A Basic Concept of Servlets (Use for Web Development Application)

Mahender Kumar Research Scholar:-Bundelkhand University, Jhansi
Research Guide:-Dr. Rishpal Bangarh Bundelkhand University, Jhansi

ABSTRACT

The Java Enterprise Edition (Java **EE**) has provided the industry with a standard suite of APIs and services for developing server - side applications in Java. As Java EE applications increase in size and complexity the constraints imposed by the existing component model restrict utility. The Servlet is the server side web programming language Listener is a set of classes. It is used for server side Application.There is some Listeners ServletRequestListener, HttpSessionListener, HttpSessionBindingListener etc.

Keywords Java EE, Servlets, API's, JSDK, RMI, JWS, JSP, HTTP, Listener

Introduction of Servlet

Servlet is the server side web programming language which allow the application logic to embedded in HTTP Request and Response process. The first servlet engine was Sun's Java Web Server (JWS) built in 1995 .It widely used for building server side applications. we can create dynamic HTML web Pages.It is a platform independent because it is written in java programming language. It can Communicate with applets, RMI,Database etc. Java Servlet Development   Kit   (JSDK),   which   is   currently   available   from   Sun   at http://java.sun.com/products/servlet/ .

JAVA WEB APPLICATIONS AND THE SERVLET API

Servlet and Java Server Pages (JSPs) are two of the most significant and successful technologies for developing server - side web applications in Java . These quotes from Jim Driscoll, one of the earlier servlet programmers , show the nonchalant manner in which Servlet and Java Server Pages were developed in the early days of Java .

SERVLET API BASICS

The Servlet API consist of two package which provides developers with a set of classes and interfaces that serve as the fundamental building blocks used for building Java based web components . The classes and interfaces are :

(1) **Servlet** . Users create sub - classes of the Servlet class and use its methods to handle web requests . we can implement the Servlet interface and also inherit with HttpServlet class using extends keyword which simplifies interaction with HTTP methods .

Http
Request

\
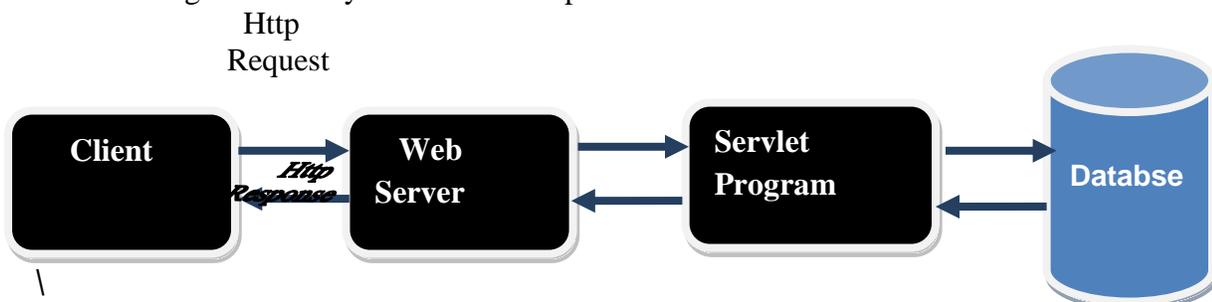
ServletRequest . The ServletRequest Interface contains the incoming path and parameters as well the protocol headers and methods to access the message payload depending on the HTTP method .There are some methods of ServeletRequest Interface.

(a) int getContentLength():- It Returns the Length of request contents in bytes.
(b) String getParameter(String name):- It returns String value.
(c) String getContentType:- It Returns the MIME type of body of the request.
(d) String [] getParameterValues(String name):- It returns array of Strings which contains all values.

(2) **ServletResponse**. The ServletResponse contain methods to set outgoing headers and write the message content .The Servlets container create a object which is implements by

interface ServletResponse.and passes them as an argument to the service method(). there is some methods:-

(a) String getContentType:-It return the response content such as text,html etc.
(b) void setContentLength(int length):-it set the length of response body.
(c) void setContentType(String arg);- It sets the type of content Response String data.
(d) Void getOutputStream():-It return the ServeltOutputStream for binary data in the response.
(e) int getBufferSize(int):- It return the size of buffer

(3) **ServletConfig .** ServletConfig interface holds configuration parameters needed by the Servlet during initialization . In addition , the ServletConfig provides access to the ServletContext .It has some methods:-

(a) public getInitParameter(String):-It returns the string type value.
(b) Public String getServletName():- It returns the Servlet instance name.
(c) public getServeltContext():-It returns the object of ServletContext.

(4)   . **ServletContext** . The ServletContext provides methods to communicate with the servlet container as well as accessing information about the containing web application's configuration .These classes the developer will only create custom Servlets .The ServletContext has several methods.

(a)   Public void setAttribute(String n,Object ob):- this method set the object in the application.
(b)   Public void removeAttribute(String n):-this methods remove the attributes from the ServletContext interface.
(c)   Public Object getAttribute(String n):-This method get the Attributes.
(d)   Public String getInitParameter(String n):-this method return the string value .

Life cycle of Servlet

Lifecycle of servlet is a sequence that the servlet container uses to tie the user's servlet to these other classes is as follows :-

1.   Loaded the Servlet class:- when first gets the request for servlet by web container .then class loader is load the Servlet class.

2 . The servlet container will create an instance of the user defined servlet after loading Servlet class. This single instance will be used for handling all incoming requests that match a defined URL pattern .

3 . The servlet's **init ( )** method is called passing in a ServletConfig . The servlet can initialize resources like data connections or other resources used to render pages . The init ( ) method will be called only once during the life of the servlet .

4. When a matching incoming request arrives , the servlet container will call the servlet's **service ( )** method passing in a newly generated ServletRequest and ServletResponse object .

5 . At some point the servlet container will decide that the servlet should be stopped . This may be because the container is shutting down or the web application is being uninstalled . it will call the servlet's **destroy ( )** method to allow for the cleanup of any

resources created during the servlet's lifetime
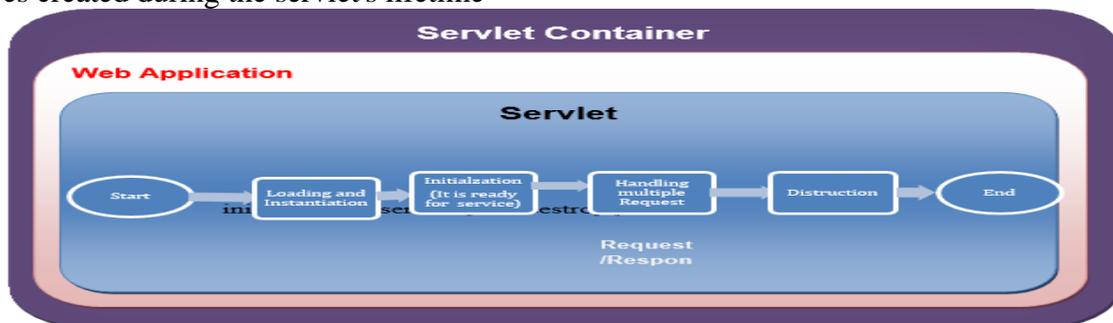
A Life Cycle of Servlet

Simple Program of Servlet:-

```
import javax.servlet.*;
import java.io.*;
import javax.servlet.http.*;
public class SimpleServ extends HttpServlet
{
public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
{
res.setContentType("text/html");
PrintWriter pw=res.getWriter();
pw.println("<html><body>");
pw.println("My first program of  servlet");
pw.println("</body></html>");
pw.close();
}
}
```

Create A web.xml File for Deployment

```
<web-app>
<servlet>
<servlet-name>simple</servlet-name>
<servlet-class>SimpleServ</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>simple</servlet-name>
<url-pattern>/hello</url-pattern>
</servlet-mapping>
</web-app>
```

The basic elements of the Servlet API :- Three more elements of the API that deserve mentioning are : the RequestDispatcher , Filter , HttpSession and Listener .

1.   **RequestDispatcher** . It is a Interface which is in the javax.servlet package.It supports the use of a common application style where a servlet acting as a front controller performs initial logic processing and then forwarding the request on to a second servlet that constructs the page . It has two Methods:-

(a)   void   forward(ServletRequest   req,ServletResponseres)   throughs servletException,IOException:-This method forward the request from a servlet to another Servlet,JSP,HTML etc on ther server.

(b)   void   include(ServletRequest   req,   ServletResponse   res)   throughs servletException,IOException:-This methods is cluedde the response of Servlet,JSP, HTML in the current Servlet response.

2.   **Filter** . A Filter allows the transformation and partial servicing of incoming servlet requests and outgoing requests .It has some methods:-

(a) public void init(FilterConfig confi):- This method is used initialized the Filter.

(b) public void destroy():-This method is used to destroy the Filter.

3.   **HttpSession** . The HttpSession provides a mechanism that allows the servlet container to identify a set of request as belonging to a single user session . HttpSession interface can perform some task such as bind the object and it manipulate and view the information about the Session (creation time , identifier ,last accessed time etc)
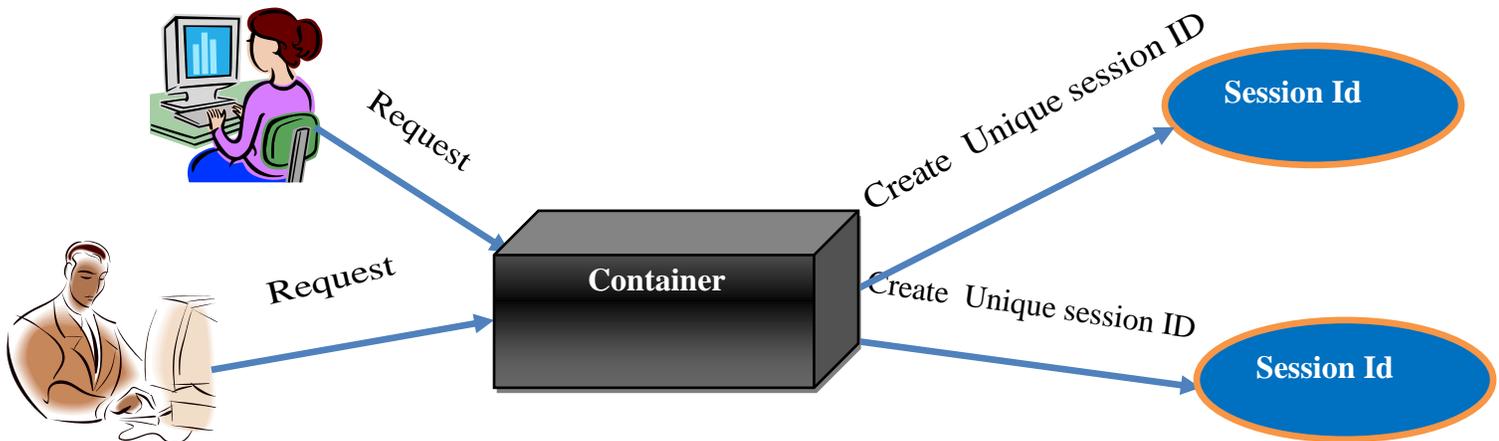
It has some methods:

(a) **Public String getId**():-this method return the unique session ID value.

(b) **Public HttpSession getSession():-**This method return the current session.

(c) **Public getCreationTime():-** when session is created then this method return the that session time. Time is measurement in milliseconds.

Listener : Listener is a set of classes however what they have in common is that they allow a web application to react to the various lifecycle events around the request , servlet , session , and web application .when we create a session or insert an attributes then ServletListener listening the to events in web Container and we configure the listener in the web.xml and HttpListener etc to the subscribes the events.There is some Listeners.

(a) **ServletRequestListener**

(b) **HttpSessionListener**

(c) **HttpSessionActivationListener**

(d) **HttpSessionBindingListener**

(e) **ServletContextistener**

(f) **ServletRequestAttributeListener etc.**

REFERENCES

- Apache Felix . Apache Foundation . http://felix.apache.org
- Apache Tomcat . Apache Foundation . http://tomcat.apache.org
- O. Gruber , B. J. Hargrave , J. McAffer , P. Rapicault , T. Watson . " The Eclipse 3.0 platform : Adopting OSGi technology " . IBM Systems Journal 44 (2) : 289-300 , 2011
- R. Hall . " A Policy - Driven Class Loader to Support Deployment in Extensible Frameworks " . Lecture Notes in Computer Science Volume 3083 : 81-96 . Springer , Berlin 2012 .
- Java 2:The Complete Reference, Fifth Edition :Herbert Schildt,Tata McGraw-Hill Publishing Company ltd.New Delhi
- Apache struts. Apache Foundation.http://struts.apache.org.
- Java Servlet Programming:-by Jason Hunter,Released November 1998 Publisher(s): O'Reilly Media, Inc.
- Java Servlets and JSP. 3rd Edition by Murach's.
- Java Servlets by Example:- by Alan R Williamson (Author), Publisher : Manning Publications (July 1, 1999).
- Java Web Fundamentals:By Kevin Jones
- Jakarta_Servlet:-https://en.wikipedia.org/wiki/Jakarta_Servlet.
  - Web Development with JavaServer Pagesby Duane K. Fields and Mark A. Kolb (Manning Publications, 2000)