

```
/*
 * Generated by the Jasper component of Apache Tomcat
 * Version: Apache Tomcat/8.0.27
 * Generated at: 2020-03-26 19:31:54 UTC
 * Note: The last modified time of this file was set to
 *       the last modified time of the source file after
 *       generation to assist with modification tracking.
 */
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import java.util.Date;

public final class DemoSession_jsp extends org.apache.jasper.
    runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent,
    org.apache.jasper.runtime.JspSourceImports {

    Integer accessCount;
    private static final javax.servlet.jsp.JspFactory
        _jspxFactory =
        javax.servlet.jsp.JspFactory.getDefaultFactory();

    private static java.util.Map<java.lang.String,java.lang.
        Long> _jspx_dependants;

    private static final java.util.Set<java.lang.String>
        _jspx_imports_packages;

    private static final java.util.Set<java.lang.String>
        _jspx_imports_classes;

    static {
        _jspx_imports_packages = new java.util.HashSet<>();
        _jspx_imports_packages.add("javax.servlet");
        _jspx_imports_packages.add("javax.servlet.http");
        _jspx_imports_packages.add("javax.servlet.jsp");
        _jspx_imports_classes = new java.util.HashSet<>();
        _jspx_imports_classes.add("java.util.Date");
    }

    private volatile javax.el.ExpressionFactory
        _el_expressionfactory;
    private volatile org.apache.tomcat.InstanceManager
        _jsp_instancemanager;

    public java.util.Map<java.lang.String,java.lang.Long>
        getDependants() {
```

```
        return _jspx_dependants;
    }

    public java.util.Set<java.lang.String> getPackageImports
        () {
        return _jspx_imports_packages;
    }

    public java.util.Set<java.lang.String> getClassImports()
    {
        return _jspx_imports_classes;
    }

    public javax.el.ExpressionFactory
        _jsp_getExpressionFactory() {
        if (_el_expressionfactory == null) {
            synchronized (this) {
                if (_el_expressionfactory == null) {
                    _el_expressionfactory = _jspxFactory.
                        getJspApplicationContext
                            (getServletConfig().
                                getServletContext()).
                                getExpressionFactory();
                }
            }
        }
        return _el_expressionfactory;
    }

    public org.apache.tomcat.InstanceManager
        _jsp_getInstanceManager() {
        if (_jsp_instancemanager == null) {
            synchronized (this) {
                if (_jsp_instancemanager == null) {
                    _jsp_instancemanager = org.apache.jasper
                        .runtime.InstanceManagerFactory.
                            getInstanceManager(getServletConfig
                                ());
                }
            }
        }
        return _jsp_instancemanager;
    }

    public void _jspInit() {
    }

    public void _jspDestroy() {
    }
```

```
}
```

```
public void _jspService(final javax.servlet.http.  
    HttpServletRequest request, final javax.servlet.http.  
    HttpServletResponse response)  
throws java.io.IOException, javax.servlet.  
    ServletException {  
  
    final java.lang.String _jspx_method = request.  
        getMethod();  
    if (!"GET".equals(_jspx_method) && !"POST".equals  
        (_jspx_method) && !"HEAD".equals(_jspx_method) &&  
        !javax.servlet.DispatcherType.ERROR.equals  
        (request.getDispatcherType())) {  
        response.sendError(HttpServletResponse.  
            SC_METHOD_NOT_ALLOWED, "JSPs only permit GET POST  
            or HEAD");  
        return;  
    }  
  
    final javax.servlet.jsp.PageContext pageContext;  
    javax.servlet.http.HttpSession session = null;  
    final javax.servlet.ServletContext application;  
    final javax.servlet.ServletConfig config;  
    javax.servlet.jsp.JspWriter out = null;  
    final java.lang.Object page = this;  
    javax.servlet.jsp.JspWriter _jspx_out = null;  
    javax.servlet.jsp.PageContext _jspx_page_context =  
        null;  
  
    try {  
        response.setContentType("text/html;charset=UTF-8");  
        pageContext = _jspxFactory.getPageContext(this,  
            request, response,  
                                                    null,  
                                                    true, 8192, true);  
        _jspx_page_context = pageContext;  
        application = pageContext.getServletContext();  
        config = pageContext.getServletConfig();  
        session = pageContext.getSession();  
        out = pageContext.getOut();  
        _jspx_out = out;  
  
        out.write("\n");  
        out.write("\n");  
        out.write("\n");  
        out.write("\n");  
    }  
}
```

```
out.write("<!DOCTYPE html>\n");
out.write("<html>\n");
out.write("    <head>\n");
out.write("        <meta http-equiv=\"Content-Type\"
            content=\"text/html; charset=UTF-8\">\n");
out.write("        <title>Session Test JSP</title>\n");
out.write("    </head>\n");
out.write("    <body>\n");
out.write("        ");
out.write("\n");
out.write("    ");

accessCount =
    (Integer)session.getAttribute("accessCount");
if (accessCount == null) {
    accessCount = 0;    // autobox int to Integer
} else {
    accessCount = new Integer(accessCount + 1);
}
session.setAttribute("accessCount", accessCount);

out.write("\n");
out.write("    Session is new? ");
out.println(session.isNew());
out.write("\n");
out.write("    <h2>You accessed this site \" ");
out.print( accessCount );
out.write(" times in this session.</h2>\n");
out.write("    <ul><li>Your session ID is \" ");
out.print( session.getId() );
out.write("</li>\n");
out.write("        <li>Session creation time is \n");
out.write("            ");
out.print( new Date(session.getCreationTime()) )
    ;
out.write(" </li>\n");
out.write("        <li>Session last access time is ");
out.print(
    new Date(session.getLastAccessedTime()
    ) );
out.write(" </li>\n");
out.write("        <li>Session max inactive interval is
            ");
out.print(
    session.getMaxInactiveInterval() );
out.write(" seconds</li>\n");
out.write("    </ul>\n");
out.write(" \n");
```

```

out.write("        <p><a href=''");
out.print( request.getRequestURI() );
out.write("'>Refresh</a>\n");
out.write("        <p><a href=''");
out.print( response.encodeURL(request.
    getRequestURI()));
out.write(">\n");
out.write("                Refresh with URL rewriting</
a>\n");
out.write("        <form method=\"GET\"
    action=\"endSession.jsp\">\n");
out.write("                <input type=\"submit\" value=\"End
    Session\">\n");
out.write("        </form>\n");
out.write("    </body>\n");
out.write("</html>\n");
} catch (java.lang.Throwable t) {
    if (!(t instanceof javax.servlet.jsp.
        SkipPageException)){
        out = _jspx_out;
        if (out != null && out.getBufferSize() != 0)
            try {
                if (response.isCommitted()) {
                    out.flush();
                } else {
                    out.clearBuffer();
                }
            } catch (java.io.IOException e) {}
        if (_jspx_page_context != null)
            _jspx_page_context.handlePageException(t
        );
        else throw new ServletException(t);
    }
} finally {
    _jspxFactory.releasePageContext
        (_jspx_page_context);
}
}
}

```