Chapter 3. Servlet Basics

Topics in This Chapter

- The basic structure of servlets
- A simple servlet that generates plain text
- A servlet that generates HTML
- Servlets and packages
- Some utilities that help build HTML
- The servlet life cycle
- How to deal with multithreading problems
- Tools for interactively talking to servlets
- Servlet debugging strategies

As discussed in Chapter 1, servlets are programs that run on a Web or application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server. Their job is to perform the following tasks, as illustrated in Figure 3-1.

1. **Read the explicit data sent by the client.**

   The end user normally enters this data in an HTML form on a Web page. However, the data could also come from an applet or a custom HTTP client program.

2. **Read the implicit HTTP request data sent by the browser.**

   Figure 3-1 shows a single arrow going from the client to the Web server (the layer in which servlets and JSP pages execute), but there are really *two* varieties of data: the explicit data the end user enters in a form and the behind-the-scenes HTTP information. Both types of data are critical to effective development. The HTTP information includes cookies, media types and compression schemes the browser understands, and so forth; it is discussed in Chapter 5.

3. **Generate the results.**

   This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly. Your real data may be in a relational database. Fine. But your database probably doesn't speak HTTP or return results in HTML, so the Web browser can't talk directly to the database. The same argument applies to most other applications. You need the Web middle layer to extract the incoming data from the HTTP stream, talk to the application, and embed the results inside a document.

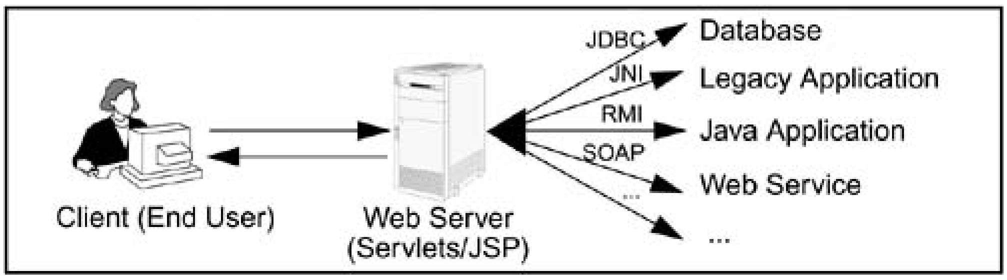
4. **Send the explicit data (i.e., the document) to the client.**

This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, or even a compressed format like gzip that is layered on top of some other underlying format.

5. **Send the implicit HTTP response data.**

   Figure 3-1 shows a single arrow going from the Web middle layer (the servlet or JSP page) to the client, but there are really *two* varieties of data sent: the document itself and the behind-the-scenes HTTP information. Both types of data are critical to effective development. Sending HTTP response data involves telling the browser or other client what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks. These tasks are discussed in Chapters 6–8.

Figure 3-1. The role of Web middleware.



In principle, servlets are not restricted to Web or application servers that handle HTTP requests but can be used for other types of servers as well. For example, servlets could be embedded in FTP or mail servers to extend their functionality. In practice, however, this use of servlets has not caught on, and we discuss only HTTP servlets.