

Advanced Internet Technologies

Nirman Kaur

This reference book can be useful for
BBA, MBA, B.Com, BMS, M.Com, BCA, MCA
and many more courses for Various Universities



**NEERAJ
PUBLICATIONS**
www.neerajbooks.com

Published by:



NEERAJ PUBLICATIONS

(Publishers of Educational Books)

Sales Office : 1507, 1st Floor,

Nai Sarak, Delhi-110 006

E-mail: info@neerajbooks.com

Website: www.neerajbooks.com

© **Reserved with the Publishers only.**

Typesetting by: Competent Computers

Terms & Conditions for Buying E-Book

- The User must Read & Accept the Terms and Conditions (T&C) carefully before clicking on the accept option for Buying the Online Soft Copy of E-books. Under this Particular Facility you may buy only the Online Soft Copy of E-books, no Hard Copy or Printed Copy shall be provided under this facility.
- These E-Books are valid for 365 days online reading only (From the Date of Purchase) and no kind of Downloading, Printing, Copying, etc. are allowed in this facility as these products are just for Online Reading in your Mobile / Tablet / Computers.
- All the online soft copy E-books given in this website shall contain a diffused watermark on nearly every page to protect the material from being pirated / copy / misused, etc.
- This is a Chargeable Facility / Provision to Buy the Online Soft Copy of E-books available online through our Website Which a Subscriber / Buyer may Read Online on his or her Mobile / Tablet / Computer. The E-books content and their answer given in these Soft Copy provides you just the approximate pattern of the actual Answer. However, the actual Content / Study Material / Assignments / Question Papers might somewhat vary in its contents, distribution of marks and their level of difficulty.
- These E-Books are prepared by the author for the help, guidance and reference of the student to get an idea of how he/she can study easily in a short time duration. Content matter & Sample answers given in this E-Book may be Seen as the Guide/Reference Material only. Neither the publisher nor the author or seller will be responsible for any damage or loss due to any mistake, error or discrepancy as we do not claim the Accuracy of these solution / Answers. Any Omission or Error is highly regretted though every care has been taken while preparing these E-Books. Any mistake, error or discrepancy noted may be brought to the publishers notice which shall be taken care of in the next edition. Please consult your Teacher/Tutor or refer to the prescribed & recommended study material of the university / board / institute / Govt. of India Publication or notification if you have any doubts or confusions before you appear in the exam or Prepare your Assignments before submitting to the University/Board/Institute.
- Publisher / Study Badshah / shall remain the custodian of the Contents right / Copy Right of the Content of these reference E-books given / being offered at the website www.studybadshah.com.
- The User agrees Not to reproduce, duplicate, copy, sell, resell or exploit for any commercial purposes, any portion of these Services / Facilities, use of the Service / Facility, or access to the Service / Facility.
- The Price of these E-books may be Revised / Changed without any Prior Notice.
- The time duration of providing this online reading facility of 365 days may be alter or change by studybadshah.com without any Prior Notice.
- The Right to accept the order or reject the order of any E-books made by any customer is reserved with www.studybadshah.com only.
- All material prewritten or custom written is intended for the sole purpose of research and exemplary purposes only. We encourage you to use our material as a research and study aid only. Plagiarism is a crime, and we condone such behaviour. Please use our material responsibly.
- In any Dispute What so ever Maximum Anyone can Claim is the Cost of a particular E-book which he had paid to Study Badshah company / website.
- If In case any Reader/Student has paid for any E-Book and is unable to Access the same at our Website for Online Reading Due to any Technical Error/ Web Admin Issue / Server Blockage at our Website www.studybadshah.com then He will be send a New Link for that Particular E-Book to Access the same and if Still the Issue is Not Resolved Because of Technical Error/ Web Admin Issue / Server Blockage at our website then His Amount for that Particular Purchase will be refunded by our website via PayTM.
- All the Terms, Matters & Disputes are Subjected to "Delhi" Jurisdiction Only.

CONTENTS

S.No.	Page
1. Servlet Programming	1
2. Java Database Connectivity	20
3. Java Server Pages–I	41
4. Java Server Pages–II	58
5. Introduction to Java Beans	74
6. Enterprise Java Beans: Architecture	90
7. EJB: Deploying Enterprise Java Beans	101
8. XML: Extensible Markup Language	117
9. Web Security Concepts	127
10. Security Implementation	133
11. Case Study	148

Sample Preview of The Chapter

Published by:



**NEERAJ
PUBLICATIONS**

www.neerajbooks.com

ADVANCED INTERNET TECHNOLOGIES

SERVLETS AND JSP PROGRAMMING

Servlet Programming



INTRODUCTION

Initially, the web sites were the collection of web pages that linked together by Hyper Text Markup Language (HTML). In addition to that, now-a-days various advance features have been added in web server technologies such as multimedia, e-commerce applications, online banking etc. These features can be created by using technologies like Java Server Pages (JSP) and Active Server Pages (ASP). Servlets are the Java platform technology that can be run on any platform or any server. Servlets enhance the functionality of web applications. They are dynamically loaded at runtime when needed. When a client sends a request the web server/container initiates the required servlet. The servlet processes the client request and sends the response back to the server. The servlets are not bind to a specific client server protocol but they are commonly used with HTTP (Hypertext Transfer Protocol). HTTP is a request-response oriented protocol. It consists of a request method, a URI (Uniform Resource Identifier), header fields and a body. An HTTP response contains a result code and header fields and a body. The service method of Http Servlet dispatches a request to different Java method for different request methods. It recognizes the standard HTTP

methods such as GET, HEAD, PUT, TRACE, OPTIONS, POST etc. Servlets use classes in Java packages `javax.servlet` and `javax.servlet.http`. Servlets which provides an advance server side extension and follow the standard framework.

CHAPTER AT A GLANCE

HOW TO INSTALL SERVLET ENGINE/WEB SERVER

Java Servlet is used for writing server side programming language; hence it is required to execute it in a Java Virtual Machine by using a service called Servlet Engine. A space servlet engine is software that increases the scope of web server software such as IIS (Internet Information Server, Apache etc.) and enables the web server software to execute servlets. Generally, this Servlet engine is contained in Servlet Engine or it could be added as a module. Few web servers have a built in Servlet engine like Sun Java web server, Gefion Software's Life web server etc. But other web servers like Netscape Enterprise Server, IIS and Apache group require a Servlet Engine add on module.

Servlets must run on web servers with built in web containers such as the Planet web server or on a standalone

2 / NEERAJ : ADVANCED INTERNET TECHNOLOGIES

Servlet container like Tomcat. Here, we will be using Apache Tomcat 4.0. It is an open source software implementation of the Java Servlet and Java Server Pages technologies.

To configure Tomcat as a stand alone server we need to download Tomcat 4.0 and JDK 1.3 (Java Development Kit) standard edition. In order to install and run this container we need to do the following:

- (a) Download and install it from the location <http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.1/bin/>.
- (b) Download jakarta-tomcat-4.0.1.zip file to C:\temp. Now, to download the Sun JDK, go to <http://java.sun.com/j2se/1.3/download.html>, choose the Windows and click download. Scroll down and accept the license agreement. Then click the FTP download and put the download file j2sdk-1_3_1_03-win.exe file to C:\temp\ directory.

Now, we need to create two environment variables, CATALINA_HOME and JAVA_HOME.

- (a) Right click on My Computer.
- (b) Select Properties
- (c) Select the Advanced tab.
- (d) Select the Environment Variables button.
- (e) In the System Variable group, select New.
- (f) In the Variable Name field, type JAVA_HOME.
- (g) In the Variable Value field, type the name of the path to your JVM. (as C:\Program Files\Java)
- (h) Select OK.
- (i) In the System Variable group, select New.
- (j) In the Variable Name field, type CATALINA_HOME.
- (k) In the Variable value field, type the name of the path to Tomcat (as C:\ProgramFiles\ Apache Tomcat 4.0)
- (l) Select OK.
- (m) Select OK.
- (n) Select OK.

Once the installation is done, the next question arises how to start and stop the server. For which follow these steps:

- To start the server, start a command prompt window and run the command:
C:\jakarta-tomcat\bin\startup.bat
- To stop the server, run the command:
C:\jakarta-tomcat\bin\shutdown.bat

YOUR FIRST JAVA SERVLET

Once the basic container is installed and configured, next step is to write the servlet programming. The servlet programming interface i.e. Java Servlet API has two packages: javax.servlet and javax.servlet.http. These

packages contain many classes which are being used by the servlet. The Java Servlet API is a part of J2EE platform and has following advantages over other scripts called CGI.

- Servlets add dynamic behaviour to servers. It makes very easy to write complex services for web based applications.
- Because servlets are written in java, it has full access to Java's advance features such as database connectivity, network awareness, object orientation etc.
- Servlets can be interfaced with different databases like Oracle and SQL server.

Now, lets' create a new file and save it as firstservlet.java

```
1. import java.io.*;
2. import java.servlet.*;
3. import java.servlet.http.*;
4. public class firstservlet extends
   HttpServlet
5. {
6.     public void doGet(Http Servlet
   Request req, HttpServletResponse
   res)
7.         throws ServletException,
   IOException
8.     {
9.         res.setContentType("text/
   html") ;
10.    PrintWriter out = res.get Writer()
   ;
11.    out.println("<HTML>
   <HEAD>");
12.    out.println("<TITLE>First   Java
   Servlet</TITLE></ HEAD>");
13.    out.println("<BODY>Hello,
   Client</BODY></HTML>");
14.    out.close();
15. }
16. }
```

Lines from 1-3: are called packages which contain the classes.

Line 4: firstservlet is a class. This is a standard base class for HttpServlets.

Lines from 6-7: The HttpServletResponse object used to set the content type of the response that we have to send. All the response headers need to be set before ServletoutputStrem.

Line 10: writes text to the response message.

The PrintWriter gets closed when out.close() is encountered.

Compiling and Running the Servlet: To compile the above written servlet, type the following command at the Dos prompt: javac firstservlet.java

If the program gets compiled successfully, a `firstservlet.class` file will be created. Now, to run the servlet, type the following address in the browser:

```
http://localhost:8080/star/servlet/  
com.stardeveloper.servlets.firstservlet
```

As we have seen certain methods are invoked by server in order to handling the requests. Each time server sends a request to a servlet. The service method() accepts two parameters - request object and a response object. The request object used to send the request whereas the response object used to return the response.

SERVLET LIFE CYCLE: A Java Servlet has a life cycle, which defines servlet processing i.e. how the servlet receives and responds to requests. The servlet life cycle is defined by the `javax.servlet.Servlet` interface. It consists of following steps:

1. Initially, the servlet class is loaded by the container.
2. The Container implements the `init()` method only once during the life of the servlet.
3. After initialization, the service method() is invoked. This method invokes two parameters `HttpServletRequest` and `HttpServletResponse`. Once the request is determined, the same is send to the appropriate method.

4. At last, the `destroy()` method is called. This method takes the servlet out of service and invoked only once during the lifecycle of a servlet.

Three main stages in the life of Java Servlet are:

1. **Servlet Initialization:** At the outset, the servlet's constructor is called with the `init()` method. The purpose of this method is to allow a servlet to perform the initialization before the `HttpRequest` being invoked. This method is called only once on any given instance.
2. **Servlet Execution:** Once the `init()` method is called, the next step is to start executing application logic in a servlet i.e. all the requests received by the servlet container are forwarded to servlet's appropriate service() method. Further, `HttpServlet` class breaks this method into various methods as `doGet()`, `doPost()`, `doDelete()`, `doPut()`, `doOptions()` and `doTrace()` depending on the type of the request it receives.
3. **Servlet Destruction:** Once the servlet finishes execution of service and there is no request pending, servlet's `destroy()` method is called. This method is used to destroy a servlet instance, which is out of service. Few activities which can be implemented in this method are performing de-allocation of those resources which were

being allocated during initialization, closing database connection etc.

Lets' understand these stages with the help of the following program:

```
import java.io.*;  
import java.servlet.*;  
import java.servlet.http.*;  
import java.io.PrintWriter ;  
import java.io.IOException ;  
public class stagesdemo extends HttpServlet  
{  
    public void init() throws  
ServletException  
    {  
        System.out.println("Invoking  
init() method");  
    }  
    public void destroy()  
    {  
        System.out.println("Invoking  
destroy() method");  
    }  
    /*Process HTTP doGet() request*/  
    public void doGet(HttpServletRequest  
Request request, HttpServletResponse  
Response response)  
        throws IOException,  
        ServletException  
    {  
        response.setContentType("text/  
html") ;  
        PrintWriter out = response.  
getWriter() ;  
        out.println("<HTML><HEAD>");  
        out.println("<TITLE>Servlet  
Life Cycle Demo</TITLE></  
HEAD>");  
        out.println("<BODY><p>Calling  
get() method </p></BODY></  
HTML>");  
        out.close();  
    }  
    /*Process HTTP doPost() request*/  
    public void doPost(HttpServletRequest  
Request request, HttpServletResponse  
Response response)  
        throws IOException,  
        ServletException  
    {  
        response.setContentType("text/  
html") ;  
        PrintWriter out = response.  
getWriter() ;  
        out.println("<HTML><HEAD>");  
    }  
}
```

4 / NEERAJ : ADVANCED INTERNET TECHNOLOGIES

```
        out.println("<TITLE>Servlet  
        Life Cycle Demo</TITLE></  
HEAD>");  
        out.println("<BODY><p>Calling  
        Post() method </p></BODY></  
HTML>");  
        out.close();  
    }  
}
```

HTTPSERVLET REQUEST INTERFACE

HttpServletRequest and HttpServletResponse are two main interfaces in the Servlet. The Servlet container creates an HttpServletRequest object and passes it as an argument to the servlet's service methods such as doGet() and doPost(). This interface provides access to an input stream and enables it to read data from the client. Various methods are provided which enables the servlet to process the clients' request. Few of them are listed below:

- 1. get_cookies
public Cookie[] get_cookies

It returns an array that contains all the cookie objects present in this request. If no cookies are present in the request, this method returns Null.

- 2. get_header
public java.lang.String
get_header(java.lang.String name)

It returns the value of the specified request header as a string. A parameter name is passed as a string specify the header name. It returns Null if the request does not have a header of that name.

- 3. get_method
public java.lang.String
get_method()

It returns the name of the HTTP method with which this request was made e.g. Get, Post or Put.

- 4. get_query_string
public java.lang.String
get_query_string()

It returns a string contained in the request URL. If no query string is present it returns NULL.

- 5. get_servlet_path
public java.lang.string.
get_servlet_path()

It returns a string that contains the name or path of the servlet as specified in the request URL.

- 6. get_session
public HttpSession
get_session(boolean create)

It returns the current HttpSession associated with this request. If no argument is passed in this method, a new session will be created. In case it is called with a Boolean

argument then, the new session will be created only if the argument is true.

- 7. get_parameter(String name)
public String
get_parameter(String name)

It returns the value associated with a parameter, which is sent to the servlet as a part of GET and POST request.

HTTPSERVLETRESPONSE INTERFACE

HttpServletResponse interface provides functionality in sending a response. It provides access to an output stream and allows the servlet to send data to client. The getWriter() method obtains a reference to PrintWriter object, which is used to send the text to the client. Following methods are used to formulate response to client:

- 1. add_cookie
public void
add_cookie(Cookie cookie);

It adds the specified cookie to the response and called be multiple times to set more than one cookie.

- 2. send_error
public void
send_error(int sc)
throws java.io.
IOException;
public void
send_error(int sc,
java.lang.String msg)
throws java.io.
IOException;

It is used to send an error message to the client using the specified status code and clearing the buffer. If the response has been committed, it throws an IllegalStateException. Once this method is used, the response should be considered to be committed. The parameter sc represents the error status code. If an input or output exception occurs then throws java.io.IOException called and if the response was committed then IllegalStateException called.

- 3. send_date_header
public void
send_date_header(java.
lang.String name, long date)

It is used to set a response header with the specified given name and date value. If the header is already set, it overwrites previous value with new value.

- 4. set_header
public void
set_header(java.
lang.String name, java.lang.
String value)

It sets a response header with the specified name and string value. It is used to check the presence of a header.

- 5. add_header
public void
add_header(java.lang.
String name, java.lang.String
value)