

Java Server Pages

*Εισαγωγή στα servlets για το εργαστήριο
Συστήματα Αξιολόγησης Διαδικτυακών
Εφαρμογών*

JSP εγκατάσταση του Apache Tomcat

- URL: <https://tomcat.apache.org/download-90.cgi>
- Αρχείο: zip ή το tar.gz
- Το συμπιεσμένο αρχείο αποσυμπιέζεται σε ένα directory και ο Tomcat ξεκινάει τρέχοντας το **startup.sh** ή το **startup.bat**, ανάλογα με το λειτουργικό σύστημα. Και τα δύο βρίσκονται στο directory **bin**.
- Οι ιστοσελίδες τοποθετούνται στο directory:
[C:\apache-tomcat-9.0.59\webapps\ROOT](#)
- Το παρακάτω HTML αρχείο:

```
<HTML>  
<BODY>  
Hello, world!  
</BODY>  
</HTML>
```

} **hello.html**

Εκτελείται από το URL: <http://localhost:8080/hello.jsp>

Εκτελώντας jsp

- Μετονομάζοντας το **hello.html** αρχείο σε **hello.jsp** εκτελείται από τον Tomcat κανονικά.
- Ο Tomcat δεν λειτουργεί όπως ο httpd, απλά στέλνοντας τα περιεχόμενα της ιστοσελίδας στον browser που την ζήτησε.
- Δημιουργεί αυτόματα μία Java κλάση, για την σελίδα και την κάνει compile.
- Η κλάση αυτή λέγεται **hello_jsp.java** και βρίσκεται στον φάκελο [C:\apache-tomcat-9.0.59\work\Catalina\localhost\ROOT\org\apache\jsp](#)
- Στον ίδιο φάκελο εμφανίζεται και το .class αρχείο που προκύπτει από το compilation.

Η κλάση

```
hello_jsp.java x
1 /*
2  * Generated by the Jasper component of Apache Tomcat
3  * Version: Apache Tomcat/8.0.20
4  * Generated at: 2015-03-18 19:29:21 UTC
5  * Note: The last modified time of this file was set to
6  *       the last modified time of the source file after
7  *       generation to assist with modification tracking.
8  */
9 package org.apache.jsp;
10
11 import javax.servlet.*;
12 import javax.servlet.http.*;
13 import javax.servlet.jsp.*;
14
15 public final class hello_jsp extends org.apache.jasper.runtime.HttpJspBase
16     implements org.apache.jasper.runtime.JspSourceDependent,
17                 org.apache.jasper.runtime.JspSourceImports {
18
19     private static final javax.servlet.jsp.JspFactory _jspxFactory =
20         javax.servlet.jsp.JspFactory.getDefaultFactory();
21
22     private static java.util.Map<java.lang.String,java.lang.Long> _jspx_dependants;
23
24     private static final java.util.Set<java.lang.String> _jspx_imports_packages;
25
26     private static final java.util.Set<java.lang.String> _jspx_imports_classes;
27
28     static {
29         _jspx_imports_packages = new java.util.HashSet<>();
30         _jspx_imports_packages.add("javax.servlet");
31         _jspx_imports_packages.add("javax.servlet.http");
32         _jspx_imports_packages.add("javax.servlet.jsp");
33         _jspx_imports_classes = null;
34     }
35
36     private javax.el.ExpressionFactory _el_expressionfactory;
37     private org.apache.tomcat.InstanceManager _jsp_instancemanager;
38
39     public java.util.Map<java.lang.String,java.lang.Long> getDependants() {
40         return _jspx_dependants;
41     }
42
43     public java.util.Set<java.lang.String> getPackageImports() {
44         return _jspx_imports_packages;
45     }
46
47     public java.util.Set<java.lang.String> getClassImports() {
48         return _jspx_imports_classes;
49     }
50 }
```

```

50
51 public void _jspInit() {
52     _el_expressionfactory = _jspxFactory.getJspApplicationContext(getServletConfig().getServletContext()).getExpressionFactory();
53     _jsp_instancemanager = org.apache.jasper.runtime.InstanceManagerFactory.getInstanceManager(getServletConfig());
54 }
55
56 public void _jspDestroy() {
57 }
58
59 public void _jspService(final javax.servlet.http.HttpServletRequest request, final javax.servlet.http.HttpServletResponse response)
60     throws java.io.IOException, javax.servlet.ServletException {
61
62     final java.lang.String _jspx_method = request.getMethod();
63     if (!"GET".equals(_jspx_method) && !"POST".equals(_jspx_method) && !"HEAD".equals(_jspx_method) && !javax.servlet.DispatcherType.ERROR.equals(request.getDispatcherType())) {
64         response.sendError(HttpServletResponse.SC_METHOD_NOT_ALLOWED, "JSPs only permit GET POST or HEAD");
65         return;
66     }
67
68     final javax.servlet.jsp.PageContext pageContext;
69     javax.servlet.http.HttpSession session = null;
70     final javax.servlet.ServletContext application;
71     final javax.servlet.ServletConfig config;
72     javax.servlet.jsp.JspWriter out = null;
73     final java.lang.Object page = this;
74     javax.servlet.jsp.JspWriter _jspx_out = null;
75     javax.servlet.jsp.PageContext _jspx_page_context = null;
76
77
78     try {
79         response.setContentType("text/html");
80         pageContext = _jspxFactory.getPageContext(this, request, response,
81             null, true, 8192, true);
82         _jspx_page_context = pageContext;
83         application = pageContext.getServletContext();
84         config = pageContext.getServletConfig();
85         session = pageContext.getSession();
86         out = pageContext.getOut();
87         _jspx_out = out;
88

```

```
88 |
89   out.write("<html>\n");
90   out.write("<head></head>\n");
91   out.write("<body>\n");
92   out.write("Hello World!\n");
93   out.write("</body>\n");
94   out.write("</html>");
95 } catch (java.lang.Throwable t) {
96   if (!(t instanceof javax.servlet.jsp.SkipPageException)){
97     out = _jspx_out;
98     if (out != null && out.getBufferSize() != 0)
99       try {
100         if (response.isCommitted()) {
101           out.flush();
102         } else {
103           out.clearBuffer();
104         }
105       } catch (java.io.IOException e) {}
106     if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
107     else throw new ServletException(t);
108   }
109 } finally {
110   _jspxFactory.releasePageContext(_jspx_page_context);
111 }
112 }
113 }
```

Προσθέτοντας Δυναμικό Περιεχόμενο

<HTML>

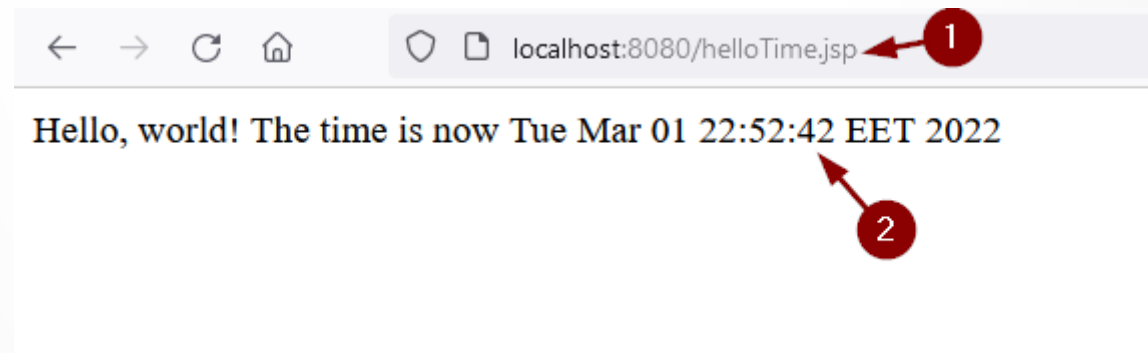
<BODY>

Hello World! The time is now

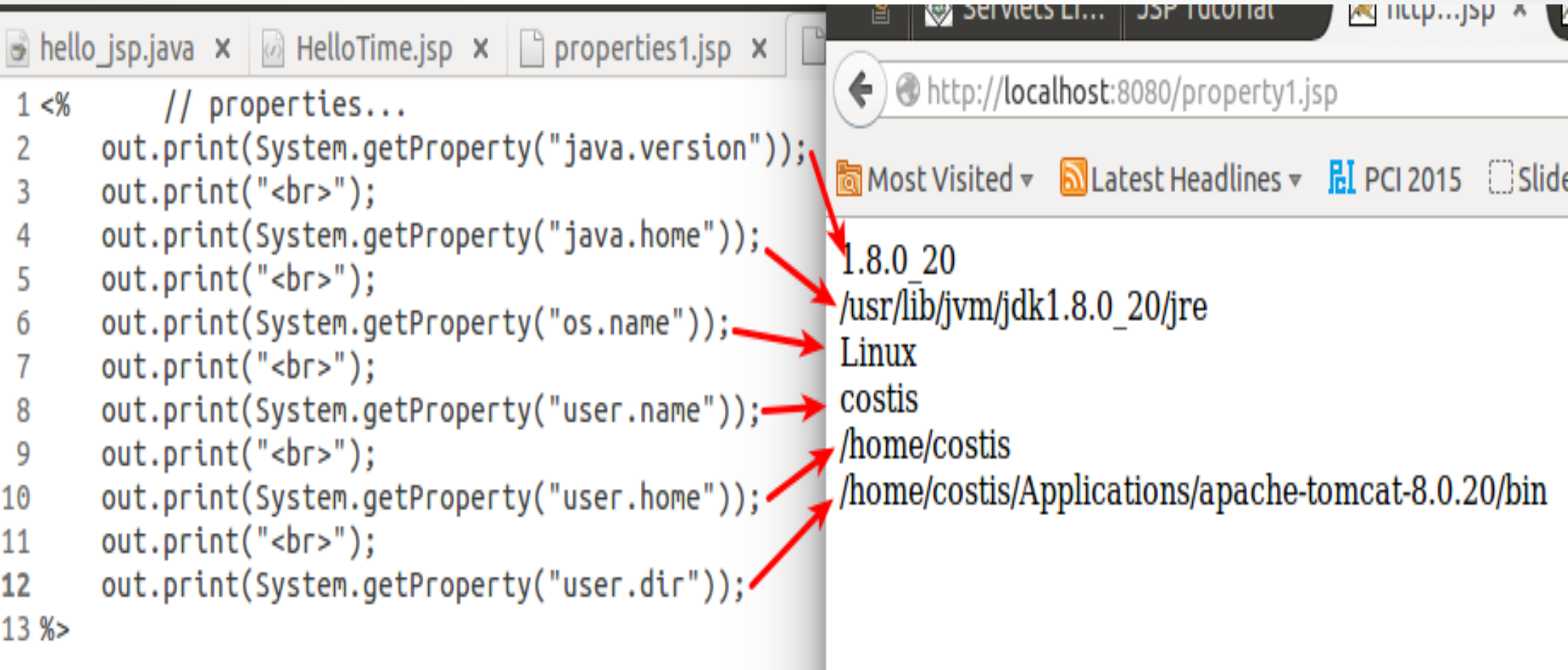
<%= new java.util.Date() %>

</BODY>

</HTML>



System.getProperty(...)



The image shows a web browser window displaying the output of a JSP page. The browser's address bar shows the URL `http://localhost:8080/property1.jsp`. The output is a list of system properties, each on a new line. Red arrows point from the corresponding lines of code in the code editor on the left to the output in the browser. The code editor shows a file named `properties1.jsp` with the following content:

```
1 <%      // properties...
2 out.print(System.getProperty("java.version"));
3 out.print("<br>");
4 out.print(System.getProperty("java.home"));
5 out.print("<br>");
6 out.print(System.getProperty("os.name"));
7 out.print("<br>");
8 out.print(System.getProperty("user.name"));
9 out.print("<br>");
10 out.print(System.getProperty("user.home"));
11 out.print("<br>");
12 out.print(System.getProperty("user.dir"));
13 %>
```

The browser output is as follows:

```
1.8.0_20
/usr/lib/jvm/jdk1.8.0_20/jre
Linux
costis
/home/costis
/home/costis/Applications/apache-tomcat-8.0.20/bin
```


conf/tomcat-users.xml

```
<!--
```

NOTE: The sample user and role entries below are wrapped in a comment and thus are ignored when reading this file. Do not forget to remove

<!-- ... --> that surrounds them.

```
-->
```

```
<!--
```

```
<role rolename="tomcat"/>
```

```
<role rolename="role1"/>
```

```
<user username="tomcat" password="tomcat" roles="tomcat"/>
```

```
<user username="both" password="tomcat" roles="tomcat,role1"/>
```

```
<user username="role1" password="tomcat" roles="role1"/>
```

```
-->
```

conf/tomcat-users.xml

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<user username="admin" password="admin"
roles="manager-gui,manager-script,manager-
jmx,manager-status,admin-gui,admin-script"/>
```

http://localhost:8080/

Apache Tomcat/8.0.20



If you're seeing this, you've successfully installed Tomcat. Congratulations!

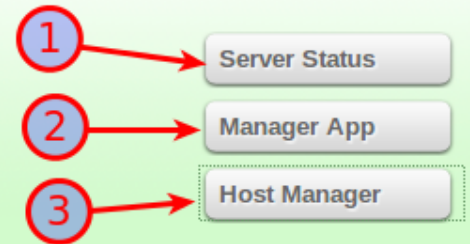


Recommended Reading:

[Security Considerations HOW-TO](#)

[Manager Application HOW-TO](#)

[Clustering/Session Replication HOW-TO](#)



Developer Quick Start

[Tomcat Setup](#)

[First Web Application](#)

[Realms & AAA](#)

[JDBC DataSources](#)

[Examples](#)

[Servlet Specifications](#)

[Tomcat Versions](#)

System.getProperty

```
<% // properties...
    out.print(System.getProperty("java.version"));
    out.print("<br>");
    out.print(System.getProperty("java.home"));
    out.print("<br>");
    out.print(System.getProperty("os.name"));
    out.print("<br>");
    out.print(System.getProperty("user.name"));
    out.print("<br>");
    out.print(System.getProperty("user.home"));
    out.print("<br>");
%>
```

System.getProperties

<%

```
java.util.Enumeration e = System.getProperties().propertyNames();  
while( e.hasMoreElements() ){  
    String prop = (String)e.nextElement();  
    out.print(prop);  
    out.print(" = ");  
    out.print( System.getProperty(prop) );  
    out.print("<br>");  
}
```

%>