

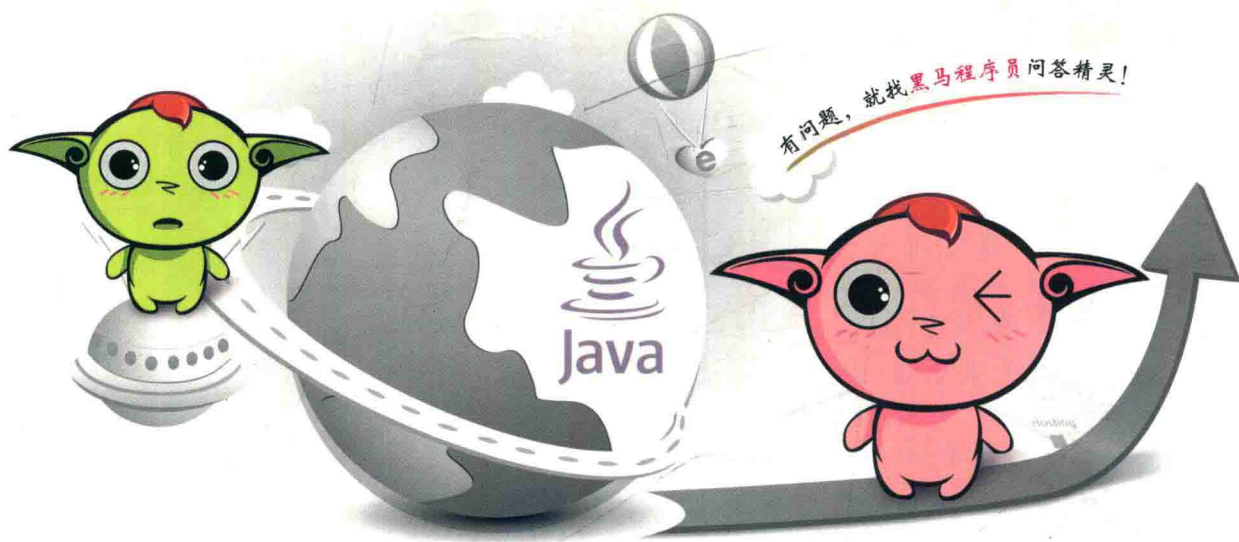


工业和信息化“十三五”
人才培养规划教材

NITE 国家信息技术紧缺人才培养工程
指定教材

Java EE 企业级 应用开发教程 (Spring+Spring MVC+MyBatis)

黑马程序员 / 编著



本书共 18 章，涵盖了轻量级 Java EE 框架 Spring+Spring MVC+MyBatis 的实际应用技术，每个知识点都配备了实战案例，共 36 道思考题。

提供免费教学资源，包含 18 个精美教学 PPT、1 个客户管理项目、1000 道测试题、长达 35 小时的教学视频等。

添加 QQ 或微信号 208695827，获取教学答案、源码和“助学红包”。

中国工信出版集团

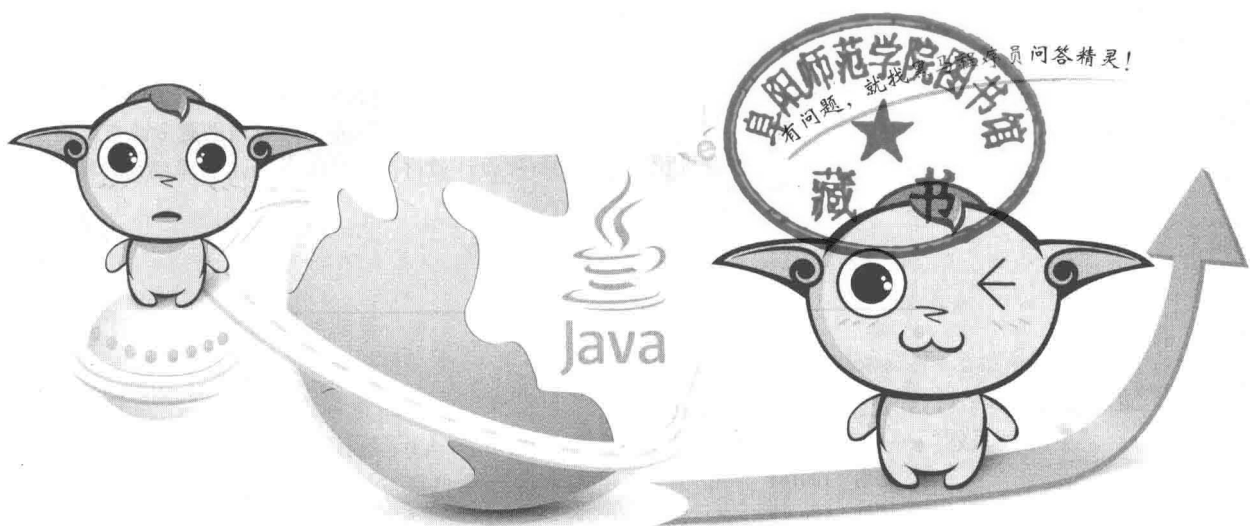
人民邮电出版社
POSTS & TELECOM PRESS

信息化“十三五”
规划教材

NITE 国家信息技术紧缺人才培养工程
指定教材

Java EE企业级 应用开发教程 (Spring+Spring MVC+MyBatis)

黑马程序员 / 编著



人民邮电出版社
北京

图书在版编目 (C I P) 数据

Java EE企业级应用开发教程 : Spring+Spring MVC+MyBatis / 黑马程序员编著. — 北京 : 人民邮电出版社, 2017.9

工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-46102-5

I. ①J… II. ①黑… III. ①JAVA语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第189003号

内 容 提 要

本书详细讲解了 Java EE 中 Spring、Spring MVC 和 MyBatis 三大框架 (以下简称“SSM”) 的基本知识和应用。本书在对知识点进行描述时采用了大量案例, 可以更好地帮助读者学习和理解 SSM 的核心技术。

本书共 18 章, 第 1~5 章主要讲解 Spring 的基本知识和应用, 其中包括 Spring 的基本应用、Spring 中的 Bean、Spring AOP、Spring 的数据库开发以及 Spring 的事务管理。第 6~10 章主要讲解了 MyBatis 的相关知识, 其中包含初识 MyBatis、MyBatis 的核心配置、动态 SQL、MyBatis 的关联映射以及 MyBatis 与 Spring 的整合。第 11~17 章主要讲解了 Spring MVC 的相关知识, 其中包含 Spring MVC 入门, Spring MVC 的核心类和注解, 数据绑定, JSON 数据交互和 RESTful 支持, 拦截器, 文件上传和下载以及 SSM 框架整合。第 18 章讲解整个 SSM 框架的总结与综合运用, 全章通过一个 BOOT 客户管理系统案例, 贴近实际地讲解了开发中 SSM 框架的应用。读者掌握了 SSM 框架技术, 就能很好地适应企业开发的技术需求, 为大型项目的开发奠定基础。

本书附有配套视频、源代码、测试题、教学 PPT、教学设计、教学大纲等配套资源。为了帮助初学者及时地解决学习过程中遇到的问题, 本书还提供了在线答疑平台, 希望可以帮助更多读者。

本书既可作为高等院校本、专科计算机相关专业的程序设计教材, 也可作为 Java 技术的培训图书, 适合广大编程爱好者阅读与使用。

◆ 编 著 黑马程序员

责任编辑 范博涛

责任印制 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>

三河市海波印务有限公司印刷

◆ 开本: 787×1092 1/16

印张: 21

2017年9月第1版

字数: 525千字

2017年9月河北第1次印刷

定价: 49.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

江苏传智播客教育科技有限公司(简称传智播客)是一家致力于培养高素质软件开发人才的科技公司,“黑马程序员”是传智播客旗下高端IT教育品牌。

“黑马程序员”的学员多为大学毕业后,想从事IT行业,但各方面条件还不成熟的年轻人。“黑马程序员”的学员筛选制度非常严格,筛选制度包括了严格的技术测试、自学能力测试,还包括性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保了学员质量,并降低了企业的用人风险。

自“黑马程序员”成立以来,教学研发团队一直致力于打造精品课程资源,不断在产、学、研3个层面创新自己的执教理念与教学方针,并集中“黑马程序员”的优势力量,有针对性地出版了计算机系列教材50多册,制作了教学视频数十套,发表各类技术文章数百篇。

“黑马程序员”不仅斥资研发IT系列教材,还为高校师生提供以下配套学习资源与服务。

为大学生提供的配套服务:

1. 专业的辅助学习平台“博学谷”(<http://yx.boxuegu.com>), 专业老师在线为您解答。
2. 针对高校学生在学习过程中存在的压力等问题,我们还面向大学生量身打造了“播妞”。播妞不仅致力推行快乐学习,还会有定期的助学红包雨。同学快来添加播妞微信/QQ: 208695827。
3. 高校学生也可扫描下方二维码,加入播妞粉丝团,获取最新学习资源,与播妞一起快乐学习。



为IT教师提供的配套服务:

针对高校教学,“黑马程序员”为IT系列教材精心设计了“教案+授课资源+考试系统+题库+教学辅助案例”的系列教学资源,高校老师可关注码大牛老师微信/QQ: 2011168841,获取配套资源,也可以扫描下方二维码,加入专为IT教师打造的师资服务平台——“教学好助手”,获取“黑马程序员”最新教师教学辅助资源相关动态。



为什么要学习本书

当前轻量级 Java EE (Java 企业版) 应用开发通常会采用两种方式: 一种是以 SSH (Struts + Spring+Hibernate) 框架为核心的组合方式, 另一种是以 SSM (Spring+Spring MVC+ MyBatis) 框架为核心的组合方式。使用这两种组合方式的项目都使 Java EE 架构具有高度的可维护性和可扩展性, 同时极大地提高了项目的开发效率, 降低了开发和维护的成本, 因此, 这两种组合方式已成为当前各个企业项目开发的首选。

两种组合框架的相同点在于都以 Spring 框架为核心, 而两者的主要不同之处在于 MVC 的实现方式 (Struts 与 Spring MVC), 以及 ORM 持久化方面 (Hibernate 与 Mybatis)。SSH 较注重配置开发, 其中的 Hibernate 对 JDBC 的完整封装更加面向对象, 对增、删、改、查的数据维护更自动化, 但 SQL 优化方面较弱, 且学习门槛稍高; SSM 更注重注解式开发, 且 ORM 实现更加灵活, SQL 优化更简便, 学习容易入门。目前来说, 传统企业项目的开发, 使用 SSH 框架比较多, 而对性能要求较高的互联网项目, 通常会选用 SSM 框架。因此, 对于想从事互联网项目开发的人员来说, 学好 SSM 框架, 就显得比较重要了。

如何使用本书

本书适用于具有 Java 基础和一定的 Java Web 相关知识的读者学习, 对于没有任何基础的读者, 建议先学习本套系教材中的《Java 基础案例教程》和《Java Web 程序设计任务教程》。

本书在 Spring 4.3 + Spring MVC 4.3 + MyBatis 3 版本的基础上, 详细讲解了这三大框架的基础知识和使用方法。在编写时, 作者力求将一些非常复杂、难以理解的问题简单化, 使读者能够轻松理解并快速掌握这些知识点。同时, 本书还对每个知识点都进行了深入的分析, 并针对重要知识点精心设计了案例, 以提高读者的实践操作能力。

全书共分为 18 个章节, 接下来分别对每个章节的内容进行简单的介绍, 具体如下。

- 第 1 章讲解了 Spring 框架入门的一些基础知识, 主要内容包括 Spring 框架的概念、作用、优点、体系结构、下载和使用、核心容器、入门程序以及依赖注入等。
- 第 2 章对 Spring 中的 Bean 进行了详细讲解, 主要内容包括 Bean 的配置、Bean 实例化的三种方式、Bean 的作用域、Bean 的生命周期以及 Bean 的三种装配方式。
- 第 3 章讲解了 Spring 框架中 AOP 的相关知识, 主要内容包括 AOP 的介绍、动态代理、基于代理类的 AOP 实现以及如何使用 AspectJ 框架来进行 AOP 开发等。
- 第 4 章对 Spring 的数据库开发进行了详细讲解。主要包括 Spring JDBC 中的核心类和配置的介绍, 以及 Spring JDBCTemplate 的常用方法。
- 第 5 章对 Spring 中的事务管理进行了详细讲解, 主要内容包括 Spring 事务管理的核心接口、事务管理的方式, 以及基于 XML 方式和基于 Annotation 方式的声明式事务处理的使用。

- 第 6 章对 MyBatis 框架的基础知识进行了讲解, 主要内容包括 MyBatis 框架的概念, 特点, 下载和使用, 工作原理以及一个简单的查、增、改、删案例。
- 第 7 章对 MyBatis 的核心配置进行了详细讲解。主要包括 MyBatis 中的两个重要核心对象 SqlSessionFactory 和 SqlSession, 以及 MyBatis 配置文件和映射文件的详细讲解。
- 第 8 章对 MyBatis 框架的动态 SQL 知识进行了详细讲解, 主要内容包括常用的动态 SQL 元素介绍以及常用动态 SQL 元素的使用。
- 第 9 章对 MyBatis 框架中的关联映射知识进行了详细讲解, 主要内容包括关联关系中的一对一、一对多和多对多的处理。
- 第 10 章对 MyBatis 与 Spring 框架的整合使用进行了详细讲解, 主要内容包括整合的环境搭建、传统 DAO 开发方式的整合和基于 Mapper 编程方式的整合。
- 第 11 章对 Spring MVC 框架的入门知识进行了详细讲解, 主要内容包括 Spring MVC 的介绍、入门程序的编写以及 Spring MVC 框架的工作流程。
- 第 12 章对 Spring MVC 的核心类及其相关注解的使用进行了详细的讲解, 主要内容包括对前端控制器 DispatcherServlet 的作用和配置的介绍, @Controller 注解和 @RequestMapping 注解类型的使用, 以及视图解析器的定义和配置。
- 第 13 章对 Spring MVC 中的数据绑定知识进行了详细讲解, 主要内容包括数据绑定介绍、简单数据绑定和复杂数据绑定。
- 第 14 章对 Spring MVC 中的 JSON 数据交互和 RESTful 支持进行了详细的讲解, 主要内容包括 JSON 的介绍、Spring MVC 中的 JSON 数据交互以及 RESTful 支持的使用。
- 第 15 章对 Spring MVC 中拦截器的使用进行了详细讲解, 主要内容包括拦截器的定义和配置, 单个拦截器和多个拦截器的执行流程, 以及拦截器的实际应用。
- 第 16 章对 Spring MVC 环境下的文件上传和下载进行了详细讲解, 主要内容包括如何实现文件上传和下载, 以及如何实现中文名称文件的下载。
- 第 17 章对 SSM 框架的整合知识进行了详细讲解, 主要内容包括 SSM 框架整合的环境搭建及 SSM 的整合过程。
- 第 18 章对 SSM 框架的实际应用 (BOOT 客户管理系统) 进行了详细讲解, 主要内容包括系统概述、数据库设计、系统环境搭建, 以及用户登录模块和客户管理模块的开发实现等。

在学习过程中, 读者一定要亲自实践书中的案例代码, 如果不能完全理解书中所讲的知识点, 可以登录博学谷平台, 通过平台中的教学视频进行辅助学习。学习完一个知识点后, 要及时在博学谷平台上进行测试以巩固学习内容。另外, 如果读者在理解知识点的过程中遇到困难, 建议不要纠结于某个地方, 可以先往后学习。通常来讲, 随着对后面知识的不断深入了解, 前面看不懂的知识点一般就能理解了。如果读者在动手练习的过程中遇到问题, 建议多思考, 理清思路, 认真分析问题发生的原因, 并在问题解决后多总结。

致谢

本书的编写和整理工作由传智播客教育科技股份有限公司完成, 其中主要的参与人员有吕春林、陈欢、韩永蒙、石荣新、杜宏、梁桐、王友军、冯佳等。全体人员在近一年的编写过程中, 付出了很多辛勤的汗水, 在此一并表示衷心的感谢。

意见反馈

尽管我们尽了最大的努力，但教材中难免会有不妥之处，欢迎各界专家和读者朋友们来函给予宝贵意见，我们将不胜感激。您在阅读本书时，如发现任何问题或有不认同之处可以通过电子邮件与我们取得联系。

请发送电子邮件至：itcast_book@vip.sina.com。

黑马程序员
2017-6-12 于北京

目录

CONTENTS

专属于老师及学生的在线教育平台
yx.boxuegu.com

让 IT 教学更简单

教师获取教材配套资源

教案

授课资源

考试系统

在线题库

教学辅助
案例

添加微信/QQ

2011168841

让 IT 学习更有效

学生获取课后作业习题答案及配套源码

添加播妞微信/QQ

208695827

学习问答精灵: ask.boxuegu.com

更多学习视频: dvd.boxuegu.com



专属大学生的圈子

第 1 章 Spring 的基本应用 1

1.1 Spring 概述	2
1.1.1 什么是 Spring	2
1.1.2 Spring 框架的优点	2
1.1.3 Spring 的体系结构	3
1.1.4 Spring 的下载及目录结构	4
1.2 Spring 的核心容器	6
1.2.1 BeanFactory	6
1.2.2 ApplicationContext	6
1.3 Spring 的入门程序	8
1.4 依赖注入	11
1.4.1 依赖注入的概念	11
1.4.2 依赖注入的实现方式	12
1.5 本章小结	13

第 2 章 Spring 中的 Bean 15

2.1 Bean 的配置	16
2.2 Bean 的实例化	17
2.2.1 构造器实例化	17
2.2.2 静态工厂方式实例化	18
2.2.3 实例工厂方式实例化	20
2.3 Bean 的作用域	21
2.3.1 作用域的种类	21
2.3.2 singleton 作用域	22
2.3.3 prototype 作用域	23
2.4 Bean 的生命周期	23
2.5 Bean 的装配方式	25
2.5.1 基于 XML 的装配	25
2.5.2 基于 Annotation 的装配	28
2.5.3 自动装配	32
2.6 本章小结	33

第 3 章 Spring AOP..... 34	第 6 章 初识 MyBatis..... 84
3.1 Spring AOP 简介..... 35	6.1 什么是 MyBatis..... 85
3.1.1 什么是 AOP..... 35	6.2 MyBatis 的下载和使用..... 86
3.1.2 AOP 术语..... 36	6.3 MyBatis 的工作原理..... 87
3.2 动态代理..... 36	6.4 MyBatis 入门程序..... 88
3.2.1 JDK 动态代理..... 36	6.4.1 查询客户..... 88
3.2.2 CGLIB 代理..... 39	6.4.2 添加客户..... 96
3.3 基于代理类的 AOP 实现..... 41	6.4.3 更新客户..... 97
3.3.1 Spring 的通知类型..... 42	6.4.4 删除客户..... 99
3.3.2 ProxyFactoryBean..... 42	6.5 本章小结..... 100
3.4 AspectJ 开发..... 45	
3.4.1 基于 XML 的声明式 AspectJ..... 45	第 7 章 MyBatis 的核心配置 ... 101
3.4.2 基于注解的声明式 AspectJ..... 51	7.1 MyBatis 的核心对象..... 102
3.5 本章小结..... 55	7.1.1 SqlSessionFactory..... 102
	7.1.2 SqlSession..... 102
第 4 章 Spring 的数据库开发 ... 56	7.2 配置文件..... 105
4.1 Spring JDBC..... 57	7.2.1 主要元素..... 105
4.1.1 Spring JdbcTemplate 的解析..... 57	7.2.2 <properties>元素..... 106
4.1.2 Spring JDBC 的配置..... 57	7.2.3 <settings>元素..... 106
4.2 Spring JdbcTemplate 的 常用方法..... 59	7.2.4 <typeAliases>元素..... 108
4.2.1 execute()..... 59	7.2.5 <typeHandler>元素..... 109
4.2.2 update()..... 63	7.2.6 <objectFactory>元素..... 110
4.2.3 query()..... 68	7.2.7 <plugins>元素..... 111
4.3 本章小结..... 71	7.2.8 <environments>元素..... 111
	7.2.9 <mapers>元素..... 113
第 5 章 Spring 的事务管理..... 72	7.3 映射文件..... 114
5.1 Spring 事务管理概述..... 73	7.3.1 主要元素..... 114
5.1.1 事务管理的核心接口..... 73	7.3.2 <select>元素..... 115
5.1.2 事务管理的方式..... 75	7.3.3 <insert>元素..... 115
5.2 声明式事务管理..... 75	7.3.4 <update>元素和<delete>元素..... 117
5.2.1 基于 XML 方式的声明式事务..... 75	7.3.5 <sql>元素..... 118
5.2.2 基于 Annotation 方式的声明式 事务..... 80	7.3.6 <resultMap>元素..... 119
5.3 本章小结..... 83	7.4 本章小结..... 122
	第 8 章 动态 SQL..... 123
	8.1 动态 SQL 中的元素..... 124

8.2	<if>元素	124	11.4	本章小结.....	178
8.3	<choose>、<when>、 <otherwise>元素	126	第 12 章 Spring MVC 的核心类和 注解		
8.4	<where>、<trim>元素	129	12.1	DispatcherServlet.....	180
8.5	<set>元素	130	12.2	Controller 注解类型.....	180
8.6	<foreach>元素	132	12.3	RequestMapping 注解 类型	181
8.7	<bind>元素	134	12.3.1	@RequestMapping 注解的使用.....	181
8.8	本章小结.....	135	12.3.2	@RequestMapping 注解的属性.....	182
第 9 章 MyBatis 的关联映射			12.3.3	组合注解	183
9.1	关联关系概述	138	12.3.4	请求处理方法的参数类型和 返回类型	184
9.2	一对一	139	12.4	ViewResolver (视图解析器)	186
9.3	一对多	146	12.5	应用案例——基于注解的 Spring MVC 应用.....	186
9.4	多对多	151	12.6	本章小结.....	188
9.5	本章小结.....	155	第 13 章 数据绑定		
第 10 章 MyBatis 与 Spring 的 整合			13.1	数据绑定介绍	191
10.1	整合环境搭建	158	13.2	简单数据绑定	192
10.1.1	准备所需 JAR 包	158	13.2.1	绑定默认数据类型.....	192
10.1.2	编写配置文件	159	13.2.2	绑定简单数据类型.....	194
10.2	传统 DAO 方式的开发 整合.....	161	13.2.3	绑定 POJO 类型.....	195
10.3	Mapper 接口方式的开发 整合.....	165	13.2.4	绑定包装 POJO	198
10.3.1	基于 MapperFactoryBean 的 整合	165	13.2.5	自定义数据绑定.....	201
10.3.2	基于 MapperScannerConfigurer 的 整合	167	13.3	复杂数据绑定	205
10.4	测试事务	168	13.3.1	绑定数组	205
10.5	本章小结.....	171	13.3.2	绑定集合	207
第 11 章 Spring MVC 入门.....			13.4	本章小结.....	210
11.1	Spring MVC 概述.....	173	第 14 章 JSON 数据交互和 RESTful 支持		
11.2	第一个 Spring MVC 应用 ...	173	14.1	JSON 数据交互	212
11.3	Spring MVC 的工作流程.....	177	14.1.1	JSON 概述.....	212

14.1.2	JSON 数据转换	213	17.1.1	整合思路	253
14.2	RESTful 支持	221	17.1.2	准备所需 JAR 包	253
14.2.1	什么是 RESTful	221	17.1.3	编写配置文件	254
14.2.2	应用案例——用户信息查询	221	17.2	整合应用测试	258
14.3	本章小结	224	17.3	本章小结	262
第 15 章 拦截器			第 18 章 BOOT 客户管理系统		
15.1	拦截器概述	226	18.1	系统概述	264
15.1.1	拦截器的定义	226	18.1.1	系统功能介绍	264
15.1.2	拦截器的配置	227	18.1.2	系统架构设计	264
15.2	拦截器的执行流程	227	18.1.3	文件组织结构	265
15.2.1	单个拦截器的执行流程	227	18.1.4	系统开发及运行环境	266
15.2.2	多个拦截器的执行流程	230	18.2	数据库设计	266
15.3	应用案例——实现用户登录 权限验证	233	18.3	系统环境搭建	267
15.4	本章小结	238	18.3.1	准备所需 JAR 包	267
第 16 章 文件上传和下载			18.3.2	准备数据库资源	269
16.1	文件上传	241	18.3.3	准备项目环境	269
16.1.1	文件上传概述	241	18.4	用户登录模块	274
16.1.2	应用案例——文件上传	243	18.4.1	用户登录	274
16.2	文件下载	247	18.4.2	实现登录验证	281
16.2.1	实现文件下载	247	18.4.3	退出登录	284
16.2.2	中文名称的文件下载	249	18.5	客户管理模块	286
16.3	本章小结	251	18.5.1	查询客户	286
第 17 章 SSM 框架整合			18.5.2	添加客户	307
17.1	整合环境搭建	253	18.5.3	修改客户	315
			18.5.4	删除客户	322
			18.6	本章小结	325

Java EE

1

Chapter

第 1 章 Spring 的基本应用

学习目标

- 了解 Spring 的概念和优点
- 理解 Spring 中的 IoC 和 DI 思想
- 掌握 ApplicationContext 容器的使用
- 掌握属性 setter 方法注入的实现



Spring 是当前主流的 Java Web 开发框架,它是为了解决企业应用开发的复杂性问题而产生的。对于一个 Java 开发者来说,掌握 Spring 框架的使用,已是其必备的技能之一。本章将对 Spring 框架的基础知识进行详细的讲解。

1.1 Spring 概述

1.1.1 什么是 Spring

Spring 是由 Rod Johnson 组织和开发的一个分层的 Java SE/EE full-stack (一站式) 轻量级开源框架,它以 IoC (Inversion of Control, 控制反转) 和 AOP (Aspect Oriented Programming, 面向切面编程) 为内核,使用基本的 JavaBean 来完成以前只可能由 EJB (Enterprise Java Beans, Java 企业 Bean) 完成的工作,取代了 EJB 的臃肿、低效的开发模式。

Spring 致力于 Java EE 应用各层的解决方案,在表现层它提供了 Spring MVC 以及与 Struts 框架的整合功能;在业务逻辑层可以管理事务、记录日志等;在持久层可以整合 MyBatis、Hibernate、JdbcTemplate 等技术。因此,可以说 Spring 是企业应用开发很好的“一站式”选择。虽然 Spring 贯穿于表现层、业务逻辑层和持久层,但它并不想取代那些已有的框架,而是以高度的开放性与它们进行无缝整合。

1.1.2 Spring 框架的优点

Spring 具有简单、可测试和松耦合等特点,从这个角度出发,Spring 不仅可以用于服务器端开发,也可以应用于任何 Java 应用的开发中。关于 Spring 框架优点的总结,具体如下。

- 非侵入式设计

Spring 是一种非侵入式 (non-invasive) 框架,它可以使应用程序代码对框架的依赖最小化。

- 方便解耦、简化开发

Spring 就是一个大工厂,可以将所有对象的创建和依赖关系的维护工作都交给 Spring 容器管理,大大地降低了组件之间的耦合性。

- 支持 AOP

Spring 提供了对 AOP 的支持,它允许将一些通用任务,如安全、事务、日志等进行集中式处理,从而提高了程序的复用性。

- 支持声明式事务处理

只需要通过配置就可以完成对事务的管理,而无须手动编程。

- 方便程序的测试

Spring 提供了对 Junit4 的支持,可以通过注解方便地测试 Spring 程序。

- 方便集成各种优秀框架

Spring 不排斥各种优秀的开源框架,其内部提供了对各种优秀框架 (如 Struts、Hibernate、MyBatis、Quartz 等) 的直接支持。

- 降低 Java EE API 的使用难度

Spring 对 Java EE 开发中非常难用的一些 API (如 JDBC、JavaMail 等),都提供了封装,使这些 API 应用难度大大降低。

1.1.3 Spring 的体系结构

Spring 框架采用的是分层架构，它一系列的功能要素被分成 20 个模块，这些模块大体分为 Core Container、Data Access/Integration、Web、AOP (Aspect Oriented Programming)、Instrumentation、Messaging 和 Test，如图 1-1 所示。

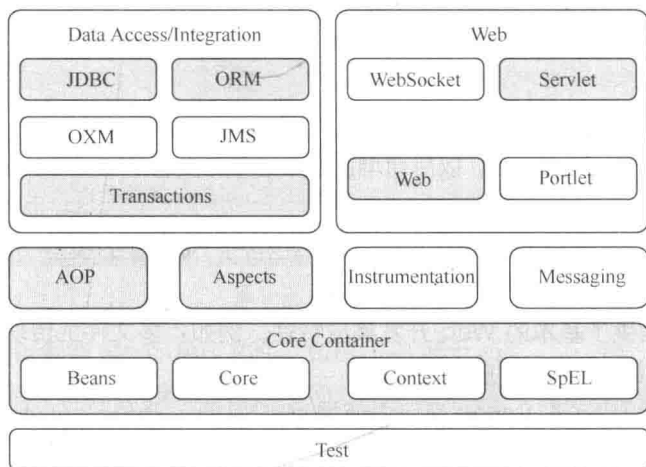


图1-1 Spring的体系结构

在图 1-1 中，包含了 Spring 框架的所有模块，其中，灰色背景模块为本书中所涉及的主要模块。接下来分别对体系结构中的模块作用进行简单介绍，具体如下。

1. Core Container (核心容器)

Spring 的核心容器是其他模块建立的基础，它主要由 Beans 模块、Core 模块、Context 模块、Context-support 模块和 SpEL (Spring Expression Language, Spring 表达式语言) 模块组成，具体介绍如下。

- Beans 模块：提供了 BeanFactory，是工厂模式的经典实现，Spring 将管理对象称为 Bean。
- Core 核心模块：提供了 Spring 框架的基本组成部分，包括 IoC 和 DI 功能。
- Context 上下文模块：建立在 Core 和 Beans 模块的基础之上，它是访问定义和配置的任何对象的媒介。其中 ApplicationContext 接口是上下文模块的焦点。
- Context-support 模块：提供了对第三方库嵌入 Spring 应用的集成支持，比如缓存 (EhCache、Guava、JCache)、邮件服务 (JavaMail)、任务调度 (CommonJ、Quartz) 和模板引擎 (FreeMarker、JasperReports、速率)。
- SpEL 模块：是 Spring 3.0 后新增的模块，它提供了 Spring Expression Language 支持，是运行时查询和操作对象图的强大的表达式语言。

2. Data Access/Integration (数据访问/集成)

数据访问/集成层包括 JDBC、ORM、OXM、JMS 和 Transactions 模块，具体介绍如下。

- JDBC 模块：提供了一个 JDBC 的抽象层，大幅度地减少了在开发过程中对数据库操作的编码。
- ORM 模块：对流行的对象关系映射 API，包括 JPA、JDO 和 Hibernate 提供了集成层

支持。

- OXM 模块: 提供了一个支持对象/ XML 映射的抽象层实现, 如 JAXB、Castor、XMLBeans、JiBX 和 XStream。

- JMS 模块: 指 Java 消息传递服务, 包含使用和产生信息的特性, 自 4.1 版本后支持与 Spring-message 模块的集成。

- Transactions 事务模块: 支持对实现特殊接口以及所有 POJO 类的编程和声明式的事务管理。

3. Web

Spring 的 Web 层包括 WebSocket、Servlet、Web 和 Portlet 模块, 具体介绍如下。

- WebSocket 模块: Spring 4.0 以后新增的模块, 它提供了 WebSocket 和 SockJS 的实现, 以及对 STOMP 的支持。

- Servlet 模块: 也称为 Spring-webmvc 模块, 包含了 Spring 的模型—视图—控制器(MVC) 和 REST Web Services 实现的 Web 应用程序。

- Web 模块: 提供了基本的 Web 开发集成特性, 例如: 多文件上传功能、使用 Servlet 监听器来初始化 IoC 容器以及 Web 应用上下文。

- Portlet 模块: 提供了在 Portlet 环境中使用 MVC 实现, 类似 Servlet 模块的功能。

4. 其他模块

Spring 的其他模块还有 AOP、Aspects、Instrumentation 以及 Test 模块, 具体介绍如下。

- AOP 模块: 提供了面向切面编程实现, 允许定义方法拦截器和切入点, 将代码按照功能进行分离, 以降低耦合性。

- Aspects 模块: 提供了与 AspectJ 的集成功能, AspectJ 是一个功能强大且成熟的面向切面编程 (AOP) 框架。

- Instrumentation 模块: 提供了类工具的支持和类加载器的实现, 可以在特定的应用服务器中使用。

- Messaging 模块: Spring 4.0 以后新增的模块, 它提供了对消息传递体系结构和协议的支持。

- Test 模块: 提供了对单元测试和集成测试的支持。

1.1.4 Spring 的下载及目录结构

Spring 的第一个版本是在 2004 年发布的, 经过 10 多年的发展, Spring 的版本也在不断地升级优化中。本书编写时, Spring 的最新版本为 4.3.6, 本书的代码都是基于该版本编写实现的, 建议读者也下载该版本。

Spring 开发所需的 JAR 包分为两个部分, 具体如下。

1. Spring 框架包

Spring 4.3.6 版本的框架压缩包, 名称为 spring-framework-4.3.6.RELEASE-dist.zip, 此压缩包可以通过地址 “<http://repo.spring.io/simple/libs-release-local/org/springframework/spring/4.3.6.RELEASE/>” 下载。下载完成后, 将压缩包解压到自定义的文件夹中, 解压后的文件目录结构如图 1-2 所示。

在图 1-2 的目录中, docs 文件夹中包含 Spring 的 API 文档和开发规范; libs 文件夹中包含

开发需要的 JAR 包和源码；schema 文件夹中包含开发所需要的 schema 文件，这些文件中定义了 Spring 相关配置文件的约束。

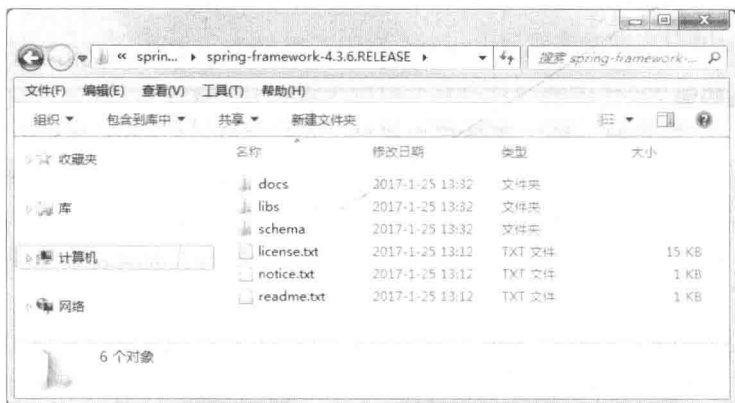


图1-2 解压后目录

打开 libs 目录可以看到 60 个 JAR 文件，如图 1-3 所示。



图1-3 libs目录

从图 1-3 可以看出，libs 目录中的 JAR 包分为三类，其中以 RELEASE.jar 结尾的是 Spring 框架 class 文件的 JAR 包；以 RELEASE-javadoc.jar 结尾的是 Spring 框架 API 文档的压缩包；以 RELEASE-sources.jar 结尾的是 Spring 框架源文件的压缩包。整个 Spring 框架由 20 个模块组成，该目录下 Spring 为每个模块都提供了这三类压缩包。

在 libs 目录中，有四个 Spring 的基础包，它们分别对应 Spring 核心容器的四个模块，具体介绍如下。

- spring-core-4.3.6.RELEASE.jar：包含 Spring 框架基本的核心工具类，Spring 其他组件都要用到这个包里的类，是其他组件的基本核心。
- spring-beans-4.3.6.RELEASE.jar：所有应用都要用到的 JAR 包，它包含访问配置文件、创建和管理 Bean 以及进行 Inversion of Control(IoC)或者 Dependency Injection(DI)操作相关的所有类。

- `spring-context-4.3.6.RELEASE.jar`: Spring 提供了在基础 IoC 功能上的扩展服务, 还提供了许多企业级服务的支持, 如邮件服务、任务调度、JNDI 定位、EJB 集成、远程访问、缓存以及各种视图层框架的封装等。

- `spring-expression-4.3.6.RELEASE.jar`: 定义了 Spring 的表达式语言。

2. 第三方依赖包

在使用 Spring 开发时, 除了要使用自带的 JAR 包外, Spring 的核心容器还需要依赖 `commons.logging` 的 JAR 包。该 JAR 包可以通过网址 “http://commons.apache.org/proper/commons-logging/download_logging.cgi” 下载。下载完成后, 会得到一个名为 `commons-logging-1.2-bin.zip` 的压缩包。将压缩包解压到自定义目录后, 即可找到 `commons-logging-1.2.jar`。

初学者学习 Spring 框架时, 只需将 Spring 的 4 个基础包以及 `commons-logging-1.2.jar` 复制到项目的 `lib` 目录, 并发布到类路径中即可。

1.2 Spring 的核心容器

Spring 框架的主要功能是通过其核心容器来实现的, 因此在正式学习 Spring 框架的使用之前, 有必要先对其核心容器有一定的了解。Spring 框架提供了两种核心容器, 分别为 `BeanFactory` 和 `ApplicationContext`。本节中将对这两种核心容器进行简单的介绍。

1.2.1 BeanFactory

`BeanFactory` 由 `org.springframework.beans.factory.BeanFactory` 接口定义, 是基础类型的 IoC 容器 (关于 IoC 的具体含义将在本章的 1.4.1 小节讲解, 这里只需知道其表示控制反转), 它提供了完整的 IoC 服务支持。简单来说, `BeanFactory` 就是一个管理 Bean 的工厂, 它主要负责初始化各种 Bean, 并调用它们的生命周期方法。

`BeanFactory` 接口提供了几个实现类, 其中最常用的是 `org.springframework.beans.factory.xml.XmlBeanFactory`, 该类会根据 XML 配置文件中的定义来装配 Bean。

创建 `BeanFactory` 实例时, 需要提供 Spring 所管理容器的详细配置信息, 这些信息通常采用 XML 文件形式来管理, 其加载配置信息的语法如下。

```
BeanFactory beanFactory = new XmlBeanFactory(new FileSystemResource("F:/applicationContext.xml"));
```

这种加载方式在实际开发中并不多用, 读者了解即可。

1.2.2 ApplicationContext

`ApplicationContext` 是 `BeanFactory` 的子接口, 也被称为应用上下文, 是另一种常用的 Spring 核心容器。它由 `org.springframework.context.ApplicationContext` 接口定义, 不仅包含了 `BeanFactory` 的所有功能, 还添加了对国际化、资源访问、事件传播等方面的支持。

创建 `ApplicationContext` 接口实例, 通常采用两种方法, 具体如下。

1. 通过 `ClassPathXmlApplicationContext` 创建

`ClassPathXmlApplicationContext` 会从类路径 `classPath` 中寻找指定的 XML 配置文件, 找