# 为单据增加分页功能

2017.8.4 杨涛

1.配置文件的修改

（1）在列表视图中添加分页控件，新增内容为红色字体部分：

<bean id=*"listView"* class=*"nc.ui.mapub.costpricebase.view.CostPriceListView"*

init-method=*"initUI"*>

<property name=*"model"* ref=*"manageAppModel"*/>

<property name=*"templateContainer"* ref=*"templateContainer"*/>

<property name=*"paginationBar"* ref=*"paginationBar"*></property>

</bean>

（2）增加分页控件和分页模型，如下，均采用平台的类

<!-- 分页控件 -->

<bean id=*"paginationBar"* class=*"nc.ui.uif2.components.pagination.PaginationBar"* init-method=*"registeCallbak"*>

<property name=*"paginationModel"* ref=*"paginationmodel"*></property>

<property name=*"context"* ref=*"context"* />

</bean>

<!-- 分页模型 -->

<bean id=*"paginationmodel"* class=*"nc.ui.pubapp.uif2app.model.pagination.PubPaginationModel"* init-method=*"init"*>

<property name=*"paginationQueryService"* ref=*"paginationProxy"*></property>

</bean>

（3）修改之前的查询服务代理，对查询结果进行分页处理，重新配置代理为：

~~<!-- 单据查询代理 -->~~

~~<bean id=~~*~~"queryProxy"~~* ~~class=~~*~~"nc.ui.mapub.costpricebase.serviceproxy.CostPriceBaseQueryProxy"~~* ~~/>~~

<!-- 分页代理 -->

<bean id=*"paginationProxy"* class=*"nc.ui.mapub.costpricebase.serviceproxy.CostPriceBasePaginationProxy"* />

(4)修改数据模型管理器，并添加分页服务接口，需要将之前的查询代理删除，并修改数据模型管理器的类为平台封装的分页数据管理类：*nc.ui.pubapp.uif2app.model.pagination.PaginationModelDataManager。*分页服务接口定义了查询时访问的方法。

<!-- 数据模型管理器，主要负责各种方式的模型初始化 -->

<bean id=*"modelDataManager"* class=*"nc.ui.pubapp.uif2app.model.pagination.PaginationModelDataManager"*>

<property name=*"model"* ref=*"manageAppModel"*></property>

~~<!-- <property name="service" ref="queryProxy"></property> -->~~

<property name=*"paginationModel"* ref=*"paginationmodel"*></property>

<property name=*"pageQueryService"* ref=*"pageQueryService"* />

</bean>

<!-- 分页服务接口 -->

<bean id=*"pageQueryService"* class=*"nc.ui.pubapp.uif2app.model.pagination.UIPageQueryService"*>

<property name=*"allPagePkQueryServiceMethod"* value=*"nc.itf.mapub.costpricebase.ICostPriceBasePaginationService.queryPKs"*></property>

<property name=*"dataOfPksQueryServiceMethod"* value=*"nc.itf.mapub.costpricebase.ICostPriceBasePaginationService.queryBillByPK"*></property>

</bean>

2.分页代理类*paginationProxy*，代理类实现了IPaginationQueryService接口。在实现的接口方法queryObjectByPks中，通过NCLocator.getInstance().lookup(ICostPriceBasePaginationService.class)向后台请求数据，因此需要定义接口ICostPriceBasePaginationService及实现CostPriceBasePaginationServiceImpl。

import nc.bs.framework.common.NCLocator;

import nc.itf.mapub.costpricebase.ICostPriceBasePaginationService;

import nc.ui.uif2.components.pagination.IPaginationQueryService;

import nc.vo.mapub.costpricebase.entity.CostPriceAggVO;

import nc.vo.pub.BusinessException;

import nc.vo.pubapp.pattern.exception.ExceptionUtils;

/\*\*

\* @since v6.3

\* @version 2017年8月2日 下午6:55:51

\* @author Administrator

\*/

public class CostPriceBasePaginationProxy implements IPaginationQueryService {

/\*根据主键数组查询，返回实际的VO

\* 参数：主键数组pks

\* @see nc.ui.uif2.components.pagination.IPaginationQueryService#queryObjectByPks(java.lang.String[])

\*/

@Override

public Object[] queryObjectByPks(String[] pks) throws BusinessException {

// TODO Auto-generated method stub

CostPriceAggVO[] rets = null;

ICostPriceBasePaginationService paginationQueryService =

NCLocator.getInstance().lookup(ICostPriceBasePaginationService.class);

try {

rets = paginationQueryService.queryBillByPK(pks);

}

catch (Exception e) {

ExceptionUtils.wrappException(e);

}

return rets;

}

}

3.分页查询接口，ICostPriceBasePaginationService

package nc.itf.mapub.costpricebase;

import nc.ui.querytemplate.querytree.IQueryScheme;

import nc.vo.mapub.costpricebase.entity.CostPriceAggVO;

import nc.vo.pub.BusinessException;

/\*\*

\* @since v6.3

\* @version 2017年8月2日 下午7:01:58

\* @author Administrator

\*/

public interface ICostPriceBasePaginationService {

//根据查询条件进行查询，返回主键数组

public String[] queryPKs(IQueryScheme queryScheme) throws BusinessException;

//根据主键数组进行查询，返回VO

public CostPriceAggVO[] queryBillByPK(String[] pks) throws BusinessException;

}

4.分页查询接口实现，CostPriceBasePaginationServiceImpl。实现类实现了接口定义的两个方法，方法queryPKs根据查询条件从数据库中获取符合条件的所有主键，返回主键数组keys，可根据业务需求对keys进行过滤，筛选出需要的数据。方法queryBillByPK根据主键数组查询实际的VO。

注：ICostPriceBasePaginationService和CostPriceBasePaginationServiceImpl实现之后需要在upm文件中进行配置

<!--费用价格库分页查询 begin-->

<component priority="0" singleton="true" remote="true" tx="CMT" supportAlias="true"> <interface>nc.itf.mapub.costpricebase.ICostPriceBasePaginationService</interface> <implementation>nc.impl.mapub.costpricebase.CostPricePaginationBaseServiceImpl</implementation>

</component>

<!--费用价格库分页查询 end-->

package nc.impl.mapub.costpricebase;

import nc.impl.pubapp.pattern.data.bill.BillQuery;

import nc.impl.pubapp.pattern.database.DataAccessUtils;

import nc.itf.mapub.costpricebase.ICostPriceBasePaginationService;

import nc.ui.querytemplate.querytree.IQueryScheme;

import nc.vo.mapub.costpricebase.entity.CostPriceAggVO;

import nc.vo.mapub.costpricebase.entity.CostPriceHeadVO;

import nc.vo.pub.BusinessException;

import nc.vo.pubapp.bill.pagination.util.PaginationUtils;

import nc.vo.pubapp.pattern.data.IRowSet;

import nc.vo.pubapp.query2.sql.process.QuerySchemeProcessor;

public class CostPricePaginationServiceImpl implements ICostPriceBasePaginationService {

/\*

\*根据查询条件构造sql语句，并进行相应的筛选处理，返回主键数组

\*/

@Override

public String[] queryPKs(IQueryScheme queryScheme) throws BusinessException {

StringBuffer sql = new StringBuffer();

QuerySchemeProcessor processor = new QuerySchemeProcessor(queryScheme);

// 默认组织权限

processor.appendFuncPermissionOrgSql();

// 获得主表别名

String mainAlias = processor.getMainTableAlias();

sql.append("select ");

sql.append(mainAlias);

sql.append(".");

sql.append(CostPriceHeadVO.CCOSTPRICEID);

sql.append(processor.getFinalFromWhere()); // 获得加工之后最终的sql

sql.append(" order by ").append(CostPriceHeadVO.CREATETIME);

DataAccessUtils dao = new DataAccessUtils();

IRowSet rowSet = dao.query(sql.toString());

String[] keys = rowSet.toOneDimensionStringArray(); // 转化为一维字符串数组

return keys; //可根据业务需求对keys进行过滤

}

/\*根据主键数组查询

\*/

@Override

public CostPriceAggVO[] queryBillByPK(String[] pks) throws BusinessException {

// TODO Auto-generated method stub

CostPriceAggVO[] bills = null;

BillQuery<CostPriceAggVO> billQuery = new BillQuery<CostPriceAggVO>(CostPriceAggVO.class);

bills = billQuery.query(pks);

return PaginationUtils.filterNotExistBills(bills, pks);

}

}