**为什么选用Handlebars**

目前Java开发,涉及到页面表示时，主要有两种解决办法。

1. **struts2/spring+vo+el表达式。**

**原理：**

VO：ViewObject，vo的属性可以是String、List、Map等等等，又可以vo套vo，因此这种方式非常灵活，也非常好用。

EL：el表达式，通常是${}、forEach标签什么的，即可构造出界面。

**缺点**：利用el表达式来生成HTML代码，这个动作时有服务器执行的，服务器解析完成后，才发送到客户端浏览器，会占用大量的服务器资源，而且响应速度缓慢。

**实例**：

1 <c:forEach var='bm' items="${contractAuditVo.borrowerModels}">

2 <table>

3 <tbody>

4 <tr>

5 <td>借款人编号：</td>

6 <td>

7 <p>

8 ${bm.borrowerId }

9 </p>

10 </td>

11 </tr>

12 <tr>

13 <td>客户编号：</td>

14 <td>

15 <p>

16 ${bm.customerId }

17 </p>

18 </td>

19 </tr>

20 <tr>

21 <td>曾用名：</td>

22 <td>

23 <p>

24 ${bm.usedName }

25 </p>

26 </td>

27 </tr>

28 </tbody>

29 </table>

30 </c:forEach>

1. **Json+ajax+拼html**

**原理：**这种方法通常是基于Ajax通信返回的Json字符串来构造HTML，因为服务器端只负责返回Json字符串，由客户端来通过JS脚本来解析，服务器负载小，响应速度快。

**缺点：**HTML是通过拼接方式来实现的，页面中出现了大量的html +=" html代码（div, td,tr･･････）"，使得页面层次结构非常零乱，不利于维护。

**实例**：

1 var contractTextHtml="";

2 $.each(jsonObject.cl,function(i,n){

3

4 contractTextHtml="";

5

6 //插入合同文本数据

7 contractTextHtml+="<div title='出借人信息---"+hiddenNull(n.cm.lenderName)+"' style='overflow:auto;padding:5px;'>";

8 contractTextHtml+="<table class='ui-table ui-table-noborder'>";

9 contractTextHtml+="<tbody>";

10 contractTextHtml+="<tr><td>合同编号：</td><td><p>"+hiddenNull(n.cm.contractId)+"</p></td></tr>";

11 contractTextHtml+="<tr class='ui-table-split'><td>出借人姓名：</td><td><p>"+hiddenNull(n.cm.lenderName)+"</p></td></tr>";

12 contractTextHtml+="<tr><td>出借人证件类型：</td><td><p>"+hiddenNull(n.cm.lenderIdType)+"</p></td></tr>";

13 contractTextHtml+="<tr class='ui-table-split'><td>出借人证件号：</td><td><p>"+hiddenNull(n.cm.lenderIdNum)+"</p></td></tr>";

14 contractTextHtml+="<tr><td>出借金额：</td><td><p>"+hiddenNull(n.cm.lenderAmount)+"</p></td></tr>";

15 contractTextHtml+="<tr class='ui-table-split'><td>出借人编号：</td><td><p>"+hiddenNull(n.cm.lenderNo)+"</p></td></tr>";

16 contractTextHtml+="<tr><td>出借人银行帐号：</td><td><p>"+hiddenNull(n.cm.lenderBankAccount)+"</p></td></tr>";

17 contractTextHtml+="<tr class='ui-table-split'><td>撮合编号：</td><td><p>"+hiddenNull(n.cm.makeMatchNo)+"</p></td></tr>";

18 contractTextHtml+="</tbody>";

19 contractTextHtml+="</table>";

20 contractTextHtml+="</div>";

21 $("#textList").append(contractTextHtml);

22 });

1. **第三种Handlebars**

el表达式构造HTML的优点是能够保持HTML的格式，JS构造HTML的优点是速度快省资源，那么二者合一，这就是Handlebars.js

Handelbars.js是一款基于Jquery的插件，它以Json为数据源，支持简单的逻辑判断，循环等操作，同时具有良好的扩展性，体积只有60KB左右，非常小