

IBATIS - CREATE OPERATION

http://www.tutorialspoint.com/ibatis/ibatis_create_operation.htm

Copyright © tutorialspoint.com

To perform any Create, Write, Update, and Delete *CRUD* operation using iBATIS, you would need to create a Plain Old Java Objects *POJO* class corresponding to the table. This class describes the objects that will "model" database table rows.

The POJO class would have implementation for all the methods required to perform desired operations.

Let us assume we have the following EMPLOYEE table in MySQL –

```
CREATE TABLE EMPLOYEE (  
  id INT NOT NULL auto_increment,  
  first_name VARCHAR(20) default NULL,  
  last_name VARCHAR(20) default NULL,  
  salary INT default NULL,  
  PRIMARY KEY (id)  
);
```

Employee POJO Class

We would create an Employee class in Employee.java file as follows –

```
public class Employee {  
  private int id;  
  private String first_name;  
  private String last_name;  
  private int salary;  
  
  /* Define constructors for the Employee class. */  
  public Employee() {}  
  
  public Employee(String fname, String lname, int salary) {  
    this.first_name = fname;  
    this.last_name = lname;  
    this.salary = salary;  
  }  
} /* End of Employee */
```

You can define methods to set individual fields in the table. The next chapter explains how to get the values of individual fields.

Employee.xml File

To define SQL mapping statement using iBATIS, we would use <insert> tag and inside this tag definition, we would define an "id" which will be used in IbatisInsert.java file for executing SQL INSERT query on database.

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE sqlMap PUBLIC "-//ibatis.apache.org//DTD SQL Map 2.0//EN"  
  "http://ibatis.apache.org/dtd/sql-map-2.dtd">  
  
<sqlMap namespace="Employee">  
  
  <insert >  
    insert into EMPLOYEE(first_name, last_name, salary)  
    values (#first_name#, #last_name#, #salary#)  
  
    <selectKey resultClass="int" keyProperty="id">  
      select last_insert_id() as id  
    </selectKey>  
  </insert>
```

```
</sqlMap>
```

Here **parameterClass** – could take a value as *string, int, float, double*, or any class *object* based on requirement. In this example, we would pass Employee object as a parameter while calling *insert* method of SqlMap class.

If your database table uses an IDENTITY, AUTO_INCREMENT, or SERIAL column or you have defined a SEQUENCE/GENERATOR, you can use the <selectKey> element in an <insert> statement to use or return that database-generated value.

IbatisInsert.java File

This file would have application level logic to insert records in the Employee table –

```
import com.ibatis.common.resources.Resources;
import com.ibatis.sqlmap.client.SqlMapClient;
import com.ibatis.sqlmap.client.SqlMapClientBuilder;

import java.io.*;
import java.sql.SQLException;
import java.util.*;

public class IbatisInsert{
    public static void main(String[] args)throws IOException,SQLException{
        Reader rd = Resources.getResourceAsReader("SqlMapConfig.xml");
        SqlMapClient smc = SqlMapClientBuilder.buildSqlMapClient(rd);

        /* This would insert one record in Employee table. */
        System.out.println("Going to insert record.....");
        Employee em = new Employee("Zara", "Ali", 5000);

        smc.insert("Employee.insert", em);

        System.out.println("Record Inserted Successfully ");
    }
}
```

Compilation and Run

Here are the steps to compile and run the above mentioned software. Make sure you have set PATH and CLASSPATH appropriately before proceeding for compilation and execution.

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

You would get the following result, and a record would be created in the EMPLOYEE table.

```
$java IbatisInsert
Going to insert record.....
Record Inserted Successfully
```

If you check the EMPLOYEE table, it should display the following result –

```
mysql> select * from EMPLOYEE;
+----+-----+-----+-----+
| id | first_name | last_name | salary |
+----+-----+-----+-----+
| 1  | Zara      | Ali      | 5000   |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js