JSP+Servlet+Tomcat

应用开发从零开始学

刘华贞 编著

（第3版）

清華大學出版社
北京

内 容 简 介

本书全面系统地介绍JSP+Servlet+Tomcat开发中涉及的相关技术要点和实战技巧。本书内容讲解循序渐进，结合丰富的示例使零基础的读者能够熟练掌握JSP+Servlet+Tomcat的应用开发和部署。本书配套示例代码、PPT课件、作者答疑服务。

本书共17章。第1~7章为Java Web基础开发，内容包括搭建Java Web开发环境、JSP基础语法、JSP内置对象、Servlet技术、请求与响应、会话管理、Servlet进阶API、过滤器、监听器等；第8~15章为Java Web高级开发，内容包括MySQL 8数据库开发、JSP与Java Bean、EL标签、JSTL标签库、自定义标签、JDBC详解、XML概述、资源国际化等；第16~17章为Java Web实战，分别讲解两个典型的系统，即家校通门户网站（JSP+HTML+CSS）和在线购物系统（JSP+Java Bean+MySQL）。

本书内容精练、结构清晰、注重实战，适合广大Java Web开发初学者学习，还可作为高等院校或高职高专计算机及相关专业的教材使用。

图书在版编目（CIP）数据

JSP+Servlet+Tomcat 应用开发从零开始学/刘华贞编著. —3 版. —北京：清华大学出版社，2023.5

ISBN 978-7-302-63617-5

Ⅰ. ①J… Ⅱ. ①刘… Ⅲ. ①JAVA 语言－程序设计 Ⅳ. ①TP312.8

中国国家版本馆 CIP 数据核字（2023）第 094055 号

责任编辑：夏毓彦
封面设计：王　翔
责任校对：闫秀华
责任印制：沈　露

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：010-83470000　　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
印 装 者：三河市天利华印刷装订有限公司
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张：**23.75　　**字　　数：**641 千字
版　　次：2015 年 1 月第 1 版　　2023 年 7 月第 3 版　　**印　　次：**2023 年 7 月第 1 次印刷
定　　价：99.00 元

产品编号：102060-01

前　言

本书是面向Java Web开发初学者的一本高质量图书。Java是当今程序开发中最流行的编程语言之一，不但可以开发手机应用、桌面应用，而且越来越多地应用于Java Web开发中。Java优越的跨平台特性使它备受欢迎。近年来，Java Web框架技术层出不穷，跨浏览器、跨系统等要求更是体现了Java Web开发的强大生命力。

目前，市面上有关Java Web的书籍非常多，初学者常常不知道应该如何选择。本书从初学者的角度出发，用浅显的实例说明复杂的知识点，为那些想在Java Web开发中大展拳脚的开发人员精心编写，虽然所讲内容未涉及当前大型项目的主流框架，但都属于Java Web中的基础知识。只有夯实基础知识，才能更好地学习其他技术框架。本书从底层原理入手，并从实战的角度进行讲解，以便让想要学习Java Web开发的初学者快速掌握相关技术，并能够根据实际需求开发出有用的Web应用。

本书特点

（1）内容丰富，知识全面。本书内容几乎涵盖Java Web基础开发的各个方面。本书还涉及Servlet 5.0版本的知识与编写规范，并利用详细的实例进行说明。

（2）循序渐进，由浅入深。为方便读者学习，本书首先介绍如何搭建Java Web开发的基础环境，然后介绍JSP的基础语法与Servlet的基本概念。帮助读者掌握这些基础知识后，让读者逐渐学习请求与响应的过程、会话管理、Servlet 5.0以上的版本中进阶API以及过滤器、监听器、自定义标签的编写等，从而更深入地掌握Java Web开发技术。

（3）编码规范，讲解详细。书中每个知识点都给出了详尽的操作示例，以供读者参考，并对代码进行详细解释。实例中的代码是严格按照Java规范进行编写的，并配有详细的代码注释。

（4）易学易用，案例丰富。本书通过简单的实例讲解每个知识点，力求用简单的实例来诠释复杂的知识，使读者快速了解并掌握Web开发所需的知识。对于各种语法几乎都配有一个实例来说明其用法。

（5）案例精讲，图文并茂。对于难以理解的知识点，编者用图表的方式进行讲解，让读者更加直观地理解知识点。编者根据多年的项目经验，在每章中尽量用一个综合示例对知识点进行整合，使读者对每章的知识点有整体掌握。

进阶路线

第1~7章，Java Web基础开发：讲解Java Web开发环境的搭建、JSP基础语法、JSP内置对象、Servlet技术、请求与响应、会话管理、Servlet进阶API、过滤器、监听器等基础知识。

第8~15章，Java Web高级开发：讲解MySQL数据库开发、JSP与Java Bean、EL标签、JSTL标签库、自定义标签、JDBC详解、XML概述、资源国际化等Java Web高级开发所需的知识。

第16~17章，Java Web实战：讲解如何运用Java Bean、MySQL、JSP技术以及标签开发家校通门户网站和带数据库的在线购物系统，使读者能够快速掌握Java Web开发技术和编写规范。

第 3 版修订说明

随着Java Web技术的快速发展，所使用的技术也在不断更新，为了方便读者学习最新技术，本书在第2版的基础上进行相应的升级。JDK的版本更新为17.0.4，Servlet升级到5.0并修改了相应的章节内容，JSP版本升级到3.1，Tomcat服务器由Tomcat 9改为Tomcat 10。本书的更新都是为了让读者跟上当前技术发展的步伐，希望读者不要停下学习的脚步，努力向前。

示例代码、PPT 课件下载

本书示例源代码、PPT课件的下载地址可扫描右侧的二维码获得。如果阅读过程中发现问题，请用电子邮件联系booksaga@163.com，邮件主题务必写“JSP+Servlet+Tomcat应用开发从零开始学（第3版）”。

本书适合的读者

- JSP+Servlet 开发初学者
- Java Web 开发初学者
- Java Web 开发工程师
- Web 应用开发人员
- 高等院校、中职学校、培训机构的学生

本书第1版本由林龙主笔，第2版本由刘华贞修订，第3版由刘华贞修订，在此表示感谢。

编　者

2023年3月

目　　录

第 1 章

搭建 Java Web 开发环境

正所谓“工欲善其事，必先利其器”，开发一个Web应用程序，首先必须搭建好开发环境，选择好开发工具，从而达到事半功倍的开发效果。现如今支持Web的应用服务器非常多，例如WebSphere、WebLogic、Tomcat等，它们的配置方法各不相同，本书选择Apache Tomcat 10.0.22作为服务器开发平台，JDK使用的是17.0.4版本。

本章主要涉及的知识点有：

- JSP支持的网络协议
- Web应用程序的运行环境和开发环境
- Tomcat软件的安装和配置
- JSP开发工具的选择

1.1 Web 开发背景知识

本节的重点是介绍Web开发的一些基础知识，首先简单介绍Web访问的基本原理，然后对超文本传输协议（Hyper Text Transfer Protocol，HTTP）进行简单介绍，最后介绍静态网页与动态网页的区别以及各种Web服务器的优缺点。通过本节的学习，读者可以掌握Java Web开发所需的背景知识。

1.1.1 Web访问的基本原理

Web访问可以简单划分为两个过程：客户端请求和服务器端响应。客户端的请求通过Servlet引擎传递给Servlet模块，Web服务器接收客户端的请求，并把处理的结果返回给客户端。客户端与服务器之间的通信协议就是超文本传输协议。客户端与服务器之间的请求模式如图1.1所示。

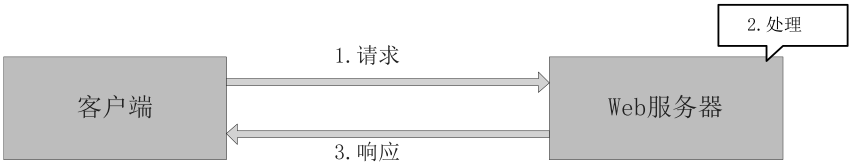

图1.1　Web访问原理

1.1.2　超文本传输协议

超文本传输协议是一种在互联网上应用最为广泛的网络协议，是一种无状态的协议。自1990年起，它就已经被应用于WWW全球信息服务系统。所有的WWW文件都必须遵守这个标准。

超文本传输协议的主要特点如下：

- 简单、快速：客户端向服务器请求服务时，只需发送请求方法和路径URL。通常请求的方法有POST、GET、PUT和DELETE。由于HTTP的协议简单，使得HTTP服务器的程序规模相对较小，因此传输速度较快。
- 灵活：HTTP允许传输任意类型的数据，例如普通文本、超文本、音频、视频等，主要由Content-Type控制。
- 无状态：无状态是指对于数据库事务处理没有记忆能力。后续的处理如果需要前面的信息，就需要重新发送。
- 无连接：无连接的含义是每次连接只处理一次请求，处理完当次请求后就断开连接。

1.1.3　静态网页和动态网页

在网站设计中，直接使用HTML标记语言编写的网页通常被称为“静态网页”。静态网页是标准的HTML文件，后缀名为.htm或.html。它所展示的内容一般是固定不变的，早期的网站一般都是静态网页。静态网页更新起来比较麻烦，需要将更新的HTML页面重新上传到网站服务器。显然，这样的网站缺乏灵活性，同时网站的维护成本也比较高。动态网页技术的出现改变了如此不灵活的状态，用户在不同时间或不同地点访问同一动态网页时显示的内容可以是不同的。

注意：所谓静态网页与动态网页，是基于用户访问网页时页面的内容有无变化而言的，与页面的视觉效果没有关系。因为动态的视觉效果大多是通过 JavaScript 或其他基于 JavaScript 的框架技术实现的，与动态网页技术没有必然的联系。

动态网页中的变化内容大部分来自数据库中数据的变化，通过增加、删除、修改、查找数据库中存储的数据来显示内容的变化。例如，在微博中发布一条微博后，查看微博时，会将所发的微博即时显示出来，这在静态网页中是无法完成的。动态网页在被访问时，首先运行服务器端脚本，通过它生成网页内容。显然，动态网页的显示内容是在访问该网页的时候动态生成的，而静态网页是提前做好放在服务器中的，因此，当前网络上的网页大多是动态网页，很少有静态网页，除非一些固定不变的内容，例如发布公告等新闻内容。

目前比较流行的动态网页技术主要包含 ASP、PHP 以及 JSP（Java Server Pages）。

- ASP更精确地说应该是一个中间件，它将Web上的请求转入IIS解释器中，IIS将全部解析执行ASP中的Script脚本。其缺点就是不能跨平台，只能在Windows平台下使用，开发受到诸多限制。其优点是微软提供了强大的IDE，所以开发者容易上手且开发效率也较高。
- PHP是当前比较流行的动态网页技术，是一种HTML内嵌式的语言，其语法融合了Java、C以及Perl，能够比CGI（Common Gateway Interface）更加快速地执行动态网页。其优点是开源、跨平台，正因为它具有开源和跨平台特性，所以很多网站都采用PHP编写自身的网页；其缺点是安装复杂，需要添加很多的外部库来支持，如图形需要gd库等。

注意：CGI 也是一种动态网页技术，虽然它的功能比较强大，但是它的由于比较编程困难、效率低下、修复复杂等缺陷，逐渐被新技术淘汰。

- JSP采用Java语言作为服务器端脚本，页面由HTML和嵌入Java代码组成。随着Java的广泛应用，JSP的应用也越来越广泛。其优点是简单易用，完全面向对象，具有Java的平台无关性、安全性和可靠性。目前大多数的动态网站开发都采用JSP技术，它具有很高的市场占有率。

1.1.4　Web浏览器和Web服务器

1．Web 浏览器

浏览器是指Web服务的客户端浏览程序。它可以向服务器发送各种请求，并对从服务器中返回的各种信息（包括文本、超文本、音频、视频等各种数据）进行解释、显示和播放。现如今，Web浏览器遍地开花，国外的有Internet Explorer（IE）、Chrome、Firefox、Safari，以及近几年逐渐步入公众视野的Microsoft Edge，国内的有360浏览器、QQ浏览器、傲游浏览器、搜狗浏览器、猎豹浏览器等，它们都各自占据着一片江山。由于IE和Edge绑定在Windows操作系统中，因此IE和Edge在市场中占有明显较高的份额，但随着互联网的不断进步，Chrome浏览器大量普及，已逐渐展露出超越IE的势头。

国内的浏览器从利用IE支持的内核逐步发展为支持多种内核，例如新版本的360浏览器同时支持Trident+Blink内核；而国外的浏览器内核是自主研发的，例如IE的内核是Trident、Chrome的内核由Webkit转为Blink、Firefox的内核是Gecko、Safari的内核是Webkit等。

2．Web 服务器

浏览器与服务器的关系可谓是“唇齿相依”，浏览器发送请求，服务器处理请求并将结果返回给浏览器显示。Web 服务器的种类繁多，目前比较流行的有 WebSphere、WebLogic、Tomcat等。它们的配置、启动方式各不相同，也有各自的优缺点。

- WebSphere服务器是IBM的产品，是一款功能很完善的Web服务器，基于Java程序研发，用于建立、部署和管理Web应用程序。WebSphere产品有开发版和商业版，但是WebSphere不开源。
- WebLogic服务器是一款多功能、基于标准的Web服务器。它遵从开放的标准、支持基于组件的开发。因为它性能比较稳定，所以国内有很多大公司在使用。

- Tomcat是一款开放源代码、基于Java的Web服务器，也是一款轻量型的Web服务器，是基于Apache许可证下开发的自由软件，根据JSP和Servlet技术标准实行。它安装简单、占有系统空间较小，却能支撑较大的Web应用系统，所以它是开发者、小型公司、学校网站建设者的首选软件。

1.2 JSP 简介

上一节简单介绍了Web开发的一些背景知识，读者已经了解了Web访问的基本原理、超文本传输协议、静态网页与动态网页的区别，以及主流的浏览器和Web服务器。本节将介绍JSP的基本概念、执行过程等，让读者了解JSP是什么、能做些什么。

1.2.1 什么是JSP

JSP技术是由SUN公司（现被Oracle收购）提出、多家公司参与的，于1999年推出的一款建设动态网页的方法。它基于Java Servlet技术来开发动态的、高性能的Web应用程序。JSP的网页实际上是由HTML文件中加入的Java代码片段和JSP的特殊标记构成的。

因为JSP是Java的成员，所以JSP具有平台无关性，即实现了跨平台功能，实现了用户界面和程序代码的解耦合，使得业务逻辑与代码的耦合度更低，开发人员可以在不更改JSP程序的情况下修改用户的界面。

JSP即Java服务器界面，实质上是一个Servlet程序。JSP页面不仅包含HTML代码，还包含用于产生动态网页内容的Java代码，这些Java代码可以是Java Bean、SQL语句、RMI（远程方法调用）对象等。例如，一个JSP页面包含了用于产生静态网页的HTML代码，同时也包含了连接数据库的JDBC代码，正因为如此才能称之为动态网页。

1.2.2 JSP的优势

JSP可以看作Java Servlet的一种扩展，JSP在使用前必须被编译为Servlet，也就是Java类，然后被调用执行，Servlet所产生的Web页面是不能包含在HTML标签中的，因为它离不开Java类文件的支持。第一次访问JSP页面时，Web服务器会将此页面翻译成一个Java源文件，并编译成为.class扩展名的字节码文件，打开源代码可以发现该文件内的类是一个继承自HttpJspBase的Java类，而HttpJspBase这个类继承自HttpServlet类。随着学习的深入，用户将体验到JSP的很多优势。

1. 开发简单方便

在JSP中的编辑跟编写HTML文件基本一样，在处理表单方面极为方便。对于设置HTTP报头，JSP同样提供了丰富的方法，使得JSP开发者在编写通用功能时十分便捷，从而花费更多的时间在业务逻辑上。

2. 跨平台

Java本身就有跨平台的特性，因此JSP程序可以在支持Java的平台上开发运行，显然这对平台移植极为有利。当JSP在更换服务平台时，若不涉及数据库等相关操作，则几乎不做任何变动就可以完成服务平台的迁移。当需要更换Web服务器时，JSP同样可以做到不修改或者少量修改就在新的Web服务器中编译、运行。

3. 高效率和高性能

JSP可以是Servlet的扩展，因此Java虚拟机为每一个请求创建一个单独的线程而不是进程，如此系统就能很快地处理请求。同时JSP只会被编译一次，即在首次加载时需要编译，这样就加快了系统的响应速率。当一个请求处理结束之后，相关JSP映射的Java类并不会从内存中删除，而是被保留在内存中，当下次同样的请求发生时，系统会提供更快的响应速度。

4. 低成本

众所周知Java是开源的开发语言，JSP也是基于Java的开源环境开发的动态网页技术，所以就省去了商业的付费项目。再有，开发者可以从众多的Java IDE中选择一款适合自己的开发工具来进行项目研发，当然，也可以利用文本编辑器直接编写，只是这样比较耗时而且容易出错。还有许多的商业软件可以使用，但是通常来说JSP的开发总成本比采用其他技术要低一些。

综上所述，采用 JSP 动态网页技术是目前 Web 开发者的最佳选择。

1.2.3　JSP的执行顺序

在编写JSP程序时，我们要了解它的执行顺序，这对于后续的学习会有很大的帮助。JSP程序的执行过程大致如下：首先客户端向Web服务器提出请求，然后JSP引擎负责将页面转换为Servlet，此Servlet经过虚拟机编译生成类文件，再把类文件加载到内存中执行，最后由服务器将处理结果返回给客户端。整个流程如图1.2所示。

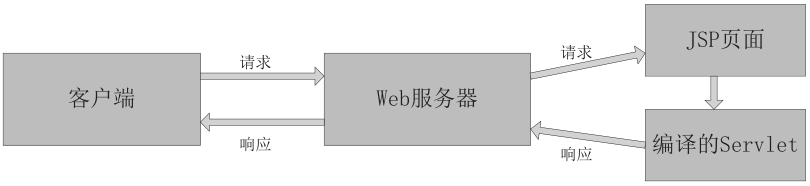

图1.2　JSP执行顺序

JSP页面代码会被编译成Servlet代码，所以从执行效率上来说肯定是没有Servlet快，但并不是每一次都需要编译JSP页面。当JSP被编译成类文件后，重复调用该JSP页面时，JSP引擎发现该JSP页面没有被改动过，那么就会直接使用编译后的类文件，而不会再次编译为新的Servlet。当然，如果页面被修改过，则需要重新加载编译。

1.2.4 一个JSP的简单实例

以下是JSP网页的一个简单实例，功能是输出01~10的相加结果：

```
-----------------------index.jsp--------------------------
<%@ page import="java.util.*" pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
  <head>
    <title>JSP简单例子</title>
  </head>
  <body>
    <%
        int count=0;
        for(int i=1;i<10;i++)
        {
           count+=i;
        }
        out.print("1到10的相加结果："+count);
     %>
  </body>
</html
```

注意：在第01行中，pageEncoding标签可以设定字符类型，第08~15行为Java代码。

该程序的主要作用是利用JSP输出1~10的和，其中代码是由简单的HTML代码和JSP表达式构成的，JSP表达式中包括一段Java程序段。

1.3 安装开发环境

上一节简单介绍了JSP中的相关概念，让读者对JSP有一个初步的了解。本节将介绍Java开发环境的安装和搭建，使读者能了解JSP所需的开发环境。

1.3.1 安装JDK17和配置环境变量

JDK是由SUN公司（现被Oracle收购）提供的Java开发工具和API，首先从Oracle公司的官网下载Java SE。需要注意的是，JDK的版本较多，各版本之间还存在不兼容问题以及操作系统的兼容性问题。JDK版本分为长期支持版本（LTS）和短期支持版本（STS）。短期支持版本一般在半年内会被更新替换，所以尽量不要使用。目前JDK8、JDK11、JDK17都是官网支持的长期版本。本书使用的是Windows 10的64位操作系统，所以下载了jdk-17_windows-x64_bin.zip版本，读者可以根据自身的操作系统下载对应的版本。

注意：API在Java世界中就是可以查找的参考内容，也就是Java方法的说明。

下载完成后解压到指定目录，笔者这里解压到F:\Program Files\Java\jdk-17.0.4路径下，解压完成后应该在bin目录下出现如图1.3所示的文件。

图1.3　JDK安装目录下的文件

该目录文件夹下包含了很多可执行文件，例如java.exe、javac.exe、javadoc.exe等。javac.exe是Java的编译器，用来编译Java文件，将它变为字节码。在DOC环境下，利用命令“javac test.java”的形式编译一个Java文件。java.exe是运行编译后的Java Class文件。java.exe也可以运行.jar文件，比如执行命令“java -jar test.jar”，就可以运行test.jar文件。

JDK11 之后默认不再安装 JRE，如果需要 JRE，可以进行手动配置。首先使用 Windows+R 组合键打开 cmd，切换到 JDK 安装目录。然后输入如下命令：

```
bin\jlink.exe --module-path jmods --add-modules java.desktop --output jre
```

执行完毕后没有任何输出，表示执行成功，如图 1.4 所示。生成成功后打开 JDK 目录，可以看到新增了 JRE 目录。

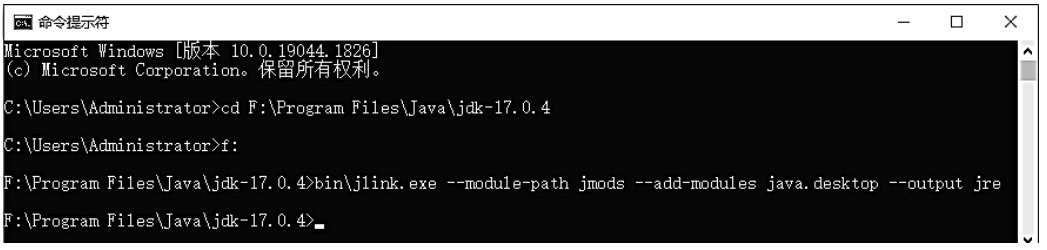

图1.4　生成JRE

接着就是配置Java的环境变量，配置的作用如下：

- 系统自动查找Java编译器路径。
- 服务器或其他需要依赖Java的程序在安装时需要知道Java路径。
- 编译和执行时指定Java路径。

配置环境变量的步骤如下：

01 以Windows为例，右击“此电脑”图标，在弹出的快捷菜单中单击“属性”命令，在打开的页面右侧单击“高级系统配置”选项，弹出“系统属性”对话框，在“系统属性”对话框中选择“高级”选项卡，单击“环境变量”按钮，如图1.5所示。

图1.5 配置环境变量

02 在弹出的“环境变量”对话框中单击“新建”按钮，分别设定JAVA_HOME、JRE_HOME、PATH和CLASSPATH这4个新的环境变量。4个变量值分别如下，其中%JAVA_HOME%、%JRE_HOME%表示引用值：

```
JAVA_HOME= F:\Program Files\Java\jdk-17.0.4
JRE_HOME= F:\Program Files\Java\jdk-17.0.4\jre
CLASSPATH= .;%JAVA_HOME%\lib;%JRE_HOME%\lib
PATH= ;%JAVA_HOME%\bin;%JRE_HOME%\bin
```

如果PATH变量已存在，则只需要在最后添加下方内容即可：

```
;%JAVA_HOME%\bin;%JRE_HOME%\bin
```

JDK9及以后的版本删除了tools.jar等JAR包，不再进行相关配置。配置完成后，打开cmd命令窗口，输入java -version命令，如果显示信息中JDK版本为17.0.4，则表示配置成功。

注意：CLASSPATH的值前面有一个“.”符号。

1.3.2 安装IntelliJ IDEA开发工具

IntelliJ IDEA是当下流行的Java IDE（Integrated Developing Environment）开发工具，深受开发者的青睐，它能支持目前主流的技术和框架，擅长企业应用、移动应用和Web应用的开发，并且拥有丰富的插件。

从官网下载IntelliJ IDEA 2022.1.4，或者使用搜索引擎搜索IntelliJ IDEA2022.1.4，然后选择合适的链接下载。下载完成后双击安装包，安装界面如图1.6所示。

图1.6　IntelliJ IDEA的安装界面

每一步都选择默认选项，直接单击“Next”按钮，进入安装进程，安装完成后会生成桌面快捷方式，也可根据操作系统的位数双击安装目录内bin目录下的idea.exe或idea64.exe运行程序，例如D:\Program Files\JetBrains\IntelliJ IDEA 2022.1.4\bin\idea.exe目录。运行IntelliJ IDEA后会提示输入注册码，这时可以输入注册码或者选择试用30天。

IntelliJ IDEA安装成功后，在新建项目或导入项目时还需配置项目所需的JDK。例如，配置已安装的JDK时，可选择File | Project Structure| Platform Settings | SDKs选项来添加JDK，如图1.7所示。

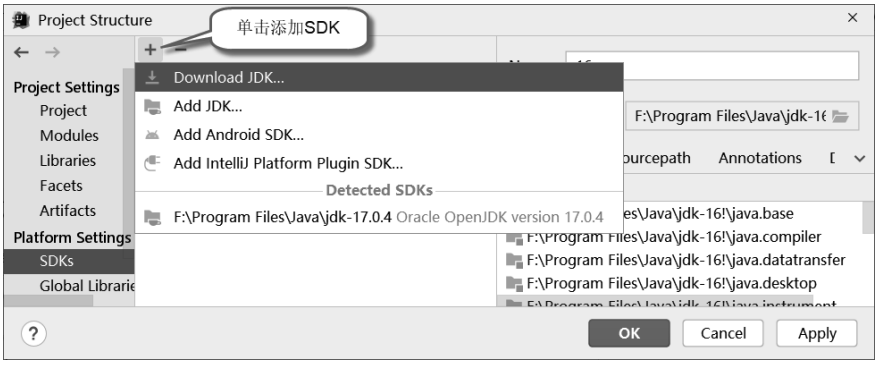

图1.7　IntelliJ IDEA查找JDK的选择界面

配置Tomcat时，可在Run | Edit Configurations中单击上方的+按钮，在左侧下拉框中选择Tomcat Server | Local选项，如图1.8所示，在弹出的窗口中配置Tomcat的安装路径，并在已安装的JDK版本中选择jdk17.0.4，如图1.9所示。

当然，还有很多Java IDE工具，例如MyEclipse、NetBeans、JCreator、JRun等。编辑JSP页面的工具还有Dreamveaver MX等。由于篇幅有限，这里就不一一介绍了。单从学习的角度来说，建议选择轻便型的IDE工具，如果是企业开发，那么人性化、智能化的IntelliJ IDEA将是最佳选择。

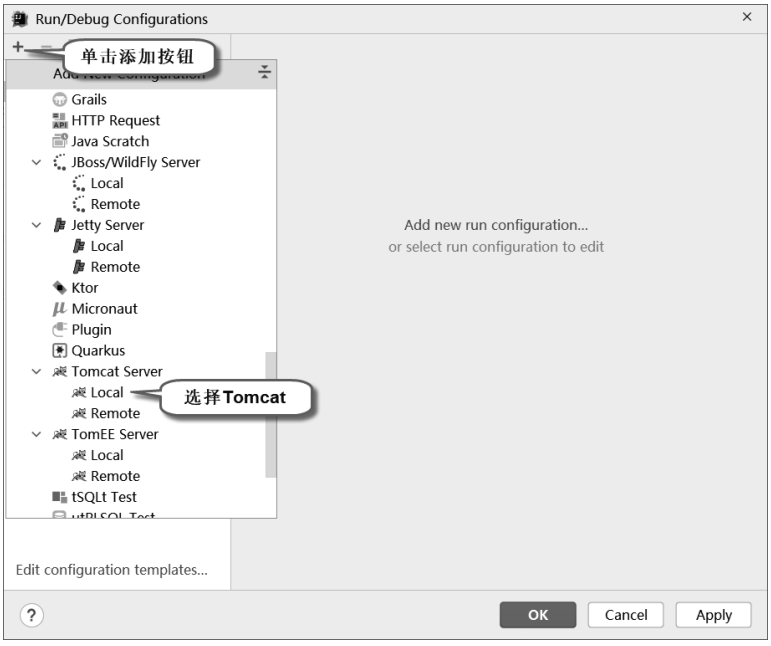

图1.8　打开配置Tomcat的界面

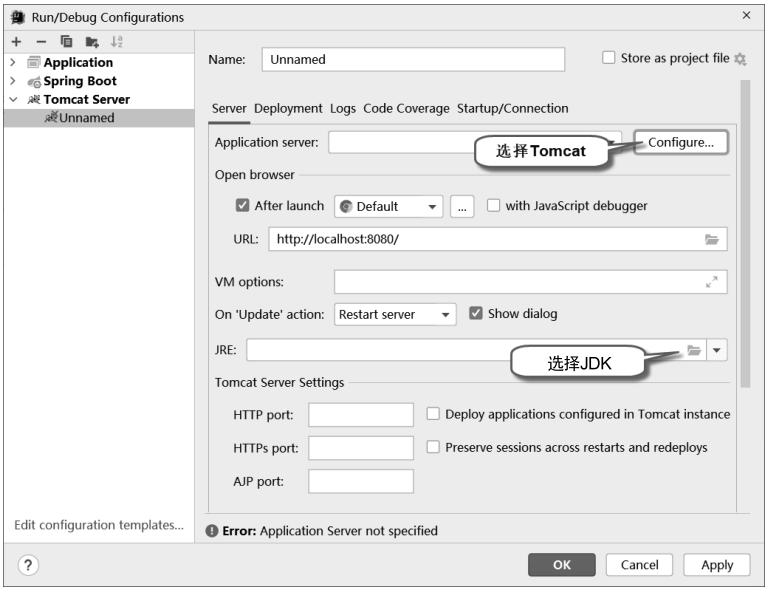

图1.9　选择Tomcat和JDK的界面

1.3.3　安装Tomcat 10服务器

1. 目录结构

Tomcat 是轻量级的 Web 应用服务器。可以从官网下载最新的 Tomcat 服务器版本，本书使用的 Tomcat 10.0.22 版本。下载完成后，直接解压 Tomcat 文件到指定的目录下，例如 F:\Program Files\apache-tomcat-10.0.22 中。下面介绍一下 Tomcat 的目录结构：

- bin文件夹，包含Tomcat服务器启动和终止的批处理文件。例如startup.bat、startup.sh、shutdown.bat、shutdown.sh、catalina.bat、catalina.sh等。其中startup.bat、shutdown.bat、catalina.bat是Windows中的批处理文件，startup.sh、shutdown.sh、catalina.sh是Linux中的脚本文件。
- conf文件夹，包含Tomcat的配置信息，主要有server.xml和web.xml两个配置文件。在server.xml中可以更改服务器端口和Web默认的访问目录。
- lib文件夹，存放Tomcat运行中需要的JAR包文件，例如catalina.jar、servlet-api.jar、tomcat-dbcp.jar等JAR包，正因为有这些包的支持，Tomcat才可以运行Web应用程序。
- logs文件夹，存放执行Tomcat的日志文件。
- temp文件夹，存放Tomcat的临时文件信息。
- webapps文件夹，是Tomcat默认的Web文件夹。本身自带两个admin应用和manager应用。开发人员可以直接将Web应用存放在该文件夹中。
- work文件夹，存放Tomcat执行应用后的缓存。

2．配置修改

在 Tomcat 服务器中，需要经常修改配置信息来满足系统的需求，例如在 server.xml 中可以更改服务器端口和 Web 默认的访问目录。

1）修改端口号

方法如下：

```
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000"
               redirectPort="8443" URIEncoding="UTF-8"/>
```

Tomcat默认的端口号为8080，也可以自定义其他端口，修改完毕后，保存server.xml，然后重启Tomcat服务器，这样服务器的端口就成功变更了。

2）修改Web默认的访问目录

方法如下：

```
<Host name="localhost"  appBase="webapps"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">
```

修改appBase中的文件夹地址。例如，将appBase的属性值webapps改为D:\test，修改后的文件如下：

```
<Host name="localhost"  appBase="D:\test"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">
```

这样就可以将Web默认的访问目录更改为D:\test，以后加载Web应用程序时就会在该目录下创建目录。

3）建立自身的Web目录

开发人员可以将应用部署在 Tomcat 服务器的默认 webapps 目录下，也可以部署在自己创建的目录下。方法如下：

（1）首先创建自身的目录 D:\test，其次配置 Web 目录，在 server.xml 文件的末尾</HOST>中加入如下语句：

```
<Context path="text" docBase="D:\test" debug="0" reloadable="true"></Context>
```

该语句的作用是将目录 D:\test 设置为 Tomcat 服务器的 Web 目录，将该文件的访问路径设置为 text。属性 docBase 的值为 D:\test，它是指应用的物理路径。修改后将 server.xml 文件进行保存。假设现在test.jsp页面位于D:\test目录下，那么页面的访问路径就为http://localhost:8080/text/test.jsp。

（2）在 bin 文件夹下，可以修改 catalina.bat 或者 catalina.sh 来更改 Tomcat 的启动配置信息。例如增加 Java 的运行内存：

```
set JAVA_OPTS=-XX:PermSize=512M -XX:MaxPermSize=512m -Xms512m -Xmx1024m
```

更多的修改内容可参见 Tomcat 官网说明。

1.4 小　　结

本章为读者介绍了搭建Web开发环境以及Web开发的基础知识，这些都是学习JSP开发技术之前必须掌握的知识。读者不仅需要理解和掌握本章内容，还需要亲自动手安装JDK、Tomcat、IntelliJ IDEA以及制作一个简单的JSP测试程序。

1.5 习　　题

（1）JSP动态网页技术有哪些优势？
（2）JSP引擎的作用有哪些？
（3）JSP页面执行的顺序是什么？
（4）如何修改Tomcat的服务器端口？
（5）如何创建一个自己的Web目录？