



Tomcat 架构解析

刘光瑞 著
刘 伟 审校

- 基于Tomcat最新版本全面解析应用服务器架构
- 涵盖Tomcat所有组件的详细配置、使用与优化
- 知识讲解广度与深度结合，系统性与完整性兼备



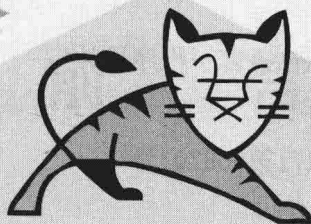
中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵原创



Tomcat 架构解析

刘光瑞 著
刘 伟 审校

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

Tomcat架构解析 / 刘光瑞著. — 北京 : 人民邮电出版社, 2017. 5
(图灵原创)
ISBN 978-7-115-45369-3

I. ①T… II. ①刘… III. ①JAVA语言—程序设计
IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第068701号

内 容 提 要

本书全面介绍了 Tomcat 的架构、各组件的实现方案以及使用方式, 包括 Tomcat 的基础组件架构以及工作原理, Tomcat 各组件的实现方案、使用方式以及详细配置说明, Tomcat 与 Web 服务器集成以及性能优化, Tomcat 部分扩展特性介绍等, 使读者全面了解应用服务器的架构以及工作原理, 学习 Tomcat 的使用、优化以及详细配置。

本书内容通俗易懂, 由浅入深, 适合基于 Java 平台的软件架构师、软件开发工程师及系统运维人员阅读使用。

-
- ◆ 著 刘光瑞
审 校 刘 伟
责任编辑 张 霞
责任印制 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京市昌平百善印刷厂印刷
 - ◆ 开本: 800×1000 1/16
印张: 23.75
字数: 561千字 2017年5月第1版
印数: 1-3 000册 2017年5月北京第1次印刷
-

定价: 79.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

前 言

Apache Tomcat 作为著名的 Servlet 容器实现以及轻量级 Java 应用服务器，是 Apache 软件基金会的顶级项目。它开源、轻量，与 JBoss、Weblogic 等企业级应用服务器相比，占用资源小，扩展性好，深受 Java 研发人员喜爱，尤其是在当前主流的基于 POJO（Without EJB）的轻量级编程方式下构建应用系统。除此之外，Apache Tomcat 还可以很容易与 Apache Http Server、Nginx 等知名的 Web 服务器集成，以实现负载均衡和集群化部署。所有这些特性都使得 Tomcat 被广泛用于开发、测试环境，甚至大规模、高并发的互联网产品部署。

由于工作的关系，笔者很早便开始接触并使用 Apache Tomcat。随着工作内容的变化，对 Tomcat 的了解及研究的广度和深度也不断变化，相信这也是大多数技术人员的学习经历。对于开发人员、测试人员或者运维人员而言，关注较多的是 Tomcat 的配置使用方式以及进一步的性能优化，而架构师则会从架构层面审视 Tomcat，研究它的各种组件设计方案、生命周期管理、运行机制等。只有这样，才可以充分利用 Tomcat 提供的特性，使得应用系统以最高效的方式部署运行。除此之外，Tomcat 的架构设计也会不同程度地为应用系统的基础技术架构提供不错的借鉴意义。

在几年之前，笔者便有计划将 Tomcat 各组件的基本概念、规范、架构设计方案以及详细配置使用方式整理成书。这不仅可以使自身相关的知识得到系统性的梳理，同时也希望能够给正在使用或者即将使用 Tomcat 的人以及对 Tomcat 架构感兴趣的技术人员带来些许帮助，这也是我所乐见的。但是由于各种原因，本书写作一直断断续续，持续近两年，而 Tomcat 也由最初的 7.x 版本更新到了 8.5.x 版本（9.x 的里程碑版本也已经发布了数版）。在最新版本中，Tomcat 增加了很多重要特性，如对 HTTP/2 和 NIO2 的支持等。为了确保本书所讲内容与 Tomcat 最新版本的架构匹配，本书进行了两次版本变更。对于版本变更带来的一些架构变更及新支持的重要特性，本书也不同程度地做了讲解。

总之，笔者希望这本书能够真实地、系统性地讲解 Tomcat 的最新架构。它涉及了 Tomcat 的代码设计，却不仅仅是围绕代码进行分析。它涉及了 Tomcat 的使用方式，却不仅仅是对其配置方式进行简单说明。对于每个技术点相关的规范、方案的考量及隐含问题、如何使用等，本书将进行系统化的讲解。希望读者读完这本书，可以清晰地知道一款 Servlet 容器所包含的组件、涉及的规范以及实现方案，也知道如何深度地定制及优化 Tomcat。

本书内容及读者对象

本书尝试从以下几个方面来讲解 Apache Tomcat。

- ❑ **基本设计**：主要讲述 Tomcat 的核心接口及概念，并阐述 Tomcat 的设计理念。
- ❑ **架构及工作原理**：主要讲述 Tomcat 的整体架构，以及各模块如何密切协调来完成应用服务器的相关工作。
- ❑ **各个模块的特性及使用方式**：详细讲述各模块的特性，以及如何使用该特性实现高质量的部署架构。每个方面均融合了相关的设计理念及架构知识的讲解，以使读者更容易从中间件产品架构的角度审视 Tomcat。

既然本书的主要内容是讲解 Tomcat，那么读者需要是熟悉 Java 语言的从业人员，而且对 Java Web 应用的开发有初步的概念，因此本书适用于以下读者。

- ❑ Java Web 应用开发者。
- ❑ 应用服务器及相关中间件开发者。
- ❑ 系统运维人员。
- ❑ 系统架构师。

章节介绍

本书的章节划分如下，不同的读者可以根据自己的背景和兴趣挑选相应的章节进行学习，而不必逐章阅读。

第 1 章照例对 Tomcat 进行简单的介绍，主要包括安装、启动、部署应用以及 Tomcat 目录结构。如果你是一位初级开发者，阅读完本章，你会对 Tomcat 有最基本的认识，基本可以覆盖应用开发过程中能够涉及的各种问题。如果你已经可以熟练部署使用 Tomcat，那么完全可以跳过这一章。

第 2 章主要介绍了 Tomcat 容器、链接器各组件的基本概念，阅读这一章有助于初步了解 Tomcat 的架构设计。

第 3~5 章、第 8~9 章对 Tomcat 架构及相关模块进行了深入的讲解。如果你希望详细了解 Tomcat 架构的各个方面，这部分应该是你的侧重点。如果你是一名中间件设计人员或者系统架构师，可以从中发现许多值得学习和借鉴的地方，包括中间件设计、系统部署架构的构建以及安全管理等多个方面。

第 6 章和第 7 章主要介绍了 Tomcat 的管理以及与 Web 服务器的集成，包括 Web 控制台、JMX、Ant 这 3 种管理方式，以及 Tomcat 与当前使用最广泛的 Apache HTTP Server、Nginx 等 Web 服务器的集成，以满足不同的运维场景需要。如果你是一名运维人员，建议你仔细阅读这两章内容，因为这有助于我们了解 Tomcat 相关的系统访问监控、集群部署以及负载均衡。

第 10 章侧重于系统的性能优化，讲解如何优化 Tomcat，以便提高请求处理速度，增加系统并发访问量。

第 11 章主要介绍了 Tomcat 提供的一些附加功能，如嵌入式启动、JNDI、Comet 和 WebSocket 实现。如果你对嵌入式启动 Tomcat 或者基于 Tomcat 进行服务器推送感兴趣，那么这一章不容错过。

最后在附录中详细整理了 Tomcat 各组件支持的配置属性，便于读者在使用过程中进行深度定制。

本书试图为读者完整地、系统化地讲解 Tomcat，甚至会在开篇简要讲解 Tomcat 的历史及现状。所有这一切，都是基于希望读者“知其然，更要知其所以然”的目的出发的。当然，作为读者，你完全可以选择自己感兴趣的部分进行阅读。

意见反馈

为了将相关知识讲述完整，笔者除了阅读及参考相关文献外，还仔细阅读并分析了 Tomcat 的源代码。但是即便如此，由于笔者能力有限，所知所写难免有疏漏和错误之处，欢迎各位读者不吝批评指正。

如果你有任何问题或者批评建议，可以发送邮件到 kunrey@163.com 与我联系。

致谢

首先，应该感谢 Sun 以及 Apache 创造了 Tomcat 这款知名的轻量级 Servlet 容器，可以让众多 Java 从业者免费使用并研究学习。

其次，感谢图灵公司的编辑张霞，正是在她的不断努力下，这本书才可以顺利出版并最终呈献给大家。在本书的编辑过程中，她细心校正每处纰漏和错误，并提供了很多非常好的改进建议。

同时，感谢四达传媒集团大视频事业部副总经理刘伟，作为前公司的领导、同事以及朋友，在百忙中抽出时间，担任本书的技术审校，他的宝贵意见和建议使我收获良多。

再次，由衷感谢每一位为本书出版付出了努力的朋友，本书的出版离不开你们的付出，感谢你们。

最后，感谢我的妻子张静以及可爱的女儿刘于飞，这本书的写作同时也伴随着对家庭付出的缺失，谢谢你们的理解和支持。

刘光瑞

2017-02-01

目 录

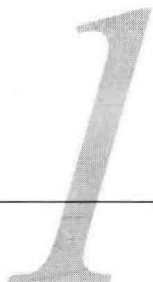
第 1 章 Tomcat 介绍.....	1	第 3 章 Catalina.....	38
1.1 简介.....	1	3.1 什么是 Catalina.....	38
1.1.1 Tomcat 历史.....	1	3.2 Digester.....	39
1.1.2 Tomcat 许可.....	2	3.2.1 对象栈.....	40
1.2 安装和启动.....	3	3.2.2 匹配模式.....	41
1.2.1 Tomcat 下载与安装.....	3	3.2.3 处理规则.....	41
1.2.2 Tomcat 启动.....	4	3.2.4 示例程序.....	42
1.2.3 IDE 启动.....	5	3.3 创建 Server.....	44
1.2.4 嵌入式启动.....	10	3.3.1 Server 的解析.....	45
1.2.5 Debug 启动.....	11	3.3.2 Engine 的解析.....	48
1.3 Tomcat 目录结构.....	12	3.3.3 Host 的解析.....	49
1.4 Tomcat 8.5 之后的新特性.....	12	3.3.4 Context 的解析.....	50
1.5 小结.....	13	3.4 Web 应用加载.....	53
第 2 章 Tomcat 总体架构.....	14	3.4.1 StandardHost.....	54
2.1 总体设计.....	14	3.4.2 HostConfig.....	56
2.1.1 Server.....	14	3.4.3 StandardContext.....	60
2.1.2 Connector 和 Container.....	15	3.4.4 ContextConfig.....	64
2.1.3 Container 设计.....	17	3.4.5 StandardWrapper.....	69
2.1.4 Lifecycle.....	20	3.4.6 Context 命名规则.....	69
2.1.5 Pipeline 和 Valve.....	23	3.5 Web 请求处理.....	71
2.1.6 Connector 设计.....	24	3.5.1 总体过程.....	71
2.1.7 Executor.....	26	3.5.2 请求映射.....	72
2.1.8 Bootstrap 和 Catalina.....	28	3.5.3 Catalina 请求处理.....	80
2.2 Tomcat 启动.....	30	3.6 DefaultServlet 和 JspServlet.....	81
2.3 请求处理.....	31	3.6.1 DefaultServlet.....	82
2.4 类加载器.....	33	3.6.2 JspServlet.....	84
2.4.1 J2SE 标准类加载器.....	33	3.7 小结.....	85
2.4.2 Tomcat 加载器.....	34	第 4 章 Coyote.....	86
2.4.3 Web 应用类加载器.....	36	4.1 什么是 Coyote.....	86
2.5 小结.....	37	4.2 Web 请求处理.....	88

4.2.1 主要概念	88	6.3.5 Filter 定义及映射	177
4.2.2 请求处理	90	6.3.6 MIME 类型映射	178
4.2.3 协议升级	92	6.3.7 欢迎文件列表	178
4.3 HTTP	94	6.3.8 错误页面	179
4.3.1 基础知识	94	6.3.9 本地化及编码映射	179
4.3.2 配置方式	95	6.3.10 安全配置	179
4.4 AJP	97	6.3.11 JNDI 配置	180
4.4.1 基础知识	97	6.3.12 其他	183
4.4.2 Web 服务器组件	103	6.4 Web 应用过滤器	185
4.4.3 配置方式	106	6.4.1 CorsFilter	186
4.5 HTTP/2.0	106	6.4.2 CsrfPreventionFilter	188
4.5.1 基础知识	106	6.4.3 ExpiresFilter	188
4.5.2 配置方式	110	6.4.4 FailedRequestFilter	190
4.6 I/O	111	6.4.5 RemoteAddrFilter	191
4.6.1 BIO	112	6.4.6 RemoteHostFilter	191
4.6.2 NIO	115	6.4.7 RemoteIpFilter	192
4.6.3 NIO2	121	6.4.8 RequestDumperFilter	195
4.6.4 APR	130	6.4.9 SetCharacterEncodingFilter	195
4.7 小结	134	6.4.10 WebdavFixFilter	196
第 5 章 Jasper	136	6.5 Tomcat 管理	196
5.1 Jasper 简介	136	6.5.1 host-manager	196
5.2 JSP 编译方式	137	6.5.2 manager	197
5.2.1 运行时编译	137	6.5.3 管理命令行	199
5.2.2 预编译	141	6.5.4 Ant 任务	200
5.3 JSP 编译原理	144	6.5.5 JMX	201
5.4 小结	151	6.6 小结	201
第 6 章 Tomcat 配置管理	152	第 7 章 Web 服务器集成	202
6.1 JVM 配置	152	7.1 Web 服务器与应用服务器的区别	202
6.1.1 JVM 配置选项	152	7.2 集成应用场景	205
6.1.2 系统属性	153	7.3 与 Apache HTTP Server 集成	206
6.2 服务器配置	158	7.3.1 Windows 环境安装	206
6.2.1 catalina.properties	158	7.3.2 Linux 环境安装	207
6.2.2 server.xml	159	7.3.3 mod_jk	209
6.2.3 context.xml	172	7.3.4 mod_proxy_ajp	229
6.3 Web 应用配置	172	7.4 与 Ngnix 集成	235
6.3.1 ServletContext 初始化参数	173	7.4.1 Ngnix 简介	235
6.3.2 会话配置	174	7.4.2 Windows 环境安装	237
6.3.3 Servlet 声明及映射	175	7.4.3 Linux 环境安装	237
6.3.4 应用生命周期监听器	177	7.4.4 Tomcat 集成	241

7.5 与 IIS 集成	243	9.5 小结	279
7.6 小结	243	第 10 章 Tomcat 性能调优	280
第 8 章 Tomcat 集群	244	10.1 Tomcat 性能测试及诊断	281
8.1 Tomcat 集群介绍	244	10.1.1 常见测试方式	281
8.1.1 Tomcat 集群基础	246	10.1.2 性能测试工具	282
8.1.2 Apache Tribes	247	10.1.3 数据采集及分析	290
8.1.3 Tomcat 集群组件实现	253	10.2 Tomcat 性能优化	303
8.2 集群配置	254	10.2.1 JVM 优化	304
8.3 会话同步	255	10.2.2 Tomcat 配置	308
8.3.1 DeltaManager	256	10.3 应用性能优化建议	311
8.3.2 BackupManager	257	10.4 小结	312
8.3.3 替代方案	257	第 11 章 Tomcat 附加功能	313
8.4 集群部署	258	11.1 Tomcat 的嵌入式启动	313
8.4.1 实现原理	258	11.1.1 为什么需要嵌入式启动	313
8.4.2 配置方式	259	11.1.2 嵌入式启动 Tomcat	314
8.5 小结	259	11.1.3 嵌入式启动服务器	316
第 9 章 Tomcat 安全	260	11.2 Tomcat 中的 JNDI	317
9.1 配置安全	260	11.2.1 什么是 JNDI	317
9.1.1 安装部署问题	260	11.2.2 Tomcat 中的 JNDI	318
9.1.2 server.xml 配置	261	11.3 Comet 和 WebSocket	320
9.2 应用安全	263	11.3.1 什么是 Comet	321
9.2.1 Realm	264	11.3.2 Tomcat 的 Comet 实现	322
9.2.2 HttpServletRequest	265	11.3.3 什么是 WebSocket	326
9.2.3 Authenticator	266	11.3.4 Tomcat 的 WebSocket 实现	328
9.3 传输安全 (SSL)	267	11.4 小结	335
9.4 Java 安全策略	273	附录 server.xml 配置	336
9.4.1 简介	273	结束语	370
9.4.2 catalina.policy	275		

第 1 章

Tomcat介绍



Tomcat是全世界最著名的基于Java语言的轻量级应用服务器，是一款完全开源免费的Servlet容器实现。同时，它支持HTML、JS等静态资源的处理，因此又可以作为轻量级Web服务器使用。作为本书的开篇，本章将简单介绍Tomcat的发展历程以及基本的安装使用，以便读者对Tomcat有个初步的印象。主要包括以下几个方面的内容。

- Tomcat的历史及许可。
- Tomcat的安装、启动和应用部署。
- Tomcat的目录结构。
- Tomcat最新版本（8.5/9.0）的特性。

1.1 简介

本节主要介绍了Tomcat的历史以及主要版本的发展情况，以及每个版本对Servlet规范的支持，以便读者能够很好地了解Tomcat的过去以及现状。此外，本节还介绍了Tomcat的授权许可，如果你计划基于Tomcat进行定制化开发并发布自己的服务器中间件产品，那么了解它的许可方式是非常有必要的。

1.1.1 Tomcat 历史

Tomcat最初由Sun公司的软件架构师James Duncan Davidson开发，名称为“JavaWebServer”，该项目作为Servlet容器的参考实现，以展示Servlet容器相关技术。随后在Davidson的帮助下，该项目于1999年与Apache软件基金会旗下的JServ项目合并，即为现在的Tomcat。

Tomcat的第一个版本(3.x)发布于1999年，该版本基本源自Sun公司贡献的代码，实现了Servlet 2.2和JSP 1.1规范。2001年，Tomcat发布了4.0版本，作为里程碑式的版本，Tomcat完全重新设计了其架构，并实现了Servlet 2.3和JSP 1.2规范。

发展至今，作为Sun相关规范的参考实现，Tomcat已经成为一款成熟的Servlet容器产品，并作为JBoss等应用服务器产品内嵌的Servlet容器（最新的JBoss版本已改为Undertow）。Tomcat不仅

广泛用于开发及测试环境，更大量应用于生产环境当中。事实证明，简单如单独服务器、主备部署，复杂至大型的集群架构，Tomcat均可以实现有效的支撑。

当前Tomcat存在5个主要版本，分别支持不同版本的规范，其对规范及JDK的版本支持（截止编写本书为止）如表1-1所示。

表1-1 Tomcat版本对照表

版本（更新） 规范及JDK	6.x（6.0.47）	7.x（7.0.72）	8.x（8.0.38）	8.5.x（8.5.6）	9.x （9.0.0.M11）
JDK	≥ 5.0	≥ 6.0	≥ 7.0	≥ 7.0	≥ 8.0
Servlet	2.5	3.0	3.1	3.1	4.0
JSP	2.1	2.2	2.3	2.3	2.3
EL	2.1	2.2	3.0	3.0	3.0
WebSocket	N/A	1.1	1.1	1.1	1.1

※如果在7.x版本下使用WebSocket，JDK最低版本为7.0。

注意 Tomcat最后两个版本并不是顺序发布的。2015年11月，Tomcat发布了重要的里程碑版本9.0（目前仍为alpha版本）。在该版本中，Tomcat依赖最新的JDK8，使用了JDK8最新的语法特性，支持最新的Servlet规范（4.0），并增加了对HTTP/2的支持。2016年3月，考虑到9.0版本的变更范围以及发布进度，Tomcat在8.0的基础上又发布了一个中间版本——8.5，它的主体架构延续自8.0，同时又实现了部分9.0的主要特性，以使用来取代8.0版本。

我们可以根据具体应用系统的部署环境要求，选择合适的版本使用。考虑到编写本书时9.0仍为alpha版本，因此我们选择基于8.5.x的最新版本8.5.6进行讲解；同时，在每一章节我们会补充9.0最新的变更内容。

1.1.2 Tomcat 许可

Tomcat以Apache License许可的方式进行发布，具体许可内容参见<http://apache.org/licenses/LICENSE-2.0>。主要概括如下。

- Tomcat完全免费，可用于任何商业或者非商业产品，而不必支付任何费用。
- 修改Tomcat并发布变更版本的用户不必公开修改部分的源代码。
- 修改Tomcat的用户不必将其变更捐献给Apache软件基金会。

此外，还需要了解如下几点限制。

- 所有针对Tomcat源代码或二进制文件的重新发布均须包含Apache许可。
- 任何重新发布所包含的资料必须经过Apache软件基金会批准。
- 无Apache基金会的许可，派生自Tomcat源代码的产品不能命名为“Tomcat”“Jakarta”“Apache”。

1.2 安装和启动

本节主要介绍了Tomcat的下载、安装以及各种场景下的启动方式。你可以根据实际情况选择不同格式的发布包安装使用。

1.2.1 Tomcat 下载与安装

在安装Tomcat之前,请确保当前系统已安装了不低于7.0版本的JDK/JRE,并添加了JAVA_HOME系统环境变量以指向JDK/JRE的安装目录。

你可以从<http://tomcat.apache.org/download-80.cgi#8.5.6>获取Tomcat的安装文件。对于Windows系统,Apache提供了ZIP和Windows安装文件两种发布形式。对于Linux系统,Apache提供了tar.gz的发布包。

下面我们对Tomcat主要的安装包进行简单说明,以便你可以选择合适的包下载使用。

- ❑ `apache-tomcat-8.5.6.zip`: Tomcat基础发布包,它不包含Windows服务相关的批处理脚本以及Windows下的APR本地库。
- ❑ `apache-tomcat-8.5.6.tar.gz`: 与ZIP包相同,只是压缩格式不同。
- ❑ `apache-tomcat-8.5.6.exe`: Windows可执行的安装包。包含功能与ZIP基本一致,但是删除了部分命令行脚本,适用于Windows快捷键以及系统服务的形式启动。
- ❑ `apache-tomcat-8.5.6-windows-x86.zip`: 32位Windows发布包,包含Windows服务相关的批处理脚本以及与32位JVM配合使用的APR本地库(操作系统为Windows 32/64位)。
- ❑ `apache-tomcat-8.5.6-windows-x64.zip`: 64位Windows发布包,包含Windows服务相关的批处理脚本以及与64位JVM配合使用的APR本地库(操作系统为Windows 64位)。

1. 在Windows上安装Tomcat

在Windows系统安装Tomcat最简单的方式莫过于下载ZIP安装包(`apache-tomcat-8.5.6.zip`),并将其解压到系统任意目录即可(注意:安装路径最好不要包含空格)。

除此之外,还可以通过Windows安装文件(`apache-tomcat-8.5.6.exe`)的方式进行安装。直接下载安装文件,双击并按照提示一步步操作即可完成安装。

通过exe方式安装的好处是,可以按照提示进行Tomcat相关配置,如Tomcat管理程序的用户名和密码、占用端口等。

此外,采用exe方式进行安装,Tomcat会默认创建Windows服务,用于启动/停止Tomcat、设置Tomcat开机启动。如果采用ZIP的方式进行安装,就需要通过Windows服务管理Tomcat,你可以运行`bin/service.bat`批处理脚本进行添加,如下所示。

```
C:\apache-tomcat-8.5.6\bin>service.bat install
```

安装完成后,最好将CATALINA_HOME添加到系统环境变量,并将\$CATALINA_HOME/bin添加到Path中,CATALINA_HOME指向Tomcat的安装目录。

2. 在Linux上安装Tomcat

在Linux上安装Tomcat，可下载tar.gz形式的发布包，执行如下命令将其解压即可：

```
liuguangrui@ubuntu:~$ tar zxvf apache-tomcat-8.5.6.tar.gz
```

与Windows安装相同，最好也将CATALINA_HOME导入到系统变量。笔者的安装目录为/home/liuguangrui/apache-tomcat-8.5.6，故执行命令如下：

```
liuguangrui@ubuntu:~$ CATALINA_HOME=/home/liuguangrui/apache-tomcat-8.5.6
liuguangrui@ubuntu:~$ export CATALINA_HOME
```

1.2.2 Tomcat 启动

Tomcat的启动非常简单，接下来我们就来看一下在不同操作系统下的启动方式。

1. 在Windows上启动Tomcat

在Windows环境下，启动Tomcat有多种方式。

- ❑ 在命令行下，打开\$CATALINA_HOME/bin目录，输入“startup.bat”运行Tomcat。
- ❑ 如果采用exe方式安装，可通过点击开始菜单中的快捷方式，打开Tomcat服务管理工具，启动Tomcat服务。
- ❑ 如果添加了Windows服务，可进入“控制面板”→“管理工具”→“服务”，找到“Apache Tomcat 8.5 Tomcat8”服务，双击并“启动”。

通过编辑\$CATALINA_HOME/bin/catalina.bat文件，可以修改Tomcat的启动配置。例如在文件中添加以下内容以调整Tomcat的内存分配：

```
Set JAVA_OPTS=-server -Xms1024m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=512m
```

2. 在Linux上启动Tomcat

在Linux环境下（以Ubuntu为例），执行如下命令启动Tomcat：

```
liuguangrui@ubuntu:~$ $CATALINA_HOME/bin/startup.sh
```

同样，我们可以通过编辑\$CATALINA_HOME/bin/catalina.sh，修改Tomcat启动配置：

```
JAVA_OPTS="-server -Xms1024m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=512m"
```

3. 测试启动

当在控制台中输出如下日志时，表示Tomcat已经启动完成。

```
信息: Server startup in 4535 ms
```

此时，在浏览器中输入http://127.0.0.1:8080，如果显示如图1-1所示的页面，即表示启动成功。需要注意的是，如果修改了Tomcat的默认端口号，上述地址需进行相应调整。

4. 应用部署

在独立启动的方式下将Web应用部署到Tomcat非常简单，只需要将应用包复制到

\$CATALINA_HOME/webapps下，并重启Tomcat即可。当然还可以通过Tomcat管理工具部署，此部分我们将在后续章节讲解。

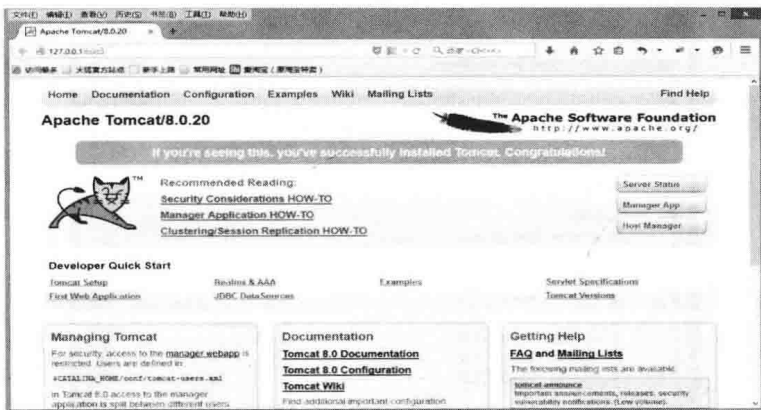


图1-1 Tomcat启动

1.2.3 IDE 启动

在项目开发过程中，我们经常需要将服务器集成到IDE中，以便直接将工程发布到服务器环境并启动，从而进行系统调试。

下面将详细展示如何将Tomcat集成到IDE（以Eclipse为例）并启动。

第一步，在Eclipse中添加Tomcat的运行环境。选择Windows → Preferences，弹出Preferences对话框，如图1-2所示。

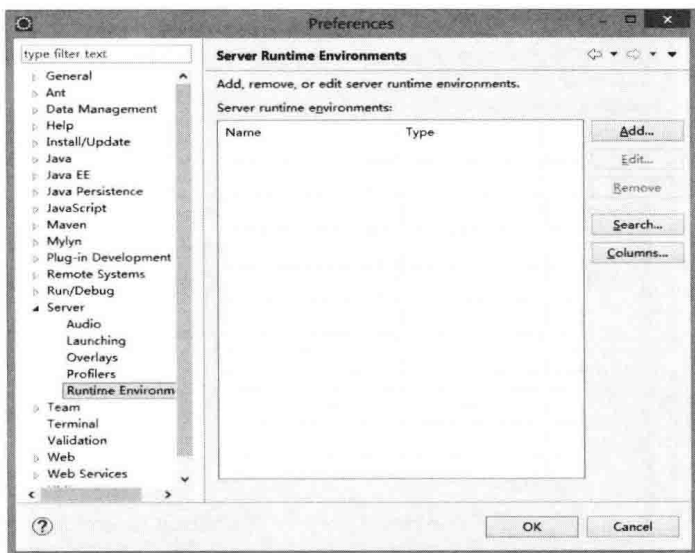


图1-2 Preferences对话框

选择Server → Runtime Environments，显示当前已经安装的服务器运行环境。点击“Add”，弹出添加服务器运行环境对话框，如图1-3所示。

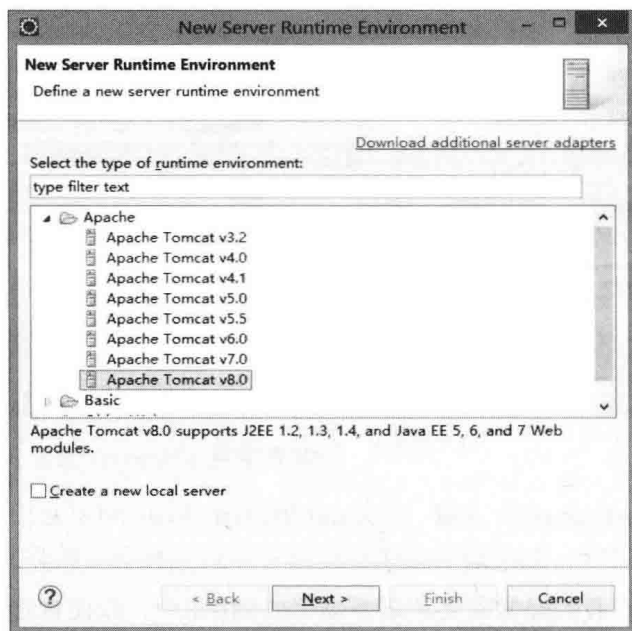


图1-3 添加服务器运行环境对话框

选择Apache → Apache Tomcat v8.0，点击“Next”，显示如图1-4所示。

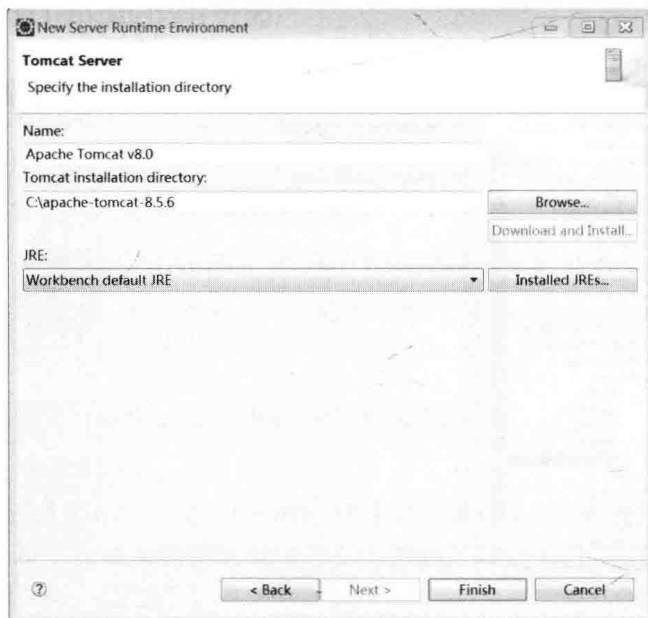


图1-4 新建服务器对话框

点击“Browse”选择Tomcat的安装根目录，点击“Finish”完成添加服务器运行环境。

第二步，新建服务器配置。选择File → New → Other弹出新建对话框，然后选择Server → Server，如图1-5所示。

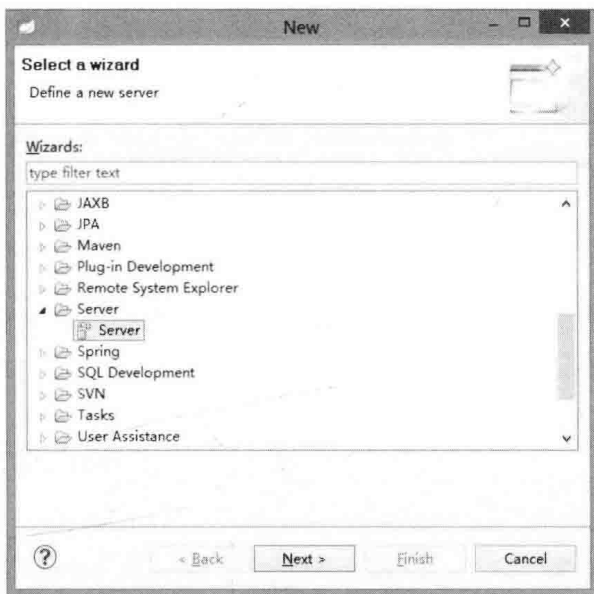


图1-5 新建服务器配置对话框

点击“Next”，如图1-6所示。

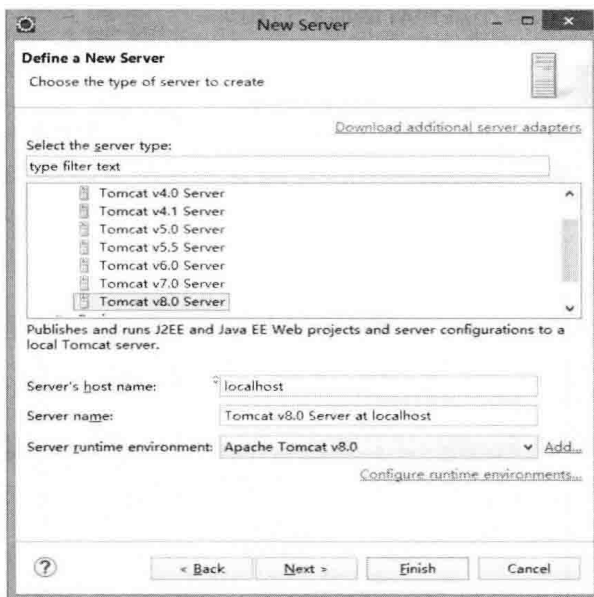


图1-6 新建服务器配置对话框