JAVA DAYS ´99



JavaServer Pages ™ and Java Servlets

Anil K. Vijendran
Sun Microsystems, Inc

Sun microsystems

Agenda



- Servlets
  - What are Servlets? Why Servlets?
  - Servlet Basics, Some Servlet Topics, What's New in Servlet 2.2?
- JavaServer Pages ™ (JSP)
  - What are JSPs? Why JSPs? Where to use JSPs?
  - JSP Basics, Building your own Tags in JSP 1.1
- What's out there?

© 1999 Sun Microsystems GmbH



Java Servlets

© 1999 Sun Microsystems GmbH

Sun microsystems

What are Servlets?



- Java ™ Objects which extend the functionality of a HTTP server
- Comparable to Apache modules, Netscape's NSAPI, Microsoft's ISAPI
- Fundamental block of web access layer in Java 2 Enterprise Edition
- Platform/Server Independent

© 1999 Sun Microsystems GmbH

Sun microsystems

Why Servlets?



- HTTP is the lingua franca of the internet
- Servlets are lightweight
  - can run in the same HTTP server process
  - can be loaded from anywhere – local file system, remote web site
  - Faster and more secure than CGI by architecture
  - Easy to Develop – its Java!

© 1999 Sun Microsystems GmbH

Why Servlets? (Contd.)



- Extensive Library Support
  - JDK Libraries
    - JavaBeans™, RMI, Serialization, JDBC, IDL
  - In Java 2 Enterprise Edition (J2EE)
    - EJB, JMS, JavaMail, Transactions, XML
- Write Once Run/Deploy Anywhere

© 1999 Sun Microsystems GmbH

A Simple Servlet




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


© 1999 Sun Microsystems GmbH

Sun microsystems

Servlet Basics



- Servlets are managed objects in a Container
  - Have a well-defined lifecycle
    - init() for initialization
    - service() for servicing HTTP requests
    - destroy()
  - Can be loaded/unloaded on demand
  - Can be mapped to any part of container's URL namespace
- Servlets are Multi-threaded

© 1999 Sun Microsystems GmbH

Anatomy of a Request



- A client makes a request on a server
- The request is resolved to a servlet by the container
- The servlet's service() method is called with a Request (input) and Response (output) object
- Servlet "writes" into the response object which is sent to the client

© 1999 Sun Microsystems GmbH

Common Servlet API Abstractions



- Request
  - Encapsulates all per-request information from clients - parameters, URLs, cookies, headers...
- Response
  - Encapsulates all communication to the client
    - Streams to write binary output, Writers to write text
    - Redirect to other pages ...

© 1999 Sun Microsystems GmbH

Sun microsystems

Common Servlet API Abstractions (Contd.)



- Session
  - An abstraction to store/retrieve per-session information
  - Keeps servlet portable from various implementations of HTTP sessions.
    - URL rewriting
    - Cookies

© 1999 Sun Microsystems GmbH

Example: Sessions in Action




```
public void doGet(HttpServletRequest request,
                  HttpServletResopnse response)
    throws ServletException, IOException
{
    HttpSession session = request.getSession();
    Cart cart = (Cart)session.getValue("cart");

    // … do logic to get the inventory number
    // and quantity of goods to add to cart

    cart.addItem(inventoryNumber, quantity);
}
```


© 1999 Sun Microsystems GmbH

Sun microsystems

Servlet Usage Scenarios



- Traditionally used to assemble HTML pages, but that is now simpler using JSPs
- Servlets power comes from the ability to
  - generate binary content (example: images)
  - generate custom data formats (example: serialization)
  - other “interesting” business-to-business communication methods
  - sequencing page flow

© 1999 Sun Microsystems GmbH

Example: Generating an Image




```
public void doGet(HttpServletRequest req,
                  HttpServletResopnse res)
    throws ServletException, IOException
{
    // do logic to build image
    Image image = buildImage(req);
    res.setContentType("image/gif");
    ServletOutputStream out = res.getOutputStream();

    // use acme.com's Gif Encoder
    GifEncoder encoder = new GifEncoder(image, out);
    encoder.encode();
}
```


© 1999 Sun Microsystems GmbH

Sun microsystems

Client-Server Communication



- Various methods available to exchange data between applications on the internet
  - text-based - most popular is XML
  - binary - object serialization
- Text-based formats make interoperation easier
- Binary formats like object serialization help transfer complex object structures easily

© 1999 Sun Microsystems GmbH

Example: Client-Server Communication




```
// servlet side
{
    Result result = getResult(); // generate object
    ServletOutputStream out = response.getOutputStream();
    ObjectOutputStream objOut = new ObjectOutputStream(out);
    objOut.writeObject(result);
}

// client side
{
    URL url = new URL("http://myserver/resultservlet");
    InputStream in = url.openStream();
    ObjectInputStream objIn = new ObjectInputStream(in);
    Result result = (Result)objIn.readObject();
}
```


© 1999 Sun Microsystems GmbH

Servlet 2.2



- History: Servlet 2.1 Released 4Q1998
- 2.2 Highlights
  - Notion of web applications
  - Web application packaging
  - Deployment support

© 1999 Sun Microsystems GmbH

Web Applications



- A web application is
  - a collection of servlets, JSPs, html pages, supporting libraries etc
  - rooted at a particular URL prefix
  - typically packaged as a WAR - Web Application Archive (like JAR) or can exist on file system
- All servlets in a web app share the same ServletContext object

© 1999 Sun Microsystems GmbH

Sun microsystems

Deployment



- Web application archive is self-contained
- XML-based deployment descriptor
- Servlet developer can communicate configuration to the deployer using the deployment descriptor

© 1999 Sun Microsystems GmbH



JavaServer Pages ™

© 1999 Sun Microsystems GmbH

Sun microsystems

What are JSPs?



- Standard web access layer to server-side business logic
- Builds on HTML while providing facilities to generate dynamic content
  - write HTML like HTML is written - No Java code necessary
  - embed JSP tags (or Java) where you need dynamic content

© 1999 Sun Microsystems GmbH

Example: A Simple JSP

```
<%@ page language="java" %>
<html>
<jsp:useBean id="clock" class="MyCalendar" />
<ul>
<li>Day of the month: <%=clock.getDayOfMonth() %>
<li>Year: <%= clock.getYear() %>
</ul>
</html>
```


© 1999 Sun Microsystems GmbH

JSP Elements



- Standard template text (usually HTML)
- Directives
- Scripting – Java code in HTML
- Standard tags or actions
- Portable tag extension runtime

© 1999 Sun Microsystems GmbH

Directives



- Specify page interpretation semantics to the JSP engine (like #pragma or #include in C/C++)
- `<%@ directive_type {attribute="value"}* %>`
- Directive types
  - page
  - taglib
  - include

© 1999 Sun Microsystems GmbH

Sun microsystems

The Page Directive



- `<%@ page {attribute="value"}* %>`
- Attributes
  - language, imports, extends
  - contentType
  - buffer, autoFlush
  - session, isThreadSafe

© 1999 Sun Microsystems GmbH

Sun microsystems

Other Directives



- Tag libraries
  - `<%@ taglib uri="http://…/wombat-taglib.jar" prefix="wombat" %>`
- Including other files
  - `<%@ include file="…" %>`

© 1999 Sun Microsystems GmbH

Sun microsystems

Scripting



- JSPs are like Java classes (Servlets, actually)
- Declarations
  - `<%! int i = 0; %>`
- Expressions
  - `<%= request.getHeader("User-Agent") %>`
- Scriptlets
  - `<% System.out.println("Hello, World"); %>`

© 1999 Sun Microsystems GmbH

Sun microsystems

Implicit Objects



- Objects that are predefined for use in JSP scripting elements
  - out
  - request, response
  - pageContext
  - application, session

© 1999 Sun Microsystems GmbH

Sun microsystems

Predefined Tags (or Actions)



- Import JavaBeans ™ for use in JSPs
  - `<jsp:useBean id="clock" class="com.wombat.ClockBean />`
- Set and get properties of imported beans
  - `<jsp:getProperty name="clock" property="day"/>`
- Include and Forward to other active sources
  - `<jsp:include page="StatusServlet" />`

© 1999 Sun Microsystems GmbH

JSP and XML



- Syntax for directives and scripts are popular but not XML
- JSP defines XML equivalents (for easy processing by tools)
- JSPs can
  - accept XML in HTTP POST
  - generate XML data to the client

© 1999 Sun Microsystems GmbH

JSP 1.1: What's New?



- Portable Tag Extension Runtime
  - Define tag libraries using Tag Library Descriptors (TLD) (which is XML)
  - Implement tag handlers as Java classes
- Integration with Servlet 2.2
- Support for precompiling JSPs
  - using new request parameter `jsp_precompile`

© 1999 Sun Microsystems GmbH

Sun microsystems

Custom Tags



- Custom tags
  - are described using Tag Library Descriptor
  - packaged in a tag library (which is a JAR)
- Tag handler objects
  - have a well-defined life-cycle
  - can generate output
  - can define implicit objects for use in scriptlets, expressions and other tag handlers

© 1999 Sun Microsystems GmbH

Example: A Tag to Generate Error Logs




```
<%@ taglib uri="http://localhost:8080/jsp/libs/taglib.jar"
           prefix="jswdk" %>

<jswdk:log>
Did you see me on the stderr window?
</jswdk:log>

<jswdk:log tobrowser="true">
Did you see me on the browser window as well?
</jswdk:log>
```


© 1999 Sun Microsystems GmbH

Sun microsystems

Example: A Custom Tag Handler




```
public class LogTag extends Tag {
    boolean toBrowser = false;

    // …

    public int doStartTag() {
        // Initialize toBrowser here
    }

    public int doAfterBody() throws JspError {
        System.err.println(bodyOut.getString());
        if (toBrowser)
            bodyOut.writeOut(getPreviousOut());
        return SKIP_BODY;
    }
}
```


© 1999 Sun Microsystems GmbH

Sun microsystems

Servlets Vs JSPs



- Servlets ARE the mechanism behind JSPs
- JSPs make content generation easier
  - Make content designer and programmer roles more distinct and separate
  - Easy to author
- JSPs can’t generate binary output

© 1999 Sun Microsystems GmbH

Servlets and JSP – Status



- Servlet and JSP specifications developed with major partners using the Java Community Process
  - `http://java.sun.com/aboutJava/communityprocess/index.html`
- Servlet 2.2, JSP 1.1 specifications in the public release stage
  - `http://java.sun.com/products/{servlets,jsp}`

© 1999 Sun Microsystems GmbH

What's Out There?



- JavaServer Web Development Kit 1.0 final
  - Reference implementation for Servlet 2.1 and JSP 1.0
  - Available at `http://java.sun.com/products/jsp`
- Support for Servlets in
  - Apache, IBM WebSphere, Netscape Enterprise Server etc...

© 1999 Sun Microsystems GmbH

The Jakarta Project



- Sun is committed to ubiquity of JSP and Servlets
- JSWDK licensed to Apache under Apache license
- Project Jakarta set up under the Apache charter (`http://jakarta.apache.org`)
- Subprojects
  - JaSPer JSP engine and Tomcat Servlet engine

© 1999 Sun Microsystems GmbH

Sun microsystems

That's it!



- For further information:
  - Websites
    - http://java.sun.com/products/servlet
    - http://java.sun.com/products/jsp
  - Mailing lists
    - servlet-interest@java.sun.com
    - jsp-interest@java.sun.com
  - My email: akv@eng.sun.com

© 1999 Sun Microsystems GmbH