atitit.atiOrmStoreService 框架part2 dsl2sqlService

[---end 4](#_Toc16332)

package com.attilax.sql;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Set;

import aaaCfg.IocX4nodb;

import aaaDbManager.DbMetaService;

import com.attilax.coll.ListX;

import com.attilax.collection.listUtil;

import com.attilax.io.filex;

import com.attilax.ioc.IocXq214;

import com.attilax.jql.DataTypeUnicomMysql;

import com.attilax.lang.MapX;

import com.attilax.store.OrmStoreService;

import com.google.inject.Inject;

public class Dsl2sqlService {

public static void main(String[] args) {

//IocX4nodb.db=new ThreadLocal<>();

IocX4nodb.db.set("shopedb");

System.setProperty("apptype", "bet");

OrmStoreService ormSvr = IocXq214.getBean(OrmStoreService.class);

Dsl2sqlService js=IocXq214.getBean(Dsl2sqlService.class);

Map m=new HashMap();

// r.put("$table", "底单申请表");

// r.put("$db", "shopedb");

// r.put("$op", "merge");

// r.put("kwidi", "shopedb\_kwidi");

// r.put("img\_col", "shopedb\_imgcol");

m.put("$table", "wxb\_order");

// m.put("$tabletype", "view");

// m.put("$view\_store\_path","com/attilax/order");

m.put("$op", "insert");

m.put("good\_id","2005");

m.put("order\_id",filex.getUUidName());

// System.out.println( js.saveOrMeger(r));

System.out.println( js.toInsertSql(m));

//saveOrMeger

System.out.println("--f4");

}

@Inject

DbMetaService metaSvr;

String prikey;

public String saveOrMeger(Map req) {

String table = (String) req.get("$table");

if(table==null)

table = (String) req.get("table");

String op = (String) req.get("$op");

if(op==null)

op = (String) req.get("op");

prikey = metaSvr.getPrimaryKey(req);

if (req.get(prikey) == null || req.get(prikey).toString().trim().length() == 0)

return toInsertSql(req);

else

return update(req);

}

public void delete(Map req) {

// TODO Auto-generated method stub

}

public String update(Map req) {

return null;

// TODO Auto-generated method stub

}

public String query(Map req) {

return null;

// TODO Auto-generated method stub

}

@SuppressWarnings("all")

public String toInsertSql(Map req) {

String table = (String) req.get("$table");

if(table==null)

table = (String) req.get("table");

Map req\_flted = fltMap(req, table);

if(req\_flted.keySet().size()==0)

throw new RuntimeException("req\_flted field is empty ");

// calc\_cols4sql\_n\_valsArr(req\_flted);

prikey = metaSvr.getPrimaryKey(req);

SqlKvUtil sqlKvUtil = new SqlKvUtil();

sqlKvUtil.prikey=prikey;

String[] a=sqlKvUtil.toInsertSqlKVPart(req\_flted);

String sql = "insert into @tab@(@cols@)values(@vals@)";

sql = sql.replace("@tab@", table);

sql = sql.replace("@cols@", a[0]);

sql = sql.replace("@vals@", a[1]);

return sql;

}

Map colTypeMap = new HashMap();

// @

/\*\*

\* ccc

\*/

List<Map> getColumnsRzt;

private Map fltMap(Map req, String table) {

List<Map> li = metaSvr.getColumns(req);

for (Map col\_map : li) {

colTypeMap.put(col\_map.get("COLUMN\_NAME") , col\_map.get("TYPE\_NAME"));

// colTypeMap.put("TYPE\_NAME", );

}

Set<String> cols = metaSvr.getColumns\_set(req);

Map ret = MapX.reduce(req, cols);

return ret;

}

public String dsl2sql(Map m) {

if(m.get("$op").equals("insert"))

return toInsertSql(m);

return null;

}

}

# ---end