第3章

Jakarta EE开发环境的安装和配置



- 市场上主流 Jakarta EE 集成开发工具的类型和比较
- Eclipse 开发工具的安装和配置
- Eclipse 开发和部署 Jakarta EE Web 项目
- Spring Tools 4 for Eclipse 集成开发工具的安装和配置
- Spring Tools 4 for Eclipse 开发和部署 Jakarta EE Web 项目
- IntelliJ IDEA Community Edition 集成开发工具的安装和配置
- IntelliJ IDEA Community Edition 开发和部署 Jakarta EE Web 项目

3.1 Jakarta EE 开发工具的比较和选择

开发 Java 企业级应用项目,不能像传统的 Java 桌面应用项目,使用类似文本编辑器加 上 javac 编译工具就可以完成编程和部署任务。Java 企业级项目涉及众多技术和配置,使 用简单的记事本开发是难以想象的。软件企业在开发基于 Java EE 或 Jakarta EE 的企业级 项目时,都使用集成的开发工具(IDE Tools)进行项目的创建、依赖库的管理、代码的编写、 部署的生成等开发任务。目前市场上有非常多的支持 Jakarta EE 及 Java EE 企业级应用项 目开发的集成工具,如下 IDE 工具是软件开发企业普遍使用的。

1. Eclipse

Eclipse 是一个优秀的平台无关的 IDE 开发环境,为开发 Java、Java EE 和 Jakarta EE 提供了强大的开发和调试功能。Eclipse 遵循 OSGi 规范,其本身只是一个框架平台,但可 依赖丰富的插件完成各种强大的功能。Eclipse 采用 SWT 本地 GUI 库,使得运行速度较 AWT 和 Swing 有了很大提高,并提供了与操作系统一致的用户界面。

2. Spring Tools 4 for Eclipse

Spring Tools 4 for Eclipse(STS)是对 Eclipse IDE 的扩展,它内置了最新版的 Eclipse,可以在 Jakarta EE 的开发、发布,以及与应用程序服务器的整合方面极大地提高工作效率,

尤其是开发 Spring 框架的应用项目有天然优势。STS 具有功能丰富的支持最新 Jakarta EE 规范的集成开发环境,完整支持 Web 开发,如 HTML、CSS、JavaScript,内置支持 Spring Framework、Spring Boot 和 Spring Cloud 等,成为许多软件企业开发项目的首选工具。

3. IntelliJ IDEA

IntelliJ IDEA 一度被认为是最好的 Java IDE 集成开发平台,其以强大的即时分析和方便的重构功能深受 Java 开发人员的喜爱。IntelliJ IDEA 提供了一整套 Java 开发工具,包括: 重构、Java EE 和 Jakarta EE 支持、Maven、JUnit,并集成了 CVS、Git 和 GitHub 等。

4. Apache NetBeans

NetBeans 是由 Sun 公司建立的开放源代码的 IDE 工具,现在 Sun 公司的购并者 Oracle 公司已经把 NetBeans 捐献给了 Apache 软件基金会,其商标已变成 Apache NetBeans。NetBeans 是一个开放的、可扩展的开发平台,支持各种插件。除了可用于 Java、 Java EE、Java ME 开发外, NetBeans 也通过插件提供 C++等应用的开发,开发人员也可以 为 NetBeans 提供各种第三方模块以扩展 NetBeans 的功能。

另外,还有其他的 IDE 工具,但由于使用不是非常普遍,因此不再一一赘述。本章简要介绍软件开发企业使用较多的 Eclipse、STS 和 IntelliJ IDEA 这 3 种工具的安装、配置,以及开发和部署 Jakarta EE Web 项目的步骤和注意事项。

3.2 Eclipse IDE 工具的安装和配置

在 Java EE 及最新的 Jakarta EE 应用开发领域, Eclipse 是软件公司普遍使用的开发工具;另外, Eclipse 是开源产品, 不需要付费, 也没有版权限制。本节详细介绍 Eclipse IDE 的安装配置和项目开发。

3.2.1 Eclipse IDE 的下载

Eclipse 官方下载页面(http://www.eclipse.org/downloads/)如图 3-1 所示。截至编 写本书时,最新的 Eclipse IDE 版本是 2023-12。下载页面会自动检查用户使用的平台类



图 3-1 Eclipse IDE 官方下载页面

型,如作者使用的是 Windows 10,则其自动提供 Windows 平台的下载链接,且默认是 EXE 文件。但是,不推荐下载 EXE 的可执行文件版本,推荐下载其压缩文件版本。

单击 Download Packages 超链接,进入压缩文件版本下载页面,如图 3-2 所示。

Eclipse IDE 2023-12 R Packages

Ğ	Eclipse IDE for Java Developers 318 MB 301,229 DOWNLOADS The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration	*	Windows x86_64 macOS x86_64 AArch64 Linux x86_64 AArch64
	Eclipse IDE for Enterprise Java and Web Developers		
	514 MB 199,650 DOWNLOADS		
۲	Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.	*	Windows x86_64 macOS x86_64 AArch64 Linux x86_64 AArch64

图 3-2 Eclipse IDE 压缩文件版本下载页面

对应 Java 项目的开发, Eclipse IDE 提供了开发普通桌面单机应用的 Eclipse IDE for Java Developers 版本,以及开发企业级应用(Jakarta EE 应用)的 Eclipse IDE for Enterprise Java and Web Developers 版本,这里选择开发企业版 Java 应用的版本。

针对不同的操作系统,Eclipse分别提供了用于 Windows、macOS和 Linux 平台的开发工具。如果使用 Windows 平台,则单击 Windows x86-64 超链接,下载其压缩版本文件 eclipse-jee-2023-12-R-win32-x86_64.zip,将其保存到计算机的本地硬盘目录即可。

3.2.2 Eclipse IDE 的安装和启动

1. 安装 Eclipse

Eclipse 是典型的绿色软件,只需将下载的 ZIP 文件解压,不需要安装即可使用。推荐将其解压到目录 D:\tools\eclipse202312。

2. 启动 Eclipse

双击 Eclipse 解压目录下的 eclipse. exe 文件,即启动 Eclipse。

3. 选择工作区目录

Eclipse要求在项目的开发过程中选择指定的目录作为工作区,每个工作区保存所有配置信息,包括工作区的字符编码集、JDK 目录、Java 编译版本、服务器类型、Maven 的版本等。工作区选择界面如图 3-3 所示。

工作区的目录名最好不要有汉字和空格,以免将来代码测试和运行时出现 Bug。

4. Eclipse 工作环境

启动 Eclipse 后,关闭其默认的 Welcome 标题页,即进入项目开发工作台界面,如图 3-4 所示。

Eclipse IDE 工作台主要包含以下区域。

Eclipse IDE Launcher				×
Select a directory as workspace				
Eclipse IDE uses the workspace directory to store its preferen	ices and dev	elopment art	ifacts.	
91		-		
Workspace: E:\lesson\JakartaEE\ws2023eclipse	~	Browse		
Use this as the default and do not ask again				
Recent Workspaces				
	Г	Launch	Can	cel

图 3-3 Eclipse IDE 启动时工作区选择界面

💭 ws2023eclipse - Eclipse IDE						-		×
File Edit Navigate Search Project Run	Window Help							
□ • □ □ + • 0 • Q • Q • G • G	8 • @ A • @ 4 0 x	■ ■ N 3. (5. n) = 20 {	• {} • {}	\$\$\$ + \$ + ₹			Q	
🗈 Project Explorer × 🛛 🖻 🖏 🏹 🖇 🗖 🗖				- 0	$\mathop{\mathrm{B}\!\!\!\!\!}{}_{\mathbb{E}} \operatorname{Outline} \times$			- 0
There are no projects in your workspace. To add a project:					There is no act an outline.	ive editor	that p	rovides
Create a Maven project								
Create a Java EE EAR project								
Create a Dynamic Web project								
Create an EJB project								
Create a Connector project								
Create a Java EE application client project								
Create a deployable web project.								
Create a JPA project								
Create a project								
🚵 Import projects								
	Markers × Properties	# Servers 贈Data Source Expl	orer 🗈 Snipp	oets 🖉 Terminal			7	8 - 0
	Description	Resource	Path	Location	Туре			
0 items selected								: 0

图 3-4 Eclipse IDE 工作台界面

(1) 功能菜单区:在工作台界面的最上方,提供了 Eclipse 所有功能选择的菜单。

(2)便捷按钮区:在菜单区下面是便捷按钮区,该区域提供了常用功能的按钮,避免了 到菜单中选择,加快了功能选择速度。

(3)项目显示区:在工作台界面的左边,用于显示和管理当前工作区内的项目列表。 由于现在没有项目开发,因此此时显示的是常用的项目创建选择功能,如 Create a Maven project(创建 Maven 项目)等。

(4)代码编写区:在工作台界面的中间,便捷按钮下方和调试标题栏的上方。Eclipse IDE的代码编辑器与普通的文本编辑器相比,具备所支持编程语言的智能补齐功能,如编写 Java 代码,会自动提示对象的属性和方法,这是普通文本编辑器所不具备的,能极大加快代 码的编写速度,提高项目的开发效率。目前因为没有代码打开,所以代码编写区处于未激活 状态。

(5) 调试和日志区: 在工作台的右下方默认提供了 Servers、Markers、Terminal 等标题 区,用于查看项目的运行、服务器的启动/停止和运行状态等。

虽然可以将各个工作区的位置进行重新调整,但推荐使用此默认的布局安排,尽量不要 调整各自的位置。

一般情况下,可以把 Outline 标题页关闭,这样可以增加代码工作区的面积,有利于代码的编写和查看。关闭 Outline 标题页后,工作台界面变成图 3-5 所示的样式。

🔘 ws2023eclipse - Eclipse IDE					-		\times
File Edit Navigate Search Project Ru	un Window Help						
□ - 	6 • @ 4 • 9 4 5	× ≫ ⊞ ₩ 3. ⊕ .e ₹ 5	인 이 🕈 🖓 🔹 🕼	\$ \$ • • • • □	ť.	۹	8
n Project Explorer × 🖻 🕏 🖓 🖇 🖱 🗖							- 0
There are no projects in your workspace. To add a project:							
Create a Maven project							
Create a Java EE EAR project							
Create a Dynamic Web project							
Si Create an EJB project							
Create a Connector project							
E Create a Java EE application client project							
Create a deployable web project.							
Streate a JPA project							
🖆 Create a project							
ia Import projects	🖹 Markers × 🗇 Propertie	es 郗 Servers 觸 Data Source Exp	olorer 🚡 Snippe	ets 🖉 Terminal		8	8 🗖 🗖
	0 items	to an and the second seco	1				
	Description	Resource	Path	Location	Туре		
0 items selected				1			: 0

图 3-5 关闭 Outline 标题页后的工作台界面

3.2.3 配置 Java SE JDK 环境

Eclipse IDE 开发 Jakarta EE 应用项目,需要先配置 Java SE 的运行环境。在安装了 Java SE 后,确保使用 java -version 指令对 Java SE 的运行环境进行了测试,并需要记住 JDK 的安装位置。

应尽可能安装 Java SE JDK 的最新版本,如 JDK 17、JDK 19 等。现在很多 Java 应用框架已经不支持比较老的 JDK 版本,如 JDK 7、JDK 8 等,如 Spring 6.0 版本就要求至少 JDK 17 版本才能运行。

启动 Eclipse IDE 后,依次选择 Windows→Preferences→Java→Installed JREs,即可进入 Java 配置界面,如图 3-6 所示。

Eclipse IDE 自身也内置了 Java SE 的 JRE 运行引擎,但是没有包含全部的 JDK 软件, 推荐配置自己计算机上安装的 JDK 版本。

要配置使用自己的 JDK 版本,可单击 Add 按钮,打开图 3-7 所示的 JDK 版本选择界面。

选择默认选中的 Standard VM 即可,单击 Next 按钮,进入 JDK 安装目录选择界面,如图 3-8 所示。

选择 JRE home 的目录,即 D:/apps/jdk17(此处需要根据自己的安装路径选择,这是作者的 JDK 安装路径);默认的 JRE name 为 jdk17,通常不需要更改此名称;在 JRE system libraries 列表框中自动选择 JDK 17 的核心库 jrt-fs. jar,该库包括所有 JDK 17 的内置的包、接口和类的定义。

单击 Finish 按钮,即完成 Java JDK 的安装配置,IDE 工具会显示刚刚配置的 JDK 的安装信息,包括 Name 和 Location(安装位置)属性及其取值。

Preferences								-	- C)	×
type filter text		Installed J	REs						\Diamond	• 🗘 •	8
> Ant > CSS (Wild Web Developer) > Data Management > Gradle	^	Add, remov created Jav Installed JF	ve or edit JRE (a projects. REs:	definitions.	By default, the checke	d JRE is a	dded to the b	ouild pa	ath of ne	ewly	
> Help		Name	Location							Add	
> HTML (Wild Web Developer)		⊡ ≞∖jre	D:\tools\eci	ipse202212	\plugins\org.eclipse.j	ustj.ope	njdk.hotspot.	jre.full	.wi	Edit	Ē
✓ Java									D	uplicate	
> Appearance										omovo	
> Build Path Code Coverage										veniove	
> Code Style										earch	_
> Compiler											
> Debug > Editor											
> Installed JREs											
JUnit									_		
> Java EE		<							>		
> Java Persistence											
> JS/TS (Wild Web Developer)	~									Apply	
	-	1									
							Apply and	Close	C	ancel	_
				, ava (j.2).	Add JRE JRE Definition Specify attributes for a	ı JRE					
Add JRE					JRE home:	D:\apps\	idk17			Direct	ory.
RE Type					JRE name:	jdk17	9				
elect the type of JRE to add to the work	space				Default VM arguments:					Variab	oles.
					JRE system libraries:				_		
istalled JRE Types:				-1	> 🗟 D:\apps\jdk17\lib	⊳\jrt-fs.jar			Add E	kternal JA	ARs.
itandard 1.1.x VM									Javad	oc Locati	on
tandard VM									Source	Attachm	hent
									Externa	l annotat	tion
									1	Remove	
										Up	
										Down	
									Rest	ore Defa	ult
? < Back Next	>	Finish	Cancel		?	Back	Next >	Fir	nish	Cano	cel

图 3-7 JDK 版本选择界面

图 3-8 JDK 安装目录选择界面

安装新的 JDK 后,需要将此 JDK 设置为默认的 Java 运行环境,即选中前面的复选框即 可,如图 3-9 所示。

选中默认的 JDK 后,需要单击 Apply 或 Apply and Close 按钮,以使新的 JDK 开始 启用。

安装完 JDK 后,还需要配置 IDE 的 Java 编译版本,即生成哪个 Java 版本的代码,这里 推荐使用与 JDK 版本相符的编译版本。依次选择 Windows → Preferences → Java → Compiler,进入 Java 编译版本配置界面,如图 3-10 所示。

这里选择 Compiler compliance level 为 17 即可,单击 Apply 或 Apply and Close 按钮, 完成 Java 编译版本的配置。

Preferences X type filter text (> - - - 8 Installed JREs > Ant Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly > CSS (Wild Web Developer) created Java projects. > Data Management Installed JREs: > Gradle > Help Name Location Add... > HTML (Wild Web Developer) **⊠ ⊒∖ jd...** D:\apps\jdk17 Edit.. > Install/Update ∏∎∖jre D:\tools\eclipse202212\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win3 ✓ Java Duplicate. > Appearance Remove > Build Path Code Coverage Search.. > Code Style > Compiler > Debug > Editor > Installed JREs JUnit **Properties Files Editor** > Java EE > Java Persistence > JS/TS (Wild Web Developer) Apply > Language Servers ? 🖬 🗹 🔘 Apply and Close Cancel

图 3-9 设置安装的 JDK 为默认的 Java 运行环境

X

Preferences

type filter text	Compiler	<p th="" §<="" ⇒="" ▼=""></p>
> Install/Update ^	Configure Pr	niect Specific Settings
✓ Java	IDK Compliance	oject opecine octango.
> Appearance		17
> Build Path	Compiler compliance level:	17 ~
Code Coverage	Use 'release' option	
> Code Style	Use default compliance settings	
> Compiler	Enable preview features for Java 19	
> Debug	Preview features with severity level:	Warning ~
> Editor	, , , , , , , , , , , , , , , , , , ,	
> Installed JREs	Generated .class files compatibility:	17 ~
JUnit	Source compatibility:	17 ~
Properties Files Editor	Disellowidentifiers colled tosserth	Freeze
> Java EE	Disanow identifiers called assert.	EIIOI V
> Java Persistence	Disallow identifiers called 'enum':	Error ~
> JS/TS (Wild Web Developer)	Charafile Comparties	
> Language Servers		
> Maven	Add variable attributes to generated class files (used by	the debugger)
> Oomph	Add line number attributes to generated class files (used	d by the debugger)
> Plug-in Development	Add source file name to generated class file (used by the	e debugger)
> Run/Debug	Preserve unused (never read) local variables	
> Server	Inline finally blocks (larger class files, but improved perfe	ormance)
> Terminal	Store information about method parameters (usable via	reflection)
> TevtMate		

图 3-10 Java 编译版本配置界面

通常配置的 JDK 版本可以支持低于其版本的 Java 编译,但无法编译高于其版本的代码编译,如安装配置了 JDK 17,不能选择编译版本为 18 或 19,因为 JDK 是向下兼容的,不能向上兼容。

3.2.4 配置 Jakarta EE 服务器

在使用 Eclipse IDE 开发 Jakarta EE 应用项目之前,需要配置符合 Jakarta EE 规范的服务器,如 Tomcat、WildFly、GlassFish等,以便将项目部署和运行在这些服务器上。本书

68

主要以 Apache Tomcat 服务器为项目的部署服务器。

下面以配置 Tomcat 10.1.17 为例配置 Jakarta EE 服务器,此版本 Tomcat 是本书编写时的最新版本。

在顶部菜单中依次选择 Window→Preferences→Server→Runtime Environments,进入 服务器配置界面,如图 3-11 所示。

Preferences				×
type filter text	Server Runtime Envir	onments	$\Leftrightarrow \bullet \Rightarrow \Rightarrow$	• 8
 > General > Ant > CSS (Wild Web Developer) 	Add, remove, or edit se Server runtime environ	erver runtime environments. ments:		
> Data Management > Gradle > Help	Name	Туре	Add Edit	•
> HTML (Wild Web Developer) > Install/Update > Java			Remov	/e
 > Java EE > Java Persistence 			Search Column	 s
 > JS/TS (Wild Web Developer) > Language Servers > Mayen 			-	
> Oomph> Plug-in Development			-	
> Run/Debug V Server Audio				
Launching Overlays				
Profilers Runtime Environments				

图 3-11 服务器配置界面

在此服务器配置界面,可以完成符合 Jakarta EE 规范的服务器的增加、修改和删除操作。因为首次选择了新的工作区,所以没有任何服务器的配置, Server runtime environments 的服务器列表为空。

按如下顺序依次进行完成服务器的配置。

1. 选择服务器类型

在图 3-11 中,单击 Add 按钮,进入服务器类型选择界面。Eclipse 支持市场上流行的各种 Jakarta EE 服务器,在此选择能支持的最新 Apache 版本 Tomcat 10.1 服务器,如图 3-12 所示。

2. 配置服务器的安装目录和 JDK 版本

选择服务器的安装目录和 JDK 版本,如图 3-13 所示。这里选择 Tomcat 10.1.17,此处 为作者计算机的安装路径 D:\apps\apache-tomcat-10.1.17,具体位置应根据读者自己的安 装目录确定。另外,还需要将服务器的运行 JRE 环境选择为新安装的 JDK 环境。

单击 Finish 按钮,即可完成 Tomcat 10.1.17 服务器的配置。

3.2.5 创建 Jakarta EE Web 项目

Eclipse IDE 内置了创建 Jakarta EE Web 项目的向导,使用此向导可以非常方便地创建符合 Jakarta EE Web Profile 规范的动态 Web 项目。

-Jakarta EE企业级应用开发实例教程

70

New Server Runtime Environment – 🗆	×		
New Server Runtime Environment		New Server Runtime Environment	
Define a new server runtime environment		Tomcat Server Specify the Tomcat installation directory and JRE for t	his runtime.
Select the type of runtime environment:		The specified Name:	
type filter text		Apache Tomcat v10.1	
Apache Tomcat v8.5	^	Tomcat installation directory:	
Apache Tomcat v9.0 Apache Tomcat v10.0 Apache Tomcat v10.1 Apache Tomcat v10.1		D:\apps\apache-tomcat-10.1.17 apache-tomcat-10.1.15 JRE:	Browse Download and Inst
> > IBM		jdk17 ~	Installed JREs
Apache Tomcat v10.1 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, 7, 8 and Jakarta EE 9 and 10 Web modules.			
(?) < Back Next > Finish Cancel		⑦ < Back Next > F	inish Cance
图 3-12 选择服务器类型和版本		图 3-13 选择服务器目录和	I JDK 版本

在 Eclipse 工作台界面依次选择 File→New→Dynamic Web Project,启动 Jakarta EE Web 项目创建向导,如图 3-14 所示。

0	ws2023eclipse - Eclipse IDE				
File	Edit Navigate Search Project	Run Window He	elp		
6	New Open File Open Projects from File System Recent Files	Alt+Shift+N>		Maven Project Enterprise Application Project Dynamic Web Project EJB Project	b • 6 • ↔
	Close Editor Close All Editors Save	Ctrl+W Ctrl+Shift+W Ctrl+S		Connector Project Application Client Project Static Web Project JPA Project	
	Save As Save All Revert	Ctrl+Shift+S	10 m %3	Project CSS File JavaScript File	
	Move Rename Refresh Convert Line Delimiters To	F2 F5 >		Servlet Session Bean (EJB 3.x) Message-Driven Bean (EJB 3.x) Web Service Folder	

图 3-14 选择 File→New→Dynamic Web Project

当工作区中没有项目时,其左侧也有创建动态 Web 项目的超链接,即 Create a Dynamic Web project,单击后,即可进入动态 Web 项目创建界面,如图 3-15 所示。

如果工作区中已经有项目,则此超链接不会出现,因此推荐使用菜单命令进行项目的创建。选择创建 Dynamic Web Project,进入动态 Web 项目输入界面,如图 3-16 所示。

在图 3-16 中输入如下动态 Web 项目参数配置:

(1) Project name(项目名称): javaweb01。

(2) Project location(项目目录):使用默认工作区目录,即选中 Use default location 复选框。

(3) Target runtime(项目目标运行环境,即服务器):选择安装的 Apache Tomcat 10.1。

(4) Dynamic web module version(Web 规范的版本): 6.0。这是 Jakarta EE 10 Web Profile 规范支持的最新版本。

٥

💓 ws2023eclipse - Eclipse IDE							
File Edit Navigate Search Project R	un Window H	elp					
Ď • 🖩 🐚 🎋 • O • Q • Q • 🖓 •	6 - 3 -	● 4 ■	& I▶ II	. N 3	s.e ₹.₹	1 图 • 图 • 1	¢¢¢••
🔓 Project Explorer × 🖻 🕏 🍸 🕴 🗖 🖬							
There are no projects in your workspace. To add a project:							
👺 Create a Maven project							
管 Create a Java EE EAR project							
🚳 Create a Dynamic Web project							
Create an EJB project							
Create a Connector project							
Eff Create a Java EE application client project							
Create a deployable web project.							
🔐 Create a JPA project							
🖻 <u>Create a project</u>							
Import projects	Markers × 0 items	Properties	्म Serve	rs 🍿 Data	Source Expl	orer 🚡 Snipp	oets 🍠 Term
	Description	^			Resource	Path	Loca

图 3-15 没有项目时直接创建动态 Web 项目的选择界面

💭 New Dynamic Web Project					×
Dynamic Web Project Create a standalone Java-based Web Application c	or add it to a n	ew or existing E	nterprise App	olication.	0
Project name: javaweb01					
Project location Use default location					
Location: E:\lesson\JakartaEE\ws2024eclipse\java	web01			Brow	vse
Target runtime					
Apache Tomcat v10.1			~	New Ru	ntime
Dynamic web module version					
6.0					~
Configuration					
Default Configuration for Apache Tomcat v10.1			\$	- Mo	dify
A good starting point for working with Apache To to add new functionality to the project.	omcat v10.1 ru	ntime. Addition	al facets can	later be in	nstalled
EAR membership Add project to an EAR					
EAR project name: EAR			~	New Pr	roject
Working sets					
Add project to working sets				Ne	w
Working sets:				Sele	ct
0	< Back	Next >	Finish	Cá	ancel

图 3-16 创建动态 Web 项目输入界面

其他参数取默认值即可,单击 Next 按钮,进入如图 3-17 所示的动态 Web 项目的站点 地址(Context root)和 Web 文档存储位置(Content directory)的配置界面。

🔯 New Dynamic Web	o Project			-		×
Web Module Configure web modu	ıle settings.				F	9
Context root: ja Content directory: sr	vaweb01 c/main/webapp deployment descriptor					
?		< Back	Next >	Finish	Cancel	

图 3-17 Web 项目站点地址和 Web 文档存储位置配置界面

图 3-17 中, Context root 指定 Web 站点的访问起始路径, 默认是项目名, 即 javaweb01。 当将此项目部署到 Tomcat 上运行时, 其访问路径是 http://localhost:8080/javaweb01。

Content directory 指定 Web 文档的存储目录,在项目的 src/main/webapp 目录下。注意,这里只能存储 Web 文档,如 HTML 文件(.html)、CSS(.css)、JavaScript(.js)、JSP(.jsp),以及 Web 应用需要的其他辅助文件,如图片、视频、纯文本文件等;此目录不能存储 Java 源代码 文件。

默认情况下,Generate web.xml deployment descriptor 复选框是不选中的,即不生成 Web应用的配置文件 web.xml。这是因为新版的 Java EE 和 Jakarta EE 支持注解类配置 方式,可以不用配置文件 web.xml;而旧版本只支持 XML 配置方式,因此必须要有 web. xml 文件。实际项目开发时,可以根据需要决定是否选择生成 Web 的配置文件。这里推荐 选中此复选框,生成配置文件 web.xml。即使不用该配置文件,也可以查看配置文件的结 构和内容。

Project Explorer × 🗈 🕏 🖓 🗣 🕴 🗖	单击 Finish 打
 ✓ 送 javaweb01 > Sin Deployment Descriptor: javaweb01 	Jakarta EE Web 应
> @ JAX-WS Web Services	在项目 javawe
 () B src/main/java マ 副 Libraries 	心的目录。
> 🛋 JRE System Library [JavaSE-17] > 🛋 Server Runtime [Apache Tomcat v10.1]	(1) Java Res
➢ build∽ ➢ src	Java代码的目录。
∽ l≥ main l≥ java	(2) Java Resou
 ✓ ⋈ webapp ✓ ⋈ META-INF 	赖库,目前有 JRE 利
	(3) build: 此
le⇒ lib ⊠ web.xml	件,即. class 和依赖
	(4) $src/main/s$

图 3-18 生成的 Jakarta EE Web 应用项目

单击 Finish 按钮,生成图 3-18 所示的 akarta EE Web应用项目。

在项目 javaweb01 目录下,生成如下主要核 心的目录。

(1)Java Resources/src/main/java:存储 Java 代码的目录。

(2) Java Resources/Libraries:项目引入的依 赖库,目前有 JRE 和 Apache Tomcat 10.1。

(3) build:此目录存储编译后的 Java 类文件,即. class 和依赖库文件. jar。

(4) src/main/webapp: 存储 Web 文档目录。

(5) src/main/webapp/META-INF:存储

项目的配置信息。此目录为受保护的目录,客户端无法请求此目录下的文件。

(6) src/main/webapp/WEB-INF:存储 Web 项目的配置信息和关键文件。此目录不能被客户端访问,是受保护的目录。

(7) src/main/webapp/WEB-INF/web. xml: 生成的 Java Web 配置文件。

通常 Web 项目需要一个启动文件,动态 Web 项目推荐的文件是 index. jsp。选择项目 的 webapp 目录,右击,在弹出的快捷菜单中选择 new→JSP File 命令,弹出 JSP 文件生成向 导界面,如图 3-19 所示,在 File name 文本框中输入 JSP 文件名 index. jsp。

Project Explorer × Verify javaweb01 Big Deployment Descriptor: jav A JAX-WS Web Services Res A Java Resources	New JSP File JSP Create a new JSP file.			×
 > Dava Resources > > > sc/main/java > = Libraries > = Libraries > = Libraries > = Server Runtime [Apac > > build > > sc > > main > java > > webapp > > WETA-INF > META-INF > MANIFEST.MF > > WEB-INF > > ib > web.xml 	Enter or select the parent folder: javaweb01/src/main/webapp			
	? < Back Next > Finis	n	Car	ncel

图 3-19 JSP 文件生成向导界面

单击 Next 按钮,进入 JSP 文件模板选择界面,如图 3-20 所示。

💽 New JSP File	$ \square$ \times
Select JSP Template	
Select a template as initial content in	ine JSF page.
Use JSP Template	
Templates:	
Name	Description ^
New JSP File (html 4.01)	JSP with html 4.01 markup
New JSP File (html 5)	JSP with html 5 markup
New JSP File (xhtml)	JSP with xhtml markup
New JSP File (xhtml, xml syntax)	JSP with xhtml markup and xml sty
New JSP File (xhtml, xml syntax, JSP	JSP with JSP 2.0 specific tags, xhtm Y
Preview:	
<%@ page language=	<i>"iava"</i> contentType=^
nageEncoding="	¢Sancoding \"%
pageEncouring-	plencouling ///
html	
<html></html>	~
<	>
Templates are 'New JSP' templates for	and in the JSP Templates preference page.
? < Back	Next > Finish Cancel

图 3-20 JSP 文件模板选择界面

在图 3-20 中自动选择 html 5 的模板语法来生成 JSP 页面代码。目前项目采用 HTML 5 的编程规范已经成为标准,因此保持默认的选择即可。使用 JSP 向导创建的 JSP 文件如 图 3-21 所示。

Jakarta EE企业级应用开发实例教程

74

```
index.jsp ×
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2 pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4*<html>
5*<head>
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9*<body>
10
11 </body>
12 </html>
```

图 3-21 使用 JSP 向导创建的 JSP 文件

默认生成的 JSP 的代码文本编码集都是 ISO-8859-1,但现在项目中都采用 UTF-8 字符编码标准,因此需要手动修改 JSP 文件中所有的 ISO-8859-1,都改成 UTF-8,也可以在项目中修改 JSP 默认的字符编码集,即选择 Window→Preferences→Web→JSP Files,打开 JSP 配置界面,如图 3-22 所示。

Preferences	- 🗆 X
type filter text	JSP Files ⇔ ♥ ♥ §
 > JS/TS (Wild Web ^ > JSON > Language Server > Maven > Plug-in Developr > Run/Debug > Server > Spring > Terminal > TextMate Validation > Version Control > Web > Client-side Jav > CSS Files > HTML Files JavaScript > JSP Files 	Creating files Add this suffix (if not specified): jsp The following encoding will apply: Encoding: ISO 10646/Unicode(UTF-8) IANA: UTF-8 Search Include JSP matches in Java searches Restore Defaults Apply
? ` '	Apply and Close Cancel

图 3-22 JSP 配置界面

在 Encoding 下拉框中选择 ISO 10646/Unicode (UTF-8),则以后创建的 JSP 文件默认 的字符编码集就是 UTF-8。

在 JSP 文件的< body ></body >之间输入 JSP 页面要显示的内容,这里只输入了简单的显示标题文本:< h1 >欢迎使用 Jakarta EE Web </h1 >。

3.2.6 部署 Jakarta EE Web 项目

Eclipse IDE 支持在不离开工作台环境下直接部署 Web 项目到配置的服务器上,大大 缩短了程序员的开发和测试时间。3.2.5 小节已经创建了动态 Web 项目,并创建了项目的 首页 index.jsp 文件,下面讲解如何将此项目部署到配置的 Tomcat 10.1.17 服务器并使用

浏览器请求该页面。

1. 部署 Jakarta EE Web 项目

选择项目并右击,在弹出的快捷菜单中选择 Run As→1Run on Server 命令,启动 Web 项目部署向导,如图 3-23 所示。

~ 營 jaka > 뿹 l	•	New Go Into) T ∕∿M haR	<pre>ranguage=_juvu concentrype=_ce. rset=UTF-8" pageEncoding="UTF-8"; }E html></pre>
> 25. > 26.		Show In Show in Local Terminal	Alt+Shift+W > >	
> @ :		Copy Copy Qualified Name Paste Delete Build Path Source Refactor	Ctrl+C Ctrl+V Delete Alt+Shift+S > Alt+Shift+T >	narset= <i>"UTF-8"></i> Insert title here 使用Jakarta EE Web
	Ð	Import Export Refresh Close Project Close Unrelated Project	> > F5	perties 卷 Servers 觸 Data Source Explorer 匼 Snippets 尋 Ter
	0. 0 *	Coverage As Run As Debug As	> > >	J 1 Run on Server Alt+Shift+X, R Lc ☑ 2 Java Application Alt+Shift+X, J

图 3-23 启动 Web 项目部署向导

2. 选择服务器并输入基本项目信息

弹出部署服务器选择界面,如图 3-24 所示。当首次执行部署任务时,选中 Manually define a new server 单选按钮,在 Select the server type 列表框中选择配置的服务器类型,这里选择 Tomcat v10.1 server。IDE 工具自动根据配置的服务器确定 Server's host name、 Server name 和 Server runtime environment 这 3 个参数值,通常不需要修改,直接取默认值即可。

How do you want to select the server?
O Choose an existing server
Manually define a new server
Select the server type:
type filter text
Tomcat v10.0 Server
🗄 Tomcat v10.1 Server
Publishes and runs J2EE, Java EE, and Jakarta EE Web projects and server
configurations to a local Tomcat server.
Server's host name:
Server name: Tomcat v10.1 Server at localhost
Server runtime environment: Apache Tomcat v10.1 VAdd.
Configure runtime environments.
Always use this server when running this project
? < Back Next > Finish Cancel

图 3-24 部署服务器选择界面

单击 Next 按钮,进入部署项目选择界面,如图 3-25 所示。

💭 Run On Server			D X
Add and Remove Modify the resources that are	e configured on the	server	
Move resources to the right to	o configure them o	n the server	
Available:		Configured:	
	Add > < Remove Add All >>	🕼 javawet	001
⑦ < Back	<< Remove All Next >	Finish	Cancel

图 3-25 部署项目选择界面

默认情况下,要部署的项目会自动出现在右侧部署区内,左侧是此工作区的所有没有选择部署的项目。当前工作区只有一个项目,因此左侧未部署项目的区域为空;如果工作区创建多个项目,此区域就会有没有被选择部署的项目列表。Eclipse IDE 支持同时部署多个项目到服务器,因此工作区内的项目名称不能相同,否则无法部署;当然,也无法创建相同项目名称的项目。

3. 启动服务器并测试 Web 项目运行

单击图 3-25 中的 Finish 按钮, Eclipse 自动启动选择的部署服务器,这里选择 Tomcat 10.1.17,在工作台的 Console 窗口会显示 Tomcat 10.1.17 的启动信息,如图 3-26 所示。

■ Problems # Servers 『Terminal 瞳Data Source Explorer □Properties □Console ×
Tomet v10.1 Server at localhost [Apache Tomeat] Diappsyidk[7]binjavaw.exe (2024年1月9日上午8:39:01)[pid: 8896]
信息: 正在启动 Servlet 引擎: [Apache Tomcat/10.1.17]
1月 09, 2024 8:39:07 上午 org.apache.catalina.util.SessionIdGeneratorI
警告: 使用[SHA1PRNG]创建会话ID生成的SecureRandom实例花费了[121]毫秒。
1月 09, 2024 8:39:07 上午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-nio-8080"]
1月 09, 2024 8:39:07 上午 org.apache.catalina.startup.Catalina start
信息: [1522]毫秒后服务器启动

图 3-26 部署服务器启动信息

图 3-26 所示的服务器的启动信息包括服务器的类型、监听 HTTP 请求的端口(当前显示为 8080)、服务器启动时间(1522 毫秒)。服务器启动后,Eclipse IDE 自动启动操作系统默认的浏览器(作者计算机默认的浏览器是 Google 的 Chrome),并自动请求 Web 项目的默认主页,显示请求的 index.jsp 的 HTTP 响应内容,如图 3-27 所示。

图 3-27 表明项目已成功部署到 Tomcat 10.1.17,并开始监听客户端浏览器的请求。

	Inser	t title h	iere		×	+				
\leftarrow	\rightarrow	C	0	localho	st:8080/ja	kartaweb03/				
	常用	. V	ue	DE IDE	Scala	Python	Spring	Books	Node	

欢迎使用Jakarta EE Web

图 3-27 Eclipse IDE 启动服务器并请求项目的默认主页

3.2.7 Maven 的安装和配置

使用创建 Dynamic Web Project 方式创建 Jakarta EE Web 项目对于初学者学习 Jakarta EE 项目编程非常简洁方便,但是这种方式在开发大型企业级项目时,尤其是当项目 中需要引入非常多的依赖库时,需要手动复制各种 JAR 文件,会导致依赖库管理混乱、版本 不统一和冲突等问题。

软件企业在开发 Jakarta EE 企业级项目时,通常使用项目构建工具对项目的代码、依赖和编译进行管理,目前使用最多的构建工具有 Maven 和 Gradle。本书使用 Maven 构建工具,对于 Gradle 的使用请读者参阅相关文档。

Maven 是 Apache 软件基金会的开源项目,访问其官网(https://maven.apache.org),可以参阅 Maven 的文档并下载发布的最新版本。Maven 官方网站主页显示如图 3-28 所示。

Welcome to Apache Maven

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's documentation from a central piece of information.

If you think that Maven could help your project, you can find out more information in the "About Maven" section of the navigation. This includes an in-depth descript some of its main features.

This site is separated into the following sections, depending on how you'd like to use Maven:

Use	Download, Install, Configure, Run Maven	Maven Plugins and Maven Extensions		
	Information for those needing to build a project that uses Maven	Lists of plugins and extensions to help with your builds.		
Extend	Write Maven Plugins	Improve the Maven Central Repository		
	Information for developers writing Maven plugins.	Information for those who may or may not use Maven, to metadata into the central repository.		
Contribute	Help Maven	Develop Maven		
	Information if you'd like to get involved. Maven is an open source community and welcomes contributions.	Information for those who are currently Maven develope contributing to the Maven project itself.		



其中,Use 栏目中包括 Download、Install、Configure、Run Maven 超链接,开发人员可以 完成 Maven 的下载、安装、配置和运行。

1. 下载 Maven

在图 3-28 中单击 Download 超链接,进入 Maven 下载页面,如图 3-29 所示。

从 Maven 下载页面可见,目前 Maven 的最新版本是 3.9.6。其中,System Requirements 中介绍了 Maven 3.9.6 需要的环境信息,包括 JDK 版本必须在 8 以上和硬盘容量在 10MB 以上,而对内存和操作系统没有最低要求,可见 Maven 能适应任何操作系统。

推荐下载 Maven 的二进制(Binary)版本。Maven 提供了两种格式的压缩文件,分别为

Downloading Apache Maven 3.9.6

Apache Maven 3.9.6 is the latest release: it is the recommended version for all users.

System Requirements

Java Development Kit (JDK)	Maven 3.9+ requires JDK 8 or above to execute. It still allows you to build against 1.3 and other JDK ν
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, disk space will be u repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts (tested on many Unix flavors) a

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the inst Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles a

	Link	Checksums
Binary tar.gz archive	apache-maven-3.9.6-bin.tar.gz	apache-maven-3.9.6-bin.tar.gz.sha512
Binary zip archive	apache-maven-3.9.6-bin.zip	apache-maven-3.9.6-bin.zip.sha512
Source tar.gz archive	apache-maven-3.9.6-src.tar.gz	apache-maven-3.9.6-src.tar.gz.sha512
Source zip archive	apache-maven-3.9.6-src.zip	apache-maven-3.9.6-src.zip.sha512

图 3-29 Maven 下载页面

ZIP和 tar.gz,对于 Windows 平台,推荐下载 ZIP 文件;而 Linux 和 macOS 平台推荐下载 tar.gz 文件。

单击 apache-maven-3.9.6-bin.zip 下载超链接,将下载的文件保存到指定目录即可。

2. 安装 Maven

Maven 也是绿色软件产品,直接将下载的 ZIP 文件解压即可。在作者的计算机上将其 解压到 D:/apps/maven396 目录下,解压后的安装目录和文件结构如图 3-30 所示。

		名称	修改日期	类型	大小
		bin	2024/1/9 9:23	文件夹	
	~	📕 boot	2024/1/9 9:23	文件夹	
	*	Conf	2024/1/9 9:23	文件夹	
	*	📕 lib	2024/1/9 9:23	文件夹	
	*] m3	2024/1/9 9:25	文件夹	
F发实例教程		LICENSE	2023/11/28 9:59	文件	19 K
		NOTICE	2023/11/28 9:59	文件	5 K
		README.txt	2023/11/28 9:59	文本文档	3 K

图 3-30 Maven 的安装目录和文件结构

注意,解压后的目录没有 m3 子目录,该目录是新创建的,用于存储 Maven 下载的依赖 库 jar 的目录。读者可以自己创建任意位置的目录作为本地仓库的位置,但推荐在 Maven 目录下创建,这样比较容易管理;同时,推荐使用 m3 目录名,使用 Maven 2 时用户大都习 惯使用 m2 作为本地仓库的目录名,到 Maven 3 时应自然过渡到 m3 目录名。

3. 配置 Maven

需要对安装好的 Maven 进行配置,以便使用创建的本地仓库目录 m3。另外, Maven 在

下载项目需要的依赖库时,默认从国外的 Maven 中央仓库中读取,对于国内的开发者,其下 载速度非常慢,经常由于下载超时导致 Maven 项目错误。为解决这一问题,阿里云在国内 创建了 Maven 的镜像仓库,其中保存了与国外中央仓库相同的库文件,并保持同步,为此一 定要配置阿里云 Maven 镜像仓库。

Maven 的配置通过安装目录下的 conf 子目录中的 settings. xml 文件完成。使用记事本打开该文件,找到< localRepository >标记,如图 3-31 所示。

图 3-31 Maven 本地仓库目录的配置

在 settings. xml 文件中增加如下配置代码,完成本地仓库存储目录的设置:

```
<localRepository>D:\apps\maven390\m3 </localRepository>
```

其中,本地仓库的位置根据自己创建的目录确定,这里设置为作者前面创建的 m3 目录。 配置好本地仓库的位置后,下一步需要配置阿里云 Maven 镜像仓库。在 settings. xml 文件中找到< mirrors >标记,在其内部增加如下 Maven 阿里云镜像的配置代码:

配置 Maven 阿里云镜像仓库后的内容如图 3-32 所示。

```
<mirrors>
<mirrors>
<id>maxen-default-http-blocker</id>
<mirrorOf>external:http:*</mirrorOf>
<name>Pseudo repository to mirror external repositories initially using HTTP.</name>
<url>http://o.0.0.0/v(vurl>
<blocked>true</blocked>
</mirror>
<id>nexus-aliyun</id>
<mirrorOf>central</mirrorOf>
<name>Nexus aliyun</name>
<url>http://maven. aliyun.com/nexus/content/groups/public</url>
</mirrors>
```



配置后,保存 settings. xml 文件即可。

3.2.8 Eclipse IDE 配置 Maven

安装完 Maven 后, Eclipse IDE 需要配置 Maven,以实现 Maven 项目的构建和管理。依次选择 Window → Preferences → Maven → Installations, 弹出 Maven 安装配置界面, 如图 3-33 所示。

an/opuate	Select the installatio	n used to launch Maven:		
FF	Name	Details		Add
istence	EMBEDDED	3.9.5/3.9.500.20231113-2345		Edit
Web	WORKSPACE	NOT AVAILABLE [3.0,)		Eun
erver				Remove
tion Pr				
es				
arnin				
ons				
figura				
ntegr				
Мар				
ooku				
iterface	Note: Embedded ru	ntime is always used for dependent	cy resolution	
ungs 🗸			Restore Defaults	Apply

图 3-33 Maven 安装配置界面

图 3-33 的 Installations 列表框中列出了 IDE 已经安装的 Maven 及其版本,可见 IDE 已经内置了 Maven,其版本是 3.9.5。如果直接使用此版本,就不需要安装 Maven,直接使用即可。

由于这里安装的是 Maven 3.9.5,而上节下载和安装的是 3.9.6,因此需要对 Maven 进行配置。单击图 3-33 图中的 Add 按钮,弹出如图 3-34 所示的 Maven 安装向导界面。

🔘 New Maven Ru			×		
Specify attributes	for a Maven installation				
Installation type:	• External O Workspace				
Installation home:	D:\apps\maven396		[Directory	/
Installation name:	maven396				
Additional extension	on libraries:				
				Project.	
				Remove	е
?		Finish		Cance	el

图 3-34 Maven 安装向导界面

选择 Maven 3.9.6 的安装目录,IDE 自动给出 Installation name,即 maven396,直接使用此名称即可。如果确实需要,可以手动修改此安装名。

单击 Finish 按钮, IDE 会自动更新 Maven 的安装列表, 如图 3-35 所示。

安装成功 Maven 后,还需要选中新安装的 Maven 3.9.6,作为 IDE 的默认 Maven。单

ype filter text	Installations			$\langle \neg \bullet \circ \rangle \bullet $
> Install/Update ^	Select the installatio	n used to launch Maven:		
 Java Java Persistence Java Persistence Java Versistence Java Versistence Maven Annotation Pr Archetypes Discovery Errors/Warnin Installations Java Configura Java Cenfigura Java EE Integra Lifecycle Map 	Name EMBEDDED WORKSPACE maven396	Details 3.9.5/3.9.500.20231113-2345 A NOT AVAILABLE [3.0,) D:\apps\maven396 3.9.6		Add Edit Remove
Source Looku	Note: Embedded ru	ntime is always used for depende	ncy resolution Restore De	faults Apply

图 3-35 Maven 的安装列表界面

击 Apply and Close 按钮,使其配置激活并启用。

完成 Maven 的安装后,下一步需要使用 Maven 的配置文件。依次选择 Window→ Preferences→Maven→User Settings,进入 Maven 配置文件选择界面,如图 3-36 所示。

Preferences		
type filter text	User Settings	> - 8
Installations ^ Java Configura	Global <u>S</u> ettings (<u>open file</u>):	
Java EE Integra	D:\apps\maven396\conf\settings.xml	Browse
Lifecycle Map	User <u>Settings (open file</u>):	
Source Looku	D:\apps\maven396\conf\settings.xml	Browse
User Settings	Update Settings	
> Oomph > Plug-in Developr	Local Repository (From merged user and global settings):	7
> Run/Debug]
<pre>> Server *</pre>	Restore Defaults	Apply
? 2 2 9	Apply and Close	Cancel

图 3-36 Maven 配置文件选择界面

将 Global Settings 和 User Settings 都选择为刚刚修改过的 Maven 配置文件 D:\apps\maven396\conf\settings. xml,可以看到 Local Repository(本地仓库)的目录自动更改为前面配置的本地仓库的目录。

单击 Apply and Close 按钮,完成 Maven 的配置工作。

3.2.9 创建 Maven Web 项目

安装并配置好 Maven 后,就可以创建 Maven Web 项目。但是,由于 Maven Web 项目 模板一直没有更新,使用的 JDK 版本是 1.5, Web 版本是 2.5,因此无法创建最新版的 Jakarta EE Web 项目。因此,在使用 Maven Web 模板创建项目后,还需要对配置进行修改,使其符合最新版 Jakarta EE 规范的 Web 项目后,才能使用 Jakarta EE Web 规范进行企

业级 Web 项目的开发。下面详细介绍创建 Maven Web 项目及修改配置的步骤。

1. 创建 Maven Web 项目

在 Eclipse IDE 中依次选择 File→New→Maven Project,弹出 Maven 项目创建界面,如 图 3-37 所示。

New Maven Project				\Box ×
New Maven project				Ð
Select project name and location				
Create a simple project (skip archetyp	e selection)			
Use default Workspace location				
Location:				Browse
Add project(s) to working set				
Working set:				More
Advanced				
0	< Back	Next >	Finish	Cancel

图 3-37 Maven 项目创建界面

在此界面中选择使用默认的工作区目录(选中 Use default Workspace location 复选框) 作为项目的目录即可,单击 Next 按钮,进入 Maven 项目的模板选择界面,如图 3-38 所示。

Catalog: Internal				~	Configure
ilter:					
Group Id	Artifact Id		Version		
org.apache.maven.archetypes	maven-arch	etype-profiles	1.0-alpha-4		
org.apache.maven.archetypes	maven-arch	etype-quickstart	1.1		
org.apache.maven.archetypes	maven-arch	etype-site	1.1		
org.apache.maven.archetypes	maven-arch	etype-site-simple	1.1		
org.apache.maven.archetypes	maven-arch	etype-webapp	1.0		
An archetype which contains a	sample Mave	n Webapp projec	:t.		
Show the last version of Arcl	netype only	Include snap	shot archetype	es Ado	l Archetype
Advanced					
/ druneed					

图 3-38 Maven 项目的模板选择界面

在 Catalog 下拉列表中选择 Internal,即内置模板库,系统会自动更新此类别下的所有 Maven 项目模板列表;再选择 Artifact Id 为 maven-archetype-webapp 1.0,此模板用于生 成 Maven Web 项目。单击 Next 按钮,进入 Maven 项目信息输入界面,如图 3-39 所示。

在此界面中输入如下信息。

(1) Group Id:项目的公司信息,通常使用域名,如 com. city、com. ibm。

(2) Artifact Id:项目的名称,如 cityoa、oaerp,此名称作为工作区的项目名称。

New Mave Specify Ar	en project chetype parameters				Ψ
Group Id:	com.city				~
Artifact Id:	oaweb00				~
Version:	1.0 ~]			
Package:	com.city.oa	-			~
Properties	✓ run archetype genera available from archetype	ation interactively e:			
Name	Value				<u>A</u> dd <u>R</u> emove
Ad <u>v</u> ance	d	,			
?		< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel

图 3-39 Maven 项目信息输入界面

(3) Version:项目的版本号,模板自动给出 0.0.1-SNAPSHOT。该参数可以改为 1.0, 也可以输入任意版本号。

(4) Package:项目的起始包名,默认给出 groupid. artifactid 作为包名,如 com. city. oaweb00。该参数可以使用默认值,也可以输入新的起始包名,如 com. city. oa。

(5) Properties available from archetype: 不需要输入任何新的属性名。

单击 Finish 按钮, Eclipse IDE 会使用 Maven 选择的模板生成 Maven Web 项目。

2. 修改 Maven Web 项目的 JDK 版本

由于 Maven Web 模板版本比较旧,生成的 Maven Web 项目默认使用 JDK 1.8,因此首 先需要将其改为现在使用的 JDK 17。

选择项目并右击,在弹出的快捷菜单中选择 Properties→Java Build Paths→Libraries 命令,进入图 3-40 所示的项目 JDK 配置界面。

Properties for oaweb0	0	– 🗆 X
	A 3 build path entries are missing.	↓ ↓ ↓
Java Build Path ^ > Java Code Style	Source ☺ Projects ➡ Libraries ⁰₀ Order and Export ♥ Module Dependencies JARs and class folders on the build path:	
Javadoc Location	> 🛋 JRE System Library [JavaSE-1.8]	Add JARs
> Java Editor	> 🛋 Maven Dependencies	Add External JARs
> Maven		Add Variable
Project Facets		Add Library
Project Natures Project References		Add Class Folder
Run/Debug Settings		Add External Class Folder
Server Service Policies		Edit
Targeted Runtimes		Remove
Task Tags		Kemove
> Validation		Migrate JAR File
Web Page Editor		
Web Project Setting ✓		Apply
?	Apply	y and Close Cancel

图 3-40 项目依赖库配置界面

选择 JRE System Library [JavaSE-1,8],单击 Edit 按钮,弹出 JDK 版本修改界面,如 图 3-41 所示。

💭 Edit Library		-	
JRE System Library			A
Select JRE for the project bu	ild path.		
System library			
O Execution environment:	JavaSE-1.8 (jdk17)	\sim	Environments
Alternate JRE:	jdk17	~	Installed JREs
○ Workspace default JRE (j	dk17)		
(?)		Finish	Cancel

图 3-41 JDK 版本修改界面

选择 Alternate JRE 复选框,并在其下拉列表中选择前面安装的 jdk17,单击 Finish 按钮,完成 JDK 版本的修改。

在修改 JDK 版本的同时,也需要修改 Java 的编译版本。选择图 3-40 左侧功能区中的 Java Compiler,进入 Java 编译版本界面,如图 3-42 所示。

ype filter text	Java Compiler		<> ▼ □	> 🕶 8
> Resource	Enable project specific settings	Configure Work	space Settir	igs
> Client-side JavaScrin	JDK Compliance			
Coverage	Use compliance from execution environment on the 'Java Build Path'			
Deployment Assemt	Compiler compliance level:		17	~
Java Build Path	Use 'release' option			_
> Java Code Style	Use default compliance settings			
> Java Compiler	Enable preview features for Java 21			
> Java Editor	Preview features with severity level:		Warning	~
JSP Fragment	Generated .class files compatibility:		17	~
> Maven	Source compatibility:		17	~

图 3-42 配置 Maven Web 项目的 Java 编译版本界面

在 JDK Compliance 下的 Compiler compliance level 下拉框中选择 17 即可,单击 Apply and Close 按钮,启用并激活配置。与此同时,需要修改项目的 Maven 配置文件 pom. xml, 增加支持编译 JDK 17 的插件(plugins)。打开项目根目录中的 pom. xml 文件,在< build > 标记下增加 2 个插件,如图 3-43 所示。

其中,插件 maven-compiler-plugin 用于指定 Maven 编译的 Java 源代码和目标类的版本,这里设置为 JDK 17; 插件 maven-war-plugin 用于 Maven 生成 Jakarta EE Web 项目的 WAR 部署文件。

修改 Maven 的配置文件 pom. xml 后,需要进行 Maven 项目的更新操作。选择项目并 右击,在弹出的快捷菜单中选择 Maven→Update Project 命令,弹出 Maven 项目更新对话 框,如图 3-44 所示。

选择要更新的项目,单击 OK 按钮即可。

٥

 \times

	Update Maven Project Select Maven projects and update options	
	Available Maven Codebases	
	⊡ i⊒ oaweb00	Select All
		Add out-of-date
		Deselect All
		Expand All
		Collapse All
<pre>tbuild> <finalname>oaweb00</finalname> <plugins></plugins></pre>	☐ Offline ✓ Update dependencies ☐ Force Update of Snapshots/Releases ☐ Force Update of Snapshots/Releases	
<pre><groupid>org.apache.maven.plugins</groupid> <artifactid>maven-war-plugin</artifactid> <version>3.4.0</version> </pre>	Opdate project configuration from pom.xml Gefresh workspace resources from local filesystem Clean projects OK	Cancel
图 3-43 Maven 项目配置文件 pom. xml 增加插件	图 3-44 Maven 项目更新对	话框

🔘 Update Maven Project

图 3-43 Maven 项目配置文件 pom. xml 增加插件

3. 增加项目的 Server 依赖

新创建的 Maven Web 项目是没有 Jakarta EE 服务器依赖的,不增加服务器依赖就无 法编写 Web 组件(如 Servlet 和 JSP 等)。增加服务器依赖与修改 JDK 依赖都在相同的界 面操作,在此Java Build Path界面中选择 Libraries 标签,如图 3-45 所示,进入项目的依赖 库配置窗口。

Properties for oawe	900	— 🗆 X
type filter text	Java Build Path	▼ §
> Resource ^ Builders	A Source B Projects Libraries Of Order and Export Module Dependence	lencies
> Client-side JavaS	JARs and class folders on the build path:	
Coverage	✓ [®] ₀ Modulepath	Add JARs
Deployment Ass Java Build Path	→ NRE System Library [JavaSE-17]	Add External JARs
> Java Code Style	> 🛋 Maven Dependencies	Add Variable
> Java Compiler		Add Library
> Java Editor		Add Class Folder
JSP Fragment > Maven		Add External Class Folder
Project Facets Project Natures		Edit
Project Referenc		Remove
Run/Debug Setti		
Service Policies		Migrate JAR File
Targeted Runtim *		Apply
(?)		Apply and Close Cancel

图 3-45 Java Build Path 界面

在图 3-45 中,首先选择 Classpath 文件夹,再单击右侧的 Add Library 按钮,弹出 Add Library 窗口,如图 3-46 所示。

选择 Server Runtime,单击 Next 按钮,进入 Server Library 依赖库选择界面,如图 3-47 所示。

library			×	
Add Library			8	
Select the library type to add.			a)	
Connectivity Driver Definition CXF Runtime EAR Libraries JRE System Library JUnit				
Maven Managed Dependencies Plug-in Dependencies				FA
Server Runtime				🛃 Add Library — 🗆 X
User Library Web App Libraries				Server Library
				Select a server runtime for the project build path.
				Runtime environments:
				🗄 Apache Tomcat v10.1
(?) < Back Next >	ish	Car	ncel	? < Back Next > Finish Cancel
图 3-46 Add Library 復	窗口			图 3-47 Server Library 依赖库选择界面

在图 3-47 选择配置的 Apache Tomcat v10.1,单击 Finish 按钮,即完成项目的服务器 依赖配置,配置后的 Libraries 在 Classpath 下新增了 Server Runtime[Apache Tomcat v10.1], 如图 3-48 所示。

type filter text	Java Build Path	(⇒ ▼ ⊂) ▼
> Resource ^ Builders	Bource Projects Libraries % Order and Export Module	Dependencies
> Client-side JavaS	JARs and class folders on the build path:	
Coverage	v ⊕₀ Modulepath	Add JARs
Deployment Ass	> 🛋 JRE System Library [JavaSE-17]	1115
Java Build Path	✓ [⊕] _☉ Classpath	Add External JARs
> Java Code Style	> 🛋 Maven Dependencies	Add Variable
> Java Compiler	Server Runtime [Apache Tomcat v10.1]	و معامل المار الم
Javadoc Locatior		Add Library
Java Editor		Add Class Folder
JSP Fragment		Add Sutarnal Class Falda
Maven		Add External Class Folde
Project Facets		Edit
Project Natures		
Project Referenc		Remove
Run/Debug Setti		a strength to be with
Server		Migrate JAR File
Service Policies		
Tardeled Kunfin		Apply

图 3-48 增加服务器依赖后的 Libraries

最后,单击 Apply 或 Apply and Close 按钮,启用项目中此服务器依赖。

配置完服务器依赖后,需要将 Maven Web 模板生成的 Web 配置文件 web. xml 内容更新为符合 Jakarta EE 10 Web Profile 规范的版本,这里直接将使用 Dynamic Web Project 创

建的 Web 项目的 web. xml 内容复制并替换即可。

Maven 模板默认创建的 web. xml 的代码如下所示:

DOCTYPE web - app PUBLIC</th
" - //Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >
< web - app >
<pre>< display - name > Archetype Created Web Application </pre>

从以上代码可见, Maven 默认创建的 Web 的版本是 2.3, 对应的 Tomcat 版本是 6.0, 无法支持 Jakarta EE Web 项目。

替换后的 web. xml 代码如下所示:

从代码可见,其Web Module 版本是 6.0,这是 Jakarta EE 10 Web 规范的最新版本。

最后修改项目的配置信息,步骤如下:选择项目并右击,在弹出的快捷菜单中选择 Properties→Project Facets 命令,进入项目的配置参数界面,如图 3-49 所示。

将 Dynamic Web Module 的值改为 6.0,将 Java 的版本改为 17,其他属性参数不用修改,单击 Apply and Close 按钮,即完成参数的配置。

再次选择项目并右击,在弹出的快捷菜单中选择 Maven→Update Project 命令,完成项目的更新。

为测试项目的部署和运行,通常在 Maven Web 中创建一个主页 JSP 文件。选择项目中的 src/main/webapp 目录,右击,在弹出的快捷菜单中选择 New→JSP File 命令,创建 Web 项目的 主页文件 index.jsp。在 index.jsp 文件的< body >和</body >之间输入 HTML 代码:

<h1>欢迎使用 Maven Web 项目</h1>

修改后的 index. jsp 文件内容如图 3-50 所示。

4. 部署和运行 Maven Web 项目

Maven Web 项目编写完成后,即可部署和运行。Eclipse IDE 工具内部署和运行 Maven Web 项目与传统的 Dynamic Web Project 相同,其执行过程如下:选择项目并右击, 在弹出的快捷菜单中选择 Run As→Run on Server 命令,进入服务器选择界面,会提示选择

ype filter text	Project Facets		\$ ▼ \$ ▼
Java Build Path Java Code Style	Configuration: <custom></custom>		✓ Save As Delete
 Java Compiler Javadoc Locatior 	Project Facet	Version 6.0 •	Details Runtimes
 Java Editor JSP Fragment Maven Project Facets Project Natures Project Referenc Refactoring Histi Run/Debug Settii Server Service Policies Targeted Runtim Task Repository Task Tags Validation Web Content Se Web Page Editor 	Axis2 Web Services CXF 2.x Web Services G Dynamic Web Module EAR G EB Module EJBDoclet (XDoclet) Java JavaServer Faces JAX-RS (REST Web Services) ++ JAXB JCA Module + UPA Static Web Module Utility Module Web Fragment Module Web ragment Module	1.0 6.0 3.1 1.2.3 17 2.3 1.7 2.3 1.1 2.2 1.6 2.2 3.0 - 1.2.3 - 1.2 - 1.2 - - - - - - - - - - - - -	Adds support for the Java or Jakarta Servlet API, for generation of dynamic Web page content. Requires the following facet: Java 11 or newer Conflicts with the following facets: Application Client module Dynamic Web Module EAR EB Module JCA Module Static Web Module Utility Module Web Fragment Module

图 3-49 Maven Web 项目的参数配置界面

🗟 ind	ex.jsp \times
1	<pre><%@ contentType="text/html; charset=UTF-8"</pre>
2	pageEncoding="UTF-8"%>
3	html
4	<html></html>
5	<head></head>
6	<meta charset="utf-8"/>
7	<title>Insert title here</title>
8	
9	<body></body>
10	<h1>欢迎使用<u>Maven</u> Web项目</h1>
11	
12	

图 3-50 修改后的 index. jsp 文件内容

指定的服务器。如果已经配置了部署的服务器,则应该选中 Choose an existing server 单选 按钮,并选择已有的 Tomcat v10.1 Server at localhost,如图 3-51 所示。

选择已有的 Tomcat 10.1 服务器后,单击 Next 按钮,进入项目部署选择界面,如 图 3-52 所示。

从图 3-52 可以看出, 左侧的 Available 列表框显示的是工作区未部署的项目, 右侧 Configured 显示的是要部署到服务器的项目。通过选择 Add 或 Add All 按钮, 可以将项目 增加到 Configured 部署区; 也可以选择 Remove 或 Remove All 按钮, 将部署区的项目移动 非部署区。

选择好要部署的项目,单击 Finish 按钮,启动 Tomcat 服务器,并运行部署的项目,可以 从 IDE 工具的 Servers 标题区看到服务器的启动状态和部署的项目,如图 3-53 所示。

🗃 Run On Server – 🗆 🗙	🗑 Run On Server	– 🗆 X
Run On Server	Add and Remove	
Select which server to use	Modify the resources that are conf	igured on the server
How do you want to select the server?	Move resources to the right to con	figure them on the server
Choose an existing server	Available:	Configured:
O Manually define a new server	🕞 javaweb01	aweb00
Select the server that you want to use:	a oaweb2024	Add
type filter text		Add
Server State		Remove
✓ ➢ localhost		
🗄 Tomcat v10.1 Server at localhost 🛛 🔓 Stopped		
?, 8 and Jakarta EE 9 and 10 Web modules. Always use this server when running this project ? <	<?< Back	Remove All Next > Finish Cancel
图 3-51 服务器选择界面	图 3-52	项目选择界面
Markers ♣ Servers × ➡ Console ▼ ♣ Tomcat v10.1 Server at localhost [Started, Synch ➡ oaweb00 [Synchronized]	ronized]	

图 3-53 Servers 的标题区

如果 Eclipse IDE 没有 Servers 标题区,可以按照如下顺序操作显示该标题区:依次选择 Window→Show Views→Other→Server→Servers,在图 3-54 所示界面中选择 Servers,单击 Open 按钮,即可打开监控 Server 运行的 Servers 标题区。

🔅 Show View			×
type filter text			
> 🗁 Maven			^
> 🗁 Mylyn			
> 🗁 Oomph			
> 🗁 Plug-in Development			
✓			
級 Servers			
> 🗁 Terminal			- 10
> 🗁 Version Control (Team)			
> 🗁 Web Services			
> 🗁 XML			
> 🗁 Other			
			~
0.5		Cane	al
Ope	511	Canc	er

图 3-54 打开监控服务器运行标题区

3.3 Spring Tools 4 for Eclipse 的安装和配置

由于在企业级 Jakarta EE 应用项目开发中都使用各种框架技术,尤其是 Spring 框架技术,而传统的 Eclipse IDE 工具没有对 Spring 项目提供支持,因此开发者需要安装各种

-Jakarta EE企业级应用开发实例教程·

90

Spring 支持插件,比较麻烦且容易出错。

Spring 项目开发组为此专门开发了 Spring Tools 4 for Eclipse 开发工具(以下简称 STS),该工具内置了最新版的 Eclipse,并自动安装好需要的各种插件,极大地方便了广大 开发者使用 Spring 框架技术,并提高了项目的开发效率,深受软件开发企业的欢迎。目前 STS 在软件企业中的使用已经超过了 Eclipse。

3.3.1 STS 的下载和安装

访问 STS 的官方网址(https://spring.io/tools/),进入 STS 下载页面,可以看到不同 平台的 STS 工具的下载超链接,如图 3-55 所示。

Spring Tools 4 for Eclipse
The all-new Spring Tool Suite 4. Free. Open source.
4.21.0 - LINUX X86_64
4.21.0 - LINUX ARM_64
4.21.0 - MACOS X86_64
4.21.0 - MACOS ARM_64
4.21.0 - WINDOWS X86_64

图 3-55 STS 下载页面

针对不同的平台,选择对应的 STS 下载超链接即可。例如,Windows 平台选择 4.21.0-WINDOWDS X86_64,下载的文件类型为 JAR 的 Java 类库,其文件名比较长,具体如下: spring-tool-suite-4-4.21.0. RELEASE-e4.30.0-win32. win32. x86_64. self-extracting.jar。

下载完成后,推荐将其改为较短的文件名,如 sts4.21.0.jar。如果计算机安装 JDK 正确无误,Windows 平台会自动将 JAR 文件关联到 Java JRE,在文件名前显示 Java 虚拟机的 图标,如图 3-56 所示。

名称	修改日期	类型	大小
🔬 sts4.21.0.jar	2023/12/7 8:52	Executable Jar File	627,580 KB

选择下载的文件,直接双击即可启动 Java 引擎并执行该 JAR 文件进行解压,解压后的 目录结构如图 3-57 所示。

图 3-56 修改文件名后的 STS 安装文件

此电脑 > D (D:) > sof	tware > Spring > STS			
	名称 ^	修改日期	类型	大小
*	sts-4.21.0.RELEASE	2024/1/11 10:59	文件夹	
*	🏂 sts4.21.0.jar	2023/12/7 8:52	Executable Jar File	627,580 KB

图 3-57 解压后的 STS 目录结构

如果 Windows 平台没有自动关联到 JRE 引擎,也可以手动选择关联,方法如下:选择文件并 右击,在弹出的快捷菜单中选择打开方式,这里选 择 Java (TM) Platform SE binary,如图 3-58 所示。

推荐选中"始终使用此应用打开.jar 文件"复选框,如此可直接双击 JAR 文件运行。

如果无法找到 JRE,则进入 cmd 命令行,导 航到下载的 JAR 文件目录,直接输入如下命令, 也可以执行解压和安装工作:

java - jar sts4.21.0.jar

需要注意的是,不要使用解压软件如 360 压 缩、WAR工具等直接解压 JAR 文件,否则生成的 目录与要求不符,影响工具的启用和运行。

3.3.2 STS 插件的安装

你要如何打开这个文件?
继续使用此应用
Java(TM) Platform SE binary
其他选项
其他选项
360压缩
番 Microsoft Store 中查找应用
更多应用↓
」 始终使用此应用打开 jar 文件
确定

图 3-58 选择 JAR 文件的打开方式

由于现在软件企业很少开发纯 Jakarta EE Web 项目,也很少直接编程 Servlet 和 JSP, 都是开发以 Spring Boot 为基础的 REST API 微服务项目,因此 STS 工具默认没有内置安 装 Java Web 的插件工具,这一点与 Eclipse IDE 不同。但是,本书内容以 Jakarta EE Web 开发为主,涉及编写其 Web 组件代码,因此需要原有 Eclipse 内置的企业级 Java Web 开发 插件,应手动安装。

Eclipse IDE 提供的开发 Java 企业级 Web 应用的插件名称是 Eclipse Enterprise Java and Web Developer Tools。

启动 STS,选择工作区目录后(此操作与 Eclipse 相同),按如下操作顺序完成此插件的 安装。选择 Help→Eclipse Marketplace,进入 Eclipse Marketplacec 安装插件界面,如 图 3-59 所示。

在 Find 文本框中输入插件的名称,即 Eclipse Enterprise Java and Web Developer Tools,单击 Go 按钮,即开始在 Marketplace 中查找此插件。如成功找到该插件,则会显示插件的名称和版本(编写本书时的版本是 3.31,该插件会不断更新),单击 Install 按钮,下载和安装此插件,并弹出插件模块选择界面,如图 3-60 所示。

这里不需要选择额外的模块,直接单击 Confirm 按钮,STS 工具即开始进行插件的安装。安装插件过程中会提示是否信任所安装插件的界面,单击 Select All 按钮,再单击 Trust Selected 按钮,表示信任所有的插件,即可继续安装过程,如图 3-61 所示。



图 3-59 Eclipse Marketplace 安装插件界面

Press Confirm to continue with the installation. Or go back to choose more

Confirm Selected Features



💌 🔳 🖗 Eclipse Enterprise Java and Web Developer Tools 3.31 https://download.eclipse.org/r
🗹 🖶 Eclipse Java EE Developer Tools (required)
🗹 🖶 Eclipse Java Web Developer Tools (required)
🗹 🖶 Eclipse Java Web Developer Tools - JavaScript Support (required)
모 🖶 Eclipse Web Developer Tools (required)
🗹 🖶 Eclipse Web JavaScript Developer Tools (required)
🗹 🖶 Eclipse XML Editors and Tools (required)
☑ 🖶 JST Server UI (required)
🗌 🖗 Axis2 Tools
CXF Web Services
🗌 🖗 Dali Java Persistence Tools - Common
🗌 🖗 Dali Java Persistence Tools - EclipseLink Common
🗌 🏇 Dali Java Persistence Tools - EclipseLink JAXB Support
🗌 🖗 Dali Java Persistence Tools - EclipseLink JPA Support
🗌 🖗 Dali Java Persistence Tools - JAXB Support
🗌 🖗 EclEmma Java Code Coverage
🗌 🖗 Eclipse JSON Editor and Tools
🗌 🕸 Eclipse XSL Developer Tools
AX-WS DOM Tools
🗌 🖗 JAX-WS Tools
□ 🖗 JSF Tools
🗌 🖗 JSF Tools - Tag Library Metadata (Apache Trinidad)
Main Market And American State (1998) Provide the American State (1998)
🗌 🖗 m2e-wtp - JPA configurator for WTP (Optional)
Main Market Mar Market Market Mark
(?) < Install More Confirm > Finish Cancel

图 3-60 插件模块选择界面

进入提示重启界面,如图 3-62 所示。

单击 Restart Now 按钮,重新启动 STS 开发工具后,需要进行与 Eclipse IDE 相同的配置任务,其配置方法与 Eclipse IDE 相同,如配置 JDK、Java 编译版本、部署的服务器、 Maven 安装、Maven 配置和 JSP 字符编码集,这里不再赘述。STS 配置完成后,即可进行 Web 项目的开发,编写 Web 组件,如 Servlet 和 JSP 等,其与 Eclipse IDE 都相同,这里不再赘述。

觉 Trust Artifacts

 93

Do you trust these signers? \triangle The displayed originator names are not necessarily a reliable certification of origin. For PGP keys, verification is typically achieved by querying the key's fingerprint against a trusted key server.

Туре	ld/Fingerprint	Name		Validity Dates	
PGP	1c6f3d42d6e6b07a66155			✓ Valid, expires 2027-11	1
✓ PGP	ae20288fb9ac6b5068749			✓ Valid, expires 2028-07	7
🗹 Remembe	r selected signers 🗌 Always	trust all content		Select All Desele	ect All
1c6f2d42	d6a6b07a6615550c70b024d	026b42220			
10015042	0000078001555507008240	540044825			
Details E	xport				
Classifier	Id		Version		
osgi.bundle	com.google.protobuf		2.4.0.v20110513110	0	
osgi.bundle	jakarta.el		4.0.0.v20210105-0527		
osgi.bundle	jakarta.servlet		5.0.0.v20210105-052	27	
osgi.bundle	jakarta.servlet.jsp		3.0.0.v20210105-052	27	
osgi.bundle	javax.activation		1.1.0.v20121113054	9	
osgi.bundle	javax.mail		1.4.0.v201005080615		
osgi.bundle	javax.wsdl		1.6.2.v201012040545		
osgi.bundle	javax.xml.rpc		1.1.0.v201209140446		
osgi.bundle	javax.xml.soap		1.2.0.v201005080501		
osgi.bundle	org.apache.xml.serializer		2.7.1.v201005080400		
osgi.bundle	org.apache.commons.disc	overy	0.5.0		
				Trust Selected Canc	el

图 3-61 选择信任插件界面

Software Updates	×
Restart Spring Tool Suite 4 to apply the software update?	
Restart Now	No

图 3-62 提示重启界面

3.4 IntelliJ IDEA 工具的安装和配置

IntelliJ IDEA(以下简称 IDEA)是 JetBrains 公司的旗舰产品,其是 Java 编程语言的集成开发环境(IDE),被业界公认为优秀的 Java 开发工具,尤其在智能代码助手、代码自动提示、重构、Java EE 支持、版本管理工具(git、svn 等)、JUnit、CVS 整合、代码分析、创新的GUI 设计等方面,其功能可以说是超常的。IDEA 的宗旨就是"Develop with pleasure",全球越来越多的开发者开始选择 IDEA 作为其 Java 项目的开发工具。

IDEA 在 2001 年 1 月发布 1.0 版本,同年 7 月发布 2.0 版本,之后基本每年发布一个版本。3.0 版本之后,IDEA 屡获大奖,其中又以 2003 年赢得的 Jolt Productivity Award 和 JavaWorld Editors's Choice Award 为标志,奠定了 IDEA 在 IDE 中的地位。IDEA 于 2018 年 8 月发布新版本 18.2.2,同时将版本更改为 2018.2.2,以后的版本号命名都遵循此标准。目前

IDEA 的最新版本 Version: 2023. 3. 2 Build: 233. 13135. 103 于 2023 年 12 月 20 日发布。

IDEA 分为 Ultimate Edition(旗舰版)和 Community Edition(社区版)两个版本。其中 旗舰版可以免费试用 30 天,到期需要付费使用;社区版完全免费,从不过期,但是功能上对 比旗舰版有所删减。图 3-63 展示了 IDEA 旗舰版和社区版的功能对比。

	IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition 🚯
Java, Kotlin, Groovy, Scala	\checkmark	\checkmark
Maven, Gradle, sbt	\checkmark	\checkmark
Git, GitHub, SVN, Mercurial, Perforce	\checkmark	~
Debugger	\checkmark	\checkmark
Docker	\checkmark	√ *
Profiling tools 🚯	~	
Spring, Jakarta EE, Java EE, Micronaut, Quarkus, Helidon, and more 🚯	\checkmark	
HTTP Client	\checkmark	
JavaScript, TypeScript, HTML, CSS, Node.js, Angular, React, Vue.js	\checkmark	
Database Tools, SQL	~	
Remote Development (Beta)	~	
Collaborative development	\checkmark	I



For Organizations 🛉 For Indi Yearly billing save 2 months Mon Intell^{ij} IDEA Ultimate The Leading Java and Kotlin IDE

Intellij IDEA
per user, first year
5,000.00CNY
second year
4,000.00CNY
third year onwards
3,000.00CNY

通过图 3-63 比较可见, IDEA 社区版不能直接支持 Jakarta EE, 因此开发 Spring 的企业级项目时选择 IDEA 旗舰版。但是, 不是所有人都有能力每年付费, IDEA 旗舰版价格不菲, 其官网给出的价格如图 3-64 所示。

从 2023 年开始 IDEA 支持人民币付费, 图 3-64 所示的价格是针对企业每个用户的第1年 价格;第2年是 4000元;第3年是 3000元。针 对个人用户第1年是 1400元;第2年是 1120元; 第3年是 840元。国内的软件企业规模大的有 上万开发者,小的也有几百开发者,如果使用 IDEA 也是一笔不小的开支。因此,国内软件公 司很少使用 IDEA 旗舰版本进行项目开发,要么 使用 IDEA 社区版,要么使用开源的其他 IDE 工 具,如 STS 或 Eclipse IDE。

图 3-64 IDEA 旗舰版购买价格

IDEA 社区版虽然不直接支持 Jakarta EE 企业级项目开发,但是通过安装第三方插件, 一般也能满足开发需要。

3.4.1 IDEA 的下载和安装

进入 IDEA 的官方网站(https://www.jetbrains.com/idea/),其主页如图 3-65 所示。



图 3-65 IDEA 主页

单击 Download 超链接,进入 IDEA 下载页面,选择 IDEA 社区版,如图 3-66 所示。



图 3-66 IDEA 社区版下载页面

IDEA 下载页面自动检查客户使用的平台,生成符合平台的安装软件超链接。对应 Windows 平台,推荐下载 EXE 文件。单击.exe 下载超链接,开始下载 IDEA 社区版的 IDE 软件。将下载的社区版安装软件 ideaIC-2023.3.2.exe 保存到指定的目录,并双击启动该安 装程序,如图 3-67 所示。

选择指定的安装目录,单击 Next 按钮,进入选择关联文件和启动配置界面,如图 3-68 所示。

这里推荐选中创建桌面快捷方式 IntelliJ IDEA Community Edition 复选框,其他复选 框不用选中,直接单击 Next 按钮,安装软件便开始复制文件操作,文件复制完成后,即完成 了 IDEA 的安装。

3.4.2 IDEA 的启动和配置

IDEA 安装完成后,操作系统的桌面上会生成此工具的快捷启动图标,双击即可启动 IDEA 开发工具,如图 3-69 所示。

IntelliJ IDEA Community Edition 安装	- □ ×
安装程序将把 IntelliJ IDEA Community E 录,请点击 [浏览(B)] 并选择其他的文f	dition 安装到以下目录。要安装到另一个目 f夹。 点击 [下一步(N)] 继续。
安装目录 tools\IDEAC\	浏览(B)
所需空间: 2.4 GB 可用空间: 309.3 GB	
	< 上一步(P) 下一步(N) → 取消(C)
图 3-67 IDEA 社区比	板安装时指定安装目录
 IntelliJ IDEA Community Edition 安装 安装选项 配置 IntelliJ 	- □ × IDEA Community Edition 安装
创建桌面快捷方式 □ Intellij IDEA Community 更新上下文菜単 □ 添加 "将文件夹作为项目打开" - 创建关联	更新 PATH 变量(需要重启) □添加"bin"文件夹到 PATH
回連大林 □.java □.gradle □.groovy	.kt .kts .pom
	< 上一步(P) 下一步(N) > 取消(C)

图 3-68 选择 IDEA 关联文件和启动配置界面

与 Eclipse 和 STS 不同, IDEA 启动时自动识别平台安装的 JDK, 前提是平台需要设置 JAVA_HOME 和 PATH 环境变量, IDEA 根据这两个环境变量的值自动配置 Java JDK, 极 大地简化了 Java 项目的开发。

由于 IDEA 社区版没有提供对 Java EE 或 Jakarta EE 的支持,因此默认情况情况下其 无法创建 Jakarta EE Web 项目。但是,IDEA 社区版提供了对 Maven 的支持,开发者可以 使用 Maven Web 模板创建 Maven Web 项目方式实现 Java EE 或 Jakarta EE Web 项目的 开发,这一点与 Eclipse 或 STS 是相同的。

为开发 Maven Web 项目, IDEA 需要配置 Maven 环境。启动 IDEA, 在图 3-69 中选择 Customize 选项卡, 再单击 All settings 超链接, 如图 3-70 所示。

进入 IDEA 配置界面,选择"Build, Execution, Deployment"→Build Tools→Maven,可 以看到图 3-71 所示的 Maven 配置界面。

Maven 主路径(Maven home path)选择 Maven 3.9.6 的安装路径,用户设置文件(User



图 3-70 IDEA 设置选择界面

settings file)选择 confi 目录下的 setting. xml 文件,修改本地仓库(Local repository)的路径为 maven396下的 m3,单击 Apply 按钮,完成 Maven 的配置。

为将来测试 Web 项目的运行和访问,推荐配置 Web 浏览器和预览。在配置界面中选择 Tools→Web Browsers and Preview,进入浏览器选择界面,如图 3-72 所示。

١

Ŀ	Settings							×
Q. > > ~	Appearance & Behavior Keymap Editor Plugins Version Control Build, Execution, Deployme > Build Tools > Maven Gradue	Build, Execution, I Work offline Execute goals re Print exception : Always update s Internet Output level: Checksum policy:		Deployment Build Tools cursively stack traces napshots ialog for new Maven projects	Maven 🕒			
> >	Gradie Gant > Compiler > Debugger Remote Jar Repositories Coverage Java Profiler ☆ Kubernetes ☆ Required Plugins Trusted Locations Languages & Frameworks Tools Settings Sync Advanced Settings		Multiproject build fai Thread count Maven home path: User settings file: Local repository: I Use settings fro	I policy: D:\apps\maven396 (Version: 3.9.6) D:\apps\maven396\conf\setti D:\apps\maven396\m3 m .mvn/maven.config	Default ings.xml	-T option		 ✓ … ✓ Override ✓ Override
(?						ок	Cancel	

图 3-71 Maven 配置界面

🖭 Settings				×		
Q	Tools $ ightarrow$ Web Browsers and Pre	Tools → Web Browsers and Preview				
 Appearance & Behavior Keymap Editor Plugins Version Control Build, Execution, Deployment Languages & Frameworks Tools Space Qodana Actions on Save 	+ 2 1 3 0 Name ✓ Chrome ✓ Firefox Safari Opera Internet Explorer ✓ Edge	Family	Path chrome firefox safari opera iexplore msedge			
Web Browsers and Preview External Tools Terminal Code With Me Database > Diff & Merge Features Suggester Features Trainer Server Certificates Shared Indexes > Tasks	Default Browser: System default Show browser popup in the editor If For HTML files For XML files Reload behavior Reload page in browser: Reload page in built-in preview:	On Save ~ On Save ~				
Settings Sync Project-level settings will be applie	id to new projects		OK Cancel	Apply		

图 3-72 浏览器选择界面

在浏览器选择界面中,IDEA 自动列出操作系统安装的浏览器类型,并默认选中。因此,通常不需要更改此设置,使用默认选中的浏览器即可。

由于 IDEA 社区版没有内置对 Jakarta EE 应用部署和运行的支持,无法直接部署和运行 Jakarta EE Web 项目,因此需要使用第三方插件解决这个问题。目前开发者大都使用 Smart Tomcat 插件,以实现 IDEA 中对 Tomcat 的配置。

在 IDEA 的配置界面中选择"Plugins",进入插件管理界面,如图 3-73 所示。

📱 Welcome to IntelliJ IDEA		- 🗆 ×
	Marketplace Insta	alled ô
2023.3.2	Q- smart tomcat ×	
Projects	Search Resu Sort By: Relevance 🔻	Smart Tomcat
Customize	Smart Tomcat Install	
Plugins		Install 4.7.0
Learn		Reviews Additional Info
		Overview What's New

图 3-73 插件管理界面

在插件搜索文本框中输入 smart tomcat, IDEA 会在 Marketplace 中搜索此插件,成功 找到后显示其安装信息;如果没有安装 Smart Tomcat 插件,则会显示"Install"按钮,单击 此按钮,即可完成 Smart Tomcat 插件的安装。

安装 Smart Tomcat 插件后,会在配置界面最下方增加 Tomcat Server 配置项,如图 3-74 所示。

🖳 Set	tings		
Q•		Tomcat Server	
> App	earance & Behavior		Name: Apache Tomcat/10.1.17
Key	map	Apache Tomcat/10	
> Edit	tor		Version: 10.1.17.0
Plug	gins		Location: D:/apps/apache-tomcat-10.1.17
> Vers	sion Control 🛛 🕀		
> Buil	d, Execution, Deployment		
> Lang	guages & Frameworks		
> Too	ls		
Sett	tings Sync		
Adv	anced Settings		
Tom	ncat Server		

图 3-74 Tomcat 服务器配置界面

选择此配置项,即可进入 Tomcat 服务器配置界面。选择 Tomcat Server 配置项,会出现 Tomcat Server 配置列表。如果没有配置过,则列表为空。单击 + 号图标,选择 Tomcat 安装的目录,IDEA 自动显示服务器的如下信息。

(1) 服务器名称: Apache Tomcat/10.1.17。

(2) Version(版本): 10.1.17。

(3) Location(安装路径): D:/apps/apache-tomcat-10.1.17。

至此完成了 IDEA 社区版开发 Jakarta EE Web 应用的所有配置工作,接下来就可以创

建 Maven Web 项目,进行 Jakarta EE Web 项目的开发工作。

3.4.3 IDEA 开发 Jakarta EE Web 项目

IDEA 社区版不支持直接创建 Jakarta EE Web 项目,可以通过创建 Maven Web 项目的方式实现 Jakarta EE 项目的编程。

在 IDEA 启动界面中单击 New Project 按钮,弹出 New Project 对话框,如图 3-75 所示。

New Preiest	To create a ge	neral Maven project, go to the New Project page.
Empty Project	Name:	jakartaweb2024
	Location:	E-Vieseon/ JakartaEE/ws2024idea/jakartaweb2024
m Mayen Archetype	Location.	
Ra JavaEX		
		Create Git repository
Compass for Desister	JDK:	□ 17 Oracle OpenJDK version 17.0.9
Compose for Desktop		
	Catalog: 🕐	Internal ~ Manage catalogs
	Archetype: 🕐	he.maven.archetypes:maven-archetype-webapp > Add
	Version:	1.0 ~
	Additional Pro	perties
	> Advanced \$	settings
		Create Cancel

图 3-75 New Project 对话框

选择 Maven Archetype,提示输入 Maven 项目的如下基本信息。

(1) Name(名称): 输入项目的名称,这里输入 jakartaweb01。

(2) Location(位置):选择项目路径。

(3) JDK: 默认选择 IDEA 自动识别出来的 JDK,这里是 JDK 17。

(4) Catalog(Maven 模板目录): 默认选 Internal(内置模板)即可。

(5) Archetype(模板类型): 选择 org. apache. maven. archetypes: maven-archetype-webapp。

(6) Version(项目版本): 默认输入 1.0。

单击 Create 按钮,即开始创建 Maven Web 项目,创建后的项目和目录结构如图 3-76 所示。

从图 3-76 可以看到,创建的 Maven Web 项目没有包含编写 Java 代码的目录,需要开 发者自己创建。在项目的 src/main 下手动创建 java 目录,创建后的目录结构如图 3-77 所示。

在 Maven 项目的依赖配置文件 pom. xml 中的< build >标记下增加如下插件代码,用于

Project ~	¢			
✓ ₽ jakartaweb2024 E:\lesson\J				
> 🗀 .idea				
✓ □ src				
🗸 🗀 main				
[] resources				
🗸 🗋 webapp				
> 🗀 WEB-INF				
JSP index.jsp				
Ø.gitignore				
/m pom.xml				
> 🗈 External Libraries				
\equiv Scratches and Consoles				





图 3-77 创建 java 目录后的目录结构

指定 Java 编译版本和 WAR 打包方式:

```
< plugins >
    < plugins >
        <plugin >
            <groupId > org.apache.maven.plugins </groupId >
            <artifactId > maven - compiler - plugin </artifactId >
            <configuration >
            <source > 17 </source >
            <target > 17 </target >
            </configuration >
            </plugin >
            <groupId > org.apache.maven.plugins </groupId >
            <artifactId > maven - war - plugin </artifactId >
            <artifactId > maven > war - plugin </artifactId >
            </plugin >
            </plugin > 3.4.0 </version >
            </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
        </plugin >
```

编写 Jakarta EE Web 的组件,如 Servlet、JSP、Filter、Listener 等,需要引入配置的 Tomcat 的依赖库。在 Maven 的中央仓库,搜索 tomcat,如图 3-78 所示。

选择 Tomcat Catalina,并选择与配置的 Tomcat 对应的版本,这里选择 Tomcat 10.1.17,如 图 3-79 所示。

tomcat

X

102

Found 1806 results

Sort: relevance | popular | newest

I

a .	1. Tomcat Catalina	608 usages
×.	org.apache.tomcat » tomcat-catalina	Apache
	Tomcat Servlet Engine Core Classes and Standard implementations	
	Last Release on Jan 9, 2024	

Search

图 3-78 May	ven 中央仓库搜索	tomcat 的显示结果
------------	------------	--------------

10.1.18		Central	2	Jan 09, 2024
10.1.17		Central	4	Dec 12, 2023
10.1.16		Central	7	Nov 15, 2023
10.1.15	1 vulnerability	Central	9	Oct 16, 2023
10.1.14	1 vulnerability	Central	3	Oct 10, 2023

图 3-79 选择指定版本的 Tomcat 依赖库

单击 10.1.17 超链接后,显示 Maven 的依赖配置代码,如图 3-80 所示。

Maven	Gradle	Gradle (Short)	Gradle (Kotlin)	SBT	lvy	Grape	Leiningen	Buildr
<pre><!-- ht <depend</th--><td>tps://mvn ency> oupId>org tifactId> rsion>10. dency></td><td>repository.com/ apache.tomcat< tomcat-catalina 1.17</td><th>artifact/org.apa /groupId> </th><th>ache. to</th><th>mcat/t</th><td>tomcat-ca</td><td>talina></td><td></td></pre>	tps://mvn ency> oupId>org tifactId> rsion>10. dency>	repository.com/ apache.tomcat< tomcat-catalina 1.17	artifact/org.apa /groupId> 	ache. to	mcat/t	tomcat-ca	talina>	

图 3-80 Tomcat 10.1.17 Maven 依赖库

将 Maven 标签下的代码复制到项目的 pom. xml 文件中的< dependencies >标记下,则 pom. xml 文件中依赖库部分的显示内容如下:

```
< dependencies >
        <dependency>
        <groupId > junit </groupId >
        <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > junit </artifact/org.apache.tomcat/tomcat - catalina -->
            <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > junit </artifactId >
            <artifactId > tomcat - catalina </artifactId >
            <artifactId > junit </artifactId > jun
```

保存 pom. xml 文件, IDEA 的 Maven 引擎自动下载指定的依赖库。

使用 Maven 模板创建的 Web 项目,其默认的 Web 模块版本是 2.3,而开发 Jakarta EE Web 项目对应的 Web Module 版本是 6.0,因此需要将项目的 Web 配置文件/webapp/WEB-INF/web. xml 的内容修改如下:

```
<?xml version = "1.0" encoding = "UTF - 8"?>
< web - app xmlns:xsi = "http://www.w3.org/2001/XMLSchema - instance"xmlns = "https://jakarta.
ee/xml/ns/jakartaee"
xsi: schemaLocation = " https://jakarta. ee/xml/ns/jakartaee https://jakarta. ee/xml/ns/
jakartaee/web - app_6_0.xsd"
id = "WebApp_ID" version = "6.0">
    <display - name > javaweb01 </display - name >
    <welcome - file - list >
        <welcome - file > index.html </welcome - file >
        <welcome - file > index.html </welcome - file >
        <welcome - file > index.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        <welcome - file > default.html </welcome - file >
        </welcome - file > default.html </welcome - file >
        </welcome - file - list >
```

</web-app>

此文件内容在 Eclipse IDE 开发 Maven Web 项目中已经介绍过,这里不再赘述。

推荐将原有 webapp 目录下的 index. jsp 文件删除,重新创建 index. jsp。 index. jsp 文件中要包括 JSP 的指令代码和 HTML 代码,将此 index. jsp 的内容修改为如下代码:

```
<% @ page language = "java" contentType = "text/html; charset = UTF - 8"
pageEncoding = "UTF - 8" % >
<! DOCTYPE html >
< html >
< head >
< meta charset = "UTF - 8">
< title > Insert title here </title >
</head >
< body >
< hl >欢迎使用 Jakarta EE Web </hl >
</body >
</html >
```

编写好 JSP 文件后,需要将此 Web 项目部署到 Tomcat 服务器,并启动 Tomcat 服务器,以测试项目的运行。

IDEA 需要配置 Tomcat 服务器,在 Current File 配置下拉列表中选择 Edit Configurations, 如图 3-81 所示。



图 3-81 选择配置 Edit Configurations

编辑配置界面中默认的配置项目为空白,单击+图标,在配置项目列表中选择 Smart Tomcat,显示 Tomcat 配置界面,如图 3-82 所示。

E Run/Debug Configurations		×
+ —	<u>N</u> ame: Tomcat10.1.17 Configuration Logs	☐ Allow multiple instances □ Store as project file ⊚
	Tomcat server: Catalina base: Deployment directory: Use classpath of module: Context path: Server port: Admin port:	Apache Tomcat/10.1.17 Configure iaido\.SmartTomcat\jakartaweb2024\jakartaweb2024\jakartaweb2024 a/jakartaweb2024 a/jakartaweb2024/jakartaweb2024/src/main/webapp c jakartaweb2024 /jakartaweb2024 /jakartaweb2024 8080 SSL port: 8005
Edit configuration templates	VM options: Environment variables: Extra JVM classpath:	
?		▷ Run ∨ OK Cancel Apply

图 3-82 Tomcat 配置界面

在 Tomcat 配置界面中输入或选择如下信息。

(1) Name(配置名称): 这里输入 Tomcat10.1.17 或任意输入其他配置名称。

(2) Tomcat server(服务器):默认显示已经配置的 Apache Tomcat/10.1.17。

(3) Deployment directory(部署的目录):默认当前项目的 webapp 目录,即 Web 站点目录,不需要修改。

(4) Context path(站点名称): 部署运行的 Web 应用的站点名称,默认为项目名,不需要修改。

(5) Server port(服务器端口): Web 应用的服务端口号, IDEA 配置默认值 8080。

(6) Admin port: 后台管理站点的端口号,这里默认为 8005。如果与系统的其他服务 不冲突,此默认值可不用修改。

其他选项可以不用配置,单击 OK 按钮,完成此配置,在"配置"下拉列表内会显示刚刚 配置好的名称,如图 3-83 所示。

۳	\equiv J jakartaweb2024 $\scriptstyle{ imes}$	Version control $ \smallsetminus $	A Tomcat10.1.17 ~	⊳¢		×
	Project ~					Ļ
	✓ Cajakartaweb2024 E:\le	sson\JakartaEE\ws2024ide	ea\jal			т
ōo	> 🛅 .idea					
	> 🗀 src					

图 3-83 配置名称显示

在配置项目的右边出现"运行"按钮▶,单击此按钮,会启动 Tomcat 服务器,并部署开发的 Maven Web 项目。Tomcat 服务器启动完成后,会在控制台窗口显示图 3-84 所示的启动日志信息。

► To	ncat Localhost Log
······································	
[main]	org.apacne.catalina.core.AprLitecyCleListener.initializeSSL UpenSSL版切初始化 [UpenSSL 3.0.
[main]	org.apache.coyote.AbstractProtocol.init 初始化协议处理器 ["http-nio-8080"]
[main]	org.apache.catalina.startup.Catalina.load 服务器在[1594]毫秒内初始化
[main]	org.apache.catalina.core.StandardService.startInternal 正在启动服务[Catalina]
[main]	org.apache.catalina.core.StandardEngine.startInternal 正在启动 Servlet 引擎: [Apache Tomcat/
[main]	org.apache.catalina.startup.HostConfig.deployDescriptor 正在部署部署描述符[C:\Users\haido\.Sn

图 3-84 启动 Maven Web 的日志信息

在日志的最后,给出了 Web 项目的访问地址,即 http://localhost:8080/jakartaweb2024。 启动浏览器,如 Google 的 Chrome,访问该地址,会显示 Web 站点的主页(index.jsp)内容,如 图 3-85 所示。



图 3-85 浏览器访问部署的 Web 站点默认主页内容

现在 Jakarta EE 的 Web 组件之一 JSP 可以正常工作,下面再增加一个 Web 组件 Servlet。在 src/main/java 目录下创建包 com. city. oa. servlet,在此包下创建 Servlet 类: Test01, IDEA 的工作台显示如图 3-86 所示。

Cijakartaweb2024 E:\lesson\JakartaEE\ws2024idea\ja	1	<pre>package com.city.oa.servlet;</pre>
> 🗋 .idea	2	<pre>import jakarta.servlet.ServletException;</pre>
∽ □ src	3	import jakarta servlet appotation WebServlet.
main m	6	import jukarta consist the Utto Consist
✓ □ java	4	import jakarta.servlet.nttp.HttpServlet;
 i com.city.oa 	5	<pre>import jakarta.servlet.http.HttpServletRequest;</pre>
> 🗈 config	6	<pre>import jakarta.servlet.http.HttpServletResponse;</pre>
> i controller.rest	7	import jova in TOExcontion:
> 🖻 factory	/	Import Java.io.ioException,
> 🖻 mapper	8	import java.io.PrintWriter;
> i model	9	
> i result	10	@WebServlet("/test01") no usages
> i service		sublic class Test01 sutends UttrCenulet (
✓ i servlet	11	public class lestor extends hitpserviet {
 ✓ is servlet © Test01 	11 12 ©†@	public trass restor extends httpservlet { public void doGet(HttpServletRequest request,
 © servlet © Test01 > © starter 	11 12 ©[†]@ 13	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html");</pre>
 Fervlet © Test01 © springMVCWebStarter © SpringMVCWebStarter 	11 12 ©î@ 13 14	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html"); response.setCharacterEncoding("UIE-8"); </pre>
 Fervlet © Test01 © springMVCWebStarter C resources 	11 12 © [†] @ 13 14	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html"); response.setCharacterEncoding("UTF-8"); PrintWrites outpersonal putWrites(); } }</pre>
 Servlet Test01 Starter SpringMVCWebStarter Resources Newbapp 	11 12 © [†] @ 13 14 15	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html"); response.setCharacterEncoding("UTF-8"); PrintWriter out=response.getWriter();</pre>
 is servlet is tarter is starter is springMVCWebStarter is resources iwebapp is test 	11 12 © [†] @ 13 14 15 16	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html"); response.setCharacterEncoding("UTF-8"); PrintWriter out=response.getWriter(); out.println("<h1>测试Servlet</h1>");</pre>
 is servlet is servlet is tarter is springMVCWebStarter is resources iwebapp it target 	11 12 © [†] @ 13 14 15 16 17	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html"); response.setCharacterEncoding("UTF-8"); PrintWriter out=response.getWriter(); out.println("<h1>测试Servlet</h1>"); out.flush(); } }</pre>
 iservlet 	11 12 © ® 13 14 15 16 17 18	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html"); response.setCharacterEncoding("UTF-8"); PrintWriter out=response.getWriter(); out.println("<h1>测试Servlet</h1>"); out.flush(); out.close(); } }</pre>
 Servlet C Test01 Starter SpringMVCWebStarter resources webapp test target g.gitignore m pom.xml Storten Liberaire 	11 12 © @ 13 14 15 16 17 18	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html"); response.setCharacterEncoding("UTF-8"); PrintWriter out=response.getWriter(); out.println("<h1>测试Servlet</h1>"); out.flush(); out.close(); } }</pre>
 Servlet C Test01 C SpringMVCWebStarter C resources Webapp Test Target ⊘.gitignore m pom.xml External Libraries Services 	11 12 © [†] @ 13 14 15 16 17 18 19	<pre>public class restor extends httpservlet { public void doGet(HttpServletRequest request, response.setContentType("text/html"); response.setCharacterEncoding("UTF-8"); PrintWriter out=response.getWriter(); out.println("<h1>测试Servlet</h1>"); out.flush(); out.close(); } </pre>



Servlet Test01 的 Java 代码如程序 3-1 所示。

```
程序 3-1 Test01Java 测试 Servlet 的 Java 代码。
```

```
package com. city. oa. servlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta. servlet. http. HttpServletResponse;
import java. io. IOException;
import java.io.PrintWriter;
@WebServlet("/test001")
public class Test001 extends HttpServlet {
     public void doGet (HttpServletRequest request, HttpServletResponse response) throws
IOException, ServletException {
        response.setContentType("text/html");
        response.setCharacterEncoding("UTF - 8");
        PrintWriter out = response.getWriter();
        out.println("< h1 >测试 Servlet </h1 >");
        out.flush();
        out.close();
```

Jakarta EE Servlet 与传统的 Java EE 的代码基本相同,不同的是所有 Servlet API 的接口和类的包名都以 jakarta 开头,而不再以 javax 开头; 传统的 Java EE Web 的请求对象类型是 javax. servlet. http. HttpServletRequest,而现在 Jakarta EE Web 的请求对象类型是 jakarta. servlet. http. HttpServletRequest,这一点之前使用 Java EE 的开发者尤其需要注意。

编写好 Servlet 后,重新启动 Tomcat 服务器,在浏览器的地址栏中输入此 Servlet 的请求地址(http://localhost:8080/jakartaeeweb01/test01),Tomcat 会运行此 Servlet,显示 图 3-87 所示的响应内容。

← → C ① localhost:8080/jakartaweb2024/
 □ ※用 □ Vue □ IDE □ Python □ Spring □ 鴻蒙

欢迎使用Jakarta EE Web

图 3-87 IDEA 启动 Tomcat 并运行 Servlet 的响应结果

至此,可以使用 IDEA 社区版进行 Jakarta EE Web 项目的编程和部署,不必再付费购 买 IDEA 旗舰版。

简答题

1. 市场上流行的 Jakarta EE 服务器产品主要有哪些? 请比较它们各自的优缺点。

2. 主流的 Jakarta EE 应用开发工具有哪些?