代码生成二次开发文档

# 简介

里面的文档可能图片偏多，主要是因为基于您是开发人员，有些说明和讲解还不如一个页面代码图片说明来得更直接，因为本人是开发人员，所以一些浅而易见的逻辑和说明就直接化为图片讲解了

# 概要说明

这个文档介绍关于通过表单生成代码具体使用，另外介绍如果直接开发进行代码开发时，如何使用页面的控件。

控件主要包括：

1.表单验证js

2.日历控件

3.数据字典

4.office控件

5.自定义对话框

6.用户选择器

7.附件选择控件

注意：控件的介绍只是介绍最基础的用法，还有更多细节和深层的用法可自己到指定的js里面查看

# 具体介绍

## 页面组件

### 表单验证js

#### 概要

表单验证主要用于form表单提交前进行表单的字段合法性判，不合法会默认在字段html后显示错误信息，错误信息的显示地方其实也可以配置的

#### 使用方法

|  |  |
| --- | --- |
| 引入js | <script type="text/javascript" src="${ctx}/js/hotent/CustomValid.js"></script> |
| 在jsp写法 | 在需要做验证的输入框，单选框，多选框，下拉框中加入validate属性validate:写法如下：{required:true,maxLength:50} required:必填；maxLength:最长长度……等等，在CustomValid.js里面有很多验证规则，具体参数和用法可自己去看一下，js顶部还有其更加高级的用法注解 |
| demo | <script type="text/javascript">  funtion submit(){  var frm=$('#form').form();//获取表单  if(frm.valid()){//检验表单  $('#form').submit();//验证成功就提交表单  }  }  </script>  <form id="form" method="post" action="save.ht">  <input name="name" type="text" validate="{required:true,maxlength:64}" /><!--必填，长度最大为64-->  <input type="button" onclick="submit" value="提交"/>  </form> |

### 日期控件

#### 概要

表单中的日期输入框

#### 使用方法

|  |  |
| --- | --- |
| 引入js | <script type="text/javascript" src="${ctx}/js/calendar/My97DatePicker/WdatePicker.js" ></script> |
| 使用介绍 | 只要在input中引入class=" Wdate "则可，具体操作看demo |
| demo | <input type="text" class=" Wdate" /> |

### 数据字典

#### 概要

数据字典，是利用了easyui的ComboTree来实现的，所以使用的时候要引入easyui包

#### 使用说明

|  |  |
| --- | --- |
| 引入js | <script type="text/javascript" src="${ctx}/js/lg/plugins/htDicCombo.js"></script> |
| 使用介绍 | 在一个input标签中，引入class="dicComboTree" 然后设置nodekey="某个数据字典key" 则可，也可设置returnField="系统内置Dictionary对象的任意属性"来设置选择树节点时的返回值，默认为dicId，nodekey可在系统这里查看 |
| demo | 省份数据字典的    <input type="text" class="dicComboTree" nodekey="sf" value="${user.home}" returnField="itemName"/> |

### office控件

#### 概要

我们系统里集合了软航office控件，这里是基于您系统能正常运行office控件的前提下使用的，如有还不能正常运行，可看其他文档。

本系统的保存实现如下，先保存office文件后，把它的文件ID(fileId)返回字段，然后下次根据fileId来打开office。

#### 使用方法

|  |  |
| --- | --- |
| 引入js | <script type="text/javascript" src="${ctx}/codegen/js/OfficePlugin.js"></script><script type="text/javascript" src="${ctx}/codegen/js/OfficeControl.js"></script> |
| 使用介绍 | 在一个input标签添加controltype="office" 则可，如果设置了value="某个office的fileID" 那么在页面初始化的时候，会打开这个文件，在提交表单的时候，需要先提交office文档，后再提交表单。具体实现和代码在demo会详说 |
| demo | 在Jsp页面中，  <input type="hidden" class="hidden" name="office" value="${office}" controltype="office" />  先把office文件提交到后台，在成功保存office文档后再提交本地表单    if(frm.valid()){  OfficePlugin.submit(function(){//提交成功后再提交本地表单  frm.sortList();  $('#zxdmsckjcsForm').submit();  });  }这里把提交成功后的执行代码以回调函数的方式传过去。 |

### 自定义对话框介绍

#### 概要

调用系统自定义对话框，这些对话框可以用户到系统那配置。

#### 使用方法

|  |  |
| --- | --- |
| 引入js | <script type="text/javascript" src="${ctx}/codegen/js/FormUtil.js"></script>  <script type="text/javascript" src="${ctx}/codegen/js/FormInit.js"></script> |
| 使用说明 | 在一个a标签内设置dialog={}属性，dialog属性如下：  name:对话框别名  fields://字段设置  [  src:对话框的其中一个字段  target:把对话框的返回值填到页面那个字段，按照$('[name=xxxxx]'来获取对象  ]  query:查询条件，eg: query:[{'id':1,'name':'a','name':'b'}] |
| demo | <a dialog="{name:'csdhk',fields:[{src:'FULLNAME',target:'zbzdy'}],query:[]}"  href="javascript:;" class="extend message">自定义按钮</a> |

### 系统内置的选择器对话框，包括：users(用户) orgs(组织) position(岗位) roles(角色)

#### 概要：

系统内置的选择器，跟自定义对话框不一样，这些选择器更加灵活和功能专一。

#### 使用方法

|  |  |
| --- | --- |
| 引入js | <script type="text/javascript" src="${ctx}/codegen/js/FormUtil.js"></script>  <script type="text/javascript" src="${ctx}/codegen/js/FormInit.js"></script> |
| 使用方法 | 系统选择器别名列表：  users(用户) orgs(组织) position(岗位) roles(角色) |
| demo | 1 先引入一个对话框按钮    2添加返回值的载体    <a href="javascript:;" class="link users" name="user">选择</a>  <input type="text" name="user"/>  <input type="text" name="userID"/> |

### 附件选择控件

#### 概要

调用系统内置的附件对话框

#### 使用方法

|  |  |
| --- | --- |
| 引入 | <script type="text/javascript" src="${ctx}/codegen/js/AttachMent.js"></script> |
| 使用方法 | 配置信息比较少，按照demo来使用比较直接 |
| demo | <div name="div\_attachment\_container">  <div class="attachement"></div>  <textarea name="fj" controltype="attachment" style="display:none;">${fj}</textarea>  <a href="javascript:;" class="link selectFile" field="fj" onclick="AttachMent.addFile(this);" name="fj">选择</a>  </div> |

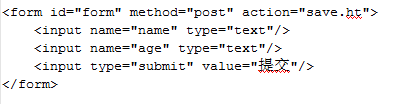
## 代码生成介绍

### 数据封装原理

本系统生成的代码都是以springmvn绑定数据原理来提交表单的，由于这个是springmvc的技术，这里就不细说，以一个简单的例子来说明便可。

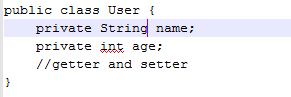
spring-mvc绑定数据原理介绍： http://www.cnblogs.com/xd502djj/archive/2013/08/08/3245485.html

页面表单：

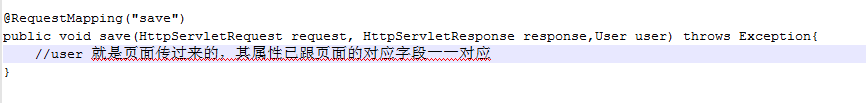


后台代码

model:



controller:



这样的好处在于不用在后台再逐个逐个属性拿出来，再封装user

### 系统新页面开发的基本过程

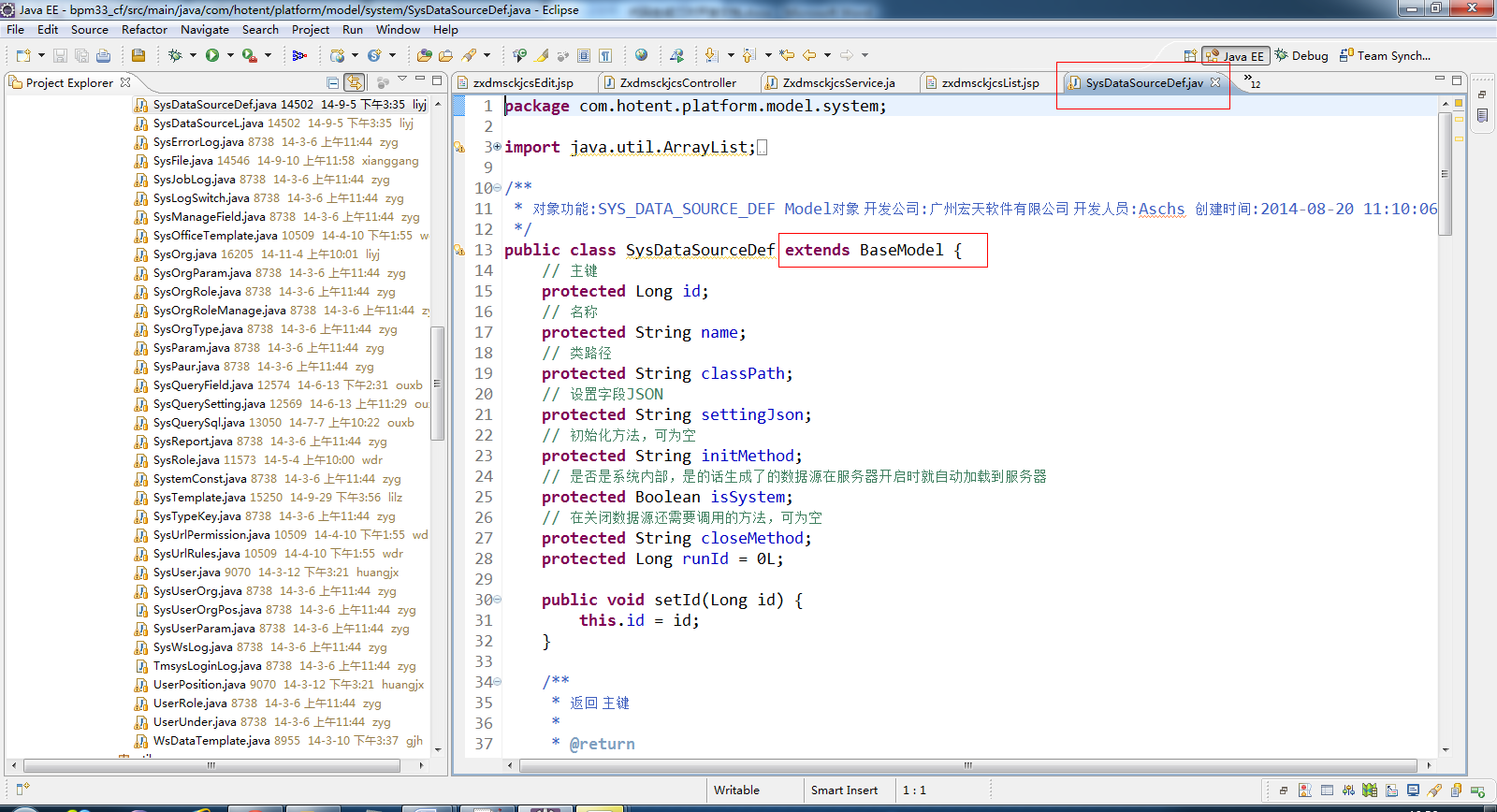
#### 概要

这里是大概说一下新功能开发步骤，实质开发时可以随便打开一个对应的文件来对比来做，如，我要做一个新model，打开一个旧的Model来对应来写则可，这个文档也是给有开发经验的用户看的。

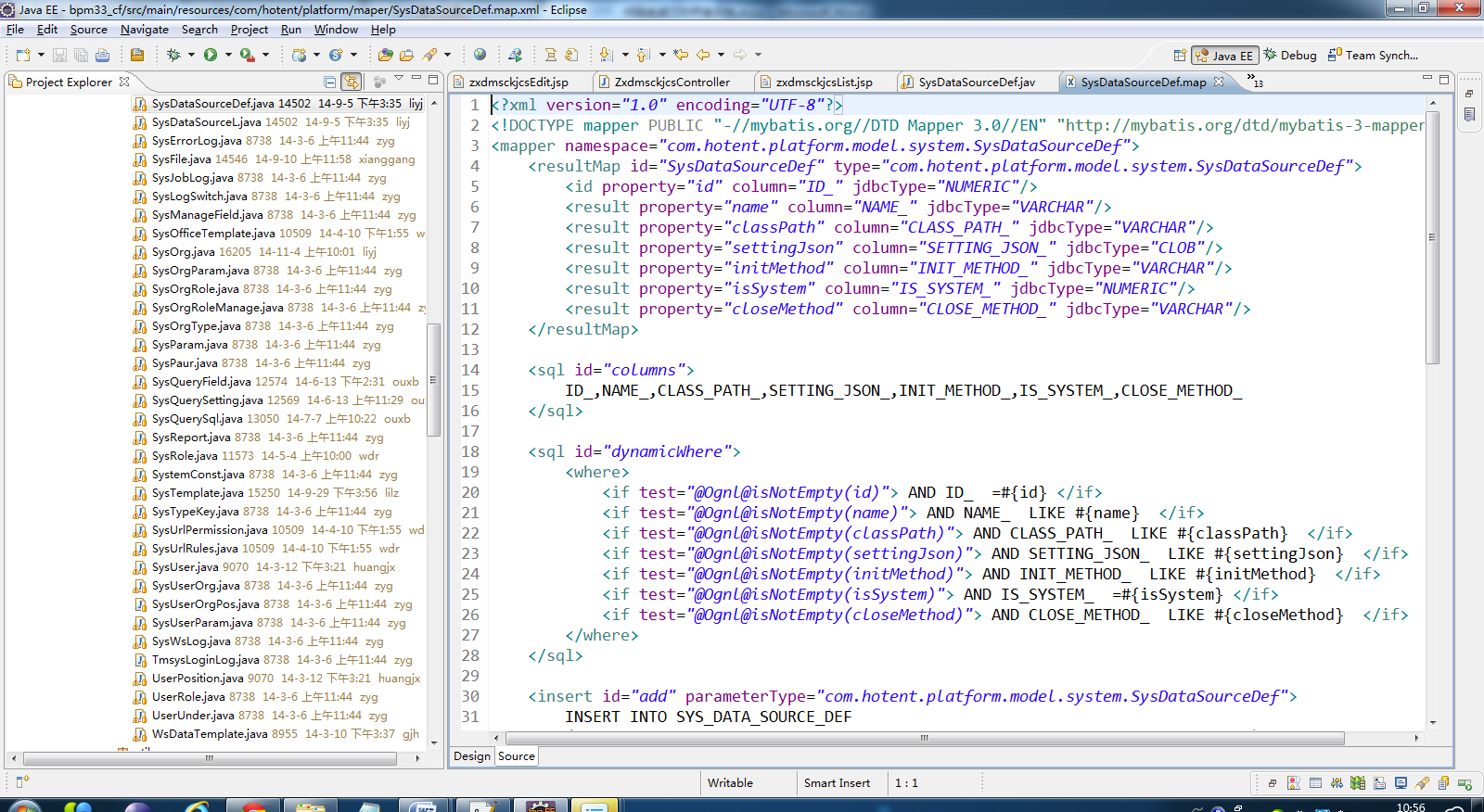
#### 具体过程

1.先定义好Model，当中包括数据库加入表和mybatis的map.xml文件。

eg:

model: 

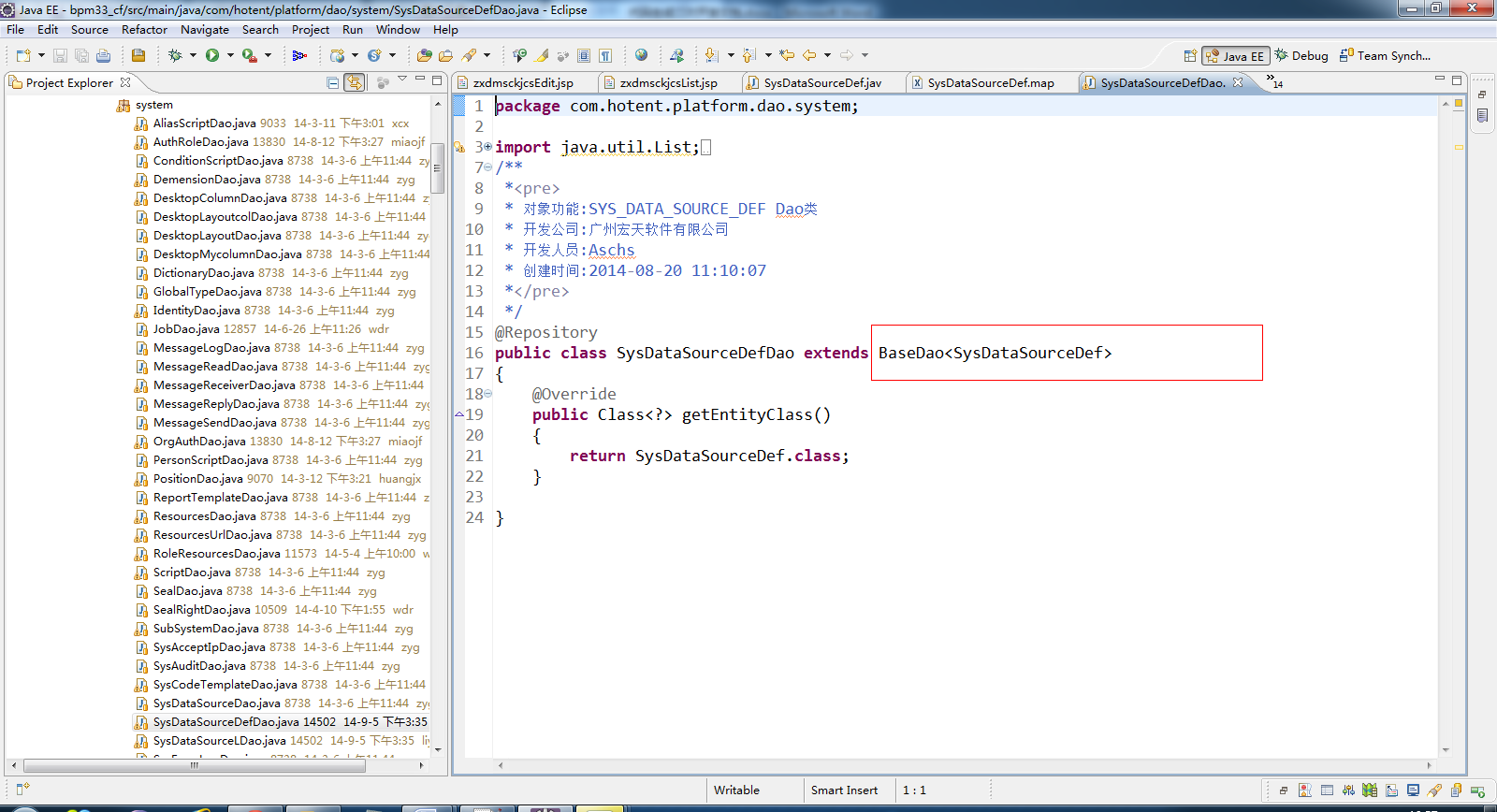
map:



2.Dao和service层

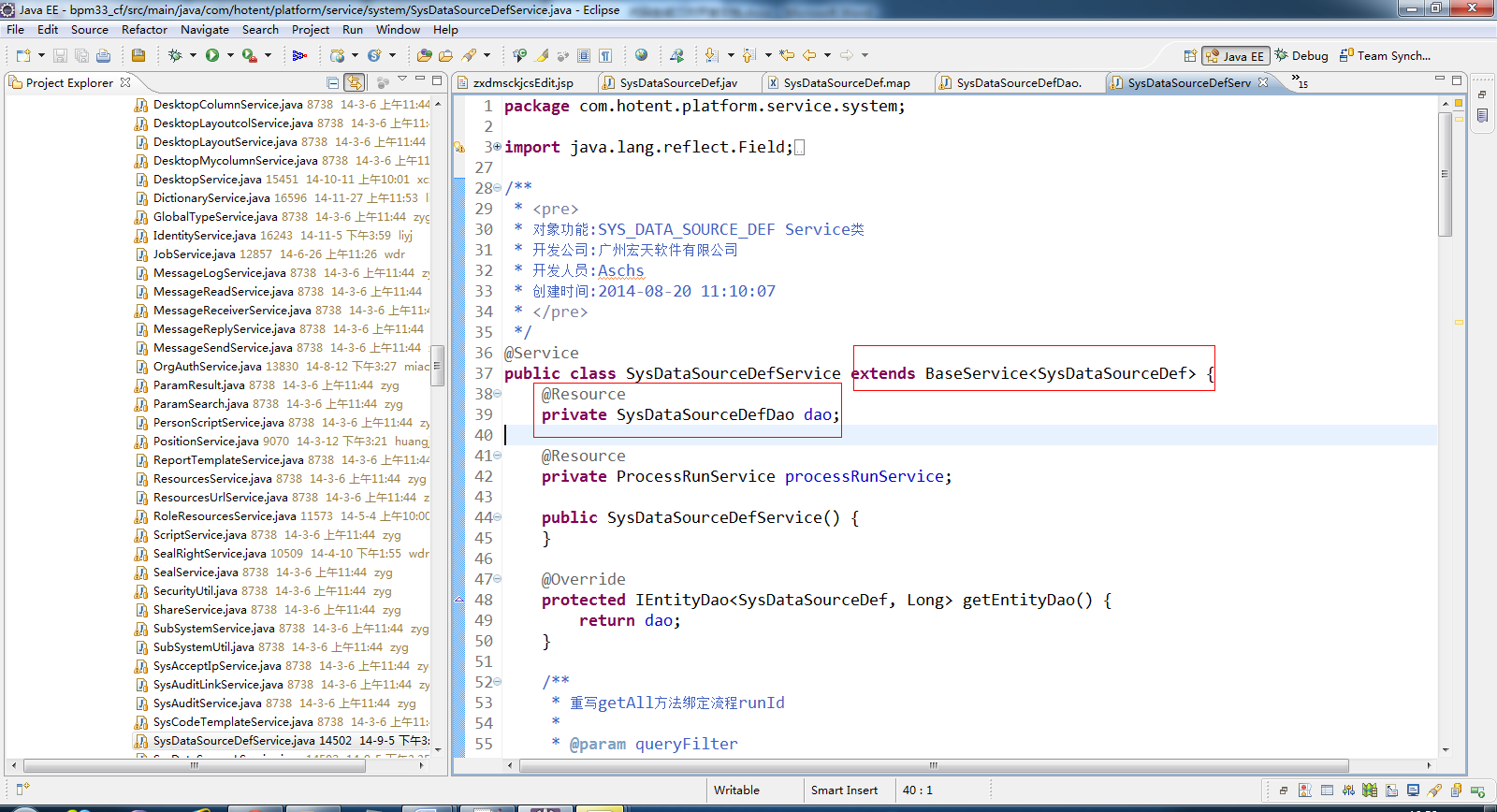
Dao:来用对数据库直接操作

eg:

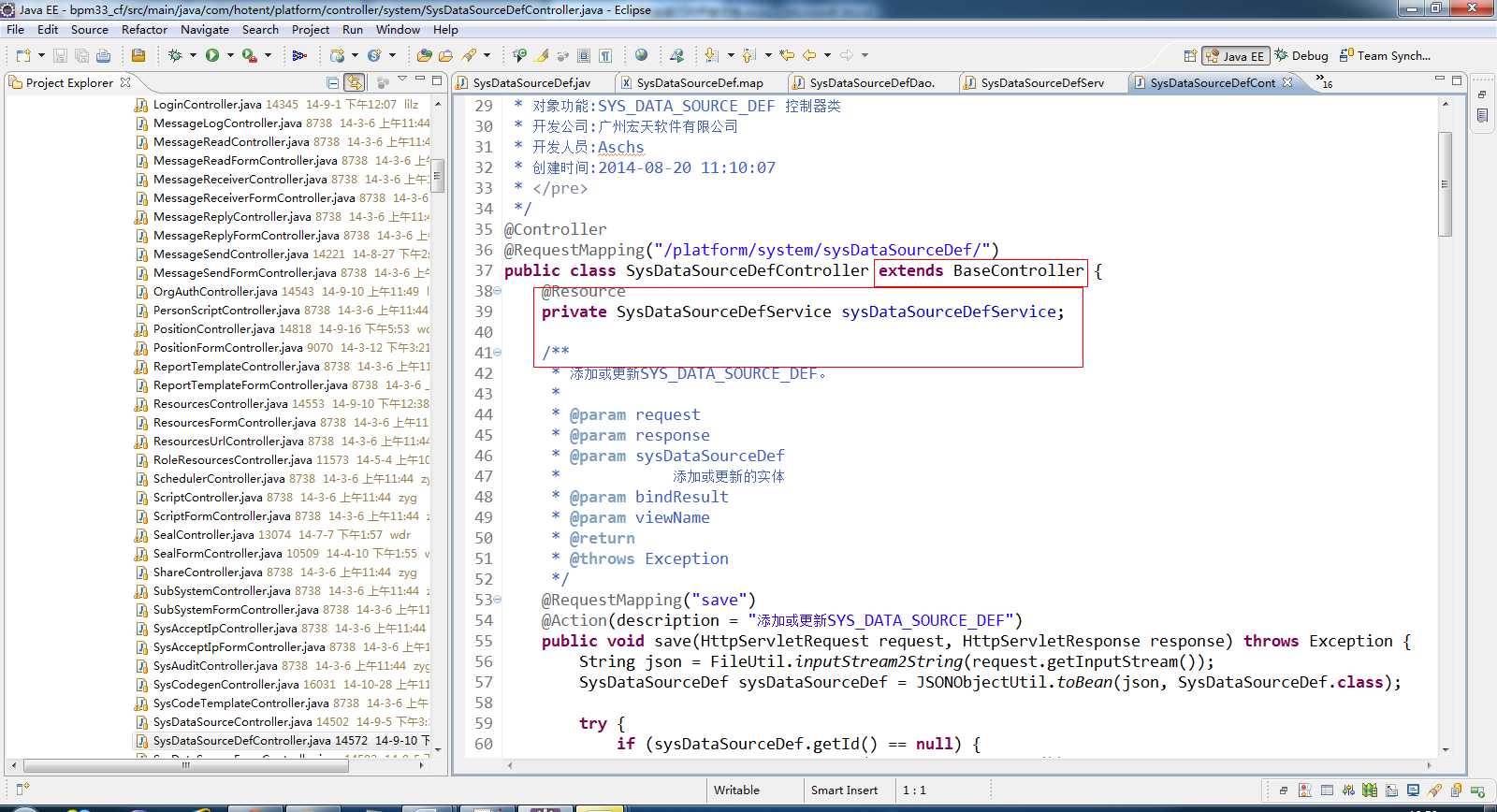


service:各种操作，通常配合dao层来进行更复杂的逻辑运算

eg:

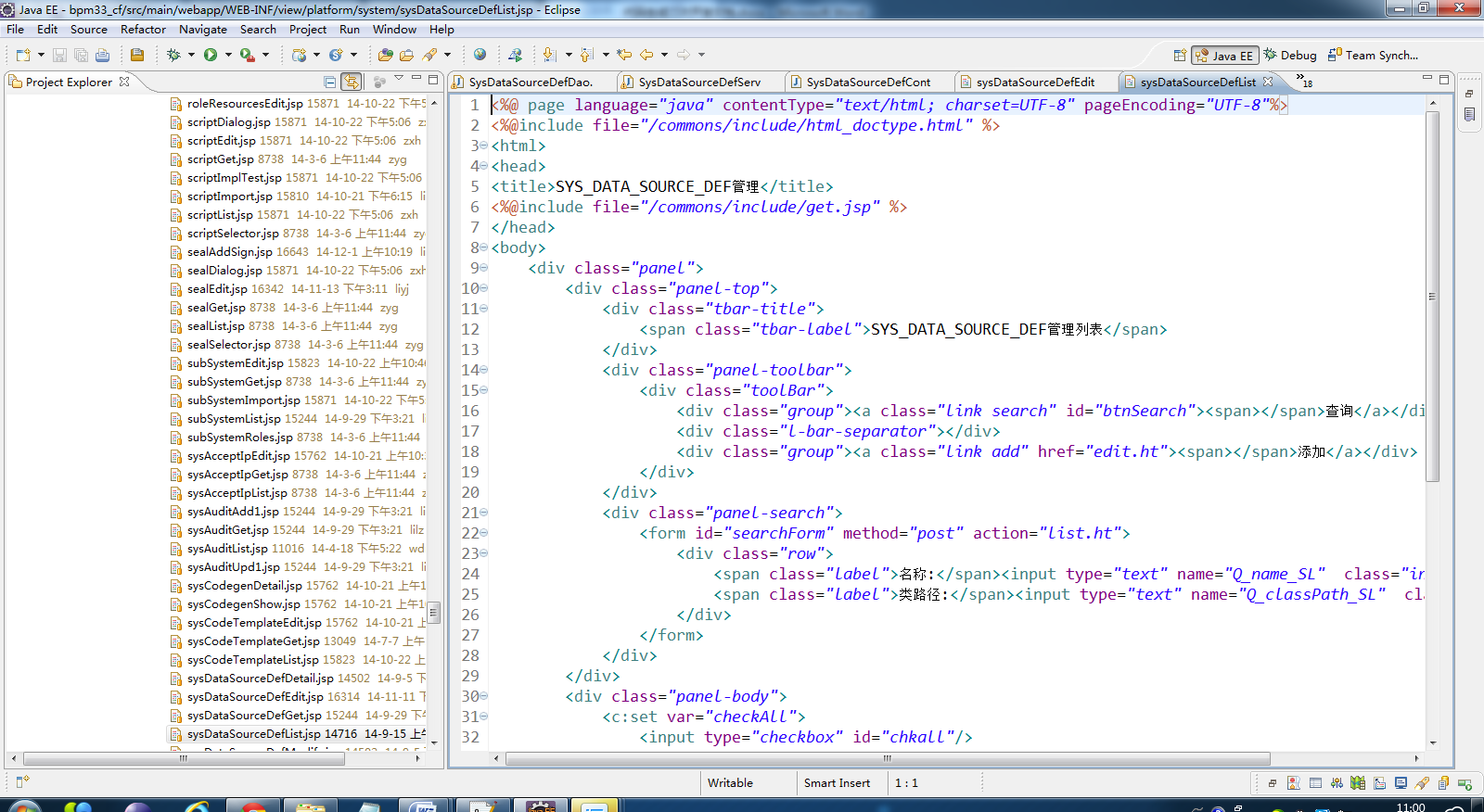


Controller：接收页面请求的地方，结合service来进行用户行为操作

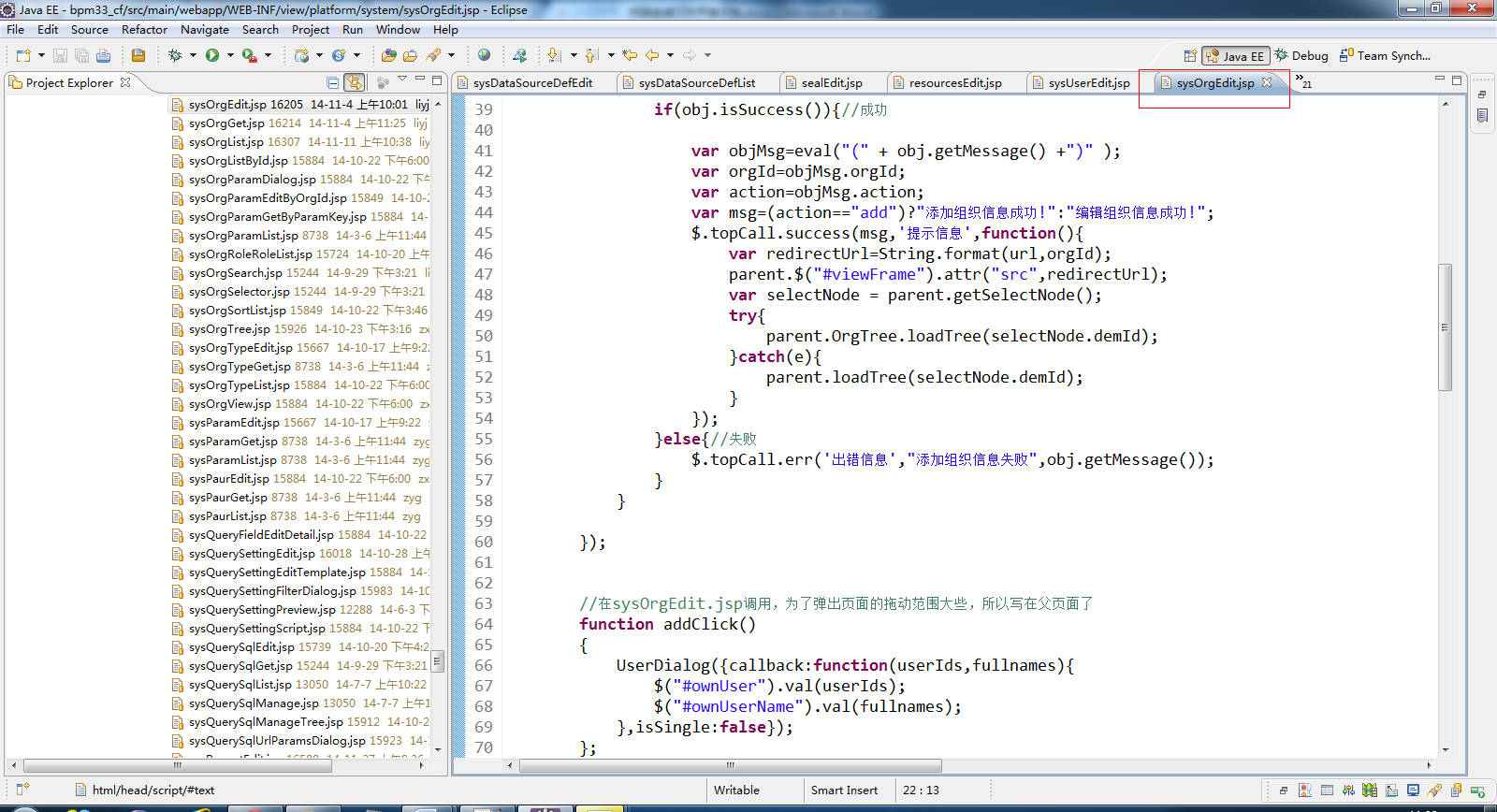


JSP页面：重要的事list edit页面，列表页和编辑页，可自己随便找一个系统本来自带的\*List.jsp和\*Edit.jsp来模仿

列表页eg:

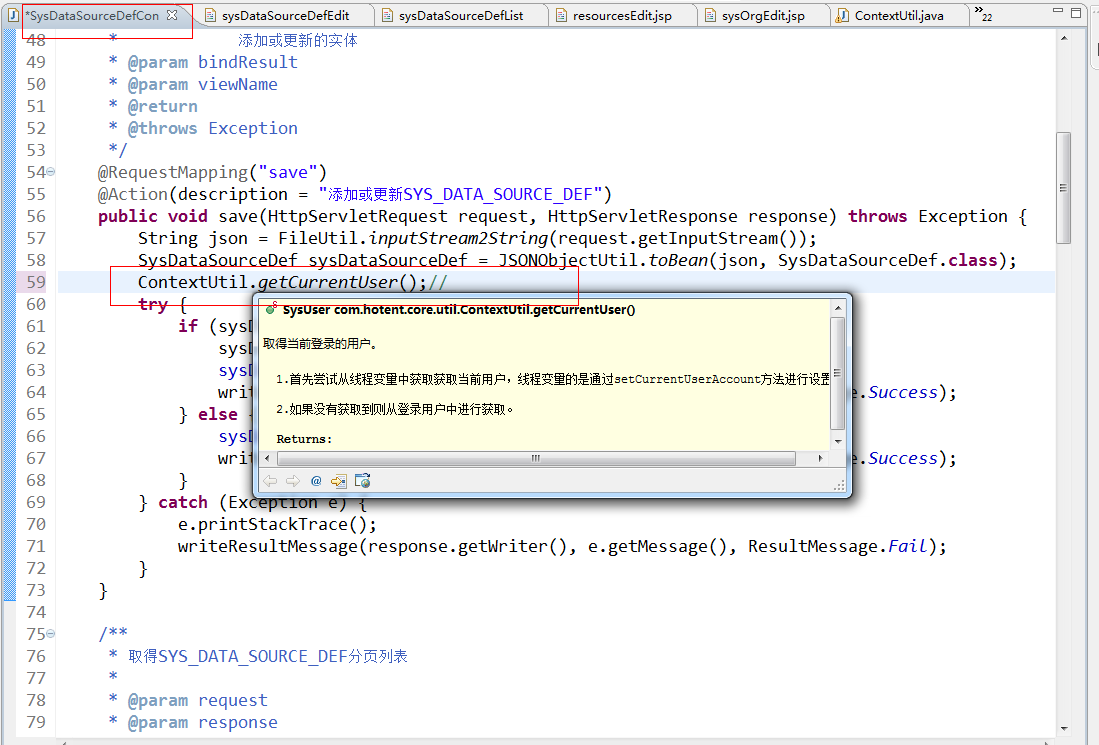


编辑页：因为sysDataSourceDefEdit.jsp页面有比较繁琐的js，这里拿了一个更简单的页面作例子



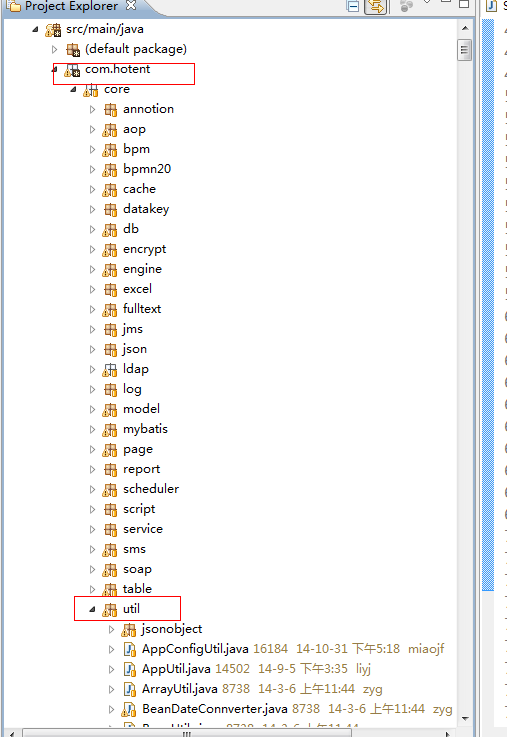
#### 常用代码的介绍

ContextUtil：通常我们需要在请求中获取当前使用者的信息等

使用例子：

里面还有很多各种各样获取当前用户的信息的方法，可自行上去查看

com.hotent.core.util里面还有很多工具包：



用户可根据命名来猜到大概用途，常用的有：StringUtil（字符串工具），JSONUtil（JSON格式化工具），TimeUtil（Date工具）