# 1、**使用hibernate调用不带参数返回值的存储过程方法**

|  |
| --- |
| public Boolean doInHibernate(Session session)      throws HibernateException, SQLException {     try{      Transaction tr = session.beginTransaction();      SQLQuery query = session.createSQLQuery("{call CommandAdapter(?,?,?,?)}");      query.setString(0,"ls");       query.setInteger(1, machineid);      query.setString(2, param);      query.setInteger(3, 0); //      query.executeUpdate();      tr.commit();           return true;     }catch(Exception e){      e.printStackTrace();      return false;     }finally{         session.close();     }    } |

# 2、**使用hibernate调用带参数返回值的存储过程方法**

### 解释：在调用带参数返回值的方法时，需使用到CallableStatement对象。因此，首先得设法获取一个Connection。

### 1、获取Connection 方法：

#### 1、session.connection();//此方法目前已被弃用

#### 2、SessionFactoryUtils.getDataSource(sessionFactory).getConnection()

不过我在使用SessionFactoryUtils.getDataSource(sessionFactory)时，一直获取的都是空值，为此参考SessionFactoryUtils的源代码直接利用ConnectionProvider对象来获取一个Connection实例。

## 2、代码

|  |
| --- |
| SessionFactory sessionFactory=this.getSessionFactory();   if (sessionFactory instanceof SessionFactoryImplementor){    ConnectionProvider cp = ((SessionFactoryImplementor) sessionFactory).getConnectionProvider();    try {          CallableStatement statement = cp.getConnection().prepareCall(                     "{call MachineDataQuery(?,?,?,?,?,?,?,?,?,?,?,?,?,?)}");       statement.setInt(1, machineId);     statement.setInt(2, jobId);     statement.setInt(3, partTypeId);     statement.setInt(4, compentId);     statement.setInt(5, opId);     statement.setInt(6, roleId);     statement.setInt(7, userId);     statement.setString(8, sd);     statement.setString(9, ed);     statement.registerOutParameter(10, Types.FLOAT);     statement.registerOutParameter(11, Types.FLOAT);     statement.registerOutParameter(12, Types.FLOAT);     statement.registerOutParameter(13, Types.FLOAT);     statement.registerOutParameter(14, Types.FLOAT);     statement.execute(); 执行完自动就有值了     System.out.println(statement.getFloat(10));     System.out.println(statement.getFloat(11));     System.out.println(statement.getFloat(12));     System.out.println(statement.getFloat(13));     System.out.println(statement.getFloat(14));    } catch (SQLException e) {     e.printStackTrace();    }finally{     cp.close();    }   } |