Servlet

Introdution to servlet

Servlet is small program that execute on the server side of a web connection. Just as applet extend the functionality of web browser the applet extend the functionality of web server.

In order to understand the advantages of servlet, you must have basic understanding of how web browser communicates with the web server.

Consider a request for static page. A user enters a URL into browser. The browser generates http request to a specific file. The file is returned by http response. Web server map this particular request for this purpose. The http header in the response indicates the content. Source of web page as MIME type of text/html.

1. What are the Advantage of Servlet Over "Traditional" CGI?

Java servlet is more efficient, easier to use, more powerful, more portable, and cheaper than traditional CGI and than many alternative CGI-like technologies. (More importantly, servlet developers get paid more than Perl programmers :-).

- **Efficient.** With traditional CGI, a new process is started for each HTTP request. If the CGI program does a relatively fast operation, the overhead of starting the process can dominate the execution time. With servlets, the Java Virtual Machine stays up, and each request is handled by a lightweight Java thread, not a heavyweight operating system process. Similarly, in traditional CGI, if there are *N* simultaneous request to the same CGI program, then the code for the CGI program is loaded into memory N times. With servlets, however, there are *N* threads but only a single copy of the servlet class.
- **Convenient.** Hey, you already know Java. Why learn Perl too? Besides the convenience of being able to use a familiar language, servlets have an extensive infrastructure for automatically parsing and decoding HTML form data, reading and setting HTTP headers, handling cookies, tracking sessions, and many other such utilities.
- **Powerful.** Java servlets let you easily do several things that are difficult or impossible with regular CGI. For one thing, servlets can talk directly to the Web server (regular CGI programs can't). This simplifies operations that need to look up images and other data stored in standard places. Servlets can also share data among each other, making useful things like database connection pools easy to implement. They can also maintain

information from request to request, simplifying things like session tracking and caching of previous computations.

- **Portable.** Servlets are written in Java and follow a well-standardized API. Consequently, servlets written for, say I-Planet Enterprise Server can run virtually unchanged on Apache, Microsoft IIS, or Web Star. Servlets are supported directly or via a plug in on almost every major Web server.
- **Inexpensive.** There are a number of free or very inexpensive Web servers available that are good for "personal" use or low-volume Web sites. However, with the major exception of Apache, which is free, most commercial-quality Web servers are relatively expensive. Nevertheless, once you have a Web server, no matter the cost of that server, adding servlet support to it (if it doesn't come preconfigured to support servlets) is generally free or cheap.

2. **What is servlet? What are the phases of servlet life cycle? Give an example.**

Servlets are small programs that execute on the server side of a web connection. Just as applet extend the functionality of web browser the applet extend the functionality of web server.

Servlet class is loaded.

Servlet class is loaded when first request to web container.

servlet instance is created:

Web container creates the instance of servlet class only once.

init method is invoked:

It class the init method when it loads the instance. It is used to intialise servlet.

Syntax of init method is

public void init(ServletConfig config) throws ServletException

Service method is invoked:

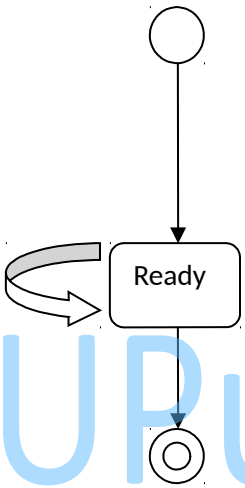
Web container calls service method each time when request for the servlet is received. If servlet is not initialized it calls init then it calls the service method. Syntax of service method is as follows

public void service(Servlet request, ServletResponse response) throws ServletException, IOException

Destroy method is invoked.

The web container calls the destroy method before it removes the servlet from service. It gives servlet an opportunity to clean up memory, resources etc. Servlet destroy method has following syntax.

public void destroy().



There are three states of servlet new, ready, end. It is in new state when servlet is created. The servlet instance is created when it is in new state. After invoking the init () method servlet comes to ready state. In ready state servlet invokes destroy method it comes to end state.

3. Explain about deployment descriptor

Deployment descriptor is a file located in the WEB-INF directory that controls the behavior of a java servlet and java server pages. The file is called the web.xml file and contains the header, DOCTYPE, and web app element. The web app element should contain a servlet element with three elements. These are servlet name, servlet class, and init-param.

The servlet name elements contain the name used to access the java servlet. The servlet class is the name of the java servlet class. init-param is the name of an initialization parameter that is used when request is made to the java servlet.

Example file:

<?xml version=”1.0” encoding-“ISO-8859=1”?>…..XML header

< !DOCTYPE web-app PUBLIC "~//Sun Microsystems, Inc.?? DTD Web Application2.2//EN"> ..doctype

```
<web-app>
        <servlet>
        <servlet-name>MyJavaservlet</servlet-name>
        <servlet-class>myPackage.MyJavaservletClass</servlet-class>
        <init-param><param-name>parameter1</param-name>
                <param-value>735</param-value>
        </init-param>
        </servlet>
</web-app>
```

4. **How to read data from client in servlet?**

- A client uses either the GET or POST method to pass information to a java servlet. Depending on the method used by the client either doGet() or doPost() method is called in servlet.
- Data sent by a client is read into java servlet by calling getParameter() method of HttpservletRequest() object that instantiated in the argument list of doGet() method and doPost() method.
- getParameter() requires one argument, which is the name of parameter that contains the data sent by the client. getParameter() returns the String object.
- String object contains the value assigned by the client. An empty string object is returned when it does not assign a value to the parameter. Also a null is retuned when parameter is not returned in the client.
- getParameterValues() used to return the array of string objects.

Example code

Html code that calls a servlet:

```
<FORM  ACTION="/servlet/myservlets.js2">
Enter Email Address :< INPUT TYPE="TEXT" NAME="email">
<INPUT TYPE="SUBMIT">
</FORM>
```

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class js2 extends Httpservlet {
public void doGet(HttpservletRequset request,HttpservletResponse response)
throws servletException , IOException {
//String email;
//Email=request.getParameter("email");
Respose.setContentType("text/html");
PrinterWriter  pw=response.getWriter();
pw.println("<HTML>\n" +
"HEAD><TITLE> Java Servlet</TITLE></HEAD>\n" +
"<BODY>\n"+
//"<p>MY Email Address :" +email +"</p>\n" +
<h1> My First Servlet
"</BODY>\n" +
</HTML>");
}
}
```

5. **How to read HTTP Request Headers?**

A request from client contains two components these are implicit data, such as email address explicit data at HTTP request header. Servlet can read these request headers to process the data component of the request.

Example of HTTP header:
Accept: image.jpg, image.gif,*/*
Accept- Encoding: Zip
Cookie: CustNum-12345
Host:www.mywebsite.com
Referer: http://www.mywebsite.com/index.html

The uses of HTTP header:
Accept: Identifies the mail extension
Accept-Charset : Identifies the character set that can be used by browser.
Cookie returns the cookies to server.
Host: contains host portal.
Referrer: Contains the URL of the web page that is currently displayed in the browser.

A java servlet can read an HTTP request header by calling the getHeader() method of the HttpservletRequest object. getHeader() requires one argument which is the name of the http request header.

getHeader()

6. **How to send data to client and writing the HTTP Response Header?**

A java servlet responds to a client's request by reading client data and HTTP request headers, and then processing information based on the nature of the request.

For example, a client request for information about merchandise in an online product catalog requires the java servlet to search the product database to retrieve product information and then format the product information into a web page, which is returned to client.

There are two ways in which java servlet replies to client request. These are sent by sending information to the response stream and sending information in http response header. The http response header is similar to the http request header.

Explicit data are sent by creating an instance of the PrintWriter object and then using println() method to transmit the information to the client.

Implcit data example: HTTP/1.1 200 OK
Content-Type:text/plain

My Response

Java servlet can write to the HTTP response header by calling setStatus() method requires one argument which is an integer that represent the status code.

Response.setStatus(100);

7. **Explain about Cookies in servlet.**

Cookies are text files stored on the client computer and they are kept for various information tracking purpose. Java Servlets transparently supports HTTP cookies.

There are three steps involved in identifying returning users:

- Server script sends a set of cookies to the browser. For example name, age, or identification number etc.
- Browser stores this information on local machine for future use.
- When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

Setting Cookies with Servlet:

Setting cookies with servlet involves three steps:

(1) Creating a Cookie object: You call the Cookie constructor with a cookie name and a cookie value, both of which are strings.

Cookie cookie = new Cookie("key","value");

(2) Setting the maximum age: You use setMaxAge to specify how long (in seconds) the cookie should be valid. Following would set up a cookie for 24 hours.

cookie.setMaxAge(60*60*24);

(3) Sending the Cookie into the HTTP response headers: You use **response.addCookie** to add cookies in the HTTP response header as follows:

response.addCookie(cookie);

Writing Cookie

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloForm extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
        throws ServletException, IOException
  {
     Cookie myCookie = new Cookie("user id", 123);
    myCookie.setMaxAge(60*60);
     response.addCookie(myCookie );
     response.setContentType("text/html");
     PrintWriter out = response.getWriter();
        out.println( "<html>\n" +
            "<head><title>" + My Cookie + "</title></head>\n" +
          "<nody>\n" +
          "<h1>+ My Cookie +"<h1>\n" +
          "<p> Cookie Written + </p>\n" +
          "</body></HTML>");
}
 }
```

Reading Cookies with Servlet:

To read cookies, you need to create an array of *javax.servlet.http.Cookie* objects by calling the **getCookies( )** method of *HttpServletRequest*. Then cycle through the array, and use getName() and getValue() methods to access each cookie and associated value.

Example:Let us read cookies which we have set in previous example:

```
import java.io.*;import javax.servlet.*;import javax.servlet.http.*;
 public class ReadCookies extends HttpServlet {
 public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
 {
    Cookie cookie;
        Cookie[]  cookies;
    cookies = request.getCookies();
     response.setContentType("text/html");
     PrintWriter out = response.getWriter();
    String title = "Reading Cookies Example";
    out.println( "<html>\n" +
            "<head><title>" + title + "</title></head>\n" );
                if( cookies != null ){
          out.println("<h2> Found Cookies Name and Value</h2>");
            for (int i = 0; i < cookies.length; i++){
              cookie = cookies[i];
             out.print("Name : " + cookie.getName( ) + ",  ");
        out.print("Value: " + cookie.getValue( )+" <br/>");
      }
```

```
}else{    out.println( "<h2>No cookies founds</h2>");
}     out.println("</body>");     out.println("</html>");
}}
```

8. **Explain Session Tracking:**
    1. A session is created each time a client requests service from a java servlet. The java. The java servlet processes the request and response accordingly, after which the session is terminated. Many times the same client follows with another request to the same client follows with another request to the same java servlet, java servlet requires information regarding the previous session to process request.
    2. However , HTTP is stateless protocol, meaning that there is not hold over from the previous sessions.
    3. Java servlet is capable of tracking sessions by using Httpsession API. It determines if the request is a continuation from an existing session or new session.
    4. A java servlet calls a getSession() method of HttpservletRequset object, which returns a session object if it is a new session. The getSession() method requires one argument which is Boolean true. Returns session object.

Syntax :

```
HttpSession s1=request.getSession(true);
```

JSP program

A jsp is java server page is server side program that is similar in design and functionality to a java servlet.

A JSP is called by a client to provide web services, the nature of which depends on client application.

A jsp is simpler to create than a java servlet because a jsp is written in HTML rather than with the java programming language. . There are three methods that are automatically called when jsp is requested and when jsp terminates normally. These are the jspInit() method , the jspDestroy()m ethod and service() method.

A jspInit() is identical ti init() method of java servlet. It is called when first time jsp is called.

A jspDestroy() is identical to destroy() method of servlet. The destroy() method is automatically called when jsp erminates normally.It is not called when jsp terminates abruptly. It is used for placing clean up codes.

1. Explain JSP tags(repeated question)

A jsp tag consists of a combination of HTML tags and JSP tags. JSP tags define java code that is to be executed before the output of jsp program is sent to the browser.

A jsp tag begin with a <%, which is followed by java code , and wnds with %>,

There ia an XML version of jsp tag <jap:TagId></jsp:TagId>

A jsp tags are embedded into the HTML component of a jsp program and are processed by Jsp virtual engine such as Tomcat.

Java code associated with jsp tag are executed and sent to browser.

There are five types of jsp tags :

Comment tag :A comment tag opens with <%-- and close with --%> and is follwed by a comment that usually describes the functionality of statements that follow a comment tag.

Declaration statement tags: A declartion statement tag opens with <%! And is followed by declaration statements that define the variables, object, and methods that are avilabe to other component of jsp program.

Directive tags: A directive tag opens with <%@ and commands the jsp virtual engine to perform a specific task, such as importing java package required by objects and methods used in a declaration statement. The directive tag closes with %> . There are commonly used in directives import, include , and taglib. The import tag is used to import java packages into the jsp program. Include is used for importing file. Taglib is used for including file.

Example:

<%@ page import=”import java.sql.*” ; %>

<%@ include file=”keogh\books.html” %>

<%@ taglib url=”myTags.tld” ; %>

Expression tags: An expression tag opens with <%= and is used for an expression statement whose result page replaces the expression tag when the jsp virtual engine resolves JSP tags. An expression tag closes with %>

Scriptlet tag: A sciptlet tag opens with <% and contains commonly used java control statements and loops. And Scriptlet tag closes with %>

2. **How variables and objcts declared in JSP program?**

   You can declare java variables and objects that are used in a JSP program by using the same codin technique used to declare them in java. JSP declaration statements must appear as jsp tag

   Ex:

```
<html>
<head>
        <title> Jsp Programming </title>
<.head>
<body>
        <%! Int age=29; %><p> Your age is : <%=age%> </p>
</body>
</html>
```


3. **How method are declared and used in jsp programs?**

   Methods are defined same way as it is defined in jsp program, except these are placed in JSP tag.methods are declared in JSP decalration tag. The jsp calls method in side the expression tag.

   Example:

```
<html>
<head>
        <title> Jsp programming</title>
</head>
<body>
<%! int add(int n1, int n2)
        {
        int c;
        c=a+b;
        return c;
```

```
        }
    %>
    <p> Addition of two numbers : <%= add(45,46)%> </p>
    </body></html>
```

4. **Explain the control statements of JSP with example program:**
   1. One of the most powerful features avilable in JSP is the ability to change the flow of the program to truly create dynamic content for a web based on conditions received from the browser.
   2. Ther are two control statements used to change the flow of program are "if " and "switch" statement , both of which are also used to direct the flow of a java program.

      Ex:

```
<html>
<head>
        <title> JSP Programming </title>
</head>
<body>
        <%! int grade=26; %>
        </body>
<% if(grade >69) { %>
        <p> You Passed !</p>
<% } else { %>
<p> Better Luck Next Time</p>
<% } %>
</body>
</html>
```

5. **Looping Statement of JSP**

   Jsp loops are nearly identical to loops that you use in your java program except you can repeat the html tags

   There are three kind of jsp loop that are commonly used in jsp program.

   Ex: for loop , while loop , do while .

   Loop plays an important role in JSP database program. The following program is example for "FOR LOOP".

```
<html><head><title>For Loop Example</title></head>
<body>
```

```
<%
   for (int i = 0; i < 10;i++) {
%>
<p> Hello World</p>
<% } %> </body>
</html>
```

6. **Explain Requset String generated by browser. how to read a request string in jsp?**
   1. A browser generate requst string whenever the submit button is selected. The user requests the string consists of URL and the query the string.
      Example of request string:
      http://www.jimkeogh.com/jsp/?fname=" Bob" & lname ="Smith"
   2. Your jsp programneeds to parse the query string to extract the values of fields that are to be processed by your program. You can parse the query string by using the methods of the request object.
   3. getParameter(Name) method used to parse a value of a specific field that are to be processed by your program
   4. code to process the requset string
      <%! String FirstName =requst.getParameter(fname);
      String LastName =requst.getParameter(lname); %>
   5. Copying from multivalued field such as selection list field can be tricky multivalued fields are handled by using getParameterValues()
   6. Other than requset string url has protocols, port no, the host name
   7. **Write the JSP program to create and read cookie called "EMPID" and that has value "AN2536".**

Cookie is small piece of information created b a JSP program that is stored on the clien's hard dik by the browser. Cookie isare used to store various kind of information, such as user preference. The cookies are created by using Cookie class.

Create cookie:

```
<html>
<head>
<title> creating cookie</title>
</head>
<body>
<%! String MyCookieName="EMPID";
```

```
        String UserValue=”AN2536”;
%>
</body>
</html>
```

Reading Cookie:

```
<html>
<head>
<title>reading cookie </title>
</head>
<body>
<% String myCookieName=”EMPID”;
String myCookieValue;
String CName, CValue;
int found=0;
Cookie[] cookies=request.getCoookies();
        for( int i=0;i<cookies.length;i++) {
        CName= cookies[i].getName();
        CValue =cookies[i].getValue();
        If(myCookieName.equals(CName)) {
        found=1;
        myCookieValue=Cvalue; } }
        If(found== 1)  { %>
        <p> Cookie Name =  < %= CName %> </p>
        <p> Cookie Value =  < %= CValue %> </p>
        <%  } %> </body></html>
```


8. **Explain steps to configure tomcat.**
   i. Jsp program programs are executed by a JSP virtual machine that run on a web server.
   ii. We can download and install JSP virtual machine.
   iii. Installation Steps
      Connect to Jakarta.apache.org.
      Select down load
      Select Binaries to display the binary Download Page.
      Create a folder from the root directory called tomcat.
      Download latest release.
      Unzip Jakarta-tomcat.zip.
      The extraction process creates the following folder in the tomcat directory: bin, conf, doc, lib, src, and webapps
      Modify the batch file , which is located in the \tomcat\bin folder. Change the JAVA_HOME variable is assigned the pathe where JDK is installed on your computer.
      Open dos window and type \tomcat\bin\tomcat to start Tomcat.
      Open your browser. Enter http://localhost:8080.

Tomcat home page is displayed on the screen verifying that Tomcat is running.

9. Explain how session objects are created.

A JSP database system ia able to share information among JSP programs within a session by using a session object. Each time a session is created , a unique ID is assigned to the session and stored as acookie.

A unique ID enables JSP program to track multiple session session simultaneously while maintaining data integrity of each session. The session ID is used to prevent the intermingling of each session.

Create session Object:

```
<html><head><title> Jsp Session</title></head>
<body>
<% ! String AtName=”Product”;
    String AtValue =”1234”;
    Session.setAttributes(AtName, AtValue);
%></body></html>
```


In session object we can store information about purchases as session attributes can be retrived and modified each time the jsp program runs. setAttributes() used for creating attributes.

Read Session Object:

getAttributeNames() methos returns names of all the attribbutes as Enumeration , the attributes are processsed.

```
<html><head><title> Jsp Session</title></head>
<body><% !
    Enumaration purchases=session.getAttributeNames();
    String AtName=(String) attributeNames.nextElement();
    String AtValu=(String) session.getAttribute(AtName); %>
    <p> Attirbute Name <%= AtName %>  </p>
    <p> Attribute Value <% = Atvalue %> </p>
    <% } %> %></body></html>
```

III Internal Questions

1. What are different types of JSP tags describe the JSP tags with example.(Dec 2011)
2. Define JSP. Explain two types of control statements with example.(Dec 2012)
3. Write the JSP program to create and read cookie called "EMPID" and that has value "AN2536"(Dec 2012)
4. What is RMI? Briefly explain working of RMI in java.(Dec 2012)
5. Department has set the grade for the subject Java as follows:

   Above 90: A, 80 - 89: B , 70 -79 : C
   Below 70 = fail. Sham enters his marks for the subject Java in the interface provided. Write a JSP program to accept the mark and display the grade.(Jun 2011)

6. Briefly explain the RMI in Java (June 2011)

7. Discuss different types of JSP tags (Jun 2011)

8. Write a program using RMI such as client and server program in which client sends hello message to server and replies to client (June 2012)

9. Develop simple java servlet that handle HTTP Request and Response (June 2012)

10. Explain javax.servlet packages(June 2012)

11. What is difference between JSP and Servlet? (june 2012)

12. What are the advantages of JSP program?(jun 2010)

13. What are servlets? Briefly explain the application of servlets in web programming (dec 2010)

14. Explain the life cycle of a servlet. (dec 2010)
15. Write a java servlet which reads two parameters from the webpage, say value 1 and value 2 , which are type integer and finds the sum of the two value and return back the result as a webpage.(dec 2010)

16. Provide java syntax for the following: (dec 2010)
    i) Handling HTTP requests and responses
    ii) Using cookies
    iii) Session tracking
17. List out difference between CGI and servlet.
18. What is cookie list out methods defined by cookie. Write a servlet program to read cookie.
19. Write a jsp program to add cookie name "User Id" and value"JB007"
20. Describe in detail how tomcat web server is configured in develop of servlet life cycle.