**南大先腾**

**数据交换平台2.0**

**技术白皮书**

2014年5月

1. 产品概述

数据交换平台主要作用是数据同步和数据集成。数据同步是指保持两个或者多个业务系统之间共享数据的一致性和完整性。数据集成是指将一个不同业务系统中的数据经过清洗、整理后汇总到一个数据中心，这样以便于数据共享、分析、挖掘。

1. 基本概念
   1. 数据角色分类

在数据交换平台中涉及的数据角色包括以下几类：

**源数据**：数据交换过程中提供数据的一方，一般是一个业务系统或者业务系统数据库。

**目标数据**：数据交换过程中接收数据的一方，本平台中为数据库、指定格式文件或Webservice接口。

**配置数据：**用来定义平台上执行的数据交换任务。包括源数据、目标数据、数据交换规则、交换任务等数据。

**中心数据库：**在数据集成中的概念，在数据交换中的角色等同于一般的应用数据库，只是没有具体的业务与之对应。这个数据库可以作为企业的数据中心供统计分析适用；也可以作为数据交换过程中的临时数据库或者中转数据库使用。

* 1. 数据交换分类

数据交换按照交换的时间点可以分为定时交换和实时交换，按照源数据和目标数据是否可以直接通讯分为在线交换和离线交换。

|  |  |  |
| --- | --- | --- |
| 交换类别 | 在线交换 | 离线交换 |
| 定时交换 | 通过定时任务交换数据 | 需要人为干预 |
| 实时交换 | 通过接口调用实时交换 | —— |

* 1. 数据交换过程

数据交换过程就是从源数据中读取数据按照交换平台配置的对应关系写入到目标数据中，在交换的过程的各个时间点可以通过事件的形式触发交换平台定义的操作。

源数据的形式可以是：对一个业务数据库的查询操作、一个离线文件、接口中的数据内容参数。目标数据只能是一个数据库。系统提供的不同的交换类别的交换过程参见下表。

|  |  |  |
| --- | --- | --- |
| 交换过程 | | 过程描述 |
| 直接连接交换 | | 定时在线交换，系统定义一个查询语句作为数据源，根据映射关系保存到目标数据库中。 |
| 离线交换 | 数据导出 | 定时或者人工触发，以查询语句作为数据源，根据规定的格式写成离线文件。 |
| 数据导入 | 人工触发或者定时监控目录，以离线文件作为数据源，根据规定的映射关系写入目标数据库中。 |
| 实时交换 | | 通过接口调用实时交换，通过调用交换平台中的WebService接口提供数据源，并根据规定的映射关系写入目标数据库中。 |

1. 平台架构

数据交换平台是基于J2EE开发的B/S结构的应用程序。相关的技术：JDBC、XML、WebService。

1. 功能设计
   1. 业务数据定义

系统中定义的数据有两类，一类是直接可以访问的数据库，还有一类是WebService接口。

* + 1. 定义业务数据库连接

在交换平台中定义业务数据库，供数据直接交换的源数据或者目标数据使用。定义数据库连接需要录入以下内容：

* 数据库名称，这个可以自己起并作为主键。
* Jdbc连接url。
* 数据库用户名。
* 数据库用户密码，用加密的方式存储。

数据交换平台支持的数据库包括：Oracle、SqlServer、DB2这三种。

* + 1. 定义WebService服务接口

数据交换平台中的WebService接口规范参见第7章。平台中的通过接口数据交换都是采用推送方式来实现的，都是数据提供方发起请求的。平台提供的WebService服务供业务系统调用向平台提供交换数据，同样业务系统需要接受数据就要实现WebService供平台调用。

WebService接口定义需要录入以下内容：

* 服务名称，这个可以自己起并作为主键。
* WebService服务的url。
* 用于验证的用户名。
* 对应验证用户的密码，用加密的方式存储。
  1. 关系与操作定义
     1. 定义数据对应关系

定义源数据库和目标数据库之间数据对应关系，需要定义的内容：

* 源数据库中的一条查询语句。
* 目标数据库中的一个表。
* 源数据查询语句的字段和目标数据库中的表字段对应关系。
* 交换触发器，数据同步过程中需要的处理。参见6.1.4触发器。
  + 1. 定义交换数据

带交换数据可以用于离线文件导出，也可以作为调用业务系统WebService接口的数据。交换数据需要定义的内容包括：

* 源数据库中的一条查询语句。
* 查询语句中各个字段的保存方式和格式。
* 接收方处理标识，这个标识供接收方识别数据内容以判断如何处理。
  + 1. 定义数据处理方式

平台通过接收方处理标识来识别处理方式。平台提供的处理方式有两种，写入数据库和调用业务系统WebService接口发送数据。

写入数据定义内容包括：

* 业务数据库中对应的表。
* 字段对应关系。
* 写入相关触发器。参见6.1.4触发器中所有针对目标数据库的触发器。

调用WebService接口：

* 服务器名，通过这名称关联4.1.2中定义的业务系统WebService接口。
  1. 定时任务定义
     1. 定义数据库直接交换任务

一个直接交换任务包括一个或者多个数据对应关系（4.2.1中定义），并指定一个执行时间计划。这个定时任务也可以手动执行。

* + 1. 定义基于WebService接口交换任务

一个基于WebService接口交换任务包括一个或多个接收方处理标识为调用WebService接口的交换数据（4.2.2中定义），并指定一个执行时间计划。这个定时任务也可以手动执行。

* + 1. 定义数据导出任务

一个数据导出任务包括一个或多个交换数据（4.2.2中定义）；忽略接收方处理标识的类别，这个接收方处理标识是给数据导入时失败的。并指定一个执行时间计划。这个定时任务也可以手动执行。

无论有多少个交换数据，数据导出的都是一个zip文件；参见7.1数据格式。

* 1. 交换与日志
     1. 手动执行任务

在交换平台中可以手动执行4.3.1中定义的三类任务。

* + 1. 数据导入

数据导入的文件为数据交换平台导出的文件，或者符合相关格式的文件。数据导入主要用于网络物理隔离的场景中。

上传离线文件到数据交换平台服务器上，交换平台解压并读取文件中的数据，根据离线文件中的接收方处理标识将文件中的数据写入到目标数据库或者调用业务系统的WebService接口将数据提交给业务系统。

这里有一个关键点接收方处理标识，在数据交换平台中必须有对应的数据处理方式（4.2.3中定义）。

* + 1. WebService接口服务

业务系统可以调用交换平台的WebService接口服务，向交换平台提供数据，交换平台根据数据中的接收方处理标识将文件中的数据写入到目标数据库或者调用业务系统的WebService接口将数据提交给业务系统。和数据导入一样数据交换平台中必须有对应的数据处理方式。

* + 1. 日志中心
       1. 日志内容

系统中所有的交换操作，包括4.4.1、4.4.2、4.4.3和自动执行的任务都会记录日志，日志内容包括：

* 执行的开始时间、结束时间。
* 执行方式：自动、手动、操作人员（自动为系统）。
* 执行结果：成功多少条、失败多少条。
* 失败数据记录，如果失败了会记录失败的数据，供跟踪和补录。
  + - 1. 日志查看

平台提供多种方式查看日志：

* 按照任务查看；查看某个任务历史交换情况。
* 按照时间查看；查看某个时间段，比如一天内的所有交换情况。
  + - 1. 日志统计

对交换结果进行统计分析，比如：

* 按时间段、任务统计数据交换总数、成功数、失败数。

1. 应用场景
   1. 基于数据库的交换

基于数据库的交换是在应用系统提供数据的读写权限的基础上执行的。

* + 1. 直接相连的定时交换

在数据交换平台中定义源数据和目标数据之间的对用关系，并创建定时任务。数据交换平台定时的执行数据同步。参见第6章直接连接交换。

* + 1. 物理隔离的离线交换

离线交换需要分两步，数据导出和数据导入。在物理隔离的系统中，需要在隔离的网络中分别安装数据交换平台。在源数据所在的交换平台上定义数据导出对应关系，在目标数据所在的交换平台上定义数据导入对应关系。这两个对应关系是通过XML格式文件和（？）来耦合的。参见第7章的数据导出和数据导入。离线数据文件需要人工的移转。

* 1. 基于接口的交换

在应用系统不允许直接访问数据库的情况下可以通过实现WebService接口的方式来实现数据交换。

* + 1. 应用作为数据提供方实时交换

应用程序实现WebService接口规范的客户端，通过调用交换平台的WebService接口向交换平台提供数据，数据交换平台根据mapinfoid来查找对应的映射关系，并将数据保存到对应的系统或者数据库中。

* + 1. 应用作为数据接收方实时交换

应用程序实现WebService接口规范的服务端，供交换平台调用。应用程序将交换平台提供的数据写入到应用系统数据库中。

* 1. 组合交换
     1. 基于接口的实时分发数据

提供数据的应用程序通过调用平台的WebService接口向平台提供数据，数据交换平台根据对应的映射关系可以同时调用多个目标应用程序的WebService服务向多个目标应用程序实时分发数据。

* + 1. 多应用系统集中离线交换

数据导出时根据数据对应关系进行导出的，一次导出可以执行多个对应关系，这些对应关系可以来自不同的应用系统。同样数据导入可以将数据导入到多个目标应用系统中。这样在隔离的两个网段中的多个应用系统数据可以通过一次导出和一次导入集中交换。

* + 1. 通过中转数据库集中离线交换

在应用系统数据库不允许或者不能直接访问时，如果需要进行离线交换可以通过中转数据库对数据进行中转。源应用系统通过调用交换平台的WebService接口向平台提供数据，平台将数据写入到中转数据库中。然后定时/手动进行数据导出和数据导入，实现离线交换。

注：离线交换都是定时/手动的无法做到实时交换。

1. 直接连接交换

直接连接交换就是数据交换平台可以同时连接到**源数据**和**目标数据**，数据交换平台根据用户配置的交换任务按用户指定的计划定时运行。

* 1. 交换任务集

交换任务集是数据交换平台调度的基本单位。一个交换任务包括多个有序的数据交换任务。数据交换任务集的相关属性参见下图：



 交换任务集的属性包括以下几类：

**基本信息**：任务名称、任务描述、创建时间、创建人员、是否启用；

**运行信息**：执行定时器、上次执行时间、下次执行时间。

**数据交互任务列表**：这个是数据交换的实际内容。

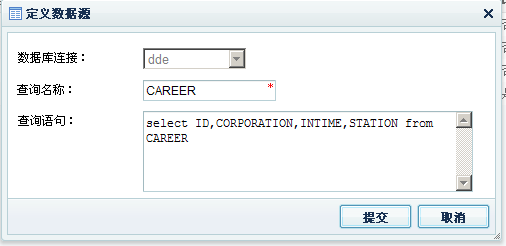
* + 1. 数据交换任务

一个数据交换任务对应一个源数据库中的查询语句和一个目标数据库中的表。数据交换任务有以下属性：数据源、数据目标、字段对应关系、交换触发器。

* + 1. 数据源和数据目标

数据源对应源数据库中的一条查询语句，这个查询语句可以简单也可以是复杂。数据目标对应目标数据库中的一个表。通常源数据库和目标数据库为不同的数据库，但是交换平台并没有做这样的约束。

数据源的定义见下图，数据目标的定义更简单点。



* + 1. 对应关系

数据源查询语句中的字段和数据目标表中的字段对应关系可以自行调整。数据源中的字段不一定都要对应到数据目标中，可以有多余的字段，这些字段内容可以用在触发器的执行过程中。



* + 1. 触发器

在数据交换前、交换后、失败时都可以触发一些动作，每一个动作都可以对应一个DML语句或者存储过程。触发器有以下属性:

**触发器类别**

|  |  |  |
| --- | --- | --- |
| **触发器名称** | **代码** | **描述** |
| 行触发器 | L | 迁移时对每一条数据执行一遍 |
| 表触发器 | T | 数据迁移前或者 完成时执行，一个对应关系只执行一次 |

**触发器时机**

|  |  |  |
| --- | --- | --- |
| **触发器时机名称** | **代码** | **描述** |
| 交换前 | B | 交换前执行触发语句 |
| 交换后 | A | 交换后执行触发语句 |
| 交换失败后 | E | 交换失败后执行触发语句 |

**执行对象**

|  |  |
| --- | --- |
| **执行对象名称** | **代码** |
| 数据源 | S |
| 数据目标 | D |

**执行顺序**

当相同类别、事件、对象的触发器有多个时，这些触发器都会得到执行，但是要给他们设定一个先后顺序。

* 1. 交换调度和日志
     1. 数据交换流程

数据交换步骤见下表，每一步骤包括其中的每一小步都是可选的。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序** | **步骤** | **定义方式** | **可能有的操作说明** | **运行方式** |
| 1 | 导出前整理数据 | 数据源交换前迁移级触发器 可以有多个，按顺序执行 | 1，检查带导出的数据是否符合相应的业务规则，对不符合的数据做上相应的标记 2，对符合导出条件的数据置上相应的标记，以锁定记录 | SQL语句 存储过程 |
| 2 | 导入前准备 | 数据目标交换前迁移级触发器 可以有多个，按顺序执行 | 1，记录导入前状态，比如：初始化导入计数 2，初始化表格；比如：需要新建表格，或者清除表格中的旧数据 | 存储过程 建表脚本 |
| 3 | 数据查询 | 源数据查询语句 | 1，查询数据 | 程序 |
| 4.1 | 数据导入前 | 目标数据交换前行级级触发器 可以有多个，按顺序执行 | 1，处理表格中可能的冲突的数据，比如：删除或者转移相同主键的记录 | SQL语句 存储过程 |
| 4.2 | 数据导入 |  | 1，导入数据，可以是insert 或者 update 2，记录导入数据迁移过程中的信息；包括：计数信息和错误信息 | 同上 |
| 4.3 | 数据导入后 | 数据目标交换后行级级触发器 可以有多个，按顺序执行 | 1，对记录进行相应的处理；比如处理异构数据，可以调用相应的存储过程 | 同上 |
| 4.4 | 数据导出后 | 数据源交换后行级级触发器 可以有多个，按顺序执行 | 1，对成功导出记录已导出状态 | 同上 |
| 4.5 | 导出出错源数据操作 | 数据源交换失败时行级级触发器 可以有多个，按顺序执行 | 1，在源数据库中 执行条语句，一般可以更改同步状态和错误消息 | 同上 |
| 4.6 | 导入出错目标数据操作 | 数据目标交换失败时行级级触发器 可以有多个，按顺序执行 | 1，在数据目标库中 执行语句，一般记录同步错误信息 | 同上 |
| 5 | 所有数据导出后 | 数据源交换后迁移级触发器 可以有多个，按顺序执行 | 1，在源数据库中记录本次导入的状态；比如：导入的数量、成功的数量、失败的数量、耗时等等 | 同上 |
| 6 | 所有数据导入后 | 数据目标交换后迁移级触发器 可以有多个，按顺序执行 | 1，在目标数据库中记录本次导入的状态；比如：导入的数量、成功的数量、失败的数量、耗时等等 | 同上 |

**注**：其中4.1~4.6是针对源数据中的每一条数据都执行一次的循环操作。（问：类似“日志”（主要是前后的处理以及失败记录）性质的数据是分别存放在源数据和目标数据中，还是统一放在目标数据中或者交换平台自带的数据库，从上面描述看，是分别存放，我个人觉得统一存放在目标数据库或者交换平台自带的数据库）

* + 1. 交换调度

调度数据交换平台中的交换任务有两种方式：

1. **手动调用**；操作人员登录数据交换平台在界面上手动调度。
2. **自动执行**；数据交换平台根据**交换任务集**的**任务执行定时器**属性中的表达式和Quartz的cron表达式基本一致，只是数据交换平台的定时器只精确到分钟，所以缺少秒部分的表达式。
   * 1. 日志与失败数据查看

 系统每执行一次交换调度，无论手动调用还是自动执行，都会记录交换日志。日志内容保存调度开始时间、结束时间、调度方式、调度人员。处理基本日志还包括交换任务集中的每个交换任务的成功交换记录条数和失败记录条数。针对失败的数据还会将失败的数据内容记录和导致失败的原因记录下来，以便用户查询和跟踪。



1. 其他形式交换

其他形式的交换都是通过离线文件、消息文本或者参数来传递数据的，这就要有一个数据格式。

* 1. 数据格式
     1. 数据内容

**交换元数据（加上编号）**

* 事务属性，整体作为一个事务，还是每一条数据作为一个事务。默认每一条数据作为一个事务。
* 从属表属性：字表单独一个事务还是和主表中的记录一起作为一个事务。默认从属于主表作为一个事务。（这句话不太懂）
* 主键冲突是覆盖（更新）还是报错。默认更新。
* 执行数据录入前所做的准备工作，一组数据库更新语句、或者建表操作。这部分可以没有。
* 回执数据内容。这部分默认记录成功条数，失败条数和失败数据内容。

注：这部分内容全部有默认值，也就是说这部分内容可以缺省。

**数据内容包括：**

* 表；内容包括：表名、表的元数据（字段名、字段类型、lob字段值存储方式）、和其他表的关系。
* 表中的记录；每一条记录中每字段一个元素，名称对应字段名内容对应字段值。

**LOB字段的处理：**

数据交换平台对lob的处理有两种方式：

* CDATA 区段；clob指端直接放入其中，blob字段通过base64编码后放入cdata区段中。
* 文件形式；每个lob字段一个文件，在记录中记录文件名，一般这个文件名可以和记录的主键相同。使用这种方式必需把xml文件和这些lob文件压缩都一个zip文件中。
  + 1. 数据格式

**文件格式**

离线数据文件是一个XML文件，可以每个表一个文件也可以多个表一个文件。Lob字段可以单独存放，一个字段一个文件，也可以经过base64编码后和其他字段一起放在项目了文件中。所以一次导出可能有多个离线文件，为了传输方便提高效率对离线文件进行打包。压缩包中有一个主文件一般为 导出任务名.xml，如果表的数据单独存放还包括多个表数据文件，以表名.xml命名；压缩文件中可以包括lob字段的文件，这些lob文件可以有自己的子目录，一般子目录的层次为表名\字段名，当然用户也可以定义任意的层次结构，和xml文件中字段值的描述一致皆可。

压缩算法采用java.util.zip中的通用zip算法。

**主XML数据格式(DTD)**

*<!DOCTYPE exchange [  
 <!ELEMENT exchange (metadata?,data)>  
 <!ATTLIST exchange id CDATA #REQUIRED>  
   
 <!ELEMENT metadata (pretreatment?,post\_processing?,receipt?)>  
 <!ELEMENT pretreatment (sql\*,call\*)>  
 <!ELEMENT post\_processing (sql\*,call\*)>  
 <!ELEMENT receipt ANY>  
   
 <!ELEMENT data (table+)>  
 <!ELEMENT table (row\*)>  
 <!ATTLIST table name CDATA #REQUIRED>****<!-- mapinfoid 为交换平台中定义的数据写入对应关系，这个很重要 ,没有这个属性系统将无法导入 -->*** *<!ATTLIST table mapinfoid CDATA #REQUIRED>  
 <!--表可以是一个单独文件存储，也可以在这个文件中 -->  
 <!ATTLIST table store (infile|embedded ) "embedded">  
 <!ATTLIST table sql CDATA #IMPLIED>  
 <!ATTLIST table transaction (each\_row|whole) "each\_row">  
 <!ATTLIST table conflict (replace|abandon) "replace">  
   
 <!ELEMENT row (item+,sub\_table\*)>  
 <!ELEMENT item (#PCDATA)>  
 <!ATTLIST item name CDATA #REQUIRED>  
 <!ATTLIST item type (text|date|timestamp|number|blob|clob) "text">  
 <!-- 如果type为blob、clob这个字段才有效 -->  
 <!ATTLIST item store (infile|embedded ) "embedded">  
 <!-- type 为date是 这个format为日期的格式，默认为"YYYY-DD-MM"； type为blob时这个格式为 plain 或则base64，clob默认为 plain， blob默认为base64 -->  
 <!ATTLIST item format CDATA>  
   
 <!ELEMENT sub\_table (row\*)>  
 <!ATTLIST sub\_table name CDATA  #REQUIRED>  
]>*

**表数据XML格式(DTD)**

参见上面的 ***table*** *格式。*

* + 1. 离线文件格式
  1. 数据导出

 离线文件交换用户不能直接联通的两个数据库之间的数据交换。这时必需在两边都要安装数据交换平台，在源数据库那边通过数据交换平台导出离线数据文件，在目标数据库那边通过数据交换平台导入离线数据文件。

导出时每个table对应的是一个查询语句，并且设定mapinfoid属性，这个属性用于和导入的对应关系进行耦合。设定导出内容时，除了给出查询语句还要给出各个字段的格式和存储形式，比如日期要规定格式，lob字段要给出是否进行base64编码，是否要单独存储为一个文件。

一个导出任务可以包括多个表，导出任务可以人工执行也可以定时执行，执行完成后会把本次导出相关的所有文件打包成一个zip文件。

导出时的对应关系定义和数据直接连接交换中的定义类似。

* 1. 数据导入

数据导入，就是将离线文件中所有的数据导入到目标数据库中。其中离线文件中table的mapinfoid属性必需在映射关系中设定好，否则不能导入成功。映射关系中自动数据文件中的表导入到目标数据库的对应表中，并制定字段的对应关系。这个和数据直接连接交换中的定义类似。

* 1. WebService交换服务

 WebService服务提供了实时交换的方案。业务可以通过调用交换平台的WebService接口将数据传递给交换平台。接口中的数据是以XML格式上传的，格式和数据导出中的表数据格式一致。同样根据mapinfoid来找对应的映射关系，保存到对应的表中。WebService作为数据实时交换的技术方案，它也是双向的，应用程序可以通过WebService向交换工具提供数据，同时也可以提供相同的WebService来接收数据。

* + 1. 用WebService提供数据

应用程序实现WebService接口规范的客户端，通过调用交换平台的WebService接口向交换平台提供数据，数据交换平台根据mapinfoid来查找对应的映射关系，并将数据保存到对应的系统或者数据库中。

* + 1. 用WebService接收数据

应用程序实现WebService接口规范的服务端，供交换平台调用。应用程序将交换平台提供的数据写入到应用系统数据库中。

* + 1. WebService接口规范

WebService对应的WSDL文件为： 。

其接口的java语言描述参见下面的代码。系统中提供了两种参数形式的接口，另外为了操作方便还提供了单独更改大字段内容的接口。

package com.centit.dde.ws;

import javax.activation.DataHandler;

import javax.jws.WebMethod;

import javax.jws.WebParam;

import javax.jws.WebService;

@WebService(targetNamespace = "http://dde.centit.com/ws/")

public interface UploadData {

/\*\*

\* 上传数据接口，写入数据库之前需要进行用户身份验证

\* @param userName 用户名

\* @param userPin 密码，加密形式

\* @param database 目标数据库或者业务

\* @param tableDate 数据格式参见离线文件数据格式，为了控制大小可以将大字段单独上传，也可以一起上传

\* @return

\*/

@WebMethod

String uploadTableAsXml(

@WebParam(name="userName") String userName,

@WebParam(name="userPin") String userPin,

@WebParam(name="database") String database,

@WebParam(name="xmlData") String tableDate);

/\*\*

\* 上传数据接口，写入数据库之前需要进行用户身份验证

\* @param userName 用户名

\* @param userPin 密码，加密形式

\* @param database 目标数据库或者业务

\* @param tableDate 数据格式参见离线文件数据格式，为了控制大小可以将大字段单独上传，也可以一起上传

\* @return

\*/

@WebMethod

String uploadTableAsDataHandler(

@WebParam(name="userName") String userName,

@WebParam(name="userPin") String userPin,

@WebParam(name="database") String database,

@WebParam(name="xmlData") DataHandler tableDate);

/\*\*

\* 单独上次大字段接口

\* @param userName 用户名

\* @param userPin 密码，加密形式

\* @param database 目标数据库或者业务

\* @param tableName 表名

\* @param columnName lob字段名

\* @param keyDesc 主键描述可以是复合主键，形式为 key=value and key2=value2 ...

\* @param isBase64 是否为 base64编码，一般Clob不需要，Blob需要编码

\* @param lobData 大字段字符串

\* @return

\*/

@WebMethod

String uploadLobAsString(

@WebParam(name="userName") String userName,

@WebParam(name="userPin") String userPin,

@WebParam(name="database") String database,

@WebParam(name="tableName") String tableName,

@WebParam(name="columnName") String columnName,

@WebParam(name="keyDesc") String keyDesc,

@WebParam(name="isBase64") boolean isBase64,

@WebParam(name="lobData") String lobData);

/\*\*

\* 单独上次大字段接口

\* @param userName 用户名

\* @param userPin 密码，加密形式

\* @param database 目标数据库或者业务

\* @param tableName 表名

\* @param columnName lob字段名

\* @param keyDesc 主键描述可以是复合主键，形式为 key=value and key2=value2 ...

\* @param lobData 大字段数据

\* @return

\*/

@WebMethod

String uploadLobAsDataHander(

@WebParam(name="userName") String userName,

@WebParam(name="userPin") String userPin,

@WebParam(name="database") String database,

@WebParam(name="tableName") String tableName,

@WebParam(name="columnName") String columnName,

@WebParam(name="keyDesc") String keyDesc,

@WebParam(name="lobData") DataHandler lobData);

}

1. 功能设计2
   1. 信息定义中心
      1. 定义业务数据库连接

在交换平台中定义业务数据库，供数据直接交换的源数据或者目标数据使用。定义数据库连接需要录入以下内容：

* 数据库名称，这个可以自己起并作为主键。
* Jdbc连接url。
* 数据库用户名。
* 数据库用户密码，用加密的方式存储。

数据交换平台支持的数据库包括：Oracle、SqlServer、DB2这三种。

* + 1. 定义WebService服务接口

数据交换平台中的WebService接口规范参见第7章。平台中的通过接口数据交换都是采用推送方式来实现的，都是数据提供方发起请求的。平台提供的WebService服务供业务系统调用向平台提供交换数据，同样业务系统需要接受数据就要实现WebService供平台调用。

WebService接口定义需要录入以下内容：

* 服务名称，这个可以自己起并作为主键。
* WebService服务的url。
* 用于验证的用户名。
* 对应验证用户的密码，用加密的方式存储。
  + 1. 定义数据对应关系

定义源数据库和目标数据库之间数据对应关系，需要定义的内容：

* 源数据库中的一条查询语句。
* 目标数据库中的一个表。
* 源数据查询语句的字段和目标数据库中的表字段对应关系。
* 交换触发器，数据同步过程中需要的处理。参见6.1.4触发器。
  + 1. 定义交换数据

带交换数据可以用于离线文件导出，也可以作为调用业务系统WebService接口的数据。交换数据需要定义的内容包括：

* 源数据库中的一条查询语句。
* 查询语句中各个字段的保存方式和格式。
* 接收方处理标识，这个标识供接收方识别数据内容以判断如何处理。
  + 1. 定义数据处理方式

平台通过接收方处理标识来识别处理方式。平台提供的处理方式有两种，写入数据库和调用业务系统WebService接口发送数据。

写入数据定义内容包括：

* 业务数据库中对应的表。
* 字段对应关系。
* 写入相关触发器。参见6.1.4触发器中所有针对目标数据库的触发器。

调用WebService接口：

* 服务器名，通过这名称关联4.1.2中定义的业务系统WebService接口。
  1. 交换控制中心
     1. 定义数据库直接交换任务

一个直接交换任务包括一个或者多个数据对应关系（4.2.1中定义），并指定一个执行时间计划。这个定时任务也可以手动执行。

* + 1. 定义基于WebService接口交换任务

一个基于WebService接口交换任务包括一个或多个接收方处理标识为调用WebService接口的交换数据（4.2.2中定义），并指定一个执行时间计划。这个定时任务也可以手动执行。

* + 1. 定义数据导出任务

一个数据导出任务包括一个或多个交换数据（4.2.2中定义）；忽略接收方处理标识的类别，这个接收方处理标识是给数据导入时失败的。并指定一个执行时间计划。这个定时任务也可以手动执行。

无论有多少个交换数据，数据导出的都是一个zip文件；参见7.1数据格式。

* + 1. 手动执行任务

在交换平台中可以手动执行4.3.1中定义的三类任务。

* + 1. 数据导入

数据导入的文件为数据交换平台导出的文件，或者符合相关格式的文件。数据导入主要用于网络物理隔离的场景中。

上传离线文件到数据交换平台服务器上，交换平台解压并读取文件中的数据，根据离线文件中的接收方处理标识将文件中的数据写入到目标数据库或者调用业务系统的WebService接口将数据提交给业务系统。

这里有一个关键点接收方处理标识，在数据交换平台中必须有对应的数据处理方式（4.2.3中定义）。

* + 1. WebService接口服务

业务系统可以调用交换平台的WebService接口服务，向交换平台提供数据，交换平台根据数据中的接收方处理标识将文件中的数据写入到目标数据库或者调用业务系统的WebService接口将数据提交给业务系统。和数据导入一样数据交换平台中必须有对应的数据处理方式。

* 1. 交换监控中心
     1. 日志内容

系统中所有的交换操作，包括4.4.1、4.4.2、4.4.3和自动执行的任务都会记录日志，日志内容包括：

* 执行的开始时间、结束时间。
* 执行方式：自动、手动、操作人员（自动为系统）。
* 执行结果：成功多少条、失败多少条。
* 失败数据记录，如果失败了会记录失败的数据，供跟踪和补录。
  + 1. 实时监控

实时查看交互平台中的交换任务和交互情况，包括以下内容：

* 查询系统中所有有效的交换任务。
* 查看系统中正在执行中的交换任务。
* 查看当天交换情况，包括各个任务交换数据总量和失败的条数。
  + 1. 日志查看

平台提供多种方式查看日志：

* 按照任务查看；查看某个任务历史交换情况。
* 按照时间查看；查看某个时间段，比如一天内的所有交换情况。
  + 1. 交换情况统计

对交换结果进行统计分析，比如：

* 按时间段、任务统计数据交换总数、成功数、失败数。