

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- =====
4:     Configuration file for Lecture 7
5:     Computer Science E-259
6:
7:     Be sure to set the value of the Server element's port
8:     attribute to an integer between 1024 and 65535, inclusive.
9:     Of course, that port cannot already be in use. Nor can it
10:    be the same value you choose for the Connector element's port.
11:
12:    Also be sure to set the value of the Connector element's port
13:    attribute to an integer between 1024 and 65535, inclusive.
14:    Of course, that port cannot already be in use. Nor can it
15:    be the same value you choose for the Server element's port.
16: ===== -->
17:
18:
19: <!-- A "Server" is a singleton element that represents the entire JVM,
20:     which may contain one or more "Service" instances. The Server
21:     listens for a shutdown command on the indicated port.
22:
23:     Note: A "Server" is not itself a "Container", so you may not
24:     define subcomponents such as "Valves" or "Loggers" at this level.
25: -->
26:
27: <Server port="8000" shutdown="SHUTDOWN">
28:
29:     <!-- Global JNDI resources -->
30:     <GlobalNamingResources>
31:
32:         <!-- Editable user database that can also be used by
33:             UserDatabaseRealm to authenticate users -->
34:         <Resource name="UserDatabase" auth="Container"
35:             type="org.apache.catalina.UserDatabase"
36:             description="User database that can be updated and saved"
37:             factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
38:             pathname="conf/tomcat-users.xml" />
39:
40:     </GlobalNamingResources>
41:
42:     <!-- A "Service" is a collection of one or more "Connectors" that share
43:         a single "Container" (and therefore the web applications visible
44:         within that Container). Normally, that Container is an "Engine",
45:         but this is not required. -->
```

```
46:
47:  <!-- Define the Tomcat Stand-Alone Service -->
48:  <Service name="Catalina">
49:
50:    <!-- A "Connector" represents an endpoint by which requests are received
51:         and responses are returned.  Each Connector passes requests on to the
52:         associated "Container" (normally an Engine) for processing. -->
53:
54:    <!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
55:    <Connector port="8080"
56:              maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
57:              enableLookups="false" acceptCount="100"
58:              connectionTimeout="20000" disableUploadTimeout="true" />
59:
60:    <!-- An Engine represents the entry point (within Catalina) that processes
61:         every request.  The Engine implementation for Tomcat stand alone
62:         analyzes the HTTP headers included with the request, and passes them
63:         on to the appropriate Host (virtual host). -->
64:
65:    <!-- Define the top level container in our container hierarchy -->
66:    <Engine name="Catalina" defaultHost="localhost">
67:
68:      <!-- Because this Realm is here, an instance will be shared globally -->
69:
70:      <!-- This Realm uses the UserDatabase configured in the global JNDI
71:           resources under the key "UserDatabase".  Any edits
72:           that are performed against this UserDatabase are immediately
73:           available for use by the Realm. -->
74:      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
75:            resourceName="UserDatabase"/>
76:
77:      <!-- Define the default virtual host -->
78:      <Host name="localhost" appBase="webapps"
79:            unpackWARs="true" autoDeploy="true"
80:            xmlValidation="true" xmlNamespaceAware="true">
81:
82:        <!-- Root Context -->
83:        <Context path="" docBase="ROOT" debug="0" reloadable="true"/>
84:
85:      </Host>
86:
87:    </Engine>
88:
89:  </Service>
90:
```

91: </Server>

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
4:
5: <!-- direct Xalan's environment to browser instead of console -->
6: <% java.io.ByteArrayOutputStream os = new java.io.ByteArrayOutputStream(); %>
7: <% java.io.PrintWriter pw = new java.io.PrintWriter(os); %>
8: <% (new org.apache.xalan.xslt.EnvironmentCheck()).checkEnvironment(pw); %>
9:
10: <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
11:   <head>
12:     <title>Environment Happiness Page</title>
13:   </head>
14:   <body>
15:     <pre>
16: <% out.print(os); %>
17:     </pre>
18:   </body>
19: </html>
```

```
1: <?xml version="1.0" encoding="iso-8859-1"?>
2: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
3:
4: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
5:   <head>
6:     <title><% out.print("hello, world"); %></title>
7:   </head>
8:   <body>
9:     <% out.print("hello, world"); %>
10:   </body>
11: </html>
```

```
1: <?xml version="1.0"?>
2: <jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
3:   version="1.2">
4:
5:   <jsp:directive.page contentType="text/html"/>
6:
7:   <![CDATA[<?xml version="1.0" encoding="iso-8859-1"?>]]>
8:   <![CDATA[<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-t
ransitional.dtd">]]>
9:
10:   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
11:     <head>
12:       <title>
13:         <jsp:scriptlet>
14:           out.print("hello, world");
15:         </jsp:scriptlet>
16:       </title>
17:     </head>
18:     <body>
19:       <jsp:scriptlet>
20:         out.print("hello, world");
21:       </jsp:scriptlet>
22:     </body>
23:   </html>
24:
25: </jsp:root>
```

```
1: <?xml version="1.0"?>
2:
3: <!-- excerpted from jakarta-tomcat-5.0.19/webapps/jsp-examples/xml/ -->
4:
5: <jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
6:   version="1.2">
7: <jsp:directive.page contentType="text/html"/>
8: <jsp:directive.page import="java.util.Date, java.util.Locale"/>
9: <jsp:directive.page import="java.text.*"/>
10:
11: <jsp:declaration>
12:   String getDateTmeStr(Locale l) {
13:     DateFormat df = SimpleDateFormat.getDateTmeInstance(DateFormat.MEDIUM, DateFormat.MEDIUM, l);
14:     return df.format(new Date());
15:   }
16: </jsp:declaration>
17:
18: <html>
19: <head>
20:   <title>Example JSP in XML format</title>
21: </head>
22:
23: <body>
24: This is the output of a simple JSP using XML format.
25: <br />
26:
27: <div>Use a jsp:scriptlet to loop from 1 to 10: </div>
28: <jsp:scriptlet>
29: // Note we need to declare CDATA because we don't escape the less than symbol
30: <![CDATA[
31:   for (int i = 1; i<=10; i++) {
32:     out.println(i);
33:     if (i < 10) {
34:       out.println(", ");
35:     }
36:   }
37: ]]>
38: </jsp:scriptlet>
39:
40: <!-- Because I omit br's end tag, declare it as CDATA -->
41: <![CDATA[
42:   <br><br>
43: ]]>
44:
45: <div align="left">
```

```
46:    Use a jsp:expression to write the date and time in the browser's locale:
47:    <jsp:expression>getDateTimeStr(request.getLocale())</jsp:expression>
48: </div>
49:
50:
51: <jsp:text>
52:     &lt;p>This sentence is enclosed in a jsp:text element.&lt;/p>
53: </jsp:text>
54:
55: </body>
56: </html>
57: </jsp:root>
```



```
1: <?xml version="1.0" encoding="ISO-8859-1"?>
2:
3: <!DOCTYPE web-app
4:     PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
5:     "http://java.sun.com/dtd/web-app_2_3.dtd">
6:
7: <!-- =====
8:     Configuration file for Lecture 7
9:     Computer Science E-259
10:  ===== -->
11:
12: <web-app>
13:     <servlet>
14:         <servlet-name>hello</servlet-name>
15:         <servlet-class>Hello</servlet-class>
16:     </servlet>
17:     <servlet>
18:         <servlet-name>parameters</servlet-name>
19:         <servlet-class>Parameters</servlet-class>
20:     </servlet>
21:     <servlet>
22:         <servlet-name>visits</servlet-name>
23:         <servlet-class>Visits</servlet-class>
24:     </servlet>
25: </web-app>
```

```
1: <?xml version="1.0"?>
2:
3: <!-- excerpted from jakarta-tomcat-5.0.19/webapps/jsp-examples/xml/ -->
4:
5: <jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
6:   version="1.2">
7: <jsp:directive.page contentType="text/html"/>
8: <jsp:directive.page import="java.util.Date, java.util.Locale"/>
9: <jsp:directive.page import="java.text.*"/>
10:
11: <jsp:declaration>
12:   String getDateTmeStr(Locale l) {
13:     DateFormat df = SimpleDateFormat.getDateTmeInstance(DateFormat.MEDIUM, DateFormat.MEDIUM, l);
14:     return df.format(new Date());
15:   }
16: </jsp:declaration>
17:
18: <html>
19: <head>
20:   <title>Example JSP in XML format</title>
21: </head>
22:
23: <body>
24: This is the output of a simple JSP using XML format.
25: <br />
26:
27: <div>Use a jsp:scriptlet to loop from 1 to 10: </div>
28: <jsp:scriptlet>
29: // Note we need to declare CDATA because we don't escape the less than symbol
30: <![CDATA[
31:   for (int i = 1; i<=10; i++) {
32:     out.println(i);
33:     if (i < 10) {
34:       out.println(", ");
35:     }
36:   }
37: ]]>
38: </jsp:scriptlet>
39:
40: <!-- Because I omit br's end tag, declare it as CDATA -->
41: <![CDATA[
42:   <br><br>
43: ]]>
44:
45: <div align="left">
```

```
46:    Use a jsp:expression to write the date and time in the browser's locale:
47:    <jsp:expression>getDateTimeStr(request.getLocale())</jsp:expression>
48: </div>
49:
50:
51: <jsp:text>
52:     &lt;p>This sentence is enclosed in a jsp:text element.&lt;/p>
53: </jsp:text>
54:
55: </body>
56: </html>
57: </jsp:root>
```