

## Re: html servlet flow

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2006-06/msg01904.html>

---

- *From:* "Oliver Wong" <[owong@xxxxxxxxxxxxxxx](mailto:owong@xxxxxxxxxxxxxxx)>
  - *Date:* Mon, 19 Jun 2006 14:13:34 GMT
- 

"Jeff Kish" <[kishjrrj@xxxxxxxxxxxxxxx](mailto:kishjrrj@xxxxxxxxxxxxxxx)> wrote in message  
[news:k4r692pmjoogfsdos562aqa7nuempqeat8@xxxxxxxxxxxxxxx](mailto:news:k4r692pmjoogfsdos562aqa7nuempqeat8@xxxxxxxxxxxxxxx)

Is this right?

The general gist of it looks right. As for the details, I'm not sure. I've never actually done JSP programming, nor used Tomcat.

Say there is a hard coded jsp page, say:  
mypage.jsp in the right dir under tomcat,  
if the user types in the browser address bar:  
<http://tomcat/jmypage.jsp>

and presses return, then...

Then the browser packs up the 'mypage.jsp' and  
formats it in the protocol of http, and send it to  
the address (tomcat here).

Probably browser sends '/jmypage.jsp'. The 'j' is probably a typo, but the slash is important. A browsers' request is always an absolute path, AFAIK.

Then tomcat recognizes the protocol, unpacks the  
request, and sees "mypage.jsp".  
Then it looks wherever it is legal for jsp pages to exist,  
and starts processing the mypage.jsp?

This is *probably* what Tomcat does, but webserver are free to not actually look for a physical file, but instead generate a response in some other fashion. Imagine for example, that the browser requests for the url '/random', and the webserver recognizes this as a special address which doesn't actually point to a file, but instead indicates that the server should generate a random 32-bit integer and send its decimal string

Re: html servlet flow

representation as a response. None of this would be violating the HTTP protocol AFAIK.

Most webservers simply look for files matching the provided name, as you've said though.

If it is a simple page, it packs up the code/html in the jsp page into the response and puts it in http format and sends it to the browser.

Right, but a minor terminology nitpick here: HTTP is not a format, but a protocol. Traditionally, no transformation is done to the data. The server sends a message to the browser, whose header says "This is an text/html file. It's length is 320 bytes." and then immediately starts sending the data as-is.

More recent servers might negotiate compression schemes. I'm not sure exactly how it works, but I imagine it's something like when the browser requests the page, it'll say "GET me /mypage.jsp. Oh and by the way, if you can gzip it, that'll be lovely." And the server will generate the response (perhaps by reading the %WEBROOT%/mypage.jsp file), and if it knows how, it'll gzip it. If it doesn't know how, it just sends the content as is.

When the browser gets the response, it recognizes it is http, unpacks it, and starts processing the html in the response.

The browser doesn't have to do any recognition at this point. It all happens under one connection. At the IP level, the browser connects to the server, and does its GET request. The server processes this, sends back its response, and closes the connection.

And this 'response' isn't the response parameter in the doGet method, it is just the general 'response' from the Tomcat.

Extra information like headers etc are in the actual doGet 'response' variable.

I didn't understand these last two parts. They sound JSP/Tomcat specific.

– Oliver

.