



# 服务器响应的生成： HTTP状态代码

JSP, Servlet, & Struts Training Courses: <http://courses.coreservlets.com>  
*Available in US, China, Taiwan, HK, and Worldwide*

JSP and Servlet Books from Sun Press: <http://www.coreservlets.com>  
*Available in English, Chinese (simplified and traditional script),  
and 12 other languages*

2

## 议程

- HTTP响应的格式
- 如何设置状态代码
- 状态代码适用于什么用途
- 处理重定向和错误页面的快捷方法
- 一个servlet：根据浏览器的不同将用户重定向到不同页面
- 各种搜索引擎的一个前端

3

# HTTP请求/响应

- 请求

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

- 响应

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

4

JSP/servlet/Stuts/JSP training: <http://www.coreservlets.com>

# 状态代码的设定

- **response.setStatus(int statusCode)**
  - 代码要是一个常量，不能是显式的整型数。这些常量定义在HttpServletResponse中。
  - 名字来源于标准消息。例如：SC\_OK，SC\_NOT\_FOUND等。
- **response.sendError(int code, String message)**
  - 将消息包装进小的HTML文档
- **response.sendRedirect(String url)**
  - 将状态代码设为302
  - 还要设置Location响应报头

5

JSP/servlet/Stuts/JSP training: <http://www.coreservlets.com>

## 常见HTTP 1.1状态代码

- **200 (OK)**
  - 一切正常；随后是文档。
  - servlet的默认值
- **204 (No Content)**
  - 浏览器应该依旧显示之前的文档。
- **301 (Moved Permanently)**
  - 所请求的文档已经被永久性地移到其他地方(在Location中标示出来)。
  - 浏览器自动跳转到新的位置。

6

JSP/servlet/Stuts/JSE training: <http://www.coreservlets.com>

## 常见HTTP 1.1状态代码 ( 续 )

- **302 (Found)**
  - 所请求的文档被临时性地移往它处 ( 在Location报头中标示 )。
  - 浏览器自动跳转到新的位置。
  - 在设定这个报头时，servlet应该使用sendRedirect，而非setStatus。参见示例。
- **401 (Unauthorized)**
  - 浏览器试图在没有提供恰当的授权报头的情况下，访问受密码保护的页面。
- **404 (Not Found)**
  - 不存在该页面。servlet应该使用sendError来发送这个报头。
  - 问题：Internet Explorer和小型的(< 512KB)错误页面。默认地，IE忽略错误页面。
  - 可以到这个地方进行试验：<http://www.plinko.net/404/>

7

JSP/servlet/Stuts/JSE training: <http://www.coreservlets.com>

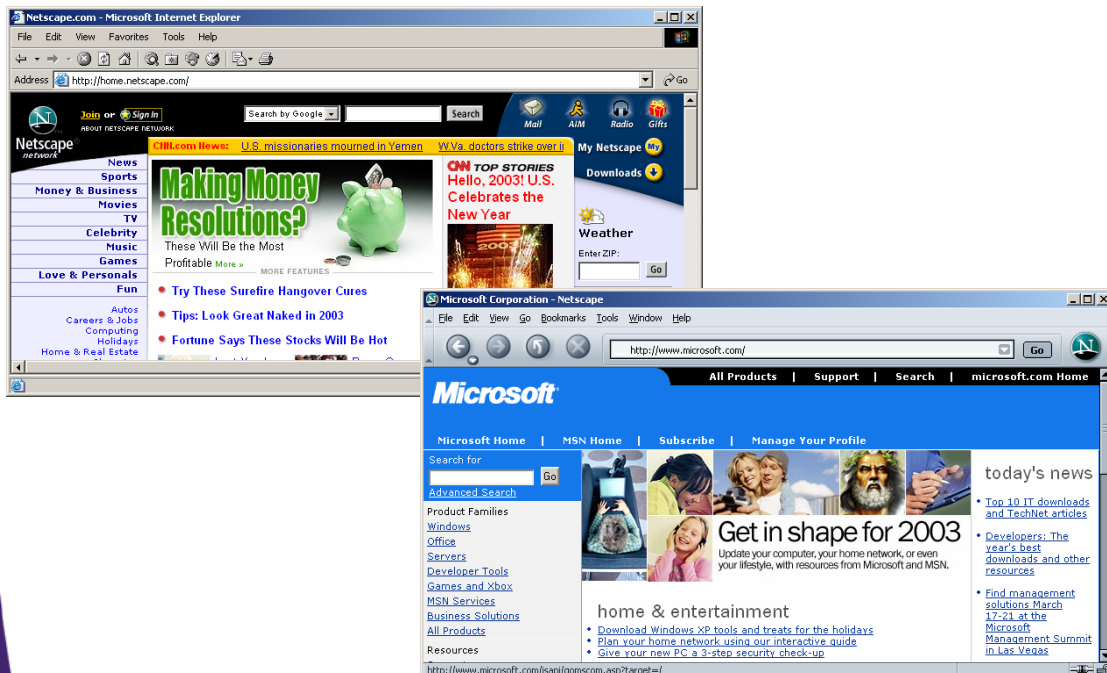
# 将用户重定向到浏览器相关页面的 servlet

```
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.indexOf("MSIE") != -1)) {
            response.sendRedirect("http://home.netscape.com");
        } else {
            response.sendRedirect("http://www.microsoft.com");
        }
    }
}
```

8

JSP/servlet/Stuts/JSF training: <http://www.coreservlets.com>

# 将用户重定向到浏览器相关页面的 servlet



9

JSP/servlet/Stuts/JSF training: <http://www.coreservlets.com>

## 各种搜索引擎的一个前端

```
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);
        String searchEngineName =
            request.getParameter("searchEngine");
        if ((searchEngineName == null) ||
            (searchEngineName.length() == 0)) {
            reportProblem(response, "Missing search engine name");
            return;
        }
    }
}
```

10

JSP/servlet/Stuts/JSP training: <http://www.coreservlets.com>

## 各种搜索引擎的一个前端 ( 续 )

```
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(response.SC_NOT_FOUND, message);
}
}
```

11

JSP/servlet/Stuts/JSP training: <http://www.coreservlets.com>

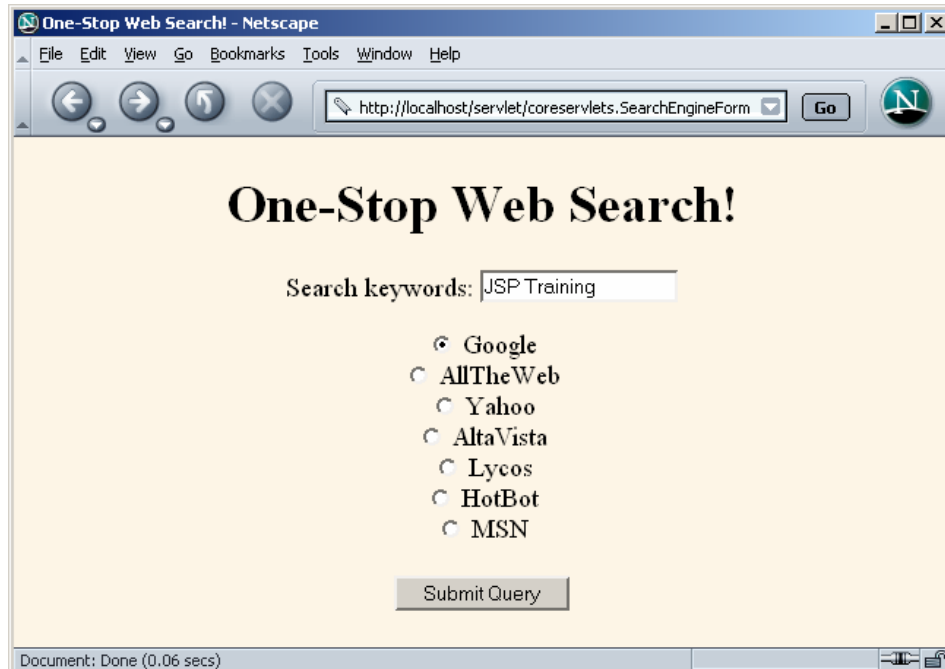
## 各种搜索引擎的一个前端 ( 续 )

```
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);  
    }  
  
    ...  
}
```

12

JSP/servlet/Stuts/JSE training: <http://www.coreservlets.com>

## 搜索引擎的前端：HTML表单



13

JSP/servlet/Stuts/JSE training: <http://www.coreservlets.com>

# 搜索引擎的前端：合法数据的结果

The image shows two browser windows side-by-side. The left window is Google Search for 'JSP Training', displaying results for 'Learn From Experts! Hands On JSP TRAINING in 40 Cities' and 'Quick Java & JSP Training'. The right window is Yahoo! Search for 'JSP Training', displaying results for 'Hands On Java Training - 5 Days \$1995', 'Learn JSP Programming Fast', and 'Learn Java's JSP - Java Server Pages'. Both windows show search results with titles, descriptions, and URLs.

14

JSP/servlet/Stuts/JSP training: <http://www.coreservlets.com>

# 搜索引擎的前端：无效数据的结果

The image shows two browser windows. The top window displays an 'HTTP Status 404 - Missing search engine name.' error message. The bottom window shows the raw HTML response for the 404 error, including the text '404 Missing search engine name' and 'Resm 2.1.1 (built Mon May 13 16:53:21 PDT 2002)'. The error message is displayed in a blue box with white text.

15

JSP/servlet/Stuts/JSP training: <http://www.coreservlets.com>

## 小结

- 许多servlet任务只能通过使用HTTP状态代码来完成
- 状态代码的设置:
  - 一般地，通过response.setStatus进行设置
  - 特殊情况 (302和404)，用response.sendRedirect和response.sendError进行设置。
- 最为重要的状态代码
  - 200 (默认)
  - 302 (转发；用sendRedirect设定)
  - 401 (需要密码)
  - 404 (未找到，用sendError设定)



## 问题？

JSP, Servlet, & Struts Training Courses: <http://courses.coreservlets.com>  
*Available in US, China, Taiwan, HK, and Worldwide*

JSP and Servlet Books from Sun Press: <http://www.coreservlets.com>  
*Available in English, Chinese (simplified and traditional script),  
and 12 other languages*