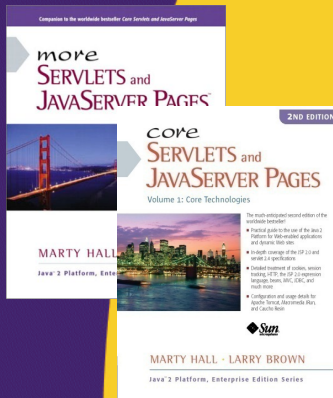




# Generating the Server Response: HTTP Response Headers

Originals of Slides and Source Code for Examples:  
<http://courses.coreservlets.com/Course-Materials/csajsp2.html>

**Customized Java EE Training:** <http://courses.coreservlets.com/>  
Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



**For live Java EE training, please see training courses at <http://courses.coreservlets.com/>.**

**JSF 2, PrimeFaces, Servlets, JSP, Ajax (with jQuery), GWT, Android development, Java 6 and 7 programming, SOAP-based and RESTful Web Services, Spring, Hibernate/JPA, XML, Hadoop, and customized combinations of topics.**



**Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization. Contact [hall@coreservlets.com](mailto:hall@coreservlets.com) for details.**

# Agenda

- Format of the HTTP response
- Setting response headers
- Understanding what response headers are good for
- Building Excel spread sheets
- Generating JPEG images dynamically
- Sending incremental updates to the browser

4

# HTTP Request/Response

- Request

```
GET /servlet/SomeName HTTP/1.1
Host: ...
Header2: ...
...
HeaderN:
(Blank Line)
```

- Response

```
HTTP/1.1 200 OK
Content-Type: text/html
Header2: ...
...
HeaderN: ...
(Blank Line)
<!DOCTYPE ...>
<HTML>
<HEAD>...</HEAD>
<BODY>
...
</BODY></HTML>
```

5

## Setting Arbitrary Response Headers

- **response.setHeader(String headerName, String headerValue)**
  - Sets an arbitrary header
- **response.setDateHeader(String name, long millisecs)**
  - Converts milliseconds since 1970 to a date string in GMT format
- **response.setIntHeader(String name, int headerValue)**
  - Prevents need to convert int to String before calling setHeader
- **addHeader, addDateHeader, addIntHeader**
  - Adds new occurrence of header instead of replacing

6

## Setting Common Response Headers

- **setContentTypes**
  - Sets the Content-Type header.
  - Servlets almost always use this.
  - See table of common MIME types.
- **setContentLength**
  - Sets the Content-Length header.
  - Used for persistent HTTP connections.
  - See Connection request header.
- **addCookie**
  - Adds a value to the Set-Cookie header.
  - See separate section on cookies.
- **sendRedirect**
  - Sets the Location header (plus changes status code).

7

# Common MIME Types

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

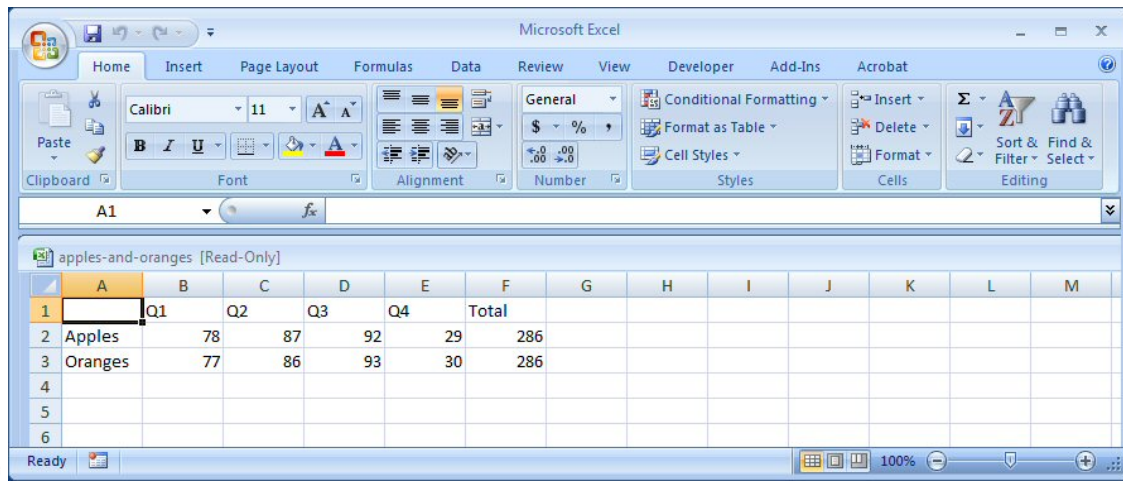
8

# Building Excel Spreadsheets

```
@WebServlet("/apples-and-oranges")
public class ApplesAndOranges extends HttpServlet {
    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\t=SUM(B2:E2)");
        out.println
            ("Oranges\t77\t86\t93\t30\t=SUM(B3:E3)");
    }
}
```

9

# Building Excel Spreadsheets



10

# Common HTTP 1.1 Response Headers

- **Cache-Control (1.1) and Pragma (1.0)**
  - A no-cache value prevents browsers from caching page.
- **Content-Disposition**
  - Lets you request that the browser ask the user to save the response to disk in a file of the given name
    - `Content-Disposition: attachment; filename=file-name`
- **Content-Encoding**
  - The way document is encoded. See earlier compression example
- **Content-Length**
  - The number of bytes in the response.
  - See `setContentLength` on previous slide.
  - Use `ByteArrayOutputStream` to buffer document before sending it, so that you can determine size. See discussion of the Connection request header

11

## Common HTTP 1.1 Response Headers (Continued)

- **Content-Type**
  - The MIME type of the document being returned.
  - Use `setContentTypes` to set this header.
- **Expires**
  - The time at which document should be considered out-of-date and thus should no longer be cached.
  - Use `setDateHeader` to set this header.
- **Last-Modified**
  - The time document was last changed.
  - Don't set this header explicitly; provide a `getLastModified` method instead. See lottery number example in book (Chapter 3).

12

## Common HTTP 1.1 Response Headers (Continued)

- **Location**
  - The URL to which browser should reconnect.
  - Use `sendRedirect` instead of setting this directly.
- **Refresh**
  - The number of seconds until browser should reload page. Can also include URL to connect to. See following example.
- **Set-Cookie**
  - The cookies that browser should remember. Don't set this header directly; use `addCookie` instead. See next section.
- **WWW-Authenticate**
  - The authorization type and realm needed in Authorization header. See security chapters in *More Servlets & JSP*.

13

## Requirements for Handling Long-Running Servlets

- **A way to store data between requests.**
  - For data that is not specific to any one client, store it in a field (instance variable) of the servlet.
  - For data that is specific to a user, store it in the HttpSession object
    - See upcoming lecture on session tracking
  - For data that needs to be available to other servlets or JSP pages (regardless of user), store it in the ServletContext
- **A way to keep computations running after the response is sent to the user.**
  - This task is simple: start a Thread. The only subtlety: set the thread priority to a low value so that you do not slow down the server.
- **A way to get the updated results to the browser when they are ready.**
  - Use Refresh header to tell browser to ask for updates

14

## Persistent Servlet State and Auto-Reloading Pages: Example

- **Idea: generate list of large (e.g., 150-digit) prime numbers**
  - Show partial results until completed
  - Let new clients make use of results from others
- **Demonstrates use of the Refresh header.**
- **Shows how easy it is for servlets to maintain state between requests.**
  - Very difficult in traditional CGI.
- **Also illustrates that servlets can handle multiple simultaneous connections**
  - Each request is in a separate thread.

15

# Finding Prime Numbers for Use with Public Key Cryptography

```
@WebServlet("/prime-numbers")
public class PrimeNumberServlet extends HttpServlet {
    private List<PrimeList> primeListCollection =
        new ArrayList<PrimeList>();
    private int maxPrimeLists = 30;

    public void doGet(HttpServletRequest request,
                     HttpServletResponse response)
        throws ServletException, IOException {
        int numPrimes =
            ServletUtilities.getIntParameter(request,
                                           "numPrimes", 50);

        int numDigits =
            ServletUtilities.getIntParameter(request,
                                           "numDigits", 120);

        PrimeList primeList =
            findPrimeList(primeListCollection,
                         numPrimes, numDigits);
    }
}
```

16

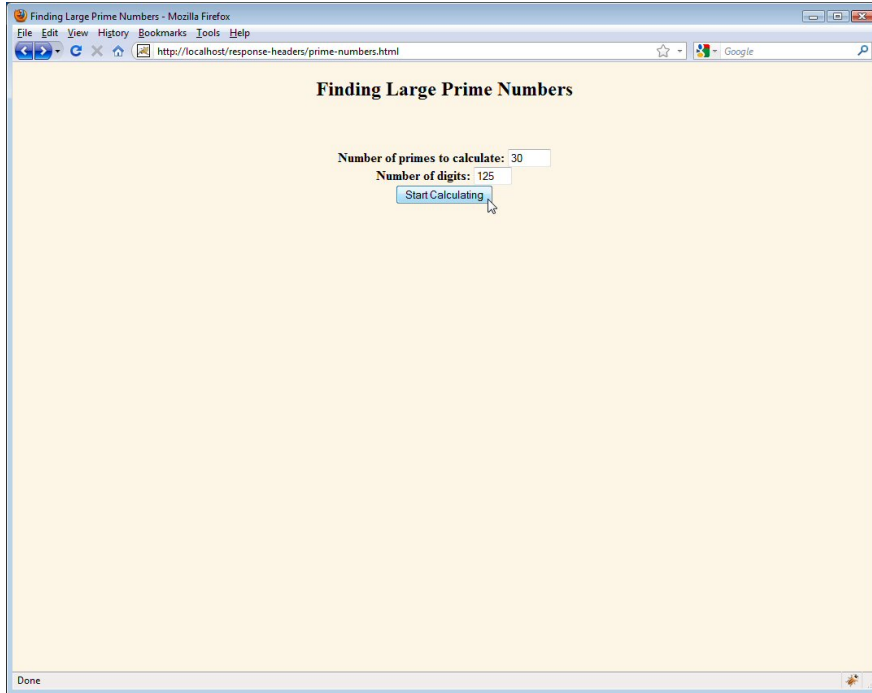
# Finding Prime Numbers for Use with Public Key Cryptography

```
if (primeList == null) {
    primeList = new PrimeList(numPrimes, numDigits, true);
    synchronized(primeListCollection) {
        if (primeListCollection.size() >= maxPrimeLists)
            primeListCollection.remove(0);
        primeListCollection.add(primeList);
    }
}
List<BigInteger> currentPrimes =
    primeList.getPrimes();
int numCurrentPrimes = currentPrimes.size();
int numPrimesRemaining = (numPrimes - numCurrentPrimes);
boolean isLastResult = (numPrimesRemaining == 0);
if (!isLastResult) {
    response.setIntHeader("Refresh", 5);
}
...
```

17

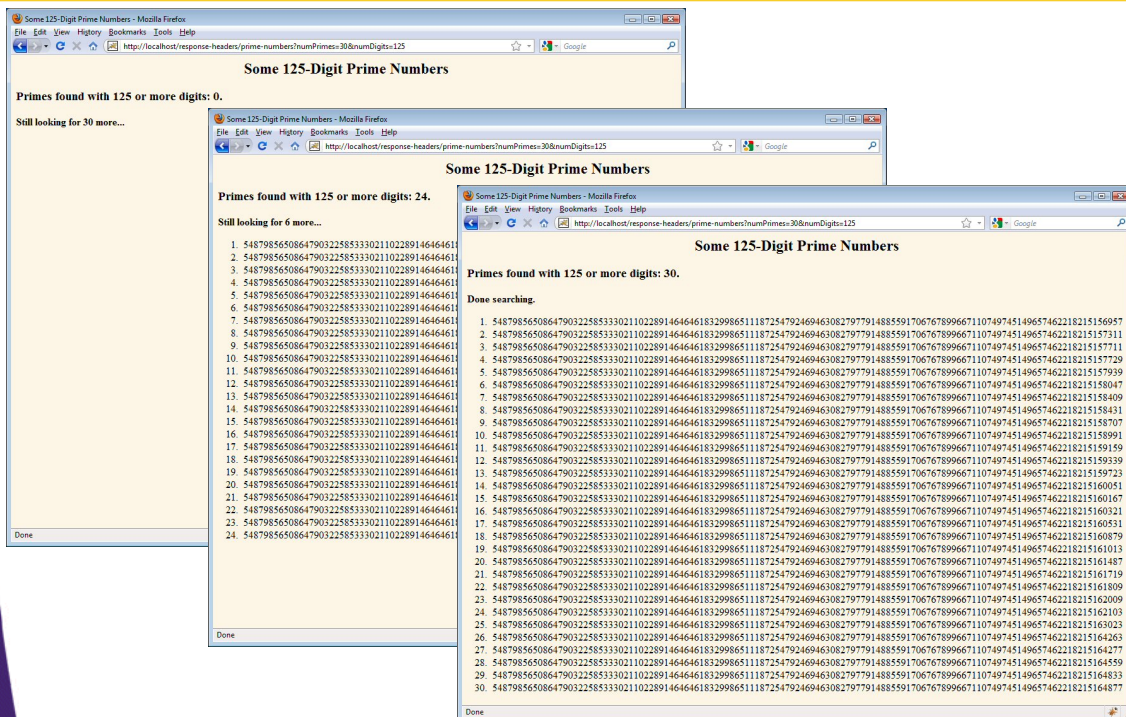


# Finding Prime Numbers for Use with Public Key Cryptography



18

# Finding Prime Numbers for Use with Public Key Cryptography



19

# Using Servlets to Generate JPEG Images

1. Create a BufferedImage
2. Draw into the BufferedImage
  - Use normal AWT or Java 2D drawing methods
3. Set the Content-Type response header  
`response.setContentType("image/jpeg");`
4. Get an output stream  
`OutputStream out = response.getOutputStream`
5. Send the BufferedImage in JPEG format to the output stream

```
try {
    ImageIO.write(image, "jpg", out);
} catch (IOException ioe) {
    System.err.println("Error writing JPEG file: "
        + ioe);
}
```

20

# Using Servlets to Generate JPEG Images

**JPEG Generation Service**

Welcome to the *free* trial edition of our JPEG generation service. Enter a message, a font name, and a font size below, then submit the form. You will be returned a JPEG image showing the message in the designated font, with an oblique "shadow" of the message behind it. Once you get an image you are satisfied with, right-click on it (or click while holding down the SHIFT key) to save it to your local disk.

The server is currently on Windows, so the font name must be either a standard Java font name (e.g., Serif, SansSerif, or Monospaced) or a Windows font name (e.g., Arial Black). Unrecognized font names will revert to Serif. Press the "Show Font List" button for a complete list.

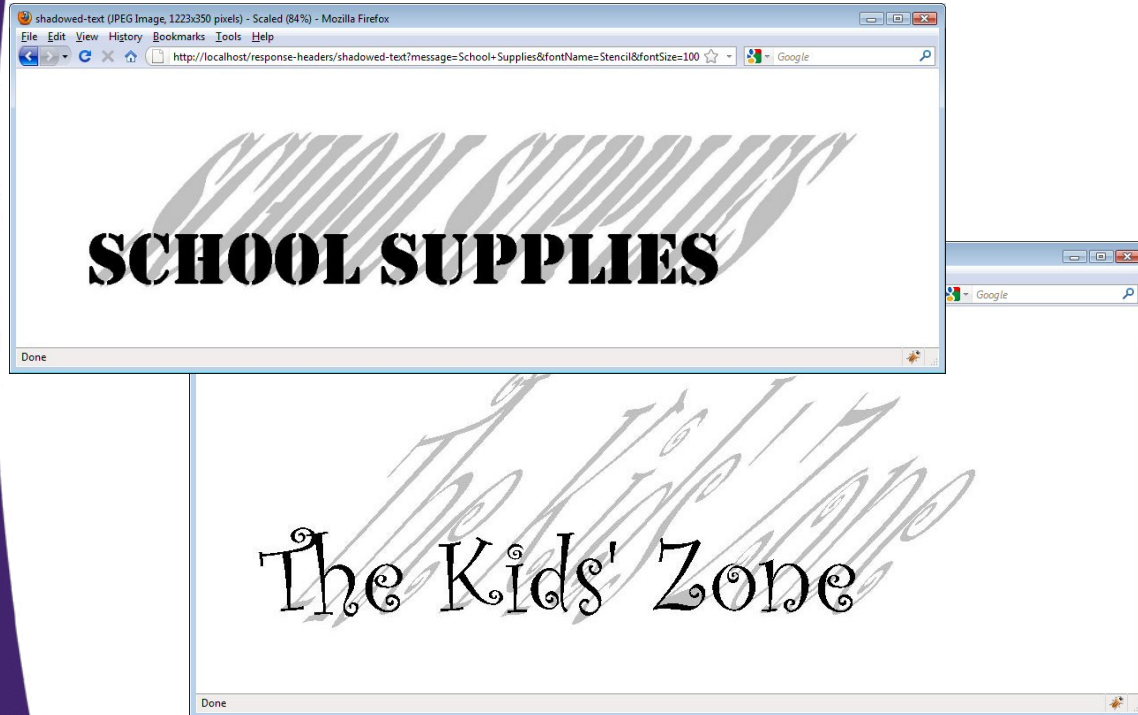
Message:   
Font name:   
Font size:

Done



21

# Using Servlets to Generate JPEG Images



## Summary

- **HTTP is important**
    - Many servlet tasks can *only* be accomplished through use of HTTP response headers
  - **Setting response headers**
    - In general, set with `response.setHeader`
    - In special cases, set with `response.setContentType`, `response.setContentLength`, `response.addCookie`, and `response.sendRedirect`
  - **Most important response headers you set directly**
    - Cache-Control and Pragma
    - Content-Disposition
    - Content-Encoding
    - Content-Length
    - Expires
    - Refresh
    - WWW-Authenticate
- 23



# Questions?

[JSF 2, PrimeFaces, Java 7, Ajax, jQuery, Hadoop, RESTful Web Services, Android, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training](#)

**Customized Java EE Training: <http://courses.coreservlets.com/>**

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.