CMP 436/774

Introduction to Servlets and Java Server Pages (JSPs)

Fall 2013
Department of Mathematics and Computer Science
Lehman College, CUNY

Why Web Applications (Apps)?

- **So why does everyone want Web apps?**
  - Universal access
    - Everyone already has a browser installed
    - Any computer on the network can access content
  - Automatic "updates"
    - Content comes from server, so is never out of date
- **Downsides to browser-based apps**
  - GUI is poor
    - HTML is OK for static documents, but not effective for dynamic documents generated/updated by the server side apps
  - Communication is inefficient
    - HTTP is poor protocol (stateless) for the way we now use Web apps

Why Build Web Pages Dynamically?

- **The Web page is based on data submitted by the user**
  - E.g., results page from search engines and order-confirmation pages at on-line stores
- **The Web page is derived from data that changes frequently**
  - E.g., a weather report or news headlines page
- **The Web page uses information from databases or other server-side sources**
  - E.g., an e-commerce site could use a servlet to build a Web page that lists the current price and availability of each item that is for sale.
- ***Servlet/JSP technologies (in EE) enable for developing dynamic, portable, secure, robust, reliable web applications.***
  - *In this lecture, we study Servlets and JSP technologies*

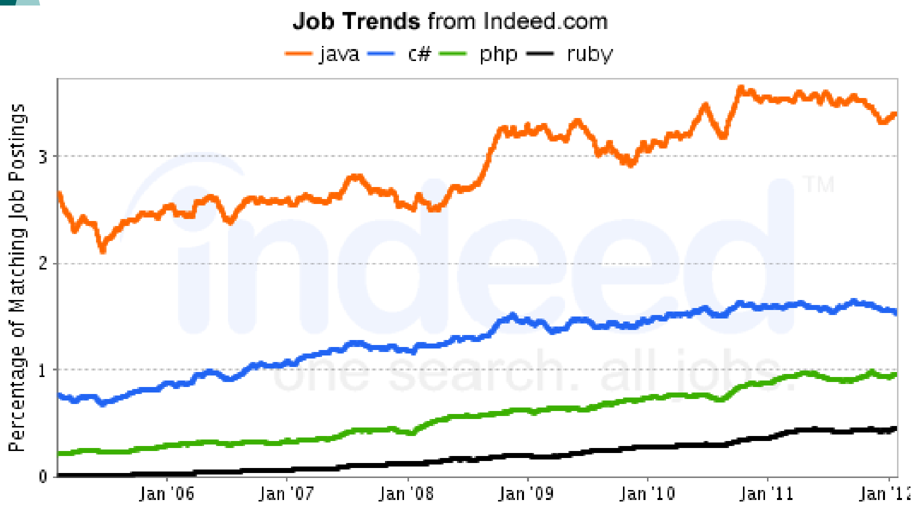
3

Servlets/JSPs vs. JSF

- **Servlets and JSP**
  - Well-established standard
  - Used by google.com, ebay.com, walmart.com, and thousands of other popular sites
  - Relatively low level by today's standards
- **JSF (JavaServer Faces) Version 2**
  - Now an official part of Java EE 6
  - Higher-level features: integrated Ajax support, field validation, page templating, rich third-party component libraries, etc. Designed around the MVC approach.
  - Not yet as widely used, but recommended for new projects to be developed
  - *JSF will be studied (after Servlets/JSPs, intro to EJBs), refer to the course outline shown in the course Web Site*

4

Web App Language Popularity



5

Servlets and JSPs

Servlets

Examples in SimpleServlets and SimpleJSPs NetBeans Projects

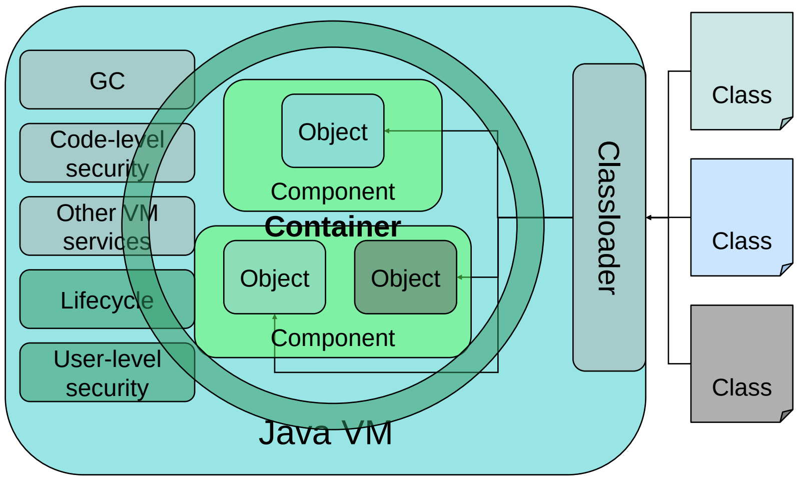
6

What is a Servlet?

- Servlets are Java programs that serve as an mediating layer between an HTTP request of a client and applications in the Web server.
- A Servlet is a dynamically loaded module that services requests from a Web server.
- A servlet runs entirely inside the a JVM (Java Virtual Machine) of a container.
- Temporary (Persistent) Servlets are activated when clients request their services
- Permanent Servlets are active when their host servers are up.
- Servlets are designated as temporary and permanent through configurations of the hosting servers.
- The GlassFish Server can cache the results of invoking a servlet, a JSP, or any URL pattern to make subsequent invocations of the same servlet, JSP, or URL pattern faster.

7

Components and Containers



8

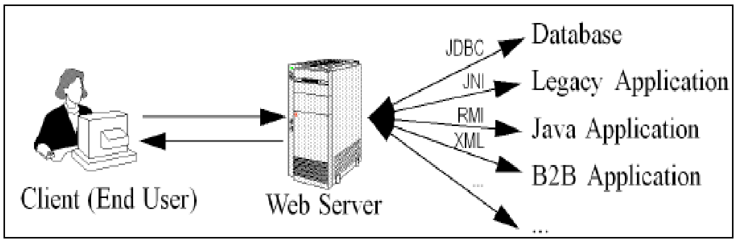
Components and Containers

- **Components:**
  - One or more objects that implement a well-defined application service
  - Make use of component services
- **Container:**
  - Runtime entity that manages components and provides their services
- **Components need to be:**
  - Written within the contracts defined by the container
    - APIs plus some rules related to the component services
  - Deployed to containers
    - Describe the components
    - Deliver all the elements (classes, resources, etc.) that implement the components
    - Provide instructions on how to manage them
    - Assemble/package components into an application assembly
- **Servlets typically run inside multithreaded servlet containers that can handle multiple requests concurrently.**
  - Developers must be aware to synchronize access to any shared resources such as files, network connections, and as well as the servlet's class and instance variables.

9

A Servlet's Job

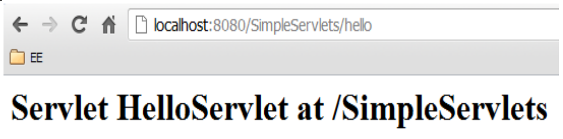
- **Read explicit data sent by client (form data).**
- **Read implicit data sent by client (request headers).**
- **Generate the results.**
- **Send the explicit data back to client (HTML).**
- **Send the implicit data to client (status codes and response headers).**



10

A Servlet that Generates a HTML doc

```
@WebServlet(name = "HelloServlet", urlPatterns = {"/hello"})
public class HelloServlet extends HttpServlet {
/*invoked by  doGet() and doPost() */
  protected void processRequest(HttpServletRequest request,
     HttpServletResponse response)
         throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        /* TODO output your page here. You may use following
     sample code. */
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet HelloServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet HelloServlet at " +
     request.getContextPath() + "</h1>");
        out.println("</body>");
        out.println("</html>");
    } finally {
        out.close();
    }
  } //continues.....
```

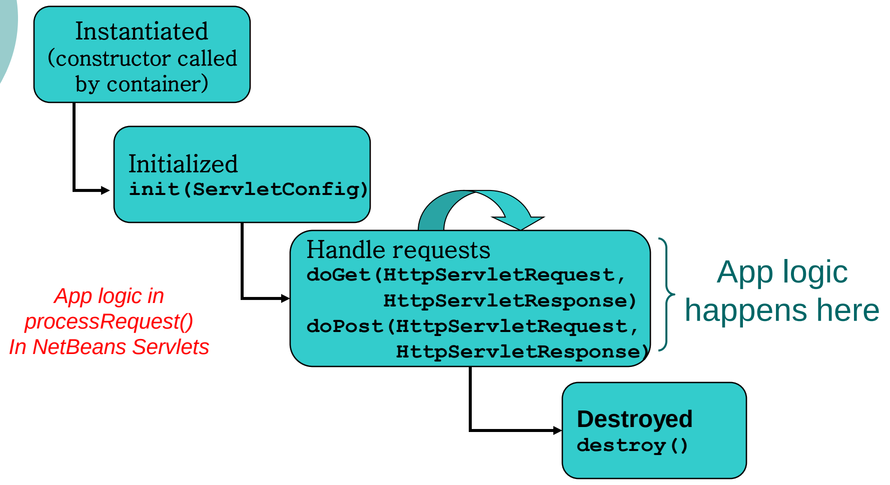


Servlet HelloServlet at /SimpleServlets

Servlet Life Cycle

- **The container loads and initializes the Servlet.**
- **The Servlet handles client requests.**
- **The container can remove the Servlet.**
- **The servlet can remain loaded to handle additional requests:**
  - Incur startup costs only once, may consume server resources if active.
- **The javax.servlet.Servlet interface introduces the methods supporting communication between the Servlets and their container.**
  - void init(ServletConfig config): to be executed when the Servlet starts running
  - void destroy()
  - void service(ServletRequest req, ServletResponse res): to be executed in response to client requests
  - ServletConfig getServletConfig():info about the servlet environment, provided by the host in init
  - String getServletInfo(): information about the Servlet

Example: HTTP Servlet Lifecycle



Instantiated (constructor called by container)
Initialized init(ServletConfig)
Handle requests doGet(HttpServletRequest, HttpServletResponse) doPost(HttpServletRequest, HttpServletResponse)
App logic happens here
App logic in processRequest() In NetBeans Servlets
Destroyed destroy()

The Servlet Life Cycle (cont'd)

- **Init:**
  - Executed once when the Servlet is first loaded:
  - Not called for each request.
  - init-params are read.
- **Service:**
  - Called in a new thread by server for each request:
  - Dispatches to doGet, doPost, etc
  - Do not override service().
    - The service method gives you automatic support for:
    - HEAD requests,
    - OPTIONS requests, //cache, encoding, etc
    - TRACE requests.
- **doGet, doPost, doXxx:**
  - Handles GET, POST, doPut, doTrace, etc.
  - Override these to provide desired behaviour.
- **Destroy:**
  - Called when server deletes Servlet instance,
  - Not called after each request.

The Servlet Life Cycle (cont'd)

- **Main servlet code goes in doGet or doPost:**
  - The HttpServletRequest contains the incoming information
  - The HttpServletResponse lets you set outgoing information
    - Call setContentType to specify MIME type
    - Call getWriter to obtain a Writer pointing to client
- **One-time setup code goes in init**
  - Servlet gets initialized and loaded once
  - Used often in Servlets:
    - e.g., initializing database connection pools.
  - Servlet gets invoked multiple times, but the initialization is done once when it's first invoked.
  - Initialization parameters set in web.xml (…/WEB-INF/web.xml)
    - @WebServlet(urlPatterns="/MyPatter", initParams={@WebInitParam(name="ccc", value="333")}) annotation can be used.
  - Use **ServletConfig.getInitParameter()** to read initialization parameters.
    - *See ShowMessages servlet example*

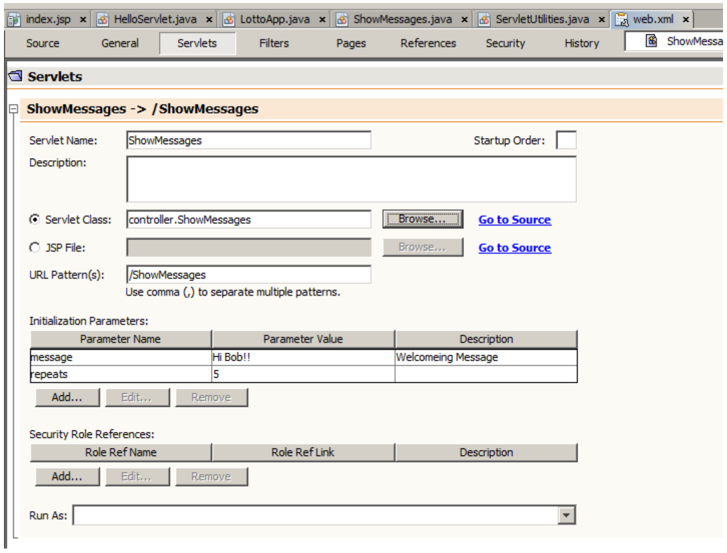
15

ShowMessages Servlet

```
@Override
  public void init() throws ServletException {
     ServletConfig config = getServletConfig();
     message = config.getInitParameter("message");
     if (message == null) {
        message = defaultMessage;
     }
     try {
        String repeatString = config.getInitParameter("repeats");
        repeats = Integer.parseInt(repeatString);
     } catch (NumberFormatException nfe) {
     }
  }
```

16

Initialization Parameters setting



Servlets

ShowMessages -> /ShowMessages

Servlet Name:	ShowMessages
Startup Order:	
Description:	
Servlet Class:	controller.ShowMessages
JSP File:	
URL Pattern(s):	/ShowMessages

Use comma (,) to separate multiple patterns.

Initialization Parameters:

Parameter Name	Parameter Value	Description
message	Hi Bob!!	Welcomeing Message
repeats	5	

Add... Edit... Remove

Security Role References:

Role Ref Name	Role Ref Link	Description

Add... Edit... Remove

Run As:

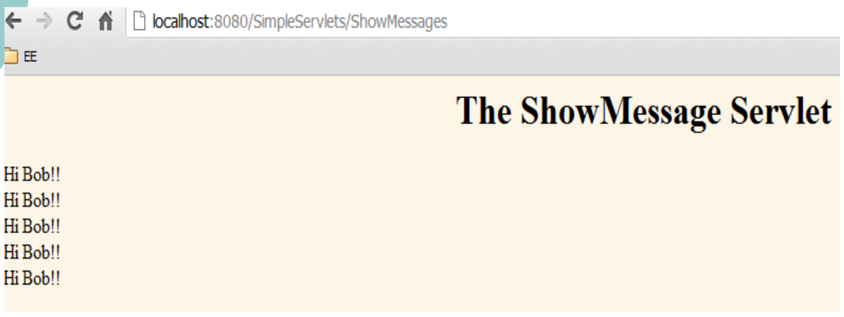
17

ShowMessage (Cont'd)

```
protected void processRequest(HttpServletRequest request, HttpServletResponse
   response)
       throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String title = "The ShowMessage Servlet";
    out.println(ServletUtilities.headWithTitle(title)
         + "<BODY BGCOLOR=\"#FDF5E6\">\n"
         + "<H1 ALIGN=CENTER>" + title + "</H1>");

    for (int i = 0; i < repeats; i++) {
       out.println(message + "<BR>");
    }
    out.println("</BODY></HTML>");
  }
```

18

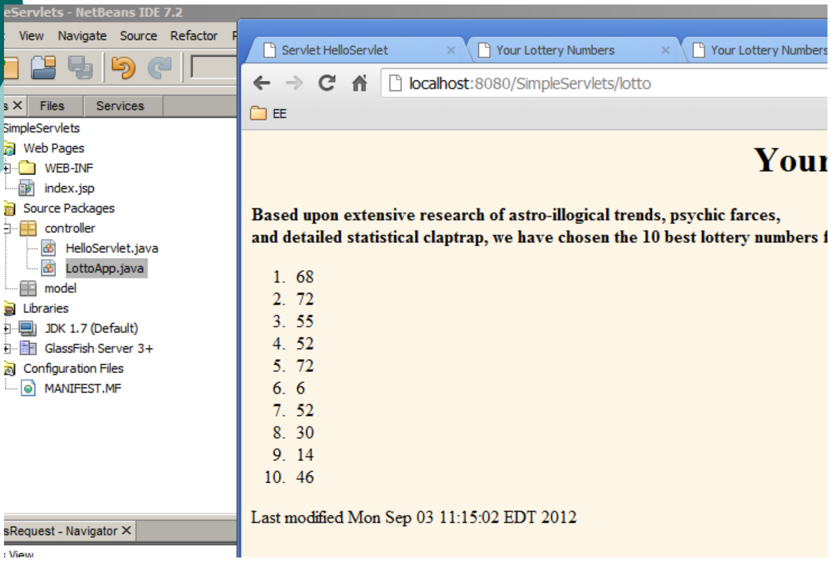
ShowMessages servlet (cont'd)



The ShowMessage Servlet

Hi Bob!!
Hi Bob!!
Hi Bob!!
Hi Bob!!
Hi Bob!!

LottoApp servlet



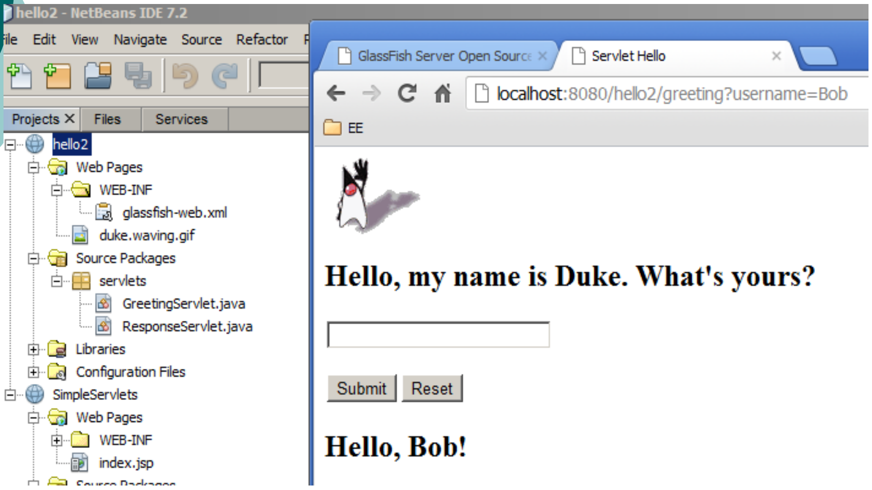
Your

Based upon extensive research of astro-illogical trends, psychic farces, and detailed statistical claptrap, we have chosen the 10 best lottery numbers f

1. 68
2. 72
3. 55
4. 52
5. 72
6. 6
7. 52
8. 30
9. 14
10. 46

Last modified Mon Sep 03 11:15:02 EDT 2012

Hello2 servlet (from EE 6 tutorial)



Hello, my name is Duke. What's yours?

Hello, Bob!

ServletUtilities (in utilities package)

```
….
public static int getIntParameter(HttpServletRequest request,
        String paramName,
        int defaultValue) {
    String paramString = request.getParameter(paramName);
    int paramValue;
    try {
        paramValue = Integer.parseInt(paramString);
    } catch (NumberFormatException nfe) { // null or bad format
        paramValue = defaultValue;
    }
    return (paramValue);
  }
```

Servlet JSP API

- Servlets and JSP API docs from
  - http://glassfish.java.net/nonav/docs/v3/api/overview-summary.html or
  - file:///C:/glassfish4/docs/api/index.html
  - Servlets 3.0/3.1 and JSP 2.2

Servlets and JSPs

Java Server pages (JSPs)

Exmaples are in SimpleJSPs SimpleServlets NetBeans Projects

The JSP Framework

- **Idea:**
  - Use regular HTML for most of page
  - Mark servlet code with special tags
  - Entire JSP page gets translated into a servlet (once), and servlet is what actually gets invoked (for each request)
- Example:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <title>Order</title>
   </head>
   <body>
      <form method=post action="OrderConfirmation.jsp">
         <h4>Please Enter The Book Name You Order.....</h4>
               <input type=text name="title">
               <input type=submit value="Click to Order Now...">
      </form>
   </body>
</html>
```

25

Benefits of JSP

- **Although JSP technically can't do anything servlets can't do, JSP makes it easier to:**
  - Write HTML
  - Read and maintain the HTML
- **JSP makes it possible to:**
  - Use standard HTML tools such as DreamWeaver
  - Have different members of your team do the HTML layout than do the Java programming
- **JSP encourages you to**
  - Separate the (Java) code that creates the content from the (HTML) code that presents it

26

Advantages of JSP Over Competing Technologies

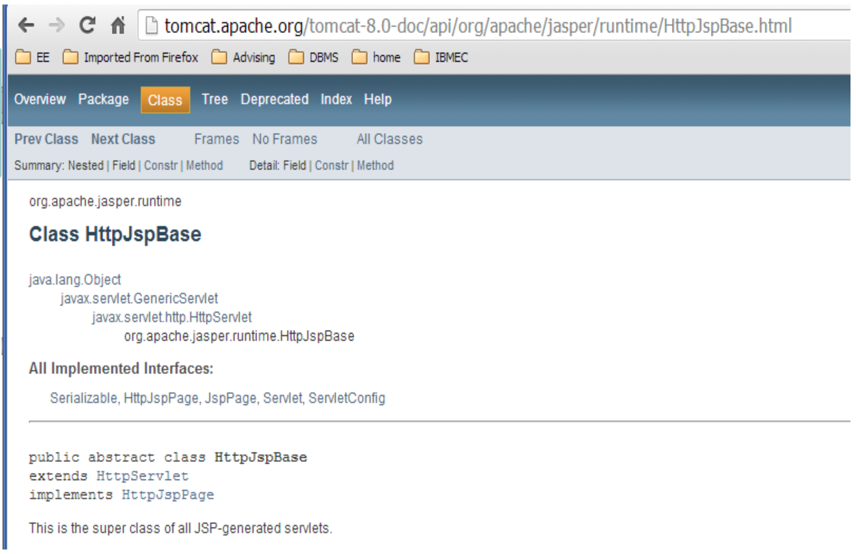
- **Versus ASP or ColdFusion**
  - Better language for dynamic part
  - Portable to multiple servers and operating systems
- **Versus PHP**
  - Better language for dynamic part
  - Better tool support
- **Versus client-side JavaScript (in browser)**
  - Capabilities mostly do not overlap with JSP, but
    - You control server, not client
    - Richer language
- **Versus static HTML**
  - Dynamic features
- **Versus pure servlets**
  - More convenient to create HTML
  - Divide and conquer
  - *JSP programmers still need to know servlet programming*

27

How Does a JSP work?

- **Web server hands a JSP request to the web container (or JSP container)**
- **Web container picks up the corresponding JSP compiles it**
  - Parses it (checks tag syntax, etc.)
  - Converts page to a Java Servlet (implementing JspPage), with your code inside the _jspService() method
  - Compiles the generated Servlet code (one-time only)
- **Original request is passed to the compiled component**
  - Init, service, destroy lifecycle events mapped into JSP versions
  - jspInit(), _jspService() and jspDestroy() are called the life cycle methods of the JSP

28



tomcat.apache.org/tomcat-8.0-doc/api/org/apache/jasper/runtime/HttpJspBase.html
EE Imported From Firefox Advising DBMS home IBMEC
Overview Package Class Tree Deprecated Index Help
Prev Class Next Class Frames No Frames All Classes
Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

org.apache.jasper.runtime

Class HttpJspBase

```
java.lang.Object
    javax.servlet.GenericServlet
        javax.servlet.http.HttpServlet
            org.apache.jasper.runtime.HttpJspBase
```

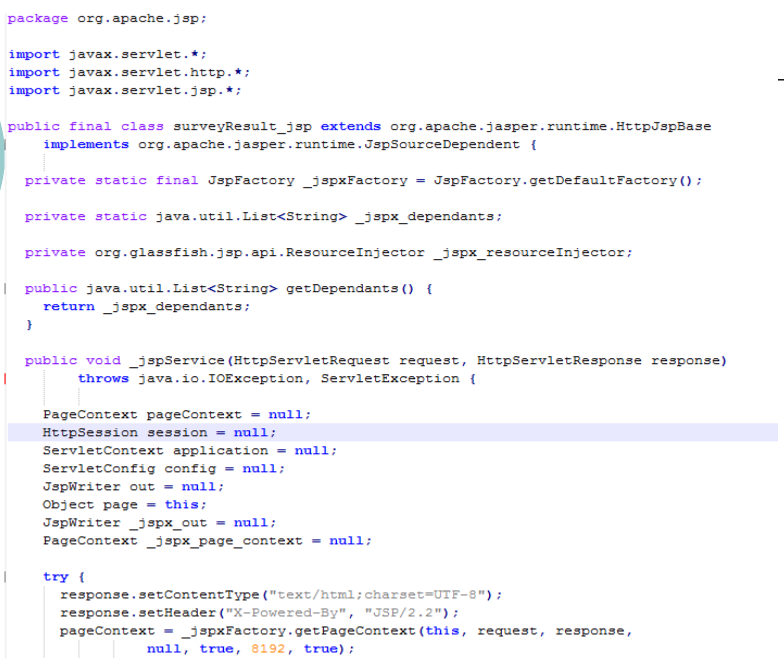
All Implemented Interfaces:

Serializable, HttpJspPage, JspPage, Servlet, ServletConfig

```
public abstract class HttpJspBase
extends HttpServlet
implements HttpJspPage
```

This is the super class of all JSP-generated servlets.

29



```
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;

public final class surveyResult_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

  private static final JspFactory _jspxFactory = JspFactory.getDefaultFactory();

  private static java.util.List<String> _jspx_dependants;

  private org.glassfish.jsp.api.ResourceInjector _jspx_resourceInjector;

  public java.util.List<String> getDependants() {
    return _jspx_dependants;
  }

  public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {

    PageContext pageContext = null;
    HttpSession session = null;
    ServletContext application = null;
    ServletConfig config = null;
    JspWriter out = null;
    Object page = this;
    JspWriter _jspx_out = null;
    PageContext _jspx_page_context = null;

    try {
      response.setContentType("text/html;charset=UTF-8");
      response.setHeader("X-Powered-By", "JSP/2.2");
      pageContext = _jspxFactory.getPageContext(this, request, response,
                null, true, 8192, true);
```

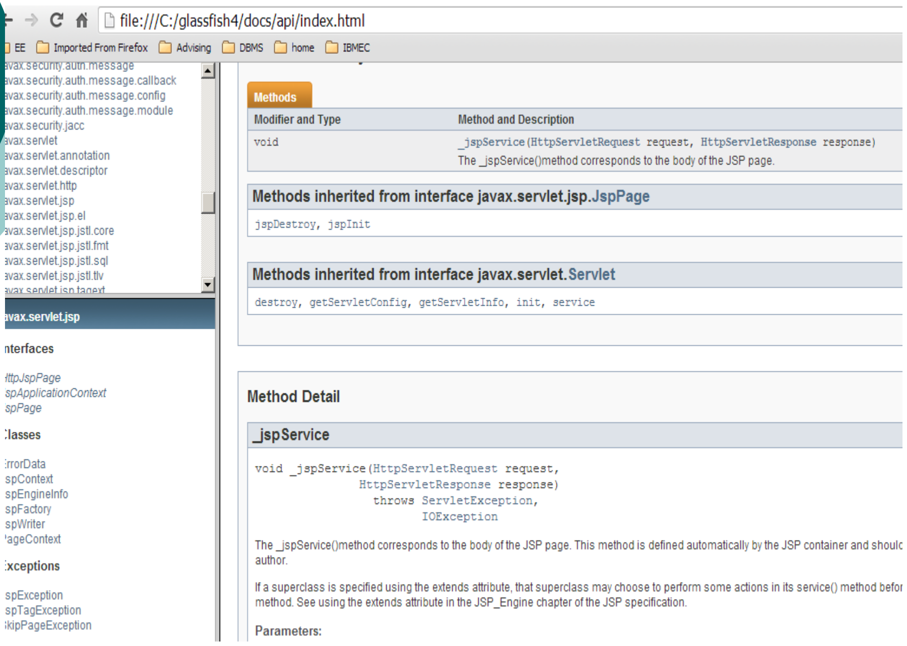
30

JSPs as Web Components

- **Very similar lifecycle, with a few exceptions**
  - They extend the class which implements *HttpJspPage* interface, which maps the callbacks into JSP callbacks
    - jspInit(), _jspService(), jspDestroy()
  - Initialization protocol:
    - *ServletConfig* is set by JSP engine
    - jspInit() is called, can access *config* using *getServletConfig()* (returns initialization and startup parameters for this Servlet)
  - See javax.servlet.jsp package; javax.servlet.jsp.PageContext abstract class (provides information that is not specific to Servlets:
    - API to manage the various scoped namespaces for tag management
    - a mechanism to obtain the JspWriter for output
    - a mechanism to expose page directive attributes to the scripting environment (e.g., for JSP-EL).

- **Biggest difference between Servlets and JSPs is that all this is hidden from you.**

31


file:///C:/glassfish4/docs/api/index.html

EE | Imported From Firefox | Advising | DBMS | home | IBMEC

avax.security.auth.message
avax.security.auth.message.callback
avax.security.auth.message.config
avax.security.auth.message.module
avax.security.jacc
avax.servlet
avax.servlet.annotation
avax.servlet.descriptor
avax.servlet.http
avax.servlet.jsp
avax.servlet.jsp.el
avax.servlet.jsp.jstl.core
avax.servlet.jsp.jstl.fmt
avax.servlet.jsp.jstl.sql
avax.servlet.jsp.jstl.tlv
avax.servlet.jsp.tagext

avax.servlet.jsp

nterfaces

ttpJspPage
spApplicationContext
spPage

lasses

rrorData
spContext
spEngineInfo
spFactory
spWriter
ageContext

xceptions

spException
spTagException
kipPageException

Methods

Modifier and Type	Method and Description
void	_jspService(HttpServletRequest request, HttpServletResponse response) The _jspService()method corresponds to the body of the JSP page.

Methods inherited from interface javax.servlet.jsp.JspPage

jspDestroy, jspInit

Methods inherited from interface javax.servlet.Servlet

destroy, getServletConfig, getServletInfo, init, service

Method Detail

_jspService

```
void _jspService(HttpServletRequest request,
               HttpServletResponse response)
                 throws ServletException,
                        IOException
```

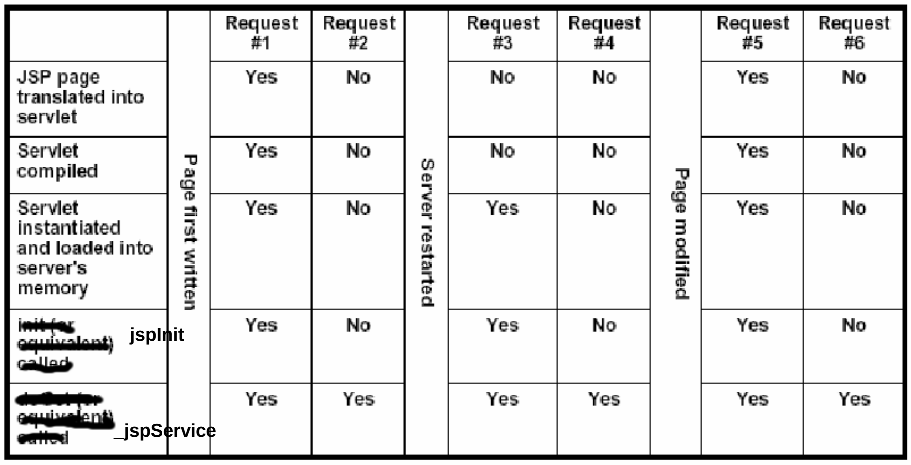
The _jspService()method corresponds to the body of the JSP page. This method is defined automatically by the JSP container and shoulc author.

If a superclass is specified using the extends attribute, that superclass may choose to perform some actions in its service() method befor method. See using the extends attribute in the JSP_Engine chapter of the JSP specification.

Parameters:

32

JSP Lifecycle



		Request #1	Request #2		Request #3	Request #4		Request #5	Request #6
JSP page translated into servlet	Page first written	Yes	No	Server restarted	No	No	Page modified	Yes	No
Servlet compiled		Yes	No		No	No		Yes	No
Servlet instantiated and loaded into server's memory		Yes	No		Yes	No		Yes	No
[redacted] jspInit		Yes	No		Yes	No		Yes	No
[redacted] _jspService		Yes	Yes		Yes	Yes		Yes	Yes

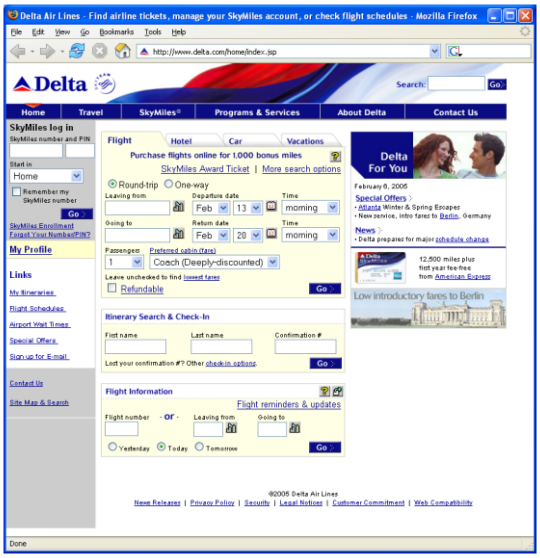
Ten Most Popular Web Sites (Alexa.com, 2010)

1. **Google**
   - Java (Web), C++ (indexing)
2. **Facebook**
   - PHP
3. **YouTube**
   - Flash, Python, Java
4. **Yahoo**
   - PHP and Java
5. **Microsoft Live.com**
   - .NET
6. **Baidu**
   - Unknown
7. **Wikipedia**
   - PHP
8. **Blogger**
   - Java
9. **MSN**
   - .NET
10. **Twitter**
    - Ruby on Rails, Scala, Java

Fall 2010: Google reports over two *billion* Web pages that use JSP

JSP/Servlets in the Real World: Airlines

- Delta Airlines
- United Airlines
- AirTran
- American Airlines
- British Airways
- KLM
- Air China
- Saudi Arabian Airlines
- Iceland Air


Delta Air Lines - Find airline tickets, manage your SkyMiles account, or check flight schedules - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.delta.com/home/index.jsp

Delta Search: Go

Home | Travel | SkyMiles® | Programs & Services | About Delta | Contact Us

SkyMiles log in
SkyMiles number and PIN
Start in: Home
Remember my SkyMiles number
Go
SkyMiles Enrollment
Forgot Your Number/PIN?
My Profile
Links
My Itineraries
Flight Schedules
Airport Wait Times
Special Offers
Sign up for E-mail
Contact Us
Site Map & Search

Flight | Hotel | Car | Vacations
Purchase flights online for 1,000 bonus miles
SkyMiles Award Ticket | More search options
[✓ Round-trip] One-way
Leaving from: | Departure date: Feb 13 | Time: morning
Going to: | Return date: Feb 20 | Time: morning
Passengers: 1 | Preferred cabin (fare): Coach (Deeply-discounted)
Leave unchecked to find lowest fares
Refundable
Go

Itinerary Search & Check-In
First name | Last name | Confirmation #
Lost your confirmation #? Other checkin options.
Go

Flight Information
Flight reminders & updates
Flight number - or - Leaving from | Going to
Yesterday [✓ Today] Tomorrow
Go

Delta For You
February 6, 2005
Special Offers
- Atlanta Winter & Spring Escapes
- New service, intro fares to Berlin, Germany
News
- Delta prepares for major schedule change
12,500 miles plus first year fee-free from American Express
Low introductory fares to Berlin

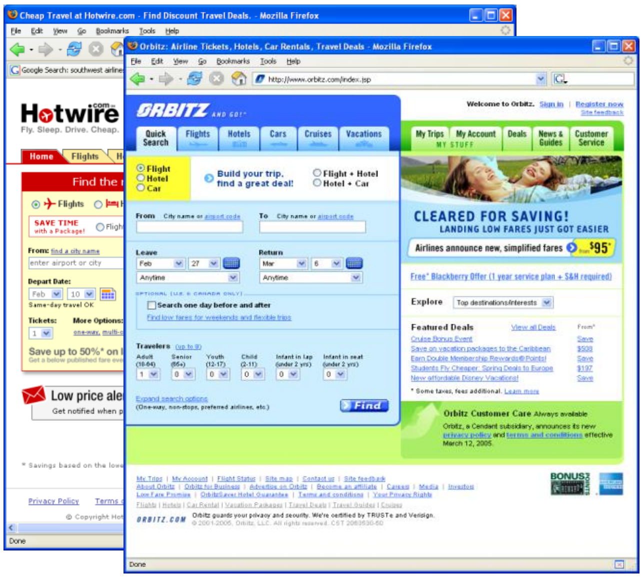
©2005 Delta Air Lines
News Releases | Privacy Policy | Security | Legal Notices | Customer Commitment | Web Compatibility

Done

35

JSP/Servlets in the Real World: Travel Sites

- Travelocity.com
- Orbitz.com
- HotWire.com
- Hotels.com
- CheapTickets.com
- National Car Rental
- Avis Car Rental
- Enterprise Car Rental
- Hertz Car Rental


Cheap Travel at Hotwire.com - Find Discount Travel Deals. - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Google Search: southwest airlines

Hotwire.com
Fly. Sleep. Drive. Cheap.
Home | Flights | H
Find the r
[✓ Flights] H
SAVE TIME with a Package! Fligh
From: find a city name
enter airport or city
Depart Date: Feb 10
Same-day travel OK
Tickets: 1 | More Options: one-way, multi-c
Save up to 50%* on l
Get a below published fare eve
Low price ale
Get notified when p
* Savings based on the lowe
Privacy Policy Terms c
© Copyright Hot
Done

Orbitz: Airline Tickets, Hotels, Car Rentals, Travel Deals - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.orbitz.com/index.jsp

ORBITZ AND GO!
Welcome to Orbitz. Sign in | Register now
Site feedback
Quick Search | Flights | Hotels | Cars | Cruises | Vacations
My Trips | My Account | Deals | News & Guides | Customer Service
MY STUFF

[✓ Flight] Hotel Car
Build your trip, find a great deal! Flight + Hotel Hotel + Car
From City name or airport code | To City name or airport code
Leave: Feb 27 Anytime | Return: Mar 6 Anytime
OPTIONAL (U.S. & CANADA ONLY)
Search one day before and after
Find low fares for weekends and flexible trips
Travelers (up to 8)
Adult (18-64): 1 | Senior (65+): 0 | Youth (12-17): 0 | Child (2-11): 0 | Infant in lap (under 2 yrs): 0 | Infant in seat (under 2 yrs): 0
Expand search options
(One-way, non-stops, preferred airlines, etc.)
Find

CLEARED FOR SAVING!
LANDING LOW FARES JUST GOT EASIER
Airlines announce new, simplified fares from $95*
Free* Blackberry Offer (1 year service plan + S&H required)
Explore: Top destinations/interests

Featured Deals	From*
Cruise Bonus Event	Save
Save on vacation packages to the Caribbean	$508
Earn Double Membership Rewards® Points!	Save
Students Fly Cheaper: Spring Deals to Europe	$197
New affordable Disney Vacations!	Save

View all Deals
* Some taxes, fees additional. Learn more

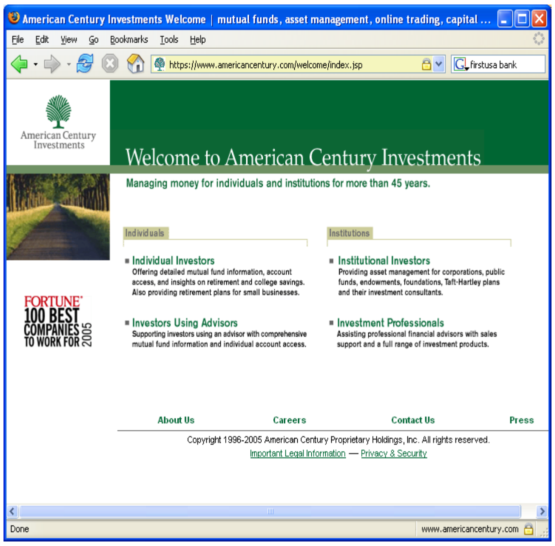
Orbitz Customer Care Always available
Orbitz, a Cendant subsidiary, announces its new privacy policy and terms and conditions effective March 12, 2005.

My Trips | My Account | Flight Status | Site map | Contact us | Site feedback
About Orbitz | Orbitz for Business | Advertise on Orbitz | Become an affiliate | Careers | Media | Investors
Low Fare Promise | OrbitzSaver Hotel Guarantee | Terms and conditions | Your Privacy Rights
Flights | Hotels | Car Rental | Vacation Packages | Travel Deals | Travel Guides | Cruises
ORBITZ.COM Orbitz guards your privacy and security. We're certified by TRUSTe and Verisign.
© 2001-2005, Orbitz, LLC. All rights reserved. CST 2063530-50
Done

36

JSP/Servlets: Financial Services

- American Century
- Vanguard
- Fidelity
- NY Stock Exchange
- First USA Bank
- Royal Bank of Scotland
- Banco Popular de Puerto Rico
- Bank of America
- China Construction Bank


American Century Investments Welcome | mutual funds, asset management, online trading, capital ...
File Edit View Go Bookmarks Tools Help
https://www.americancentury.com/welcome/index.jsp
firstusa bank
American Century Investments
FORTUNE 100 BEST COMPANIES TO WORK FOR 2005
Individuals
Institutions
About Us
Careers
Contact Us
Press
Done
www.americancentury.com

Welcome to American Century Investments

Managing money for individuals and institutions for more than 45 years.

Individual Investors
Offering detailed mutual fund information, account access, and insights on retirement and college savings. Also providing retirement plans for small businesses.

Institutional Investors
Providing asset management for corporations, public funds, endowments, foundations, Taft-Hartley plans and their investment consultants.

Investors Using Advisors
Supporting investors using an advisor with comprehensive mutual fund information and individual account access.

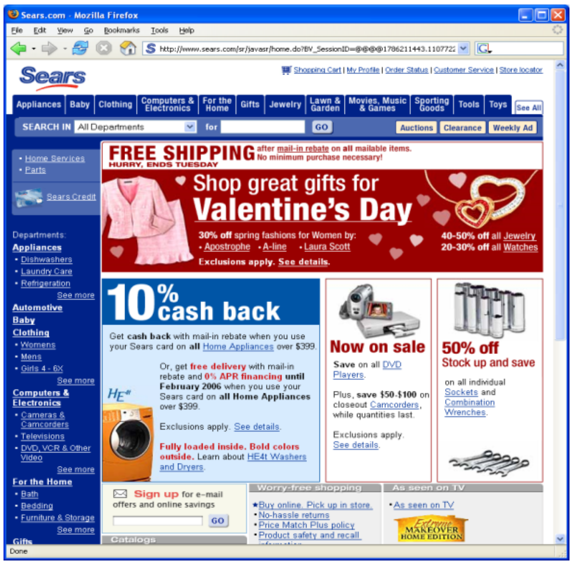
Investment Professionals
Assisting professional financial advisors with sales support and a full range of investment products.

Copyright 1996-2005 American Century Proprietary Holdings, Inc. All rights reserved.
Important Legal Information — Privacy & Security

37

JSP/Servlets in the Real World: Retail

- Sears.com
- Walmart.com
- HomeDepot.com
- SamsClub.com
- Macys.com
- llbean.com
- Kohls.com
- Ikea.com
- Target.com
- Longaberger.com
- Nike.com
- CircuitCity.com



FREE SHIPPING after mail-in rebate on all mailable items. No minimum purchase necessary! HURRY, ENDS TUESDAY

Shop great gifts for Valentine's Day

30% off spring fashions for Women by: Apostrophe • A-line • Laura Scott

40-50% off all Jewelry
20-30% off all Watches

Exclusions apply. See details.

10% cash back

Get cash back with mail-in rebate when you use your Sears card on all Home Appliances over $399.

Or, get free delivery with mail-in rebate and 0% APR financing until February 2006 when you use your Sears card on all Home Appliances over $399.

Exclusions apply. See details.

Fully loaded inside. Bold colors outside. Learn about HE4t Washers and Dryers.

Now on sale

Save on all DVD Players.

Plus, save $50-$100 on closeout Camcorders, while quantities last.

Exclusions apply. See details.

50% off

Stock up and save on all individual Sockets and Combination Wrenches.

Sign up for e-mail offers and online savings

Buy online. Pick up in store.
No-hassle returns
Price Match Plus policy
Product safety and recall information

As seen on TV

38

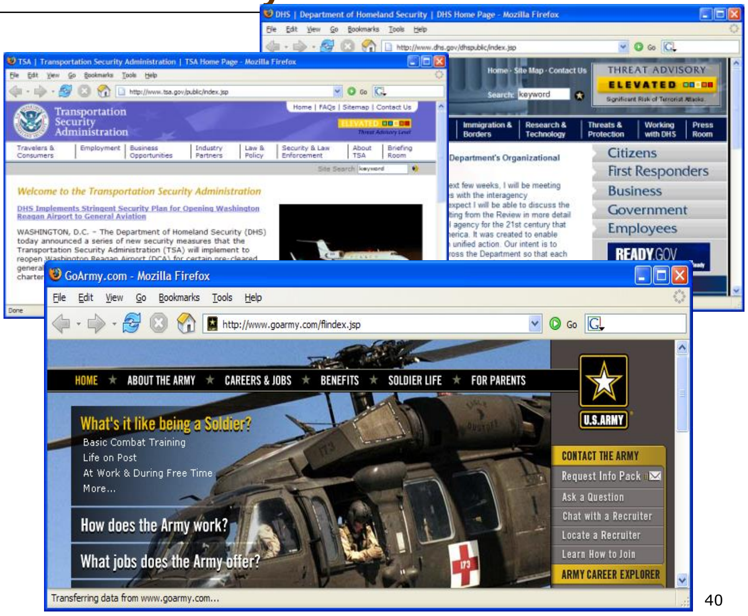
JSP/Servlets in the Real World: Entertainment

- WarnerBrothers.com
- Billboard.com
- E! (eonline.com)
- PBS.org
- Comcast
- games.atari.com

Welcome to Scooby Doo.com - Mozilla Firefox
Billboard Music Charts - Latest Music News - Music Videos - Mozilla Firefox
Billboard.com

JSP/Servlets: Military and Federal Government

- DHS
- TSA
- FAA
- CIA
- NSA
- GSA
- IRS
- Army
- Navy
- USPS


DHS | Department of Homeland Security | DHS Home Page - Mozilla Firefox
TSA | Transportation Security Administration | TSA Home Page - Mozilla Firefox
GoArmy.com - Mozilla Firefox
http://www.goarmy.com/flindex.jsp

Science and Research

- NSF
- UN Oceans
- diabetes.org
- fas.org
- dlse.org
- science.edu.sg
- gbif.net
- collegeboard .com

US NSF - National Science Foundation - Microsoft Internet Explorer

Address http://www.nsf.gov/index.jsp

National Science Foundation
WHERE DISCOVERIES BEGIN

HOME | FUNDING | AWARDS | DISCOVERIES | NEWS | PUBLICATIONS | STATISTICS | ABOUT | FastLane

Take An Underwater Tour

UN Atlas of the Oceans - Mozilla Firefox

http://www.oceansatlas.org/index.jsp

UN Atlas of the Oceans

Login for Members

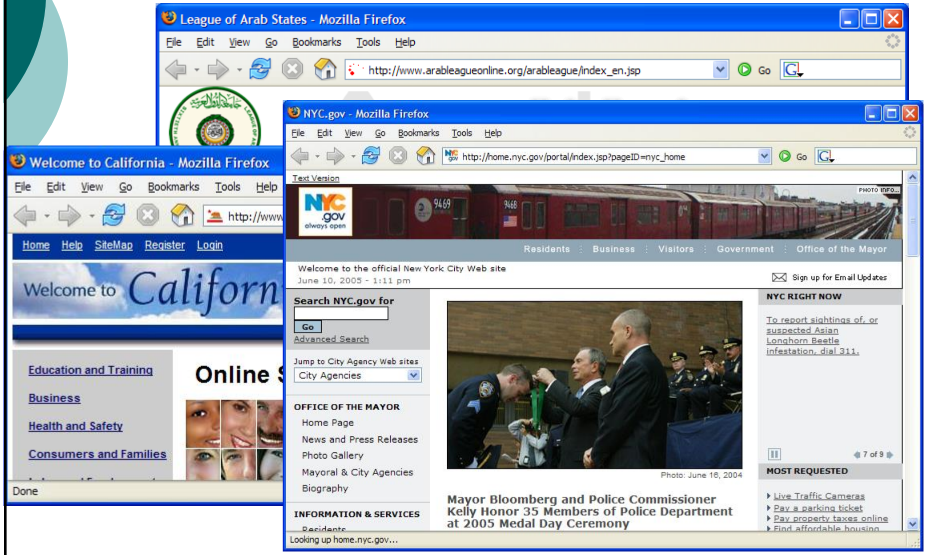
Welcome to the United Nations Atlas of the Oceans

The Atlas is an information system designed for use by policy makers who need to become familiar with ocean issues and by scientists, students and resource managers who need access to underlying data bases and approaches to sustainability. Information about the Atlas...

Members: Forgotten your Password?

Indian Ocean Tsunami 26 December 2004 To contribute to the tsunami information exchange, become a Member of the UN Atlas community - Join Now

JSP/Servlets: State, Local, International


League of Arab States - Mozilla Firefox

http://www.arableagueonline.org/arableague/index_en.jsp

Welcome to California - Mozilla Firefox

Home Help SiteMap Register Login

Welcome to California

Education and Training
Business
Health and Safety
Consumers and Families

Online S

NYC.gov - Mozilla Firefox

http://home.nyc.gov/portal/index.jsp?pageID=nyc_home

Residents | Business | Visitors | Government | Office of the Mayor

Welcome to the official New York City Web site
June 10, 2005 - 1:11 pm

Search NYC.gov for
Advanced Search

Jump to City Agency Web sites
City Agencies

OFFICE OF THE MAYOR
Home Page
News and Press Releases
Photo Gallery
Mayoral & City Agencies
Biography

INFORMATION & SERVICES

Photo: June 16, 2004

Mayor Bloomberg and Police Commissioner Kelly Honor 35 Members of Police Department at 2005 Medal Day Ceremony

NYC RIGHT NOW
To report sightings of, or suspected Asian Longhorn Beetle infestation, dial 311.

MOST REQUESTED
Live Traffic Cameras
Pay a parking ticket
Pay property taxes online
Find affordable housing

Looking up home.nyc.gov...

JSP/Servlets in the Real World: Sports

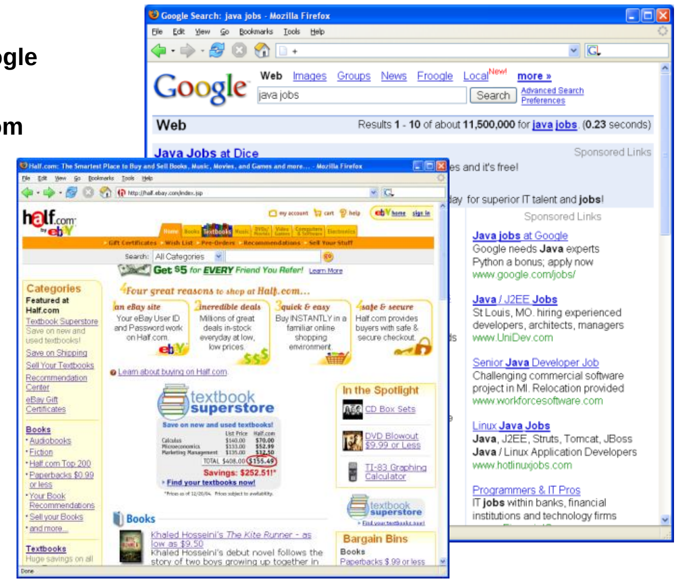
- Baltimore Orioles
- Washington Redskins
- Washington Nationals
- Major League Baseball
- NHL.com
- Nascar.com

The Official Site of The Baltimore Orioles | orioles.com: Homepage - Internet Explorer
http://baltimore.orioles.mlb.com/index.jsp?c_id=bal
News
Audio & Video
Scoreboard
Schedule
Roster
Stats
Community
Fans
Oriole Park
Tickets
Shop
Auction
Mobile
Fantasy
WATCH
LISTEN
GAMEDAY
VIDEO HIGHLIGHTS
Enjoy the Left Field Club Picnic Perch
ORIOLES INSIDER
HEADLINES
FAN VALUE CORNER
MLB.TV PREMIUM New Features
Show your Love.
Internet | Protected Mode: On

43

JSP/Servlets in the Real World: Search/Portals

- Most of Google
- All of Ebay
- netscape.com
- excite.com
- dice.com
- hi5
- Paypal



Google Search: java jobs - Mozilla Firefox
Web
Images
Groups
News
Froogle
Local New!
more »
java jobs
Search
Advanced Search
Preferences
Half.com: The Smartest Place to Buy and Sell Books, Music, Movies, and Games and more... - Mozilla Firefox
Categories
textbook superstore
In the Spotlight
Bargain Bins
Books

Web Results 1 - 10 of about 11,500,000 for java jobs. (0.23 seconds)

Java Jobs at Dice

Sponsored Links

Java jobs at Google
Google needs Java experts
Python a bonus; apply now
www.google.com/jobs/

Java / J2EE Jobs
St Louis, MO. hiring experienced developers, architects, managers
www.UniDev.com

Senior Java Developer Job
Challenging commercial software project in MI. Relocation provided
www.workforcesoftware.com

Linux Java Jobs
Java, J2EE, Struts, Tomcat, JBoss
Java / Linux Application Developers
www.hotlinuxjobs.com

Programmers & IT Pros
IT jobs within banks, financial institutions and technology firms

44

JSP example

```
<%--
    Document   : index
    Created on : Sep 1, 2012, 1:12:57 PM
    Author     : gjung
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

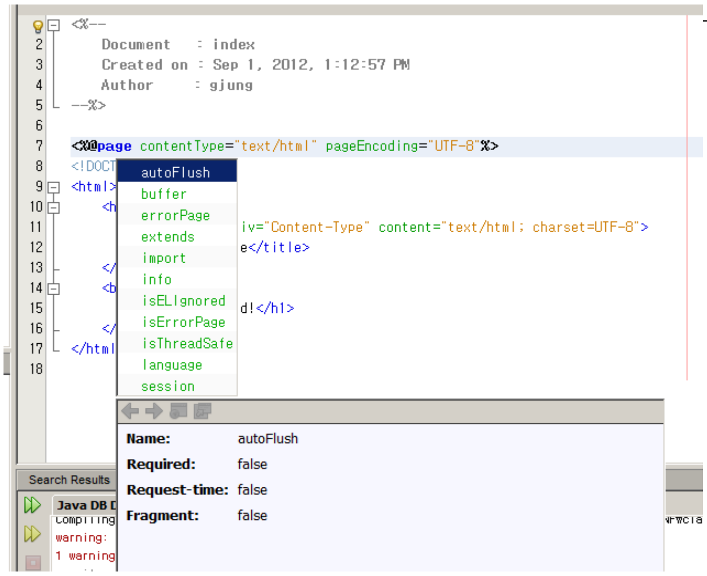
45

JSP comments, page directives

- <%-- and --%> tags delineate JSP comments, everything between two tags is ignored by the JSP compiler. These types of comments will not be rendered on the page.
- Standard HTML comments <!-- and -- >, they will only visible by viewing the source of the rendered page.
- JSP page directives define attributes that apply to the entire page.

46

Page directives



```
<%--
    Document   : index
    Created on : Sep 1, 2012, 1:12:57 PM
    Author     : gjung
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCT
<html>
    <h
                         iv="Content-Type" content="text/html; charset=UTF-8">
                         e</title>
    </
    <b
                         d!</h1>
    </
</html
```

Name:	autoFlush
Required:	false
Request-time:	false
Fragment:	false

```
Compiling
warning:
1 warning
```

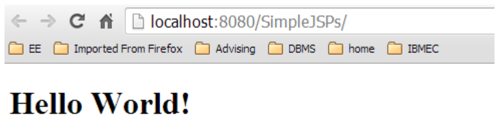
ContextPath

- **By default the context path for the application is the name of the project.**
  - This is the path at which your application can be accessed after it is deployed to the server.
  - For example, GlassFish uses 8080 as its default port number, so during development you'll be able to access the project in a browser window from: http://localhost:8080/ProjectName/

Run a JSP page

- Run the new SimpleJSP project. In the Projects window, you can do this by right-clicking the project node and choosing Run, otherwise, click the Run Project ( ) button in the IDE's main toolbar.

  A browser window opens to display the project's index.jsp page.



Hello World!

JSP/Servlet Correspondence

- Original JSP:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

- Possible resulting Servlet code:

/glassfish4_home/glassfish/domains/domainname/generated/jsp/ProjectName/org/apache/jsp/index_jsp.java

Invoking Dynamic Java Code fromJSPs

Simple Application by
a Small Development Team

- Call Java Code Directly
  (Expressions, Declarations, Scriptlets)
- Call Java Code Indirectly
  (by means of separate Utility Classes)
- Use Beans
  (jsp:useBean, jsp:getProperty, jsp:setProperty)
- Use MVC architecture (Servlet, JSP, JavaBean)
- Use JSP expression Language
  (shorthand to access bean properties, etc)
- Use custom tags
  (Develop tag handler classes; use xml-like custom tags)

Complex Application by
a Big Development Team

Limit Java Code in JSP Pages

- **You have two options for writing JSP**
  - Put 25 lines of Java code directly in the JSP page
  - Put those 25 lines in a separate Java class and put 1 line in the JSP page that invokes it
- **Why is the second option *much* better?**
  - Modular development
  - Debugging
    - If you have syntax errors, you see them immediately at compile time. Simple print statements can be seen.
  - Testing
    - More effective
  - Reuse
    - You can use the same class from multiple pages.

Basic Syntax

- **HTML Text**
  - <H1>……..</H1>
  - Passed through to client. Really turned into Servlet code that looks like out.print("<H1>……..</H1>");
- **HTML Comments**
  - <!-- Comment -->
  - Same as other HTML: passed through to client
- **JSP Comments**
  - <%-- Comment --%>
  - Not sent to client
- **To get <% in output, use <\%**

53

Types of Scripting Elements

- **Expressions:**
  - Format <%=expression %>
  - Evaluated and inserted into the Servlet's output, i.e., results in something like out.println(expression).
- **Scriptlets:**
  - Format <%code%>
  - Inserted verbatim into the Servlet's _jspService method (called a service).
- **Declarations:**
  - Format <%! code%>
  - Inserted verbatim into the body of the Servlet class, outside of any existing methods.

54

JSP Expressions

- **Format:**
  - <%= Java Expression %>
- **Result:**
  - Expression evaluated, converted to String, and placed into HTML page at the place it occurred in the JSP page.
  - That is, expression is placed in _jspService inside out.print.
- **Examples:**
  - Current time: <%=new java.util.Date()%>
  - Your hostname: <% = request.getRemoteHost()%>
- **XML-compatible syntax**
  - <jsp:expression>Java Expression</jsp:expression>
  - You cannot mix versions within a single page. You must use XML for *entire* page if you use jsp:expression.

55

Predefined (implicit) Variables

- **Request:**
  - The HttpServletRequest – 1st arg to doGet();
- **Response:**
  - The HttpSerlvetRepsonse – 2nd arg to doGet();
- **session:**
  - The HttpSession associated with the request (unless disabled with the session attribute directive).
- **out:**
  - The stream (of type JspWriter) used to send output to the client.
- **application (javax.servlet.ServletContext):**
  - The ServletContext (for sharing data) as obtained via getServletConfig().getContext().
- **config, jspContext, page (i.e., *this*), pageContext, exception**

56

JSP Scriptlets

- **Format:**
  - <%Java Code%>
- **Result:**
  - Code is inserted verbatim into Servlet's _jspService.
- **Example:**
  - <%String queryData = request.getQueryString();
    out.println("Attached GET data: " + queryData);%>

57

JSP/Servlet Correspondence

- **Original JSP:**

```
<%= foo() %>
<%= bar() %>
<% baz(); %>
```

- **Possible resulting servlet code:**

```
public void _jspService(HttpServletRequest request,
    HttpServletResponse response) throws servletException
    IOException{
  response.setContentType("text/html");
  HttpSession session = request.get Session(true);
 JspWriter out = response.getWriter();
 out.println(foo());
 out.println(bar());
 baz();
…
}
```

58

JSP Declarations

- **Format:**
  - <%! Java Code %>
- **Result:**
  - Code is inserted verbatim into the Servlet's class definition, outside of any existing methods..
- **Examples:**
  - <%! private int someField=5;%>
  - <%! private void someMethod(...) {...} %>
    - Design consideration:
      - Fields are useful. For methods, it is usually better to define the method in a separate Java class.

59

Example Using JSP Declarations

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
    Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>JSP Declarations</TITLE>
  </HEAD>
  <BODY>
    <H1>JSP Declarations</H1>
    <%! private int accessCount = 0; %>
    <H2>Accesses to page since server reboot:
    <%= ++accessCount %></H2>
  </BODY>
</HTML>
```

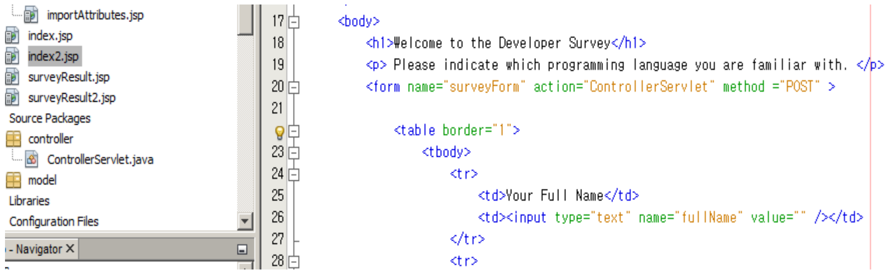
60

Purpose of the Page Directive

- **Will give high-level information about the Servlet that will result from the JSP page.**
- **Can control:**
  - Which classes are imported,
  - What class the Servlet extends,
  - What MIME type is generated,
  - How multi-threading is handled.
  - If the Servlet participates in sessions,
  - The size and behaviour of the output buffer,
  - What page handles unexpected errors.

The Import Attribute

- **Format:**
  - <%@ page import="package.class" %>
  - <%@ page import="paqckage.class1,…, package.classM"%>
- **Purpose:**
  - Generate import statements at top of the Servlet.
- **Notes:**
  - Although JSP pages can be almost anywhere on the server, classes used by JSP pages must be in normal Servlet directories.
    - *Always try to* use packages for utilities that will be used by JSP.



```
importAttributes.jsp
index.jsp
index2.jsp
surveyResult.jsp
surveyResult2.jsp
Source Packages
controller
ControllerServlet.java
model
Libraries
Configuration Files
- Navigator
```

```
<body>
    <h1>Welcome to the Developer Survey</h1>
    <p> Please indicate which programming language you are familiar with. </p>
    <form name="surveyForm" action="ControllerServlet" method ="POST" >

        <table border="1">
            <tbody>
                <tr>
                    <td>Your Full Name</td>
                    <td><input type="text" name="fullName" value="" /></td>
                </tr>
                <tr>
```

The contentType Attribute

- **Format:**

```
<%@ page contentType="MIME-Type"%>
<%@ page contentType="MIME-Type;
                 charset=Character-Set"%>
<%@ page pageEncoding="Character-Set" %>
```

- **Purpose:**
  - Specify the MIME type of the page generated by the Servlet that results from the JSP page.

63

The isThreadSafe Attribute

- **Format:**
  - `<%@ page isThreadSafe="true"%> <%-- Default--%>`
  - `<%@ page isThreadSafe="false"%>`
- **Purpose:**
  - To tell the system when your code is not threadsafe, so that the system can prevent concurrent accesses – instructs the Servlet to implement a SingleThreadModel.
- **Notes:**
  - Default is true – system assumes you have synchronized updates to fields and other shared data.
  - Supplying a value of false can degrade the performance.

64

Non-threadsafe Code

- What is the potential problem with this code?

```
<%! private int idNum = 0;%>
<%
  String UserID = "userID" + idNum;
  out.println("Your ID is " + userID + ".");
  idNum = idNum + 1;
%>
```

65

Thread-safe Code

```
<%! private int idNum = 0;%>
<%
  synchronized(this) {
    String UserID = "userID" + idNum;
    out.println("Your ID is " + userID + ".");
    idNum = idNum + 1;
}
%>
```

- Thread-safe code: access to a shared variable using the normal synchronized statement.

66

Other attributes of the Page Directive

- **session:**
  - Lets you choose NOT to participate in sessions.
- **buffer:**
  - Changes the minimum size of the buffer used by the JspWriter.
- **autoflush:**
  - Requires the developer to explicitly flush the buffer.
- **extends:**
  - Changes the parent class of the generated servlet.
- **errorPage:**
  - Designates a page to handle unplanned errors.
- **isErrorPage:**
  - Declares that page can be used as an error page.

67

JSP and Servlet Interactions

68

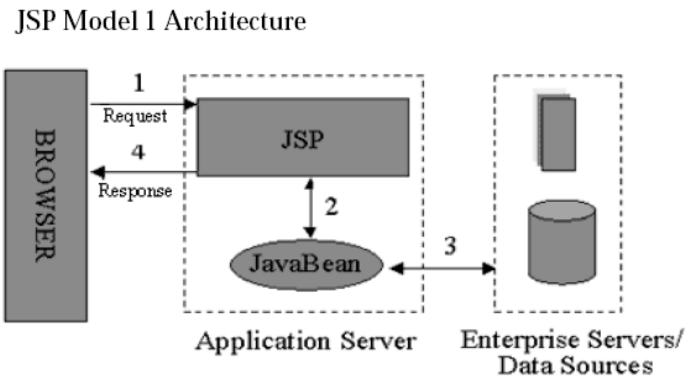
Handling Requests in Web Environments

- **Servlet only works well when**
  - Output is a binary type. (e.g., an image)
  - There is no output handling. (e.g., forwarding or redirection)
  - Format/layout of page is highly variable (e.g., portal service)
  - Low-content, high business-logic situations
- **JSP only works well when**
  - Output is mostly character data. (e.g., HTML)
  - Format/layout mostly fixed (although you can forward or redirect to other pages).
  - High-content, low business-logic situations
- **Combination (MVC architecture) is recommended when**
  - A single request will result in multiple substantially different look and feel results.
  - You have a large development team with different team members doing the Web development and the business logic.
  - You perform complicated data processing, but have a relatively fixed layout.

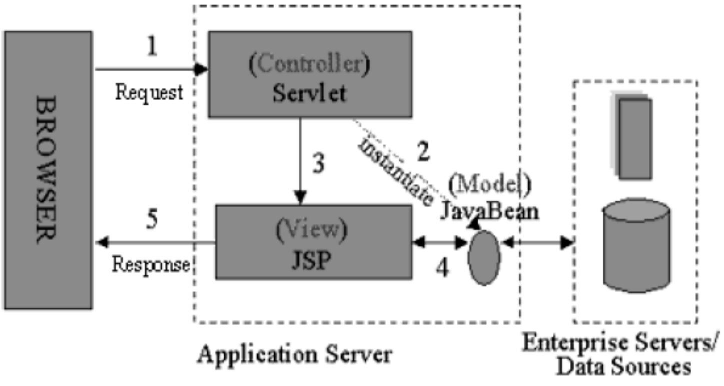
Model-View-Controller

- **Architectural tool**
  - Addresses a common need: Separation of UI and business logic
- **Various tools support the approach**
  - Layered on top of standard web components
  - Struts is an example
- **Enterprise infrastructures and applications evolve**
  - UI and business tiers evolve on different timelines
- **Enterprise development involves multiple parties**
  - Several roles (web developers, software architects, software engineers, database engineers)
  - Many hands in the soup at the same time
- **MVC and multi-tiered applications help keep things manageable in enterprise environment.**

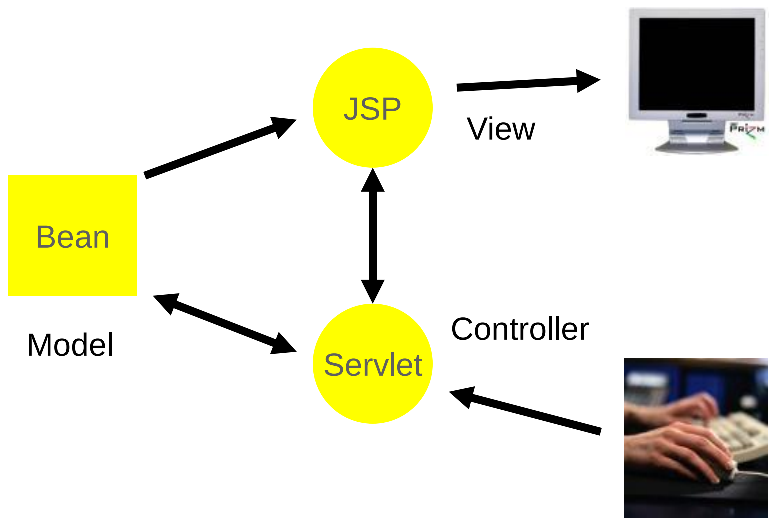
JSP/Servlet Model 1 Architecture



JSP/Servlet Model 2 Architecture



Model-View-Controller (Model 2)



73

Implementing MVC with Request Dispatcher

- **Define beans to represent the data and use a Servlet to handle requests**
  - Servlet reads request parameters, checks for missing and malformed data, etc.
- **Populate the beans**
  - The Servlet invokes business logic (application- specific code) or data-access code to obtain the results. Results are placed in the beans that were defined in step 1.
- **Store the bean in the request, session, or Servlet context**
  - The Servlet calls `setAttribute` on the request, session, or Servlet context objects to store a reference to the beans that represent the results of the request.

74

MVC with Request Dispatcher (cont'd)

- **Forward the request to a JSP page.**
  - The Servlet determines which JSP page is appropriate to the situation and uses the forward() method of RequestDispatcher to transfer control to that page.
- **Extract the data from the Beans.**
  - The JSP page accesses Beans with jsp:useBean and a scope declared.
  - The page then uses jsp:getProperty to use the Bean properties.
  - The JSP page does not create or modify the bean; it merely extracts and displays data that the Servlet created.
  - JSP page does not create all the data objects (Beans). To guarantee that the JSP page will not create Beans you should use
    - <jsp:useBean ... type="package.Class" />
    - instead of
    - <jsp:useBean ... class="package.Class" />
    - you should use jsp:getProperty but not jsp:setProperty.

75

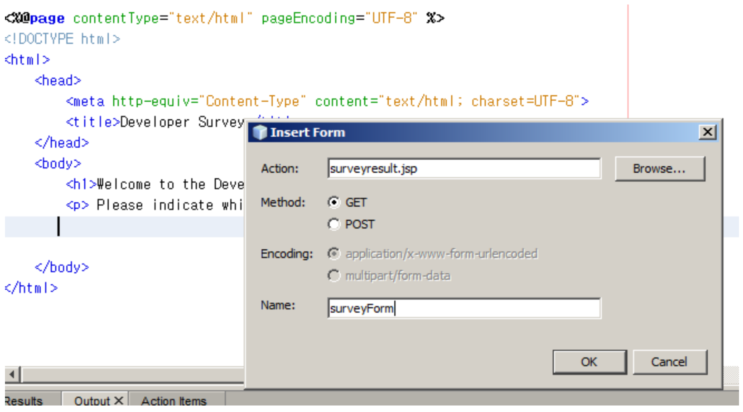
Request Forwarding Example

```
protected void processRequest(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
    String operation = request.getParameter("operation");
    if (operation == null) {
            operation = "unknown";
    }
    String address;
    if (operation.equals("order")) {
            address = "/Order.jsp";
    }   else if (operation.equals("cancel")) {
            address = "/Cancel.jsp";
    }   else {
            address = "/UnknownOperation.jsp";
    }
    RequestDispatcher dispatcher = request.getRequestDispatcher(address);
    dispatcher.forward(request, response);
```

76

SimpleJSPs MVC example

- Survey App (Index2.jsp, Controller servlet, surveyResult2.jsp)
- Form element of the NetBeans Palette



```
<%@page contentType="text/html" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Developer Survey
    </head>
    <body>
        <h1>Welcome to the Deve
        <p> Please indicate whi

    </body>
</html>
```

Insert Form

Action: surveyresult.jsp

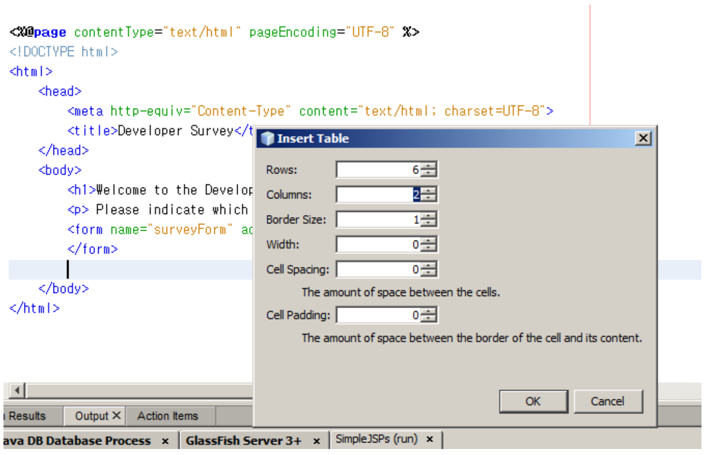
Method: [✓ GET] POST

Encoding: [✓ application/x-www-form-urlencoded] multipart/form-data

Name: surveyForm

77

Table Element



```
<%@page contentType="text/html" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Developer Survey</t
    </head>
    <body>
        <h1>Welcome to the Develop
        <p> Please indicate which
        <form name="surveyForm" ac
        </form>

    </body>
</html>
```

Insert Table

Rows: 6

Columns: 2

Border Size: 1

Width: 0

Cell Spacing: 0

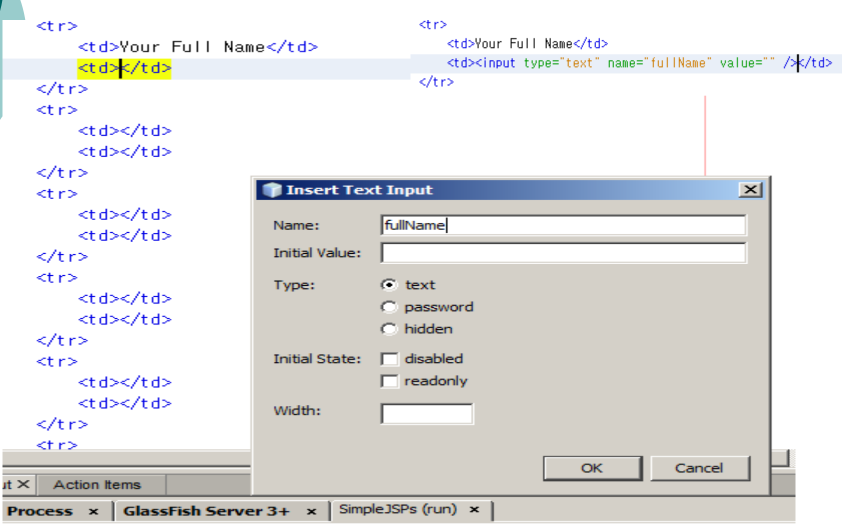
The amount of space between the cells.

Cell Padding: 0

The amount of space between the border of the cell and its content.

78

Text element (NetBeans Pallette)



```
<tr>
    <td>Your Full Name</td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>
</tr>
<tr>
```

```
<tr>
    <td>Your Full Name</td>
    <td><input type="text" name="fullName" value="" /></td>
</tr>
```

Insert Text Input

Name: fullName

Initial Value:

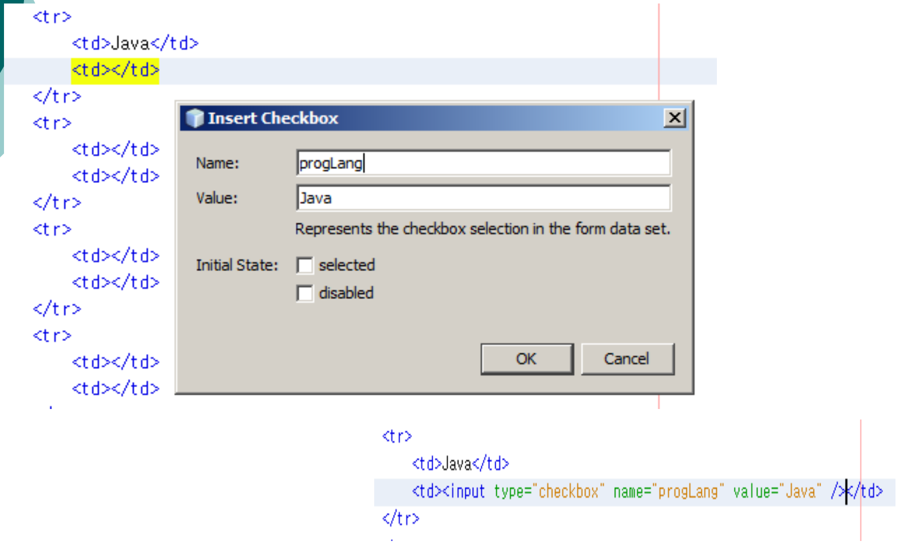
Type: [✓ text] password hidden

Initial State: ☐ disabled ☐ readonly

Width:

OK Cancel

Ckeckbox Element



```
<tr>
    <td>Java</td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>
```

Insert Checkbox

Name: progLang

Value: Java

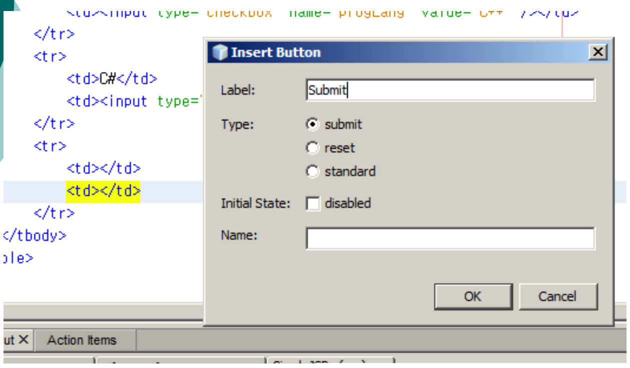
Represents the checkbox selection in the form data set.

Initial State: ☐ selected ☐ disabled

OK Cancel

```
<tr>
    <td>Java</td>
    <td><input type="checkbox" name="progLang" value="Java" /></td>
</tr>
```

Button Element



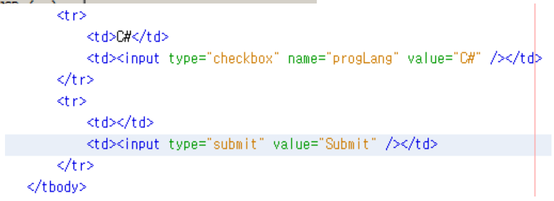
Insert Button

Label: Submit

Type: [✓ submit] reset standard

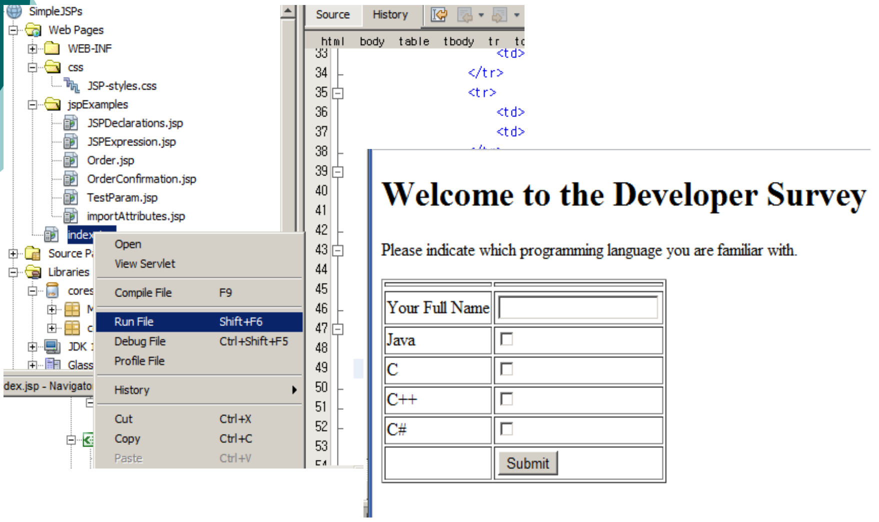
Initial State: disabled

Name:



```
<tr>
    <td>C#</td>
    <td><input type="checkbox" name="progLang" value="C#" /></td>
</tr>
<tr>
    <td></td>
    <td><input type="submit" value="Submit" /></td>
</tr>
</tbody>
```

Run File – index2.jsp

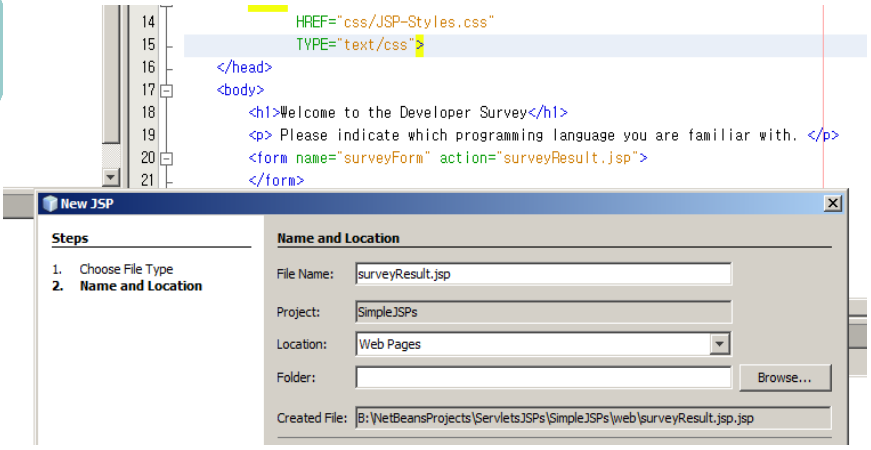


Welcome to the Developer Survey

Please indicate which programming language you are familiar with.

Your Full Name	
Java	☐
C	☐
C++	☐
C#	☐
	Submit

surveyResul2.jsp



```
            HREF="css/JSP-Styles.css"
            TYPE="text/css">
    </head>
    <body>
        <h1>Welcome to the Developer Survey</h1>
        <p> Please indicate which programming language you are familiar with. </p>
        <form name="surveyForm" action="surveyResult.jsp">
        </form>
```

New JSP

Steps

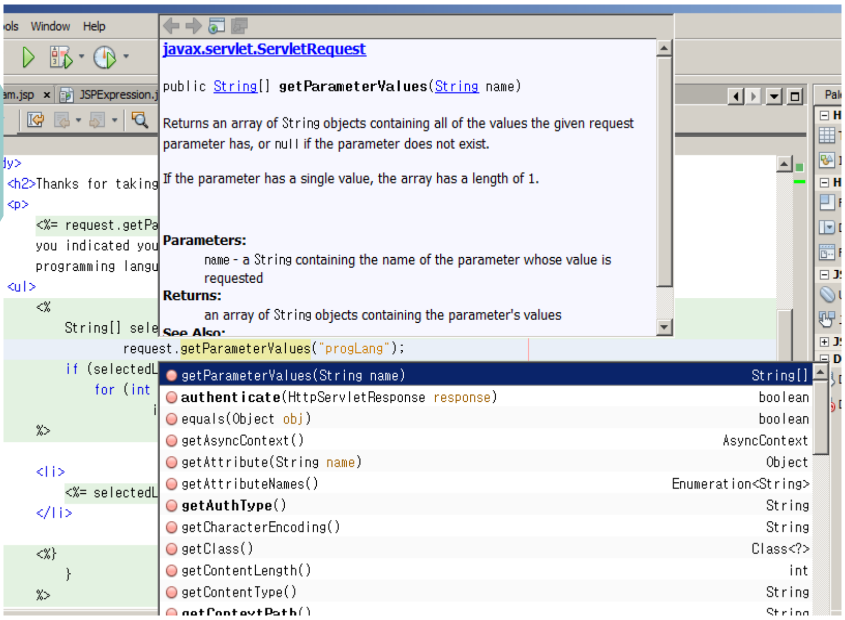
1. Choose File Type
2. **Name and Location**

Name and Location

File Name:	surveyResult.jsp
Project:	SimpleJSPs
Location:	Web Pages
Folder:	
Created File:	B:\NetBeansProjects\ServletsJSPs\SimpleJSPs\web\surveyResult.jsp.jsp

Browse...

83



```
dy>
<h2>Thanks for taking
<p>
    <%= request.getPa
    you indicated you
    programming langu
<ul>
    <%
        String[] sele
            request.getParameterValues("progLang");
        if (selectedL
            for (int
                i
    %>
    <li>
        <%= selectedL
    </li>
    <%}
        }
    %>
```

javax.servlet.ServletRequest

public String[] **getParameterValues**(String name)

Returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist.

If the parameter has a single value, the array has a length of 1.

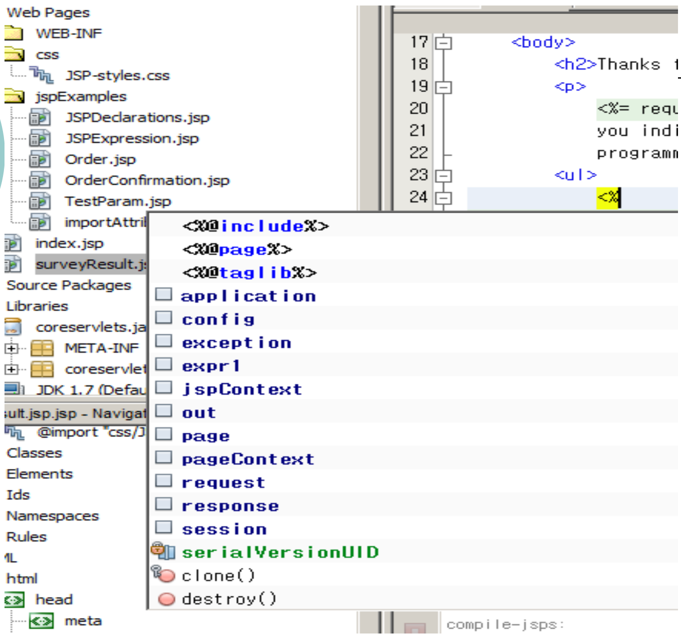
Parameters:
name - a String containing the name of the parameter whose value is requested

Returns:
an array of String objects containing the parameter's values

See Also:

```
getParameterValues(String name)                              String[]
authenticate(HttpServletResponse response)                    boolean
equals(Object obj)                                            boolean
getAsyncContext()                                        AsyncContext
getAttribute(String name)                                      Object
getAttributeNames()                               Enumeration<String>
getAuthType()                                                  String
getCharacterEncoding()                                         String
getClass()                                                   Class<?>
getContentLength()                                                int
getContentType()                                               String
getContextPath()                                               String
```

84



Web Pages
- WEB-INF
- css
  - JSP-styles.css
- jspExamples
  - JSPDeclarations.jsp
  - JSPExpression.jsp
  - Order.jsp
  - OrderConfirmation.jsp
  - TestParam.jsp
  - importAttril
- index.jsp
- surveyResult.js

Source Packages
Libraries
- coreservlets.ja
- META-INF
- coreservlet
- JDK 1.7 (Defau

sult.jsp.jsp - Navigat
- @import "css/J

Classes
Elements
Ids
Namespaces
Rules
ML
html
- head
  - meta

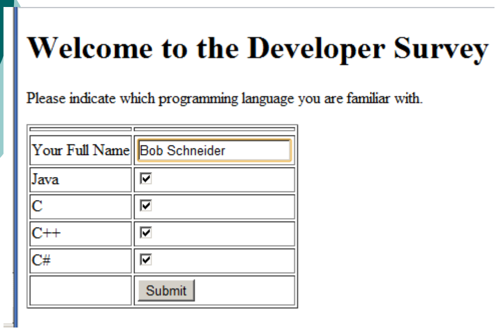
```
        <body>
            <h2>Thanks f
            <p>
                <%= requ
                you indi
                programm
            <ul>
                <%
```

```
<%@include%>
<%@page%>
<%@taglib%>
application
config
exception
expr1
jspContext
out
page
pageContext
request
response
session
serialVersionUID
clone()
destroy()
```

compile-jsps:

85

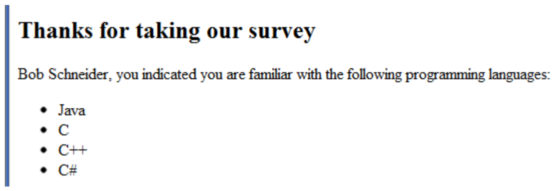
Developer App



Welcome to the Developer Survey

Please indicate which programming language you are familiar with.

Your Full Name	Bob Schneider
Java	☑
C	☑
C++	☑
C#	☑
	Submit



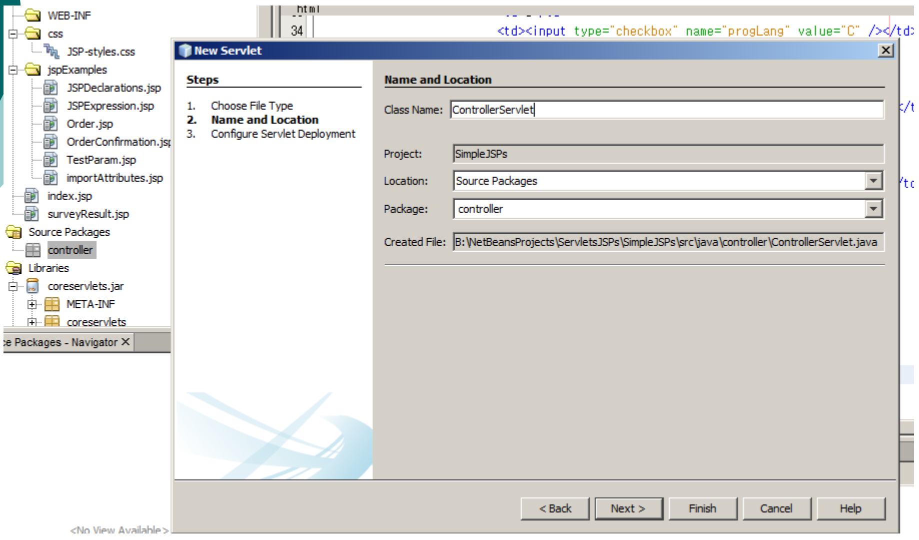
Thanks for taking our survey

Bob Schneider, you indicated you are familiar with the following programming languages:

- Java
- C
- C++
- C#

86

Adding Controller Servlet (MVC)



```
34   <td><input type="checkbox" name="progLang" value="C" /></td>
```

Name and Location

Class Name: ControllerServlet

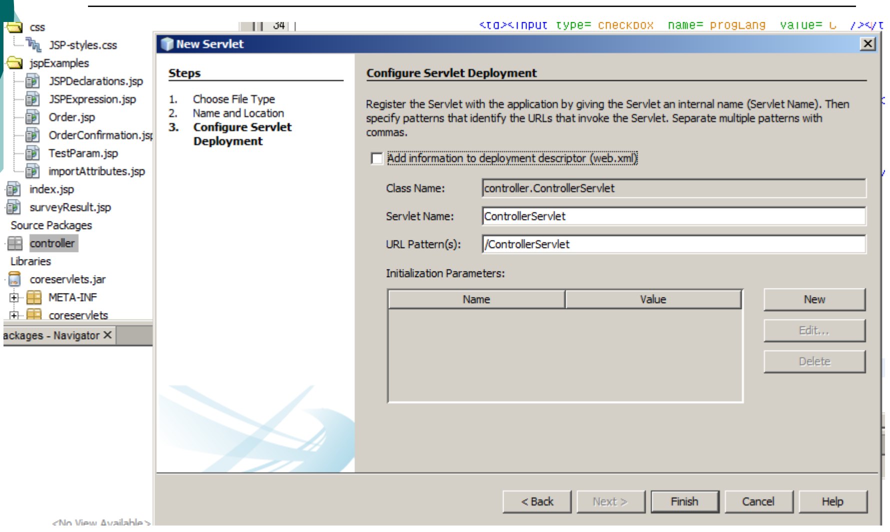
Project: SimpleJSPs

Location: Source Packages

Package: controller

Created File: B:\NetBeansProjects\ServletsJSPs\SimpleJSPs\src\java\controller\ControllerServlet.java

87



Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☐ Add information to deployment descriptor (web.xml)

Class Name: controller.ControllerServlet

Servlet Name: ControllerServlet

URL Pattern(s): /ControllerServlet

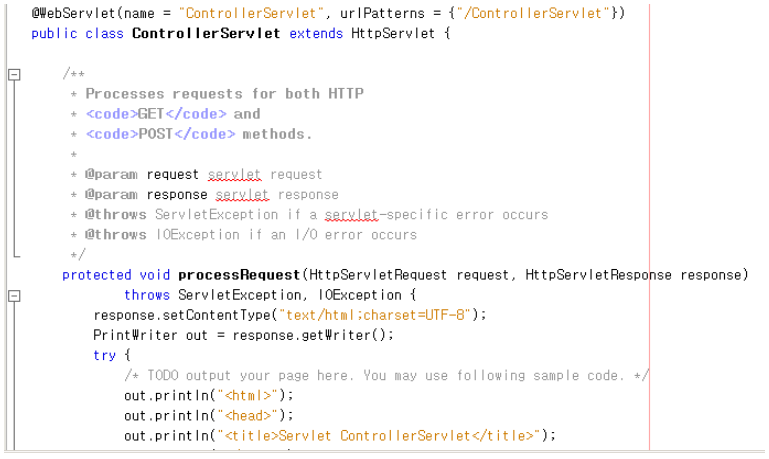
Initialization Parameters:

Name	Value

88

WebServlet annotation

- @WebServlet annotation by default when generating servlets



```
@WebServlet(name = "ControllerServlet", urlPatterns = {"/ControllerServlet"})
public class ControllerServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            /* TODO output your page here. You may use following sample code. */
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet ControllerServlet</title>");
```

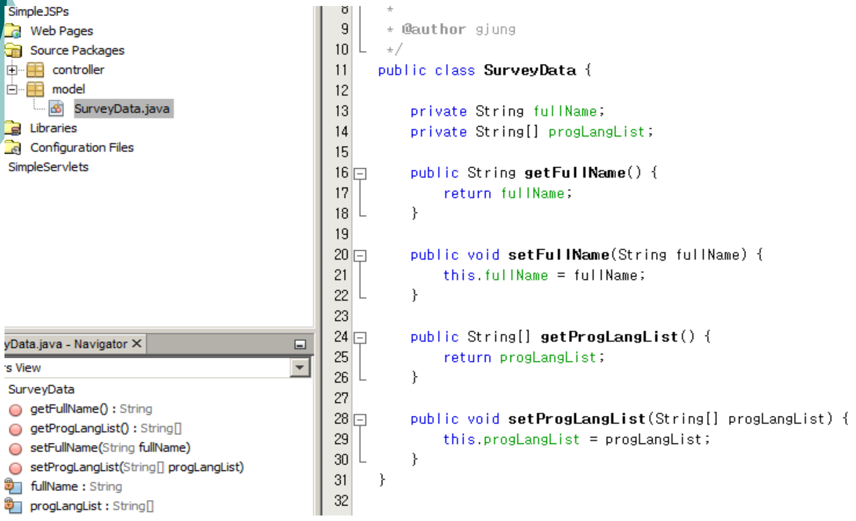
89

processRequest() method

- The generated servlet contains a processRequest() method that will be executed every time the servlet receives an HTTP GET or an HTTP POST request from the browser.
- This method takes an instance of javax.servlet.http.HttpServletRequest and an instance of javax.servlet.http.HttpServletResponse as parameters. These parameters are equivalent to the request and response implicit objects in JSPs.
- The processRequest() method is a NetBeans specific method that is generated (in most cases the servlet executes the same code regardless GET or POST method from the browser).
- doGet() doPost() methods handle GET or POST request, respectively.

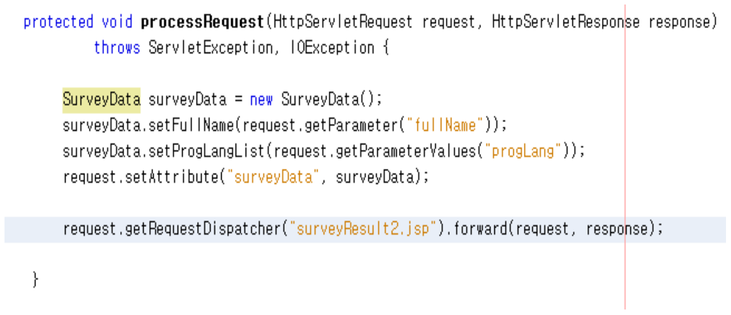
90

Add model (data, JavaBean)



```
 *
 * @author gjung
 */
public class SurveyData {

    private String fullName;
    private String[] progLangList;

    public String getFullName() {
        return fullName;
    }

    public void setFullName(String fullName) {
        this.fullName = fullName;
    }

    public String[] getProgLangList() {
        return progLangList;
    }

    public void setProgLangList(String[] progLangList) {
        this.progLangList = progLangList;
    }
}
```

Using Model (Data)

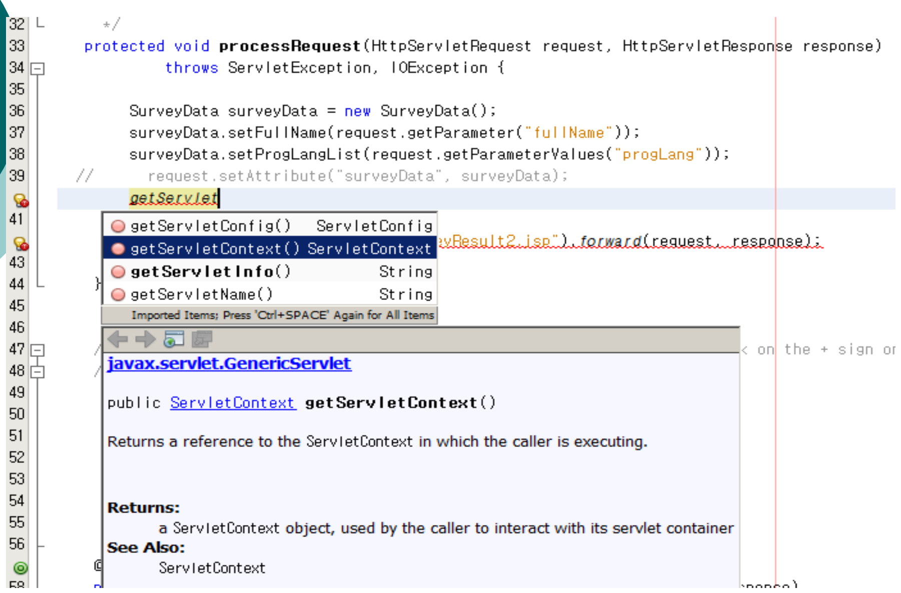


```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    SurveyData surveyData = new SurveyData();
    surveyData.setFullName(request.getParameter("fullName"));
    surveyData.setProgLangList(request.getParameterValues("progLang"));
    request.setAttribute("surveyData", surveyData);

    request.getRequestDispatcher("surveyResult2.jsp").forward(request, response);

}
```

Scope of the Bean (model)

- Objects can be stored by a servlet as attributes at the session or application scope.
  - Request.getSession().setAttribute(“SurveyData”, surveyData);
  - getSession() method of the javax.servlet.http.HttpServletRequest interface returns an instance of javax.servlet.http.HttpSession representing the user’s session.
  - Session attributes are visible to all pages in a user’s session and are preserved across requests.
  - getServletContext() method is defined in javax.servlet.GenericSurvlet, which is the parent class of javax.servlet.http.HttpServlet, that is in turn is the parent class of every servlet in a web application. This method returns an instance of javax.servlet.ServletContext.
    - Storing an object as an attribute of the servlet context makes it visible across user sessions; therefore all users in the application have access to the attribute.

```
<jsp:useBean id="surveyData" scope="application" class="model.SurveyData" />
<head>
```



```
   */
  protected void processRequest(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {

      SurveyData surveyData = new SurveyData();
      surveyData.setFullName(request.getParameter("fullName"));
      surveyData.setProgLangList(request.getParameterValues("progLang"));
//      request.setAttribute("surveyData", surveyData);
      getServlet
```

```
getServletConfig()    ServletConfig
getServletContext()   ServletContext
getServletInfo()      String
getServletName()      String
Imported Items; Press 'Ctrl+SPACE' Again for All Items
```

javax.servlet.GenericServlet

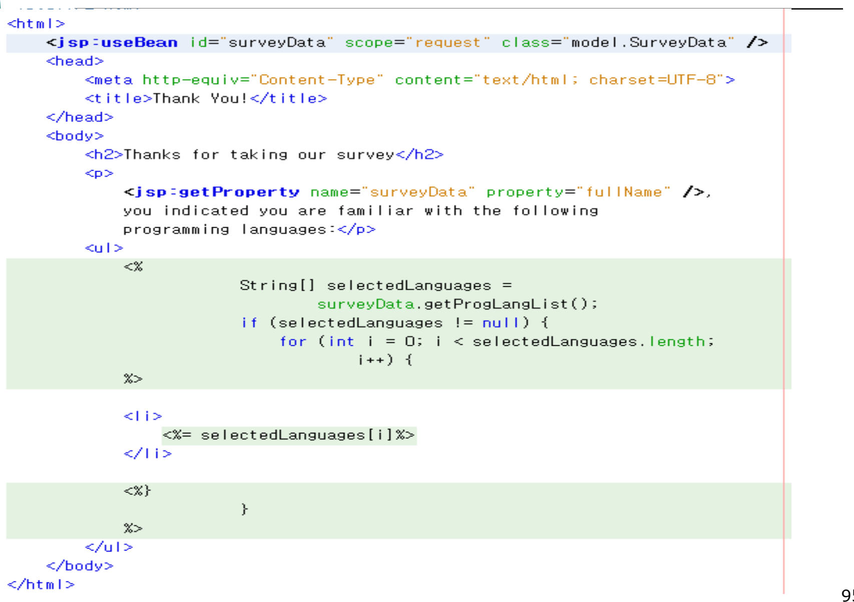
```
public ServletContext getServletContext()
```

Returns a reference to the ServletContext in which the caller is executing.

Returns:
a ServletContext object, used by the caller to interact with its servlet container

See Also:
ServletContext

SurveyResult2.jsp



```
<html>
    <jsp:useBean id="surveyData" scope="request" class="model.SurveyData" />
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Thank You!</title>
    </head>
    <body>
        <h2>Thanks for taking our survey</h2>
        <p>
            <jsp:getProperty name="surveyData" property="fullName" />,
            you indicated you are familiar with the following
            programming languages:</p>
        <ul>
            <%
                        String[] selectedLanguages =
                                surveyData.getProgLangList();
                        if (selectedLanguages != null) {
                            for (int i = 0; i < selectedLanguages.length;
                                    i++) {
            %>

            <li>
                <%= selectedLanguages[i]%>
            </li>

            <%}
                        }
            %>
        </ul>
    </body>
</html>
```

JSP and Servlet Interactions

Securing Web Applications

Examples are in SecureWebApp NB Project

Securing Web Apps

- **It is a common requirement to only allow certain users to access certain pages in a web application.**
  - Requires security realm set up
  - Each security realm allows the application server to obtain security information from some sort of permanent storage (i.e., file, relational database, LDAP repository, or any kind of persistent storage)
    - Developers not to worry about the specific implementation.
    - Simply configure the application to use a defined security realm for authentication
  - Setting up security realm varies from AS to AS.
- **Four different ways to authenticate a user**
  - Basic authentication (browser pop up window based)
  - Digest authentication (~~ BA, password is encrypted)
  - Client side certificate (issued by certificate authorities such as Verisign or Thawte): not very common due to the expense and lack of convenience of issuing client-side certificates
  - Form-based authentication: most common, need to develop a JSP or HTML page used to collect user credentials, use HTTPS (HTTP over SSL)

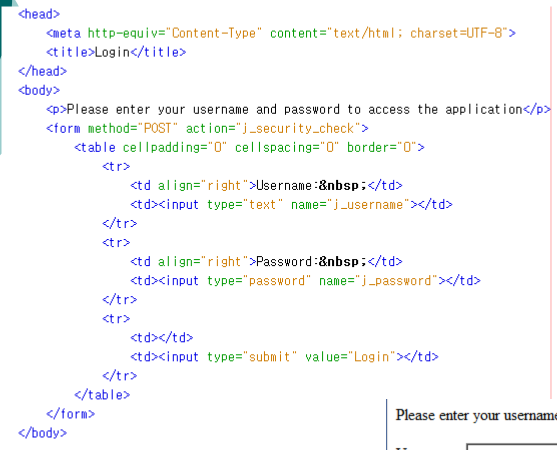
97

Implementing Form-based Authentication

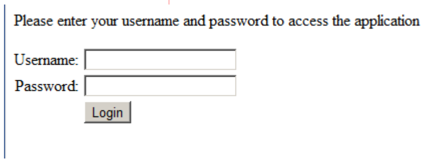
- **A login page needs to be created**
  - Every login page created for form based authentication must contain an HTML form with a method POST and an action of j_security_check
  - Every EE-compliant AS has a security servlet deployed on installation, the security servlet is mapped to the j_security_check
  - URL, as such, its doPost() method is executed when the form is submitted.
  - Login page must have j_username and j_password (used by the security servlet to check these values match those in security realm)
- **A login error page needs to be created, this page will be displayed when a user enters incorrect credentials**
- **The web app needs to be configured to use a security realm for authentication**

98

SecureWebApp –login.jsp



```
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Login</title>
</head>
<body>
    <p>Please enter your username and password to access the application</p>
    <form method="POST" action="j_security_check">
        <table cellpadding="0" cellspacing="0" border="0">
            <tr>
                <td align="right">Username:&nbsp;</td>
                <td><input type="text" name="j_username"></td>
            </tr>
            <tr>
                <td align="right">Password:&nbsp;</td>
                <td><input type="password" name="j_password"></td>
            </tr>
            <tr>
                <td></td>
                <td><input type="submit" value="Login"></td>
            </tr>
        </table>
    </form>
</body>
```



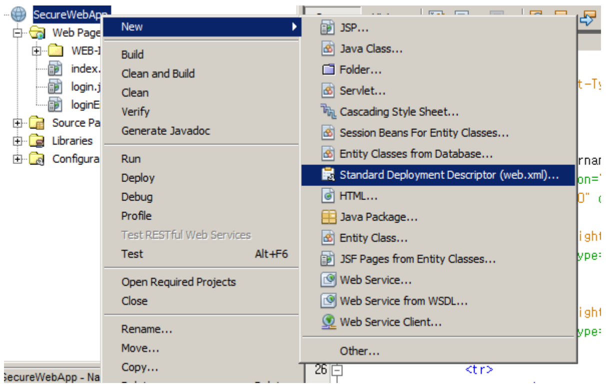
Please enter your username and password to access the application

Username:

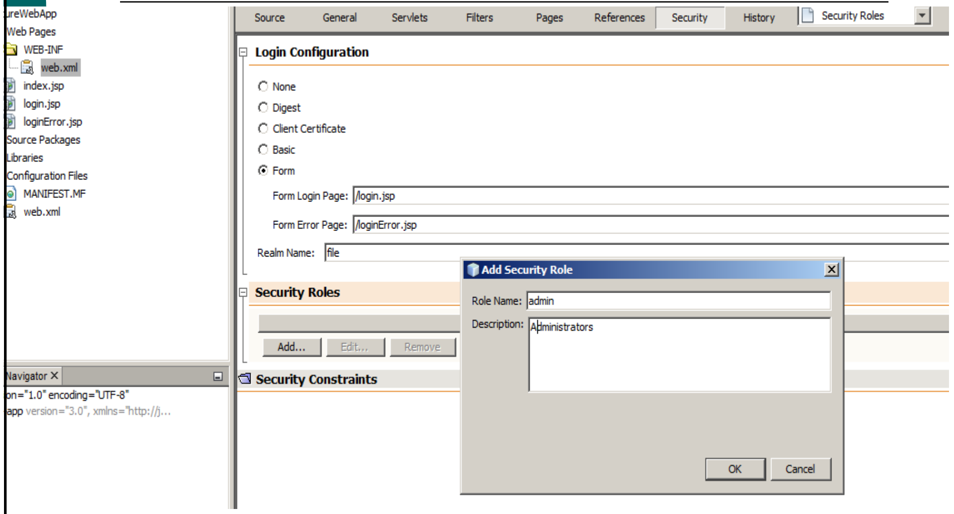
Password:

Login

Configuring Web app for form-based AU



Add Security Role



Source | General | Servlets | Filters | Pages | References | Security | History | Security Roles

Login Configuration

- None
- Digest
- Client Certificate
- Basic
- [✓ Form]

Form Login Page: /login.jsp

Form Error Page: /loginError.jsp

Realm Name: file

Security Roles

Security Constraints

Add Security Role

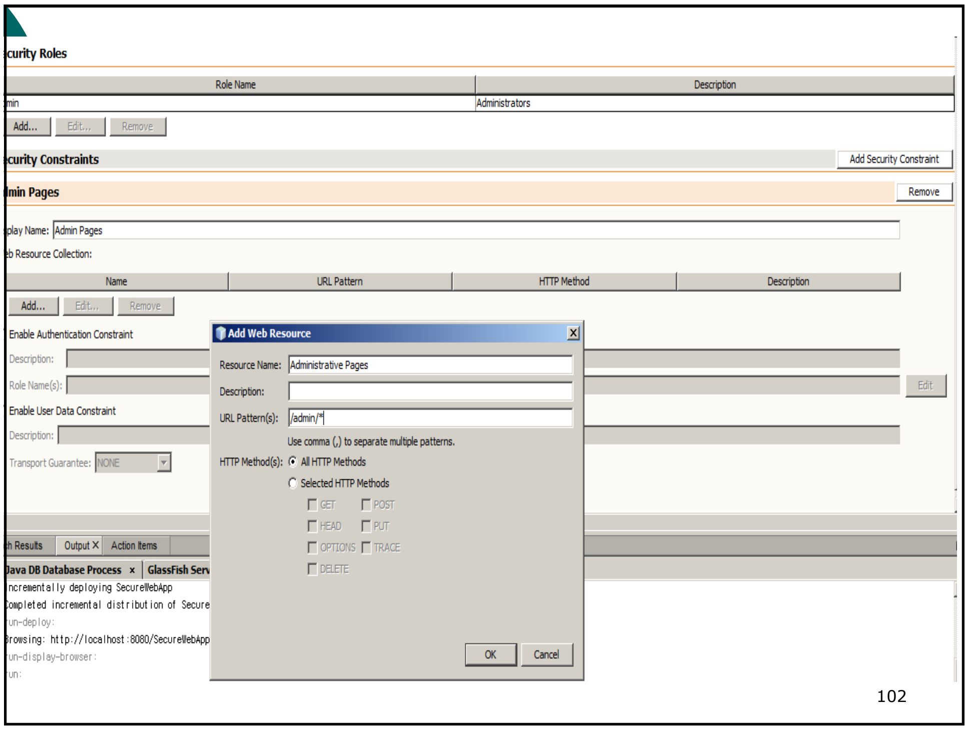
Role Name: admin

Description: Administrators

OK | Cancel

```
on="1.0" encoding="UTF-8"
app version="3.0", xmlns="http://j...
```

101



curity Roles

Role Name	Description
min	Administrators

curity Constraints

lmin Pages

play Name: Admin Pages

b Resource Collection:

Name	URL Pattern	HTTP Method	Description

Enable Authentication Constraint

Enable User Data Constraint

Transport Guarantee: NONE

Add Web Resource

Resource Name: Administrative Pages

Description:

URL Pattern(s): /admin/*

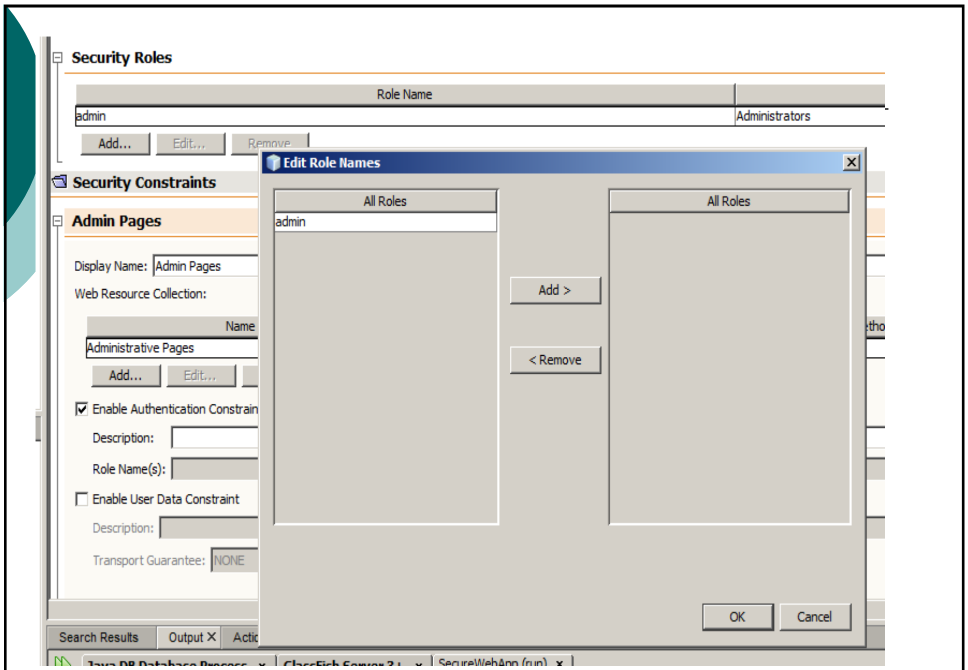
Use comma (,) to separate multiple patterns.

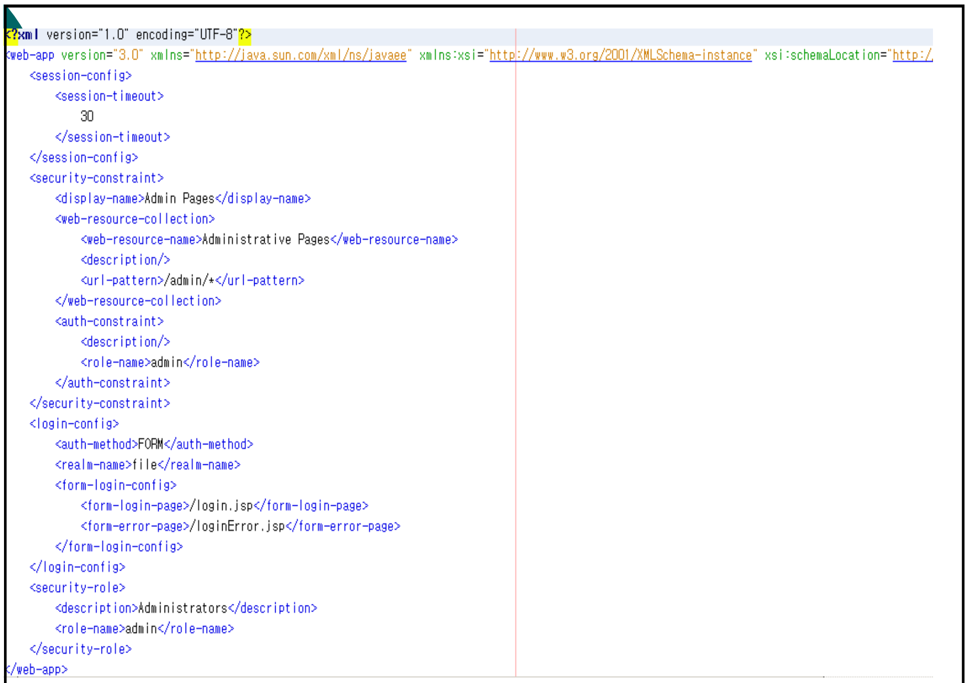
HTTP Method(s): [✓ All HTTP Methods] Selected HTTP Methods

OK | Cancel

```
ncrementally deploying SecureWebApp
ompleted incremental distribution of Secure
un-deploy:
rowsing: http://localhost:8080/SecureWebApp
un-display-browser:
un:
```

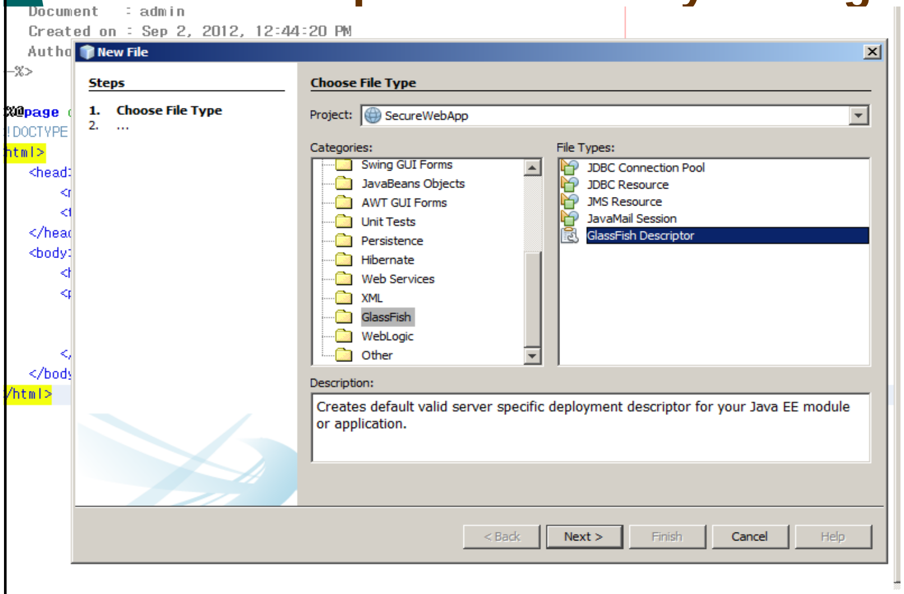
102





```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http:/
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <security-constraint>
        <display-name>Admin Pages</display-name>
        <web-resource-collection>
            <web-resource-name>Administrative Pages</web-resource-name>
            <description/>
            <url-pattern>/admin/*</url-pattern>
        </web-resource-collection>
        <auth-constraint>
            <description/>
            <role-name>admin</role-name>
        </auth-constraint>
    </security-constraint>
    <login-config>
        <auth-method>FORM</auth-method>
        <realm-name>file</realm-name>
        <form-login-config>
            <form-login-page>/login.jsp</form-login-page>
            <form-error-page>/loginError.jsp</form-error-page>
        </form-login-config>
    </login-config>
    <security-role>
        <description>Administrators</description>
        <role-name>admin</role-name>
    </security-role>
</web-app>
```

GlassFish Specific Security Config


Document : admin
Created on : Sep 2, 2012, 12:44:20 PM

New File

Steps
1. Choose File Type
2. ...

Choose File Type

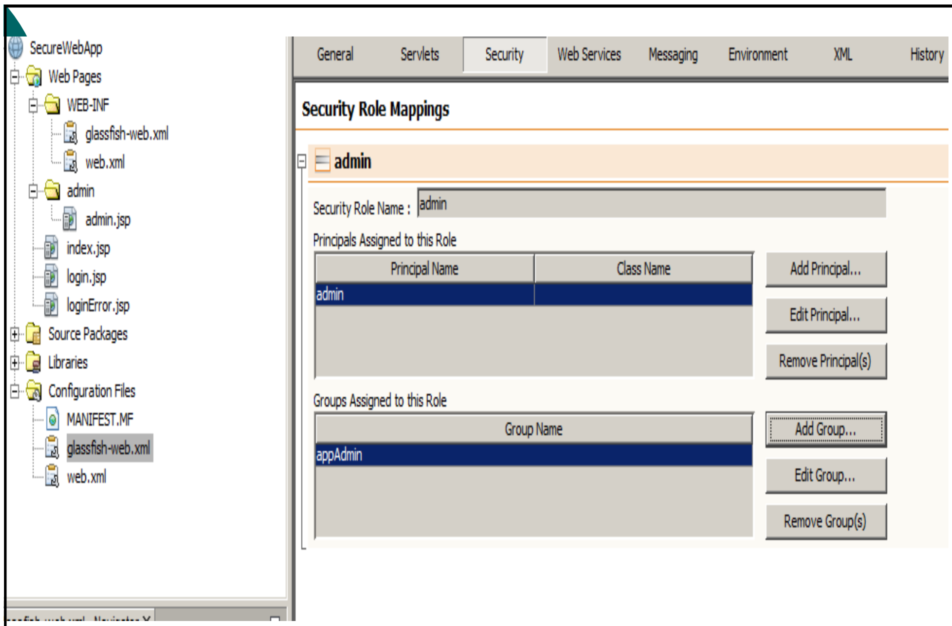
Project: SecureWebApp

Categories:
- Swing GUI Forms
- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- Persistence
- Hibernate
- Web Services
- XML
- GlassFish
- WebLogic
- Other

File Types:
- JDBC Connection Pool
- JDBC Resource
- JMS Resource
- JavaMail Session
- GlassFish Descriptor

Description:
Creates default valid server specific deployment descriptor for your Java EE module or application.

105


```
SecureWebApp
  Web Pages
    WEB-INF
      glassfish-web.xml
      web.xml
    admin
      admin.jsp
    index.jsp
    login.jsp
    loginError.jsp
  Source Packages
  Libraries
  Configuration Files
    MANIFEST.MF
    glassfish-web.xml
    web.xml
```

Security Role Mappings

admin

Security Role Name : admin

Principals Assigned to this Role

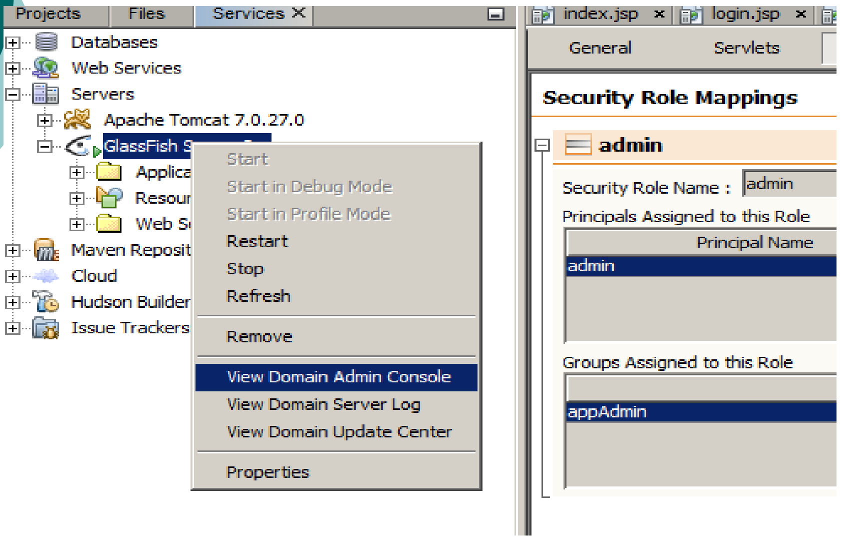
Principal Name	Class Name
admin	

Groups Assigned to this Role

Group Name
appAdmin

106

Configure GF AS Security Realm



Security Role Mappings

admin

Security Role Name : admin

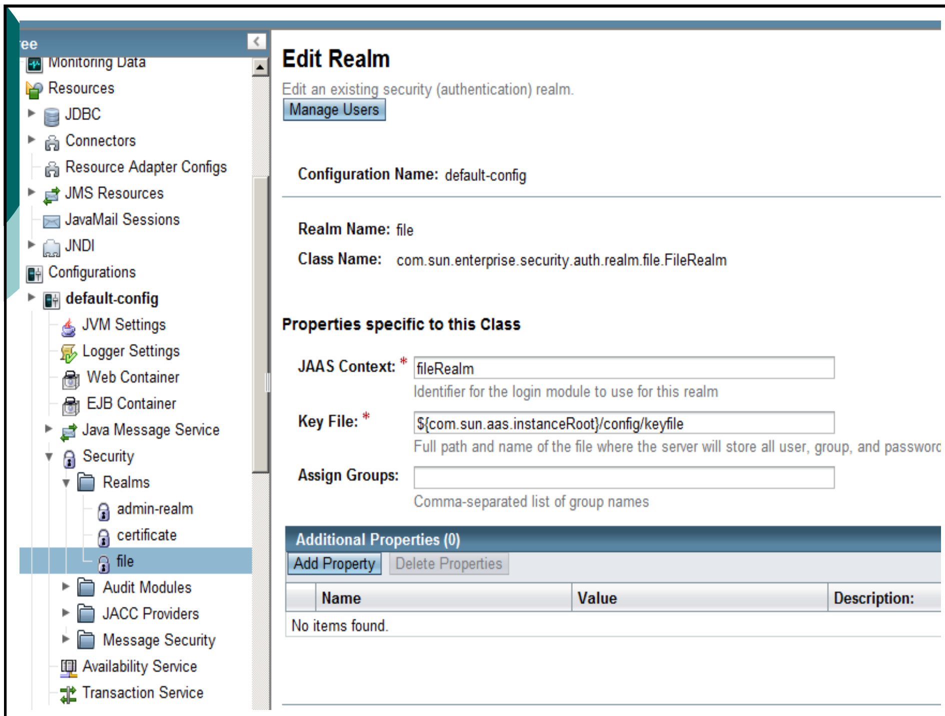
Principals Assigned to this Role

Principal Name
admin

Groups Assigned to this Role

appAdmin

107



Edit Realm

Edit an existing security (authentication) realm.

Manage Users

Configuration Name: default-config

Realm Name: file

Class Name: com.sun.enterprise.security.auth.realm.file.FileRealm

Properties specific to this Class

JAAS Context: * fileRealm

Identifier for the login module to use for this realm

Key File: * ${com.sun.aas.instanceRoot}/config/keyfile

Full path and name of the file where the server will store all user, group, and passworc

Assign Groups:

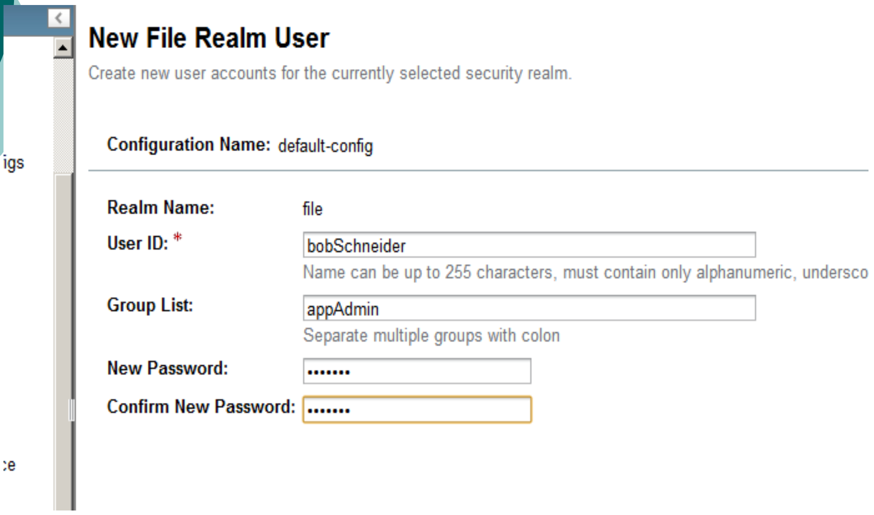
Comma-separated list of group names

Additional Properties (0)

Add Property Delete Properties

Name	Value	Description:
No items found.		

New File Realm User



New File Realm User

Create new user accounts for the currently selected security realm.

Configuration Name: default-config

Realm Name: file

User ID: * bobSchneider

Name can be up to 255 characters, must contain only alphanumeric, undersco

Group List: appAdmin

Separate multiple groups with colon

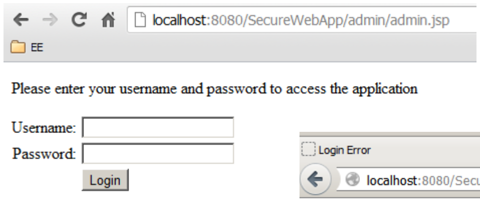
New Password: •••••••

Confirm New Password: •••••••

109

Run SecureWebApp

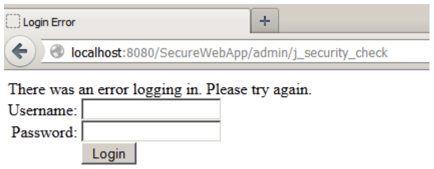
- After deploying the App and accessing admin/admin.jsp
  - The user is automatically directed to the App's login page



localhost:8080/SecureWebApp/admin/admin.jsp

Please enter your username and password to access the application

Username:
Password:
Login

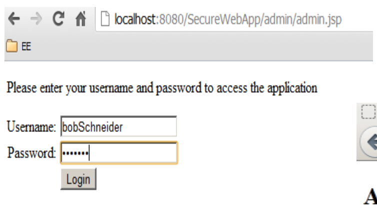


Login Error

localhost:8080/SecureWebApp/admin/j_security_check

There was an error logging in. Please try again.

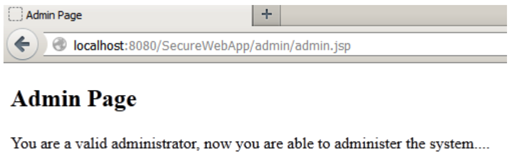
Username:
Password:
Login



localhost:8080/SecureWebApp/admin/admin.jsp

Please enter your username and password to access the application

Username: bobSchneider
Password: ••••••
Login



Admin Page

localhost:8080/SecureWebApp/admin/admin.jsp

Admin Page

You are a valid administrator, now you are able to administer the system....

110

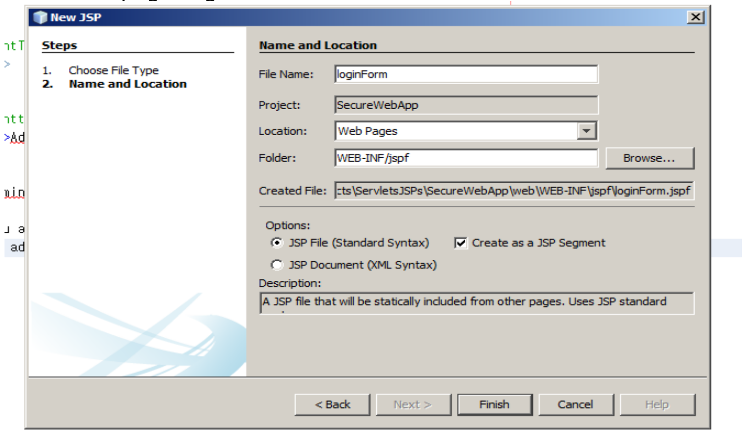
JSP and Servlet Interactions

JSP Fragments

Examples are in SecureWebApp NB Project

JSP fragments

- **In a typical web application, most pages share certain common web page areas such as a navigation menu, a header, a footer, etc.**
  - Create JSP fragments for common page areas, which can be included in every page.
  - JSP fragments need to be updated when there are some changes on those common page segments.



Name and Location

File Name: loginForm

Project: SecureWebApp

Location: Web Pages

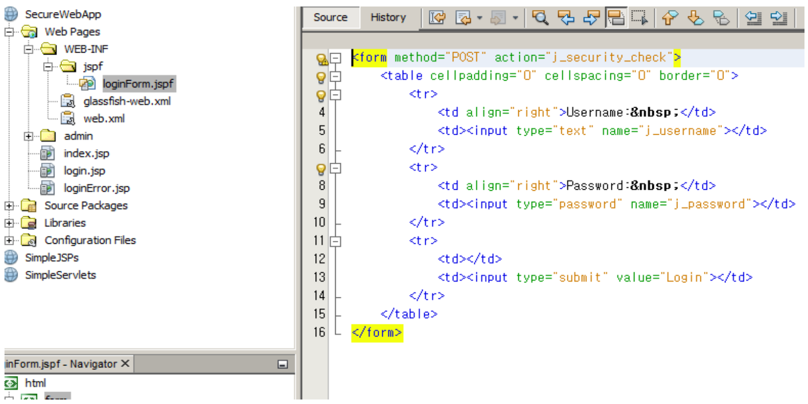
Folder: WEB-INF/jspf

Created File: :ts\ServletsJSPs\SecureWebApp\web\WEB-INF\jspf\loginForm.jspf

Options:
[✓ JSP File (Standard Syntax)] [✓ Create as a JSP Segment]
JSP Document (XML Syntax)

Description:
A JSP file that will be statically included from other pages. Uses JSP standard

SecureWebApp- loginForm.jspf

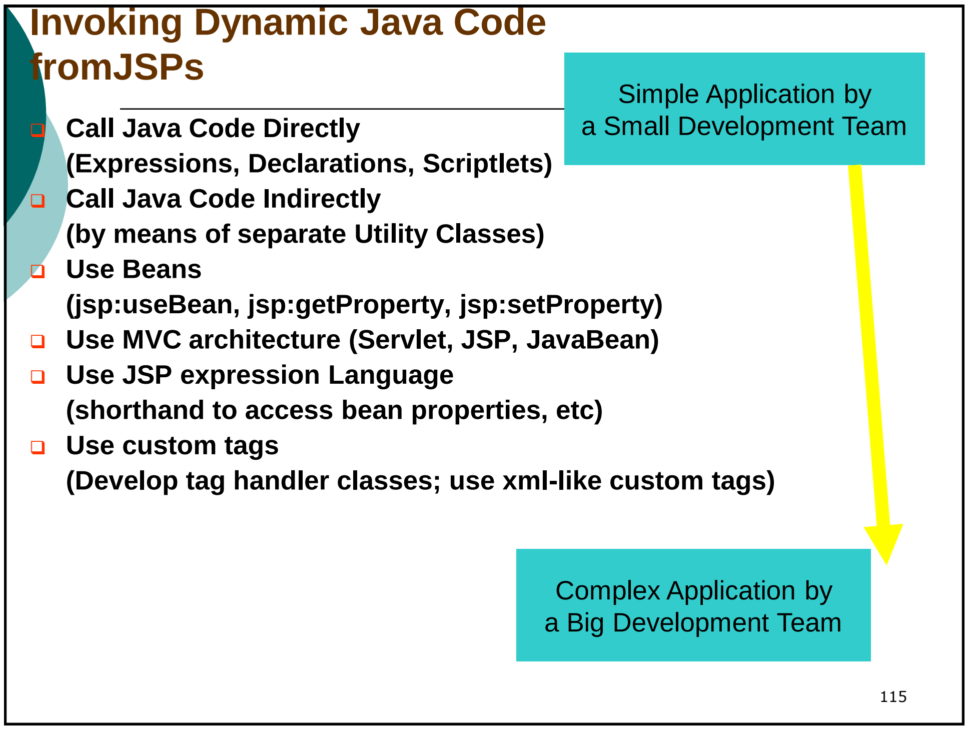


```
SecureWebApp
  Web Pages
    WEB-INF
      jspf
        loginForm.jspf
      glassfish-web.xml
      web.xml
    admin
    index.jsp
    login.jsp
    loginError.jsp
  Source Packages
  Libraries
  Configuration Files
SimpleJSPs
SimpleServlets
```

```
<form method="POST" action="j_security_check">
    <table cellpadding="0" cellspacing="0" border="0">
        <tr>
            <td align="right">Username:&nbsp;</td>
            <td><input type="text" name="j_username"></td>
        </tr>
        <tr>
            <td align="right">Password:&nbsp;</td>
            <td><input type="password" name="j_password"></td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Login"></td>
        </tr>
    </table>
</form>
```

```
    <body>
        <p>Please enter your username and password to access the application</p>
        <%@include file="WEB-INF/jspf/loginForm.jspf" %>
     </body>
</html>
```

JSP Scripting Elements Revisited



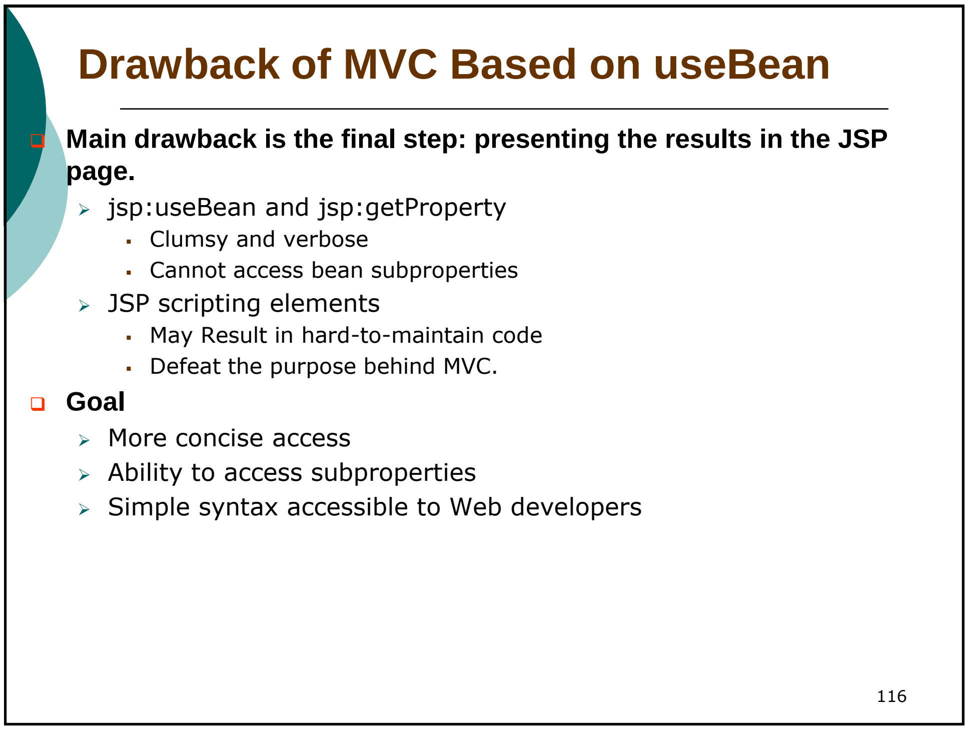
Invoking Dynamic Java Code fromJSPs

- Call Java Code Directly
  (Expressions, Declarations, Scriptlets)
- Call Java Code Indirectly
  (by means of separate Utility Classes)
- Use Beans
  (jsp:useBean, jsp:getProperty, jsp:setProperty)
- Use MVC architecture (Servlet, JSP, JavaBean)
- Use JSP expression Language
  (shorthand to access bean properties, etc)
- Use custom tags
  (Develop tag handler classes; use xml-like custom tags)

Simple Application by
a Small Development Team

Complex Application by
a Big Development Team

115



Drawback of MVC Based on useBean

- **Main drawback is the final step: presenting the results in the JSP page.**
  - jsp:useBean and jsp:getProperty
    - Clumsy and verbose
    - Cannot access bean subproperties
  - JSP scripting elements
    - May Result in hard-to-maintain code
    - Defeat the purpose behind MVC.
- **Goal**
  - More concise access
  - Ability to access subproperties
  - Simple syntax accessible to Web developers

116

JSP and Servlet Interactions

By Expression Language

Examples are in JSPEL NB Project

Advantages of Expression Language

- **Concise access to stored objects.**
  - To output a "scoped variable" (object stored with setAttribute in the PageContext, HttpServletRequest, HttpSession, or ServletContext) named saleItem, you use ${saleItem}.
- **Shorthand notation for bean properties.**

  Examples:
  - To output the companyName property (i.e., result of the getCompanyName() method) of a scoped variable named company, you use ${company.companyName}.
  - To access the firstName property of the president property of a scoped variable named company, you use ${company.president.firstName}.
- **Simple access to collection elements.**
  - To access an element of an array, List, or Map, you use ${variable[indexOrKey]}. Provided that the index or key is in a form that is legal for Java variable names.

Advantages of EL (cont'd)

- **Succinct access to request parameters, cookies, and other request data.**
  - To access the standard types of request data, you can use one of several predefined implicit objects.
- **A small but useful set of simple operators.**
  - To manipulate objects within EL expressions, you can use any of several arithmetic, relational, logical, or empty-testing operators.
  - (E.g.,) For conditional output, you can use `${test ? option1 : option2}`.
- **Automatic type conversion.**
  - The expression language removes the need for most typecasts and for much of the code that parses strings as numbers.
- **Empty values instead of error messages.**
  - In most cases, missing values or NullPointerExceptions result in empty strings, not thrown exceptions.

119

Invoking the Expression Language

- **Basic form: ${expression}**
  - These EL elements can appear in ordinary text or in JSP tag attributes, provided that those attributes permit regular JSP expressions. For example:
    - `<UL>`
    - `<LI>Name: ${expression1}`
    - `<LI>Address: ${expression2}`
    - `</UL>`
    - `<jsp:include page="${expression3}" />`
- **The EL in tag attributes**
  - You can use multiple expressions (possibly intermixed with static text) and the results are strings and concatenated. For example:
    - `<jsp:include page="${expr1}……..${expr2}" />`
- **Escaping special characters**
  - To get ${ in the page output, Use \${ in the JSP page. To get a single quote within an EL expression Use \' and to get a double quote within an EL expression Use \"

120

Accessing Scoped Variables

- **${varName}**
  - Means to search the PageContext, the HttpServletRequest, the HttpSession, and the ServletContext, *in that order*, and output the object with that attribute name.
- **Bean object variable**
  - ${name}

```
<%= pageContext.findAttribute("name") %>

<jsp:useBean id="person" type="somePackage.SomeClass" scope="...">
<jsp:getProperty name="person" property="name" />
```

121

JSPELExamples Project-- ScopedVars

```
@WebServlet(name = "ScopedVars", urlPatterns = {"/ScopedVars"})
public class ScopedVars extends HttpServlet {
protected void processRequest(HttpServletRequest request, HttpServletResponse
   response)
       throws ServletException, IOException {
     request.setAttribute("attribute1", "First Value");
     HttpSession session = request.getSession();
     session.setAttribute("attribute2", "Second Value");
     ServletContext application = getServletContext();
     application.setAttribute("attribute3",
          new java.util.Date());
     request.setAttribute("repeated", "Request");
     session.setAttribute("repeated", "Session");
     application.setAttribute("repeated", "ServletContext");
     RequestDispatcher dispatcher =
          request.getRequestDispatcher("/scopedVariables.jsp");
     dispatcher.forward(request, response);
  }
```

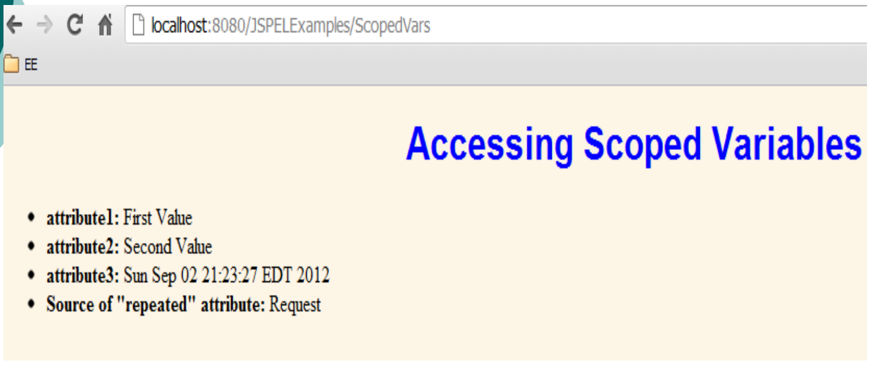
122

scopedVars.jsp

```
<!DOCTYPE ...>
...
<TABLE BORDER=5 ALIGN="CENTER">
<TR><TH CLASS="TITLE">
Accessing Scoped Variables
</TABLE>
<P>
<UL>
   <LI><B>attribute1:</B> ${attribute1}
   <LI><B>attribute2:</B> ${attribute2}
   <LI><B>attribute3:</B> ${attribute3}
   <LI><B>Source of "repeated" attribute:</B> ${repeated}
</UL>
</BODY></HTML>
```

123

ScopedVars (cont'd)



Accessing Scoped Variables

- **attribute1:** First Value
- **attribute2:** Second Value
- **attribute3:** Sun Sep 02 21:23:27 EDT 2012
- **Source of "repeated" attribute:** Request

124

Accessing Bean Properties

- **${varName.propertyName}**
  - Means to find scoped variable of given name and output the specified bean property
- **Equivalent forms**

```
${customer.firstName}

<%@ page import="packageName.NameBean" %>
<%
   NameBean person =
   (NameBean)pageContext.findAttribute("customer");
%>
<%= person.getFirstName() %>
```

125

Accessing Bean Properties (cont'd)

- **Equivalent forms**
  - ${customer.firstName}
  - <jsp:useBean id="customer" type="coreservlets.NameBean" scope="*request, page, session, or application*" />
  - <jsp:getProperty name="customer" property="firstName" />
- **This may be better than script on previous slide.**
  - But, requires you to know the scope and fails for sub-properties. (E.g.,) ${customer.address.zipCode}
- **Equivalent forms**
  - – ${name.property}
  - – ${name["property"]}
- Reasons for using array notations
  - – To access arrays, lists, and other collections
  - – To calculate the property name at request time.

126

BeanProperty servlet

```
@WebServlet(name = "BeanProperty", urlPatterns = {"/BeanProperty"})
public class BeanProperty extends HttpServlet {
protected void processRequest(HttpServletRequest request,
   HttpServletResponse response)
        throws ServletException, IOException {
     NameBean name = new NameBean("Bob", "Feiner");
     CompanyBean company =
          new CompanyBean("ee6Consulting.com",
          "Enterprise Java Consulting Co.");
     EmployeeBean employee = new EmployeeBean(name, company);
     request.setAttribute("employee", employee);

   request.getRequestDispatcher("beanProperties.jsp").forward(request,
   response);
   }
```

127

BeanProperties (cont'd)-- EmployeeBean

```
public class **EmployeeBean** {
 private NameBean name;
 private CompanyBean company;

 public EmployeeBean(NameBean name, CompanyBean company) {
  setName(name);
  setCompany(company);
 }
 public NameBean getName() { return(name); }

 public void setName(NameBean newName) {
  name = newName;
 }

 public CompanyBean getCompany() { return(company); }

 public void setCompany(CompanyBean newCompany) {
  company = newCompany;
 }
}
```

128

NameBean

```
public class NameBean {
  private String firstName = "Missing first name";
  private String lastName = "Missing last name";

  public NameBean() {}

  public NameBean(String firstName, String lastName) {
    setFirstName(firstName);
    setLastName(lastName);
  }
  public String getFirstName() {
    return(firstName);
  }
  public void setFirstName(String newFirstName) {
    firstName = newFirstName;
  }
  public String getLastName() {
    return(lastName);
  }

  public void setLastName(String newLastName) {
    lastName = newLastName;
  }
}
```

129

CompanyBean

```
public class CompanyBean {
  private String companyName;
  private String business;

  public CompanyBean(String companyName, String business) {
    setCompanyName(companyName);
    setBusiness(business);
  }

  public String getCompanyName() { return(companyName); }

  public void setCompanyName(String newCompanyName) {
    companyName = newCompanyName;
  }

  public String getBusiness() { return(business); }

  public void setBusiness(String newBusiness) {
    business = newBusiness;
  }
}
```

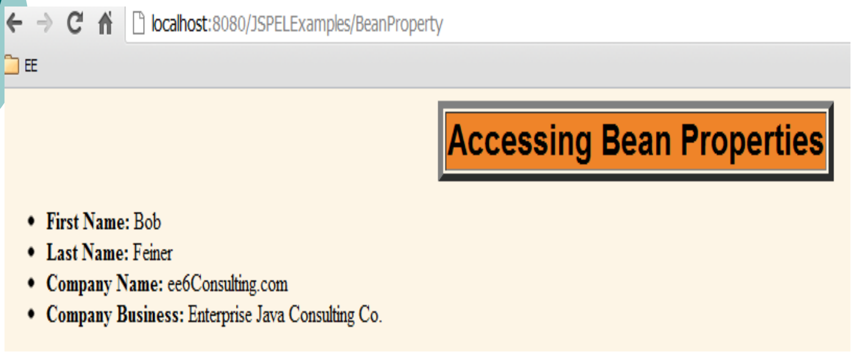
130

beanProperties.jsp

```
<!DOCTYPE …>
…
<UL>
  <LI><B>First Name:</B>
            ${employee.name.firstName}
  <LI><B>Last Name:</B>
            ${employee.name.lastName}
  <LI><B>Company Name:</B>
            ${employee.company.companyName}
  <LI><B>Company Business:</B>
            ${employee.company.business}
</UL>
</BODY>
</HTML>
```

131

BeanProperties (cont'd)



localhost:8080/JSPELExamples/BeanProperty

Accessing Bean Properties

- **First Name:** Bob
- **Last Name:** Feiner
- **Company Name:** ee6Consulting.com
- **Company Business:** Enterprise Java Consulting Co.

132

Accessing Collections

- • ${attributeName[entryName]} works for
  - Array.
    - Equivalent to theArray[index]
  - List.
    - Equivalent to theList.get(index)
  - Map.
    - Equivalent to theMap.get(keyName)
- **Equivalent forms (for HashMap)**
  - ${stateCapitals["maryland"]}
  - ${stateCapitals.maryland}
  - But the following is illegal since 2 is not a legal var name
    ${listVar.2}

133

Collections servlet

```
@WebServlet(name = "Collections", urlPatterns = {"/Collections"})
public class Collections extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    String[] firstNames = {"Bill", "Scott", "Larry"};
    ArrayList<String> lastNames = new ArrayList();
    lastNames.add("Ellison");
    lastNames.add("Gates");
    lastNames.add("McNealy");
    HashMap<String, String> companyNames = new HashMap();
    companyNames.put("Ellison", "Intel");
    companyNames.put("Gates", "Oracle");
    companyNames.put("McNealy", "Microsoft");
    request.setAttribute("first", firstNames);
    request.setAttribute("last", lastNames);
    request.setAttribute("company", companyNames);

    request.getRequestDispatcher("collections.jsp").forward(request, response);
  }
```

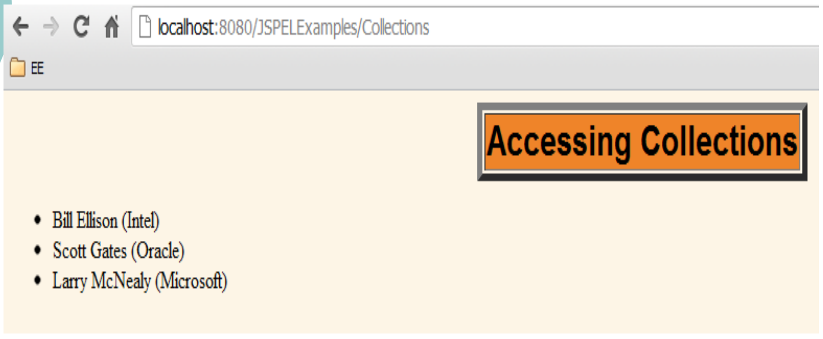
134

collections.jsp

```
<!DOCTYPE ...>
...
<BODY>
<H3>
Accessing Collections
</H3>
<P>
<UL>
  <LI>${first[0]} ${last[0]} (${company["Ellison"]})
  <LI>${first[1]} ${last[1]} (${company["Gates"]})
  <LI>${first[2]} ${last[2]} (${company["McNealy"]})
</UL>
</BODY></HTML>
```

135

Accessing Collections



Accessing Collections

- Bill Ellison (Intel)
- Scott Gates (Oracle)
- Larry McNealy (Microsoft)

136

Accessing Implicit Objects

- **pageContext**
  - E.g., ${pageContext.session.id}
- **param and paramValues**
  - Request params. – E.g. ${param.custID}
- **header and headerValues**
  - Request headers. – E.g. ${header.Accept} or ${header["Accept"]}, ${header["Accept-Encoding"]}
- **cookie**
  - Cookie object (not cookie value). – E.g. ${cookie.userCookie.value} or ${cookie["userCookie"].value}
- **initParam**
  - Context initialization param.
- **pageScope, requestScope, sessionScope, appliationScope.**

Expression Language support

- **implicit variables defined in the EL:**

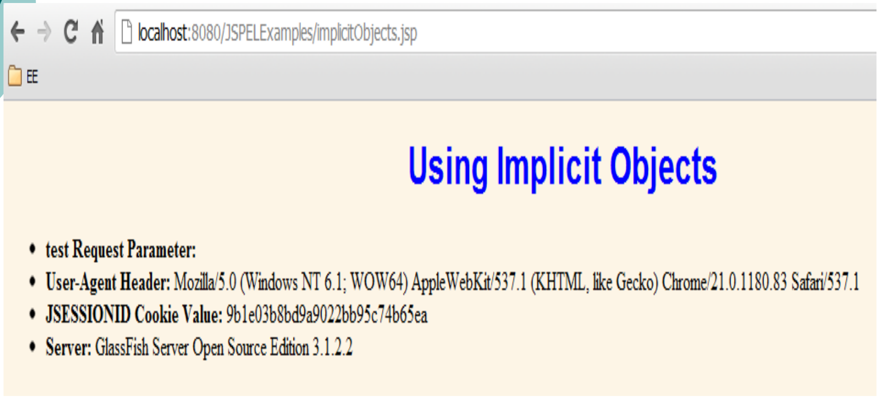
Variable name	Description
pageScope	A collection (a java.util.Map) of all page scope variables.
requestScope	A collection (a java.util.Map) of all request scope variables.
sessionScope	A collection (a java.util.Map) of all session scope variables.
applicationScope	A collection (a java.util.Map) of all application scope variables.
param	A collection (a java.util.Map) of all request parameter values as a single String value per parameter.
paramValues	A collection (a java.util.Map) of all request parameter values as a String array per parameter.
header	A collection (a java.util.Map) of all request header values as a single String value per header.
headerValues	A collection (a java.util.Map) of all request header values as a String array per header.
cookie	A collection (a java.util.Map) of all request cookie values as a single javax.servlet.http.Cookie value per cookie.
initParam	A collection (a java.util.Map) of all application initialization parameter values as a single String value per parameter.
pageContext	An instance of the javax.servlet.jsp.PageContext class, providing access to various request data.

Example: Implicit Objects

```
<!DOCTYPE …>
…
<P>
<UL>
   <LI><B>test Request Parameter:</B>
               ${param.test}
   <LI><B>User-Agent Header:</B>
               ${header["User-Agent"]}
   <LI><B>JSESSIONID Cookie Value:</B>
               ${cookie.JSESSIONID.value}
   <LI><B>Server:</B>
               ${pageContext.servletContext.serverInfo}
</UL>
</BODY></HTML>
```

139

Example: Implicit Objects


localhost:8080/JSPELExamples/implicitObjects.jsp

EE

Using Implicit Objects

- **test Request Parameter:**
- **User-Agent Header:** Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.83 Safari/537.1
- **JSESSIONID Cookie Value:** 9b1e03b8bd9a9022bb95c74b65ea
- **Server:** GlassFish Server Open Source Edition 3.1.2.2

140

Operators & Conditional Expressions

- **Arithmetic**
  - + - * / div % mod
- **Relational**
  - == eq != ne < lt > gt <= le >= ge
- **Logical**
  - && and || or ! not
- **Empty**
  - empty
  - True for null, empty string, empty array, empty list, empty map. False otherwise.
- **${ test ? expression1 : expression2 }**
  - Evaluates test and outputs either expression1 or expression2
  - Problems
    - Relatively weak
    - c:if and c:choose from JSTL may be better

141

Examples

${1.2 + 2.3} => 3.5	**<%= 1.2 + 2.3 %>**
${3/0} => Infinity	
\${1} => ${1}	
${10 mod 4} => 2	
${4.0 >= 3} => true	
${4.0 ge 3} => true	**Not in Java**
${100.0 == 100} => true	
${(10*10) ne 100} => false	**Not in Java**
${'hip' > 'hit'} => false	**Not in Java**
${'a' < 'b'} => true	**Not in Java**

142

Expression Language support

Operator	Description
.	Access a bean property or Map entry.
[]	Access an array or List element.
()	Group a subexpression to change the evaluation order.
? :	Conditional test: *condition* ? *ifTrue* : *ifFalse*.
+	Addition.
-	Subtraction or negation of a value.
*	Multiplication.
/ or div	Division.
% or mod	Modulo (remainder).
== or eq	Test for equality.
!= or ne	Test for inequality.
< or lt	Test for less than.
> or gt	Test for greater than.
<= or le	Test for less than or equal.
>= or ge	Test for greater than or equal.
&& or and	Test for logical AND.
|| or or	Test for logical OR.
! or not	Unary Boolean complement.
empty	Test for an empty variable value: null, an empty String, or an array, Map, or Collection without entries).
func(*args*)	A function call, where *func* is the function name and *args* is zero, one or more comma-separated function arguments.

143

Example: operator.jsp

```
.....
<TABLE BORDER=1 ALIGN="CENTER">
 <TR><TH CLASS="COLORED" COLSPAN=2>Arithmetic Operators
     <TH CLASS="COLORED" COLSPAN=2>Relational Operators
 <TR><TH>Expression<TH>Result<TH>Expression<TH>Result
 <TR ALIGN="CENTER">
   <TD>\${3+2-1}<TD>${3+2-1}  <%-- Addition/Subtraction --%>
   <TD>\${1&lt;2}<TD>${1<2}      <%-- Numerical comparison --%>
 <TR ALIGN="CENTER">
   <TD>\${"1"+2}<TD>${"1"+2}     <%-- String conversion --%>
   <TD>\${"a"&lt;"b"}<TD>${"a"<"b"} <%-- Lexical comparison --%>
 <TR ALIGN="CENTER">
   <TD>\${1 + 2*3 + 3/4}<TD>${1 + 2*3 + 3/4}  <%-- Mult/Div --%>
   <TD>\${2/3 &gt;= 3/2}<TD>${2/3 >= 3/2}       <%-- >= --%>
 <TR ALIGN="CENTER">
   <TD>\${3%2}<TD>${3%2}                <%-- Modulo --%>
   <TD>\${3/4 == 0.75}<TD>${3/4 == 0.75} <%-- Numeric = --%>
 <TR ALIGN="CENTER">
   <%-- div and mod are alternatives to / and % --%>
   <TD>\${(8 div 2) mod 3}<TD>${(8 div 2) mod 3}
   <%-- Compares with "equals" but returns false for null --%>
   <TD>\${null == "test"}<TD>${null == "test"}
```

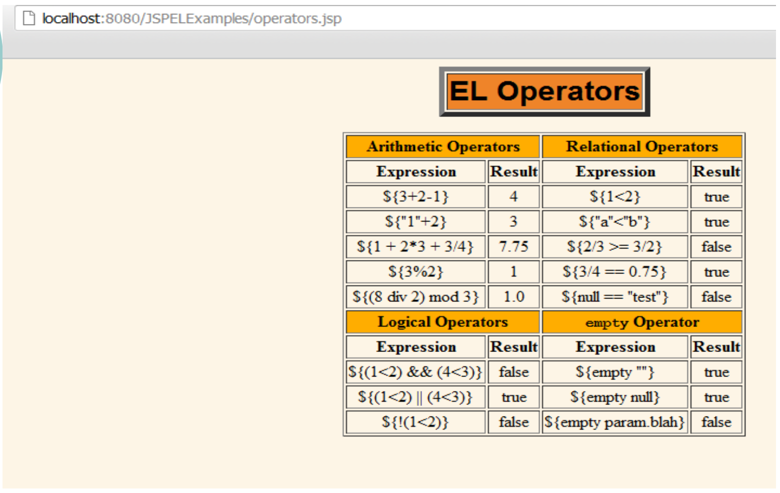
144

Example: operators.jsp (cont'd)

```
<TR><TH CLASS="COLORED" COLSPAN=2>Logical Operators
    <TH CLASS="COLORED" COLSPAN=2><CODE>empty</CODE> Operator
 <TR><TH>Expression<TH>Result<TH>Expression<TH>Result
 <TR ALIGN="CENTER">
  <TD>\${(1&lt;2) && (4&lt;3)}<TD>${(1<2) && (4<3)} <%--AND--%>
  <TD>\${empty ""}<TD>${empty ""} <%-- Empty string --%>
 <TR ALIGN="CENTER">
  <TD>\${(1&lt;2) || (4&lt;3)}<TD>${(1<2) || (4<3)} <%--OR--%>
  <TD>\${empty null}<TD>${empty null} <%-- null --%>
 <TR ALIGN="CENTER">
  <TD>\${!(1&lt;2)}<TD>${!(1<2)}  <%-- NOT -%>
  <%-- Handles null or empty string in request param --%>
  <TD>\${empty param.blah}<TD>${empty param.blah}
</TABLE>
</BODY></HTML>
```

145

Example: operators.jsp (cont'd)


localhost:8080/JSPELExamples/operators.jsp

EL Operators

Arithmetic Operators		Relational Operators	
Expression	Result	Expression	Result
${3+2-1}	4	${1<2}	true
${"1"+2}	3	${"a"<"b"}	true
${1 + 2*3 + 3/4}	7.75	${2/3 >= 3/2}	false
${3%2}	1	${3/4 == 0.75}	true
${(8 div 2) mod 3}	1.0	${null == "test"}	false
Logical Operators		**empty Operator**	
Expression	Result	Expression	Result
${(1<2) && (4<3)}	false	${empty ""}	true
${(1<2) || (4<3)}	true	${empty null}	true
${!(1<2)}	false	${empty param.blah}	false

146

Example: Conditionals

```
@WebServlet(name = "Conditionals", urlPatterns = {"/Conditionals"})
public class Conditionals extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
      SalesBean apples =
           new SalesBean(150.25, -75.25, 22.25, -33.57);
      SalesBean oranges =
           new SalesBean(-220.25, -49.57, 138.25, 12.25);
      request.setAttribute("apples", apples);
      request.setAttribute("oranges", oranges);

      request.getRequestDispatcher("conditionalEval.jsp").forward(request,
    response);
    }
```

147

SalesBean.java

```
public class SalesBean {
  private double q1, q2, q3, q4;

  public SalesBean(double q1Sales,
                   double q2Sales,
                   double q3Sales,
                   double q4Sales) {
    q1 = q1Sales;
    q2 = q2Sales;
    q3 = q3Sales;
    q4 = q4Sales;
  }

  public double getQ1() { return(q1); }

  public double getQ2() { return(q2); }

  public double getQ3() { return(q3); }

  public double getQ4() { return(q4); }

  public double getTotal() { return(q1 + q2 + q3 + q4); }
}
```

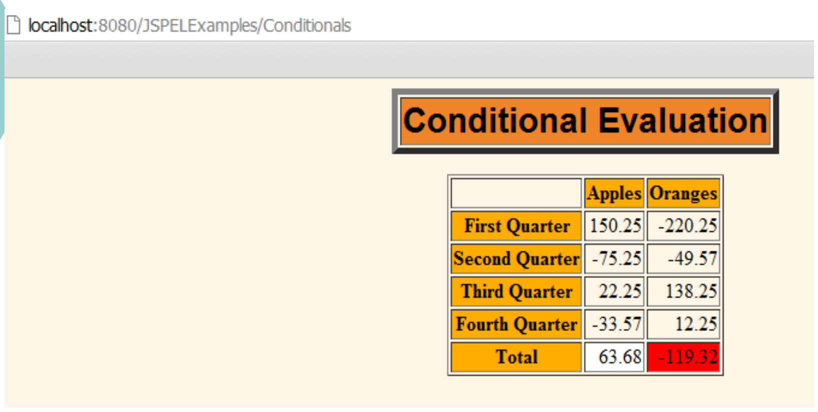
148

conditional-eval.jsp

```
…
<TABLE BORDER=1 ALIGN="CENTER">
<TR><TH>
    <TH CLASS="COLORED">Apples
    <TH CLASS="COLORED">Oranges
<TR><TH CLASS="COLORED">First Quarter
    <TD ALIGN="RIGHT">${apples.q1}
    <TD ALIGN="RIGHT">${oranges.q1}
<TR><TH CLASS="COLORED">Second Quarter
    <TD ALIGN="RIGHT">${apples.q2}
    <TD ALIGN="RIGHT">${oranges.q2}
…
<TR><TH CLASS="COLORED">Total
    <TD ALIGN="RIGHT"
      BGCOLOR="${(apples.total < 0) ? "RED" : "WHITE" }"> ${apples.total}
    <TD ALIGN="RIGHT"
      BGCOLOR="${(oranges.total < 0) ? "RED" : "WHITE" }"> ${oranges.total}
</TABLE>…
```

149

Conditional Evaluation



localhost:8080/JSPELExamples/Conditionals

Conditional Evaluation

	Apples	Oranges
First Quarter	150.25	-220.25
Second Quarter	-75.25	-49.57
Third Quarter	22.25	138.25
Fourth Quarter	-33.57	12.25
Total	63.68	-119.32

150

JSP Tag Libraries

JSP Tag Libraries allow you to define and use JSP tags in much the same way as you define and use functions in standard programming languages.

Examples are in JSTLExample NB Project

151

JSP Tag libraries

- **The debut of JavaServer Pages (JSP) followed by the inclusion of support for JSP tags were logical evolutionary steps toward fast, maintainable Java Web page implementation.**
- **The JSTL (JSP Standard Tag Library) further enables speeding and simplifying the development process.**
- **JSTL 1.2 was intended to align JSTL with JSP 2.1 (JSP2.2)**
- **JSTL pages are also JSP pages:**
- **Also, all JSTL tags are valid XML:**
- **A primary design goal for JSTL and the EL was to simplify Web page development and implementation.**
- **Can use JSTL either with JSP expressions or with EL (or both).**

152

The JSTL Tag Libraries

- **JSTL is often spoken of as a single-tag library.**
  - JSTL consists of five tag libraries.
- **Five tag libraries are:**
  - Core Tag Library – tags that are essential to nearly any Web application. Examples of core tag libraries include looping, expression evaluation, and basic input and output.
  - Formatting/Internationalization Tag Library – tags that are used to and parse data. Some of these tags will parse data, such as dates, differently based on the current locale.
  - Database Tag Library – tags that can be used to access SQL databases.
  - XML Tag Library – tags that can be used to access XML elements. Because XML is used in many Web applications, XML processing is an important feature of JSTL.
  - Function Library – tags that are used for testing and manipulating String and collection.

153

JSTL tag libraries

- Core: `http://java.sun.com/jsp/jstl/core`
- XML: `http://java.sun.com/jsp/jstl/xml`
- Internationalization: `http://java.sun.com/jsp/jstl/fmt`
- SQL: `http://java.sun.com/jsp/jstl/sql`
- Functions: `http://java.sun.com/jsp/jstl/functions`

154

JSTL tag libraries

- To use the JSTL core tag library in your page, include the following example directive at the top of your page:

```
<%@ taglib prefix="c" uri=http://java.sun.com/jsp/jstl/core%>
```

- To use the tags in that core library, prefix each tag in your page with the prefix you have designated in your include statement:

```
<c:out value="${anExpression}"/>
```

155

JSTL Tags

Area	Subfunction	Prefix
Core	Variable support	c
	Flow control	
	URL management	
	Miscellaneous	
XML	Core	x
	Flow control	
	Transformation	
I18N	Locale	fmt
	Message formatting	
	Number and date formatting	
Database	SQL	sql
Functions	Collection length	fn
	String manipulation	

156

JSTL Examples

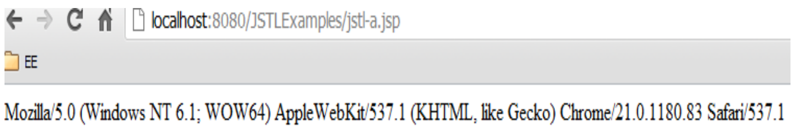
Examples in JSTLExamples NetBeans Project

157

Example: jstl-a.jsp

- The set action creates a variable named browser and assigns it the value of the User-Agent property.
- The out action then prints the value of the browser.

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
  <head> <title>Simple Example</title> </head>
  <body>
    <c:set var="browser" value="${header['User-Agent']}"/>
    <c:out value="${browser}"/>
  </body>
</html>
```



```
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.83 Safari/537.1
```

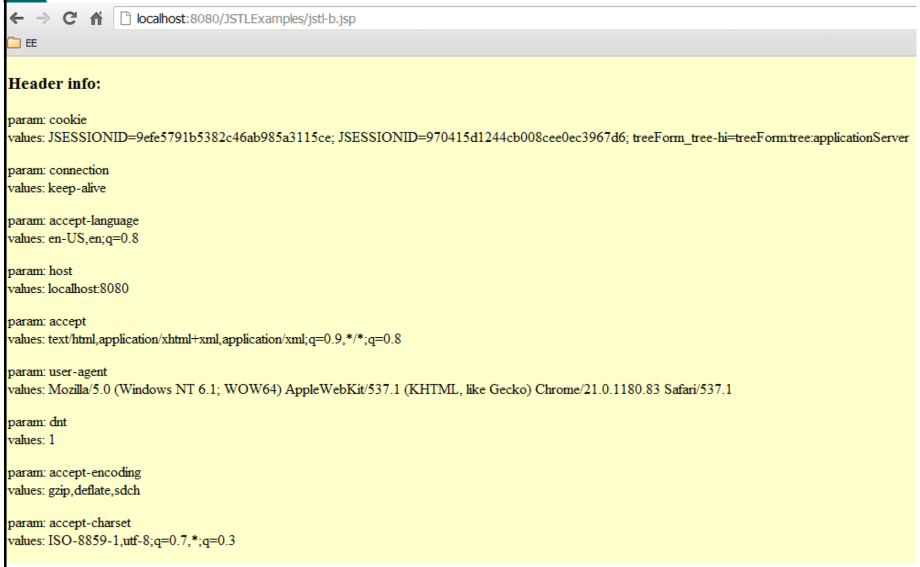
158

Example: jstl-b.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
  <head>
    <title>JSTL headers</title>
  </head>
  <body bgcolor="#FFFFCC">
    <h3>Header info:</h3>
    <c:forEach var="head" items="${headerValues}">
      param: <c:out value="${head.key}"/><br>
      values:
        <c:forEach var="val" items="${head.value}">
          <c:out value="${val}"/>
        </c:forEach>
        <p>
    </c:forEach>
  </body>
</html>
```

159

Example: jstl-b.jsp



```
Header info:

param: cookie
values: JSESSIONID=9efe5791b5382c46ab985a3115ce; JSESSIONID=970415d1244cb008cee0ec3967d6; treeForm_tree-hi=treeForm:tree:applicationServer

param: connection
values: keep-alive

param: accept-language
values: en-US,en;q=0.8

param: host
values: localhost:8080

param: accept
values: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

param: user-agent
values: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.83 Safari/537.1

param: dnt
values: 1

param: accept-encoding
values: gzip,deflate,sdch

param: accept-charset
values: ISO-8859-1,utf-8;q=0.7,*;q=0.3
```

160

XML Tags

- XML is becoming increasingly important to page authors, and the JSTL provides XML actions that address the basic needs of those developers.
- The XML actions can be divided into core, control flow, and transformation actions.
- The XML core actions are similar to those provided in the core actions discussed above, and include <x:out>, <x:set>, and <x:parse>.
- The main difference between the core actions discussed above and the XML core actions is that XML actions support XPath expressions, while the core actions do not.
  - As a matter of fact, the XML actions are based on XPath expressions.
- Xpath is a language for defining parts of an XML document; XPath uses path expressions to identify nodes in an XML document.

161

XML Tags (cont'd)

- The XML control flow actions are the same as the core actions.
  - They include: <x:if>, <x:choose>, <x:when>, <x:otherwise>, and <x:forEach>.
- The main difference is that you use the select attribute to specify XPath expressions.
  - An example:
  - <x:forEach select="$output/portfolio/stock"> <x:out select="price"/> </x:forEach>
- In the next slide, xml-ex1.jsp uses the core and XML tag libraries to process an XML document.
  - In this example, the XML document is embedded within the page and the <x:parse> tag is used to parse the document.
  - Then, an expression is used to select items from the document.

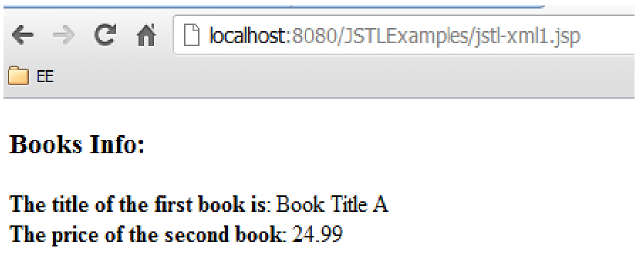
162

jstl-xml1.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="x" uri="http://java.sun.com/jsp/jstl/xml" %>

<html>
<head><title>JSTL Support for XML</title></head>
<h3>Books Info:</h3>
<c:set var="xmltext">
  <books>
    <book>
      <title>Book Title A</title>
      <author>A. B. C.</author>
      <price>17.95</price>
    </book>
    <book>
      <title>Book Title B</title>
      <author>X. Y. Z.</author>
      <price>24.99</price>
    </book>
  </books>
</c:set>

<x:parse xml="${xmltext}" var="output"/>
<b>The title of the first book is</b>: <x:out select="$output/books/book[1]/title"/>
<br>
<b>The price of the second book</b>: <x:out select="$output/books/book[2]/price"/>
</body>
</html>
```

163

Jstl-xml1.jsp (cont'd)



```
localhost:8080/JSTLExamples/jstl-xml1.jsp
```

Books Info:

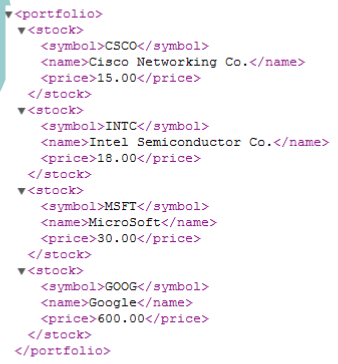
The title of the first book is: Book Title A
The price of the second book: 24.99

164

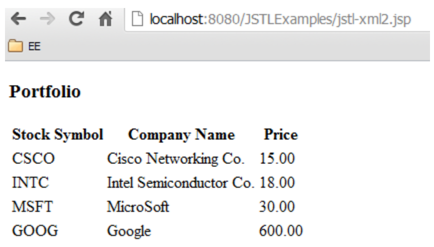
Jstl-xml2.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="x" uri="http://java.sun.com/jstl/xml" %>

<html>
<head> <title>JSTL Support for XML</title></head>
<h3>Portfolio</h3>
<c:import url="stocks.xml" var="xmldoc"/>
<x:parse xml="${xmldoc}" var="output"/><p>
<table border="2" width="50%">
 <tr>
 <th>Stock Symbol</th>
 <th>Company Name</th>
 <th>Price</th>
 </tr>
 <tr>

<x:forEach select="$output/portfolio/stock" var="item">
  <td><x:out select="symbol"/></td>
  <td><x:out select="name"/></td>
  <td><x:out select="price"/></td></tr>
</x:forEach>
</table>

</body>
</html>
```

Jstl-xml2.jsp



```
▼<portfolio>
  ▼<stock>
     <symbol>CSCO</symbol>
     <name>Cisco Networking Co.</name>
     <price>15.00</price>
   </stock>
  ▼<stock>
     <symbol>INTC</symbol>
     <name>Intel Semiconductor Co.</name>
     <price>18.00</price>
   </stock>
  ▼<stock>
     <symbol>MSFT</symbol>
     <name>MicroSoft</name>
     <price>30.00</price>
   </stock>
  ▼<stock>
     <symbol>GOOG</symbol>
     <name>Google</name>
     <price>600.00</price>
   </stock>
 </portfolio>
```



Portfolio

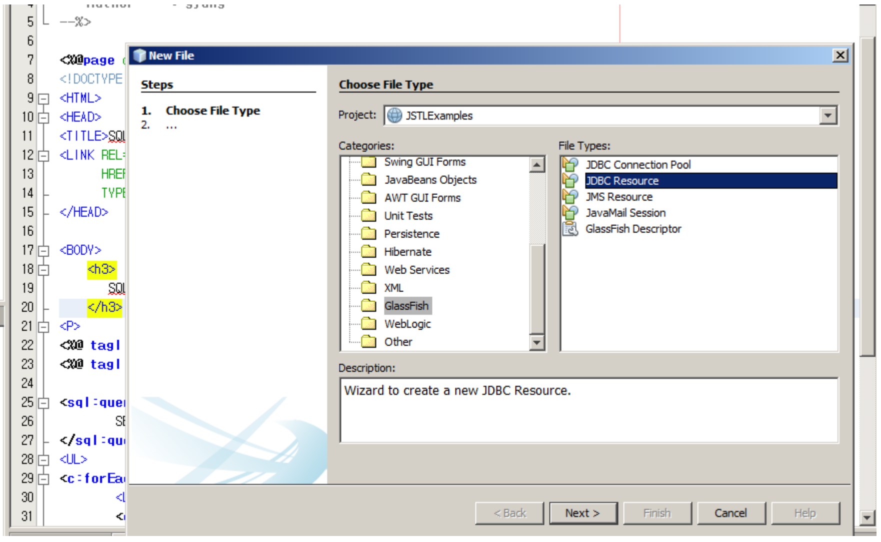
Stock Symbol	Company Name	Price
CSCO	Cisco Networking Co.	15.00
INTC	Intel Semiconductor Co.	18.00
MSFT	MicroSoft	30.00
GOOG	Google	600.00

sqlaccess.jsp

```
……………
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

<sql:query var="products" dataSource="jdbc/affablebean">
    SELECT * FROM product
</sql:query>
<UL>
<c:forEach var="row" items="${products.rows}">
    <LI><c:out value="${row.name}"/>
    <c:out value="${row.price}"/>
</c:forEach>
</UL>
```

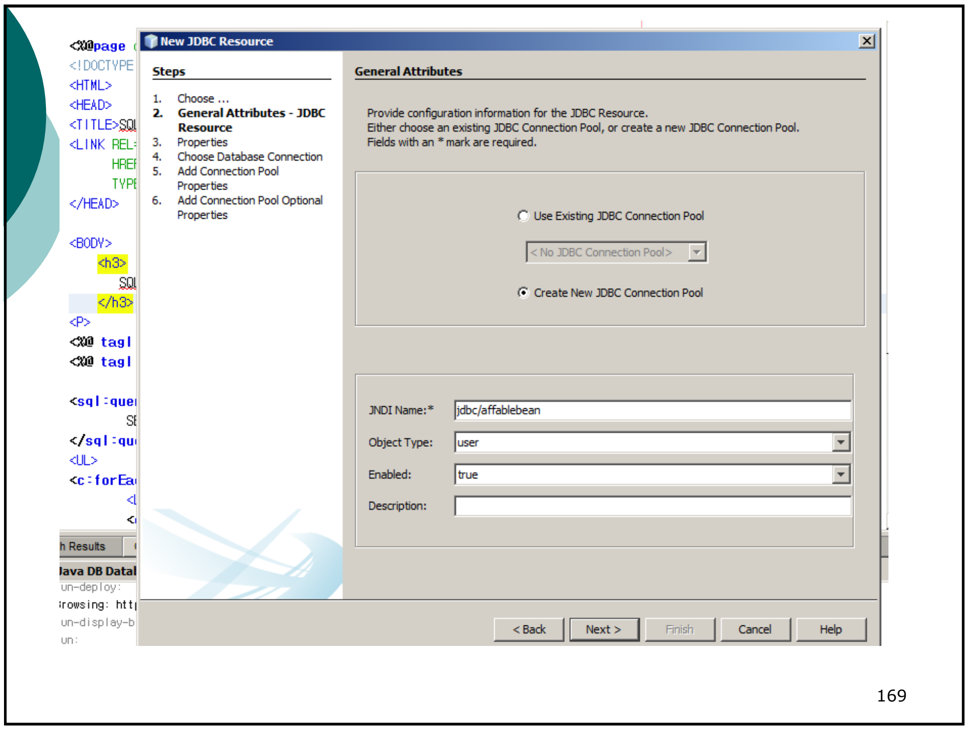
167

JDBC Resource Setting



Wizard to create a new JDBC Resource.

168



New JDBC Resource

Steps

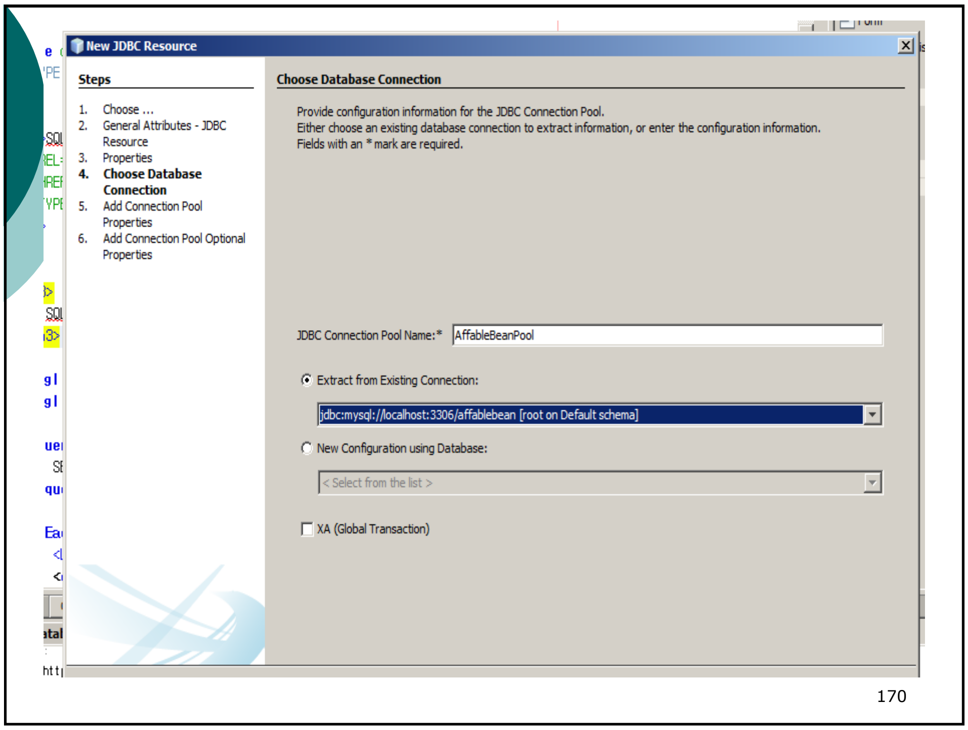
1. Choose ...
2. **General Attributes - JDBC Resource**
3. Properties
4. Choose Database Connection
5. Add Connection Pool Properties
6. Add Connection Pool Optional Properties

General Attributes

Provide configuration information for the JDBC Resource.
Either choose an existing JDBC Connection Pool, or create a new JDBC Connection Pool.
Fields with an * mark are required.

( ) Use Existing JDBC Connection Pool
< No JDBC Connection Pool >
(•) Create New JDBC Connection Pool

JNDI Name:*	jdbc/affablebean
Object Type:	user
Enabled:	true
Description:	



New JDBC Resource

Steps

1. Choose ...
2. General Attributes - JDBC Resource
3. Properties
4. **Choose Database Connection**
5. Add Connection Pool Properties
6. Add Connection Pool Optional Properties

Choose Database Connection

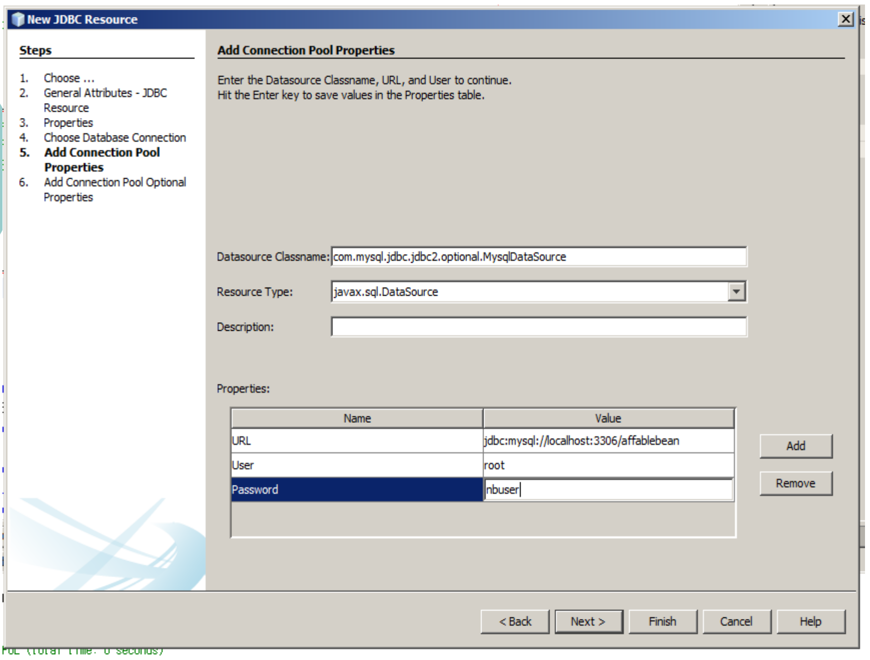
Provide configuration information for the JDBC Connection Pool.
Either choose an existing database connection to extract information, or enter the configuration information.
Fields with an * mark are required.

JDBC Connection Pool Name:* AffableBeanPool

(•) Extract from Existing Connection:
jdbc:mysql://localhost:3306/affablebean [root on Default schema]

( ) New Configuration using Database:
< Select from the list >

[ ] XA (Global Transaction)



New JDBC Resource

Steps

1. Choose ...
2. General Attributes - JDBC Resource
3. Properties
4. Choose Database Connection
5. **Add Connection Pool Properties**
6. Add Connection Pool Optional Properties

Add Connection Pool Properties

Enter the Datasource Classname, URL, and User to continue.
Hit the Enter key to save values in the Properties table.

Datasource Classname: com.mysql.jdbc.jdbc2.optional.MysqlDataSource

Resource Type: javax.sql.DataSource

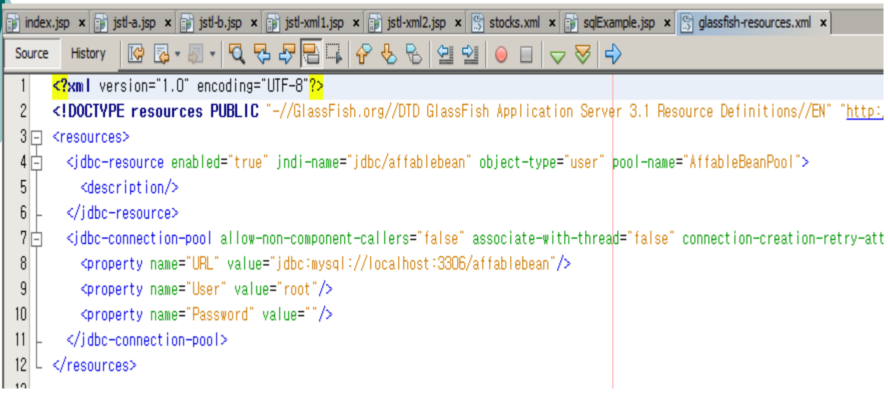
Description:

Properties:

Name	Value
URL	jdbc:mysql://localhost:3306/affablebean
User	root
Password	nbuser

171

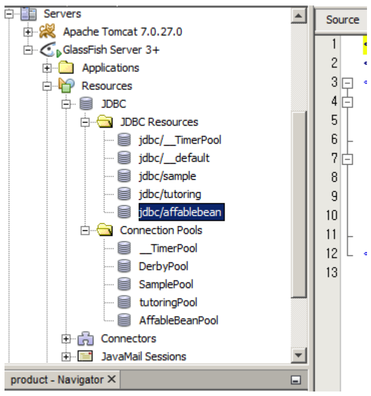
Glassfish-resources.xml updated



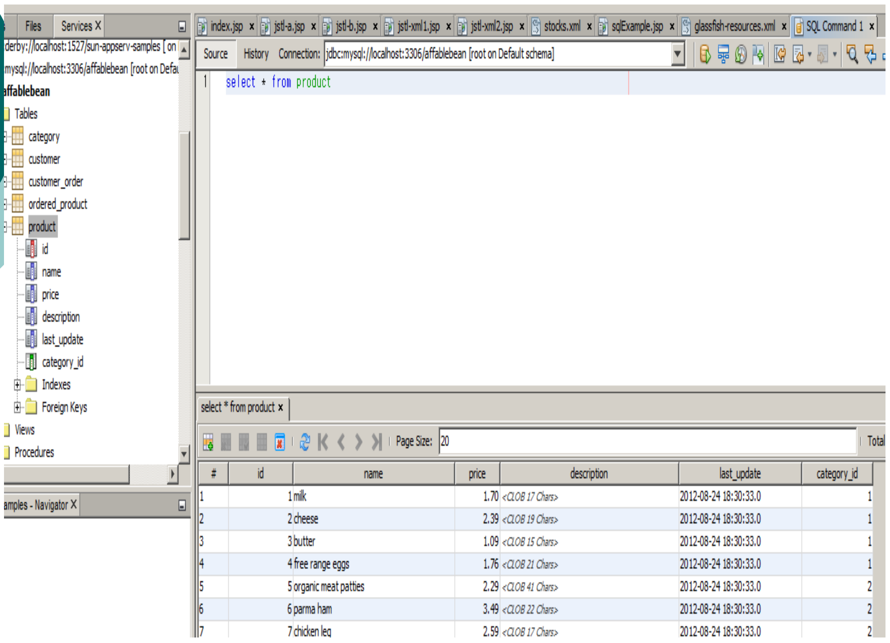
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Resource Definitions//EN" "http:
<resources>
  <jdbc-resource enabled="true" jndi-name="jdbc/affablebean" object-type="user" pool-name="AffableBeanPool">
    <description/>
  </jdbc-resource>
  <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connection-creation-retry-att
    <property name="URL" value="jdbc:mysql://localhost:3306/affablebean"/>
    <property name="User" value="root"/>
    <property name="Password" value=""/>
  </jdbc-connection-pool>
</resources>
```

172

Affablebean JDBC Resource


Servers
Apache Tomcat 7.0.27.0
GlassFish Server 3+
Applications
Resources
JDBC
JDBC Resources
jdbc/__TimerPool
jdbc/__default
jdbc/sample
jdbc/tutoring
jdbc/affablebean
Connection Pools
__TimerPool
DerbyPool
SamplePool
tutoringPool
AffableBeanPool
Connectors
JavaMail Sessions

173



derby://localhost:1527/sun-appserv-samples [on
mysql://localhost:3306/affablebean [root on Defau
ffablebean
Tables
category
customer
customer_order
ordered_product
product
id
name
price
description
last_update
category_id
Indexes
Foreign Keys
Views
Procedures

Connection: jdbc:mysql://localhost:3306/affablebean [root on Default schema]

```
select * from product
```

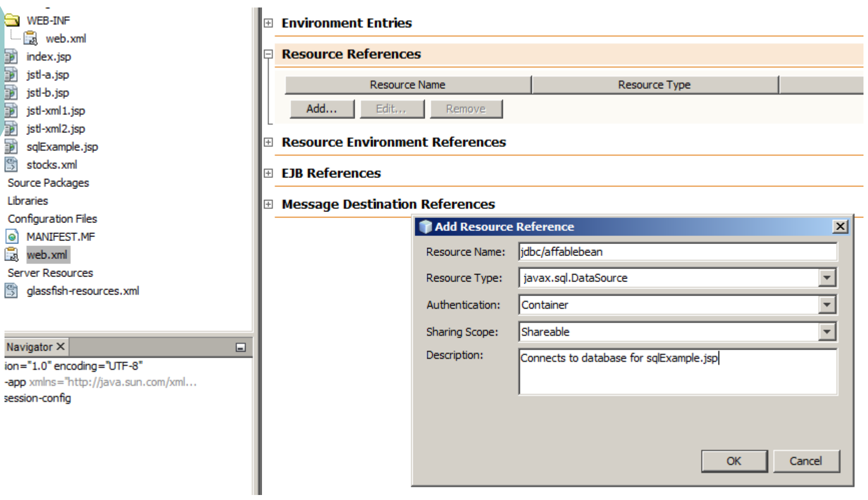
select * from product

Page Size: 20

#	id	name	price	description	last_update	category_id
1	1	milk	1.70	<CLOB 17 Chars>	2012-08-24 18:30:33.0	1
2	2	cheese	2.39	<CLOB 19 Chars>	2012-08-24 18:30:33.0	1
3	3	butter	1.09	<CLOB 15 Chars>	2012-08-24 18:30:33.0	1
4	4	free range eggs	1.76	<CLOB 21 Chars>	2012-08-24 18:30:33.0	1
5	5	organic meat patties	2.29	<CLOB 41 Chars>	2012-08-24 18:30:33.0	2
6	6	parma ham	3.49	<CLOB 22 Chars>	2012-08-24 18:30:33.0	2
7	7	chicken leg	2.59	<CLOB 17 Chars>	2012-08-24 18:30:33.0	2

174

Update web.xml to Add Resource reference



WEB-INF
web.xml
index.jsp
jstl-a.jsp
jstl-b.jsp
jstl-xml1.jsp
jstl-xml2.jsp
sqlExample.jsp
stocks.xml
Source Packages
Libraries
Configuration Files
MANIFEST.MF
web.xml
Server Resources
glassfish-resources.xml

Navigator
```
ion="1.0" encoding="UTF-8"
-app xmlns="http://java.sun.com/xml...
session-config
```

Environment Entries
Resource References

Resource Name	Resource Type

Add... Edit... Remove

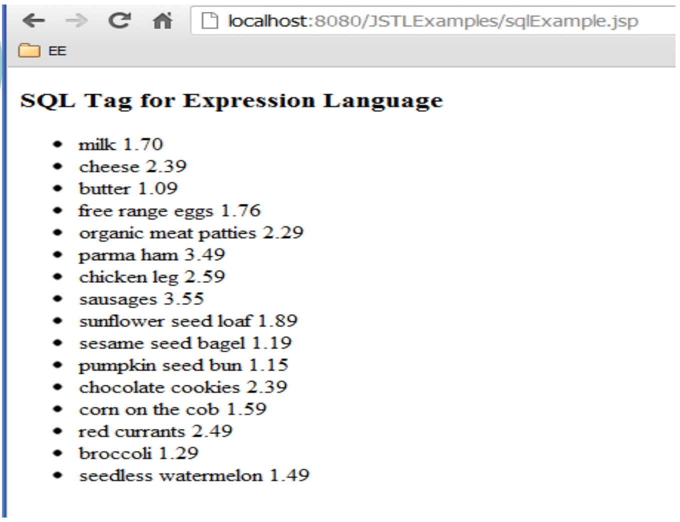
Resource Environment References
EJB References
Message Destination References

Add Resource Reference

Resource Name:	jdbc/affablebean
Resource Type:	javax.sql.DataSource
Authentication:	Container
Sharing Scope:	Shareable
Description:	Connects to database for sqlExample.jsp

OK Cancel

sqlExample.jsp



localhost:8080/JSTLExamples/sqlExample.jsp
EE

SQL Tag for Expression Language

- milk 1.70
- cheese 2.39
- butter 1.09
- free range eggs 1.76
- organic meat patties 2.29
- parma ham 3.49
- chicken leg 2.59
- sausages 3.55
- sunflower seed loaf 1.89
- sesame seed bagel 1.19
- pumpkin seed bun 1.15
- chocolate cookies 2.39
- corn on the cob 1.59
- red currants 2.49
- broccoli 1.29
- seedless watermelon 1.49

Customized Tags

177

Extending JSTL

- **It is also possible that you can use JSTL as a base for developing your own custom tags.**
- **Some abstract classes are provided that can assist with rapid development of tags and promote integration of your own custom tags with JSTLs tag set.**
- **For instance, you can build your own custom tags that make use of the JSTL.**
  - By extending javax.servlet.jsp.jstl.core.ConditionalTagSupport, you could write a conditional tag by simply implementing a single method that returns a boolean value corresponding with your tag's desired conditional behaviour.
  - *Or, by implementing javax.servlet.jsp.tagext.IterationTag interface*

178