

Broadview
www.broadview.com.cn

Java技术大系

Complete Reference Tapestry

深入浅出 Tapestry

董黎伟 编著
易为科技 审校

相关程序代码和
项目案例



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
HTTP://WWW.PHEI.COM.CN



Java技术大系

深入浅出 Tapestry

董黎伟 编著
易为科技 审校

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书以循序渐进的方式,从 Tapestry 框架技术的基本概念入手,讲解 Tapestry 框架在 J2EE Web 应用程序中的整体架构实现。使读者在学习如何使用 Tapestry 框架技术的同时,还能够获得在 J2EE Web 应用程序中应用 Tapestry 框架的先进经验。

本书详细介绍了 HiveMind 框架的原理与应用,使读者不但可以通过 HiveMind 来重构 Tapestry 的官方实现,还可以使用 HiveMind 来搭建 J2EE Web 应用程序的业务层。在本书还提供了—个以 Hibernate+HiveMind+Tapestry 4.0 为架构的 J2EE Web 应用程序示例,为读者提供—个完整的 J2EE Web 应用程序解决方案。本书还介绍了作者为 Tapestry 扩展的额外功能,即页面流程,详细阐述了该功能的作者思维与源代码实现过程,并提供了应用示例。

本书在介绍内容安排上,充分考虑到 Tapestry 对主流 Java 应用支持的功能。介绍了对 Spring、Hibernate、Ajax 等技术的融合。这样用 Tapestry 可以开发非常热门的 Web 应用。

为了方便读者在应用 Tapestry 框架时对各种知识点进行查阅,本书的各个章节相互独立,因此本书不但可以作为—部学习教程,也可以作为—本工具参考书。

本书适合 Web 框架技术的爱好者、Tapestry 入门学习者、Tapestry 表现层开发人员和关注下一代的 Web 开发技术的公司与个人用户。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

深入浅出 Tapestry / 董黎伟编著. —北京:电子工业出版社, 2007.3

(Java 技术大系)

ISBN 978-7-121-03739-9

I. 深… II. 董… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 165002 号

责任编辑:韩 明

印 刷:北京市通州大中印刷厂

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:24 字数:502 千字

印 次:2007 年 3 月第 1 次印刷

印 数:4000 册 定价:49.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话:(010) 68279077;邮购电话:(010) 88254888。

质量投诉请发邮件至 zlt@s@phei.com.cn,盗版侵权举报请发邮件至 dbqq@s@phei.com.cn。

服务热线:(010) 88258888。

Tapestry 颠覆了 J2EE Web 应用程序表现层技术的传统思维, 完全实现表现层技术的面向对象编程。Tapestry 作为下一代 Web 框架技术, 不仅仅是一个最有实力的 Web 应用框架, 同时也是最快速、最容易吸收的技术, 无论你是否拥有 Java、Perl、XML 或 PHP 技术背景。随着面向元素的 Web 框架技术逐渐被淘汰, Tapestry 作为最具实力的面向组件 Web 框架之一, 受到国外很多资深技术社区的推崇。

在国内, Tapestry 技术已经拥有比较庞大的应用群体, 越来越多的软件开发公司正在尝试此项技术, 以便借助 Tapestry 基于组件的开发方式实现最大程度的项目积累, 从而极大地提高开发效率。同时, 在开源论坛上, 对 Tapestry 的关注也非常多, 相关的讨论也很热烈。Tapestry 广泛应用的基础已经开始诞生。但是, 比较可惜的是, Tapestry 还没有中文的技术文档, 也没有相关的技术图书, 这给很多希望学习 Tapestry 的群体设置了一些障碍。而现在越来越多的开发企业与开发者已经开始关注 Tapestry 相关技术的应用, 对于开发经验的相关讲解更是热点。

作者已经有多年 Tapestry 开发的实践经验, 而且作者所在公司的 Web 项目也是基于 Tapestry 的。在应用了 Tapestry 技术之后, 作者充分体验到 Tapestry 给公司带来的开发效率的提升。同时, 作为一线的开发程序员, 一直以来都在为 Tapestry 的推广做一些力所能及的工作, 包括翻译了大量的开发文档, 同时在 Javaeye 论坛上成立专门的 Tapestry Wiki 中文社区, 并且通过 QQ、E-mail、论坛等交流方式与广大 Tapestry 爱好者分享作者的经验心得和自定义 Tapestry 组件。

为了推广 Tapestry, 作者尝试了许多方式, 但是效果均不理想。只有通过写书的方式, 不仅能够详细介绍 Tapestry 4.0 的基本应用方法, 还能够详细且系统地阐述积累多年的开发经验。本书作为国内第一本 Tapestry 图书, 努力做到让希望入门的读者能入门, 入了门的读者提高技术能力, 同时也希望能为 Java 优秀的开源技术的推广尽绵薄之力。

Tapestry 作为 Java 优秀的开源技术, 具有的优势是不言而喻的, 也不断带给开发人员意外的惊喜。为了更直观地理解, 将 Tapestry 的特点给罗列如下。

- Tapestry 的灵魂是组件重用, 随着项目开发越多, 组件积累得越多, 项目的效率就越高;
- 清晰明了的页面模板, 使得美工和表现层程序员能够实现完全分工;
- 由于所有页面逻辑全部在独立的 java 文件中, 因此项目的维护显得轻而易举;
- Tapestry 开发具有一致性, 不论项目大小, 不同的开发者都会在相似问题上找到相似的解决方式;
- 任何异常或错误都不会导致 Tapestry 框架本身发生错误。Tapestry 框架会将异常或错误信息以非常友好而详细的方式反馈出来;
- Tapestry 对底层进行了全面合理的封装, 使得开发者从乏味重复的底层代码开发中解脱出来, 极大地提高了工作效率;

- Tapestry 能够非常容易地与各种底层框架结合，包括 EJB 和 Spring；
- Tapestry 能够非常容易地创建新组件，就如同以常规的步骤来构建应用程序一样。官方提供了超过 50 种组件，这些组件包括各种各样简单的输出组件，以及非常复杂的表格组件和树型目录组件；
- Tapestry 框架是开源框架，允许开发者任意研究源代码，甚至重构官方源代码，以实现特殊功能；
- Tapestry 构建在 HiveMind 框架之上，且其源代码均是面向接口编程，具有非常良好的可扩展性；
- Tapestry 具有非常活跃的官方邮件列表，并获得了 theserverside 社区、apache wiki 社区、IBM developerworks 社区、Java 视线社区等国内外知名社区支持。

以上 Tapestry 框架的种种优势已经有足够的理由让我们放弃传统的 J2EE 表现层框架，而选择使用 Tapestry 框架。本书会将这些优势一一展现出来，我们将会发现，因为使用 Tapestry 框架而获得工作效率的提升是多么显而易见、激动人心。

本书特色

- 本书基于案例来讲解，为本书专门开发的配套项目案例让读者更容易登入 Tapestry 的开发殿堂。
- 代码是程序员的第二语言，本书以代码示例为先导，使得读者在学习理论的同时，掌握到 Tapestry 框架应用的实践经验。
- 本书不仅详细阐述 Tapestry 框架怎样使用，还会深入浅出地剖析 Tapestry 框架的原理。
- 本书的目标是将读者培养成为 Tapestry 框架的开发者，而并非仅仅是使用者。
- 本书不仅仅局限于为读者提供表现层技术的 Tapestry 框架解决方案，还为读者提供了 J2EE Web 应用程序的整体架构解决方案。
- 本书在介绍内容安排上，充分考虑到 Tapestry 对主流 Java 应用支持的功能，介绍了对 Spring、Hibernate、Ajax 等技术的融合。这样可以用 Tapestry 开发非常热门的 Web 应用。

适合的读者

本书适合 Web 应用开发的爱好者、Tapestry 入门学习者、Tapestry 表现层开发人员、关注下一代的 Web 开发技术的公司与个人，以及 Java 开源技术的推动者。

光盘内容

随书光盘包含 IceRoom 项目案例的源文件，相关的环境说明如下：

• IV •

1. Java JDK 5.0 以上版本;
2. Web 容器用 Resin3.0 以上版本;
3. IceRoom 项目案例使用 SQL Server2000 sp3 版本;
4. 开发的 IDE 使用 Eclipse 3.0 以上版本。

代码的相关配置说明, 在书中相应的章节已经说明, 请参考图书内容学习。

致谢

本书由董黎伟主笔。程小飞先生参与了本书内容的筛选, 并且给出指导性意见。姚新军先生参与本书的前期策划和后期的质量监控。参与本书校对和排版工作的有王斌、万雷、张赛桥、周忻、王晓艳、肖静、田元、刘军华、黄北军等。全书由成都易为科技有限责任公司审定。在本书写作期间, 还获得 Tapestry 的创始人 Mr.Howard 的鼓励和支持。感谢电子工业出版社辛勤工作的老师, 你们的付出才使本书能在第一时间和读者见面。本书前后花了近半年的时间完成写作, 在这期间获得家人和朋友的大力支持。对所有支持和关心本书的同仁在此一致谢过, 特别是好友王莉的支持和关心, 给了我非常大的鼓励。本书特意献给我亲爱的老爹老妈。

限于作者水平和精力有限, 疏漏之处难免, 欢迎读者批评指正。

编 者
2007 年 1 月

《深入浅出 Tapestry》读者调查表

尊敬的读者:

感谢您对我们的支持与爱护。为了今后为您提供更优秀的图书,请您抽出宝贵的时间将您的意见以下表的方式及时告知我们(可另附页)。我们将从中评选出热心读者若干名,免费赠阅我们以后出版的图书。

姓名:	性别: ☐ 男 ☐ 女	年龄:	职业:
通信地址:			邮政编码:
电话:	传真:	E-mail:	

1. 影响您购买本书的因素(可多选):

- ☐ 封面封底 ☐ 价格 ☐ 内容提要、前言和目录 ☐ 书评广告 ☐ 出版物名声
☐ 作者名声 ☐ 正文内容 ☐ 其他 _____

2. 您对本书的满意度:

- 从技术角度 ☐ 很满意 ☐ 比较满意 ☐ 一般 ☐ 较不满意 ☐ 不满意
 ☐ 改进意见 _____
- 从文字角度 ☐ 很满意 ☐ 比较满意 ☐ 一般 ☐ 较不满意 ☐ 不满意
 ☐ 改进意见 _____
- 从版面、封面设计角度 ☐ 很满意 ☐ 比较满意 ☐ 一般 ☐ 较不满意
 ☐ 不满意 ☐ 改进意见 _____

3. 您最喜欢书中的哪篇(或章、节)? 请说明理由。

4. 您最不喜欢书中的哪篇(或章、节)? 请说明理由。

5. 您希望本书在哪些方面进行改进?

6. 您感兴趣或希望增加的图书选题有:

通信地址: 北京万寿路 173 信箱 博文视点(100036) 电话: 010-51260888

如果您对我们出版的图书有任何意见和建议,也可以发邮件给我们,我们将及时回复。

E-mail: jsj@phei.com.cn, editor@broadview.com.cn

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

CONTENTS

第 1 章 搭建 Tapestry 运行环境..... 1	3.1.3 Tapestry 水果商店应用程序 项目配置 31
1.1 初步认识 Tapestry 2	3.1.4 Tapestry 页面组成结构 32
1.1.1 从 MVC 框架开始 2	3.1.5 水果商店的业务逻辑类 34
1.1.2 为什么要选择 Tapestry 3	3.1.6 登录页面 36
1.2 开发工具安装 4	3.1.7 水果列表页面 42
1.2.1 安装 JDK 4	3.1.8 水果图示页面 47
1.2.2 Eclipse 开发环境的搭建 5	3.1.9 购买水果页面 49
1.2.3 配置 Servlet 容器 6	3.1.10 已购买水果列表页面 57
1.2.4 Tapestry 框架支持包 9	3.1.11 Border 组件 60
1.3 Hello Tapestry 例程 9	3.2 Tapestry 属性指定 71
1.3.1 创建我的第一个 Tapestry 应用程序 10	3.3 Tapestry 属性的生命周期 73
1.3.2 Hello Tapestry 实例 11	3.4 监听方法 76
1.3.3 让 Hello Tapestry 运行起来 13	3.5 Visit 对象和 Global 对象 77
第 2 章 Tapestry 框架概述 15	3.5.1 Visit 对象 77
2.1 什么是面向组件开发方式 16	3.5.2 Global 对象 78
2.1.1 传统的面向元素开发方式 16	3.6 页面类和页面事件 79
2.1.2 Tapestry 的面向组件开发方式 17	3.6.1 页面类 79
2.1.3 面向元素 VS 面向组件 ——思维方式的变革 18	3.6.2 页面事件 80
2.2 Tapestry 框架概述 19	3.7 Form 组件原理 81
2.2.1 Tapestry 框架的配置文件 19	3.7.1 HTML form 与 Tapestry Form 81
2.2.2 Tapestry 页面的组成 22	3.7.2 Form 表单组件的基本应用 84
2.2.3 Tapestry 组件的组成 23	3.7.3 Form 表单组件运行原理 86
2.3 OGNL 25	3.7.4 表单校验 89
2.3.1 OGNL 概述 25	3.7.5 校验对象 90
2.3.2 OGNL 表达式规则 25	3.7.6 校验代理 93
第 3 章 Tapestry 框架基础应用 28	3.7.7 自定义校验代理 97
3.1 快速上手 29	3.7.8 自定义校验对象 99
3.1.1 Tapestry 水果商店项目介绍 29	3.7.9 客户端校验 101
3.1.2 Tapestry 水果商店应用程序 流程图 30	3.7.10 表单输入转换 102
	3.7.11 失效链接 103
	3.8 Tapestry 的 service 106
	3.8.1 Tapestry 用于 Servlet 的

底层对象	106	5.3.3 带有表单校验的自定义 表单组件	156
3.8.2 理解应用程序 Servlet	108	5.3.4 带有表单输入转换功能的 自定义表单组件	160
3.8.3 Servlet 请求处理	109	5.4 Tapestry 动态脚本	162
3.8.4 理解服务器端状态	110	5.4.1 带有动态脚本的自定义 组件	163
3.8.5 在一个集群中管理服务器 端状态	111	5.4.2 Tapestry 动态脚本的语法	168
3.8.6 理解 Tapestry 引擎	112	5.5 自定义 service	171
3.8.7 传统应用程序的 URL 有什么问题?	113	5.6 创建自己的 Ajax 组件	177
3.8.8 Tapestry 怎样操作应用 程序的运作?	114	5.6.1 创建自己的 Ajax 组件	178
3.8.9 理解 home service	114	5.6.2 在水果商店示例中创建演示 页面	179
3.8.10 理解 page service	116	5.6.3 Tapestry 官方的 PropertySelection 组件	181
3.8.11 理解 direct service	117	5.6.4 创建联动下拉菜单组件	184
3.8.12 理解 external service	123	第 6 章 Tapestry 对 J2SE 5.0 的支持	197
3.8.13 理解 ResponseRender	125	6.1 J2SE 5.0 的新特性 Annotation	198
第 4 章 Tapestry 框架配置	127	6.1.1 Annotation 介绍	198
4.1 本地化 i10n	128	6.1.2 Annotation 与 Tapestry 的 property 属性注入	201
4.1.1 Java 实现国际化	128	6.2 Tapestry 中 Annotations 的使用	201
4.1.2 Tapestry 的本地化	130	6.2.1 Asset	202
4.2 应用程序规范文件的配置	132	6.2.2 Bean	202
4.2.1 应用程序规范文件的标签 元素定义	133	6.2.3 Component	203
4.2.2 应用程序配置	135	6.2.4 InitialValue	203
4.3 Hivemodule 配置	136	6.2.5 InjectObject	203
4.3.1 HiveMind 配置文件	137	6.2.6 InjectPage	204
4.3.2 调用 HiveMind 的 service	137	6.2.7 InjectState	204
4.3.3 Friendly URL	138	6.2.8 InjectScript	204
4.4 重定义异常页面	140	6.2.9 Persist	205
4.5 子项目分包与页面分包	142	6.2.10 Parameter	205
第 5 章 创建自定义组件	148	第 7 章 HiveMind 框架	207
5.1 组件的种类	149	7.1 HiveMind 框架概述	208
5.2 组件的相关配置	149		
5.3 单一组件	150		
5.3.1 最简单的自定义普通组件	150		
5.3.2 最简单的自定义 Form 表单组件	152		

7.2	HiveMind 框架应用	211	8.5	项目需求分析	250
7.2.1	HiveMind 框架基础应用	211	8.5.1	用户权限管理	250
7.2.2	HiveMind 配置文件	212	8.5.2	论坛	255
7.2.3	启动 HiveMind 框架	212	8.5.3	软件上传下载	260
7.2.4	声明 service-id 并注入 service 实例	213	8.5.4	贴图上传及评论	261
7.2.5	HiveMind 的 BuildFactory 详解	216	8.5.5	RAID 活动管理	264
7.2.6	Configuration 简介	221	8.5.6	行会公告、冰羽轩制度、 DKP 制度	270
7.2.7	HiveMind 拦截器	228	8.6	冰羽轩项目的结构	272
7.2.8	HiveMind 如何与 Hibernate 融合	228	8.7	冰羽轩的论坛功能实现	279
7.2.9	HiveMind 如何实现 AOP 编程	232	8.7.1	ForumList 页面	280
7.2.10	Tapestry 如何通过 HiveMind 与 Spring 融合	235	8.7.2	Forum 页面	286
8	Tapestry 应用实践		8.7.3	ForumTopic 页面	296
8.1	关于项目案例的说明	238	第 9 章	Tapestry 底层探秘	304
8.2	架构的选择	239	9.1	Tapestry 源代码结构	305
8.2.1	认识 B/S 模式	240	9.2	Tapestry 如何实现页面 之间对象传递	307
8.2.2	三层 J2EE 架构的选择	240	9.3	Tapestry 属性的持久化 策略	311
8.2.3	Hibernate 框架简介	241	第 10 章	为 Tapestry 添加一个 翅膀 TWF	318
8.2.4	架构的整合	241	10.1	解析 XML 配置文件	319
8.3	HibernateSynchronizer 插件	242	10.1.1	bean-xml 配置文件	319
8.3.1	创建 Hibernate 配置文件	242	10.1.2	flow-xml 配置文件	322
8.3.2	创建 Hibernate 映射文件	244	10.1.3	页面流程流转的过程	324
8.3.3	创建 BO 和 DAO	246	10.1.4	TWF 在 Tapestry 中的应用	328
8.4	安装冰羽轩	248	10.2	TWF 与 Tapestry 融合	329
			附录 A	常见问题解答 (FAQ)	335
			附录 B	官方基础组件	344

第 1 章

搭建 Tapestry 运行环境

本章内容包括：

- ✎ 安装 JDK
- ✎ Eclipse
- ✎ Servlet 容器
- ✎ Tapestry 框架支持包
- ✎ 创建我的第一个 Tapestry 应用程序
- ✎ 编写 Hello Tapestry 实例
- ✎ 让 Hello Tapestry 运行起来

古人云：“工欲善其用，必先利其器”。所以我们首先将介绍如何搭建开发工具，再介绍如何搭建 Tapestry 应用程序。搭建 Tapestry 运行环境作为开发的初始步骤，一般包括 JDK、Eclipse、Servlet 容器，以及相关框架支持包的安装等。本章从简单的“Hello Tapestry”例程开始，让我们对 Tapestry 运行有一个直观地认识。

1.1 初步认识 Tapestry

Tapestry 是一个使用 Java 语言编写的开源框架，用于创建动态的、健壮的、高灵活性的 Web 应用程序。Tapestry 框架构筑在标准的 Java Servlet API 之上，因此它能够很好地兼容任何 Servlet 容器或者应用服务。Tapestry 和 HiveMind 框架是由 Howard Lewis Ship 开发出来的。我们可以登录 Howard 的个人主页 <http://howardlewisship.com>，了解更多信息。

Tapestry 是属于 MVC 框架的范畴，我们下面将从 MVC 框架开始，初步认识 Tapestry。

1.1.1 从 MVC 框架开始

MVC 是模型 (Model) ~视图 (View) ~控制器 (Controller) 的简称，它是 20 世纪 80 年代在 Smalltalk 编程语言中出现的一种软件设计模式。MVC 将应用程序的输入、处理和输出分开成为相互独立的 3 个核心层次：Module、View、Controller。

Model 用于表示企业数据和业务规则。在 MVC 的 3 个层次中，Model 层处理的业务最多。Model 层可能用像 EJB、Spring、HiveMind 这样的中间件来构建对象处理数据库中的数据，并返回处理业务逻辑之后的数据。这些数据是中立的，也就是说，Model 与数据格式无关，因此一个 Model 可以向多个 View 提供数据。

View 是用户看到的应用程序界面，用户通过操作界面的功能来与应用程序进行交互。对于大多数 Web 应用程序来说，View 就是由 HTML 元素组成的界面。不过现在一些新技术层出不穷，View 也指由包括 Flash、XML/XSL、WML 等技术实现的用户界面。

Controller 用于接受用户在 View 的输入，并调用 Model 和 View 去完成用户的业务要求。比如当用户在 Web 页面上单击一个链接或提交一个表单的时候，Control 只是接收 View 发出的请求并决定调用哪个 Model 去响应请求，最后确定用哪个 View 去向客户展现 Model 处理业务逻辑之后返回的数据。Controller 只是起一个桥梁的作用，它本身并不处理任何业务逻辑或输出任何数据。

MVC 是一种设计模式，它将“数据从何而来”与“数据如何显示”这两个代码层次分开，这样将带来如下好处。

- 多个 View 能够共享一个 Model，即有许多个 View 所显示的数据都可能出自同一个数据库中的同一张表，那么我们只需要编写一个 Model 就可以满足多个 View 显示数据的需要，使我们能够尽可能多地重用代码；
- 由 Model 返回的数据是没有格式化的，因此能够满足多个 View 的显示需要。比如一个 View 可能使用 HTML 来显示界面，另一个 View 可能使用 Flash 来显示界面；
- 即便我们的数据库从 SQL Server 移植到 MySQL，只是改变 Model 而不会改变 View，方便我们进行重构和维护；

- 由于 Controller 是独立的，我们也可以通过 Controller 去改变 Model 与 View 之间的连接关系，从而可以根据用户需求来选择不同的 Model 处理数据，并通过不同的 View 展现数据。

MVC 的优点显而易见，但是它也是有缺点的。因为 MVC 没有明确定义各个层次的界限，因此要在应用程序中实现 MVC 设计模式，必须对 MVC 有深刻的理解并具备非常丰富的项目架构经验，所以我们不得不耗费大量的精力去分析如何将 Model、View、Controller 进行严格的分离，而其中任何一点设计缺陷都可能导致我们的应用程序在实现、维护和重构上带来很大的困难。为了帮助绝大多数功力不够深厚的程序员和团队，很多资深的程序员或开发团队组织起来，共同开发了一些 MVC 框架。这些框架本身也是应用程序，我们通过使用这些应用程序来开发我们自己的应用程序，使我们不用过多地去考虑 MVC 的设计模式实现，从而提高我们应用程序的健壮性和可伸缩性。我们甚至可以认为这些 MVC 框架是一种工具，通过使用这些工具，我们可以避免一些对开发经验要求极高的结构设计，从而降低不必要的甚至是致命的风险。

常用的 MVC 框架有 Tapestry、Struts、WebWork2、Spring MVC、JSF 等。它们都有各自的优点和缺点。

1.1.2 为什么要选择 Tapestry

Tapestry 将一个 Web 应用程序分离为一组页面，而每个页面又由多个组件组成。这就提供了一个协调一致的结构，使得 Tapestry 框架能够专注于任何一个关键细节，例如，URL 解析和派遣、持久化状态储存在客户端还是储存在服务器端、用户输入校验、本地化或国际化和异常报告。对 Tapestry 应用程序的开发包括使用普通的 HTML 标签创建 HTML 模板，然后使用 XML 描述文件（可选）将这些模板与少量的 Java 代码结合在一起。在 Tapestry 中，创建应用程序是通过使用对象、方法和对象的属性，而并非 URLs 和查询参数。Tapestry 带来了真正的 Java Web 应用程序 OOP 编程。

Tapestry 能够非常容易地创建新组件，就如同以常规的步骤来构建应用程序一样。而官方提供了超过 50 种组件，这些组件包括各种各样简单的输出组件，以及非常复杂的表格组件和树型目录组件。

Tapestry 能够构建从小型的应用程序到由各种大型团队开发的具有几百个独立页面的大型应用程序。Tapestry 能够非常容易地与各种底层框架结合，包括 J2EE、Hivemind 和 Spring。

Tapestry 发布于 Apache Software Licence 2.0。现在 Tapestry 是 Apache 下的一级项目，称为 Apache Tapestry。

前面我们介绍了 MVC 框架比较多，那我们为什么要选择 Tapestry 呢？

- Tapestry 颠覆了 J2EE Web 应用程序表现层技术的传统思维，完全实现了表现层技术的面向对象编程；
- Tapestry 的灵魂是组件重用，项目开发越多，组件积累得越多，项目开发的效率就越高；

- 清晰明了的页面模板，使得美工和表现层程序员能够实现完全分工；
- 由于所有页面逻辑全部在独立的 Java 文件中，因此项目的维护显得轻而易举；
- Tapestry 开发具有一致性，不论项目大小，不同的开发者都会在相似问题上找到相似的解决方式；
- 任何异常或错误都不会导致 Tapestry 框架本身发生错误。Tapestry 框架会将异常或错误信息以非常友好而详细的方式反馈出来；
- Tapestry 对底层进行了全面合理的封装，使得开发者从乏味重复的底层代码开发中解脱出来，极大地提高了工作效率；
- Tapestry 能够非常容易地与各种底层框架结合，包括 EJB 和 Spring；
- Tapestry 能够非常容易地创建新组件，就如同以常规的步骤来构建应用程序一样。而官方提供了超过 50 种组件，这些组件包括各种各样简单的输出组件，以及非常复杂的表格组件和树型目录组件；
- Tapestry 框架是开源框架，允许开发者任意研究源代码，甚至重构官方源代码，以实现特殊功能；
- Tapestry 构建在 Hivemind 框架之上，且其源代码均是面向接口编程，具有非常好的可扩展性；
- Tapestry 具有非常活跃的官方邮件列表，并获得了 TheServerSide 社区、Apache wiki 社区、IBM developerworks 社区、Java 视线社区等国内外知名社区支持；

Tapestry 框架的种种优势，让我们有足够的理由放弃传统的 J2EE 表现层框架，而选择 Tapestry 框架。本书会将这些优势——展现出来，我们将会发现，因使用 Tapestry 框架而获得工作效率的提升是多么显而易见、激动人心。

1.2 开发工具安装

Tapestry 的开发工具有很多种，现在用的比较多的是开源的 Eclipse，下面我们将介绍如何使用 Eclipse 来搭建开发环境。一般来说，在构建开发环境时，我们会先安装 JDK、安装 IDE 工具（如 Eclipse 或 JBuilder）、Servlet 容器、相关的框架支持包等。下面的内容将会详细介绍相关工具的安装与配置，搭建 Tapestry 的开发环境。

1.2.1 安装 JDK

Tapestry 框架是使用纯 Java 语言编写、构建在标准 Java Servlet API 之上的开源框架。Tapestry 框架本身运行于 JDK 1.3 版本。因此只要我们安装了 1.3 版本或更高版本的 JDK，就可以支持 Tapestry 框架良好运行。

在 Tapestry 官方网站 http://jakarta.apache.org/site/downloads/downloads_tapestry.cgi，我们可以下载 Tapestry 框架最新的压缩包。其中包含了 4 个 Java 包文件。

- tapestry-4.0.jar
- tapestry-annotations-4.0.jar

- `tapestry-contrib-4.0.jar`
- `tapestry-portlet-4.0.jar`

除了 `tapestry-4.0.jar` 以外,其他 3 种包文件都属于扩展包。`tapestry-4.0.jar` 包是 Tapestry 框架的主程序包,里面提供 Tapestry 框架的所有功能和基本组件。为了使用 Tapestry 框架,我们的应用程序必须加载 `tapestry-4.0.jar` 包。

为了支持最新的 J2SE 5.0 版本的新功能, Tapestry 框架提供了 `tapestry-annotations-4.0.jar` 包用于支持 J2SE5.0 的 Annotation 功能。因此,除非我们需要使用 J2SE 5.0 的 Annotation 功能更快捷地编写 Tapestry 框架的页面类,否则可以不加载这个包文件。

`tapestry-contrib-4.0.jar` 是官方扩展组件包。将若干 Tapestry 组件连同一个组件包规范文件一起打包为一个 jar 文件,可以方便我们跨项目地积累 Tapestry 组件或者向第三方提供组件。通过在应用程序规范文件中配置组件包路径, Tapestry 框架可以非常方便地调用组件包中的组件。`tapestry-contrib-4.0.jar` 官方扩展组件包中的组件,是官方直接或间接对基础组件已有功能的扩展。如果我们的 Tapestry 应用程序中没有使用到这些扩展组件,那么可以不加载这个包文件。

在 Tapestry 4.0 版本,提供了对 JSR-168 Portlets 的支持。JSR-168 Portlets 是 Portlets 和 Portals 之间交互的规范,有兴趣的读者可以在 <http://www.jcp.org/en/jsr/detail?id=168> 查询到该规范的详细定义。所以,如果不需要支持 Portlets,我们不用下载 `tapestry-portlet-4.0.jar` 包文件。

一般来说,安装目前流行的 JDK 1.4 版本就够用了。如果我们理解 J2SE 5.0 的 Annotation 功能,而且在开发过程中需要用到 Annotation,那么就需要安装 Annotation JDK 1.5 版本。因为 Tapestry 框架的 Annotation 使用起来更方便,能够减少代码量,而且 J2SE 5.0 本身的新特性能够加快应用程序的开发效率,提高应用程序的运行效率。

各个版本的 JDK 可以在 http://www.java.com/zh_CN/ 下载。

1.2.2 Eclipse 开发环境的搭建

Eclipse 是目前比较流行的开发工具,另外用的比较广泛的还有 JBuilder。实际上, Tapestry 框架作为 Web 表现层框架,只要能够搭建 J2EE Web 应用程序,使用什么开发工具并不重要。我们甚至可以不使用任何工具,手工搭建应用程序。应用 Eclipse 能够提高我们的开发效率,所以我们一般用 Eclipse 来辅助开发。

本书中的应用程序都使用 Eclipse 进行搭建,版本为 3.1.0。我们可以在 <http://www.eclipse.org/downloads/> 下载各种不同版本的 Eclipse SDK。

Eclipse 是一个开源免费的开发工具,通过安装不同的插件,可以非常容易地扩展 Eclipse 的功能,因此 Eclipse 受到越来越多程序员和开发团队的欢迎。

Eclipse 不需要安装,从官方下载到 Eclipse SDK 压缩文件包之后,将压缩文件包中的文件解压到硬盘上,然后单击解压目录中的 `eclipse.exe` 文件即可启动 Eclipse 开发工具。

启动 Eclipse 之后,我们首先需要配置 Java 的 JRE 路径,方可使用 Eclipse 进行开发工作。在 Eclipse 中执行【Window】/【Preferences】命令,如图 1-1 所示。弹出【Preferences】

对话框，展开【Java】/【Installed JREs】选项，然后在右边窗口配置 JRE，如图 1-2 所示，打开 Eclipse 配置窗口并配置 JRE。

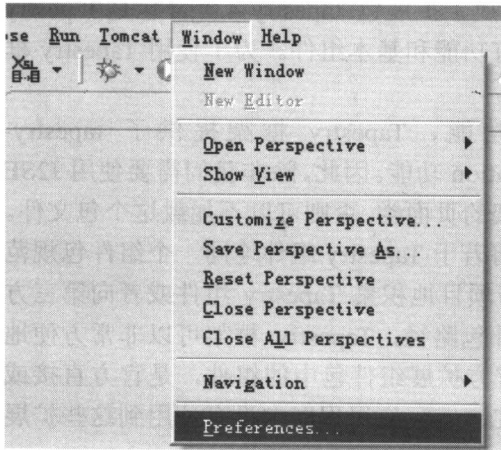


图 1-1 打开 Eclipse 配置窗口

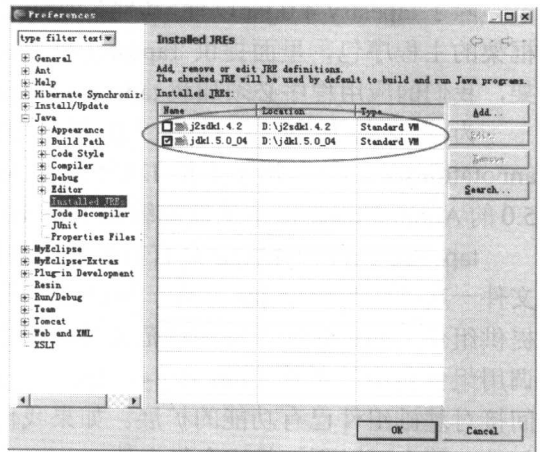


图 1-2 配置 JRE

1.2.3 配置 Servlet 容器

Tapestry 框架构建在标准的 Java Servlet API 之上，因此能够很好地兼容任何 Servlet 容器和应用服务。目前比较流行的 Servlet 容器有 Tomcat、Resin、Jetty 等。我们根据应用程序的需求和部署环境，选择不同的 Servlet 容器。而 Tapestry 框架本身却与 Servlet 容器毫无关系。

本书中的所有示例使用 Resin 作为 Servlet 容器，版本号为 3.0.14。我们可以在 <http://www.caucho.com/download/index.xtp> 下载 Resin-3.0.14 的压缩包文件，将下载后的压缩包解压到硬盘上即可使用。

在 <http://www.improve-technologies.com/alpha/updates/site.xml> 可以下到 Eclipse 开发工具的 Resin 插件，该插件能够让我们在 Eclipse 中更方便快捷地使用 Resin，目前该插件只支持 Eclipse 3.1.0 版本。整个配置过程如下。

步骤 1 启动 Eclipse 工具，执行【Help】/【Software Updates】/【Find and Install】命令，如图 1-3 所示。在弹出的【Install / Update】窗口，单击【Search for new features to install】单选按钮，单击【Next】按钮，如图 1-4 所示，开始搜索 Resin 插件。

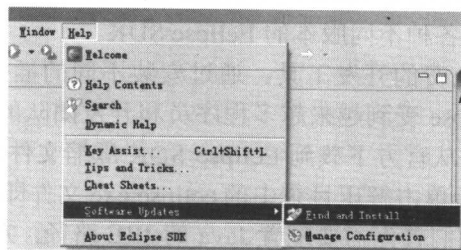


图 1-3 开启 Eclipse 插件安装功能