* **JDBC Demo using Java file**
* **Add Required dependencies in pom.xml file**

<dependencies>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.3.25</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.25</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.3.25</version>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

</dependency>

</dependencies>

**Beanconfig file**

package com.psl.training.config;

import javax.sql.DataSource;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

@Configuration // this annotation will tell spring that this class provides configurations

@ComponentScan(basePackages = "com.psl.training")

public class BeanConfig {

@Bean

public DataSource getDataSource() {

DriverManagerDataSource datasource = new DriverManagerDataSource();

datasource.setDriverClassName("com.mysql.cj.jdbc.Driver");

datasource.setUrl("jdbc:mysql://localhost:3306/employeesystem?allowPublicKeyRetrieval=true&useSSL=false");

datasource.setUsername("root");

datasource.setPassword("root");

return datasource;

}

// JDBC Template will help to execute Queries

@Bean

public JdbcTemplate getJdbcTemplate() {

JdbcTemplate jdbcTemplate = new JdbcTemplate();

jdbcTemplate.setDataSource(getDataSource());

return jdbcTemplate;

}

}

**Entity / Model class**

**package** com.psl.training.entity;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.stereotype.Component;

@Component("emp")

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

**private** **int** id;

**private** String name;

**public** **int** getId() {

**return** id;

}

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

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + "]";

}

}

**DAO Layer**

**package** com.psl.training.dao;

**import** java.util.List;

**import** com.psl.training.entity.Employee;

**public** **interface** EmployeeDAO {

**void** create(Employee emp);// insert record

**void** update(Employee e);

**void** delete(Employee e);

**void** select(**int** id);// access single record

List<Employee> readall();// Access all data

}

**package** com.psl.training.dao;

**import** java.util.List;

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

**import** org.springframework.jdbc.core.JdbcTemplate;

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

**import** com.psl.training.entity.Employee;

@Repository

**public** **class** EmployeeDAOImpl **implements** EmployeeDAO {

@Autowired

JdbcTemplate jdbctemplate;

**public** **void** create(Employee emp) {

String sql = "insert into emp values(?,?)";

jdbctemplate.update(sql, emp.getId(), emp.getName());

System.***out***.println("Data inserted sucessfully");

}

**public** **void** update(Employee e) {

String sql = "update emp set id=? where name=?";

jdbctemplate.update(sql, e.getId(), e.getName());

System.***out***.println("Data updated sucessfully");

}

**public** **void** delete(Employee e) {

String sql = "delete from emp where id=?";

jdbctemplate.update(sql, e.getId());

System.***out***.println("Data deleted sucessfully");

}

**public** **void** select(**int** id) {// select one record from database table

EmployeeRowMapperImpl r = **new** EmployeeRowMapperImpl();

String sql = "select \* from emp where id=?";

Employee e = jdbctemplate.queryForObject(sql, r, id);

System.***out***.println(e);

}

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

EmployeeRowMapperImpl r = **new** EmployeeRowMapperImpl();

String sql = "select \* from emp";

List<Employee> list = jdbctemplate.query(sql, r);

**return** list;

}

}

**RowMapper implementation for select query**

package com.psl.training.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import com.psl.training.entity.Employee;

public class EmployeeRowMapperImpl implements RowMapper<Employee> {

public Employee mapRow(ResultSet rs, int rowNum) throws SQLException {

Employee e = new Employee();

e.setId(rs.getInt(1));

e.setName(rs.getString(2));

return e;

}

}

**Service Layer**

**package** com.psl.training.service;

**import** java.util.List;

**import** com.psl.training.entity.Employee;

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

**void** creatEmp(Employee emp);

**void** updatemp(Employee emp);

**void** deleteemp(Employee emp);

**void** selectempbyid(**int** id);

List<Employee> fetchallEmployee();

}

package com.psl.training.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.psl.training.dao.EmployeeDAO;

import com.psl.training.entity.Employee;

@Service

public class EmployeeServiceImpl implements EmployeeService {

@Autowired

EmployeeDAO dao;

public void creatEmp(Employee emp) {// insert query service

dao.create(emp);

}

public void updatemp(Employee emp) {

dao.update(emp);

}

public void deleteemp(Employee emp) {

dao.delete(emp);

}

public void selectempbyid(int id) {

dao.select(id);

}

public List<Employee> fetchallEmployee() {

return dao.readall();

}

}

Test class

package com.psl.training;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.psl.training.config.BeanConfig;

import com.psl.training.entity.Employee;

import com.psl.training.service.EmployeeService;

public class Main {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(BeanConfig.class);

Employee emp = context.getBean("emp", Employee.class);

// emp.setId(101);

EmployeeService service = context.getBean(EmployeeService.class);

// service.creatEmp(emp);

// service.updatemp(emp);

// service.deleteemp(emp);

// service.selectempbyid(101);

List<Employee> list = service.fetchallEmployee();

for (Employee e : list) {

System.out.println(e);

}

}

}

* **JDBC Demo using XML file**

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

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd"*>

<context:component-scan

base-package=*"com.psl.training"*></context:component-scan>

<bean name=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*

p:driverClassName=*"com.mysql.cj.jdbc.Driver"*

p:url=*"jdbc:mysql://localhost:3306/employeesystem"* p:username=*"root"*

p:password=*"root"*>

</bean>

<bean name=*"jdbctemplate"*

class=*"org.springframework.jdbc.core.JdbcTemplate"*

p:dataSource-ref=*"dataSource"*>

<!-- <property name="dataSource" ref="dataSource"></property> -->

</bean>

</beans>