

# 基于 Java Servlet 的 Web 数据库技术<sup>\*</sup>

Developing Web-Database Based on Java Servlet

杨百龙 王 健 张树生

(第二炮兵工程学院 西安710025)(西北工业大学 CAD/CAM 国家专业实验室 西安710072)

**Abstract** This paper gives an introduction of servlet's definition, functions and features, illuminates the working process and principle of Java servlet in detail. With an instance, the process and technology of developing Web-Database using Java servlet are stressed.

**Keywords** Java, Servlet, Web-Database

## 1 引言

Web 数据库由于它的跨地域、跨平台和“0客户端维护”特性,从一开始就受到了人们的高度重视,Microsoft、Netscape 等纷纷提出了各自的技术规范, Sbase、Oracle 等也推出了各自的开发工具;从早期的 CGI(Common Gateway Interface)到低层的 Web server APIs,再到近期的 Object Web,新的开发技术和方法不断涌现。Java Servlet 就是开发支持 Object Web 的应用系统的 Java 解决方案,本文介绍了 Java Servlet 的基本概念,重点讨论了利用 Java Servlet 开发 Web 数据库的基本原理和具体方法。

## 2 什么是 Servlet?

正如 Applet 运行在浏览器上用来扩展浏览器的处理能力一样,Servlet 是用 Java 语言编写的运行在服务器端用来扩展服务器功能的 Java 小程序。Servlet 通过建立在 WWW 上提供请求和响应服务的运行框架来扩展服务器的功能,当客户端发送请求给服务器,服务器就将请求信息传递给一个 Servlet,Servlet 通过 JDBC 访问数据库,构造响应结果信息,经由服务器返回给客户端,从而实现了 Browser→Web→Database 的三级模式交互处理过程。

## 3 Servlet 的功能

Servlets 拓展了服务器的功能,使用户浏览器访问到的不再是一成不变的静态内容,而是基于数据库的、实时交互的动态页面。Servlets 能完成以下功能:

1)建立并返回根据客户请求产生的包含动态内容

的整个页面;

2)生成部分 HTML 页面,该部分页面可以插到已存在的页面之中;

3)同服务器上的数据库或其它 Java 应用程序等资源进行通信;

4)同多个客户通信,接受多个用户请求,并返回处理结果给各用户,例如一个 Servlet 可以是一个多人游戏服务器;

5)建立从服务器到浏览器 Applet 的连接,维持连接的畅通,允许在单个连接上进行多个数据传输;

6)提供对服务器标准例程的定制处理,如一个 Servlet 能修改用户的身份鉴别方式。

## 4 Servlet 的优点

同早期的 CGI(Common Gateway Interface)编程和低层的 Web server APIs,如 ISAPI(Internet Services Application Programming Interface)和 NSAPI(Netscape Connection Application Programming Interface)相比,Java Servlet 具有更好的性能和开发效率,Java 很快就会成为工业标准,人们可以使用单一的、灵活的、功能强大的 Java 工具来满足各种编程需要,Java 的跨平台、跨服务器的支持能力能最大限度地保护 Servlet 的开发投资,因此越来越多的人选择 Java Servlet 来扩展服务器功能,与 CGI 程序相比的优点是:

1)CGI是大多数 Web 服务器支持的标准,它定义了 Web 服务器与外部程序即 CGI 程序交换数据的接口。尽管也可以用 C 语言等编译型语言开发,但 CGI 程序常常用 UNIX 的外壳脚本或 PERL 等解释性语

<sup>\*</sup> 本文研究得到国家863基金资助(863-511-942-007),杨百龙 博士,主要研究方向是 PDM, CIMS 工程,网络数据库,王

健 博士,张树生 教授,博士生导师。

言编写,解释性语言比 Java 等编译型语言运行速度慢;

2) CGI 程序运行在一个与服务器相独立的进程中,每次客户端请求一个 CGI 程序时都要花费大量的启动时间;Servlet 同服务器运行在同一个进程中,启动后,它就保持启动状态,减少了后续客户请求的启动时间。

与 ISAPI, NSAPI 相比的优点是:

1) Web SAPIs 是平台相关的,Java Servlet 是平台无关的;

2) 低层的 SAPIs 常是 C 语言程序,而 Servlet 为 Java 程序,Java 继承了 C 语言的文法,也从 C 语言多年的开发实践中吸取了有益的经验教训;

3) Java 程序不会因内存分配错误而产生内存漏洞,C 程序员常常要花费大量时间查找和修复内存漏洞;

4) Java 程序不使用指针,C 程序中,指针操作常导致难以修复的错误、段违例和潜在的安全问题;

5) Java 程序比 C 程序更加小巧,例如 C 本身不支持线程而 Java 却支持。

## 5 Servlet 的处理过程

如图1所示,用户通过浏览器输入操作命令和数据,向 Web 服务器发出请求,调用 Servlet,Web 服务器收到请求后,加载相应的 Servlet,同时把用户请求信息传给该 Servlet;Servlet 分析用户请求信息,生成对应的 SQL 语句并执行,通过 JDBC 访问数据库服务器中的数据库,获得相应的结果集,生成 HTML 页面返回到客户端。

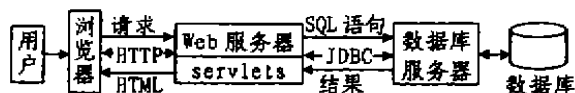


图1 Servlet 工作原理

用户可以通过 URL 调用 Servlet,在<SERVLET>标签中指定 Servlet,或者在 JSP 文件中调用 Servlet,但最常见的是在<FORM>中通过 GET()或 POST()方法调用 Servlet。当一个客户端的请求到达服务器,服务器生成一个请求对象和一个响应对象与该请求相对应,然后服务器调用 Servlet 的 service()方法,并将请求和响应对象传给方法;service()方法通过请求对象获得请求信息,处理请求,并利用响应对象把响应信息传递给客户端;service()方法可以调用其它方法来处理请求如 doGet(),doPost()及自己编写的各种方法等。

当服务器不再需要 Servlet 或 Servlet 的一个新实

例被装入,服务器调用 Servlet 的 destroy()方法。

## 6 Servlet 的开发过程

Servlet 可以用 JSAPI (Java Servlet Application Programming Interface) 及其相关的类和方法来开发,常用的开发工具包有 JSDK 1.0、JSDK1.1,另外也有一些公司所开发的 Java 类库和开发平台,较典型的如 IBM WebSphere,IBM Websphere 是 IBM 公司于98年推出的支持 CORBA/JAVA 标准的 Web 服务器应用开发平台,它除了内嵌了 JSDK 1.1外,还增加了大量附加类和扩展类,支持 CORBA 和动态页面 JSP 及 JavaBeans 技术;它提供了基于 Java,与 Web 服务器及操作系统无关的 Servlet 引擎和与大多数 Web 服务器应用编程接口(API)兼容的服务器插件,支持 Domino Go Webserver, Netscape Enterprise Server, Netscape FastTrack Server, Misoft Internet Information Server, Apache Server 等 Web 服务器,从而能让你的 Servlet 开发实现“开发一次,到处运行”,是一个高度集成的 Servlet 管理和开发平台。

下面结合实例简单说明利用 IBM Websphere 开发 Servlet 的基本过程和方法,该实例的运行过程是:用户通过浏览器输入用户名和口令,Web 服务器上的 Servlet 接受用户请求,从数据库(DBMS 为 ORACLE7.3)用户表(USERS)中查询出用户操作权限(RIGHTS),通过 HTML 页面返回给用户,设计实现环节包括 Servlet 的调用、获得客户端请求信息、访问数据库、返回处理结果,该实例经过适当修改,可作为系统用户登录和权限检查模型块。

### 6.1 Servlet 的调用

在页面<FORM>中通过 Get()或 Post()方法来调用 Servlet 是最常用的方法。下面的代码段用 Get()方法向服务器上的 Servlet (UserLoginServlet) 发请求,例如当用户输入“user1”、“pwd1”后,按“确定”键,则其请求信息以“UserID=user1 & Password=pwd1”的形式被发往 Web 服务器。

```

<FORM METHOD="get"
ACTION="/Servlet/UserLoginServlet">
  用户名<INPUT type="text" name="UserID"><BR>口令
  <INPUT type="password" name="Password"><BR>
  <INPUT type="submit" value="确定"><INPUT type="reset" value="取消"></FORM>
  
```

### 6.2 获得客户端请求信息

用户请求到达 Web 服务器后,服务器生成一个请求对象 HttpServletRequest 和一个响应对象 HttpServletResponse,加载相应的 Servlet(如 UserLoginServlet),调用该 Servlet 的 service()方法,并将请求和响应对象传给该方法.service()方法可以通过服

(下转第89页)

- 2 Georgiadis L, et al. Efficient network QoS provisioning based on per node traffic shaping. *IEEE/ACM Trans. Networking*, 1996, 4(4): 482~501
- 3 Goyal P, Chen H, Vin H. Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks. In: *Proc. SIGCOMM*, Stanford University, CA, 1996. 157~166
- 4 Parekh A K, Gallager R G. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Trans. Networking*, 1994, 2(2): 137~150
- 5 Stiliadis D, Varma A. Latency rate servers: A general model for analysis of traffic scheduling algorithms. In: *Proc. INFOCOM*, San Francisco, CA, 1996. 111~119
- 6 Golestani S J. A self-clocked fair queueing scheme for broadband applications. In: *Proc. INFOCOM*, Toronto, Canada, 1994. 636~646
- 7 Shenker S, Partridge C, Guerin R. Specification of guaranteed quality of service. *IETF RFC 2212*, Sep. 1997
- 8 Toutain F. Decoupled generalized processor sharing: a fair queueing principle for adaptive multimedia applications. In: *Proc. of the 17th Annual Joint Conf. of the IEEE Computer and Communications Societies (IEEE INFOCOM'98)*, San Francisco, CA, IEEE Computer Society, 1998. 291~298
- 9 Bennett J C R, Zhang H. WF<sup>2</sup>Q: worst-case fair weighted fair queueing. In: *Proc. INFOCOM*, San Francisco, CA, 1996. 120~128
- 10 Foster I, Kesselman C. *The Grid: A Blueprint for the New Computing Infrastructure*. Morgan Kaufman, San Francisco, CA, 1998
- 11 Shreedhar M, Varghese G. Efficient fair queueing using deficit round-robin. *IEEE/ACM Trans. Networking*, 1996, 4(3): 375~385
- 12 Katevenis M, Sidiropoulos S, Courcoubetis C. Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. *IEEE J. Select. Areas Commun.*, 1997, SAC-9(8): 1265~1279
- 13 Stiliadis D, Varma A. Frame-based fair queueing: A new traffic scheduling algorithm for packet-switched networks. In: *Proc. SIGMETRICS'97*, 1997
- 14 Georgiadis L, Cuerni R, Parekh A. Optimal multiplexing on a single link: Delay and buffer requirements. *IEEE Trans. Inform. Theory*, 1997, 43(5): 1518~1535
- 15 Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Networking*, 1995, 1(4): 397~413

(上接第98页)

务器生成的请求对象获得用户请求信息,并调用 do-Get()、doPost()或用户开发的各种方法来处理请求。具体过程如下:

```
public void service (HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException
{ String UserID, Password
  UserID=req.getParameter("UserID");
  Password=req.getParameter("Password");
}
```

### 6.3 访问数据库

IBM Websphere 提供了丰富的数据库访问功能,它支持一种 JDBC 连接池(Connection Pool)技术使得不必为每次请求建立新的数据库连接,因此,通常把建立数据库连接操作安排在 Servlet 的 init()方法中,该方法仅在 Servlet 加载时执行一次,建立数据库连接后,对用户的每次请求,服务器都会在调用 Servlet 的 service()方法时把用户的请求信息传递过来,因此可以在 Servlet 的 service()中根据用户请求生成相应的 SQL 语句并执行。下面的代码段说明了 Servlet 从系统配置文件中取得连接参数,通过 JDBC 建立数据库连接的基本过程。

```
ConfigBundle = PropertyResourceBundle (PropertyResourceBundle.getBundle("Myconfig.ini"));
DbUserid = ConfigBundle.getString ("JDBCServlet.dbUserid");
DbPasswd = ConfigBundle.getString ("JDBCServlet.dbPasswd");
HostName = ConfigBundle.getString ("JDBCServlet.HOST-NAME");
TcpPort = ConfigBundle.getString ("JDBCServlet.TCP-PORT");
Instance = ConfigBundle.getString ("JDBCServlet.IN-STANCE");
url ="jdbc:oracle:thin:@"+HostName+": "+TcpPort +
```

```
"," + Instance;
con= DriverManager.getConnection(url, DbUserid, DbPasswd);
```

下面的代码段说明了根据从请求对象中获得的用户请求信息生成并执行 SQL 语句、通过连接对象获得处理结果集的基本过程。

```
Qry-stmt="Select RIGHTS From USERS WHERE USER-NAME="+UserID+" AND
PASSWORD="+Password+";
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery(Qry-stmt);
UserRight=rs.getObject(1); //得到用户权限结果
```

### 6.4 返回处理结果

Servlet 根据数据库处理结果,生成相应的 HTML 页面,通过响应对象返回给用户。具体过程如下:

```
res.setContentType("text/html");
//以下3行禁止动态内容在浏览器中缓存
res.setHeader("Pragma","no-cache");
res.setHeader("Cache-Control","no-cache");
res.setDateHeader("Expires",0);
out=res.getOutputStream();
pw=new PrintWriter(out);
pw.println("<TITLE>(您的权限是:"+UserRight+"</TITLE>");
```

**结束语** 本文介绍了 Java Servlet 的定义、功能、特点和开发过程,同 Microsoft 的 ASP(ActiveX Server Page)相比,Java Servlet 的最大优势是支持目前常用的各种 Web 服务器,良好的跨平台性将最大限度地保护用户的开发投资。

### 参考文献

- 1 宋关福,钟耳顺,王尔琪. WebGIS-基于 Internet 的地理信息系统. *中国图象图形学报*, 1998, 3(3): 251~254
- 2 IBM WebSphere Application Server. Available at: <http://www.software.ibm.com/webservers/>