|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 修改内容 | 日期 |
| V0.1 | 刘昕 | 创建 | 2011/5/19 |

1. 概述

通过4个接口来实现excel和database之间的数据相互导入导出.

**com.zjhcsoft.exceldb.support.IexcelReader excel读取接口**

##### com.zjhcsoft.exceldb.support.IexcelWriter excel写入接口

##### com.zjhcsoft.exceldb.support.IdatabaseReader database读取接口

##### com.zjhcsoft.exceldb.support.IdatabaseWriter database 写入接口

* 1. excel导入数据库

使用IexcelReader将excel的数据根据配置文件配置的信息以Map<String, Object>的形式读取到内存(其中key 为数据库列名 value单元格的数据, 一个map代表一行数据).再使用IdatabaseWriter将读取到的数据写入到数据库中.

**public** **void** testWrite() **throws** IOException {

// 1.读取excel文件

File file = **new** File("E:\\excel\\myexcel.xls");

InputStream is = **null**;

// 2.读取配置文件

XslDb xslDb = MappingLoader.*getXslDb*("DATA");

**try** {

is = **new** FileInputStream(file);

// 3.创建excelReader

IExcelReader excelReader = **new** PoiExcelReader(is, xslDb);

// 4.创建databaseWriter

IDatabaseWriter databaseWriter = **new** JdbcDatabaseWriter();

// 5.迭代读取excel的每一行

**while** (excelReader.hasNext()) {

Map<String, Object> map = excelReader.next();

// 在这里可以添加额外的业务逻辑

// 6.数据写入database

databaseWriter.write(map, "tbl\_data");

}

} **finally** {

is.close();

}

}

* 1. 数据库导出excel

原理与excel导入数据库相似,使用IdatabaseWriter将database中的数据以List<Map<String,Object>>的形式读取到内存中(如果有配置文件的情况下key为Excel的列,如A,B,C…,如果没有配置文件的情况下key为数据库的列名.value对应的数据, 一个map代表一行数据),再使用IexcelWrite将数据写入excel.

**public** **void** testWrite() **throws** IOException {

// 1.创建要写入的Excel文件

File file = **new** File("E:\\excel\\test.xls");

**if** (!file.exists()) {

file.createNewFile();

}

// 2.读取配置文件

XslDb xslDb = MappingLoader.*getXslDb*("DATA");

OutputStream os = **null**;

**try** {

os = **new** FileOutputStream(file);

// 3.创建excelWriter

IExcelWriter poiExcelWriter = **new** PoiExcelWriter(os);

// 4.创建databaseReader

JdbcDatabaseReader databaseReader = **new** JdbcDatabaseReader();

// 4.或者初始化databaseReader

ibatisDatabaseReader.setCurrentRowNum(0);

ibatisDatabaseReader.setCountSql("find\_account\_for\_count\_sql");

ibatisDatabaseReader.setQuerySql("find\_account\_for\_page\_sql");

ibatisDatabaseReader.setXslDb(xslDb);

// 5.设置配置文件(可选操作)

databaseReader.setXslDb(xslDb);

List<Map<String, Object>> all = **new** ArrayList<Map<String, Object>>();

// 6.迭代数据库读取数据

**while** (databaseReader.hasNext()) {

List<Map<String, Object>> list = databaseReader.next();

all.addAll(list);

}

// 7.将读取到的数据写入excel

poiExcelWriter.write(all);

} **finally** {

os.close();

}

}

1. 配置文件

<mapping>

<dbxsl id=*"ASS\_KPI\_ACCEPT\_BASE\_DATA"* ignore=*"1"* sheets=*"0"* idGenerator=*"sequence"*

idGeneratorValue=*"SEQ\_ASS\_KPI\_ACCEPT\_BASE\_DATA"* idColumn=*"ID"*>

<col dbcol=*"CITY\_ID"* xslcol=*"A"* title=*"CITY\_ID"* />

<col dbcol=*"UNIT"* xslcol=*"B"* title=*"UNIT"* />

<col dbcol=*"TYPE"* xslcol=*"C"* title=*"TYPE"* />

<col dbcol=*"ACTION"* xslcol=*"D"* title=*"ACTION"* />

<col dbcol=*"SUB\_ACTION"* xslcol=*"E"* title=*"SUB\_ACTION"* />

<col dbcol=*"STATUS"* xslcol=*"F"* title=*"STATUS"* />

<col dbcol=*"AMOUNT"* xslcol=*"G"* title=*"AMOUNT"* />

</dbxsl>

</mapping>

id:java中读取配置文件的标识符

ignore:读取excel是忽略的行数

idGenerator:主键生成策略(目前支持sequence和UUID)

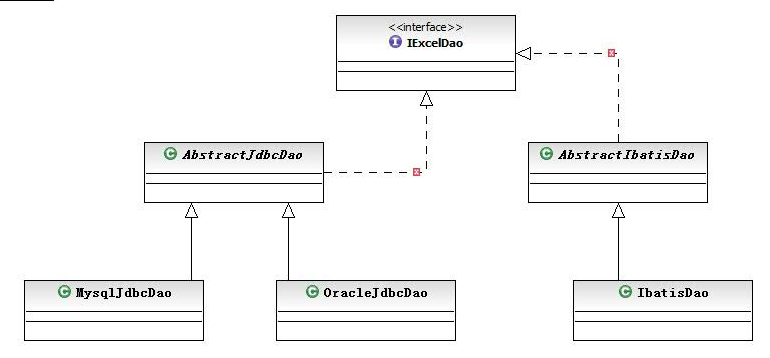
idGeneratorValue:生成主键的参数(如果是sequence生成策略为sequence的名称)

idColumn:数据库中id列的列名

col:数据库列名和excel列名的映射

1. Database操作

由于兼容不同的数据库和不同的数据库操作框架(Ibatis,JDBC,SevenStar)IdataBaseRead类Dao继承结构如下:



数据库操作类结构如下:

