

基于 Struts2 Ajax 校验的资源管理系统 槽位端口添加方案

董卓亚

(商丘师范学院 计算机与信息技术学院, 河南 商丘 476000)

摘要:为了实现通信领域资源管理系统中设备槽位端口添加的条件限制,在系统中采用 Struts + Spring 框架、结合 Struts2 Ajax 校验,通过对校验条件和配置文件的合理配置,完全实现了添加属性的条件限制功能,满足了通信领域的实际使用需求。

关键词:资源管理系统;槽位;板卡;Struts Ajax 校验;模糊匹配

中图分类号:TP311 **文献标志码:**A **DOI:**10.3969/j.issn.2095-476X.2013.04.021

The scheme of adding slot ports to resource management system based on Struts2 Ajax validation

DONG Zhuo-ya

(College of Computer and Information Technology, Shangqiu Normal University, Shangqiu 476000, China)

Abstract: In order to achieve the constraints of equipment slot port addition in the communication resource management system, the Struts + Spring framework was adopted with Struts2 Ajax validation in this system, through the rational allocation of check conditions and configuration files. It fully realized the constraints function of addition attributes, thus met the actual needs of the communication.

Key words: resource management system; slot; board; Struts Ajax validation; fuzzy matching

0 引言

目前国内的通信资源管理系统对设备的管理是将其分为设备、槽位、端口3类分别进行管理,这是一种单一的数据管理模式。其功能主要集中在批量数据处理、单一数据查看、数据维护和权限控制上。文献[1]介绍的通信领域使用的设备,大多是机柜式设备:在一个机柜上,存在很多个槽位,槽位上安装有板卡,而通信用端口就存在于板卡上,这些端口通过数据线、光缆线等进行连接。设备、槽位、端口这些数据都应该是紧密相连的,所以实际上设

备、槽位、端口数据是一个多维的数据,应该建立一个多维数据库。文献[2]根据设备(即机柜)与槽位(板卡)和端口的关系,设计了一种将多种数据关联在一起的数据模型,便于数据的引用及数据的安全保护。这种将多种数据关联在一起的数据模型是未来数据库或者数据仓库必然的发展方向。文献[3]认为必须将几类数据共同管理,并且凸显出其之间的联系,将数据紧密结合。在实践中,发现通信领域设备基本是以机柜形式存在的大型组合设备。每个机柜上存在若干槽位,在槽位中安装需要的板卡,此槽位被使用。但是,机柜中槽位的放置并不是有

收稿日期:2013-02-05

基金项目:国家自然科学基金青年基金项目(61103195);国家博后基金项目(20100480048)

作者简介:董卓亚(1984—),女,河南省商丘市人,商丘师范学院助教,主要研究方向为计算机网络。

规律的,而是因设备散热、使用高度等原因,为了使用方便而无规律放置的.同时要对槽位进行编码,编码须具有唯一性、规律性.在资源管理系统中,需要对每一个设备进行数据的录入,包括设备的每一个槽位及每一个槽位上使用的板卡的具体数据,其中包括端口的属性^[4].在槽位添加需求上主要有以下问题需要解决:槽位的添加必须在机柜所含有的槽位数的范围内;使用槽位编号是按照工作人员实际使用的槽位号来编写的;同一槽位编号只能被使用1次.

在端口添加需求上实际需要解决的问题有:端口号符合槽位上板卡端口数量范围;端口号是真实使用的端口号;端口编号具有唯一性;端口只能与其他的1个端口连接.

为了解决上述问题,本文拟应用 Struts2 Ajax 校验技术,设计一种独特的关联式的添加方式,完成数据添加时的数据校验,以期解决这一系列问题.

1 方案设计

1.1 整体设计

在设备列表中添加槽位/端口管理按钮,点击进入管理页面.先进行槽位的添加,此时需要输入所使用的槽位号、端口数以及板卡类型,当点击增加按钮时,系统会通过 Struts2 Ajax 校验对输入的数据进行检验,检验是否符合上述要求,如符合则添加成功,否则报错并拒绝添加.添加成功后,将出现此槽位管理板块,可对此槽位进行管理.

在槽位管理板块进行端口添加,输入使用端口号、缆线编号、连接端口号、端口类型1、端口类型2,点击添加,同样 Struts2 Ajax 校验会对输入数据进行检验,以判断其是否符合前文提出的要求,如符合则进行添加.

Struts2 框架 Ajax Validate 就是在 Struts 框架的基础上,添加一个效验机制. Struts 框架提供了在访问请求发出后,当 Web 容器接收到这个请求后,会将其转交给一个过滤链过滤,然后就由 FilterDispatcher 核心控制器调用 ActionMapper 确定请求那个 Action. 核心控制器将控制权转交给 ActionProxy,由它来调用 Action. 在调用 Action 之前会依次调用配置文件中的拦截器,笔者就是在这些拦截器中增加了一个对提交内容进行效验的拦截器 jsonValidationWorkflowStack. 由它来对笔者提交的内容进行效

验,以判断其是否符合要求,如果不符合,将通过页面提醒用户,并阻止后续的动作调用等事件的发生.如图1所示.

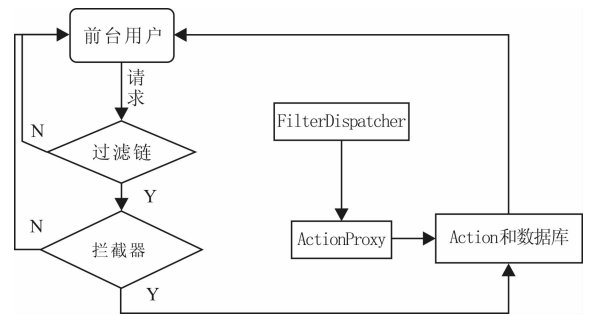


图1 Struts 2 框架 Ajax Validate 原理

1.2 Struts2 框架 Ajax Validate 配置

1.2.1 页面设置 1) 在页面表单 form 中,每一个 submit 都具有3次实际意义上的交互:第1次,就是 onclick,即为点击按钮瞬间发生的时间;click 时间结束后返回一个非 false,就会是第2次的 onsubmit 发生;当 onsubmit 返回一个非 false,表单 form 就会被提交,即为 submit 本身.所以在需要验证表单中的内容时,就需要把 Validate 放在 form 表单被提交之后,但是要在 submit 的 Action 发生之前^[5]. Submit 标签提供了一个属性就是 Validate = "true",同时对 Validate 进行属性 ValidateFunction = "kiwiValidation" 配置.

本文经过改进,在一个 JS 页面里面用 JS 语言写了一个 kiwiValidation 的方法,就是用来将所有的验证功能在 JSP 页面上显示出来.

```

function kiwiValidation( form, errors ) {
    if ( errors.fieldErrors ) {
        $.each( errors.fieldErrors,
            function( index, value ) {
                index = index.replace(/。/g, '_');
                var input_field_elem = $( "# " +
                    form_id + "_" + index );
                console.log( form_id + "_" + index );
                console.log( input_field_elem );
                if ( input_field_elem.length == 0 )
                    { alert( value[0] ); }
                input_field_elem.before( " <span class = 'emsg'
                    style = 'font - size:12px;
                    color: red; display: block' > " + value[0] + " </
                    span > " );
            }
        );
    }
}
  
```

这个 function 就是 submit 配置中配置的 " ki-wiValidation", 它的作用就是把后台的 Validate 配置文件和 Action 中 ValidateCreate () 产生作用后向 FieldError 传递警告.

2) 在 JSP 中加入如下代码

```
<%@ taglib prefix = "s" uri = "/struts - tags" %>
<%@ taglib prefix = "sx" uri = "/struts - dojo - tags" %>
```

1.2.2 后台配置 Struts2 Ajax Validate 的框架实现, 规范已经说得很清楚了, 但是并未给出一个有效的配置过程, 笔者给出其过程如下:

1) 要调用的 Action 需要继承 ActionSupport 类; 如 public class EXAction extends ActionSupport.

2) 对 Validate 的配置文件以及在 class 中的配制方法例如 ValidateCreate () 都需要遵守以下命名规则: 在对应的 class 中, 要调用 Action 所在 class, 需要配置一个 Validate, 名字为 Validate [Action] (), 其中 [Action] 为所调用 Action 的名字; 在对应 Action 所在的文件夹下, 即 EXAction 所在 class 所在的文件夹下, 需要建立一个 xml 文件, 其名字为 Class-Name - Action - validation. xml, ClassName 为 Action 所在 class 名字^[6].

3) 在 Struts2 的配置文件的 Action Map 中添加 < interceptor - ref name = " jsonValidationWorkflow-Stack"/ >, 并增加 input result. 如下:

```
< Action name = " EXAction" class = " org. apache. Struts2. showcase. validation. QuizAction" >
< interceptor - ref name = " jsonValidationWorkflow-Stack"/ >
< result name = " input" > quiz - ajax. jsp </result >
< result > quiz - success. jsp </result >
</action >
```

这样, Struts2 Ajax Validate 才算完全配置成功.

2 方案实现

2.1 槽位添加前台方法

在“添加槽位”页面中, “使用槽位号”即为员工真实使用的槽位. “端口数”即是每个槽位上板卡所拥有的端口数量, 为端口添加做了限制. 在页面中, 以标签形式出现的就是添加成功的槽位, 槽位的编码是后台自动生成的, 以机柜编码和槽位号相合生成.

2.2 槽位添加后台方法

每当点击“增加”按钮, 页面会向后台发出命令 create 一个新的 slot (槽位), 调用方法 “ create (Slot slot) ”. 在此方法中调用了另一个方法 genSlotCode (String devCode), 如下:

```
public String genSlotCode ( String devCode ) {
while ( true ) { slotCode = devCode + " - " + Slot.
getSlotNumber ( );
break; } return slotCode;
```

其目的是为此槽位生成一个唯一的槽位号, 此槽位号是后台自动生成的, 由机柜的设备号码与“使用槽位号”连接生成.

利用 Struts Ajax 验证之 Json Validate 技术. 在 Struts 框架中, 以 Json 形式传输的数据都可以使用这种验证^[7]. 在本资源管理系统中, 很多提交 form 表单的添删改功能都会用到 Validate 技术.

所以在 create () 方法所在的 Action 里面需要配置一个 validateCreate () 方法, 用于在 create () 方法运行前检验 create 的内容, 方法如下:

```
public void validateCreate ( ) {
if ( slotNumber > maxslotCount ) {
addFieldError ( " slotNumber", " 此设备只能添加"
+ maxslotCount + " 个槽位" );
} if ( null ! = slotService. getSlotBySlotCode ( slot-
Code ) ) {
addFieldError ( " slot. slotNumber", " 此槽位已
使用" );
}}
```

在上面的 ValidateCreate () 方法中, 首先获得前台传回的 form 表单中的 slot 属性 DevCode, 即此槽位所在设备编码, 然后通过 deviceService 中的 getDeviceByDevCode 方法获得 device (设备) 实例, 获得其中的属性 slotCount (槽位数), 与要添加的使用槽位编码对比, 以判断是否可以添加. 紧接着, 通过和 create () 调用相同的方法 genSlotCode () 获得槽位编码, 通过 getSlotBySlotCode () 在数据库中调取使用此槽位编码的槽位, 如果不为空, 说明已被使用, 不可添加此槽位. 这样就保证了槽位添加的唯一性、任意性, 同时符合范围要求, 如图 2 所示.

2.3 端口添加前台方法

页面中, 上部为添加端口的区域, 下部位端口列表. 在添加端口区域, 可以看到 2 个虚线的输入框, 在这里, 采用的是模糊匹配选择框. 当你输入

时,会根据你输入的内容自动在下拉框中显示模糊匹配的内容.

这里使用的是 Struts2-jquery 标签 autocompl-eter,它的作用就是自模糊匹配.在后台,通过 getby-

filter 方法获得 list,在 list 中,已经剔除了拥有端口连接的端口,显示的是尚无连接的端口,如果手动输入一个已连接或错误的端口,在提交时就会通过 validation 机制跳出警告.如图 3 所示.



图 2 添加槽位页面

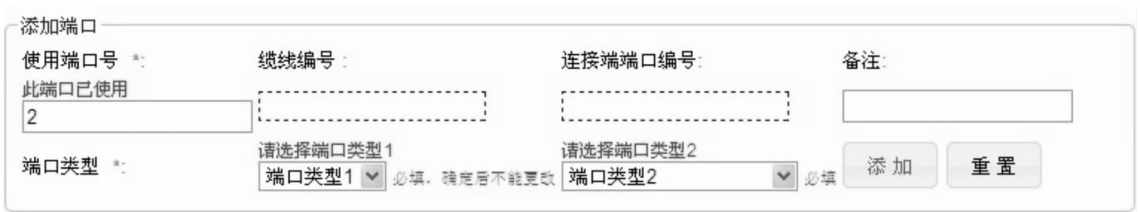


图 3 添加端口页面

添加的端口号是员工根据实际使用情况自助添加的.下部的端口列表有一句 if 语句,如下:

```
<s:if test = "ports. size < 1" >暂无端口,请添加
</s:if >
<s:else >当列表无内容时,显示暂无端口,请添加
```

2.4 端口添加后台方法

在端口添加必须输入的内容输入完全后,点击“添加”按钮,首先进行 Validate 检验,代码如下:

```
public void validateCreate() {
    if( exsistPortCount > = slotPortCount ) {
        addFieldError( " slotCount ", " 此槽位只能添加" + slot-
PortCount + " 个端口" );
    }
    if ( null ! = portService. getPortByPortCode ( port-
Code) ) {
        addFieldError( " port. portNumber ", " 此端口已
使用" ); }
    if ( ! port. getPortType1 CateCode ( ). equals ( PortE-
num. CUSTOMER. toCate ( ). getCode ( ) ) )
    if ( port. getPortCode ( ). equals ( port. getConnectTo
```

```
(( )) {
        addFieldError( " port. connectTo ", " 不
可以自连" );
    } } else {
        addFieldError( " port. connectTo ", " 非法的
端口编码" );
    } } } else {
```

根据其中中文,就可看出每一句代码所具有的检验功能,最重视的端口唯一连接性就是在这里进行功能实现的.分为 2 种情况:添加时连接一个已连接了的端口;自己连自己.在上述代码中倒数第 3 个 if 中可以看到“此端口已被占用”的字段,它和前面一个 if 一起解决了第 1 种情况,即连接到的是一个已被使用端口,调用了 portService 里面的 isUsedPort 这个方法,用来验证是否已使用:

```
public boolean isUsedPort( String portCode ) {
    Port p = getPortByPortCode( portCode );
    Port p2 = getPortByConnectTo( portCode );
    if ( p. getConnectTo ( ) == null && null == p2 ) {
        return false;
```

```

} else {
    return true;
}

```

紧接着第2种情况端口自己连接自己的检验开始了,会向前台输出一个"不可以自连"的中文提醒.与槽位添加类似,这里不再赘述.

3 Struts2 Ajax 校验的技术先进性

目前常用的2种校验方式中,“客户端纯 JavaScript 校验方式”在输入画面上根据 XML 文件中的配置内容生成对应的 JavaScript 校验脚本,在提交数据的时候首先使用这些生成的 JavaScript 校验输入信息.如输入错误,在画面上显示错误信息,不需要向服务器提交内容,只有输入数据校验正确时才向服务器提交请求.但是这种方式的校验不支持所有 Struts2 内置的校验,只支持一部分校验^[8].“普通的服务器端校验方式”是客户端提交信息到服务器,在服务器上使用 XML 配置信息,校验客户端提交的信息,如信息错误,返回 input 页面,显示校验出错误信息.此方式占用了大量的网络资源.

以往的校验方式主要分为服务器端校验和客户端校验.但是服务器端校验需要不停刷新页面,占用大量网络资源,客户端校验不能完全完成 Struts 校验方法.现在 Struts 框架为我们提供了一个在 Action 执行前就进行验证的机制,即 Struts2 Ajax 校验方式.其主要特点是在提交请求的时候,首先使用 AJAX 的方式将数据提交到服务器,在校验出错的情况下在页面显示错误信息但不刷新页面,在校验成功的情况下才向服务器提交请求^[9].

由于客户端纯 JavaScript 校验方式不刷新页面,但是也不能使用所有的 Struts2 校验方法;服务端校验可以使用所有 Struts2 的校验方法,但是需要占用较多资源.而本文部署利用的 Struts2 Ajax 校验,既不需要刷新页面占用资源,又可以使用 Struts2 内置的所有的校验方法,所以 Ajax 校验具两者之长,是校验首选.Struts2 Ajax 校验的配置很简单,而且校验内容也可以写得很丰富.在类中,可以根据要求写出自定义校验方法,这是其他校验方法不可比拟的.在本管理系统中,利用 Struts2 Ajax 校验完成了设备到槽位、槽位到端口的层层相连,使得数据具有关联性,使得设备管理具有多层次、智能化、高深度的优点,完善了当前通信领域资源管理系统的功能模块.使得资源管理系统在数据整合方面具有了

新的发展方向.

4 结语

本文通过使用 Struts2 Ajax 校验技术,成功地完成了通信领域设备资源管理中遇到的问题.当设备添加校验通过后,才可进行槽位添加;槽位添加校验完成后,才可进行端口添加,同时对端口的添加数据进行校验,有效地将实际中设备、槽位、端口间的数据关系和关联体实现在软件中,是软件设计中的一次成功设计.同时,模糊匹配技术也使操作者的劳动大大减少.利用添加校验实现端口对端口的链接,完成了设备与设备之间的关联,将所有设备组成了一个大的联通的通信网络,便于数据关联,为下一步进行基于物理层的自动拓扑图生成功能完成数据库设计.

Struts2 Ajax 校验是目前比较先进的校验技术,既具有客户端 JavaScript 校验的不占用网络流量的优势,又能进行所有 Struts 校验.但是目前的 Struts2 Ajax 校验还存在很多问题,在配置文件中,命名过于死板,且每一个 Struts.xml 中都需要配置校验返回语句,对于 JAVA 这种可移植性很强的语言来讲,目前的 Struts2 Ajax 校验方法仍具有过高的耦合度,尚有待改进.

参考文献:

- [1] 袁建巧.基于.NET的移动通信基站设备管理系统的设计与开发[D].成都:电子科技大学,2011.
- [2] 杨志宇.信息通信设备管理系统研究与设计[D].保定:华北电力大学,2012.
- [3] 熊勇.重庆电信数据通信网络的系统安全研究[D].重庆:重庆大学,2004.
- [4] 赵得成,柴英杰.标准通信机柜的结构创新与造型设计[J].机械设计与制造,2006(10):135.
- [5] 李超.ASP 表单两种提交模式辨析[J].内江科技,2008(2):149.
- [6] 邓芳.基于MVC模式的Struts框架中Ajax研究与应用[J].湖北教育学院学报,2007(24):8.
- [7] 屈展,李婵.JSON在Ajax数据交换中的应用研究[J].西安石油大学学报,2011,26(1):95.
- [8] 齐新力,郭亚莎,邹振宇.基于Struts2框架数据校验的分析与应用[J].仪器仪表用户,2011,18(5):48.
- [9] 蒲子明,许勇,王黎.Struct + Hibernate + Spring 整合开发技术详解[M].北京:清华大学出版社,2010.