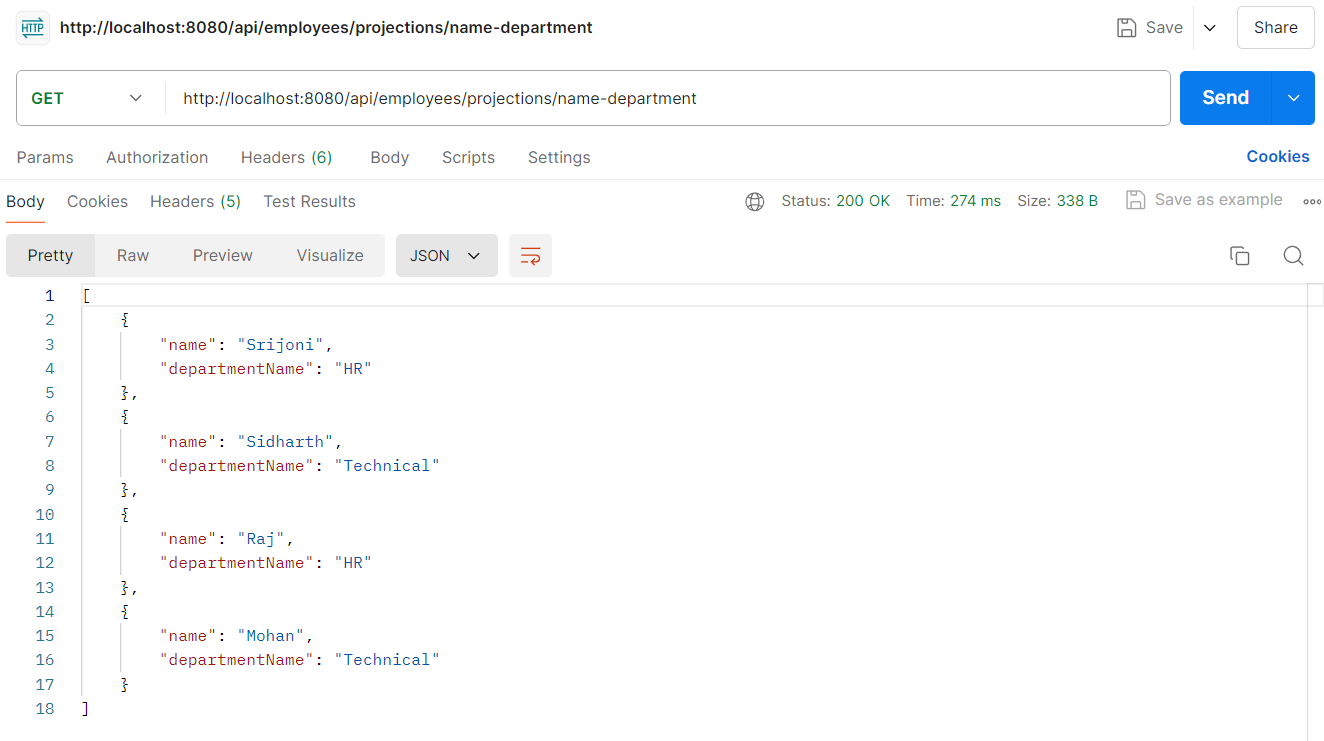
**Definition:** Class-based projections allows to create a class that holds the subset of data that wanted to be fetched. This is useful when needed more control over the projection and want to include logic or additional properties.

**Implementation:**

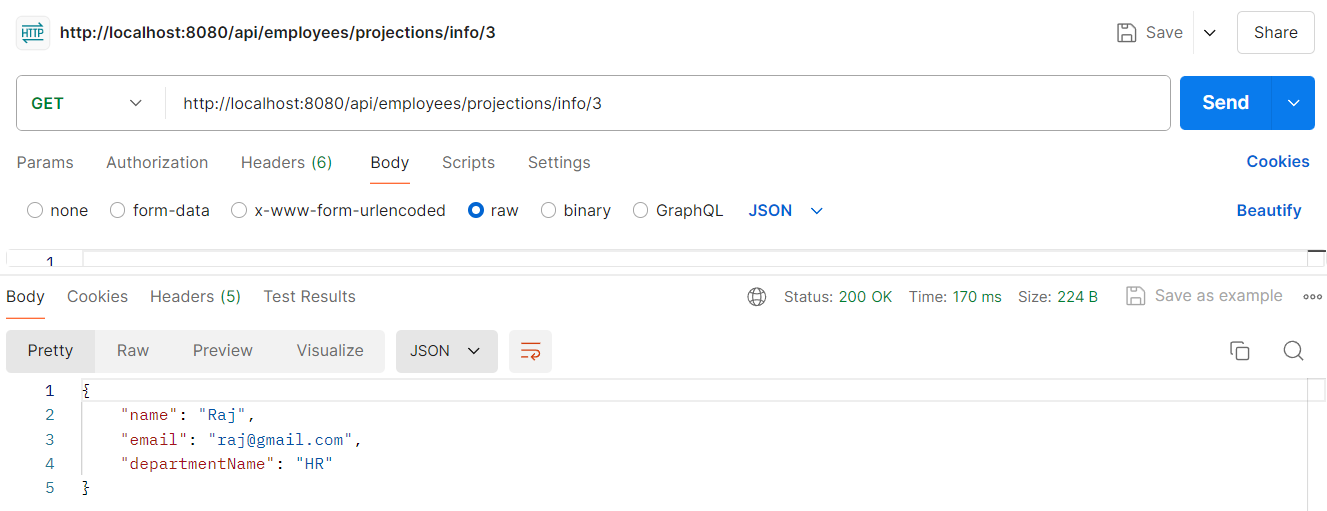
* **Class Name:** EmployeeInfo
* **Purpose:** To fetch detailed employee information, including name, email, and department name.
* **Constructor Expressions:**
  + Fields are populated using a constructor.
  + **Constructor:** EmployeeInfo(String name, String email, String departmentName)
* **Repository Usage Method:** findEmployeeInfoById(@Param("id") Long id)
* **Query:** Uses JPQL with constructor expressions to fetch employee details.

1. Added methods to **EmployeeRepository** for the projections:
   * findEmployeeNameAndDepartment()
   * findEmployeeInfoById(Long id)
2. Integrated into **EmployeeController.**

* **Interface-Based Projection (EmployeeNameAndDepartment):**
  + **Purpose:** Fetch specific fields (employee name and department name) using an interface.
  + **Repository Method:** findEmployeeNameAndDepartment()
  + **Controller Endpoint:** /api/employees/projections/name-department
* **Class-Based Projection (EmployeeInfo):**
  + **Purpose:** Fetch detailed employee information using a class-based projection.
  + **Repository Method:** findEmployeeInfoById(@Param("id") Long id)
  + **Controller Endpoint:** /api/employees/projections/info/{id}

**GET Employee Name and Department Name using interface-based projection:**

**GET Employee info using class-based projection:**

****