

基于 JSP 与 JavaBean 技术的 Web 应用开发

赵晓峰

(无锡商业职业技术学院信息工程系, 江苏 无锡 214063)

摘要: 随着 Web 技术的迅速发展, 计算技术也由早期基于 C/S 的模式向基于 B/S 的模式进行转变。因此, 基于 JSP 技术, 本文探讨了 B/S 模式下 Web 应用的设计与实现, 包括 JSP 动态网页技术、JSP 与 JavaBean 的结合和 JSP 与 JavaBean 的应用三个部分。

关键词: JSP 技术; JavaBean 技术; 数据库

中图分类号: TP3

文献标识码: A

文章编号: 1672-6332 (2005) 02-0039-05

随着 Web 技术的迅速发展, 动态网页技术日趋成熟, 大多数的应用由 C/S 结构向 B/S 结构过渡。目前的 Web 应用较多是使用三层结构, 通过 HTML 中嵌入 ASP、PHP 等脚本程序形成动态网页, 实现客户端(浏览器)与服务器的交互, Web 服务器充当中间层完成业务逻辑计算。这种结构存在着三个方面的不足: 第一, 包含大量的业务逻辑计算脚本程序, 执行效率不高; 第二, 脚本代码被嵌入到 HTML 中, 开发效率不高且不利于维护; 第三, 代码可移植性差且很少能够复用。

JSP (Java Server Page) 技术由 SUN 公司推出, 使用 Java 作为脚本语言。它不仅克服了早期 CGI 技术维护复杂、运行效率低下等缺点, 而且相对 ASP 等其它动态网页技术, JSP 还支持跨平台开发与运行、组件开发与使用方便的优点, 提高了 WEB 应用的开发与执行效率。

JavaBean 是一种基于 Java 的软件组件。JSP 对于在 Web 应用中集成 JavaBean 组件提供了良好的支持。这种支持主要体现在代码复用, 缩短了开发时间, 同时给 Web 应用带来了一定的可伸缩性。JavaBean 组件可以用来执行复杂的计算任务, 负责与数据库的交互, 实现具体业务逻辑的封装等。

1 JSP 动态网页技术

1.1 JSP 运行方式

[收稿日期] 2005-03-21

[作者简介] 赵晓峰 (1977-), 男 (回族), 江苏常州人, 高级工程师, E-mail: xfzhao_xd@163.com

JSP 是 SUN 公司推出的一种动态网页技术标准。它在传统的静态页面文件中加入 Java 程序代码和 JSP 标记, 构成了 JSP 页面。JSP 页面的执行过程是服务器在接收到客户端发来的一个 JSP 页面请求时, 首先 JSP 引擎把 JSP 代码转换成 Servlet 代码 (其实是 Java 源码), 然后由 JSP 引擎调用服务器端的 Java 编译器对 Servlet 代码进行编译, 生成字节码文件, 再由 JVM 执行此字节码文件, 最后将执行结果以 HTML 格式返回客户端 (浏览器)。过程如图 1 所示。

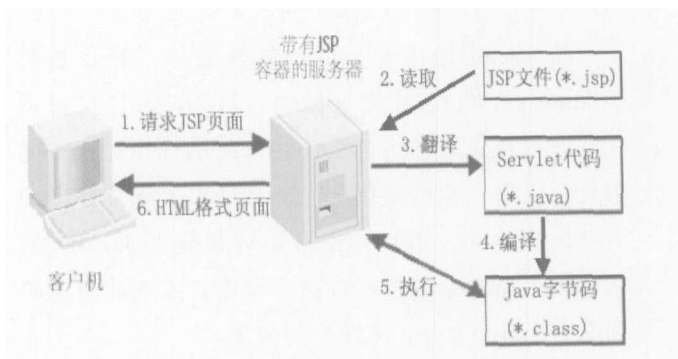


图 1 JSP 执行过程

Fig.1 procedure of running JSP

实际上, 这个过程只有第一次运行 JSP 时才执行, 这也是第一次浏览 JSP 页面速度比较慢的原因, 而生成了字节码文件后, 下一次请求就由 JVM 直接调用此字节码文件执行, 将执行结果以 HTML

格式返回。当 JSP 代码有所修改,则服务器就要求 JSP 代码重新转换成 Servlet 代码,再进行编译和执行。

1.2 运行环境及服务器配置

目前可支持 JSP 的应用服务器产品较多,其中 SUN 公司推荐使用的 Tomcat 应用服务器占有较大的市场份额。本文以 Tomcat 为例说明 JSP 运行环境的建立过程。

1) JSP 运行环境的建立,首先需要安装 J2SDK,目前最新的 Java 2 平台标准版开发工具包 5.0 可以在 SUN 公司的网站上直接下载。

2) J2SDK 安装结束后,再安装 Tomcat 应用服务器,默认 Web 服务端口是 8080 端口,用户也可改变该端口设置,但不能与其它应用程序冲突。

3) 最后,在操作系统的环境变量中设置 CLASSPATH 变量,并在 PATH 变量中添加有关的变量。

要想让 JSP 程序正常运行,还需要进行一定的配置,如建立应用子目录及其相关配置文件,类似于 IIS 中建立网站虚拟目录。如果使用 NetBeans IDE 开发环境来创建 Web 应用,就不需要再进行配置了,SUN 已经将该开发环境放到 J2SDK 5.0 的软件包中,可随 J2SDK 5.0 一起下载安装。

2 JSP 与 JavaBean 的结合

JavaBean 是基于 Java 的软件组件,有点类似于 Microsoft 的 COM 组件概念。JavaBean 支持可视化和非可视化两种组件,可视化的组件在运行中能够显示出来,非可视化组件通常用来处理程序中的一些复杂事务。JSP 只支持非可视化的 JavaBean,常用来封装业务逻辑和数据库操作,实现业务逻辑和前台程序(如 jsp 文件)的分离。

在 JSP 中使用 JavaBean 需要用到 JSP 标记的 useBean 动作,该语法格式如下:

```
<jsp: useBean id="BeanName"
scope="page|request|session|application"
class="package.class"/>
```

在当前 JSP 页面中使用<jsp: useBean>动作相当于创建了类 package.class 的一个名为 BeanName 的对象。实际上,在 JSP 页面中可以同 Java 一样,

使用 new 关键字来创建 JavaBean 对象,只要在页面开始处用 page 指令引入(import)该 package。

<jsp: useBean>动作标记内定义了几个属性,其中 id 定义了整个 JSP 页面内该 JavaBean 的标识,在 JSP 页面中可以通过 BeanName 引用该对象;scope 属性定义了该 JavaBean 的生存时间;class 属性说明了该 JavaBean 的类文件。

创建 JavaBean 对象后,在 JSP 页面中还可通过 jsp: setProperty 动作设置 JavaBean 对象的值,也可以通过 jsp: getProperty 动作读取 JavaBean 对象的值。

编写 JavaBean 与编写普通的 java 程序一样,可使用普通的文本编辑器或者像 NetBeans IDE 这样的集成开发环境。此外,JSP 页面中使用的 JavaBean,必须放在一个 package 中,否则 JSP 无法访问到该 JavaBean。在 Tomcat 环境下,经过编译的 JavaBean 字节码文件随 package 一起要存放到该 Web 应用目录的 WEB-INF\classes 目录中。

3 JSP 与 JavaBean 的应用

在此,我们应用 JSP 技术为某文体公司开发了一个在线购物系统,用户可以在网上实现浏览、检索、在线购买喜欢的文化、体育用品。该系统采用 Tomcat 为应用服务器,使用 MS SqlServer 7.0 为数据库服务器。应用程序通过 JDBC 与数据库连接。

该系统包括网上交易和后台管理两大部分。其中网上交易主要针对客户,用户登录、浏览选择商品、提交订单、以及留言、投诉等配套服务。后台管理则由系统管理员完成,包括商品信息的添加、修改和删除,顾客购物信息分类统计,新产品推荐,订单处理等。这个在线网上购物系统的建立,技术上要解决后台大型数据库的连接和数据库数据的存取,用户购物信息的处理等问题。

3.1 使用 JavaBean 访问数据库

利用 JSP 开发 Web 应用时,对数据库的访问是必不可少的,如何获取数据、增加数据、删除数据,以及如何对数据库进行管理,是每个 Web 应用必须要解决的问题。在 Java 中连接数据库的技术是 JDBC (Java Database Connectivity)。很多数据库系统带有 JDBC 驱动程序,Java 程序就通过 JDBC 驱动程序与数据库相连,执行查询、提取数

据等操作。在 JavaBean 访问数据库的整个过程与在 Java 中是一致的, 使用 JDBC API 来访问数据库, JDBC API 由 Sun 和其他数据库厂商共同建立, 并已得到广泛的应用。

JDBC API 由多个类和接口组成, 进行数据库编程时, 有 4 个常用接口: DriverManager、Connection、Statement、ResultSet。(1) DriverManager 用于管理 JDBC 驱动程序的接口, 主要通过 getConnection 方法取得 Connection 对象引用。(2) Connection 对象是通过 DriverManager.getConnection() 方法取得的, 表示驱动程序提供的与数据库连接的一次对话, 使用 createStatement() 返回一个 Statement 对象。(3) Statement 是向数据库提交 SQL 语句并返回相应结果的接口, 如是 SELECT 语句, 则使用 executeQuery (String sql) 返回 ResultSet 对象; 如是 UPDATE、DELETE、INSERT 语句, 则使用 executeUpdate (String sql) 方法返回发生改变的记录条数; (4) ResultSet 接口定义访问由于执行 Statement 产生结果集的方法, ResultSet 结果集可以按照名称或列号访问。下面是通过 JDBC 访问数据库的 JavaBean 的源代码。

```
package connect;
import java.sql.*;
public class DBConn {
    String DBDriver =" sun.jdbc.odbc.JdbcOdbc -
Driver"; //注册 JDBC-ODBC 驱动
    String ConnStr="jdbc: odbc: csdb"; //数据库
    连接
    Connection connect=null;
    ResultSet rs=null;
    public DBConn () {
        try {Class.forName (DBDriver); //注册数据库
        驱动
        } catch ( java.lang.ClassNotFoundException e)
        {
            System.err.println (e.getMessage ());
        }
        public ResultSet executeQuery (String sql) {
            //该方法用于数据查询操作
            try { connect =DriverManager.getConnection
```

```
(ConnStr); //获取对应数据库的连接
            Statement st =connect.createStatement
            (); //返回一个 Statement 对象
            rs=st.executeQuery (sql);
            } catch (SQLException e) {
                System.err.println ( e.getMessage
                ());
            }
            return rs; //返回结果集
        }
        public int executeUpdate (String sql) { //该
        方法用于数据更新操作
            int result=0;
            try { connect =DriverManager.getConnection
            (ConnStr);
                Statement st =connect.createStatement
                ();
                result=st.executeUpdate (sql); //执行
                更新操作
            } catch (SQLException e) {
                System.err.println (e.getMessage ());
            }
            return result; //返回被更新记录的条数
        }
    }
}
```

该 JavaBean 经过编译生成字节文件后, 放到应用的 WEB-INF/class 目录中。需要访问数据库的 JSP 页面中使用 JSP 的 useBean 标记进行引用, 产生数据库连接的对象。具体引用如下: <jsp: useBean id="conn" scope="page" class="connect. DBConn" />。而在 JSP 文件中执行 SQL 语句时, 可直接通过该方法访问, 如 ResultSet rs=conn.executeQuery ("select * from lqtable")。

3.2 JavaBean 封装业务逻辑

在线购物网站中, 需要用到购物车保存用户的购物信息, 并且在用户购物过程中要动态记录购物信息, 结束时要能进行结算。如果用数据库来跟踪购物信息, 系统开销较大, 因为每个客户购物时都要使用购物车。利用 JavaBean 来构建这样一辆“购物车”, 可以将业务逻辑进行封装, 购物有关的页面引用该 JavaBean 即可。

把购物车看作一个对象,已购商品的信息是该对象的属性,购物过程的一些操作构成它的方法。如要实现购物车中添加、删除一件商品这样的功能,就要建立一个 addGoods 成员方法和一个 removeGoods 成员方法。在购物过程中,根据 Jsp 页面文件传回操作的类型,直接调用相关的方法来实现。

当然,也可以把这些处理操作完全写在 jsp 程序中,不过这样会增加 JSP 页面中的代码,造成 jsp 页面臃肿不堪,增加了应用维护的复杂度,不利于 Web 应用的开发与维护。以下给出实现购物车 JavaBean 基本代码,可根据页面传回的操作类型完成购物。

```
package shop;
import java.util.Vector;
import java.util.Enumuration;
public class ShoppingCar {
    Vector v=new Vector ();
    String submit=null;
    String item=null;
    private void addGoods (String name)  { //添加商品
        v.addElement (name);}
    private void removeGoods (String name)  { //减少商品
        v.removeElement (name);}
    public void setGoods (String name)  { //设定本次操作的商品
        item=name;}
    public void setSubmit (String s)  {
        submit=s; }
    public void procRequest ()  { //由 JSP 页面传
```

回的参数选择执行的操作

```
    if (submit.equals ("add"))
        addGoods (item);
    else if (submit.equals ("remove"))
        removeGoods (item); }
}
```

JSP 页面引用: <jsp: useBean id="car" scope="session" class =" shop. ShoppingCar " />, 该 JavaBean 的生存时间是一次会话 (session)。通过 <jsp: setProperty name="car" property="item" value="<% request.getParameter ("item") %>" /> 得到从 JSP 页面传递过来的商品名称, <jsp: setProperty name =" car" property =" submit" value =" <% request.getParameter ("submit") %>" /> 得到从页面传递的操作类型; 最后调用购物车对象的 <% car. procRequest (); %> 进行业务的处理。

这个 JavaBean 只记录了所购商品的名称, 如需要考虑添加商品的时候判断库存是否有货物, 没有货物不得购买 (需要查询数据库), 在这个时候我们可以直接修改 JavaBean 的 addItem 方法, 加入处理语句来实现, 这样就完全不用修改动态网页中的 jsp 脚本程序了。当然, 实际应用过程中, 还要根据具体网站的要求, 进一步完善购物车的功能。

4 结语

从上述实例中不难看出, 在 JSP 中使用 JavaBean, 不仅起到了封装信息的作用, 还可以将数据库查询等一些可重用的业务逻辑运算程序隐藏在 JavaBean 的内部。在 Web 应用开发过程中, 为具体的 Web 应用设计出符合业务逻辑要求的 JavaBean, 可提高程序的可重用性, 更有利于页面与业务逻辑的分离, 提高开发效率。

参考文献 (References)

- [1] Gary B, Bharathi N. JSP 程序设计指南[M]. 天宏工作室. 北京: 清华大学出版社, 2002.
Gary B, Bharathi N. Guide of JSP designs program[M]. Tian hong room. Beijing: Tsing-Hua University Press, 2002. (in Chinese)
- [2] 林上杰, 林康司. JSP 2.0 技术手册[M]. 北京: 电子工业出版社, 2004.
LIN Shangjie, LIN Kangsi. JSP 2.0 technical manuals[M]. Beijing: Electronic Industry Press, 2004. (in Chinese)
- [3] 朱敏. JSP WEB 应用教程[M]. 北京: 北京交通大学出版社, 2004.
ZHU Min. WEB application of JSP. Beijing: Beijing Transportation University Press, 2004. (in Chinese)
- [4] 管贻生. Java 高级实用编程[M]. 北京: 清华大学出版社, 2004.
GUAN Yisheng. Advanced practical programming of JAVA[M]. Beijing: Tsing-Hua University Press, 2004. (in Chinese)

Web application development based on JSP and JavaBean

ZHAO Xiaofeng

(Department of Information Engineering, Wuxi Institute of Commerce, Wuxi 214063, Jiangsu, P. R. China)

Abstract: With the development of web technology, the technique of computation transfer from C/S model to B/S. Based on JSP The paper discusses the design and realization of web application under on B/S model, including JSP dynamic web page technology, the combination and application of JSP and JavaBean.

Keywords: JSP; JavaBean; Database

(责任编辑: 贺小凤)