

基于Servlet链的 Web 数据库开发



汪小栋 董红斌 (武汉大学软件工程国家重点实验室 430072)

摘要: Servlet是一种在服务器端开发Web应用的Java技术,它能比CGI更好地扩展Web服务器的功能。本文简要的介绍了Servlet的基本体系结构,并针对Servlet的高级功能Servlet链提出了一种基于Servlet链的Web数据库开发模型,并给出了一个应用实例。

关键词: Servlet CGI Servlet链 Web数据库

1 引言

随着互联网的高速发展,Web日益成为人们获取、传播和交换信息的重要途径,访问人数的不断增加以及诸如电子商务等数字化产业的崛起,对Web的交互性及数据处理能力提出了更加严格的要求,传统的CGI程序虽然能在一定程度上扩展Web服务器的功能,但CGI对服务器资源的巨大消耗使得CGI无法胜任请求频繁的Web应用。

Servlet是JAVA 2.0中新增的功能,它能象CGI一样扩展Web服务器的功能,但比CGI要更加强大,而且占用的服务器资源要比CGI小的多,因此在性能上要远远超过CGI,更能适应Internet的迅速发展。大部分的主流Web服务器如Apache, IIS, NES等都支持JAVA Servlet。本文简要介绍了Servlet的基本概念和体系结构,并在此基础上提出了一种基于Servlet链的Web数据库开发模型。

2 Servlet与CGI比较

通用网关接口(CGI)在很大程度上扩展了Web服务器的功能,利用CGI可以动态的创建WWW页面,处理HTML表格输入,在Web和其他Internet服务之间架设沟通的渠道。对于请求不十分频繁的Web应用CGI是一种很好的选择。但是对于请求频繁的应用,CGI的性能较差,例如大多数的CGI程序是用perl这样的脚本语言来编写的,针对来自客户端的每一个请求,perl解释器都将单独的创建一

个进程来响应,这会占用大量的服务器资源,另外传统的CGI在安全性上也存在缺陷。Sun公司在Java 2.0中引入的Servlet技术可以很好的解决这些问题。

Servlet运行在Web服务器上,就象Applet必须在浏览器中运行的一样,Servlet与Web服务器之间也有类似的依赖关系。与CGI相比Servlet具有许多优点:

(1) 每个Servlet并是以多线程的形式来运行的,无需为每个运行的Servlet创建一个进程,这就节省了为每个等待响应的请求都创建一个进程所需的巨大开销;

(2) Servlet并发地响应所有来自客户端的请求,Web服务器只需加载一次Servlet,就可以在不同的客户端请求之间保持服务,因此不必像CGI那样多次的装载、启动、卸载程序,既提高了响应速度,又节省了宝贵的内存空间。

(3) Servlet具有Java语言的高度的健壮性,跨平台性等优点,不必像CGI程序那样关心脚本是如何装载的,脚本运行的服务器环境是什么等细节,因此Servlet可以方便的融合到各种Web服务器中。

(4) 编写Servlet十分简单,快捷,并且要比CGI程序容易调试和维护。

3 Servlet体系结构简介

通过与CGI的比较可以看出,Servlet确实在功能上十分强大。下面我们对Servlet的基本体系结构作一个简要介绍。

3.1 Servlet 的类层次结构

Servlet 是利用 Java 的标准扩展包 javax.Servlet 和 javax.Servlet.http 来开发的。一般通过继承 javax.Servlet.http 中的 HttpServlet 来编写自定义的 Servlet,图 1 描述了这两个包中主要的类和接口间的继承关系:

3.2 Servlet 的生命周期

Servlet 的生命周期分为三个阶段:

(1) 装载与初始化 Servlet: 在初始化阶段,服务器创建一个 Servlet 实例,然后调用它的 init 方法。init 在整个周期中只执行一次,调用它的目的是为了初始化一些在运行中要用到的一些参数。

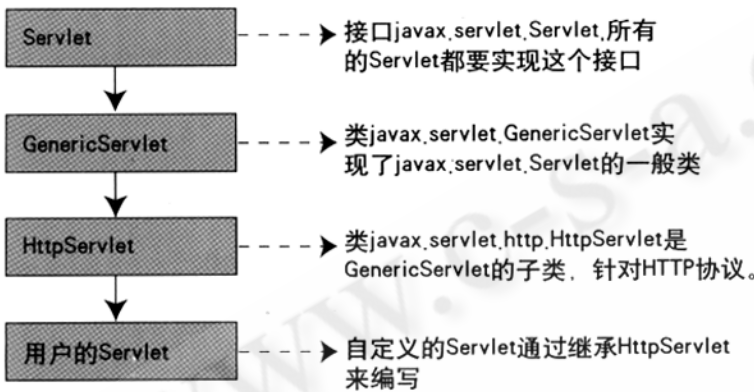


图 1 Servlet 的基本层次结构

(2) 运行 Servlet: 在经过初始化后,Servlet 调用 service 方法来对客户端发出的各个请求进行响应,service 方法根据客户端发送 HTTP 请求的不同方法 (GET,POST 等) 将客户端请求发送给相应的内部方法处理,例如对应于 POST 方法的 HTTP 请求,service 将把它们传给 doPost 方法,由 doPost 方法来进一步处理。在 Servlet 的生命周期中 service 方法会被多个线程并发的调用。

(3) 卸载 Servlet: 当所有的客户端请求都响应过,或者 Servlet 运行的时间超过了一定的时限,服务器将调用 Servlet 的 destroy 方法卸载 Servlet。

一个 Servlet 的运行过程及生命周期如图 2 所示:

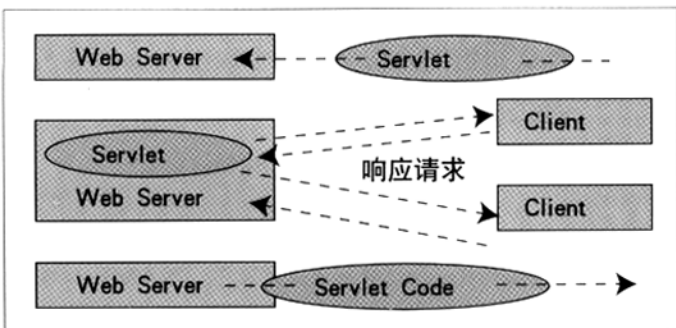


图 2 Servlet 的生命周期

4 Servlet 在 Web 数据库中的应用

Servlet 除了具有持久性,快速性,平台无关性等优点外,还具有一些其他的强大功能,Servlet 链(Servlet Chain)就是其中之一,所谓的 Servlet 链与 UNIX 命令中的管道类似,使我们可以按照特定的顺序将多个 Servlet 链接起来,一个 Servlet 的输出作为下一个 Servlet 的输入,而链中最后一个 Servlet 的输出被返回给客户端,并且这种链接是无缝的,即在链中的每一个 Servlet 察觉不到其他 Servlet 的存在。Servlet 的这一功能能够很好的应用于 Web 数据库的开发,下面我们给出一个基于 Servlet 链的 Web 数据库开发模型如图 3 所示。

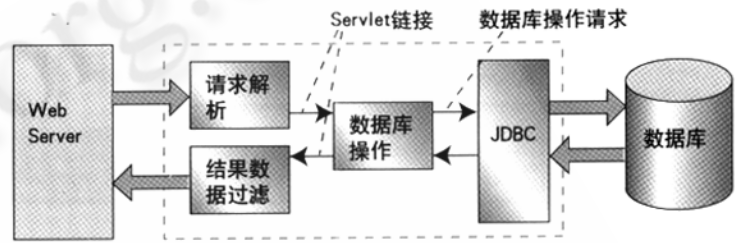


图 3 基于 Servlet 链的 Web 数据库

我们将整个 Web 数据库设计分为三个层次:

(1) 请求解析层: 主要任务是解析从 Web Server 传来的 HTTP 请求,从中得到用户传来的各个参数,需要执行的数据库操作,查询的条件等信息,最终生成相应的 SQL 语句,传给数据库操作层。

(2) 数据库操作层: 主要任务是为每个传送过来的 SQL 语句建立数据库连接,将 SQL 语句提交数据库执行,并将数据库返回的结果按照预先定义的规范进行初始化。

(3) 结果数据过滤层: 将数据库操作层传来的经过初始化的结果数据进行过滤,根据应用的需要转换成相应的 html 文档。

虽然在层次之间存在一定的依赖关系(例如请求解析层的输出是数据库操作层的输入),但 Servlet 链的无缝性,把这种依赖关系降低到最小,在具体的实现中可以根据实际情况将每个层次对应的模块进一步细化为一个或多个 Servlet 来实现,并且各 Servlet 之间基本上是相互独立的。采用这种方式设计的 Web 数据库有以下几个优点:

(4) 结构化好,三个层次的划分,使得在软件设计阶段可以很方便的对 Web 数据库进行模块划分,不同的模块实现不同层次的功能,并且由于功能独立模块之间的依赖程度很小。

(5) 集中式的处理方式,将所有类似的工作集中在了同一层次中进行处理,避免了重复开发,代码的通用性和可移植性较强。

(6) 提供灵活多样的显示界面,通过在结果过滤层中编写多个不同的Servlet,我们可以将相同的查询结果,以各种不同风格的界面显示给客户端的用户。

5 应用实例

在后台数据库webdb中有一张保存用户基本信息的表 user_info, 其表结构如表 1 所示:

表 1 user_info 的表结构

字段	数据类型
Account	Varchar (20)
name	Varchar (20)
Gender	Char (2)
Phone	Char (12)
Email	Varchar (20)
Address	Varchar (20)

这个例子要实现的功能就是根据用户输入的账号(Account), 查询此账号的基本信息, 并将结果返回给客户端的浏览器。利用四个 Servlet 来实现这个功能, 限于篇幅只给出主要代码:

```

· class createSql extends HttpServlet{
    public void doGet(HttpServletRequest req,
        HttpServletResponse res)
        throws IOException{
        /* 提取用户输入的账号, 并创建查询语句 */
        String account=res.getParameter("account");
        String sql = "select * form user_info where
            account='"+account+"'";
        ServletOutputStream out=res.getOutputStream();
        /* 输出查询语句 */
        .....
        out.println("<!-- SQL-->");
        out.println(sql);
        .....
    }
}

```

createSql 对应于请求解析层将客户端的 HTTP 请求

解析成 SQL 语句。

```

· class executeSQL extends HttpServlet{
    public void service(HttpServletRequest req,
        HttpServletResponse res)
        throws IOException{
        /* 复制前一个 Servlet 的头信息 */
        .....
        /* 取出查询语句 */
        ServletInputStream in = req.getInputStream();
        BufferedReeder br=new BufferedReeder(new
            InputStreamReader(in));
        while (!element.equals("<!-- SQL-->"))element =
            br.readLine();
        sql = br.readLine();
        /* 联接数据库执行查询 */
        .....
        rs=sqlstatement.executeQuery(sql);
        .....
        /* 初始化结果数据 */
        .....
        out.println("<!-- NAME- ->");
        out.println(rs.getString(name));
        .....
    }
}

```

executeSQL 对应于数据库操作层执行数据库查询, 并对结果数据初始化。

```

· class showInTable extends HttpServlet{
    public void service(HttpServletRequest req,
        HttpServletResponse res)
        throws IOException{
        /* 复制前一个 Servlet 的头信息 */
        .....
        String name,gender,email,address,telephone;
        /* 从初始化的结果数据中取出查询结果 */
        while(true){
            String element=br.readLine();
            if(element==null)break;
            else if(element.equals("<!-- NAME- ->"))
                name=br.readLine();
            .....
        }
    }
}

```

```

}
/* 将查询结果以表格的形式显示 */
.....
out.println("<table border=1 width=45%
height=176>");
.....
out.println("</table>");
.....
}
}

```

showInTable对应于数据过滤层将初始化的结果数据以表格的形式显示出来。

· showInForm与showInTable的基本相同只是将结

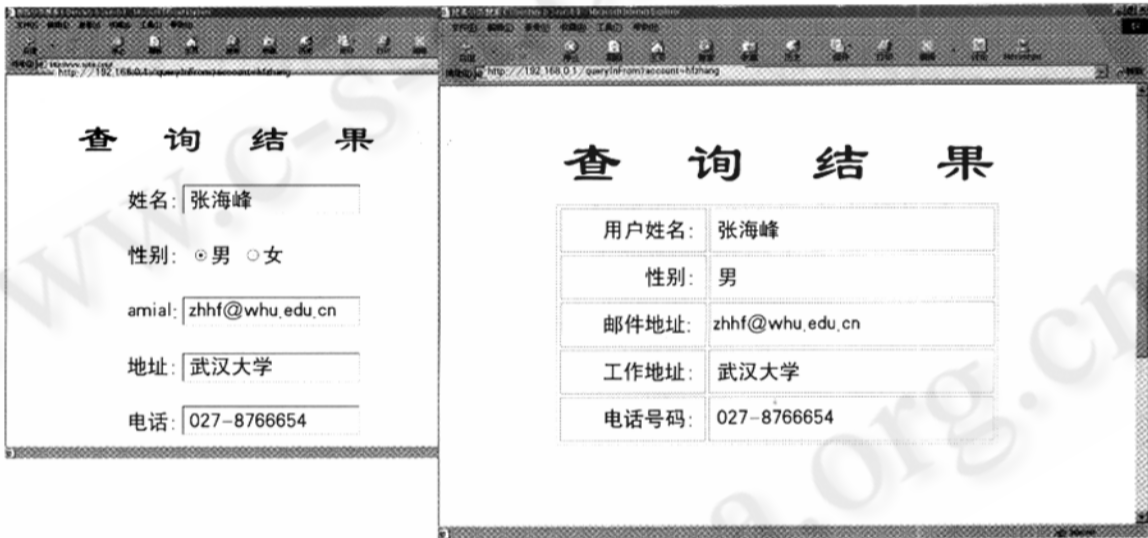
果数据以表单的形式显示出来。

我们采用JRun2.3作为Servlet引擎，在JRun的配置中添加如下两个Servlet链：`/queryInTable createSQL,executeSQL,showInTable;`

`/queryInForm createSQL,executeSQL,showInForm;`

建立了这两个Servlet链后就可以进行查询了，图4给出了它们的显示界面：

从这个例子我们可以看到采用层次划分和Servlet链接，既使得各个Servlet之间相互独立，又可以顺利的协同工作，要得到不同风格的显示界面，只需要在数据过滤层上编写两个Servlet就可以实现，而对应于其他两层的Servlet则无需作任何改动。



(a) queryInForm 的显示界面

(b) queryInTable 的显示界面

图 4

6 结论

Servlet是一种基于Java，应用于Web服务器的新技术，它能很好的替代传统的CGI程序完成各种复杂的Web应用。随着信息产业尤其是电子商务的发展，WWW日益成为人们工作，生活和进行商务活动必不可少的工具和媒介，Servlet以其强大的功能必将在各种Internet应用尤其是Web数据库中大显身手。■

参考文献

- 1 Stefan Zeiger. *Servlet Essentials*. <http://www.novocode.com/docs/servlets-essentials/>.
- 2 Cynthia Bloch. *Servlet Tutorial*. <http://java.sun.com/docs/books/tutorial/servlets/>.
- 3 陆新年, 陆新宇, 刘昊飞. *Java Servlets 编程指南*. 科学出版社.
- 4 Jeffry Dwight, Michael Erin, Robert Niles. *CGI开发使用手册*. 机械工业出版社.