

# Chapter#3 Spring Boot JDBC CURD

---

## **application.properties**

```
server.port=9898  
spring.h2.console.enabled=true
```

## **1. Model class**

```
package in.nit.model;  
  
import java.sql.ResultSet;  
  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class Employee {  
  
    private Integer empId;  
    private String empName;  
    private Double empSal;  
  
    public static Employee of(ResultSet rs) {  
        Employee emp=null;  
        try {  
            emp=new Employee(  
                rs.getInt("eid"),  
                rs.getString("ename"),  
                rs.getDouble("esal"));  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        return emp;  
    }  
}
```

## 2. RowMapper class

```
package in.nit.mapper;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import in.nit.model.Employee;

public class EmployeeRowmapper
    implements RowMapper<Employee>
{

    public Employee mapRow(ResultSet rs,int count)
        throws SQLException
    {
        Employee e=new Employee();
        e.setEmpId(rs.getInt("eid"));
        e.setEmpName(rs.getString("ename"));
        e.setEmpSal(rs.getDouble("esal"));

        return e;
    }

}
```

## 3. IEmployee Dao

```
package in.nit.dao;

import java.util.List;

import in.nit.model.Employee;

public interface IEmployeeDao {

    int saveEmployee(Employee e);

}
```

```
int deleteEmployee(Integer id);
int updateEmployee(Employee e);

Employee getOneEmployee(Integer id);
List<Employee> getAllEmployees();

}
```

#### 4. EmployeeDaoImpl

```
package in.nit.dao.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

import in.nit.dao.IEmployeeDao;
import in.nit.model.Employee;

@Repository
public class EmployeeDaoImpl implements IEmployeeDao {

    @Autowired
    private JdbcTemplate jt; //HAS-A

    public int saveEmployee(Employee e) {
        String sql="INSERT INTO EMPTAB VALUES(?,?,?)";
        int
count=jt.update(sql,e.getEmpId(),e.getEmpName(),e.getEmpSal());
        return count;
    }

    public int deleteEmployee(Integer id) {
        String sql="DELETE FROM EMPTAB WHERE EID=?";
        int count=jt.update(sql,id);
        return count;
    }

    public int updateEmployee(Employee e) {
        String sql="UPDATE EMPTAB SET ENAME=?,ESAL=? WHERE EID=?";
```

```
        int
count=jt.update(sql,e.getEmpName(),e.getEmpSal(),e.getEmpId());
        return count;
    }

    public Employee getOneEmployee(Integer id) {
        String sql="SELECT * FROM EMPTAB WHERE EID=?";
        /*
        EmployeeRowmapper er=new EmployeeRowmapper();
        Employee e=jt.queryForObject(sql,er,id);
        */
        /*
        Employee e=jt.queryForObject(
            sql, (rs,count)->
            new Employee(
                rs.getInt("eid"),
                rs.getString("ename"),
                rs.getDouble("esal")),
            id);
        */
        Employee e=jt.queryForObject(sql,(rs,count)->
Employee.of(rs),id);
        return e;
    }

    @Override
    public List<Employee> getAllEmployees() {
        String sql="SELECT * FROM EMPTAB";

        /*EmployeeRowmapper er=new EmployeeRowmapper();
        List<Employee> emps=jt.query(sql,er);*/

        /*List<Employee> emps=jt.query(sql,
            (rs,count)->
            new Employee(
                rs.getInt("eid"),
                rs.getString("ename"),
                rs.getDouble("esal"))
            );
        */
        List<Employee> emps=jt.query(sql, (rs,count)->
Employee.of(rs));
        return emps;
    }
}
```

## 5. Runner code

```
package in.nit.runner;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;

import in.nit.model.Employee;
import in.nit.service.IEmployeeService;

@Component
public class JdbcTestRunner
implements CommandLineRunner {
    @Autowired
    private IEmployeeService service;

    @Override
    public void run(String... args) throws Exception {

        //update data
        /*Employee e=new Employee();
        e.setEmpId(10);
        e.setEmpName("IJKL");
        e.setEmpSal(999.99);
        service.updateEmployee(e);*/

        //remove data
        // service.deleteEmployee(12);

        //save data
        /*
        * Employee e=new Employee(); e.setEmpId(14);
        e.setEmpName("RR");
        * e.setEmpSal(66.66); service.saveEmployee(e);
        System.out.println("done");
        */

        //fetch one row
        Employee e=service.getOneEmployee(12);
        System.out.println(e);
    }
}
```

- by RAGHU SIR [NareshIT, Hyderabad]

```
        System.out.println("-----");  
        //fetch all  
        service.getAllEmployees().forEach(System.out::println);  
    }  
}
```

FB: <https://www.facebook.com/groups/thejavatemple/>

email: javabyraghu@gmail.com