Server-side Web Programming

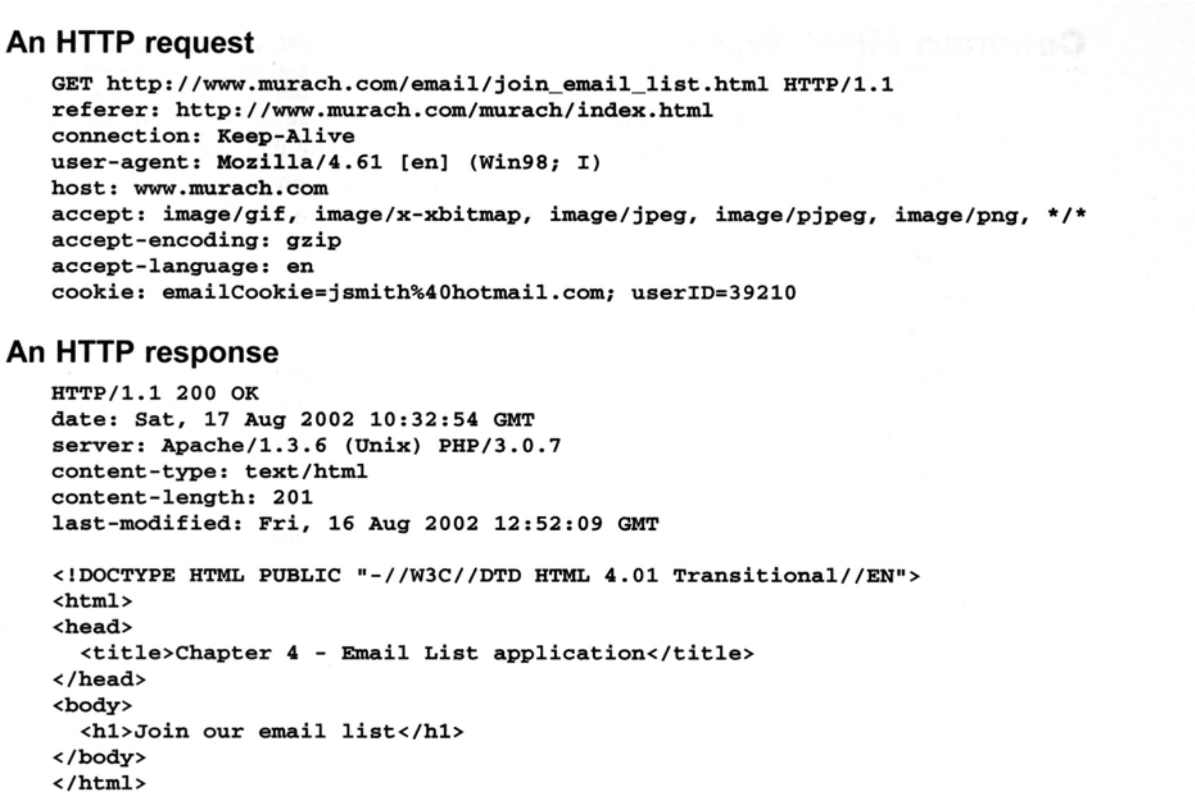
Lecture 15:

The Request and Response Objects

Http Requests and Responses

- **`request`** object
  - Properties of browser
  - IP address and host name of referring machine
    - **`request.getRemoteAddr()`**
    - **`request.getHost()`**
    - Not particularly useful for identification (too easy to fake)
- **`response`** object
  - Can be used to tell browser more than just html page to display
  - Format to display response page, etc.

Http Requests and Responses



An HTTP request

```
GET http://www.murach.com/email/join_email_list.html HTTP/1.1
referer: http://www.murach.com/murach/index.html
connection: Keep-Alive
user-agent: Mozilla/4.61 [en] (Win98; I)
host: www.murach.com
accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
accept-encoding: gzip
accept-language: en
cookie: emailCookie=jsmith%40hotmail.com; userID=39210
```

An HTTP response

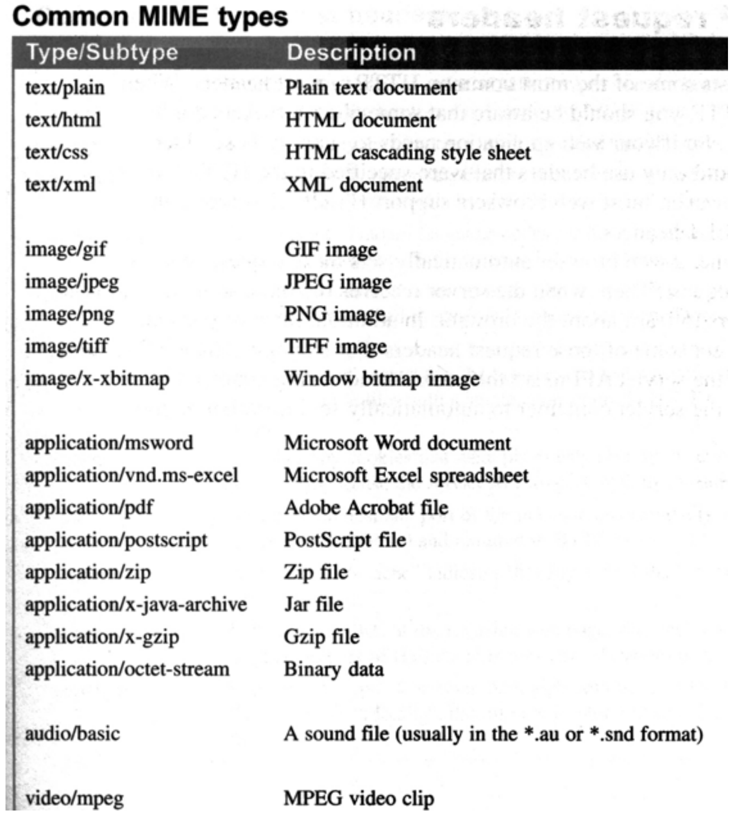
```
HTTP/1.1 200 OK
date: Sat, 17 Aug 2002 10:32:54 GMT
server: Apache/1.3.6 (Unix) PHP/3.0.7
content-type: text/html
content-length: 201
last-modified: Fri, 16 Aug 2002 12:52:09 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>Chapter 4 - Email List application</title>
</head>
<body>
  <h1>Join our email list</h1>
</body>
</html>
```

Requests

- Contains information about browser that submitted request
- Main components:
  - **Referrer**: Page from which request was submitted
  - **Accept**: Preferred order of MIME types accepted by browser
  - **Accept-Encoding**: Types of compression understood by browser
    - gzip, etc.
  - **Accept-Language**: Language codes for accepted languages
    - "**en**", "**en-us**", etc.
  - **User-Agent**: Browser type
    - Long string containing identifiers specific to browser
      - "**MSIE**", etc.

MIME Types

- Multipurpose Internet Mail Extensions: Formats for transmitting data via email / internet
  - Text formats
  - Image formats
  - Application formats (programs browser can run to display page)
  - Audio and video multimedia formats
- Can use ***/*** to indicate that accept anything (usually last resort)



Common MIME types

Type/Subtype	Description
text/plain	Plain text document
text/html	HTML document
text/css	HTML cascading style sheet
text/xml	XML document
image/gif	GIF image
image/jpeg	JPEG image
image/png	PNG image
image/tiff	TIFF image
image/x-xbitmap	Window bitmap image
application/msword	Microsoft Word document
application/vnd.ms-excel	Microsoft Excel spreadsheet
application/pdf	Adobe Acrobat file
application/postscript	PostScript file
application/zip	Zip file
application/x-java-archive	Jar file
application/x-gzip	Gzip file
application/octet-stream	Binary data
audio/basic	A sound file (usually in the *.au or *.snd format)
video/mpeg	MPEG video clip

Accessing Request Properties

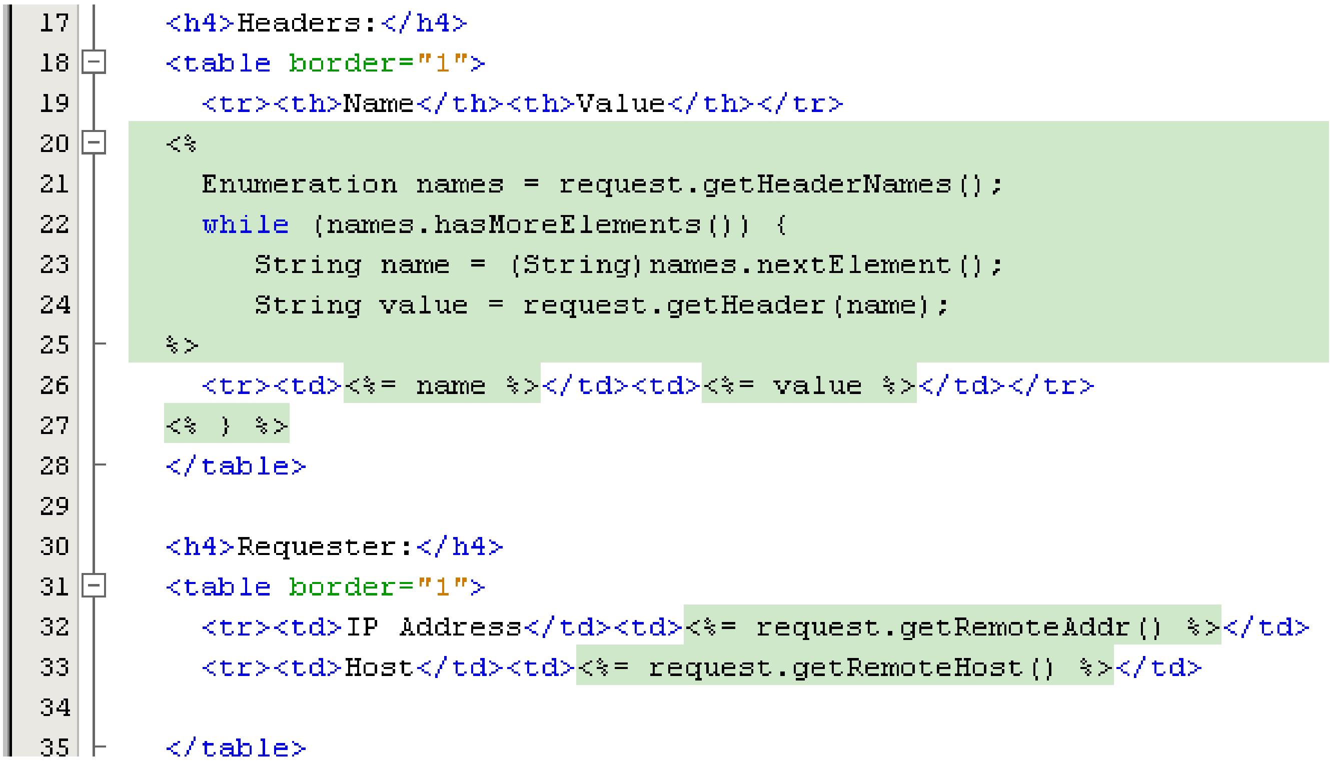
- Can get these properties using
  `request.getHeader(headername)`
- Example:

```
String browser =
        request.getHeader("Accept-Encoding");
```

  might return "**gzip, deflate**" for example

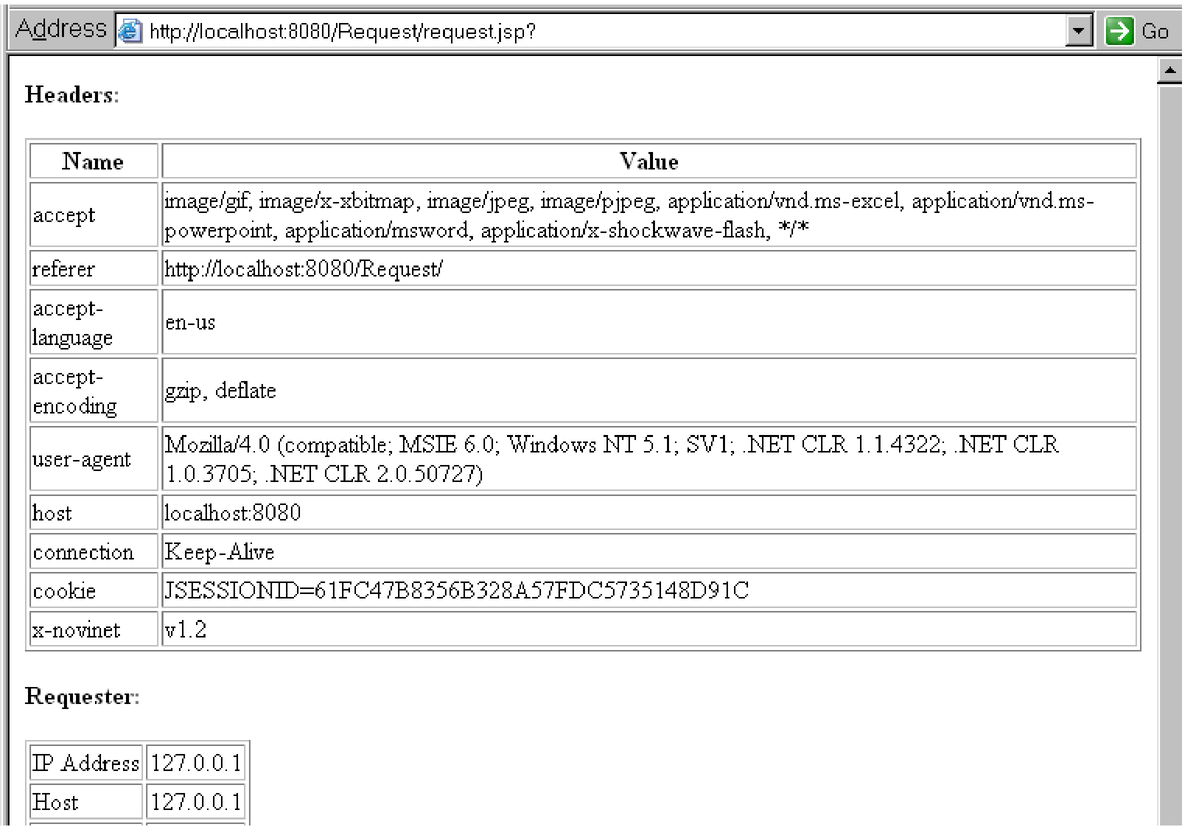
- Main use: Customizing response to abilities of browser
  - Only send information over if form browser can handle!
- Can use **request.getHeaderNames()** to get list of all property names sent over from browser

Accessing Request Properties



```
<h4>Headers:</h4>
<table border="1">
  <tr><th>Name</th><th>Value</th></tr>
<%
  Enumeration names = request.getHeaderNames();
  while (names.hasMoreElements()) {
     String name = (String)names.nextElement();
     String value = request.getHeader(name);
%>
  <tr><td><%= name %></td><td><%= value %></td></tr>
<% } %>
</table>

<h4>Requester:</h4>
<table border="1">
  <tr><td>IP Address</td><td><%= request.getRemoteAddr() %></td>
  <tr><td>Host</td><td><%= request.getRemoteHost() %></td>

</table>
```

Accessing Request Properties



Headers:

Name	Value
accept	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-flash, */*
referer	http://localhost:8080/Request/
accept-language	en-us
accept-encoding	gzip, deflate
user-agent	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 1.0.3705; .NET CLR 2.0.50727)
host	localhost:8080
connection	Keep-Alive
cookie	JSESSIONID=61FC47B8356B328A57FDC5735148D91C
x-novinet	v1.2

Requester:

IP Address	127.0.0.1
Host	127.0.0.1

Using Request Properties

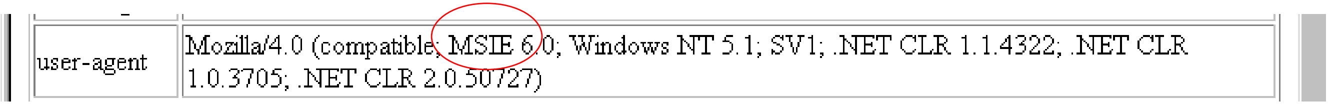
- Example: Sending custom image types
  - Send **.png** image if supported
  - Send **.jpg** image otherwise

```
String imagetypes = request.getHeader(“Accept”);
boolean acceptsPng = imagetypes.contains(“PNG”);
if (acceptsPng) {
   // insert link to .png image
   }
else {
   // insert link to .jpg image
}
```

Search method for strings

Using Request Properties

- Example: Customizing response to browser type
  - Will contain the string “**MSIE**” if Internet Explorer used

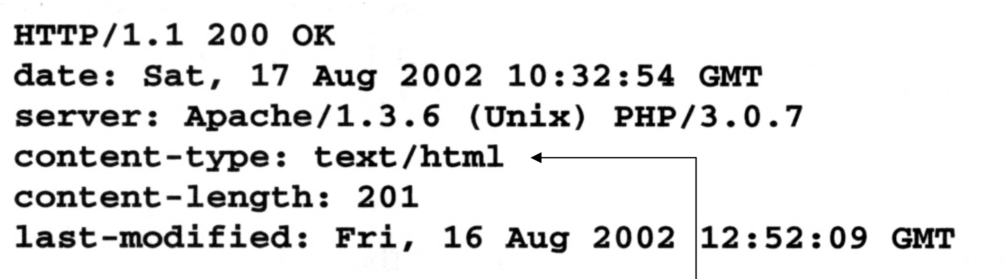


user-agent	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 1.0.3705; .NET CLR 2.0.50727)

```
String browser = request.getHeader(“User-Agent”);
boolean isIE = browser.contains(“MSIE”);
if (isIE) {
   // forward to IE specific page
   }
else {
   // forward to general response page
}
```

Response Properties

- Can set properties of response



```
HTTP/1.1 200 OK
date: Sat, 17 Aug 2002 10:32:54 GMT
server: Apache/1.3.6 (Unix) PHP/3.0.7
content-type: text/html
content-length: 201
last-modified: Fri, 16 Aug 2002 12:52:09 GMT
```

- Useful type to set: Content type
  - Form in which browser should display information sent
  - Default: **`text/html`** (standard html format)
  - Should first examine request to make sure that form is supported!

Setting Content Type

- Syntax: **`response.setContentType(“`*`MIME type`*`”);`**

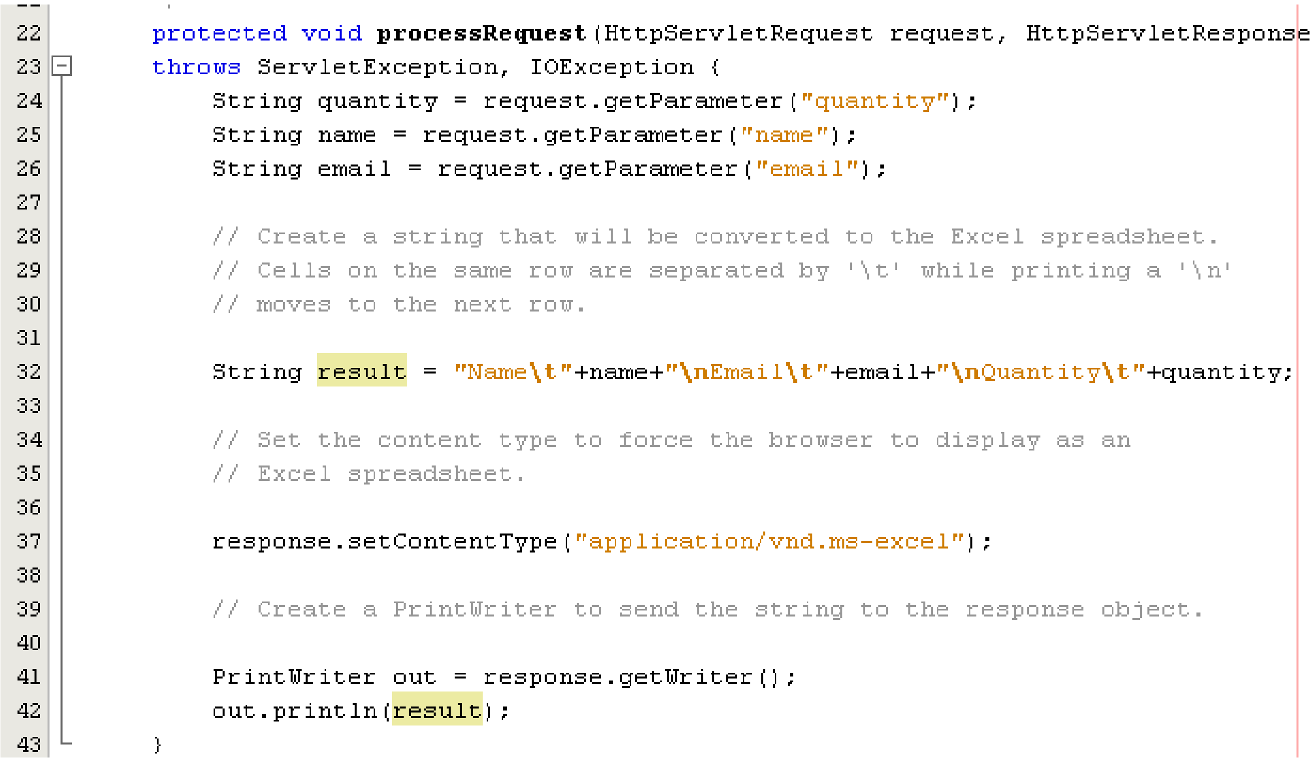
- Example: forcing browser to display response as Excel spreadsheet

  - **`response.setContentType(“application/vnd.ms-excel”);`**

  - Send response back in simple format:
    - Cells in same row separated by tab **`‘\t’`**
    - Move to next row with return **`‘\n’`**
  - Write that string to **`response`** object using **`PrintWriter`** (like old style response page)

  - Much more efficient than sending an entire spreadsheet as file!

Setting Content Type



```
protected void processRequest(HttpServletRequest request, HttpServletResponse
throws ServletException, IOException {
    String quantity = request.getParameter("quantity");
    String name = request.getParameter("name");
    String email = request.getParameter("email");

    // Create a string that will be converted to the Excel spreadsheet.
    // Cells on the same row are separated by '\t' while printing a '\n'
    // moves to the next row.

    String result = "Name\t"+name+"\nEmail\t"+email+"\nQuantity\t"+quantity;

    // Set the content type to force the browser to display as an
    // Excel spreadsheet.

    response.setContentType("application/vnd.ms-excel");

    // Create a PrintWriter to send the string to the response object.

    PrintWriter out = response.getWriter();
    out.println(result);
}
```

Controlling Caching

- For efficiency, most browsers <u>cache</u> pages received from server
  - Stored in local memory
- Next time user requests page, check to see whether in cache before downloading again
- Problem for pages that <u>change</u> regularly
  - Stock price pages, etc.

- Can <u>force</u> browser to <u>remove</u> page after certain interval of time
  - Browser will then download <u>current</u> version of page
- Syntax:
  `response.setHeader("cache-control", "no-cache");`

Forcing Page Refresh

- Can <u>force</u> browser to <u>refresh</u> page after certain interval of time
  - Gamecasts, etc.

- Syntax:
  response.setIntHeader(“refresh”, *time in seconds*);

  Time after which browser
  refreshes page

- Example:
  response.setIntHeader(“refresh”, 60);