



servlet基础

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

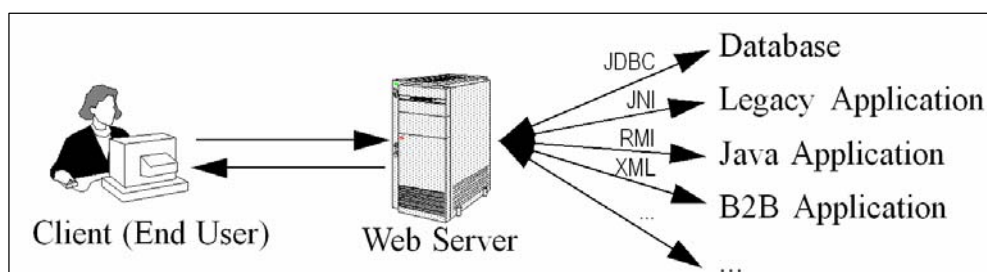
议程

- **servlet的基本结构**
- **生成纯文本的servlet**
- **生成HTML的servlet**
- **servlet和包**
- **协助构建HTML的一些实用工具**
- **servlet的生命周期**
- **servlet的调试**

3

servlet的功用

- 读取客户程序发送来的显式数据 (表单数据)
- 读取客户程序发送来的隐式数据 (请求报头)
- 生成相应的结果
- 发送显式的数据给客户程序 (HTML)
- 发送隐式的数据给客户程序 (状态代码和响应报头)



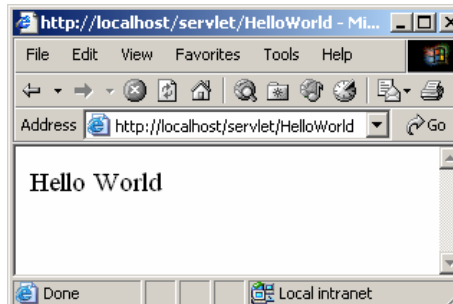
4

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

生成纯文本的servlet (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");
    }
}
```



5

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

生成HTML的servlet

- 告知浏览器接下来发送的是HTML
 - response.setContentType("text/html");
- 修改println语句以构建合法的Web页面
 - 打印语句应该输出相应的HTML标签。
- 用形式语法验证器 检查生成的HTML
 - <http://validator.w3.org/>
 - <http://www.htmlhelp.com/tools/validator/>

6

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

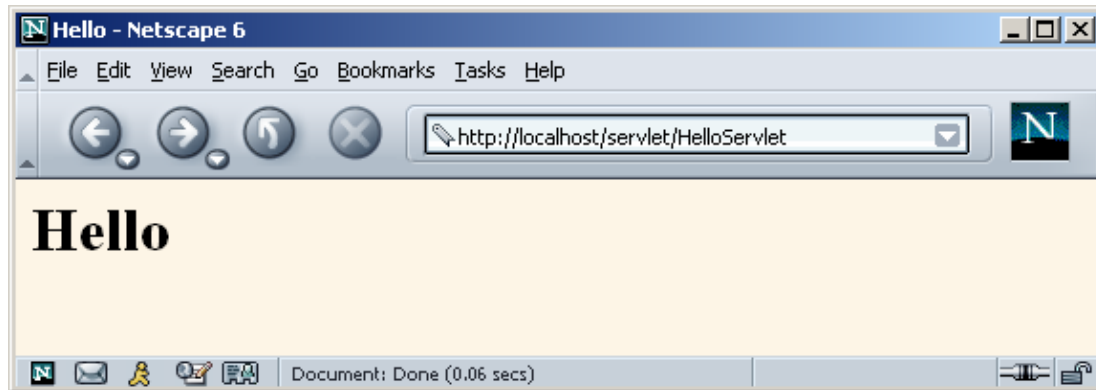
生成HTML的servlet (代码)

```
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 \"/></pre>
```

7

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

生成HTML的servlet (结果)



8

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

servlet的打包

- 将文件移动到与预想的包名相匹配的子目录中
 - 例如，本课程中接下来使用的servlet中大部分都使用 `coreservlets` 包。因此，类文件需要放置在名为 `coreservlets` 的子目录中。
- 在类文件中插入包语句
 - 比如，在 `HelloServlet2.java` 的顶部：
`package coreservlets;`
- 要保证 **CLASSPATH** 指出顶层目录
 - E.g., `C:\Servlets+JSP`. (No changes to CLASSPATH!)
- 在 **URL** 中包括包名
 - `http://localhost/servlet/coreservlets>HelloServlet2`

9

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

servlet的打包：HelloServlet2（代码）

```
package coreservlets;

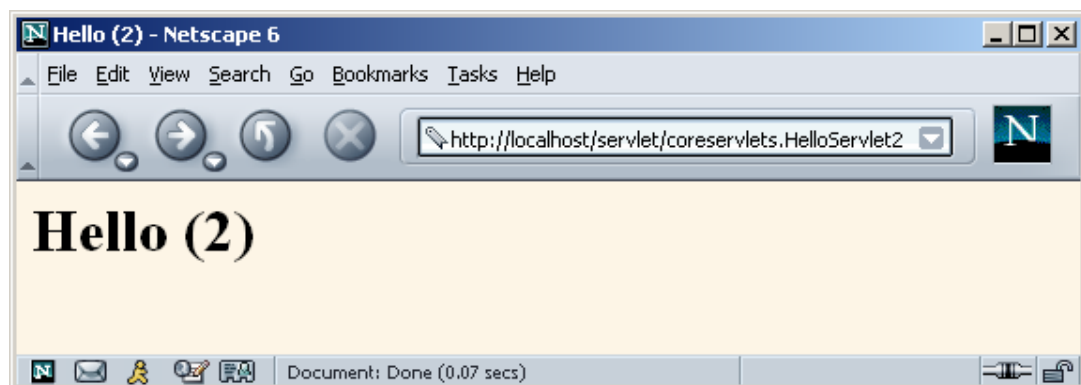
...

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>");
    }
}
```

10

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

servlet的打包：HelloServlet2（结果）



11

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

一些简单的、用以构建HTML的实用工具类

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

- **不要走极端**

- 完整的HTML生成包常常不能很好地工作
- 应用JSP框架是更好的解决方案

12

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

HelloServlet3: 用到包和实用工具类的HelloServlet

```
package coreservlets;

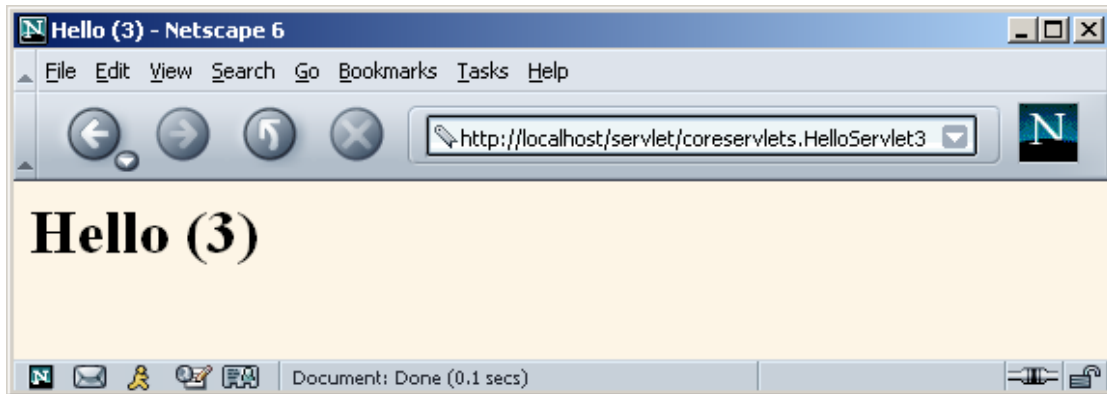
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

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>");
    }
}
```

13

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

HelloServlet3: 结果



14

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

servlet的生命周期

- **init**
 - 仅在servlet首次载入时执行一次。
不是每次请求都要调用。
- **service**
 - 在新线程中由服务器为每个请求而调用。发送到doGet, doPost等。
不要覆盖这个方法！
- **doGet, doPost, doXxx**
 - 处理GET, POST, 等请求。
 - 覆盖这些方法以提供期望的行为。
- **destroy**
 - 在服务器删除servlet的实例时调用。
不是每次请求后都调用。

15

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

为什么不应该覆盖service

- 之后可以通过加入doPut , doTrace等支持其他服务。
- 通过添加getLastModified方法 , 就可以支持修改日期。
- **service**方法自动提供下述支持 :
 - HEAD请求
 - OPTIONS请求
 - TRACE请求
- **其他方案 : 让doPost调作doGet**

servlet的调试

- **使用打印语句; 在桌面计算机上运行服务器**
- **使用Apache Log4J**
- **使用IDE集成的调试器**
- **直接查看HTML源码**
- **返回错误页面给客户**
 - 需要针对缺失或异常数据预先规划
- **使用日志文件**
 - log("message") 或 log("message", Throwable)
- **分别处理请求和响应数据**
 - 请求: 参见EchoServer (www.coreservlets.com)
 - 响应: 参见WebClient (www.coreservlets.com)
- **停止并重新启动服务器**

小结

- **主要的servlet代码都在doGet或doPost中:**
 - HttpServletRequest包含输入信息
 - HttpServletResponse允许我们设置发送出去的信息
 - 调用setContentType指定MIME类型
 - 调用getWriter获取针对客户的Writer
- **一次性的设置代码放在init中**
 - servlet仅仅初始化并载入一次
 - servlet被多次调用
 - 初始化参数在web.xml中设定 (在*More Servlets & JavaServer Pages* 一书的第5章详细介绍)

18

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

© 2004 Marty Hall



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

19