

Core Servlets and JavaServer Pages / 2e
Volume 1: Core Technologies
Marty Hall • Larry Brown

Generating the Server Response: HTTP Response Headers

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**

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>
```

Setting Arbitrary Response Headers

- **The most general way to specify headers is to use the `setHeader` method of the `HttpServletResponse` class**
 - `public void setHeader(String headerName, String headerValue)`
- **Two specialized methods set headers with dates and integers**
 - `public void setDateHeader(String name, long millisecs)`
 - Converts milliseconds since 1970 to a date string in GMT format
 - `public void 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

Setting Common Response Headers

- **setContentType (String mimeType)**
 - Sets the Content-Type header (MIME types)
- **setContentLength (int length)**
 - Sets the Content-Length header (number of bytes in the response), which is useful if the browser supports persistent HTTP connections.
- **addCookie (Cookie c)**
 - Adds a value to the Set-Cookie header.
- **sendRedirect (String address)**
 - Sets the Location header (plus changes status code).

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
- **Content-Length**
 - The number of bytes in the response.
 - Use `ByteArrayOutputStream` to buffer document before sending it, so that you can determine size.

Common HTTP 1.1 Response Headers (Continued)

- **Content-Type**

- The MIME type of the document being returned.

- **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.
- Use the getLastModified method instead

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.

- **Set-Cookie**

- The cookies that browser should remember. Don't set this header directly; use `addCookie` instead.

Common MIME Types

Type

application/msword
application/octet-stream
application/pdf
application/postscript
application/vnd.ms-excel
application/vnd.ms-powerpoint
application/x-gzip
application/x-java-archive
application/x-java-vm
application/zip
audio/basic
audio/x-aiff
audio/x-wav
audio/midi
text/css
text/html
text/plain
text/xml
image/gif
image/jpeg
image/png
image/tiff
video/mpeg
video/quicktime

Meaning

Microsoft Word document
Unrecognized or binary data
Acrobat (.pdf) file
PostScript file
Excel spreadsheet
Powerpoint presentation
Gzip archive
JAR file
Java bytecode (.class) file
Zip archive
Sound file in .au or .snd format
AIFF sound file
Microsoft Windows sound file
MIDI sound file
HTML cascading style sheet
HTML document
Plain text
XML document
GIF image
JPEG image
PNG image
TIFF image
MPEG video clip
QuickTime video clip

Building Excel Spreadsheets

- Though servlets usually generate HTML output, other types of output are possible
- Microsoft Excel content can be generated so that the features of Excel can be exploited
- The key is to include the following code

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

- Example will generate output in tab-separated format (include `\t` in the output strings)

Building Excel Spreadsheets

```
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)");
    }
}
```

Building Excel Spreadsheets

Address: <http://localhost/servlet/coreservlets.ApplesAndOranges>

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

coreservlets.ApplesAndOranges

Unknown Zone

Requirements for Handling Long-Running Servlets

- **What to do if a calculation requires a long time to complete (20 seconds) or whose results change periodically**
- **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
 - For data that needs to be available to other servlets or JSP pages (regardless of user), store it in the ServletContext
- **Keep computations running after the response is sent to the user.**
 - Start a Thread but set the thread priority to a low value so that it does not slow down the server.

Requirements for Handling Long-Running Servlets

- **Send updated results to the browser when they are ready.**
 - Browser does not maintain an open connection to the server. Use Refresh header to tell browser to ask for updates

```
if (!isLastResult) {  
    response.setIntHeader("Refresh", 5);
```

Using Servlets to Generate JPEG Images

- 1. Create a BufferedImage**
- 2. Draw into the BufferedImage**
- 3. Set the Content-Type response header**
- 4. Get an output stream**
- 5. Send the BufferedImage in JPEG format to the output stream**

```
response.setContentType("image/jpeg");
```

```
OutputStream out = response.getOutputStream
```

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