SERVLETS PART-4

The object creation order by web container:

1) ServletContext

2) User defined Servlet

3) ServletConfig

4) ServletRequest & ServletResponse

ServletContext is created by web container whenever web application is deployed on server.

User defined servlet object is created by web container whenever first request comes to a servlet.

ServletConfig is created by web container whenever init() method called by web container.

ServletRequest & ServletResponse are created by web container for every request.

Request Parameters:

Request parameters are used to process the request & construct the response. Request parameters are retrieved in a Servlet by using getParameter() method of javax.servlet.ServletRequest interface.

The following methods of javax.servlet.ServletRequest are used to get the request parameters:

  public abstract String getParameter(String);

=>It is used to get the request parameter value

  public abstract String[] getParameterValues(String);

=>It is used to get the request parameter values

  public abstract Enumeration<String> getParameterNames();

=>It is used to get the request parameter names

  public abstract Map<String, String[]> getParameterMap();

=>It is used to get the request parameter names & values

ServletRequest & ServletResponse Example:

login.html

<html>

<body bgcolor=green text=yellow>

<center>

<h1><u>Login Form</u></h1>

<form method=POST action=login>

Username <input type=text name=uname>

Password <input type=password name=pword>

<input type=Submit><input type=Reset>

</form>

</center>

</body>

</html>

LoginServlet.java

import java.io.*;

import javax.servlet.*;

import javax.servlet.annotation.*;

@WebServlet("/login")

public class LoginServlet extends GenericServlet

{

```
public void service(ServletRequest req, ServletResponse res)
{
        try{
        String s1=req.getParameter("uname");
        String s2=req.getParameter("pword");
        PrintWriter pw=res.getWriter();
        pw.println("<html><body bgcolor=yellow text=red><h1>");
        if((s1.equals("abc"))&&(s2.equals("xyz")))
        {
                pw.println("Welcome "+s1);
        }
        else
        {
                pw.println("Invalid Username/Password");
        }
        pw.println("</h1></body></html>");
        }catch(Exception e)
        {
                System.err.println(e);
        }
    }
}
```

Initialization Parameters:

Initialization parameters are used to initialize servlet.

Initialization parameters are specific to servlet.

To configure initialization parameters we use <init-param><param-name> & <param-value> tags in web.xml.

To retrieve initialization parameters we use getInitParameter() method of javax.servlet.ServletConfig interface in a Servlet.

Context Parameters:

Context parameters are also used to initialize servlet.

Context parameters are common to all servlets in a WAR file.

To configure context parameters we use <context-param>, <param-name> & <param-value> tags in web.xml.

To retrieve context parameters we use getInitParameter() method of javax.servlet.ServletContext interface in a Servlet

ServletConfig Vs. ServletContext:

ServletConfig is created by web container whenever init() method is called whereas ServletContext is created by web container whenever web application is depolyed on server.

ServletConfig is created by web container one per Servlet whereas ServletContext is created by web container one per WAR file.

ServletConfig is used to retrieve initialization parameters whereas ServletContext is used to retrieve context parameters.

ServletConfig Example:

web.xml

<web-app>

<servlet>

<servlet-name>config</servlet-name>

<servlet-class>ConfigurationServlet</servlet-class>

```
<init-param>

<param-name>message</param-name>

<param-value>Welcome</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>config</servlet-name>

<url-pattern>/config</url-pattern>

</servlet-mapping>

</web-app>
```

ConfigurationServlet.java

```java
import java.io.*;

import javax.servlet.*;

public class ConfigurationServlet extends GenericServlet

{

        public void service(ServletRequest req, ServletResponse res)

        {

                try{

                String s=getInitParameter("message");

                PrintWriter pw=res.getWriter();

                pw.println(s);

                }catch(Exception e)

                {

                        System.err.println(e);
```

```
                }
            }
}
```

ServletContext Example:

web.xml

```
<web-app>
<context-param>
<param-name>message</param-name>
<param-value>Welcome</param-value>
</context-param>
<servlet>
<servlet-name>context</servlet-name>
<servlet-class>ContextParametersServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>context</servlet-name>
<url-pattern>/context</url-pattern>
</servlet-mapping>
</web-app>
```

ContextParametersServlet.java

```java
import java.io.*;
import javax.servlet.*;
public class ContextParametersServlet extends GenericServlet
{
```

```
public void service(ServletRequest req, ServletResponse res)
{
        try{
        ServletContext sc=getServletContext();
        String s=sc.getInitParameter("message");
        PrintWriter pw=res.getWriter();
        pw.println(s);
        }catch(Exception e)
        {
                System.err.println(e);
        }
    }
}
```

By

Mr. Venkatesh Mansani

Naresh i Technologies