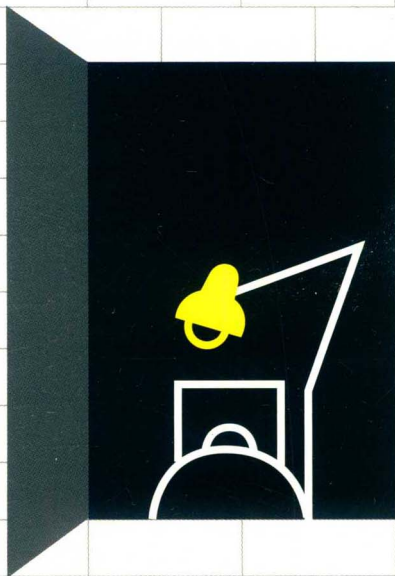




RocketMQ社区布道师和技术专家撰写

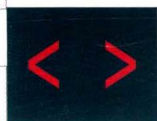
RocketMQ创始人
Alibaba Messaging开源技术负责人
Linux OpenMessaging创始人

作序推荐

RocketMQ 技术内幕

RocketMQ架构设计与实现原理

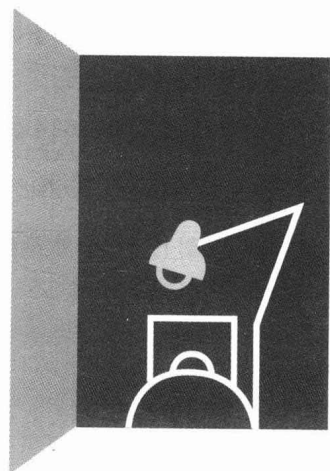
丁威 周继锋◎著



RocketMQ Internals: Architecture, Design and Principle



机械工业出版社
China Machine Press



RocketMQ

技术内幕

RocketMQ架构设计与实现原理

丁威 周继锋◎著



RocketMQ Internals: Architecture, Design and Principle



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

RocketMQ 技术内幕: RocketMQ 架构设计与实现原理 / 丁威, 周继锋著. —北京: 机械工业出版社, 2019.1

ISBN 978-7-111-61421-0

I.R… II. ①丁… ②周… III. 计算机网络 - 软件工具 IV. TP393.07

中国版本图书馆 CIP 数据核字 (2018) 第 263145 号

RocketMQ 技术内幕

RocketMQ 架构设计与实现原理

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 张锡鹏

责任校对: 殷虹

印刷: 北京市兆成印刷有限责任公司

版次: 2019 年 1 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 18.75

书号: ISBN 978-7-111-61421-0

定价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

华章IT
HZBOOKS | Information Technology



Foreword 推荐序

当前，全球商业正面临着产业、组织和价值观三大重构。首先，数字技术重新定义了商业模式，颠覆了原有的产业组织，许多行业中的领头羊，不是被行业内部的追赶者所取代，就是因技术进步而受到挑战，技术变革影响行业竞争格局的速度和深度都发生了数量级上的变化；其次，企业的组织也需要重构，过去的中心、多中心式的组织结构已经不能适应数字时代的需要，新时代的组织要全面向分布式升级。再次，价值观也在重构，数字经济体奉行的价值观是开放、分享、透明和责任，这 and 传统企业的价值观大相径庭。在这一全新价值观驱动下，企业之间为了更好地推进自身组织结构升级，提高产能，已经将目光逐渐转向开放技术，尤其是云计算领域。而开放、无技术绑定、技术标准恰恰是云计算三个最为典型的特征。在这一领域，RedHat 无疑是冲在了最前头，其最新推出的《开源故事》，由一系列旨在弘扬开源价值（如英才培养、社区构建、透明文化）的精彩影片构成，详细阐述了这一价值正在对我们的工作与生活所产生的影响。从教育领域到制造社区，再到慈善组织和环保事业，无不充分体现了开源价值理念，以及拥抱这一理念的个人如何推动开放文化成为 21 世纪创新的新模式。

人的本性中有一种崇尚自由、希望共享的心理。毕竟，隐私只是我们个人生活的一小部分而已，而我们的社会团体生活中，开放却是促进整个集体进步的重要因素。当我们分清楚什么不能公开和什么该公开的时候，以及当我们很好地平衡这两者而非抑制其中一方的时候，我们的生活就会更加美好。开放它们不但没有使公开者蒙受损失，反而引起了人们更多的兴趣和热情，使得相关的技术发展进入一个良性循环而稳步前进，这就产生了一种良好的社会风气。

RocketMQ 的开源正是源于对这种开源文化的认同，开放是为了更好的协同创新，并将这

一技术推向新的高度。在经历了阿里巴巴集团内部多年“双 11”交易核心链路工业级场景的验证，2016 年 11 月，团队将 RocketMQ 捐献给全球享有盛誉的 Apache 软件基金会，正式成为孵化项目。至此，RocketMQ 开启了迈向全球顶级开源软件的新征程。通过社区与团队半年多的开放式创新。2017 年 9 月 25 日，Apache 软件基金会官方宣布，阿里巴巴捐赠给 Apache 社区的开源分布式消息引擎 RocketMQ 从社区正式毕业，成为与 Apache Spark “同款”的顶级项目（TLP）。最近，我们也看到越来越多的中国本土开源软件进入 Apache，Apache 优秀的社区理念驱动着更多立志打造世界级品牌的团队不断发展，促进其生态更加健康和活跃。

在过去几年里，RocketMQ 在第 10 届、第 16 届中北亚开源高峰论坛，以及 2017 年工信部开源峰会的开源软件评比中，都代表中国更高的开源技术，获得了傲人的开源大奖。在这里，请允许我代表 RocketMQ 团队，感谢大家这些年对 RocketMQ 的厚爱与支持。丁威同学是 RocketMQ 社区里比较早期的“布道师”，也非常感谢他能将自己这些年的积累整理成书，帮助大家更好地理解 RocketMQ，参与到 RocketMQ 社区建设中来。目前，RocketMQ 团队与社区正在构建下一代 RocketMQ 以及领域标准 OpenMessaging，我们希望它是 Cloud Native 编程范式下首选的金融级高可靠、高性能数据消息计算平台，也非常欢迎大家能够加入到社区建设中来。

人类的生活正在因为开源软件而变得更加美好，让我们一起来构筑美好未来！

冯嘉

Alibaba Messaging 开源技术负责人

Apache RocketMQ 创始人

Linux OpenMessaging 创始人、主席

为什么要写这本书

随着互联网技术蓬勃发展，微服务架构思想的兴起，系统架构开始追求小型化、轻量化，原有的大型集中式的 IT 系统通常需要进行垂直拆分，孵化出颗粒度更小的众多小型系统，因此对系统间松耦合的要求越来越高，目前 RPC、服务治理、消息中间件几乎成为互联网架构的标配。

引入消息中间件，服务之间可以通过可靠的异步调用，降低系统之间的耦合度，提高系统的可用性。消息中间件的另一个重要应用场景是解决系统之间数据的一致性（最终一致性）。

RocketMQ 作为阿里开源的一款高性能、高吞吐量的消息中间件，承载了阿里“双 11”大部分业务，可以说是一名久经战场的“精英”、值得信任的“伙伴”。同时它的开发语言为 Java，自然而然地得到了广大互联网架构师们的青睐，成为互联网行业首选的消息中间件。

初次接触 RocketMQ 是在听到阿里巴巴正式将 RocketMQ 捐献给 Apache 基金会，成为 Apache 的顶级开源项目时，这意味着承载阿里“双 11”巨大流量的消息中间件完全走向开源，对广大 Java 开发者来说无疑是一个巨大的利好，让我们有机会一睹高性能消息中间件 RocketMQ 的“真容”。作为一名阿里技术崇拜者，我内心异常激动，于是不假思索地在 CSDN 上开通了专栏“源码研究 RocketMQ”，受到了广大技术朋友的支持。

RocketMQ 作为一款高性能消息中间件，其核心优势是可靠的消息存储、消息发送的高性能与低延迟、强大的消息堆积能力与消息处理能力、严格的顺序消息模式等。RocketMQ 的另一个核心思想是懂得取舍。软件设计不可能做到面面俱到，消息中间件的理想状态是一条消息能且只能被消费一次，但要做到这一点，必然需要牺牲性能。RocketMQ 的设计者解决这一难题的办法是不去解决，即保证消息至少被消费一次，但不承诺消息不会被消费者多次消费，

其消费的幂等由消费者实现，从而极大地简化了其实现内核，提高了 RocketMQ 的整体性能。

自从 RocketMQ 被捐献给 Apache 基金会后便在快速发展，RocketMQ 的设计者们正在制定消息中间件的新规范，其模块为 openmessaging。本书主要是抛砖引玉，与各位读者朋友们探讨 RocketMQ 的实现原理，使读者能更好地在实际项目中应用 RocketMQ。

读者对象

- ❑ RocketMQ 用户和爱好者
- ❑ RocketMQ 代码开发志愿者
- ❑ Java 中高级开发工程师
- ❑ Java 架构师
- ❑ 有志于从事 Java 开源的相关技术从业者

本书特色

本书从源码的角度对 RocketMQ 的实现原理进行详细剖析，从中阐述了作者学习阅读源码的方法。本书作为一本源码阅读类书籍，其讲解切入点并不是以组成 RocketMQ 的一个个源码包进行展开，而是基于功能模块（如 Topic 路由中心、消息发送、消息存储、消息消费、事务消息）来展开，更加贴近实战需求。

如何阅读本书

本书分为三大部分。

第一部分为准备篇（第 1 章），简单介绍了 RocketMQ 的设计理念与目标，并介绍了在开发工具中如何对 RocketMQ 进行代码调试。

第二部分为实现篇（第 2~8 章），重点讲解了 RocketMQ 各个功能模块的实现原理，包括 NameServer、消息发送、消息存储、消息消费、消息过滤、顺序消息、事务消息等。

第三部分为实例篇（第 9 章），通过示例展示 RocketMQ 的使用技巧，着重讲解了 RocketMQ 的监控命令与监控管理界面。

本书在最后的附录中给出了 RocketMQ 的主要参数列表及含义，供读者参考。

本书的行文思路主要是根据消息发送的全流程进行展开，从路由管理到消息发送、消息

存储、消息消费，再到顺序消息、事务消息，从而实现消息链路的闭环。建议读者按照该思路，带着问题来阅读本书，或许会事半功倍。

勘误和支持

除封面署名外，参加本书编写工作的还有陈鹏飞。由于水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。为此，大家可以通过 CSDN 博客专栏（<https://blog.csdn.net/column/details/20603.html>）留言反馈。书中的全部源文件可以从 github rocketmq 官方仓库中下载，我也会将相应的功能及时更新。如果你有更多的宝贵意见，也欢迎发送邮件至 dw19871218pmz@126.com，期待能够得到你的真挚反馈。

致谢

首先要感谢 MyCAT 开源社区负责人周继锋对我的提携与指导，为我的职业发展指明前进的方向。

感谢 RocketMQ 联盟中每一位充满创意和活力的朋友——奔腾、zenk、共产国际史派克、水动力皮划艇、张登、张凤凰、曾文、季永超，以及名单之外的很多朋友，感谢你们对我的支持与帮助。感谢杨福川老师的引荐，是你的努力才促成了本书的成功出版。

感谢机械工业出版社华章公司的编辑张锡鹏，在这一年多的时间中始终支持我的写作，你的鼓励和帮助引导我能顺利完成全部书稿。

最后感谢我的爸爸、妈妈、爷爷、奶奶，感谢你们将我培养成人，并时时刻刻为我灌输着信心和力量！感谢我的老婆、女儿，你们是我持续努力的最大动力。

谨以此书献给我最亲爱的家人，以及众多热爱 RocketMQ 的朋友们！

丁威

感谢 RocketMQ 团队，是你们的付出才有这么好的产品，同时感谢杨福川编辑对本书出版工作的支持。

谨以此书献给我最亲爱的家人和同事，以及帮助过、关注过我的人，以及使用、学习过 RocketMQ 的朋友们！

周继锋

目 录 Contents

推荐序

前言

第1章 阅读源代码前的准备 1

1.1 获取和调试 RocketMQ 的源代码 ... 1

1.1.1 Eclipse 获取 RocketMQ 源码 ... 2

1.1.2 Eclipse 调试 RocketMQ 源码 ... 9

1.1.3 IntelliJ IDEA 获取 RocketMQ 源码 15

1.1.4 IntelliJ IDEA 调试 RocketMQ 源码 20

1.2 RocketMQ 源代码的目录结构 ... 27

1.3 RocketMQ 的设计理念和目标 ... 28

1.3.1 设计理念 28

1.3.2 设计目标 28

第2章 RocketMQ路由中心 NameServer 31

2.1 NameServer 架构设计 31

2.2 NameServer 启动流程 32

2.3 NameServer 路由注册、故障 剔除 36

2.3.1 路由元信息 36

2.3.2 路由注册 38

2.3.3 路由删除 43

2.3.4 路由发现 46

2.4 本章小结 47

第3章 RocketMQ消息发送 49

3.1 漫谈 RocketMQ 消息发送 49

3.2 认识 RocketMQ 消息 50

3.3 生产者启动流程 51

3.3.1 初识 DefaultMQProducer 消息发送者 51

3.3.2 消息生产者启动流程 54

3.4 消息发送基本流程 56

3.4.1 消息长度验证 56

3.4.2 查找主题路由信息 56

3.4.3 选择消息队列 60

3.4.4 消息发送 65

3.5 批量消息发送 71

3.6 本章小结 74

第4章 RocketMQ消息存储 75

4.1 存储概要设计 75

4.2	初识消息存储	76	5.4	消息拉取	133
4.3	消息发送存储流程	78	5.4.1	PullMessageService 实现 机制	133
4.4	存储文件组织与内存映射	83	5.4.2	ProcessQueue 实现机制	136
4.4.1	MappedFileQueue 映射文件 队列	84	5.4.3	消息拉取基本流程	138
4.4.2	MappedFile 内存映射文件	87	5.5	消息队列负载与重新分布 机制	154
4.4.3	TransientStorePool	93	5.6	消息消费过程	162
4.5	RocketMQ 存储文件	94	5.6.1	消息消费	163
4.5.1	Commitlog 文件	95	5.6.2	消息确认 (ACK)	167
4.5.2	ConsumeQueue 文件	97	5.6.3	消费进度管理	171
4.5.3	Index 索引文件	100	5.7	定时消息机制	176
4.5.4	checkpoint 文件	104	5.7.1	load 方法	177
4.6	实时更新消息消费队列与索引 文件	105	5.7.2	start 方法	178
4.6.1	根据消息更新 ConsumeQueue	107	5.7.3	定时调度逻辑	179
4.6.2	根据消息更新 Index 索引 文件	108	5.8	消息过滤机制	181
4.7	消息队列与索引文件恢复	109	5.9	顺序消息	186
4.7.1	Broker 正常停止文件恢复	112	5.9.1	消息队列负载	187
4.7.2	Broker 异常停止文件恢复	114	5.9.2	消息拉取	187
4.8	文件刷盘机制	115	5.9.3	消息消费	188
4.8.1	Broker 同步刷盘	116	5.9.4	消息队列锁实现	195
4.8.2	Broker 异步刷盘	119	5.10	本章小结	196
4.9	过期文件删除机制	122	第5章	RocketMQ消息消费	127
4.10	本章小结	126	5.1	RocketMQ 消息消费概述	127
第5章	RocketMQ消息消费	127	5.2	消息消费者初探	128
5.1	RocketMQ 消息消费概述	127	5.3	消费者启动流程	130
5.2	消息消费者初探	128	5.4	消息拉取	133
5.3	消费者启动流程	130	5.4.1	PullMessageService 实现 机制	133
5.4	消息拉取	133	5.4.2	ProcessQueue 实现机制	136
5.4.1	PullMessageService 实现 机制	133	5.4.3	消息拉取基本流程	138
5.4.2	ProcessQueue 实现机制	136	5.5	消息队列负载与重新分布 机制	154
5.4.3	消息拉取基本流程	138	5.6	消息消费过程	162
5.5	消息队列负载与重新分布 机制	154	5.6.1	消息消费	163
5.6	消息消费过程	162	5.6.2	消息确认 (ACK)	167
5.6.1	消息消费	163	5.6.3	消费进度管理	171
5.6.2	消息确认 (ACK)	167	5.7	定时消息机制	176
5.6.3	消费进度管理	171	5.7.1	load 方法	177
5.7	定时消息机制	176	5.7.2	start 方法	178
5.7.1	load 方法	177	5.7.3	定时调度逻辑	179
5.7.2	start 方法	178	5.8	消息过滤机制	181
5.7.3	定时调度逻辑	179	5.9	顺序消息	186
5.8	消息过滤机制	181	5.9.1	消息队列负载	187
5.9	顺序消息	186	5.9.2	消息拉取	187
5.9.1	消息队列负载	187	5.9.3	消息消费	188
5.9.2	消息拉取	187	5.9.4	消息队列锁实现	195
5.9.3	消息消费	188	5.10	本章小结	196
5.9.4	消息队列锁实现	195	第6章	消息过滤FilterServer	198
5.10	本章小结	196	6.1	ClassFilter 运行机制	198
第6章	消息过滤FilterServer	198	6.2	FilterServer 注册剖析	199
6.1	ClassFilter 运行机制	198	6.3	类过滤模式订阅机制	202
6.2	FilterServer 注册剖析	199	6.4	消息拉取	205
6.3	类过滤模式订阅机制	202	6.5	本章小结	206
6.4	消息拉取	205			
6.5	本章小结	206			

第7章 RocketMQ主从同步(HA)**机制** 207

7.1 RocketMQ 主从复制原理 207

7.1.1 HAService 整体工作机制 208

7.1.2 AcceptSocketService 实现
原理 2087.1.3 GroupTransferService 实现
原理 210

7.1.4 HAClient 实现原理 211

7.1.5 HACConnection 实现原理 214

7.2 RocketMQ 读写分离机制 220

7.3 本章小结 223

第8章 RocketMQ事务消息 225

8.1 事务消息实现思想 225

8.2 事务消息发送流程 226

8.3 提交或回滚事务 232

8.4 事务消息回查事务状态 233

8.5 本章小结 240

第9章 RocketMQ实战 242

9.1 消息批量发送 242

9.2 消息发送队列自选择 243

9.3 消息过滤 243

9.3.1 TAG 模式过滤 244

9.3.2 SQL 表达模式过滤 244

9.3.3 类过滤模式 245

9.4 事务消息 247

9.5 Spring 整合 RocketMQ 250

9.6 Spring Cloud 整合 RocketMQ 251

9.7 RocketMQ 监控与运维命令 258

9.7.1 RocketMQ 监控平台搭建 258

9.7.2 RocketMQ 管理命令 261

9.8 应用场景分析 280

9.9 本章小结 281

附录A 参数说明 282

阅读源代码前的准备

研究一款开源中间件，首先我们需要了解它的整体架构以及如何在开发环境调试源码，从代码入手才能快速熟悉一个开源项目，只有这样才能抽丝剥茧地理解透彻，了解作者的设计思想和实现原理。本章将重点介绍 RocketMQ 的整体设计理念以及如何调试 RocketMQ，为后续源码阅读打下扎实的基础。

本章重点内容如下。

- 获取和调试 RocketMQ 源代码
- RocketMQ 源代码的目录结构
- RocketMQ 的设计理念和设计目标

1.1 获取和调试 RocketMQ 的源代码

RocketMQ 原先是阿里巴巴内部使用的消息中间件，于 2017 年提交到 Apache 基金会成为 Apache 基金会的顶级开源项目，GitHub 代码库链接：<https://github.com/apache/rocketmq.git>。在 Github 网站上搜索 RocketMQ，如图 1-1 所示。

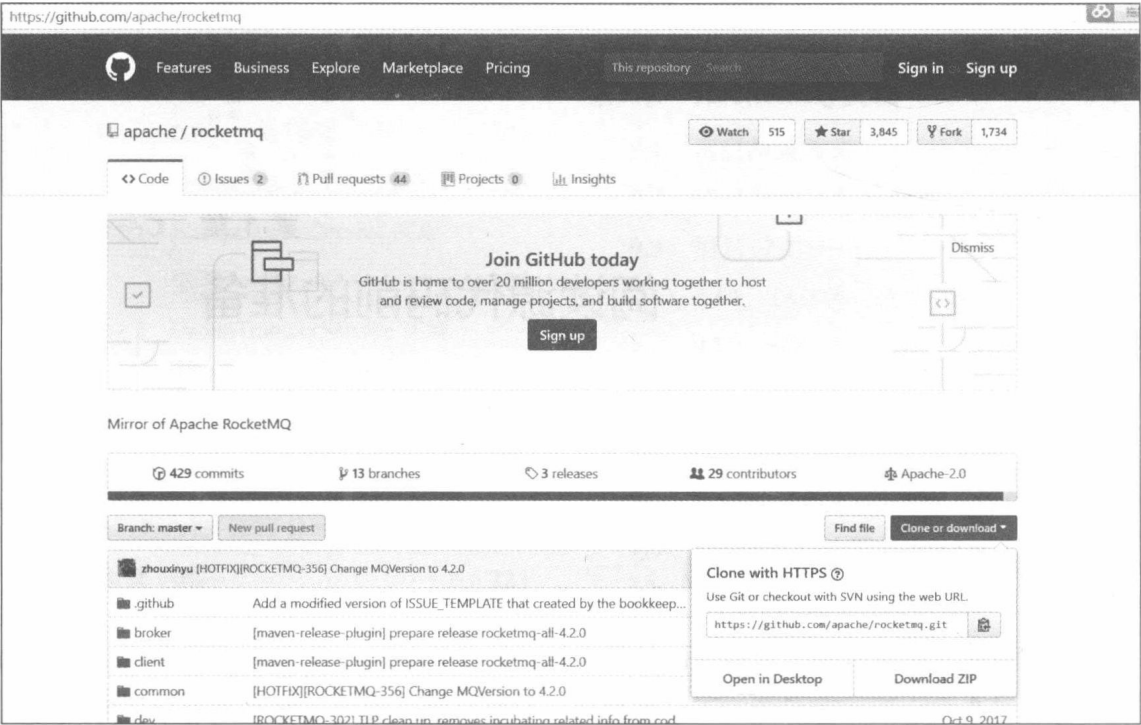


图 1-1 GitHub RocketMQ 搜索界面

1.1.1 Eclipse 获取 RocketMQ 源码

Step1: 单击右键从菜单中选择 import git，弹出如图 1-2 所示的对话框。

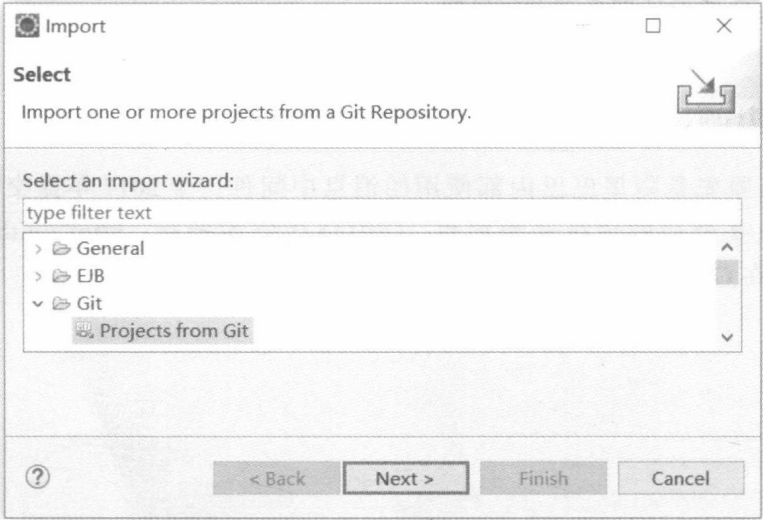


图 1-2 Import 对话框

Step2: 点击 Next 按钮, 弹出 Projects from Git 对话框, 如图 1-3 所示。

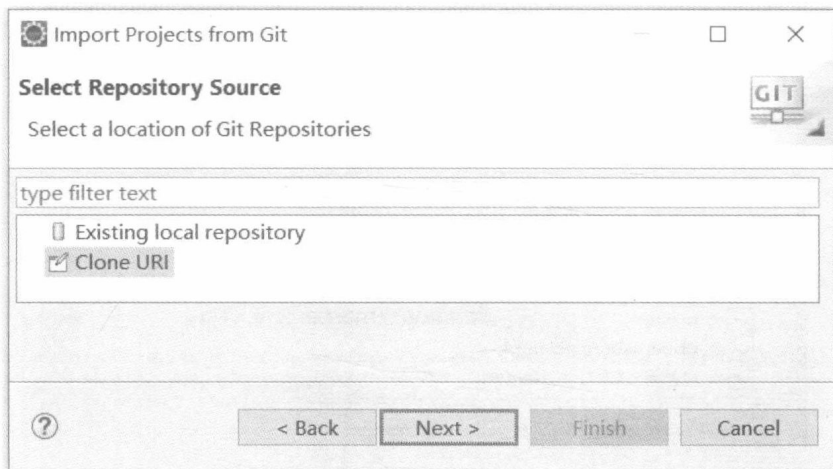


图 1-3 Import Projects from Git 对话框

Step3: 点击 Next 按钮, 弹出 Clone URI 对话框, 如图 1-4 所示。

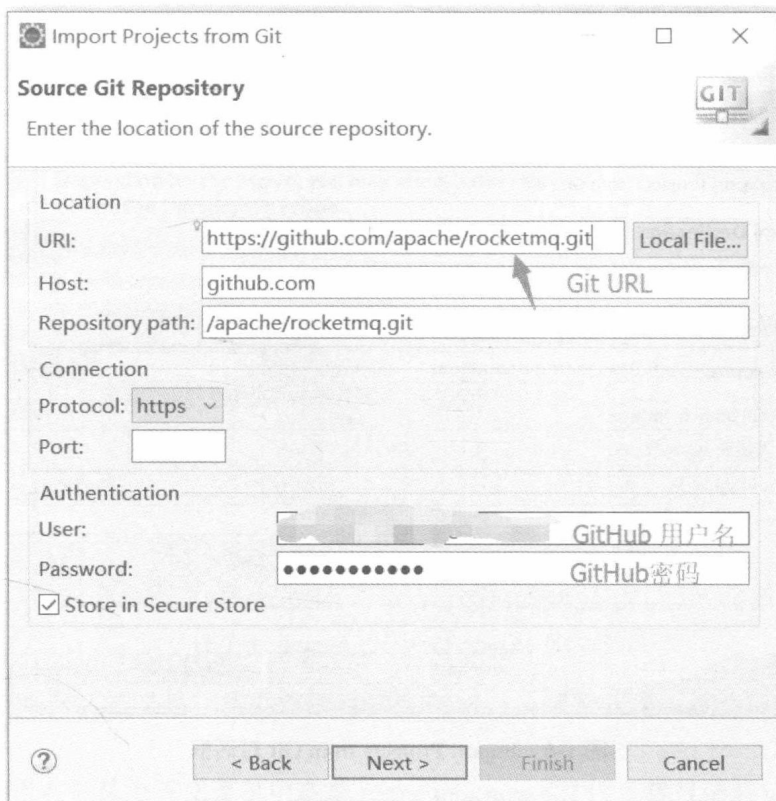


图 1-4 Import Projects from Git 对话框

Step4: 继续点击 Next 进入下一步，选择代码分支，如图 1-5 所示。

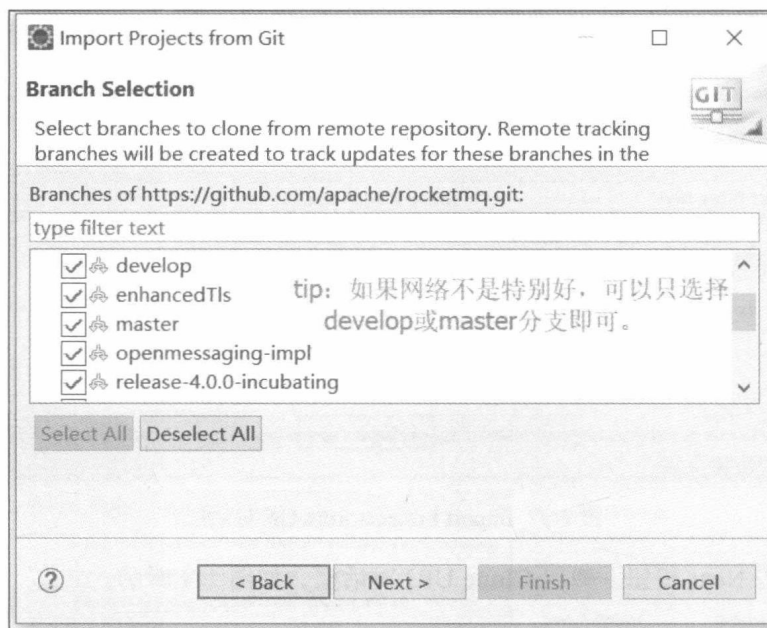


图 1-5 Import Projects from Git 对话框

Step5: 选择所需要的分支后点击 Next，进入代码存放目录选择，如图 1-6 所示。

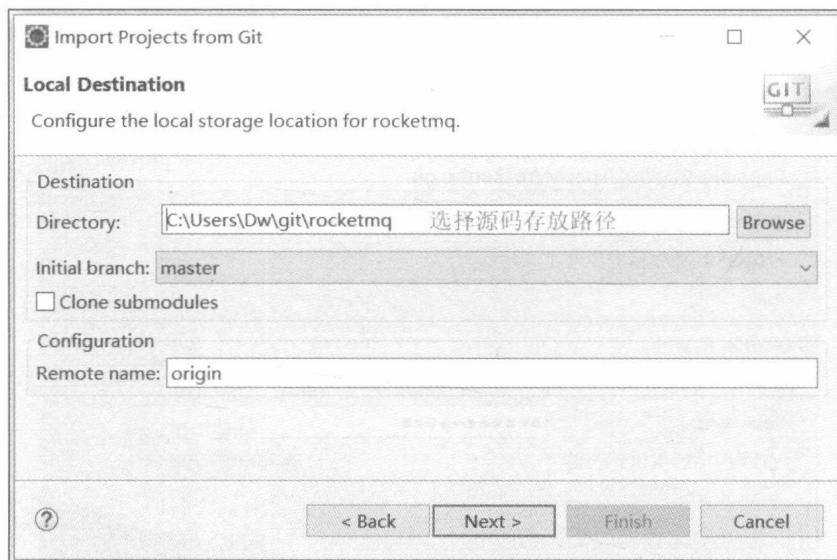


图 1-6 Import Projects from Git 对话框

Step6: 点击 Next，Eclipse 将从远程仓库下载代码，如图 1-7 所示。

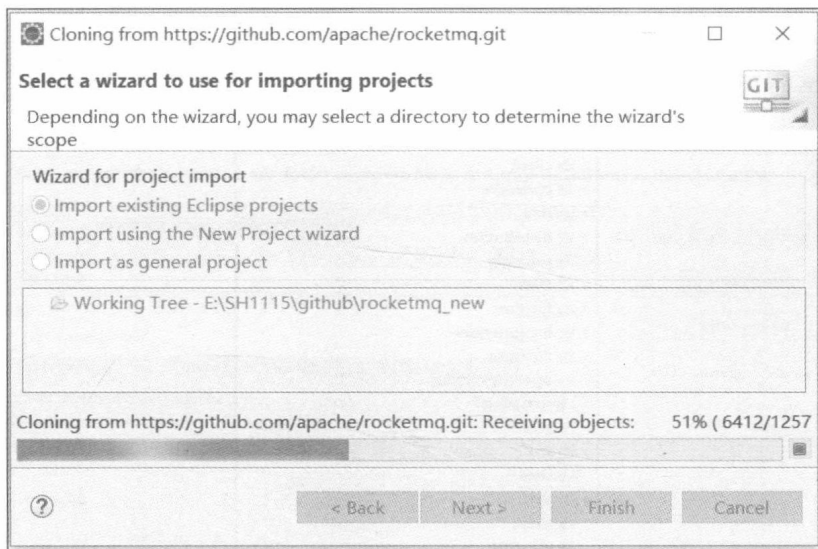


图 1-7 Import Projects from Git 对话框

Step7: 代码下载到指定目录后, 默认选择 Import existing projects (单分支), 这里手动选择 Import as general projects (多分支), 点击 Finish, 成功导入, 如图 1-8 所示。

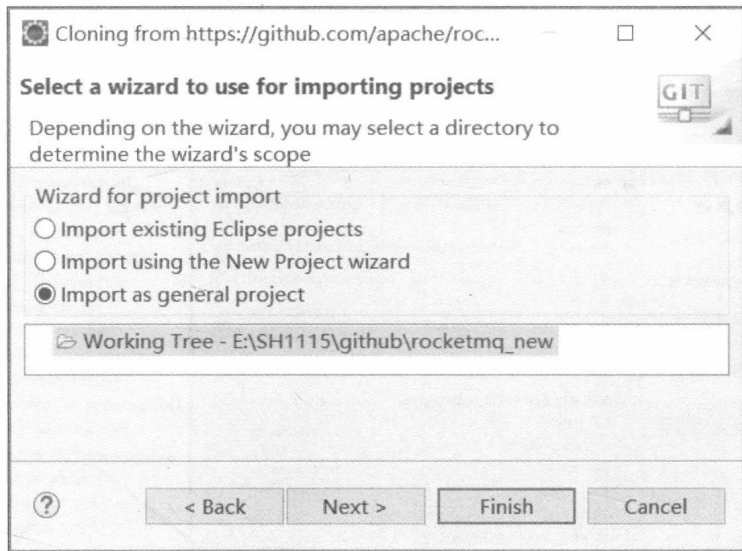


图 1-8 Import Projects from Git 对话框

Step8: 代码导入成功后, 需要将项目转换成 Maven 项目, 导入成功后的效果图, 如图 1-9 所示。

Step9: 单击右键从上下文菜单中选择 rocketmq_new (文件下载目录名) → Configure → Convert and Detect Nested Projects 转换成 Maven 项目, 如图 1-10 所示。