Java EE:

JavaServer Pages

SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Contents

Questions

Software University

sli.do

#java-web

JavaServer Pages

Overview

What are JavaServer Pages?(1)

- Java Server Pages (JSP)
  - Server-side programming technology
  - Dynamic
  - Platform-independent
  - Method for building Web-based applications
- JSP have access to the entire family of Java APIs

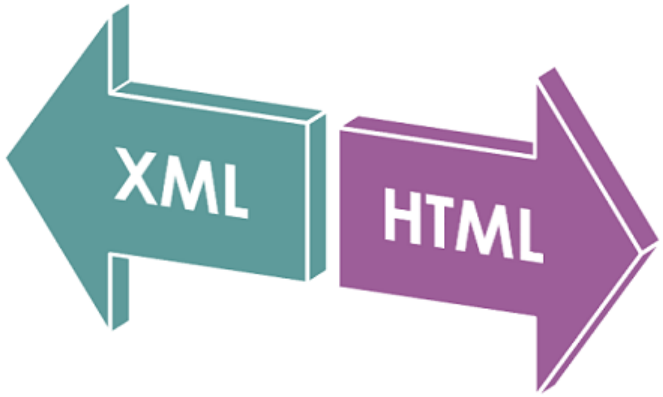
What are JavaServer Pages?(2)

- JavaServer Pages (JSP):
  - Technology for developing Webpages that support:
    - Dynamic content.
    - Insertion of java code in HTML pages
    - Use of special JSP tags - <% … %>

```
<%! int i = 0; %>
<%! int a, b, c; %>
<%! Circle a = new Circle(2.0); %>
```

What are JavaServer Pages?(3)

- JavaServer Pages(JSP) component:
  - Type of Java servlet
  - Designed to fulfill the role of a user interface for a Java web application
- JSPs are text files that combine:
  - HTML
  - XHTML
  - XML
  - Embedded JSP actions and commands



What are JavaServer Pages?(4)

- Using JSP, you can:
  - Collect input from users through:
    - Webpage forms
    - Present records from a database or another source
  - Create Webpages dynamically

What are JavaServer Pages?(5)

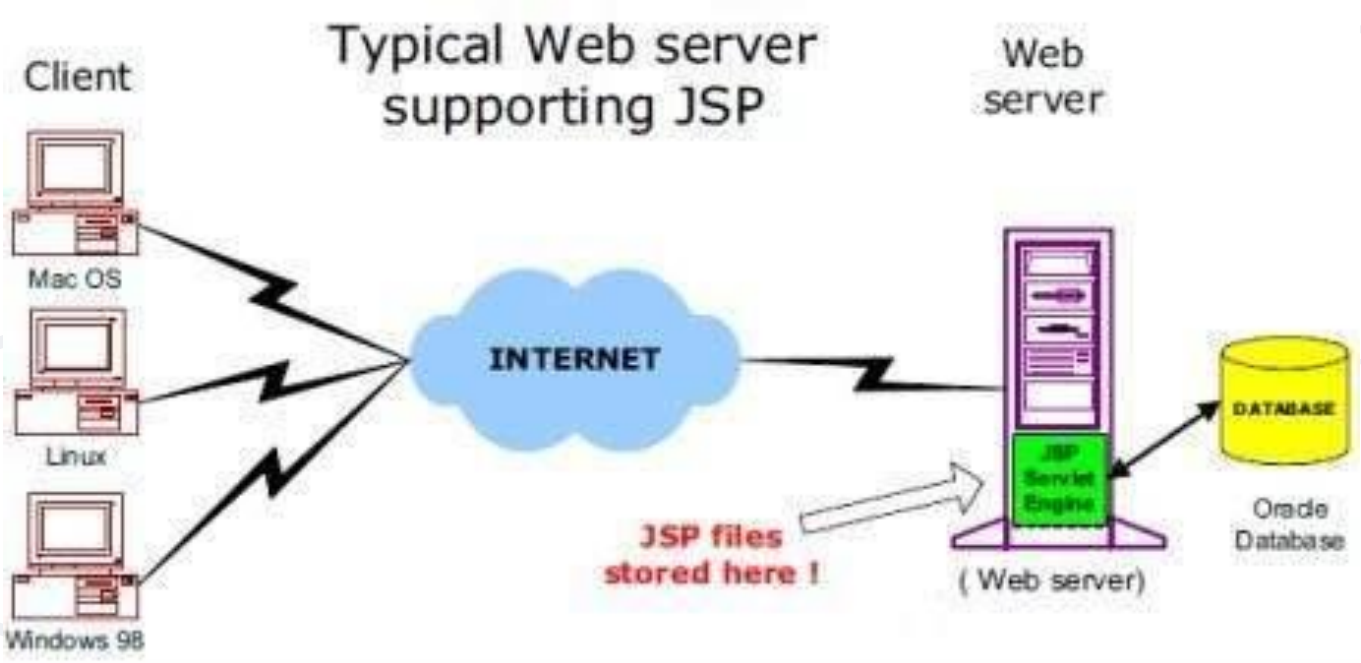
- JSP tags purposes:
  - Retrieving information from a database
  - Registering user preferences
  - Accessing JavaBeans components
  - Passing control between pages
  - Sharing information between requests, pages etc

JSP Architecture(1)

- The web server needs:
  - JSP engine, i.e, a container to process JSP pages
- The JSP container:
  - Responsible for intercepting requests for JSP pages
  - Works with the Web server
  - Provides the runtime environment and other services a JSP needs
  - Knows how to understand the special elements that are part of JSPs

JSP Architecture(2)

- Following diagram shows the position of JSP container and JSP files in a Web application.



JSP Processing(1)

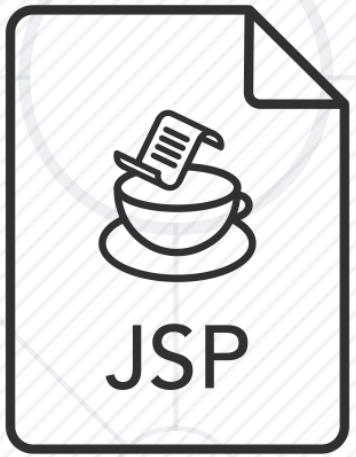
- Browser sends an HTTP request to the web server
- Web server:
  - Recognizes that the HTTP request is for a JSP
  - Forwards it to a JSP engine using the URL or JSP page which ends with .jsp instead of .html

JSP Processing(2)

- JSP engine:
  - Loads the JSP page from disk
  - Converts it into a servlet content:
    - Template text is converted to println( ) statements
    - All JSP elements are converted to Java code that implements the corresponding dynamic behavior of the page

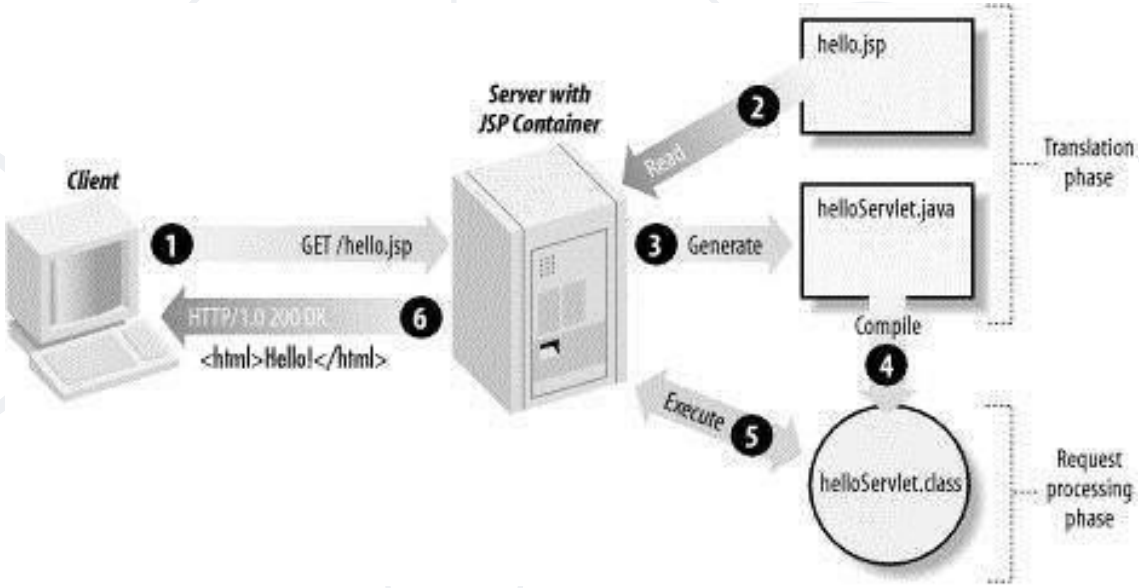
JSP Processing(3)

- JSP engine:
  - Compiles the servlet into an executable class
  - Forwards the original request to a servlet engine
- Servlet engine:
  - Loads the Servlet class and executes it:
    - The servlet produces an output in HTML format
    - Output is further passed inside a HTTP response



JSP Processing(4)

- Web server:
  - Forwards the HTTP response to your browser
- Web browser:
  - Handles the dynamically-generated HTML as if it is a static page



JavaServer Pages

Examples

Hello World!(1)

Servlet.java

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

  private String message;

  protected void init(…) {
    this.message = “Hello World!”;
  }

  protected void doGet(…) {
    request.setAttribute(“message”, this.message);
    request.gerRequestDispatcher(“/home.jsp”).forward(request, response);
  }
}
```

Hello World!(2)

home.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>HelloWorld!</title>
</head>
<body>
    <h1><%= request.getAttribute("message") %></h1>
</body>
</html>
```

Register Users!(1)

HomeServlet.java

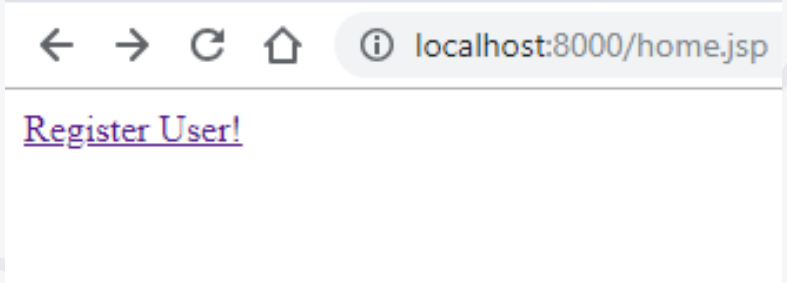
```
@WebServlet("/")
public class HomeServlet extends HttpServlet {

  protected void doGet(…) {
    request.gerRequestDispatcher("/home.jsp").forward(request, response);
  }
}
```

Register Users!(2)

home.jsp

```
<%@ page import="java.util.List" %>
<%@ page import="domain.User" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Register Users!</title>
</head>
<body>
    <a href="/register">Register User!</a>
    <% if ((request.getSession().getAttribute("users")) != null &&
((List<User>)request.getSession().getAttribute("users")).size() > 0) {%>
        <h2>Our Users!</h2>
        <% for (User user : ((List<User>) request.getSession().getAttribute("users"))) {%>
            <p><%= String.format("Username: %s; Password: %s;", user.getUsername(),
user.getPassword())%></p>
        <%}%>
    <% }%>
</body>
</html>
```



Register Users!(3)

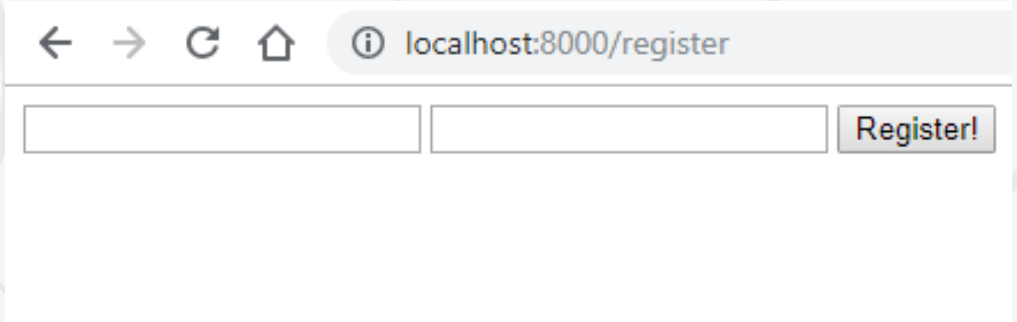
RegisterServlet.java

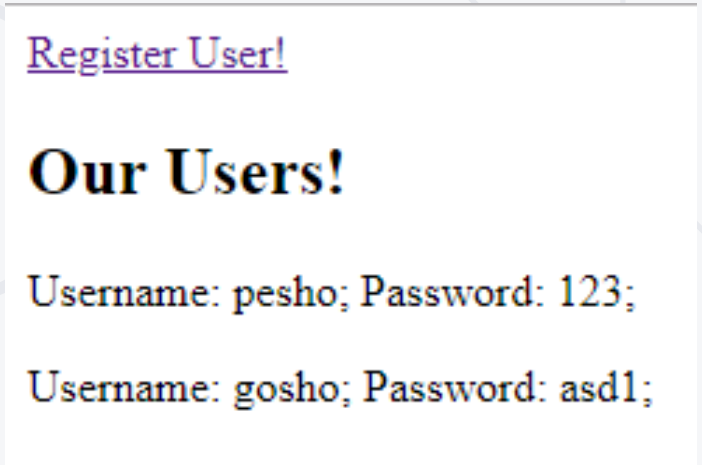
```
@WebServlet("/register")
public class RegisterServlet extends HttpServlet {
  private List<User> users;
  protected void doGet(…) {
    if (this.users == null) {
        this.users = new ArrayList<>();
        request.getSession().setAttribute("users", this.users);
    }
    request.getRequestDispatcher("/register-user.jsp").forward(reqest, response);
  }
  protected void doPost(…) {
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    this.users.add(new User(username, password));
    response.sendRedirect("home.jsp");
  }
}
```

Register Users!(4)

register-user.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Register Users!</title>
</head>
<body>
  <form action="/register" method="post">
    <label for="usernameId">
      <input type="text" id="usernameId" name="username">
    </label>
    <label for="passwordId">
      <input type="password" id="passwordId" name="password">
    </label>
    <button>Register!</button>
  </form>
</body>
</html>
```





Filters

Overview

What are Servlet Filters?

- Servlet Filters:
  - Pluggable java components
  - Used to intercept and process:
    - Requests *before* they are sent to servlets
    - Response *after* servlet code is finished and before container sends it back to the client

Servlet Filter Tasks

- Filter tasks:
  - Logging request parameters to log files
  - Authentication and authorization of request for resources
  - Formatting of request body or header before sending it to servlet
  - Compressing the response data sent to the client
  - Alter response by adding some cookies, header information etc.

Servlet Filter interface(1)

- Servlet Filter interface:
  - Similar to Servlet interface
  - Managed by servlet container
  - Contains lifecycle methods of a Filter:
    - init() method – called when container initializes the filter
    - doFilter() method – called when container needs to apply filter to a resource. FilterChain parameter is used to invoke the next filter in the chain
    - destroy() method

Servlet Filter Configuration

web.xml

```
<?xml version = "1.0" encoding = "UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
   http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
         id="WebApp_ID" version="2.5">
    <filter>
        <filter-name>ServletFilter</filter-name>
        <filter-class>demo.web.filters.ServletFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>ServletFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
</web-app>
```

Servlet Filter class

ServletFilter.java

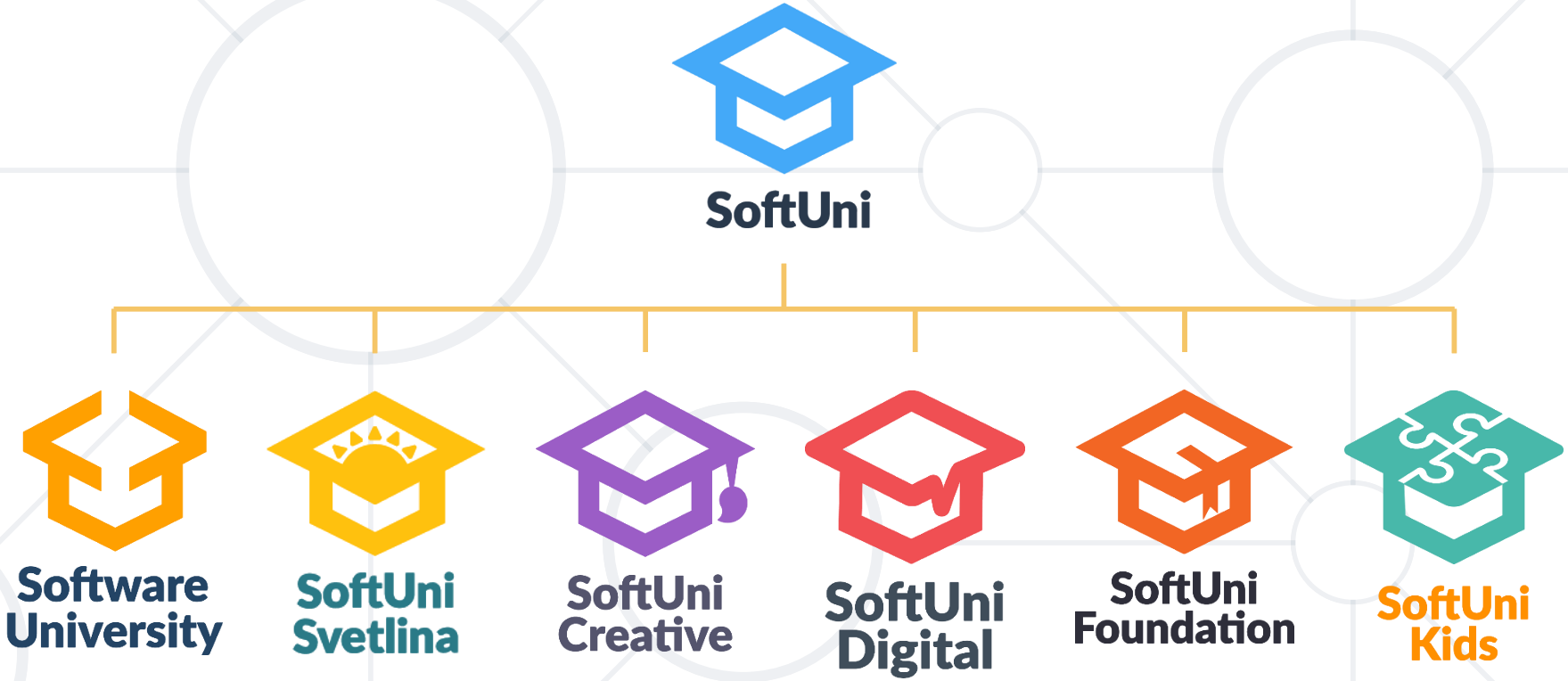
```
@WebFilter("/ServletFilter")
public class ServletFilter implements Filter {

  public void doFilter(…) {
    HttpServletRequest req = (HttpServletRequest) request;
    HttpServletResponse resp = (HttpServletResponse) response;
    req.getSession().setAttribute("username", "pesho");
    chain.doFilter(req, resp);
  }
}
```

Summary

- JavaServer Pages:
  - Server-side programming technology.
  - Enables the creation of dynamic, platform-independent method for building Web-based applications.
- Servlet Filters:
  - Pluggable java components
  - Used to intercept and process requests and responses

Questions?


SoftUni
Software University
SoftUni Svetlina
SoftUni Creative
SoftUni Digital
SoftUni Foundation
SoftUni Kids

SoftUni Diamond Partners

Software University

XSsoftware

SBTech
we know sports

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

SUPER HOSTING .BG

INDEAVR
Serving the high achievers

INFRAGISTICS®

LIEBHERR

æternity

SoftUni Organizational Partners

Software University

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

Lukanet.com

WORLD OF MYTHS

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license