

Servlets & JSP

(with Tomcat & Apache)

Kai Tödter
CT SE 2
Siemens AG

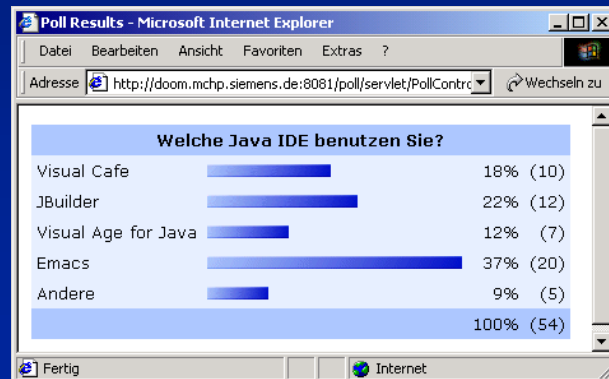
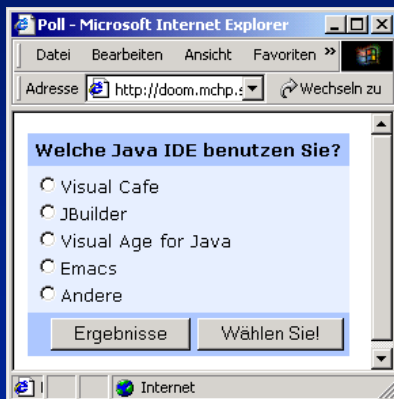
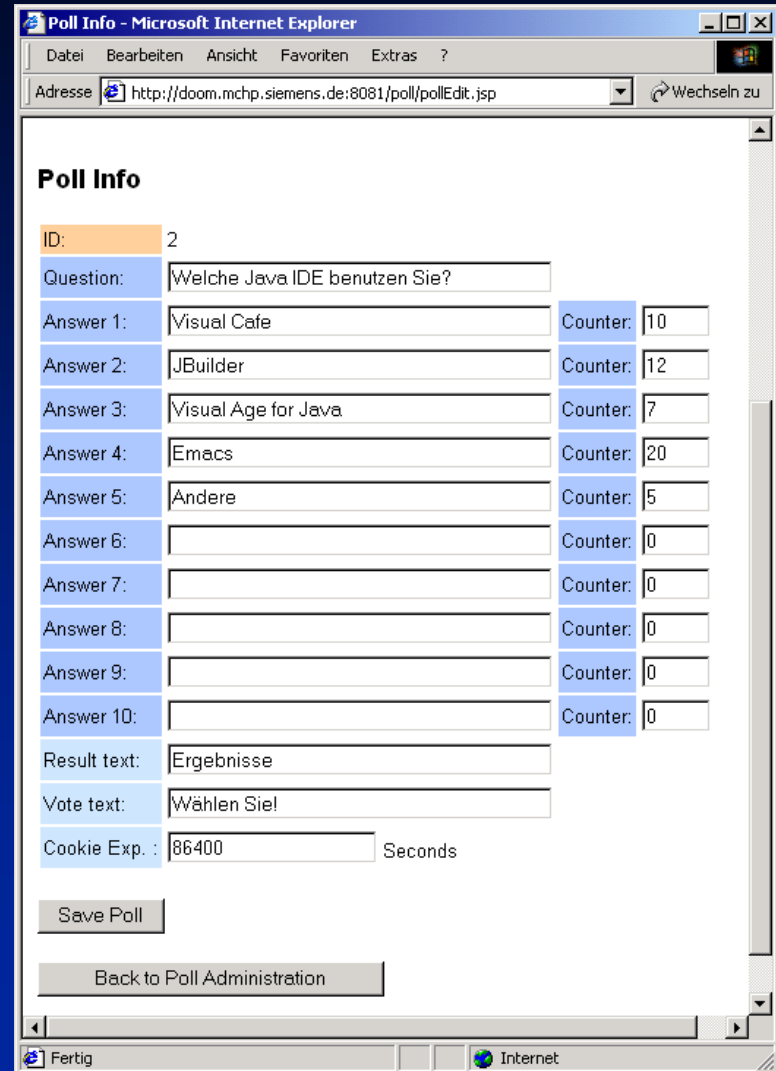
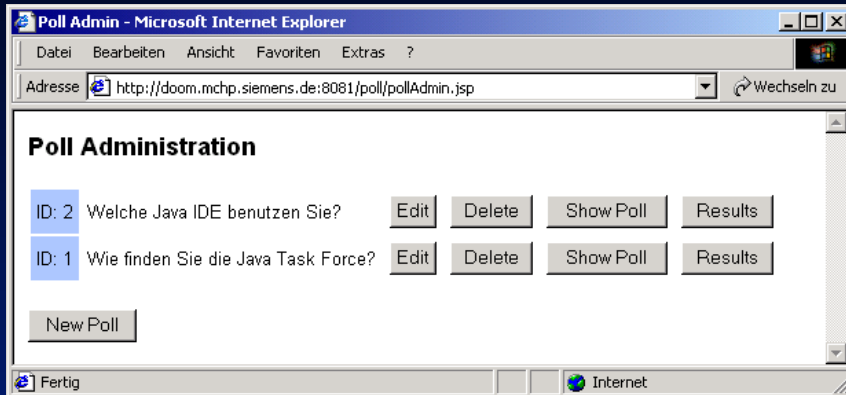
E-mail: kai.toedter@mchp.siemens.de

www: <http://www.toedter.com>

Outline

- Motivation: Poll Web Application
- Date Servlet/Date JSP
- Servlet Engine: Tomcat
- Web Server: Apache
- Basic Servlet/JSP
 - Request
 - Cookies
 - Authorization
 - Session
 - Tag Libraries
- Web Applications
- JDBC
- Books & Links

Poll Servlet/JSP Demo



Date Servlet (1)

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

public class DateServlet extends HttpServlet {
    public void doGet( HttpServletRequest request, HttpServletResponse response )
        throws ServletException, IOException {

        response.setContentType( "text/html" );
        PrintWriter out = response.getWriter();

        out.println("<HTML>");
        out.println("<HEAD><TITLE>Date Servlet</TITLE></HEAD>");
        out.println("<BODY><H2>Date Servlet: " + (new Date()) + "</H2></BODY>");
        out.println("</HTML>");
    }
}
```

Date Servlet (2)

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



```
public class DateServlet extends HttpServlet {
    public void doGet( HttpServletRequest request,
                      HttpServletResponse response )
        throws ServletException, IOException {
        // ...
    }
}
```

Date Servlet (3)

....

```
response.setContentType( "text/html" );
```

```
PrintWriter out = response.getWriter();
```

```
out.println("<HTML>");
```

```
out.println("<HEAD><TITLE>Date Servlet</TITLE></HEAD>");
```

```
out.println("<BODY><H2>Date Servlet: " +  
            (new Date()) + "</H2></BODY>");
```

```
out.println("</HTML>");
```

```
}
```

Date JSP

```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>Date JSP</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
  <H2> Date JSP: <%= new java.util.Date() %>
```

```
  </H2>
```

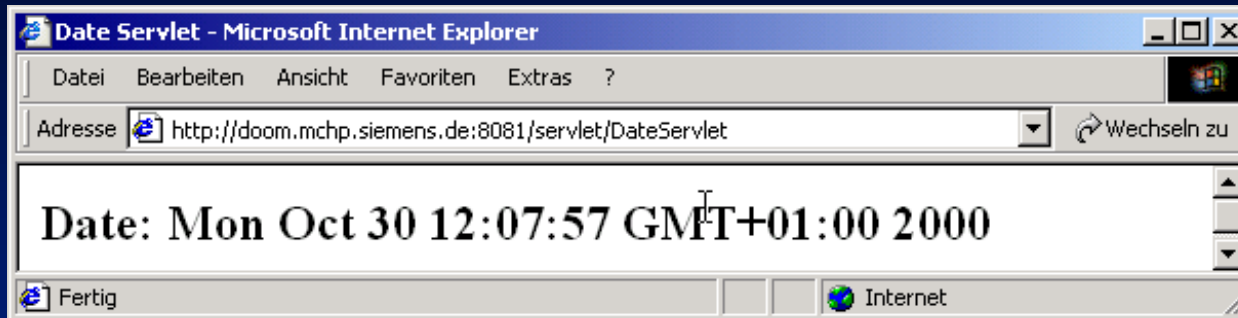
```
</BODY>
```

```
</HTML>
```

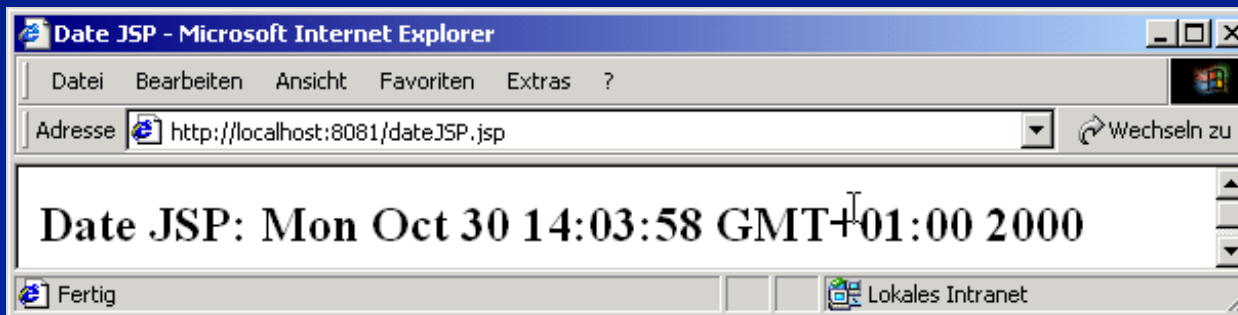


Date Servlet/JSP Demo

Servlet



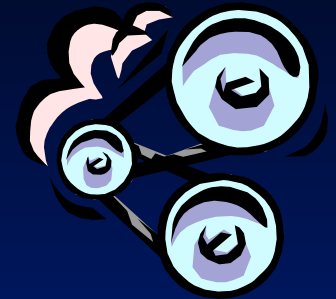
JSP



Servlet/JSP Engines

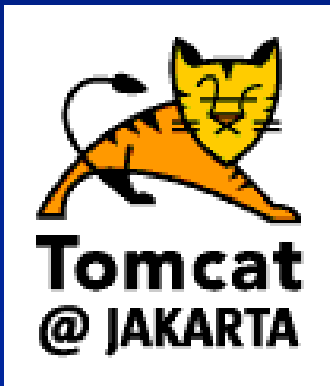
- List of Servlet engines:

- <http://www.servlets.com/resources/urls/engines.html>



- Tomcat

- Reference Implementation for Servlets & JSPs
 - Implementing the latest API
 - Open Source
 - Apache Project Jakarta






Installing Tomcat

- Download Tomcat from jakarta.apache.org
- Unzip it!
 - ◆ E.g. c:\java\jakarta-tomcat
- Set 2 environment variables
 - ◆ JAVA_HOME: E.g. c:\java\jdk1.3
 - ◆ Tomcat_HOME: E.g. c:\java\jakarta-tomcat
- Probably run Tomcat as DEFAULT web server:
 - ◆ Change c:\java\jakarta-tomcat\conf\server.xml
 - ◆ `<Parameter name="port" value="8080"/>` to `<Parameter name="port" value="80"/>`
- Start Tomcat
 - ◆ Double click c:\java\jakarta-tomcat\bin\startup.bat
- Test Tomcat
 - ◆ Type "http://localhost:8080" in your browser
 - ◆ (or "http://localhost", if you use port 80)

Tomcat Start Page



Tomcat v3.2.1
Version 3.2.1

This is the default Tomcat home page. This page serves as a quick reference guide to related resources and is located at:

- `</path/to/tomcat>/webpages/index.html`

Included within this release are functional examples with associated source code, API documentation for servlets and JSP, a **README**, a technical FAQ on this release and an assortment of jar files which are pre-requisites for continued development of web technologies including JSP and Servlets.

Examples:

- [JSP Examples](#)
- [Servlet Examples](#)

Documentation:

- [API docs for Servlet and JSP Packages](#)


The **README** file, which can be found at `</path/to/tomcat>/README`, contains a list of known bugs, incompatibilities and limitations.

You can find more information about the Servlet and JSP technologies at:

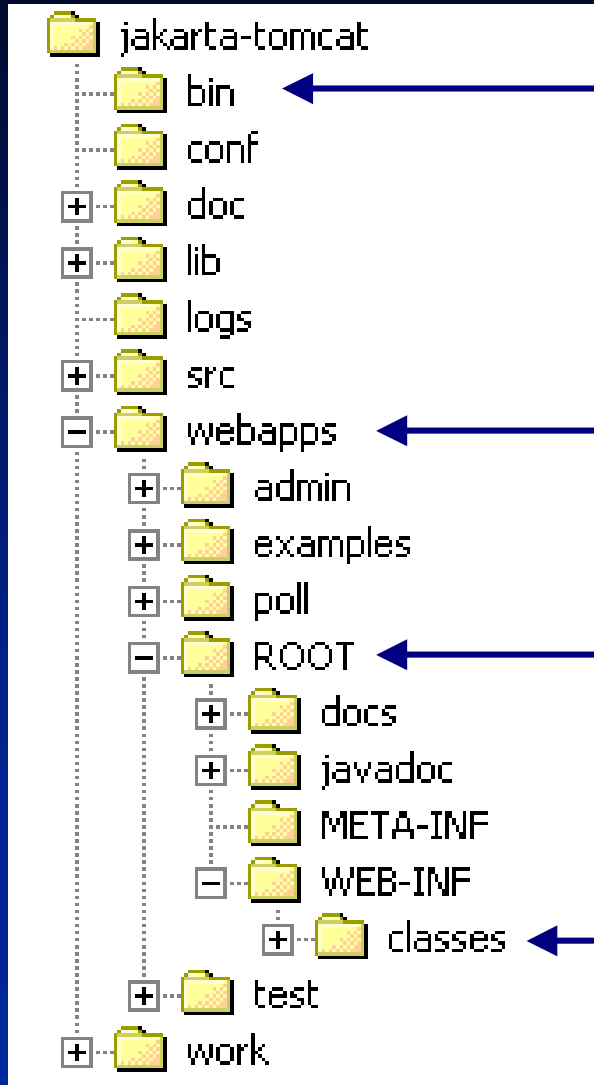
- [Sun's Java Server Pages Site](#)
- [Sun's Servlet Site](#)

and by subscribing to one or more of the following Servlet and JSP related interest lists:

- jsp-interest@java.sun.com
- servlet-interest@java.sun.com

Powered by

TOMCAT
Copyright © 1999 Apache Software Foundation
All Rights Reserved

Tomcat File Structure



Use `startup.bat` & `shutdown.bat` to start & stop Tomcat

Put your web application directory here

Put your test JSP here

Put your test servlets here

Apache Web Server



- Web-Site: httpd.apache.org
- Free Software!
- Robust, fast, powerful, many features, ...
- Available for almost every operating system
 - ◆ E.g. Solaris, Windows, Linux, ...
- Most widely used web server in the world

BUT:

- Not very easy to manage and administrate

Apache Installation



- Download latest version from httpd.apache.org
- Under Windows: Run setup.exe
- Edit Apache\httpd.conf
 - Change ServerAdmin to your E-Mail
 - Change ServerName to your computers name
 - Perhaps change Port from 80 (Default) to another port
- Start server
 - Run Apache Web Server\Start Apache from Start Menu
 - You can also install Apache as a service (from Start Menu)
- Test server
 - Use "http://localhost" in your web browser
 - (Or "http://<your computer name>:<your port>")

Apache Test



Testseite fuer die Apache-Installation auf dieser Web Site - Microsoft Internet Explorer

Datei Bearbeiten Ansicht Favoriten Extras ?

Adresse <http://doom:8085> Wechselt zu

Links [Poll Admin](#) [Future Team](#) [JAR5](#) [gratis2000](#) [MySpace - Browse](#) [Kostenlos.de](#) [BB](#) [TelasWeb](#) [SCD](#) [Planet GeForce](#) [Meta-Search](#) [Google](#)

Es klappt! Der Apache Web-Server ist auf dieser Web-Site installiert!

Wenn Sie diese Seite sehen, dann bedeutet das, dass die Eigentümer dieser Domäne soeben einen neuen [Apache Web-Server](#) erfolgreich installiert haben. Jetzt muss noch der richtige Web-Inhalt zugefügt und diese Platzhalter-Seite ersetzt werden (oder der Web-Server für den Zugriff auf den richtigen Inhalt umkonfiguriert werden).

Wenn Sie diese Seite an Stelle einer anderen erwarteten Web-Site sehen sollten, dann **nehmen Sie bitte Kontakt mit dem Eigentümer dieser Site auf** (Versuchen Sie, eine EMail an `<Webmaster@domänename>` zu senden)!

Obwohl der Eigentümer dieser Domäne die Apache Web-Server Software verwendet, hat diese Web-Site ziemlich sicher keinerlei Verbindung mit der *Apache Software Foundation* (die diese Software kostenlos vertreibt). Es besteht also **keinerlei Veranlassung**, eine EMail an die Entwickler der Software zu senden. Sollten Sie das dennoch tun, wird Ihre Mail stillschweigend **ignoriert**.

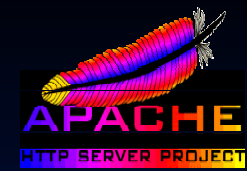
Die englische [Online-Dokumentation](#) für die Apache Web-Server Software ist Bestandteil dieser Software-Distribution.

Dem *WebMaster* dieser Web-Site steht es frei, das untenstehende "Powered by Apache"-Logo auf einem Apache-basierten Web-Server zu verwenden. Vielen Dank, dass Sie Apache gewählt haben!



Fertig Lokales Intranet

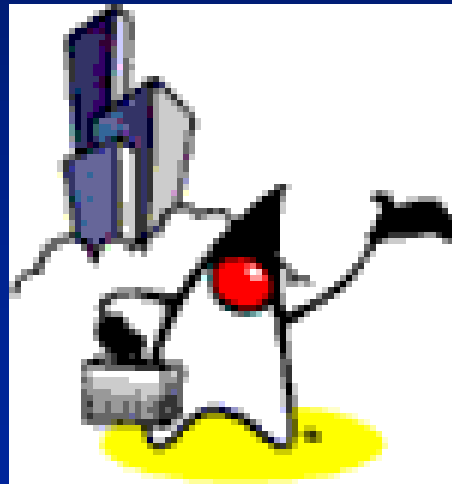
Tomcat & Apache Integration



- Download DLLs from binary Tomcat distribution
 - ApacheModuleJserv.dll
 - isapi_redirect.dll
 - jni_connect.dll
 - mod_jk.dll
- Put these DLLs in your Apache\modules directory
- Edit Apache\conf\httpd.conf
 - Put the line
`include <Your tomcat directory>/conf/mod_jk.conf-auto`
at the end of httpd.conf
- Start Tomcat & Apache
- Test your servlet/JSP with Apache

What is a Servlet?

A servlet is a web component, managed by a container, that generates dynamic content.



Advantages over other techniques

- Generally much faster than CGI scripts
 - A different process model is used.
- Standard API
 - Supported by many web servers.
- All the advantages of the Java
 - Ease of development
 - Platform independence
 - Large API



What is a Java Server Page?

A JSP is a text-based document that describes how to process a *request* to create a *response*. The description intermixes template data with some dynamic actions and leverages the Java 2 Platform.

E.g. a HTML page that contains Java code

JSP Benefits

- Easy to combine static templates, including HTML or XML
- JSP pages compile dynamically into servlets
- JSP makes it easier to author pages "by hand"
- JSP tags for invoking JavaBeans™ components
- Availability of authoring tools
 - E.g. Macromedia Dreamweaver Ultra Dev



pollEdit.jsp in Dreamweaver Ultra Dev

Poll Info

ID:

Question:

Answer Counter:

Result text:

Vote text:

Cookie Exp.: Seconds

Save Poll

Back to Poll Administration

<body> 447 x 427 7 K / 2 Sek

Poll Info

ID: 2

Question:

Answer 1: <input type="text" value="Visual Cafe"/>	Counter: <input type="text" value="10"/>
Answer 2: <input type="text" value="JBuilder"/>	Counter: <input type="text" value="12"/>
Answer 3: <input type="text" value="Visual Age for Java"/>	Counter: <input type="text" value="7"/>
Answer 4: <input type="text" value="Emacs"/>	Counter: <input type="text" value="20"/>
Answer 5: <input type="text" value="Andere"/>	Counter: <input type="text" value="5"/>
Answer 6: <input type="text" value=""/>	Counter: <input type="text" value="0"/>
Answer 7: <input type="text" value=""/>	Counter: <input type="text" value="0"/>
Answer 8: <input type="text" value=""/>	Counter: <input type="text" value="0"/>
Answer 9: <input type="text" value=""/>	Counter: <input type="text" value="0"/>
Answer 10: <input type="text" value=""/>	Counter: <input type="text" value="0"/>

Result text:

Vote text:

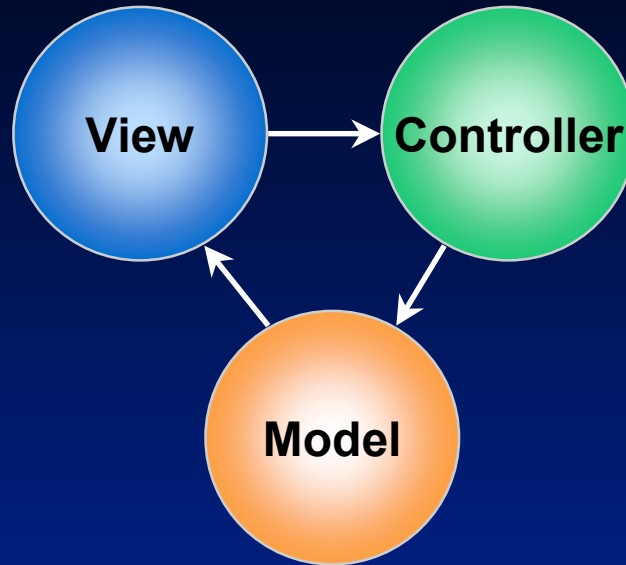
Cookie Exp.: Seconds

Save Poll

Back to Poll Administration

Fertig Internet

Model-View-Controller Architecture



■ Model

An object that defines the component's state

■ View

A visual screen representation of a component

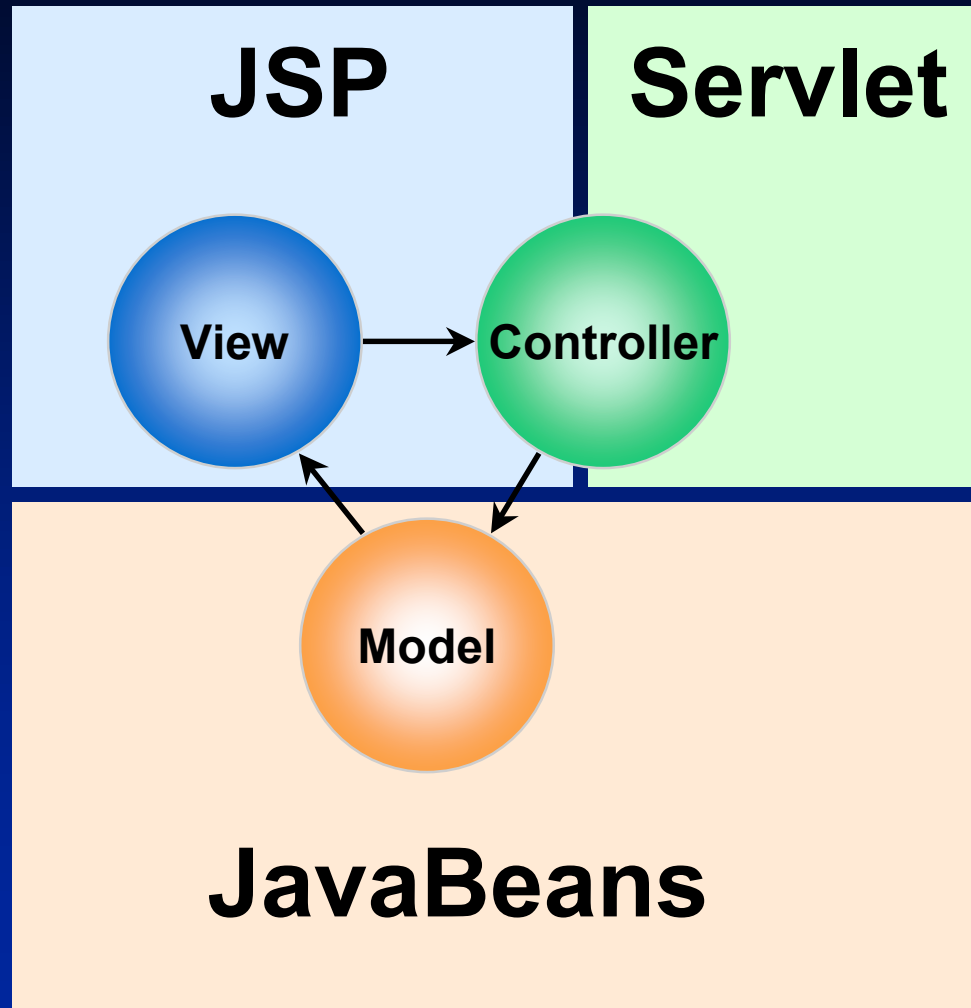
■ Controller

An object that makes a component respond to user input

Advantages of MVC Architecture

- Multiple appearances, multiple behaviors
 - Without disturbing the component's underlying architecture.
- Easy extension and change of existing views or controllers
- Appearance and behavior can change at runtime

MVC with Servlets/JSP/JavaBeans



Poll MVC Architecture

JSP

- authorization
- errorpage
- pollAdmin
- pollVote
- pollEdit
- pollResult

Servlet

- pollController

Java / JavaBeans

- poll
- pollFactory

JSP Scripting Elements

■ Declarations

- Are inserted outside any methods (in the body) of the servlet
- Example: `<%! private int counter; %>`

■ Expressions

- Are evaluated and inserted into the output
- Example: `<%= poll.getAnswer() %>`

■ Scriptlets

- Java code, which is inserted into the servlet
- Example: `<% for(int i=0; i<poll.getAnswerCount(); i++) %>`

Request

- Class HttpServletRequest
- Provides information about the client request
 - Method (GET, POST, ...)
 - URI (e.g. /poll/pollController)
 - Protocol (e.g. HTTP/1.0)
 - **Parameters**
 - Header (User Agent, Host, ...)
 - **Cookies**
 - **Authorization**
 - User
 - **Session**
 - ...

Parameters in Servlets

■ Getting all parameters:

```
Enumeration pn = request.getParameterNames();  
for (;pn.hasMoreElements();) {  
    String parName = (String) pn.nextElement();  
    String parValue = (String) request.getParameter(parName);  
    out.println("Parameter: " + parName + " ");  
    out.println("Value: " + parValue + "<br>" );  
}
```

■ Getting a specific parameter:

```
String results = request.getParameter( "results" );
```

Parameters in JSP

■ Getting all parameters:

```
<H3>Parameters</H3>
```

```
<% for (Enumeration e =request.getParameterNames();  
    e.hasMoreElements();) {
```

```
    String parName = (String) e.nextElement();
```

```
    Object parValue = request.getParameter(parName);
```

```
%>
```

```
<%=parName%>: <%=parValue%><br>
```

```
<% } %>
```

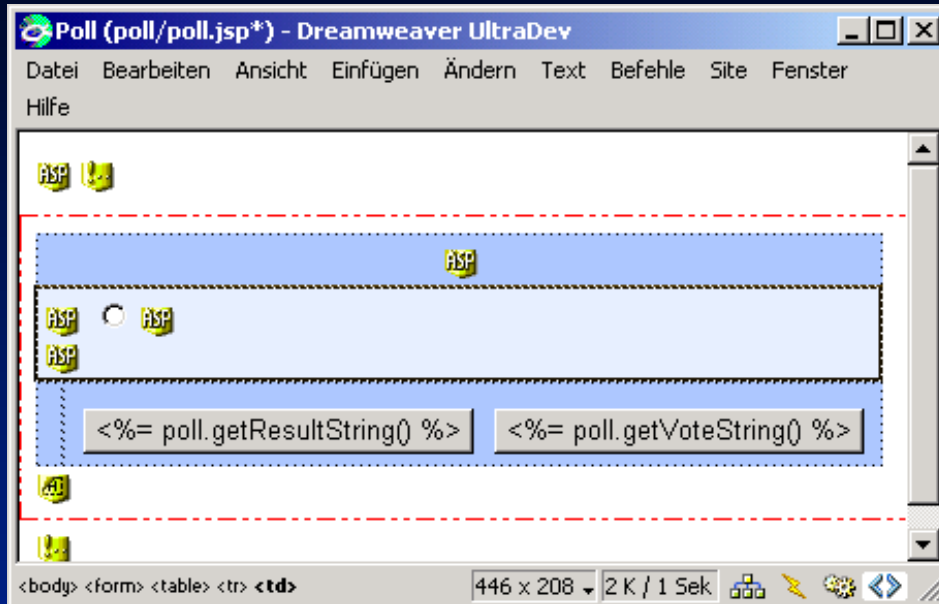
■ Getting a specific parameter:

```
Results: <%= request.getParameter( "results" ); %>
```

Setting a Parameter in votePoll Form

```
<td colspan="3">
  <font face="Verdana, Arial, Helvetica, sans-serif" size="-1">
    <% for( int i=0; i<poll.getAnswerCount(); i++ ) {
      %>
      <input type="radio" name="votevalue" value="<%= i %>">
        <%= poll.getAnswerText(i) %> <br>
      <% }
      %>
    </font>
  </td>
```

pollVote.jsp Form in DreamWeaver



```
IL-Quelle
e
?
Umbruch Liniennummern
method="post" action="http://<%= request.getHeader
border="0" cellspacing="0" cellpadding="4" borde
align="center">
d colspan="3" bgcolor="#ACC5FC"><font face="Verde
</tr>
<tr bgcolor="#E6EEFF">
  <td colspan="3"><font face="Verdana, Arial, Helvet
    <% for( int i=0; i<poll.getAnswerCount(); i++ )
      %>
      <input type="radio" name="votevalue" value="<%=
    <% )
      %>
    </font>
  </td>
</tr>
<tr>
```

Error Page in JSP

- Useful to have an own error page
 - Catching uncaught exceptions at runtime
- Using an error page in a JSP
 - `<%@ page errorPage="/util/errorpage.jsp" %>`



- A JSP error page

```
<%@ page isErrorPage="true" ... % >
```

```
<html>
```

```
<head><title>JSP Error Page</title></head>
```

```
...
```

JSP Error Page Example

JSP Error Page

Exception: java.lang.ArithmeticException

Reason: / by zero

Stack Trace: java.lang.ArithmeticException: / by zero at
_0002fpollEdit_0002ejspollEdit_jsp_1._jspService
(_0002fpollEdit_0002ejspollEdit_jsp_1.java:83) at
org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:119) at
javax.servlet.http.HttpServlet.service(HttpServlet.java:853) at
org.apache.jasper.servlet.JspServlet\$JspServletWrapper.service(JspServlet.java:177) at
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:309) at
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:382) at
javax.servlet.http.HttpServlet.service(HttpServlet.java:853) at
org.apache.tomcat.core.ServletWrapper.doService(ServletWrapper.java:387) at
org.apache.tomcat.core.Handler.service(Handler.java:263) at
org.apache.tomcat.core.ServletWrapper.service(ServletWrapper.java:371) at
org.apache.tomcat.core.ContextManager.internalService(ContextManager.java:786) at
org.apache.tomcat.core.ContextManager.service(ContextManager.java:732) at
org.apache.tomcat.service.http.HttpConnectionHandler.processConnection
(HttpConnectionHandler.java:210) at org.apache.tomcat.service.TcpWorkerThread.runIt
(PoolTcpEndpoint.java:407) at org.apache.tomcat.util.ThreadPool\$ControlRunnable.run
(ThreadPool.java:498) at java.lang.Thread.run(Thread.java:484)

Parameters: edit: Edit
command: edit
id: 3

Attributes:

Attribute	Value
javax.servlet.jsp.jspException	java.lang.ArithmeticException: / by zero

Classpath: .;D:\java\jakarta-tomcat\classes;D:\java\jakarta-tomcat\lib\ant.jar;D:\java\jakarta-tomcat\lib\jasper.jar;D:\java\jakarta-tomcat\lib\jaxp.jar;D:\java\jakarta-tomcat\lib\parser.jar;D:\java\jakarta-tomcat\lib\servlet.jar;D:\java\jakarta-tomcat\lib\webserver.jar;D:\java\jdk1.3\lib\tools.jar;D:\java\jdk1.3\jre\lib\rt.jar

Fertig Lokales Intranet

Page Forwarding

■ In JSP

- `<jsp:forward page = "pollVote.jsp" />`

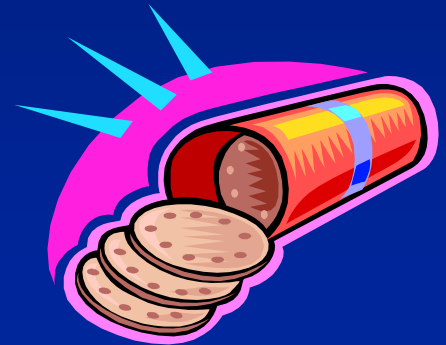
■ In servlets

- `getServletContext().
getRequestDispatcher("/pollResult.jsp").
forward(request, response);`

Cookies

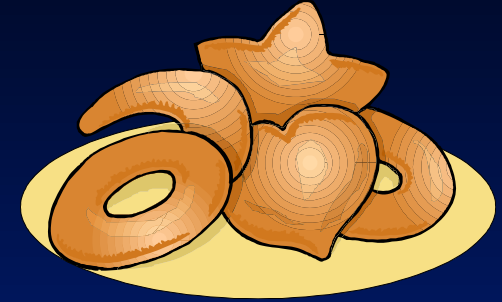
What is a cookie?

- A cookie is a piece of textual information
- Send by the Web server to the client browser
- Every time the browser visits the Web site again, the cookie is sent unchanged back to the server



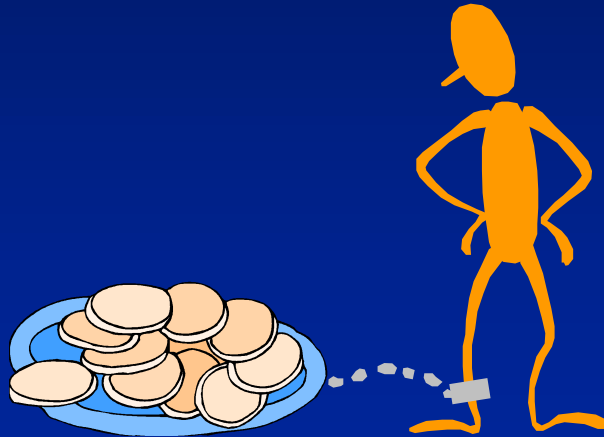
Benefits of Cookies

- Identification of a user
 - E-commerce
 - Poll
- Customizing a site
 - Portals
 - Personal Views
- Avoiding login
 - Useful for sites that don't require high security



Drawbacks of cookies

- User can deny cookies in his Web browser
- Might be a privacy thread
 - E.g. search engines remember user specific search topics



Setting a Cookie

■ Per session

...

```
String idString = Integer.toString( pollID );  
Cookie cookie = new Cookie( "PowerPollVote", idString );  
response.addCookie( cookie );
```

■ Persistent

...

```
cookie.setMaxAge( 60 * 60 * 24 ); // For 24 hours  
response.addCookie( cookie );
```

Getting a Cookie

```
Cookie[] cookies = request.getCookies();
```

```
String powerPollVote = "-1";
```

```
for( int i=0; i<cookies.length; i++ ) {
```

```
    if( cookies[i].getName().equals( "PowerPollVote" ) )
```

```
        powerPollVote = cookies[i].getValue();
```

```
}
```

Session Tracking

- HTTP is a “stateless” protocol
- A HttpSession object keeps state during a session
- HttpSession provides a rich API
 - `public void setAttribute(String name, Object value)`
 - `public Object getAttribute(String name)`
 - `public Object removeAttribute(String name)`
 - `public String getId()`
 - `public long getCreationTime()`
 - ...



Session Examples

■ In JSP

```
<%@ page errorPage="/util/errorpage.jsp" session="true" %>  
<%  
...  
String user = "Kai";  
session.setAttribute( "user", user );  
%>
```

■ In Servlet

```
...  
HttpSession session = request.getSession( true );  
String user = (String) session.getAttribute("user");
```

Authorization

- Needed for security aspects
- Algorithm:
 - Check for Authorization header
 - If there is no Authorization header, ask for password
 - Set response status to SC_UNAUTHORIZED
 - Set response header “WWW-Authenticate“ to you realm
 - If there is an Authorization header
 - Create a BASE64Decoder
 - Get user and password
 - Check against stored user & password



Authorization Example (1)

```
<%@ page errorPage="/util/errorpage.jsp" session="true"
import="sun.misc.*" %>
```

```
<%
```

```
String aut = request.getHeader("Authorization");
```

```
if (aut == null) {
```

```
    response.setStatus(HttpServletResponse.SC_UNAUTHORIZED);
```

```
    response.setHeader( "WWW-Authenticate",
                        "BASIC realm=\"Web Poll\", ");
```

```
    return;
```

```
}
```

```
else {
```

```
    // See next slide
```

```
}
```


```
%>
```

Authorization Example (2)

```
else {
    aut = aut.substring( aut.indexOf(' ')+1 );
    BASE64Decoder decoder = new BASE64Decoder();
    String s = new String( decoder.decodeBuffer( aut ));
    String user = s.substring( 0,s.indexOf(':') );
    String passwd = s.substring( s.indexOf(':')+1 );
    if( !(user.equals( "Kai" ) && ( passwd.equals( "jtf" ) ))) {
        response.setStatus( HttpServletResponse.SC_UNAUTHORIZED );
        response.setHeader( "WWW-Authenticate", "BASIC realm=\"Poll\"");
        return;
    }
}
```

Authorization Example (3)

Netzwerkennwort eingeben [?] [X]

 Geben Sie Benutzernamen und Kennwort ein.

Site: doom.mchp.siemens.de

Bereich: Web Poll

Benutzername:

Kennwort:

Kennwort in Kennwortliste speichern

JavaBeans

- Java component model
- Minimal requirements to a bean:
 - Public empty constructor (no parameters)
 - Should have no public attributes
 - attribute XXX should be accessed through
 - public Type getXXX()
 - public void setXXX(Type xxx)
- JavaBeans are easy to use in JSP



- Examples:

```
<jsp:useBean id = "myBean" class = "mypackage.MyClass" />
```

```
<%= myBean.getXXX() %>
```

```
<jsp:setProperty name = "myBean" property = "XXX" value = "test" />
```

Setting Parameters to Bean Properties

■ Setting a specific bean property

```
<jsp:setProperty name = "myBean"  
  property = "XXX"  
  value = '<% request.getParameter( "XXX" ) %>' />
```

■ Setting the properties of a bean from all parameters

```
<jsp:setProperty name = "myBean" property= "*" />
```

JSP Tag Libraries

- Creating of custom tags
- Tags can manipulate JSP content
 - JavaBeans can not!
- Easy to use for HTML Designers
 - Complex operations can be hidden by simple tags
- Better separation of HTML Design & Java source code



Tag Library Descriptor (1)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib
  PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.1//EN"
  "http://java.sun.com/j2ee/dtds/web-jsptaglibrary_1_1.dtd">
<taglib>
  <tlibversion>1.0</tlibversion>
  <jspversion>1.1</jspversion>
  <shortname>poll</shortname>
  <uri></uri>
  <info>
    A tag lib for a web poll
  </info>
  ... Tags, see next slide
</taglib>
```

Tag Library Descriptor (2)

... See previous slide

```
<taglib>
```

... See previous slide

```
<tag>
```

```
  <name>htmlfilter</name>
```

```
  <tagclass>polltaglib.HtmlFilterTag</tagclass>
```

```
  <bodycontent>JSP</bodycontent>
```

```
  <info>
```

A filter for HTML characters

```
  </info>
```

```
</tag>
```

```
</taglib>
```

JSP Tag Java Source

```
public class HtmlFilterTag extends BodyTagSupport {
    public int doEndTag() throws JspException {
        try {
            JspWriter out = pageContext.getOut();
            out.print( filter( getBodyContent().getString()));
        }
        catch( IOException e ) {
            throw new JspException(e .getClass().getName() + ": " + e.getMessage());
        }
        return (EVAL_PAGE);
    }

    public int doStartTag() throws JspException {
        return EVAL_BODY_TAG;
    }
}
```

JSP Tag Lib Example

```
<%@ page session="false" errorPage="/util/errorpage.jsp" %>
<%@ taglib uri="/taglib/polltaglib.tld" prefix="poll" %>
```

```
<html>
<body>
<pre>
  <poll:htmlfilter>
    <html>
      <body>
        <h3 align=center>The Tag poll:htmlfilter</h3>
        <p>Paragraph: HTML except JSP tags
      </body>
    </html>
  </poll:htmlfilter>
</pre>
</body>
</html>
```

The Tag poll:htmlfilter

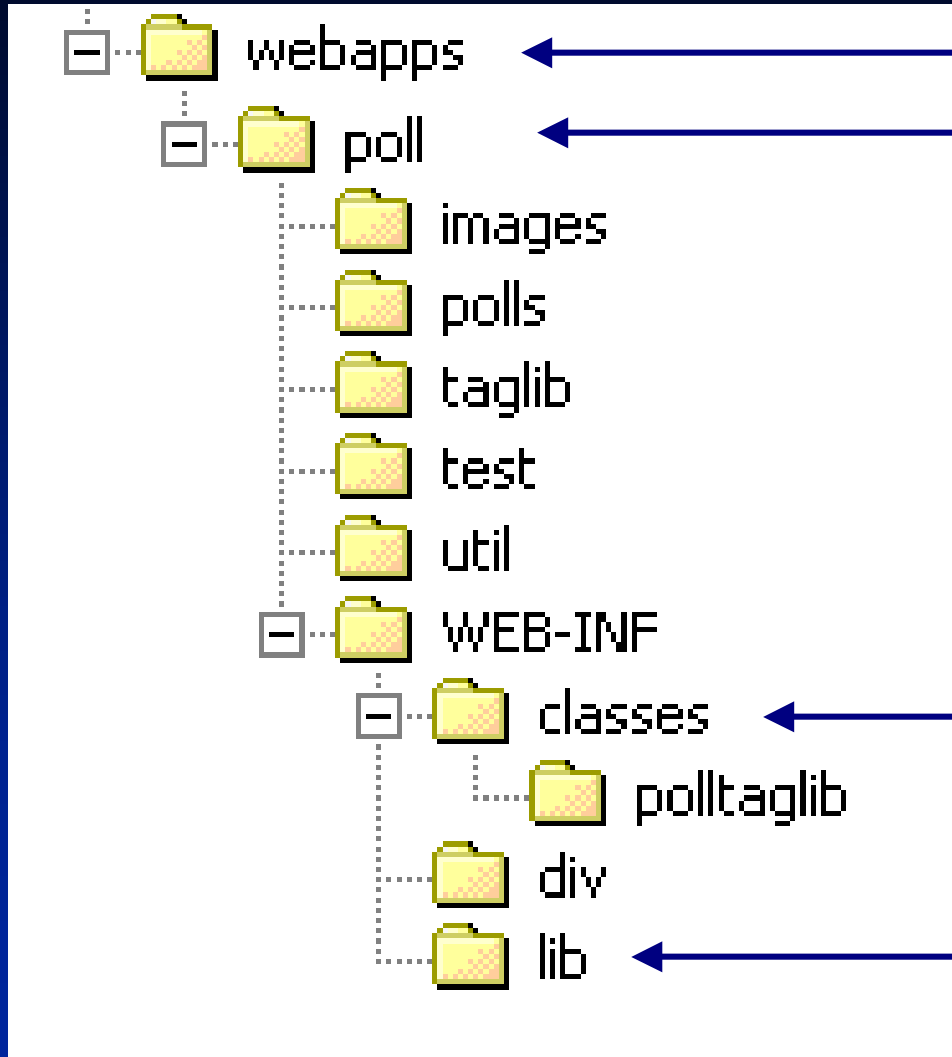
```
<html>
<body>
  <h3 align=center>The Tag poll:htmlfilter</h3>
  <p>Paragraph: HTML except JSP tags
</body>
</html>
```

Web Applications

- Java Runtime Environment(s)
 - Running in the server (required)
- JSP(s)
 - Handle requests and generate dynamic content
- Servlet(s)
 - Handle requests and generate dynamic content
- Server-side JavaBeans components
 - Encapsulate behavior and state
- Static HTML, DHTML, XHTML, XML and similar pages.
- Client-side Java classes
 - Applets, JavaBeans, other Java class files
- Java Runtime Environment(s)
 - Downloadable via the Plugin
 - Running in client(s)



Web Application Structure



Root for Web Applications
Poll Web Application

Classes

Additional JAR Files

Web Application Deployment

- Web Applications can be deployed as
 - File Structure
 - WAR file (Web Application Archive, JAR format)
- Optional Deployment Descriptor
 - Located in \WEB-INF\web.xml
 - Simple XML structure (see next slide)
 - Description of the web application, icons, ...
 - Servlet descriptors
 - Tag Library descriptors
 - ...

Simple Deployment Descriptor

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE web-app  
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"  
  "http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">
```

```
<web-app>  
  <display-name>  
    Web Poll (Kai Toedter)  
  </display-name>  
  <description>  
    A Web Poll with Java, Servlets and JSP  
  </description>  
</web-app>
```

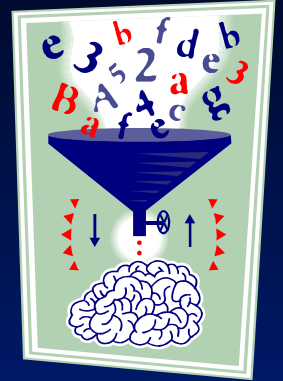
JDBC

- No Acronym
 - Unofficially: Java Database Connectivity
- Standard Java API for database access
- Free database software
 - MySQL
 - Download at www.mysql.com
- Free JDBC driver
 - Download at www.mysql.com/downloads/api-jdbc.html



JDBC Standard Procedure

- Load Driver
- Get Connection from database URL
- Create Statement
- Execute Query
- Process ResultSet
- Close
 - ResultSet
 - Statement
 - Connection



JDBC Example (1)

```
try {
    Class.forName( "org.gjt.mm.mysql.Driver" );
    Connection connection = DriverManager.getConnection(
        "jdbc:mysql:///Poll", user, password );
    Statement statement = connection.createStatement();
    ResultSet rs = statement.executeQuery(
        "SELECT * FROM InnovationTest ORDER BY id" );
    rs.beforeFirst();
    while ( rs.next() ) {
        int id = rs.getInt( "id" );
        String poll = rs.getString( "poll" );
        System.out.println( id + " : " + poll );
    }
}
catch (SQLException e) { ...
```

JDBC Example (2)

```
private boolean createTestData( Statement statement)
    throws java.sql.SQLException
{
    try {
        statement.executeUpdate("DROP TABLE Poll");
    }
    catch (SQLException e) {}
    statement.executeUpdate( "CREATE TABLE Poll (id int PRIMARY KEY,
                                poll char(32))" );
    for (int i = 0; i < pollCount; i++) {
        statement.executeUpdate( "INSERT INTO Poll VALUES ("
            + pollNr(i) + ", ' " + pollQuestion(i) + " )" );
    }
    return true;
}
```

Connection Pooling

- Opening/Closing a database connection is time-consuming
- Short queries may take less time

It might be a good idea to

- Probably preallocate connections
 - In the servlet init method
- Store connections
- Reuse connections

Connection Pooling Example (1)

```
public void init() {  
    // initialize JDBC driver, url, ...  
    jdbcConnPool = new JDBCConnectionPool( jdbcDriver, url,  
                                             user, password, initialConnections, ... )  
    // ...  
}  
  
public void destroy() {  
    jdbcConnPool.closeAll();  
}
```

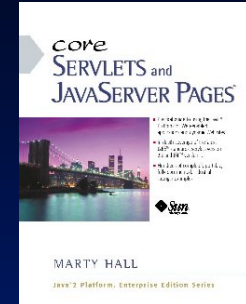
Connection Pooling Example (2)

```
public void doGet( ... ) {  
    // ...  
    JDBCConnection conn = jdbcConnPool.getConnection();  
    // Execute query  
    conn.release();  
  
    // ...  
}
```

Books

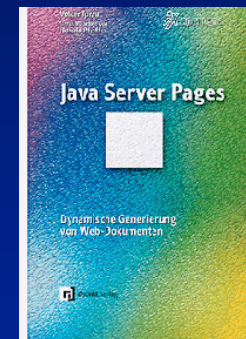
■ Core Servlets and Java Server Pages

- Author: Marty Hall
- ISBN: 0-13-089340
- www.coreservlets.com



■ Java Server Pages (German)

- Author: Volker Thurau
- ISBN 3-932588-66-5
- shannon.informatik.fh-wiesbaden.de/jsp



Links

■ Java Server Pages

- java.sun.com/products/jsp

■ Servlets

- java.sun.com/products/servlet

■ Tomcat

- jakarta.apache.org

■ Apache Web Server

- httpd.apache.org

■ Download slides

- www.toedter.com



Summary

- Poll Web Application
- Date Servlet/Date JSP
- Servlet Engine: Tomcat
- Web Server: Apache
- Basic Servlet/JSP
 - Request
 - Cookies
 - Authorization
 - Session
 - Tag Libraries
- Web Applications
- JDBC
- Books & Links

Discussion + Q&A

