Spring core +java +mysql +jdbc crud opration

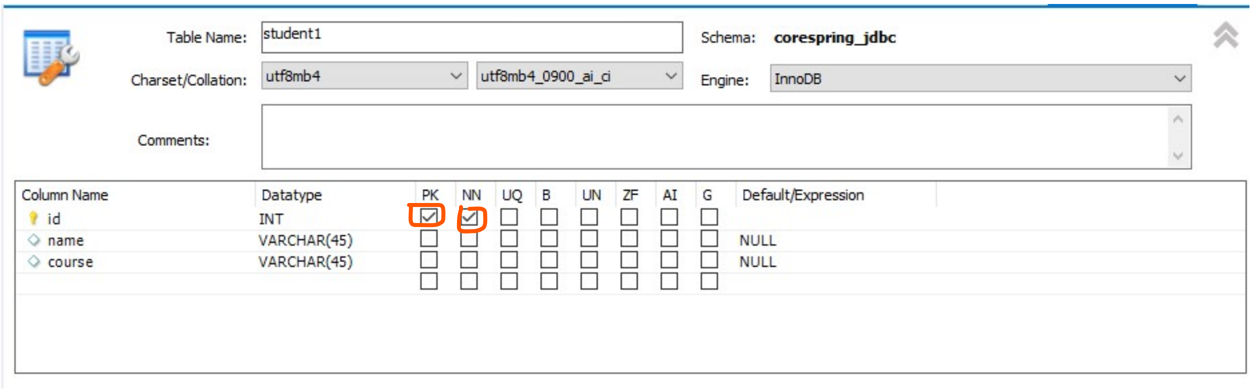


Table Name: student1

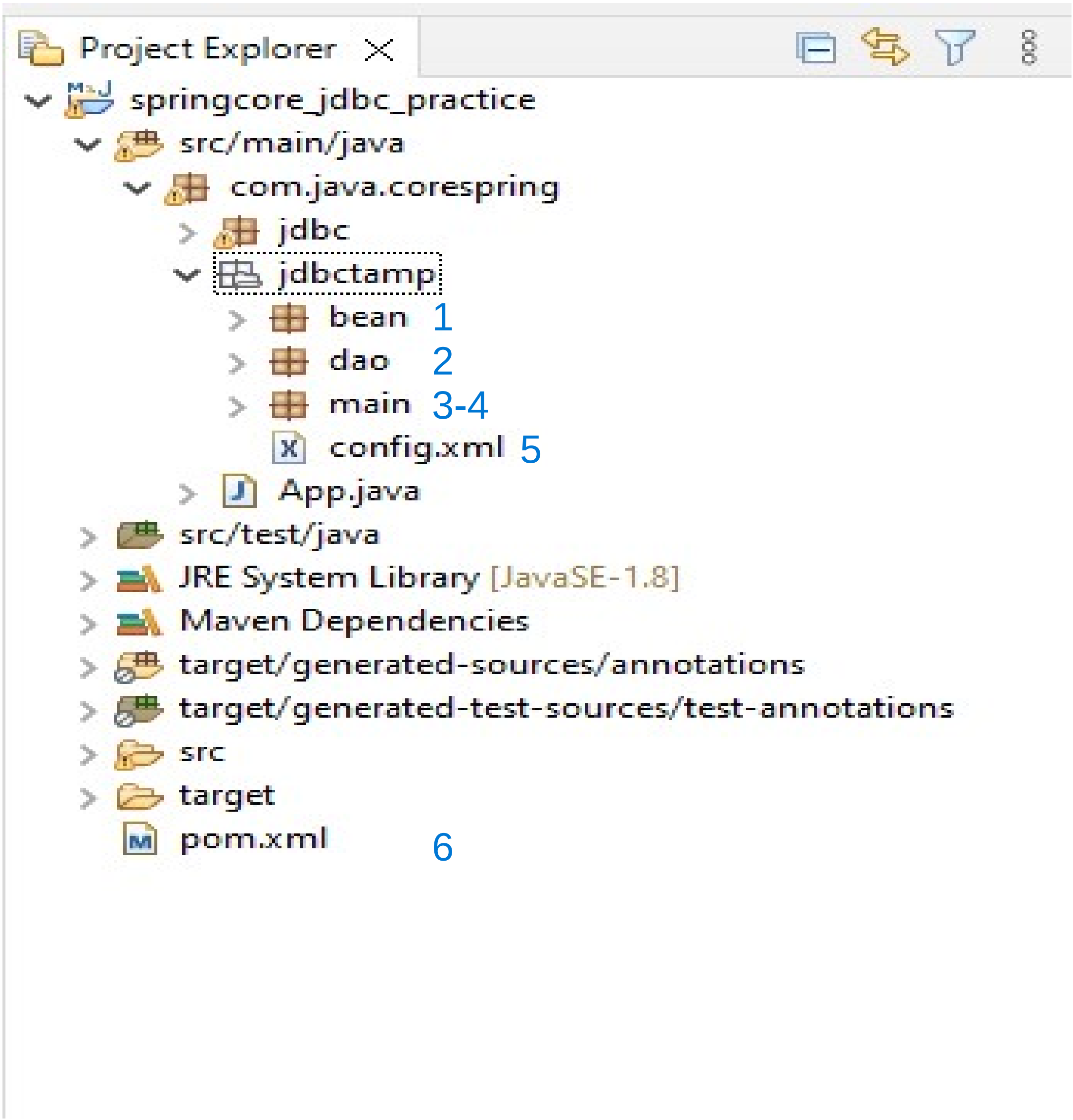
Schema: corespring_jdbc

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	☑	☑	□	□	□	□	□	□	
name	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
course	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
		□	□	□	□	□	□	□	□	



```
Project Explorer
springcore_jdbc_practice
  src/main/java
    com.java.corespring
      jdbc
      jdbctamp
        bean        1
        dao         2
        main        3-4
        config.xml  5
      App.java
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
  src
  target
  pom.xml           6
```

1. Package com.java.corespring.jdbctamp.bean

```
package com.java.corespring.jdbctamp.bean;

public class Student {

private int id;

private String name;
private String course;
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 String getCourse() {
return course;
}
public void setCourse(String course) {
this.course = course;
}
@Override
public String toString() {
```

```
return "Student [id=" + id + ", name=" + name + ", course="
+ course + "]";

}

}
```

2. Package com.java.corespring.jdbctamp.dao

```
package com.java.corespring.jdbctamp.dao;

import java.util.List;

import com.java.corespring.jdbctamp.bean.Student;

public interface StudentDAO {

//create

public void save(Student s);

//read

public Student getById(int id);

//update

public void update(Student s);

//delete

public void deleteById(int id);

//read all

public List<Student> getAll();

}
```

3. com.java.corespring.jdbctamp.dao

```java
package com.java.corespring.jdbctamp.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import javax.sql.DataSource;

import com.java.corespring.jdbctamp.bean.Student;

public class StudentDAOImpl implements StudentDAO

{

//create datasource

private DataSource dataSource;

public void setDataSource(DataSource dataSource)

{

this.dataSource=dataSource;

}

//----------------------------------------------------------
-----------------------------------

@Override

public void save(Student s) {
```

```java
String query="insert into
student1(id,name,course)values(?,?,?)";

Connection con=null;

PreparedStatement ps=null;

try

{

con=dataSource.getConnection();

ps=con.prepareStatement(query);

ps.setInt(1, s.getId());

ps.setString(2, s.getName());

ps.setString(3,s.getCourse());

int result=ps.executeUpdate();

if(result!=0)

{

System.out.println("Student Save with id:"+s.getId());

}else

{

System.out.println("Sorry faild Your request:"+s.getId());

}

}catch (SQLException e) {

e.printStackTrace();

}finally
```

```
{
try {
ps.close();
con.close();
} catch (SQLException e) {
e.printStackTrace();
}
}
}
//----------------------------------------------------------
------------------------------------------------------------
---
@Override
public Student getById(int id) {
String query="select name,course from student1 where id=?";
Student st=null;
Connection con=null;
PreparedStatement ps=null;
ResultSet rs=null;
try {
con=dataSource.getConnection();
ps=con.prepareStatement(query);
```

```
ps.setInt(1, id);
rs=ps.executeQuery();
System.out.println("Id"+"\t"+"name"+"\t"+"Cource");
if(rs.next()) {
st=new Student();
st.setId(id);
st.setName("name");
st.setCourse("course");
System.out.println("Found:"+st);
}else
{
System.out.println("Not Found!");
}
} catch (SQLException e) {
e.printStackTrace();
}finally {
try {
ps.close();
con.close();
} catch (Exception e2) {
// TODO: handle exception
```

```
}
}
return st;
}
//-------------------------------------------------------------------------------------------------------------
@Override
public void update(Student s) {
String query="update student1 set name=?,course=? where id=?";
Connection con=null;
PreparedStatement ps=null;
try {
con=dataSource.getConnection();
ps=con.prepareStatement(query);
ps.setString(1,s.getName());
ps.setString(2,s.getCourse());
ps.setInt(3,s.getId());
int result=ps.executeUpdate();
if(result!=0)
{
System.out.println("Student Update Details By iD:"+s.getId());
```

```
}else

{

System.out.println("Sorry Not Found!");

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

ps.close();

con.close();

} catch (Exception e2) {

e2.printStackTrace();

}

}

}

//----------------------------------------------------------
-------------------------------------------------

@Override

public void deleteById(int id) {

String query="delete from student1 where id=?";

// Student st=null;

Connection con=null;
```

```
PreparedStatement ps=null;
try
{
con=dataSource.getConnection();
ps=con.prepareStatement(query);
ps.setInt(1, id);
int result=ps.executeUpdate();
if(result!=0)
{
System.out.println("Student Details deleted by id:"+id);
}
else
{
System.out.println("Not match id:"+id);
}
}catch (SQLException e) {
e.printStackTrace();
}finally
{
try
{
```

```
ps.close();

con.close();

}catch (Exception e) {

}

}

}

//------------------------------------------------------------------------------------------------------------

@Override

public List<Student> getAll() {

String query="select*from student1";

List<Student> list=new ArrayList<Student>();

Connection con=null;

PreparedStatement ps=null;

ResultSet rs=null;

try

{

con=dataSource.getConnection();

ps=con.prepareStatement(query);

rs=ps.executeQuery();

while(rs.next())

{
```

```java
Student st=new Student();

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

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

st.setCourse(rs.getString(3));

list.add(st);

}}catch(SQLException e){

e.printStackTrace();

}finally{

try {

rs.close();

ps.close();

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

return list;

}

}
```

4. com.java.corespring.jdbctamp.main

```java
package com.java.corespring.jdbctamp.main;
```

```java
import java.util.List;
import java.util.Random;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.java.corespring.jdbctamp.bean.Student;
import com.java.corespring.jdbctamp.dao.StudentDAO;
public class Main {
public static void main(String[] args) {
ClassPathXmlApplicationContext context=
new ClassPathXmlApplicationContext("com/java/corespring/jdbctamp/config.xml");
//get the dao class
StudentDAO student=context.getBean("studentDAO",StudentDAO.class);
//run some for test jdbc crud opration
Student st=new Student();
int rand=new Random().nextInt(1000);
st.setId(rand);
st.setName("MerajAlam");
st.setCourse("Java");
//create
student.save(st);
//read data from datacbase
Student st1=student.getById(rand);
System.out.println("Student Retrive data:"+st1);
//update
st.setCourse("Java Developer");
student.update(st1);
//get All deatils
List<Student> list=student.getAll();
System.out.println(list);
//delete
student.deleteById(rand);
//close spring context
context.close();
}
}
```

5. Config.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"
xmlns:context="http://www.springframework.org/schema/context"
```

```
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd">
<bean
class="org.springframework.jdbc.datasource.DriverManagerDataSource" id="ds">
<property name="driverClassName"
value="com.mysql.jdbc.Driver"/>
<property name="url"
value="jdbc:mysql://localhost:3306/corespring_jdbc"/>
<property name="username" value="root"/>
<property name="password" value="123456"/>
</bean>
<bean
class="com.java.corespring.jdbctamp.dao.StudentDAOImpl"
id="studentDAO">
<property name="dataSource" ref="ds">
</property>
</bean>
</beans>
```

6. Pom.xml

```
<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
http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.java.corespring</groupId>

<artifactId>SpringCore</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>SpringCore</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>
```

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

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>6.0.4</version>

</dependency>

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

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>6.0.4</version>

</dependency>

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

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>6.0.4</version>

</dependency>
```

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

<dependency>

<groupId>com.mysql</groupId>

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

<version>8.0.32</version>

</dependency>

</dependencies>

</project>

======================**ends***======================
```