1. Orion-2.1.0版本升级说明
   1. 业务框架增加Spring事务开关

STIRA\_FLOW表增加TX\_FLAG列，默认’Y’；DM流程定义增加Spring事务开关。

全量脚本：

db\business-stria\stria\_mysql.sql

db\business-stria\stria\_oracle.sql

增量DB脚本：

db\patch\V2\orion\2.1.0\stira\_to\_commit\_mysql.sql

db\patch\V2\orion\2.1.0\stira\_to\_commit\_oracle.sql

* 1. SCP冲正功能完善

FW\_FLOW\_LOG表增加REVERSAL\_COUNT列,流程冲正次数

全量脚本：

db\scp\scp\_mysql.sql

db\scp\scp\_oracle.sql

增量DB脚本：

db\patch\V2\orion\2.1.0\scp\_to\_commit\_mysql.sql

db\patch\V2\orion\2.1.0\scp\_to\_commit\_oracle.sql

A 外系统冲正接口:

1）业务实现抽象类AbstractReversalSystem

|  |
| --- |
| **public class** ReversalSystem **extends** AbstractReversalSystem {  @Override  **public** Map flowLogCheck(List<Map> flowLogs,Map request)  {  **if** (flowLogs == **null** || flowLogs.isEmpty())  {  **return** SystemFactory.*getDefaultSystem*().execute(request);  }  **if** (flowLogs.size() > 1)  {  **return** ScpUtils.*errorReturn*(**"XXXX"**,**"被冲正交易不唯一"**);  }  Map flowLog = flowLogs.get(0);  String flowStatus = (String) flowLog.get(**"STATUS"**);  **if** (!**"S"**.equals(flowStatus))  **return** ScpUtils.*errorReturn*(**"XXXX"**,**"原交易未成功完成，不允许冲正!"**);  String flowReversalStatus = (String)flowLog.get(**"REVERSAL\_STATUS"**);  **if** (**"P"**.equals(flowReversalStatus) || **"R"**.equals(flowReversalStatus))  {  **return** ScpUtils.*errorReturn*(**"XXXX"**,**"原交易正在冲正中!"**);  }  **if** (**"S"**.equals(flowReversalStatus) )  {  **return** ScpUtils.*errorReturn*(**"XXXX"**,**"原交易已被冲正!"**);  }  **return null**;  } } |

2）将实现的类注入为Spring Bean

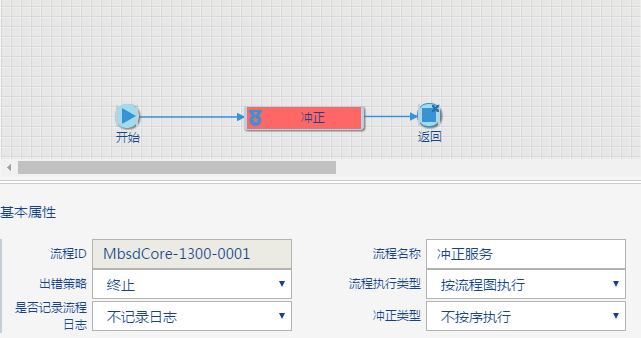
|  |
| --- |
| <**bean id="reversalSystem" class="com.dcits.platform.integration.ReversalSystem"**>  <**property name="keyValueExpr" value="[in][BODY][CHANNEL\_SEQ\_NO]"**/> </**bean**> |

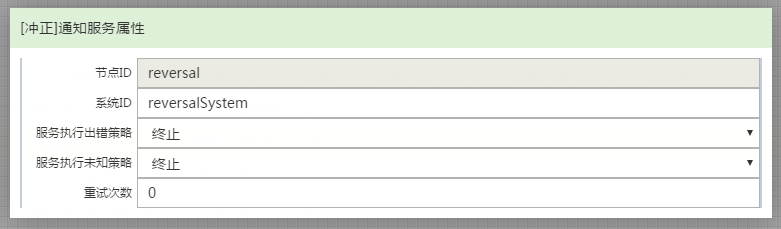
属性keyValueExpr为外系统上送冲正关键字段的表达式，用于从fw\_flow\_log查询已完成的交易。

3）配置数据映射：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**mapping**>  <**flow id="MbsdCore-1300-0001"**>  <**service id="reversal"**>  <**in**>  <**item name="SYS\_HEAD"**>[in][SYS\_HEAD]</**item**>  <**item name="APP\_HEAD"**>[in][APP\_HEAD]</**item**>  <**item name="LOCAL\_HEAD"**>[in][LOCAL\_HEAD]</**item**>  <**item name="BODY"**>[in][BODY]</**item**>  </**in**>  </**service**>  <**out**>  <**if test="[reversal\_out][0][REVERSAL\_RESULTS] == null"**>  <**item name="SYS\_HEAD"**>[reversal\_out][0][SYS\_HEAD]</**item**>  <**item name="BODY"**>[reversal\_out][0][BODY]</**item**>  </**if**>  <**if test="[reversal\_out][0][REVERSAL\_RESULTS] != null"**>  <**map name="SYS\_HEAD"**>  <**item name="RET\_STATUS"**>'S'</**item**>  <**list name="RET"**>  <**item name="RET\_CODE"**>'000000'</**item**>  <**item name="RET\_MSG"**>'冲正请求已受理'</**item**>  </**list**>  </**map**>  </**if**>  </**out**>  </**flow**> </**mapping**> |

4）配置冲正服务的SCP流程：





B 定时任务冲正：

batch\_time\_def 增加记录：



其中类名为：com.dcits.orion.scp.job.TimerDeal

方法名为：deal

* 1. 输入输出报文数据映射功能

核心系统与外系统对接时，外系统的报文结构与核心定义的报文结构有差异时，需要核心提供数据映射功能对输入报文和输出报文进行加工，满足核心和外系统的要求。

数据映射配置文件路径：classpath下parserMapping/\*.xml

数据映射配置文件示例：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* **<!DOCTYPE mappings SYSTEM "http://www.dcits.com/xml/dtd/parserMapping.dtd"*>*** <**mappings**>  <**baseDefine id="json2mapByMapping"**>  <**globalMappingId**>global</**globalMappingId**>  <**mappingIdSpel**>[in][SYS\_HEAD][SERVICE\_CODE]+'-'+[in][SYS\_HEAD][MESSAGE\_TYPE]+'-'+[in][SYS\_HEAD][MESSAGE\_CODE]]</**mappingIdSpel**>  </**baseDefine**>  <**mapping id="global"**>  <**in ref="[in]"**>*<!--输入数据映射-->* <**map name="SYS\_HEAD" ref="[in][SYS\_HEAD]"**>  <**item name="TRAN\_TIMESTAMP"**>T(com.dcits.galaxy.base.util.DateUtils).getDateTime()</**item**>  </**map**>  </**in**>  <**out ref="[out]"**>  <**map name="SYS\_HEAD" ref="[out][SYS\_HEAD]"**>  <**item name="TRAN\_TIMESTAMP"**>T(com.dcits.galaxy.base.util.DateUtils).getDateTime()</**item**>  </**map**>  </**out**>  </**mapping**> </**mappings**> |

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* **<!DOCTYPE mappings SYSTEM "http://www.dcits.com/xml/dtd/parserMapping.dtd"*>*** <**mappings useMapping="global"**>  <**mapping id="Financial-9999-0101"**>  <**in ref="[in]"**>  <**map name="BODY" ref="[in][BODY]" remove="SERV\_DETAIL,SERV\_DETAIL1"**>  <**list name="SERR" alias="SERV\_DETAIL" source="[in][BODY][SERV\_DETAIL]" ref="#SERV\_DETAIL" index="index" remove="AMOUNT"**>  <**item name="SEO\_NO"**>#index+1</**item**>  <**item name="BASE\_ACCT\_NO"**>[in][BODY][SERV\_DETAIL1][#index][BASE\_ACCT\_NO]</**item**>  </**list**>  <**if test="[in][BODY][SER] != null"**>  <**item name="SEO\_NO"**>'0'</**item**>  </**if**>  <**if test="[in][BODY][SER] == null"**>  <**item name="SEO\_NO"**>'1'</**item**>  </**if**>  </**map**>  </**in**>  </**mapping**>  <**mapping id="Financial-9999-0102" useMapping="common"**>  <**in ref="[in]"**>  </**in**>  <**out ref="[out]"**>  </**out**>  </**mapping**>  <**mapping id="Financial-9999-0103" useMapping="global"**>  <**in ref="[in]"**>  </**in**>  <**out ref="[out]"**>  </**out**>  </**mapping**>  <**mapping id="Financial-9999-0104"**>  <**in ref="[in]"**>  </**in**>  <**out ref="[out]"**>  </**out**>  </**mapping**>  <**mapping id="Financial-9999-0105" useMapping=""**>  <**in ref="[in]"**>  </**in**>  <**out ref="[out]"**>  </**out**>  </**mapping**>  <**mapping id="Financial-9999-0106" useMapping="global"**>  <**in ref="[in]" useMapping=""**>  <**map name="BODY" ref="[in][BODY]" remove="ACCT\_NO"**>  <**item name="BASE\_ACCT\_NO"**>[in][BODY][ACCT\_NO]</**item**>  </**map**>  </**in**>  <**out ref="[out]" useMapping=""**>  <**map name="BODY" ref="[in][BODY]" remove="BASE\_ACCT\_NO"**>  <**item name="ACCT\_NO"**>[in][BODY][BASE\_ACCT\_NO]</**item**>  </**map**>  </**out**>  </**mapping**> </**mappings**> |

标签说明：

1. mappings ：根标签
2. baseDefine 数据映射基本定义

globalMappingId 全局数据映射值为采用哪个映射做为全局数据映射，如果不定义，则不进行全局数据映射

mappingIdSpel 差异化数据映射SPEL表达式值为表达式，通过此表达式运算得到映射id，如果不定义，则不进行差异化的数据映射

1. mapping 数据映射标签

in 输入报文映射规则 ：

ref: 引用如果为空，则新建一个Map对象

out 输出报文映射规则

ref: 引用如果为空，则新建一个Map对象

1. map 结构标签

name: name

ref: 引用如果为空，则新建一个Map对象

remove: 删除map中多余的元素，如果多个，中间用‘,’分隔

1. item 元数据标签

name: name属性

1. 数组标签：

name: name

ref: 引用 如果为空，则新建一个Map对象

remove: 删除map中多余的元素，如果多个，中间用‘,’分隔

source: 源如果为空，则新建一个LIST,里面只有一个对象

index: 遍历时索引值的别外，从0开始

alias: 遍历时当前元素的别名

1. if 判断标签：

test: 表达值成立时，进行映射

配置说明：

此映射时在connector层做的数据映射，需要修改galaxy.connector.properties文件，设置msgParser的属性如下：**jetty.msgParser**=**json2mapByMapping**

* 1. 批处理调度优化

Batch\_run\_job表增加FAIL\_CNT, SPLIT\_FAIL\_CNT列,失败task数，分段失败数

全量脚本：

db\bussiness-batch\ batch\_mysql.sql

db\bussiness-batch\ batch\_oracle.sql

增量DB脚本：

db\patch\V2\orion\2.1.0\scp\_to\_commit\_mysql.sql

db\patch\V2\orion\2.1.0\scp\_to\_commit\_oracle.sql

1. Galaxy-2.1.0版本升级说明
   1. Galaxy2.1 中galaxy-core工程拆分

Galaxy-core分别迁移至galaxy-base和galaxy-dubbo两个工程。Galaxy-base增加了线程池管理、spring容器、序列、插桩管理功能，galaxy-dubbo增加了服务代理功能，整合了dubbo扩展和插桩部分。

* + 1. 业务系统项目依赖

将主工程的pom依赖调整为2.1.0。

|  |
| --- |
| <galaxy.version>2.1.0</galaxy.version> |

* + 1. 原galaxy中的依赖版本调整

|  |  |  |  |
| --- | --- | --- | --- |
|  | 功能 | 调整前 | 更新后 |
| Galaxy | 公共组件 | <dependency>  <groupId>com.dcits.galaxy</groupId>  <artifactId>galaxy-core</artifactId>  </dependency> | <dependency>  <groupId>com.dcits.galaxy</groupId>  <artifactId>galaxy-dubbo</artifactId>  </dependency> |

* + 1. 包路径调整

无

* 1. Galaxy-sequences升级
     1. 增加Oracle序列实现

对于Oracle项目提供了Oracle序列的实现。 由于Durid通过$方式开启wall的防SQL注入，创建序列无法创建cache方式，只能创建nocache的。所有通过程序自动创建的序列，格式如下：步长1、最小值：1、最大值：9223372036854775807（最大长整型）、缓存：否、周期序列：否。

使用oracle实现序列，需要修改resources\ext\spring\business-seq.xml配置文件，将配置文件的内容修改为以下表格中的内容。



|  |
| --- |
| <bean id="oracleSeqDao" class="com.dcits.galaxy.sequences.dao.OracleSeqDao">  <property name="shardSqlSessionTemplate" ref="shardSqlSessionTemplate"/>  <property name="autoCreateSeq" value="true"/> </bean> <bean id="localSequences" class="com.dcits.galaxy.sequences.impl.OracleSequences">  <property name="oracleSeqDao" ref="oracleSeqDao"/> </bean> <bean id="shardSequences" class="com.dcits.galaxy.sequences.impl.OracleSequences">  <property name="oracleSeqDao" ref="oracleSeqDao"/> </bean> <dcits:service group="${galaxy.business.sequences.group}"  interface="com.dcits.galaxy.sequences.ShardSequences"  ref="shardSequences"/> |

* + 1. 增加序列集合接口

|  |  |  |
| --- | --- | --- |
| 类名 | com.dcits.galaxy.sequences.AbstractGenerateSeq | |
| 描述 | 序列抽象类 | |
| **Public** List<Long>  getSeqList(**long** value) | | 获得value数量的序列号 |
| **public void addSeqListToThreadLocal(**  **long value)** | | 添加value数量的序列到线程缓存 |

序列集合使用说明：

1. 创建序列。
2. 添加指定数量序列到线程缓存：addSeqListToThreadLocal(long value)。
3. 获得序列号：generateSeqNo()。
4. 获得序列号generateSeqNo()接口说明：

先从线程缓存中读取序列号，当线程缓存中序列为空，则按原方式从序列服务器上获得序列。

使用场景：业务批量TASK作业中，循环获取序列。可以通过上述方式处理。

|  |
| --- |
| XXXSequeneces.addSeqListToThreadLocal(20000);// 一次缓存20000个序列到ThreadLocal  ……  for(业务循环) {  ……  原获取序列的方式不变。  XXXSequeneces.generateSeqNo();// 此时，序列优先通过ThreadLocal获取  ……  } |

* + 1. 序列集合可取最大数量配置

不配置情况下，默认序列集合可取最大数量为20000。也可通过配置sequenceListLimitOver，设置序列集合可取最大数量。

|  |
| --- |
| <!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Oracle序列服务 sequences start\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->  <bean id="oracleSeqDao" class="com.dcits.galaxy.sequences.dao.OracleSeqDao">  <property name="shardSqlSessionTemplate" ref="shardSqlSessionTemplate"/>  <property name="autoCreateSeq" value="true"/>  </bean>  <bean id="localSequences" class="com.dcits.galaxy.sequences.impl.OracleSequences">  <property name="oracleSeqDao" ref="oracleSeqDao"/>  <property name="sequenceListLimitOver" value="30000"/>  </bean>  <bean id="shardSequences" class="com.dcits.galaxy.sequences.impl.OracleSequences">  <property name="oracleSeqDao" ref="oracleSeqDao"/>  <property name="sequenceListLimitOver" value="30000"/>  </bean>  <dcits:service group="${galaxy.business.sequences.group}"  interface="com.dcits.galaxy.sequences.ShardSequences"  ref="shardSequences"/>  <!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Oracle序列服务 sequences end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* --> |

1. Orion-2.0.2版本升级说明
   1. 响应报文Json打包，对与Bigdecimal类型返回科学计数法问题修复

针对这个问题在jsonConvert转换组包的时候，增加了ValueFilter过滤配置，可以在json序列化过程中对Bigdecimal的值使用toPlainString函数输出。修复后对于JavaBean中的对象为Bigdecimal类型原输出科学计数法形式将输出具体数值，例如：{"AMT":1E-9} 过滤后 {"AMT":0.000000001}。

具体的调整business.xml文件调整原来的jsonCovert的bean定义，增加初始函数。若不调整配置不影响原转换逻辑。具体配置如下：

|  |
| --- |
| <!-- json报文 -->  <bean id="jsonConvert" class="com.dcits.orion.base.json.JsonConvert" *init-method="initJsonConfig"*>  *<property name="jsonConfig">*  *<bean class="com.dcits.orion.base.json.JsonConfig"/>*  *</property>*  </bean> |

1. Orion-2.0.1版本升级说明

无

1. Galaxy-2.0.0版本升级说明
   1. Galaxy2.0工程拆分

Galaxy技术平台由原Galaxy拆分为Galaxy和Orion两个工程。Galaxy主要负责技术框架及技术组件的提供，Orion主要负责业务框架部分的功能实现。

* + 1. 业务系统项目依赖

将主工程的pom依赖调整为2.0.0。

|  |
| --- |
| <galaxy.version>2.0.0</galaxy.version>  <orion.version>2.0.0</orion.version> |

* + 1. 原galaxy中的依赖版本调整

|  |  |  |  |
| --- | --- | --- | --- |
|  | 功能 | 调整前 | 更新后 |
| Galaxy | 公共组件 | <dependency>  <groupId>com.dcits.galaxy</groupId>  <artifactId>galaxy-common</artifactId>  </dependency> | <dependency>  <groupId>com.dcits.galaxy</groupId>  <artifactId>galaxy-base</artifactId>  </dependency> |
| 序列组件 | <dependency>  <groupId>com.dcits.galaxy</groupId>  <artifactId>galaxy-business-sequences</artifactId>  </dependency> | <dependency>  <groupId>com.dcits.galaxy</groupId>  <artifactId>galaxy-sequences</artifactId>  </dependency> |
| Orion | 接口包 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-api</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-api</artifactId> </dependency> |
| 基础组件 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-convert</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-base</artifactId> </dependency> |
| 批处理调度组件 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-batch</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-batch</artifactId> </dependency> |
| 批处理公共 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-batch-common</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-batch-common</artifactId> </dependency> |
| 批处理接口 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-batch-api</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-batch-api</artifactId> </dependency> |
| 批处理作业组件 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-batch-task</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-batch-task</artifactId> </dependency> |
| 业务框架核心组件 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-common</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-core</artifactId> </dependency> |
| Hadoop作业调度组件 | <dependency>  <groupId>com.dcits.galaxy</groupId>  <artifactId>galaxy-schedule</artifactId>  </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-schedule </artifactId> </dependency> |
| Stria服务流程处理组件 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-stria</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-stria</artifactId> </dependency> |
| Stria服务流程接口 | <dependency>  <groupId> com.dcits.galaxy</groupId>  <artifactId>galaxy-business-stria-api</artifactId> </dependency> | <dependency>  <groupId>com.dcits.orion</groupId>  <artifactId>orion-stria-api</artifactId> </dependency> |

* + 1. 包路径调整

|  |  |
| --- | --- |
| 调整前 | 更新后 |
| com.dcits.galaxy.common | com.dcits.galaxy.base |
| com.dcits.galaxy.util | com.dcits.galaxy.base.util |
| com.dcits.galaxy.business.common | com.dcits.orion.core |
| com.dcits.galaxy.business.model | com.dcits.orion.api.model |
| com.dcits.galaxy.business.sequences | com.dcits.galaxy.sequences |
| com.dcits.galaxy.business | com.dcits.orion.core |
| com.dcits.galaxy.platform | com.dcits.core.galaxy.platform |
| com.dcits.galaxy.spring.SpringApplicationContext | com.dcits.galaxy.core.spring.SpringApplicationContex |
| com.dcits.galaxy.business.sequences.AbstractGenerateSeq | com.dcits.galaxy.sequences.AbstractGenerateSeq |
| com.dcits.galaxy.client.builder | com.dcits.galaxy.client.core.builder |
| com.dcits.galaxy.access | com.dcits.galaxy.core.access |

* + 1. 数据库脚本调整

batch\_oracle\_upright\_patch2.0.0.sql\batch\_mysql\_upright\_patch2.0.0.sql含了框架提供的Sql脚本修改，各业务系统检查业务自身数据脚本根据实际情况，调整数据库脚本中的包路径。

* + 1. 配置文件调整

conf下扩展ext/spring文件夹，business开头的xml Spring配置将会全部被加载。



其中conf下的galaxy.properties、galaxy.db.properties、galaxy.cache.properties与ext\spring下的business.xml、business-cache.xml、business-db.xml、business-dtp.xml为必须加载的配置。（注：各业务系统可以参考示例内容，对原有文件进行调整。原有文件中的部分无需扩展和业务关注的内容，体现在orion-core的jar中。），其它的配置文件与相关依赖的组件有关，示例的conf是框架全量的配置。具体使用根据业务依赖的jar包而定。

* 1. 添加SQL执行计时插件

可以通过在business-db.xml中sqlSessionFactory增加如下属性：

|  |
| --- |
| <property name="plugins">  <array value-type="org.apache.ibatis.plugin.Interceptor">  <bean class="com.dcits.galaxy.dal.mybatis.plugins.TimerPlugin" />  </array> </property> |

并在logback.xml中开启com.dcits.galaxy.dal.mybatis.plugins.TimerPlugin的trace级别日志

* 1. 批处理增加分布式事务一致性方案：

1. 在BATCH\_STD\_JOB表增加DTP\_FLAG字段
2. 在核心所能操作的所有数据库（包括但不限于teller表）新增BATCH\_TO\_COMMIT表
   1. 批处理增加JOB控制定时任务的的功能

在BATCH\_TIMER\_DEF增加PAUSE\_BATCH和REC\_FLAG字段

* 1. 数据库脚本：

db\business-batch\batch\_to\_commit\_mysql.sql

db\business-batch\batch\_to\_commit\_oracle.sql

db\patch\V2\orion\2.0.0\batch\_mysql\_upright\_patch2.0.0.sql

db\patch\ V2\orion\2.0.0\batch\_oracle\_upright\_patch2.0.0.sql

* 1. 批处理JOB增加分组功能

1. 在BATCH\_STD\_JOB表增加DTP\_FLAG字段
2. OM也需要更新为最新版本
   1. 数据库脚本：

db\patch\V2\orion\2.0.0\beta6\ batch\_mysql\_upright\_patch2.0.0\_beta6.sql

db\patch\ V2\orion\2.0.0\beta6\ batch\_oracle\_upright\_patch2.0.0\_beta6.sql

* 1. 关于Mybaits一级缓存配置调整说明

为全局关闭Mybaits一级缓存以及将更多的参数配置提供给业务开发人员使用，现对配置文件做以下调整：

1. 将Mybaits配置文件调整至用于打包的工程XXX—integration中的config\ext目录下（galaxy-dal jar包中的原配置文件继续保留，不做修改）

* 示例如下：

|  |
| --- |
|  |

* Mybaits基本配置模板（模板中的基本设置不可修改，仅能添加新的参数）：



1. 如需全局关闭一级缓存，则调整以下配置
2. 则在mybatis-config.xml中添加标红的参数

|  |
| --- |
| <?xml version=**"1.0"** encoding=**"UTF-8"**?>  <!DOCTYPE configuration PUBLIC  "-//mybatis.org//DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd">  <configuration>  <properties resource=**"galaxy.db.properties"**/>  <settings>  <!-- 数据库字段与JavaBean属性采用驼峰命名映射 -->  <setting name=**"mapUnderscoreToCamelCase"** value=**"true"**></setting>  <!-- Oracle库JDBC空类型兼容性处理 -->  <setting name=**"jdbcTypeForNull"** value=**"NULL"**></setting>  <!--返回对象是Map时，空值需要 put null -->  <setting name=**"callSettersOnNulls"** value=**"true"**></setting>  <setting name=**"localCacheScope"** value=**"STATEMENT"**></setting>  </settings>  <typeAliases>  <typeAlias type=**"com.dcits.galaxy.dal.mybatis.proactor.page.ParameterWrapper"** alias=**"ParameterWrapper"**/>  </typeAliases>  <plugins>  <plugin interceptor=**"com.dcits.galaxy.dal.mybatis.plugins.PagePlugin"**/>  </plugins>  </configuration> |

1. 修改business-db.xml中sqlSessionFactoryBean配置

|  |
| --- |
| <!-- 集成mybatis，统一在此加载，对业务开发而言约定：其配置文件在各个jar包中的/META-INF/mapper/\*\*/\*.xml -->  <bean id=**"sqlSessionFactory"** class=**"org.mybatis.spring.SqlSessionFactoryBean"**>  <property name=**"dataSource"** ref=**"dataSource0"**/>  <property name=**"databaseIdProvider"** ref=**"databaseIdProvider"**/>  <property name=**"transactionFactory"** ref=**"transactionFactory"**/>  <property name=**"configLocation"**  value=**"classpath:ext/mybatis/mybatis-config.xml"**/>  <property name=**"mapperLocations"** value=**"classpath\*:META-INF/mapper/\*\*/\*.xml"**/>  <property name=**"typeAliasesPackage"** value=**"${typeAliasesPackage}"**/>  </bean> |