

Assignment 13

//CaseStudy-1

```
package com.spring.JdbcTemplate2;

public class Employee {

    private int id;

    private String name;

    private int salary;

    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;

    }

    public int getSalary() {

        return salary;

    }

    public void setSalary(int salary) {

        this.salary = salary;

    }

    public Employee() {

    }

    public Employee(int id, String name, int salary) {

        super();

        this.id=id;
```

```

this.name=name;

this.salary=salary;
}
}

package com.spring.JDBCTemplate2;

import org.springframework.jdbc.core.JdbcTemplate;

import java.util.*;

public class Employee2 {

private JdbcTemplate jdbcTemplate;

public void setJdbcTemplate (JdbcTemplate jdbcTemplate) {

this.jdbcTemplate=jdbcTemplate;

}

public int saveStudent( Employee e) {

String query="insert into student values('"+
e.getId()+"','"+e.getName()+"','"+e.getSalary()+"'");

return jdbcTemplate.update(query);

}

}

public List< Employee>listEmployee (){

String SQL="Select * from student";

package com.spring.PTM2;

import java.util.List;

import javax.sql.DataSource;

import org.springframework.dao.DataAccessException;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.transaction.PlatformTransactionManager;

import org.springframework.transaction.TransactionDefinition;

import org.springframework.transaction.TransactionStatus;

import org.springframework.transaction.support.DefaultTransactionDefinition;

public class EmployeeJDBCTemplate implements Employee2 {

private DataSource dataSource;

```

```

private JdbcTemplate jdbcTemplateObject;

private PlatformTransactionManager transactionManager;

public void setDataSource(DataSource ds) {
    this.dataSource=ds;
    this.jdbcTemplateObject=new JdbcTemplate(dataSource);
}

public void setTransactionManager(PlatformTransactionManager transactionManager) {
    this.transactionManager=transactionManager;
}

public void create(String name, int age, int marks, int year) {
    TransactionDefinition def=new DefaultTransactionDefinition();
    TransactionStatus status=transactionManager.getTransaction(def);
    try {
        String SQL1="insert into Student (name, age) values(?,?)";
        jdbcTemplateObject.update(SQL1,name,age);
        String SQL2="select max(id) from Student";
        int sid=jdbcTemplateObject.queryForObject(SQL2, Integer.class);
        String SQL3="insert into Marks(sid,marks,year)"+ "values(?,?,?)";
        jdbcTemplateObject.update(SQL3,sid,marks,year);
        System.out.println("Created Name= "+name+", Age= "+age);
        transactionManager.commit(status);
    }
    catch(DataAccessException e) {
        System.out.println("Error in creating record, rolling back");
        transactionManager.rollback(status);
    }
    return;
}

public List<Employee> listStudents(){
    String SQL="select * from Employee, marks where ID=marks.sid";
    List<Employee>employee=jdbcTemplateObject.query( SQL, new EmployeeMapper());
}

```

```

return employee;
}

public void create(String name, Integer id, Integer salary) {
// TODO Auto-generated method stub
}
}

```

Beans.xml

```

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

<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="com.mysql.cj.jdbc.Driver"/>
<property name="url" value="jdbc:mysql://localhost:3306/mydb"/>
<property name="username" value="root"/>
<property name="password" value="Likhi2123"/>
</bean>

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate2">
<property name="dataSource" ref="dataSource"></property>
</bean>

<bean id="jdbc" class="com.spring.JdbcTemplate.EMployee2">
<property name="jdbcTemplate2" ref="jdbcTemplate2"/>
</bean>
</beans>

```

```

List < Employee>em=jdbcTemplate.query(SQL,new EmployeeMapper());
return em;
}
}

```

//Casestudy-2

```
package com.spring.Tm;

public class Employee {

    private String name;

    private int salary;

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public int getSalary() {

        return salary;

    }

    public void setSalary(int salary) {

        this.salary = salary;

    }

    public void displayinfo() {

        System.out.println("Name: "+name);

        System.out.println("Salary_of_EMployee:"+salary);

    }

}

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

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

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

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">
```

```

<!-- <bean id="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="com.mysql.cj.jdbc.Driver"/>

<property name="url" value="jdbc:mysql://localhost:3306/mydb"/>

<property name="username" value="poojitha"/>

<property name="password" value="pooja@12"/>

</bean>-->

<!-- <bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

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

</bean>

<bean id="studentJdbcTemplate" class="com.spring.PTM.StudentJdbcTemplate">

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

<property name="transactionManager" ref="transactionManager"/>

</bean>

</beans>-->

<bean id="a" class="com.spring.Tm.Employee">

<property name="name" value ="Ravanan"/>

<property name="salary" value ="9000"/>

</bean>

</beans>

```

```

package com.spring.Tm;

import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.ClassPathResource;
import org.springframework.core.io.Resource;

public class Mainapp {

    public static void main(String[] args) {

        Resource r=new ClassPathResource("Beans.xml");

        BeanFactory factory=new XmlBeanFactory(r);

        Employee e=(Employee)factory.getBean("a");
    }
}

```

```
e.displayinfo();
```

```
}
```

```
}
```

```
//caseStudy-3
```

```
package com.spring.InnerBean2;
```

```
public class Employee {
```

```
    private String name;
```

```
    public String getName() {
```

```
        return name;
```

```
    }
```

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

```
        this.name = name;
```

```
    }
```

```
    @Override
```

```
    public String toString() {
```

```
        // TODO Auto-generated method stub
```

```
        return super.toString();
```

```
    }
```

```
}
```

```
package com.spring.InnerBean2;
```

```
public class EMP {
```

```
    private Employee employee;
```

```
    public EMP(Employee employee) {
```

```
        this.employee= employee;
```

```
    }
```

```
    public void setEmployee(Employee employee) {
```

```
        this.employee= employee;
```

```
    }
```

```
    public String toString() {
```

```
        return "Hai\n"+employee;
```

```

}
}

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

<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<!--<bean id="h" class="com.spring.InnerBean.Helloword">

<constructor-arg>

<bean class="com.spring.InnerBean.Foo">
<property name="name" value="fooName"/>
</bean>

</constructor-arg>

</bean>-->

<bean id="ok" class="com.spring.InnerBean2.EMP">
<constructor-arg>
<bean class="com.spring.InnerBean2.Employee">
<property name="name" value="Ravanan"/>
</bean>

</constructor-arg>

</bean>

</beans>

package com.spring.InnerBean2;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.spring.InnerBean.Helloword;

public class MAinAPP {

public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext context=new ClassPathXmlApplicationContext("Beans.xml");

EMP e=(EMP)context.getBean("ok");

```



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

```
}
```

```
}
```