

基于JavaScript和Java语言的动态网页设计方法

雷 虹

摘 要 针对网页制作过程中单独使用Java或JavaScript语言出现的问题进行了分析,研究并提出将二者结合使用,既可以减少程序处理流程,为网上客户提供方便的解决方案。

关键词 Java JavaScript 服务程序 事件处理

Design Method of Web Page Based JavaScript and Java

Lei Hong

(Computer Center of Shenyang Railway Administration, Shenyang, 110001)

Abstract: The writer discussed the problems while using Java, JavaScript independently for Web page. As a solution, the author combines JavaScript with Java to reduce the perform flows and provides a convenience options to Web users.

Keywords: Java, JavaScript, small service program, event handling

1 问题的提出

我们在应用Java语言制作互联网上售票系统过程中,庞大的数据库中包含了全国几千个车站的站名信息。考虑到用户对各车站的查询频率、网页的响应速度、时间及资源占用等因素,可供用户直接选取的站名列表中只包含了全国特、一、二等站的站名字典,其他多数站名无法全部列出。当用户需要查询不在列表中的站名时,需要用户手工输入。如果使用JavaScript语言制作该页面,则无法动态从数据库中提取站名列表,给用户带来不便,也造成频率资源的浪费。因此,单独使用上述两种语言制作Web页面,均无法兼顾查询效率和使用方便。

2 JavaScript和Java语言的特点

Java是一种使建立分布式应用程序(即由网络上的多个计算机执行的程序)更简单、易行的技术。由于其强大的移植能力、多线程处理和连网能力而充满魅力,成为网络编程的最优秀工具之一。它使Internet的作用从通信工具扩展到能够运行成熟应用程序的网络。这种突破性技术使企业可以在Intranet上部署全方位的业务服务和实时交互式信息内容。Java程序有三种类型:小程序、小程序和服务程序。小程序就是在Web服务器上运行的特殊程序,它的主要功能是响应服务器上的新连接,这个连接导致调用小程序的service()。

JavaScript是与Java有松散联系的另一编程语言,可以在HTML文档中直接编码,支持Web应用程序的客户和服务方组件的开发。在客户机中,它可用于编写Web

浏览器在Web页面上下文中执行的程序;在服务器中,它可用于编写用于处理Web浏览器提交的信息并相应地更新浏览器显示的Web服务器程序。JavaScript得到了Netscape Navigator、Microsoft Internet Explorer和Opera Browser等浏览器的支持,是WWW上新的强大的编程语言。它不仅用于开发真正交互式的Web页面,而且像胶水一样将Java小程序、ActiveX控件、浏览器插入件、服务器脚本和其他Web对象集成起来,使开发人员能够生成Internet和公司Intranet上使用的分布式应用程序。

尽管Java与JavaScript名称相似,但这两种语言却有许多显著的差别,这种差别并不是一种语言优于另一种语言,两者各有所长。Java用于实现Web嵌入页面的可执行内容,它具有更强的编程功能,但这些功能限制在有限的浏览器窗口区中。JavaScript用于HTML环境,它提供了能够响应Web页面时间的脚本。这样,它可以完全访问浏览器窗口的各个方面。

3 JavaScript的事件控制的解决方案

为解决动态网页设计嵌套事件控制机制问题,在Java小程序中嵌入了事件控制的JavaScript。事件是浏览器响应用户操作的机制,JavaScript的事件处理功能可改变浏览器响应这些操作的标准方式。这样,一方面可以实现与数据库交互产生动态网页,另一方面可实现用户与Web页面的交互。

对于在字典中存在(即使用频率较高)的站名,用户可直接在列表中选择;对于字典中不存在(即不常用)的站名,用户可直接输入。



首先, 要使用小服务程序从数据库中读出站名列表:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import ConnPool.*;

public class query_stop_time_from_to extends
javax.servlet.http.HttpServlet {
public void init() {
    /*建立数据库连接*/
    Connection conn = ConnPool.getConn_t0();
}

public void service(
javax.scrvlet.ServletRequest req,javax.scrvlet.
ServletResponse res)
    throws javax.scrvlet.ServletException,
java.io.IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    out.println("<html>");
    out.println("<head>");
    . . . . .

    /*建立站名输入表单*/
    out.println("<form name=form1>");
    out.println("<select name=\"from_station\"
size=\"5\" onClick=\"MM_goToURL_from()\">");
    /*查询站名字典*/
    try{
        String str = "SELECT station_name FROM
station_dictionary";
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(str);
        While(rs.next()){
            out.println("<OPTION value=\"\" +
rs.getString(1) + \"\">\" + rs.getString(1));
        }
        stmt.close();
    }catch(Exception e){}
    . . . . .

    /*发站站名输入文本域*/
    out.println("<FORM NAME=\"form3\"
METHOD=\"POST\"");
    out.println("<tr>");
    out.println("<td width=30 nowrap
```

```
align=\"center\" valign=\"bottom\">发站</td>");
    out.println("<td width=106
align=\"center\" valign=\"top\">");
    out.println("<input name=from_station_name
size=13 maxlength=\"13\" class=\"text\">");
    . . . . .
```

在建立上面的站名列表时, 我们捕捉了点击事件 onClick:

```
<select name="from_station" size="5"
onClick="MM_goToURL_from()">
```

当用户点击站名列表中的某项之后, 执行 JavaScript 函数 MM_goToURL_from(), 将用户选择的站名赋给发站站名文本域:

```
out.println("<script
language=\"JavaScript\">");
out.println("<!--");
/*将用户从发站列表中选择站名赋给发站字段*/
out.println("function MM_goToURL_from() { //
v3.0");
    out.println("    from_station_idx =
document.form1.from_station.options.selectedIndex;
");
    out.println("    from_station_str =
document.form1.from_station.options
[from_station_idx].value;");
    out.println("    document.form3.from_station_name.value = from_station_str;");
    out.println("    return false;");
    out.println("    ");
    到站站名的选择方式与发站相同。
```

4 结束语

JavaScript 和 Java 两种脚本语言相结合, 既可以减少程序处理流程, 也为网上客户的输入提供了极大的方便。

JavaScript 和 Java 都是功能强大的编程语言, 将它们有机地结合起来, 能够建立更优秀的动态交互网页。既能在动态网页开发编程方面提供方便, 又能替搞网上访问网页的效率。

5 参考文献

- 1 翟裕忠. 电子商务应用开发技术. 北京: 高等教育出版社, 2000

(收稿日期: 2001-01-15)