IBatis (MyBatis): Working with Stored Procedures

March 29, 2011 | By Loiane

This tutorial will walk you through how to setup iBatis (MyBatis) in a simple Java project and will present how to work with stored procedures using MySQL.

The goal os this tutorial is to demonstrate how to execute/call stored procedures using iBatis/MyBatis.

mybatis
SQL Mapping Framework for Java

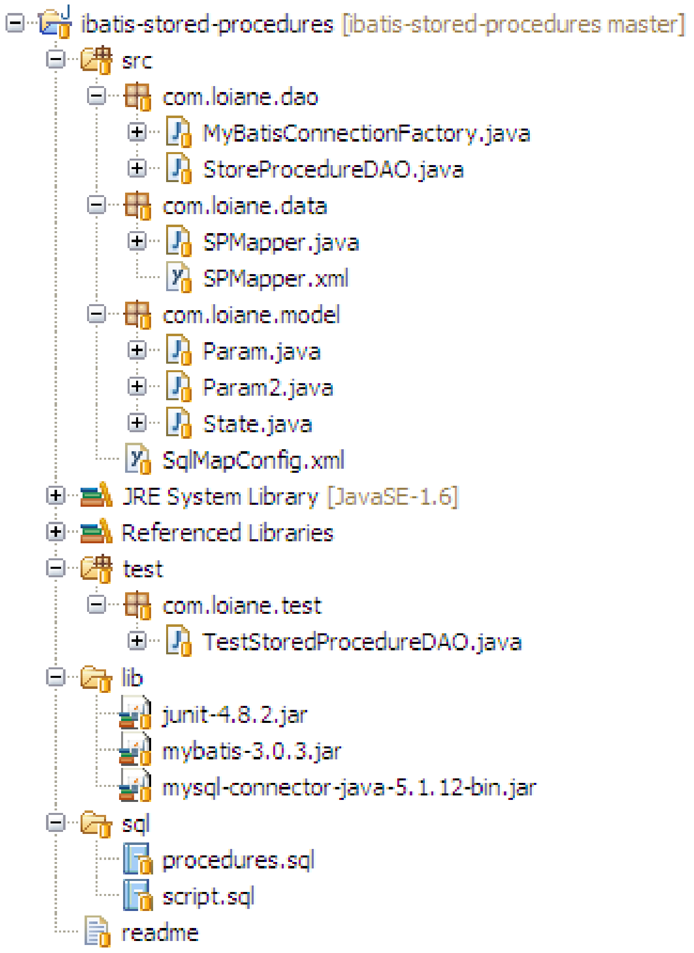
iBATIS

Pre-Requisites

For this tutorial I am using:

IDE: Eclipse (you can use your favorite one)
DataBase: MySQL
Libs/jars: Mybatis, MySQL conector and JUnit (for testing)

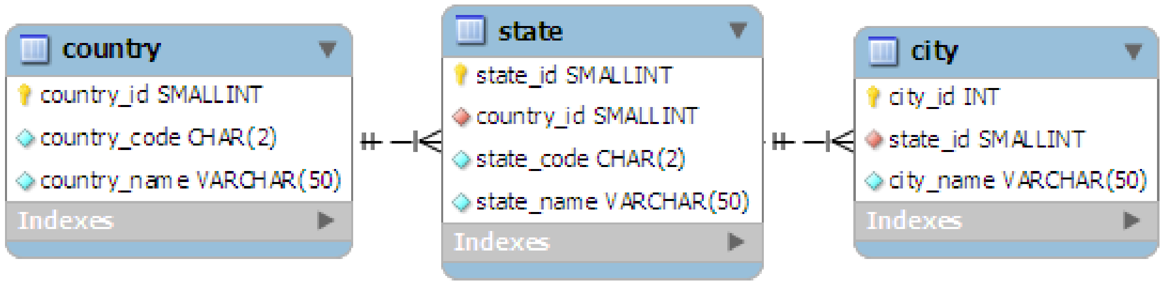
This is how your project should look like:



```
ibatis-stored-procedures [ibatis-stored-procedures master]
  src
    com.loiane.dao
      MyBatisConnectionFactory.java
      StoreProcedureDAO.java
    com.loiane.data
      SPMapper.java
      SPMapper.xml
    com.loiane.model
      Param.java
      Param2.java
      State.java
    SqlMapConfig.xml
  JRE System Library [JavaSE-1.6]
  Referenced Libraries
  test
    com.loiane.test
      TestStoredProcedureDAO.java
  lib
    junit-4.8.2.jar
    mybatis-3.0.3.jar
    mysql-connector-java-5.1.12-bin.jar
  sql
    procedures.sql
    script.sql
  readme
```

Sample Database

Please run the script into your database before getting started with the project implementation. You will find the script (with dummy data) inside the sql folder.



As we are going to work with stored procedures, you will also have to execute a script with procedures. Here are the procedures:

```
USE `blog_ibatis`;
DROP procedure IF EXISTS `getTotalCity`;
DELIMITER $$
USE `blog_ibatis`$$
CREATE PROCEDURE `blog_ibatis`.`getTotalCity` (OUT total INTEGER)
BEGIN
    SELECT count(*) into total
    FROM city;
END
$$
DELIMITER ;

-- ------------------------------------------------------------------

USE `blog_ibatis`;
DROP procedure IF EXISTS `getTotalCityStateId`;
DELIMITER $$
USE `blog_ibatis`$$
CREATE PROCEDURE `blog_ibatis`.`getTotalCityStateId` (IN stateId SMAl
BEGIN
    SELECT count(*) into total
    FROM city
    WHERE state_id = stateId;
END
$$
DELIMITER ;

-- ------------------------------------------------------------------

USE `blog_ibatis`;
DROP procedure IF EXISTS `getStates`;
DELIMITER $$
USE `blog_ibatis`$$
CREATE PROCEDURE `blog_ibatis`.`getStates` ()
BEGIN
    SELECT state_id, state_code, state_name
    FROM state;
END
$$
DELIMITER ;
```

1 – SPMapper – XML

I did not find anything on the user manual about how to call stored procedures, so I decided to search on the mailing list. And I found some tips of how to call stores procedures.

On the previous version, iBatis has a special XML tag for stored procedures. But there is no XML tag for it on current MyBatis version (version 3).

To call a stored procedure usgin MyBatis/iBatis 3 you will have to follow some tips:

1. Must set the statement type to **CALLABLE**
2. Must use the JDBC standard escape sequence for stored procedures: {**call xxx (parm1, parm2)**}
3. Must set the **MODE** of all parameters (**IN, OUT, INOUT**)
4. All IN, OUT, and INOUT parameters must be a part of the **parameterType** or parameterMap (discouraged). The only exception is if you are using a Map as a parameter object. In that case you **do not need to add OUT parameters to the map before calling**, MyBatis will add them for you

automatically.
5. resultType or resultMap (more typically) is only used if the procedure returns a result set.
6. ***IMPORTANT***: Oracle ref cursors are usually returned as parameters, NOT directly from the stored proc. So with ref cursors, resultMap and/or resultType is usually not used.

First Example:

We want to call the procedure *getTotalCity* and this procedure only have one OUT parameter, and no IN/INOUT parameter. How to do it?

We are going to ser inline parameters in this first example. To use inline parameters, create a POJO class to represent your parameters, set the parameterType to the class you created and you are going to use this notation to represent each parameter:

#{parameterName, mode=OUT, jdbcType=INTEGER}

- mode can be IN, OUT, INOUT
- and specify the jdbcType of your parameter

To create the Mybatis XML configuration, you can use the **select** ou **update** tag. Do not forget to set the statementType to ***CALLABLE***.

Here is how our MyBatis statement is going to look like:

```
<select id="callGetTotalCity" parameterType="Param" statementType="C
    { CALL getTotalCity(#{total, mode=OUT, jdbcType=INTEGER})}
</select>
```

And this is the POJO class which represents the parameter for *getTotalCity* procedure:

```
package com.loiane.model;

public class Param {

    private int total;

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        this.total = total;
    }
}
```

Second Example:

Now we are going to try to call the same stored procedure we demonstrated on the first example, but we are going to use a parameterMap, like you used to do in version 2.x.

A very important note: this is discouraged, please use inline parameters.

Let’s declare the Param POJO class as a parameterMap:

```
<parameterMap type="Param" id="testParameterMap">
    <parameter property="total" jdbcType="INTEGER" mode="OUT" />
</parameterMap>
```

And the stored procedure statment:

```
<update id="callGetTotalCity2" parameterMap="testParameterMap" state
    { CALL getTotalCity(?) }
</update>
```

Note that now we use “*?*” (question mark) to represent each parameter.

Third Example:

Now we are going to call a stored procedure with IN and OUT parameters. Let’s follow the same rules as the fisrt example.

We are going to use inline parameters and we are going to create a POJO class to represent our parameter.

MyBatis code:

```
<select id="callGetTotalCityStateId" parameterType="Param2" statemen
    { CALL getTotalCityStateId(
        #{stateId, mode=IN, jdbcType=INTEGER},
        #{total, mode=OUT, jdbcType=INTEGER})}
</select>
```

Param2 POJO:

```
package com.loiane.model;

public class Param2 {

    private int total;
    private int stateId;

    public int getTotal() {
        return total;
    }
    public void setTotal(int total) {
        this.total = total;
    }
    public int getStateId() {
        return stateId;
    }
    public void setStateId(int stateId) {
        this.stateId = stateId;
    }
}
```

Fourth Example:

Now let's try to retrieve a resultSet from the stored procedure. For this we are going to use a resultMap.

```
<resultMap type="State" id="resultState">
    <result property="id" column="state_id"/>
    <result property="name" column="state_name"/>
    <result property="code" column="state_code"/>
</resultMap>

<select id="callGetStates" resultMap="resultState" statementType="CALL
    { CALL getStates()}
</select>
```

State POJO class:

```
package com.loiane.model;

public class State {

    private int id;
    private String code;
    private String name;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getCode() {
        return code;
    }
    public void setCode(String code) {
        this.code = code;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

2- SPMapper – Annotations

Now let's try to do the same thing we did using XML config.

Annotation for First Example (XML):

```
@Select(value= "{ CALL getTotalCity( #{total, mode=OUT, jdbcType=INT
@Options(statementType = StatementType.CALLABLE)
Object callGetTotalCityAnnotations(Param param);
```

It is very similiar to a simple select statement, but we have to set the statement type to CALLABLE. To do it, we can use the annotation ***@Options***.

With annotations, we can only use inline parameters, so we will not be able to represent the second exemple using annotations.

Annotation for Third Example (XML):

The explanation is the same as first example, I am just going to list the code:

```
@Select(value= "{ CALL getTotalCityStateId( #{stateId, mode=IN, jdbc
@Options(statementType = StatementType.CALLABLE)
Object callGetTotalCityStateIdAnnotations(Param2 param2);
```

Annotation for Fourth Example (XML):

I tried to set the fourth example with annotation, but the only thing I've got is this:

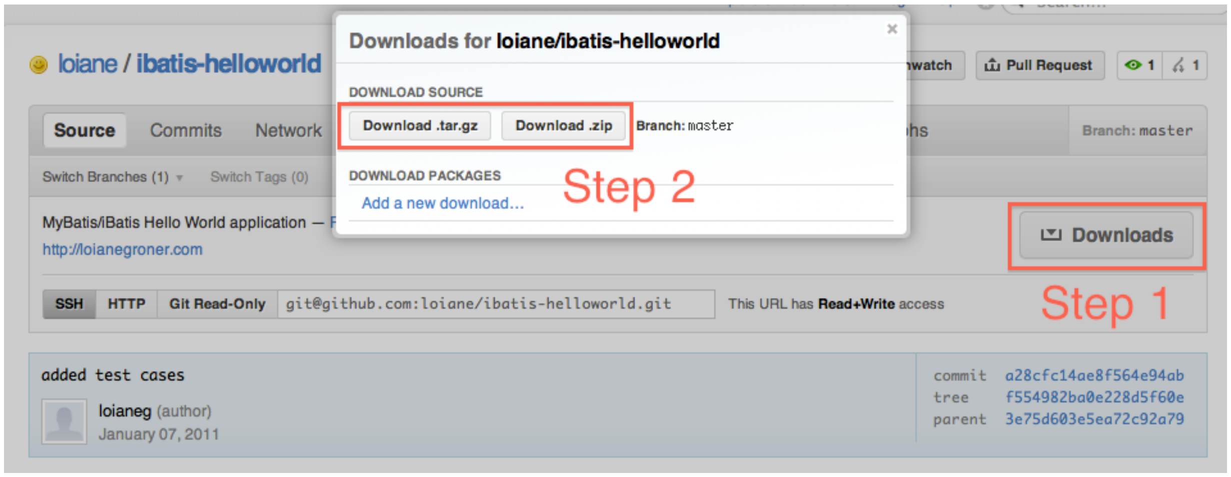
```
//TODO: set resultMap with annotations
/*@Select(value= "{ CALL getTotalCityStateId()}")
@Options(statementType = StatementType.CALLABLE)
/*@Results(value = {
    @Result(property="id", column="state_id"),
    @Result(property="name", column="state_name"),
    @Result(property="code", column="state_code"),
})*/
List<State> callGetStatesAnnotations();
```

And it does not work. I tried to search on the mailing list, no luck. I could not find a way to represent a resultMap with annotation and stored procedures. I don't know if it is a limitation. If you have any clue how to do it, please leave a comment, I will appreciate it! 🙂

Download

If you want to download the complete sample project, you can get it from my GitHub account: https://github.com/loiane/ibatis-stored-procedures

If you want to download the zip file of the project, just click on download:



MyBatis/iBatis Hello World application
http://loianegroner.com

```
git@github.com:loiane/ibatis-helloworld.git
```

```
added test cases
loianeg (author)
January 07, 2011
commit  a28cfc14ae8f564e94ab
tree    f554982ba0e228d5f60e
parent  3e75d603e5ea72c92a79
```

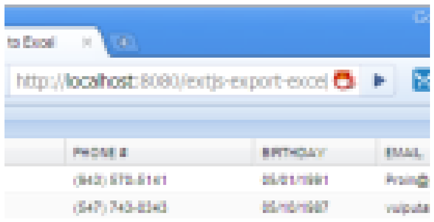
There are more articles about iBatis to come. Stay tooned!

Happy Coding! 😃

- Post to Delicious
  5

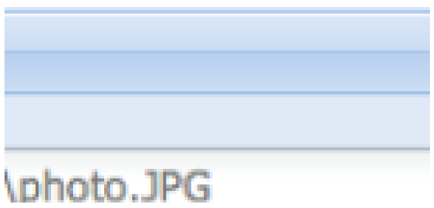
You may also like:

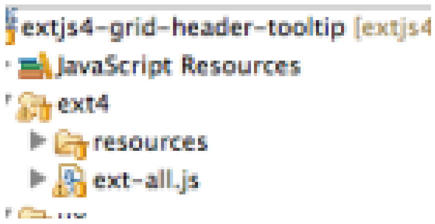
Importing an Excel Spreadsheet into an ExtJS DataGrid usi...


ExtJS: How to Export DataGrid to Excel

Introduction to iBatis (MyBatis), An alternative to Hiber...

Getting Started with iBatis (MyBatis): Annotations


ExtJS 4 File Upload + Spring MVC 3 Example


ExtJS 4: How to add Tooltip to Grid Header

Filed in: iBatis (MyBatis) | Tags: iBatis, Java, MyBatis, mySQL, stored procedure

Comments (6)

1. *Sunil Vashisth*

   Thanks for this Nice Article
   Very clear and concise

   [May 23, 2011 at 9:01 AM]

2. *Santosh*

   Hi,
   I checked your code https://github.com/loiane/ibatis-stored-procedures.
   But i want call **getTotalCity** procedure without using annotaion code(SqlMapper.java)
   Using Session object

   I am trying to do this :

   Session session = sqlSessionFactory.openSession(ExecutorType.SIMPLE,true);
   Param pobj = new Param();
   Param p = session.selectList("getTotalCity", pobj);
   System.out.prinln("Result" + p.getTotal());

   DEBUG [main] – ==> Executing: { CALL getTotalCity(?)}
   DEBUG [main] – ==> Parameters:
   Null pointer exception.

   Can you please help me out.

   [June 3, 2011 at 9:22 AM]

   - *Loiane*

     Hi Santosh,
     Please check the class StoreProcedureDAO.java that is within the source code.
     You will find how to call it in this class.

     [July 25, 2011 at 10:15 AM]

3. *Chi*

   Any luck with using the annotations to deal with results of a stored procedure call? Thanks!

   [August 29, 2011 at 7:15 PM]

   - *Loiane*

     Hi Chi,
     Not yet.
     Thanks

     [September 14, 2011 at 1:12 PM]

4. *Chi*

   I meant to post this earlier, but didn't get a chance. I was able to get the results from a stored procedure call. For reference, here is the code -

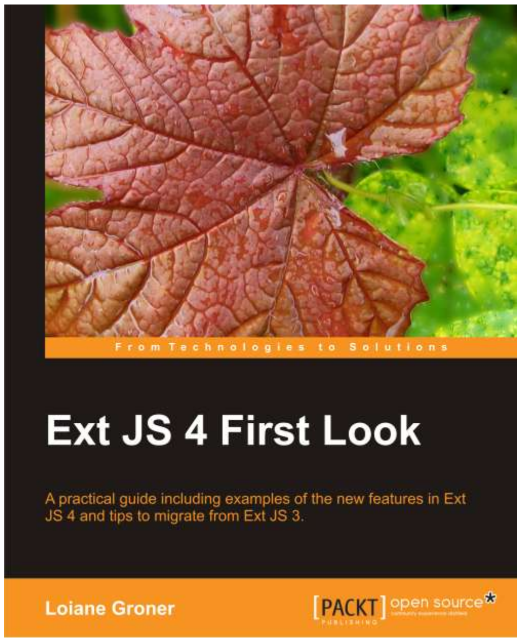
```
public interface XYZMapper {
  @Select(value = "{call xyz_storedproc( #{cId, jdbcType=NUMERIC, mode=IN} )}")
  @Results(
  {
     @Result(property = "severity", column = "Severity", javaType = String.class, jdbcType =
JdbcType.VARCHAR),
     @Result(property = "descr", column = "Descr", javaType = String.class, jdbcType =
JdbcType.VARCHAR)
  })
  @Options(statementType = StatementType.CALLABLE)
  public List execute(XYZParams xYZParams);

}
```

   [January 25, 2012 at 4:36 PM]

« [IBatis (MyBatis): Working with Dynamic Queries (SQL)]
[JAXB Custom Binding – Java.util.Date / Spring 3 Serialization] »

My Books!

RAW
Community Experience Distilled
Mastering Ext JS: RAW
Discover how you can take your skills with Ext JS to the next level
Loiane Groner
[PACKT] open source


From Technologies to Solutions
Ext JS 4 First Look
A practical guide including examples of the new features in Ext JS 4 and tips to migrate from Ext JS 3.
Loiane Groner
[PACKT] open source

Connect

Connect with us on the following social media platforms.

We're on Follow

+540

Buy me a coffee!

Do you like this blog? Feel free to donate 1 dollar - or any amount you want! :)

PayPal Donate

Categories

Select Category

Loiane Groner

- Contact
- Blog
- About Me
- ExtJS Plugins