**Project : Employee Management System**

# 1 ) Create Spring Boot Project with Spring Initializer

# 2) Maven Dependencies – Code for pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.0.2</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>CrudDemoApplication</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>CrudDemoApplication</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

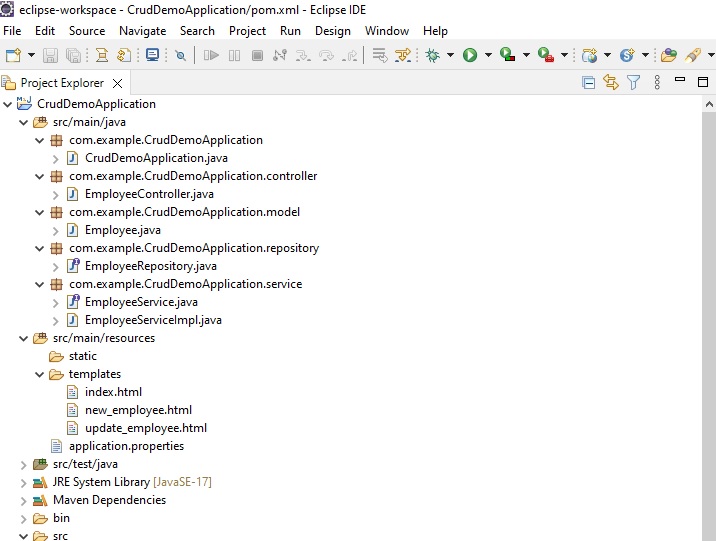
</plugin>

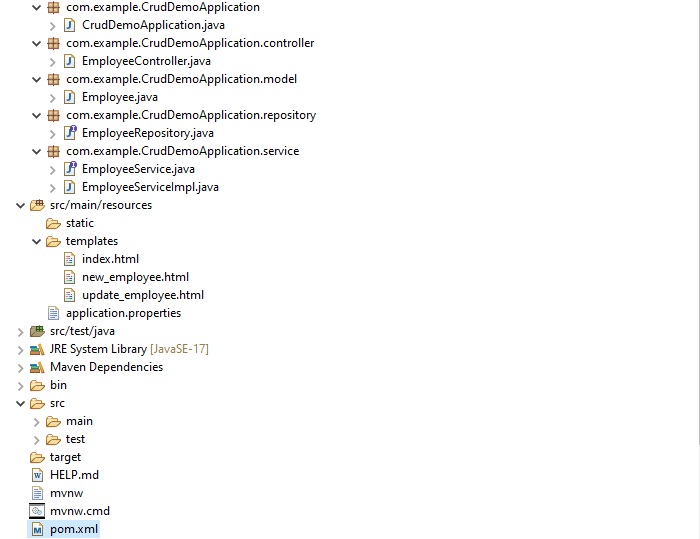
</plugins>

</build>

</project>

# 3) Project Structure – 1





# 4. Database Setup

Create a database with the name "**demo**" in the MySQL database server.

We’ll need to configure MySQL database **URL**, **username**, and **password**so that Spring can establish a connection with the database on startup.

In *application.properties*  following MySQL database configuration to be done:

spring.datasource.url=jdbc:mysql://localhost:3306/demo?useUnicode=true&useJDBCComplaintTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC

server.error.whitelabel.enabled=false

server.port=8093

spring.datasource.username=root

spring.datasource.password=

spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

spring.jpa.hibernate.ddl-auto=update

logging.level.org.hibernate.SQL=DEBUG

logging.level.org.hibernate.type=TRACE

#spring.jpa.open-in-view=false

#spring.thymeleaf.cache=false

api.base.path = http://localhost:8090

# 5) Create JPA Entity - Employee.java

**package** com.example.CrudDemoApplication.model;

**import** jakarta.persistence.Column;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.GeneratedValue;

**import** jakarta.persistence.GenerationType;

**import** jakarta.persistence.Id;

**import** jakarta.persistence.Table;

@Entity

@Table(name="employees")

**public** **class** Employee {

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

**private** **long** id;

@Column(name= "first\_name")

**private** String firstName;

@Column(name="last\_name")

**private** String lastName;

**private** String email;

**public** **long** getId() {

**return** id;

}

**public** **void** setId(**long** id) {

**this**.id = id;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

}

# 6) back-end changes.

## EmployeeRepository.java

**package** com.example.CrudDemoApplication.repository;

**import** org.springframework.stereotype.Repository;

**import** org.springframework.data.jpa.repository.JpaRepository;

**import** com.example.CrudDemoApplication.model.Employee;

@Repository

**public** **interface** EmployeeRepository **extends** JpaRepository<Employee,Long>{

}

# 7) EmployeeService.java

**package** com.example.CrudDemoApplication.service;

**import** java.util.List;

**import** com.example.CrudDemoApplication.model.Employee;

**public** **interface** EmployeeService {

List<Employee>getAllEmployees();

**void** saveEmployee(Employee employee);

Employee getEmployeeById(**long** id);

**void** deleteEmployeeById(**long** id);

}

# 8) EmployeeServiceImpl.java

**package** com.example.CrudDemoApplication.service;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.example.CrudDemoApplication.model.Employee;

**import** com.example.CrudDemoApplication.repository.EmployeeRepository;

**import** java.util.List;

**import** java.util.Optional;

@Service

**public** **class** EmployeeServiceImpl **implements** EmployeeService {

@Autowired

**private** EmployeeRepository employeeRepository;

@Override

**public** List<Employee> getAllEmployees(){

**return** employeeRepository.findAll();

}

@Override

**public** **void** saveEmployee(Employee employee) {

**this**.employeeRepository.save(employee);

}

**public** Employee getEmployeeById(**long** id) {

Optional < Employee > optional = employeeRepository.findById(id);

Employee employee = **null**;

**if** (optional.isPresent()) {

employee = optional.get();

} **else** {

**throw** **new** RuntimeException(" Employee not found for id :: " + id);

}

**return** employee;

}

@Override

**public** **void** deleteEmployeeById(**long** id) {

**this**.employeeRepository.deleteById(id);

}

}

# 9) EmployeeController.java

**package** com.example.CrudDemoApplication.controller;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.PostMapping;

**import** com.example.CrudDemoApplication.model.Employee;

**import** com.example.CrudDemoApplication.service.EmployeeService;

**import** org.springframework.web.bind.annotation.PathVariable;

@Controller

**public** **class** EmployeeController {

@Autowired

**private** EmployeeService employeeService;

@GetMapping("/")

**public** String viewHomePage(Model model) {

model.addAttribute("listEmployees" ,employeeService.getAllEmployees());

**return** "index";

}

@GetMapping("/showNewEmployeeForm")

**public** String showNewEmployeeForm(Model model) {

Employee employee = **new** Employee();

model.addAttribute("employee", employee);

**return** "new\_employee";

}

@PostMapping("/saveEmployee")

**public** String saveEmployee(@ModelAttribute("employee")Employee employee) {

employeeService.saveEmployee(employee);

**return** "redirect:/";

}

@GetMapping("/showFormForUpdate/{id}")

**public** String showFormForUpdate(@PathVariable(value = "id") **long** id, Model model) {

// get employee from the service

Employee employee = employeeService.getEmployeeById(id);

// set employee as a model attribute to pre-populate the form

model.addAttribute("employee", employee);

**return** "update\_employee";

}

@GetMapping("/deleteEmployee/{id}")

**public** String deleteEmployee(@PathVariable(value = "id") **long** id) {

// call delete employee method

**this**.employeeService.deleteEmployeeById(id);

**return** "redirect:/";

}

}

# 10)Front-end changes

## **index.html**

<!DOCTYPE html>

<html lang=*"en"* xmlns:th=*"http://www.thymeleaf.org"*>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Employee Management System</title>

<link rel=*"stylesheet"* href=*"https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css"*

integrity=*"sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"*

crossorigin=*"anonymous"*>

</head>

<body>

<div class=*"container my-2"*>

<h1> Employees List</h1>

<a th:href = *"@{/showNewEmployeeForm}"* class=*"btn btn-primary btn-sm mb-3"*> Add Employee</a>

<br> <br>

<table border=*"1"* class=*"table table-striped table-responsive-md"*>

<thead>

<tr>

<th> Employee First Name </th>

<br>

<th> Employee Last Name </th>

<br>

<th> Employee Email </th>

<br>

<th>Actions</th>

<br>

</tr>

</thead>

<tbody>

<tr th:each=*"employee : ${listEmployees}"*>

<td th:text= *"${employee.firstName}"*></td>

<td th:text= *"${employee.lastName}"*></td>

<td th:text= *"${employee.email}"*></td>

<td><a th:href=*"@{/showFormForUpdate/{id}(id=${employee.id})}"* class=*"btn btn-primary"*>Update</a>

<a th:href=*"@{/deleteEmployee/{id}(id=${employee.id})}"* class=*"btn btn-danger"*>Delete</a>

</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

# 11) Create new\_employee.html

<!DOCTYPE html>

<html lang=*"en"* xmlns:th=*"http://www.thymeleaf.org"*>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Employee Management System</title>

<link rel=*"stylesheet"* href=*"https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css"*

integrity=*"sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"*

crossorigin=*"anonymous"*>

</head>

<body>

<div class=*"container"*>

<h1> Employee Management System</h1>

<hr>

<h2> Save Employee</h2>

<form action= *"#"* th:action=*"@{/saveEmployee}"* th:object=*"${employee}"* method=*"POST"*>

<input type =*"text"* th:field = *"\*{firstName}"*

placeholder= *"Employee First Name"* class=*"form-control mb-4 col-4"*>

<input type =*"text"* th:field = *"\*{lastName}"*

placeholder= *"Employee Last Name"* class=*"form-control mb-4 col-4"*>

<input type =*"text"* th:field = *"\*{email}"*

placeholder= *"Employee Email"* class=*"form-control mb-4 col-4"*>

<button type=*"submit"* class=*"btn btn-info col-2"*>Save Employee</button>

</form>

</div>

</body>

</html>

## **12) Create update\_employee.html**

<!DOCTYPE html>

<html lang=*"en"* xmlns:th=*"http://www.thymeleaf.org"*>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Employee Management System</title>

<link rel=*"stylesheet"* href=*"https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css"*

integrity=*"sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"*

crossorigin=*"anonymous"*>

</head>

<body>

<div class=*"container"*>

<h1> Employee Management System</h1>

<hr>

<h2> Update Employee</h2>

<form action= *"#"* th:action=*"@{/saveEmployee}"* th:object=*"${employee}"* method=*"POST"*>

<input type=*"hidden"* th:field=*"\*{id}"* />

<input type=*"text"* th:field=*"\*{firstName}"* class=*"form-control mb-4 col-4"*>

<input type=*"text"* th:field=*"\*{lastName}"* class=*"form-control mb-4 col-4"*>

<input type=*"text"* th:field=*"\*{email}"* class=*"form-control mb-4 col-4"*>

<button type=*"submit"* class=*"btn btn-info col-2"*> Update Employee</button>

</form>

<hr>

<a th:href=*"@{/}"*> Back to Employee List</a>

</div>

</body>

</html>