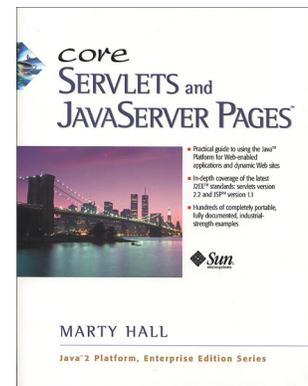


Server-side Java

Richard A. Slywczak



Core Servlets and JavaServer Pages

Marty Hall, The Johns Hopkins University Applied Physics Laboratory
 Published by Sun Microsystems Press/Prentice Hall PTR, Upper Saddle River, NJ, 2000,
 575 pp., \$42.99

During the past decade, the World Wide Web has experienced tremendous growth. From its origins as a document sharing system, e-business has transformed the Web into a full-application processing system. In the process, we have been introduced to a veritable alphabet soup of new technologies from HTTP and HTML to CSS, XML, and ASP. Two of the more significant technologies, developed by Sun Microsystems, have been servlets and JavaServer Pages (JSP). *Core Servlets and JavaServer Pages* (Sun Microsystems Press) by Marty Hall is an excellent text on both of these technologies. For the novice, it provides a how-to manual for servlets and JSP in a clear and straightforward manner. For the advanced developer, the book is an excellent reference for class definitions and advanced topics such as using JavaBeans with JSP, developing custom tag libraries, and using Java database connectivity (JDBC) and database connection pooling. The text is divided into three parts: servlets, JSP, and associated supporting technologies to develop front and back ends for servlets and JSP. Numerous code samples throughout help reinforce the reader's understanding. Since its debut at JavaOne 2000, *Core Servlets and JavaServer Pages* has reached number 37 at Amazon.com's overall bestseller list and has been translated into Bulgarian, Chinese, Czech, French, German, Hebrew, Japanese, Korean, Polish, Russian, and Spanish.

Part I provides extensive coverage of servlets based on the 2.1 and 2.2 specifications developed by Sun, and I highlight some examples here. Servlets are Java's approach to common gateway interface (CGI) programming. Servlets, written in Java, run via Web servers and provide an interface between the client (i.e., Web browser) and some resource on the server. Chapter 2 offers a nice introduction to servlet basics. Their basic structure and life cycle are developed, and the servlet sample code demonstrates the principles shown in the text. Since servlets are a server-side technology, a communications mechanism is needed between them and the Web application. The strength of the book is shown by the introduction and development of the communications between the browser and servlet. Chapter 4 explains the HTTP codes and headers that the servlet can receive and process from the browser. In Chapters 6 and 7, the HTTP codes that the servlet can send to the browser are defined. In each of these chapters, Marty has done an excellent job of listing and explaining each code and header in great detail. He also gives a brief introduction on how information is passed to the servlet via form data; this topic is covered in complete detail in Chapter 16. Part I concludes with two very important topics. First, cookies, a much talked-about but misunderstood issue, are covered in depth with an explanation of the cookie application programming interface (API). Example programs demonstrate how cookie data

are passed between the client and server. This section concludes with a discussion of session tracking, a mechanism that maintains contextual information about the clients that visit a site. An introduction to the API is presented and its implementation is demonstrated with a very nice example of a fictional bookstore. This chapter is a must for e-business developers.

In Part II, *Core Servlets and JavaServer Pages* offers the reader in-depth coverage of JSP based on the 1.0 and 1.1 specification. JSP enables the developer to mix regular, static HTML with dynamically built, generated content by interspersing Java code into HTML. Another outstanding feature of this book is that JSP and custom tag libraries are developed and explained incrementally to simplify each topic. The three different types of JSP scripting elements—expressions, scriptlets, and declarations—are covered in detail, and each is supported by numerous examples. In Chapter 14, Marty tackles how to create custom JSP tag libraries. Starting with basic tags, each step adds attributes, body content, and, finally, tags that can modify their own body content. Each part of the tag library is explained in a clear, straightforward manner and reinforced by numerous examples. To readers interested in JavaBeans, the book explains how to create beans and access them through JSP. While not a definitive guide on JavaBeans, enough information is presented to help the developer understand how the two technologies interact and write useful applications. This section ends with a discussion on integrating servlets and JSP. An all-inclusive example of a travel reservation system that makes use of servlets and JSP demonstrates most of the major topics covered in these two sections.

Both servlets and JSP are server-side technologies and are part of the middleware layer. To provide a complete application, you need a user interface and, depending on your application, you might need access to a data repository. This is the function of Part III; it fills in the missing

pieces for a complete application. In Chapter 16, HTML Forms, used to create a user interface, are reviewed in detail. Here, Marty explains and shows examples of every form element that exists in HTML. If forms are not dynamic enough, the next chapter provides a look at using applets as the user interface. This chapter is not meant to teach applet programming but demonstrates how to access servlets through an applet. Both of these chapters provide enough information to show the developer how to construct an interface to a servlet and/or JSP. The section ends with a discussion of JDBC and database connection pooling. While not in depth, this is an excellent introduction to JDBC, presenting the information you need to be up and running with JDBC in no time. Marty concisely outlines the steps on loading the driver, making a connection, issuing a query, and processing the results. Since inefficient use of a database system can impede performance, the chapter covers optimization topics such as precompiled queries and database connection pooling. This chapter covers a lot of material and does so very efficiently.

Any good technical manual must give the reader a way to find information quickly and easily. For *Core Servlets and JavaServer Pages*, that is Appendix A, which summarizes the book, chapter by chapter, and emphasizes the major points and references Web sites and APIs. The appendix is easy to search through to find topics of interest.

There have been a number of books written about servlets and JSP, but it is hard to beat *Core Servlets and JavaServer Pages*. Marty does an excellent job of presenting the material in a clear, concise manner, and the many examples help reinforce the topics presented. This publication is one for your bookshelf.

Richard A. Slywczak
Glenn Research Center
NASA

Server Side Programming(Server.java). Server class : The main server implementation is easy and similar to the previous article. The following points will help understand Server implementation : The server runs an infinite loop to keep accepting incoming requests. When a request comes, it assigns a new thread to handle the communication part. The sever also stores the client name into a vector, to keep a track of connected devices. The vector stores the thread object corresponding to the current request. Server-Side Row Model. However many real world applications contain much larger datasets, often involving millions of records. In this case it simply isn't feasible to load all the data into the browser in one go. Instead data will need to be lazy-loaded as required and then purged to limit the memory footprint in the browser.Â Server-Side Row Model Interfaces. Our Java service will use the following request and response objects Is series mein mein ham servlet (java) and jsp ki technology ki asani se sikhenge . Servlet technology is used to create a web application (resides at server side and generates a dynamic web page). Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.