endava

UNIVERSITATEA DIN BUCUREȘTI

VIRTUTE ET SAPIENTIA

WEB TECHNOLOGIES USING JAVA

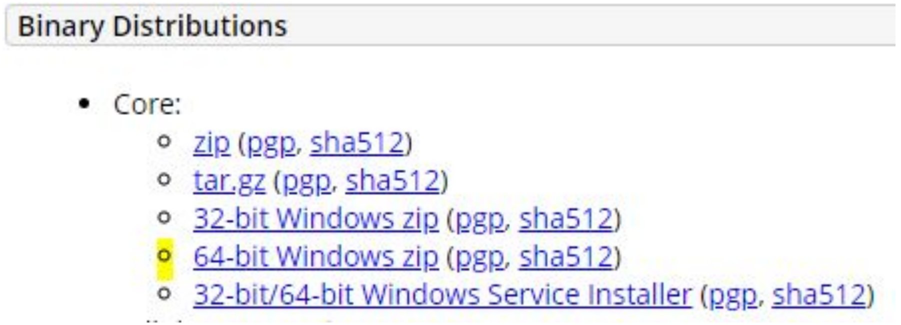
Laboratory 1 - 09.10.2020

AGENDA

- ➢ Application Server Setup - Tomcat
- ➢ JAVA EE Application Setup
- ➢ Servlet API
- ➢ JSP (Java Server Pages)
- ➢ Filters
- ➢ Package and Deploy Java web applications
- ➢ Homework

Application Server - Setup (I)

- Download Tomcat from https://tomcat.apache.org/ - Tomcat 9.0 zip file
- Unzip and add the folder in a location of your preference

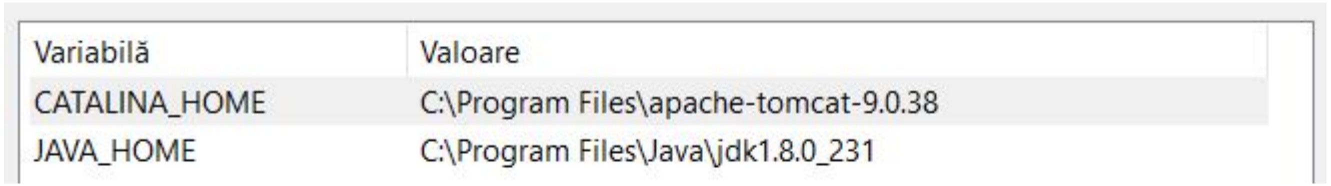


Binary Distributions

- Core:
  - zip (pgp, sha512)
  - tar.gz (pgp, sha512)
  - 32-bit Windows zip (pgp, sha512)
  - 64-bit Windows zip (pgp, sha512)
  - 32-bit/64-bit Windows Service Installer (pgp, sha512)

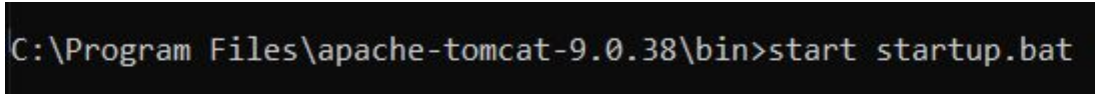
Application Server - Setup (II)

- Set up environment variables: JAVA_HOME and CATALINA_HOME



Variabilă	Valoare
CATALINA_HOME	C:\Program Files\apache-tomcat-9.0.38
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_231

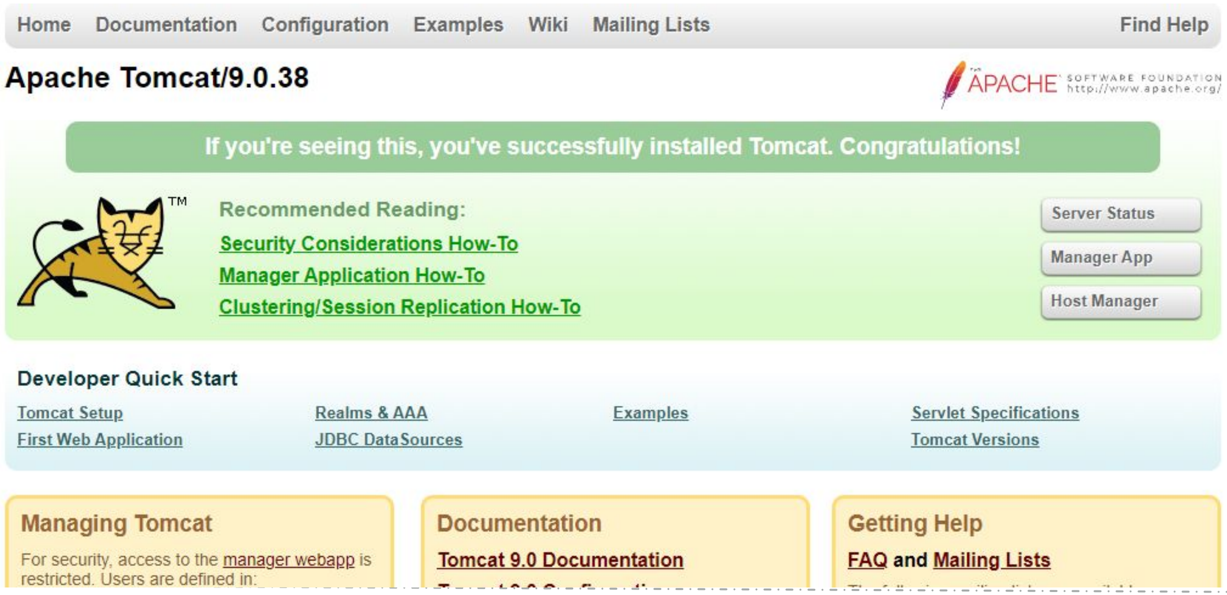
- Start the application server:



```
C:\Program Files\apache-tomcat-9.0.38\bin>start startup.bat
```

Application Server - Setup (III)

➢ Navigate to http://localhost:8080/:



Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/9.0.38

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

Server Status

Manager App

Host Manager

Developer Quick Start

Tomcat Setup

First Web Application

Realms & AAA

JDBC DataSources

Examples

Servlet Specifications

Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

Documentation

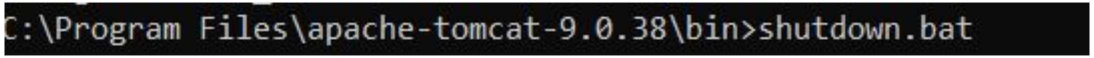
Tomcat 9.0 Documentation

Getting Help

FAQ and Mailing Lists

Application Server - Setup (IV)

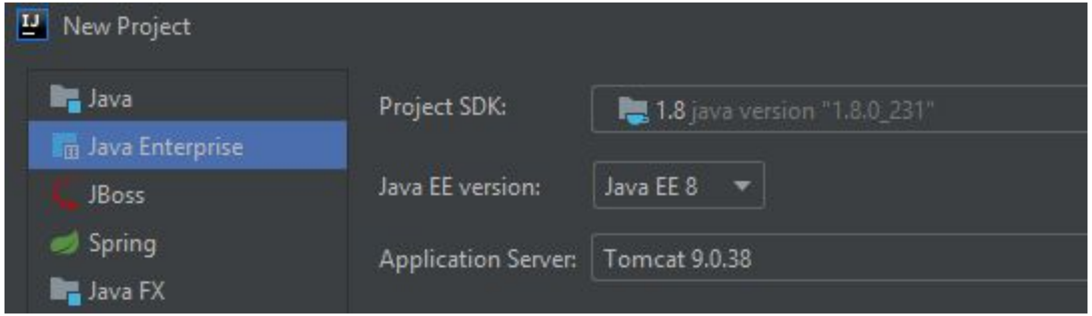
- Shutdown the application server:



```
C:\Program Files\apache-tomcat-9.0.38\bin>shutdown.bat
```

Java EE Application - Setup

- In your IDE create a project:
  - Java Enterprise
  - Set up the Application Server as the downloaded tomcat web server.



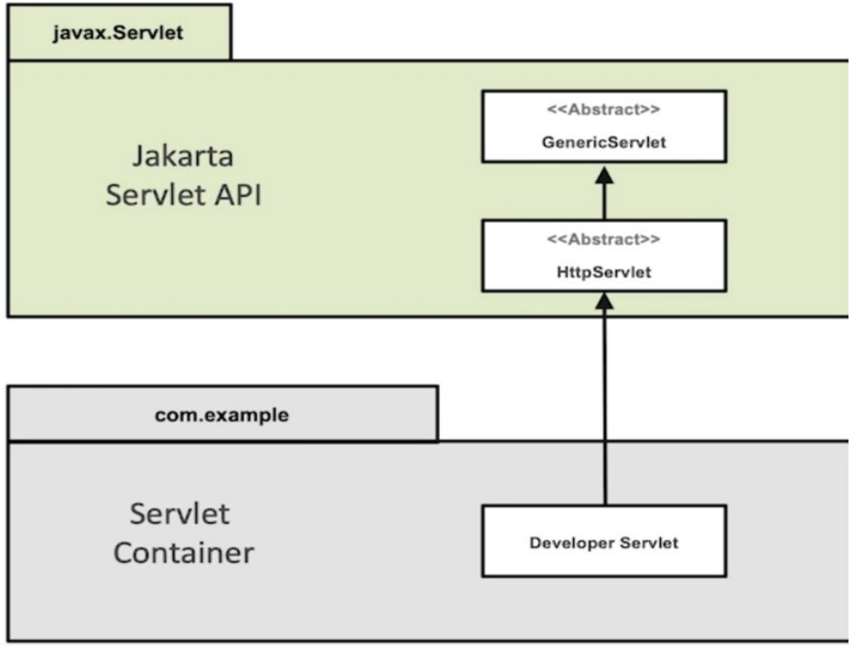
Project SDK: 1.8 java version "1.8.0_231"

Java EE version: Java EE 8

Application Server: Tomcat 9.0.38

Servlet API (I)

- JAVA class
- extends HttpServlet class



Servlet API (II) - Demo Servlet

```java
// tell the application server the HTTP path to the servlet
@WebServlet("/hello")
public class ServletHelloWorld extends HttpServlet {

   // in order to process HTTP get request -> need to override the doGet method
   @Override
   protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
       // HttpServletResponse object has a method that provides an IO print writer
       // we can use that print writer to output text
       resp.getWriter().println("Hello everybody!");
   }
}
```

Servlet API (III) - Demo Servlet (HTTP Parameters)

➢ http://localhost:8080/lab1/params?name=silvia

```
@WebServlet("/params")
public class ParamsServlet extends HttpServlet {

   @Override
   protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
       String name = req.getParameter("name");
       resp.getWriter().println(String.format("Hello %s!", name));
   }
}
```

Servlet API (IV) - Demo Servlet (HTTP Headers)

```
@WebServlet("/headers")
public class HeadersServlet extends HttpServlet {

   @Override
   protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
       String name = req.getParameter("name");
       -----------
       resp.setHeader("Content-Language", languageTag);
       resp.getWriter().println(String.format("%s %s!", greeting, name));
   }
}
```

Servlet API (IV) - Demo Servlet (ServletContext)

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
   ServletContext context = req.getServletContext();
   int count = 1;
   Object counter = context.getAttribute("counter");
   if (counter != null) {
       AtomicInteger existingCount = (AtomicInteger) counter;
       count = existingCount.incrementAndGet();
   } else {
       AtomicInteger newCount = new AtomicInteger(1);
       context.setAttribute("counter", newCount);
   }
   resp.getWriter().println("The count is: " + count);
}
```

Servlet API (V) - Demo Servlet (HttpSession)

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
   HttpSession session = req.getSession();
   int count = 1;
   Object counter = session.getAttribute("counter");
   if (counter != null) {
       AtomicInteger existingCount = (AtomicInteger) counter;
       count = existingCount.incrementAndGet();
   } else {
       AtomicInteger newCount = new AtomicInteger(1);
       session.setAttribute("counter", newCount);
   }
   resp.getWriter().println("The count is: " + count);
}
```

JSP (I) - JSP Expressions

➢ Templating language - provides markup for generating dynamic web pages

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
   <head>
       <title>Another Hello Example</title>
   </head>
   <body>
       <h1>Hello <%= request.getParameter("name") %></h1>
   </body>
</html>
```

JSP (II) - JSP Expressions

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
   <head>
       <title>Another Hello Example</title>
   </head>
   <body>
       <h1>Hello ${param.name} </h1>
   </body>
</html>
```

JSP (III) - JSP Scriplets

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
   <head>
       <title>Another Hello Example</title>
   </head>
   <body>
   <%@page import="java.util.Locale" %>
   <%
       String name = request.getParameter("name");
       ------------------------------------------
       response.setHeader("Content-Language", languageTag);
       out.print(String.format("%s %s!", greeting, name));
   %>
   </body>
</html>
```

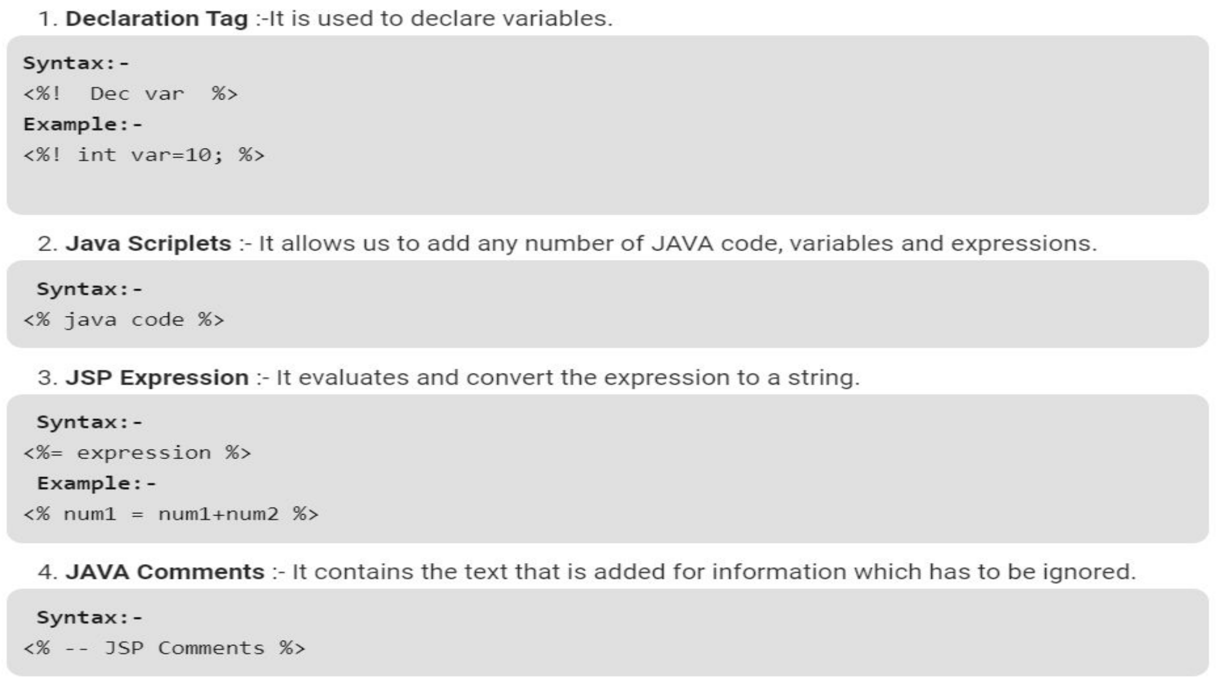
JSP (IV) - Java Beans

- A been must have a no-args constructor
- A bean may have state in the form of properties
- Properties must be accessible via getters and setters
- A bean’s properties must be serializable
- JSP was design to work with Java Beans components

JSP (IV) - Java Beans

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ page import="webprogramming.lab1.ex6.HelloHelper" %>
<%--<%@ =  directive--%>
<html>
   <head>
       <title>Beans Demo Example</title>
   </head>
   <body>
       <h1>
           <jsp:useBean id="helper" class="webprogramming.lab1.ex6.HelloHelper" scope="application" />
           <%= helper.getGreeting(request.getLocale()) %> ${param.name}
       </h1>
   </body>
</html>
```

JSP (IV) - Cheat Sheet



1. **Declaration Tag** :-It is used to declare variables.

```
Syntax:-
<%!  Dec var  %>
Example:-
<%! int var=10; %>
```

2. **Java Scriplets** :- It allows us to add any number of JAVA code, variables and expressions.

```
 Syntax:-
<% java code %>
```

3. **JSP Expression** :- It evaluates and convert the expression to a string.

```
 Syntax:-
<%= expression %>
 Example:-
<% num1 = num1+num2 %>
```

4. **JAVA Comments** :- It contains the text that is added for information which has to be ignored.

```
 Syntax:-
<% -- JSP Comments %>
```

Filters (I)

➢ Filters are interceptors - they modify behaviour

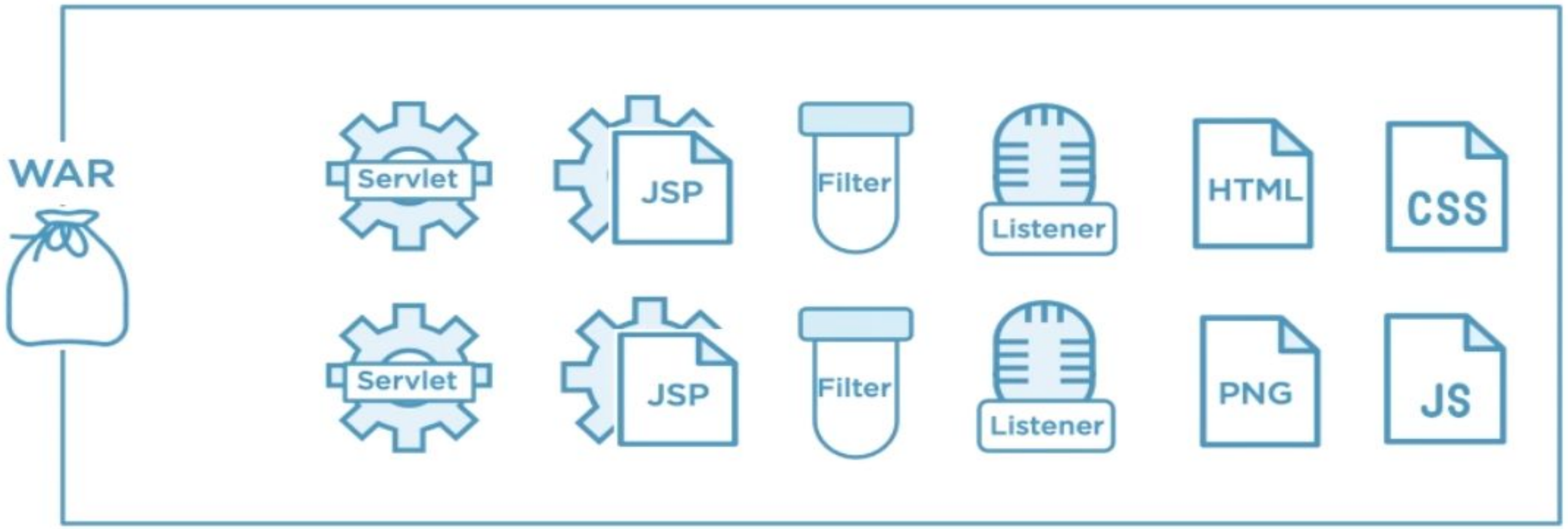
```
public class FilterDemo extends HttpFilter {

   @Override
   public void doFilter(ServletRequest request, ServletResponse response, FilterChain 
chain) throws IOException, ServletException {
       // code to execute before servlet

       chain.doFilter(request, response);

       // code to execute after servlet
   }
}
```

Package and Deploy Java web applications (I)



Package and Deploy Java web applications (II)

- Create war file: `jar cfv lab1.war index.jsp WEB-INF`
- Move created war file to: apache-tomcat-9.0.38\webapps
- Start tomcat and test app was successfully deployed:
    - http://localhost:8080/lab1/hello

Homework

- Create a web application using Servlets and JSPs that contains:
  - Login page with input forms for username and password
  - Home page containing a greeting of the logged in user

THANK YOU

Silvia Butan