



# Servlet & JSP技术概述

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

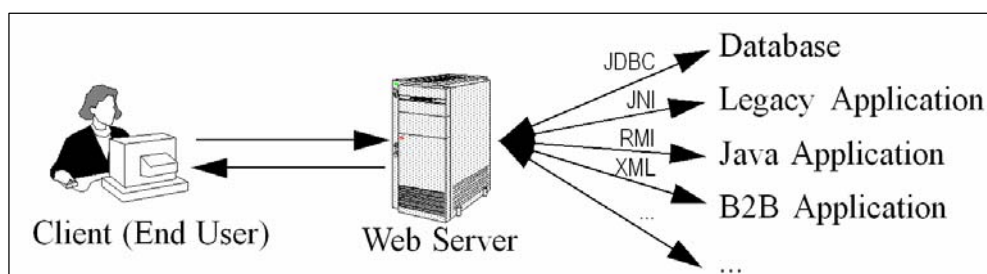
## 议程

- **servlet的功能和用途**
- **动态构建Web页面**
- **servlet和其他技术的对比**
- **JSP的功能和用途**
- **服务器的配置**
- **开发环境的配置**
- **测试安装是否正确**
  - 服务器基本功能测试
  - HTML/JSP测试
  - servlet测试

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

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

## 为什么要动态地构建Web页面?

- **Web页面的内容建立在用户提交的数据之上**
  - 比如：搜索引擎的结果页面、在线商店中反映订单信息的页面。
- **Web页面的内容由频繁变动的数据导出**
  - 比如：天气预报或新闻头条页面
- **Web页面用到了来自于数据库或其他服务器端资源的信息**
  - 比如：电子商务网站有可能使用servlet动态地构建Web页面，列出待售商品的最新价格。

5

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

# Servlet相比于“传统”CGI技术的优点

- **高效率**
  - 使用线程取代操作系统进程，单一servlet副本
- **便利**
  - 众多高层实用程序
- **功能强大**
  - 提供数据共享，连接共享，持续存储等诸多功能
- **可移植性强**
  - 几乎可以在所有操作系统、所有服务器上运行
- **花费少**
  - 有众多免费或收费低廉的服务器可以选择
- **安全**
  - 不涉及外壳程序，也就杜绝了使用转义符的攻击。没有缓冲区溢出的问题。
- **主流**
  - 参见下一页

6

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

## 主流

- **流行:**
  - Java技术最为广泛的应用
  - 中型、大型Web应用的主导技术
- **支持广泛:**
  - Apache, Oracle, IBM, Sybase, BEA, Macromedia, Caucho, Sun/iPlanet, New Atlanta, ATG, Fujitsu, Lutris, Silverstream, the World Wide Web Consortium (W3C), 还有许多
  - 适用于IIS和Zeus的插件
- **运行平台:**
  - Windows, Unix/Linux, MacOS, VMS, and IBM大型机OSs
- **主要应用:**
  - 航空公司，旅馆，
  - 电子商务站点，搜索引擎，
  - 银行，金融财经网站等等.....



7

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

## JavaServer Pages (JSP) - 对servlet功能的扩展

- **思想:**
  - 页面的大部分使用常规HTML
  - 使用特殊的标签标记动态内容
  - 本课程的第二部分详细论述相关细节

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Welcome to Our Store</TITLE></HEAD>
<BODY>
<H1>Welcome to Our Store</H1>
<SMALL>Welcome,
<!-- User name is "New User" for first-time visitors -->
<%= coreservlets.Utils.getUserNameFromCookie(request) %>
To access your account settings, click
<A HREF="Account-Settings.html">here.</A></SMALL>
<P>
Regular HTML for rest of on-line store's Web page
</BODY></HTML>
```

8

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

## 服务器的安装和配置

1. 下载并安装Java软件开发包 (SDK)
  2. 下载某种服务器程序
  3. 配置该服务器
  4. 设置自己的开发环境
  5. 测试安装设置是否正确
  6. 实现一种简单的部署方法
  7. 创建定制Web应用
- **如何配置Tomcat 4和5的详细信息，参见：**
    - <http://www.coreservlets.com/Apache-Tomcat-Tutorial/>

9

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

## 下载和安装Java SDK (或JDK)

- **推荐采用的Java版本**
  - JDK 1.4
- **在<http://java.sun.com/j2se/1.4/>获取**
  - 一定要按照Java文档中的描述正确地设置PATH变量
- **提供支持的Java最低版本**
  - Servlets 2.3 and JSP 1.2 (独立服务器).
    - Java 1.2 or later.
  - J2EE 1.3 (其中包括servlets 2.3 and JSP 1.2).
    - Java 1.3 or later.
  - Servlets 2.4 and JSP 2.0 (独立服务器).
    - Java 1.3 or later.
  - J2EE 1.4 (其中包括servlets 2.4 and JSP 2.0).
    - Java 1.4 or later.

10

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

## 为桌面开发下载免费服务器

- **Apache Tomcat**
  - <http://jakarta.apache.org/tomcat/>
  - 安装与设置方面的细节，参见：  
<http://www.coreservlets.com/Apache-Tomcat-Tutorial/>
- **Macromedia JRun**
  - <http://www.macromedia.com/software/jrun/>
- **Caucho Resin**
  - <http://caucho.com/products/resin/>
- **New Atlanta ServletExec**
  - <http://www.newatlanta.com/products/servletexec/>
- **Jetty**
  - <http://jetty.mortbay.org/jetty/>

11

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



## 配置服务器程序

- 标出SDK的安装目录
  - 如果为Tomcat，则需要设置JAVA\_HOME
- 指定端口
  - 将端口从默认值（一般为8080）改为80
- 完成各个服务器专有的相关定制
  - 如果为Tomcat，则
    - 启用servlet重新载入功能
    - 启用ROOT上下文
    - 打开servlet调用器
    - **完成这些更改之后。**安装时可以参见书籍的相关内容或者参考<http://www.coreservlets.com/Apache-Tomcat-Tutorial/>

## 设置开发环境

- 创建开发目录
  - 选择一个位置，用以开发servlet，JSP文档和支持类（比如：C:\Servlets+JSP）。
- 设定CLASSPATH
  - 告诉编译器servlet和JSP的JAR文件，以及开发目录所在的位置。
  - *对于初学者来讲，不正确地设置这个变量是最常见的问题根源。*
- 制作启动和停止服务器的快捷方式
  - 确保能够方便地启动和停止服务器。
- 标记或安装servlet和JSP的API文档
  - 经常需要参考这份文档，因此要将它放在手边。

## 测试系统的设置

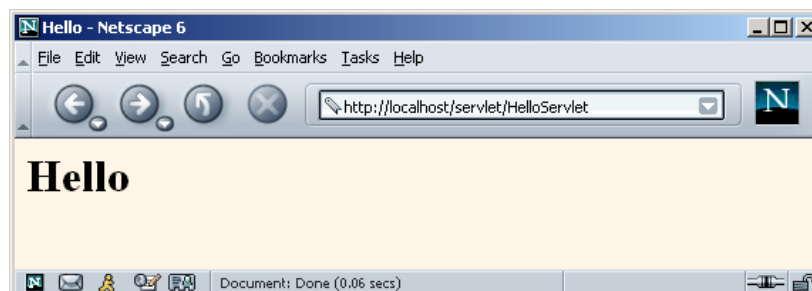
- 检验Java的安装
  - 一定要保证下面的命令都要得到有意义的结果：
    - java -version
    - javac -help
- 检查服务器的基本配置
  - 启动服务器并访问服务器的主页 ( http://localhost/ )
  - 访问一个简单的用户自定义HTML页面。
    - 从本书的源代码档案中下载Hello.html
    - 放在 *install\_dir/webapps/ROOT* 目录中
    - 用 <http://localhost/Hello.html> 访问该页面
  - 访问一个简单的用户自定义JSP页面。
    - 下载Hello.jsp并将其放到 *install\_dir/webapps/ROOT* 中
    - 用 <http://localhost/Hello.jsp> 访问它

14

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

## 测试系统的设置 ( 续 )

- 编译并部署一个不使用包的servlet
  - 从源代码档案中下载HelloServlet.java
  - 放到开发目录中 ( 比如C:\Servlets+JSP )
  - 编译 ( 如果有错误, 请检查CLASSPATH )
  - 将HelloServlet.class移动到 *install\_dir/webapps/ROOT/WEB-INF/classes*
  - 用 <http://localhost/servlet/HelloServlet> 进行访问

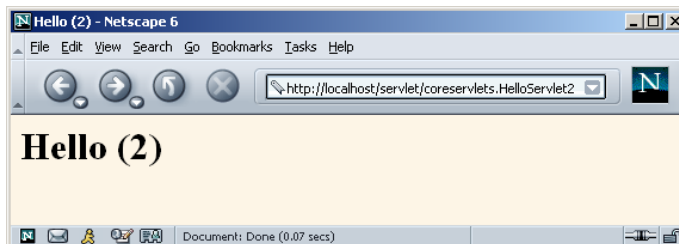


15

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

## 测试系统的设置 ( 续 )

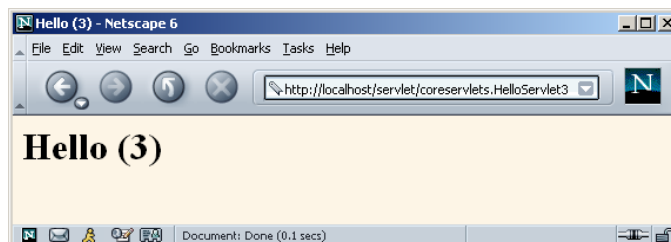
- 编译并部署一个使用包的servlet
  - 从源代码档案中下载HelloServlet2.java
  - 放在开发目录的coreservlets子目录中 ( 例如 : C:\Servlets+JSP\coreservlets )
  - 编译 ( 如果有错误 , 请检查CLASSPATH )
  - 将HelloServlet2.class移动到 *install\_dir/webapps/ROOT/WEB-INF/classes/coreservlets*
  - 用http://localhost/servlet/coreservlets.HelloServlet2进行访问



16

## 测试系统的设置 ( 续 )

- 编译并部署使用包且用到辅助类的servlet
  - 下载HelloServlet3.java和ServletUtilities.java
  - 放置在开发目录的coreservlets子目录中
  - 编译 ( 如果有错误 , 请检查 CLASSPATH )
  - 将全部类文件移动到 *install\_dir/webapps/ROOT/WEB-INF/classes/coreservlets*
  - 用http://localhost/servlet/coreservlets.HelloServlet3进行访问



17



## 实现一种简单的部署方法

- 复制成快捷方式或符号链接
  - 制作`install_dir/webapps/ROOT/WEB-INF/classes`的快捷方式
  - 对不使用包的servlet，将.class文件复制到这个快捷方式
  - 对使用包的servlets，将整个目录复制到快捷方式
    - 对于初学者这是最简单的方法
    - 这也是我在处理类时将采用的方法。
- 使用javac的-d选项
  - 使用这种方式可以将源文件放置在一个位置，自动将.class文件放置到另外的位置
- 由IDE来负责部署工作
- 使用ant或类似工具
  - 在使用定制Web应用时，Ant尤其普遍

18

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

## Web应用：预览

- 学习
  - 使用默认Web应用（在Tomcat中为ROOT）
  - 使用默认URL（`http://.../servlet/ServletName`）
  - 优点
    - 更简单
    - 不用重新启动服务器或编辑web.xml就能够进行测试
- 部署
  - 使用定制Web应用（在Tomcat中为install\_dir/webapps中的一个目录，结构与ROOT类似）
  - 在WEB-INF/web.xml注册定制URL
  - 优点
    - URL看起来更好一些
    - 通过使用注册的URL，可以获得许多高级特性（初始化参数、安全、过滤器等）

19

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

## 制作定制Web应用

1. 以默认Web应用的结构为范本生成一个目录。
  - HTML ( 以及JSP , 如果有的话 ) 文档放在顶层目录
  - web.xml文件安置在WEB-INF子目录
  - servlet和其他类不是在WEB-INF/classes中就是在WEB-INF/classes中与包名相匹配的子目录中。
  - 在Tomcat中 , 整个目录都在*install\_dir/webapps*中。
2. 更新CLASSPATH。
  - 将*webAppDir/WEB-INF/classes*加入其中。

20

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

## 制作定制Web应用

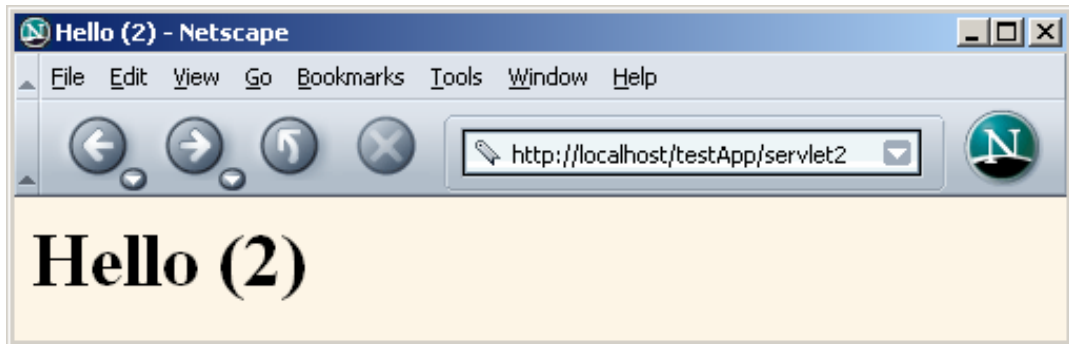
3. 在URL中使用目录名
  - 所有的URL都应该是下面的形式  
`http://host/webAppDir/...`
4. 使用web.xml指派定制URL
  - 使用servlet和servlet-mapping元素

```
<servlet>
  <servlet-name>Servlet2</servlet-name>
  <servlet-class>
    coreservlets.HelloServlet2
  </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>Servlet2</servlet-name>
  <url-pattern>/servlet2</url-pattern>
</servlet-mapping>
```

21

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

# 制作定制Web应用



22

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

## 小结

- **Servlet** 高效、可移植、功能强大且在行业中广泛采用
- 和实际的部署服务器无关，可以在桌面计算机上运行免费的服务器用于开发
- 入门之始：
  - 设置 CLASSPATH
    - servlet JAR 文件
    - 包的顶级目录
  - 将类文件存放到正确的位置
    - .../WEB-INF/classes
  - 使用正确的 URL，一般为 `http://host/servlet/ServletName`
- 首先下载已有的 **Servlet**
  - 从 [www.coreservlets.com](http://www.coreservlets.com) 的 HelloServlet 开始

26

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



# 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*