

第1章

搭建 Java Web 开发环境

俗话说“工欲善其事，必先利其器”，学习 Java Web 开发也离不开好的开发工具。在进行 Java Web 开发之前，需要先搭建开发环境，只有这样才能开发并运行 Java Web 程序。本章介绍如何搭建 Java Web 开发环境，主要内容包括 Web 开发技术、认识 Java Web、JDK 的安装与配置、Java 集成开发环境等。

1.1 Web 开发技术



因为网页分为静态网页和动态网页，所以 Web 开发技术可分为静态 Web 开发技术和动态 Web 开发技术。

1.1.1 静态 Web 开发技术

静态 Web 开发技术只能开发出内容固定不变的网页和网站，现实中常用的 Web 静态技术有 HTML 和 XML 两种。

1. HTML 技术

HTML 文件以<HTML>开头，以</HTML>结束。<head>...</head>之间是文件的头部信息，除了<title>...</title>之间的内容以外，其余内容都不会显示在浏览器上。<body>...</body>之间的代码是 HTML 文件的主体，客户浏览器中显示的内容主要在这里定义。

HTML 是制作网页的基础，大家在现实中所见到的静态网页就是以 HTML 为基础制作的网页。早期的网页都是直接用 HTML 代码编写的，不过现在有很多智能化的网页制作软件（常用的有 FrontPage、Dreamweaver 等）通常不需要人工编写代码，而是由这些软件自动生成代码。尽管不需要人工编写代码，但用户了解 HTML 代码仍然非常重要，这是学习 Web 开发技术的基础。

2. XML 技术

XML 是 eXtensible Markup Language 的缩写，意为可扩展的标记语言。与 HTML 相似，XML 是一种显示数据的标记语言，它能使数据通过网络无障碍地进行传输，并显示在用户的浏览器上。XML 是一套定义语义标记的规则，这些标记将文档分成许多部件并对这些部件加以标识。它也是元标记语言，即定义了用于定义其他与特定领域有关的、语义的、结构化的标记语言的句法语言。

使用上述静态 Web 技术也能实现页面的绚丽效果，并且静态网页相对于动态网页来说，其显示速度比较快。所以，在现实应用中，为了满足页面的特定需求，需要在站点中使用静态

Web 技术来显示访问速度比较快的页面。例如，国内综合站点搜狐和新浪的信息详情页面都使用静态网页技术来显示。

1.1.2 动态 Web 开发技术

静态 Web 技术只能实现页面内容的简单显示，而不能实现页面的交互效果。随着网络技术的发展和现实需求的提高，静态 Web 技术越来越不能满足客户的需求，因此更新、更高级的网页技术便登上了 Web 领域的舞台，这就是动态 Web 开发技术。

除了本书讲解的 Java Web 技术以外，现实中常用的 Web 动态技术还有 ASP、ASP.NET、PHP 和 JSP 等。

1. ASP 技术

ASP（Active Server Pages，动态网页）是微软公司推出的一种用于取代 CGI 的技术。ASP 具有微软操作系统的强大普及性，一经推出，便迅速成为最主流的 Web 开发技术。

ASP 是 Web 服务器端的开发环境，利用它可以创建和执行动态、高效、交互的 Web 服务应用程序。ASP 技术是 HTML、Script 与 CGI 的结合体，其运行效率却比 CGI 更高，程序的编写也比 HTML 更方便且更有灵活性。

2. PHP 技术

PHP 也是流行的生成动态网页的技术之一。PHP 是完全免费的，可以从 PHP 官方站点（<http://www.php.net>）自由下载。用户可以不受限制地获得 PHP 源代码，甚至可以加入自己需要的特色。PHP 在大多数 UNIX 平台、GUN/Linux 平台和微软公司的 Windows 平台上均可以运行。

3. JSP 技术

JSP 是 Sun 公司为创建高度动态的 Web 应用提供的一个独特的开发环境。和 ASP 技术一样，JSP 提供了在 HTML 代码中混合某种程序代码，由语言引擎解释执行程序代码的能力。JSP 技术是本书讲解的 Java Web 技术的一部分。

4. ASP.NET 技术

ASP.NET 是微软公司动态服务网页技术的最新版本，提供了一个统一的 Web 开发模型，其中包括开发人员生成企业级 Web 应用程序所需的各种服务。ASP.NET 的语法在很大程度上与 ASP 兼容，同时它还提供了一种新的编程模型和结构，可以生成伸缩性和稳定性更好的应用程序，并提供更好的安全保护。



微视频

1.2 认识 Java Web

Java Web 是 Java 技术的一个分支应用，是指利用 Java 开发 Web 项目。由此可见，要想学好 Java Web，首先要了解 Java 语言的相关知识。纵观当今各大主流招聘媒体，大家总是会看到多条招聘 Java 程序员的广告，由此可以看出 Java 程序员很受市场欢迎。

1.2.1 Java 语言介绍

Java 是一门跨平台的、面向对象的高级程序设计语言，Java 程序能够在不同的计算机、不同的操作系统中运行，甚至在支持 Java 的硬件上也能正常运行。

Java 是于 1995 年由 Sun 公司推出的一种面向对象的程序设计语言。它由被称为“Java 之父”的 Sun 研究院院士 James Gosling（詹姆斯·高斯林）亲手设计而成。Java 最初的名称为 OAK，1995 年被重命名为 Java 后正式发布。

Java 是一种通过解释方式来执行的语言，其语法规则和 C++ 类似。与 C++ 不同的是，它摒弃了 C++ 中难以理解的多继承、指针等概念，这使得 Java 语言简洁了很多，而且还提高了语言的可靠性与安全性，可以说 Java 是一门非常卓越的编程语言。

Java 的口号是“Write Once, Run Anywhere”，这体现了 Java 语言的跨平台特性，所以 Java 常被应用于企业网络和 Internet 环境中。当前，Sun 公司被 Oracle（甲骨文）公司收购，Java 也随之成为 Oracle 公司的产品。

1.2.2 Java Web 体系介绍

Java Web 是指利用 Java 语言开发 Web 项目的一种技术，和 ASP、ASP.NET、PHP 等功能类似。如果要开发 Java Web 应用，就需要掌握一系列相关的技术，必要时可应用相应的框架，以提高 Java Web 应用的开发效率，并确保其可扩展性与可维护性。

Java Web 应用开发的主要技术包括 HTML、XML、JavaScript、Java、JDBC、JSP、JavaBean、Servlet、Ajax 等。

（1）JDBC：JDBC（Java DataBase Connectivity，Java 数据库连接）是一种用于执行 SQL 语句的 Java API（Application Programming Interface，应用编程接口或应用程序编程接口），由一组用 Java 语言编写的类和接口组成，可让程序员以 Java 的方式连接数据库，并完成相应的操作。

（2）Servlet：Servlet 是一种用 Java 编写的与平台无关的服务器端组件，其实例化后的对象运行在服务器端，可用于处理来自客户端的请求，并生成相应的动态网页。

（3）JavaBean：JavaBean 是 Java 中的一种可重用组件技术（类似于微软公司的 COM 技术），其本质是一种通过封装属性和方法而具有某种功能的 Java 类。

（4）Ajax：Ajax（Asynchronous JavaScript and XML，异步 JavaScript 和 XML）是一种创建交互式网页应用的开发技术，其核心理念为使用 XMLHttpRequest 对象发送异步请求，可实现 Web 页面的动态更新。

（5）JSP：JSP 与 ASP、PHP 类似，是目前 Web 应用开发的主流技术之一。作为一种动态 Web 技术标准，JSP 在 Java Web 应用开发中的使用是相当普遍的。

1.3 JDK 的安装与配置



学习 Java Web 需要一个编译和运行的环境，因此进行 JDK（Java Development Kit）的下载、安装和环境配置是必须的。

1.3.1 JDK 的下载

JDK 是整个 Java 的核心，包括 Java 运行环境 JRE、Java 工具和 Java 基础类库，下面详细介绍 JDK 的下载，下载 JDK 的具体步骤如下。

步骤 1：在浏览器的地址栏中输入网址“<https://www.oracle.com/java/technologies/javase->

downloads.html”，按 Enter 键确认，进入 JDK 的下载页面，选择最新版本。单击 JDK Download 下载链接，如图 1-1 所示。

步骤 2：进入下载文件选择页面，根据操作系统和需求选择适合的版本。这里选择 Windows x64 Installer 版本，如图 1-2 所示。

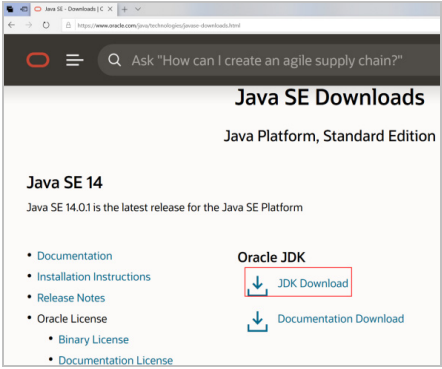


图 1-1 JDK 下载页面

Product / File Description	File Size	Download
Linux Debian Package	15792 MB	jdk-14.0.1_linux-x64_bin.deb
Linux RPM Package	165.04 MB	jdk-14.0.1_linux-x64_bin.rpm
Linux Compressed Archive	182.04 MB	jdk-14.0.1_linux-x64_bin.tar.gz
macOS Installer	175.77 MB	jdk-14.0.1_osx-x64_bin.dmg
macOS Compressed Archive	176.19 MB	jdk-14.0.1_osx-x64_bin.tar.gz
Windows x64 Installer	162.07 MB	jdk-14.0.1_windows-x64_bin.exe
Windows x64 Compressed Archive	181.53 MB	jdk-14.0.1_windows-x64_bin.zip

图 1-2 选择安装版本

☆大牛提醒☆

在图 1-2 中，Installer 表示安装版本，安装过程由系统自动配置；Compressed Archive 表示压缩版本，安装过程需要用户自己配置。

步骤 3：选择版本之后进入下载页面，选中“reviewed and accept...”协议复选框，然后单击 Download jdk-14.0.1_windows-x64_bin.exe 按钮进行下载，如图 1-3 所示。

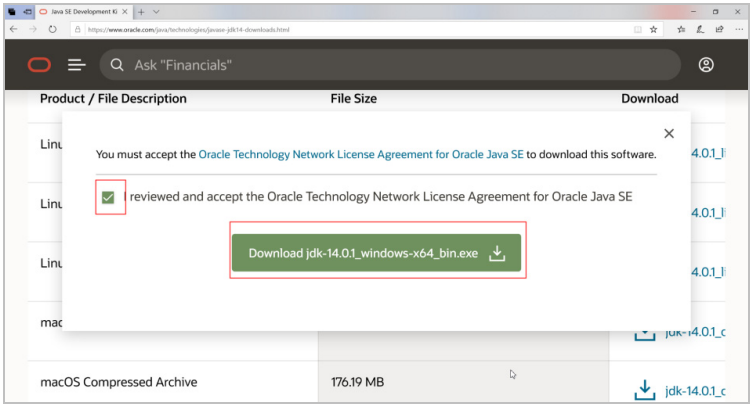


图 1-3 选择接受协议并下载 JDK

1.3.2 JDK 的安装

JDK 安装包下载完毕后就可以进行安装，具体安装步骤如下。

步骤 1：双击下载的 jdk-14.0.1_windows-x64_bin.exe 文件，进入“安装程序”对话框，单击“下一步”按钮，如图 1-4 所示。

步骤 2：弹出“目标文件夹”对话框，用户可根据自己的需要更改安装路径，然后单击“下一步”按钮，如图 1-5 所示。



图 1-4 “安装程序”对话框



图 1-5 “目标文件夹”对话框

步骤 3: JDK 开始自动安装。安装成功后进入“完成”对话框，提示用户 Java 已成功安装，单击“关闭”按钮即可完成 JDK 的安装，如图 1-6 所示。

1.3.3 JDK 环境配置

在 JDK 安装完成后，还需要配置环境变量才能使用 Java 开发环境。这里配置环境变量 Path，具体实现步骤如下。

步骤 1: 在桌面上选择“此电脑”图标，然后右击，在弹出的快捷菜单中选择“属性”菜单命令，打开“系统”窗口，如图 1-7 所示。

步骤 2: 单击“高级系统设置”选项，弹出“系统属性”对话框，切换到“高级”选项卡，单击“环境变量”按钮，如图 1-8 所示。



图 1-6 “完成”对话框

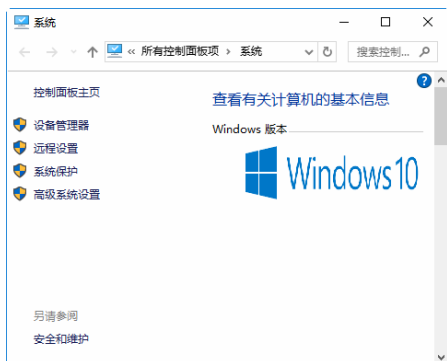


图 1-7 “系统”窗口



图 1-8 “系统属性”对话框

步骤 3: 弹出“环境变量”对话框，在“系统变量 (S)”列表框中选择 Path 变量，如图 1-9 所示。

步骤 4: 双击 Path 变量，弹出“编辑环境变量”对话框，单击“编辑文本(T)”按钮，如图 1-10 所示。

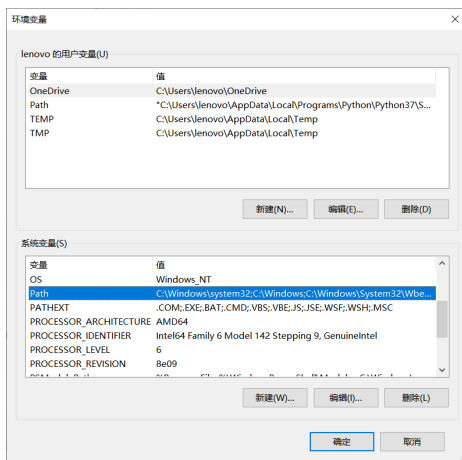


图 1-9 “环境变量”对话框

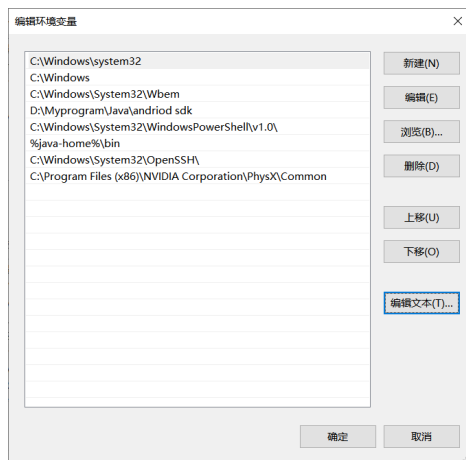


图 1-10 “编辑环境变量”对话框

步骤 5: 打开“编辑系统变量”对话框，在“变量值”参数的最前面加入 JDK 安装路径下的 bin 文件路径，这里所加的路径为“E:\20200526\java\bin;”，如图 1-11 所示。

步骤 6: 单击“确定”按钮，返回“环境变量”对话框，这时可以发现 Path 变量的最前面有刚添加的路径“E:\20200526\java\bin;”，如图 1-12 所示。最后单击“确定”按钮，即可完成 JDK 环境配置。

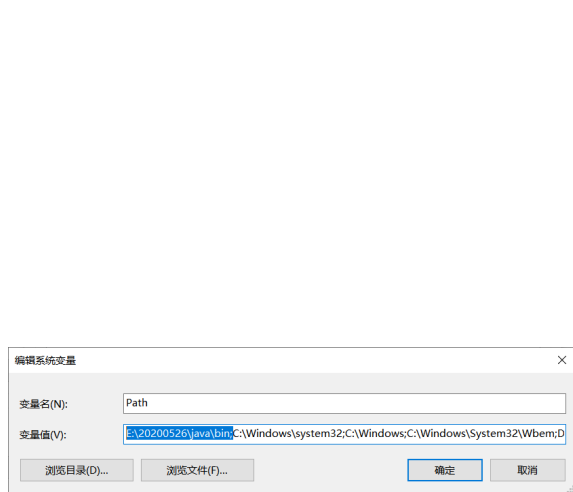


图 1-11 “编辑系统变量”对话框

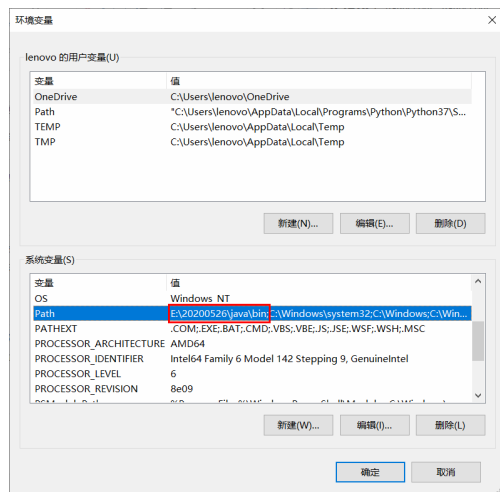


图 1-12 “环境变量”对话框

1.3.4 测试开发环境

完成 JDK 的安装并成功配置环境后，需要测试一下配置的准确性，具体操作步骤如下。

步骤 1: 右击“开始”按钮，在弹出的快捷菜单中选择“运行”菜单命令，打开“运行”对话框，在“打开”文本框中输入“cmd”命令，如图 1-13 所示。

步骤 2: 单击“确定”按钮，即可打开“命令提示符”窗口，在其中输入命令“javac”并

按下 Enter 键,即可显示 JDK 的编译器信息,这说明开发环境配置成功,如图 1-14 所示。

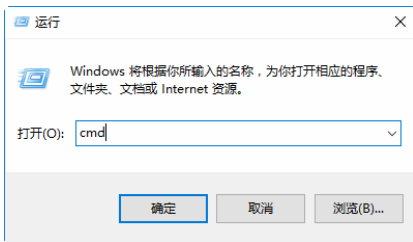


图 1-13 “运行”对话框

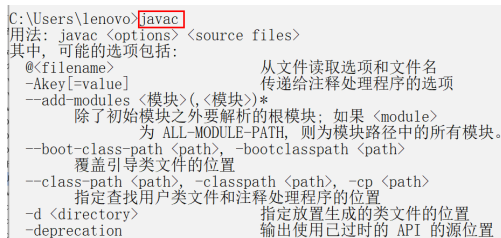


图 1-14 JDK 编译器信息

1.4 我的第一个 Java 程序



微视频

在 Java 开发环境配置好以后,下面用第一个 Java 程序——输出文字“Hello! Java!”来体验一下 Java 语言的魅力。

编写“Hello! Java!”程序的具体步骤如下。

步骤 1: 新建记事本文件,输入如图 1-15 所示的内容,并保存成“hello.java”文件。

☆大牛提醒☆

“java”文件名与代码中的 class (类) 名字必须是一致的,如图 1-15 所示。另外,由于 Java 是解释性语言,所以这里可以用记事本来编写 Java 代码。

步骤 2: 程序写好之后,现在开始编译该程序的代码。打开“命令提示符”窗口,进入 hello.java 所在的文件夹,输入编译命令“javac hello.java”,按下 Enter 键,这时在 hello.java 文件所在的目录下会生成一个 class 文件,如图 1-16 所示。

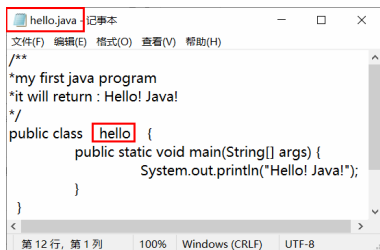


图 1-15 第一个 Java 程序的代码

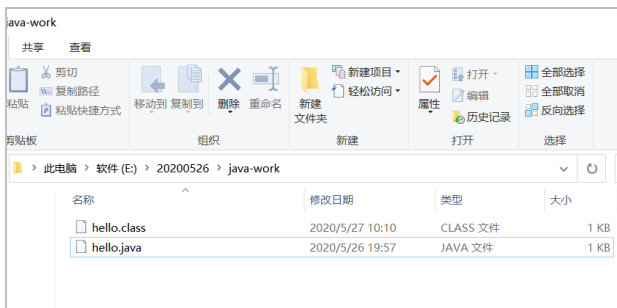


图 1-16 编译之后生成 class 文件

步骤 3: 程序代码编译好之后,现在开始执行程序输出相应的内容。在“命令提示符”窗口中程序文件所在的路径下输入执行命令“java hello”,按下 Enter 键,这时就会输出 hello.java 程序中的内容——“Hello! Java!”,如图 1-17 所示。

☆大牛提醒☆

在 hello.java 程序中用到了 System.out.println() 函数,当使用该函数输出文字时,必须将所输出文字用英文双引号引起来,例如 System.out.println("Hello! Java! ")。

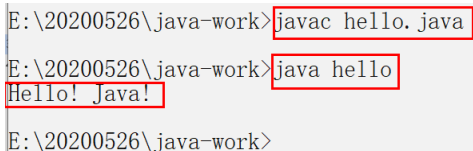


图 1-17 用记事本编写的 Java 程序的运行结果



微视频

1.5 选择 Java 开发工具

当前主流的 Java 开发工具是 Eclipse，它不仅免费，而且功能齐全，使用起来非常方便且易上手。下面详细介绍 Eclipse 的下载、安装与使用。

1.5.1 Eclipse 的下载

用户可以到 Eclipse 工具的官方网站下载 Eclipse 开发工具，具体步骤如下。

步骤 1：在浏览器的地址栏中输入网址“<https://www.eclipse.org/downloads>”，进入 Eclipse 下载页面，单击 Download Packages 超链接，如图 1-18 所示。

步骤 2：进入版本选择页面，这里选择 Eclipse IDE for Enterprise Java and Web Developers 选项，并根据自己的系统需求来选择相应的版本，这里选择 windows 64 bit，如图 1-19 所示。

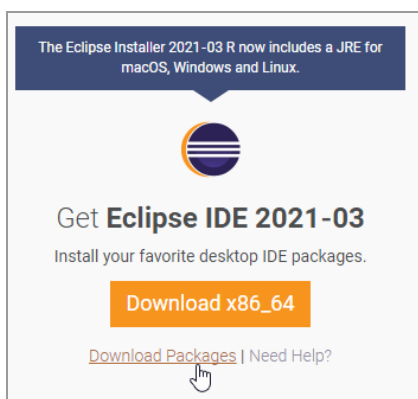


图 1-18 Eclipse 下载首页

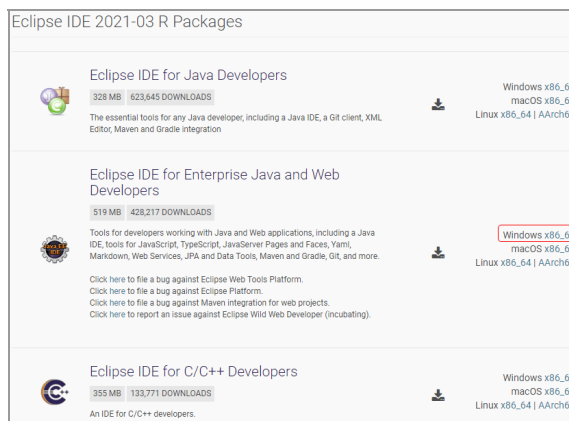


图 1-19 Eclipse 版本选择页

步骤 3：单击相应的版本进入下载页，然后单击 Download 按钮进行下载，如图 1-20 所示。

☆ 大牛提醒 ☆

如果下载不成功，或者很久都没有下载提示，可以单击 Select Another Mirror 超链接，选择更多的镜像来下载，如图 1-21 所示。不过，如果网络环境好，默认镜像就能成功下载。



图 1-20 Eclipse 下载页

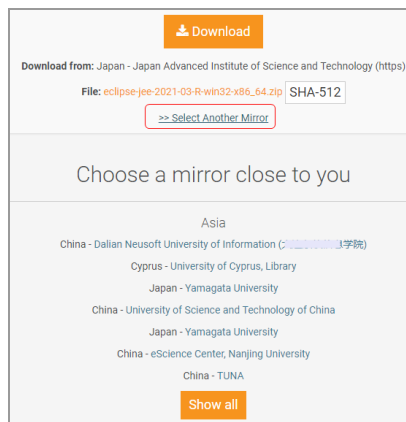


图 1-21 镜像选择页面

1.5.2 Eclipse 的安装与配置

将下载好的压缩文件解压到自己指定的文件夹，然后运行文件 `eclipse.exe`，会弹出工作空间目录选择页面，也就是在 Eclipse 上创建 Java 项目文件存放的位置，这里根据自己的需要更改文件夹的路径，并将下面的默认路径选上，这样就不需要每次运行 Eclipse 时再确认工作空间路径了，如图 1-22 所示。

单击 **Launch** 按钮进入 Eclipse 工作台欢迎界面，如图 1-23 所示，这样就完成了 Eclipse 开发工具的配置，以后再运行 `eclipse.exe` 文件就可以直接进入和使用 Eclipse 开发工具。

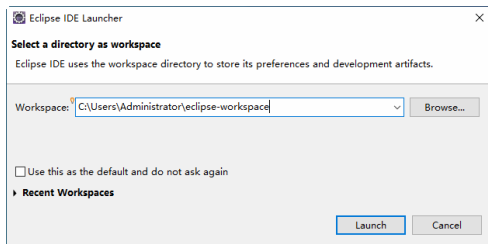


图 1-22 Eclipse 工作空间设置

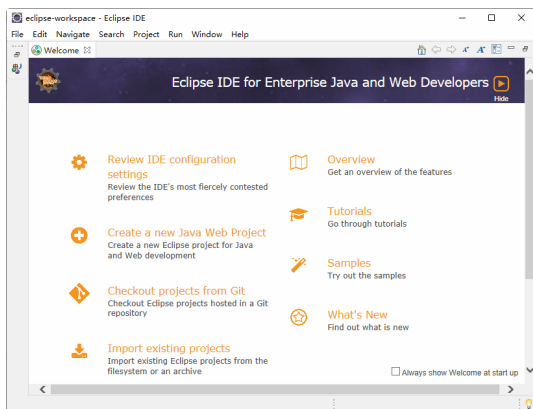


图 1-23 Eclipse 欢迎界面

启动并运行 Eclipse 开发工具，进入如图 1-24 所示的工作界面。

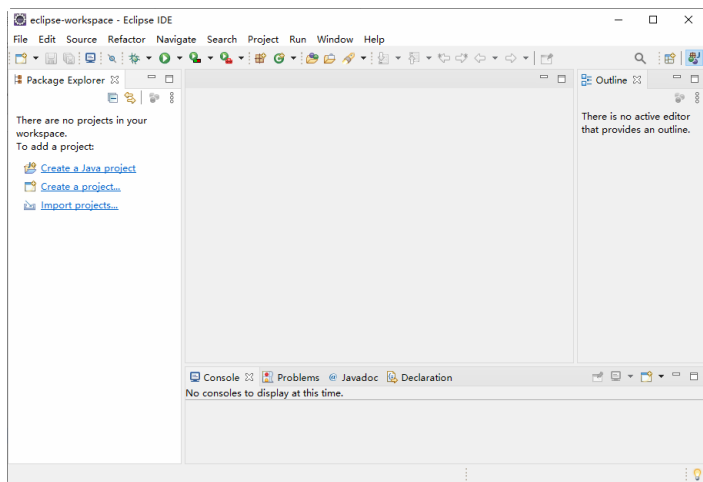


图 1-24 Eclipse 工作界面

1.5.3 用 Eclipse 创建 Java 项目

成功安装和配置好 Java 及 Eclipse 程序以后，学习者就可以更轻松地学习 Java，现在开始 Eclipse 下的 Java 学习。这里创建一个 Java 项目，具体步骤如下。

步骤 1：单击项目视图界面中的 **Create a Java project** 超链接，如图 1-25 所示，或者选择 **File→New→Java Project** 菜单命令，如图 1-26 所示。

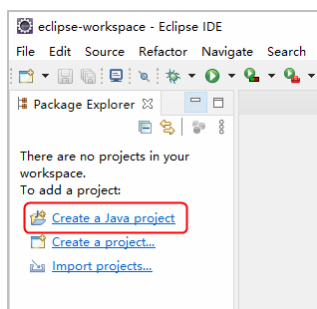


图 1-25 快捷创建项目

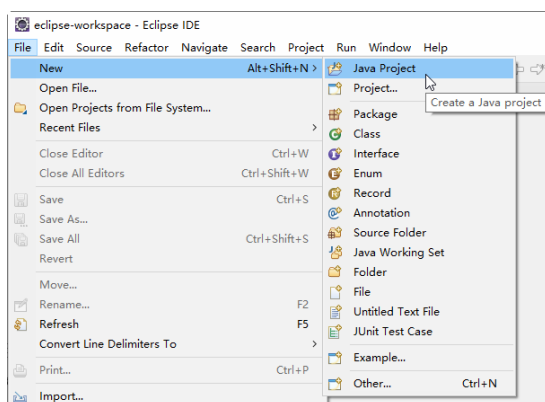


图 1-26 通过菜单创建项目

步骤 2: 打开 New Java Project 对话框, 在其中输入 Java 项目的名称, 其他设置可以选择默认, 单击 Finish 按钮, 如图 1-27 所示。

步骤 3: 打开 New module-info.java 对话框, 在其中可以输入程序模块化的名称, 不过在学习初期模块化文件没有必要, 还有可能影响 Java 项目的运行, 所以这里不建议创建程序模块, 如图 1-28 所示。

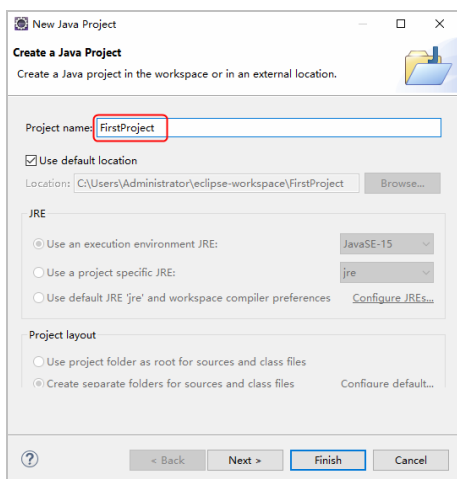


图 1-27 Java 项目命名对话框

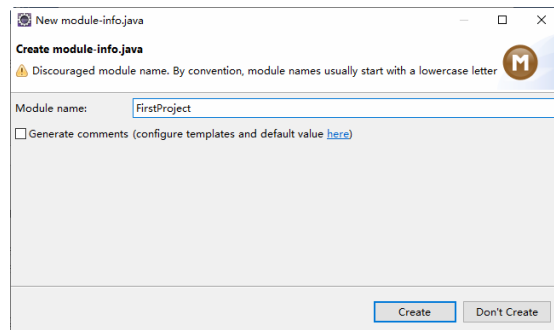


图 1-28 项目模块命名对话框

步骤 4: 单击 Don't Create 按钮, 即可实现 Java 项目的创建, 如图 1-29 所示。

1.5.4 创建 Java 类文件

在创建 Java 类文件时会自动打开 Java 编辑器, 创建 Java 类文件的具体操作步骤如下。

步骤 1: 选择 File→New→Class 菜单命令, 如图 1-30 所示。

步骤 2: 弹出 New Java Class 对话框, 在 Package 文本框中输入文件包的名称“myPackage”, 在 Name 文本框中输入类名称“HelloWorld”, 并选中 public static void main(String[] args)复选

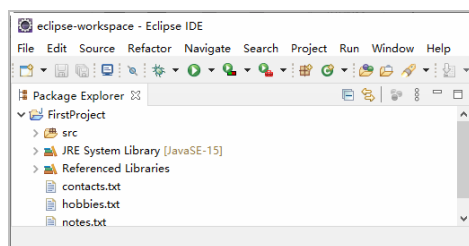


图 1-29 项目创建成功界面

框，以保证所创建的类是能运行的主类，然后单击 Finish 按钮完成创建 Java 类文件的操作，如图 1-31 所示。

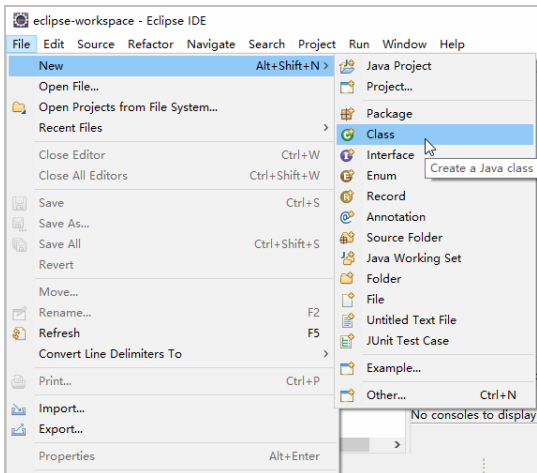


图 1-30 创建新的 Java 类文件

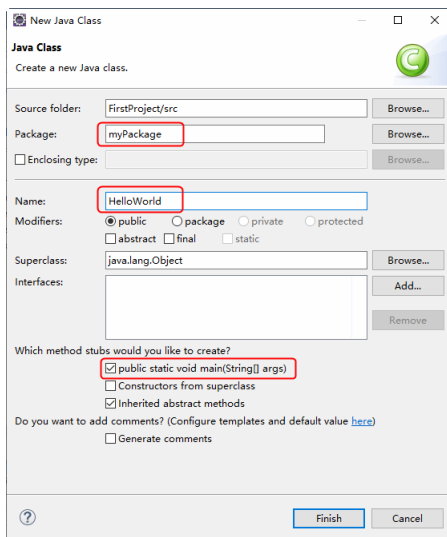


图 1-31 为 Java 类文件命名并设置

1.5.5 编写和运行 Java 程序

创建的 Java 类文件会在 Eclipse 的编辑区中打开，该区域可以重叠放置多个文件进行编辑。编辑前面创建的 Java 类文件进行 Java 程序的编写，Eclipse 在运行 Java 程序时会先自动编译，再运行输出相应的程序内容，具体操作步骤如下。

步骤 1：编写 Java 程序。在 Java 类文件中输入代码，按快捷键 Ctrl+S 进行保存，这里所编写代码的功能是输出“Hello World！”，如图 1-32 所示。

步骤 2：运行 Java 程序。可以直接单击  按钮，或者选择 Run→Run 菜单命令来运行 Java 程序，如图 1-33 所示。

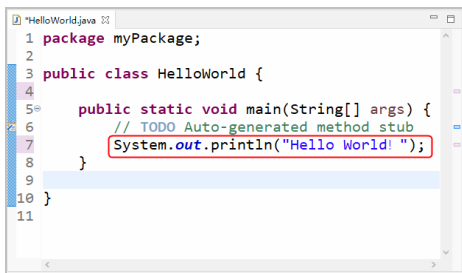


图 1-32 编写 Java 程序

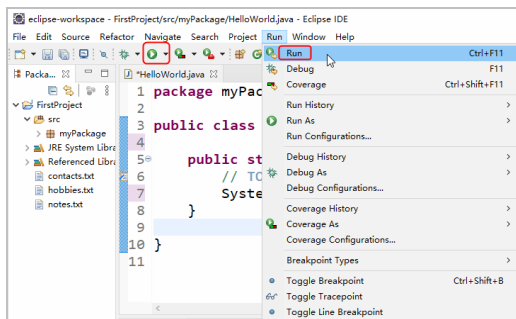


图 1-33 运行 Java 程序

步骤 3：运行结果在下面的 Console 视图界面中显示，如图 1-34 所示。

☆ **大牛提醒** ☆

使用 Eclipse 编写程序要比使用记事本省事得多，而且 Eclipse 能够提示代码书写，这就加快了代码的编写并提高了代码书写的正确率。

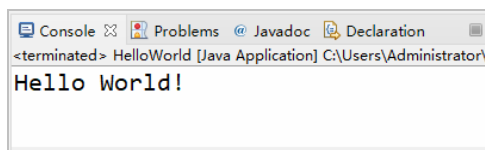


图 1-34 Eclipse 下第一个 Java 程序的运行结果

1.6 新手疑难问题解答

问题 1：环境变量配置后，在命令提示符中仍找不到“javac”命令？

解答：这是环境配置的问题，需要重新进行环境变量的配置，以确保新添加的 JDK 的 bin 路径在 Path 变量的最前面，且与原变量内容以英文分号“;”隔开。

问题 2：在保存 Java 源代码后，为什么会出现不正确的 Java 源文件名？

解答：初学者在保存 Java 源代码时会出现这样的问题，即用“TestGreeting.java”作为文件名保存的，但在磁盘对应目录中查看时文件名却成为“TestGreeting.java.txt”，这是不正确的 Java 源文件名。这需要对 Windows 系统进行如下配置：双击“此电脑”，在打开的“此电脑”窗口中切换到“查看”选项卡，在“显示/隐藏”设置区域中将“文件扩展名”设置为未选中状态。

1.7 实战训练

实战 1：输出“乾坤未定！你我皆是黑马！”。

编写程序，在窗口中输出“乾坤未定！你我皆是黑马！”，程序运行效果如图 1-35 所示。

实战 2：打印星号字符图形。

编写程序，在窗口中输出用星号组成的三角形，效果如图 1-36 所示。

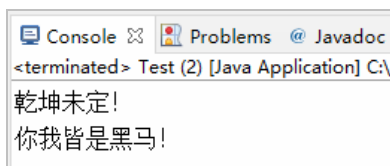


图 1-35 输出信息

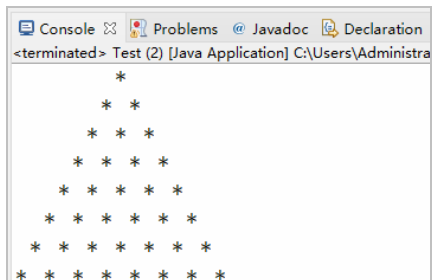


图 1-36 打印星号字符图形

第 2 章

Web 服务器的搭建

随着互联网的普及和电子商务的迅速发展，人们对站点的要求越来越高，为此开发动态、高效的 Web 站点已经成为社会发展的需求，这样就需要用到 Web 服务器。常用的 Web 服务器有很多，本章以 Tomcat 服务器为例来介绍 Web 服务器的搭建过程。

2.1 Web 开发背景知识

学习 Web 开发，不得不提本地计算机和远程服务器的概念。顾名思义，本地计算机是指用户正在使用的、浏览站点页面的计算机。对于本地计算机来说，最重要的构成模块是 Web 浏览器。



微视频

2.1.1 Web 浏览器

浏览器是 WWW 系统的重要组成部分，它是运行在本地计算机上的程序，负责向服务器发送请求，并且将服务器返回的结果显示给用户。用户可以通过浏览器窗口来分享网上丰富的资源。常见的网页浏览器包括 Internet Explorer、Firefox、Opera、Google Chrome 等。如图 2-1 所示为 Google Chrome 浏览器的工作界面。



图 2-1 Google Chrome 浏览器的工作界面

2.1.2 远程服务器

远程服务器是一种高性能计算机，它作为网络的节点，存储、处理网络上大约 80% 的数据、信息，因此也被称为网络的灵魂。远程服务器是网络上一种为客户端计算机提供各种服务的高性能的计算机，它在网络操作系统的控制下，将与其相连的硬盘、打印机、Modem 及各种专用通信设备提供给网络上的客户站点共享，也能为网络用户提供集中计算、信息发表及数据管理等服务。它的高性能主要体现在高速度的运算能力、长时间的可靠运行、强大的外部数据吞吐能力等方面。如图 2-2 所示为本地计算机和远程服务器的工作流程。

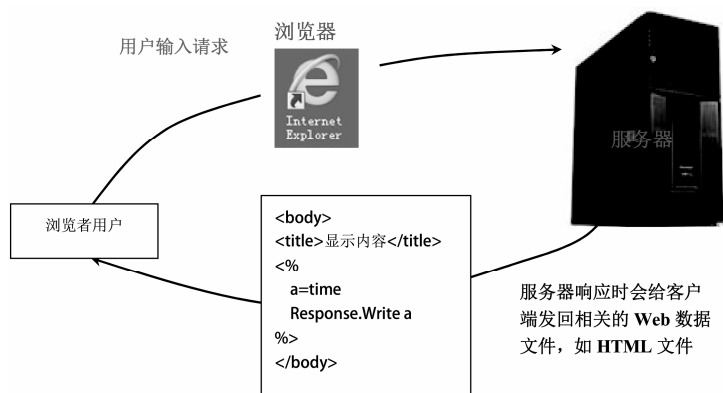


图 2-2 本地计算机和远程服务器的工作流程

2.1.3 Web 应用程序的工作原理

用户访问互联网资源的前提是必须首先获取站点的地址，然后通过页面链接来浏览具体页面的内容。上述过程是通过浏览器和服务端进行的。下面以访问搜狐网为例，了解 Web 应用程序的工作原理。

- (1) 在浏览器的地址栏中输入搜狐网的首页地址“<http://www.sohu.com>”。
- (2) 用户浏览器向服务器发送访问首页的请求。
- (3) 服务器获取客户端的访问请求。
- (4) 服务器处理请求。如果请求的页面是静态文档，则只需将此文档直接传送给浏览器；如果是动态文档，则将处理后的静态文档发送给浏览器。
- (5) 服务器将处理后的结果在客户端浏览器中显示。

2.1.4 Web 服务器简介

Web 服务器一般是指网站服务器，可以向浏览器等 Web 客户端提供文档。Web 服务器不仅能够存储信息，还能在用户通过 Web 浏览器提供的信息的基础上运行脚本和程序。

Web 服务器不仅可以放置网站文件，让全世界网友浏览，还可以放置数据文件，让全世界网友下载。常用的 Web 服务器有很多，在本节简单介绍便于 Java Web 使用的 Tomcat、Nginx 和 Jetty 服务器。

1. Tomcat 服务器

Tomcat 服务器是一款免费开放源代码的 Web 应用服务器。该服务器是由 Apache 开发的一个 Servlet 容器，实现了对 Servlet 和 JSP 的支持，并提供了作为 Web 服务器的一些特有功能，例如 Tomcat 管理和控制平台、安全域管理和 Tomcat 阀等。Tomcat 服务器属于轻量级应用服务器。

Tomcat 的官方网站下载地址为“<http://tomcat.apache.org>”，如图 2-3 所示。

2. Nginx 服务器

Nginx 服务器是一款高性能且功能丰富的 Web 服务器，可作为 HTTP 服务器，也可作为反向代理服务器和邮件服务器。Nginx 的官方网站下载地址为 <http://www.nginx.org/>，如图 2-4 所示。

Nginx 服务器的特点是占有的内存少，并发能力强，在连接高并发的情况下，Nginx 是 Apache 服务器不错的替代品，能够支持高达 50 000 个并发连接数的响应。

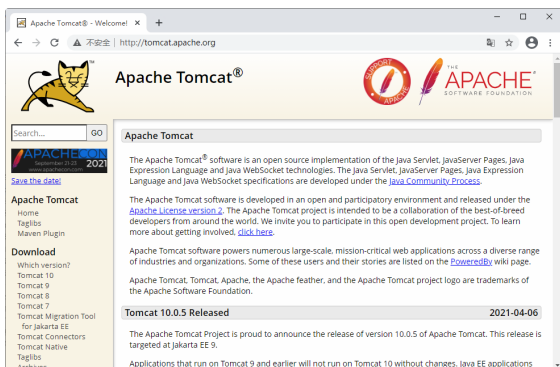


图 2-3 Tomcat 服务器下载页面

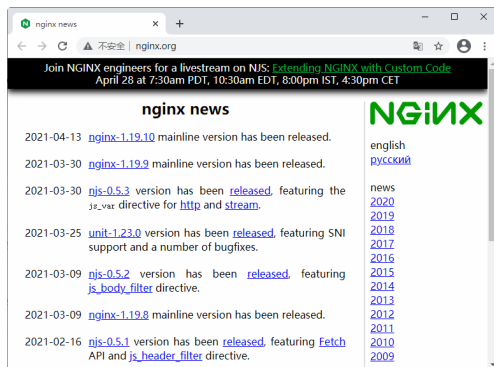


图 2-4 Nginx 服务器下载页面

3. Jetty 服务器

Jetty 服务器是目前比较被看好的一款 Servlet 服务器。该服务器的架构比较简单，但在可扩展性方面表现得非常灵活。它有一个基本数据模型，这个数据模型就是 Handler，所有可以被扩展的组件都可以作为一个 Handler 添加到 Server 中。Jetty 就是帮助管理这些 Handler 数据模型，以便于更迅速地开发。

Jetty 的官方网站下载地址为“<https://www.eclipse.org/jetty/download.php>”，如图 2-5 所示。

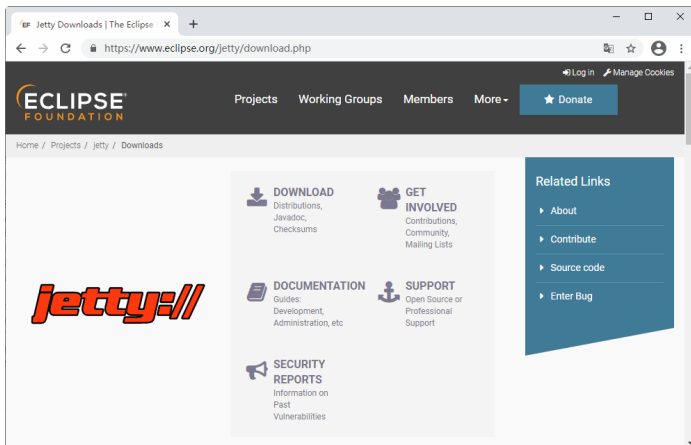


图 2-5 Jetty 服务器下载页面

由于 Tomcat 服务器技术先进、性能稳定且免费，深受广大 Java 爱好者的喜爱，同时也得到了部分软件开发商的认可，因此成为目前比较流行的 Web 应用服务器。接下来以 Tomcat 服务器为例，学习 Web 服务器的搭建、启动及配置方法。

2.2 Tomcat 的下载与安装

Tomcat 的安装有两种方式，一种是解压之后不需要安装就可以直接使用的方式，称为解压版；另一种是应用程序需要安装之后才能使用的方式，称为安装版。在介绍 Tomcat 的安装方法之前，先了解一下 Tomcat 各个版本的区别，以帮助读者更好地选择适合自己的软件版本。



微视频

2.2.1 了解 Tomcat 版本的区别

当前 Tomcat 服务器主要包含 Tomcat 10、Tomcat 9 和 Tomcat 8 等版本。

1. Tomcat 10 版本

Tomcat 10 是较新版本，建立在 Tomcat 9 版本的基础之上。Tomcat 10 及更高版本的用户应注意，作为从 Java EE 到 Eclipse Foundation 迁移的一部分，从 Java EE 迁移到 Jakarta EE 的结果是所有已实现 API 的主要软件包已从 `javax.*` 更改为 `jakarta.*`，以使应用程序能够从 Tomcat 9 及更低版本迁移到 Tomcat 10 及更高版本。这包括以下功能改进：

- (1) 更新到 Jakarta Servlet 5.0、Jakarta Server Pages 3.0、Jakarta Expression Language 4.0、Jakarta WebSocket 2.0、Jakarta Authentication 2.0 和 Jakarta Annotations 2.0 规范。
- (2) 在 `conf/web.xml` 中使用和将默认请求及响应字符编码设置为 UTF-8。
- (3) 删除 HTTP/2 UpgradeProtocol 元素上的 HTTP/1.1 配置重复项。

2. Tomcat 9 版本

Tomcat 9 建立在 Tomcat 8 版本的基础之上，符合 Servlet 4.0 规范，执行 JSP 2.4、EL 3.1、Web Socket 的 1.2 和 JASPIC 1.1 规格，包括以下功能改进：

- (1) 添加对 HTTP/2 的支持（需要 APR/本地库）。
- (2) 添加对 TLS 虚拟主机的支持。
- (3) 添加对使用 JSSE 连接器（NIO 和 NIO2）和使用 OpenSSL for TLS 支持的支持。

3. Tomcat 8 版本

Tomcat 8 建立在 Tomcat 7 版本的基础之上，是符合 Servlet 3.1、JSP 2.3、EL 3.0 和 Web Socket 的 1.1 规格的版本。除此之外，它在用单个公共资源实现来替换早期版本中提供的多个资源扩展特性方面做了重大改进。

4. Tomcat 7 版本

Tomcat 7 是 Tomcat 6 的改进版本，符合 Servlet 3.0、JSP 2.2、EL 2.2 和 Web Socket 的 1.1 规格。除此之外，它还包括以下改进：

- (1) Web 应用程序内存泄漏检测和预防。
- (2) 提高了 Manager 和 Host Manager 应用程序的安全性。
- (3) 通用 CSRF 保护。
- (4) 支持直接在 Web 应用程序中包含外部内容。
- (5) 重构（连接器，生命周期）和大量的内部代码清理。

5. Tomcat 6 版本

Tomcat 6 是 Tomcat 5.5 的改进版本，符合 Servlet 2.5 和 JSP 2.1 规范。除此之外，它还包括以下改进：

- (1) 内存使用优化。
- (2) 高级 IO 功能。
- (3) 重构聚类。

Tomcat 是一个开源的 Java Servlet 的软件实现和 Java Server Pages 技术的服务器。不同版本的 Tomcat 可用于不同版本的 Servlet 和 JSP 规范。它们之间的映射规范和相应的 Tomcat 版本如表 2-1 所示。

表 2-1 Tomcat 版本映射表

Servlet 支持	JSP 支持	EL 支持	Web Socket 规范	Tomcat 版本	支持 Java 版本
5.0	3.0	4.0	2.0	10.0.x	8 以后
4.0	2.3	3.0	1.1	9.0.x	8 以后
3.1	2.3	3.0	1.1	8.5.x	7 以后
3.1	2.3	3.0	1.1	8.0.x	7 以后
3.0	2.2	2.2	1.1	7.0.x	6 以后
2.5	2.1	2.1	N/A	6.0.x	5 以后
2.4	2.0	N/A	N/A	5.5.x	1.4 以后
2.3	1.2	N/A	N/A	4.1.x	1.3 以后
2.2	1.1	N/A	N/A	3.3.x	1.1 以后

每个版本的 Tomcat 支持任何稳定的 Java 版本，在选择版本时只要满足上面表格中最后一栏的要求即可。本书以 Tomcat 10 版本为例进行 Tomcat 服务器的搭建。

2.2.2 安装 Tomcat 解压版

在完成 JDK 的安装和环境配置后就可以安装 Tomcat。下面介绍 Tomcat 解压版的安装与环境配置，具体操作步骤如下。

步骤 1：打开浏览器，在地址栏中输入网址“http://tomcat.apache.org”进入 Tomcat 官网，Tomcat 官网界面如图 2-6 所示。

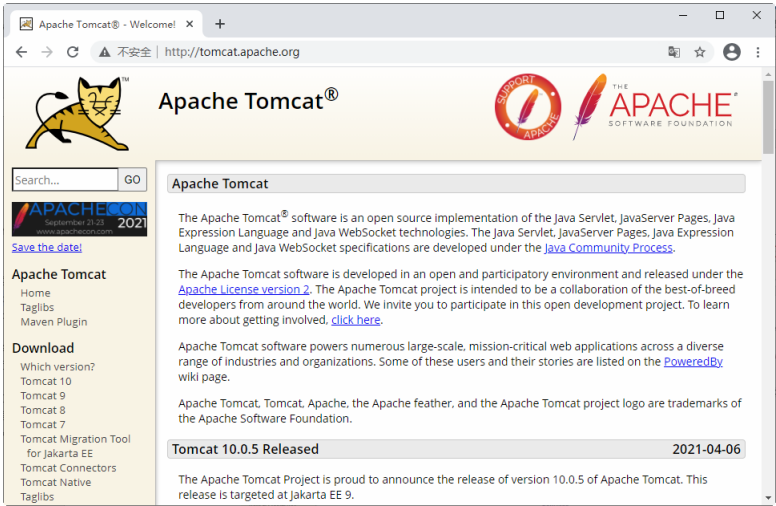


图 2-6 Tomcat 官网界面

- 步骤 2：在 Tomcat 官网界面中找到软件下载（Download）区域，如图 2-7 所示。
- 步骤 3：在下载区域中选择 Tomcat 10 版本选项，在 Tomcat 10 下载界面的快速导航（Quick Navigation）栏中单击 10.0.5 选项，如图 2-8 所示。
- 步骤 4：进入选择压缩包的下载界面中，根据自己计算机的 CPU 支持的是 32 位或 64 位以及计算机的配置，选择 32-bit Windows zip 或 64-bit Windows zip 压缩版进行下载，如图 2-9 所示。

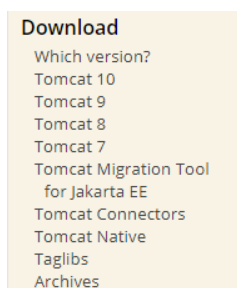


图 2-7 Download 区域

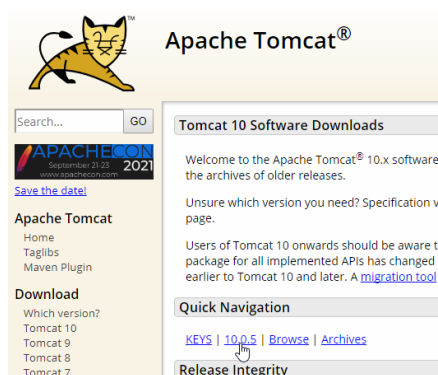


图 2-8 Tomcat 10 下载快速导航界面

步骤 5：下载完成后，选择并将该压缩文件解压到英文路径的盘符下，如图 2-10 所示为放到“C:\Tomcat”下，注意只要是纯英文路径下都可以，至此解压版的安装便完成。

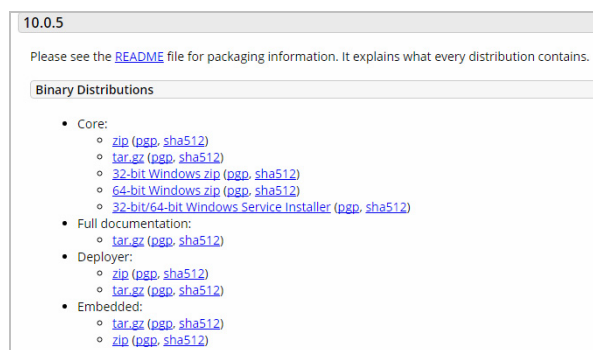


图 2-9 选择适合自己计算机的解压版下载

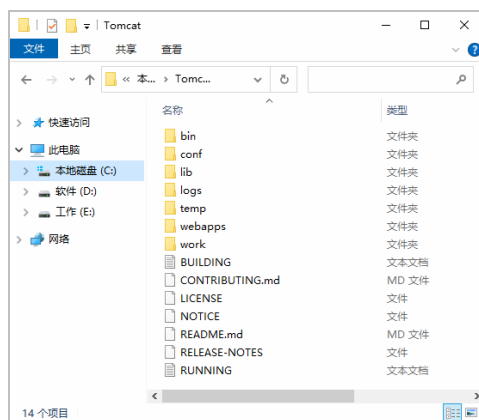


图 2-10 Tomcat 路径

2.2.3 安装 Tomcat 安装版

下面详细介绍 Tomcat 安装版的安装，具体操作步骤如下。

步骤 1：进入 Tomcat 10 下载快速导航界面，选择 32-bit/64-bit Windows Service Installer 下载选项，如图 2-11 所示。

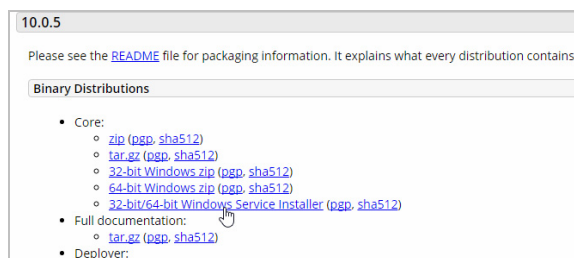


图 2-11 选择安装版下载

步骤 2：下载完成后，解压并打开软件压缩包，然后双击软件安装包中的 Setup.exe 文件，打开 Apache Tomcat Setup 界面，如图 2-12 所示。

步骤3：单击 Next 按钮，进入 License Agreement 界面，如图 2-13 所示。

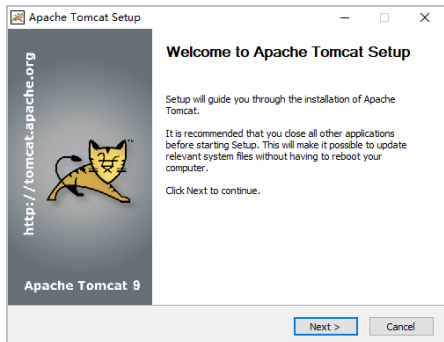


图 2-12 安装界面

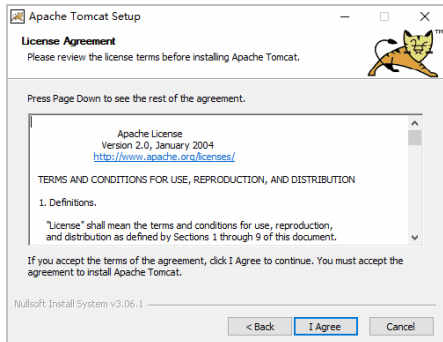


图 2-13 License Agreement 界面

步骤4：单击 I Agree 按钮，进入 Choose Components 界面，在其中选择需要安装的选项，如图 2-14 所示。

步骤5：单击 Next 按钮，进入 Configuration 界面，此时安装程序会提示设置端口和用户信息，也可采用默认设置，如图 2-15 所示。

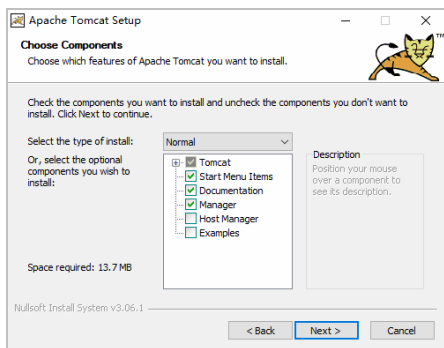


图 2-14 Choose Components 界面

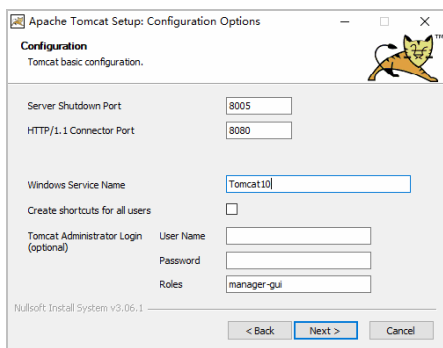


图 2-15 设置端口及用户信息界面

步骤6：单击 Next 按钮，进入 Java Virtual Machine 界面，在其中可以设置 Java 软件的安装路径，如图 2-16 所示。

步骤7：单击 Next 按钮，进入 Choose Install Location 界面，然后单击 Browse 按钮为程序指定安装路径，如图 2-17 所示。

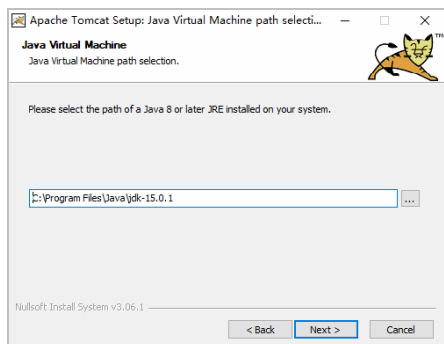


图 2-16 设置 Java 的安装路径

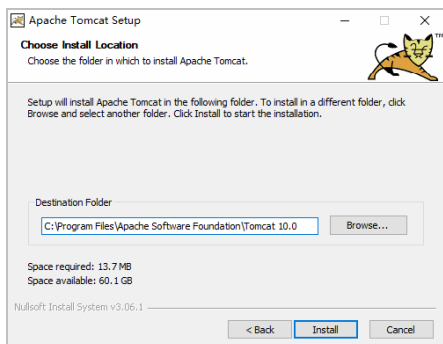


图 2-17 选择 Tomcat 的安装路径

步骤 8：完成路径的设置后，单击 Install 按钮即可安装软件，并显示安装进度，如图 2-18 所示。

步骤 9：安装完成后，弹出 Completing Apache Tomcat Setup 对话框，提示用户安装完成，如图 2-19 所示。

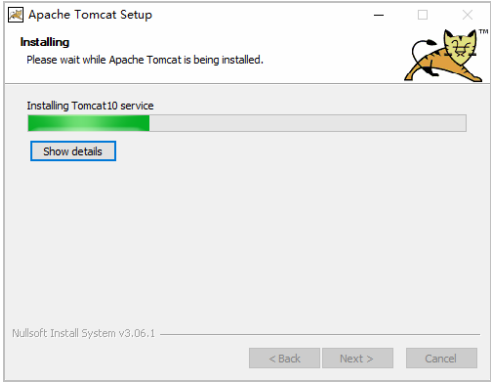


图 2-18 软件安装进度

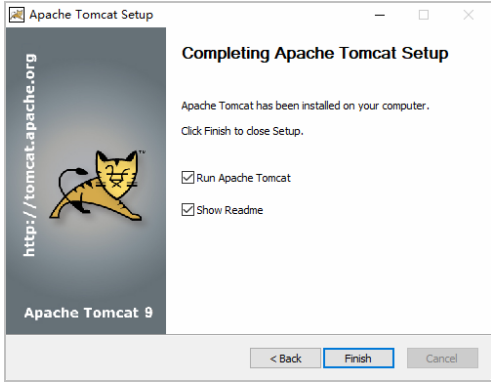


图 2-19 完成软件的安装

2.2.4 环境变量的配置

在完成 Tomcat 服务器的安装后，还需要进行环境变量的配置，才能使用 Tomcat Web 服务器，具体操作步骤如下。

步骤 1：选择计算机桌面上的“此电脑”图标并右击，在弹出的快捷菜单中选择“属性”菜单命令，打开“系统属性”对话框，单击“高级”选项卡中的“环境变量”按钮，打开“环境变量”对话框，如图 2-20 所示。

步骤 2：在“环境变量”对话框中单击“新建”按钮，弹出“新建系统变量”对话框，在“变量名”文本框中填写 CATALINA_HOME，在“变量值”文本框中填写前面所解压文件存放的路径，这里输入“C:\Tomcat”，该目录下有 lib、bin 等文件夹，如图 2-21 所示。

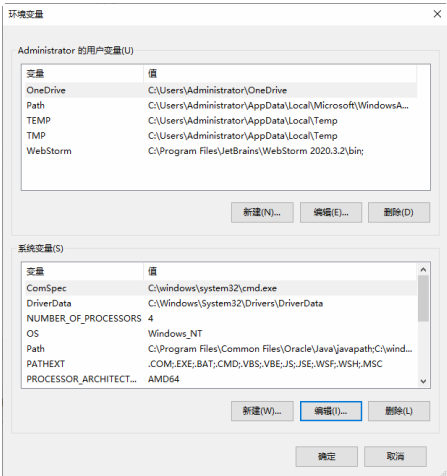


图 2-20 “环境变量”对话框

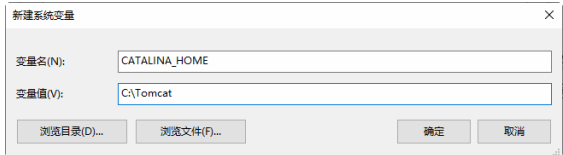


图 2-21 添加 CATALINA_HOME 变量

步骤 3：完成变量名和变量值的设置后，单击“确定”按钮，完成 Tomcat 的安装操作。



微视频

2.3 Tomcat 的启动与关闭

在安装好 Tomcat 之后还需要学会如何启动与关闭。下面介绍 Tomcat 的启动与关闭。

2.3.1 在服务器中启动与关闭

在服务器中启动与关闭 Tomcat 解压版的方法如下。

步骤 1: 在 Tomcat 安装完成后, 打开 Tomcat 安装路径下的 bin 文件夹, 找到 startup.bat 文件双击运行, 即可启动 Tomcat 服务器, 如图 2-22 所示。

步骤 2: 打开任意浏览器, 并在浏览器的地址栏中输入 “http://localhost:8080/” (8080 是 Tomcat 默认端口号), 若出现如图 2-23 所示的界面, 则说明 Tomcat 服务器启动成功。

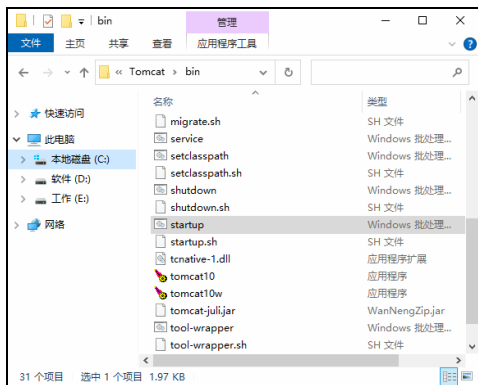


图 2-22 运行 startup.bat 文件

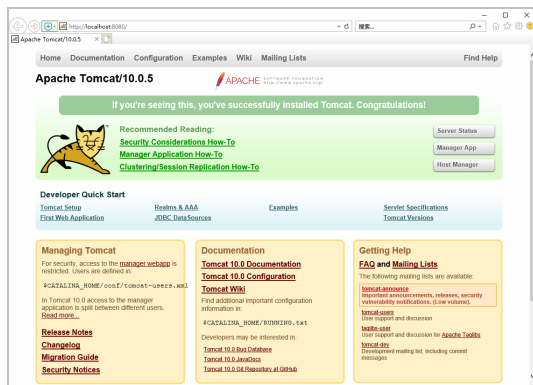


图 2-23 服务器启动成功

步骤 3: 关闭 Tomcat 服务器。双击运行安装路径下 bin 目录中的 shutdown.bat 文件, 即可关闭 Tomcat 服务器, 如图 2-24 所示。

在服务器中启动与关闭 Tomcat 安装版的方法如下。

步骤 1: 单击 “开始” 按钮, 选择 Apache Tomcat 10→Monitor Tomcat 菜单命令, 打开 Apache Tomcat 10.0 Tomcat10 Properties 对话框, 如图 2-25 所示。

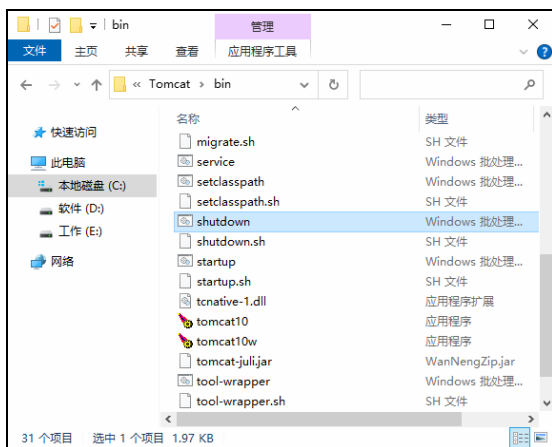


图 2-24 关闭服务器

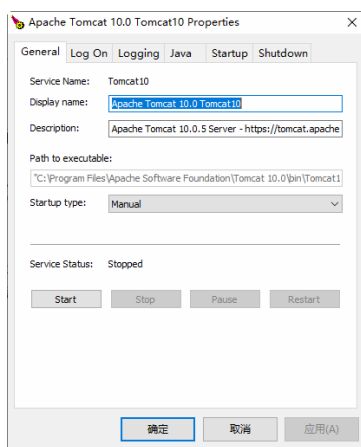


图 2-25 Tomcat 10 对话框

步骤 2：单击 **Start** 按钮启动 Tomcat 服务器，如图 2-26 所示。

步骤 3：启动完成后，在 Apache Tomcat 10.0 Tomcat10 Properties 对话框中可以看到服务器的状态为“**Started**”，如图 2-27 所示。

步骤 4：在浏览器中输入测试地址即可显示 Tomcat 服务器主页，测试地址是“**http://127.0.0.1:8089**”，其中 8089 是在前面设置的 HTTP 端口号，如图 2-28 所示。

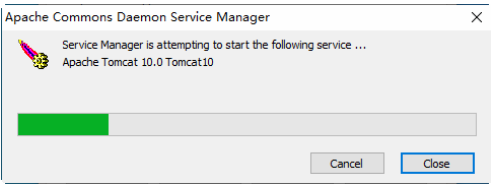


图 2-26 启动 Tomcat 服务器

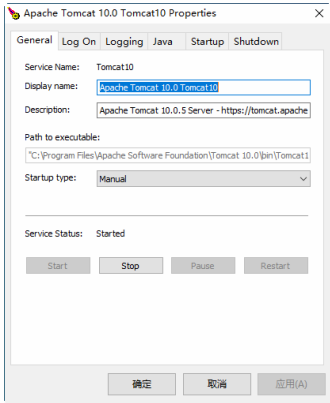


图 2-27 查看服务器的状态

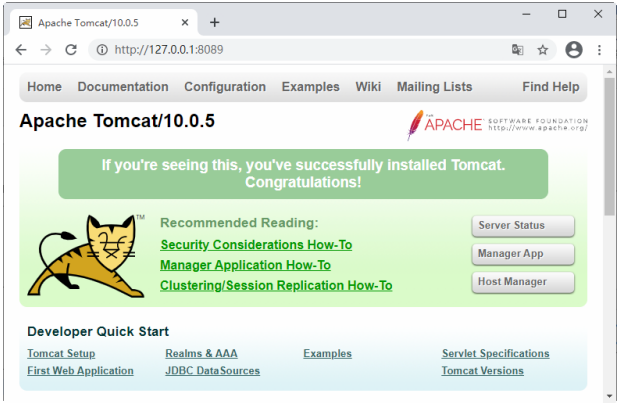


图 2-28 Tomcat 服务器主页

2.3.2 在 Eclipse IDE 中启动与关闭

所谓 IDE 就是像 Eclipse 这样的编译器，在此以 Eclipse 为例介绍 Tomcat 的启动与关闭。

步骤 1：在 Eclipse 工作界面中选择 **Window→Show View→Other** 菜单命令，如图 2-29 所示。

步骤 2：打开 Show View 对话框，在其中展开 **Server** 文件夹并选择 **Servers** 选项，如图 2-30 所示。

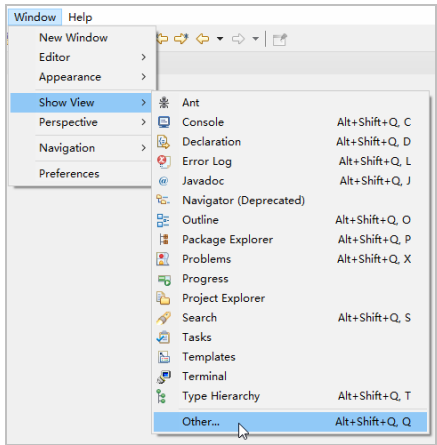


图 2-29 选择 Other 菜单命令

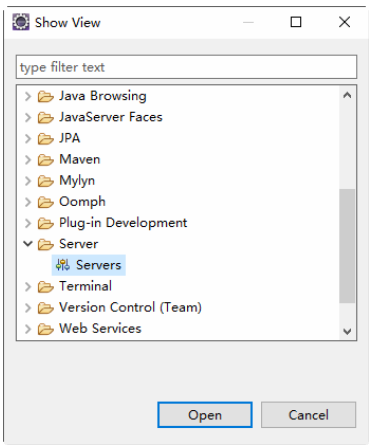


图 2-30 Show View 对话框

步骤 3：单击 **Open** 按钮，即可在 Eclipse 主界面下方显示 **Servers** 选项，如图 2-31 所示。

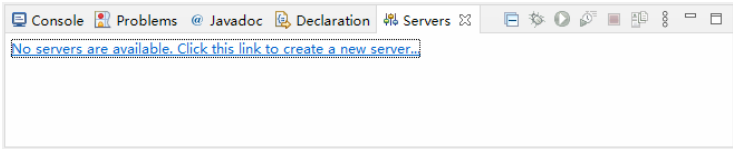


图 2-31 Servers 选项

步骤 4：单击 **Servers** 选项下面的超链接，在弹出的 **New Server** 对话框中找到 **Apache** 文件夹并打开，选择安装的 Tomcat 版本，如图 2-32 所示。

步骤 5：单击 **Next** 按钮，打开 **Tomcat Server** 对话框，如图 2-33 所示。

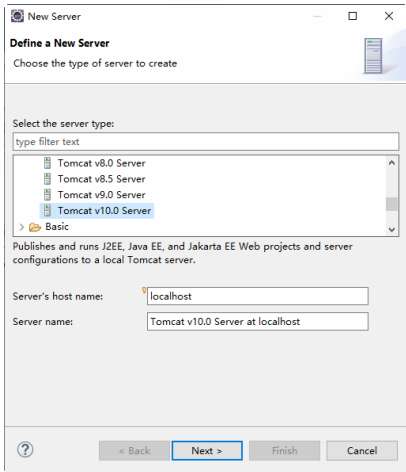


图 2-32 选择 Tomcat 版本

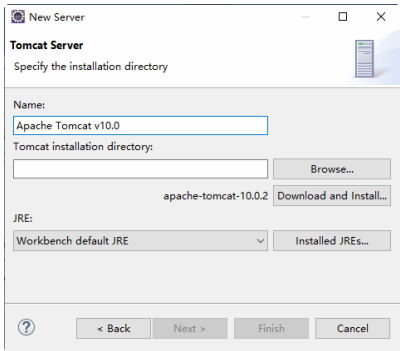


图 2-33 Tomcat Server 对话框

步骤 6：单击 **Browse** 按钮，打开“选择文件夹”对话框，在其中选择 Tomcat 的安装路径，如图 2-34 所示。

步骤 7：单击“选择文件夹”按钮，返回到 **New Server** 对话框中，可以看到添加的 Tomcat 安装路径，如图 2-35 所示。

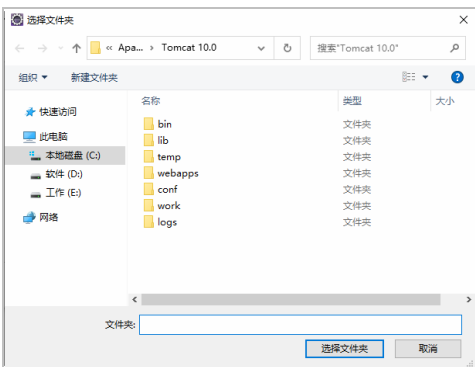


图 2-34 选择 Tomcat 的安装路径

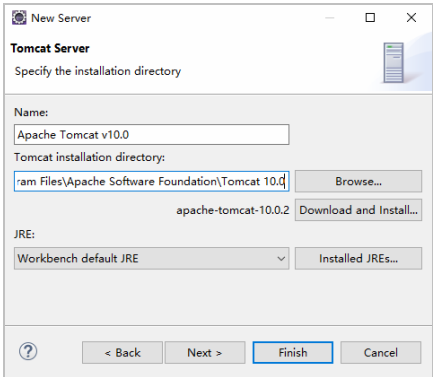


图 2-35 New Server 对话框

步骤 8：单击 **Finish** 按钮，返回到 Eclipse 工作界面，在 **Servers** 选项下即可看到添加的服务，如图 2-36 所示。

步骤 9：服务添加成功后，选中添加的服务，然后单击服务窗口右上角的启动或停止服务按钮，即可启动或停止 Tomcat 服务，其中第 3 个按钮为启动按钮，第 5 个按钮为停止按钮，如图 2-37 所示。

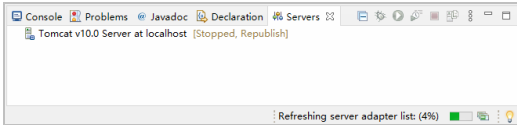


图 2-36 添加的服务

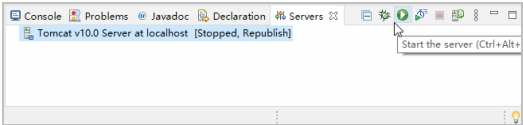


图 2-37 激活启动或停止按钮

步骤 10：单击启动按钮，Tomcat 服务启动成功后会在 Console 窗口中显示启动信息，如图 2-38 所示。这时，在 Eclipse 左侧的项目列表中便可以看到 Servers 的服务信息，如图 2-39 所示。

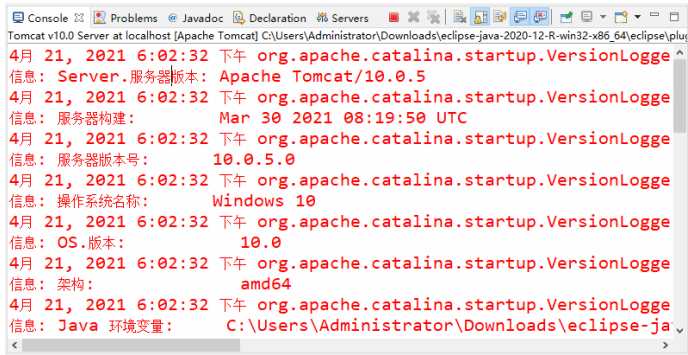


图 2-38 显示启动信息

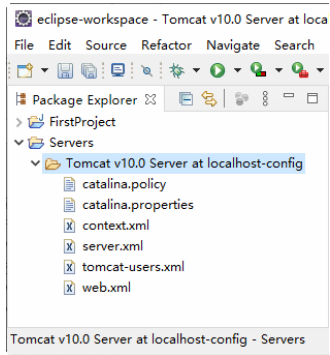


图 2-39 Servers 的服务信息

☆ 大牛提醒 ☆

若启动不成功，大多数情况下都是在外部已经启动了，这时 Eclipse 会报错，如图 2-40 所示。如果出现这种情况，可以打开 Tomcat 安装目录下的 bin 文件夹，找到 shutdown.bat 文件，双击关闭 Tomcat 服务，或者在 Apache Tomcat 10.0 Tomcat10 Properties 对话框中单击 Stop 按钮来关闭 Tomcat 服务，然后再到 Eclipse 中启动，如图 2-41 所示。

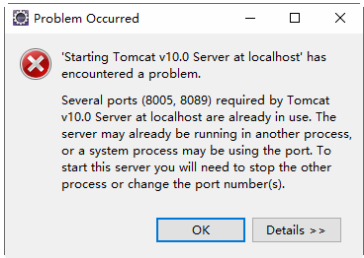


图 2-40 Tomcat 启动失败

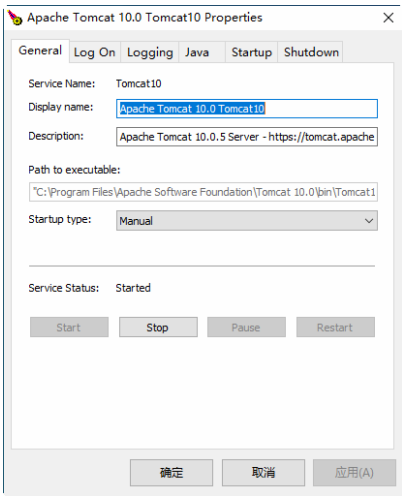


图 2-41 关闭 Tomcat 服务



微视频

2.4 修改 Tomcat 端口号

在默认情况下, Tomcat 的端口号是 8080, 但如果使用了两个 Tomcat, 就需要修改其中一个 Tomcat 的端口号才能使得两个 Tomcat 同时正常工作。那么, 如何修改 Tomcat 的端口号呢? 下面分别介绍在服务器和 IDE 中修改端口号的方法。

2.4.1 在服务器中修改端口号

在服务器中修改端口号的具体方法如下。

步骤 1: 在 Tomcat 安装目录 (或者解压目录) 下找到并打开 conf 文件夹, 在其中找到 server.xml 文件。

步骤 2: 用文件编辑工具或者记事本打开 server.xml 文件, 并找到如下代码段。

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
```

步骤 3: 将 port="8080" 的端口号值改为其他数值, 便完成了修改端口号的操作。

2.4.2 在 Eclipse IDE 中修改端口号

在 Eclipse IDE 中也可以修改端口号, 具体方法如下。

步骤 1: 在 Eclipse IDE 集成开发环境中双击 Servers 选项下的 Tomcat 本地服务器, 如图 2-42 所示。

步骤 2: 在窗口中找到 Ports 选项, 在 HTTP/1.1 对应栏中输入想修改的端口号值 (默认为 8080), 如图 2-43 所示。

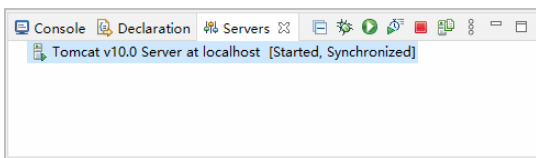


图 2-42 Servers 下的服务器

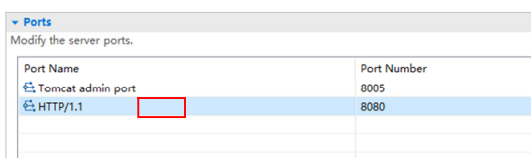


图 2-43 在 Eclipse 中修改端口号

步骤 3: 完成端口号的修改后, 选择 File→Save 菜单命令保存即可, 如图 2-44 所示。

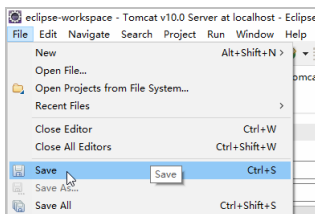


图 2-44 保存修改的端口号

2.5 将 Web 项目部署到 Tomcat 中



微视频

已完成的项目需要部署到 Tomcat 中才能被浏览器正常浏览和访问, 本节介绍在 Tomcat 和

Eclipse 中部署 Web 项目的方法。

2.5.1 在服务器中部署 Web 项目

在 Tomcat 服务器配置完成后，还需要将 Web 项目部署到服务器中，然后通过 ip+端口号+项目名来访问 Web 项目。将 Web 项目部署到 Tomcat 中的方法就是直接把项目复制到 Tomcat 的 webapps 文件夹下，具体步骤如下。

步骤 1：新建文件夹 myweb，然后新建文本文件并改名为 web.jsp。web.jsp 的代码如下：

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h2>这是我的个人主页</h2>
</body>
</html>
```

把 myweb 项目放到 Tomcat 的 webapps 文件夹下，如图 2-45 所示。

步骤 2：打开 Tomcat 服务器（确保服务器打开），在地址栏中输入“http://localhost:8089/myweb/web.jsp”就可以在浏览器中访问 Web 项目，这样就完成在 Tomcat 服务器中部署项目的操作，如图 2-46 所示。

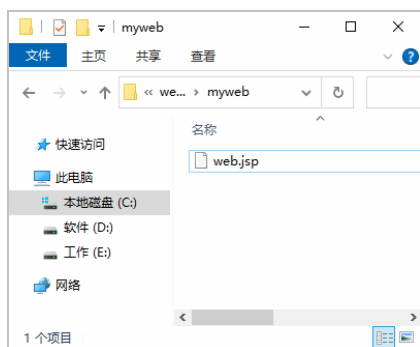


图 2-45 在服务器中部署 Web 项目



图 2-46 在浏览器中访问 Web 项目

2.5.2 在 Eclipse IDE 中部署 Web 项目

通过 Eclipse IDE 也可以将 Web 项目部署到 Tomcat 服务器中。

1. 建立新项目

步骤 1：在 Eclipse 工具页面左侧 Project Explorer 窗口的空白处右击，在弹出的快捷菜单中选择 New→Other 菜单命令，如图 2-47 所示。

步骤 2：弹出 Select a wizard 对话框，展开 Web 文件夹，然后选择 Dynamic Web Project 子节点，如图 2-48 所示。

步骤 3：单击 Next 按钮，打开 Dynamic Web Project 对话框，在 Project name 文本框中输入“myWeb”，如图 2-49 所示。

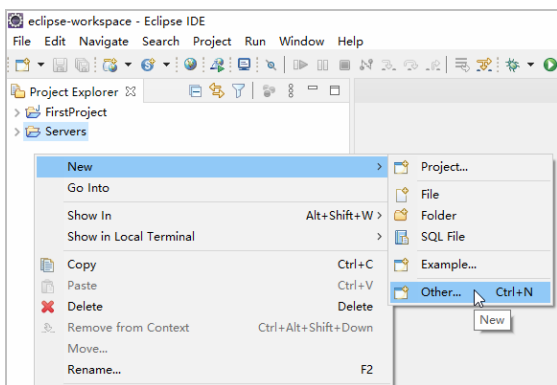


图 2-47 选择 Other 菜单命令

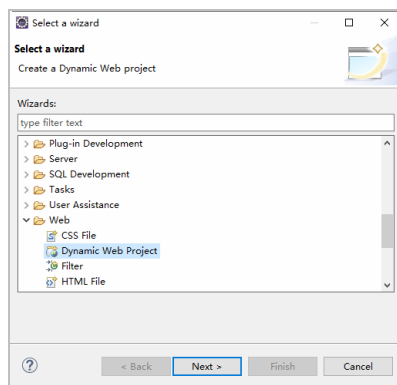


图 2-48 Select a wizard 对话框

步骤 4: 单击 Next 按钮, 弹出 Java 对话框, 在该对话框中选择默认设置, 如图 2-50 所示。

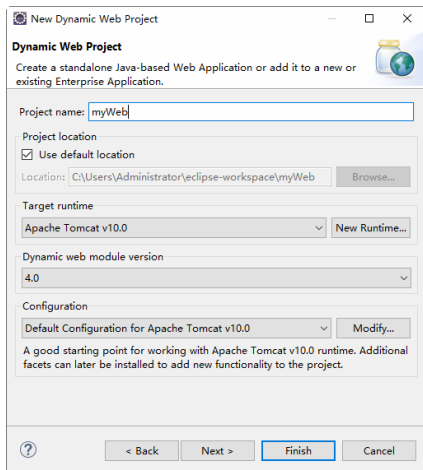


图 2-49 输入“myWeb”

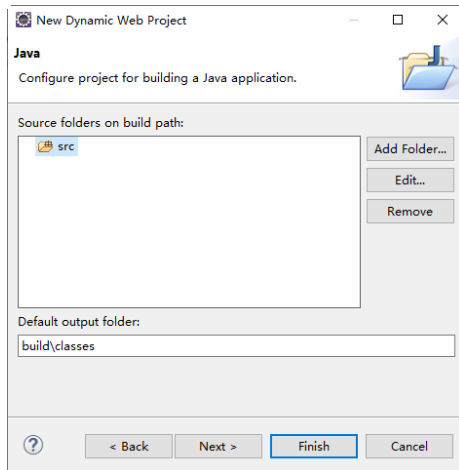


图 2-50 Java 对话框

步骤 5: 单击 Next 按钮, 打开 Web Module 对话框, 在 Context root 文本框中输入“myWeb”, 在 Content directory 文本框中输入“WebContent”, 如图 2-51 所示。

步骤 6: 单击 Finish 按钮, 返回到 Eclipse 工具页面, 这样就创建了一个 Web 项目工程, 展开 WebContent 文件夹, 可以看到整体项目结构, 如图 2-52 所示。

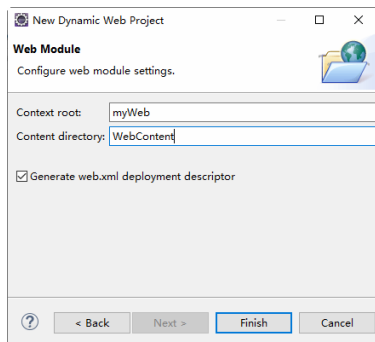


图 2-51 输入“WebContent”

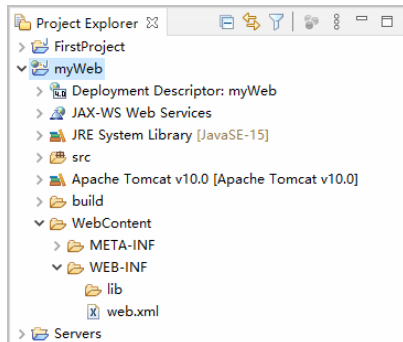


图 2-52 展开 WebContent 文件夹

2. 创建 JSP 页面

步骤 1：在 Project Explorer 窗口中选择 WebContent 文件夹，然后右击，在弹出的快捷菜单中选择 New→Other 菜单命令，如图 2-53 所示。

步骤 2：打开 Select a wizard 对话框，在 Web 文件夹中选择“JSP File”，如图 2-54 所示。

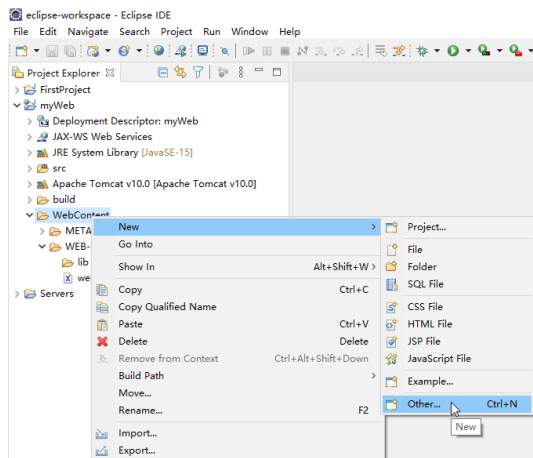


图 2-53 选择 Other 菜单命令

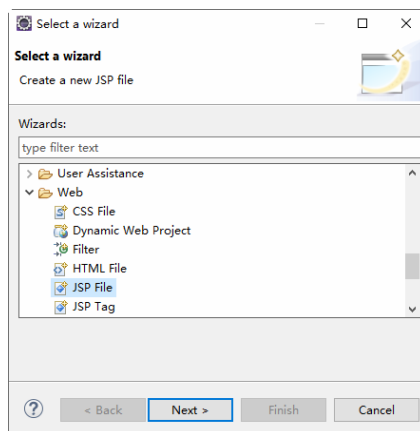


图 2-54 选择“JSP File”

步骤 3：单击 Next 按钮，打开 JSP 对话框，在 File name 文本框中输入“web.jsp”，如图 2-55 所示。

步骤 4：单击 Finish 按钮，即可在项目目录中看到新建立的 JSP 页面，如图 2-56 所示。

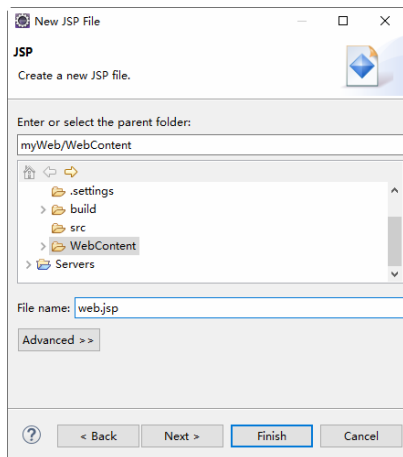


图 2-55 输入“web.jsp”

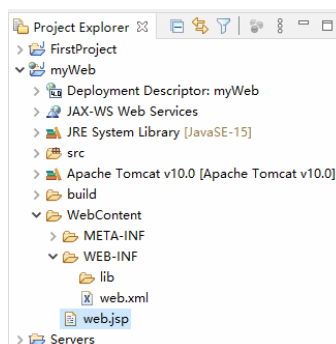


图 2-56 新建 JSP 页面

3. 编辑 JSP 页面

双击 web.jsp 页面，在打开的界面中输入以下代码，如图 2-57 所示。

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
```

```

<body>
<h2>这是我的个人主页</h2>
</body>
</html>

```



图 2-58 Server Locations 区域

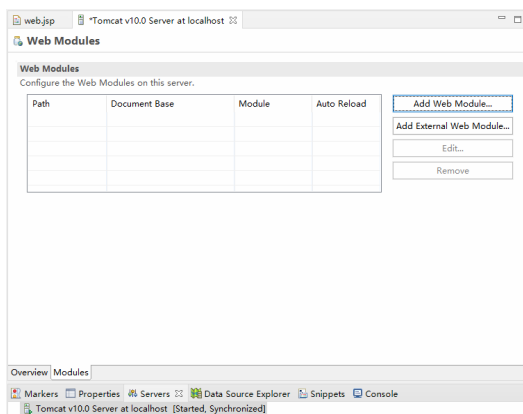


图 2-59 Modules 选项卡

步骤 3: 单击 Add Web Modules 按钮, 打开 Add Web Modules 对话框, 在其中选择刚添加的 myWeb 项目, 如图 2-60 所示。

步骤 4: 单击 OK 按钮, 返回到 Web Modules 界面, 在其中可以看到添加的 myWeb 项目, 然后按下 Ctrl+S 键, 即可保存项目, 如图 2-61 所示。



图 2-60 选择 myWeb 项目

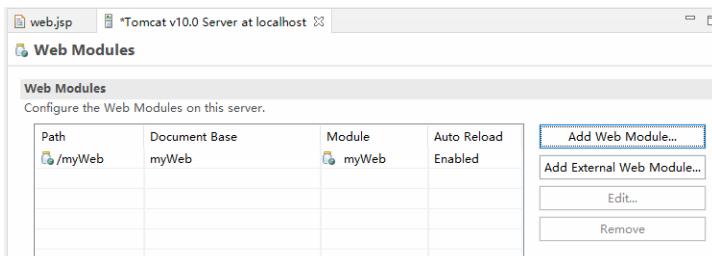


图 2-61 Web Modules 界面

步骤 5：保存成功后关闭页面，可以在 Servers 服务信息下看到已经成功添加的 myWeb 项目，如图 2-62 所示。

步骤 6：在 Servers 选项下选中 Tomcat 服务并右击，在弹出的快捷菜单中选择 Start 菜单命令，如图 2-63 所示。

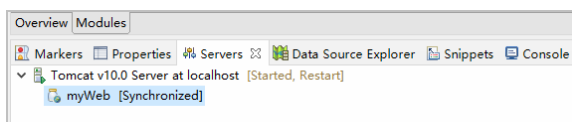


图 2-62 成功添加 myWeb 项目

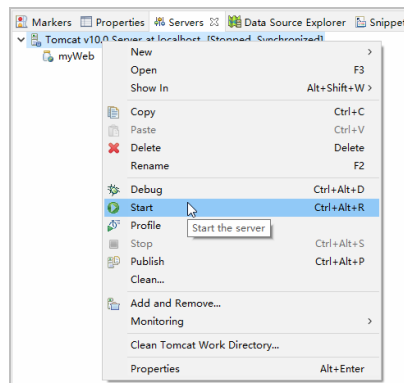


图 2-63 选择 Start 菜单命令

☆ 大牛提醒 ☆

用户还可以单击 Servers 选项右上角的启动按钮来启动 Tomcat 服务，如图 2-64 所示。

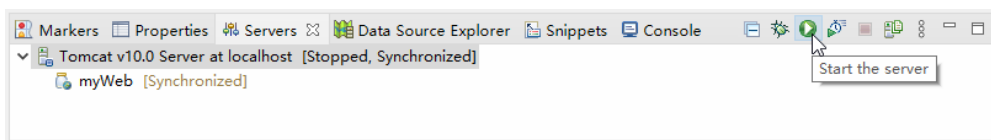


图 2-64 启动 Tomcat 服务

步骤 7：在 Eclipse 浏览器的地址栏中输入“http://localhost:8089/myWeb/web.jsp”，如果浏览器中打开如图 2-65 所示的界面，说明 Web 项目部署成功。



图 2-65 运行界面

2.6 新手疑难问题解答

问题 1：Tomcat 中使用的连接器是什么？

解答：在 Tomcat 中使用了两种类型的连接器，一种是 HTTP 连接器，另一种是 AJP 连接器。HTTP 连接器有许多可以更改的属性，以确定它的工作方式和访问功能，例如重定向和代理转发；AJP 连接器以与 HTTP 连接器相同的方式工作，使用的是 HTTP 的 AJP 协议。AJP 连接器通常通过插件技术 mod_jk 在 Tomcat 中实现。

问题 2：如何使用 WAR 文件部署 Web 应用程序？

解答：在 Tomcat 的 Web 应用程序目录下，JSP、Servlet 和它们的支持文件被放置在适当的

子目录中。用户可以将 Web 应用程序目录下的所有文件压缩到一个压缩文件中，文件的扩展名为.war。用户可以通过在 webapps 目录中放置 WAR 文件来执行 Web 应用程序。当一个 Web 服务器开始执行时，它会将 WAR 文件的内容提取到适当的 webapps 子目录中。

2.7 实战训练

实战 1：获取本机的 IP 地址。

编写程序，通过使用脚本小程序来获取本机的 IP 地址。脚本程序可以包含任意数量的 Java 语句、变量、方法或表达式，只要它们在脚本语言中是有效的。程序的运行结果如图 2-66 所示。

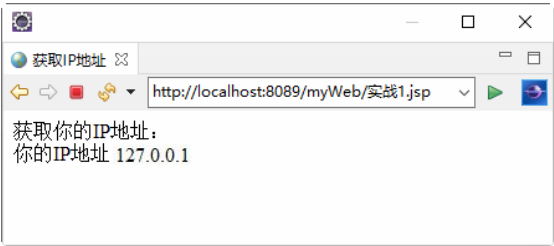


图 2-66 获取 IP 地址

实战 2：获取用户的注册信息。

编写程序，获取用户的注册信息。这里综合应用 JSP 的有关内置对象设计一个系统登录功能页面。如图 2-67 所示为“用户注册”页面，单击“确定”按钮后即可跳转至如图 2-68 所示的“注册信息”页面，并在其中显示用户所输入的信息。

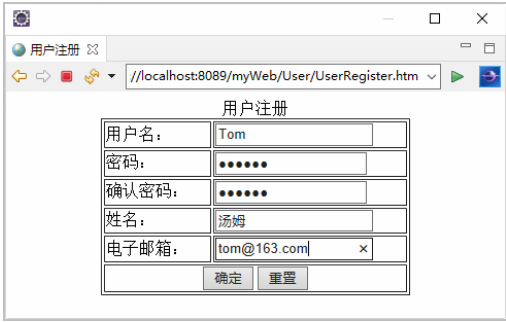


图 2-67 “用户注册”页面

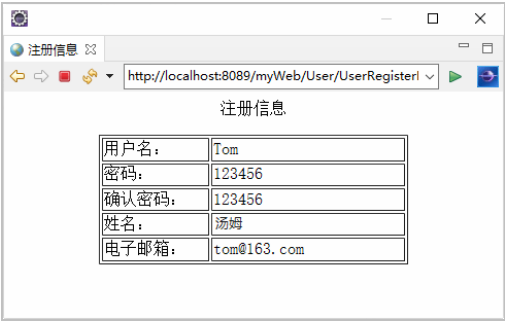


图 2-68 “注册信息”页面