Web Development with Java EE*

Introduction to Servlets and JSP

Federico Pecora

AASS Mobile Robotics Lab, Teknik

Room T2222

fpa@aass.oru.se

* Course adapted from A.J. Lillenthal

0 Administrative Issues – Lectures

- Mar 31 (Mon), 9 – 12 o’clock
  - L1 – Introduction to JSP & Servlets
- Apr 2 (Wed), 9 – 12 o’clock
  - L2 – JavaBeans and JDBC
- Apr 7 (Mon), 9 – 12 o’clock
  - *Definition of course project*
  - L3 – Custom Tags, JSTL and Deployment
- Apr 11 (Fri), 13 – 16 o’clock
  - L4 – XML Processing with JSP

0 Administrative Issues – Labs

- Mar 31 (Mon): 13 – 16 o’clock
  - Refreshing your Java (+ First JSP/Servlet exercise)
- Apr 1 (Tue): 13 – 16 o’clock
  Apr 2 (Thu): 13 – 16 o’clock
  - JSP/Servlet, JavaBeans, JDBC exercises
- Apr 4 (Fri): 13 – 16 o’clock
  Apr 9 (Wed): 13 – 19 o’clock
  Apr 16-18 (Wed-Fri): 13 – 19 o’clock
  Apr 21 (Mon): 13 – 19 o’clock
  Apr 23-25 (Wed-Fri): 13 – 16 o’clock

Unsupervised (regular visits)

0 Administrative Issues – Exam

- Apr 30 (Wed), 14 – 19
  - written examination
  - in English → dictionary is allowed
- May 5 (Mon)
  - *Projects due*
- May 12 (Mon), [Time TBD]
  - Distribution of Examinations (tentavisning)
- May TBD (TBD), [Time TBD]
  - re-exam
  - in English → dictionary is allowed

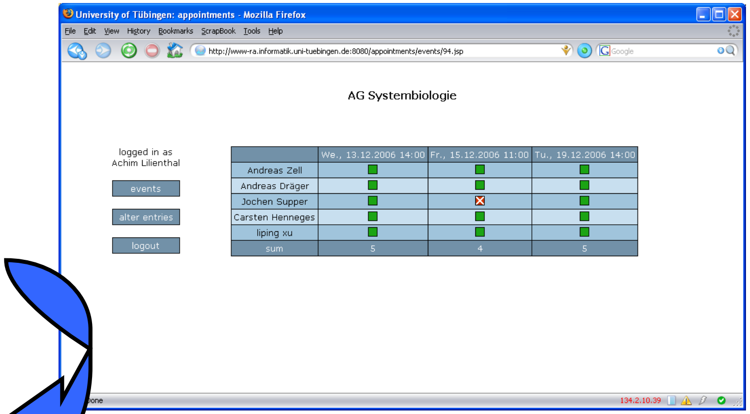
Contents

1. Introduction
   - Web Applications, Java EE, Servlets, JSP, Literature
2. Installation
   - JDK, Tomcat, NetBeans
3. JSP – Basics
4. Servlets – Basics
5. JSP – Details
   - Comments, Thread Safety, Predefined Variables, Scopes, JSP Actions, Exceptions, …

Introduction

1 Introduction – Web Applications

- Web Development
  - particularly dynamic content
  - interaction → web applications


University of Tübingen: appointments - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help
http://www-ra.informatik.uni-tuebingen.de:8080/appointments/events/94.jsp
Google
events
alter entries
logout
one
134.2.10.39

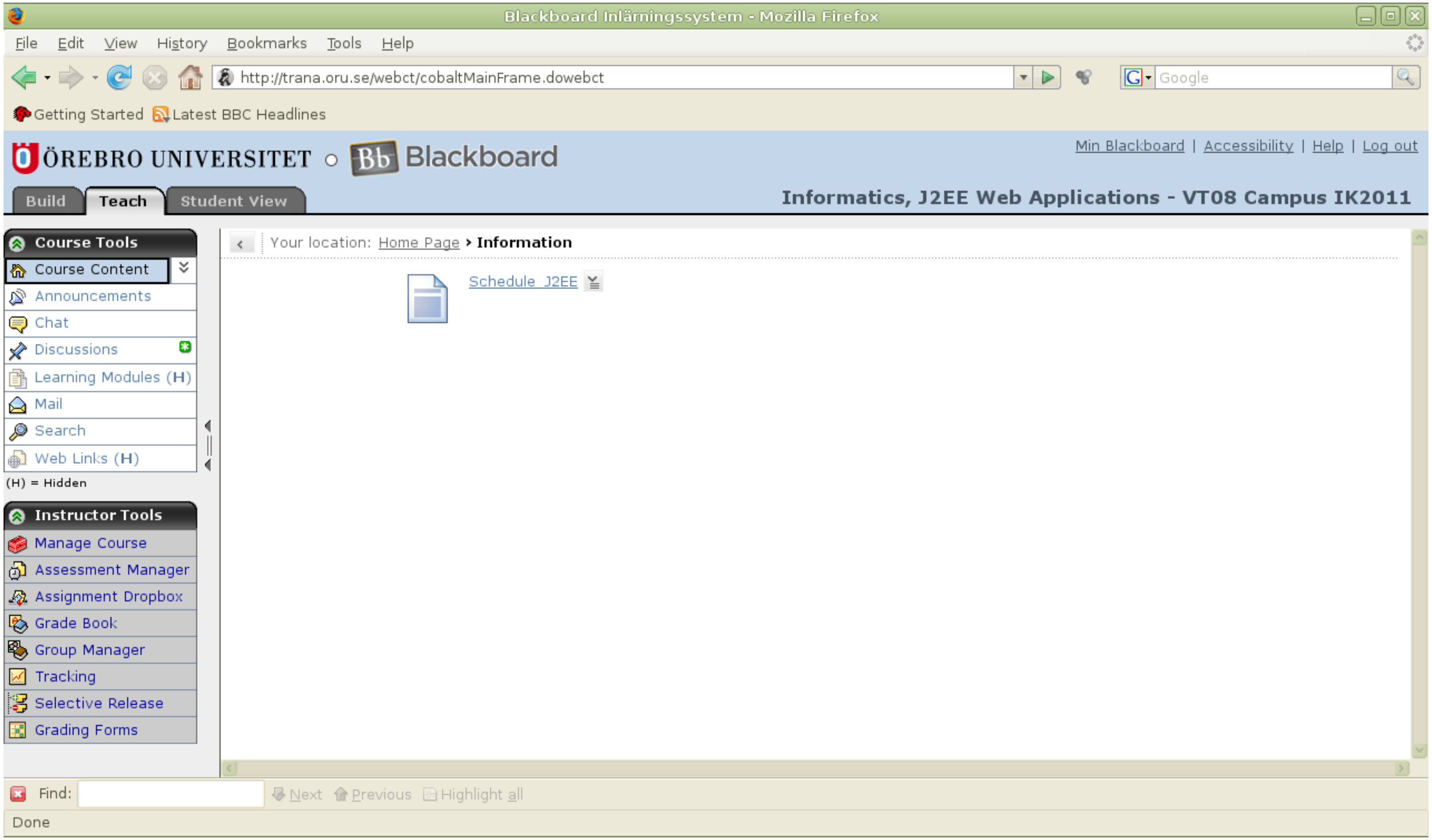
AG Systembiologie

logged in as
Achim Lilienthal

	We., 13.12.2006 14:00	Fr., 15.12.2006 11:00	Tu., 19.12.2006 14:00
Andreas Zell	🟩	🟩	🟩
Andreas Dräger	🟩	🟩	🟩
Jochen Supper	🟩	☒	🟩
Carsten Henneges	🟩	🟩	🟩
liping xu	🟩	🟩	🟩
sum	5	4	5

- Web Applications
  - delivered to the users over a network
  - require a thin client (web browser) only – universal client
  - updating and maintaining is simple
  - platform independent in theory (referring to the client)
  - webmail, wikis, weblogs, online auctions (ebay), …

1 Introduction – Web Applications



1 Introduction – Web Applications

- Web Applications versus “Normal” Applications
  - user input through web form elements
  - dynamically generated static HTML/XHTML pages
    - the sequence of pages provides an interactive experience
  - Java Script for client-side dynamic elements
- Limitations of Web Applications
  - everything has to be triggered by an HTML request
    - no progress bars, limited drag-and-drop, drawing on the screen, etc.
  - limited interface (thin client) has to be used

1 Introduction – Java EE

- Writing Web Applications Directly
  - CGI/Perl, CGI/Bash, CGI/C, …
  - PHP
  - mod_perl

Common Gateway Interface

PHP: Hypertext Processor

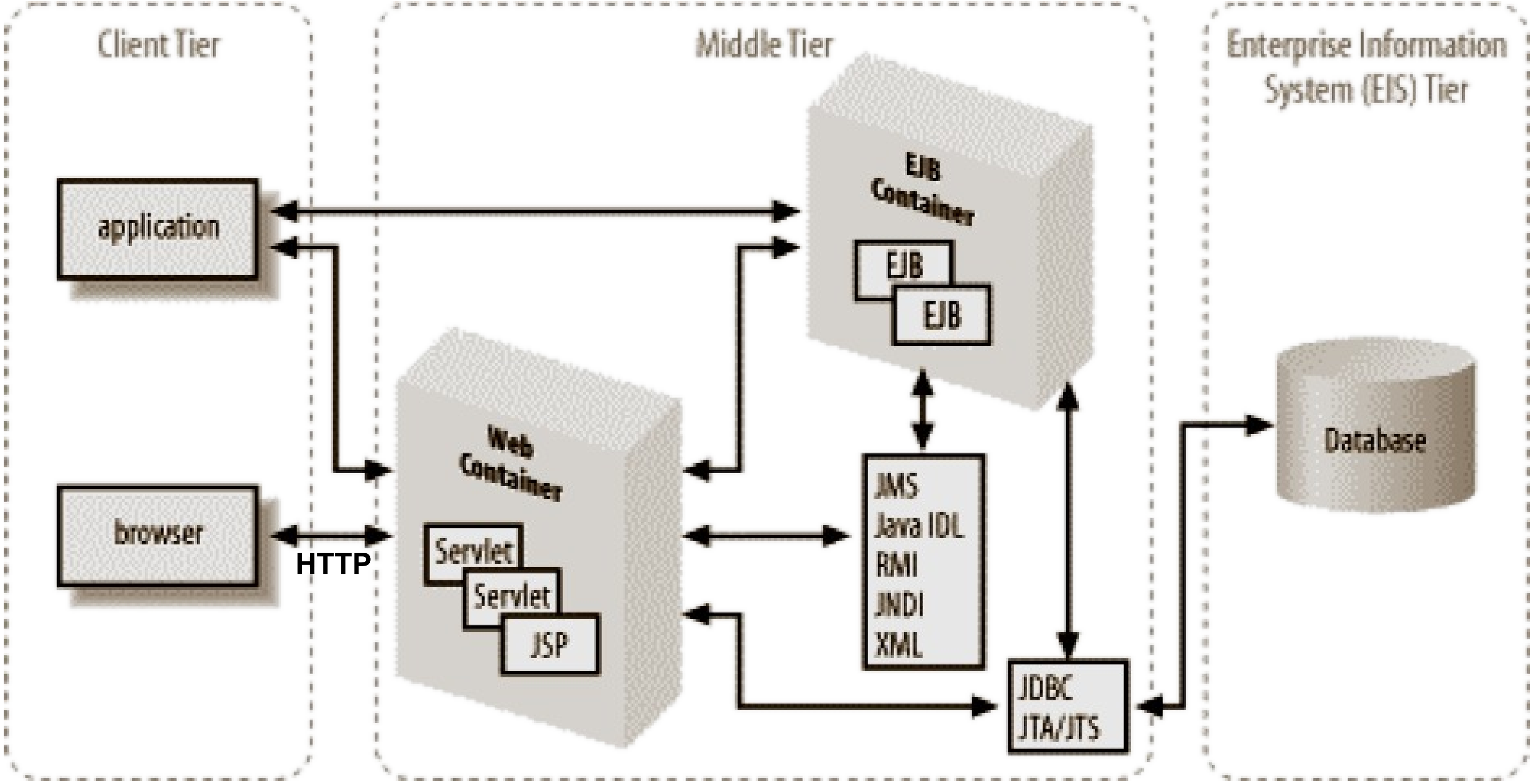
- Writing Web Applications Using a Framework
  - Java Servlets, JSP (Java EE)
  - Apache Struts
  - ASP.NET
  - Java Server Faces
  - ... e.g., AJAX (Asynchronous JavaScript and XML)

1 Introduction – Java EE

- Java Platform, Enterprise Edition (Java EE)
  - Formerly known as J2EE (J2EE 1.4 SDK, Java EE 5)
  - platform/specification for multi-tier server applications
  - runs on an application server (web container)
  - based on modular components
  - includes several APIs:
    - Servlets, Java Server Pages (JSP)
    - Java Database Connection (JDBC, SQL)
    - Java API for XML Parsing (JAXP)
    - Java Authentication and Authorization Service (JAAS)
    - Enterprise Java Beans (EJB)

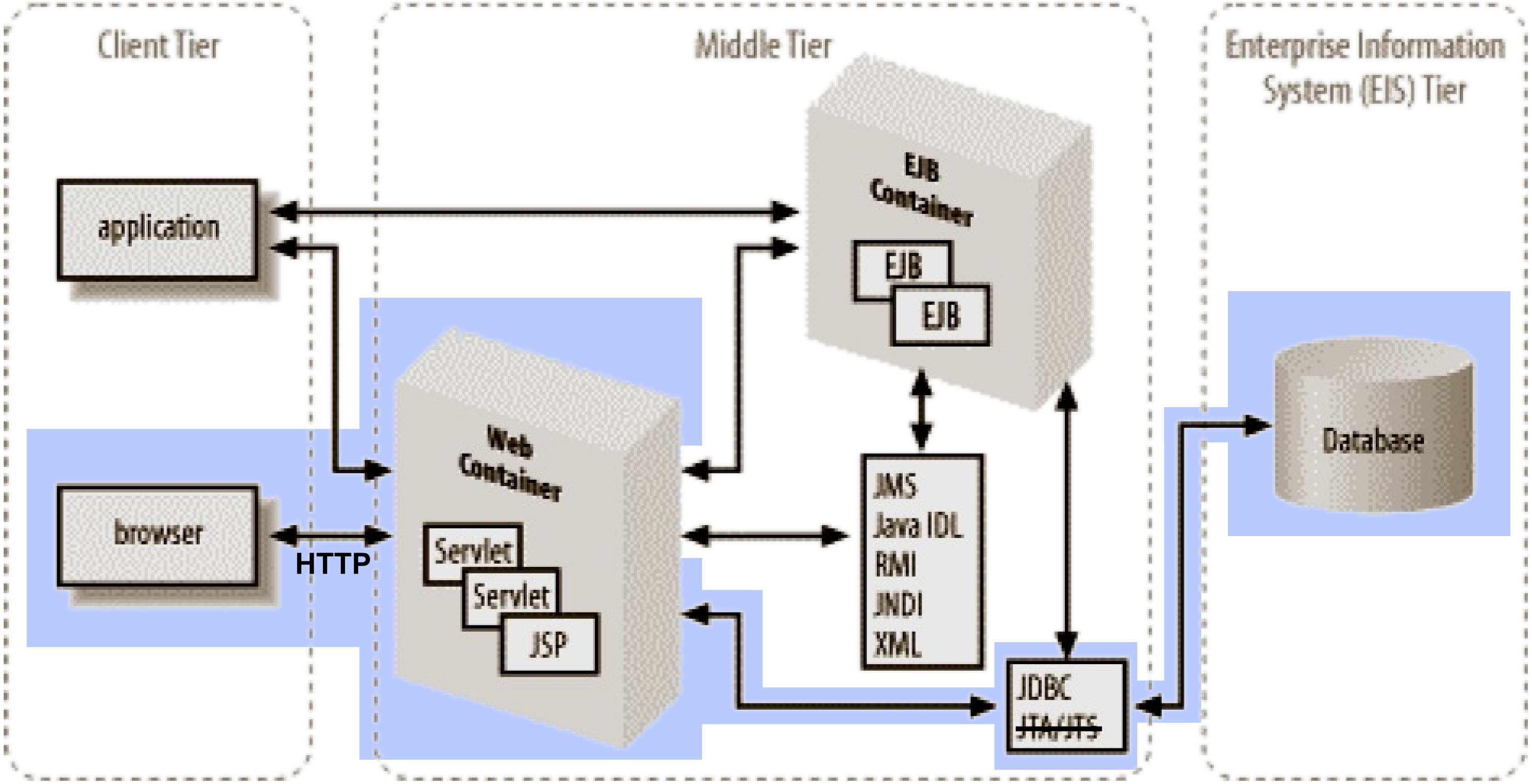
1 Introduction – Java EE

Java EE Architecture



1 Introduction – Java EE

Java EE Architecture



1 Introduction – Java EE

- Java EE Benefits
  - supports multiple platforms (as opposed to .NET)
  - an implementation can be downloaded for free from Sun
  - many open source tools are available:
    - IDEs: NetBeans, Eclipse
    - Web Container: Jakarta Tomcat (implements Servlets and JSP)
    - Development Tools: Apache Ant
    - Testing Tools: JMeter, JUnit
    - Web Application Framework: Apache Struts

1 Introduction – Java EE

- Current Specification Versions
  - Java 2 Enterprise Edition ver. 1.4 (“J2EE 1.4”)
    Stable Release: 2003/11/23
  - Java Platform Enterprise Edition ver. 5 (“Java EE 5”)
    Stable Release: 2006/05/11
  - Java Platform Enterprise Edition ver. 6 (“Java EE 6”)
    In Progress: expected in 2008

http://java.sun.com

1 Introduction – Servlets

- Servlets = Particular Java Programs
  - run on a server
  - act as a middle layer
    - HTTP request ↔ Servlet ↔ application on the server (database)
  - advanced API (high level networking utilities)
    - session tracking
    - cookies
    - HTTP headers

1 Introduction – Servlets versus CGI

- Efficiency
  - CGI: a new process for each HTTP request
  - Servlets: a Java thread for each HTTP request
- Capabilities
  - Servlets can share data and database connections, and can maintain information between HTTP requests
- Security
  - string bounds are not checked in CGI/C, CGI/C++
  - often executed by shells ("\" or ";" have to be filtered!)

1 Introduction – JSP

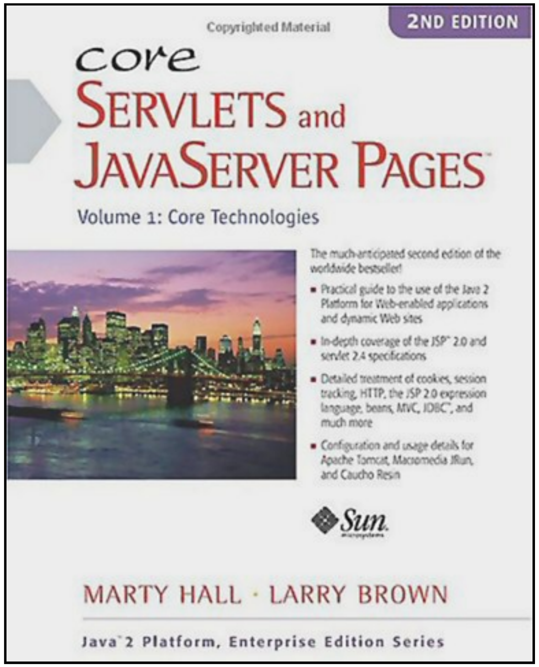
- Java Server Pages
  - JSP = “HTML with Java code” (dynamic content)
  - Servlets = “Java code with HTML” (produces HTML)
  - JSP is a direct extension of Java Servlets
- Benefits of JSP
  - dynamic parts separated from static parts
  - inherits the advantages of Java
    - capabilities, security, portability
  - sometimes more convenient to use than Servlets

1 Literature

- JSP and Servlets
  - *JavaServer Pages*
    Hans Bergsten (O'Reilly)
  - focus almost entirely on JSP and supporting technologies
  - explains Servlets only if necessary to understand JSP
  - covers the largest part of the course (in comparison with the other books)

1 Literature

- JSP and Servlets
  - *Core Servlets and JavaServer Pages* Marty Hall, Larry Brown (Sun)
  - Free Online Version of the 1st Edition
  - focus on Servlets and how to handle HTTP sessions (first part of the book)
  - less JSP than in other books
  - some relevant JSP issues are not addressed in Volume 1
  - database connection pooling as a supporting technology (third part of the book).



Copyrighted Material

2ND EDITION

core SERVLETS and JAVASERVER PAGES

Volume 1: Core Technologies

The much-anticipated second edition of the worldwide bestseller!

- Practical guide to the use of the Java 2 Platform for Web-enabled applications and dynamic Web sites
- In-depth coverage of the JSP™ 2.0 and servlet 2.4 specifications
- Detailed treatment of cookies, session tracking, HTTP, the JSP 2.0 expression language, beans, MVC, JDBC™, and much more
- Configuration and usage details for Apache Tomcat, Macromedia JRun, and Caucho Resin

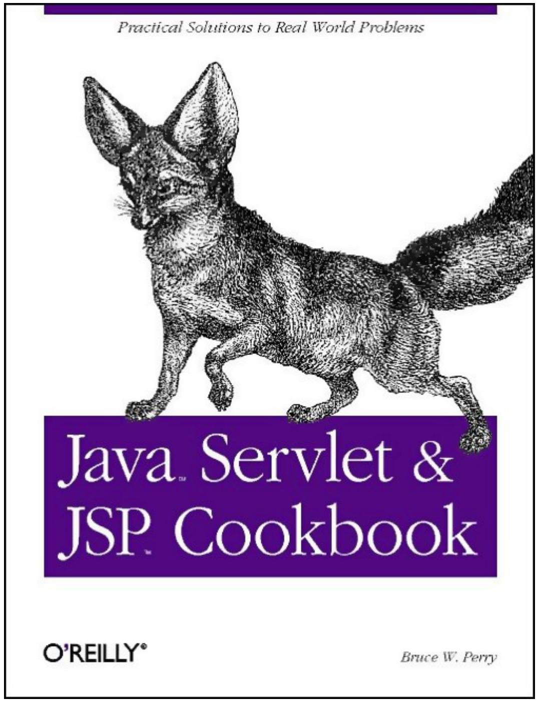
Sun microsystems

MARTY HALL · LARRY BROWN

Java™2 Platform, Enterprise Edition Series

1 Literature

- JSP and Servlets
  - *Java Servlet & JSP Cookbook*
    Bruce W.Perry (O'Reilly)
  - follows a practical approach
  - many examples
  - some of the examples address
    quite advanced issues
    (cannot be found in the other books)
  - not designed as a course book in the first place


Practical Solutions to Real World Problems
Java Servlet & JSP Cookbook
O'REILLY
Bruce W. Perry

1 Literature

- Specifications
  - Java Server Pages Specification 2.1
    - Delisle, Luehe, Roth (Sun)
    - Released May 8th 2006
      http://jcp.org/aboutJava/communityprocess/final/jsr245
  - Java Servlets Specification 2.4
    - Coward, Yoshida (Sun)
    - Released Nov 24th 2003 (maintenance release 2.5 Sep 11 2007)
      http://jcp.org/aboutJava/communityprocess/mrel/jsr154

Installation

2 Installation

- Requirements
  - Web Server ⇒ Apache/Jakarta Tomcat
    - version 6.0.16 (stable)
    - requires Servlet/JSP Spec 2.5/2.1
  - Java Servlets and JSP
    - Servlets version 2.5 (included with Tomcat 6)
    - JSP version 2.1 (included with Tomcat 6)
  - Texteditor, IDE
    - NetBeans, Eclipse, Jbuilder, ...

2 Installation – JDK

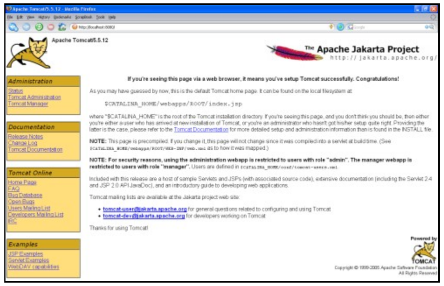
- JDK 6.0 (= JDK 1.6)
  - for development (the JRE only is sufficient to run Tomcat)
  - JDK 6.0 Update 5 from Sun
    - link: http://java.sun.com/
    - Linux, Windows (jdk-6u5-windows-i586-p.exe)
  - install (Windows)
    - for example to C:\Program Files\Java\jdk1.6.0_5
  - set the environment variable JAVA_HOME
    - JAVA_HOME=C:\Program Files\Java\jdk1.6.0_5

2 Installation – Tomcat

- Tomcat 6.0.16
  - download zip file from the Apache Tomcat page
    - http://tomcat.apache.org/download-60.cgi
    - apache-tomcat-6.0.16.zip
  - unzip into a convenient folder
    - for example to C:\Program Files\apache-tomcat-6.0.16
  - the Tomcat folder’s name: CATALINA_HOME
    - $CATALINA_HOME=C:\Program Files\apache-tomcat-6.0.16

2 Installation – Testing Tomcat

- Tomcat 6.0.16
  - start Tomcat
    - ${CATALINA_HOME}\bin\startup.bat (Windows)
    - ${CATALINA_HOME}/bin/startup.sh (Linux)
  - request the Tomcat start page
    - http://localhost:8080/



If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
$CATALINA_HOME/webapps/ROOT/index.jsp
```

where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Jakarta project web site:

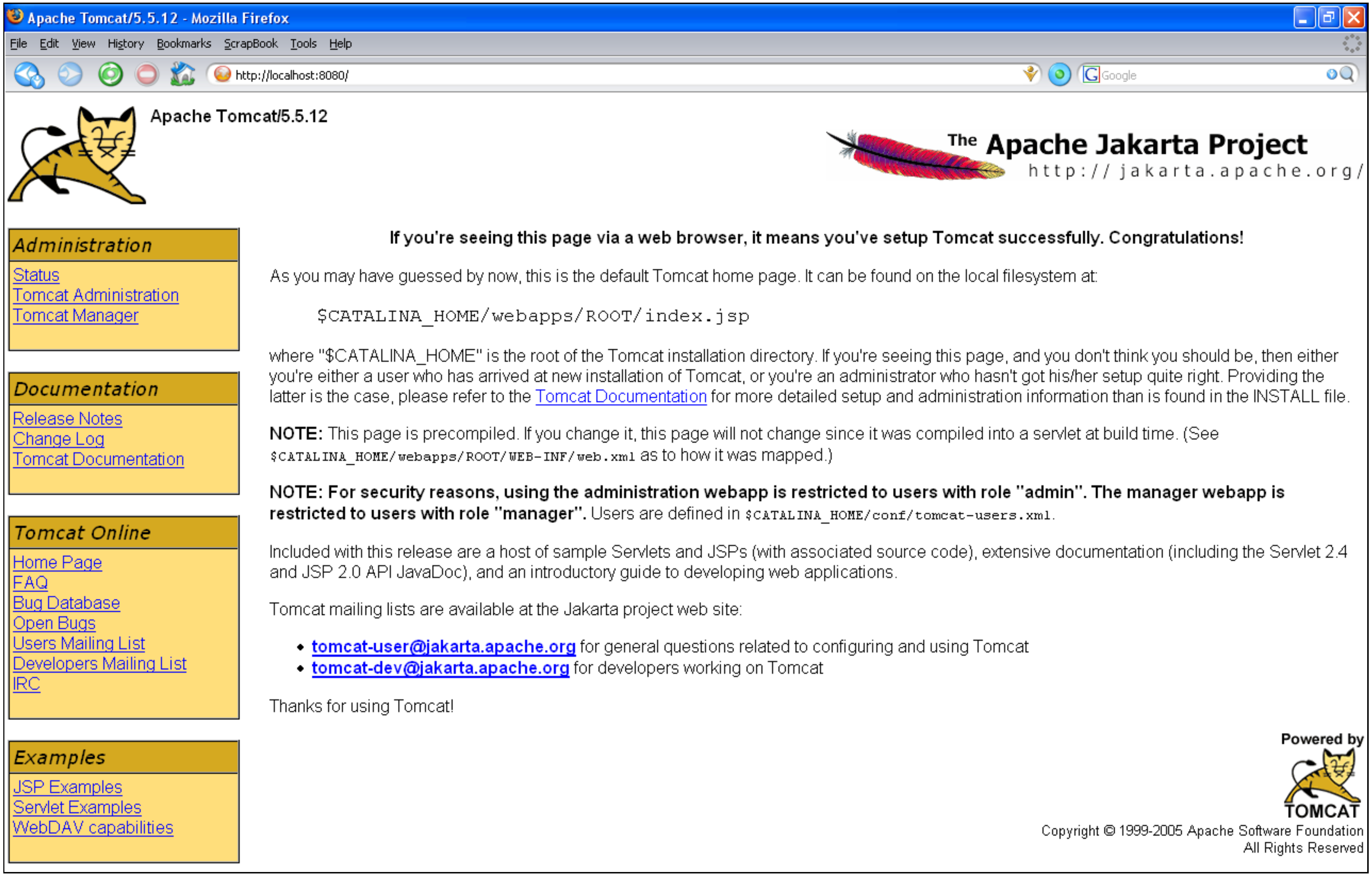
- **tomcat-user@jakarta.apache.org** for general questions related to configuring and using Tomcat
- **tomcat-dev@jakarta.apache.org** for developers working on Tomcat

Thanks for using Tomcat!

Copyright © 1999-2005 Apache Software Foundation
All Rights Reserved

  - shutdown Tomcat
    - ${CATALINA_HOME}\bin\shutdown.bat (Windows)
    - ${CATALINA_HOME}/bin/shutdown.sh (Linux)

2 Installation – Testing Tomcat


Apache Tomcat/5.5.12 - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help
http://localhost:8080/
Google

Apache Tomcat/5.5.12

The Apache Jakarta Project
http://jakarta.apache.org/

Administration

Status
Tomcat Administration
Tomcat Manager

Documentation

Release Notes
Change Log
Tomcat Documentation

Tomcat Online

Home Page
FAQ
Bug Database
Open Bugs
Users Mailing List
Developers Mailing List
IRC

Examples

JSP Examples
Servlet Examples
WebDAV capabilities

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
    $CATALINA_HOME/webapps/ROOT/index.jsp
```

where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Jakarta project web site:

- **tomcat-user@jakarta.apache.org** for general questions related to configuring and using Tomcat
- **tomcat-dev@jakarta.apache.org** for developers working on Tomcat

Thanks for using Tomcat!

2 Installation – Tomcat Configuration

- Changing the port
  - edit ${CATALINA_HOME}\conf\server.xml

```
<Server port="8005" shutdown="SHUTDOWN">
  …
  <Service name="Catalina">
    …
    <!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
    <Connector port="8080" maxHttpHeaderSize="8192"
      maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
      enableLookups="false" redirectPort="8443" acceptCount="100"
      connectionTimeout="20000" disableUploadTimeout="true"
    />
  </Service>
</Server>
```

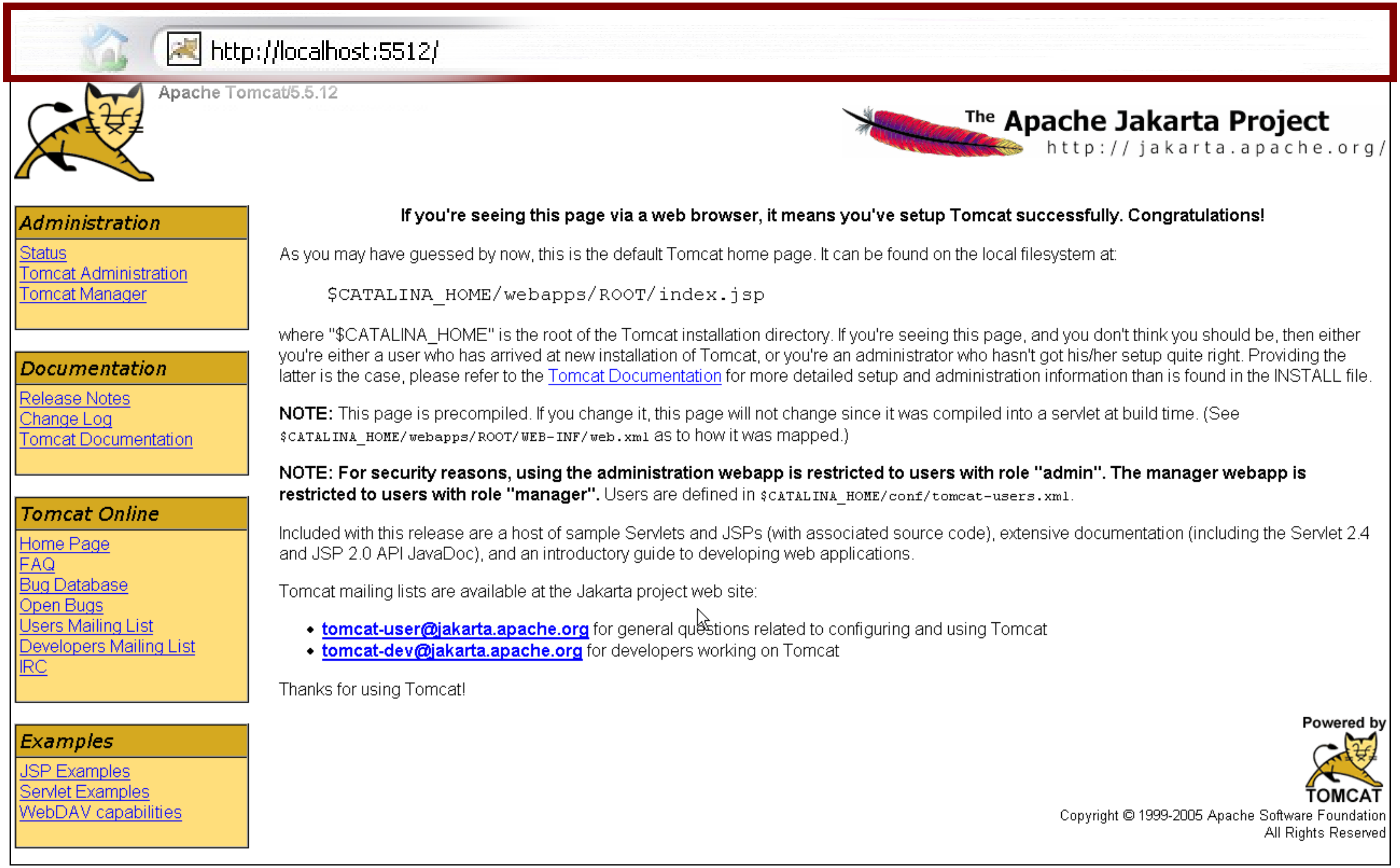
2 Installation – Tomcat Configuration

- Changing the port to 5512
  - edit ${CATALINA_HOME}\conf\server.xml

```
<Server port="8005" shutdown="SHUTDOWN">
  …
  <Service name="Catalina">
    …
    <!-- Define a non-SSL HTTP/1.1 Connector on port 5512 (def:8080) -->
    <Connector port="5512" maxHttpHeaderSize="8192"
      maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
      enableLookups="false" redirectPort="8443" acceptCount="100"
      connectionTimeout="20000" disableUploadTimeout="true"
    />
  </Service>
</Server>
```

→ **Restart Tomcat!**

2 Installation – Testing Tomcat



http://localhost:5512/

Apache Tomcat/5.5.12

The Apache Jakarta Project
http://jakarta.apache.org/

Administration

- Status
- Tomcat Administration
- Tomcat Manager

Documentation

- Release Notes
- Change Log
- Tomcat Documentation

Tomcat Online

- Home Page
- FAQ
- Bug Database
- Open Bugs
- Users Mailing List
- Developers Mailing List
- IRC

Examples

- JSP Examples
- Servlet Examples
- WebDAV capabilities

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
$CATALINA_HOME/webapps/ROOT/index.jsp
```

where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Jakarta project web site:

- **tomcat-user@jakarta.apache.org** for general questions related to configuring and using Tomcat
- **tomcat-dev@jakarta.apache.org** for developers working on Tomcat

Thanks for using Tomcat!

Powered by TOMCAT

Copyright © 1999-2005 Apache Software Foundation
All Rights Reserved

2 Installation – Tomcat Configuration

- Define users and set passwords
  - edit ${CATALINA_HOME}\conf\tomcat-users.xml

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="dari82" password="tomcat" roles="tomcat,role1"/>
  <user username="esthLp" password="tomcat" roles="role1"/>
</tomcat-users>
```

2 Installation – Tomcat Configuration

- Define users and set passwords
  - edit ${CATALINA_HOME}\conf\tomcat-users.xml

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="user"/>
  <role rolename="manager"/>
  <user username="admin" password="KikuNoGomon" roles="manager"/>
  <user username="guest" password="tomcat12345" roles="user"/>
</tomcat-users>
```

  - a manager role is required to access the Tomcat manager

2 Installation – Net Beans

- Net Beans 6.0.1
  - available bundled with the JDK
    - link: http://download.netbeans.org/netbeans/6.0/final/

JSP – Basics

3 JSP – Basics

- Integration into a web page
  - scripting elements

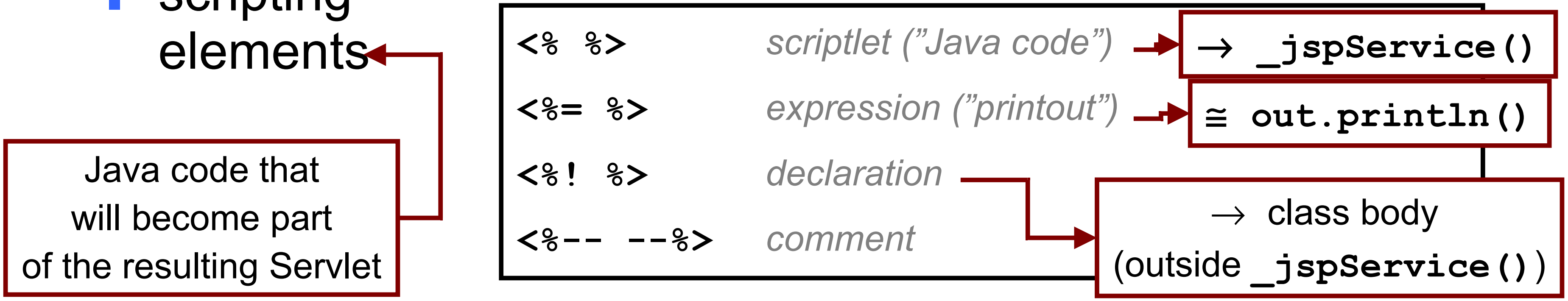
`<% %>`	*scriptlet ("Java code")*
`<%= %>`	*expression ("printout")*
`<%! %>`	*declaration*
`<%-- --%>`	*comment*

  - directives

```
<%@ include file="aFile" %>
<%@ page import="aClass" %>
<%@ page contentType="text/html" %>
<%@ page isThreadSafe="true" %>
<%@ taglib uri="address" prefix="prefix" %>
```

3 JSP – Basics

- Integration into a web page
  - scripting elements
  - directives



Java code that will become part of the resulting Servlet

```
<% %>          scriptlet ("Java code")     → _jspService()
<%= %>         expression ("printout")     ≅ out.println()
<%! %>         declaration                 → class body (outside _jspService())
<%-- --%>      comment
```

```
<%@ include file="aFile" %>
<%@ page import="aClass" %>
<%@ page contentType="text/html" %>
<%@ page isThreadSafe="true" %>
<%@ taglib uri="address" prefix="prefix" %>
```

3 JSP – Basics

- Integration into a web page
  - scripting elements

Java code that will become part of the resulting Servlet

```
<% %>         scriptlet ("Java code")      → _jspService()
<%= %>        expression ("printout")      ≅ out.println()
<%! %>        declaration                  → class body (outside _jspService())
<%-- --%>     comment
```

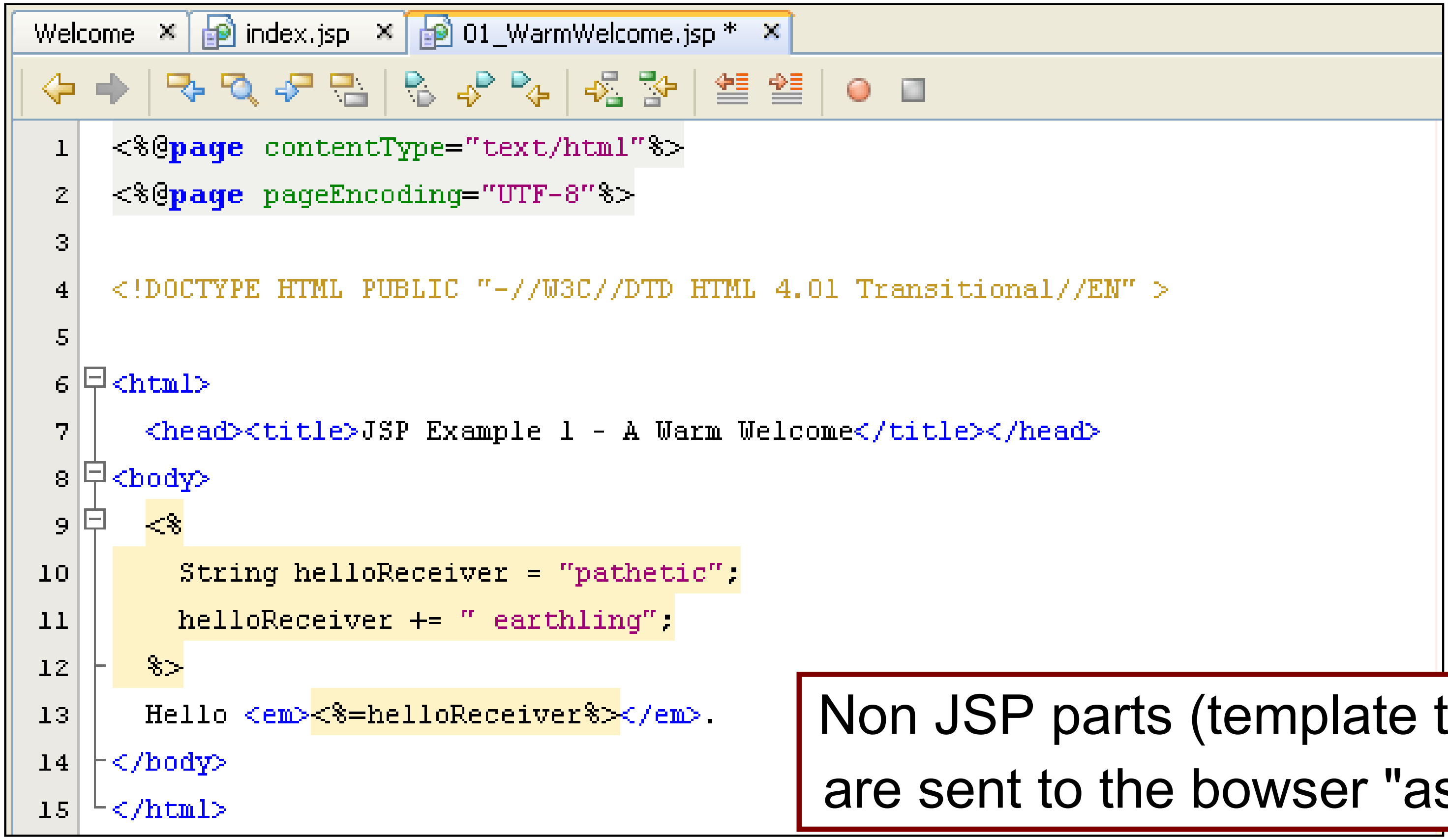
  - directives

allow to control the overall structure of the resulting Servlet (specify attributes of the page itself)

```
<%@ include file="aFile" %>
<%@ page import="aClass" %>
<%@ page contentType="text/html" %>
<%@ page isThreadSafe="true" %>
<%@ taglib uri="address" prefix="prefix" %>
```

3 JSP – First Examples

- Example 1 – “A Warm Welcome”

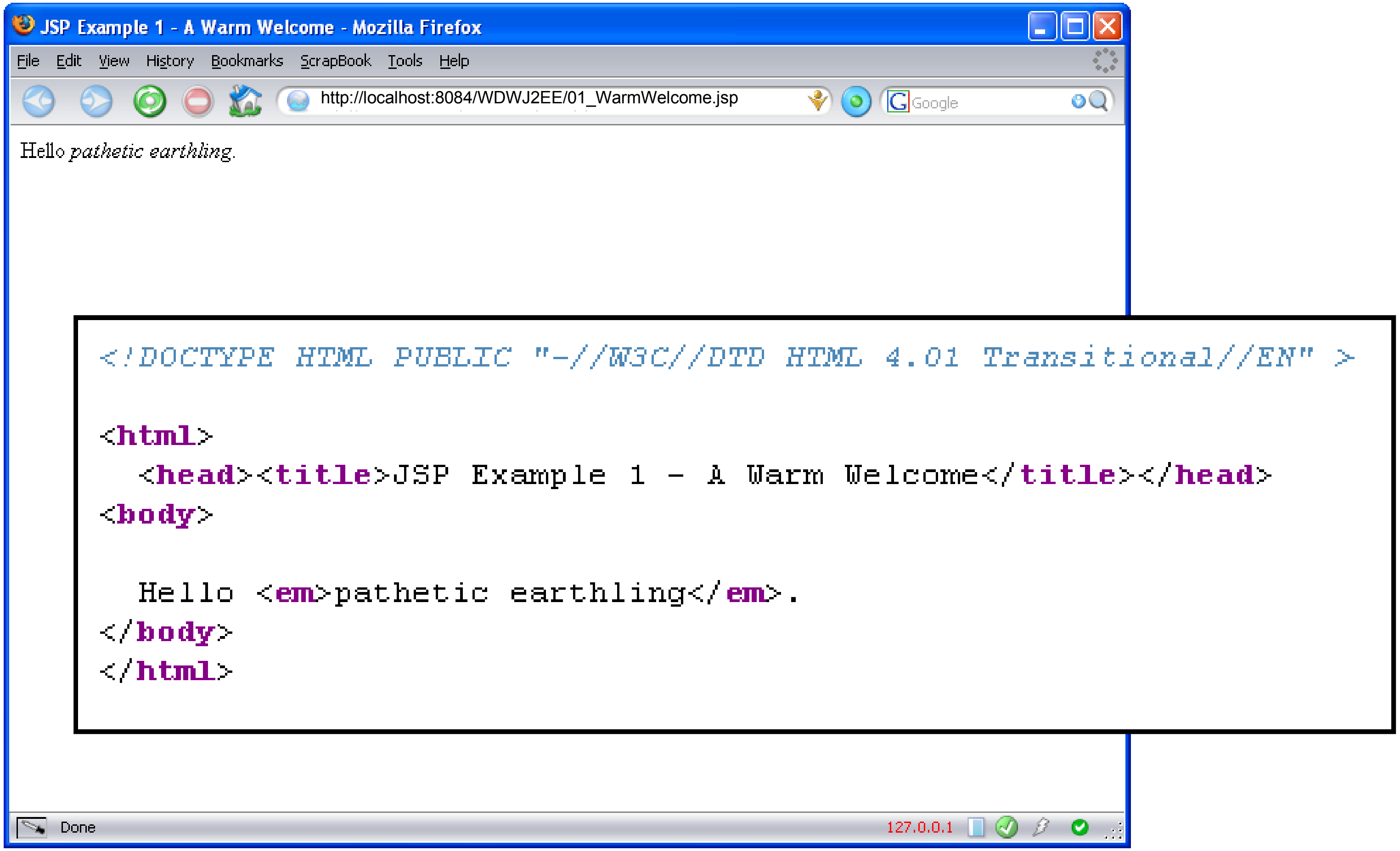


```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >

<html>
  <head><title>JSP Example 1 - A Warm Welcome</title></head>
<body>
  <%
    String helloReceiver = "pathetic";
    helloReceiver += " earthling";
  %>
  Hello <em><%=helloReceiver%></em>.
</body>
</html>
```

Non JSP parts (template text) are sent to the bowser "as is".

3 JSP – First Examples

- http://localhost:5512/JspExamples/JspExample01.jsp

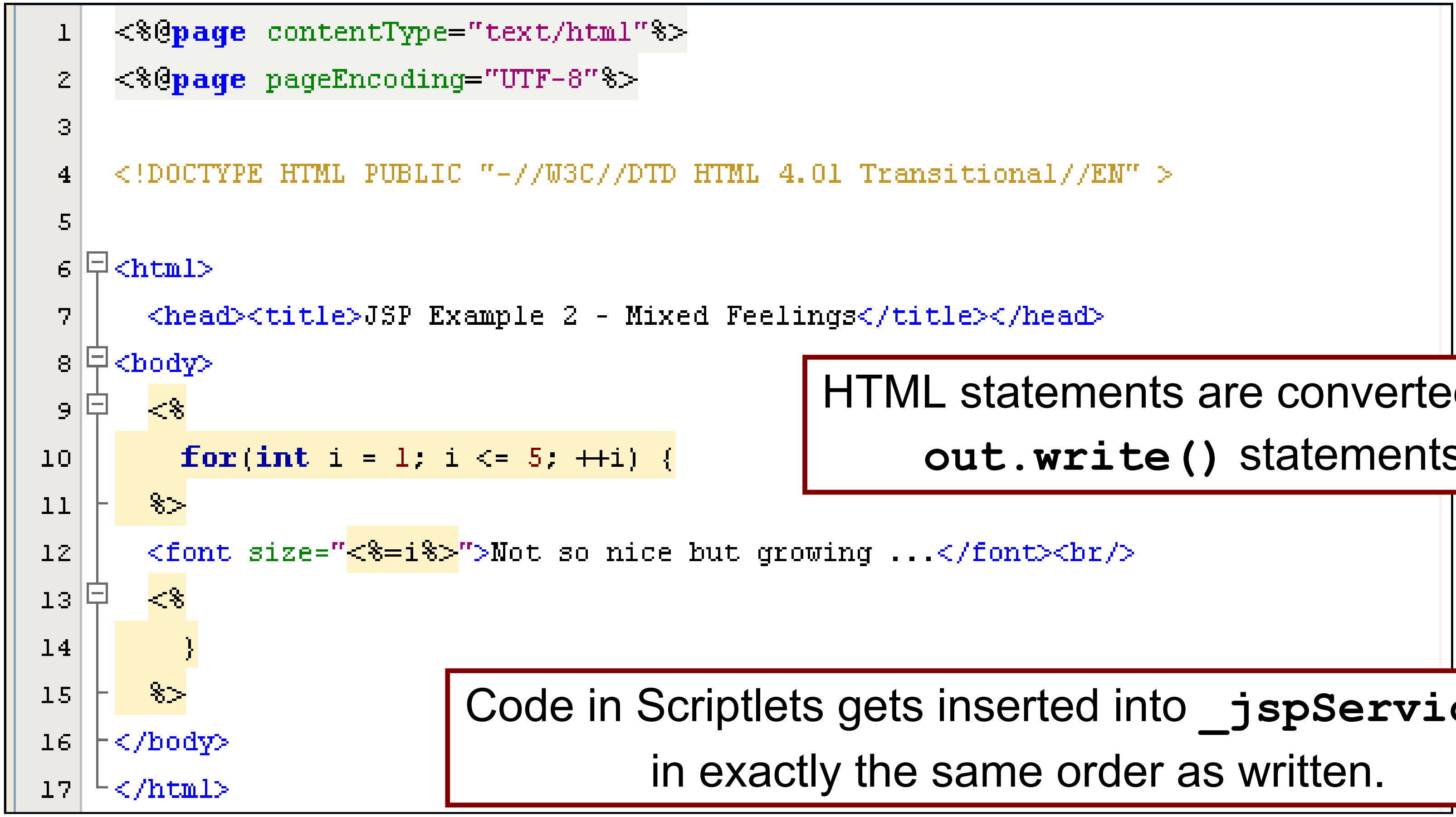

JSP Example 1 - A Warm Welcome - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help
http://localhost:8084/WDWJ2EE/01_WarmWelcome.jsp
Google
Done
127.0.0.1

Hello *pathetic earthling*.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >

<html>
  <head><title>JSP Example 1 - A Warm Welcome</title></head>
<body>

  Hello <em>pathetic earthling</em>.
</body>
</html>
```

3 JSP – First Examples

Example 2 – “Mixed Feelings”

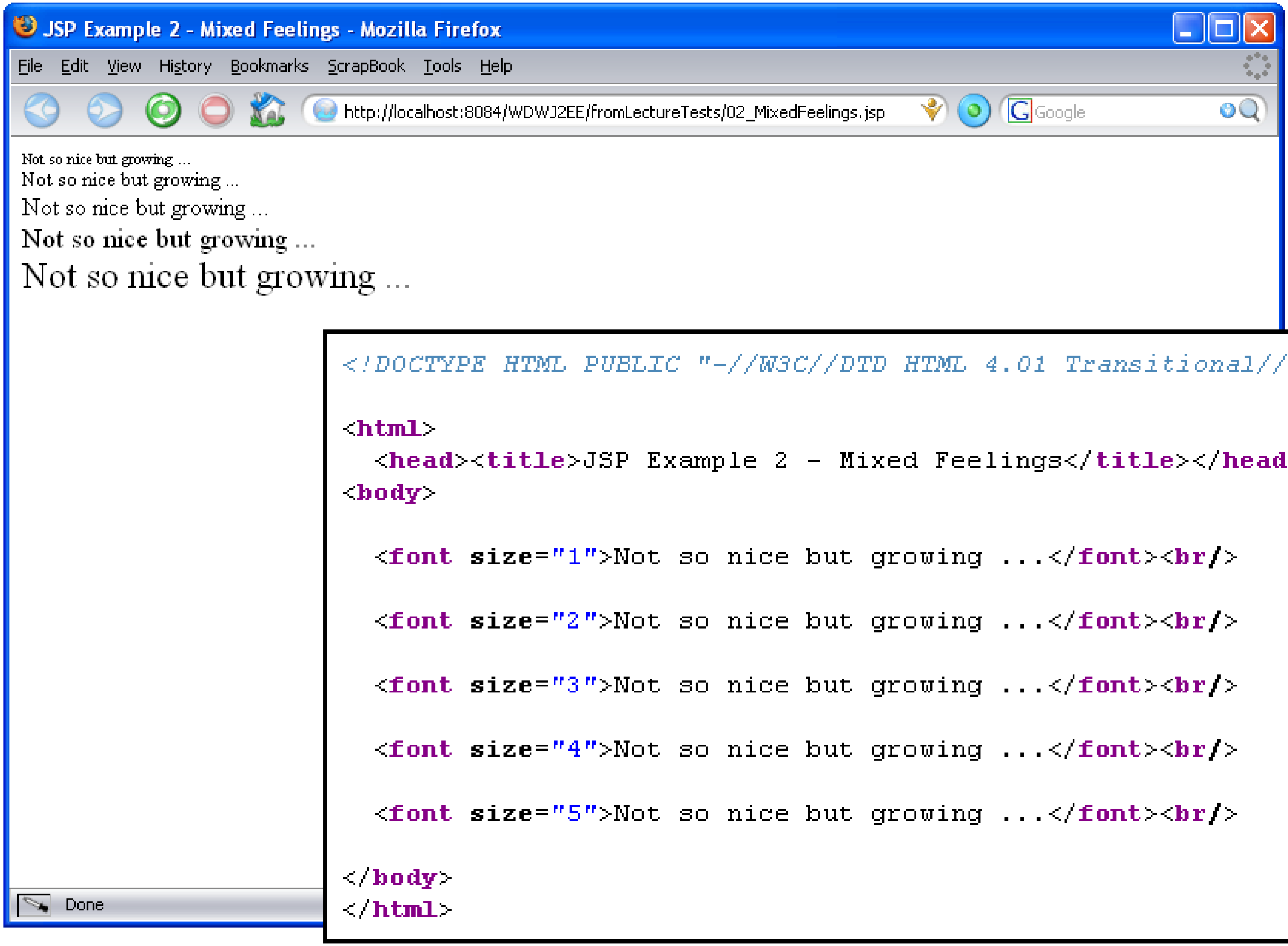


```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >

<html>
  <head><title>JSP Example 2 - Mixed Feelings</title></head>
<body>
  <%
    for(int i = 1; i <= 5; ++i) {
  %>
  <font size="<%=i%>">Not so nice but growing ...</font><br/>
  <%
    }
  %>
</body>
</html>
```

HTML statements are converted into **`out.write()`** statements.

Code in Scriptlets gets inserted into **`_jspService()`** in exactly the same order as written.

3 JSP – First Examples

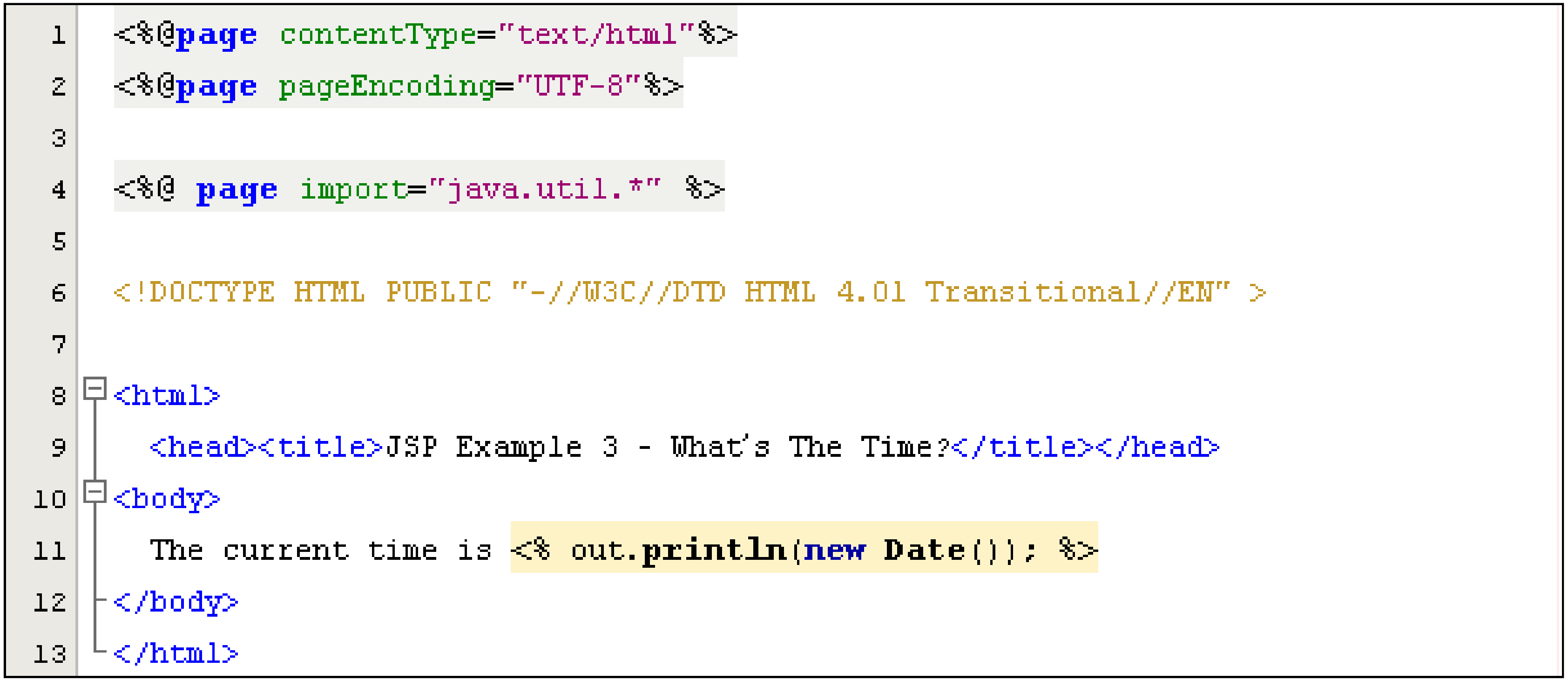

JSP Example 2 - Mixed Feelings - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help
http://localhost:8084/WDWJ2EE/fromLectureTests/02_MixedFeelings.jsp
Google
Done

Not so nice but growing ...
Not so nice but growing ...
Not so nice but growing ...
Not so nice but growing ...
Not so nice but growing ...

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >

<html>
  <head><title>JSP Example 2 - Mixed Feelings</title></head>
<body>

  <font size="1">Not so nice but growing ...</font><br/>

  <font size="2">Not so nice but growing ...</font><br/>

  <font size="3">Not so nice but growing ...</font><br/>

  <font size="4">Not so nice but growing ...</font><br/>

  <font size="5">Not so nice but growing ...</font><br/>

</body>
</html>
```

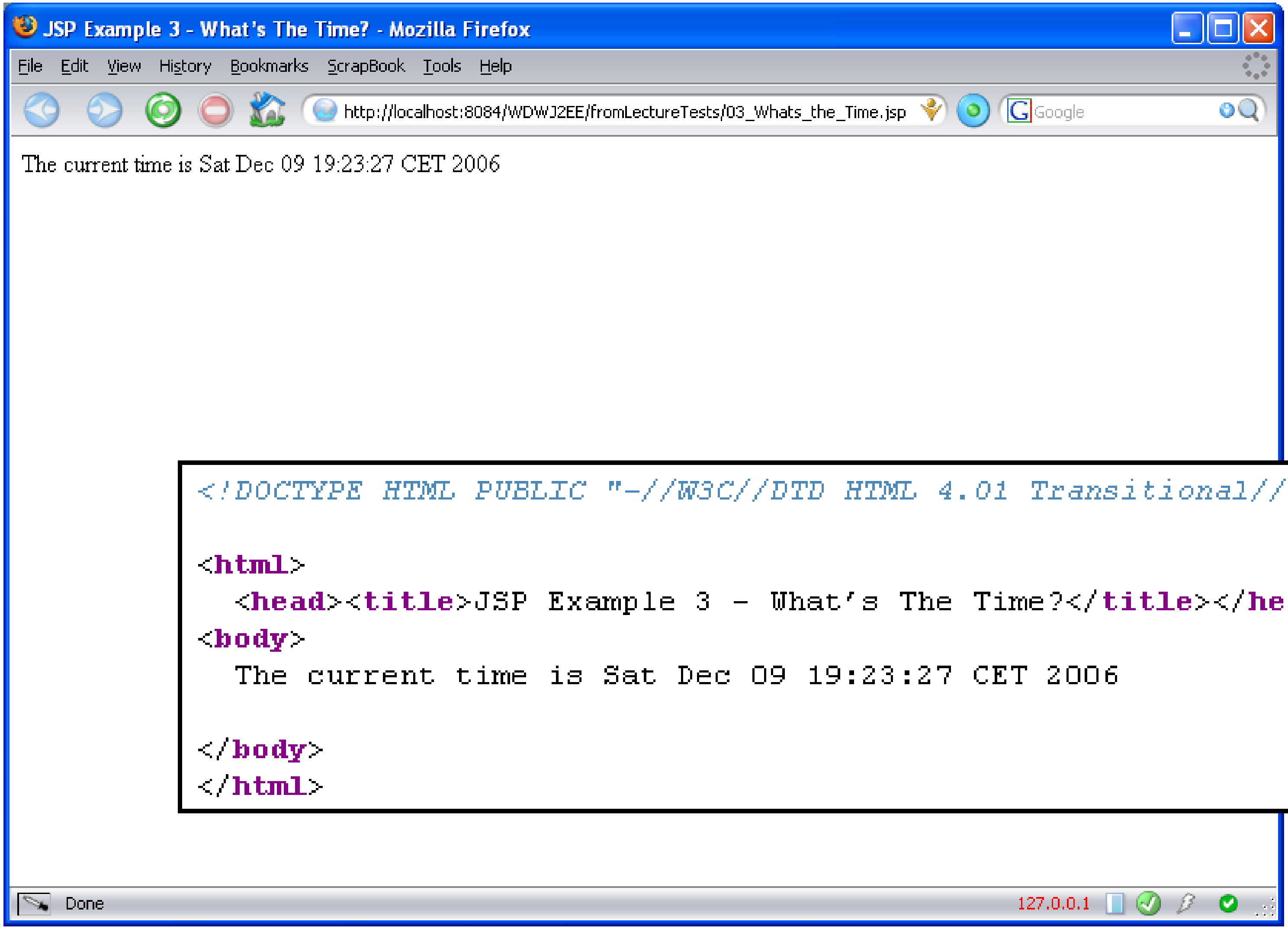
3 JSP – First Examples

Example 3 – “What’s The Time?”



```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<%@ page import="java.util.*"  %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >

<html>
  <head><title>JSP Example 3 - What's The Time?</title></head>
<body>
  The current time is <% out.println(new Date()); %>
</body>
</html>
```

3 JSP – First Examples

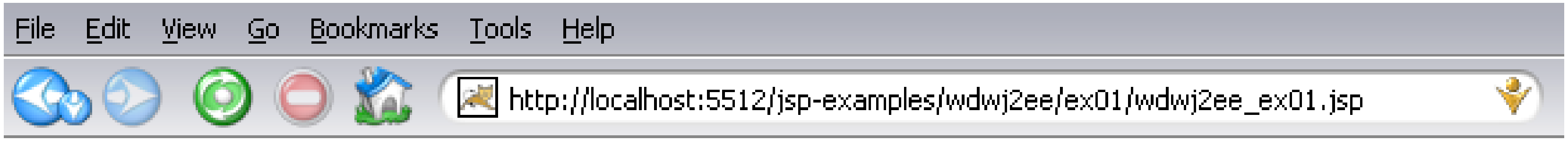

JSP Example 3 - What's The Time? - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help
http://localhost:8084/WDWJ2EE/fromLectureTests/03_Whats_the_Time.jsp
Google
Done
127.0.0.1

The current time is Sat Dec 09 19:23:27 CET 2006

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >

<html>
  <head><title>JSP Example 3 - What's The Time?</title></head>
<body>
  The current time is Sat Dec 09 19:23:27 CET 2006

</body>
</html>
```

3 JSP – Basics

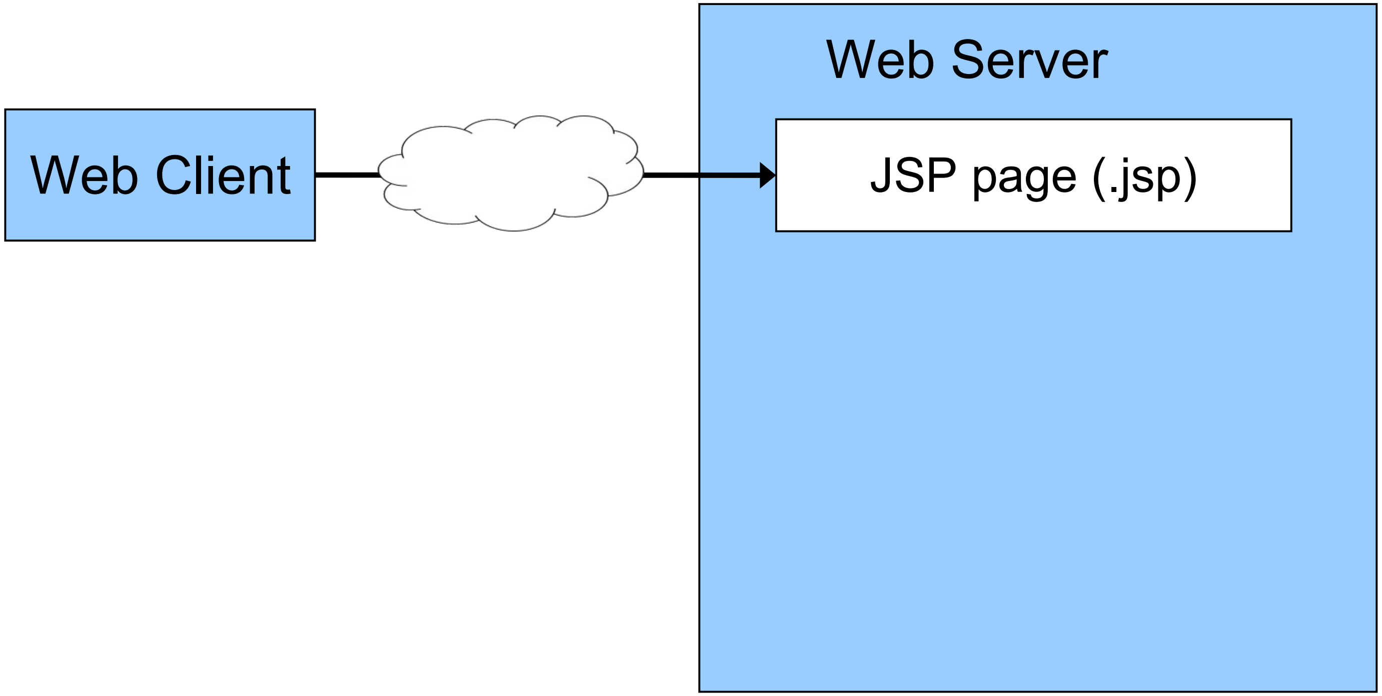
- How does it work?
  - a client requests a JSP page



http://localhost:5512/jsp-examples/wdwj2ee/ex01/wdwj2ee_ex01.jsp

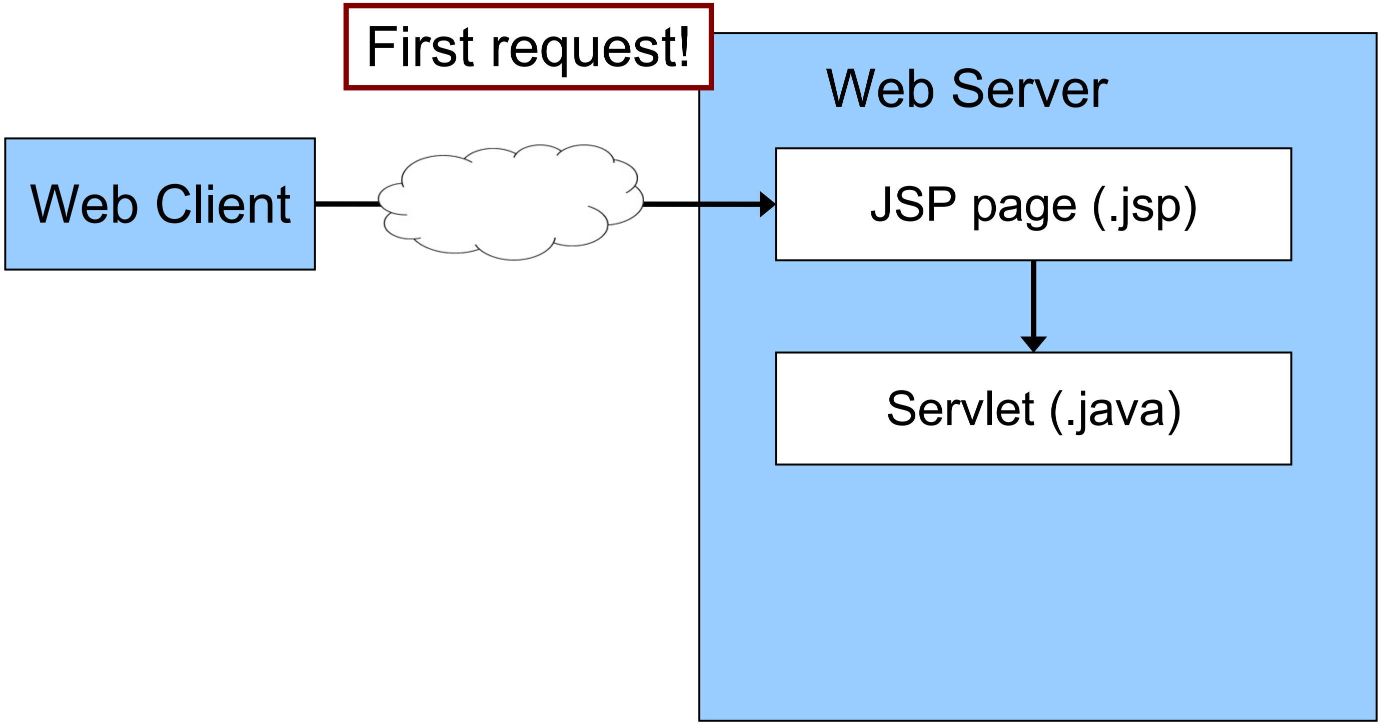
3 JSP – Basics

- How does it work?
  - a client requests a JSP page



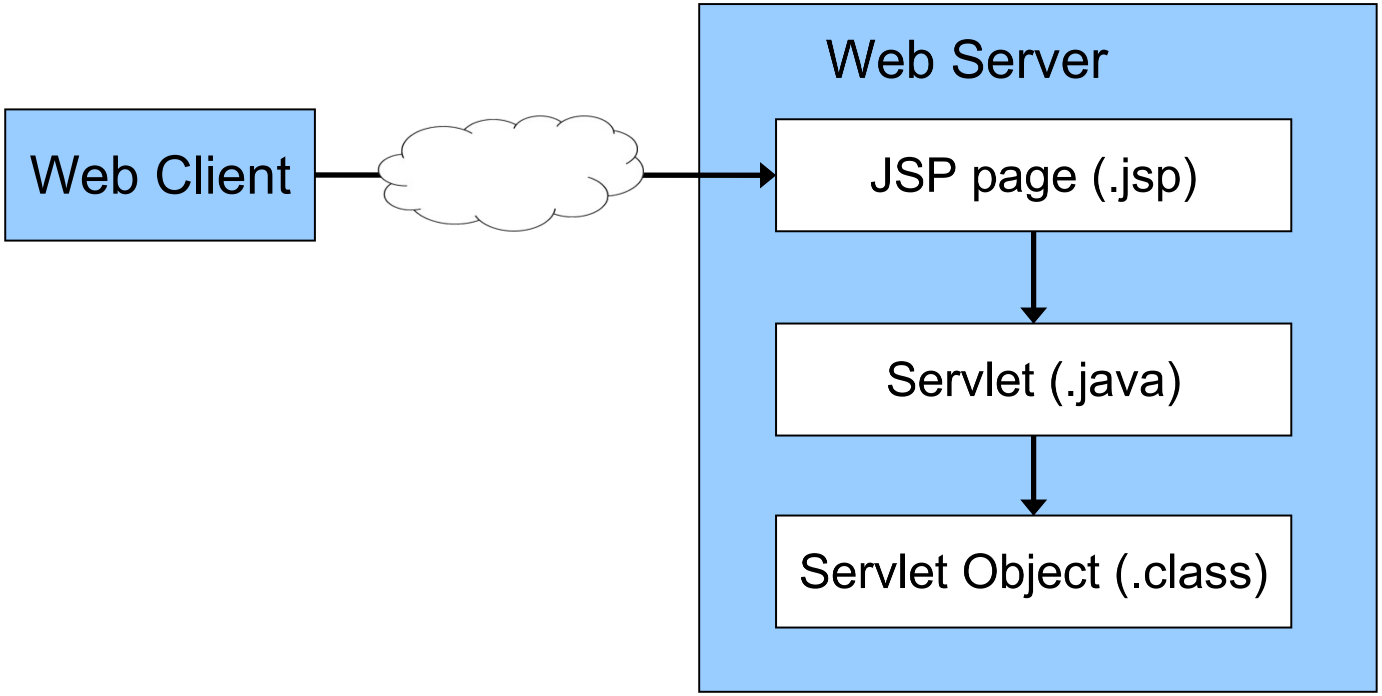
3 JSP – Basics

- How does it work?
  - the JSP engine generates a Servlet from the JSP page



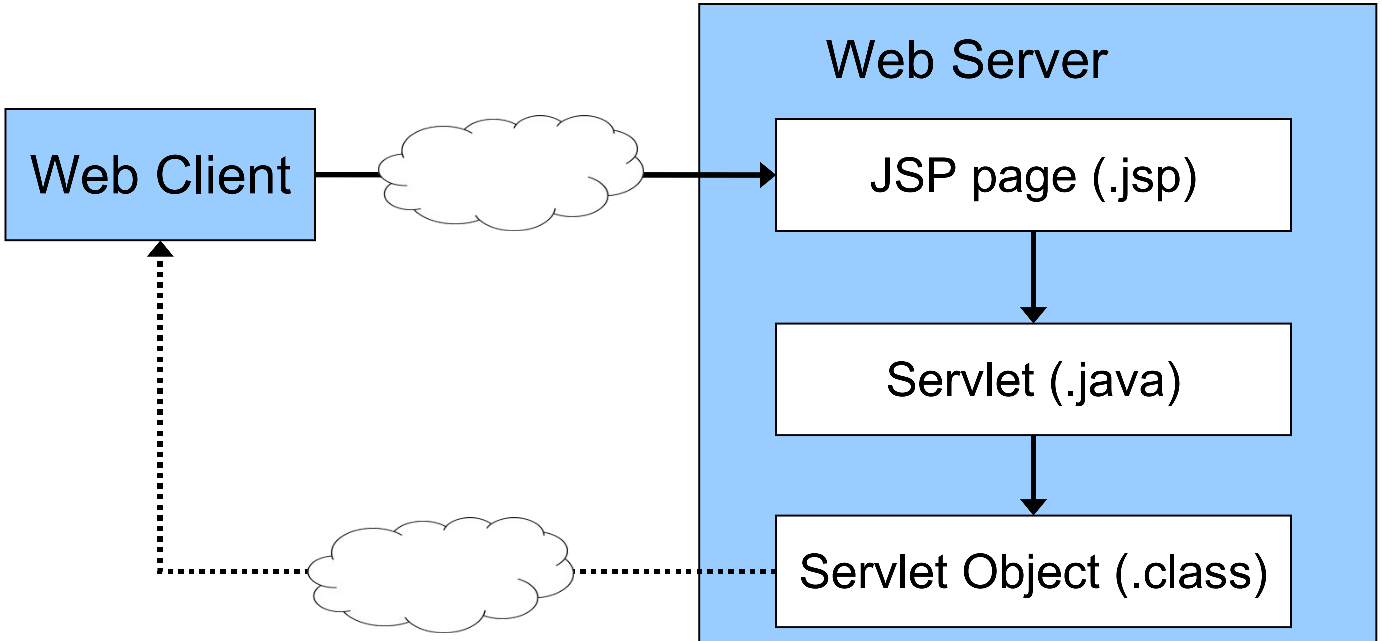
3 JSP – Basics

- How does it work?
  - the Server compiles and loads the Servlet object



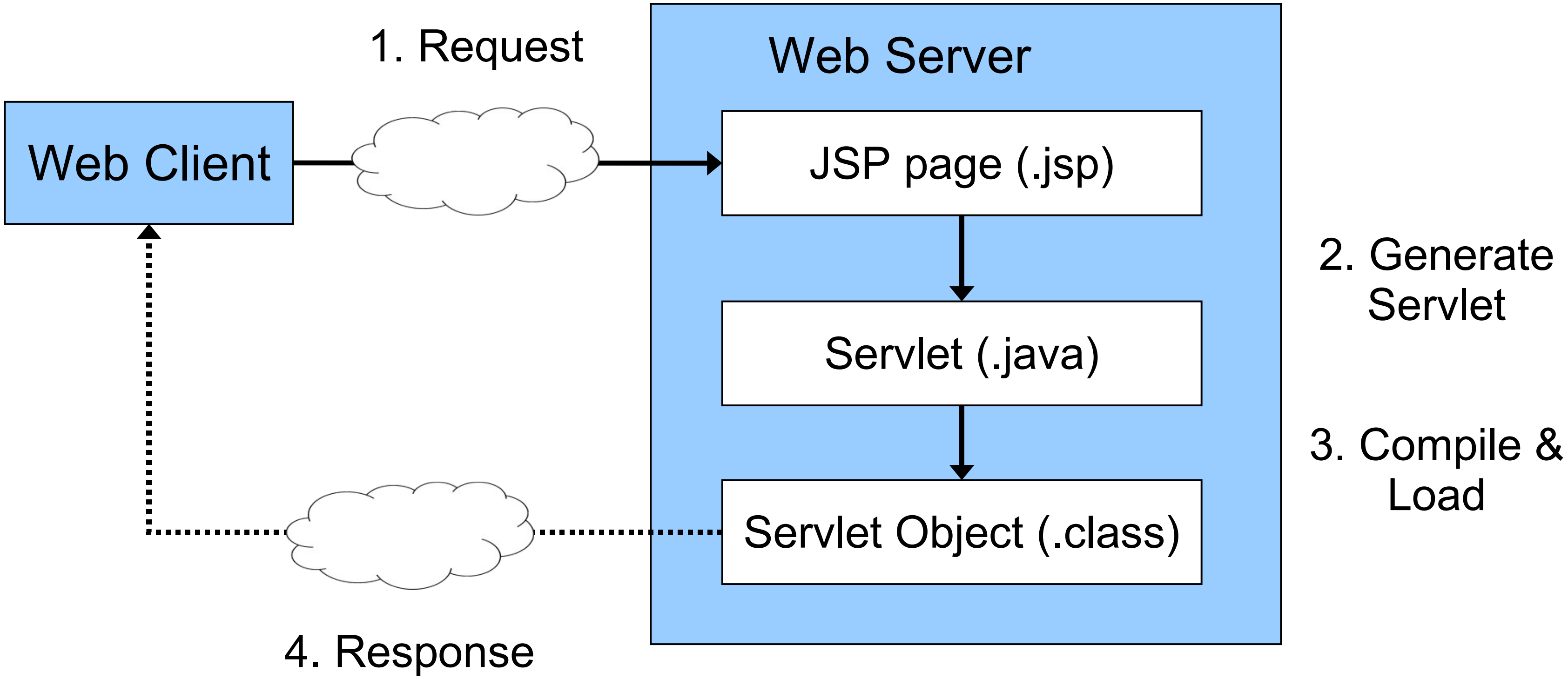
3 JSP – Basics

- How does it work?
  - the Servlet is executed (it sends back the response)



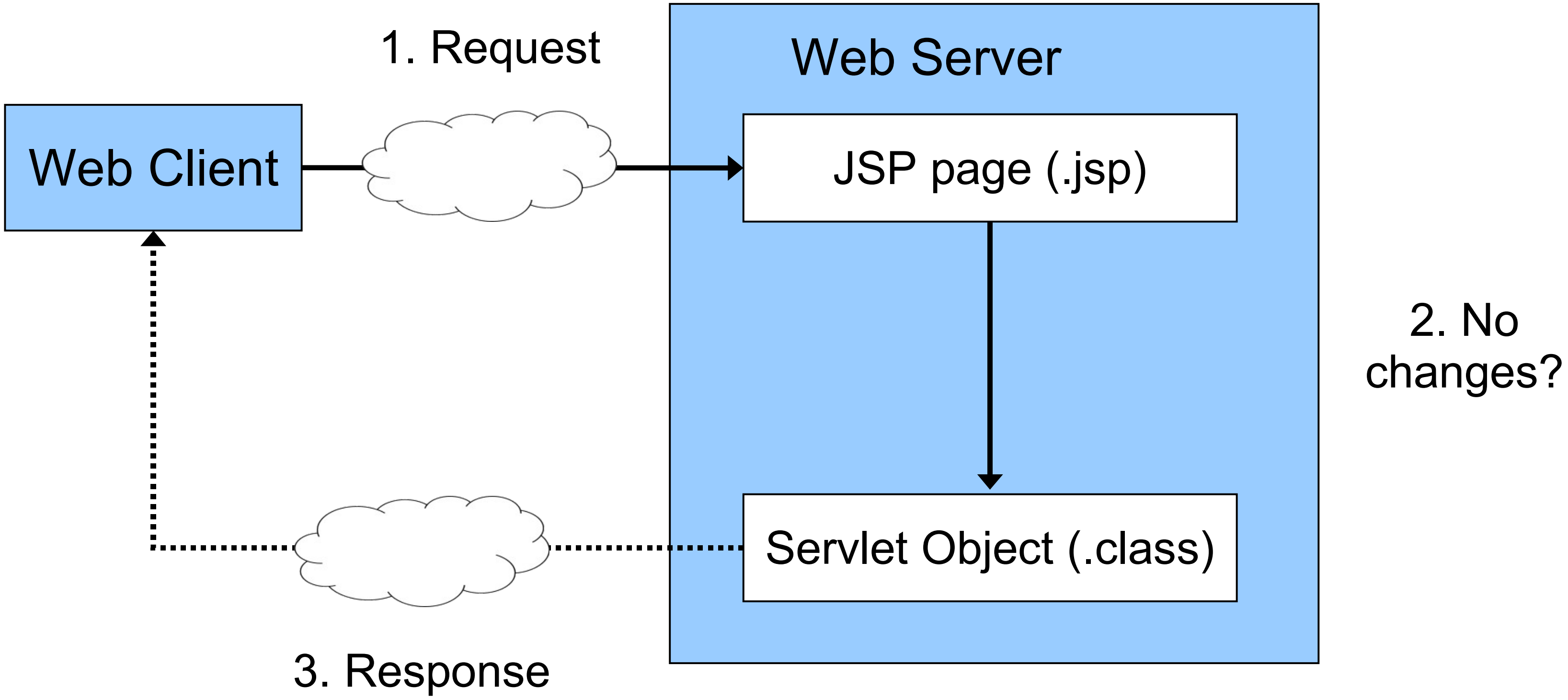
3 JSP – Basics

- How does it work?
  - first request



3 JSP – Basics

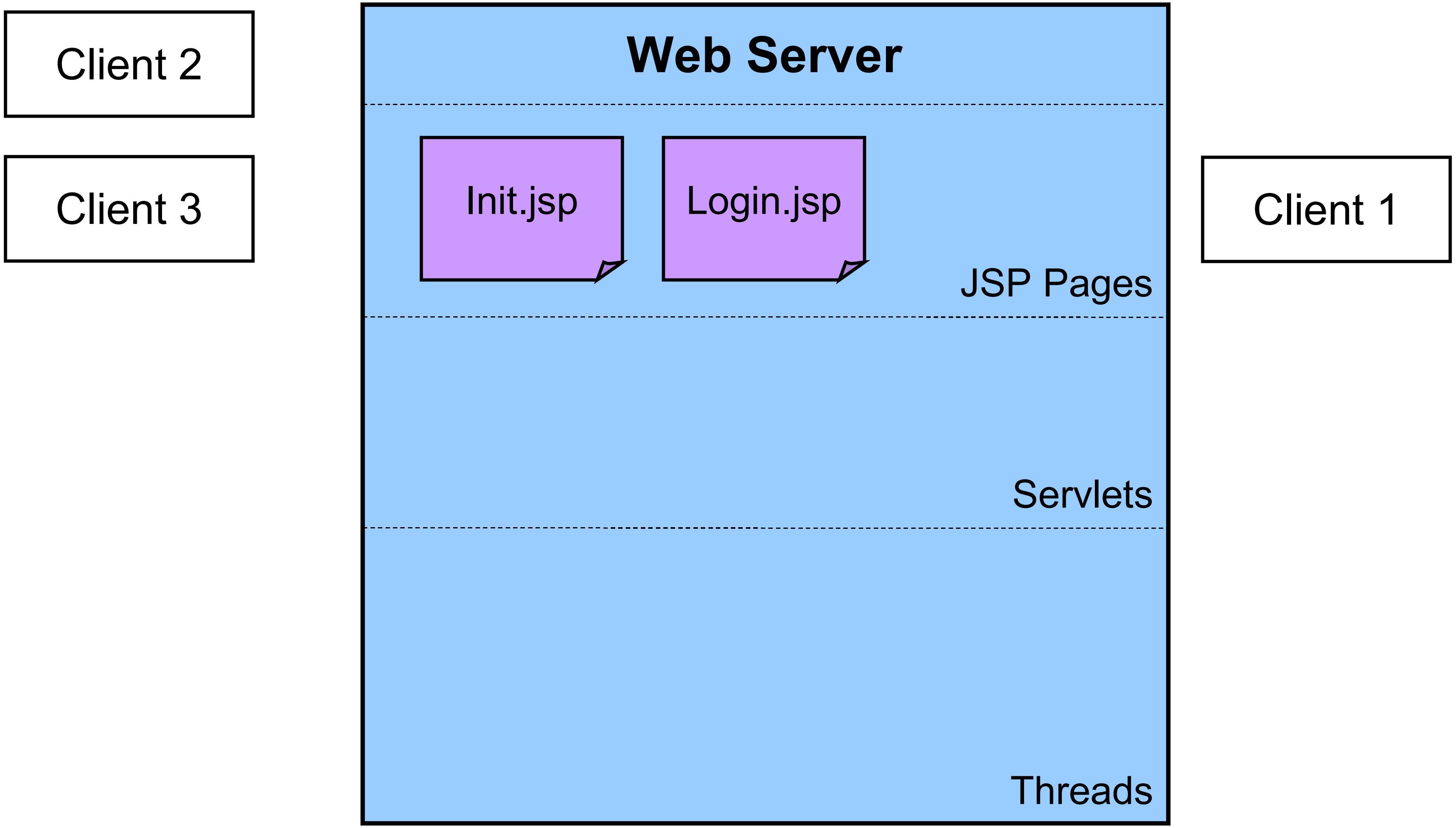
- How does it work?
  - subsequent requests



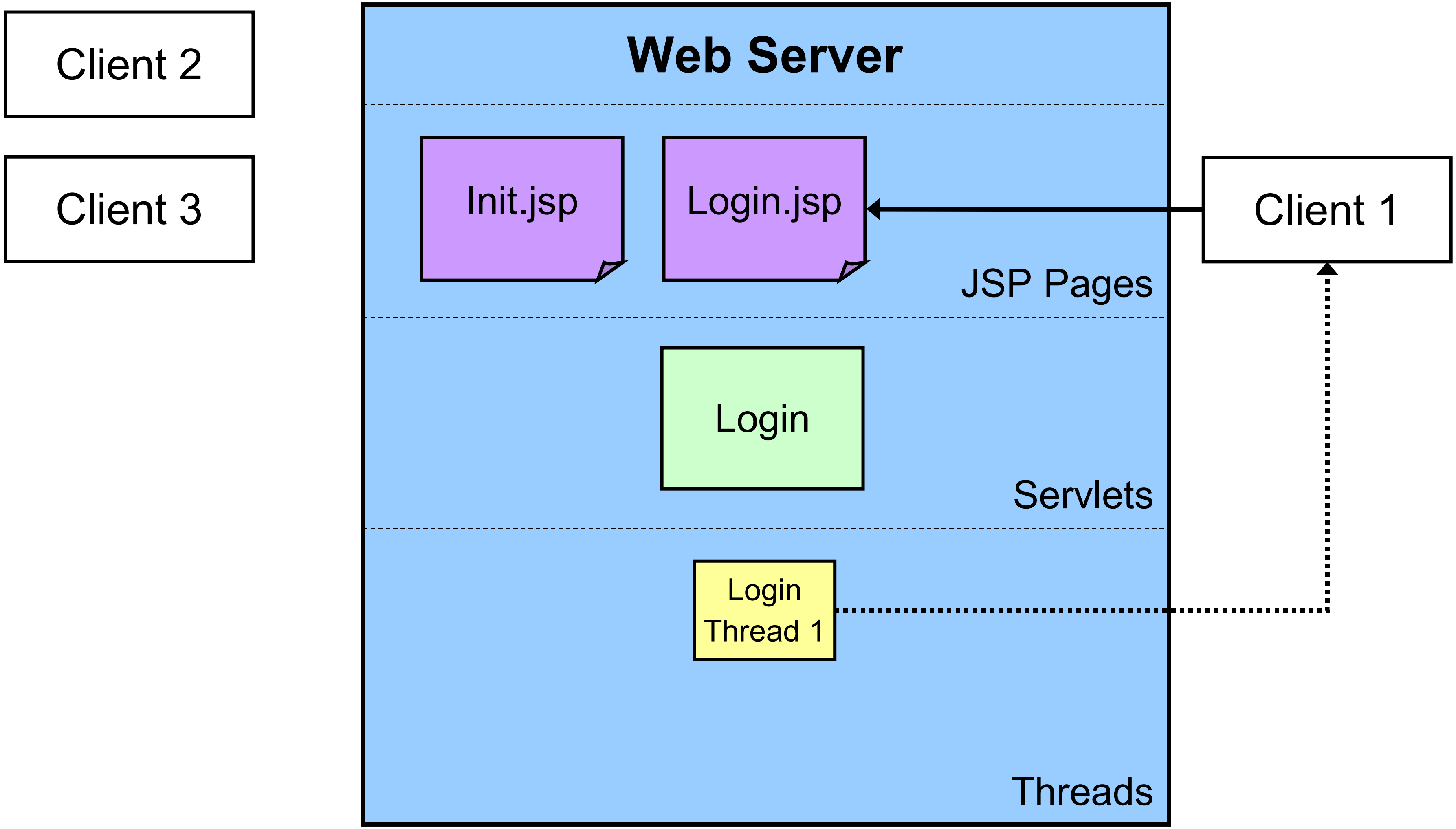
3 JSP – Basics

- JSPs are
  - a servlet that is mapped to the extension *.jsp
  - as powerful as Servlets
  - created from mixed HTML and Java code

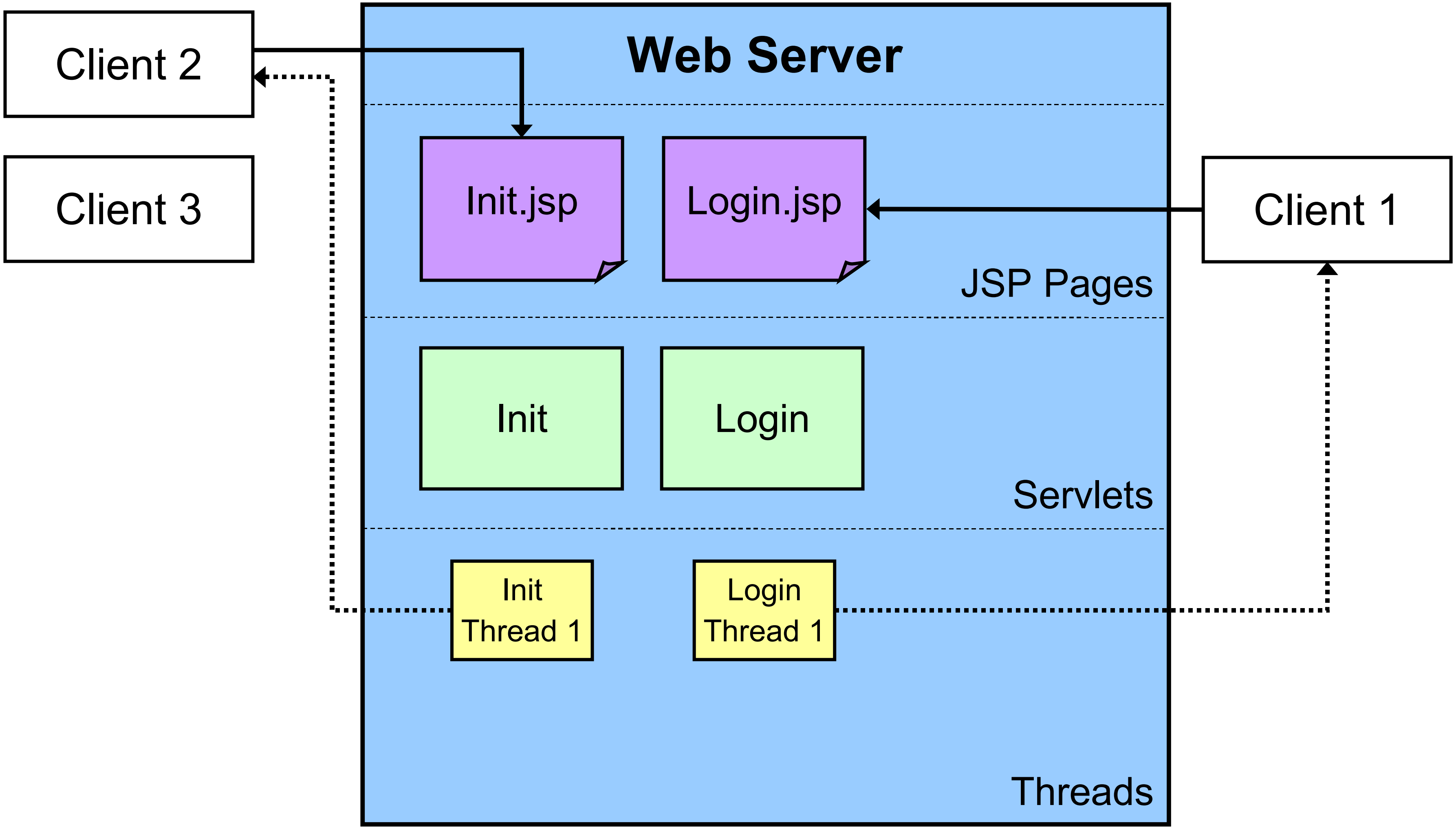
3 JSP – Performance



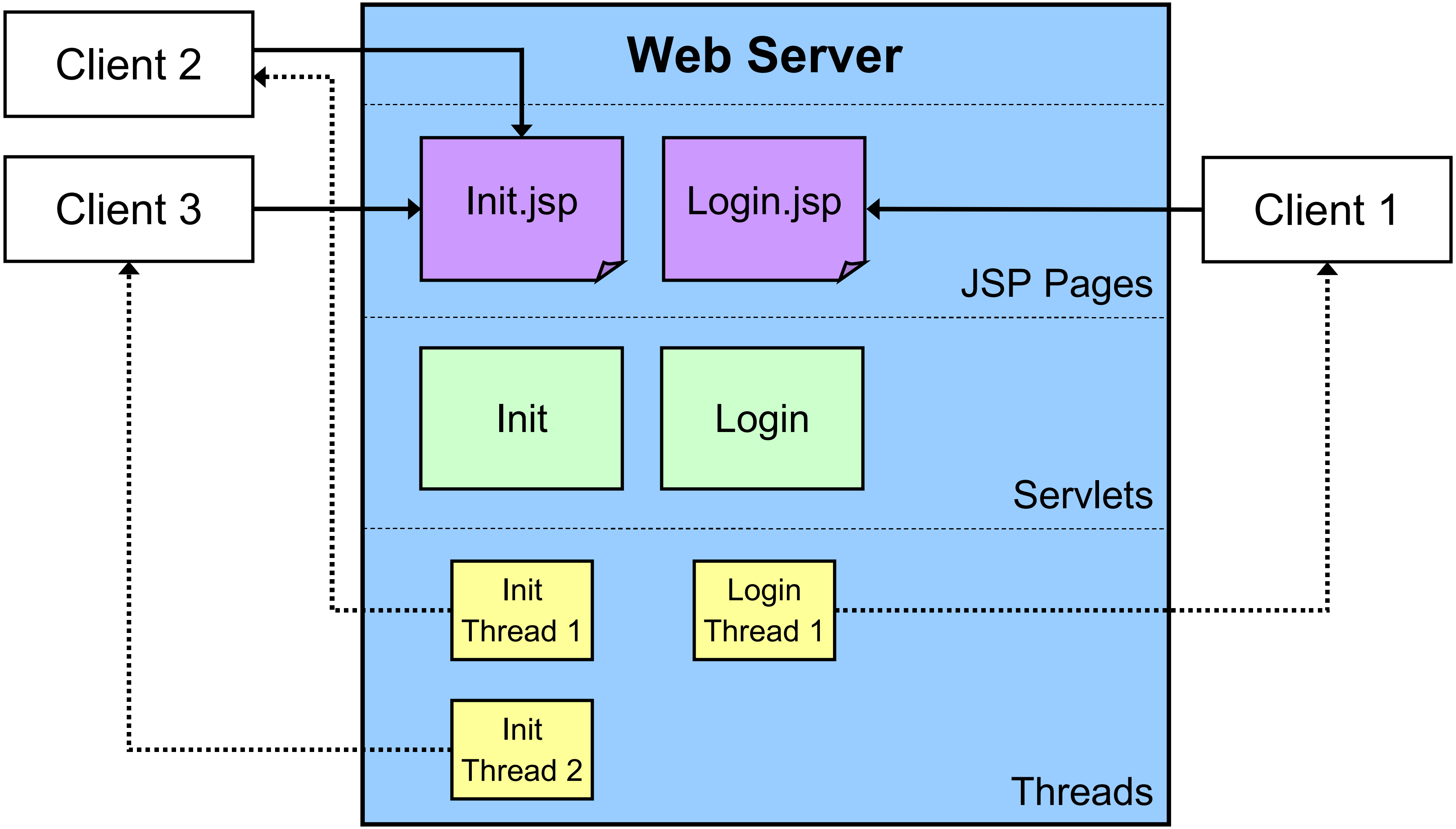
3 JSP – Performance



3 JSP – Performance



3 JSP – Performance



Servlets – Basics

4 Servlets – Basics

- Servlets
  - inherit `HttpServlet` (which inherits `GenericServlet`)
  - override `doGet()` and/or `doPost()`
  - are mapped to a URL
    - for example MyServlet.java → http://localhost/servlet/MyServlet

4 Servlets – Basics

- The Servlet Lifecycle
  - one time setup: `init()`
    - NOTE: only a single Servlet instance is created
  - on any user request: `service()`
    - the server creates a new thread and calls `service()`
    - ! don't overload `service()`
    - checks the HTTP request and calls `doGet()`, `doPost()`, …
  - to handle POST/GET requests
    - `doGet()`, `doPost()` ← most things happen here
  - destructor: `destroy()`

4 Servlets – First Examples

Example 1 – “Another Warm Welcome”

```
package WdwJ2ee;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ServletExample01 extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.println("Hello <em>pathetic earthling</em>.");
    out.close();
  }
  public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request,response);
  }
}
```

4 Servlets – First Examples

Example 1 – “Another Warm Welcome”

```
package WdwJ2ee;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ServletExample01 extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.println("Hello <em>pathetic earthling</em>.");
    out.close();
  }
  public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request,response);
  }
}
```

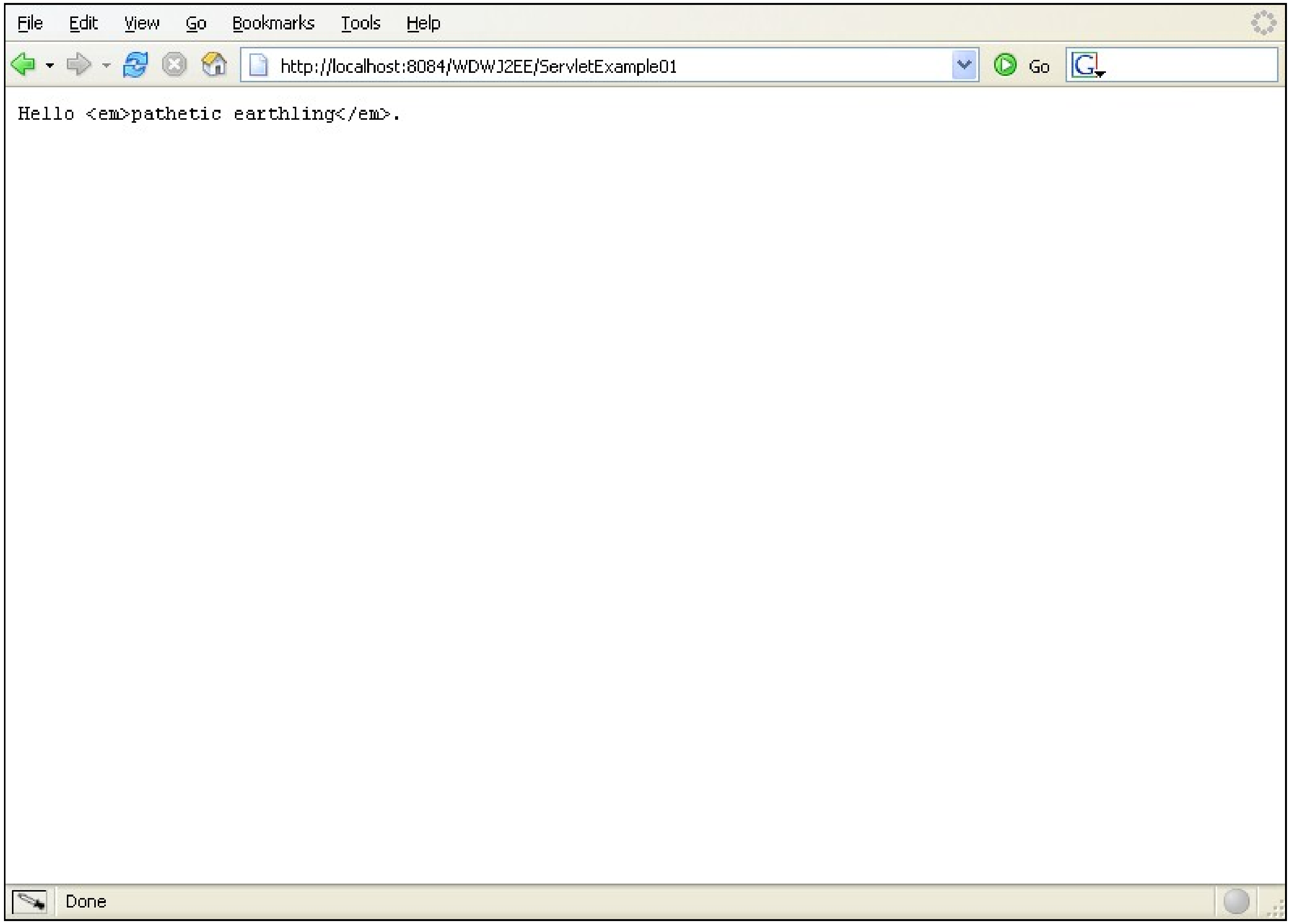
4 Servlets – First Examples

Example 1 – “Another Warm Welcome”

```
package WdwJ2ee;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ServletExample01 extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.println("Hello <em>pathetic earthling</em>.");
    out.close();
  }
  public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request,response);
  }
}
```

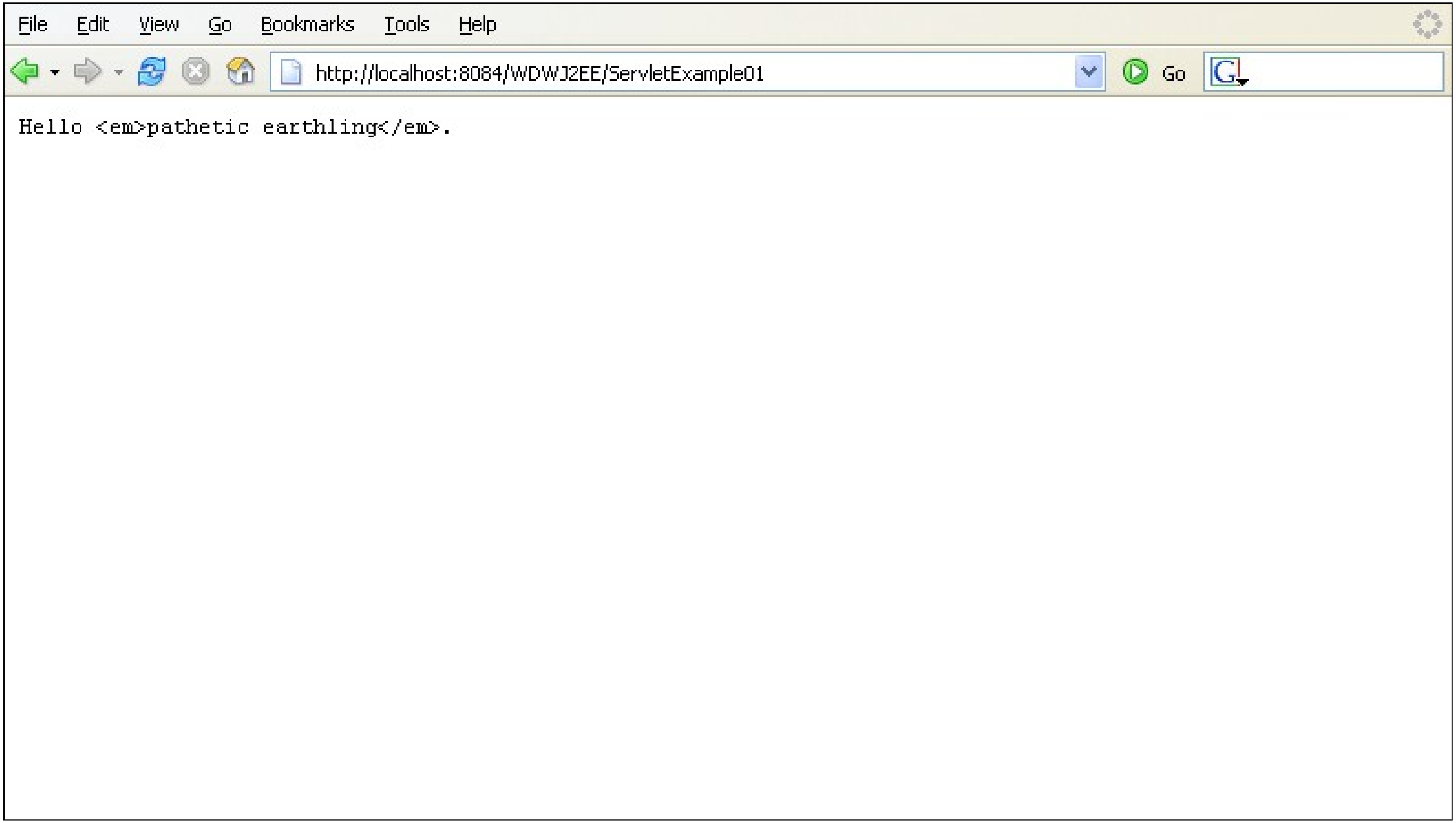
4 Servlets – First Examples

- Example 1 – “Another Warm Welcome”



```
Hello <em>pathetic earthling</em>.
```

4 Servlets – First Examples



```
Hello <em>pathetic earthling</em>.
```

4 Servlets – Basics

- doGet/doPost Arguments
  - Incoming data (**`HttpServletRequest`**)
    - form data
    - HTTP request headers
    - client’s hostname
  - Outgoing data (**`HttpServletResponse`**)
    - lets you obtain the **`PrintWriter`** to create the document content
    - response headers (Content-type, Set-Cookie, …)
    - HTTP status codes (200, 404, …)

4 Servlets – First Examples

Example 2 – How To Declare The Content Type

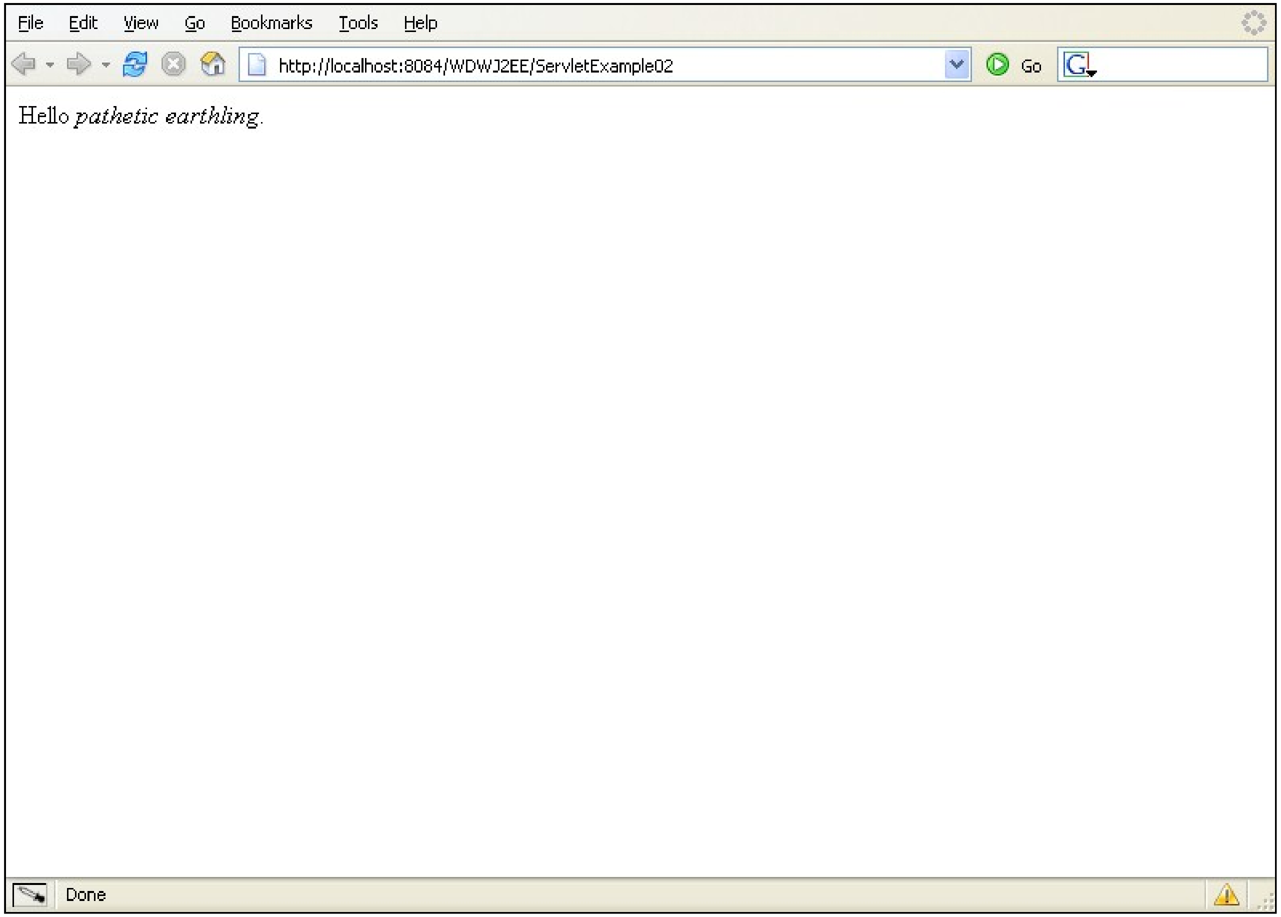
```
...
public class ServletExample02 extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("Hello <em>pathetic earthling</em>.");
    out.close();
  }
  public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request,response);
  }
}
```

! set content before using the PrintWriter (belongs to the response header)

! other content types: "text/xml", "application/vnd.msexcel", "image/gif", ...

4 Servlets – First Examples

- Example 2 – How To Declare The Content Type



Hello *pathetic earthling*.

Example 3 – More Decent HTML

```
package WdwJ2ee;
...
import javax.util.Calendar;
public class ServletExample03 extends HttpServlet {
  public void init(ServletConfig config)
      throws ServletException {
      cal = Calendar.getInstance();
      initTime = cal.get(Calendar.DAY_OF_MONTH) + "/" +
                  (cal.get(Calendar.MONTH)+1) + "/" +
                 cal.get(Calendar.YEAR) + "-" + cal.get(Calendar.MINUTE) + ":" +

                 cal.get(Calendar.SECOND) + ":" + cal.get(Calendar.MILLISECOND);
  }
  ...
  protected Calendar cal;
  protected String initTime;
}
```

Example 3 – More Decent HTML

```
...
public class ServletExample03 extends HttpServlet {
  private final String strContentType = "text/html;charset=UTF-8";
  public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    cal = Calendar.getInstance();
    String date = cal.get(Calendar.DAY_OF_MONTH) + "/" + ...
    String requestTime = cal.get(Calendar.DAY_OF_MONTH) + "/" + ...
    response.setContentType(strContentType);
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Servlet Example 3</title></head>");
    out.println("<body>");
    out.println("  <h1>Servlet Example 3</h1>");
    ...
```

<%!declaration%>

or <%=expression%>

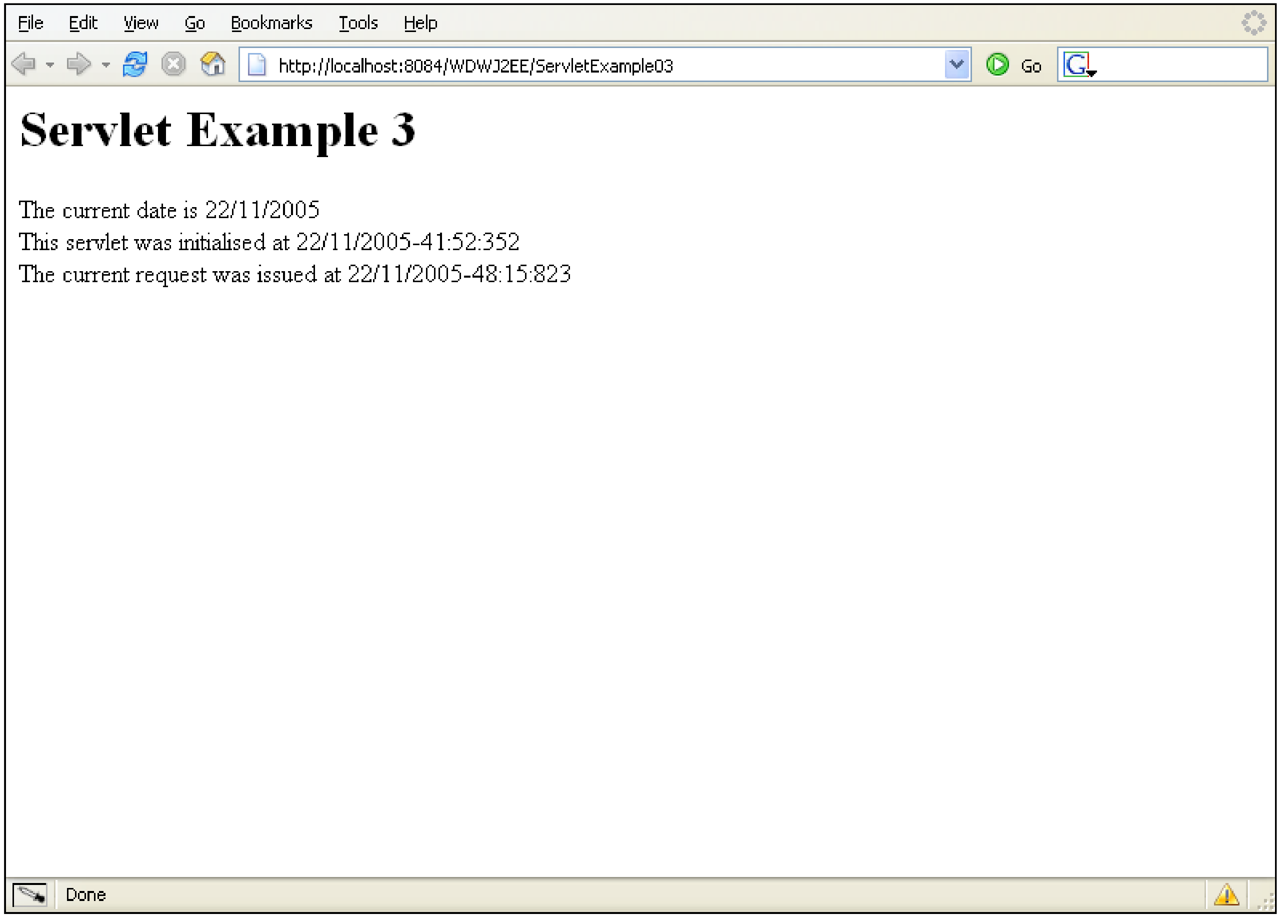
4 Servlets – First Examples

Example 3 – More Decent HTML

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
   throws ServletException, IOException {
   out.println("<html>");
   out.println("<head><title>Servlet Example 3</title></head>");
   out.println("<body>");
   out.println("  <h1>Servlet Example 3</h1>");
   out.println("<p>");
   out.println("  The current date is " + date + "<br/>");
   out.println("  This servlet was initialised at " + initTime + "<br/>");
   out.println("  The current request was issued at " + requestTime);
   out.println("</p>");
   out.println("</body></html>");
   out.close();
 }
```

4 Servlets – First Examples

Example 3 – More Decent HTML

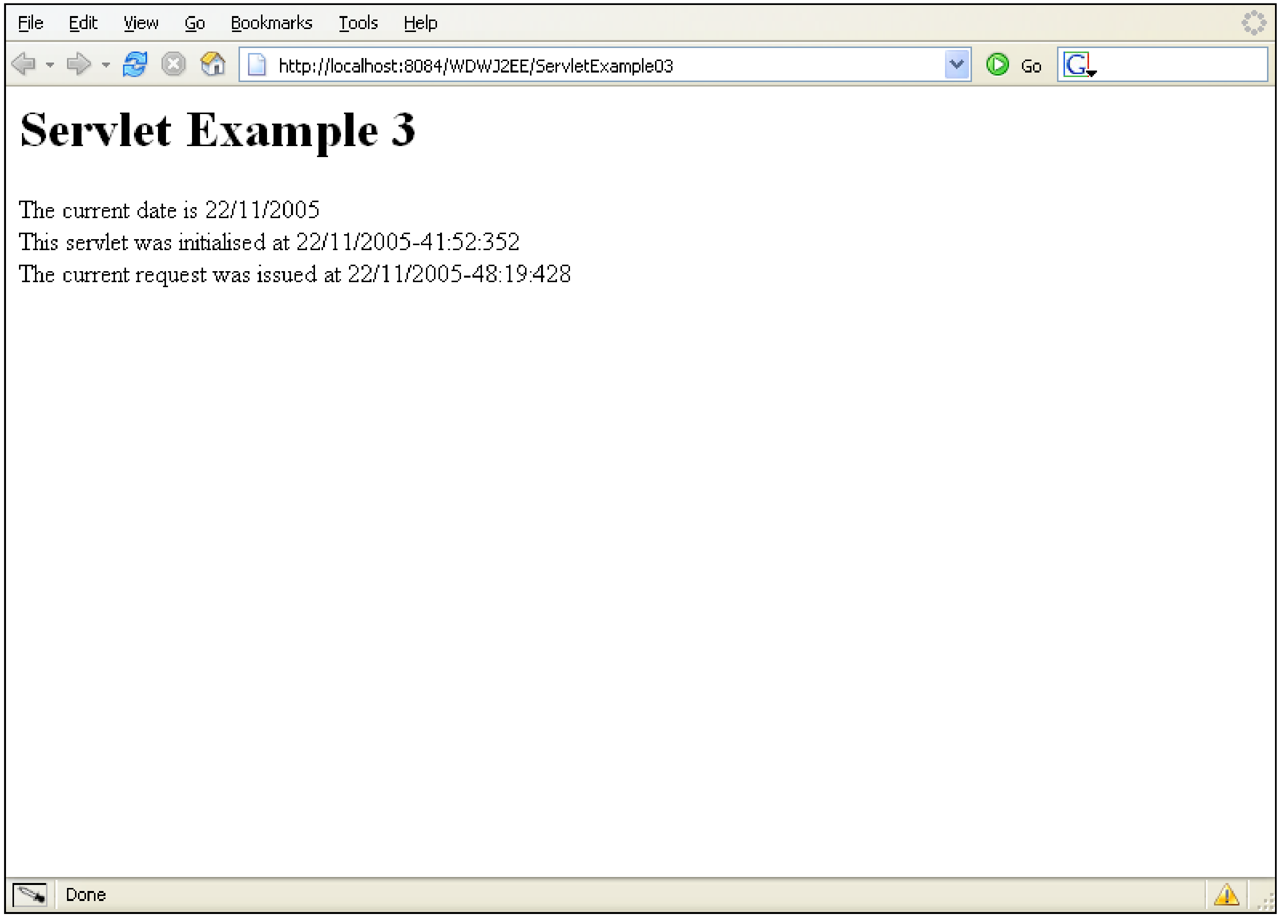


Servlet Example 3

```
The current date is 22/11/2005
This servlet was initialised at 22/11/2005-41:52:352
The current request was issued at 22/11/2005-48:15:823
```

4 Servlets – First Examples

Example 3 – More Decent HTML



Servlet Example 3

The current date is 22/11/2005
This servlet was initialised at 22/11/2005-41:52:352
The current request was issued at 22/11/2005-48:19:428

4 Servlets – JSPs are Servlets

JSPs are Servlets – Consequences (1)

- it’s easier to understand how JSP works

Where is the error here?

```
<%
  import java.util.Calendar;
  Calendar cal = Calendar.getInstance();
  String date = cal.get(Calendar.DAY_OF_MONTH) + "/" +
    (cal.get(Calendar.MONTH)+1) + " " + c.get(Calendar.YEAR);
%>
<html>
  <head><title>A Bad JSP Example</title></head>
<body>
 <p>The current date is <%=date%></p>
</body></html>
```

4 Servlets – JSPs are Servlets

- JSPs are Servlets – Consequences (1)
  - it’s easier to understand how JSP works

Where is the error here?

```
<%
  ~~import java.util.Calendar;~~
  Calendar cal = Calendar.getInstance();
  String date = cal.get(Calendar.DAY_OF_MONTH) + "/" +
    (cal.get(Calendar.MONTH)+1) + " " + c.get(Calendar.YEAR);
%>
<html>
  <head><title>A Bad JSP Example</title></head>
<body>
 <p>The current date is <%=date%></p>
</body></html>
```

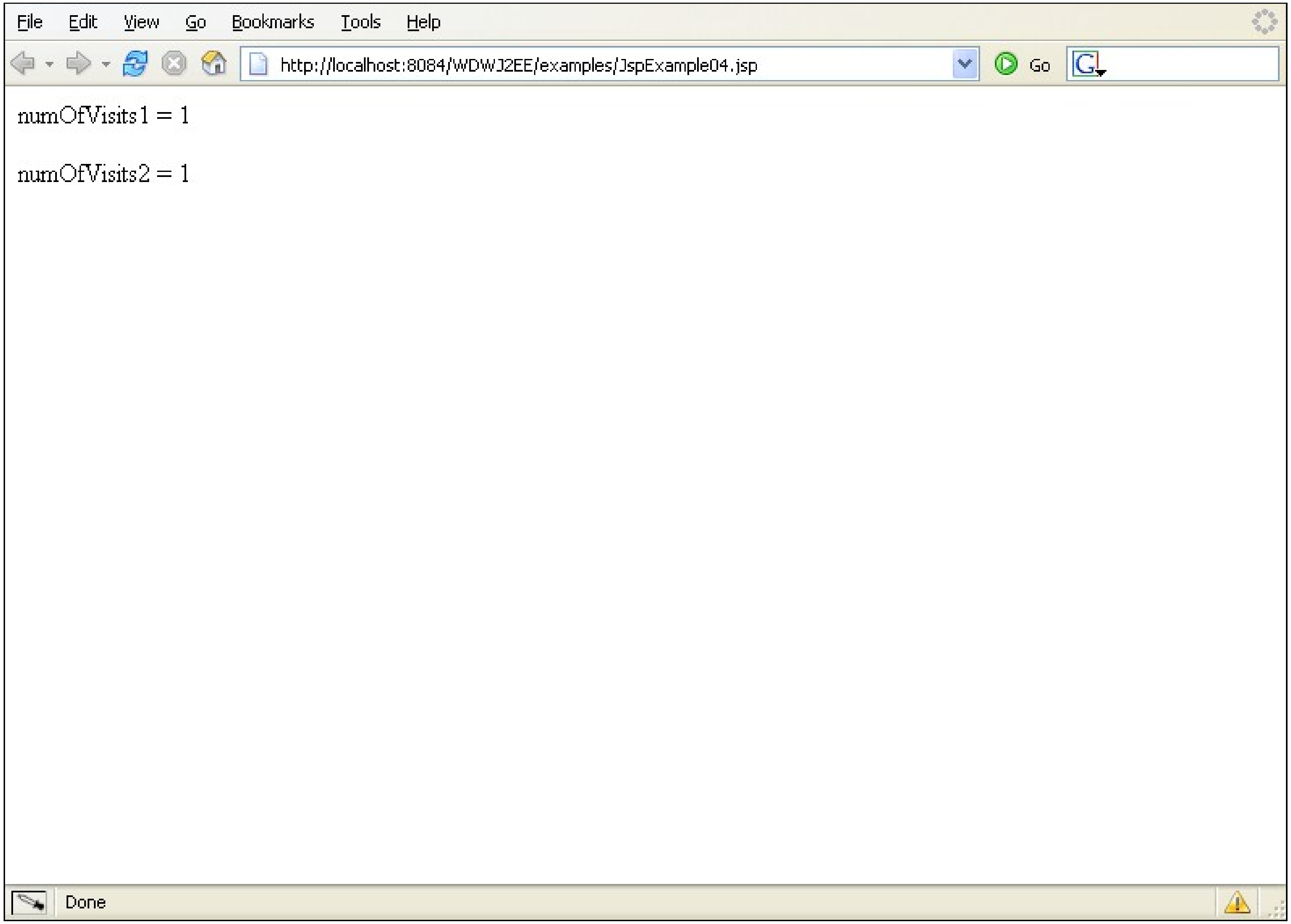
4 Servlets – JSPs are Servlets

JSPs are Servlets – Consequences (2)

- different ways of declaring variables

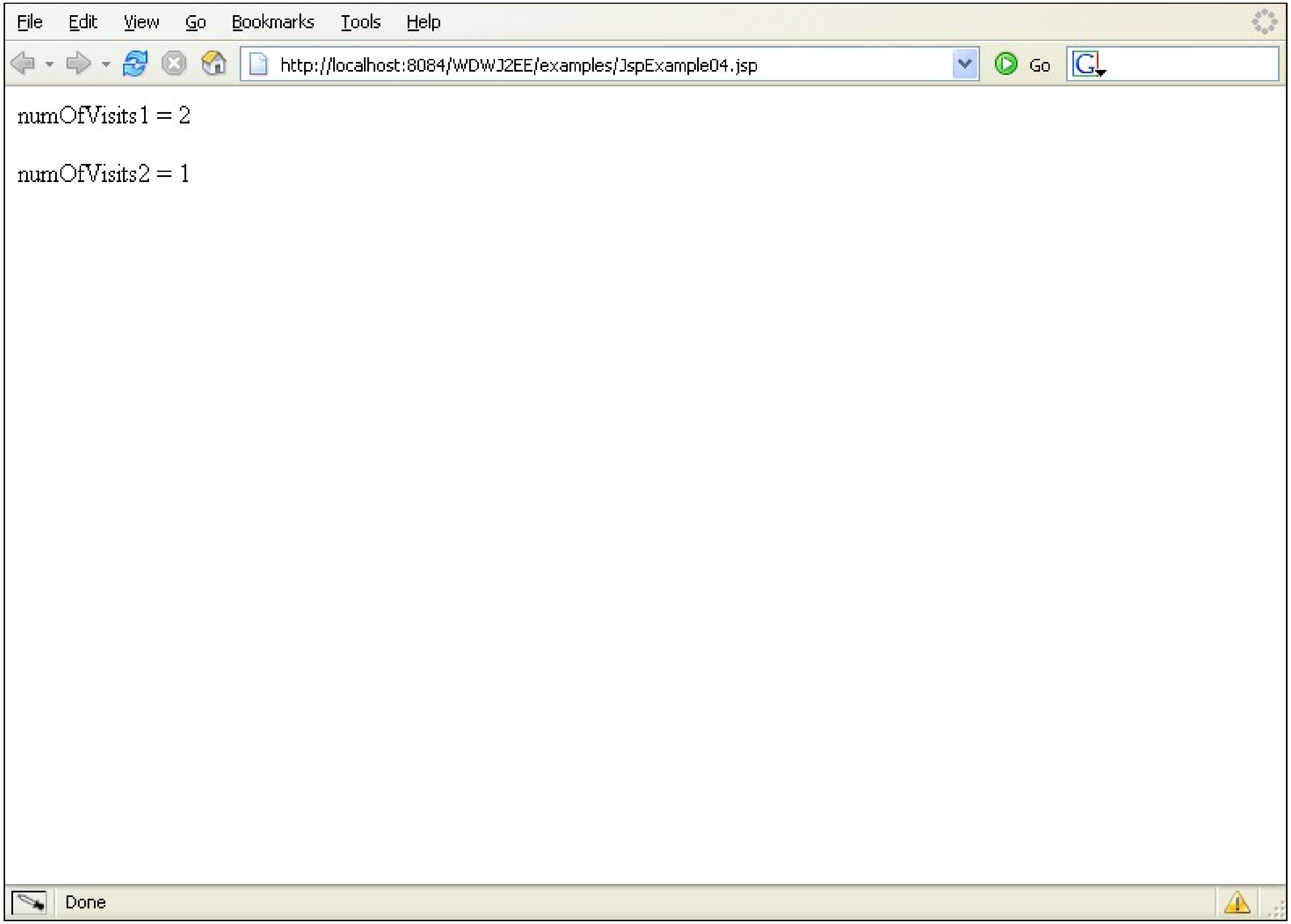
```
<%! int numOfVisits1 = 0; %>
<%  int numOfVisits2 = 0; %>
<html>
  <head><title>
    JSP Example 04 - Different Ways of Declaring Variables
  </title></head>
<body>
 <p>numOfVisits1 = <%=++numOfVisits1%></p>
 <p>numOfVisits2 = <%=++numOfVisits2%></p>
</body></html>
```

4 Servlets – JSPs are Servlets



```
numOfVisits1 = 1

numOfVisits2 = 1
```

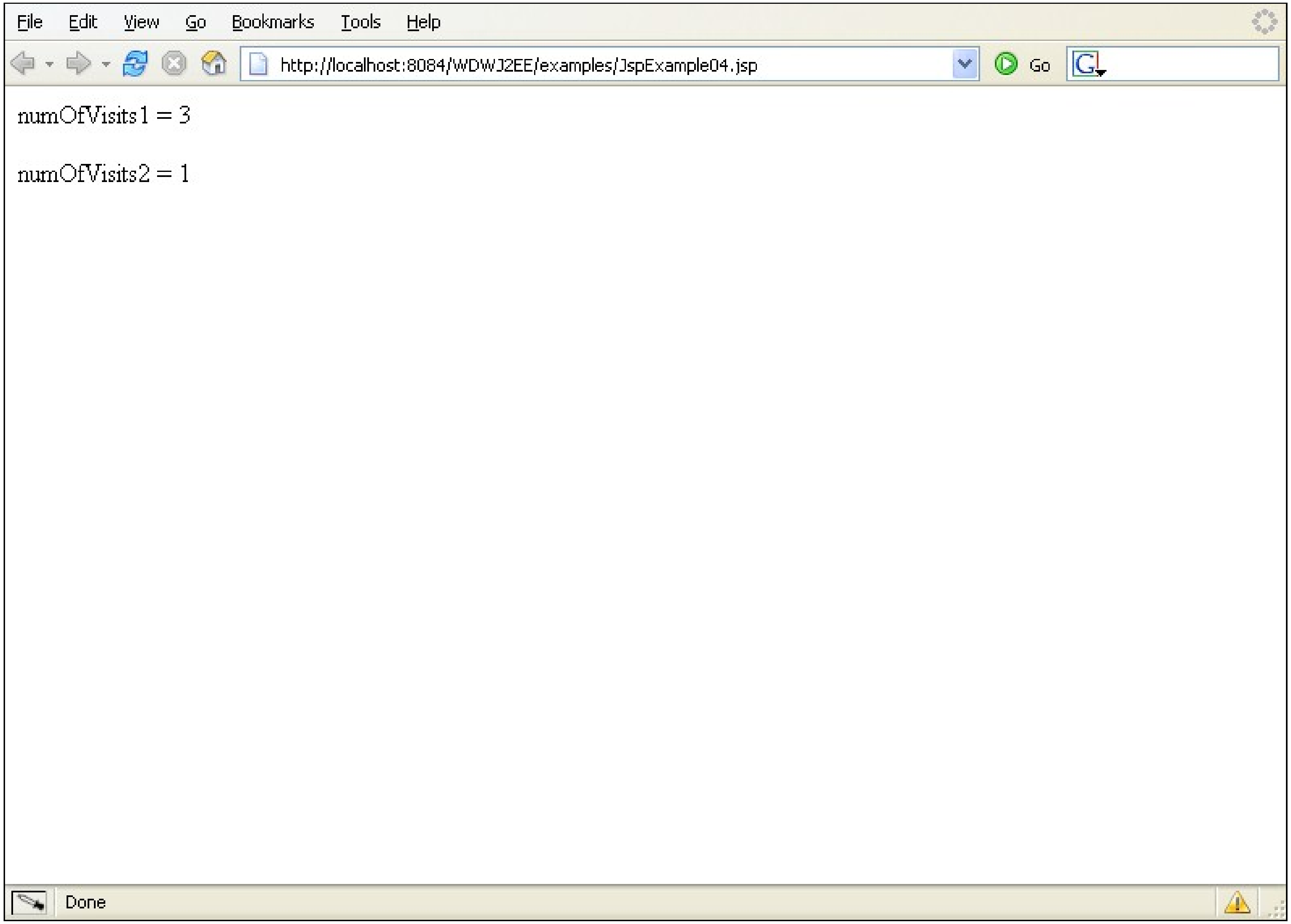
4 Servlets – JSPs are Servlets



File Edit View Go Bookmarks Tools Help
http://localhost:8084/WDWJ2EE/examples/JspExample04.jsp
Go
Done

```
numOfVisits1 = 2

numOfVisits2 = 1
```

4 Servlets – JSPs are Servlets



```
numOfVisits1 = 3

numOfVisits2 = 1
```

4 Servlets – JSPs are Servlets

JSPs are Servlets – Consequences (2)

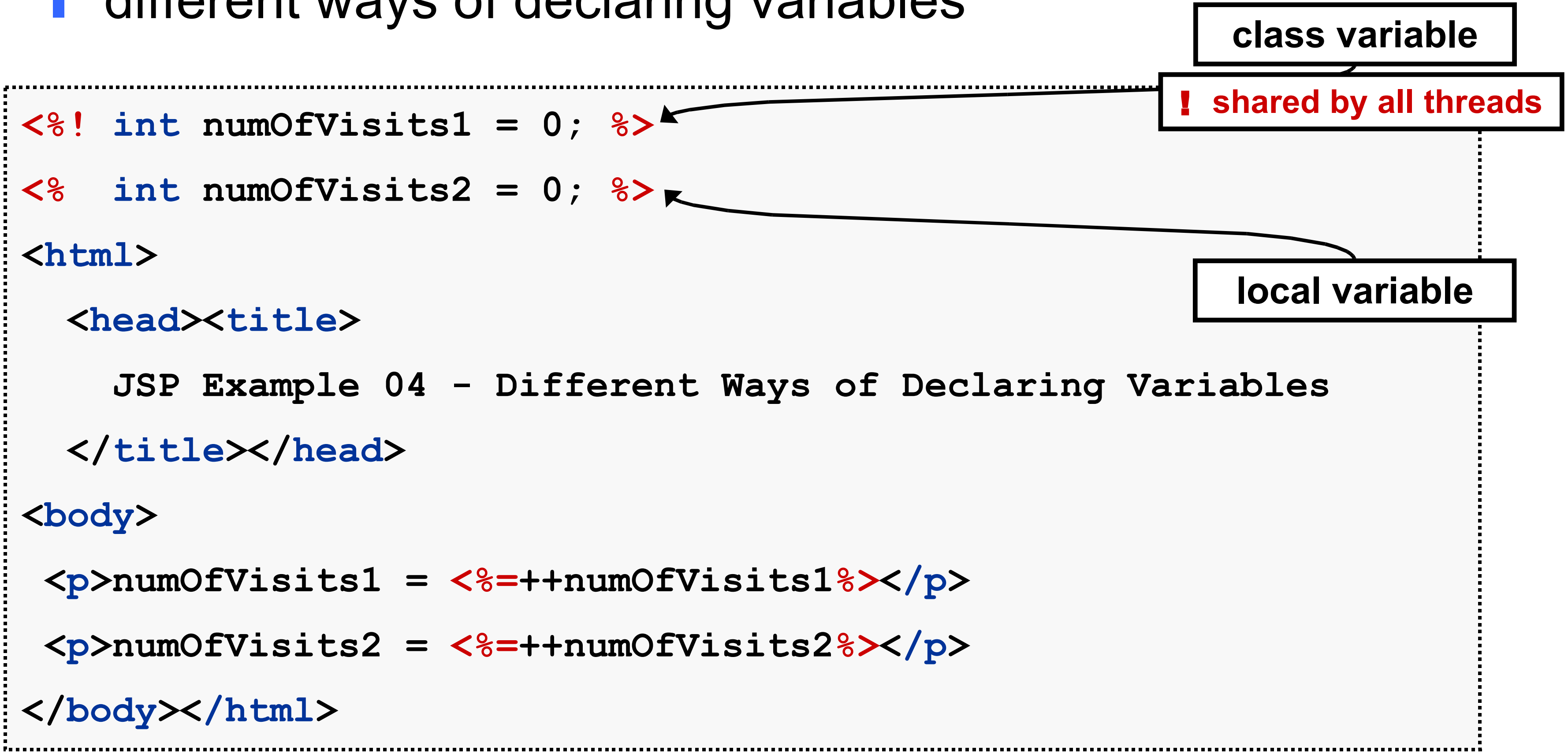
- JspExample04.jsp → JspExample04_jsp.java

```
...
public final class JspExample04_jsp extends org.apache.jasper.runtime.HttpJspBase
  implements org.apache.jasper.runtime.JspSourceDependent {
  int numOfVisits1 = 0;
  ...
  public void _jspService(HttpServletRequest request, HttpServletResponse response)
    throws java.io.IOException, ServletException {
    ...
    _jspxFactory = JspFactory.getDefaultFactory();
    response.setContentType("text/html");
    ...
    int numOfVisits2 = 0;
    ...
  }
}
```

4 Servlets – JSPs are Servlets

JSPs are Servlets – Consequences (2)

- different ways of declaring variables



```
<%! int numOfVisits1 = 0; %>
<%  int numOfVisits2 = 0; %>
<html>
  <head><title>
    JSP Example 04 - Different Ways of Declaring Variables
  </title></head>
<body>
 <p>numOfVisits1 = <%=++numOfVisits1%></p>
 <p>numOfVisits2 = <%=++numOfVisits2%></p>
</body></html>
```

class variable

! shared by all threads

local variable

JSP – Details

5 JSP – Comments

- A Comment on Comments
  - JSP comments do not appear in the output …
    - `<%-- A JSP comment --%>`
  - … while HTML comments do
    - `<!-- An HTML comment -->`

5 JSP – Thread Safety

- Thread Safety
  - `<%@ page isThreadSafe="false" %>`
  - declares that *the servlet code* is not thread safe
  - must implement `SingleThreadModel`

```
<%@ page isThreadSafe="false" %>
...
<%! private int nUserId = 0; %>
<%
  String strUserId = "uid." + nUserId;
  out.println("Your ID is " + strUserId + ".");
  nUserId += 1;
%>
```

! can be slow

→ instance variables not necessarily unique because servlet engines are permitted (not required) to create multiple servlet instances

5 JSP – Thread Safety

Thread Safety

- `<%@ page isThreadSafe="true" %>` (Default)
- declares that *the servlet code* is thread safe

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

! it’s up to the programmer to make the code thread safe

→ two threads cannot enter the block at the same time

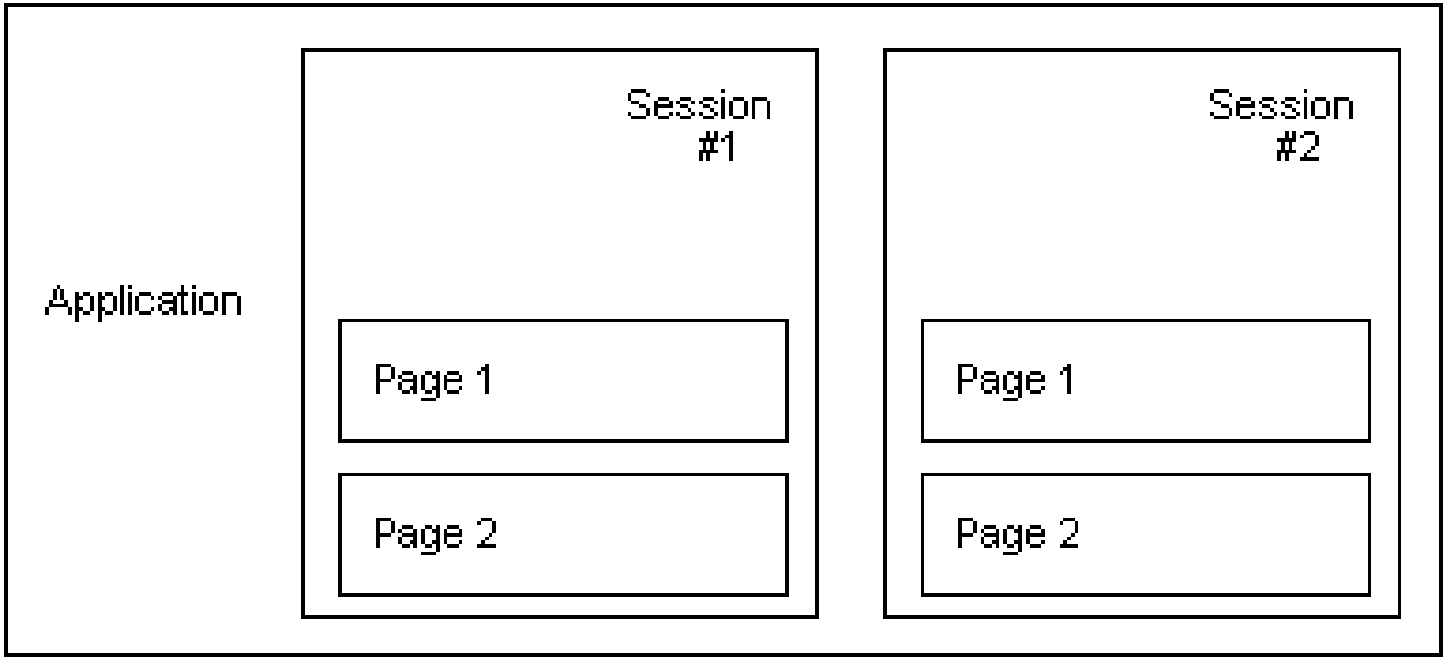
5 JSP – Predefined Variables

- Predefined Variables (Implicit Objects)
  - to simplify writing JSPs
  - available in `_jspService()` (not in declarations!)

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

    JspFactory _jspxFactory = JspFactory.getDefaultFactory();
    PageContext pageContext = _jspxFactory.getPageContext(...);
    HttpSession session = pageContext.getSession();
    ServletContext application = pageContext.getServletContext();
    ServletConfig config = pageContext.getServletConfig();
    JspWriter out = pageContext.getOut();
    Object page = this;
    ...
```

5 JSP – Scope

- Information in a web application can be stored in
  - **application** scope, **session** scope, **request** scope and **page** scope



5 JSP – Scope

- Information in a web application can be stored in
  - **page** scope: each JSP has it's own page scope, so objects in the page scope of one JSP are not accessible to other JSPs
  - **request** scope: all JSPs and servlets that share a request share the request scope (e.g., forwarded, included pages, *not* redirect)
  - **session** scope: all requests from the same user are in the same session, each user has his own session
  - **application** scope: all users share the same values of the application scope, therefore all users can see changes made by other users

5 JSP – Scope

- Information in a web application can be stored in
  - **page** scope: accessible through `pageContext` object
    - released when the response is returned or the request forwarded
  - **request** scope: accessible through `request` object
    - released when the request has been processed
  - **session** scope: accessible through `session` object
    - released when the session ends
  - **application** scope: accessible through `application` object
    - released when the runtime environment reclaims the ServletContext

5 JSP – Scope

- JspScopeExample.jsp

```
<%   if (pageContext.getAttribute("pageCount")==null) {
        pageContext.setAttribute("pageCount", new Integer(0));
     }
     if (session.getAttribute("sessionCount")==null) {
        session.setAttribute("sessionCount",new Integer(0));
     }
     if (application.getAttribute("appCount")==null) {
        application.setAttribute("appCount",new Integer(0));
     }
%>
...
```

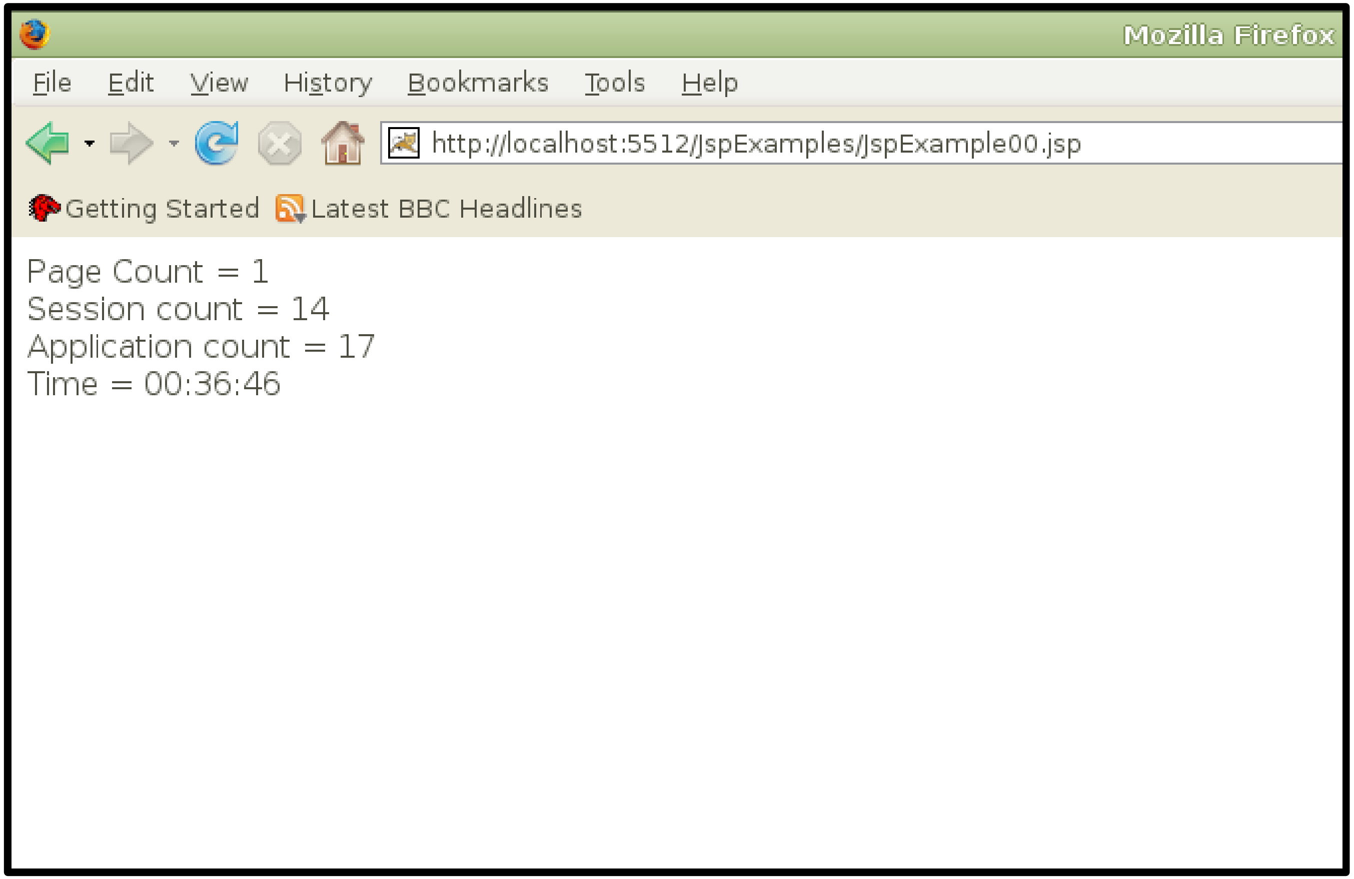
5 JSP – Scope

- JspScopeExample.jsp

```
...

<HTML>
  <BODY>
     <% Integer count = (Integer)pageContext.getAttribute("pageCount");
        pageContext.setAttribute("pageCount", new Integer(count.intValue()+1));
        Integer count2 = (Integer)session.getAttribute("sessionCount");
        session.setAttribute("sessionCount",new Integer(count2.intValue()+1));
        Integer count3 = (Integer)application.getAttribute("appCount");
        application.setAttribute("appCount",new Integer(count3.intValue()+1));
     %>
     Page Count = <%= pageContext.getAttribute("pageCount")%>
     <br/>Session count = <%= session.getAttribute("sessionCount")%>
     <br/>Application count = <%= application.getAttribute("appCount")%>
     <br/>Time = <%=  new java.sql.Time(System.currentTimeMillis()) %>
   </BODY>
</HTML>
```

5 JSP – Scope

- JspScopeExample.jsp

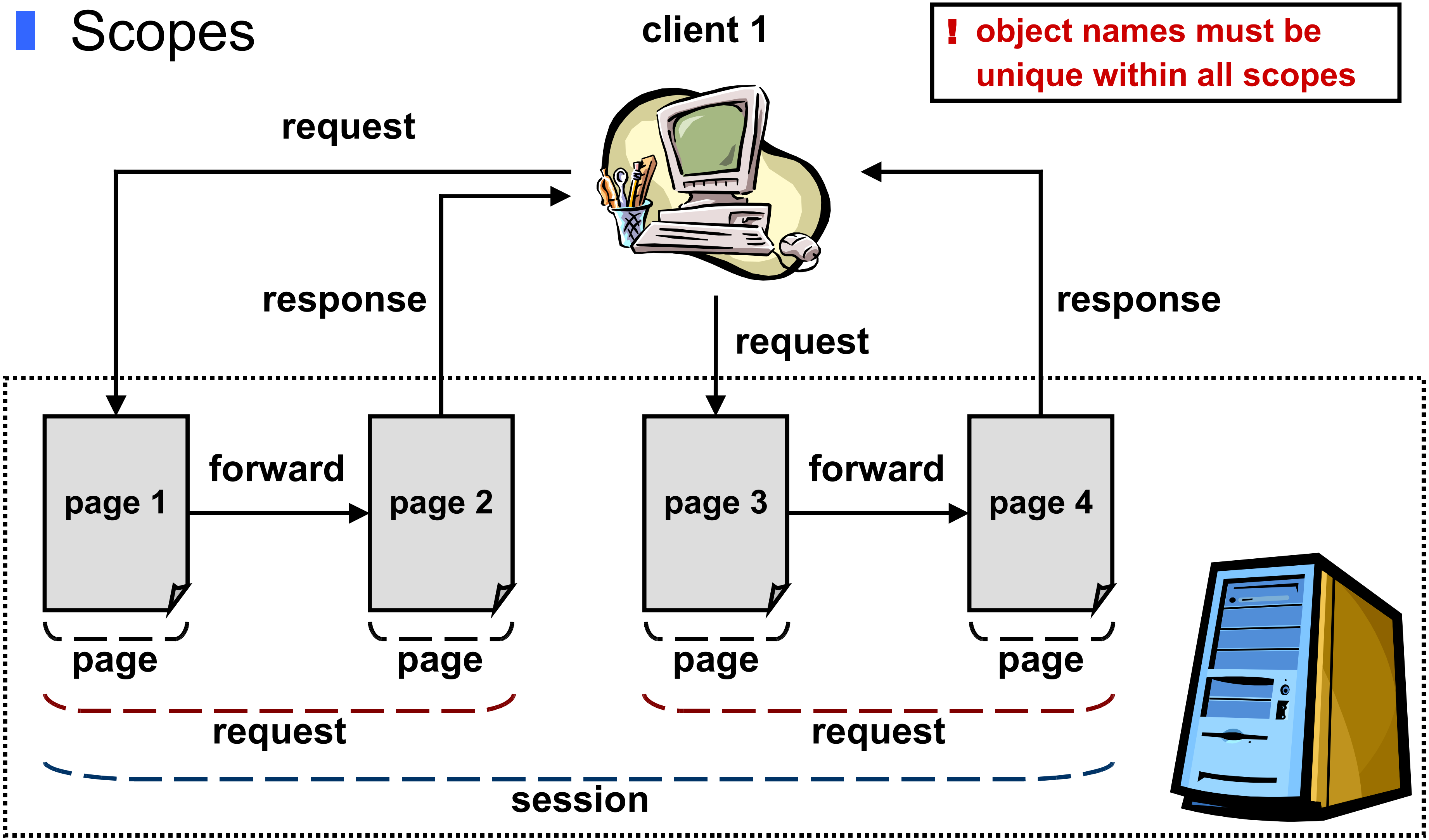

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:5512/JspExamples/JspExample00.jsp
Getting Started
Latest BBC Headlines

```
Page Count = 1
Session count = 14
Application count = 17
Time = 00:36:46
```

5 JSP – Scopes

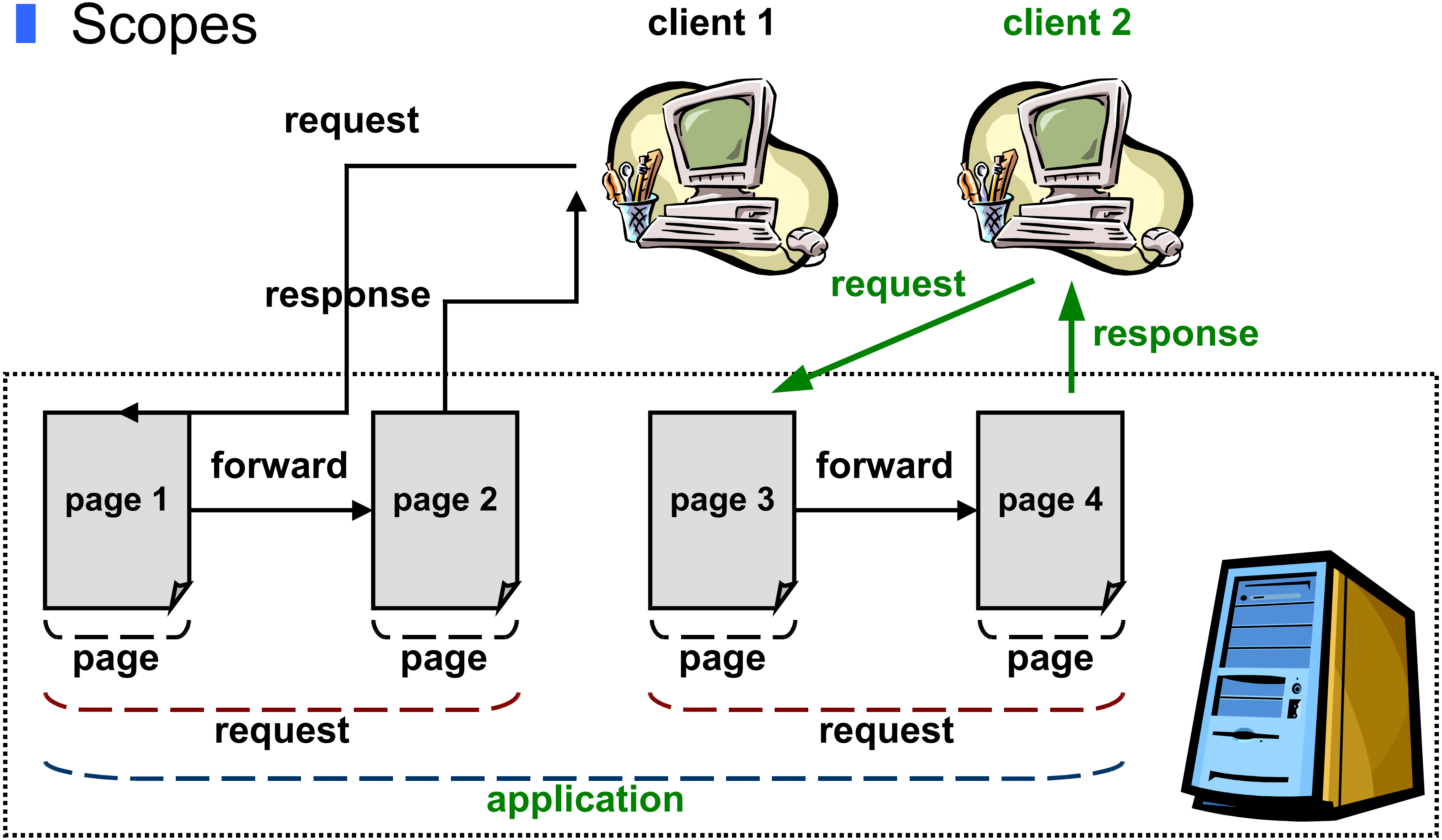
- Scopes

! object names must be unique within all scopes



5 JSP – Scopes

- Scopes



5 JSP – Predefined Variables – request

- Predefined Variables – request
  - request ↔ **`HttpServletRequest`**
  - provides access to the incoming HTTP header
    - `public String getParameter("ParName")`
    - `public String[] getParameterValues("ParName")`
    - `public String getMethod()`
    - `public Cookies[] getCookies()`
    - `public Object getAttribute("AttName")`
    - `public Enumeration getAttributeNames()`
    - `public void setAttribute("AttName", obj)`
    - `public String getQueryString()`
    - `public String getRemoteHost()` …

objects
in request
scope

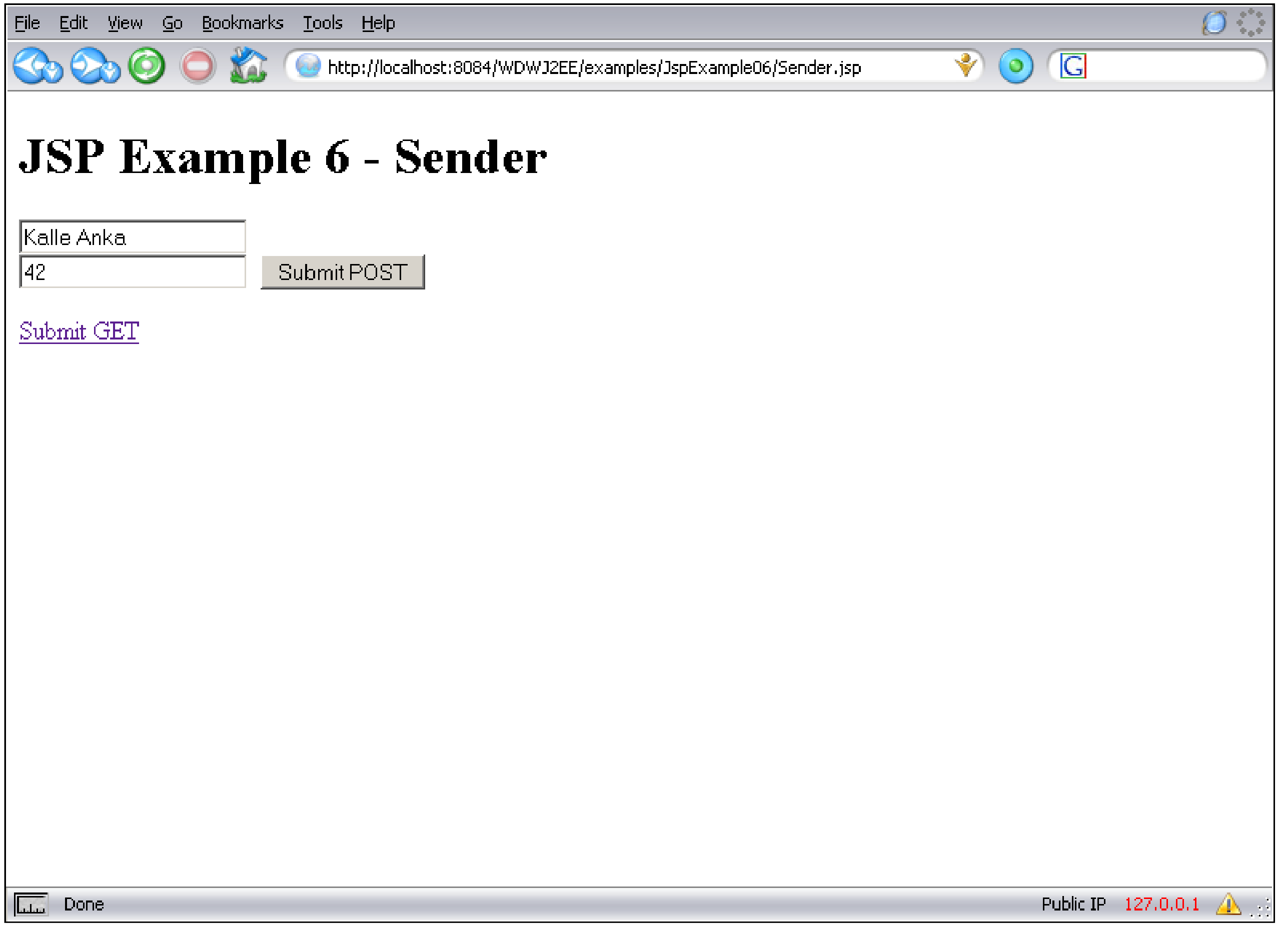
5 JSP – Predefined Variables – request

- POST / GET
  - specify with the attribute `method`

```
<html>
  <head><title>Sender</title></head>
<body>
  <form action="Receiver.jsp" method="post">
    <input type="text" name="name" value="Kalle Anka" /><br>
    <input type="text" name="age" value="42" />&nbsp;
    <input type="submit" name="Submit" value="Submit POST" />
  </form>
  <a href="Receiver.jsp?name=Kalle+Anka&age=42">Submit GET</a>
</body>
</html>
```

5 JSP – Predefined Variables – request

- POST / GET



File Edit View Go Bookmarks Tools Help
http://localhost:8084/WDWJ2EE/examples/JspExample06/Sender.jsp
Done
Public IP 127.0.0.1

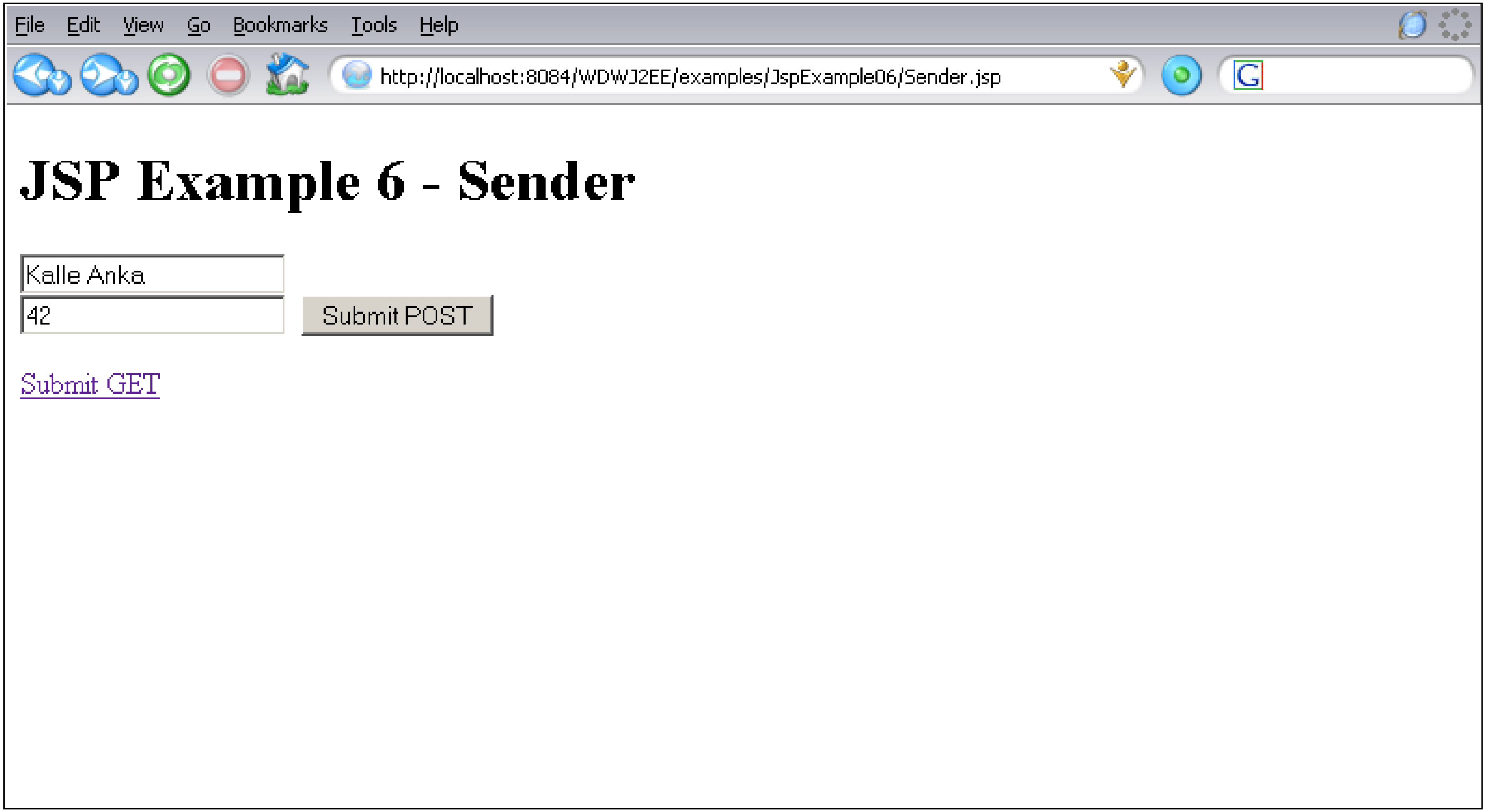
JSP Example 6 - Sender

Kalle Anka

42 Submit POST

Submit GET

5 JSP – Predefined Variables – request


File Edit View Go Bookmarks Tools Help
http://localhost:8084/WDWJ2EE/examples/JspExample06/Sender.jsp

JSP Example 6 - Sender

Kalle Anka

42

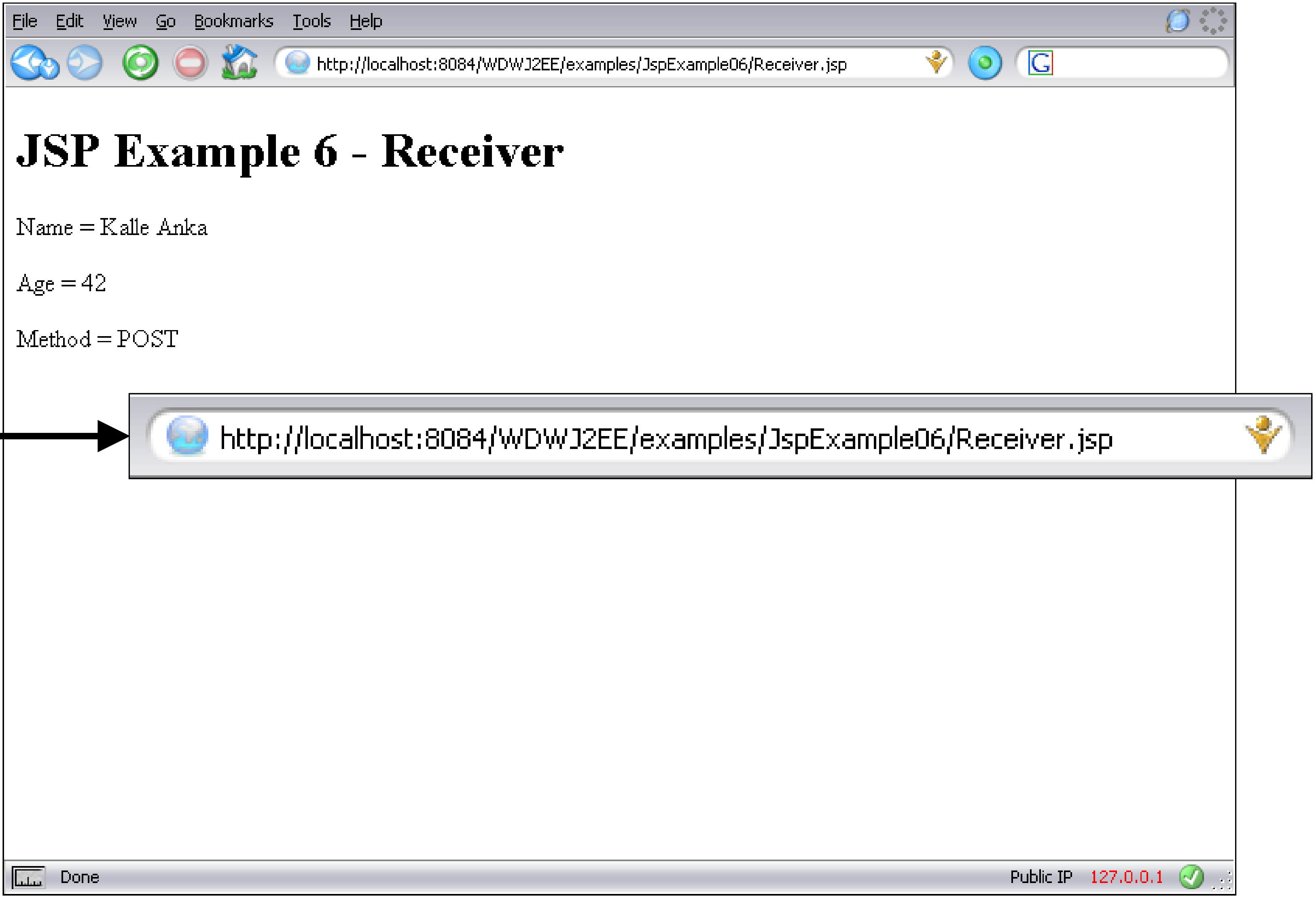
5 JSP – Predefined Variables – request

- POST / GET
  - retrieved from the **`request`** in the same way

```
<html>
  <head><title>Receiver</title></head>
<body>
  <p>Name = <%=request.getParameter("name")%></p>
  <p>Age = <%=request.getParameter("age")%> </p>
  <p>Method = <%=request.getMethod()%> </p>
</body>
</html>
```

5 JSP – Predefined Variables – request

POST



File
Edit
View
Go
Bookmarks
Tools
Help
http://localhost:8084/WDWJ2EE/examples/JspExample06/Receiver.jsp
Done
Public IP
127.0.0.1

JSP Example 6 - Receiver

Name = Kalle Anka

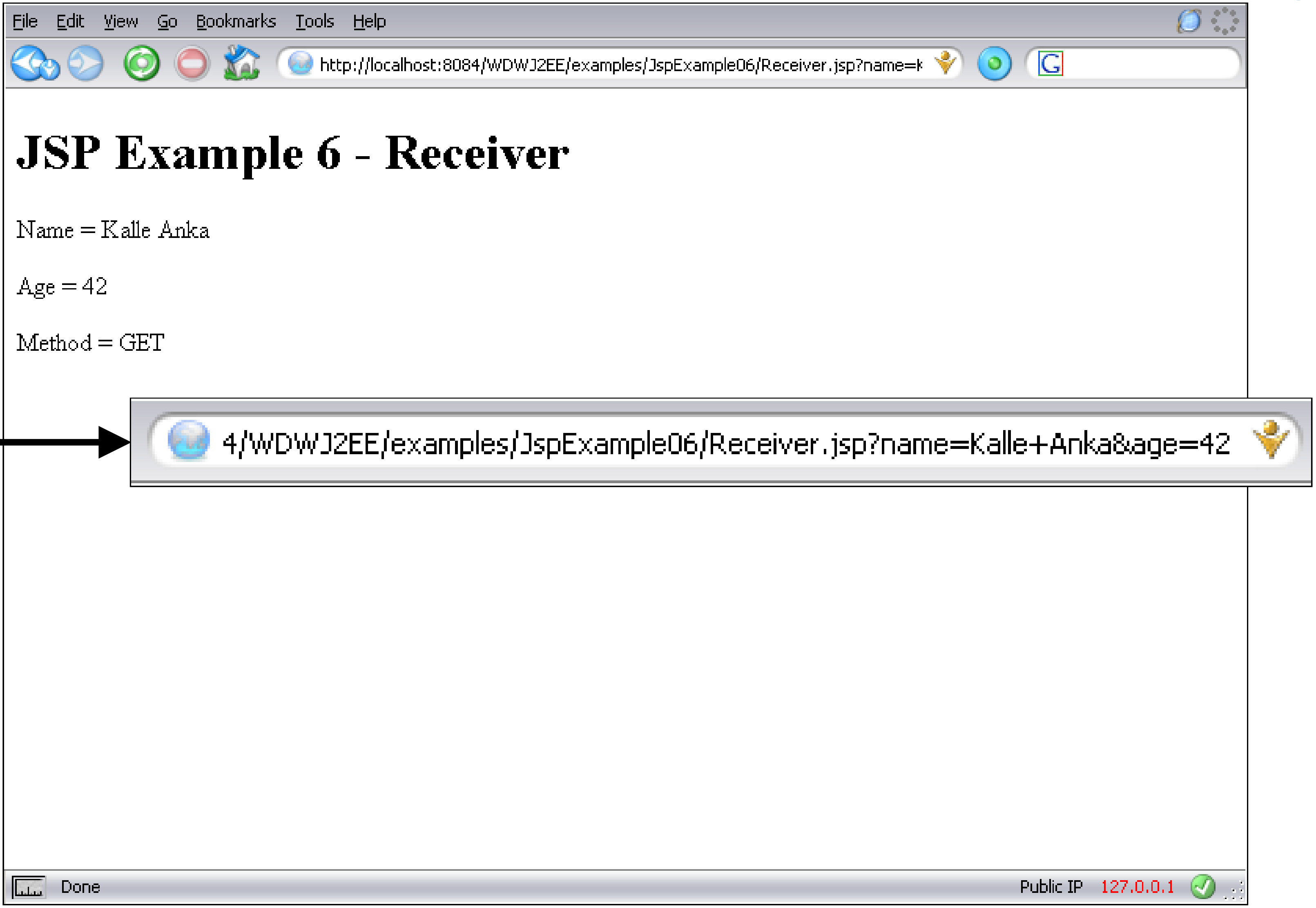
Age = 42

Method = POST

http://localhost:8084/WDWJ2EE/examples/JspExample06/Receiver.jsp

5 JSP – Predefined Variables – request

GET



http://localhost:8084/WDWJ2EE/examples/JspExample06/Receiver.jsp?name=k

JSP Example 6 - Receiver

Name = Kalle Anka

Age = 42

Method = GET

4/WDWJ2EE/examples/JspExample06/Receiver.jsp?name=Kalle+Anka&age=42

5 JSP – Predefined Variables – response

- Predefined Variables – response
  - response ↔ **HttpServletResponse**
  - set Content-Type
    - `public void setContentType("text/html")`

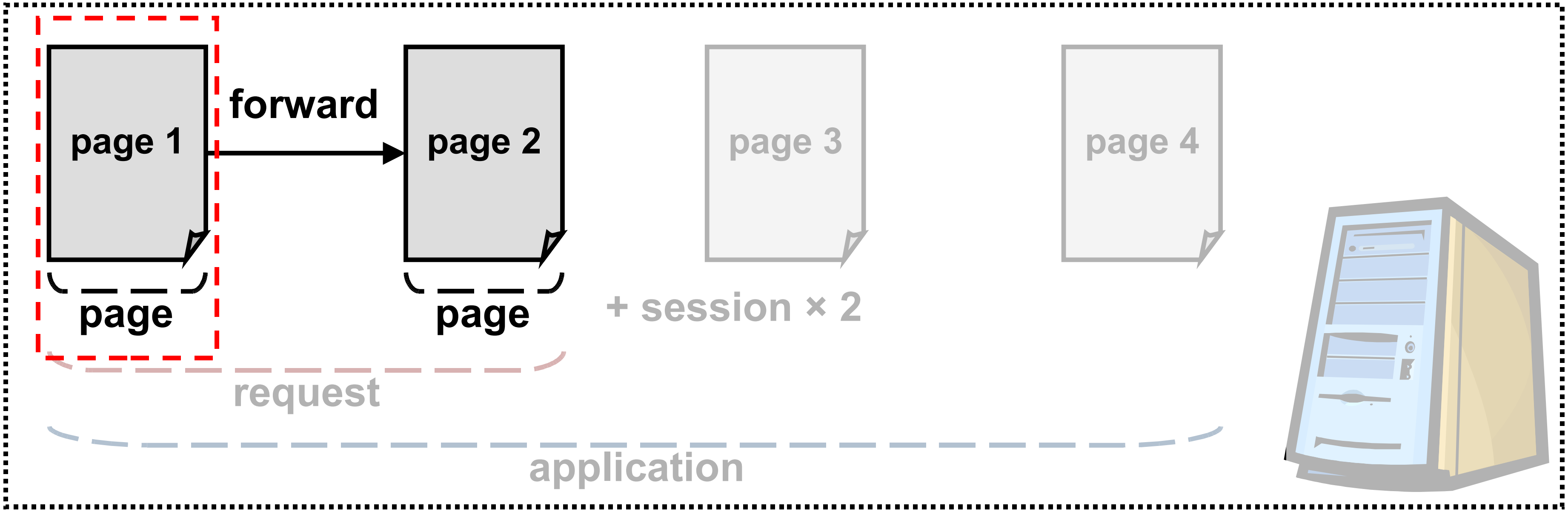
5 JSP – Predefined Variables – response

- Predefined Variables – response
  - response ↔ `HttpServletResponse`
  - set Content-Type
    - `public void setContentType("text/html")`

- alternative:
  - `<%@ page contentType="text/html" %>`
- content types
  - text/html;charset=UTF-8
  - text/plain;charset=UTF-8
  - …

5 JSP – Predefined Variables – pageContext

- Predefined Variables – pageContext
  - pageContext ↔ `PageContext`
  - can be thought of as the “environment” of the page/servlet
  - provides access to the page attributes } objects in page scope



5 JSP – Predefined Variables – session

- Predefined Variables – session
  - session ↔ **HttpSession**

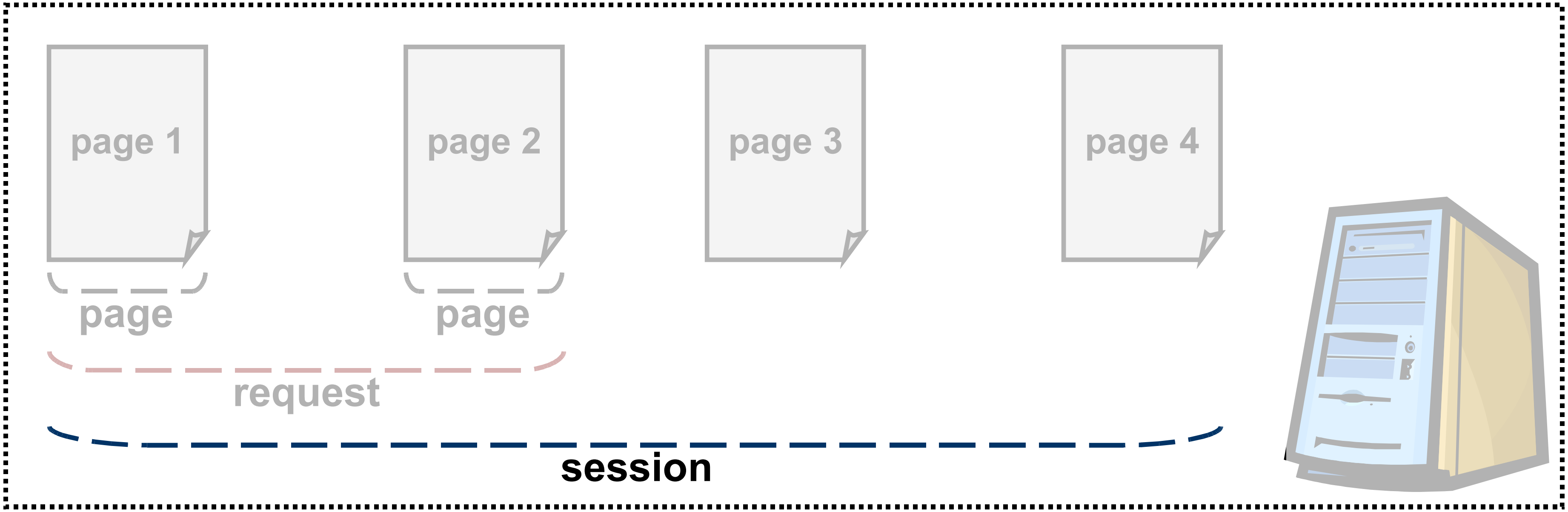
objects
in session
scope

```
public Object getAttribute("AttName")
public Enumeration getAttributeNames()
public void setAttribute("AttName", obj)
```

```
public String getId()
public long getCreationTime()
public long getLastAccessedTime()
public long getMaxInactiveInterval()
```

5 JSP – Predefined Variables – session

- Predefined Variables – session
  - a session remains active
    - as long the client web browser is not closed and …
    - … the session is not “timed out” by the web server.
  - often used to store client specific objects for a certain time



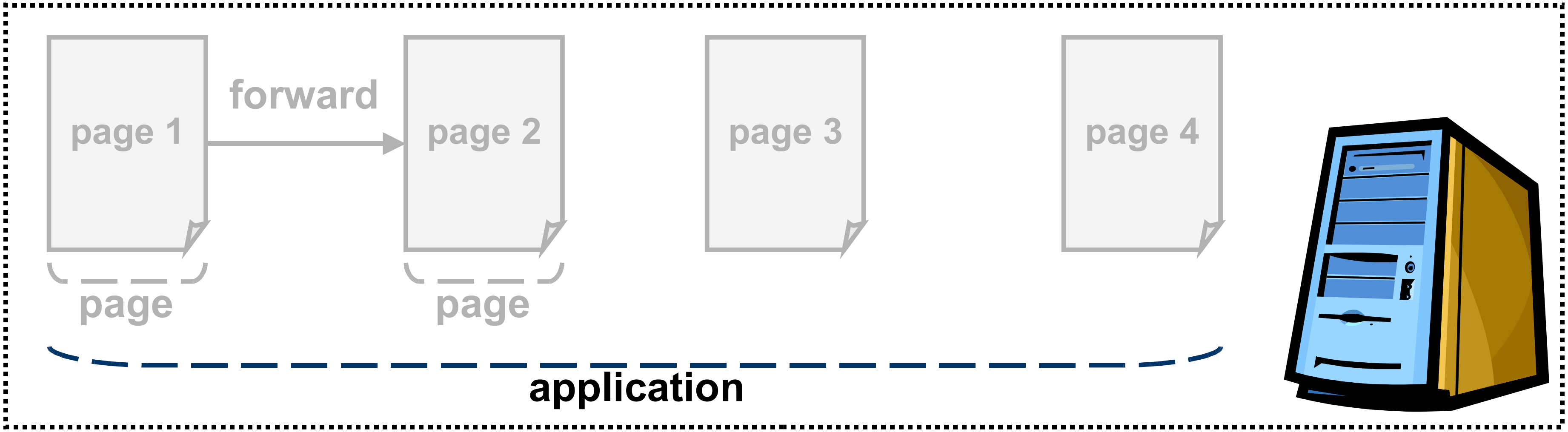
5 JSP – Predefined Variables – application

- Predefined Variables – application
  - application ↔ **ServletContext**
  - to store persistent data (rather than in instance variables)
    - `public void setAttribute("att1","att1Val")`
    - `public String getAttribute("att1")`

```
<%
  Object tempRef = application.getAttribute("hitCounter");
  if(tempref != null) {
    HitCounter counter = (HitCounter)tempRef;
    counter.increaseNumberOfPageHitsSynchronized();
  }
  else {
    application.setAttribute("hitCounter", new HitCounter());
  }
%>
```

5 JSP – Predefined Variables – application

- Predefined Variables – application
  - an application remains active
    - as long the web server runs
  - typically used to store shared objects
    - database connection
    - information from an XML file that determines the servlet behaviour



5 JSP – Predefined Variables – out

- Predefined Variables
  - out ↔ **`JspWriter`** → **`PrintWriter`**
  - a reference to the output stream

```
<p>
  <%="Hey"%>
</p>
```

```
<%
  out.println("<p>");
  out.println("Hey");
  out.println("</p>");
%>
```

5 JSP – Actions

- JSP Actions
  - XML encapsulation of “ordinary” processes
  - take place at runtime

`<jsp:useBean …/>`	*create a JavaBean for use in this JSP*
`<jsp:setProperty …/>`	*change the value of a property of a JavaBean*
`<jsp:getProperty …/>`	*read the value of a property of a JavaBean*
`<jsp:include …/>`	*include a page*
`<jsp:forward …/>`	*forward to a request*
`<jsp:param …/>`	*specification of parameters for* ***`<jsp:include …/>`*** *or* ***`<jsp:forward …/>`***
`<jsp:plugin …/>`	*specification of plugin in a non browser-specific way* *(for example instead of* ***`APPLET`*** *by* ***`OBJECT`*** *or* ***`EMBED`****)*

5 JSP – Include

- Include Action
  - `<jsp:include page="msg.jspf" [flush="false"] />`
  - *output* of the included JSP inserted at request time
    - elements are not included as JSP code
    - included elements can be the output of a JSP
  - corresponds to **`RequestDispatcher.include()`**
  - allows to send parameters
  - the parameter `page` can be determined at request time
  - used for simple insertion of HTML or plain text

- flush specifies whether the response body should be flushed before the page is included
  → "true" = the main page cannot set headers or forward to another page after a <jsp:include>

Include Action

- `<jsp:include page="msg.jspf" [flush="false"] />`
- *output* of the included JSP inserted at request time
  - elements are not included as JSP code
  - included elements can be the output of a JSP
- corresponds to **`RequestDispatcher.include()`**
- allows to send parameters
- the parameter `page` can be determined at request time
- used for simple insertion of HTML or plain text

5 JSP – Include

- Include Directive
  - `<%@ include file="incFile.jspf" %>`
  - *file* inserted at translation time (not at request time)
    - included elements can contain JSP code
    - included elements can contain method declarations
    - included elements can contain class variables
    - JSP files might not be updated if an included file changes
  - allows reuse of
    - navigation bars,
    - repeated messages,
    - common declarations, …

5 JSP – Include

- Path To The Included File
  - relative to the page
    - `<%@ include file="incFile.jspf" %>`
    - `<jsp:include page="incFile.jspf" flush="true" />`
  - relative to the application context path
    - `<%@ include file="/incFile.jspf" %>`
    - `<jsp:include page="/incFile.jspf" flush="true" />`

5 JSP – Include

ref_to_mail.jspf

```
<center>
  If you have further questions please contact
  <a href="mailto:fpa@aass.oru.se">me</a>.
</center>
```

style_inc.jspf

```
<%
  String strStyleTemplate = null;
  if(session.getAttribute("VisHandicaped") != null)
    { strStyleTemplate = "VisHandicaped.css"; } else
    { strStyleTemplate = "Standard.css"; }
%>
<link rel="stylesheet" href="<%=strStyleTemplate%>" type="text/css"/>
```

```
<html>
  <head>
    <title>A Page with Includes</title>
    <jsp:include page="style_inc.jspf" flush="false" />
  </head>
<body>
  ...
  <%@ include file="ref_to_mail.jspf" %>
</body>
</html>
```

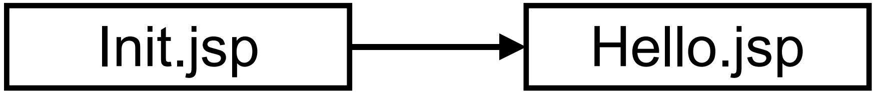
5 JSP – Forward

- Forwarding Requests
  - `<jsp:forward page="RelativeURL" />`
  - the page attribute permits JSP expressions
  - parameters can be sent as well

    -
      ```
      <jsp:forward page="RelativeURL" >
        <jsp:param name="ParName" value="ParValue" />
      </jsp:forward>
      ```

5 JSP – Forward

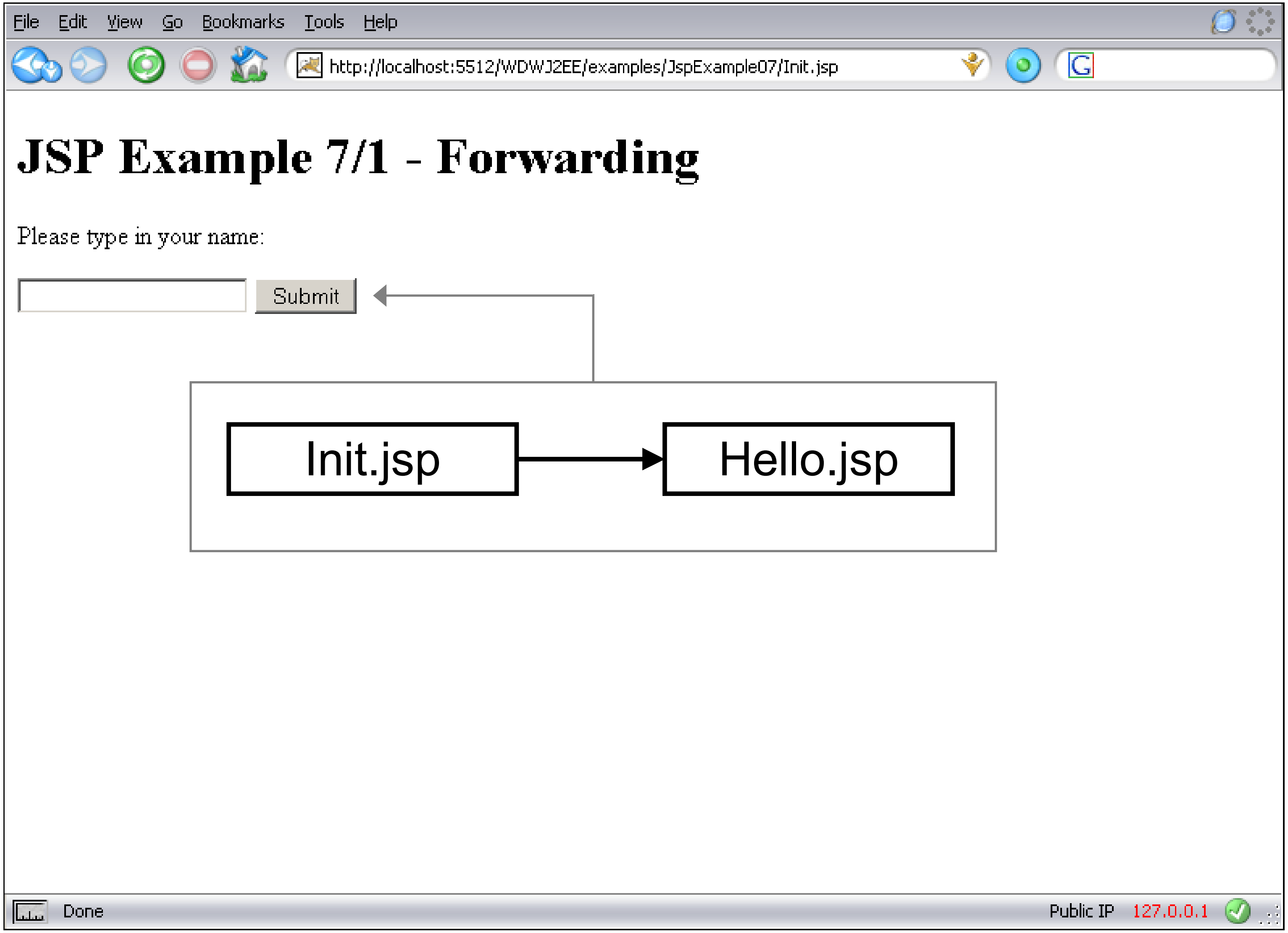
Forwarding Requests – Example 1



```
<html>
  <head>
    <title>A Forwarding Example</title>
  </head>
<body>
  <h1>JSP Example 7/1 - Forwarding</h1>
  <p>Please type in your name:</p>
  <form action="Hello.jsp" method="post">
    <input type="text" name="Name" />
    <input type="submit" name="Submit" value="Submit" />
  </form>
</body>
</html>
```

Init.jsp

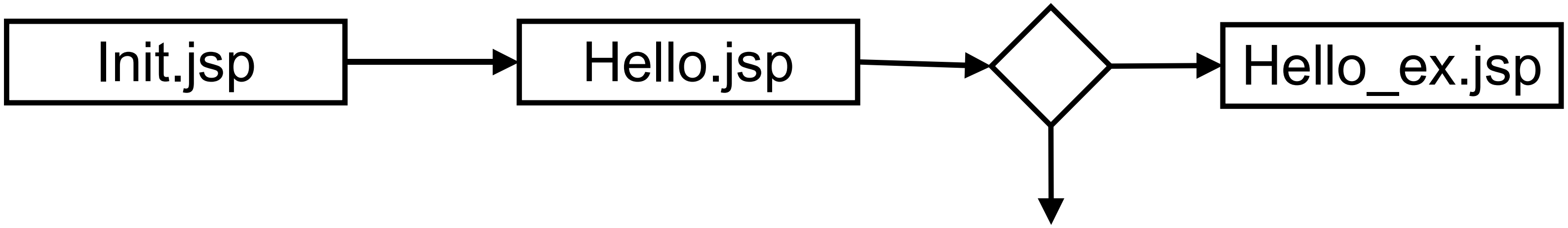
5 JSP – Forward


File Edit View Go Bookmarks Tools Help
http://localhost:5512/WDWJ2EE/examples/JspExample07/Init.jsp
Submit
Init.jsp
Hello.jsp
Done
Public IP 127.0.0.1

JSP Example 7/1 - Forwarding

Please type in your name:

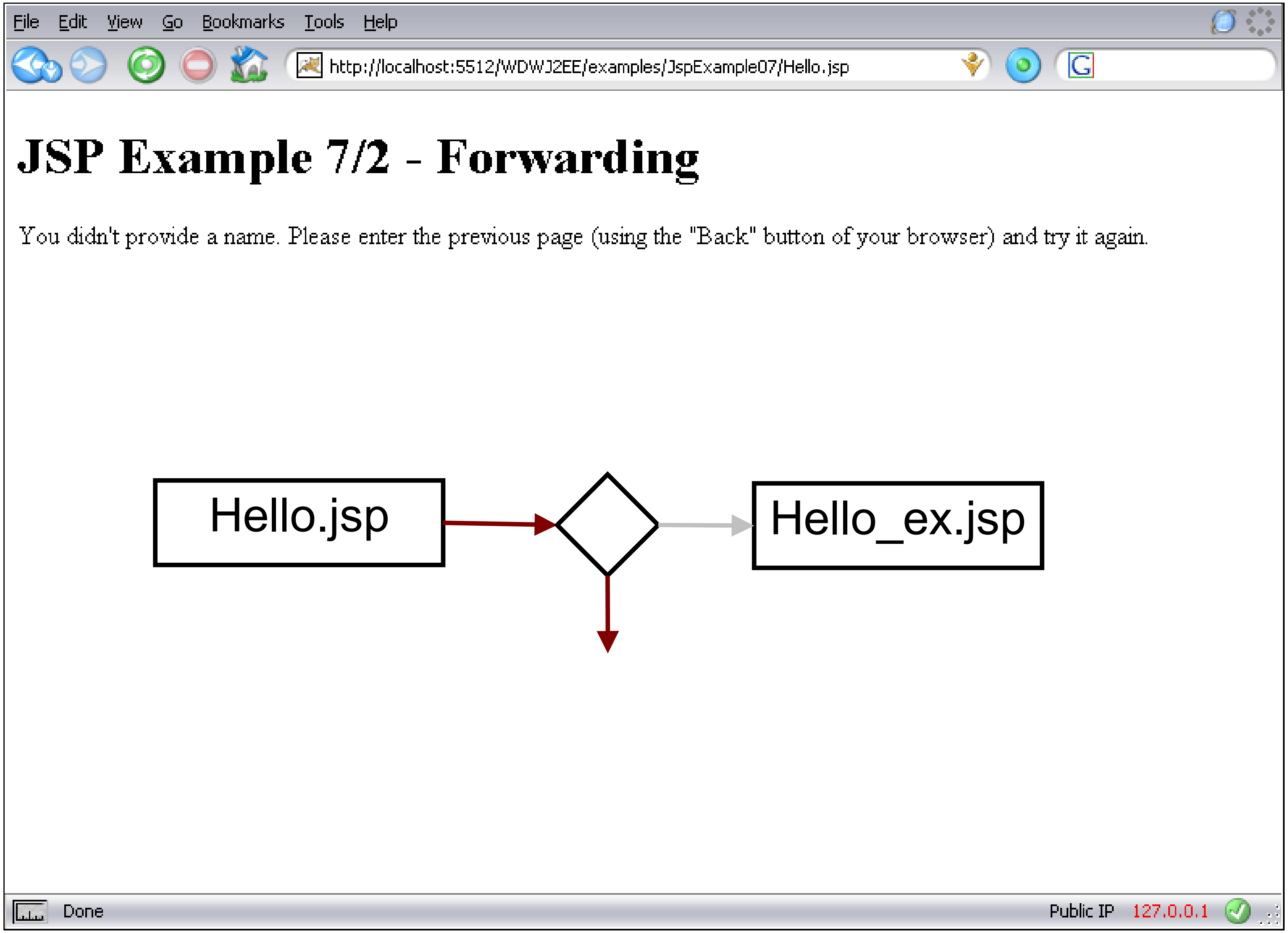
5 JSP – Forward



```
<% String strGreeting = request.getParameter("Name");
 if((strGreeting != null) && (strGreeting.length() != 0)) {
   if(Calendar.getInstance().get(Calendar.HOUR_OF_DAY) < 12) {
     strGreeting = "God förmiddag ";
   } else {
     strGreeting = "God eftermiddag ";
   } %>
  <jsp:forward page="Hello_ex.jsp">
    <jsp:param name="Greeting" value="<%=strGreeting%>" />
  </jsp:forward>
<% } %>
...
  <h1>JSP Example 7/2 - Forwarding</h1>
  <p>You didn't provide a name. Please enter the previous page (using the "Back"
button of your browser) and try it again.</p>
</body>
</html>
```

Hello.jsp

5 JSP – Forward

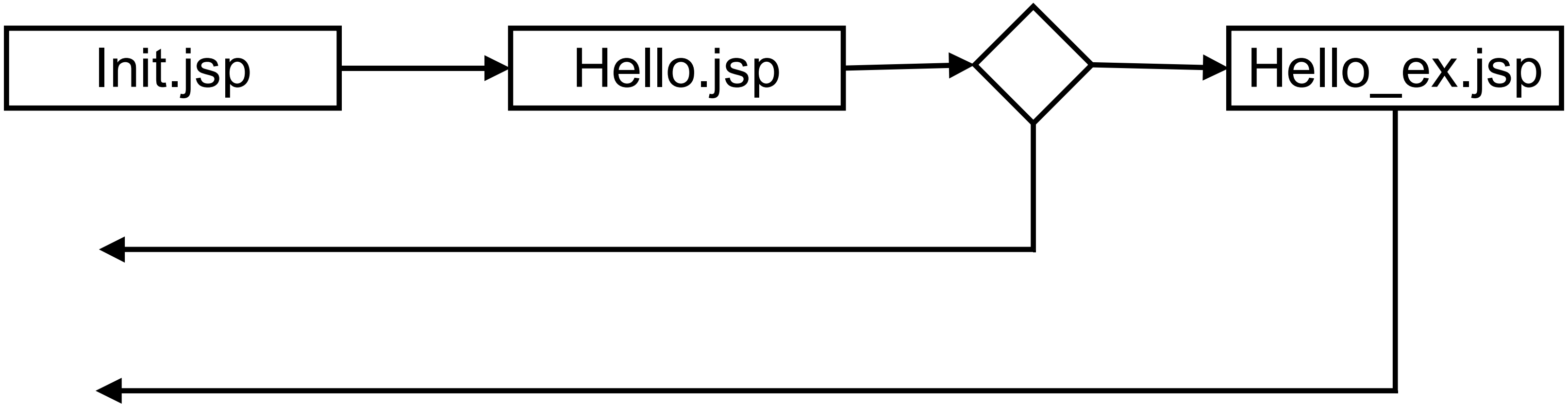


http://localhost:5512/WDWJ2EE/examples/JspExample07/Hello.jsp

JSP Example 7/2 - Forwarding

You didn't provide a name. Please enter the previous page (using the "Back" button of your browser) and try it again.

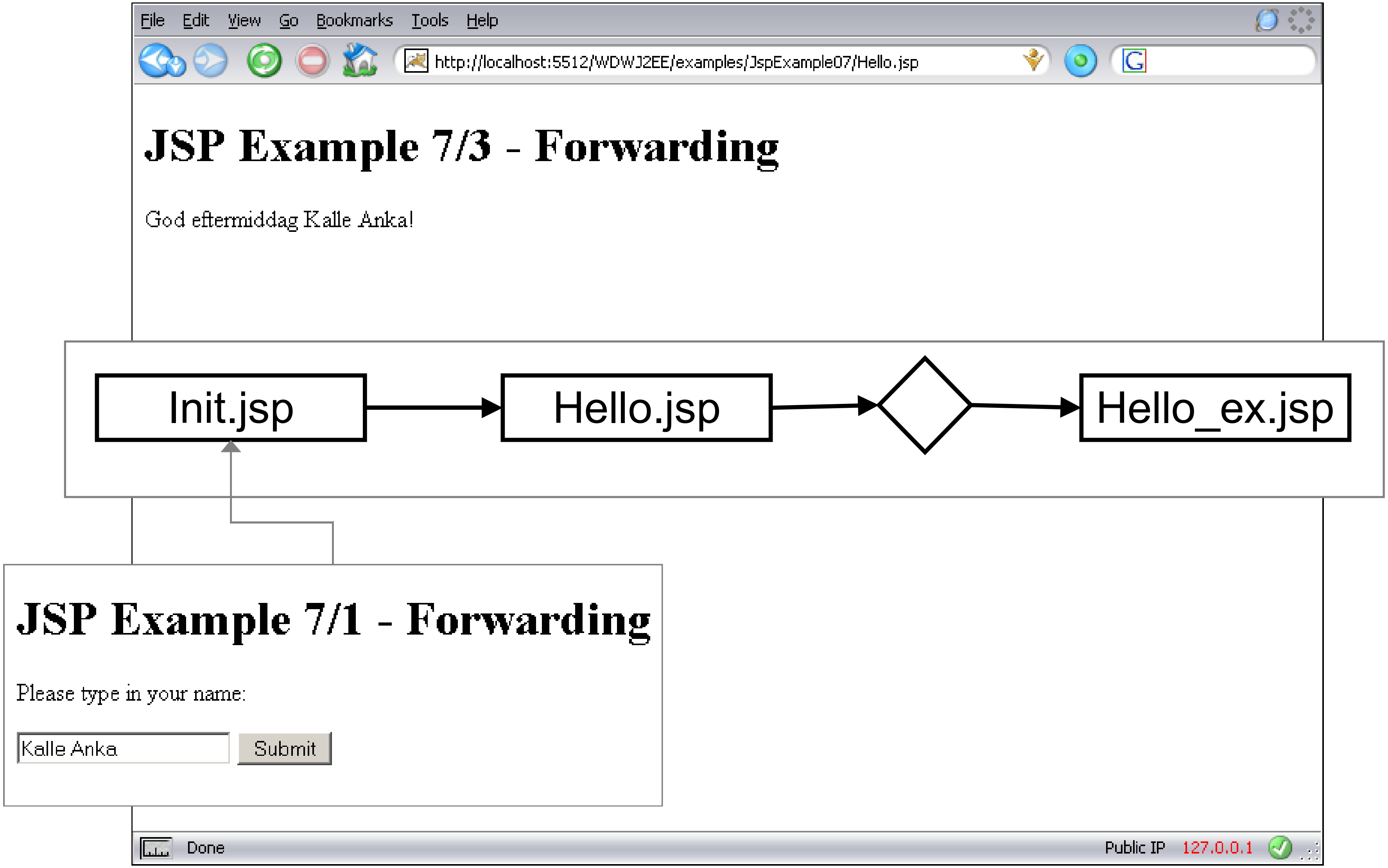
5 JSP – Forward



```
<%
   // Collect parameters from the original request and the ones inserted by forward
   String strName = request.getParameter("Name");
   String strGreeting = request.getParameter("Greeting");
%>

   <h1>Forwarding Example - Execute Standard Hello</h1>
   <p><%=strGreeting%><%=strName%>!</p>
</body>
</html>
```

Hello_ex.jsp

5 JSP – Forward


File Edit View Go Bookmarks Tools Help
http://localhost:5512/WDWJ2EE/examples/JspExample07/Hello.jsp
Init.jsp
Hello.jsp
Hello_ex.jsp
Submit
Done
Public IP 127.0.0.1

JSP Example 7/3 - Forwarding

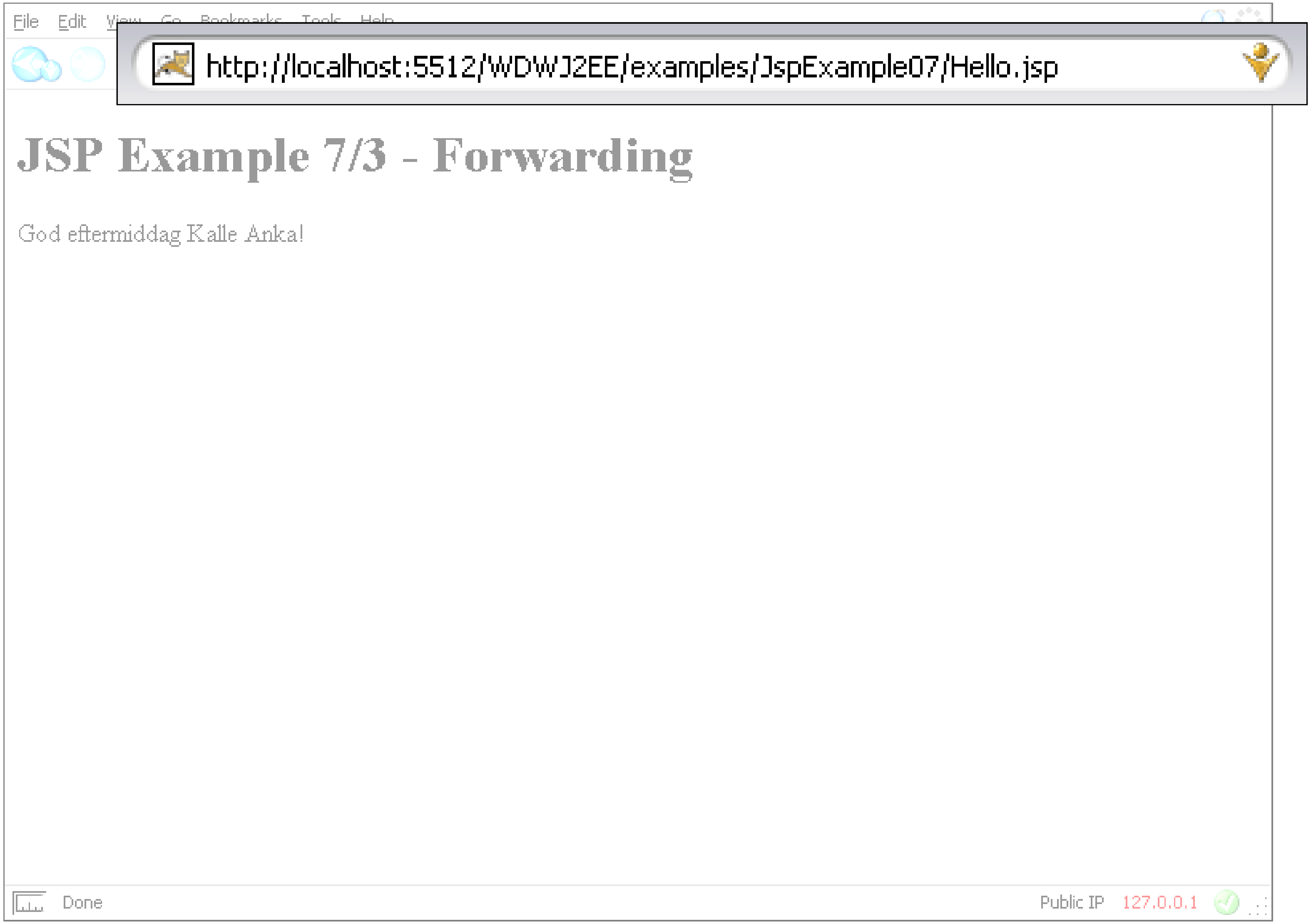
God eftermiddag Kalle Anka!

JSP Example 7/1 - Forwarding

Please type in your name:

Kalle Anka

5 JSP – Forward



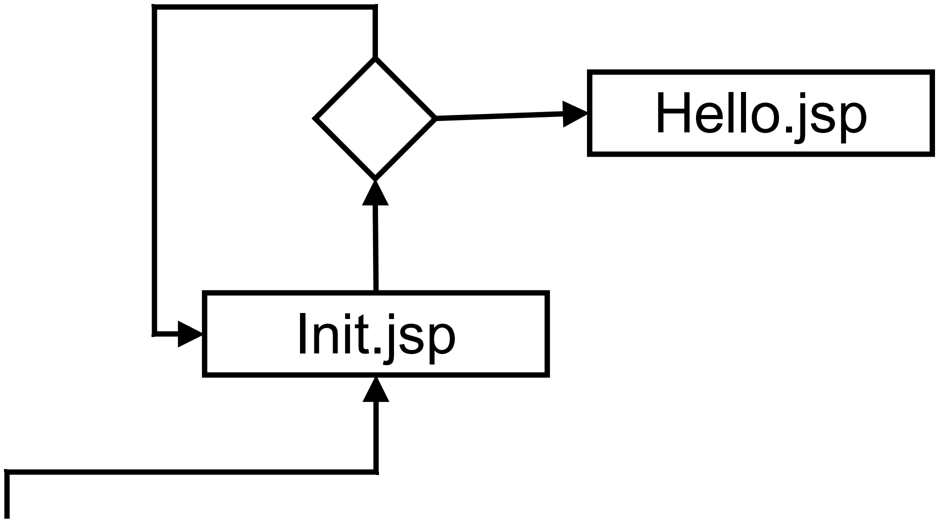
http://localhost:5512/WDWJ2EE/examples/JspExample07/Hello.jsp

JSP Example 7/3 - Forwarding

God eftermiddag Kalle Anka!

5 JSP – Forward

- Forwarding Requests – Example 2 (“Self Forwarding”)

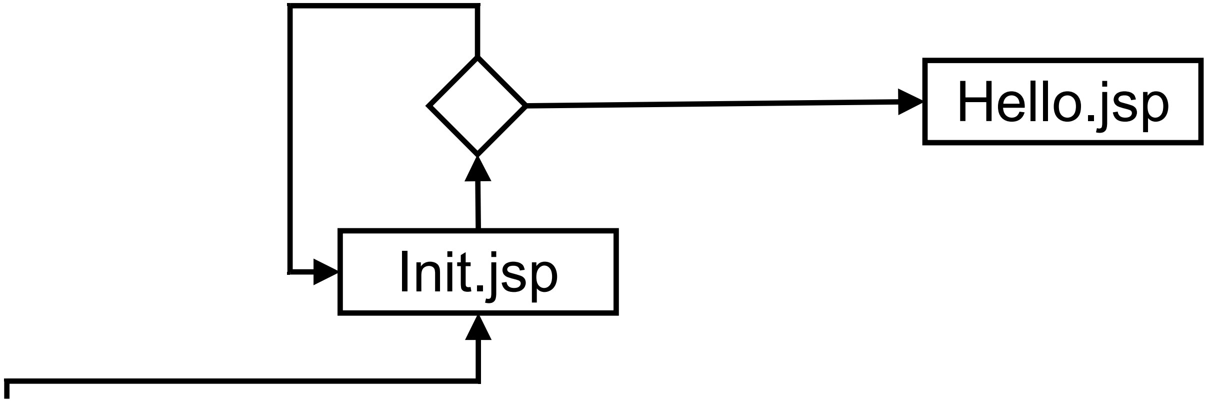


5 JSP – Forward

```
<html>
  <head><title>A Self Forwarding Example</title></head>
<body>
  <h1>JSP Example 8/1 - Self Forwarding</h1>
<% String strName = request.getParameter("Name");
  if(strName == null) { %>
  <p>Please type in your name:</p>
  <form action="Init.jsp" method="post">
    <input type="text" name="Name" />
    <input type="submit" name="Submit" value="Submit" />
  </form>
<% } else {
     if(strName.length() == 0) { %>
 <p>You didn't provide a name. Please try again:</p>
  <form action="Init.jsp" method="post">
    <input type="text" name="Name" />
    <input type="submit" name="Submit" value="Submit" />
  </form>
<%   } else { %>
  <jsp:forward page="Hello.jsp" />
<%   }
   } %>
</body></html>
```

Init.jsp

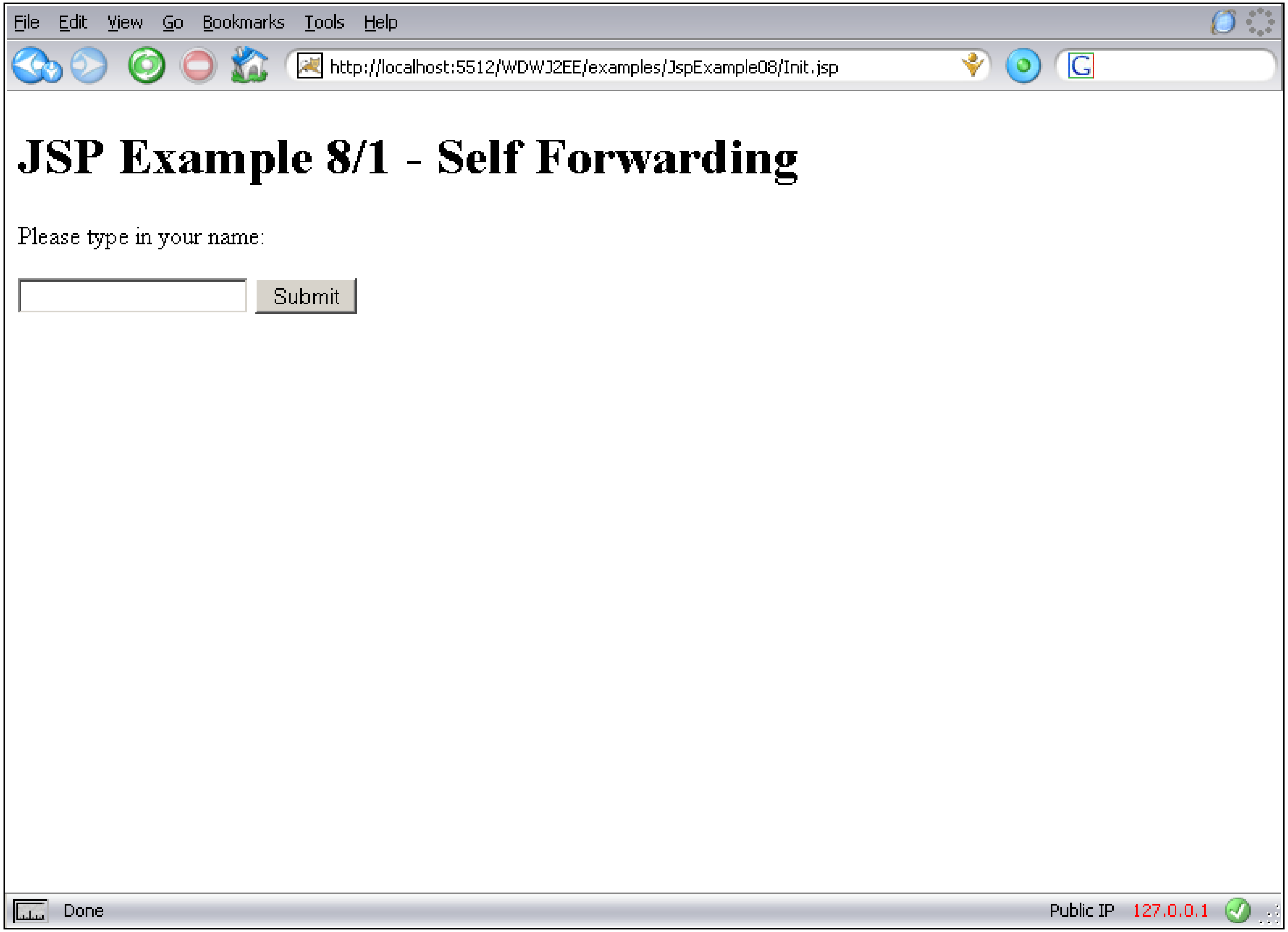
5 JSP – Forward



```
<%
   String strName = request.getParameter("Name");
   String strGreeting;
   if(Calendar.getInstance().get(Calendar.HOUR_OF_DAY) < 12) {
     strGreeting = "God förmiddag ";
   } else {
     strGreeting = "God eftermiddag ";
   }
%>
<html>
   <head><title>A Self Forwarding Example</title></head>
<body>
   <h1>JSP Example 8/2 - Self Forwarding</h1>
   <p><%=strGreeting%> <%=strName%>!</p>
</body></html>
```

Hello.jsp

5 JSP – Forward

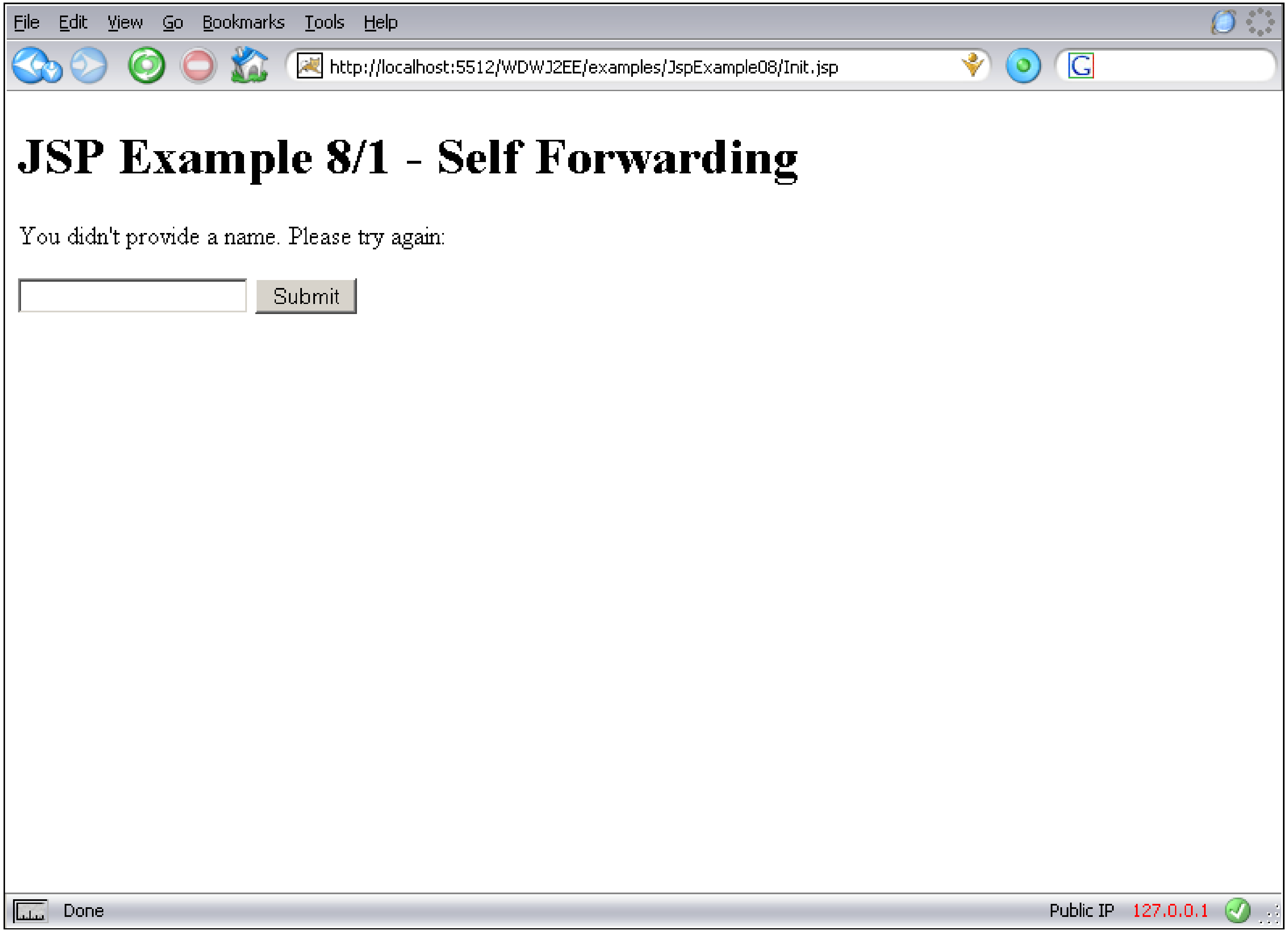

File Edit View Go Bookmarks Tools Help
http://localhost:5512/WDWJ2EE/examples/JspExample08/Init.jsp
Done
Public IP 127.0.0.1

JSP Example 8/1 - Self Forwarding

Please type in your name:

Submit

5 JSP – Forward



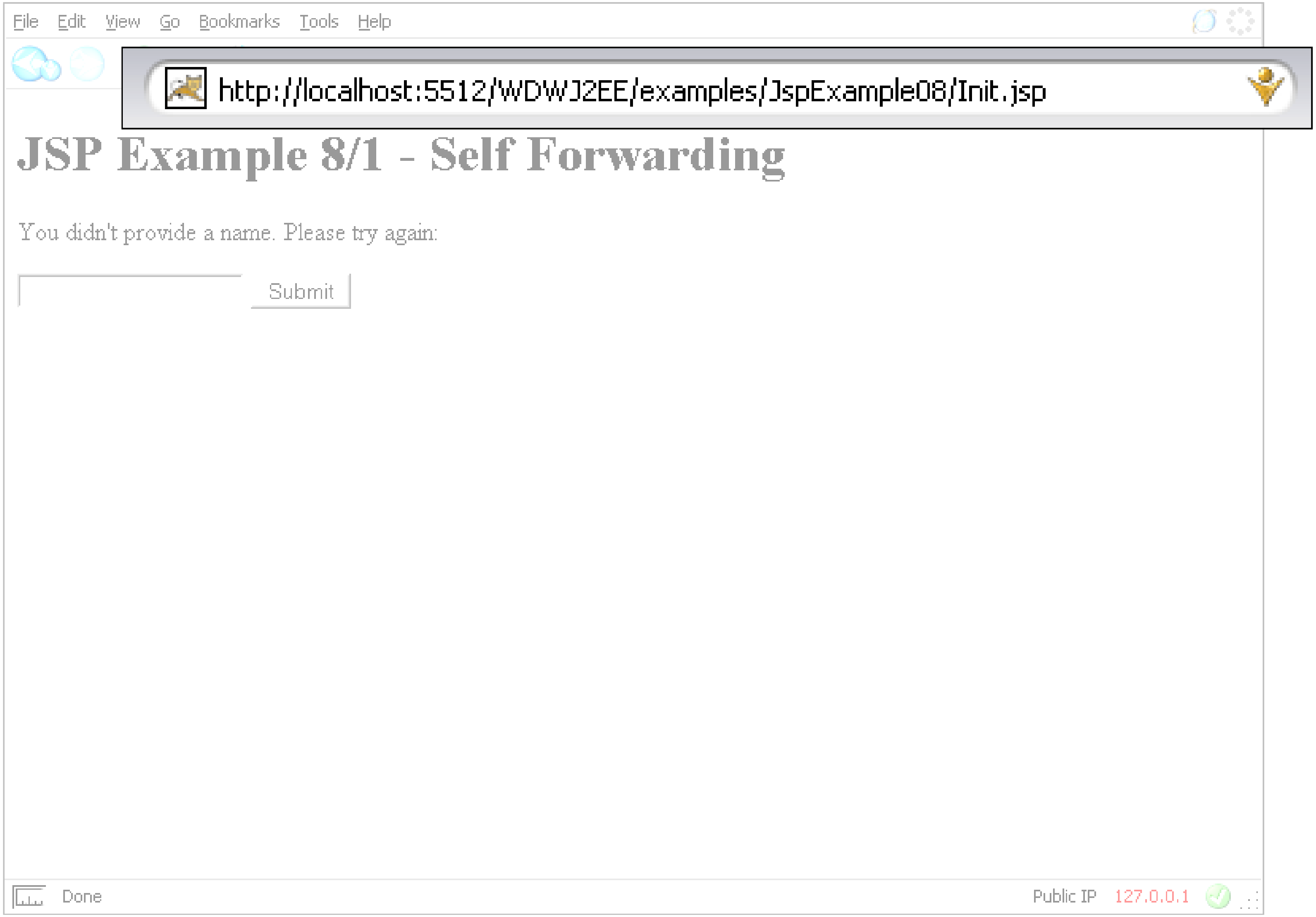
File Edit View Go Bookmarks Tools Help
http://localhost:5512/WDWJ2EE/examples/JspExample08/Init.jsp
Done
Public IP 127.0.0.1

JSP Example 8/1 - Self Forwarding

You didn't provide a name. Please try again:

Submit

5 JSP – Forward



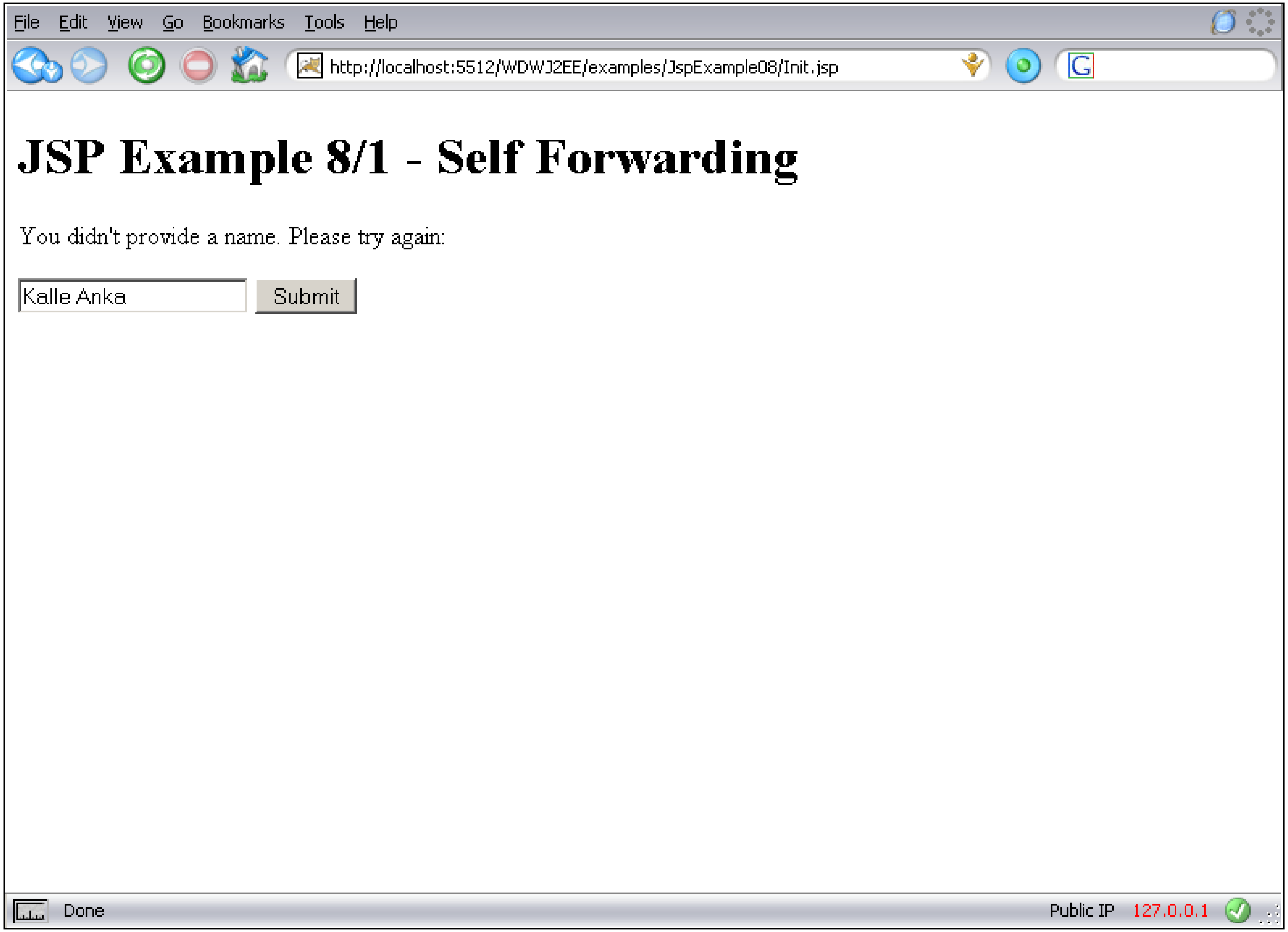
http://localhost:5512/WDWJ2EE/examples/JspExample08/Init.jsp

JSP Example 8/1 - Self Forwarding

You didn't provide a name. Please try again:

Submit

5 JSP – Forward

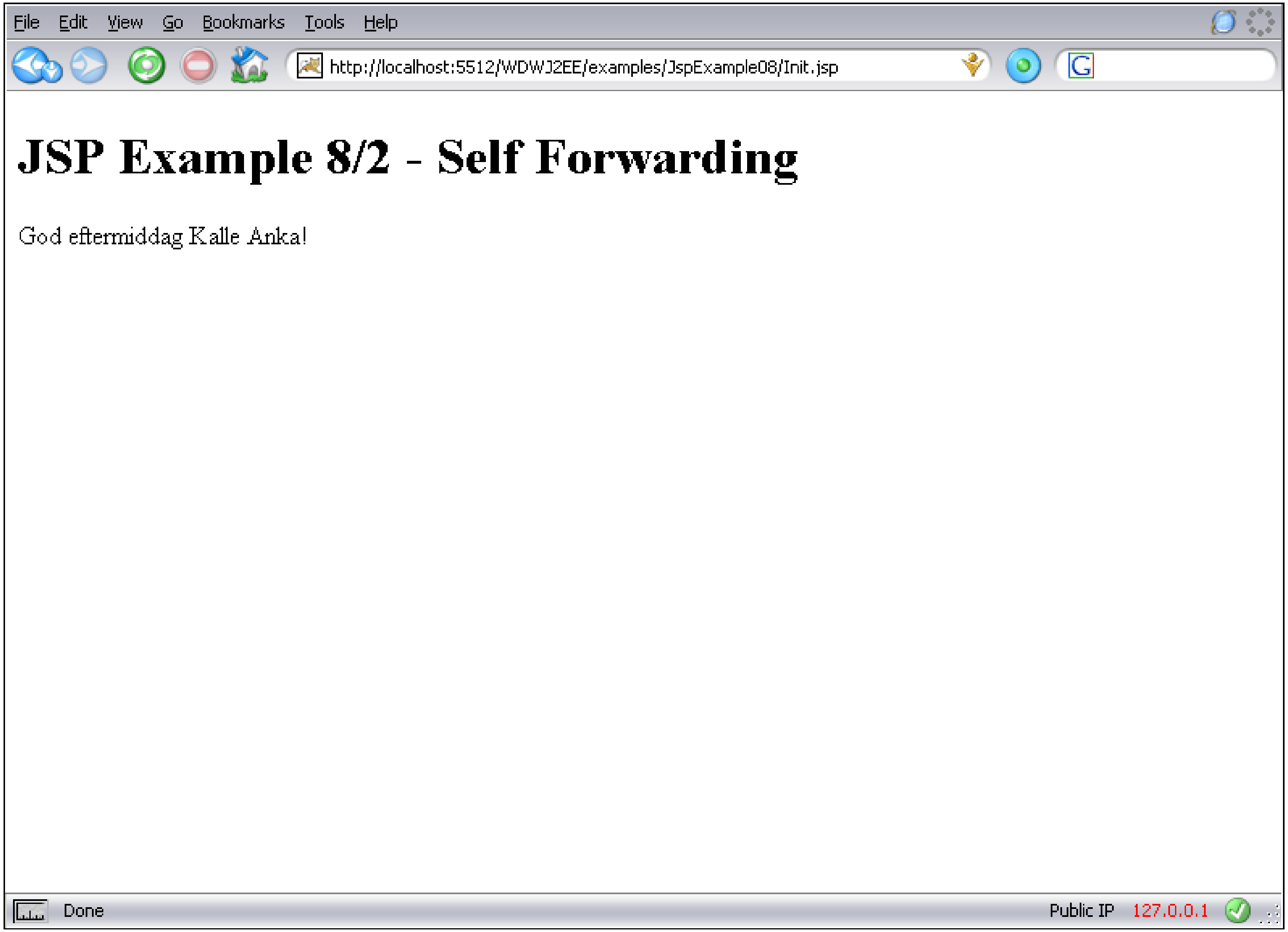

File Edit View Go Bookmarks Tools Help
http://localhost:5512/WDWJ2EE/examples/JspExample08/Init.jsp
Done
Public IP 127.0.0.1

JSP Example 8/1 - Self Forwarding

You didn't provide a name. Please try again:

Kalle Anka [Submit]

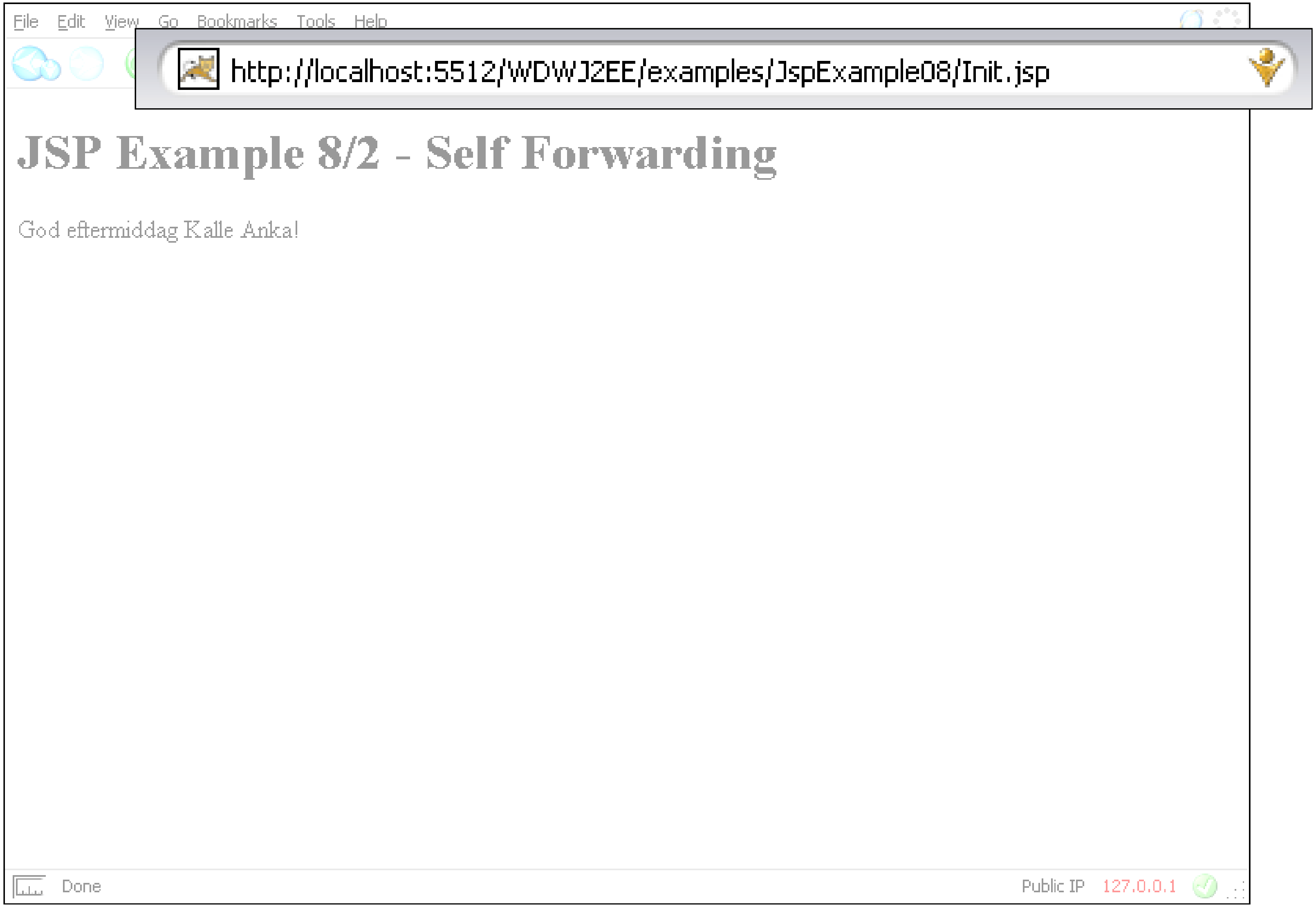
5 JSP – Forward


File
Edit
View
Go
Bookmarks
Tools
Help
http://localhost:5512/WDWJ2EE/examples/JspExample08/Init.jsp
Done
Public IP
127.0.0.1

JSP Example 8/2 - Self Forwarding

God eftermiddag Kalle Anka!

5 JSP – Forward



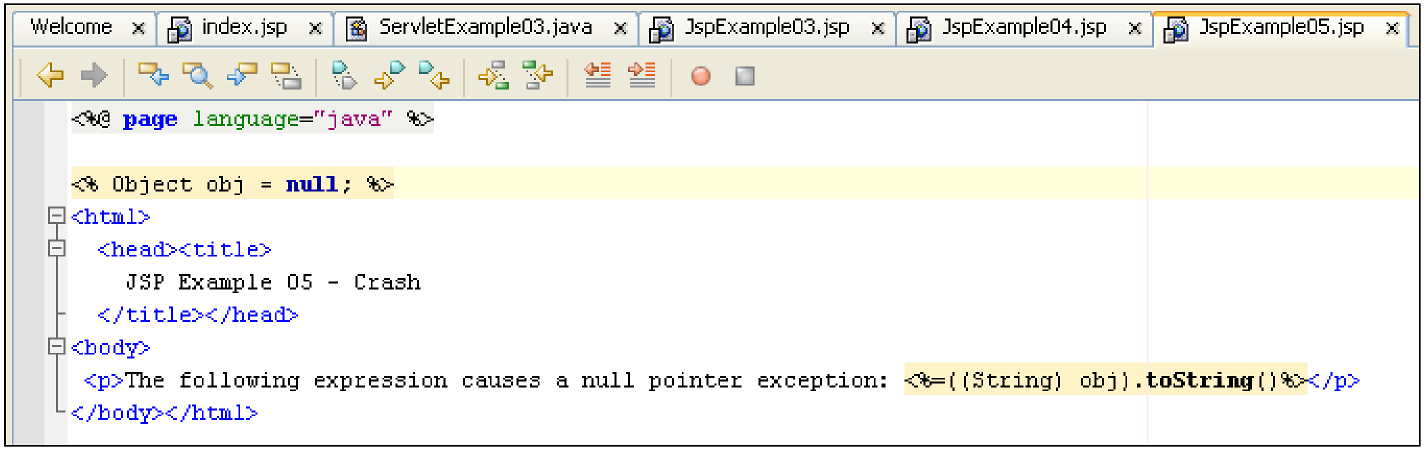
http://localhost:5512/WDWJ2EE/examples/JspExample08/Init.jsp

JSP Example 8/2 - Self Forwarding

God eftermiddag Kalle Anka!

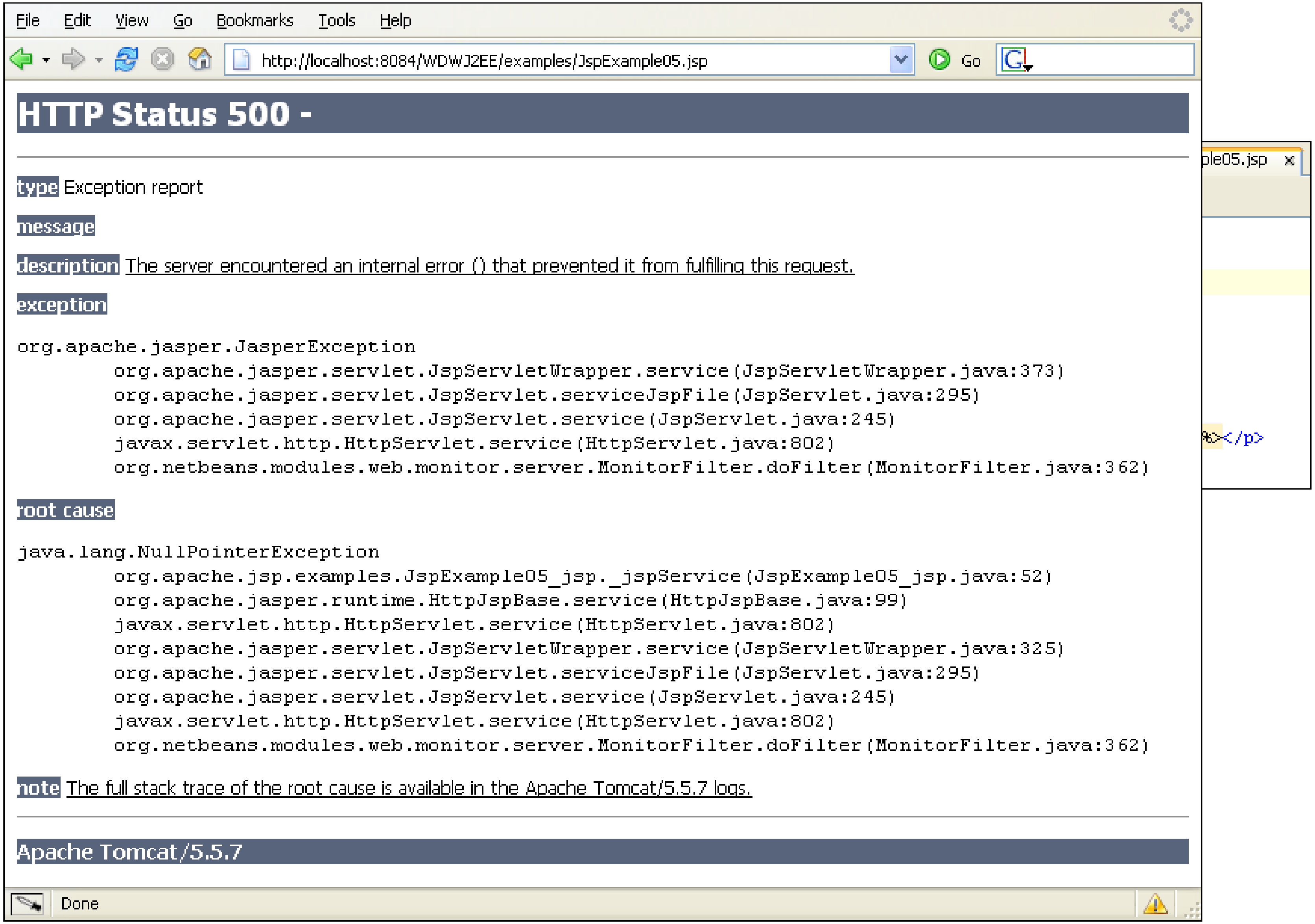
5 JSP – Exceptions

- A null pointer exception



```
<%@ page language="java" %>

<% Object obj = null; %>
<html>
  <head><title>
    JSP Example 05 - Crash
  </title></head>
<body>
 <p>The following expression causes a null pointer exception: <%=((String) obj).toString()%></p>
</body></html>
```

5 JSP – Exceptions



http://localhost:8084/WDWJ2EE/examples/JspExample05.jsp

HTTP Status 500 -

type Exception report

message

description The server encountered an internal error () that prevented it from fulfilling this request.

exception

```
org.apache.jasper.JasperException
	org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:373)
	org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:295)
	org.apache.jasper.servlet.JspServlet.service(JspServlet.java:245)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
	org.netbeans.modules.web.monitor.server.MonitorFilter.doFilter(MonitorFilter.java:362)
```

root cause

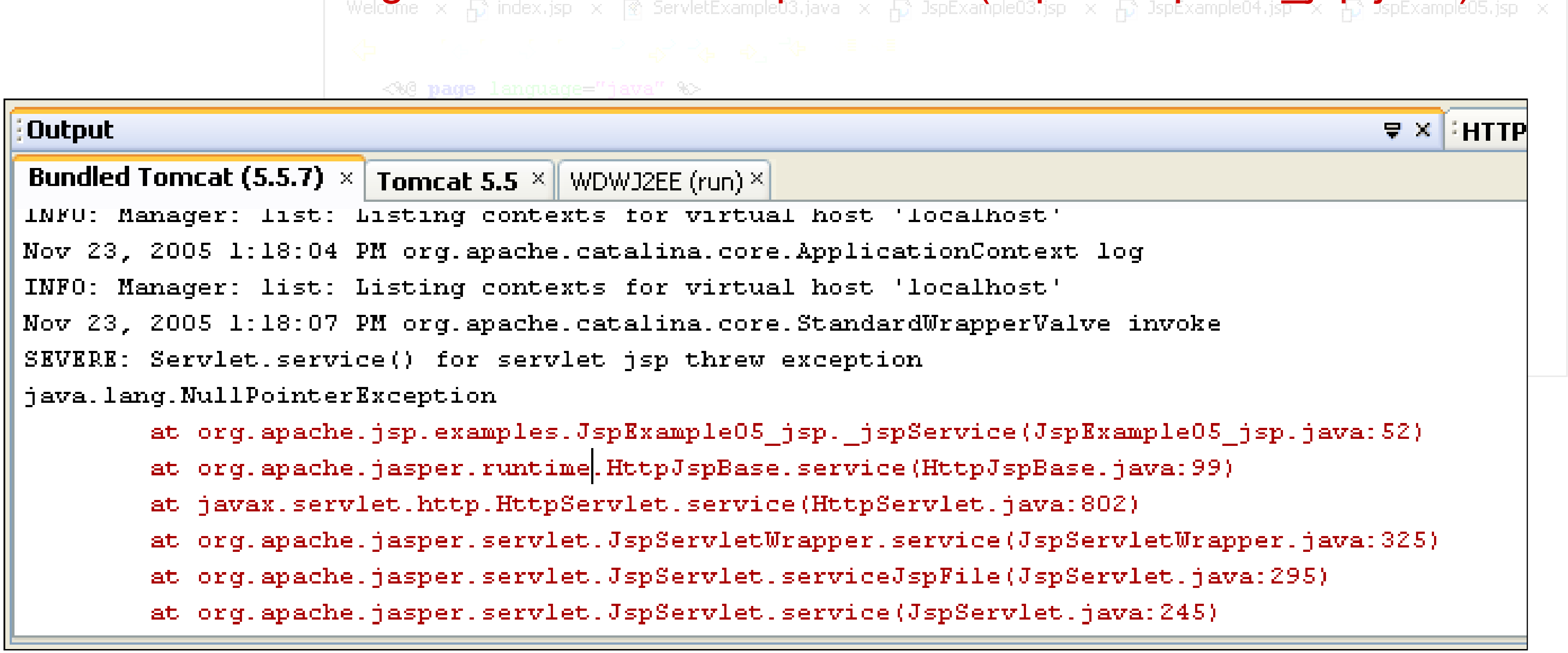
```
java.lang.NullPointerException
	org.apache.jsp.examples.JspExample05_jsp._jspService(JspExample05_jsp.java:52)
	org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:99)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
	org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:325)
	org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:295)
	org.apache.jasper.servlet.JspServlet.service(JspServlet.java:245)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
	org.netbeans.modules.web.monitor.server.MonitorFilter.doFilter(MonitorFilter.java:362)
```

note The full stack trace of the root cause is available in the Apache Tomcat/5.5.7 logs.

Apache Tomcat/5.5.7

5 JSP – Exceptions

- A null pointer exception
  - error messages refer to the compiled JSP (JspExample05_jsp.java)



```
INFO: Manager: list: Listing contexts for virtual host 'localhost'
Nov 23, 2005 1:18:04 PM org.apache.catalina.core.ApplicationContext log
INFO: Manager: list: Listing contexts for virtual host 'localhost'
Nov 23, 2005 1:18:07 PM org.apache.catalina.core.StandardWrapperValve invoke
SEVERE: Servlet.service() for servlet jsp threw exception
java.lang.NullPointerException
        at org.apache.jsp.examples.JspExample05_jsp._jspService(JspExample05_jsp.java:52)
        at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:99)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
        at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:325)
        at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:295)
        at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:245)
```

5 JSP – Exceptions

- Exception Handling
  - `try … catch` blocks in the Java code
  - what happens with uncaught exceptions ?
- redirecting the control flow in case of exceptions:
  - refer to an “error page” from JSPs

    ```
    <%@ page errorPage="/ShowException.jsp" %>
    ```

  - declare the linked page as “error page”

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

5 JSP – Exceptions

Exception Handling

```
<%@ page errorPage="ShowException.jsp" %>
<% Object obj = null; %>
<html>
  <head><title>JSP Example 9 – Exception Handling</title></head>
<body>
  <h1>JSP Example 9 – Exception Handling</h1>
  <p>
    The following expression causes a null pointer exception
    <%=((String) obj).toString()%>
  </p>
</body></html>
```

Init.jsp

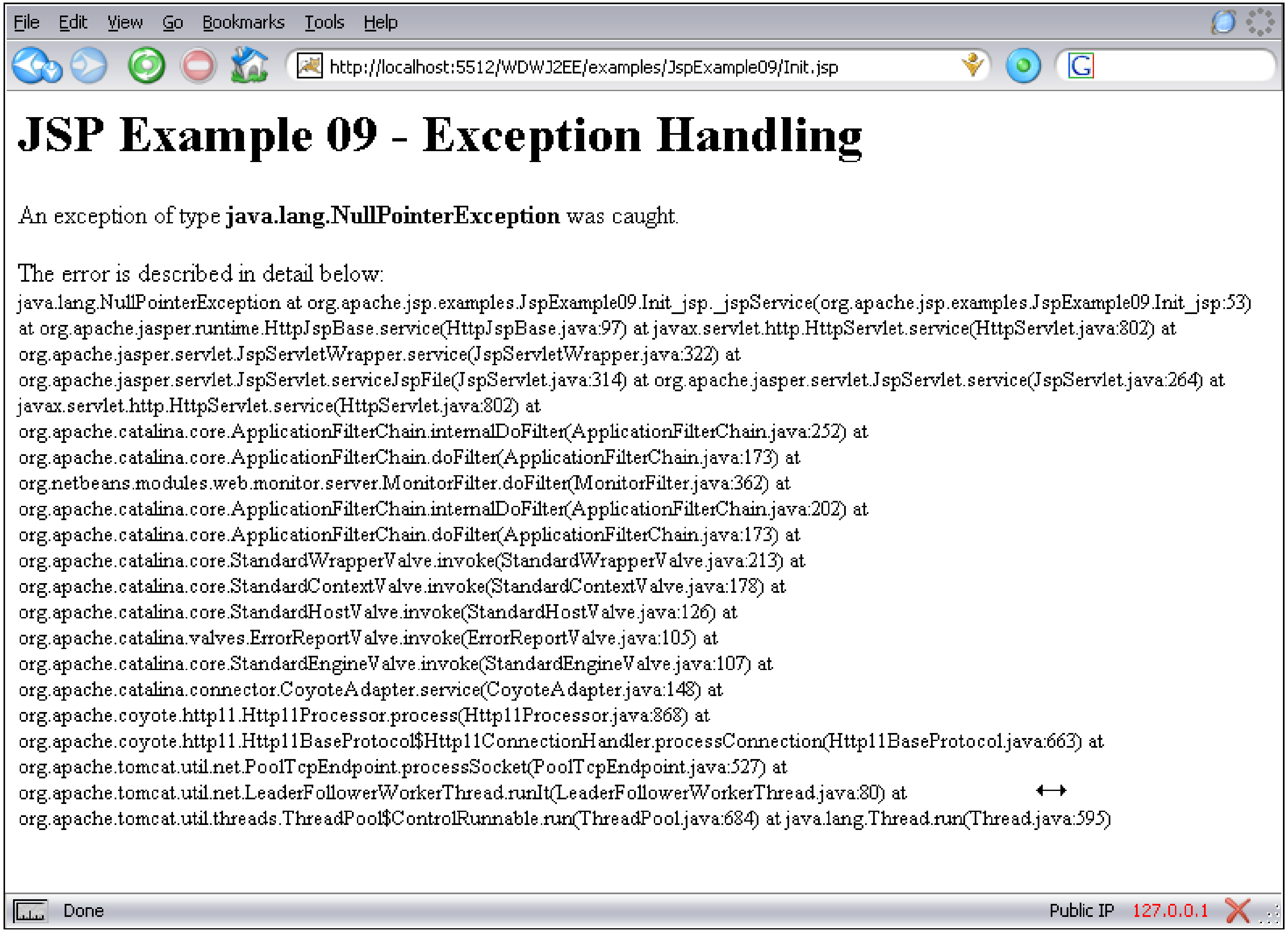
5 JSP – Exceptions

! the exception property of the pageContext variable is initialized with a reference to the exception thrown (java.lang.Throwable)

```
<%@ page isErrorPage="true" %>
<%@ page import="java.lang.io.*" %>
<%
  Throwable e = exception;
  String strException = "";
  if(e instanceof NullPointerException) {
    strException = e.toString();
  }
%>
<html>
  <head><title>JSP Example 9 - Exception Handling</title></head>
<body>

  <h1>JSP Example 9 - Exception Handling</h1>
<% if(strException.length() != 0) { %>
  <p>
    An exception of type <b><%=exception%></b> was caught.
  </p>
<% } %>
 <p>
   The error is described in detail below:<br/>
   <font size="-1"><% e.printStackTrace(new java.io.PrintWriter(out)); %></font>
 </p>
</body></html>
```

ShowException.jsp

5 JSP – Exceptions



http://localhost:5512/WDWJ2EE/examples/JspExample09/Init.jsp

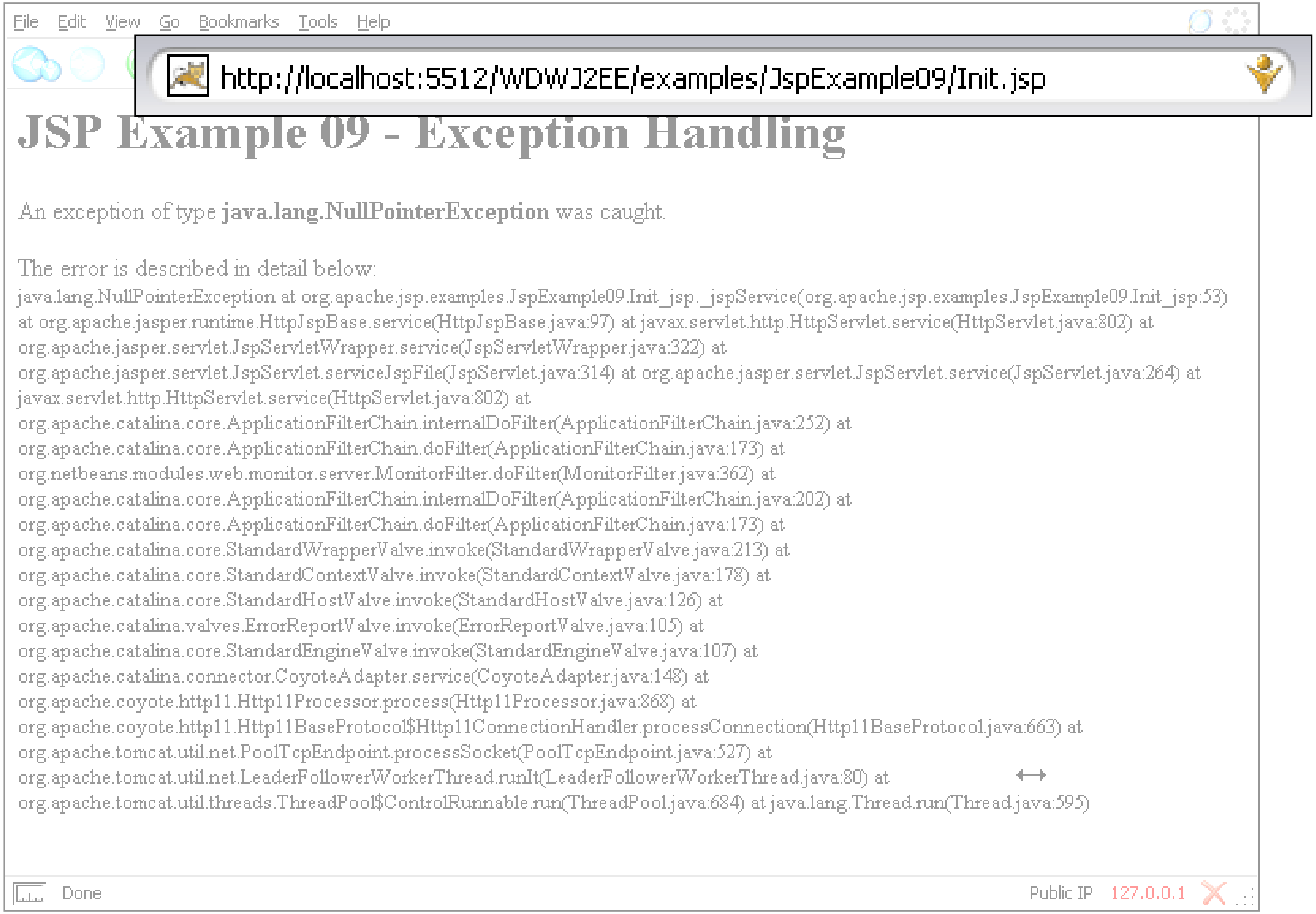
JSP Example 09 - Exception Handling

An exception of type **java.lang.NullPointerException** was caught.

The error is described in detail below:

```
java.lang.NullPointerException at org.apache.jsp.examples.JspExample09.Init_jsp._jspService(org.apache.jsp.examples.JspExample09.Init_jsp:53)
at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:97) at javax.servlet.http.HttpServlet.service(HttpServlet.java:802) at
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:322) at
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:314) at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:264) at
javax.servlet.http.HttpServlet.service(HttpServlet.java:802) at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:252) at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:173) at
org.netbeans.modules.web.monitor.server.MonitorFilter.doFilter(MonitorFilter.java:362) at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:202) at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:173) at
org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:213) at
org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:178) at
org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:126) at
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:105) at
org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:107) at
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:148) at
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:868) at
org.apache.coyote.http11.Http11BaseProtocol$Http11ConnectionHandler.processConnection(Http11BaseProtocol.java:663) at
org.apache.tomcat.util.net.PoolTcpEndpoint.processSocket(PoolTcpEndpoint.java:527) at
org.apache.tomcat.util.net.LeaderFollowerWorkerThread.runIt(LeaderFollowerWorkerThread.java:80) at
org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(ThreadPool.java:684) at java.lang.Thread.run(Thread.java:595)
```

5 JSP – Exceptions



http://localhost:5512/WDWJ2EE/examples/JspExample09/Init.jsp

JSP Example 09 - Exception Handling

An exception of type **java.lang.NullPointerException** was caught.

The error is described in detail below:

```
java.lang.NullPointerException at org.apache.jsp.examples.JspExample09.Init_jsp._jspService(org.apache.jsp.examples.JspExample09.Init_jsp:53)
at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:97) at javax.servlet.http.HttpServlet.service(HttpServlet.java:802) at
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:322) at
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:314) at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:264) at
javax.servlet.http.HttpServlet.service(HttpServlet.java:802) at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:252) at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:173) at
org.netbeans.modules.web.monitor.server.MonitorFilter.doFilter(MonitorFilter.java:362) at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:202) at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:173) at
org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:213) at
org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:178) at
org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:126) at
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:105) at
org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:107) at
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:148) at
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:868) at
org.apache.coyote.http11.Http11BaseProtocol$Http11ConnectionHandler.processConnection(Http11BaseProtocol.java:663) at
org.apache.tomcat.util.net.PoolTcpEndpoint.processSocket(PoolTcpEndpoint.java:527) at
org.apache.tomcat.util.net.LeaderFollowerWorkerThread.runIt(LeaderFollowerWorkerThread.java:80) at
org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(ThreadPool.java:684) at java.lang.Thread.run(Thread.java:595)
```

5 JSP – Dynamic Attributes

- Attributes That Permit JSP Expressions
  - `jsp:setProperty` (Attribute: `value`)
  - `jsp:include` (Attribute: `page`)
  - `jsp:forward` (Attribute: `page`)
  - `jsp:param` (Attribute: `value`)

5 JSP – JSP 1.2 / JSP 2.0

JSP 1.2 vs. JSP 2.0

JSP 1.2

```
<% %>

<%= %>

<%! %>

<%@ include file = "..."/>
<%@ page import = "..."/>
<%@ page contentType = "..."/>
...
```

JSP 2.0

```
<jsp:scriptlet>
  java code
</jsp:scriptlet>

<jsp:expression>
  an expression
</jsp:expression>

<jsp:declaration>
  a declaration
</jsp:declaration>

<jsp:directive.include file = "..."/>
<jsp:directive.page import = "..."/>
<jsp:directive.page contentType = "..."/>
...
```

5 JSP – Special Characters

- Escape Sequences
  - what if the character " is needed in an attribute value?
    - use \"
  - what if I need a <% in HTML?
    - use <\% or \%>

Contents

1. Introduction
   - Web Applications, J2EE, Servlets, JSP, Literature
2. Installation
   - JDK, Tomcat, NetBeans
3. JSP – Basics
4. Servlets – Basics
5. JSP – Details
   - Comments, Thread Safety, Predefined Variables, Scopes, JSP Actions, Exceptions, …

Web Development with Java EE

Introduction to Servlets and JSP

Thank you!

Appendix – Tomcat setup for development

- Servlet reloading
  - deploy a modified servlet .class file without restarting the server
- Invoker servlet
  - drop a servlet .class file into `webapps/appName/WEB-INF/classes/pkgName` and immediately run it
- Directory listings
  - show directory listing when requested URL ends in /

Appendix – Tomcat setup for development

- Servlet reloading
  - in file $CATALINA_HOME/conf/context.xml, add:

```
<Context>
```

```
<Context reloadable="true" privileged="true">
```

Appendix – Tomcat setup for development

- Invoker servlet
  - in file $CATALINA_HOME/conf/web.xml, un-comment:

```
<servlet>
  <servlet-name>invoker</servlet-name>
  <servlet-class>
    org.apache.catalina.servlets.InvokerServlet
  </servlet-class>
</servlet>
```

Appendix – Tomcat setup for development

- Directory listings
  - in file $CATALINA_HOME/conf/web.xml, set:

```
<init-param>
  <param-name>listings</param-name>
  <param-value>true</param-value>
</init-param>
```