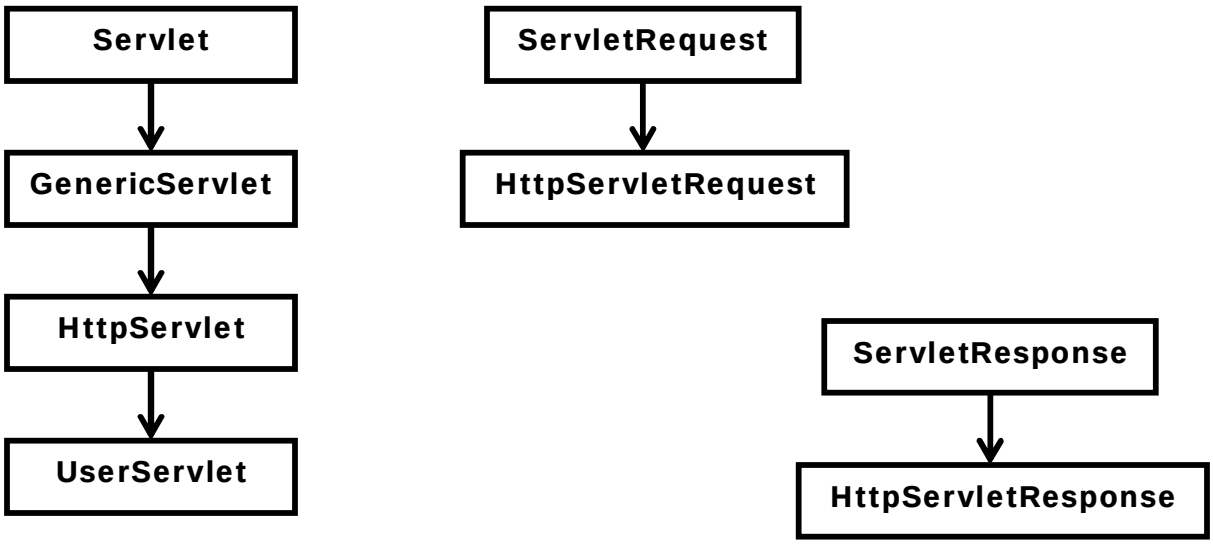
Servlets

Servlets are modules of java code that run in a **Server** application to answer the Clients' requests. Servlets make use of the java standard extension classes in the packages **javax.servlet** and **javax.servlet.http**.

The traditional way of adding functionality to a web server is the **Common Gateway Interface** (CGI), a language independent interface that allows the server to start an external process which gets information about a request through environment variables, the command line and its standard input stream and writes response data to its standard output stream. Each request is answered in a separate process by a separate instance of the CGI program, or **CGI script** (often these scripts are written in languages like **Perl**).

Servlets have several advantages over CGI. A servlet does not run in a separate process. This removes the overhead of creating a new process for each request. It stays in memory between requests. But a CGI script needs to be loaded and started for each CGI request. There is only a single instance which answers all requests concurrently. This saves memory and allows a servlet to easily manage persistent data.

Inheritance Hierarchy of Servlets



Life Cycle of a Servlet / Methods of Servlet interface

1. **init()**

   This method gets executed, when the server application loads the Servlet. This method is called only once during the Servlet's life cycle. It is guaranteed to finish before any service requests are accepted. After initialization, the network service does not call the init method again unless it reloads the servlet after it has unloaded and destroyed it.

2. **service()**

   When the Servlet is initialized, its **service** (reqst, respns) method is called for every request to the Servlet. This method is called concurrently, so it should be implemented in a thread-safe manner. Service requests are not handled until servlet initialization has completed

3. **destroy()**

   When the Servlet needs to be unloaded this method is called. This method is called only once during the life cycle of a Servlet. In the case of long-running operations, there could be other threads running service requests when destroy is called. The servlet writer is responsible for making sure that any threads still in the service method complete.

When the Servlet container first loads a Servlet, it invokes the Servlet's **init()** method to initialize the servlet. Then, as requests are made to execute the Servlet, the Servlet container repeatedly invokes the Servlet's **service()** method to provide the required service. Finally, when the Servlet container no longer needs the Servlet, it invokes the Servlet's **destroy()** method and unloads it from memory. Note that during the lifetime of a single Servlet instance, the init() and destroy() methods will be invoked only once, whereas the service() method will be invoked many times, once each time a request is made to execute the Servlet.

Hyper Text Transfer Protocol (HTTP) is a request-response oriented protocol which is used by a client (browser) to send a request to a web server. An **HTTP Request** consists of a **URI**, header fields and a body. An **HTTP Response** contains a **result code** and again header fields and a body.

An **HTTP Request** consists of a request method, a request URL, header fields, and a body. HTTP defines the following request methods:

GET	: Retrieves the resource identified by the request URL
HEAD	: Returns the headers identified by the request URL
POST	: Sends data of unlimited length to the Web server
PUT	: Stores a resource under the request URL
DELETE	: Removes the resource identified by the request URL
OPTIONS	: Returns the HTTP methods the server supports
TRACE	: Returns the header fields sent with the TRACE request

An **HTTP Response** contains a result code, header fields, and a body. The HTTP protocol expects the result code and all header fields to be returned before any body content. Some commonly used status codes include the following:

404: Indicates that the requested resource is not available

401: Indicates that the request requires HTTP authentication

500: Indicates an error inside the HTTP server which prevented it from fulfilling the request

503: Indicates that the HTTP server is temporarily overloaded, and unable to handle the request

Methods of HttpServlet interface:

1. **doGet()**
2. **doPost()**
3. **doPut()**
4. **doHead()**
5. **doOptions()**
6. **doTrace()**

All the above methods take two parameters: **HttpServletRequest**, and **HttpServletResponse**. The **doGet()** method is used for the request. The **doHead()** method is performed by calling doGet(). With a doGet() request the parameters are encoded in the URL, with a **doPost()** request they are transmitted in the body. The **doPut()** method is used to upload the resources to a web server and **doDelete()** request deletes the resources. The methods **doOptions()** and **doTrace()** are usually not overridden.

ServletRequest interface Methods:

1. int contentLength()
2. String getContentType()
3. String getProtocol()
4. String getServerName()
5. int getServerPort()
6. ServletInputStream getInputStream()
7. BufferedReader getReader()
8. String getParameter(String parameter-name)
9. Enumeration getParameterNames()
10. String[] getParameterValues(String param-name)

ServletResponse interface Methods:

1. void setContentLength(int length)
2. void setContentType(String type)
3. PrintWriter getWriter()
4. ServletOutputStream getOutputStream()

HttpServletRequest interface Methods:

1. Cookie[] getCookies()
2. String getRequestURI()
3. String getServletPath()
4. String getPathInfo()
5. HttpSession getSession()
6. String getRequestSessionId()
7. boolean isRequestedSessionIdValid()

HttpServletResponse interface Methods:

1. void addCookie(Cookie x)
2. void setStatus(int code)
3. void setStatus(int code, String message)
4. void sendRedirect(String location)

To run a Servlet, we should go for any Web Server, such as Apache **Tomcat,** Bea **Web Logic**, Sun Java Web Server, or **ServletRunner** which comes with Java Servlet Development Kit (**JSDK**).

The Directories under JSDK are as follows:

src : contains the Servlets' class files which you can import them into your program.

bin : contains **ServletRunner** utility.

examples : contains the user defined Servlets .class files. The web server takes the Servlets from this directory.

doc : Contains documentation files.

Steps to Run a Servlet:

Phase - I

1. Write code for the Servlet.
2. Generate class files (Compile the Source).

Phase - II

Use any of the above available Web Servers.

1. ServletRunner Utility:

a. Install JSDK2.0

b. Set the environment variable classpath to **"C:\JSDK2.0\Src;."**

c. Copy the **Servlets' .class** files to **C:\JSDK2.0\examples** directory

d. Run the **ServletRunner** utility in DOS, providing port number, as follows. The default port number is 8080.

```
C:\JSDK2.0\Bin > ServletRunner –p 5050
```

-p option binds the port number 5050 to the server

2. Apache Tomcat:

1. Install the latest version jakarta-tomcat. This will load the server to **C:\Program Files\Apache Software Foundation \Tomcat**.

2. **Copy** the Servlet **.class files** to **\Tomcat\webapps\ROOT \WEB-INF\classes** directory.

3. **Remove** the **comments** in **\Tomcat\conf\web.xml** file.

4. Make changes in **Servlet mappings** in the file **\Tomcat\ webapps\ROOT\WEB-INF\web.xml,** if you need**.**

5. Start the Tomcat Server.

Phase – III:

Open the browser, and type **http://localhost:8080/servlet /Servlet-name** in the address bar, in case you are running the Servlet under **ServletRunner** utility. You will find the Servlet gets executed.

(or)

Open the browser, and type **http://localhost:8080/servlet /Servlet-name** in the address bar, in case you are running the Servlet under **Tomcat** server. You will find the Servlet gets executed.

1. A simple servlet that writes some text on a webpage.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class FirstServlet extends HttpServlet
{
      static int counter=0;
      public FirstServlet()
      {
            super();
            System.out.println("Servlet is Instantiated");
      }
      public void init()
      {
            System.out.println("Servlet is Initialized");
      }
```

```
public void service(HttpServletRequest req,HttpServletResponse res) throws
                                                    ServletException, IOException
{
  res.setContentType("text/html");
  PrintWriter pw = res.getWriter();
  pw.println("<html><head>");
  pw.println("<title>HelloServlet Example</title></head>");
  pw.println("<body><center><h1>Welcome To The Servlets World</h1>");
  pw.println("<h1> Your Visitor Number : "+counter++"</h1>");
  pw.println("</center></body></html>");
}
public void destroy()
 {
    System.out.println("HelloServlet Derstroyed...");
 }
}
```

2. Passing parameters to Servlets from an html program

a. HTML program that passes parameters m1, m2 & m3 to a Servlet

```
<html>

<head>
      <title> Passing parameters to a Servlet</title>
</head>

<body>
<center>

      <form method=post
           action="http://localhost:8080/servlet/ParameterizedServlet">

           Enter marks for M1: <input type =text name=m1><br>
           Enter marks for M2: <input type =text name=m2><br>
           Enter marks for M3: <input type =text name=m3><br>
          <br>

           <input type=submit value="Calculate Total">
      </form>

</center>
</body>

</html>
```

b. Servlet which processes the parameters sent from the HTML program

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ParameterizedServlet extends HttpServlet
{
      public void service(HttpServletRequest req,HttpServletResponse res)
                                          throws ServletException,IOException
      {
            res.setContentType("text/html");
            PrintWriter pw = res.getWriter();

            int m1 = Integer.parseInt(req.getParameter("m1"));
            int m2 = Integer.parseInt(req.getParameter("m2"));
            int m3 = Integer.parseInt(req.getParameter("m3"));
            int tot=m1+m2+m3;
            pw.println("<html><head>");
            pw.println("<body bgcolor = chocolate> <center>");
            pw.println("<font face = verdana color = navy>");
            pw.println("<h1>The total marks are  " + tot + "</h1>");
            pw.println("</font></center></body></html>");
      }
}
```

3. Calling a Servlet from another Servlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class CallServlet extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse res)
                                          throws ServletException,IOException
{
    res.setContentType("text/html");
    PrintWriter pw = res.getWriter();
    pw.println("<html><head>");
    pw.println("<title>Calling another Servlet</title></head>");
    pw.println("<body><form method=post action='JDBCServlet'>");
    pw.println("<center><h1>Welcome To The Servlets World</h1>");
    pw.println("<input type=submit value='Call JDBCServlet'>");
    pw.println("</center></form></body></html>");
      }
}
```

4. Accessing Data of a DataBase through a Servlet

a. HTML program

```
<html>
 <body>
      <center>
          <form method=post
                action="http://localhost:8080/servlet/JDBCServlet">

               <input type=submit value="Display Students Details">
          </form>
     </center>
</body>
</html>
```

b. Servlet program

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class JDBCServlet extends HttpServlet
{
public void doPost(HttpServletRequest req,HttpServletResponse res) throws
                                                ServletException, IOException
  {
      res.setContentType("text/html");
      PrintWriter pw = res.getWriter();
      pw.println("<html><head>");
      pw.println("<title>All Students Information</title></head>");
      pw.println("<body bgcolor=gray text=navy><center>");
      pw.println("<font face = verdana>");
      pw.println("<h1>All Students Information</h1><br>");
      pw.println("<hr color = blue>");
      pw.println("<table border=2 bordercolor=blue>");
      pw.println("<tr> <th width=50> RNO </th>
                                  <th width=150> NAME </th> </tr>");
      try
      {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      Connection con=
      DriverManager.getConnection("jdbc:odbc:oracledsn","scott","tiger");
      Statement st = con.createStatement();
      ResultSet rs = st.executeQuery("select * from student order by rno");
      while(rs.next())
```

```
        {
        pw.println("<tr><td>"+rs.getInt("rno")+"</td><td>"+
                                rs.getString("name")+"</td></tr>");
        }
        pw.println("</table><font>");
        pw.println("<br><hr color = blue>");
        pw.println("</center></body></html>");
        }
        catch(Exception e)
        {
        pw.println("<strong>The Error:= " + e.getMessage() + "</strong>");
        }
    }
}
```

Sessions & Cookies

The mechanism to maintain the state about a series of requests from the same user across a period of time is called a **Session**. Sessions are shared among the Servlets a client access. It is convenient for the application made up of multiple Servlets. Sessions are used to maintain state and user identity across multiple page requests.

With the method **getSession()** of **HttpServletRequest object**, we can create an HttpSession. This method should be called before sending any response data with the help of a Writer.

e.g.

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

Methods of HttpSession interface

1. String getId()
2. long getCreationTime()
3. long getLastAccessedTime()
4. void invalidate()
5. void putValue(String name, object value)
6. Object getValue(String name)
7. void removeValue(String name)
8. String[] getValuenames()
9. boolean isNew()

Cookies are a way for the server to send information to a client to store and for the server to later retrieve its data from that client. A server can provide one or more cookies to a client.

The class **Cookie** is used for session management with HTTP and HTTPS protocols. Cookies are used to get user agents to hold small amounts of state associated with a user's web browsing. Common applications for cookies include storing user preferences, automating low security user sign on facilities, and helping collect data used for "shopping cart" style applications.

Cookies are named, and have a single value. They may have optional attributes, including a comment presented to the user, path and domain qualifiers for which hosts see the cookie, a maximum age, and a version.

Cookies are assigned by servers, using fields added to HTTP response headers. Cookies are saved one at a time into HTTP response headers, using the **HttpServletResponse.addCookie()** method. User agents are expected to support 20 Cookies per host, of at least 4KB each; use of large numbers of cookies is discouraged. Cookies are passed back to those servers using fields added to HTTP request headers. HTTP request fields are retrieved using the **HttpServletRequest.getCookies()** method. This returns all of the cookies found in the request. Several cookies with the same name can be returned; they have different path attributes.

Cookie class Methods

1. Cookie(String name, String value)
2. void setValue(String newValue)
3. String getValue()
4. String getName()
5. void setPath(String uri)
6. String getPath()
7. void setMaxAge(int expiry) in seconds
8. int getMaxAge()
9. void setComment(String purpose)
10. String getComment()

5. Servlet program on Sessions

a. HTML Program that calls the a SessionServlet

```
<html>
<head><title>Program on Creating Sessions</title></head>
<body>
<center>
<form method=post name=form1
                    action="http://localhost:8080/servlet/SessServlet">
<table>
       <tr>
       <td>Enter Name :</td>
       <td><input type=text name="name"></td>
       </tr>
       <tr>
       <td>Enter Marks :</td>
       <td><input type=text name="marks"></td>
       </tr>
       <tr>
       <td><input type=submit value="Create a Session"></td>
       <td><input type=reset value="Reset"
                          onClick='document.form1.name.focus()'></td>
       </tr>
</table></form></body></html>
```

b. Servlet that creates a Session

```
import java.io.*;
import javax.servlet.http.*;
public class SessServlet extends HttpServlet
{
  public void service(HttpServletRequest req,HttpServletResponse res)
 {
     try
    {
       String s1=req.getParameter("name");
       String s2=req.getParameter("marks");
       HttpSession hs=req.getSession(true);
       hs.putValue(s1,s2);
       res.sendRedirect("http://localhost:8080/servlet/SessionReadServlet");
     }
     catch(Exception e)
       {
             System.out.println(e);
       }
   }
}
```

c. Servlet that reads the value of a Session

```
import java.io.*;
import javax.servlet.http.*;
public class SessionReadServlet extends HttpServlet
{
      public void service(HttpServletRequest req,HttpServletResponse res)
      {
            try
            {
              HttpSession hs=req.getSession(true);
              PrintWriter pw=res.getWriter();
              pw.println("The Session value is : " + hs.getValue("hello"));
            }
            catch(Exception e)
            {
                  System.out.println(e);
            }
      }
}
```

6. Servlet program on Cookies

a. HTML program that calls a CookieServlet

```
<html>
<head><title>Program on Creating Cookies</title></head>
<body><center>
      <form method=post name=form1
                  action="http://localhost:8080/servlet/CookieServlet">
      <table>
            <tr>
                  <td>Enter Name :</td>
                  <td><input type=text name="name"></td>
            </tr>
            <tr>
                  <td>Enter Marks :</td>
                  <td><input type=text name="marks"></td>
            </tr>
            <tr>
                  <td><input type=submit value="Create Cookie"></td>
                  <td><input type=reset value="Reset"
                        onClick='document.form1.name.focus()'></td>
            </tr>
      </table></form>
</body>
</html>
```

b. Servlet that creates a Cookie

```
import java.io.*;
import javax.servlet.http.*;
public class CookieServlet extends HttpServlet
{
      public void service(HttpServletRequest req,HttpServletResponse res)
      {
            try
            {
             String s1=req.getParameter("name");
             String s2=req.getParameter("marks");
             Cookie c=new Cookie(s1,s2);
             res.addCookie(c);
             PrintWriter pw=res.getWriter();
             pw.println("<html><body><center><form action =
                    'http://localhost:8080/servlet/CookieReadServlet'>");
             pw.println("<h1>Cookie Created Successfully</h1>");
             pw.println("<input type=submit value=ReadCookies>");
             pw.println("<input type=button value=GoBack
                                    onClick='javascript:history.go(-1)'>");
             pw.println("</form></body></html>");
            }
            catch(Exception e)
            {
                  System.out.println(e);
            }
      }
}
```

c. Servlet that reads Cookies from the HTTP Request

```
import java.io.*;
import javax.servlet.http.*;

public class CookieReadServlet extends HttpServlet
{
 public void service(HttpServletRequest req,HttpServletResponse res)
 {
    try
   {
        PrintWriter pw=res.getWriter();
        Cookie c[]=req.getCookies();

        pw.println("<html><head><title>List of Cookies Created</title>
                                              </head><body><center>");
```

```
        pw.println("<table><tr><th width=150 align=left>Name</th>
                            <th width=100 align=left>Marks</th></tr>");
        for(int i=0;i<c.length;i++)
        pw.println("<tr><td nowrap>"+c[i].getName()+"</td><td nowrap>"
                                        +c[i].getValue()+"</td></tr>");
        pw.println("</table></body></html>");
    }
     catch(Exception e)
    {
          System.out.println(e);
    }
  }
}
```