

Lecture 6

Generating the Server Responses: HTTP Response Headers

Lecture Agenda

Applied

- 1 ➤ Format of the HTTP response.
- 2 ➤ Setting Response Headers.
- 3 ➤ Understanding What Response Headers are good for.
- 4 ➤ Building Excel Spreadsheets
- 5 ➤ Caching Web Pages
- 6 ➤ Generating JPEG images dynamically.

Review: Anatomy of HTTP Request/Response

A Typical HTTP GET Request Header

Review of HTTP GET Request Header

GET request with data passed (**color=dark**, **taste=malty**) on request header line.

HTTP GET Request

GET /select/selectBeerTaste.jsp?**color=dark&taste=malty** HTTP/1.1

Host: www.wickedlysmart.com

User-Agent: Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.4) Gecko/20030624 Netscape/7.1

Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

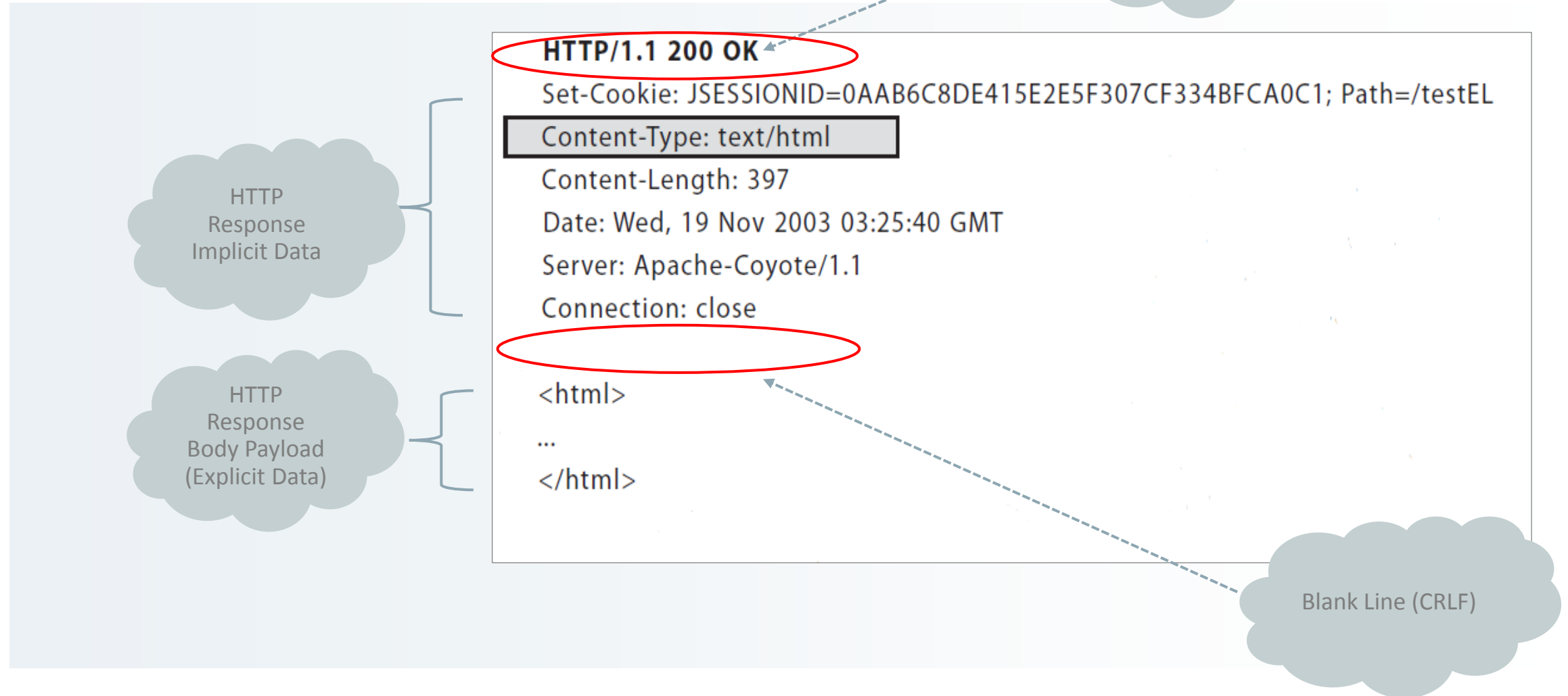
Connection: keep-alive

HTTP Request Implicit Data

Blank Line (CRLF)

A Typical HTTP Response

Review of HTTP Response



Setting Arbitrary Response Headers

Response Method	Description
setHeader(String headerName, String headerValue)	<ul style="list-style-type: none">• Sets arbitrary header.
setDateHeader(String name, long millisecs)	<ul style="list-style-type: none">• Converts milliseconds to date string in GMT format. <code>long date = new Date().getTime();</code> <code>response.setDateHeader("Date", date);</code>
setIntHeader(String name, int headerValue)	<ul style="list-style-type: none">• Prevents the need to convert int to String before calling setHeader().
addHeader(String, String), addDateHeader(String, String) addIntHeader(String, String)	<ul style="list-style-type: none">• Adds a new occurrence of header instead of replacing, not replacing the old occurrence.

Setting Common Response Headers

Response Method	Description
setContentTypes()	<ul style="list-style-type: none">• Sets the Content-Type header.• Servlets almost always use this.• Refer to table of common MIME types. <code>response.setContentType("text/html");</code>
setContentLength()	<ul style="list-style-type: none">• Sets the Content-Length header (ie. entity body length). <code>int size = getFile().length;</code> <code>response.setContentLength(size);</code>
addCookie()	<ul style="list-style-type: none">• Adds a value to the Set-Cookie header. <code>Cookie cookie = new Cookie("user", "john");</code> <code>cookie.setMaxAge(60*60);</code> <code>response.addCookie(cookie);</code>
sendRedirect()	<ul style="list-style-type: none">• Sets the Location header (plus changes status code). <code>sendRedirect("/my/Servlet");</code>

Common MIME Types

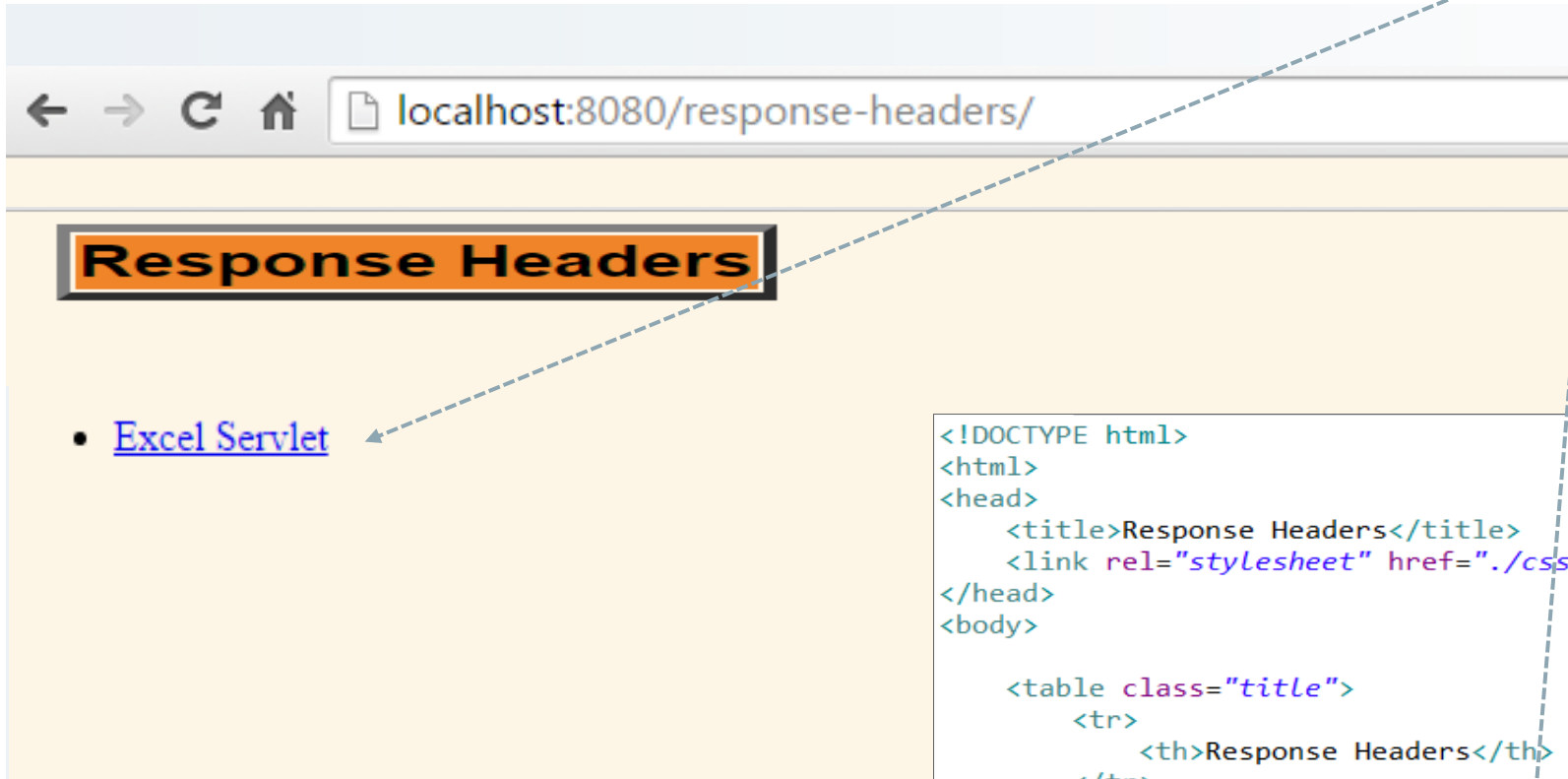
MIME Type Summary

Type	Meaning
application/msword	Microsoft Word document
application/octet-stream	Unrecognized or binary data
application/pdf	Acrobat (.pdf) file
application/postscript	PostScript file
application/vnd.ms-excel	Excel spreadsheet
application/vnd.ms-powerpoint	Powerpoint presentation
application/x-gzip	Gzip archive
application/x-java-archive	JAR file
application/x-java-vm	Java bytecode (.class) file
application/zip	Zip archive
audio/basic	Sound file in .au or .snd format
audio/x-aiff	AIFF sound file
audio/x-wav	Microsoft Windows sound file
audio/midi	MIDI sound file
text/css	HTML cascading style sheet
text/html	HTML document
text/plain	Plain text
text/xml	XML document
image/gif	GIF image
image/jpeg	JPEG image
image/png	PNG image
image/tiff	TIFF image
video/mpeg	MPEG video clip
video/quicktime	QuickTime video clip

Example: Building a Excel Spreadsheet

Building Excel Spreadsheets

Example



Link downloads creates and downloads excel spreadsheet.

```
<!DOCTYPE html>
<html>
<head>
  <title>Response Headers</title>
  <link rel="stylesheet" href="./css/styles.css" type="text/css" />
</head>
<body>

  <table class="title">
    <tr>
      <th>Response Headers</th>
    </tr>
  </table>

  <ul>
    <li><a href="apples-and-oranges">Excel Servlet</a></li>
  </ul>

</body>
</html>
```

Building Excel Spreadsheets

Example

```
import java.io.IOException;

@WebServlet("/apples-and-oranges")
public class ApplesAndOranges extends HttpServlet {

    private static final long serialVersionUID = 1L;

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        response.setContentType("application/vnd.ms-excel");
        PrintWriter out = response.getWriter();

        out.println("\tQ1\tQ2\tQ3\tQ4\tTotal");
        out.println("Apples\t78\t87\t92\t29\tSUM(B2:E2)");
        out.println("Oranges\t77\t86\t93\t30\tSUM(B3:E3)");

    }
}
```

Setting excel content-type:
application/vnd.ms-excel. Downloads file with open with .xls file extension.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Q1	Q2	Q3	Q4	Total							
2	Apples	78	87	92	29	286							
3	Oranges	77	86	93	30	286							
4													
5													
6													

Caching Web Pages

What is Caching?

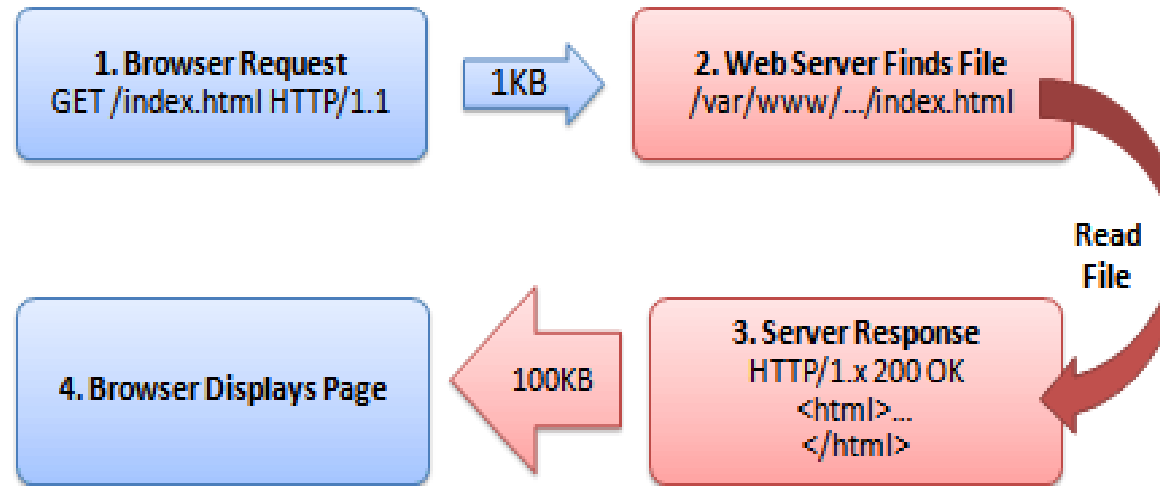
Understanding the purpose of caching.

Term	Definition
Caching (Web Cache)	<ul style="list-style-type: none">• A web cache (or HTTP cache) is an information technology for temporary storage (caching) of web documents, such as HTML pages and images, to reduce bandwidth usage, server load and perceived lag.• In the case of websites, the browser can save a copy of images, stylesheets, JavaScript or even an entire page. The next time then user needs that resource (such as a script or logo), the browser doesn't have to download it again.

HTTP Request/Response Refresher

Review of HTTP Request/Response

HTTP Request and Response

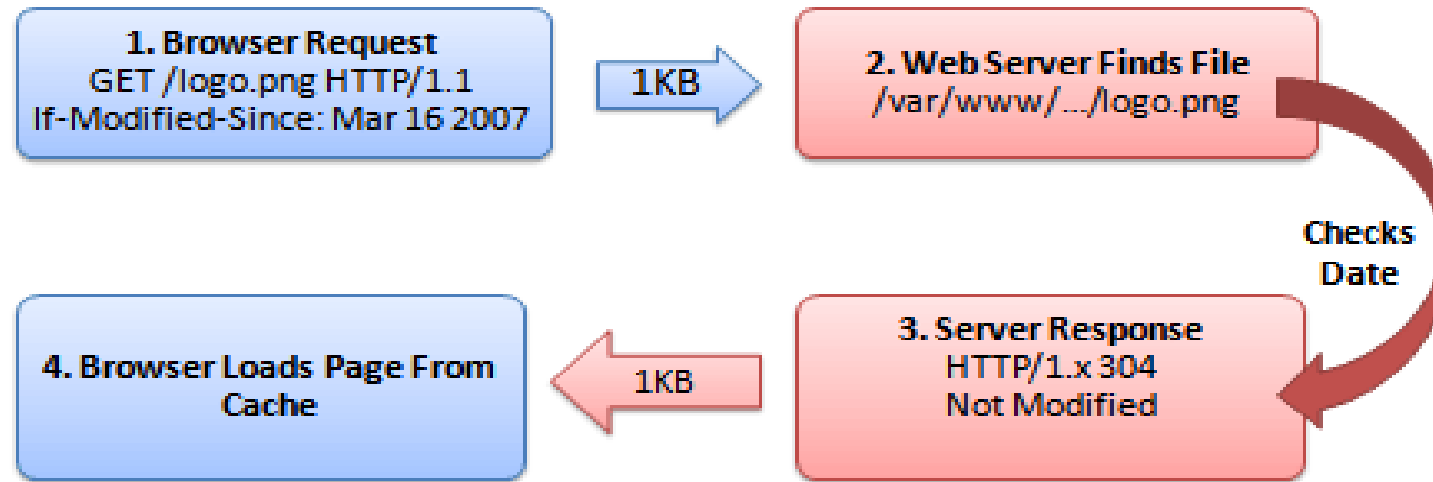


1. Client: Browser request page (index.html)
2. Server: Will look up the resources.
3. Server: Finds resources, returns resources
4. Client: Downloads the page.

Caching Background

HTTP Cache: Last-Modified

HTTP Cache: Last-Modified

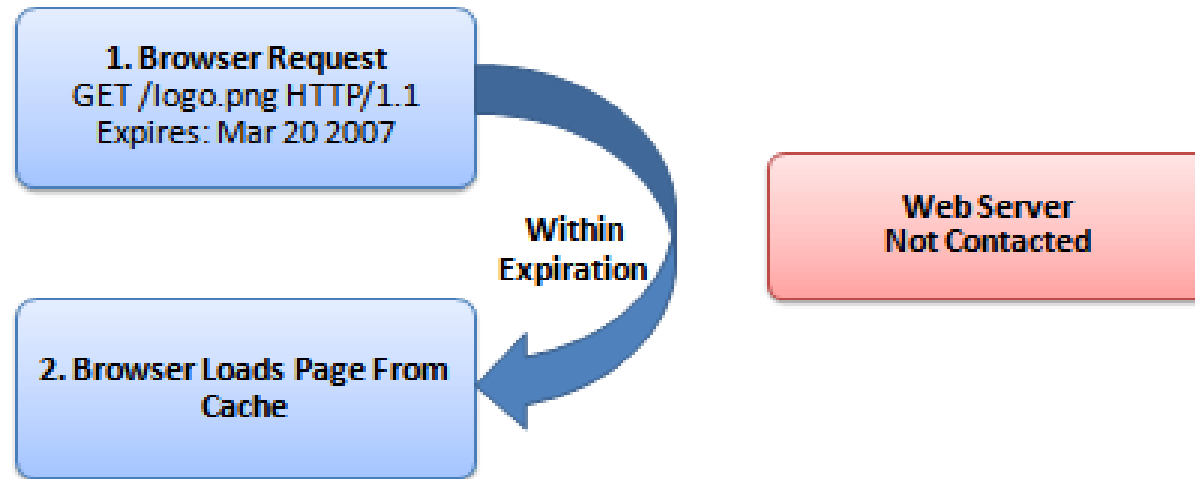


1. **Client**: Browser request file (logo) but only if its been modified since Mar 16th, 2007.
2. **Server**: Checking the modification date.
3. **Server**: Determines the file was **not modified** since that date. Client has the latest version of that file.
4. **Client**: Browser, continues to show that cached version.

Java Servlet: Set Expires Header with Cache

Example HTTP Cache: Expires

HTTP Cache: Expires



1. **Client:** There is no conversation here, this is a complete client/browser monologue.
2. **Client:** Checks, the expiration date on its own (Mar 20th, 2007).
3. **Client:** Verified, show cached data.
4. **Client:** Complete, the server does nothing, the user see cached file instantly.

Cache “How-To”

Java Example: Setting Cache Response Headers

Caching

```
final int CACHE_DURATION_IN_SECOND = 60 * 60 * 24 * 2; // --> Two Days
final long CACHE_DURATION_IN_MS = CACHE_DURATION_IN_SECOND * 1000;

long now = System.currentTimeMillis();

response.addHeader("Cache-Control", "max-age=" + CACHE_DURATION_IN_SECOND);
response.setDateHeader("Last-Modified", now);
response.setDateHeader("Expires", now + CACHE_DURATION_IN_MS);
```

No Caching

```
response.setHeader("Pragma", "No-cache");
response.setHeader("Cache-Control", "no-cache, no-store, max-age=0");
response.setDateHeader("Expires", 1);
```

Using Servlet to Create JPEG Images

Using Servlets to Generate JPEG Images

Steps	Description
STEP 1	<ul style="list-style-type: none">Create a BufferedImage
STEP 2	<ul style="list-style-type: none">Draw into the BufferedImage/ read image into BufferedImage
STEP 3	<ul style="list-style-type: none">Set the content-type of response header. response.setContentType("image/jpeg");
STEP 4	<ul style="list-style-type: none">Get an output stream OutputStream out = response.getOutputStream()
STEP 5	<ul style="list-style-type: none">Send the BufferedImage in JPEG format to the output stream <pre>try { ImageIO.write(image, "jpg", out); } catch(IOException ioe){ System.err.println("Error writing JPEG file: " + ioe); }</pre>

Image Servlet

Return an image from Servlet using ImageIO

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.io.OutputStream;

import javax.imageio.ImageIO;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/image-servlet")
public class ImageServlet extends HttpServlet {

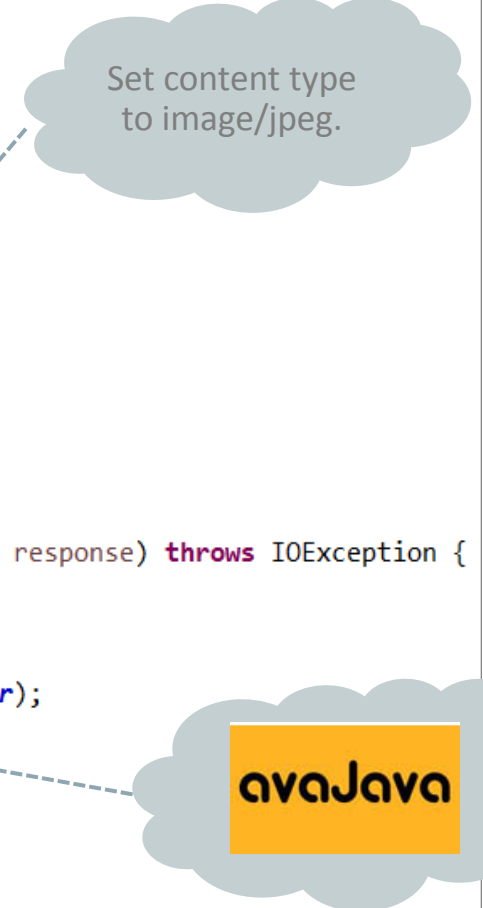
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {

        response.setContentType("image/jpeg");

        String pathToWeb = getServletContext().getRealPath(File.separator);
        File f = new File(pathToWeb + "avajava.jpg");

        BufferedImage bi = ImageIO.read(f);
        OutputStream out = response.getOutputStream();
        ImageIO.write(bi, "jpg", out);
        out.close();
    }
}
```



Set content type to image/jpeg.

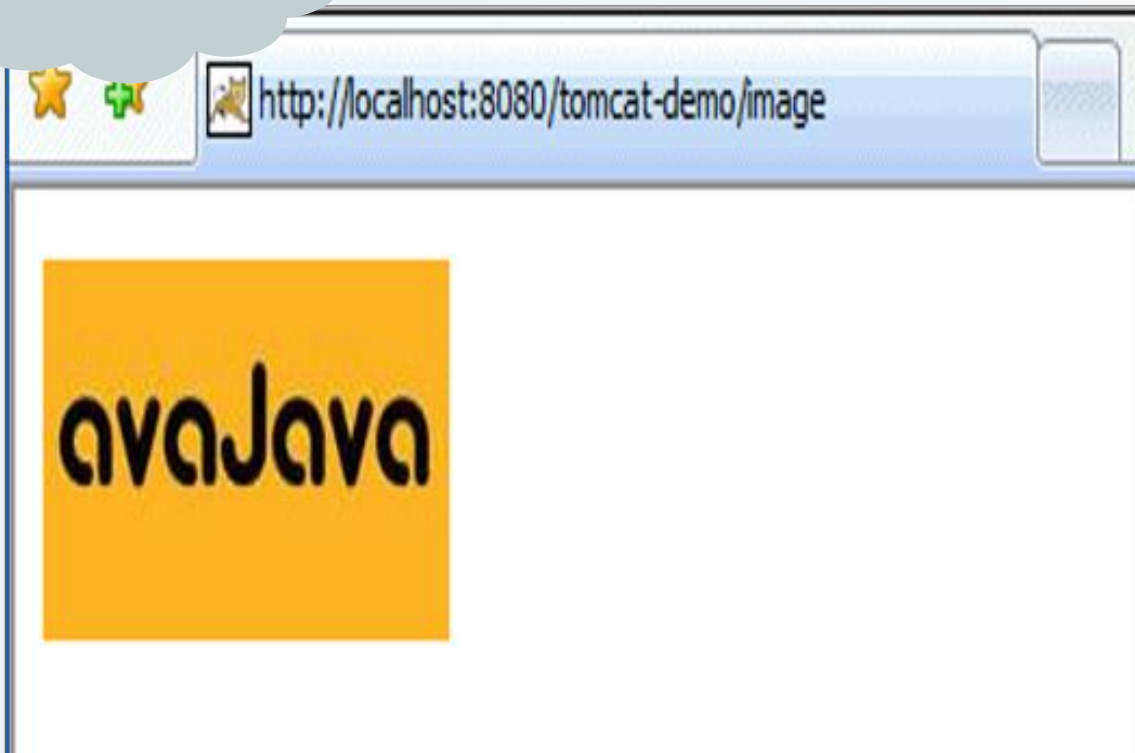
avaJava

1. Create Buffered Image
2. Create OutputStream
3. Send BufferedImage into OutputStream

Image Servlet

Result using servlet to return JPEG

Result



Environment

- response-headers
 - .settings
 - build
 - src
 - WebContent
 - css
 - META-INF
 - WEB-INF
 - avalogo.jpg
 - index.html

Questions?