

Aplicatii Web cu Servleti si pagini JSP

Se consideră că există un folder cu distributia apache-tomcat; vom numi acest director “**tomcat6**” (în documentatia Apache este valoarea variabilei de mediu CATALINA_HOME). Aceasta distributie poate fi obtinuta gratuit de la <http://tomcat.apache.org/download-60.cgi>.

1. Utilizarea unui servlet simplu

Creati un director (folder) pentru aplicatie cu numele “hello1”.

In acest director creati un fisier sursa numit “HelloServlet.java” cu urmatorul continut:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

Compilati fisierul “HelloServlet.java” cu biblioteca “servlet-api.jar” din “tomcat6\lib”. Intr-un IDE trebuie adaugata biblioteca respectiva la căile unde se caută bibliotecile de clase. In linie de comandă se compilează astfel (**tomcat6** se inlocuieste cu calea corespunzatoare instalarii):

```
javac -cp c:\tomcat6\lib\servlet-api.jar HelloServlet.java
```

Creati subdirectorul “WEB-INF” in directorul “hello1”.

Creati subdirectorul hello1/WEB-INF/classes si copiat (sau mutati) în el fisierul “HelloServlet.class”.

In subdirectorul hello1/WEB-INF creati fisierul “web.xml” cu urmatorul continut:

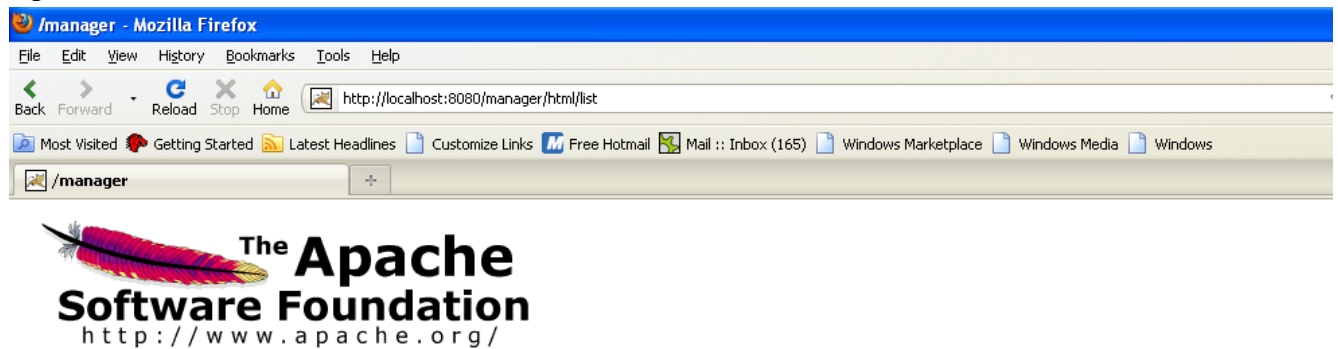
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" version="2.4">
    <servlet>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>HelloServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
</web-app>
```

Copiat directorul “hello1” in directorul tomcat6\webapps (“deployment”).

Intr-un browser Web introduceti urmatorul URL:

http://localhost:8080/hello1/hello

Aplicatia este vizibila si in consola de administrare Tomcat:



Tomcat Web Application Manager

Message: OK

Manager

[List Applications](#) [HTML Manager Help](#) [Manager Help](#)

Applications				
Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minu
/docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minu
/examples	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minu
/hello1		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minu

Modificati textul afisat in “Salut lume !” (in clasa servlet) si refaceti toate operatiile necesare pentru ca sa se afiseze acest mesaj la client (cu acelasi URL).

2. Extragere parametri dintr-o cerere

Creati un director (folder) pentru aplicatie cu numele “hello2”.

In acest director creati un fisier sursa numit “ParamServlet.java” cu urmatorul continut:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ParamServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
        String first,last;
        first=request.getParameter("first");
        last=request.getParameter("last");
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");
    }
}
```

```

    if (first != null || last != null) {
        out.println("<html>");
        out.println("<body>");
        out.print("<hr><h1>");
        out.print("Hello " + first + " " + last + "!");
        out.println("</h1><hr>");
        out.println("</body>");
        out.println("</html>");
    } else
        out.println("No Parameters, Please enter some");
}
}

```

Compilati fisierul “ParamServlet.java” cu biblioteca “servlet-api.jar” din “**tomcat6**\lib”.

Creati subdirectorul “WEB-INF” in directorul “hello2”.

Creati subdirectorul hello2/WEB-INF/classes si copiat (sau mutati) în el fisierul “ParamServlet.class”.

In subdirectorul hello2/WEB-INF copiat fisierul “web.xml” din directorul “hello1” si modificati numele de servlet si de clasa in “ParamServlet” (in loc de “HelloServlet”).

Copiat directorul “hello2” in directorul tomcat6\webapps.

Intr-un browser Web introduceti urmatorul URL:

```
http://localhost:8080/hello2/hello?first=good&last=student
```

3. Servlet cu afisare formular si preluare parametri din formular

Creati un director (folder) pentru aplicatie cu numele “hello3”.

In acest director creati un fisier sursa numit “FormServlet.java” cu urmatorul continut:

```

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FormServlet extends HttpServlet {
    String first,last;
    public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Request Parameters Example</title>");
        out.println("</head>");
        out.println("<body>");
        out.print("<form method=\"get\">");
        out.println("<p>");
        out.println("First Name:");
        out.println("<input type=text size=20 name=firstname>");
        out.println("<br>");
        out.println("Last Name:");
        out.println("<input type=text size=20 name=lastname>");
        out.println("<br>");
        out.println("<input type=submit value=\"Submit\">");
        out.println("</form>");
    }
}

```



```

package beans;
public class NameHandler {
    private String username;
    public NameHandler() { username = null; }
    public void setUsername( String name ) { username = name; }
    public String getUsername() { return username; }
}

```

Compilati fisierul java folosind un IDE sau prin comanda:

```
javac -d . NameHandler.java
```

(Se va crea un subdirector “beans” cu un fisier “NameHandler.class”).

Mutati directorul “beans” in hello5/WEB-INF/classes.

Copiatu directorul “hello5” in subdirectorul “tomcat6/webapps”.

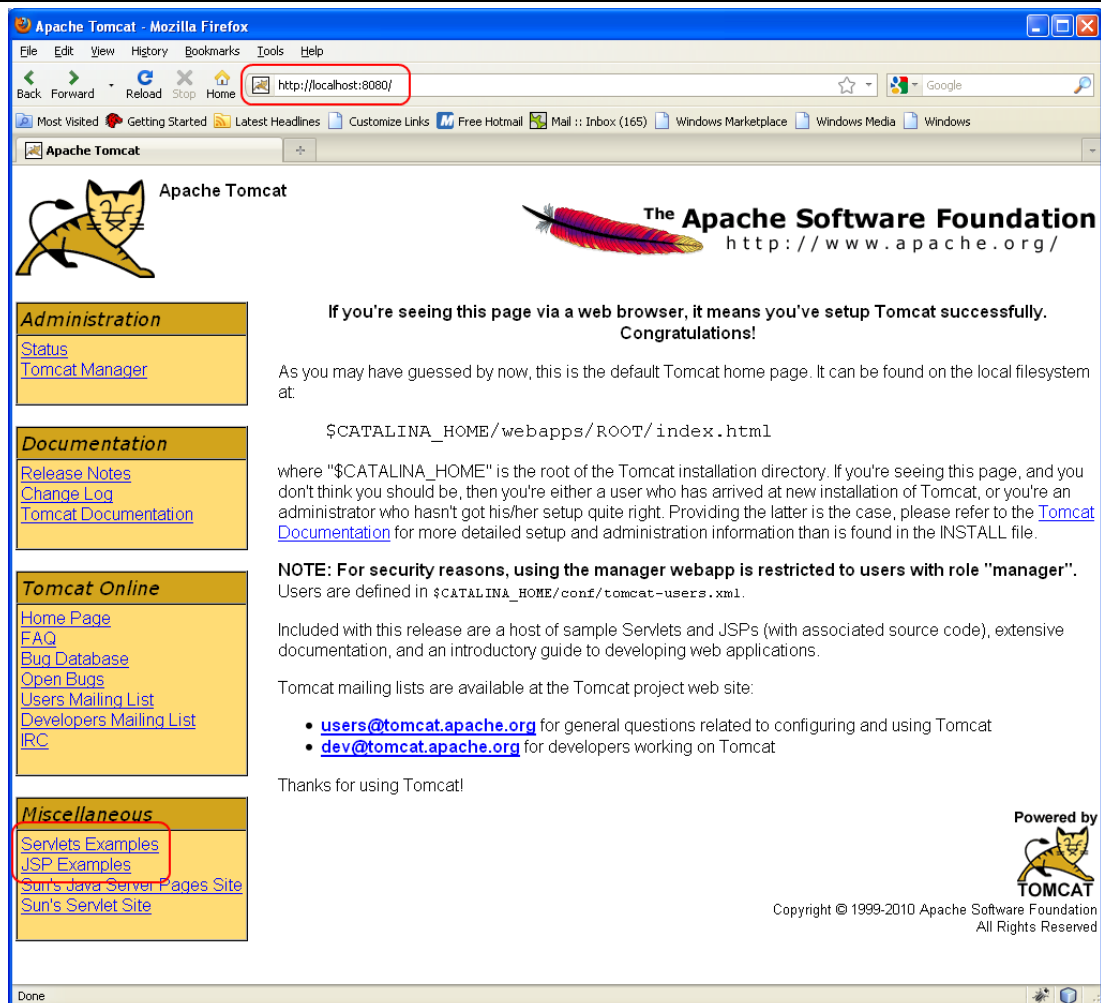
Intr-un browser Web introduceti urmatorul URL:

```
http://localhost:8080/hello5
```

6. Exemple

Cu serverul Tomcat pornit, introduceti intr-un browser urmatorul URI:

```
http://localhost:8080/examples
```



Executati exemplele si examinati sursele acestor exemple (servleti si pagini JSP).

Folosind ca model operatiile prezentate anterior creati o aplicatie Web cu un servlet din exemple (de exemplu “sessions”) si apoi o aplicatie cu o pagina JSP din exemple (de exemplu “Calendar” din JSP 1.2 Examples).