MYBATIS - STORED PROCEDURES

http://www.tutorialspoint.com/mybatis/mybatis_stored_procedures.htm

You can call a stored procedure using MyBatis. First of all, let us understand how to create a stored procedure in MySQL.

We have the following EMPLOYEE table in MySQL −

```
CREATE TABLE details.student(
   ID int(10) NOT NULL AUTO_INCREMENT,
   NAME varchar(100) NOT NULL,
   BRANCH varchar(255) NOT NULL,
   PERCENTAGE int(3) NOT NULL,
   PHONE int(11) NOT NULL,
   EMAIL varchar(255) NOT NULL,
   PRIMARY KEY (`ID`)
);
```

Let us create the following stored procedure in MySQL database −

```
DELIMITER //
   DROP PROCEDURE IF EXISTS details.read_recordById //
   CREATE PROCEDURE details.read_recordById (IN emp_id INT)

   BEGIN
      SELECT * FROM STUDENT WHERE ID = emp_id;
   END//

DELIMITER ;
```

Assume the table named STUDENT has two records as −

```
mysql> select * from STUDENT;
+----+----------+--------+------------+-----------+----------------------+
| ID |   NAME   | BRANCH | PERCENTAGE |   PHONE   |       EMAIL          |
+----+----------+--------+------------+-----------+----------------------+
|  1 | Mohammad |  It    |     80     | 900000000 | mohamad123@yahoo.com |
|  2 | Shyam    |  It    |     75     | 984800000 | shyam@gmail.com      |
+----+----------+--------+------------+-----------+----------------------+
2 rows in set (0.00 sec)
```

STUDENT POJO Class

To use stored procedure, you do not need to modify the Student.java file. Let us keep it as it was in the last chapter.

```
public class Student {

   private int id;
   private String name;
   private String branch;
   private int percentage;
   private int phone;
   private String email;

   public Student(int id, String name, String branch, int percentage, int phone, String
email) {
      super();
      this.id = id;
      this.name = name;
      this.setBranch(branch);
      this.setPercentage(percentage);
      this.phone = phone;
      this.email = email;
   }

   public Student() {}

   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 getPhone() {
      return phone;
   }

   public void setPhone(int phone) {
      this.phone = phone;
   }

   public String getEmail() {
      return email;
   }

   public void setEmail(String email) {
      this.email = email;
   }

   public String getBranch() {
      return branch;
   }

   public void setBranch(String branch) {
      this.branch = branch;
   }

   public int getPercentage() {
      return percentage;
   }

   public void setPercentage(int percentage) {
      this.percentage = percentage;
   }

   public String toString(){
      StringBuilder sb = new StringBuilder();

      sb.append("Id = ").append(id).append(" - ");
      sb.append("Name = ").append(name).append(" - ");
      sb.append("Branch = ").append(branch).append(" - ");
      sb.append("Percentage = ").append(percentage).append(" - ");
      sb.append("Phone = ").append(phone).append(" - ");
      sb.append("Email = ").append(email);

      return sb.toString();
   }

}
```

Student.xml File

Unlike IBATIS, there is no **<procedure>** tag in MyBatis. To map the results of the procedures, we have created a resultmap named Student and to call the stored procedure named read_recordById. We have defined a select tag with id callById, and we use the same id in the application to call the procedure.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace = "Student">

   <resultMap id = "result" type = "Student">
      <result property = "id" column = "ID"/>
      <result property = "name" column = "NAME"/>
      <result property = "branch" column = "BRANCH"/>
      <result property = "percentage" column = "PERCENTAGE"/>
      <result property = "phone" column = "PHONE"/>
      <result property = "email" column = "EMAIL"/>
   </resultMap>

   <select id = "callById" resultMap = "result" parameterType = "Student" statementType
= "CALLABLE">
      {call read_record_byid(#{id, jdbcType = INTEGER, mode = IN})}
   </select>

</mapper>
```

mybatisSP.java File

This file has application level logic to read the names of the employees from the Employee table using ResultMap −

```
import java.io.IOException;
import java.io.Reader;

import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;

public class getRecords {

   public static void main(String args[]) throws IOException{

      Reader reader = Resources.getResourceAsReader("SqlMapConfig.xml");
      SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(reader);
      SqlSession session = sqlSessionFactory.openSession();

      //select a particular student  by  id
      Student student = (Student) session.selectOne("Student.callById", 3);

      //Print the student details
      System.out.println("Details of the student are:: ");
      System.out.println("Id :"+student.getId());
      System.out.println("Name :"+student.getName());
      System.out.println("Branch :"+student.getBranch());
      System.out.println("Percentage :"+student.getPercentage());
      System.out.println("Email :"+student.getEmail());
      System.out.println("Phone :"+student.getPhone());
      session.commit();
      session.close();

   }

}
```

Compilation and Run

Here are the steps to compile and run the getRecords program. Make sure, you have set PATH and CLASSPATH appropriately before proceeding for compilation and execution.

- Create Student.xml as shown above.
- Create Student.java as shown above and compile it.
- Create getRecords.java as shown above and compile it.
- Execute getRecords binary to run the program.

You will get the following result −

```
Details of the student are::
Id :2
Name :Shyam
Branch :It
Percentage :75
Email :shyam@gmail.com
Phone :984800000
```