

<<Java Web应用开发技术>>

图书基本信息

书名：<<Java Web应用开发技术>>

13位ISBN编号：9787115271297

10位ISBN编号：7115271291

出版时间：2012-2

出版时间：人民邮电出版社

作者：王电钢，刘孙俊 主编

页数：236

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Java Web应用开发技术>>

内容概要

本书系统地介绍了使用Java语言开发Web应用的基础技术。首先，从Web应用的基础——HTTP入手；然后，逐章讲述了Java开发Web应用的各种技术和规范，涉及了Java Web容器、Servlet、JSP、过滤器、监听器、标记库、Web应用安全、Web开发模式；每章为读者配备了简明而又实用的示例。通过本书的学习，读者能够全面掌握Java语言Web编程技术，并对Java Web应用的基础知识有更深刻的了解。

本书可作为普通高等院校计算机及相关专业的教材，同时也可作为Java编程爱好者及开发人员的参考用书。

书籍目录

第1章 Java Web工作原理

1.1 解析HTTP

1.1.1 建立连接

1.1.2 发送请求信息

1.1.3 回送响应信息

1.1.4 关闭连接

1.2 Web服务器的缺点

1.3 服务器端网页编程

1.3.1 最早的解决方案CGI

1.3.2 Java的解决方案

1.4 Java Web 应用程序的组成

1.4.1 /WEB-INF和web.xml

1.4.2 Java类和资源文件

1.4.3 JAR文件

1.4.4 Web应用程序资源(WAR)文件

1.5 Web容器Tomcat简介

1.5.1 Tomcat目录结构

1.5.2 发布Web应用

1.5.3 Tomcat其他功能简介

本章小结

第2章 Servlet基础

2.1 什么是Servlet

2.2 Servlet API

2.3 Servlet生命周期

2.4 Servlet与HTTP

2.5 Servlet编码与部署步骤

2.5.1 编译Servlet代码

2.5.2 部署Servlet

2.6 Servlet配置

2.7 ServletContext

2.7.1 初始化Web应用程序参数

2.7.2 在Servlet中共享信息

2.7.3 虚拟路径转换为物理路径

2.7.4 临时目录

本章小结

第3章 使用Servlet处理HTTP响应

3.1 使用Servlet发送服务器端响应信息

3.2 设置响应状态行

3.2.1 HTTP状态码

3.2.2 设置状态码

3.3 设置响应头

3.4 设置响应消息体

3.4.1 传递文本流

3.4.2 传递二进制流

3.5 设置请求重定向

<<Java Web应用开发技术>>

3.5.1 使用sendRedirect方法实现重定向

3.5.2 设置自动刷新和等待页

本章小结

第4章 使用Servlet处理HTTP请求

4.1 使用Servlet接收服务器端请求信息

4.2 获取HTTP请求行

4.3 获取HTTP请求头

4.4 获取请求消息体

4.4.1 获取表单数据

4.4.2 获取原始表单数据

4.5 请求转发和请求范围

4.5.1 请求转发

4.5.2 请求范围

本章小结

第5章 Web应用程序状态管理

5.1 概述

5.2 Cookies

5.2.1 Cookies原理

5.2.2 在Servlet中管理Cookies

5.3 Session

5.3.1 Session原理

5.3.2 会话跟踪机制

5.3.3 HttpSession接口

5.3.4 会话超时管理

5.3.5 Application 与Session域范围的属性比较

5.3.6 Session持久化管理

5.4 URL地址重写

本章小结

第6章 对象作用域与Servlet事件监听器

6.1 对象作用域

6.1.1 ServletContext应用上下文

6.1.2 会话作用域

6.1.3 请求作用域

6.2 监听器概述

6.3 监听Web应用程序范围内的事件

6.4 监听会话范围内事件

6.5 监听请求生命周期内事件

本章小结

第7章 过滤器

7.1 过滤器概述

7.2 HelloWorld过滤器

7.3 过滤器API

7.3.1 Filter接口

7.3.2 FilterConfig接口

7.3.3 FilterChain接口

7.3.4 请求和响应包装器类

7.4 配置Filter

<<Java Web应用开发技术>>

7.4.1 元素

7.4.2 元素

7.4.3 配置过滤器链

7.5 使用请求和响应包装器

本章小结

第8章 JSP基础

8.1 JSP简介

8.2 JSP运行机制与生命周期

8.3 JSP语法和语义

8.4 脚本元素

8.4.1 Scriptlet

8.4.2 脚本表达式

8.4.3 声明

8.5 注释

8.6 JSP指令

8.7 JSP标准动作

8.7.1 动作

8.7.2 动作

8.8 JSP隐式对象

8.8.1 out对象

8.8.2 pageContext对象

本章小结

第9章 Java Web开发模式

9.1 Java Web开发模式的变迁

9.2 在JSP中使用JavaBean

9.3 JSP开发模式1

9.4 JSP开发模式2

9.4.1 MVC架构模式

9.4.2 MVC架构模式的Java Web实现

9.4.3 JSP模式2的开发步骤

9.5 在MVC中使用过滤器

本章小结

第10章 编写Java无脚本的JSP 页面

10.1 JSP表达式语言EL

10.1.1 EL表达式和JSP脚本表达式

10.1.2 在EL表达式中使用隐式变量

10.1.3 运算符

10.1.4 EL函数

10.2 JSP自定义标记库与标准标记库JSTL

10.2.1 通用标记

10.2.2 变量支持标记

10.2.3 流程控制

10.2.4 使用JSTL访问URL信息

本章小结

第11章 JSP自定义标记

11.1 JSP Tag Library简介

11.1.1 标记库和JavaBean的区别

<<Java Web应用开发技术>>

- 11.1.2 标记库的工作原理
- 11.2 标记库的使用
 - 11.2.1 一个简单的标记库程序
 - 11.2.2 标记处理
 - 11.2.3 标记描述程序
 - 11.2.4 在JSP文件中使用自定义标记
 - 11.2.5 在web.xml中定义标记
- 11.3 自定义标记开发步骤
- 第12章 Web应用程序异常处理
 - 12.1 JSP和Servlet
 - 12.2 Web应用程序异常处理
 - 12.3 日志
 - 12.3.1 Log4j简介
 - 12.3.2 Log4j组件
 - 12.3.3 Log4j的配置
 - 12.3.4 Log4j在Web应用程序的使用
 - 12.3.5 日志和性能
- 第13章 JavaWeb应用程序安全
 - 13.1 基本概念
 - 13.2 理解验证机制
 - 13.2.1 验证机制类型
 - 13.2.2 为Web应用程序定义验证机制
 - 13.3 声明式安全
 - 13.4 程序式安全
- 第14章 Web服务
 - 14.1 异构系统的交互
 - 14.2 解决方案
 - 14.2.1 CORBA
 - 14.2.2 消息队列
 - 14.3 Web服务简介
 - 14.3.1 Web服务基本规范
 - 14.3.2 扩展Web服务规范
 - 14.3.3 Web服务是如何工作的？
 - 14.4 JAX-RPC开发Web服务简介
 - 14.5 JAX-WS开发Web服务
 - 14.6 简单对象访问协议(SOAP)
 - 14.6.1 HTTP协议层
 - 14.6.2 SOAP信封
 - 14.6.3 SOAP协议头
 - 14.6.4 SOAP协议体
 - 14.6.5 SOAP错误响应
 - 14.6.6 SOAP样式
 - 14.6.7 SOAP消息交互模式
 - 14.7 Web服务描述语言(WSDL)
 - 14.7.1 数据类型
 - 14.7.2 定义消息

<<Java Web应用开发技术>>

14.7.3 接口定义

14.7.4 定义绑定

14.7.5 服务定义

编辑推荐

王电钢、刘孙俊主编的这本《Java Web应用开发技术》从Web应用程序工作原理入手，让读者对Web应用程序的网络基础协议HTTP有所了解；介绍了Web应用程序请求应答的工作方式，让读者对Web的基本原理有一个较为清晰的认识；然后再对Java EE平台上Web应用程序开发的核心技术和技术规范进行介绍，如Web容器、Servlet、过滤器、监听器、JSP等。

在对核心技术本身进行介绍的同时，介绍如何应用这些技术以及应用这些技术的各种技巧，如上传文件、动态生成图片等。

在介绍了核心技术之后，本书还从Web应用程序的整体架构角度介绍了使用Java开发Web应用程序的常用开发模式；介绍了各种MVC框架的基础技术JSP自定义标记；介绍了增强Java Web应用程序可靠性的异常处理机制和安全机制；最后介绍了用于异构系统整合的Web服务。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>