

**BOS物流项目**

# 教学计划

1、实现区域导入功能

* + OCUpload一键上传插件使用
  + 将文件上传到Action
  + POI简介
  + 使用POI解析Excel文件
  + 完成数据库操作
  + 使用pinyin4J生成简码和城市编码

2、区域分页查询

* + 页面调整
  + 服务端实现

3、重构分页代码

* + BaseAction
  + 子类Action

4、分区添加功能

* + 什么是分区
  + 页面调整（combobox使用）
  + 服务端实现

# 实现区域导入功能

## jquery OCUpload一键上传插件使用

OCUpload（One Click Upload）

第一步：将js文件引入页面

<script type=*"text/javascript"* src=*"*${pageContext.request.contextPath }*/js/jquery-1.8.3.js"*></script>

<script type=*"text/javascript"* src=*"*${pageContext.request.contextPath }*/js/jquery.ocupload-1.1.2.js"*></script>

第二步：在页面中提供任意一个元素



第三步：调用插件提供的upload方法，动态修改HTML页面元素

<script type="text/javascript">

$(**function**(){

//页面加载完成后，调用插件的upload方法，动态修改了HTML页面元素

**$("#myButton").upload({**

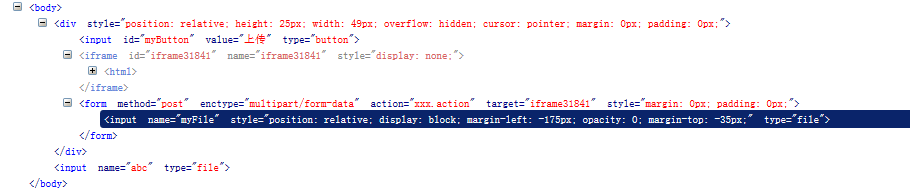
**action:'xxx.action',//链接**

**name:' regionFile '//这个的名字一定要和action对应（文件名）**

**});**

});

</script>



## 在服务端接收上传的文件

在Action中提供一个File类型的属性，名称和上传的文件输入框名称一致regionFile

@Controller

@Scope("prototype")

**public** **class** RegionAction **extends** BaseAction<Region>{

//属性驱动，接收上传的文件

**private** File regionFile;

**public** **void** setRegionFile(File regionFile) {

**this**.regionFile = regionFile;

}

/\*\*

\* 区域导入

\*/

**public** String importXls(){

System.***out***.println(regionFile);

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

}

}

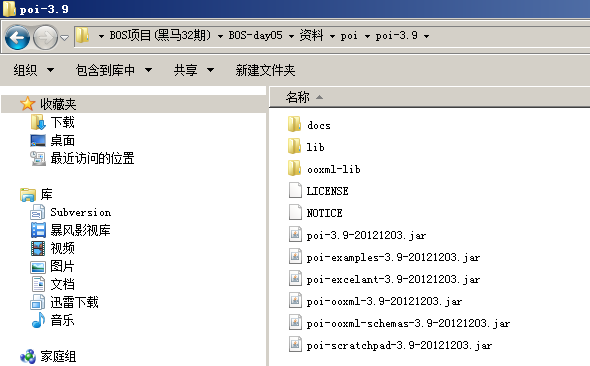
## apache POI技术

Apache POI是[Apache软件基金会](http://baike.baidu.com/view/7044910.htm" \t "_blank)的开放源码函式库，POI提供API给Java程序对Microsoft Office格式档案读和写的功能。

下载开发包：



解压上面的zip文件：



在项目中引入POI的依赖：

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.11</version>

</dependency>

POI使用：

@Test

**public** **void** test1() **throws** FileNotFoundException, IOException{

String filePath = "C:\\Users\\zhaoqx\\Desktop\\BOS项目(黑马32期)\\BOS-day05\\资料\\区域导入测试数据.xls";

//包装一个Excel文件对象 （参数是文件的输入流 可以进行读的操作）

HSSFWorkbook workbook = **new** HSSFWorkbook(**new** FileInputStream(**new** File(filePath)));

//读取文件中第一个Sheet标签页

HSSFSheet hssfSheet = workbook.getSheetAt(0);

//遍历标签页中所有的行

**for** (Row row : hssfSheet) {//row表示每行数据

System.***out***.println();

**for** (Cell cell : row) {//表示每个单元格

String value = cell.getStringCellValue();

System.***out***.print(value + " ");

}

}

}

## pinyin4J



<!-- 引入pinyin4J的依赖 -->

<dependency>

<groupId>com.belerweb</groupId>

<artifactId>pinyin4j</artifactId>

<version>2.5.0</version>

</dependency>

@Test

**public** **void** test1(){

//河北省 石家庄市 桥西区

String province = "河北省";

String city = "石家庄市";

String district = "桥西区";

//简码---->>HBSJZQX

province = province.substring(0, province.length() - 1);

city = city.substring(0, city.length() - 1);

district = district.substring(0, district.length() - 1);

String info = province + city + district;

String[] headByString = PinYin4jUtils.*getHeadByString*(info);

//得到每个字符的的首字母 返回一个字符串数组

String shortcode = StringUtils.*join*(headByString);

System.***out***.println(shortcode);

//城市编码---->>shijiazhuang

String citycode = PinYin4jUtils.*hanziToPinyin*(city, "");

//汉字转拼音

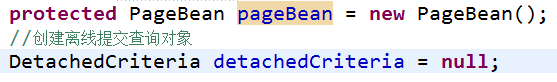
System.***out***.println(citycode);

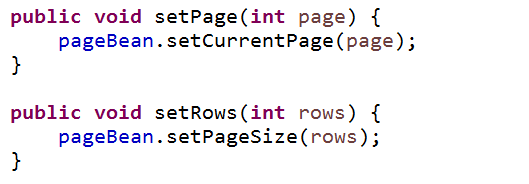
}

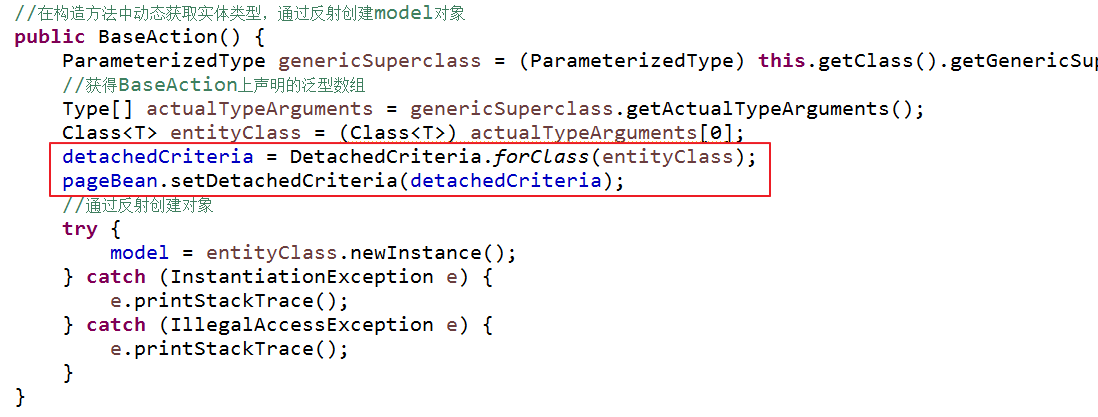
# 区域数据分页查询

# 重构分页代码

将重复分代码统一抽取到BaseAction中







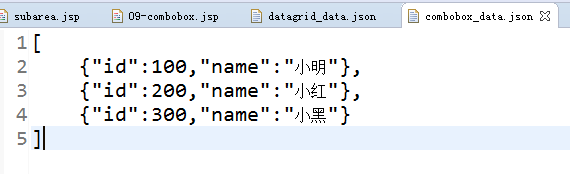


# 分区数据添加功能

区域范围很大，对区域进行细分产生分区

## easyUI combobox下拉框使用

制作一个json文件：

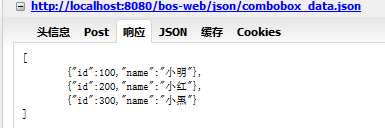


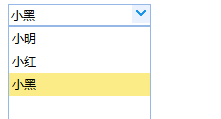
<input data-options=*"url:'*${pageContext.request.contextPath }*/json/combobox\_data.json',*

*valueField:'id',*

*textField:'name'"*

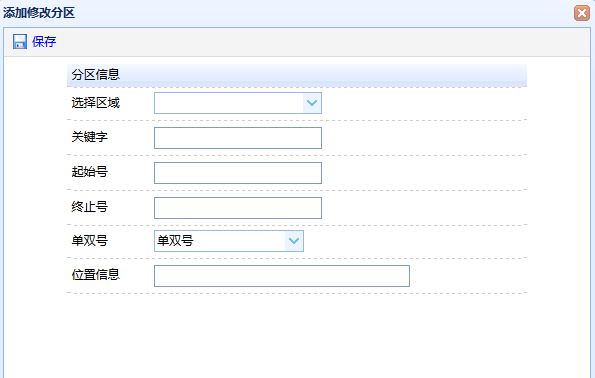
class=*"easyui-combobox"*>





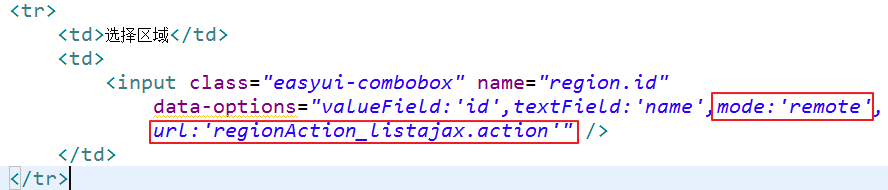
## 使用combobox展示区域数据

页面：WEB-INF/pages/base/subarea.jsp



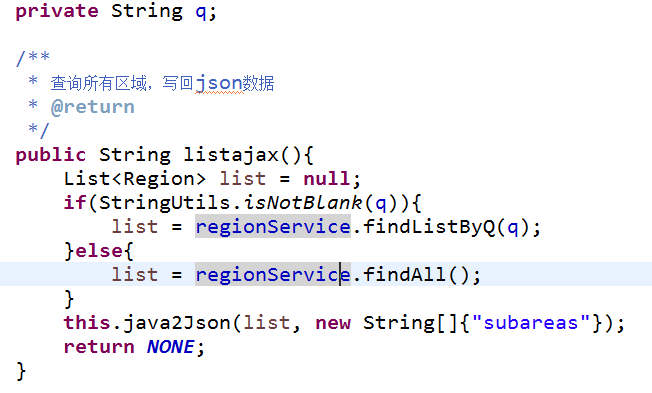


第一步：修改页面中combobox：

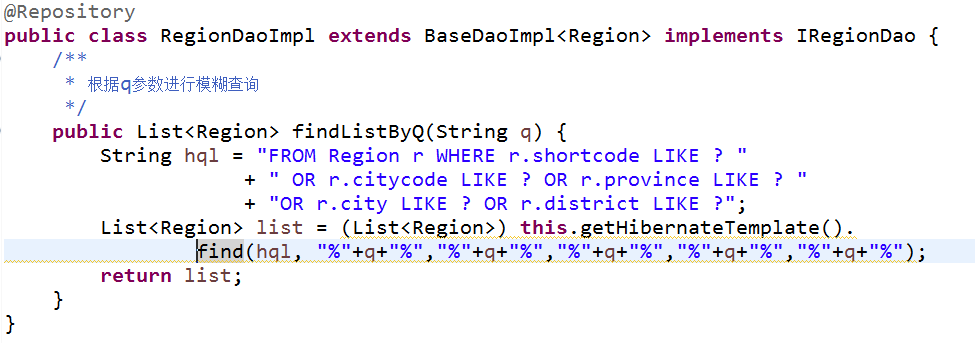




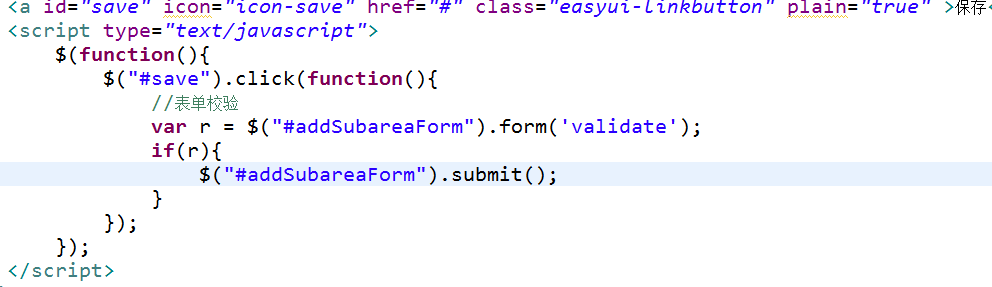
第二步：在RegionAction中提供listajax方法



第三步：在RegionDao中扩展方法



第四步：为添加窗口中保存按钮绑定事件





## 服务端实现

创建分区对应的Action、Service、Dao

# 分页查询问题

将Java对象转为json过程中，因为对象之间有相互引用关系，会发生死循环问题，解决方式，排除其中属性

