



Servlet Basics

JSP, Servlet, Struts, JSF, AJAX, & Java 5 Training: <http://courses.coreservlets.com>
J2EE Books from Sun Press: <http://www.coreservlets.com>



For live J2EE training, see training courses on JSP, servlets, Struts, JSF, AJAX, and Java 5 at <http://courses.coreservlets.com/>.



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.
Additional topics available upon request.

Agenda

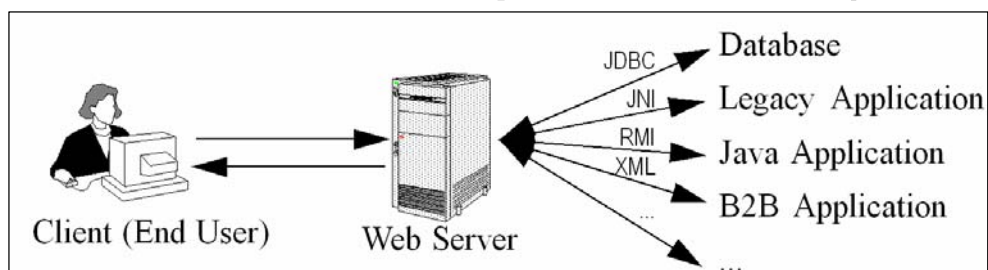
- The basic structure of servlets
- A simple servlet that generates plain text
- A servlet that generates HTML
- Servlets and packages
- Some utilities that help build HTML
- The servlet life cycle
- Servlet debugging strategies

4

J2EE training: <http://courses.coreservlets.com>

A Servlet's Job

- Read explicit data sent by client (form data)
- Read implicit data sent by client (request headers)
- Generate the results
- Send the explicit data back to client (HTML)
- Send the implicit data to client (status codes and response headers)



5

J2EE training: <http://courses.coreservlets.com>

A Servlet That Generates Plain Text (HelloWorld.java)

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("Hello World");
    }
}
```



J2EE training: <http://courses.coreservlets.com>

6

A Servlet That Generates HTML

- **Tell the browser that you're sending it HTML**
 - `response.setContentType("text/html");`
- **Modify the `println` statements to build a legal Web page**
 - Print statements should output HTML tags
- **Check your HTML with a formal syntax validator**
 - <http://validator.w3.org/>
 - <http://www.htmlhelp.com/tools/validator/>

7

J2EE training: <http://courses.coreservlets.com>

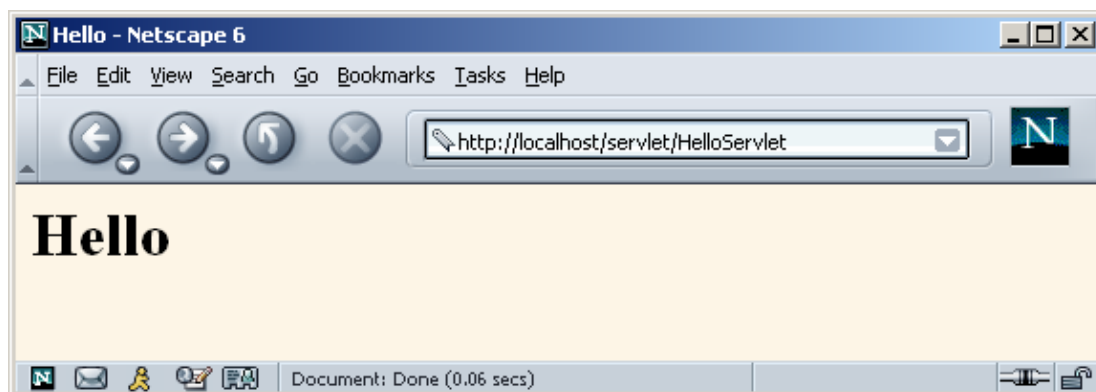
A Servlet That Generates HTML (Code)

```
public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String docType =
            "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 "+
            "Transitional//EN">\n";
        out.println(docType +
                    "<HTML>\n" +
                    "<HEAD><TITLE>Hello</TITLE></HEAD>\n"+
                    "<BODY BGCOLOR=\"#FDF5E6\"\n" +
                    "<H1>Hello</H1>\n" +
                    "</BODY></HTML>");
    }
}
```

8

J2EE training: <http://courses.coreservlets.com>

A Servlet That Generates HTML (Result)



9

J2EE training: <http://courses.coreservlets.com>

Packaging Servlets

- **Move the files to a subdirectory that matches the intended package name**
 - For example, I'll use the `coreservlets` package for most of the rest of the servlets in this course. So, the class files need to go in a subdirectory called `coreservlets`.
- **Insert a package statement in the class file**
 - E.g., top of `HelloServlet2.java`:
`package coreservlets;`
- **Keep CLASSPATH referring to top-level dir**
 - E.g., `C:\Servlets+JSP`. (No changes to CLASSPATH!)
- **Include package name in URL**
 - `http://localhost/servlet/coreservlets>HelloServlet2`

10

J2EE training: <http://courses.coreservlets.com>

Packaging Servlets: HelloServlet2 (Code)

```
package coreservlets;

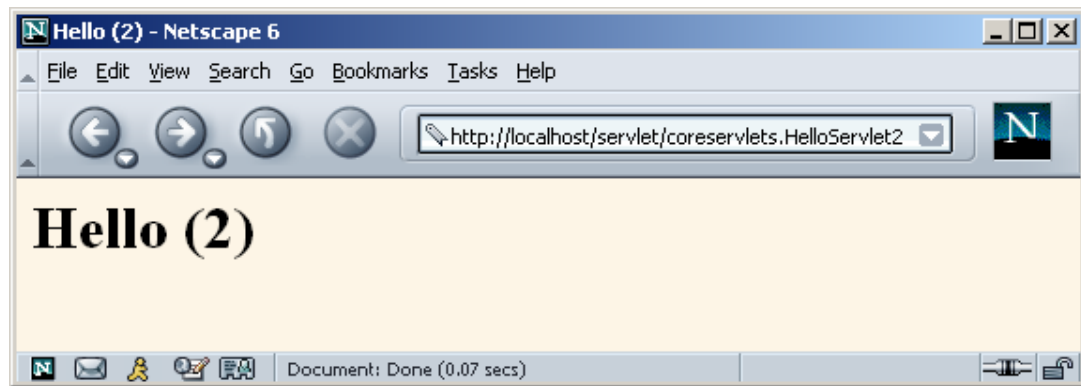
...

public class HelloServlet2 extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String docType =
            "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 "+
            "Transitional//EN">\n";
        out.println(docType +
            "<HTML>\n" +
            "<HEAD><TITLE>Hello (2)</TITLE></HEAD>\n"+
            "<BODY BGCOLOR=\"#FDF5E6\">\n" +
            "<H1>Hello (2)</H1>\n" +
            "</BODY></HTML>");
    }
}
```

11

J2EE training: <http://courses.coreservlets.com>

Packaging Servlets: HelloServlet2 (Result)



12

J2EE training: <http://courses.coreservlets.com>

Some Simple HTML-Building Utilities

```
public class ServletUtilities {
    public static final String DOCTYPE =
        "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 \" +
        \"Transitional//EN\">";

    public static String headWithTitle(String title) {
        return(DOCTYPE + "\n" +
            "<HTML>\n" +
            "<HEAD><TITLE>" + title +
            "</TITLE></HEAD>\n");
    }
    ...
}
```

- **Don't go overboard**
 - Complete HTML generation packages usually work poorly
 - The JSP framework is a better solution

13

J2EE training: <http://courses.coreservlets.com>

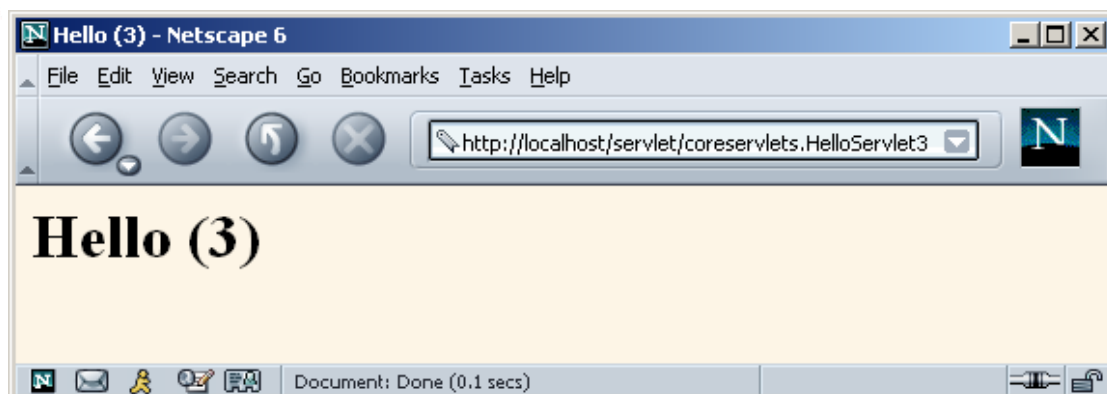
HelloServlet3: HelloServlet with Packages and Utilities

```
package coreservlets;  
  
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
public class HelloServlet3 extends HttpServlet {  
    public void doGet(HttpServletRequest request,  
                      HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        String title = "Hello (3)";  
        out.println(ServletUtilities.headWithTitle(title) +  
                   "<BODY BGCOLOR=\"#FDF5E6\">\n" +  
                   "<H1>" + title + "</H1>\n" +  
                   "</BODY></HTML>");  
    }  
}
```

14

J2EE training: <http://courses.coreservlets.com>

HelloServlet3: Result



15

J2EE training: <http://courses.coreservlets.com>

The Servlet Life Cycle

- **init**
 - Executed once when the servlet is first loaded.
Not called for each request.
- **service**
 - Called in a new thread by server for each request.
Dispatches to doGet, doPost, etc.
Do not override this method!
- **doGet, doPost, doXxx**
 - Handles GET, POST, etc. requests.
 - Override these to provide desired behavior.
- **destroy**
 - Called when server deletes servlet instance.
Not called after each request.

16

J2EE training: <http://courses.coreservlets.com>

Why You Should Not Override service

- You can add support for other services later by adding doPut, doTrace, etc.
- You can add support for modification dates by adding a getLastModified method
- The service method gives you automatic support for:
 - HEAD requests
 - OPTIONS requests
 - TRACE requests
- **Alternative: have doPost call doGet**

17

J2EE training: <http://courses.coreservlets.com>

Debugging Servlets

- **Use print statements; run server on desktop**
- **Use Apache Log4J**
- **Integrated debugger in IDE**
- **Look at the HTML source**
- **Return error pages to the client**
 - Plan ahead for missing or malformed data
- **Use the log file**
 - `log("message")` or `log("message", Throwable)`
- **Separate the request and response data .**
 - Request: see EchoServer at www.coreservlets.com
 - Response: see WebClient at www.coreservlets.com
- **Stop and restart the server**

18

J2EE training: <http://courses.coreservlets.com>

Summary

- **Main servlet code goes in doGet or doPost:**
 - The `HttpServletRequest` contains the incoming information
 - The `HttpServletResponse` lets you set outgoing information
 - Call `setContentType` to specify MIME type
 - Call `getWriter` to obtain a `Writer` pointing to client
- **One-time setup code goes in init**
 - Servlet gets initialized and loaded once
 - Servlet gets invoked multiple times
 - Initialization parameters set in `web.xml` (covered in detail in *More Servlets & JavaServer Pages* Chapter 5)

19

J2EE training: <http://courses.coreservlets.com>



Questions?

JSP, Servlet, Struts, JSF, AJAX, & Java 5 Training: <http://courses.coreservlets.com>
J2EE Books from Sun Press: <http://www.coreservlets.com>