# Employee Management System with Spring Boot: Setter Injection and RESTful API

# OVERVIEW

Create a simple Spring Application and inject the literal values by setter injection. So, create a simple class Employee having three attributes ID, Name, and Designation. Create setter methods for these attributes and a simple method to print the details of the employee.

# FUNCTIONAL REQUIREMENTS

* Build an application for getting employee details.
* Have to create a folder named “Controller” inside the springapp in src.
* Inside the Controller create a java file named “ApiController.java”.
* Next have to create another folder inside the springapp named as

“Models”.

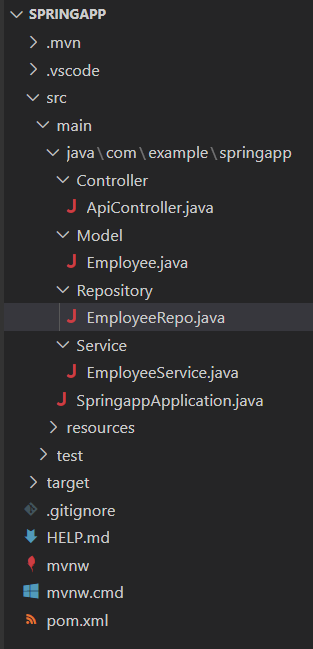
* Inside Models, you should create the java file named “Employee.java”.
* And create Repository folder and inside that create "EmployeeRepo.java".

**Create 3 variables**

* Id - int
* Name- string
* Designation – string

as well as create setters and corresponding variables. Finally, create the "Service" folder, and inside that create the "EmployeeService.java" file.

The project structure looks like this image



**Core Platform**

OpenJDK 11

**API:**

POST - "/" --> true/false

GET - "/{id}" --> List of employee object GET - "/" --> List of employee object

GET - "/employees/groupBy/{attribute}" --> Returns grouped employee data based on the specified attribute.

GET - "/employees/findBy/{attribute}?value={value}" --> Returns employee data filtered by the specified attribute value.

GET -“/employees/salaryRange?minSalary={minSalary}&maxSalary={maxSalary}" --> Returns employee data within the specified salary range.

**Note:**

Copy and paste it into the **application.properties** file

**spring.jpa.hibernate.ddl-auto=update**

**spring.datasource.url=jdbc:mysql://localhost/employee?createDatabaseIfNo tExist=true**

**spring.datasource.username=root spring.datasource.password=examly**

**spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver spring.jpa.show-sql= true**

**spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect**

**API endpoint: 8080**

**Platform Guidelines:**

To run the command use Terminal in the platform.

**Spring Boot:**

To start/run the application **'mvn spring-boot:run'**