

基于 Java Applet 与 Servlet 的动态网页技术

北京理工大学 9 系

瞿有利 樊孝忠

北京 394 信箱网络技术室

瞿宏阳

WEB 的动态网页已经成为 WEB 站点信息资源组织、共享、发布与信息交互的主要技术。本文介绍了基于 Java Applet 与 Servlet 的三种动态网页技术,详细介绍了系统环境及其配置。

引言

随着 internet/intranet 技术的发展,基于 web 的信息管理应用系统越来越广泛,典型的 Internet/intranet 应用系统,一般设计为所谓三层模型,即系统由数据及信息存储与管理的数据库服务器,对信息进行组织、管理与提供服务的应用服务器,以及客户端浏览器构成。用户在浏览器发出信息请求,服务器对不同的信息请求提供不同的信息服务,这就需要服务器端根据不同的信息请求动态生成网页提供给浏览器。最初动态网页技术多为 CGI 程序设计,但 CGI 程序本身有诸多缺点(如运行效率低、编写困难、不够灵活、平台相关等),而 JAVA 具有面向对象性、简单性、平台独立性和安全性和网络的适应性等特色,所以近年来的动态网页技术转为基于 JAVA 的编程技术。本文采用的 Java Applet 与 Servlet 技术就是一种基于 JAVA 的动态网页技术。

Applet 的特点与 Servlet 简介

1. Applet 的特点

Applet 使 web 的内涵变得更加丰富,例如即时播放声音与背景音乐、即时演示动画、产生各种图形效果等等。它还能在浏览器中显示漂亮美观的电子表格等控件,弥补了 HTML 语言的不足,为用户提供了美观大方的控制界面。

Applet 的作用并不局限于提供美观的界面,良好的交互性使它受到很多程序员的青睐,程序员可以控制 Applet 在发送数据回 web 服务器前做许多额外的工作,比如对数据进行检查,及时发现错误,而不是等数据传回服务器后才发现其中的错误,这就能够降低网络传输代价,提高响应速度。

2. Servlet 的简介

Servlet 是 sun 公司推出的 java 组件。在 web 服务器中引入的 Servlet 程序象 cgi 程序一样,能够接收标准 HTTP 请求(例如,从客户端通过 POST 方式传来的请

求),然后对该请求进行响应,并返回动态生成的数据;Servlet 可以同时接受多个请求,并及时响应这些请求,这样就能够支持类似网络会议那样的应用;它还能够把请求转发给其他服务器或 Servlet 程序,对于具有多个服务器的站点,可以做到负载均衡。

Servlet 是用纯 JAVA 编写的,对系统和服务器的运行环境没有特殊的要求,因此可以不加修改的在多种系统和多种服务器中运行。目前,Servlet 最广泛的应用是嵌入在 HTTP 服务器中,扩展服务器功能。

Applet 端与 Servlet 的通信

由于 Servlet 也遵守 CGI 规范,所以 Applet 向 Servlet 发送数据与向 CGI 程序发送数据的过程是一样的。Applet 发送数据的基本方式有两种:GET 方式和 POST 方式。下面以 POST 方式为例介绍 Applet 和 Servlet 的通信方法(GET 方式的通信方法和 POST 方式的通信方式基本相同)。

Applet 端和 Servlet 通信前的准备工作

准备工作主要包括两项内容:(1)Applet 与 Servlet 程序建立连接;(2)明确通知浏览器将要进行的操作设置。

可用如下语句实现:

```
URL url = new URL(getCodeBase()).toString()
+"myservlet");
URLConnection urlConn = url.openConnection();
urlConn.setDoInput(true);
urlConn.setDoOutput(true);
urlConn.setRequestProperty("Content-type",
"application/x-www-form-urlencoded");
```

其中,myservlet 是 Servlet 程序名,这里也按照 JAVA 的约定,不需写后缀“class”。上述第一条语句生成 myservlet 的 URL。由于 JAVA Applet 的安全限制,Applet 只许与加载 Applet 的 web 服务器建立连接,所以

使用 getCodeBase()来取得此 Applet 的 URL。

第三、四条语句分别设置 applet 为可接收输入和进行输出的。由于 Applet 需要接受 Servlet 的实时响应,所以第五条语句设置 Servlet 和 Applet 传输中不使用 cache。最后一条语句将 HTTP 请求中的 content-type 域设置为: "application/x-www-form-urlencoded"。

与 Servlet 交换数据

发送数据: 首先打开数据输出流,接下来将传给 Servlet 的数据写入流中 (Servlet 程序能够从该流中获得发来的数据),写入流中的数据要遵守标准的 CGI 规范,为此,每一个参数的值需要用 URLEncoder 将其转换成 URL 编码。可用如下语句实现:

```
DataOutputStream printout = new
DataOutputStream(urlConn.getOutputStream());
String content =
    "name"+URLEncoder.encode("qyl")
+"&age="+URLEncoder.encode("25");
printout.writeBytes(content);
printout.flush();
```

发送数据完成后,要关闭数据输出流。

接收数据: 发送完数据后,从输入流中获得 Servlet 程序返回的数据。打开输入流: DataInputStream input = new DataInputStream(urlConn.getInputStream());

然后就可以用 DataInputStream 的各种成员方法从流中读取响应数据了。最后要关闭数据输入流。

Servlet 端与 Applet 的通信

Servlet 程序有三种类型: 实现 javax.servlet.Servlet 接口、实现 javax.servlet.GenericServlet 接口和实现 javax.servlet.http.HttpServlet 接口。一般用于 web 服务器的 Servlet 程序采用第三种类型。HttpServlet 为处理 HTTP 请求提供了一个框架,使得程序员编写处理 HTTP 请求的程序非常容易。本文主要讨论这种 HttpServlet。HttpServlet 是一个抽象类,继承它的 Servlet 类必须至少实现一种它规定的方法。这些方法主要是: init, destroy, doGet, doPost。前两个方法是实现 Servlet 初始化和销毁时所作的一些额外工作,后两个是处理不同的 HTTP 请求类型的方法,分别对应于 GET 类型和 POST 类型请求。当 Servlet 接到 POST 类型请求后,就调用 doPost 方法来处理该 POST 类型请求。当 Servlet 接到 GET 类型请求后,就调用 doGet 方法来处理该 Get 类型请求。

doGet 方法的声明格式如下:

```
public void doGet(HttpServletRequest req,
HttpServletResponse resp)
throws ServletException, IOException
doPost 方法的声明格式如下:
public void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException
```

其中,第一个参数包含了从 applet 端发来的请求所包含的数据,从该参数中还可以访问特定的 HTTP 协议头信息。第二个参数用于 Servlet 返回对此请求的响应数据。

动态网页技术的设计原理

基于 Java Applet 与 Servlet 通信的动态网页技术可有以下三种方法:

1. Applet 采用 GET 方式

基本过程描述如下:

- (1)Applet 调用 showDocument()方法 (这是 Get 方式请求) 请求一个动态网页;
- (2)Servlet 在 doGet 方法中将所生成的动态网页内容写入输出流中;
- (3)浏览器显示该动态页面。

2. Applet 采用 POST 方式 + Java bean

基本过程描述如下:

- (1)Applet 采用 POST 方式向 Servlet 发送数据并请求一个动态网页;
- (2)Servlet 在 doPost 方法中将所生成的动态网页内容写入输出流中;
- (3)Applet 从输入流中接受该动态网页内容,并调用专用的 java bean 在浏览器中显示该动态网页。

3. Applet 采用 POST+GET 方式

基本过程描述如下:

- (1)Applet 将数据通过 POST 方式发送给 Servlet。
- (2)Servlet 利用 POST 传送来的信息构造一个 HTML 页面。
- (3)Servlet 将此 HTML 页面存储在 web 服务器上。
- (4)Servlet 将页面文件名返回给 Applet。
- (5)Applet 利用此文件名生成一个 URL,在 URL 中,包含了一个 GET 请求。
- (6)Applet 调用 showDocument()方法显示 HTML 页面。
- (7)Servlet 接到 GET 请求后,从其参数中取得页面文件名,并将文件内容返回给浏览器。
- (8)浏览器显示 HTML 页面内容。

上述第一种方法最容易实现,但是 Applet 不能通过 GET 方式向 Servlet 发送大量数据;第二种方法需要使用专用的 java bean,而通常 java bean 比较庞大,网上传输费时,并且需要购买。第三种方法(即 POST+GET 方式)克服了上述两种方法的缺点,虽然增加了一次 Applet 与 Servlet 之间的通信,但并未明显降低响应速度。

系统运行环境及其配置

1. 运行 Applet 的系统环境

由于 HTML 协议已经提供了对 Applet 的支持,所以 Applet 得到几乎全部浏览器的支持。

2. 运行 Servlet 的系统环境及其配置

由于 Servlet 出现得较晚,所以不是所有的 web 服务器都支持它,例如,Windows 平台上的 IIS (Internet Information Server) 就不支持它。可喜的是,另一种常用的 web 服务器 Apache 支持 Servlet。本文就以 Apache 为例介绍运行 Servlet 的系统环境。

(1)系统环境(Apache web server)简介。Apache web 服务器也称为 Apache HTTP 服务器,是由 Apache 软件基金会支持的 Apache 开发计划推出的,最早运行在 Unix 系统上,后来逐渐推广到许多系统平台上,如 Linux, Windows NT 等。在本文写作之时,Apache 服务器的最新版本是 1.3.9。随着 Servlet 的推广,Apache 也推出了 Apache JServ 来支持 Servlet。Apache JServ 是 Apache 服务器的一个插件,它不能单独运行,只能运行在 Apache 服务器之上。目前,Apache JServ 的发行版是 1.0。Apache 服务器和 Apache JServ 都是免费发放的,可以从 Apache 的网站 (<http://www.apache.org>) 上下载它们。在系统中安装它们时,首先安装 Apache 服务器,再安装 Apache JServ。

(2)Apache web server 的配置。安装了 Apache 和 Apache JServ 之后,需要对 Apache JServ 进行配置,才能正确运行 Servlet 程序。Apache JServ 和 Apache 服务器都采用配置脚本记录服务器配置,配置脚本是文本文件,存储在 Apache 安装目录 conf 下。对 Apache web server 的配置实际上就是对配置文件进行修改,整个配置过程如下:

①第一步创建 Servlet 区域。修改配置文件的 zone.properties。该参数的格式如下:

repositories=<存储目录 1>, <存储目录 2>, ...et 的目录名、或者包含 Servlet 的 zip 文件名和 jar 文件名。一个 Servlet 区域可以包含多个 Servlet 存储目录,这些存储

目录之间用逗号隔开。如:

```
repositories=e:\servlets1, e:\servlets2
```

修改后的配置文件可以重新命名。

②第二步是将所创建的 Servlet 区域引入 Apache JServ 中。修改配置文件的 jserv.zones。该参数的格式如下:

```
zones= 区域名 1, 区域名 2, ...
```

Apache JServ 中可以引入多个 Servlet 区域,但是一个用户只能有一个区域。区域之间用逗号隔开,区域名可以任意指定。如:

```
zones=qyl, qhy
```

然后,为每一个区域制定相应的配置文件。格式如下:

```
<区域名>.properties= 区域配置文件的全路径
```

例如:

```
qyl.properties=e:\apache\conf\qyl.poperties
```

```
qhy.properties=e:\apache\conf\qhy.properties
```

③第三步是激活 Servlet 区域。修改配置文件 mod_jserv.conf 中的 ApJServMount 参数。该参数的格式如下:

```
ApJServMount 激活名称 jserv-url
```

激活名称指 url 中主机名和所请求的 Servlet 之间的路径名。Jserv-url 和 url 的格式相同,即形如: protocol:/host:port/zone, 其中的 protocol、host 和 port 均可使用缺省值,zone 指在第二步中引入的区域名。例如:

```
ApJServMount /qyl/servlets /qyl
```

```
ApJServMount /qhy/servlets /qhy
```

当用户 qyl 和它的 web 客户请求执行区域 qyl 中的 myservlet.class 这个 Servlet 时,可使用如下的 url: <http://hostname/qyl/servlets/myservlet>。其余类似。

结束语

基于 Applet 与 Servlet 的动态网页技术可广泛适应各种系统平台,具有较强的可移植性,能够方便灵活地对数据进行处,应用前景十分广泛。