**11.ServletContext Interface:**

* An object of ServletContext is created by the web container at time of deploying the project.
* This object can be used to get configuration information from web.xml file.
* There is only one ServletContext object per web application.
* If any information is shared to many servlet, it is better to provide it from the web.xml file using the **<context-param>** element.

**Advantage of ServletContext:**

* **Easy to maintain** if any information is shared to all the servlet, it is better to make it available for all the servlet.
* We provide this information from the web.xml file, so if the information is changed, we don't need to modify the servlet.
* Thus it removes maintenance problem.

### Methods of ServletContext interface:

1. **public String getInitParameter(String name):**Returns the parameter value for the specified parameter name.
2. **public Enumeration getInitParameterNames():**Returns the names of the context's initialization parameters.
3. **public void setAttribute(String name,Object object):**sets the given object in the application scope.
4. **public Object getAttribute(String name):**Returns the attribute for the specified name.
5. **public Enumeration getInitParameterNames():**Returns the names of the context's initialization parameters as an Enumeration of String objects.
6. **public void removeAttribute(String name):**Removes the attribute with the given name from the servlet context.

### Syntax:

<web-app>

......

**<context-param>**

**<param-name>parametername</param-name>**

**<param-value>parametervalue</param-value>**

**</context-param>**

......

</web-app>

### Example:

**import** java.io.\*;

**import** javax.servlet.\*;

**import** javax.servlet.http.\*;

**public** **class** DemoServlet **extends** HttpServlet{

**public** **void** doGet(HttpServletRequest req,HttpServletResponse res)

**throws** ServletException,IOException

{

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

//creating ServletContext object

ServletContext context=getServletContext();

//Getting the value of the initialization parameter and printing it

String driverName=context.getInitParameter("dname");

pw.println("driver name is="+driverName);

pw.close();

}}

### **web.xml:**

<web-app>

<servlet>

<servlet-name> DemoServlet </servlet-name>

<servlet-**class**>DemoServlet</servlet-**class**>

</servlet>

<context-param>

<param-name>dname</param-name>

<param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>

</context-param>

<servlet-mapping>

<servlet-name> DemoServlet </servlet-name>

<url-pattern>/context</url-pattern>

</servlet-mapping>

</web-app>