Broadview®
www.broadview.com.cn

基于“开箱即用”原则的Spring Boot，令企业应用开发更加快速和高效

Spring Boot 2
企业应用实战

疯狂软件 编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Spring Boot 2

企业应用实战

疯狂软件 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书介绍了 Java EE 领域的全新开源框架：Spring Boot 2。本书的示例建议在 Tomcat 8 上运行。

本书重点介绍如何使用 Spring Boot 进行 Java EE 快速开发，从内容上可以划分为四个部分。第一部分详细介绍了 Spring Boot 的核心知识。第二部分详细介绍了 Spring Boot 的 Web 开发。第三部分重点介绍了 Spring Boot 的数据访问。第四部分重点介绍了 Spring Boot 的 Spring Security 安全控制。书中示范开发了一个包含 7 个表、表之间具有复杂的关联映射关系，且业务功能也相对完善的信息管理系统案例，希望让读者理论联系实际，将 Spring Boot 框架真正运用到实际开发当中去。该案例采用目前最流行、最规范的 Java EE 架构，整个应用分为 DAO 持久层、领域对象层、业务逻辑层、控制器层和视图层，各层之间分层清晰，层与层之间以松耦合的方法组织在一起。所有代码完全基于 Eclipse IDE 来完成，一步步带领读者深入两个框架的核心。

阅读本书之前，建议先阅读疯狂软件教育的《疯狂 Java 讲义》一书。本书适合有较好的 Java 编程基础，JSP、Servlet、JDBC 基础，Spring 框架基础的读者，尤其适合于对 Spring Boot 了解不够深入，或对 Spring Boot 整合开发不太熟悉的开发人员阅读。

图书在版编目（CIP）数据

Spring Boot 2 企业应用实战 / 疯狂软件编著. —北京：电子工业出版社，2018.6
ISBN 978-7-121-34116-8

Ⅰ. ①S… Ⅱ. ①疯… Ⅲ. ①JAVA 语言—程序设计 Ⅳ. ①TP312.8

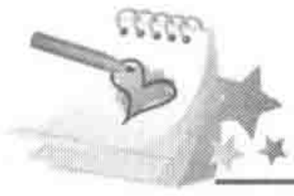
中国版本图书馆 CIP 数据核字（2018）第 083111 号

策划编辑：张月萍
责任编辑：刘　舫
印　　刷：三河市华成印务有限公司
装　　订：三河市华成印务有限公司
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：16　　字数：410 千字
版　　次：2018 年 6 月第 1 版
印　　次：2018 年 6 月第 1 次印刷
印　　数：4000 册　　定价：58.00 元

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，联系及邮购电话：（010）88254888，88258888。

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

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。



前　言

时至今日，以 Spring 为核心的轻量级 Java EE 企业开发平台在企业开发中占有绝对的优势，Java EE 应用以其稳定的性能、良好的开放性以及严格的安全性，深受企业应用开发者的青睐，应用的性能、稳定性都有很好的保证。

Spring 在 Java EE 开发中是实际意义上的标准，但是在实际项目开发中使用 Spring 的时候经常遇到两个让人非常头疼的问题：

（1）大量的配置文件

（2）与第三方框架整合

特别是在今天，脚本语言和敏捷开发大行其道之时，Spring 的开发显得尤其烦琐。而 Spring Boot 的推出正具有颠覆和划时代的意义。如果说 Spring 框架的目标是帮助开发者写出更好的系统，那 Spring Boot 的目标就是帮助开发者用更少的代码，更快地写出好的系统。

Spring Boot 从无数知名企业的实践中吸取经验，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。Spring Boot 遵循“约定优于配置”原则，从而使开发人员不再需要定义样板化的配置，只需要很少的配置，或者大部分时候只是使用默认配置就可以快速搭建项目，无须配置整合第三方框架。通过这种方式，Spring Boot 在蓬勃发展的快速应用开发（rapid application development）领域已经成为领导者。

本书基于 Spring Boot 2.0 版本，重点介绍 Spring Boot 框架，采用 Tomcat 8 作为 Web 服务器，Eclipse IDE 作为开发工具，详细介绍了 Spring Boot 框架的绝大部分功能。希望读者在阅读、学习完本书之后，能够掌握 Spring Boot 技术，更快更好地开发出 Java EE 项目，为 Java 开发者带来更多的就业机会与竞争力。

本书有什么特点

本书是一本介绍 Spring Boot 框架的实用图书，全面介绍了最新的 Spring Boot 和常用第三方框架整合等各方面的知识。

本书针对每一个知识点都通过相应的程序给出了示范，第 7 章的实战项目“信息管理系统”采用目前非常流行、规范的 Java EE 架构，整个应用分为 DAO 持久层、领域对象层、业务逻辑层、控制器层和视图层，各层之间分层清晰，层与层之间以松耦合的方法组织在一起。

笔者既担任过软件开发的技术经理，也担任过软件公司的培训导师，如今从事专业、高端的职业技术培训，所有应用范例都密切契合企业开发实际场景，例如用户权限验证、文件上传下载等都是企业开发中的实际功能，同时采用了目前企业最流行、最规范的开发架构，严格遵守 Java EE 开发规范。读者参考本书的架构，完全可以身临其境地感受企业实际项目开发。

本书并不是一本关于所谓“思想”的书，也没有一堆“深奥”的新名词和“高深”的理念，只会让读者学会实际的 Spring 和 Spring Boot 技术。本书的特点是操作步骤详细，编程思路清晰，语言平实易懂。只要读者认真阅读本书，并掌握书中知识，那么就完全可以胜任企业中的 Spring Boot 项目开发。

阅读本书需要具备一定的计算机知识以及编程功底。熟练掌握Java语言和Spring框架的IoC、AOP和持久层的ORM设计模式等知识对于学习本书是很有必要的。

可访问www.crazyit.org或www.broadview.com.cn/34116下载本书配套资源。

本书写给谁看

如果你已经掌握Java SE的内容，或已经学完疯狂软件教育的《疯狂Java讲义》一书，那么非常适合阅读此书。除此之外，如果你已有初步的JSP、Servlet、JDBC基础，甚至对Spring、Spring Boot等框架有所了解，但希望掌握它们在实际开发中的应用，本书也将非常适合你。如果你对Java的掌握还不熟练，则建议遵从学习规律，循序渐进，暂时不要购买、阅读此书，而是按照“疯狂Java学习路线图”中的建议顺序学习。

衷心感谢

衷心感谢李刚老师，他是一位非常好的朋友，在本书的创作过程中，他提供了大量切实、有用的帮助。同时衷心感谢疯狂软件教育中心所有同事提供的帮助。

感谢所有参加疯狂软件实训的学生，他们在实际工作场景的应用证明了本书的价值，他们的反馈让本书更加实用。

编者

2018年3月1日

目 录 CONTENTS

CHAPTER

1

第1章

Spring Boot入门

本章要点

- Spring 简介
- Spring Boot 简介
- 下载和安装 Maven
- Eclipse 集成 Maven
- Eclipse 构建基于 Maven 的 Spring Boot 项目

时至今日，轻量级 Java EE 平台在企业开发中占有绝对的优势，Java EE 应用以其稳定的性能、良好的开放性以及严格的安全性，深受企业应用开发者的青睐。实际上，对于信息化要求较高的行业，如银行、电信、证券以及电子商务等行业，都不约而同地选择了 Java EE 开发平台。

而 Spring 在 Java EE 开发中已经是实际意义上的标准，绝大部分的软件开发公司在开发 Java EE 应用的时候都会用到 Spring。不仅如此，围绕 Spring，以 Spring 为核心还衍生出了一系列框架，如 Spring Boot、Spring Cloud、Spring Cloud Data Flow、Spring Web Flow、Spring Security、Spring for Android 等（登录 Spring 官方网站 https://spring.io/了解具体内容），Spring 越来越强大，在开发中带来越来越多的便捷。本书所介绍的正是现阶段非常流行的 Spring Boot 框架。

1.1 Spring 简介

Spring 框架由 Rod Johnson 开发，是一个轻量级的企业级开发一站式解决方案。2004 年发布了 Spring 框架的第一个版本。经过十多年的发展，Spring 已经发展成 Java EE 开发中最重要的框架之一。对于 Java 开发者来说，Spring 已经成为必须掌握的框架。

1.1.1 Spring 概述

Spring 是一个从实际开发中抽取出来的框架，因此它完成了大量开发中的通用步骤，留给开发者的仅仅是与特定应用相关的部分，从而大大提高了企业应用的开发效率。

Spring 为企业应用的开发提供了一个轻量级的解决方案。该解决方案包括：基于依赖注入的核心机制、基于 AOP 的声明式事务管理、与多种持久层技术的整合以及优秀的 Web MVC 框架等。Spring 致力于 Java EE 应用各层的解决方案，而不是仅仅专注于某一层的方案。可以说，Spring 是企业应用开发的“一站式”选择，Spring 贯穿表现层、业务层、持久层。然而，Spring 并不想取代那些已有的框架，而是以高度的开放性与它们无缝整合。

总结起来，Spring 具有如下优点。

- 低侵入式设计，代码的污染极低。
- 独立于各种应用服务器，基于 Spring 框架的应用，可以真正实现 Write Once，Run Anywhere 的承诺。
- Spring 的 IoC 容器降低了业务对象替换的复杂性，提高了组件之间的解耦。
- Spring 的 AOP 支持允许将一些通用任务如安全、事务、日志等进行集中式处理，从而提供了更好的复用性。
- Spring 的 ORM 提供了与第三方持久层框架的良好整合，并简化了底层的数据库访问。
- Spring 的高度开放性，并不强制应用完全依赖于 Spring，开发者可自由选用 Spring 框架的部分或全部。

图 1.1 显示了 Spring 框架的组成结构图。

正如图 1.1 所示，Spring 是模块化的，这意味着开发者可以选择自己需要的 Spring 模块。其中最重要的模块包括 Core Container（核心容器）、AOP（面向切面编程）、Web（Web 集成功能）、Data Access（数据访问功能）等。

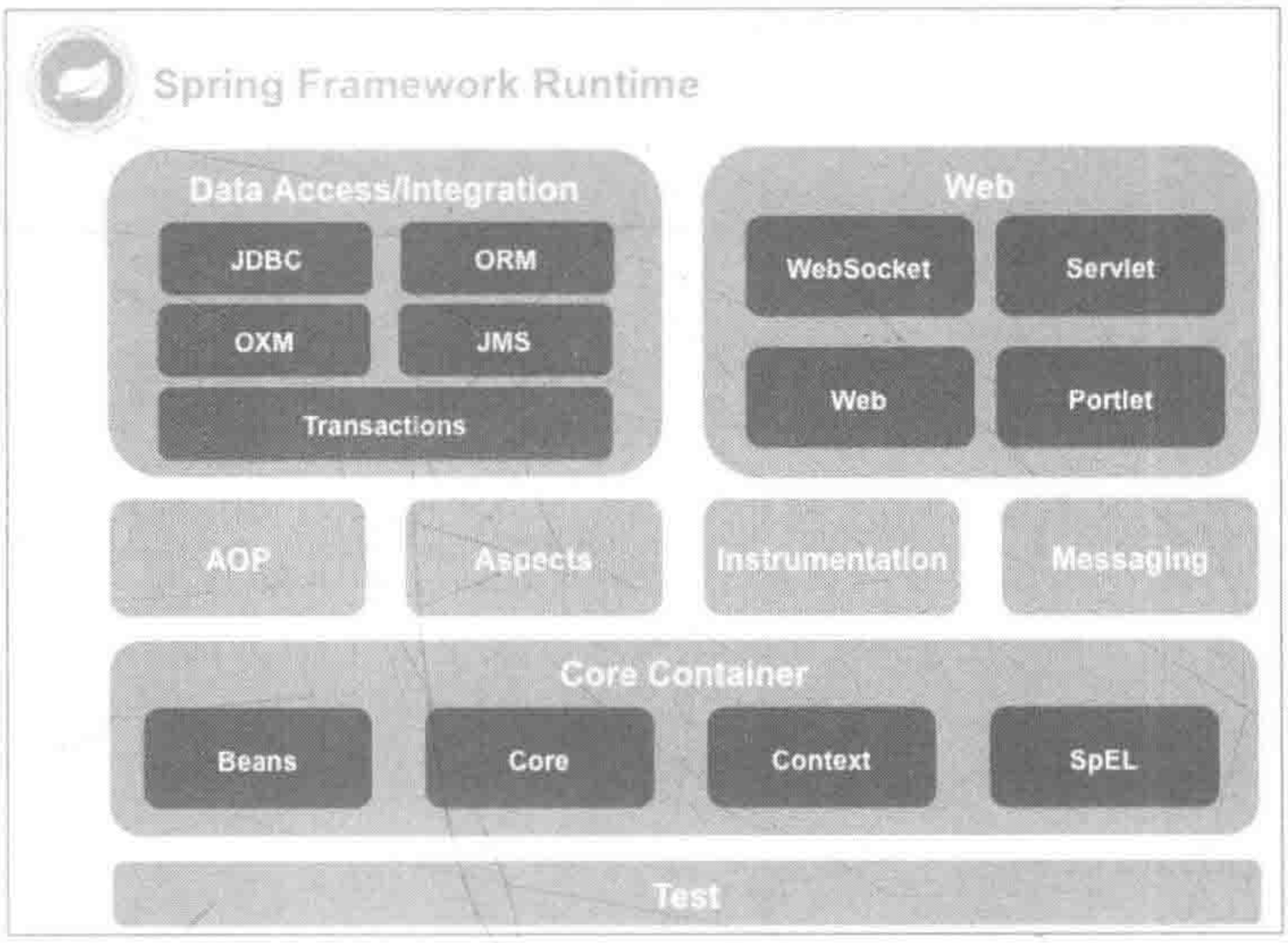


图 1.1 Spring 框架的组成结构图

当使用 Spring 框架时，必须使用 Spring Core Container（即核心容器），它代表了 Spring 框架的核心机制，Spring Core Container 主要由 org.springframework.core、org.springframework.beans、org.springframework.context 和 org.springframework.expression 四个包及其子包组成，主要提供 Spring IoC 容器支持。

1.1.2 Spring 的生态圈

Spring 发展到今天已经不仅仅是 Spring 框架本身的内容了，Spring 目前提供了非常多的基于 Spring 的项目，可以用来更加快捷地开发项目。

Spring 目前主要有以下项目。

- Spring Boot：使用默认 Java 配置来实现快速开发。
- Spring XD：用于简化大数据应用的开发。
- Spring Cloud：为分布式系统开发提供工具集。
- Spring Data：对主流的关系型数据库和 NoSQL 数据库提供支持。
- Spring Integration：通过消息机制对企业集成模式（EIP）提供支持。
- Spring Batch：简化及优化大量数据的批处理操作。
- Spring Security：通过认证和授权保护应用。
- Spring HATEOAS：基于 HATEOAS 原则简化 REST 服务开发。
- Spring Social：与社交网络 API（如 Facebook、新浪微博等）的集成。
- Spring AMQP：对基于 AMQP 的消息的支持。
- Spring Mobile：提供对手机设备进行检测的功能，给不同的设备返回不同的支持。
- Spring for Android：提供在 Android 上消费 RESTful API 的功能。
- Spring Web Flow：基于 Spring MVC 提供的 Web 应用开发。
- Spring Web Services：提供基于协议有限的 SOAP/Web 服务。
- Spring LDAP：简化使用 LDAP 开发。
- Spring Session：提供一个 API 及实现来管理用户会话信息。

1.1.3 Spring 5 的变化

与 Spring 4.x 相比，Spring 5 发生了一些变化，这些变化包括如下这些。

- Spring 5 已经全面支持 Java 8。
- 删除了一些已过时的包和类。
- 核心 IoC 容器新增了泛型限定式依赖注入、Map 依赖注入、List（数组）注入、延迟注入等功能，这些功能主要体现在基于注解的配置上。
- Spring 5 的 Web 支持已经升级为支持 Servlet 3.0 以及更高的规范。
- 从 Spring 5 开始，Spring 支持使用 Groovy DSL 进行 Bean 配置。
- Spring 5 新增了一个 spring-websocket 模块，该模块支持 WebSocket、SockJS、STOMP 通信。

从以上介绍可以看出，Spring 5 的升级主要就是为了支持 Java 8 和 Servlet 3.0 规范，也为核心 IoC 容器增强了一些注解。

1.1.4 Spring 的配置简化

Spring 最大的特点是通过配置简化开发。

- 在 Spring 1.x 的时代，使用 Spring 开发都是使用 xml 配置 Bean。
- 在 Spring 2.x 的时代，Spring 提供了声明 Bean 的注解（如@Component、@Service 等），大大减少了 XML 的配置量。随之而来的是开发者的讨论：注解配置和 XML 配置到底哪个更好？开发者最终的选择是应用的基本配置（如数据源、事务管理）使用 XML，业务配置使用注解。
- 从 Spring 3.x 到今天，Spring 提供了 Java 配置的能力，使用 Java 配置可以让开发者更加理解所配置的 Bean。Spring 5 和 Spring Boot 都推荐使用 Java 的注解配置，所以本书都将使用 Java 的注解配置。

1.2 Spring Boot 简介

1.2.1 Spring Boot 概述

Spring 框架非常优秀，然而它最大的问题在于“配置过多”。基于 Spring 的企业级开发项目，需要大量的配置文件，Spring Boot 的出现就是为了解决 Spring 框架存在的问题。

Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化 Spring 应用的创建、运行、调试、部署等。使用 Spring Boot 可以做到专注于 Spring 应用的开发，而无须过多关注 XML 的配置。Spring Boot 使用“约定优先于配置（COC，Convention Over Configuration）”的理念，简单来说，Spring Boot 提供了针对企业应用开发各种场景的很多 spring-boot-starter 自动配置依赖模块，这些模块都基于“开箱即用”的原则，使得企业应用开发中各种场景的 Spring 应用更加快速和高效。

Spring Boot 是开发者和 Spring 框架的中间层，帮助开发者统筹管理应用的配置，提供基于实际开发中常见配置的默认处理（即约定优先于配置），简化应用的开发和运维；总体来说，Spring Boot 的目的就是为了对 Java Web 的开发进行“简化”和“加速”，简化开发过程中引入或启动相关 Spring 功能的配置。这样带来的好处就是降低开发人员对于框架的关注度，可以把更多的精力放在自己的业务代码上。

同时随着微服务概念的推广和实践，Spring Boot 的精简理念又使其成为 Java 微服务开发

的不二之选，也可以说，Spring Boot 是最适合微服务的 Java Web 框架。关于微服务的更多知识请参考疯狂软件系列图书中的《疯狂 Spring Cloud 微服务架构实战》。

现如今，Spring Boot 已经在蓬勃发展的快速应用开发领域成为领导者。

1.2.2 Spring Boot 解决的问题

Spring Boot 的出现带来了以下优点。

- **使编码变得简单：**推荐使用注解。
- **使配置变得简单：**自动配置、快速构建项目、快速集成新技术的能力。
- **使部署变得简单：**内嵌 Tomcat、Jetty 等 Web 容器。
- **使监控变得简单：**自带项目监控。

1.2.3 Spring Boot 的主要特性

Spring Boot 的主要特点如下。

- Spring Boot 是伴随着 Spring 4.0 诞生的，继承了 Spring 框架原有的优秀基因。
- 遵循“约定优先于配置”的原则，使用 Spring Boot 只需很少的配置，大部分的时候直接使用默认的配置即可。
- 对主流开发框架无配置集成，自动整合第三方框架。
- 可独立运行 Spring 项目，Spring Boot 可以以 jar 包的形式独立运行。使用 java –jar 命令或者在项目的主程序中执行 main 函数就可以成功运行项目。
- 内嵌 Servlet 容器，可以选择内嵌 Tomcat、Jetty 等 Web 容器，无须以 war 包形式部署项目。
- 提供 starter 简化 Maven 配置，基本上可以做到自动化配置，高度封装，开箱即用。
- Spring Boot 会根据项目依赖来自动配置 Spring 框架，极大减少了项目所使用的配置。
- Spring Boot 提供了准生产环境的应用监控。
- 无代码生成和 XML 配置，纯 Java 的配置方式，很简单，很方便。
- 分布式开发，与 Spring Cloud 的微服务无缝结合。

1.2.4 Spring Boot 2.0 的重要改变

- 基于 Spring 5 构建，Spring 5 的新特性都可以在 Spring Boot 2.0 中使用。
- 为各种组件的响应式编程提供了简化配置，如：Reactive Spring Data、Reactive Spring Security 等。
- 要求 Java 版本必须是 Java 8 或更高版本，支持最新的 Java 9。
- 要求 Gradle 4 或更高版本、Maven 3.2 或更高版本。
- 要求 Tomcat 8 或更高版本，Hibernate 5.2 或更高版本，Thymeleaf 3 或更高版本。

1.3 “开箱即用”的依赖模块

Spring Boot 提供了针对企业应用开发各种场景的很多 spring-boot-starter 自动配置依赖模块，它们都约定以 spring-boot-starter-作为命名的前缀，并且都位于 org.springframework.boot 包或者命名空间下。

访问 http://start.spring.io，页面如图 1.2 所示。

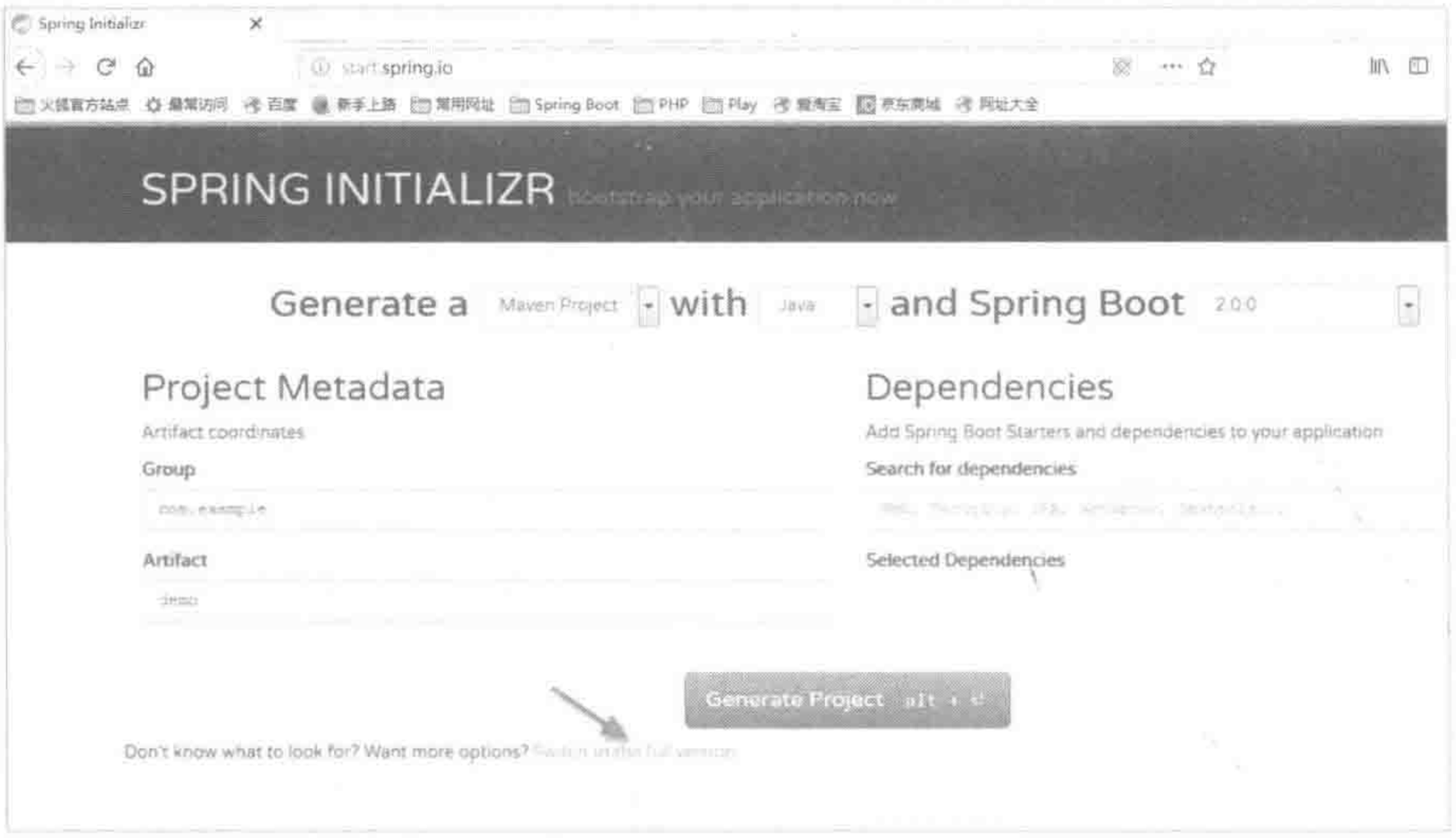


图 1.2　Spring Initializr 页面

单击图 1.2 所示的“Switch to the full version”超链接，可以看到展开的页面上显示的 Spring Boot 默认支持和提供了大约 80 个自动配置依赖模块。如此之多的依赖模块，开发者在实际项目中也不一定都会用到。此处重点讲解几个最常用的 spring-boot-starter 模块，其实所有的 spring-boot-starter 模块的使用都大同小异，读者可以之后在工作中灵活使用。

所有的 spring-boot-starter 模块都有约定的默认配置，但是允许开发者调整这些默认的配置用以改变默认的配置行为，这就是所谓的“约定优先于配置”。

简单来讲，Spring Boot 的配置主要可以分为以下几类：

- 命令行参数
- 系统环境变量
- 位于文件系统中的配置文件
- 位于 classpath 中的配置文件
- 固化到代码中的配置

以上几种方式按照优先级从高到低排列，高优先级方式提供的配置项会覆盖或者优先生效，比如通过命令行参数传入的配置项会覆盖通过环境变量传入的同一配置项。

实际项目开发中最常用的配置是配置文件，不管是位于文件系统还是位于 classpath，Spring Boot 应用默认的配置文件名叫作 application.properties，可以放在项目的 src/main/resources 目录下或者在类路径的/config 目录下。

1.3.1　日志依赖模块 spring-boot-starter-logging

Java 的日志系统多种多样，有默认的 java.util 提供的日志支持，有 log4j、log4j2、commons logging 等日志框架。可以通过如下配置在 Maven 中添加 spring-boot-starter-logging 依赖模块。

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-logging</artifactId>
</dependency>
```

Spring Boot 将自动使用 logback 作为项目的日志框架。Spring Boot 为开发者提供了很多默认的日志配置，所以，只要将 spring-boot-starter-logging 依赖模块添加到当前项目，即可以“开箱即用”，不需要做任何多余的配置。

如果要对 Spring Boot 默认提供的日志设置做调整，则可以遵循 logback 的约定，使用自己定制的 logback.xml 日志文件，logback.xml 日志文件建议放在项目的 src/main/resources 目录下，然后在 application.properties 中指定。

```
logging.config = logback.xml
```

如果更习惯使用 log4j2，只需要采用类似的方式将对应的 spring-boot-starter 依赖模块加到 Maven 中。

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-log4j2</artifactId>
</dependency>
```

1.3.2 Web 开发依赖模块 spring-boot-starter-web

在互联网时代，大部分项目都是基于 Web 开发的，Spring 框架的使用者几乎都会使用 Spring MVC 来开发 Web 项目。为了帮助开发者简化快速搭建过程并开发 Web 项目，Spring Boot 提供了 spring-boot-starter-web 自动配置依赖模块。

可以通过如下配置在 Maven 中添加 spring-boot-starter-web 依赖模块。

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

这样，就得到了一个可以直接执行的 Web 项目，运行当前项目的 App 的 main 方法就可以直接启动一个使用了嵌入式 Tomcat 服务器的 Web 应用。只不过，在还没有提供任何服务 Web 请求的 Controller 时，访问任何路径都会返回一个 Spring Boot 默认提供的错误请求页面（称为 Whitelabel Error Page），如图 1.3 所示。

图 1.3 Spring Boot 默认错误页面

可以在当前项目下新建一个 Spring MVC 的 Controller 控制器，代码如下。

```
@RestController
public class IndexController {
    // 映射"/"请求
    @RequestMapping("/")
    public String index(){
        return "Hello Spring Boot!";
    }
}
```

重新运行 App 的 main 方法并访问 http://127.0.0.1:8080，将正常显示 Controller 返回的信息“Hello Spring Boot!”，一个简单的 Web 项目就完成了。

提示

本节重点讲解 spring-boot-starter-web 依赖模块的作用，关于 Spring Boot 项目的搭建，Spring MVC 的 Controller 控制器的使用，1.4 节会重点介绍。

Spring Boot 使用 spring-boot-starter-web 依赖模块开发 Web 项目，非常简单、方便。但是，在简单的背后，其实有很多的“约定”，只有充分了解这些“约定”，才能更好地使用 spring-boot-starter-web 依赖模块。

（1）项目结构层面的约定

在传统的 Java Web 项目中，所有的静态文件和页面都放在 WebContent 目录下，而 Spring Boot 项目的静态文件和页面统一放在 src/main/resources 目录对应的子目录下。src/main/resources/static 目录用于存放各类静态资源文件，比如 css、js 和 image 等；src/main/resources/templates 目录用于存放页面模板文件，比如 html 和 jsp 等。

（2）Spring MVC 框架层面的约定

spring-boot-starter-web 依赖模块默认自动配置一些 Spring MVC 必要的组件。

- 将 ViewResolver 自动注册到 Spring 容器。
- 将 Converter 和 Formatter 等 bean 自动注册到 Spring 容器。
- 将对 Web 请求的支持和相应的类型转换的 HttpMessageConverter 自动注册到 Spring 容器。
- 将 MessageCodesResolver 自动注册到 Spring 容器。

（3）嵌入式 Web 容器层面的约定

spring-boot-starter-web 依赖模块默认使用嵌入式 Tomcat 作为 Web 容器对外提供服务，默认使用 8080 端口对外监听和提供服务。

如果在开发中不想使用默认的嵌入式 Tomcat，可以引入 jetty 作为替代方案。

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jetty</artifactId>
</dependency>
```

或者使用 undertow 作为替代方案。

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-undertow</artifactId>
</dependency>
```

如果不想使用默认的 8080 端口，可以通过更改 application.properties 配置文件中的 server.port 使用自己指定的端口。

```
server.port=8088
```

Spring Boot 提供了大量的 spring-boot-starter-依赖模块，后面的章节会详细讲解。

1.4 开发第一个 Spring Boot 应用

本书重点介绍 Spring Boot，使用 Eclipse 作为 Java IDE 开发工具，使用 Maven 作为项目构建工具，关于 Eclipse 和 Maven 的详细安装和使用、Spring 的使用请参考疯狂软件系列图书中的《轻量级 Java EE 企业应用实战（第 5 版）：Struts 2+Spring 5+Hibernate 5/JPA 2 整合开发》，关于 Spring MVC 的使用请参考疯狂软件系列图书中的《Spring+MyBatis 企业应用实战》。

1.4.1 下载和安装 Maven

下载和安装 Maven 请按如下步骤进行。

① 登录 http://maven.apache.org/download.cgi 站点下载 Maven 最新版，本书成书之时，Maven 的最新稳定版是 3.5.0，建议下载该版本。

虽然 Maven 是基于 Java 的生成工具，具有平台无关的特性，但考虑到解压缩的方便性，

通常建议 Windows 平台下载*.zip 压缩包，而 Linux 平台则下载*.gz 压缩包。

② 将下载到的压缩文件解压缩到任意路径，此处将其解压缩到 D:\路径下，解压缩后生成文件夹 apache-maven-3.5.0。解压缩后看到如下文件结构。

- bin：保存 Maven 的可执行命令。其中 mvn 和 mvn.bat 就是执行 Maven 工具的命令。
- boot：该目录只包含一个 plexus-classworlds-2.5.2.jar。plexus-classworlds 是一个类加载器框架，与默认的 Java 类加载器相比，它提供了更丰富的语法以方便配置，Maven 使用该框架加载自己的类库。通常无须理会该文件。
- conf：保存 Maven 配置文件的目录，该目录包含 settings.xml 文件，该文件用于设置 Maven 的全局行为。
- lib：该目录包含了所有 Maven 运行时需要的类库，Maven 本身是分模块开发的，因此用户能看到诸如 maven-core-3.5.0.jar、maven-repository-metadata-3.5.0.jar 等文件。此外，还包含 Maven 所依赖的第三方类库。
- LICENSE、NOTICE、README.txt 等说明性文档。

③ 配置 Maven 本地资源库。

在 apache-maven-3.5.0 文件夹下新建文件夹 repository，用于充当本地资源库。打开 apache-maven-3.5.0\conf\setting.xml 文件，在<settings xmlns/>元素下增加

```
<localRepository>D:/apache-maven-3.5.0/repository</localRepository>
```

<localRepository>元素的内容是一个路径字符串，该路径用于设置 Maven 的本地资源库的路径。如果用户不设置该参数，Maven 本地资源库默认保存在用户 Home 目录的 m2/repository 路径下。考虑到 Windows 有可能需要重装、恢复系统，因此建议该 Maven 本地资源库设置到其他路径。

资源库是 Maven 的一个重要概念，Maven 构建项目所使用的插件、第三方依赖库都集中存放在本地资源库中。

1.4.2 Eclipse 集成 Maven

打开 Eclipse，选择 Window→Preferences→Maven→User Settings，如图 1.4 所示。

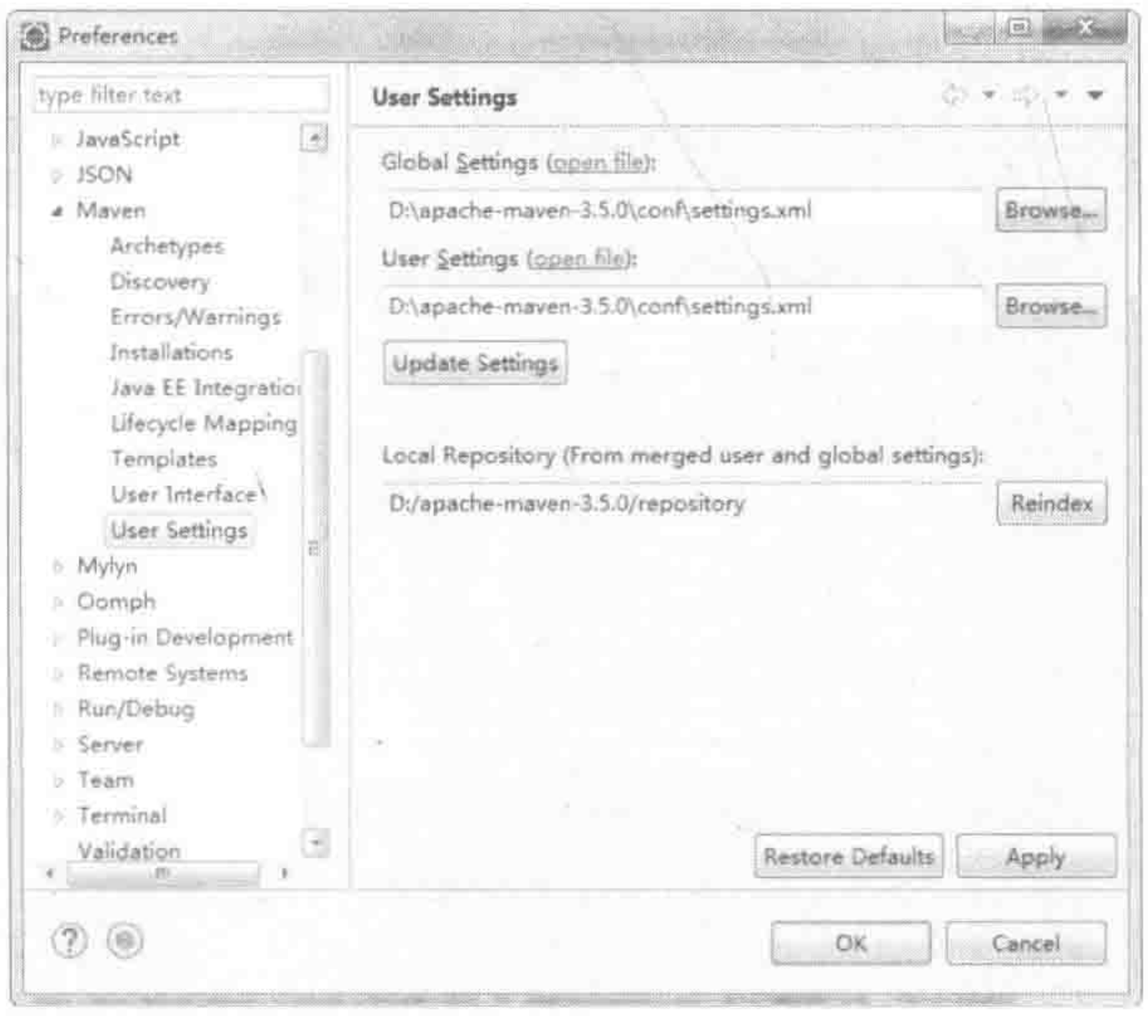


图 1.4 Eclipse 集成 Maven

在 User Settings 窗格中包含如下选项。

- Global Settings：全局设置，此处选择 Maven 的 conf 目录下的 settings.xml。

- User Settings：用户设置，此处选择 Maven 的 conf 目录下的 settings.xml。
- Local Repository：Maven 的本地资源库，即创建的 repository 目录。

1.4.3　示例：第一个 Spring Boot 应用

1.4.3.1　创建一个新的 Maven 项目

打开 Eclipse，选择 File→New→Others→Maven→Maven Project 命令，打开如图 1.5 所示的对话框。

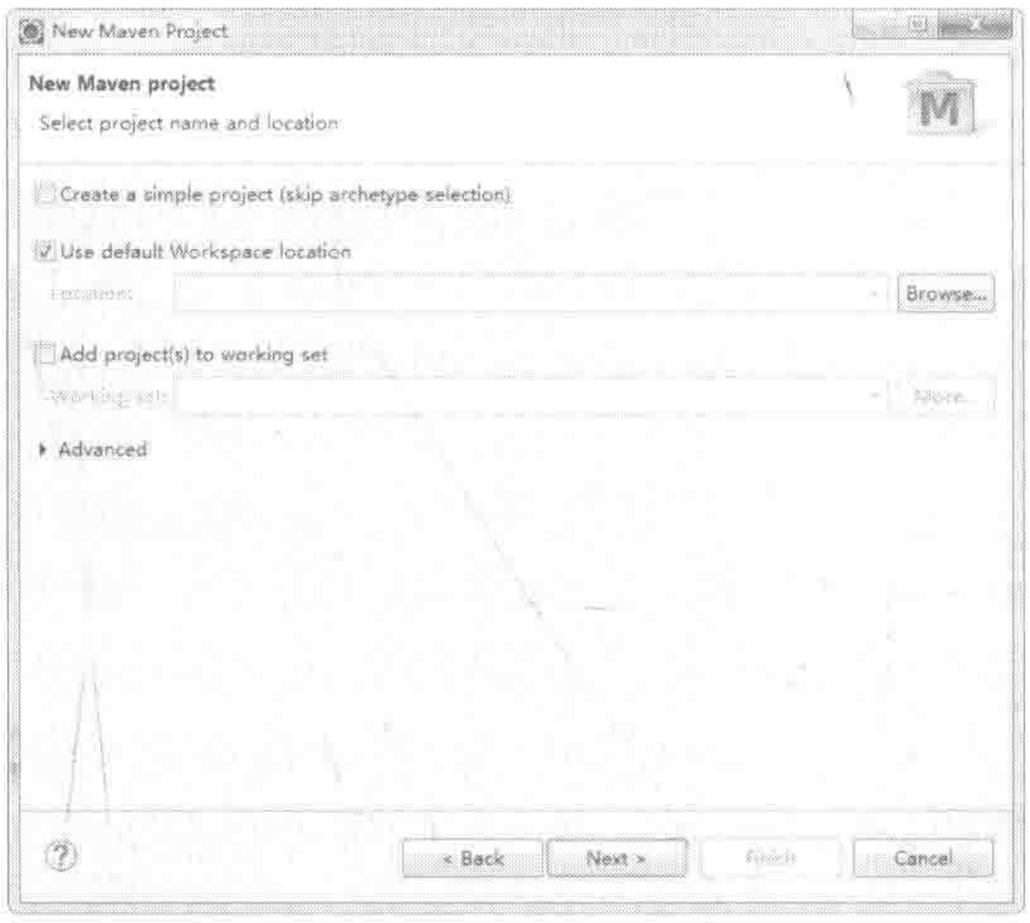


图 1.5　创建新的 Maven 项目

选择项目路径：

- Create a simple project (skip archetype selection)，创建一个简单项目（跳过原型选择）。如果选择该选项将直接跳过 Maven 的项目原型（模板）选择，建议不要勾选，可以使用内置的模板。
- Use default Workspace location，使用默认的工作区间，勾选后，建立的项目将放在默认工作区间，如不选则单击 Browse（浏览）按钮，选择一个工作区间。
- Add project(s) to working set，添加项目到工作集，选中则将新建的项目放入工作集，这里的工作集的概念就是项目归类，类似文件归档一样，方便区分。此选项可选可不选。

单击 Next 按钮，打开如图 1.6 所示的对话框。

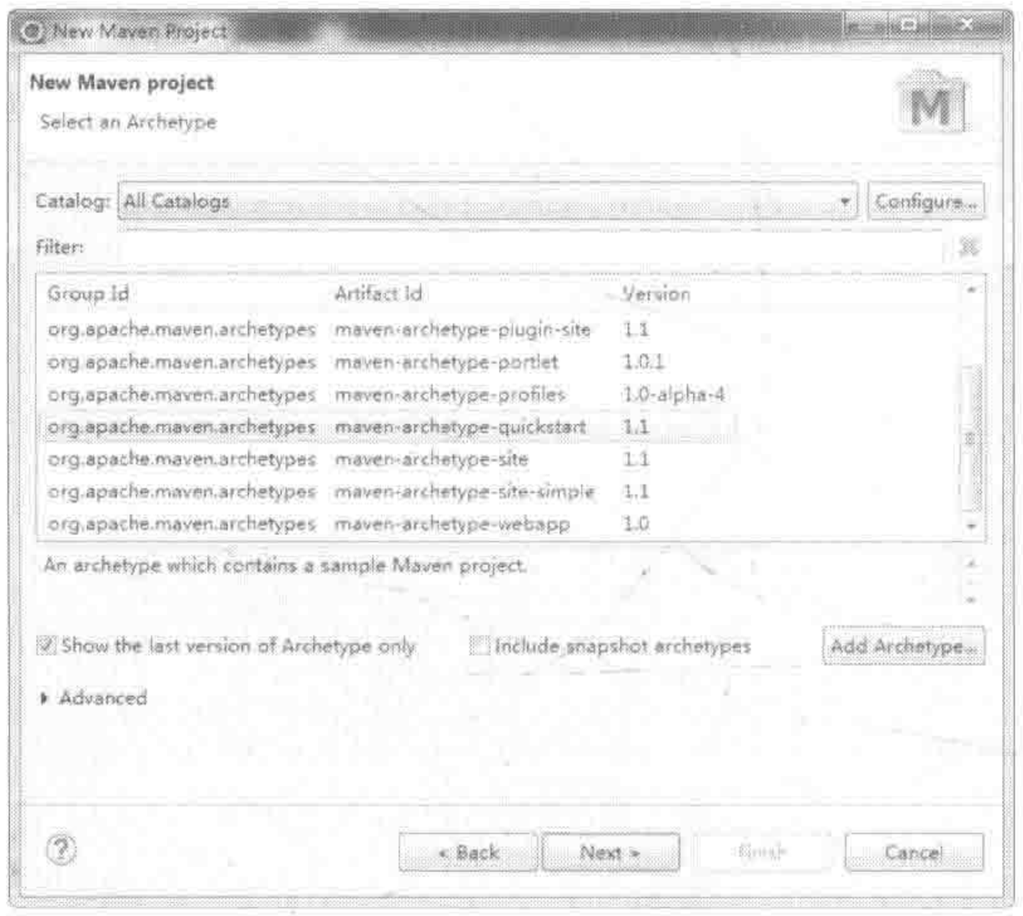


图 1.6　选择项目类型