1.关闭弹出的窗口：

getContext().bubbleMessage(**new**com.jiuqi.dna.ui.template.launch.TemplateWindow.CloseMessage());

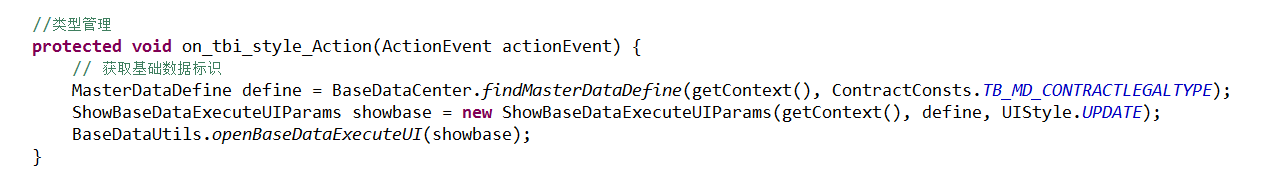
1. 设置弹出的窗口右上角最大化按钮JWT.MAXIMUM

设置窗口不关闭，其他页面操作不可行JWT.MODAL

1. 使用bap平台的页面的时候，使用内部语句进行实现

// 获取基础数据标识

打开bap平台内置的页面



//类型管理

protected void on\_tbi\_style\_Action(ActionEvent actionEvent) {

// 获取基础数据标识

MasterDataDefine define = BaseDataCenter.findMasterDataDefine(getContext(), ContractConsts.TB\_MD\_CONTRACTLEGALTYPE);

ShowBaseDataExecuteUIParams showbase = new ShowBaseDataExecuteUIParams(getContext(), define, UIStyle.UPDATE);

openBaseDataExecuteUI(showbase);

}

protected BusinessModel openBaseDataExecuteUI(ShowBaseDataExecuteUIParams params){

Map<String, Object> templateParameters = new HashMap();

FBusinessModelDefine define = params.getModelDefine();

BusinessModel baseModel = ((BusinessModelDefine)define).createModel(params.getContext(), null);

String masterTableName = define.getMasterTable().getName();

templateParameters.put("baseDataModel", baseModel);

templateParameters.put("TABLENAME", masterTableName);

win = TemplateLauncher.openTemplateWindow(null, "BaseDataMaintainPage", templateParameters, JWT.MAXIMUM|JWT.CLOSE|JWT.MODAL, JWT.NONE);

String title = INLUtil.getINLStringValue(define.getMasterTable().getRecid(), define.getMasterTable().getTitle());

win.setTitle(title);

win.addWindowListener(new WindowListener() {

@Override

public void windowClosing(WindowEvent arg0) {

provider = new ContractMgrProvider(getContext());

initTree();

}

});

win.addWidgetListener(new WidgetListener() {

@Override

public void widgetInited(WidgetEvent arg0) {

}

@Override

public void widgetDisposed(WidgetEvent arg0) {

provider = new ContractMgrProvider(getContext());

initTree();

}

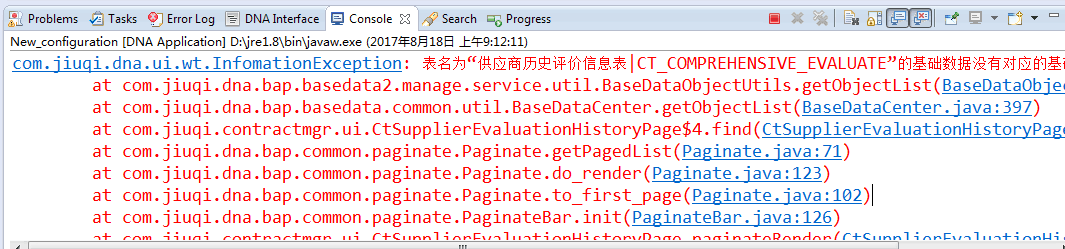
});

return baseModel;

}

删除前的数据校验事件监听



1. 

这个是因为通过表名获取基础数据项的时候发现这个表没有对应的基础数据定义

就会抛出这个异常,只要把基础数据定义放出来就好了

1. 打增量的时候，class文件、form文件、controls文件、同时需要看一下mate-inf目录下的mf文件，也有可能被动更改。Mf文件是程序自动更改的
2. 基础数据和单据设计在底层是相同的。只不过单据设计之后，在页面显示的数据是从数据库中读取出来的。相应的数据的操作都是与数据库进行的交互的。基础数据是从内存中读取的数据。涉及到的数据都是最基础和公用的。数据量不是非常大。枚举的。而且数据的变动也是非常小的。比如公司的部门，很少有变动，这类的就适合作为基础数据。
3. 基础数据管理---页面中的左侧树形，右上侧的表格，都是系统自动生成的，只需要配置右下的页面。
4. 增加功能按钮。实现IAction接口。也可以继承BaseBusinessAction类，BaseBusinessAction是实现IAction接口的。但是需要重写executeAction方法。设置图标ActionConst.ico\_upgrade
5. model.getModelState获取模型状态-----ModelState.NEW新建状态。ModelState.EDIT编辑状态。
6. 基础数据新增操作

Context context = **null**;//上下文

String tableName = **null**;//表名

IBaseDataContextHandle handle = **null**;//基础数据情景

BusinessModelDefine define = (BusinessModelDefine)BaseDataCenter.*findMasterDataDefine*(context, tableName);

BaseDataModel model = **new** BaseDataModel(context, define, handle);

model.add();

//设置各个字段的值并保存

BusinessObject object = model.getModelData().getMaster();

object.setFieldValue("", "");

model.save();

model.beginUpdate();

model.endUpdate();

1. 系统选项功能中，添加多个选项，在代码中是用list集合

List<String> list = new ArrayList<String>();

list.add("CT\_INFO");

list.add("PA\_REGCONTRACT\_MAIN");

Item itemCtInfo =

new Item(ContractInfoTableName, "合同基本信息主表名称", "合同基本信息主表标识", new StringListType(list, list.get(0),

StorageType.SYSTEM\_DEFAULT));

1. 使用docx4j的时候，需要引入slf4j-api的jar包，但是在DNA平台，引入jar包会有点问题，需要在mf文件中自己手动添加jar包依赖。
2. 查找字段关联的表使用字段对象IField field = billModel.getDefine().findField(billModel.getDefine().getMasterTable().getName(), WaterMarkID);

ield.getType();

TableDefine relTableDefine = field.getRelationTable();

relTableDefine.getName();

1. 当改动的代码不被执行，有可能是编译错误了。重新clean，然后build。看一下class文件是不是最新的。有可能library出错。
2. 页面配置的时候，有可能是同一个按钮但是配置的不同，只是在单据设计页面把按钮的名称改了
3. 当报错信息显示是自己代码的时候，如果代码中没有，有可能是基础数据中表的关联有问题，找到基础数据表，查看表标识，查看单据定义信息表B0104\_BILLDEFINE。

select \* from B0104\_BILLDEFINE where NAME = 'JCKJ';

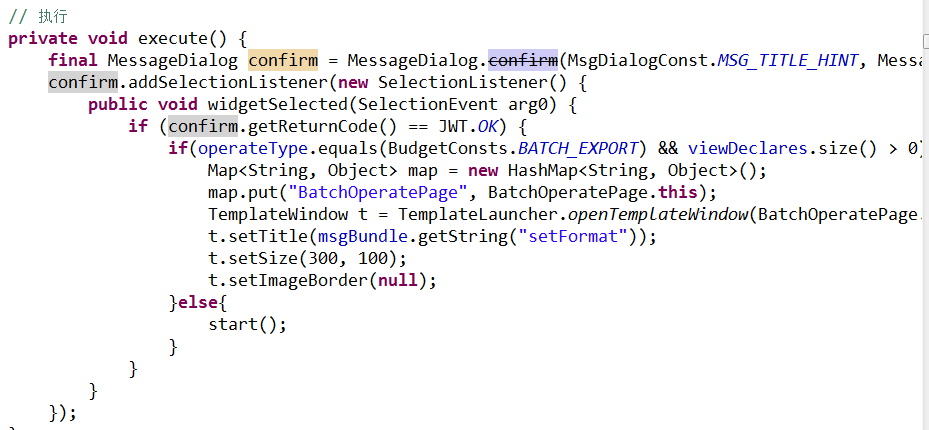
可以查看信息，其中的define字段存储的是配置信息。在配置信息中配置的class有可能在自己代码中进行了位置的更改，重新找到对应的类，然后在配置信息中替换新的类路径就可以了。

1. billModel 获取页面所有信息。Inputmain也可以获取页面所有信息，更强大。inputmain.getGrid().getContent()获取页面中grid所有的信息。
2. 查找某些字段的时候，打断点看data中有哪些信息。是什么类型的。不要定义成实现类。要定义成接口对象。
3. 别人已经开发好的方法可以进行借鉴，做数据保存的时候，已经有了保存功能，最重要的就是grid和provider。但是在保存功能中的provider时私有的，并且如果自己写会非常困难，这时候可以为provider增加get方法。就可以获取到provider了。并且也效仿保存中的数据处理，设置可编辑，记录操作行。然后设置showtext。这时候调用保存功能。就会进行原本存在的保存操作。对数据进行处理。具体内部怎么改变值就让保存按钮进行操作了。

对固定表和浮动表不能使用getdata这类的方法。Floatorder

1. final MessageDialog md = MessageDialog.confirm("提示", "确认转入风险警示录？");

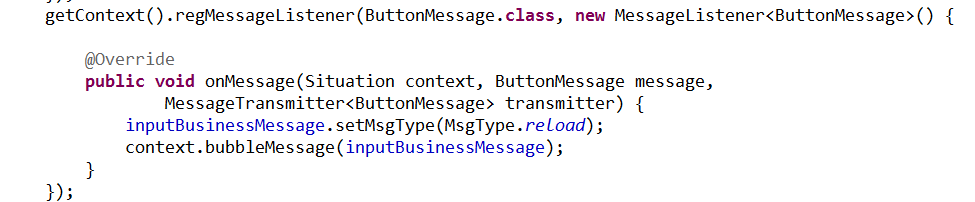
弹出确认框。Md.addselection()添加监听，md.getReturnCode()==jwt.ok。对消息确认和取消的监听



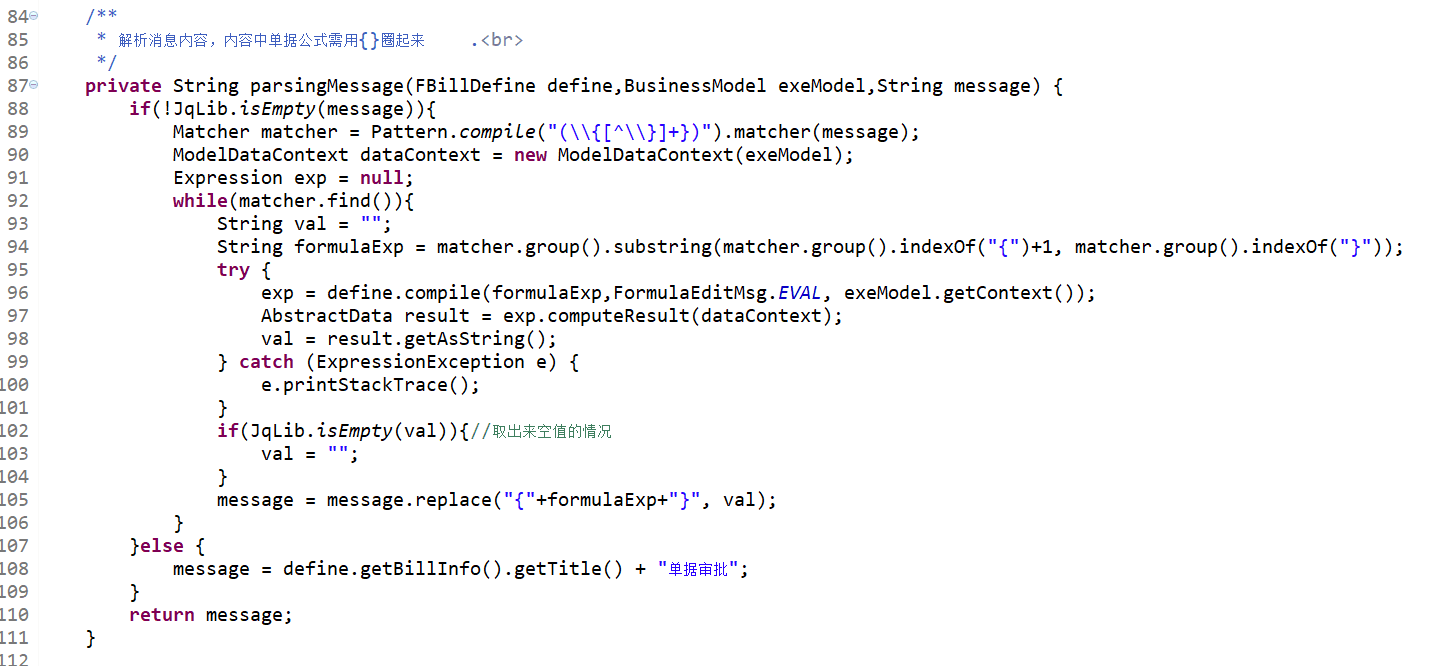
1. 浮动表中，inputmain中存储的是页面中所有的信息。

inputBusinessMessage中存的是界面业务协调消息

1. 浮动表中向下级传递信息用



可以实现浮动表页面的刷新操作

1. 添加依赖关系的时候，在runtime的classpath中或者dependencies中，两者中添加一个就行了。否则会造成冲突。 java.lang.NoSuchMethodError报错很多情况下都是class冲突引起的，比如说：某个java编译成class后，放到classes下面，然后lib目录下，也有这个class所在的jar包，这样就导致classpath实际上有两个相同的class 。也有可能是class文件编译的jdk版本不一致引起的
2. BillConst类设置的是工作流状态。常量是大写，里面写的非常量用小写
3. 

这个取值更灵活，在已知主表的情况下，可以设定公式。

26.Display.getCurrent().exportFile导出指定数据

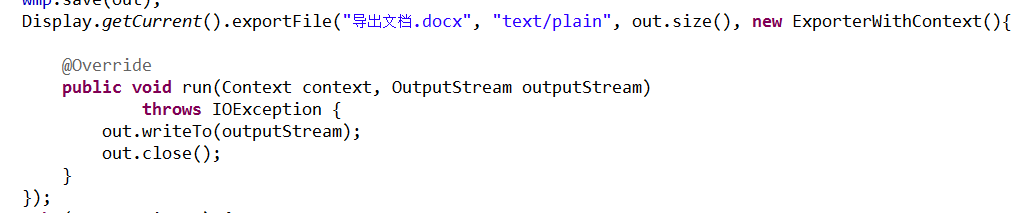
参数：

fileName 输出客户端的文件名称

contentType 输出内容的类型，如文本文件为：text/plain

exporter 导出器

contentLength



1. 单据服务接口：billcenter
2. 查询语句定义 定义一个可执行的查询语句结构

QueryStatementDeclare m\_query = (QueryStatementDeclare) getContext().parseStatement(dnaSql);

RecordSet rs = getContext().openQuery(m\_query);

扩展jq的公式ifunction

29.//获取流程实例关联数据

GUID businessInstanceId = GUID.valueOf(process.getWorkRef());

BtDataStateImpl datastate = context.find(BtDataStateImpl.class, businessInstanceId);

1. 获取流程中多维表相关数据

// datastate.getYear();

// datastate.getPeriodNum();

// datastate.getPeriod();

// DataPeriod d = new DataPeriod(year, type, period);

//

// datastate.getOrgId();

//

// datastate.getLinkId();

//

// LinkidUtils.getDimMap(linkid);

// getDimMap.put(version,0) zbtype=1 unitid=orgid

//

// DimConsts.SYS\_DIMEN\_ORG\_NAME;

// ZHDF =

// "PARENTID"

1. 对页面的更改不生效，可以尝试清楚浏览器缓存试试
2. 自动任务继承AbstractWorkHandler，

自动任务的参数配置界面继承AbstractBindingWorkConfig

1. 配置系统选项，实现SystemOptionDefine类、在xml中注册

<systemoptions>

<systemoption class="com.jiuqi.contractmgr.common.ContractMgrSystemOption"/>

</systemoptions>

1. 获取billmodel单据表名billModel.getDefine().getMasterTable().getName()
2. 内置单据模型继承BillDataModel类，实现defineActions，defineFunctions，initFieldValues(ModelData md)初始化单据。
3. Oracle创建用户并赋权限create user ljy IDENTIFIED by ljy;

grant dba to ljy;最高权限

删除密码select \* from core\_authuser;

update core\_authuser set password='00000000000000000000000000000000';

Oracle导入dmp文件imp [test/test@10.1.21.104/orcl](mailto:test/test@10.1.21.104/orcl) file=d:ll.dmp full=y log=d:ss.log

用户时间过期：

|  |
| --- |
| update sm\_user set ACTIVESTOPTIME='21-10月-29 06.16.42.821000000 下午' |

1. 还原数据库的时候可以删除掉无用的数据，比如日志表、Core\_mate\*\*\*\_\*\*的表。都是可以删除的,ddb格式的文件放在da目录的word下。名称与要还原到的数据库名称相同。
2. 实现按钮后置任务继承类AbstractBudgetActionHandler。实现dowork方法。在xml中进行注册<BudgetActionHandlers>

<handler name="CheckViewReportHander" title="校验指定表样是否都上报" class="com.jiuqi.budget.input.modeAction.handler.impl.CheckViewReportHander"/>

1. 获取获得当前时期机构下的所有表List<HybercubeViewDefine> allviewlist = SolutionUtil.getHyberViewsByPeriod(getContext(), solution, mainDataPeriod);
2. 扩展后置任务：继承类AbstractBudgetActionHandler，dna.xml中注册
3. 异步任务：1.定义接口，返回任务类型，名称，说明，页面等。2.抽象类实现接口，定义界面的实现以及增加或者更新日志记录。？抽象类的日志类？3.抽象类继承抽象类。定义自己的方法。

BtAsynHandlerGather？？？继承的类，元素工厂

BtAsynHandlerElement？获取到xml中的注册对象

BtAsynHandlerGather类通过工厂元素，获取xml注册对象。解析xml对象参数，根据注册的类类型初始化类，返回初始化之后的map集合。loadClass参数中的类类型？？？

执行异步自动任务，自动任务创建新的上下文。获取配置的参数以及选择的异步自动任务，执行dowork。Service使用的是异步service。context.asyncHandle(task);

两个页面配置，一个是异步任务自动的页面配置，另一个新建异步任务的时候设置的默认参数弹出界面。将参数传递到父页面。

重新实现自动任务的界面。Xml中注册

<AutoWorkItemHandlerConfigs>

<AutoWorkItemHandlerConfig handler="BtAsynWorkFlowHandler" class="com.jiuqi.budget.asynTask.workFlow.BtAsynWorkFlowHandlerPage"/>

</AutoWorkItemHandlerConfigs>

计划任务？？

定义一个实现接口的统一的抽象类，执行dowork方法。工作流和编制表分别定义抽象类实现抽象任务。自动任务处分别定义工作流自动任务以及编制表自动任务。通过获取初始化后选择的自动任务列表，执行doAsynHandle方法。进而执行异步自动任务。

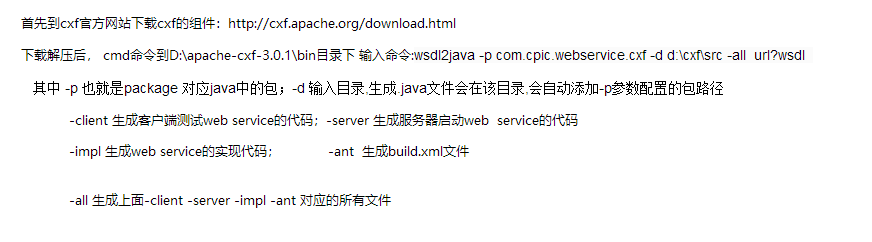
1. 给测试人员sql语句的时候，不需要转成DNA sql。同时需要把问题一步到位。比如删除的，直接给delete语句



1. 执行delete from B0101\_MAINBODY where MBNAME='MD\_ORG' and state=0;

mainbody多了，但是zb没有配套多出来

1. 基础数据内置的时候在基础数据设置那里也内置。参考ContractSupplyService
2. 查看报错的时候，如果是框架的错误，并且这个流程在其他地方都好用，只在这里不好用。那就有可能是数据的问题。查找框架的报错类。定位哪里有问题，然后查看自己的代码。
3. 处理代码冲突的情况：远端是有小叹号，本地代码有黄色框标记，右键本地代码，看对比，左侧是远端，右侧是本地，下面的是合并之后应有的样子。然后处理



1. 打开单据列表页面

TemplateWindow tw = WindowUtil.openWindow2(SendAcceptanceCheckSetting.this, "BatchCreateBillMgrDefinePage", "单据选择", WindowUtil.COMMON\_DIALOG\_STYLE, JWT.OK|JWT.CANCEL, null);

tw.setHeight(400);

tw.setWidth(400);

48.//判断固定表单元格是不是指标表达式

ObjectobjGD=gridData.getGridCellData(p.x,p.y).getCellData(HyberConstants.CELL\_MAP\_EXP\_KEY)

//判断浮动表单元格是不是指标表达式

Object objFD = gridData.getGridCellData(p.x,p.y).getData(); xcelldata

1. -agentpath:C:\ProgramData\YourKit\yjpagent\_14110\_64\_C4C0A79F.dll=disablestacktelemetry,disableexceptiontelemetry,delay=10000
2. 旧单据列表刷新BillMgrMessageNotifier.reQuery((Situation)context, true, true,true);
3. 看超大日志文件的软件PilotEdit
4. 时期和机构的维度，有可能在表内，虽然表外也能看到，但是实际上是没有的。
5. 添加小问号提示框：

controls.lbl\_3.setBackimage(getContext().find(ImageDescriptor.class, BapImages.ico\_help));HoverInfoWindowTool.bindTargetControl(controls.lbl\_3, "描述");

1. 机构类型的基础数据查询使用

Organization organization = OrgVersionUtils.getOrgItemByOrgId(getContext(), table, null, recid, null);

if (organization != null) {

obj = OrgVersionUtils.convert2(getContext(), table, organization, null);

}

54.设置连接形式的单元格

cell.getStyle().setFont(new Font(9, "宋体", JWT.FONT\_STYLE\_PLAIN | JWT.FONT\_STYLE\_UNDERLINE));

cell.getStyle().setForeground(Color.COLOR\_BLUE.getColor());

1. 附件关联的表thr\_attachment\_index这张表的 attachmentId这个字段
2. if(null != System.getProperty("com.jiuqi.dna.debug")

&& "true".equalsIgnoreCase(System.getProperty("com.jiuqi.dna.debug")))

1. ReturnObject 用于错误返回
2. -Declipse.ignoreApp=true -Dosgi.noShutdown=true
3. 记录系统日志 context.handle(new AddLogInfoTask(AddLogInfoTask.Constant.INFOMATION, "模型维护", "模型维护，发布模型", loginfo));
4. Git安装目录 c://x86//git
5. 查看数据库表路径以及表名select file\_name,tablespace\_name from dba\_data\_files;
6. 增加表空间alter TABLESPACE USERS add datafile 'D:\JQ\APP\ORADATA\ORCL\USERS02.DBF' size 30720M;
7. 查看数据库表空间使用情况select a.tablespace\_name,a.bytes/1024/1024||'M' total,b.bytes/1024/1024||'M' used,c.bytes/1024/1024||'M' free ,

trunc(b.bytes/a.bytes,2) useded, trunc(c.bytes/a.bytes,2) freeed

from sys.sm$ts\_avail a,sys.sm$ts\_used b,sys.sm$ts\_free c

where a.tablespace\_name = b.tablespace\_name

and a.tablespace\_name = c.tablespace\_name;

63.DNA平台下获取JDBC的方式1、context.find(java.sql.Connection.class);

2、context.find(java.sql.Connection.class,DataSourceName);

3、context.find(java.sql.DatabaseMetaData.class);

4、context.find(java.sql.Statement.class);

5、context.find(java.sql.Statement.class,Integer resultSetType, Integer resultSetConcurrency);

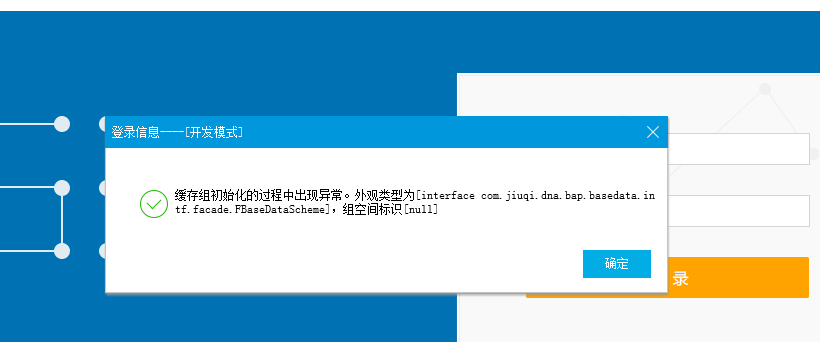
6、context.find(java.sql.PreparedStatement,SQL);

7、context.find(java.sql.PreparedStatement,SQL,Integer resultSetType, Integer resultSetConcurrency);

8、context.find(java.sql.PreparedStatement,SQL,Integer autoGeneratedKeys);

9、context.find(java.sql.CallableStatement,SQL);

10、context.find(java.sql.CallableStatement,SQL,Integer resultSetType, Integer resultSetConcurrency);

执行sql解决，多执行几遍

DELETE FROM MD\_SYSINFO S WHERE S.RECID NOT IN (SELECT MAX(RECID) FROM MD\_SYSINFO K GROUP BY K.NAME);

DELETE FROM MD\_SYSSCHEME WHERE MD\_SYSSCHEME.BASEDATAID NOT IN (SELECT MD\_SYSINFO.RECID FROM MD\_SYSINFO);

DELETE FROM MD\_SYSINFO S WHERE S.MASTERTABLE NOT IN (SELECT MBNAME FROM B0101\_MAINBODY );

DELETE FROM MD\_ORG\_VERSION T WHERE T.TABLENAME NOT IN (SELECT K.MBNAME FROM B0101\_MAINBODY K);

DELETE FROM B0102\_MASTERDATADEFINE M WHERE M.NAME NOT IN (SELECT K.MBNAME FROM B0101\_MAINBODY K);

1. 产品需要授权 <http://gmc6.jiuqi.com.cn/>这个地址<http://IP:PORT/app/licenseMgr>
2. Logger logger = DNALogManager.getLogger("budget/input");
3. 启动服务时缺失参数，导致服务停止。-Dcom.jiuqi.dna.bap.datastructure.boot.ignore.error=true加参数
4. 当N很小时，快速排序慢，归并排序快

当N很大时，并且有序程度高时，快速排序最快

当N很大时，并且有序程序低时，堆排序最快

1. 启动的时候报丢失字段的问题，加参数

-Dcom.jiuqi.dna.bap.datastructure.boot.ignore.error=true

68．

69. Linux下生成线程dump的方式

第一步：先找jdk安装路径：命令：echo $JAVA\_HOME

第二步：进入jdk安装路径 cd “路径”

第三步：找bin文件夹下的jstack文件 ls

第四步：查找进程号 ps -ef|grep java

第五步：运行此命令

./jstack “进程号” >/home/appuser/`date "+%H:%M:%S:%N"`.txt

`date "+%H:%M:%S:%N"`.txt为以当前时间作为文件名。

70.调试端口参数-Xrunjdwp:transport=dt\_socket,address=8133,server=y,suspend=n

**71.浮动表指标外露**

1、执行如下sql查询一下是否有垃圾数据

select \* from RP\_MAPCELL as t where t.rpt\_guid=(select t.recid from bt\_hybercube\_view as t where

t.name='表样标识') and t.mc\_row\_no=6 and t.mc\_col\_no=2；

上述sql中：

rpt\_guid 报表id

mc\_row\_no 表格上的行号

mc\_col\_no 表格上的列号

2、如果上面的结果查出来多余1行的时候，执行如下sql

DECLARE

CURSOR INFOS IS

select RPT\_GUID,MA\_GUID,MC\_ROW\_NO,MC\_COL\_NO from RP\_MAPCELL group by RPT\_GUID,MA\_GUID,MC\_ROW\_NO,MC\_COL\_NO having count(1)>1;

tempStr varchar2(4000);

BEGIN

FOR INFO IN INFOS LOOP

tempStr:='delete from RP\_MAPCELL where RPT\_GUID=:a and MA\_GUID=:b and MC\_ROW\_NO=:c and MC\_COL\_NO=:d' ;

EXECUTE IMMEDIATE tempStr using INFO.RPT\_GUID,INFO.MA\_GUID,INFO.MC\_ROW\_NO,INFO.MC\_COL\_NO;

COMMIT;

END LOOP;

COMMIT;

END;

3、然后重启

4、然后重新拖拽指标

70 恢复dmp

--第一次操作时，按1、2、3、4、5、7顺利执行,如表空间不够时，单独执行6、8、3、4、5、7。

现场发过来的数据库后缀为.DMP时，先将文件放到D:\dbfile目录下，然后按照以下顺序执行：8、3、4、5、7

create DIRECTORY dbfile as 'D:\dbfile'; --1、创建文件夹，存放导出数据库地址

create tablespace dtys --2、创建名为user\_data的数据表空间

logging

datafile 'E:\app\zhuruijuan\oradata\orcl\dtys.dbf ' --数据表空间文件地址

size 50m --初始大小

autoextend on --自动扩容

next 50m maxsize 20480m --每次扩容50m，最大20480m

extent management local;

create user dtys identified by dtys default tablespace dtys; --3、创建用户

grant dba to dtys; --4、授权DBA权限

alter user dtys IDENTIFIED by dtys; --5、强制修改用户名及密码

ALTER TABLESPACE dtys ADD DATAFILE 'E:\app\zhuruijuan\oradata\orcl\dtys02.DBF' SIZE 20000M;--6、表空间不够时，扩容表空间

drop user dtys CASCADE; --8、删除用户，扩容后从新导入的时候，先drop掉用户，再重新生成用户（SQL Developer 中，在SYSTEM连接中，执行删除用户，创建用户，授权等操作）

impdp dtys/dtys DIRECTORY=dbfile dumpfile=dtys.dmp schemas=dtys --7、cmd中执行导入语句（cmd中直接执行此行命令即可）

impdp dtys/dtys DIRECTORY=dbfile dumpfile=dtys.dmp tables=bt\_solution （只导入指定表,数据泵导出文件时）

imp dtys/dtys@orcl3  file=D:\dbfile\bt\_solutionbakup.dmp  tables=(bt\_solutionbakup)（一般数据导出文件时）

－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－

删除用户时，提示无法删除连接的用户，执行以下语句：

select \* from v$session where username='DTYS';--查询该用户的SID、SERIAL#字段，

alter system kill session '204,4';----SID、SERIAL#字段

imp dtys/dtys@orcl3  file=D:\dbfile\bt\_solution.dmp  tables=(bt\_solution)

71:打印sql

-Dcom.jiuqi.dna.debug.dnasql=true  
-Dcom.jiuqi.dna.debug.sql.dmlparam=true  
-Dcom.jiuqi.dna.debug.sql.dml=true

72:查询待办总条数

select i.STATE piStateField,p.NIGUID NIGUID, n.piguid piguid

from participant p

left join NODEINSTANCE n on n.recid = p.NIGUID

left join ProcessInstance i on i.recid = n.piguid

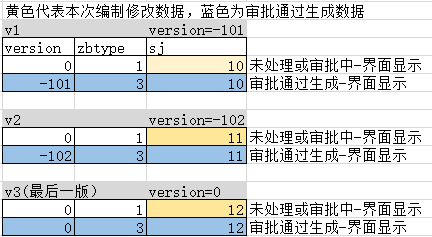
where 1=1 and i.STATE = 3 and p.ACTION = 0

and p.WORKCATEGORY is not null

and p.WORKCATEGORY <> '00000000000000000000000000000000'

and (p.USERID ='9797BBBCC96B6266787437A8706FC772'

or p.commission ='9797BBBCC96B6266787437A8706FC772');



1. 还原dmp文件

create DIRECTORY dbfile as 'D:\dbfile';

create tablespace CWAG datafile 'E:\Oracle\oradata\orcl\CWAG.dbf' Size 10240M AUTOEXTEND ON next 10M;

create user CWAG identified by CWAG default tablespace CWAG; --3、创建用户

grant dba to CWAG;

ALTER TABLESPACE CWAG ADD DATAFILE 'E:\Oracle\oradata\orcl\CWAG02.dbf' SIZE 30G;

imp 用户/密码 file=文件路径 full=y ignore=y;

1. 修改oracle用户密码

alter user system identified by system;

1. 索引

|  |
| --- |
| 1、创建索引  create index 索引名 on 表名(列名);  2、删除索引  drop index 索引名;  3、创建组合索引  create index 索引名 on 表名(列名1,,列名2);  4、查询索引  --根据索引名，查询表索引字段 select \* from user\_ind\_columns where index\_name='索引名'; --根据表名，查询一张表的索引 select \* from user\_indexes where table\_name='表名'; |

1. 第三方