

**BOS物流项目**

# 教学计划

1、定区关联客户

* + 完善CRM服务中的客户查询方法
  + 在BOS项目中配置代理对象远程调用crm服务
  + 调整定区关联客户页面
  + 实现定区关联客户

2、查看定区中包含的分区

* + 页面调整
  + 服务端实现

3、查看定区关联的客户列表数据

* + 页面调整
  + 服务端通过代理对象远程调用CRM服务活动客户信息

# 定区关联客户

## 在BOS项目中配置代理对象远程调用crm

第一步：在BOS项目的pom.xml中引入CXF的依赖

<dependency>

<groupId>org.apache.cxf</groupId>

<artifactId>cxf-rt-frontend-jaxws</artifactId>

<version>3.0.1</version>

</dependency>

<dependency>

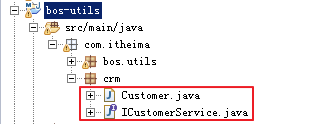
<groupId>org.apache.cxf</groupId>

<artifactId>cxf-rt-transports-http</artifactId>

<version>3.0.1</version>

</dependency>

第二步：使用wsimport命令解析wsdl文件生成本地代码，只需要接口文件和实体类



第三步：在spring配置文件中注册crm客户端代理对象

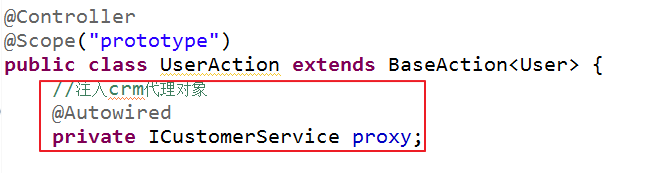
<!-- 注册crm客户端代理对象 -->

<jaxws:client id=*"crmClient"*

serviceClass=*"com.itheima.crm.ICustomerService"*

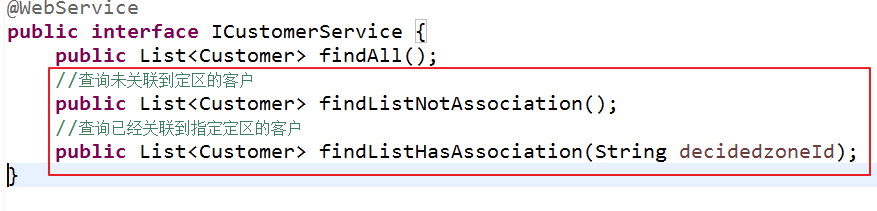
address=*"http://192.168.115.89:8080/crm\_heima32/service/customer"*/>

第四步：通过注解方式将代理对象注入给Action



## 完善CRM服务中的客户查询方法

* 在crm服务中的接口中扩展两个方法



* 在实现类中实现方法

//查询未关联到定区的客户

**public** List<Customer> findListNotAssociation() {

String sql = "select \* from t\_customer where decidedzone\_id is null";

List<Customer> list = jdbcTemplate.query(sql, **new** RowMapper<Customer>(){

**public** Customer mapRow(ResultSet rs, **int** arg1) **throws** SQLException {

**int** id = rs.getInt("id");//根据字段名称从结果集中获取对应的值

String name = rs.getString("name");

String station = rs.getString("station");

String telephone = rs.getString("telephone");

String address = rs.getString("address");

String decidedzone\_id = rs.getString("decidedzone\_id");

**return** **new** Customer(id, name, station, telephone, address, decidedzone\_id);

}

});

**return** list;

}

//查询已经关联到指定定区的客户

**public** List<Customer> findListHasAssociation(String decidedzoneId) {

String sql = "select \* from t\_customer where decidedzone\_id = ?";

List<Customer> list = jdbcTemplate.query(sql, **new** RowMapper<Customer>(){

**public** Customer mapRow(ResultSet rs, **int** arg1) **throws** SQLException {

**int** id = rs.getInt("id");//根据字段名称从结果集中获取对应的值

String name = rs.getString("name");

String station = rs.getString("station");

String telephone = rs.getString("telephone");

String address = rs.getString("address");

String decidedzone\_id = rs.getString("decidedzone\_id");

**return** **new** Customer(id, name, station, telephone, address, decidedzone\_id);

}

},decidedzoneId);

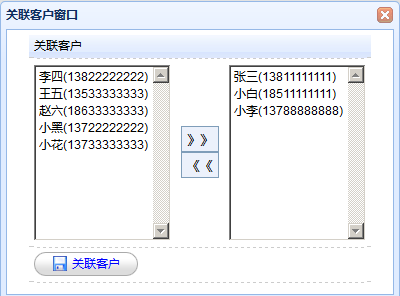
**return** list;

}

注意：服务端扩展了新的方法，需要客户端重新生成客户端代码

## 调整定区关联客户页面

页面：WEB-INF/pages/base/decidedzone.jsp



第一步：修改定区页面中关联客户按钮绑定事件处理函数

**function** doAssociations(){

//获取当前数据表格所有选中的行，返回数组

**var rows = $("#grid").datagrid("getSelections");//得到的是选中的行 json 对象的数组**

**if**(rows.length != 1){

//弹出提示

$.messager.alert("提示信息","请选择一个定区操作！","warning");

//类比 alter 这个有标题有图标

}**else**{

//选中了一个定区

$('#customerWindow').window('open');

//类比$('#customerWindow').hide()

//清理下拉框

$("#noassociationSelect").empty();

$("#associationSelect").empty();

//发送ajax请求，请求定区Action，在定区Action中通过crm代理对象完成对于crm服务远程调用获取客户数据

**var** url\_1 = "decidedzoneAction\_findListNotAssociation.action";

$.post(url\_1,**function**(data){

**//遍历json数组**

**for**(**var** i=0;i<data.length;i++){

**var** id = data[i].id;

**var** name = data[i].name;

**var** telephone = data[i].telephone;

name = name + "(" + telephone + ")";

**$("#noassociationSelect").append("<option value='"+id+"'>"+name+"</option>");**

},

“json”

});

//发送ajax请求，请求定区Action，在定区Action中通过crm代理对象完成对于crm服务远程调用获取客户数据

**var** url\_2 = "decidedzoneAction\_findListHasAssociation.action";

**var** decidedzoneId = rows[0].id;

$.post(url\_2,{"id":decidedzoneId},**function**(data){

//遍历json数组

**for**(**var** i=0;i<data.length;i++){

**var** id = data[i].id;

**var** name = data[i].name;

**var** telephone = data[i].telephone;

name = name + "(" + telephone + ")";

**$("#associationSelect").append("<option value='"+id+"'>"+name+"</option>");**

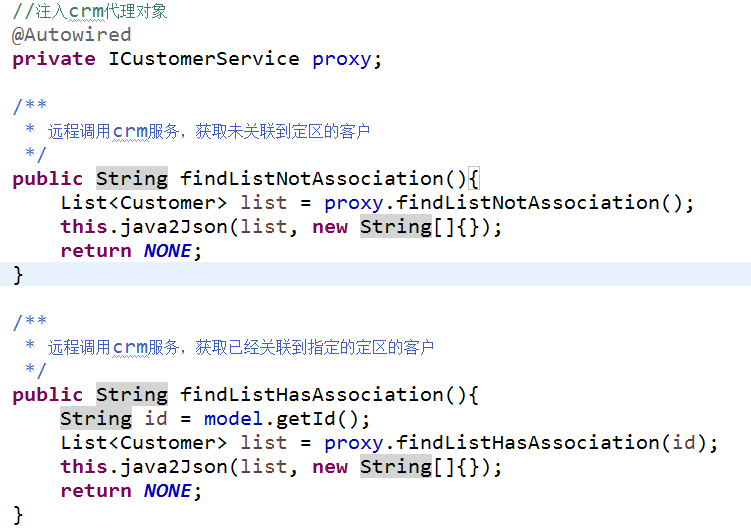
}

});

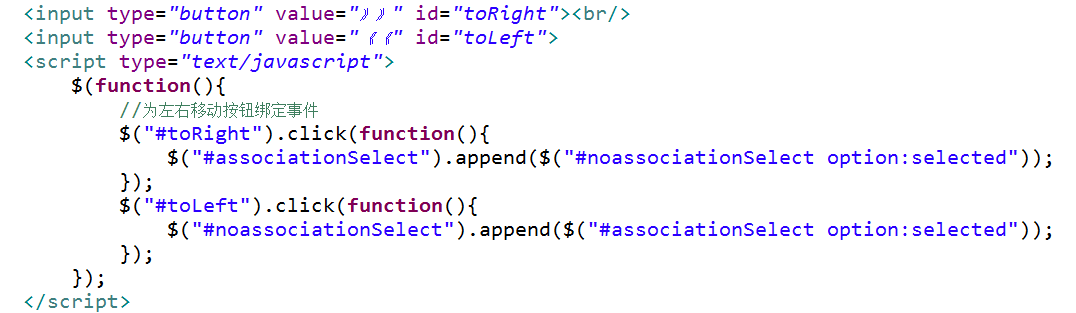
}

}

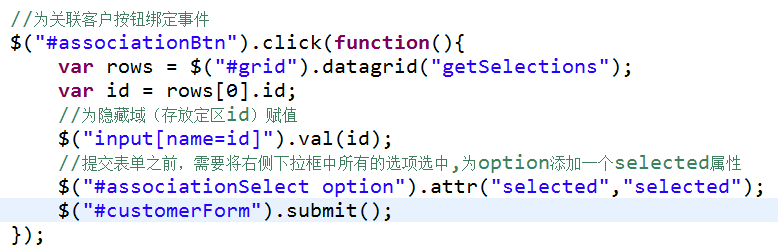
第二步：在定区Action中注入crm代理对象，并且提供方法远程调用crm服务

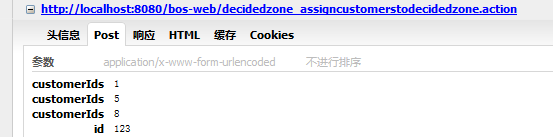


**第三步：为关联客户窗口中左右移动两个按钮绑定事件**

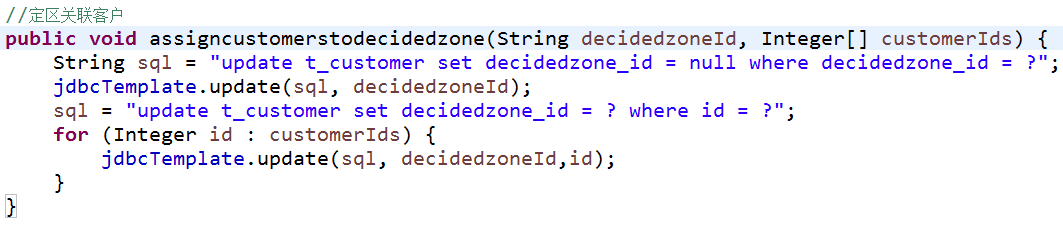


第四步：为定区关联客户窗口中的“关联客户”按钮绑定事件



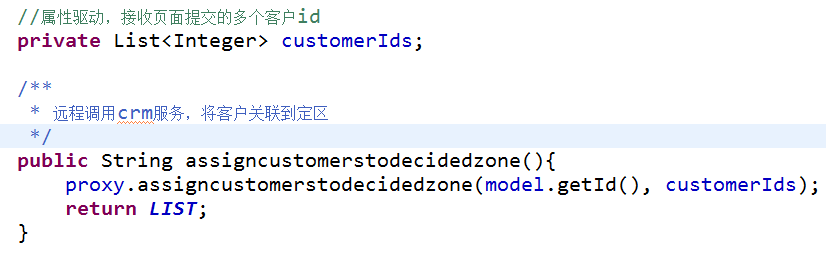


## 在crm服务端扩展定区关联客户方法



注意：crm服务端扩展方法后，需要bos客户端重新生成客户端代码

## 在定区Action中提供关联客户方法

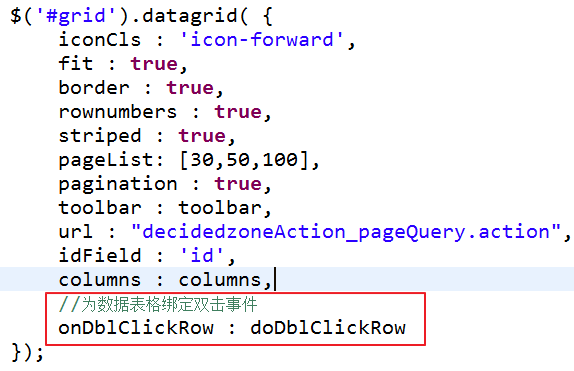


# 查看定区中包含的分区

页面：WEB-INF/pages/base/decidedzone.jsp

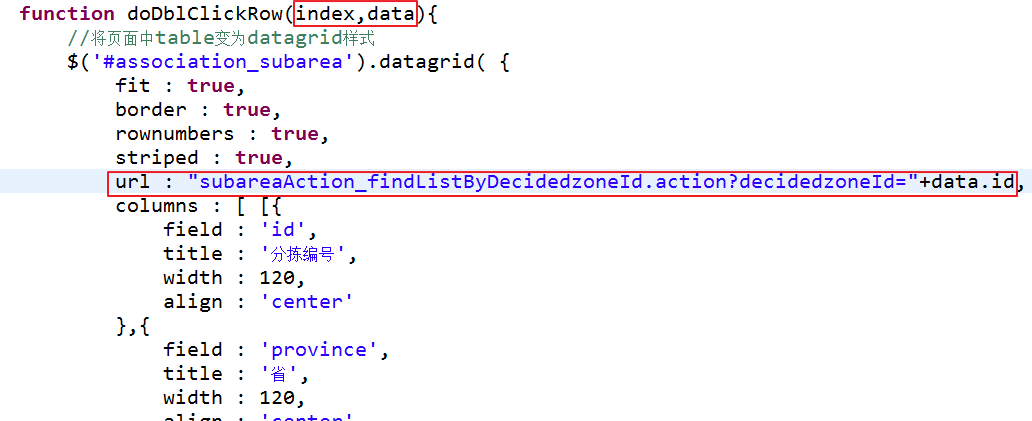
## 页面调整

第一步：为datagrid绑定数据事件



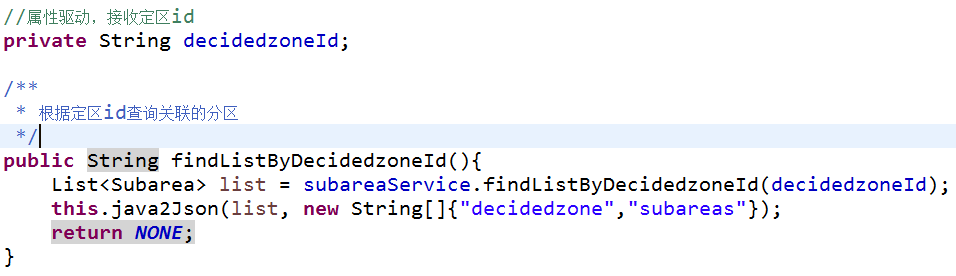
选中是有的 但是双击事件需要自己指定

第二步：定义doDblClickRow函数

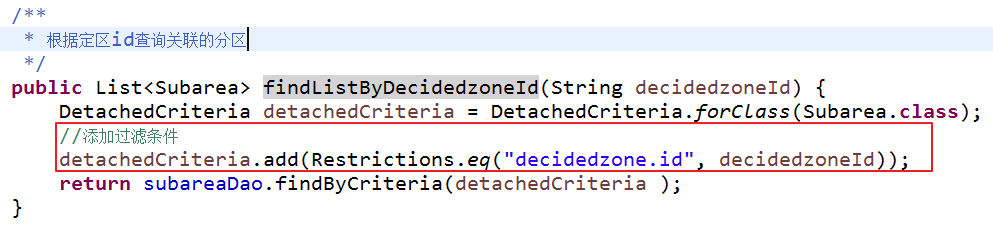


## 服务端实现

* 在分区Action中提供方法，根据定区id查询关联的分区



* 在SubareaServiceImpl中提供方法，根据定区id查询关联的分区



# 查看定区中关联的客户信息

