文件上传与下载

注：文件上传与下载相关类全部在包com.yonyou.bq.file下，比较关键的类如下：

1、FileSourceController

与前端对接入口类，用于接受用户请求，文件上传，最佳，替换等相关请求

方法1：将文件上传到临时目录，并返回文件模型给前端，文件模型BqFile包含BqSheet等信息，每一个BqSheet包含很多个SheetColumn列对象

**public** String parseFileSource(

@RequestParam("file") MultipartFile mFile,

HttpServletRequest request)

方法2：创建文件数据源，将临时文执行入库处理。

@RequestMapping(value = ***FILENEW***+"/{dirUid}", method = RequestMethod.***POST***, produces = "application/text; charset=utf-8")

@ResponseBody

**public** String createFileSource(@PathVariable("dirUid") String dirUid,

@RequestBody String body){

入库逻辑如下：

1. 判断文件属于csv还是Excel，不同的文件类型调用不同的处理逻辑
2. Csv处理逻辑需要将第一行当做列名，遍历一遍csv解析其列的数据类型，只要某一列有一个数据无法转换成double，就将此列设置为字符串类型，否则为数字类型。
3. Excel处理逻辑需要将首行做列名，第二行取数据类型。

2、FileInputTask

该类封装了Excel，Csv的入库前处理，Csv调用DBInsertFilter进行入库，而Excel调用ExcelDbInsertFilter进行入库，而ExcelDbInsertFilter底层还是DbInsertFilter，总之核心都是调用DBInsertFilter进行入库。

Excel执行入库时，Excel07解析速度较慢，故需要开启并发解析，即启动多个线程解析Excel，进程数最大为6，如果低于6，取Sheet数量，这样可以实现Sheet间并发入库。

核心方法：startCSV()

完成Csv入库逻辑，需要调用DBInsertFilter，根据BqSheet模型创建表，然后通过JDBC完成入库逻辑。

核心方法：start03

完成Excel03的入库逻辑，由于Excel033解析速度特别快，故没有启动并发，直接调用ExcelDbInsertFilter执行入库。

核心方法：start07

这里完成了多个线程的创建和同步等操作，每一个线程的处理逻辑和Excel03类似，结尾处需要多个线程之间做join，以实现多线程之间的同步。

1. FileTempUtils

文件上传到临时目录的工具类，该类可以将文件写入指定临时目录，可以创建临时上传和下载路径等，该类用于将文件上传到tomcat本地临时目录，并且将指定的文件写入到临时下载目录等

1. Excel03Reader

Excel03版解析器类，用于解析Excel03版数据，解析方式是基于事件流的方式进行解析。只需要定义相关的filter，即可监控相关的时间，在该类中已经做相关数据类型的处理，无需关注内部逻辑

1. Excel07Reader

Excel07版解析器类，用于解析Excel03版数据，解析方式是基于事件流的方式进行解析。只需要定义相关的filter，即可监控相关的时间，在该类中已经做相关数据类型的处理，无需关注内部逻辑

1. CsvReader

Csv解析器类，为了将Csv的处理逻辑和Excel相同，此处又按照Excel处理逻辑进行对Csv解析进行了封装。

1. ReaderFactory

Reader工厂类，封装CSVReader，Excel03Reader，Excel07Reader三个reader类，可以根据文件类型，选择创建不同的reader

1. ExcelDbInsertFilter

执行Excel入库逻辑，处理规则如下:

每一个Sheet页面取第一行，根据第一行，创建SheetColumn，根据sheet页面的第二行数据设置列类型，即：数值，日期，字符串等

当读取到sheet页面的第二行的时候，列名称和列类型都以确定，执行建表逻辑。Varchar类型默认长度为4000,当执行完入库逻辑以后，再将列长度修改为合适的值。Kudu没有varchar2，而是用String，所以无需修改列长度。

核心方法：filter

该方法为过滤方法，过滤Excel文件时，文件的每一个Sheet页的每一行记录都调用此方法，该方法会将Excel的数据做一些预处理，然后调用DBInsertFilter的filter方法，以完成入库。

1. DbInsertFilter

Jdbc入库核心类，该类封装了jdbc数据入库逻辑，以过滤器方式进行入库，使调用十分简洁。该类包含了sql，connection，commitBytesSize，commitSize基本方法。

Sql用于指定插入数据库的sql语句

Connection为目标数据库的连接

commitBytesSize，用以确定提交字节数，当已处理的字节数大于等于该字节数时，将会执行提交操作。

commitSize，该方法用于确定提交数，当处理的记录数达到此数量时，也会执行提交操作，该参数与commitBytesSize同时控制提交，二者有一个满足，即执行提交操作。

Concurrent变量，用于控制是否进行并发提交，如果为true，则新启动一个线程进行提交操作，故提交不阻塞主线程的时间。

核心方法：filter

该方法为核心过滤方法，记录处理行数和处理字节数，并将处理的数据添加到statement中，并调用addBatch，当处理字节数和处理记录数有一个达到限制时，则执行入库操作。

1. KuduInsertFilter

继承于DBInsertFilter类，重写其filter方法，调用kuduAPI执行入库

KUDU API入库简要逻辑如下：

**public** **static** **void** doInsert(String tableName) **throws** Exception{

KuduClient client = **new** KuduClient.KuduClientBuilder("172.20.17.213:7051").build();

KuduTable table = client.openTable(tableName);

KuduSession session = client.newSession();

session.setFlushMode(FlushMode.***MANUAL\_FLUSH***);

**try** {

**for**(**int** i=0; i<10; i++){

Insert insert = table.newInsert();

PartialRow row = insert.getRow();

row.addLong(0, i);

row.addString(1, "1");

session.apply(insert);

}

}

**catch**(Exception e){

e.printStackTrace();

}

**finally** {

session.flush();

client.close();

}

}

1. 模型类BqFile

该类用于存储Excel模型，该模型包含多个BqSheet对象，存于数组中。目前而言，该模型只用于数据交互，不用于数据存储。

1. 模型类BqSheet

该模型包含很多SheetColumn，用该模型可以完整的描述一张表的信息，可以用于该对象进行建表操作。

1. ISheetService和ISheetMapper

BqSheet模型维护服务，用于维护BqSheet模型的增删改查等相关操作

14、SheetColumnMapper

用于维护SheetColumn的增删改查等