

**BOS物流项目**

# 教学计划

1、分区组合条件分页查询

* + 分区分页查询（没有过滤条件）
  + 分区分页查询（带有过滤条件）

2、分区导出

* + 页面调整
  + 使用POI将数据写到Excel文件
  + 通过输出流进行文件下载

3、定区添加

* + 定区概念
  + 定区添加页面调整
  + 服务端实现

4、定区分页查询

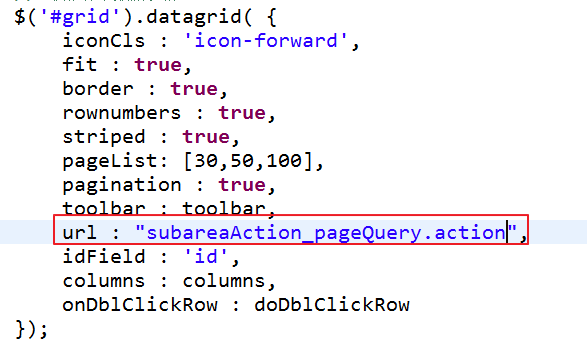
* + 页面调整
  + 服务端实现
  + 分页问题总结

# 分区组合条件分页查询

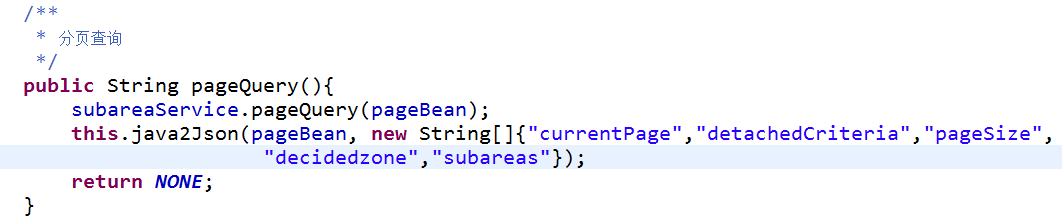
## 分区分页查询（没有过滤条件）

页面：WEB-INF/pages/base/subarea.jsp

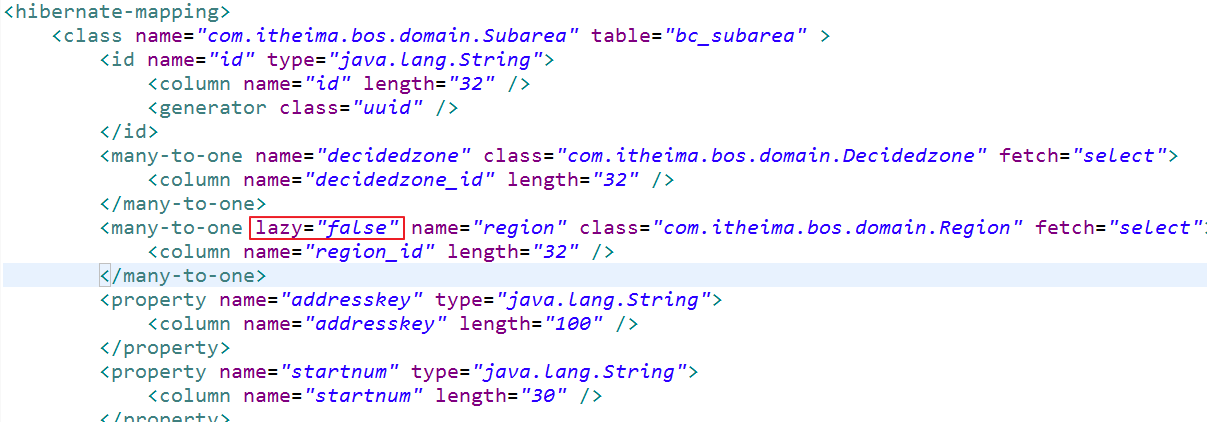
第一步：修改jsp页面中datagrid的URL地址



第二步：在SuareaAction中提供分页查询方法pageQuery



第三步：修改subarea.hbm.xml

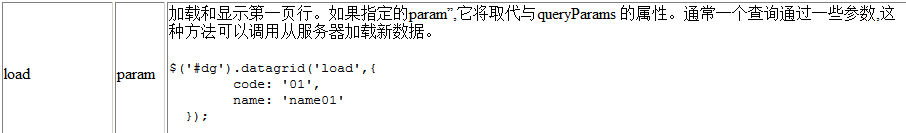


## 分区分页查询（带有过滤条件）

### 页面调整



datagrid提供的方法：用于重新发送ajax请求，并且可以提交参数



第一步：提供一个工具方法，可以将指定的form表单中所有的输入项转为json数据，用于参数提交

//定义一个工具方法，用于将指定的form表单中所有的输入项转为json数据{key:value,key:value}

$.fn.serializeJson=**function**(){

**var** serializeObj={};

**var** array=**this**.serializeArray();

$(array).each(**function**(){

**if**(serializeObj[**this**.name]){

**if**($.isArray(serializeObj[**this**.name])){

serializeObj[**this**.name].push(**this**.value);

}**else**{

serializeObj[**this**.name]=[serializeObj[**this**.name],**this**.value];

}

}**else**{

serializeObj[**this**.name]=**this**.value;

}

});

**return** serializeObj;

};

第二步：为查询窗口中查询按钮绑定事件

$("#btn").click(**function**(){

//将指定的form表单中所有的输入项转为json数据{key:value,key:value}

**var** p = $("#searchForm").serializeJson();

console.info(p);

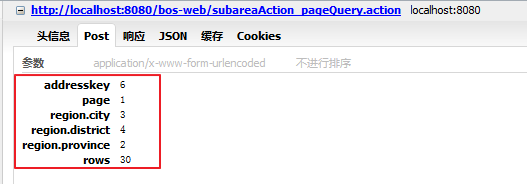
//调用数据表格的load方法，重新发送一次ajax请求，并且提交参数

$("#grid").datagrid("load",p);

//关闭查询窗口

$("#searchWindow").window("close");

});



### 修改Action中分页查询方法

/\*\*

\* 分页查询

\*/

**public** String pageQuery(){

DetachedCriteria dc = pageBean.getDetachedCriteria();

//动态添加过滤条件

String addresskey = model.getAddresskey();

**if**(StringUtils.*isNotBlank*(addresskey)){

//添加过滤条件，根据地址关键字模糊查询

dc.add(Restrictions.*like*("addresskey", "%"+addresskey+"%"));

}

Region region = model.getRegion();

**if**(region != **null**){

String province = region.getProvince();

String city = region.getCity();

String district = region.getDistrict();

dc.createAlias("region", "r");

**if**(StringUtils.*isNotBlank*(province)){

//添加过滤条件，根据省份模糊查询-----多表关联查询，使用别名方式实现

//参数一：分区对象中关联的区域对象属性名称

//参数二：别名，可以任意

dc.add(Restrictions.*like*("r.province", "%"+province+"%"));

}

**if**(StringUtils.*isNotBlank*(city)){

//添加过滤条件，根据市模糊查询-----多表关联查询，使用别名方式实现

//参数一：分区对象中关联的区域对象属性名称

//参数二：别名，可以任意

dc.add(Restrictions.*like*("r.city", "%"+city+"%"));

}

**if**(StringUtils.*isNotBlank*(district)){

//添加过滤条件，根据区模糊查询-----多表关联查询，使用别名方式实现

//参数一：分区对象中关联的区域对象属性名称

//参数二：别名，可以任意

dc.add(Restrictions.*like*("r.district", "%"+district+"%"));

}

}

subareaService.pageQuery(pageBean);

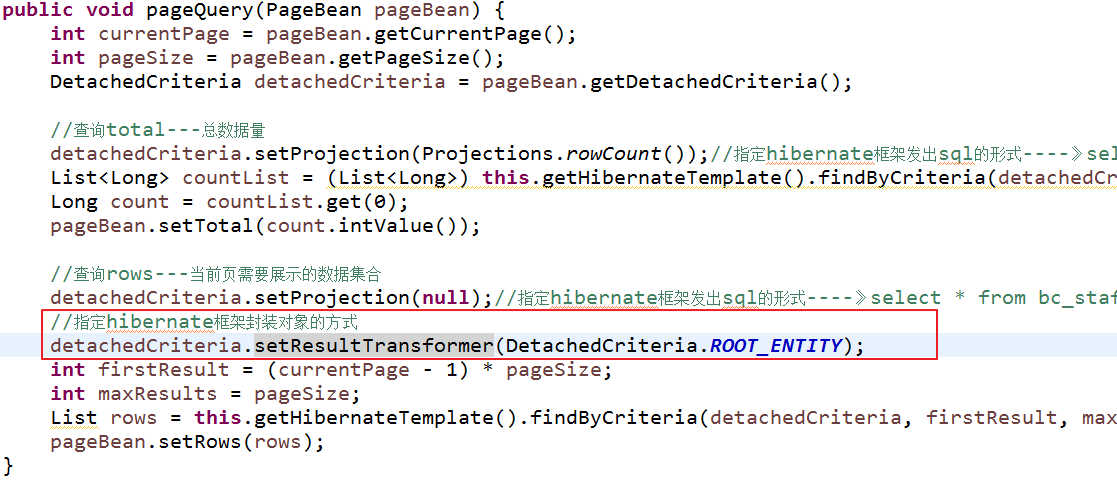
**this**.java2Json(pageBean, **new** String[]{"currentPage","detachedCriteria","pageSize",

"decidedzone","subareas"});

**return** ***NONE***;

}

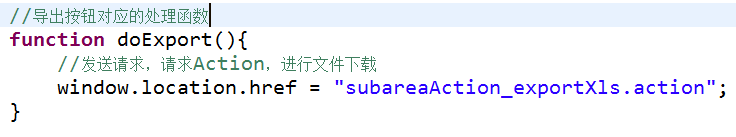
修改BaseDao中的分页查询方法



# 分区数据导出功能

## 页面调整

为页面中导出按钮绑定事件



## 服务端实现

第一步：查询所有的分区数据

第二步：使用POI将数据写到Excel文件中

第三步：使用输出流进行文件下载

/\*\*

\* 分区数据导出功能

\* **@throws** IOException

\*/

**public** String exportXls() **throws** IOException{

//第一步：查询所有的分区数据

List<Subarea> list = subareaService.findAll();

//第二步：使用POI将数据写到Excel文件中

//在内存中创建一个Excel文件

HSSFWorkbook workbook = **new** HSSFWorkbook();

//创建一个标签页

HSSFSheet sheet = workbook.createSheet("分区数据");

//创建标题行

HSSFRow headRow = sheet.createRow(0);

headRow.createCell(0).setCellValue("分区编号");

headRow.createCell(1).setCellValue("开始编号");

headRow.createCell(2).setCellValue("结束编号");

headRow.createCell(3).setCellValue("位置信息");

headRow.createCell(4).setCellValue("省市区");

**for** (Subarea subarea : list) {

HSSFRow dataRow = sheet.createRow(sheet.getLastRowNum() + 1);

dataRow.createCell(0).setCellValue(subarea.getId());

dataRow.createCell(1).setCellValue(subarea.getStartnum());

dataRow.createCell(2).setCellValue(subarea.getEndnum());

dataRow.createCell(3).setCellValue(subarea.getPosition());

dataRow.createCell(4).setCellValue(subarea.getRegion().getName());

}

//第三步：使用输出流进行文件下载（一个流、两个头）

String filename = "分区数据.xls";

String contentType = ServletActionContext.*getServletContext*().getMimeType(filename);

ServletOutputStream out = ServletActionContext.*getResponse*().getOutputStream();

ServletActionContext.*getResponse*().setContentType(contentType);

//获取客户端浏览器类型

String agent = ServletActionContext.*getRequest*().getHeader("User-Agent");

filename = FileUtils.*encodeDownloadFilename*(filename, agent);

ServletActionContext.*getResponse*().setHeader("content-disposition", "attachment;filename="+filename);

workbook.write(out);

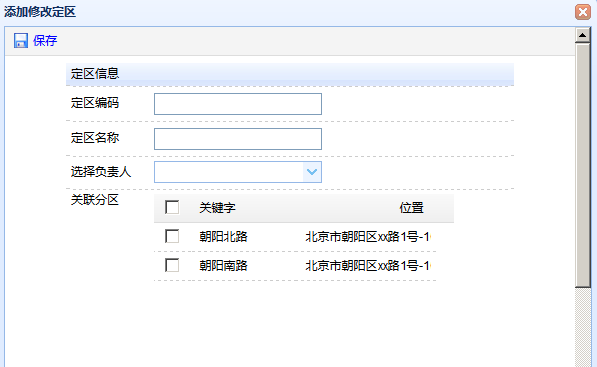
**return** ***NONE***;

}

# 定区添加

定区是进行物流分配的基本单位，可以将取派员、分区、客户信息进行关联，为自动分单提供数据。

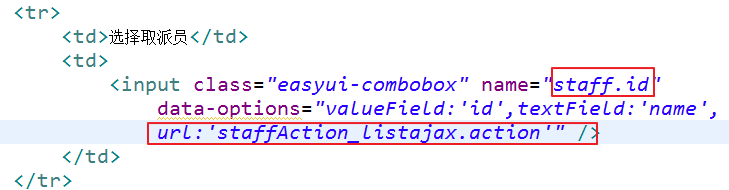
页面：WEB-INF/pages/base/decidedzone.jsp



## 页面调整

### 使用combobox展示取派员数据

第一步：修改定区页面中combobox下拉框URL地址





第二步：在StaffAction中提供listajax方法，查询所有未删除的取派员，返回json

/\*\*

\* 查询所有未删除的取派员，返回json

\*/

**public** String listajax(){

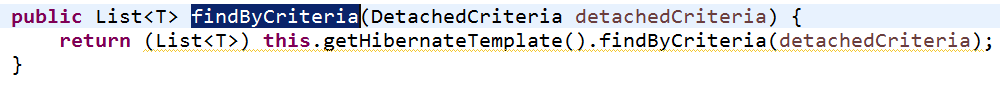
List<Staff> list = staffService.findListNotDelete();

**this**.java2Json(list, **new** String[]{"decidedzones"});

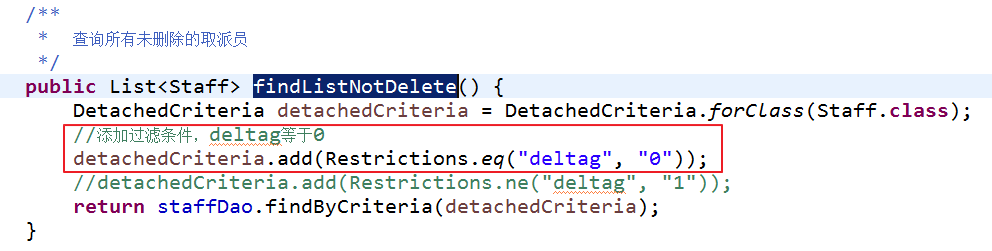
**return** ***NONE***;

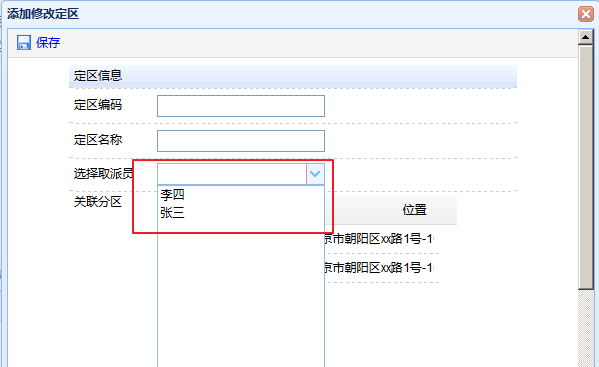
}

第三步：在BaseDao中扩展一个通用查询方法



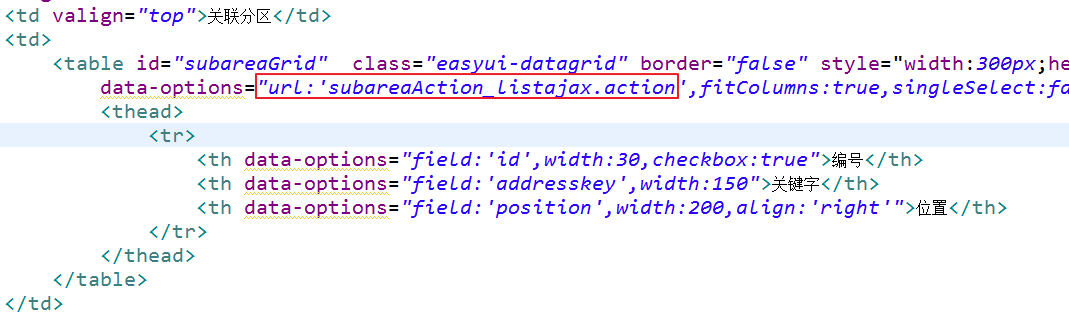
第四步：在StaffService中扩展方法，查询未删除的取派员



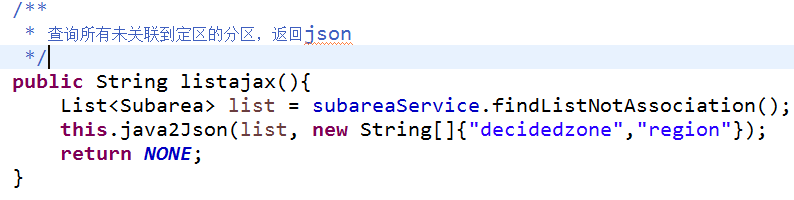


### 使用datagrid展示分区数据

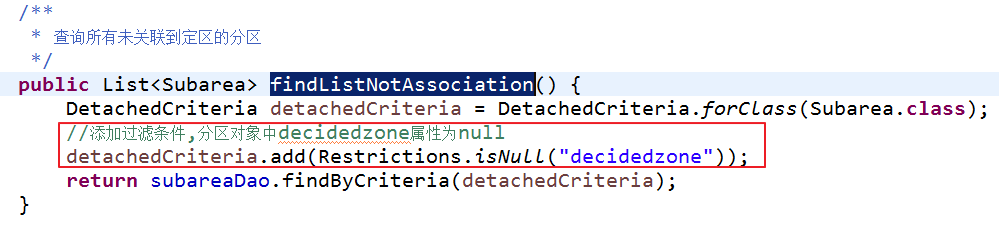
第一步：修改页面中datagrid的URL地址



第二步：在SubareaAction中提供listajax方法，查询所有未关联到定区的分区，返回json

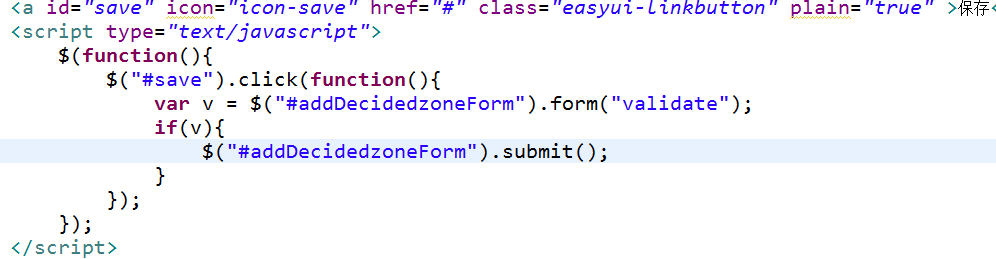


第三步：在SubareaService中扩展方法，查询未关联到定区的分区

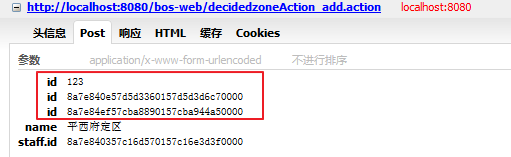




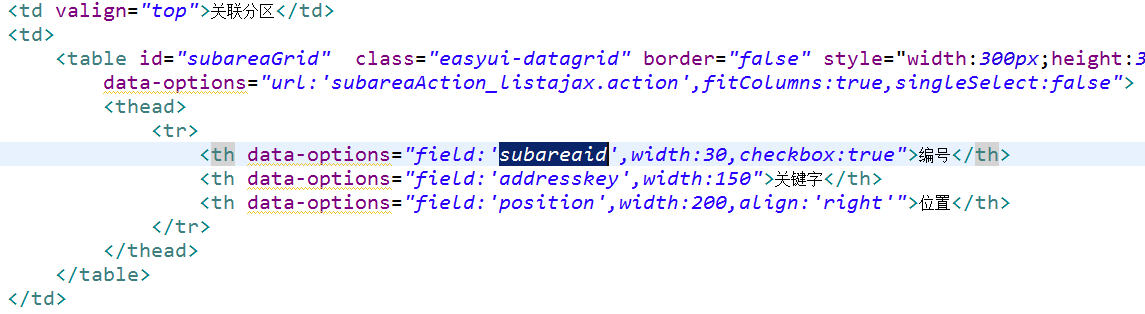
### 为保存按钮绑定事件提交表单



问题一：提交的表单中存在多个id参数



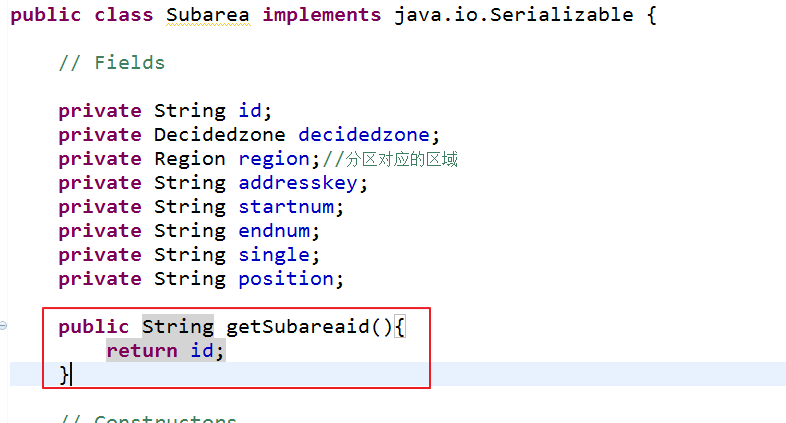
解决方案：将datagrid的filed由id改为subareaid

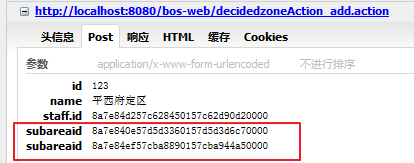


问题二：提交的表单中subareaid参数的值为null



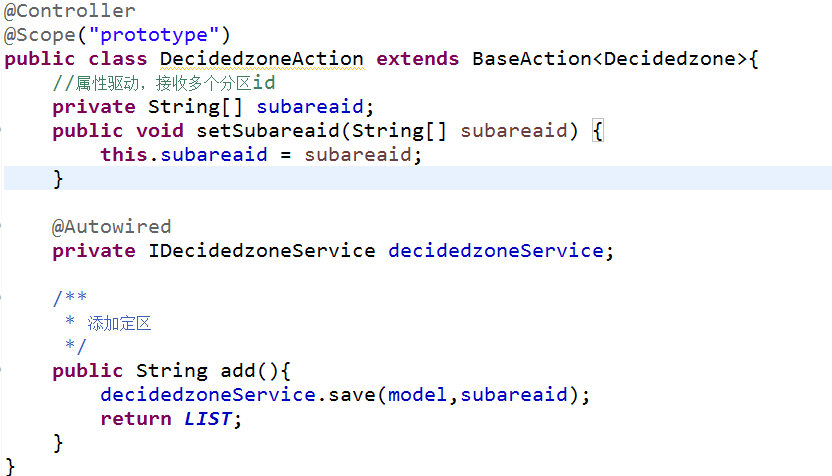
解决方案：在分区类中提供getSubareaid方法



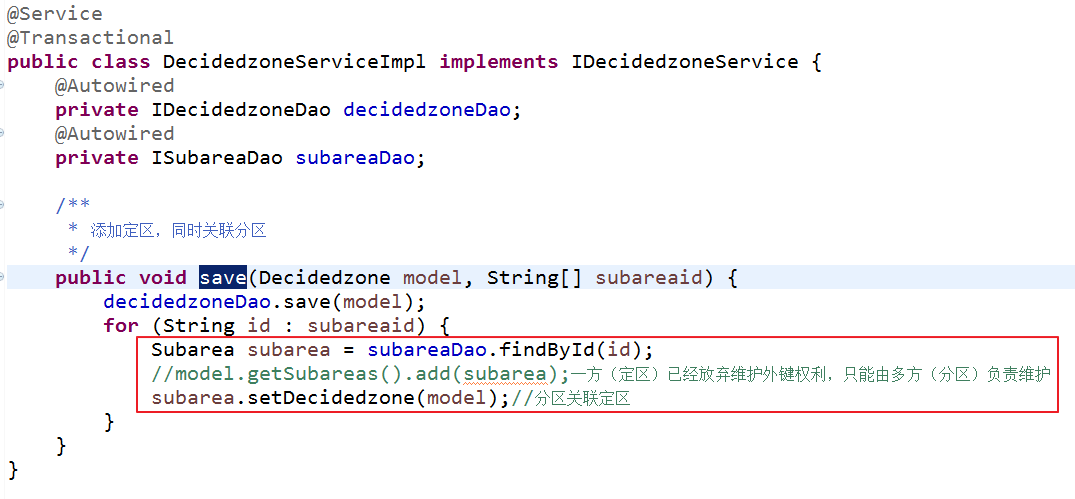


## 服务端实现

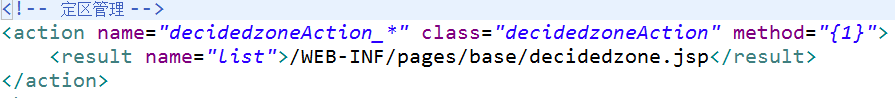
创建定区的Action、Service、Dao



Service代码：

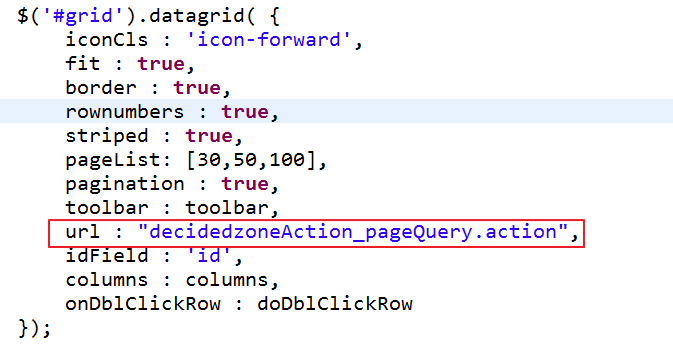


配置struts.xml

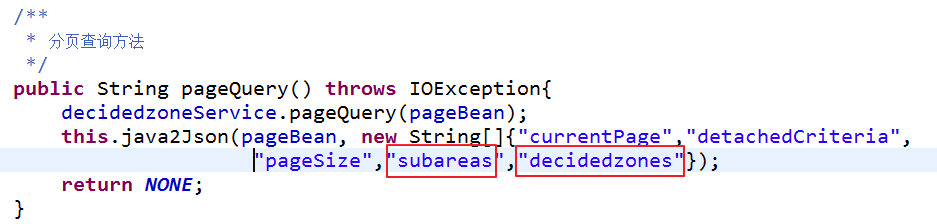


# 定区分页查询

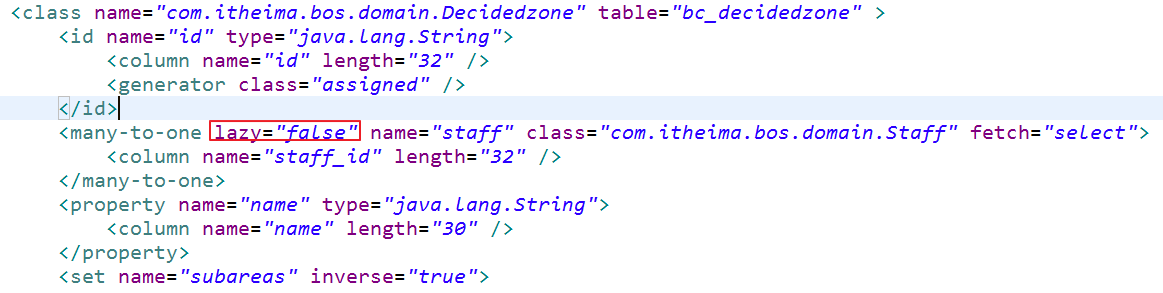
第一步：修改定区jsp页面中datagrid的URL地址



第二步：在定区Action中提供pageQuery方法



第三步：在Decidedzone.hbm.xml中修改，查询定区对象时需要立即加载关联的取派员对象



# 分页查询死循环问题

1、页面不需要展示关联数据时

解决：将关联对象属性排除掉

2、页面需要展示关联数据时

解决：将关联对象改为立即加载，并且将关联对象中的属性排除