Generating the Server Response: HTTP Status Codes
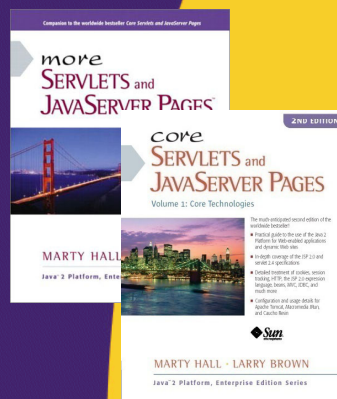
Originals of Slides and Source Code for Examples:
http://courses.coreservlets.com/Course-Materials/csajsp2.html

2

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

Servlets, JSP, Struts, JSF 1.*x*, JSF 2.0, Ajax (with jQuery, Dojo, Prototype, Ext-JS, Google Closure, etc.), GWT 2.0 (with GXT), Java 5, Java 6, SOAP-based and RESTful Web Services, Spring, Hibernate/JPA, and customized combinations of topics.

Agenda

- **Format of the HTTP response**
- **How to set status codes**
- **What the status codes are good for**
- **Shortcut methods for redirection and error pages**
- **A servlet that redirects users to browser-specific pages**
- **A front end to various search engines**

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 Status Codes

- **response.setStatus(int statusCode)**
  - Use a constant for the code, not an explicit int. Constants are in HttpServletResponse
  - Names derived from standard message. E.g., SC_OK, SC_NOT_FOUND, etc.
- **response.sendError(int code, String message)**
  - Wraps message inside small HTML document
- **response.sendRedirect(String url)**
  - Sets status code to 302
  - Sets Location response header also

Common HTTP 1.1 Status Codes

- **200 (OK)**
  - Everything is fine; document follows.
  - Default for servlets.
- **204 (No Content)**
  - Browser should keep displaying previous document.
- **301 (Moved Permanently)**
  - Requested document permanently moved elsewhere (indicated in Location header).
  - Browsers go to new location automatically.
  - Browsers are technically supposed to follow 301 and 302 (next page) requests only when the incoming request is GET, but do it for POST with 303. Either way, the Location URL is retrieved with GET.

Common HTTP 1.1 Status Codes (Continued)

- **302 (Found)**
  - Requested document temporarily moved elsewhere (indicated in Location header).
  - Browsers go to new location automatically.
  - Servlets should use sendRedirect, not setStatus, when setting this header. See example.
- **401 (Unauthorized)**
  - Browser tried to access password-protected page without proper Authorization header.
- **404 (Not Found)**
  - No such page. Servlets should use sendError to set this.
  - Problem: Internet Explorer and small (< 512 bytes) error pages. IE ignores small error page by default.
  - Fun and games: http://www.plinko.net/404/

A Servlet That Redirects Users to Browser-Specific Pages

```
@WebServlet("/wrong-destination")
public class WrongDestination extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    String userAgent = request.getHeader("User-Agent");
    if ((userAgent != null) &&
        (userAgent.contains("MSIE")) {
      response.sendRedirect("http://home.mozilla.com");
    } else {
      response.sendRedirect("http://www.microsoft.com");
    }
  }
}
```

A Servlet That Redirects Users to Browser-Specific Pages

- **Original URL for** *both*
  - http://localhost/status-codes/wrong-destination

A Front End to Various Search Engines

```java
@WebServlet("/search-engines")
public class SearchEngines extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    String searchString =
      request.getParameter("searchString");
    if ((searchString == null) ||
        (searchString.length() == 0)) {
      reportProblem(response, "Missing search string");
      return;
    }
    searchString = URLEncoder.encode(searchString, "utf-8");
    String searchEngineName =
      request.getParameter("searchEngine");
    if ((searchEngineName == null) ||
        (searchEngineName.length() == 0)) {
      reportProblem(response, "Missing search engine name");
      return;
    }
```

A Front End to Various Search Engines (Continued)

```
      String searchURL =
        SearchUtilities.makeURL(searchEngineName,
                                searchString);
    if (searchURL != null) {
      response.sendRedirect(searchURL);
    } else {
      reportProblem(response,
                    "Unrecognized search engine");
    }
  }

  private void reportProblem(HttpServletResponse response,
                             String message)
      throws IOException {
    response.sendError(HttpServletResponse.SC_NOT_FOUND,
                       message);
  }
}
```

A Front End to Various Search Engines (Continued)

```
public class SearchSpec {

  /** Builds a URL for the results page by
   *  simply concatenating the base URL
   *  (http://...?someVar=") with the
   *  URL-encoded search string (jsp+training).
   */

  public String makeURL(String searchString) {
    return(baseURL + searchString);
  }
…
}
```
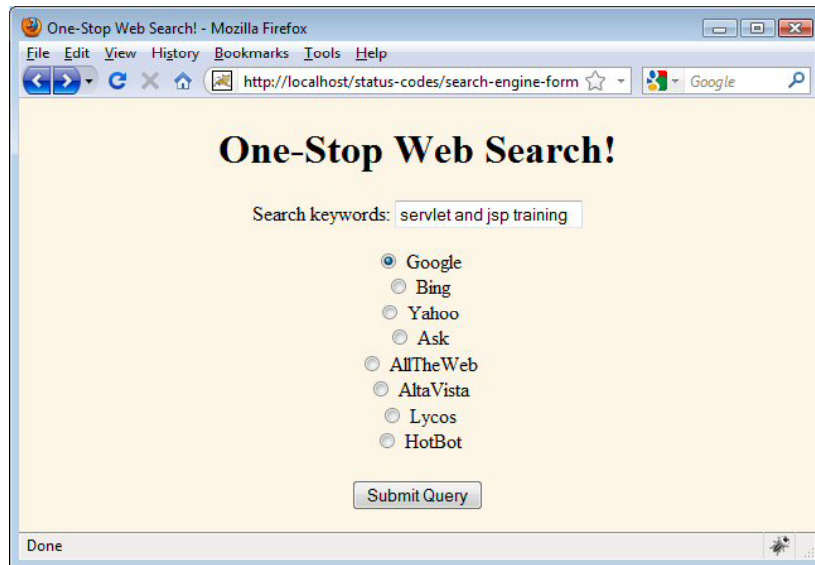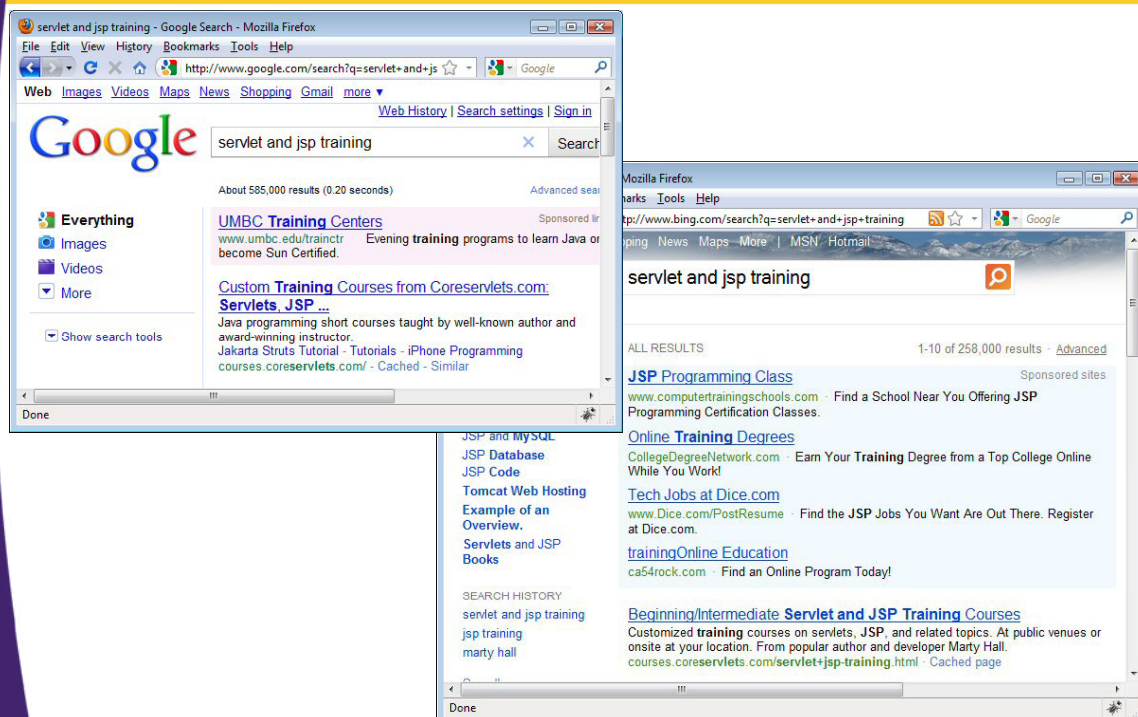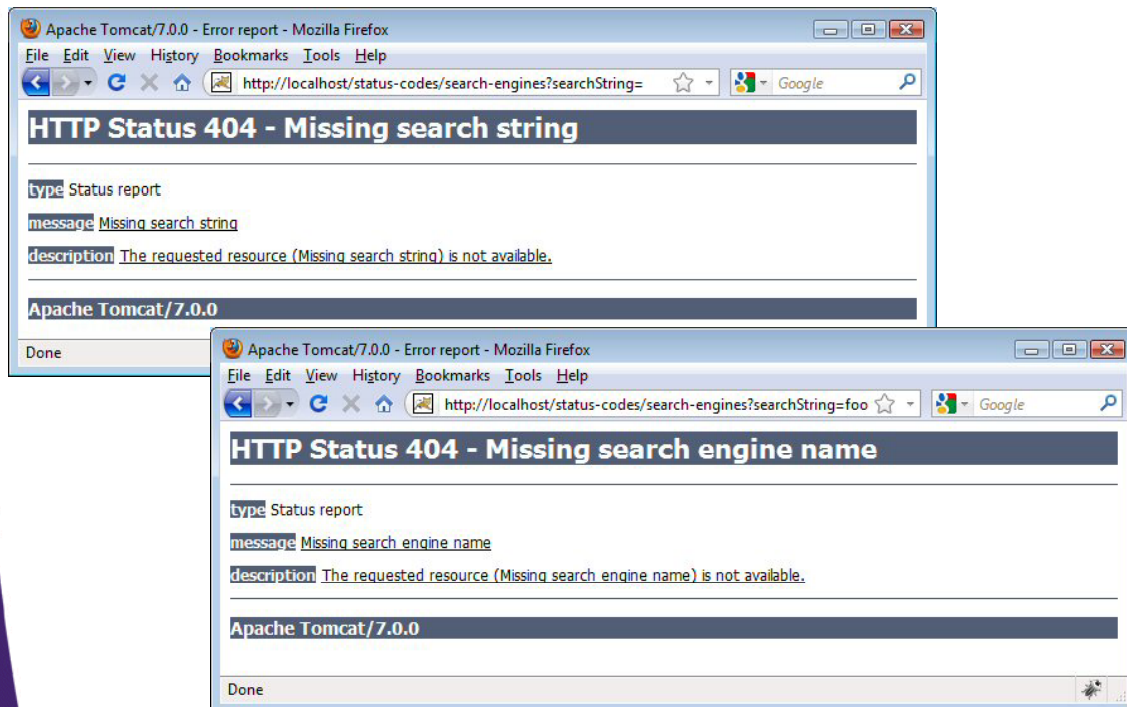
Front End to Search Engines: HTML Form

Front End to Search Engines: Result for Valid Data

Front End to Search Engines: Result for Invalid Data

Summary

- **Many servlet tasks can *only* be accomplished with HTTP status codes**
- **Setting status codes:**
  - Redirect user with response.sendRedirect(someURL)
    - If you insert user-supplied data into the URL, encode with URLEncoder.encode
  - Send 404 error pages with sendError
  - In general, set via response.setStatus
- **Most important status codes**
  - 200 (default)
  - 302 (forwarding; set with sendRedirect)
  - 401 (password needed)
  - 404 (not found; set with sendError)

Questions?

18