# How To Call Stored Procedure In Hibernate

Posted on February 15, 2010 , Last modified : August 30, 2012
By mkyong

In this tutorial, you will learn how to call a store procedure in Hibernate.

## MySQL store procedure

Here's a MySQL store procedure, which accept a stock code parameter and return the related stock data.

```
DELIMITER $$

CREATE PROCEDURE `GetStocks`(int_stockcode VARCHAR(20))
BEGIN
   SELECT * FROM stock WHERE stock_code = int_stockcode;
   END $$

DELIMITER ;
```

In MySQL, you can simple call it with a **call** keyword :

```
CALL GetStocks('7277');
```

## Hibernate call store procedure

In Hibernate, there are three approaches to call a database store procedure.

## 1. Native SQL – createSQLQuery

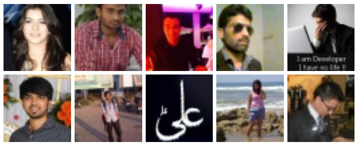
You can use **createSQLQuery()** to call a store procedure directly.

```
Query query = session.createSQLQuery(
	"CALL GetStocks(:stockCode)")
	.addEntity(Stock.class)
	.setParameter("stockCode", "7277");

List result = query.list();
for(int i=0; i<result.size(); i++){
	Stock stock = (Stock)result.get(i);
	System.out.println(stock.getStockCode());
}
```

## 2. NamedNativeQuery in annotation

Declare your store procedure inside the **@NamedNativeQueries** annotation.



Recent Posts

- Java - Convert IP address to Decimal Number
- Java and "& 0xFF" example
- Spring MVC - How to include JS or CSS files in a JSP page
- Eclipse - java.lang.OutOfMemoryError: Java heap space
- MongoDB - find all documents where an array / list size is greater than N

```
//Stock.java
...
@NamedNativeQueries({
        @NamedNativeQuery(
        name = "callStockStoreProcedure",
        query = "CALL GetStocks(:stockCode)",
        resultClass = Stock.class
        )
})
@Entity
@Table(name = "stock")
public class Stock implements java.io.Serializable {
...
```

Call it with **getNamedQuery()**.

```
Query query = session.getNamedQuery("callStockStoreProcedure")
        .setParameter("stockCode", "7277");
List result = query.list();
for(int i=0; i<result.size(); i++){
        Stock stock = (Stock)result.get(i);
        System.out.println(stock.getStockCode());
}
```

## 3. sql-query in XML mapping file

Declare your store procedure inside the “**sql-query**” tag.

```
<!-- Stock.hbm.xml -->
...
<hibernate-mapping>
    <class name="com.mkyong.common.Stock" table="stock" ...>
        <id name="stockId" type="java.lang.Integer">
            <column name="STOCK_ID" />
            <generator class="identity" />
        </id>
        <property name="stockCode" type="string">
            <column name="STOCK_CODE" length="10" not-null="true" unique="true" />
        </property>
        ...
    </class>

    <sql-query name="callStockStoreProcedure">
        <return alias="stock" class="com.mkyong.common.Stock"/>
        <![CDATA[CALL GetStocks(:stockCode)]]>
    </sql-query>

</hibernate-mapping>
```

Call it with **getNamedQuery()**.

```
Query query = session.getNamedQuery("callStockStoreProcedure")
        .setParameter("stockCode", "7277");
List result = query.list();
for(int i=0; i<result.size(); i++){
        Stock stock = (Stock)result.get(i);
        System.out.println(stock.getStockCode());
}
```

## Conclusion

The above three approaches are doing the same thing, call a store procedure in database. There are not much big different between the three approaches, which method you choose is depend on your personal prefer.

Tags : hibernate stored procedure

Popular Tutorials

Android Tutorial

JSF 2.0 Tutorial

Spring Tutorial

Maven Tutorial

Hibernate Tutorial