针对 Spring MVC 初学者的详尽概览和实践指南

学习指南

[美] Paul Deck 著
林仪明 崔毅 译

人民邮电出版社
POSTS & TELECOM PRESS

Spring MVC 学习指南

[美] Paul Deck 著
林仪明 崔毅 译

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

Spring MVC学习指南 / （美）戴克（Deck,P.）著 ；林仪明，崔毅译. -- 北京 ：人民邮电出版社，2015.5
ISBN 978-7-115-38639-7

Ⅰ. ①S… Ⅱ. ①戴… ②林… ③崔… Ⅲ. ①JAVA语言－程序设计－指南 Ⅳ. ①TP312-62

中国版本图书馆CIP数据核字(2015)第059323号

版 权 声 明

◆ 著 [美] Paul Deck
译 林仪明 崔 毅
责任编辑 陈冀康
责任印制 张佳莹 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市中晟雅豪印务有限公司印刷

◆ 开本：800×1000 1/16
印张：18.5
字数：350 千字 2015 年 5 月第 1 版
印数：1－3 000 册 2015 年 5 月河北第 1 次印刷

著作权合同登记号 图字：01-2014-2667 号

定价：49.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315

内容提要

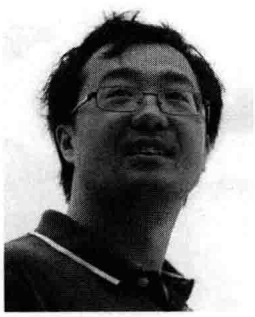
Spring MVC 是 Spring 框架中用于 Web 应用快速开发的一个模块，被广泛用作当今业界最主流的 Web 开发框架。

本书重在讲述如何通过 Spring MVC 来开发基于 Java 的 Web 应用。全书共计 12 章，分别从 Spring 框架、模型 2 和 MVC 模式、Spring MVC 介绍、控制器、数据绑定和表单标签库、转换器和格式化、验证器、表达式语言、JSTL、国际化、上传文件、下载文件多个角度介绍了 Spring MVC。除此之外，本书还配有丰富的示例以供读者练习和参考。

本书是一本 Spring MVC 的教程，内容细致、讲解清晰，非常适合 Web 开发者和想要使用 Spring MVC 开发基于 Java 的 Web 应用的读者阅读。

译者简介

林仪明，男，现为 IBM 高级工程师。2004 年毕业于厦门大学软件学院，主要研究软件架构、应用中间件。目前在福州生活和工作，先后从事软件技术开发，软件架构设计以及团队管理等工作，有多年的开发设计和管理经验，目前提供 IBM 中间件产品支持工作。



崔毅，男，JustCommodity Software Solution Pte Ltd 技术部经理。2007 年毕业于北京航空航天大学计算机学院，获硕士学位，主要研究 Web 服务、信息交换中间件。目前在新加坡生活和工作，先后从事技术开发，系统分析，系统实施，咨询顾问，和产品研发管理等工作，有多年的开发设计和管理经验，目前负责一个产品线。

前言

Spring MVC 是 Spring 框架中用于 Web 应用快速开发的一个模块。Spring MVC 的 MVC 是 Model-View-Controller 的缩写。它是一个广泛应用在图形化用户交互开发中的设计模式，不仅常见于 Web 开发，也广泛应用于桌面开发，如 Java Swing。

作为当今业界最主流的 Web 开发框架，Spring MVC（有时也称 Spring Web MVC）的开发技能相当热门。本书可供想要学习如何通过 Spring MVC 开发基于 Java 的 Web 应用的开发人员阅读。

Spring MVC 基于 Spring 框架、Servlet 和 JSP（JavaServer Page），在掌握这 3 种技术的基础上学习 Spring MVC 将非常容易。本书第 1 章针对 Spring 新手提供一个快速教程，附录 B 和附录 C 将帮助你快速学习 Servlet 和 JSP。如果希望深入学习 Servlet 和 JSP，推荐阅读由 Budi Kurniawan 所著的 *Servlet and JSP：A Tutorial* 一书。

本章接下来将讨论 HTTP、基于 Servlet 和 JSP 的 Web 编程，以及本书的章节结构。

HTTP

HTTP 使得 Web 服务器与浏览器之间可以通过互联网或内网进行数据交互。作为一个制定标准的国际社区，万维网联盟（W3C）负责和维护 HTTP。HTTP 第一版是 0.9，之后是 HTTP 1.0，当前最新版本是 HTTP 1.1。HTTP 1.1 版本的 RFC 编号是 2616，下载地址为 http://www.w3.org/Protocols/HTTP/1.1/rfc2616.pdf。

Web 服务器每天 24 小时不间断运行，并等待 HTTP 客户端（通常是 Web 浏览器）来连接并请求资源。通常，客户端发起一个连接，服务端不会主动连接客户端。

注意

2011 年，标准化组织 IETF（因特网工程任务组）发布了 WebSocket 协议，即 RFC 6455 规范。该协议允许一个 HTTP 连接升级为 WebSocket 连接，支持双向通信，这就使得服务端可以通过 WebSocket 协议主动发起同客户端的会话通信。

互联网用户需要通过点击或者输入一个 URL 链接或地址来访问一个资源，如下为两个示例：

```
http://google.com/index.html
http://facebook.com/index.html
```

URL 的第一个部分是 HTTP，代表所采用的协议。除 HTTP 外，URL 还可以采用其他类型的协议，如下为两个示例：

```
mailto:joe@example.com
ftp://marketing@ftp.example.com
```

通常，HTTP 的 URL 格式如下：

```
protocol://[host.]domain[:port][/context][/resource][?query string | path variable]
```

或者

```
protocol://IP Address[:port][/context][/resource][?query string | path variable]
```

中括号中的内容是可选项。因此，一个最简单的 URL 是 http://yahoo.ca 或者是 http://192.168.1.9。

需要说明的是，除了输入 http://google.com 外，还可以用 http://173.194.46.35 来访问谷歌。可以用 ping 命令来获取域名对应的 IP 地址。

```
ping google.com
```

由于 IP 地址不容易记忆，所以实践中更倾向于使用域名。一台计算机可以托管不止一个域名，因此，不同的域名可能指向同一个 IP。另外，example.com 或者 example.org 无法被注册，因为他们被保留作为各类文档手册举例使用。

URL 中的 Host 部分用来表示在互联网或内网中一个唯一的地址。例如，http://yahoo.com（没有 host）访问的地址完全不同于 http://mail.yahoo.com（有 host）。多年以来，作为最受欢迎的主机名，www 是默认的主机名。通常，http://www.domainName 会被映射到 http://domainName。

HTTP 的默认端口是 80 端口。因此，对于采用 80 端口的 Web 服务器，无需输入端口号。有时，Web 服务器并未运行在 80 端口上，此时必须输入相应的端口号。例如，Tomcat 服务器的默认端口号是 8080，为了能正确访问，必须提供输入端口号。

```
http://localhost:8080/index.html
```

localhost 作为一个保留关键字，用于指向本机。

URL 中的 context 部分用来代表应用名称，该部分也是可选的。一台 Web 服务器可以运行多个上下文（应用），其中一个可以配置为默认上下文。若访问默认上下文中的资源，可以跳过 context 部分。

最后，一个 context 可以有一个或多个默认资源（通常为 index.html、index.htm 或者 default.htm）。一个没有带资源名称的 URL 通常指向默认资源。当存在多个默认资源时，其中最高优先级的资源将被返回给客户端。

资源名后可以有一个或多个查询语句或者路径参数。查询语句是一个 Key/Value 组，多个查询语句间用“&”符号分隔。路径参数类似于查询语句，但只有 value 部分，多个 value 部分用“/”符号分隔。

HTTP 请求

一个 HTTP 请求包含 3 部分内容。

1．方法-URI-协议/版本。

2．请求头信息。

3．请求正文。

下面为一个 HTTP 请求示例：

```
POST /examples/default.jsp HTTP/1.1
Accept: text/plain; text/html
Accept-Language:en-gb
Connection:Keep-alive
Host:localhost
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.6)
Gecko/20100625 Firefox/3.6.6
Content-Length:30
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate

lastName=Blanks&firstName=Mike
```

请求的第一行 POST /examples/default.jsp HTTP/1.1 是方法-URI-协议/版本。

请求方法为 POST，URI 为/examples/default.jsp，而协议/版本为 HTTP/1.1。

HTTP 1.1 规范定义了 7 种类型的方法，包括 GET、POST、HEAD、OPTIONS、PUT、DELETE 以及 TRACE，其中 GET 和 POST 广泛应用于互联网。

URI 定义了一个互联网资源，通常解析为服务器根目录的相对路径。因此，通常用“/”符号打头。另外，URL 是 URI 的一个具体类型。（详见 http://www.ietf.org/rfc/rfc2396.txt）。

HTTP 请求包含的请求头信息包含关于客户端环境以及实体内容等非常有用的信息。例如，浏览器设置的语言，实体内容长度等。每个 header 都用回车/换行（即 CRLF）分隔。

HTTP 请求头信息和请求正文用一行空行分隔，HTTP 服务器据此判断请求正文的起始位置。因此，在一些关于互联网的书籍中，CRLF 被作为 HTTP 请求的第 4 种组件。

示例中，请求正文是 lastName=Blanks&firstName=Mike。

在正常的 HTTP 请求中，请求正文的内容不止如此。

HTTP 响应

同 HTTP 请求一样，HTTP 响应也包含 3 部分。

1. 协议-状态码-描述。

2. 响应头信息。

3. 响应正文。

下面为一个 HTTP 响应示例：

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Date: Thu, 29 Sep 2013 13:13:33 GMT
Content-Type: text/html
Last-Modified: Web, 28 Sep 2013 13:13:12 GMT
Content-Length: 112

<html>
<head>
<title>HTTP Response Example</title>
</head>
<body>
Welcome to Brainy Software
</body>
</html>
```

类似于 HTTP 请求报文，HTTP 响应报文的第一行说明了 HTTP 的版本是 1.1，并且请求

结果是成功的（状态代码 200 为响应成功）。

同 HTTP 请求报文头信息一样，HTTP 响应报文头信息也包含了大量有用的信息。HTTP 响应报文的响应正文是 HTML 文档。HTTP 响应报文的头信息和响应正文也是用 CRLF 分隔的。

状态代码 200 表示 Web 服务器能正确响应所请求的资源。若一个请求的资源不能被找到或者理解，则 Web 服务器将返回不同的状态代码。例如，访问未授权的资源将返回 401，而使用被禁用的请求方法将返回 405。完整的 HTTP 响应状态代码列表详见网址 http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html。

Servlet 和 JSP

Java Servlet 技术是 Java 体系中开发 Web 应用的底层技术。1996 年，Servlet 和 JSP 由 SUN 系统公司发布，以替代 CGI 技术，随后标准化来支持产生 Web 动态内容。CGI 技术为每一个请求创建相应的进程，但是，创建进程会耗费大量的 CPU 周期，最终导致很难编写可伸缩的 CGI 程序。相对于 CGI 程序，一个 Servlet 则快多了，这是因为当一个 Servlet 为响应第一次请求而被创建后，会驻留在内存中，以便响应后续请求。

JSP（JavaServer Pages）技术于 1999 年发布，以便简化 Servlet 开发。

从 Servlet 技术出现的那天起，人们开发了大量的 Web 框架来帮助程序员快速编写 Web 应用程序。这些开发框架让开发人员能更关注业务逻辑，减少编写“相似”的代码片段。尽管如此，开发人员依然需要去理解 Servlet 技术的基础概念。尽管实践中会应用一些诸如 Spring MVC、Struct 2 或者 JSF 等强大的开发框架，但如果没有深入理解 Servlet 和 JSP 技术，则无法有效和高效地开发。Servlet 是运行在 Servlet 容器中的 Java 程序，而 Servlet 容器或 Servlet 引擎相当于一个 Web 服务器，但是可以产生动态内容，而不仅是静态资源。

Servlet 当前的版本为 3.1，其规范定义可见 JSR （Java Specification Request）340（http://jcp.org/en/jsr/detail?id=340），基于 Java 标准版本 6 及以上。JSP 2.3 规范定义在 JSR 245（http://jcp.org/en/jsr/detail?id=245）。本书假定读者已经了解 Java 以及面向对象编程技术。对于 Java 新手，推荐阅读《Java 7：A Beginner's Tutorial》第 3 版（中文翻译版《Java 7 程序设计》已由机械工业出版社出版）。

一个 Servlet 是一个 Java 程序，一个 Servlet 应用包含了一个或多个 Servlet，一个 JSP 页面会被翻译并编译成一个 Servlet。

一个 Servlet 应用运行在一个 Servlet 容器中，它无法独立运行。Servlet 容器将来自用户的请求传递给 Servlet 应用，并将 Servlet 应用的响应返回给用户。由于大部分 Servlet 应用都会包含一些 JSP 界面，故称 Java Web 应用为“Servlet/JSP”应用会更恰当些。

Web 用户通过一个诸如 IE、火狐或者 Chrome 等 Web 浏览器来访问 Servlet 应用。Web 浏览器又被称为 Web 客户端。下图展示了一个典型的 Servlet/JSP 应用架构。

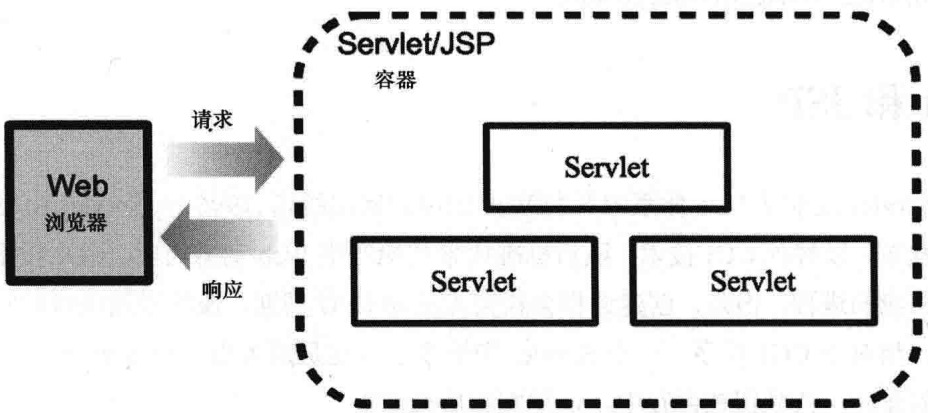


图 Servlet/JSP 应用架构

Web 服务端和 Web 客户端基于 HTTP 通信。因此，Web 服务端往往也称为 HTTP 服务端。

一个 Servlet/JSP 容器是一个能处理 Servlet 以及静态资源的 Web 服务端。过去，由于 HTTP 服务器更加健壮，所以人们更愿意将 Servlet/JSP 容器作为 HTTP 服务器（如 Apache HTTP 服务器）的一个模块来运行，在这种场景下，Servlet/JSP 容器用来产生动态内容，而 HTTP 服务器处理静态资源。今天，Servlet/JSP 容器已经更加成熟，并且被广泛地独立部署。Apache Tomcat 和 Jetty 作为最流行的 Servlet/JSP 容器，免费而且开源。下载地址为 http://tomcat.apache.org 以及 http://jetty.codehaus.org。

Servlet 和 JSP 仅是 Java 企业版众多技术之一，其他 Java 企业版技术包括 JMS、EJB、JSF 和 JPA 等，Java 企业版 7（当前最新版）完整的技术列表见 Http://www.oracle.com/technetwork/java/javaee/tech/index.html。

运行一个 Java 企业版应用需要一个 Java 企业版容器，常见的有 GlassFish、JBoss、Oracle

Weblogic 以及 IBM WebSphere。虽然可以将一个 Servlet/JSP 应用部署到 Java 企业版容器中，但一个 Servlet/JSP 容器其实就足够了，而且更加轻量级。Tomcat 和 Jetty 不是 Java 企业版容器，故它们不能运行 EJB 或 JMS。

关于本书

第 1 章：Spring 框架，介绍了最流行的开源框架。

第 2 章：模型 2 和 MVC 模式，讨论了 Spring MVC 实现的设计模式。

第 3 章：Spring MVC 介绍，本章编写了第一个 Spring MVC 应用。

第 4 章：基于注解的控制器，讨论了 MVC 模式中最重要的一个对象——控制器。本章，我们将学习如何编写基于注解的控制器，该方式由 Spring MVC 2.5 版本引入。

第 5 章：数据绑定和表单标签库，讨论 Spring MVC 最强大的一个特性，并利用它来展示表单数据。

第 6 章：转换器和格式化，讨论了数据绑定的辅助对象类型。

第 7 章：验证器，展示如何通过验证器来验证用户输入数据。

第 8 章：表达式语言。

第 9 章：JSTL。

第 10 章：国际化。

第 11 章：上传文件。

第 12 章：下载文件。

附录 A：Tomcat。

附录 B：Servlet。

附录 C：JavaServer Pages。

附录 D：部署描述符。

下载示例应用

本书所有的示例应用压缩包可以通过如下地址下载：

http://books.brainysoftware.com/download

目 录