

The Definitive Guide to Spring Web Flow

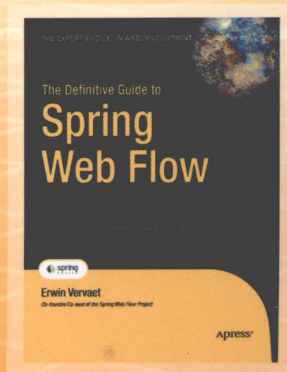
Spring Web Flow

权威指南

[比] Erwin Vervaet 著
乔晶峰 孙爱红 译



- Spring Web Flow创始人著作
- 体验全新的Web用户界面开发方式
- 深入剖析开源框架的设计与实现



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Java 系列

The Definitive Guide to Spring Web Flow

Spring Web Flow

权威指南

[比] Erwin Vervaet 著
乔晶峰 孙爱红 译

人民邮电出版社
北京

图书在版编目(CIP)数据

Spring Web Flow 权威指南 / (比) 弗维特
(Vervaeke, E.) 著; 乔晶峰, 孙爱红译. —北京: 人民
邮电出版社, 2009.12

(图灵程序设计丛书)

书名原文: The Definitive Guide to Spring Web
Flow

ISBN 978-7-115-21593-2

I. ① S… II. ①弗…②乔…③孙… III. ①
JAVA 语言—程序设计 IV. ① TP312

中国版本图书馆CIP数据核字(2009)第190164号

内 容 提 要

本书介绍了 Spring Web Flow 的构建系统 Spring Jumpstart、Spring Web Flow 架构和基础知识, 并结合
示例讲述了 Spring Web Flow 的高级概念、流执行管理和驱动, 同时说明了如何测试流。

本书适合各层次 Java Web 开发人员阅读和参考。

图灵程序设计丛书

Spring Web Flow权威指南

-
- ◆ 著 [比] Erwin Vervaeke
 - 译 乔晶峰 孙爱红
 - 责任编辑 傅志红
 - 执行编辑 杨 爽
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
 - 邮编 100061 电子函件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京铭成印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 17.5
 - 字数: 414千字 2009年12月第1版
 - 印数: 1—2 000册 2009年12月北京第1次印刷
 - 著作权合同登记号 图字: 01-2009-2912号

ISBN 978-7-115-21593-2

定价: 49.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *The Definitive Guide to Spring Web Flow* by Erwin Vervae, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2008 by Erwin Vervae. Simplified Chinese-language edition copyright © 2009 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

译者序

在企业级应用开发中，Java技术及其相关平台一直占据着巨大的市场份额。最近的权威市场调查表明，Java语言持续位列各种编程语言之榜首。而这其中，大多数又都是基于Web的应用程序。在活跃的Java开源社区，各种各样的开源框架不断推陈出新，使得各种新的应用需求得以实现，并推进企业应用开发技术的成熟和完善。

如果说前几年J2EE平台是企业级应用开发的重量级选择，那么这些年的后起之秀Spring无疑是轻量级产品的首选。重量与轻量之间，你到底会选择谁？在Spring诞生之前，人们一想到企业级应用开发，就会认为这样的应用开发只有选择重量级的平台才可靠，他们会认为诸如Struts这样的框架只适合于简单的Web应用程序，而无法承担企业应用之重。不过，随J2EE技术而来的问题则是高额的开发费用和维护费用。现在我们更有理由相信我们需要减负，需要更具灵活性的开发过程、更简单的测试过程以及更前沿的技术，因此，所谓的敏捷开发和轻量级产品越来越引起人们的注意。

Spring家族中有很多让人叫绝的产品，其中Spring Web Flow就是一个值得学习并应用到项目中的产品。到底什么是Spring Web Flow呢？它主要解决的问题又是什么？本书的作者Erwin Vervaeke是Spring Web Flow的创作者，他在本书中通过一个完整的实例循序渐进地讲解了这一切。我非常认同作者创作本书的思路，同时也认为这是学习一门新技术的最佳方式，很值得每一位学习者借鉴。作者认为在学习一门新技术（产品）时，应当很明确地知道自己所学的对象要解决的问题是什么，它又是通过哪种方式来解决这些问题的。为此，作者在第1章不惜笔墨，非常详细地解释了Spring Web Flow要解决的问题，即导航的控制和状态管理。众所周知，现在基于Web的企业应用已经越来越复杂了，复杂的交互式对话的应用越来越广泛。当这些交互式对话存在时，你不得不考虑如何控制好导航，如何管理好应用程序的状态，以避免由于用户任意使用浏览器上的后退按钮或者前进按钮而带来的各种问题。在这种情况下，如果你使用传统的框架技术，可能不得不花费很多的精力来处理这些底层的東西。请仔细看作者在第1章中如何将这内容娓娓道来吧。本书的第2章介绍了Spring Web Flow的构建系统Spring Jumpstart，它同时也是构建本书示例程序的系统。我建议读者仔细阅读这一章，把开发环境搭建好，有了这把利器，相信学习起来会事半功倍。第3章介绍Spring Web Flow架构，学习完这一章，你对Spring Web Flow就有了一个宏观的了解，知道它都由哪些模块组成，每一部分又都扮演什么样的角色。第4章主要介绍Spring Web Flow的基础知识，学习完这一章，你就能掌握Spring Web Flow的基础，这些基础知识在大多数场合中都能用到。第5章至第7章分别讲解Spring Web Flow的高级概念、流执行管理和驱动流执

行的一些知识。其中包括用于表达流定义的OGNL（对象图导航语言）、使用XML定义流的方法，如何管理流、驱动执行流以及和其他框架的集成等知识。第8章说明了如何测试Spring Web Flow的流。第9章和第10章演示了使用Spring Web Flow的示例应用程序。这些示例不仅可以帮助你更好地了解前面学到的知识，还可以作为开发Spring Web Flow应用程序的一个很好的模板。当然，这些示例都是贯穿全书的，在前面各章中都能找这些例子的影子。第11章则说明了更加高级的内容，即如何扩展Spring Web Flow。因为Spring Web Flow是完全开源的，你可以很轻松地修改这些源代码以满足自己的应用程序中的特殊需求。

作为Spring大家族的一员，Spring Web Flow的使用是离不开Spring框架的。所以，读者在学习它之前，有一定的Spring框架的知识是必需的。通过参考Spring官方参考手册（可从其网站<http://www.springsource.org>获得），你能找到该框架的大部分知识。当然，中文版的图书中，由人民邮电出版社出版的王福强所著的《Spring揭秘》也是值得一看的一本好书，该书以幽默诙谐的语言将抽象的Spring讲解得透彻而清晰。

尽管我们尽自己最大的努力用准确而贴切的语言翻译好本书，但是由于水平有限，不当之处在所难免，敬请广大读者批评指正。

引 言

2004年底，当我开始使用Spring Web Flow的时候，Web应用开发已经占据了Java企业级开发的半壁江山。当时，我已经在多个项目中使用了Struts，但是总觉得Struts还少点什么东西。在金融行业的几个项目中使用的一个专属框架激发了我的兴趣。该框架包括一个工作流引擎，这对于面向高端企业级应用开发的框架来说，是相当典型的一个特性。工作流引擎比较新颖的是，它也可以在Web应用程序中定义页面流。这为Java Web应用程序开发引入了非常直观的方法。

相对于当时一些主流框架所提供的以请求为中心的解决方案来说，如果把状态图作为Web应用程序中页面流的基础，则看起来更加自然，特别是在完成一个业务流程需要有很多复杂的用例，同时需要用户经过很多不同的步骤的情况下。2004年，Spring发展得非常迅猛，而我也已经在学习Spring框架了，我被它的设计和实现质量深深折服。我开始往Spring Web MVC框架中添加页面流控制器，最终诞生了后来的Spring Web Flow。

最初，Spring Web Flow着重采用基于状态图的方法，以便在Web应用程序中定义页面导航能变得更简单和更直观。这为Web应用程序开发者提供了一种强大的方式来表达页面导航规则。表达型的页面流定义也突出了对更好的导航控制的需要。对于试图控制页面导航的Web应用程序来说，臭名昭著的后退按钮造成了各种各样的麻烦。显然，Spring Web Flow要解决这个问题。

那段时间，我阅读了一篇探讨在Web应用程序中使用延续性（continuation）来解决导航问题的文章。这篇文章给了我启发，它非常适合Spring Web Flow的流执行模型，而且还补充了一些遗失的环节。现在Spring Web Flow融合了两种非常具有吸引力的补充特性：

- 一种在Web应用程序中表达页面导航规则的直观且易于使用的方法；
- 一个强大的、健壮的导航控制系统。

而要引入第三种极具吸引力的特性仅需要很小的一步：

- 通过良好定义的输入/输出契约，将页面流封装为黑盒应用程序模型。

Spring Web Flow的发展已经走过了几个年头，它已经从一个不起眼的Spring Web MVC框架的“流控制器”发展成为如今的“下一代Java Web应用程序控制器框架，通过它，开发者可以将用户动作建模为称为流的高级模型。该框架不仅能提高生产力及增强程序的可测试性，而且提供了一个很强大的解决方案来强制导航规则和管理应用程序状态”（Johnson et al 2003）。它的下一代Spring Web Flow 2引入了激动人心的特性来构建和运行富Web应用程序。很显然，Spring Web Flow目前已经是一个成熟的项目了，它已经应用到很多产品的部署中，而且还有非常活跃的用户社区。

本书不仅教你如何使用Spring Web Flow，也帮助你理解该框架背后的基本理论和动机。我真诚地希望你喜欢本书，并且能够在工作中快乐地使用Spring Web Flow。

关于 Spring Web Flow 项目

Spring Web Flow项目最初起源于一个小的开源项目——Ervacon Spring Web Flow(<http://www.ervacon.com/products/springwebflow>)，由我于2004年10月创建。该项目引起了Spring 框架团队中的开发者Keith Donald的注意，最终在2005年2月成为Spring框架的官方子项目。经过近两年的积极开发及数次预览发行版和试发行版，2006年10月发布了产品版1.0。该版本就是本书要介绍的主题。

在Spring Web Flow 1.0建立的坚固基础上，本产品的下一代开发工作还在继续着。2008年6月发布的Spring Web Flow 2做了架构上的一些调整，从而能让它更好地无缝集成到富Web环境中。它增强了对JSF和AJAX技术的支持，同时简化了流定义语法。

Spring Web Flow遵循著名的Apache 2许可协议，它是一个自由（开源）软件协议（Apache Software Foundation 2004）。许多开源项目都采用了Apache 2许可协议，其中包括Spring框架本身和Apache HTTP Server。按照该协议，用户可以为了任何商业或非商业目的使用Spring Web Flow，甚至可以修改和再分发。

Spring Web Flow有时被错误地写成“Spring WebFlow”，造成这种错误的原因大概是因为Spring Web Flow的包名——`org.springframework.webflow`，它把webflow写在一个单词中了。“Spring Web Flow”通常也简写为“SWF”。

Spring Web Flow的官方主页是<http://www.springframework.org/webflow>。对于想紧跟该项目发展步伐的用户来说，这是个主要的资源。如果读完本书之后，你还有没有弄明白的问题，那么可以到非常活跃的Spring用户论坛<http://forum.springframework.org>上获得答案。另外，Ervacon Spring Web Flow Portal (<http://www.ervacon.com/products/swf>)上也有一些实用的信息，如使用Spring Web Flow的一些小技巧和实践经验等。

关于本书

本书的目的是教你如何使用Spring Web Flow。书中包括了基础知识和高级用例，并且深入介绍了当前Spring Web Flow所提供的全部特性。你还能学到如何扩展该框架的特性。阅读完全书后，你就可以自称为Spring Web Flow专家了！

Spring Web Flow 1 和 Spring Web Flow 2

进一步讲解之前，需要声明的是本书主要讲解Spring Web Flow 1，而其下一版本Spring Web Flow 2将是另一本书探讨的主题。

版本1和版本2实质上是两个不同的产品。两个版本的核心概念是相同的，但是技术上却截然不同的。因此，Spring Web Flow 2不是向后兼容Spring Web Flow 1的。从版本1过渡到版本2并不是一个简单的升级，而是一次迁移。本书的后记中将详细说明这两个版本之间的不同点，帮助你

选择适合自己的版本。

在本书中，Spring Web Flow指的就是Spring Web Flow 1，不带版本号。当提及Spring Web Flow 2的时候，会带上版本号。

目标读者

本书旨在作为新用户和高级Spring Web Flow用户的参考书。如果你是个新用户，可以学到如何开始使用Spring Web Flow以及它所有的强大特性。作为高级用户，你将学到如何扩展该框架和一些最佳实践，同时你会发现本书提供了很多有趣的Spring Web Flow的设计内幕。

在开始阅读本书之前，你应该已经掌握了Java和Java Web应用程序开发的一些知识，如servlet、JSP等。本书中大部分例子都使用了Spring Web MVC框架。当然，如果你熟悉任何一个Web MVC框架（如Struts或者WebWork），这些例子对你来说应该也不成问题。

本书还假定读者已经了解Spring框架的基本知识及其指导思想，如反转控制模式、依赖注入等。读者不必是Spring专家，但是，如果你至少已经接触过Spring的类，像ApplicationContext和BeanFactory，并且理解Spring是如何把Bean连接起来的话，那么你将会有一个更加轻松的学习之旅。

与其在这里夸夸其谈Java Web应用开发的细节、Spring Web MVC，甚至Spring知识，倒不如推荐你去看Apress出版社的如下图书：*Beginning Spring 2: From Novice to Professional*，作者Dave Minter（2005）；*Pro Spring 2.5*^①，作者Jan Machacek、Jessica Ditt、Aleksa Vukotic和Anirvan Chakraborty（2008）；*Pro Java™ EE Spring Patterns: Best Practices and Design Strategies Implementing Java EE Patterns with the Spring Framework*^②，作者Dhrubojyoti Kayal（2008）。在Apress网站<http://www.apress.com>还可以找到其他的相关书籍。

概述

本书从Spring Web Flow的介绍性的内容开始一直讲到其深层次的知识。下面概要介绍每章的重点内容。当然，你也可以从头到尾地阅读本书，而且，如果你是Spring Web Flow初学者的话，我也建议你这么做。但如果你已经熟悉了该框架，就可以略过前两章，直接从第3章开始。

第1章：初识Spring Web Flow

这一章从宏观的角度来审视Spring Web Flow要解决的问题。除了要说明Spring Web Flow赖以生存的环境，还给出一个概念性的认识，即到底Spring Web Flow是什么。

第2章：入门

通过第1章宽泛的介绍后，第2章将开始Spring之旅。这一章将要介绍如何开始使用Spring Web Flow的所有细节。通过这一章的环境搭建，你将能很容易理解后续各章中的例子，便于自己动手尝试。

① 本书中文版《Spring高级程序设计》由人民邮电出版社出版。——编者注

② 本书中文版将由人民邮电出版社出版。——编者注

第3章：Spring Web Flow架构

在第1章和第2章中我们初步接触了Spring Web Flow的一些概念，第3章开始深入挖掘。这一章将介绍Spring Web Flow的架构，让你深入认识不同子系统，为后续各章学习Spring Web Flow的高级特性打下基础。

第4章：Spring Web Flow基础

这一章将介绍Spring Web Flow的基本特性，这些基本特性会在大多数的用例中用到。你将学到如何使用基于XML的流定义语言和Java的流定义语言来设计和实现Web流。

第5章：高级Web流概念

继第4章介绍了基本概念之后，这一章将详细讲述高级特性。你将学到如何从一个流中重用其他流作为其子流，从而实现Spring Web Flow所承诺的模块化。处理HTML表单数据也是本章的内容。

第6章：流执行管理

使用Spring Web Flow的时候，大多数的功夫都花在了如何定义流上。同样重要的是理解Spring Web Flow是如何管理流的执行和相关联的数据的。这些主题将在这一章介绍。

第7章：驱动流执行

这一章介绍如何将Spring Web Flow集成到宿主框架中，如Spring Web MVC和JSF，以及在这些环境中如何驱动流的执行。这一章中还讨论如何为Web流开发视图。

第8章：Spring Web Flow的测试

这一章将介绍如何对Spring Web Flow应用做单元测试。你将学到如何对流定义做集成测试，如何独立测试流相关的工作（如动作Action）等。

第9章：示例应用程序

这一章将展示一个非同寻常的应用程序，它融合了自由浏览和可控导航两种情形。这里要介绍的示例应用程序就是Spring银行。Spring银行是一个简单的电子银行应用，用户可以做管理银行账号、输入支付信息等事情。这个应用程序的用例将贯穿全书，用以解释和说明Spring Web Flow的特性集。

第10章：真实用例

本章讨论现实世界中与用例相关的常见问题。你将了解如何使你的流更加安全，如何跟踪面包屑（breadcrumb）^①，以及如何对Spring Web Flow应用程序进行压力测试等。

第11章：扩展Spring Web Flow

最后一章涵盖如何扩展和自定义Spring Web Flow。你将学会如何通过源代码构建Spring Web Flow等。

后记

书末的后记部分留给你一些思考，并带你一起浏览一下Spring Web Flow 2中的新功能和增强特性。

① 面包屑导航（Breadcrumb Navigation）这个概念来自童话故事《汉泽尔和格雷特儿》，当汉泽尔和格雷特儿穿过森林时，他们在沿途走过的地方都撒下了面包屑，以便找到回家的路。所以，面包屑导航的作用是告诉访问者他们目前在网站中的位置以及如何返回。——译者注

版式约定

本书使用简单易理解的版式约定。所有需要解释的文字，如Java类名称、代码段、文件名、XML元素等，均使用等宽字体。特别重要的主题用楷体表示，新的或者重要的代码段使用粗等宽字体。在命令行中执行的命令使用粗体。

本书中的多数程序清单已经根据页面重排了格式，方便阅读。在某些场合下会有多余的换行。反斜杠 (\) 用于表示多余的换行。

关于示例

本书中所有的示例应用程序都使用Java 5、Servlet API 2.4和JSP 2.0。请确保你已经安装了JDK 5，并且安装了合适的应用程序服务器或者Servlet引擎（如Tomcat 5或者Jetty 6）。

示例应用程序使用Spring Web Flow 1.0.6和Spring 2.5.4开发。未来的Spring Web Flow 1.x和Spring 2.x版本将兼容。第2章详细介绍了如何搭建开发环境，用于运行示例应用程序。

Spring银行示例应用程序和本书中提到的其他示例可以在Ervacon的Subversion存储库中找到：<https://svn.ervacon.com/public/spring>。你可以通过浏览器浏览源代码，也可以通过Subversion客户端来签出源代码。如果你使用Windows平台，TortoiseSVN (<http://tortoisesvn.tigris.org>) 是个不错的Subversion客户端。本书的源代码也可以从Apress出版社的网站<http://www.apress.com>获取。

致谢

本书和Spring Web Flow的存在得感谢很多人的大力帮助。

Spring Web Flow很幸运，从一开始就受到社区的积极支持。特别感谢早期的使用者和论坛的成员，他们无价的意见反馈帮助Spring Web Flow成长为今天的这个样子。要特别感谢好几个人积极参与这个项目，他们是Juergen Hoeller、Colin Sampaleanu、Rod Johnson、Ben Hale、Christian Dupuis、J. Enrique Ruiz、C_sar Ordiñana、Rossen Stoyanchev、Jeremy Grelle、Rob Harrop、Seth Ladd、Colin Yates、Steven Devijver、Greame Rocher、Sam Brannen、Maxim Petrashev、Marten Deinum以及Dave Syer。不用说，还有其他很多人也值得感谢。

写书是一项大工程，没有很多人的支持和反馈是无法完成的。我要感谢Keith Donald，感谢他对Spring Web Flow所做的很多贡献以及对本书的审阅工作。Karl David Moore甚至超越了审稿的职责，不仅相当彻底地审阅了很多章的内容，而且帮助我润色书中的文字。Apress出版社的一群优秀的人同样值得赞扬，他们让本书成为现在的样子，他们是Heather Lang、Laura Cheu和Kylie Johnston。还有许多人直接或间接地为本书作出了贡献，他们是Philip Van Bogaert、Henri Shih、Kris Meukens和Mike Seghers。在此感谢你们所有人。

我要特别感谢我的妻子Bieke。在我全身心地写作本书时，她不知疲倦地照顾着孩子们，承揽了很多本该是我做的事情。说句实在话，没有Bieke的支持，就不可能有这本书。

目 录

第 1 章 初识 Spring Web Flow.....	1	3.2.2 执行引擎.....	42
1.1 自由浏览	3	3.2.3 执行者.....	43
1.2 可控导航	4	3.2.4 测试支持.....	43
1.2.1 导航的控制.....	5	3.2.5 系统配置.....	44
1.2.2 状态管理.....	8	3.3 小结	44
1.2.3 模块化问题.....	10	第 4 章 Spring Web Flow 基础.....	45
1.3 传统解决方案.....	11	4.1 设计流	45
1.4 Spring Web Flow	14	4.1.1 UML 状态图	46
1.5 小结	16	4.1.2 第一个流定义.....	48
第 2 章 入门.....	17	4.2 流生成器	50
2.1 下载 Spring Web Flow	17	4.2.1 XML 流生成器.....	52
2.2 运行时需求.....	17	4.2.2 Java 流生成器.....	53
2.3 构建系统的集成.....	18	4.2.3 选择合适的流生成器.....	55
2.3.1 手工集成.....	18	4.3 定义流	56
2.3.2 与 Ivy 的集成	20	4.3.1 流	56
2.3.3 与 Maven 的集成.....	21	4.3.2 状态	57
2.3.4 Spring Jumpstart.....	21	4.3.3 转移	60
2.4 Hello World	24	4.3.4 流定义结构.....	63
2.5 开发环境中的 Spring Web Flow.....	28	4.4 流执行	63
2.5.1 安装 IDE	28	4.4.1 流会话.....	65
2.5.2 Eclipse 中的“Hello World”	30	4.4.2 流执行环境.....	66
2.5.3 使用 Spring IDE	32	4.4.3 请求环境.....	67
2.6 给 Hello World 增添些趣味.....	33	4.5 实现动作	72
2.7 小结	36	4.5.1 AbstractAction.....	73
第 3 章 Spring Web Flow 架构.....	37	4.5.2 MultiAction.....	74
3.1 语言	37	4.5.3 动作部署.....	76
3.2 架构层	39	4.6 基本状态类型.....	79
3.2.1 执行核心.....	40	4.6.1 视图状态.....	80
		4.6.2 动作状态.....	84

4.6.3 结束状态	85	6.2.4 请求处理	151
4.7 流定义注册表	86	6.2.5 配置流执行者	152
4.7.1 XML 流定义注册表	88	6.3 流执行存储库	155
4.7.2 Java 流定义注册表	91	6.3.1 对话管理	157
4.7.3 合并流定义注册表	91	6.3.2 简单存储库	159
4.8 小结	92	6.3.3 单键存储库	161
第 5 章 高级 Web 流概念	94	6.3.4 Continuation 存储库	162
5.1 OGNL	94	6.3.5 客户 Continuation 存储库	165
5.1.1 通过示例学习 OGNL	95	6.3.6 选择存储库	167
5.1.2 OGNL 实战	98	6.4 流执行监听器	167
5.2 转换服务	111	6.4.1 监听器调用示例	170
5.3 注解流定义工件	114	6.4.2 监听器配置	173
5.4 处理异常	115	6.5 小结	175
5.5 视图选择	119	第 7 章 驱动流执行	176
5.5.1 空字符串	120	7.1 流执行者集成	177
5.5.2 viewName	120	7.2 Spring Web Flow 视图开发	178
5.5.3 redirect:viewName	120	7.2.1 模型数据	178
5.5.4 externalRedirect:url	121	7.2.2 构建请求	180
5.5.5 flowRedirect:flowId?input1 =value&...&inputN=valueN	121	7.3 宿主框架集成	184
5.5.6 bean:id	122	7.3.1 Spring Web MVC	185
5.5.7 自定义视图选择器	122	7.3.2 Spring Portlet MVC	186
5.6 数据绑定和验证	123	7.3.3 Struts	187
5.7 子流	130	7.3.4 JSF	190
5.7.1 内联流	132	7.4 小结	199
5.7.2 重温流会话	132	第 8 章 Spring Web Flow 的测试	200
5.7.3 声明输入/输出契约	133	8.1 单元测试	200
5.7.4 映射输入输出参数	136	8.1.1 MockRequestContext	201
5.7.5 增强“输入支付款”流	138	8.1.2 MockRequestControlContext	201
5.8 流的开始动作和结束动作	140	8.1.3 MockExternalContext	201
5.9 完整的“输入支付款”流定义	141	8.1.4 MockParameterMap	201
5.10 小结	144	8.1.5 MockFlowExecutionContext	201
第 6 章 流执行管理	145	8.1.6 MockFlowSession	201
6.1 流执行存储库介绍	145	8.1.7 MockFlowServiceLocator	201
6.2 流执行者	147	8.1.8 使用 Mock 对象测试	202
6.2.1 启动流执行	148	8.2 流执行测试	203
6.2.2 恢复流执行	149	8.2.1 测试 Java 流定义	203
6.2.3 刷新流执行	150	8.2.2 测试 XML 流定义	204
		8.2.3 测试“输入支付款”流	206

8.2.4 使用流执行监听器	209	10.3 使用监听器	234
8.3 集成测试	211	10.3.1 安全化流	234
8.4 小结	213	10.3.2 全局后退转移	235
第 9 章 示例应用程序	214	10.3.3 面包屑	237
9.1 功能需求	214	10.4 负载测试和压力测试	238
9.2 下载和构建	215	10.5 Spring Web Flow 和 AJAX	240
9.3 领域模型	217	10.6 小结	241
9.4 应用程序搭建	219	第 11 章 扩展 Spring Web Flow	242
9.5 表现层	220	11.1 通用扩展点	242
9.5.1 Spring Web MVC 搭建	221	11.1.1 使用 Bean 引用	242
9.5.2 实现“输入支付款”用例	224	11.1.2 扩展流定义结构	244
9.5.3 国际化	227	11.1.3 定制流生成器	248
9.5.4 异常处理	228	11.2 数据库支持的对话管理器	249
9.5.5 页面布局	228	11.3 流 Servlet	255
9.6 小结	229	11.4 构建 Spring Web Flow	258
第 10 章 真实用例	230	11.5 小结	260
10.1 访问宿主环境	230	后记	261
10.2 流定义的参数化	232	参考文献	265

第1章

初识Spring Web Flow

企业级应用程序，尤其是Web应用程序，占有用Java语言开发的应用程序的一大部分。大多数Java开发者在他们的职业生涯中都或多或少开发过Java Web应用程序。因此，有大量所谓的MVC（模型-视图-控制器）Web开发框架可供选择不足为奇。比较著名的框架有：

- Apache Software Foundation的Struts框架（<http://struts.apache.org>）；
- Spring Web MVC，基于Spring框架的Web MVC框架（<http://www.springframework.org>）；
- WebWork，由OpenSymphony项目开发的一个Web框架（<http://www.opensymphony.com/webwork>）。Struts 2是在WebWork上发展起来的。WebWork的开发已经停止，它已融入了Struts 2中。

在过去的几年中，这些基于请求的传统框架受到了基于组件的Web MVC框架的巨大挑战，包括：

- JavaServer Faces (JSF)，该框架由JCP（Java Community Process）开发，并定义于JSR 127、JSR 252以及JSR 314中（<http://java.sun.com/javaee/jaserverfaces>），作为Java EE的标准规范；
- Tapestry，由Apache Software Foundation开发（<http://tapestry.apache.org>）。

基于请求的框架把HTTP请求作为“一等公民”来对待。通常，通过应用程序的动作或者控制器来完成请求的处理，处理完成后在新页面中显示处理的结果。而基于组件的框架则把HTTP请求抽象为对象并把应用程序的功能封装在可复用的组件中。这种方式和桌面应用程序中的GUI（graphical user interface，图形用户界面）工具集采用的方式非常类似。Web应用程序组件对事件做出响应，然后操作自己的内部状态，而将页面的显示交给控制框架。

尽管这两种类型的框架存在着这些不同点，但它们都采用了MVC这种设计模式来组织Web应用程序，因而可以很容易地理解和维护。MVC模式最早起源于Smalltalk社区，用于组织桌面应用程序的GUI。如今，事实证明它在Web应用开发中也非常有效。

MVC模式通过引入控制器作为中间协调者，试图将视图（用户界面）的相关问题从模型（领域或业务模型）中分离出来。在Web应用程序中，控制器将接收到的请求委托给业务组件（如服务层），并为要输出的视图准备好要显示的模型数据。视图通常是使用模板系统（如JSP或者Velocity）实现。

说明 Web应用程序中使用的MVC设计模式略有一定的变化,有时也被称作模型2架构或者Web MVC。在原始的MVC模式中,控制器并不是和视图联系在一起的。相反,视图是作为模型的观察者,用于接收模型改变时所发出的事件通知 (Gamma et al 1995)。

Web应用程序中的MVC模式实现,大多数都引入了另外一个组件:前端控制器(Fowler 2003)。前端控制器实质上是技术性的,通过良好定义的请求处理生命周期,它能够协调请求的处理:它将请求映射给特定的控制器,并呈现由该控制器指定的视图内容。前端控制器也可以管理一些常见问题,如安全和国际化等。著名的前端控制器的实现有Struts的ActionServlet、Spring MVC的Dispatcher Servlet以及JSF的FacesServlet。从应用程序的角度来看的话,控制器相当于一个驾驶位。一旦前端控制器为某一请求选择了适当的控制器,该控制器就会自行决定如何与业务组件交互,应当显示哪一个视图以及要显示哪一个模型数据等。这个过程可以用图1-1来说明。

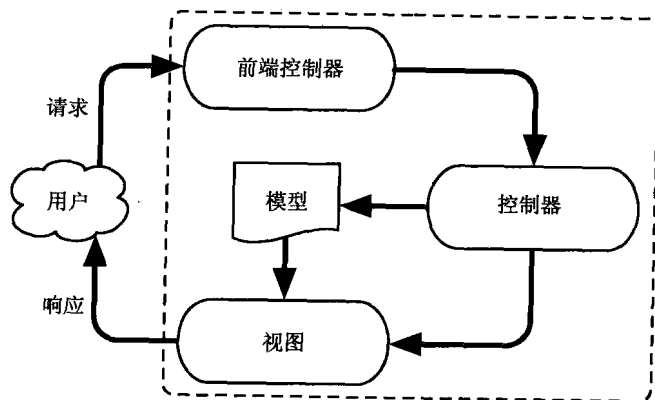


图1-1 Web MVC

Web MVC框架帮助我们开发有效的、结构化的、可维护的Web应用程序:应用在万维网上的软件程序。相对于静态的HTML网页提供静态数据而言,这样的应用可以动态地提供数据,如从数据库中读取数据等。

Web应用程序正在推动着Web技术的边缘。我们已经习惯于这些应用程序有复杂的对话,如跨越多请求的交互式对话框等。在线机票预订流程就是一个例子:当和这样的在线应用程序交互时,你需要跨越多个页面,如选择航班、指定机上座位、输入支付信息等页面。

基于MVC模式的Java Web应用开发是很好理解的。在Java世界中,关于某个框架的优缺点,总有非常激烈的争论。但基本而言,大多数争论者都承认,一些老的框架(如Struts)多年以来,已经成功地应用于很多产品的部署中。不过,虽然在这些框架的开发上,人们已经投入了如此多的精力,而且基于这些框架的应用的数量又是如此之多,但竟然没有一个主流的Java Web应用开发框架提供真正引人注目的方法来实现复杂对话功能,不得不让人惊讶。

Spring Web Flow旨在填补这个空白。作为Web应用程序控制器组件,它关注的重点就是Web应用程序中复杂对话的定义和执行。Spring Web Flow与其他的Web MVC框架集成起来,共同实

现这一个功能，而并不是它们的替代产品。它作为“MVC”中的“C”，即控制器，负责安排与业务组件的交互，准备模型数据，选择适当的视图等（Spring Web Flow作为控制器这一角色，在