“在实践中成长”丛书

Java EE

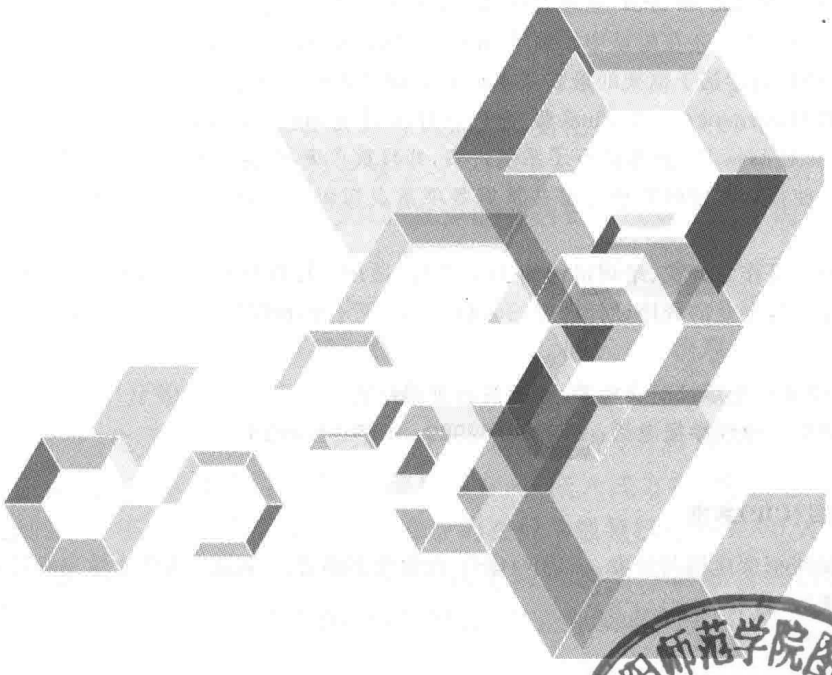
轻量级框架应用与开发——S2SH

QST青软实训 编著

- 采用新版S2SH框架，扩展Spring MVC和MyBatis
- 理论和实践结合，深入剖析企业轻量级框架解决方案
- GIFT-EMS企业真实项目贯穿全书
- 结合微信、支付宝应用，符合互联网创业潮流

清華大學出版社

“在实践中成长”丛书



Java EE 轻量级框架应用与开发——S2SH

QST青软实训 编著

清华大学出版社
北京

内 容 简 介

本书深入介绍了 Java EE 领域的三个开源框架：Struts 2、Hibernate 和 Spring，涵盖了 MVC 设计思想、Struts 2 的基本原理、处理流程及常用标签库的使用，Hibernate 的 ORM 设计理念、配置、实体映射文件以及 HQL 查询等，Spring 的 IoC 和 AOP 原理及应用、Bean 对象管理及事务处理等。除了 Struts 2、Hibernate 和 Spring 三个开源框架，本书还在附录中扩展了 Spring MVC 和 MyBatis 框架的使用。

书中所有代码都通过基于框架的最新版本环境下调试运行。其中，Struts 2 升级到 Struts 2.3.16.3 版，Hibernate 升级到 Hibernate 4.3.8.Final 版，Spring 升级到 Spring 4.1.5 版。

本书由浅入深对 Java EE 技术进行了系统讲解，并且重点突出、强调动手操作能力，以一个项目贯穿所有章节的任务实现，使得读者能够快速理解并掌握各项重点知识，全面提高分析问题、解决问题以及动手编码的能力。

本书适用面广，可作为高校、培训机构的 Java 教材，适用于计算机科学与技术、软件外包、计算机软件、计算机网络、电子商务等专业的程序设计课程的教材。本书适合各种层次的 Java 学习者和工作者阅读。

图书在版编目(CIP)数据

Java EE 轻量级框架应用与开发：S2SH/QST 青软实训编著. --北京：清华大学出版社，2016
"在实践中成长"丛书
ISBN 978-7-302-41371-4

Ⅰ. ①J…　Ⅱ. ①Q…　Ⅲ. ①JAVA 语言－程序设计　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 209109 号

责任编辑：刘　星　李　晔
封面设计：刘　键
责任校对：李建庄
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954
印 刷 者：清华大学印刷厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：32.5　　**字　　数**：811 千字
版　　次：2016 年 1 月第 1 版　　**印　　次**：2016 年 1 月第1 次印刷
印　　数：1～2000
定　　价：69.00 元

产品编号：065167-01

目 录

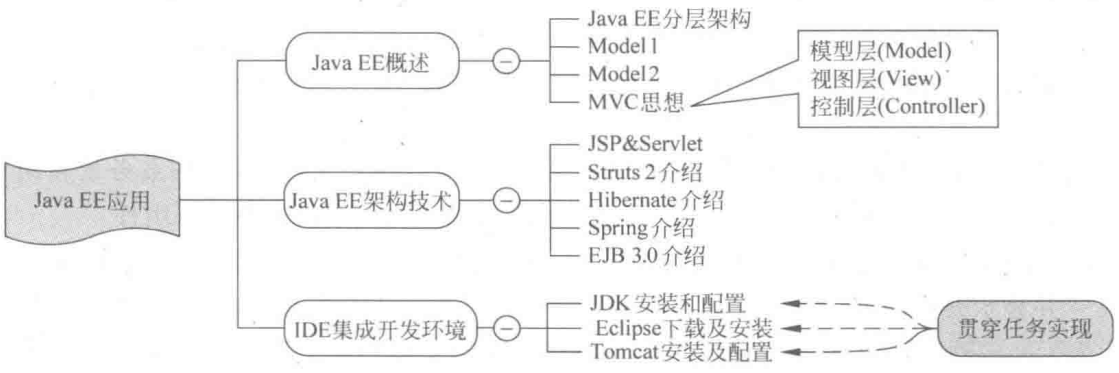
第1章 Java EE应用

任务驱动

开发任何项目之前都需要进行一些环境搭建的准备工作，本章任务是完成“GIFT-EMS礼记”系统所需要的开发环境搭建工作：

- 【任务 1-1】 JDK 的下载、安装和配置。
- 【任务 1-2】 Eclipse 的下载及安装。
- 【任务 1-3】 Tomcat 的下载、安装及配置。

学习路线



本章目标

知 识 点	Listen(听)	Know(懂)	Do(做)	Revise(复习)	Master(精通)
Java EE 概述	★	★			
Java EE 架构技术	★	★			
Java EE 开发环境搭建	★	★	★	★	★

1.1 Java EE 概述

Java 平台分为 Java SE、Java EE 和 Java ME 三个版本。其中，Java EE(Java Platform Enterprise Edition)是 Sun 公司为企业级应用推出的标准开发平台，该版本以前称为“J2EE”，

Sun 公司在 1998 年发布 JDK 1.2 版本时，使用了新的名称 Java 2 Platform，即“Java 2 平台”，修改后的 JDK 称为 J2SDK(Java 2 Platform Software Developing Kit)；并将平台分为 J2SE(Standard Edition，标准版)、J2EE(Enterprise Edition，企业版)、J2ME(Micro Edition，微型版)，J2ME 便由此诞生。2005 年 6 月，Java One 大会召开，SUN 公司发布 Java SE 6。此时，Java 的各种版本已经更名以取消其中的数字“2”，即 J2SE 更名为“Java SE”，J2EE 更名为“Java EE”，J2ME 更名为“Java ME”。

随着 Java 技术的发展，Java EE 平台得到迅速推广，已成为 Java 语言中最活跃的体系之一。现如今，Java EE 不仅仅是指一种软件技术，更多的表达着一种软件架构和设计思想，是一系列技术标准所组成的平台。

1.1.1 Java EE 分层架构

在企业级应用的开发过程中，软件的可维护性和可复用性是降低开发成本所必须要考虑的两个重要指标。因此，在开发过程中需要对项目结构进行分层，Java EE 的分层架构模式不仅使开发得到简化，而且提高了开发的效率。目前，针对 Java EE 企业级应用开发已出现许多优秀的框架，无论是经典的、还是轻量级的 Java EE 架构，都可以大致分为以下几个层次：

- **实体层(POJO 层)**：由 POJO(Plain Old Java Object，普通的传统 Java 对象)组成，这些对象代表系统的实体，通常与数据库中的表对应，主要作用是将数据保存起来，即持久化数据，一般保存在数据库或文件中。
- **数据访问层(DAO 层)**：由 DAO(Data Access Object)组件组成，这些 DAO 组件提供对实体对象的创建、查询、删除和修改等操作。
- **业务逻辑层(Service 层)**：由业务逻辑对象组成，用于实现系统所需要的业务逻辑方法。
- **控制器层(Controller 层)**：由控制器组成，用于响应用户请求，并调用业务逻辑组件的对应业务方法处理用户请求，然后根据处理结果转发到不同的表现层组件。
- **表现层(View 层)**：由页面(如 JSP、HTML)或其他视图组件组成，负责收集用户请求，并显示处理结果。

为了架构的可扩展性，各层组件之间需要以松散的方式耦合在一起，而不能以硬编码方式耦合。层与层的组件之间应符合面向接口编程的原则，使各层组件之间的依赖仅仅在接口层次。如图 1-1 所示，各层的 Java EE 组件以松耦合的方式耦合在一起，各组件并不以硬编码方式耦合，上面的组件实现依赖于下面组件的功能，下面组件支持了上面组件的实现，这种架构模式为应用提供了高度的可扩展性。对应这种 Java EE 应用架构，通常与工厂模式联系在一起，如业务逻辑对象仅仅需要访问 DAO 组件的工厂，而无须理会 DAO 对象的实现。在轻量级 Java EE 应用架构中，通常会交给类似于 Spring 框架的 IoC(Inversion of Control，控制反转)容器来管理组件之间的依赖，耦合度更低。

1.1.2 Model 1

Java 平台的动态网站编程技术经历了 Model 1 和 Model 2 时代。在早期的 Model 1 时期，整个网站应用主要由 JSP 页面组成，JSP 页面接收并处理客户端请求，辅助以少量的 JavaBean 完成数据库的连接、访问等特定的重复操作。

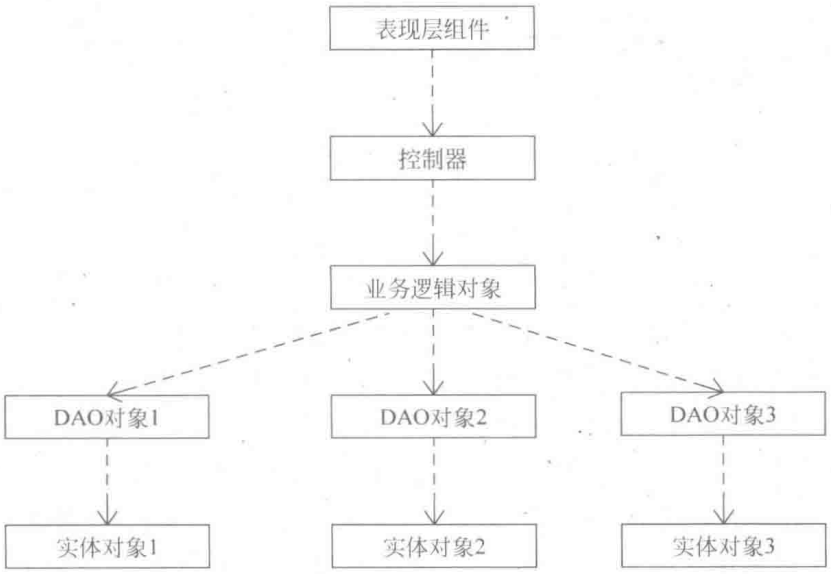


图 1-1 Java EE 应用分层架构

如图 1-2 所示，在 Model 1 体系中，JSP 页面负责响应用户请求并将处理结果返回用户，JSP 既要负责业务流程控制，又要负责提供表示层数据，同时充当视图和控制器，因此开发效率非常高。

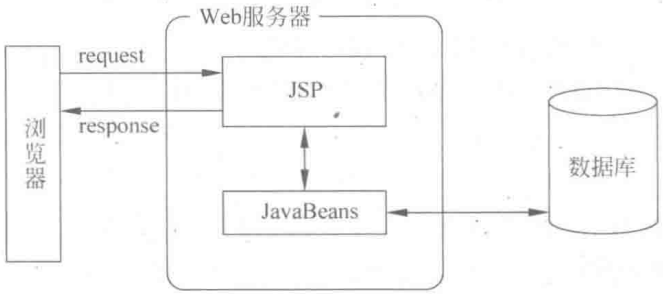


图 1-2 Model 1 体系

尽管 Model 1 体系十分适合简单应用的需要，但从工程化角度来看，局限性非常明显：JSP 页面身兼 View 和 Controller 两种角色，控制逻辑和表现逻辑混杂在一起，从而导致代码的重用性非常低，增加了应用的扩展和维护的难度，不适合开发复杂的大型应用程序。另外，不加选择地随意运用 Model 1，会导致 JSP 页内嵌入大量的 Java 代码，尽管这对 Java 程序员来说不是什么大问题，但如果 JSP 页面是由网页美工设计人员开发并维护的，则增加了其维护难度。从本质上分析，Model 1 体系将导致角色定义不清和职责分配不明，给项目管理带来很多麻烦，且在后期维护过程中非常困难。于是，软件开发人员开始寻找新的开发模式，基于 Java EE 的 Model 2 在这种背景下应运而生。

1.1.3 Model 2

在 Model 2 模式下，JSP 继续实现视图的功能，而控制器的功能用 Servlet 技术实现，模型功能用 JavaBean 技术实现，如图 1-3 所示。

Model 2 体系结构是一种联合使用 JSP 与 Servlet 来提供动态内容服务的方法，吸取了

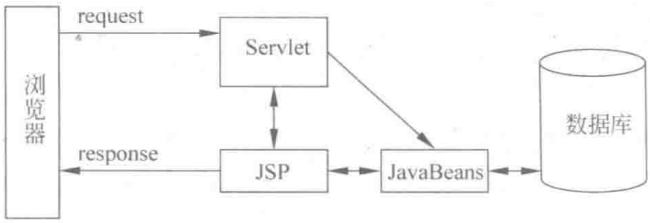


图 1-3　Model 2 体系

JSP 和 Servlet 两种技术各自的突出优点，由 JSP 生成表示层的内容，Servlet 完成对业务逻辑的处理。在此，Servlet 充当控制器的角色，负责处理用户请求，创建 JSP 页面需要使用的 JavaBean 对象，根据用户请求选择合适的 JSP 页面返回给用户。在 JSP 页面内没有处理逻辑，只承担检索原先由 Servlet 创建的 JavaBean 对象，从 Servlet 中提取动态内容插入到静态模板中。这是一种有突破性的软件设计方法，清晰地分离了表达和内容，明确了角色定义以及后台处理与前台页面设计的分工。事实上，项目越复杂，使用 Model 2 设计模式的好处就越大。

1.1.4　MVC 思想

MVC(Model-View-Controller)是软件开发的一种设计模式，即把一个应用的输入、处理、输出流程按照模型层(Model)、视图层(View)、控制层(Controller)的方式进行划分。主要目的是将模型层和视图层的代码分离，从而使同一个应用可以使用不同的表现形式，只需通过控制层确保两者的更新同步即可。对于 MVC 各层的详细介绍如下所述。

1. 模型层

模型层是对业务流程/状态的处理以及业务规则的制定。业务流程的处理过程对其他层来说是黑箱操作，模型接受视图请求的数据，并返回最终的处理结果。业务模型的设计是 MVC 最主要的核心。

2. 视图层

视图层代表用户交互界面，对于 Web 应用来说，可以概括为前台网页。随着应用的复杂性和规模性，界面的处理也变得具有挑战性。一个应用可能有很多不同的视图，MVC 设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理。业务流程的处理交予模型层处理。比如一个订单的视图只接受来自模型的数据并显示给用户，以及将用户界面的输入数据和请求传递给控制器和模型。

3. 控制层

控制层可以理解为从用户接收请求，将模型与视图匹配在一起，共同完成用户的请求。控制层就像一个分发器，对于选择的模型和视图以及需要完成的用户请求。控制层并不做任何的数据处理。例如，用户点击一个连接，控制层接受请求后，并不处理业务信息，只把用户请求传递给模型，告诉模型做什么，选择符合要求的视图返回给用户即可。因此，一个模型可能对应多个视图，一个视图可能对应多个模型。

模型、视图与控制器的分离，使得一个模型可以具有多个显示视图。如果用户通过某个视图的控制器改变了模型的数据，所有其他依赖于这些数据的视图都应反映到这些变化。因此，无论何时发生何种数据变化，控制器都会将变化通知所有的视图，导致显示的更新。MVC 三层架构的关系如图 1-4 所示。

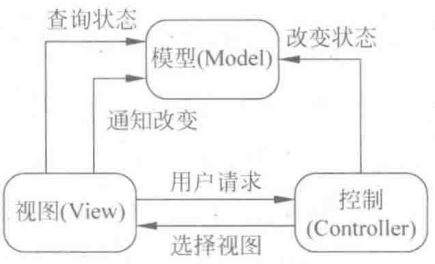


图 1-4 MVC 三层架构关系图

综上所述，概括 MVC 设计思想有以下几个优点：

- 低耦合性，提高了应用的可扩展性和可维护性；
- 高重用性和可适用性；
- 有利于软件工程化管理。

1.2 Java EE 架构技术

Java EE 架构技术经历多年的发展，出现了很多实用技术。从最初的 JSP 和 Servlet，到 JSP Model 2 基础上的各种 MVC 框架，以及轻量级容器和数据持久化工具等。随着各框架的不断发展，Spring、Struts、Hibernate 三个开源框架的组合，以其优异的性能，为开发人员熟悉接受，成为最流行的轻量级 Java EE 架构。

1.2.1 JSP&Servlet

JSP 技术给开发人员一个简单的与 HTML 类似的接口来创建 Servlet。JSP 文件中可以包含 HTML 代码、Java 代码以及被称为 JavaBean 的编程模块。但实际上 JSP 技术提供与 Servlet 同样的功能，因为在运行时 JSP 必须被 Web 服务器编译成 Servlet，所以服务器端真正运行的是 Servlet。当前，JSP 在 Java Web 应用开发中主要充当表现层，并广泛使用。

Servlet 是为接受来自浏览器的 HTTP 请求并返回其应答的服务器端技术。因为 Servlet 是用 Java 代码来实现的，所以可以满足平台间的完全兼容。

1.2.2 Struts 2 介绍

Struts 是一个为开发基于 MVC 模式的应用架构的开源框架，是利用 Servlet 和 JSP 构建 Web 应用的一项非常有用的技术。

Struts 最早是 Apache Jakarta 项目的组成部分，项目的创立者希望通过对该项目的研究，改进和提高 JSP、Servlet、标签库以及面向对象的技术水准。使用 Struts 可以为业务应用的每一层提供支持，提供软件开发的“支撑”，减少在运用 MVC 设计模型来开发 Web 应用的时间。

引入 MVC 模式后，系统中各组件只负责相对应的逻辑，具有组件化的特点，更适合大规模应用的开发。由于 Struts 能充分满足应用开发的需求，简单易用，敏捷迅速，因而吸引了众多的开发人员的关注。而 Struts 2 是在 Struts 框架和 WebWork 框架基础上发展起来的，由于结合了两个框架的优点，因此在稳定性、开发速度、性能等方面都有很好的体现，使 Web 开发变得更容易。