Atitit 调试工具模块 保存序列化map参数 mybatis调试sql selectid 查看mybatis真实sql

[1.1. 对象序列化功能 序列化为bytearr 文件等 1](#_Toc7268)

[1.2. Ide本身的序列化 失去类型信息 2](#_Toc23492)

[1.3. mybatis调试sql 最终使用map判断类型结合mybatsi文件获取实际sql 2](#_Toc13791)

[1.4. mybatis读取mapper配置的sql语句 使用xml类库 2](#_Toc5452)

## 对象序列化功能 序列化为bytearr 文件等

Apache lang3 的 SerializationUtils.serialize

**package** com.attilax.util;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.ObjectInputStream;

**import** java.io.ObjectOutputStream;

**import** java.io.Serializable;

**import** java.util.Map;

**import** java.util.UUID;

**import** org.apache.commons.lang3.SerializationUtils;

**import** com.alibaba.fastjson.JSON;

**import** com.google.common.collect.Maps;

**public** **class** serilizeUtil {

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

// **TODO** Auto-generated method stub

Map **m**=Maps.*newHashMap*();

m.put("kk", "vv");

SerializationUtils.*serialize* ((Serializable) m,**new** FileOutputStream( "c:\\logs\\mo2.txt"+UUID.*randomUUID*()));

Map **m2**=(Map) *serizGetObjFromFile*("c:\\logs\\mo2.txt");

System.***out***.println( JSON.*toJSONString*(m2));

}

## Ide本身的序列化 失去类型信息

**private** **static** String getValueByTypeinSql(Object value) {

**if**(value.getClass()==String.**class**)

**return** "'"+value.toString()+"'";

**if**(value.getClass()==Integer.**class**)

**return** value.toString();

**throw** **new** RuntimeException("getValueByTypeinSql:: value not jude type");

}

params="{aParam60=0, beh01=null, bbx09=1, vaf58=0, abc02=普通, sqltext=%葡萄糖%, aParamno=0, yiyuanId=-5, bce01=1, acf01=1, aParam1=0, bdp02=自费, offset=0, bckyf=0, bck01=108, aParam106=1}";

//

**private** **static** Map getM(String params) {

Map **m**=Maps.*newConcurrentMap*();

params=params.substring(1,params.length()-1); //de start end one char ..openclose char

String[] **entry\_arr**=params.split(",");

**for** (String **e** : entry\_arr) {

String[] **kva**=e.split("=");

m.put(kva[0], kva[1]);

}

**return** m;

}

## mybatis调试sql 最终使用map判断类型结合mybatsi文件获取实际sql

## mybatis读取mapper配置的sql语句 使用xml类库

C:\0wkspc\oploggerPrj\src\com\attilax\util\MybatisUtil.java

package com.attilax.util;

import java.io.IOException;

import java.sql.Connection;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.List;

import java.util.Map;

import org.apache.ibatis.session.SqlSession;

import org.apache.zookeeper.common.PathUtils;

import org.jdom.Document;

import org.jdom.Element;

import org.jdom.JDOMException;

import org.jdom.input.SAXBuilder;

import com.cnhis.cloudhealth.clidoctorweb.gzip.PathUtil;

import com.google.common.collect.Maps;

import cn.freeteam.util.MybatisSessionFactory;

public class MybatisUtil {

protected static Statement getStt() {

SqlSession session = getSqlSession();

Connection conn = session.getConnection();

Statement st = null;

try {

conn.setAutoCommit(true);

st = conn.createStatement();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

return st;

}

private static SqlSession getSqlSession() {

MybatisSessionFactory.CONFIG\_FILE\_LOCATION = "/com/attilax/db/mybatisutil/mybatis\_postgresql.xml";

SqlSession session = MybatisSessionFactory.getSession();

return session;

}

public static void main(String[] args) {

String sqlid="adviceSousuo\_kucui";

String s=getMybaticsCfgedSqlBysqlid("C:\\0wkspc\\clinical\\src\\main\\java\\com\\cnhis\\cloudhealth\\clinical\\clidoctor\\clischemedefine\\mapper\\CliSchemeDefineMapper.xml","adviceSousuo\_kucui");

System.out.println(s);

// String f="D:\\0workspace\\atiplat\_restapi\\src\\aaaPKg\\flow\_design\_cs.xml";

//

// Map m=(Map) serilizeUtil.serizGetObjFromFile("c:\\logs\\adviceSousuo\_kucui\_map\_8080f6ab-35b1-440f-b1b5-8c1b0ea2de32");

//

// Statement st = getStt(); // ini envi

//

// List li= getSqlSession().selectList(sqlid, m);

// System.out.println(li.size());

}

private static String getMybaticsCfgedSqlBysqlid(String f,String sqlid) {

// TODO Auto-generated method stub

SAXBuilder builder=new SAXBuilder(false);

Document doc;

try {

doc = builder.build(f);

} catch (JDOMException | IOException e) {

throw new RuntimeException(e);

}

Element books=doc.getRootElement();

List< Element> definitions\_eles=books.getChildren("select");

for (Element e : definitions\_eles) {

System.out.println(e.getAttribute("id"));

if(e.getAttribute("id").getValue().equals(sqlid))

return e.getText();

}

// Element process\_ele=books.getChild("process");

// System.out.println(process\_ele.getAttributeValue("deadlineLimit"));

System.out.println("--f");

return sqlid;

}

}