

Dt : 7/6/2022

**imp*

ServletContext:

=>ServletContext is an interface from javax.servlet package and which is instantiated automatically when WebApplication is deployed(loaded) into Server.

=>This ServletContext object is loaded with Server information.

=>we use getServletContext() method to access the reference of ServletContext object.

=>This getServletContext() method is available from ServletRequest, GenericServlet, HttpSession and ServletConfig.

syntax:

```
ServletContext sct = req.getServletContext();
```

=>we use <context-param> tag part of web.xml to initialize parameters to ServletContext object.

syntax:

```
<web-app>
```

```
  <context-param>
```

```
    <param-name> name </param-name>
```

```
    <param-value> value </param-value>
```

```
  </context-param>
```

.

.

</web-app>

=>we use `getInitParameter()` method to access the initialized parameter from `ServletContext` and `ServletConfig` objects.

Note:

=>Total `WebApplication` will have only one `ServletContext` object and the information in `ServletContext` object can be used by all Servlet Programs of `WebApplication`.

***imp**

`ServletConfig`:

=>`ServletConfig` is an interface from `javax.servlet` package and which is instantiated automatically when Servlet program is loaded for execution.

=>Every Servlet program in `WebApplication` wil have its own `ServletConfig` object.

=>This `ServletConfig` object is loaded with Servlet name.

=>we use `getServletConfig()` method to access the reference of `ServletConfig` object.

=>This `getServletConfig()` method is available from `GenericServlet`.

syntax:

`ServletConfig sc = this.getServletConfig();`

=>we use `<init-param>` tag part of `<servlet>` to initialize parameters to `ServletConfig` object.

syntax:

```
<web-app>
```

```
.
```

```
<servlet>
```

```
.
```

```
<init-param>
```

```
<param-name> name </param-name>
```

```
<param-value> value </param-value>
```

```
</init-param>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
.
```

```
</servlet-mapping>
```

```
.
```

```
</web-app>
```

Ex Application:

input.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
```

```
<body>
  <form action="dis" method="post">
    Name:<input type="text" name="name"><br>
    <input type="submit" value="Display">
  </form>
</body>
</html>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <context-param>
    <param-name>a</param-name>
    <param-value>2000</param-value>
  </context-param>

  <servlet>
    <servlet-name>DisplayServlet</servlet-name>
    <servlet-class>test.DisplayServlet</servlet-class>
    <init-param>
      <param-name>b</param-name>
      <param-value>3000</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>DisplayServlet</servlet-name>
    <url-pattern>/dis</url-pattern>
  </servlet-mapping>

  <welcome-file-list>
    <welcome-file>input.html</welcome-file>
  </welcome-file-list>
</web-app>
```

DisplayServlet.java

```
package test;

import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

@SuppressWarnings("serial")
```

```

public class DisplayServlet extends HttpServlet{

    protected void doPost(HttpServletRequest req,HttpServletResponse res)

    throws ServletException,IOException{

        PrintWriter pw = res.getWriter();

        res.setContentType("text/html");

        pw.println("User : "+req.getParameter("name"));

        pw.println("<br>====ServletContext====");

        ServletContext sct = req.getServletContext();

        //Accessing Context object reference

        pw.println("<br>ServerInfo:"+sct.getServerInfo());

        pw.println("<br>ContextValue:"+sct.getInitParameter("a"));

        pw.println("<br>====ServletConfig====");

        ServletConfig sc = this.getServletConfig();

        //Accessing Config Object

        pw.println("<br>ServletName:"+sc.getServletName());

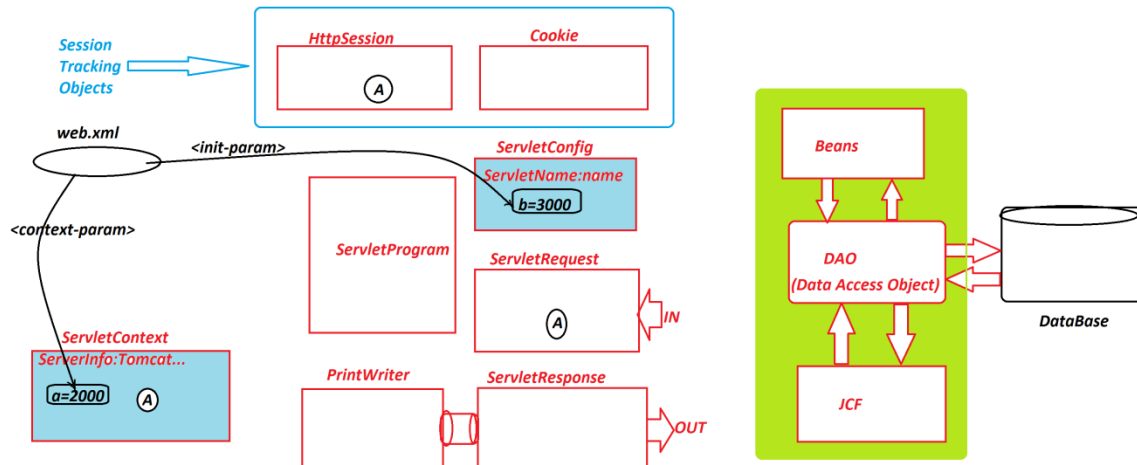
        pw.println("<br>ConfigValue:"+sc.getInitParameter("b"));

    }

}

```

Diagram of Object Layout:



Venkatesh Maipal