第1章

课 前 准 备

关于 Maven 的资料和书籍很多。有的比较简洁，点到为止；有的是厚厚的一本书，把方方面面都阐述得很详细；还有些是针对某个方面进行说明讲解。

本书的目的和追求：用通俗易懂的语言，以实际项目为案例，按项目自身发展线索，介绍 Maven 在项目中每个环节的使用方法，使读者由浅入深地了解 Maven、运用 Maven，最后实践与理论结合，掌握 Maven。

1.1 项目经理的工作

一支团队在接到公司下达的项目后，项目经理或项目架构师往往要在团队正式介入以前，做好充分的前期准备。具体工作如下。

(1) 确定系统架构。

(2) 收集框架相关的 jar 包。

(3) 搭建 SSM 框架。

(4) 编写测试代码。

(5) 寻找框架依赖的 jar 包。

(6) 剔除冲突 jar 包。

(7) 制定需求设计文档规范。

(8) 测试文档和代码规范。

(9) 相关报告文档规范。

这里面的内容很多、很杂，必须进行充分细致的准备。一支团队在每承接一个项目时，都要重复做类似的事情。这种工作劳动强度大，而且缺少技术含量。

1.2 Maven 的作用

上述问题用 Maven 可以直接解决。

(1) Maven 统一集中管理好所有的依赖包，不需要程序员再去寻找。

(2) 对应第三方组件用到的共同 jar，Maven 自动解决重复和冲突问题。

(3) Maven 作为一个开放的架构，提供了公共接口，方便同第三方插件集成。程序员可以将自己需要的插件，动态地集成到 Maven，从而扩展新的管理功能。

(4) Maven 可以统一每个项目的构建过程，实现不同项目的兼容性管理。

第2章

开始学习 Maven

2.1 Maven 简介

Maven 是 Apache 开源组织奉献的一个开源项目。Maven 的本质是一个项目管理工具，将项目开发和管理过程抽象成一个项目对象模型(POM)。开发人员只需做一些简单的配置，就可以批量完成项目的构建、报告和文档的生成工作。

当然，Maven 除了是一个优秀的项目构建方面的管理工具外，还有项目管理相关的其他特殊优势。比如，项目相关的第三方依赖包，这是每个 Java 程序员不可回避的问题。

每个老程序员都在自己的计算机里有个专门目录，分类保存过去项目开发过程中使用的第三方 jar 包，需要的时候从里面筛选。新程序员就麻烦了，测试项目的时候，经常会遇到 Class No Found Exception，导致一整天在搜索和重启中度过。

Maven 可以统一管理所有的依赖 jar，甚至是不同的版本。程序员也可以动态地将自己写好的模块打包成 jar 包让它管理。需要的时候，可以直接通过简单的描述文件告诉 Maven，它会自动帮助程序员找出来，集成到项目中。

本书的所有案例都是基于 Maven 3.3.9 版本测试的。

2.2 安装 Maven 前的准备

因为 Maven 本身就是基于 Java 写的，所以在安装配置 Maven 之前，有必要将 Java 的运行环境安装配置好。接下来介绍 Java 的安装和配置过程，有相关基础的读者可以跳过本节。但要注意，Maven 3.3.9 版本要求的 JDK 一定是 JDK 1.7 或以后的版本。

JDK 的安装和配置主要分如下步骤。

2.2.1 下载合适的 JDK 安装软件安装

Oracle 软件的下载链接为 https://www.oracle.com/downloads/index.html。

程序员可以在里面找到需要的 JDK 安装包。本书是在 Windows 操作系统中完成的所有案例，所以请下载 Windows 版本。下载时注意匹配跟自己的系统统一的安装包(32 位或 64 位)。现在的计算机基本上是 64 位，所以建议 Java 相关软件统一安装 64 位的。

安装好后的目录结构如图 2-1 所示。



图 2-1　JDK 安装目录

jdk1.7.0_17_64 是安装 JDK 后的总目录，bin、db、include 等是它里面的子目录和文件。

2.2.2　配置 JDK 环境变量

JDK 的环境变量只要配置 JAVA_HOME 和 Path。右击“我的电脑”，选择“属性”命令，出现如图 2-2 所示窗口。



图 2-2　Windows 系统信息

单击左边的“高级系统设置”按钮，出现如图 2-3 所示对话框。

单击“高级”标签中的“环境变量”按钮，显示如图 2-4 所示对话框。

(1) 配置 JAVA_HOME 目录。查看“系统变量”中有没有 JAVA_HOME(之前有配置过，就会有 JAVA_HOME 这项显示)。如果没有，单击“新建”按钮，在弹出的窗口中输入 JAVA_HOME 和 Java 的安装目录。



图 2-3　系统属性



图 2-4　系统环境变量

（2）配置 Path 目录。同样在配置 JAVA _HOME 的地方查找 Path 变量，如图 2-5 所示。

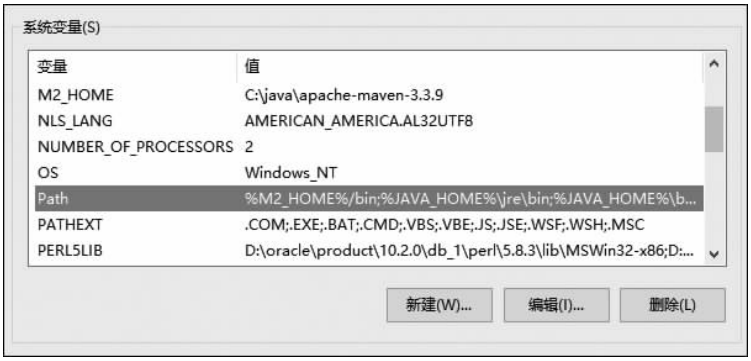


图 2-5　系统变量

单击“编辑”按钮，在以前的 Path 值的前面新添加 JDK 的 bin 目录。这里的 JDK 的 bin 目录是 C:\java\jdk1.7.0_17_64\bin，就是前面的 JAVA_HOME 目录后面添加\bin。添加完成后，同后面的内容要用分号分开，而且是英文输入法的分号。结果如下：

```
C:\java\jdk1.7.0_17_64\bin;以前的内容
```

2.2.3　测试 JDK 是否安装成功

打开一个 CMD 窗口，分别输入 javac -version 命令和 java -version 命令，出现如图 2-6 所示的 Java 版本信息显示，表示 JDK 安装成功。

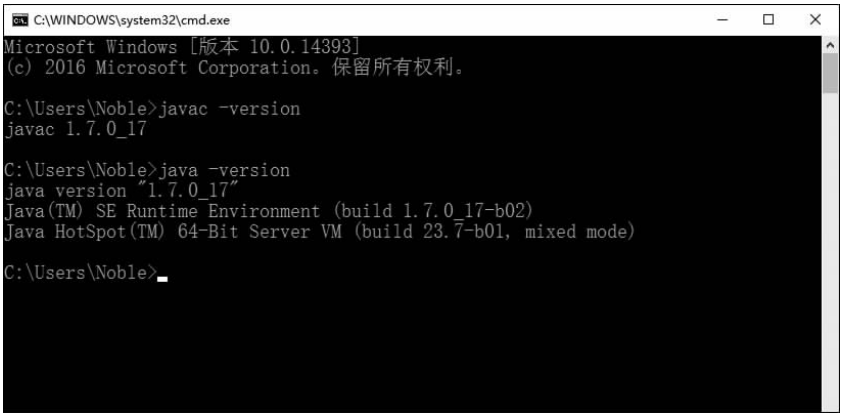


图 2-6　DOS 命令窗口

2.3　Maven 的安装与配置

JDK 环境安装好了，接下来正式安装配置 Maven。先要下载 Maven（http://maven.apache.org/download.cgi）。

它是以压缩包形式提供的，下载 Binary 形式的压缩包就行。

下载完成后，直接用解压工具解压到自己的一个空目录下。这里是解压到 C:\java\apache-maven-3.3.9，最后的目录结构如图 2-7 所示。

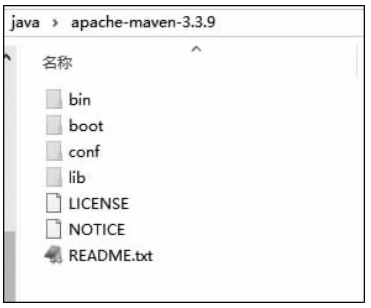


图 2-7　Maven 安装目录

安装好后，接下来对 Maven 进行配置。配置内容同 JDK 的配置内容一样，也是两部分：一部分是在系统环境变量中配置一个 M2_HOME；另一部分也是将 Maven 里面的 bin 目录添加到 Path 环境变量。

（1）配置 M2_HOME。同配置 JAVA_HOME 一样，在系统环境变量中添加 M2_HOME，参考 JDK 的 JAVA _HOME 配置，这里就不再赘述了，效果如图 2-8 所示。

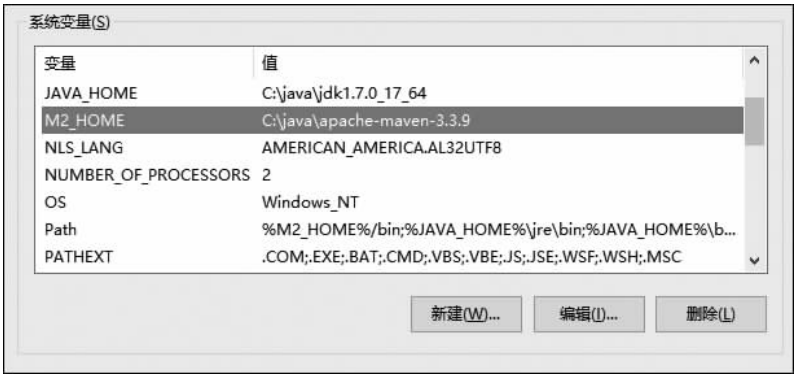


图 2-8　设置 M2_HOME 环境变量

（2）追加 Path。类似前面 JDK Path 环境变量的配置，请参考 JDK 配置，效果如图 2-9 所示。

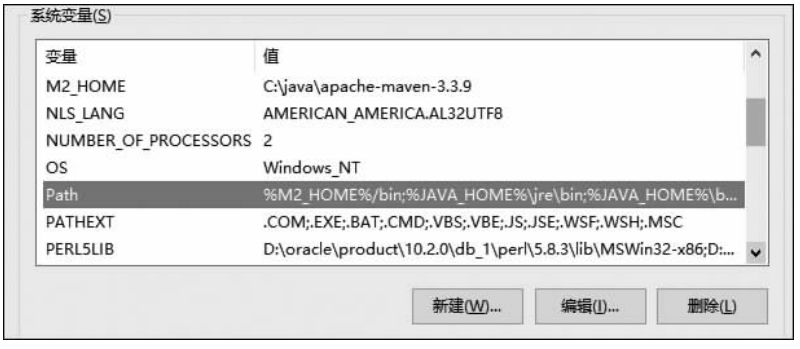


图 2-9　Maven Path 环境变量

其中，%M2_HOME%/bin 是作为 Maven 的 Path 添加的。

（3）测试安装是否成功。打开一个 CMD 窗口（一定要重新打开一个，不能使用配置

环境变量之前的旧 CMD 窗口），输入“mvn -v”命令运行，出现如图 2-10 所示内容表示安装成功。



图 2-10　Maven 安装检测