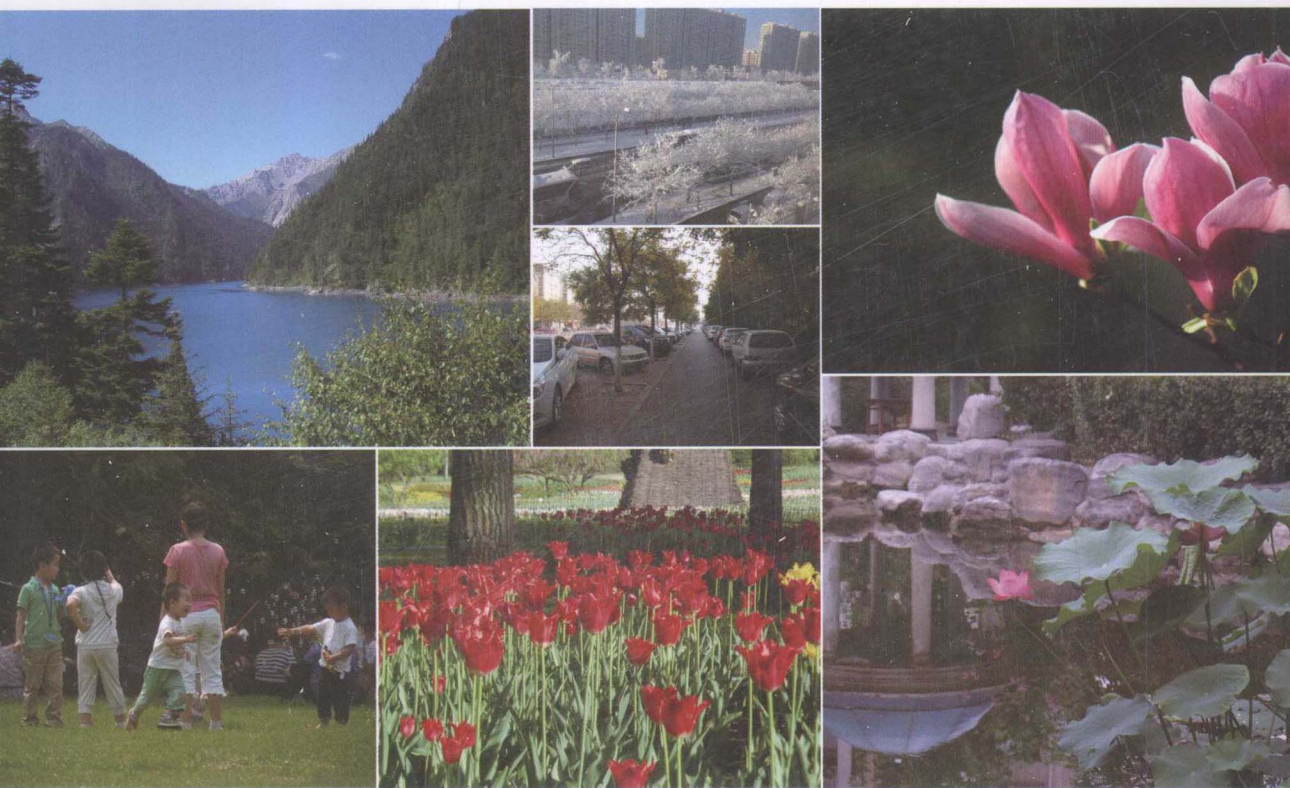




The Open Source Media Server

视频应用开发指南



郑虎 编著

- ➡ 全面讲解Red5视频应用开发技巧
- ➡ 综合Red5、Java、Flash、ActionScript、ASP.NET技术
- ➡ 点播、直播、电子白板、视频会议一个都没有少



本书代码提供云下载



清华大学出版社



视频应用开发指南

郑虎 编著

清华大学出版社
北 京

内 容 简 介

Red5 作为免费开源的 Flash 流媒体服务器, 广泛用在 Flash 视频网站、Flash 直播系统、视频会议系统、网游等产品领域。这些领域的开发人员需要全面掌握 Red5 视频应用开发的知识和技巧。

全书共 10 章, 全面讲解了 Red5 视频应用开发的基础知识、开发技巧、开发注意事项, 并相对完整地实现了视频播放器、直播系统、点播系统、电子白板系统、视频会议系统等应用。本书所有的代码和资源提供网络下载。

本书适合 Red5 流媒体产品的软件开发人员、系统分析员、产品经理, 也可以作为高等院校和培训机构相关专业课程的教学参考书和实验教程。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Red5 视频应用开发指南 / 郑虎编著. -- 北京: 清华大学出版社, 2014

ISBN 978-7-302-32711-0

I. ①R… II. ①郑… III. ①服务器软件—软件开发—指南 IV. ①TP393.09-62

中国版本图书馆 CIP 数据核字 (2013) 第 125564 号

责任编辑: 王金柱

封面设计: 王 翔

责任校对: 闫秀华

责任印制: 刘海龙

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社总机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn



印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 190mm×260mm

印 张: 31.25

字 数: 800 千字

版 次: 2014 年 1 月第 1 版

印 次: 2014 年 1 月第 1 次印刷

印 数: 1~3000

定 价: 69.00 元

前言

Red5 是一个采用 Java 开发的开源 Flash 流媒体服务器。Red5 的主要功能和 Adobe 公司的 Flash Media Server (FMS) 类似, 它提供基于 Flash 的流媒体服务, 使用 RTMP 作为流媒体传输协议, 与 FMS 完全兼容。Red5 具有流化 FLV、MP3 文件, 实时录制客户端流为 FLV 文件, 发布现场直播流, 共享对象, 远程调用等功能。

Red5 作为免费开源的服务器, 可以替换 Flash Media Server, 广泛应用于 Flash 视频网站、Flash 直播系统、视频会议系统、网游等产品领域。这些领域的开发人员需要全面掌握 Red5 视频应用开发的所有细节, 才能开发出受市场欢迎的、技术过硬的产品出来。

全书共 10 章, 全面讲解了 Red5 视频应用开发的基础知识、开发技巧、开发注意事项以及主要视频应用类型相对完整的实现, 书中完整的实例包括直播系统、点播系统、视频会话系统、电子白板系统、视频点播系统和视频直播系统。本书提供下载的代码包中包括本书所有的代码和资源, 以方便读者学习, 也可利用书中完整的实例代码尝试开发自己的视频应用。本书各章内容安排如下:

第 1 章讲解 Red5 与 Flash Media Server (FMS) 的背景知识, Red5 架构以及 FMS 应用迁移到 Red5 的注意事项。

第 2 章讲解 Red5 安装与配置, Demo 的用法以及 Red5 应用的调试方法。

第 3 章讲解客户端 Flash 与 Red5 服务器端的通讯方法, 简单介绍了 ActionScript 3.0 语言, 并详细讲述了 NetConnection 对象的使用。

第 4 章讲解使用连接进行通信的方法, 以及一个简单的视频聊天实例的实现过程。

第 5 章讲解视频播放原理, NetStream 对象的使用方法以及一个视频播放器实例的实现方法。

第 6 章讲解直播的基本概念和流程、摄像头与麦克风对象的使用, 以及直播过程的实现方法。

第 7 章在第 6 章的基础上重点讲解一个简单的视频会话实例的流程和实现方法。

第 8 章讲解共享对象 SharedObject, 绘图的实现方法, 并实现一个 SharedObject 电子白板应用。

第 9 章讲解视频点播系统的开发方法, 包括视频格式转换软件的使用。

第 10 章讲解视频直播系统的开发方法。

Red5 视频应用涉及的知识面很宽, 本书尽量使用简洁清晰的表达, 对视频应用开发的主

要基础和编程技巧进行讲解，知识点整体安排从易到难、由浅入深。在学习本书时，读者可以按顺序逐章学习，牢固掌握相关知识点和相关编程技巧，并在项目中多加实践。本书使用到以下工具和技术：

- Eclipse Indigo 3.7.1，Red5 服务器端程序开发环境。
- JDK 6.0 Update 26，用于开发 Red5 服务器端程序，必须掌握 Java 编程技术。
- Flash CS5，用于构建 Flash 客户端，必须掌握 ActionScript 3.0 编程技术。
- ASP.NET 4.0，用于构建视频网站系统。读者可以只参看系统的实现逻辑，使用自己更熟悉的 J2EE 开发技术，甚至可以只用 JSP 实现一个视频系统。

本书适合 Red5 流媒体产品的软件开发人员、系统分析员、产品经理，也可以作为高等院校和培训机构相关专业课程的教学参考书和实验教程。

本书代码包下载地址如下：<http://pan.baidu.com/s/1svPbN>。

由于编者水平有限，书中难免有疏漏之处，望广大读者批评指正，意见与建议请 E-mail 至 booksaga@163.com，方便编者在新一版本的书中做出改进。如果代码包下载有问题，请电子邮件联系 booksaga@163.com，邮件标题为“求代码，Red5”。

编者

2013 年 10 月

目 录

第 1 章 Red5 与 Flash Media Server

1.1	Flash Media Server 的起源与发展	2
1.2	Red5 的起源与发展	3
1.3	两者的对比	3
1.4	Red5 的设计与架构	4
1.4.1	Jetty 的架构	5
1.4.2	Mina 的架构	9
1.5	从 Flash Media Server 迁移到 Red5	11
1.5.1	类 ApplicationAdapter (Red5)	11
1.5.2	接受或拒绝客户端	12
1.5.3	当前连接与客户端	12
1.5.4	附加句柄	13
1.5.5	通过配置文件配置句柄	13
1.5.6	在应用端代码里配置句柄	13
1.5.7	调用客户端方法	14
1.5.8	SharedObject	14
1.5.9	服务器端修改 SharedObject 监听者	15
1.5.10	服务器端修改 SharedObject	16
1.5.11	SharedObject 事件句柄	16

第 2 章 Red5 安装与配置

2.1	Red5 运行所需要的软件环境	19
-----	-----------------------	----

2.2	安装 Red5	19
2.3	访问 Red5 的 demo 页面	30
2.4	Red5 的配置与说明	34
2.4.1	red5.properties	34
2.4.2	red5.xml	38
2.4.3	red5-common.xml	38
2.4.4	red5-core.xml	38
2.4.5	red5-web.xml	38
2.5	如何调试 Red5	38
2.5.1	查看日志	39
2.5.2	red5-debug.bat	39
2.5.3	使用 Eclipse 进行调试	40

第 3 章 连接到 Red5

3.1	创建一个 Red5 应用程序	42
3.1.1	创建一个 Red5 应用	42
3.1.2	发布应用到 Red5 的服务器	51
3.2	Flash ActionScript 3 的基本介绍	56
3.3	NetConnection 对象的说明和使用	57
3.4	创建一个连接对象	61
3.5	管理连接状态	63

第 4 章 使用连接进行通信

4.1	连接通信概述	73
4.2	Red5 的服务器端对象	73
4.3	通信流程分析	75
4.4	SharedObject 的说明与使用	78
4.5	获取用户列表	81
4.5.1	服务器端	82
4.5.2	Flash 客户端	86

4.6 广播信息到所有用户	90
4.6.1 服务器端	90
4.6.2 Flash 客户端	91
4.7 单播信息到特定用户	93
4.8 一个简单的聊天程序	97

第 5 章 视频播放

5.1 视频播放原理概述	110
5.2 播放流程分析	111
5.3 NetStream 的说明与使用	112
5.4 播放视频文件	119
5.5 控制视频播放	121
5.6 一个简单的视频播放器实例	125
5.6.1 简单的视频播放器界面	125
5.6.2 播放按钮	126
5.6.3 停止按钮的逻辑	128
5.6.4 播放进度条的实现逻辑	129
5.6.5 缓冲条的进度	132
5.6.6 快进和快退功能	132
5.6.7 点击位置的快进或者快退	134
5.6.8 音频控制按钮	139
5.6.9 音量控制条	143
5.6.10 音量	144
5.6.11 停止拖曳动作	145
5.6.12 全屏按钮 fullScreenBtn	146

第 6 章 摄像头与直播

6.1 摄像头与直播原理概述	149
6.2 摄像头与直播流程分析	150
6.3 Camera 的说明与使用	151

6.4	Microphone 的说明与使用	158
6.5	客户端摄像设备的识别与显示	165
6.6	客户端声音设备的识别与显示	168
6.7	上行视频到服务器端	171
6.8	播放一个用户的上行视频	177
6.9	Respond 的说明与使用	180
6.10	检测服务器端与客户端的连接情况	181
6.10.1	客户端到服务器端的带宽检测	185
6.10.2	服务器端到客户端的带宽检测	213
6.11	控制音频/视频质量与播放参数	249

第 7 章 一个简单的视频会话实例

7.1	登录逻辑	265
7.2	断开连接时服务器端执行的动作	269
7.3	发送公开消息和私聊消息	275
7.4	FlashPlayer 客户端	286
7.5	视频发布的功能	288
7.6	视频播放和同步的客户端	290

第 8 章 SharedObject 与电子白板实例

8.1	SharedObject 的工作原理	299
8.2	SharedObject 的同步与控制	301
8.3	ActionScript 3 的绘画 API 的使用与说明	306
8.4	一个简单的电子白板程序	318
8.4.1	服务器端 WhiteBoard	319
8.4.2	Flash 客户端——创作端的实现	319
8.4.3	Flash 客户端——订阅端的实现	344
8.4.4	通信部分的实现	345

第 9 章 视频点播系统的开发与实践

9.1 视频格式的转换与实现	356
9.1.1 FFmpeg	356
9.1.2 mencoder	366
9.2 视频点播集成与开发	375
9.2.1 上传视频文件	375
9.2.2 上传文件的 Web 界面	390
9.2.3 基于 Flash 实现的 SWFUpload	398
9.3 FileReference 类的属性和方法	401
9.3.1 FileReference 类的属性	401
9.3.2 FileReference 类的方法	405
9.4 视频转换	417
9.4.1 视频转换功能	417
9.4.2 转换视频的质量	423
9.4.3 Process 进程转换文件	429
9.5 调用服务的客户端	439
9.6 视频播放的实现	455
9.6.1 视频列表页面	455
9.6.2 播放视频的页面	460

第 10 章 视频直播系统的开发与实践

10.1 视频直播系统的界面与开发	472
10.2 获取 Camera 对象和 Microphone 对象	475
10.3 发布视频功能的实现	482

第 1 章

Red5 与 Flash Media Server

从本章节可以学习到:

- ❖ Flash Media Server 的起源与发展
- ❖ Red5 的起源与发展
- ❖ 两者的对比
- ❖ Red5 的设计与架构
- ❖ 从 Flash Media Server 迁移到 Red5

1.1 Flash Media Server 的起源与发展

Flash Media Server 的前身 Flash Communication Server 最早是由 Macromedia 创造于 2002 年，伴随着 Flash Player 6 的一些关于网络通信的新特性而产生。这些新特性包括：网络通信、语音视频编码与解码、客户端信息共享等。有了它们，Flash Player 从以前的单一解决多媒体播放质量和速度的普通控件，一跃变成了一个可以进行网络通信、完成各种各样通信服务的高级控件。Flash Communication Server 的产品标识如图 1-1 所示。

在 Macromedia 的年代，Flash Media Server 从 1.0 版本一直演化到了 1.5 版本。随后在 2005 年 4 月，Adobe 斥巨资收购了 Macromedia，全盘接收和继承了 Macromedia 的所有产品，并且在当年 11 月推出了 2.0 版本的 Flash Communication Server，产品名称也正式变更为 Flash Media Server。此后数年，Adobe 一直保持着约每两年推出一个新版本 Flash Media Server 的惯例，产品线也随着用户类型和需求划分得更细。目前最新版本的 Flash Media Server 是 4.5，包含 Flash Media Streaming Server、Flash Media Interactive Server、Flash Media Enterprise Server 和 Flash Media Server 4 on Amazon Web Services 共 4 个分支。Flash 产品标识如图 1-2 所示，Flash Media Server 的版本情况如表 1-1 所示。



图 1-1



图 1-2

表 1-1 Flash Media Server 的版本情况

版本	出品方	时间	新特性
1.0	Macromedia	2002 年 7 月	网络通信、语音视频编码与解码等
1.5	Macromedia	2003 年 3 月	Http Tunneling 与 Linux 版本支持等
2.0	Adobe	2005 年 11 月	性能与稳定性改进、新的视频音频编码
3.0	Adobe	2007 年 12 月	性能与稳定性改进、多产品线尝试
3.5	Adobe	2009 年 1 月	多产品线正式推出、尝试 P2P 网络模式
4.0	Adobe	2010 年 9 月	P2P 网络通信模式、Amazon 云平台集成
4.5	Adobe	2011 年 9 月	多产品线调整，正式推出 Amazon 云平台服务

由于 Flash Media Server 的所有客户端应用的基础是安装量和安装范围极大的 Flash

Player, 因此基于它的应用形式目前主要集中在 WEB 上; 而伴随着 Adobe 推出了 Flash AIR 模式, Flash Player 的应用也开始逐渐向桌面端转移, Flash Media Server 的应用范畴也开始随之扩散到桌面端。

Flash Media Server 的应用范畴包括但不仅限于以下范畴: 语音视频聊天室、Web IM、客服系统、电子白板、在线教育等。

1.2 Red5 的起源与发展

Red5 最早是由 Dominick Accattato (<http://blog.infrared5.com/tag/dominick-accattato/>) 发起的, 旨在创建一个可以和 Flash Media Server 并存的开源解决方案。它的主要目的是为了创建一个可以部分代替 Flash Media Server 的开源版本服务器。目前该开源组织一共有 4 个主要成员以及 10 个代码贡献者, 其代码主要存放在 <http://code.google.com/p/red5/> 网站上。

由于是开源性质的组织和开发方式, 因此 Red5 在 release 的时间点的控制上并没有像 Adobe 那么严格, 但一般也都会紧随着 Flash Media Server 而更新。其 release 的时间一般是在 Flash Media Server 新版本之后 2~6 个月的时间内。Red5 各版本情况如表 1-2 所示。

表 1-2 Red5 各版本情况

版本	出品方	时间	新特性
0.6.3 Final	DA	2007 年 9 月	支持 RTMP、支持远程共享
0.7.0 Final	DA	2008 年 2 月	负载均衡模式、支持多线程、修改 BUG
0.8.0 Final	DA	2009 年 6 月	远程共享 BUG 修改、性能改善
0.9.0 Final	DA	2010 年 1 月	架构改善、性能改善、增加多媒体格式支持
0.9.1 Final	DA	2010 年 2 月	重要 BUG 修复
1.0 RC1	DA	2011 年 2 月	重要 BUG 修复, 本书写作在此版本基础上完成的
1.0 Final		2012 年 12 月	
1.0.1 Final		2013 年 1 月	
1.0.2 Snapshot		2013 年 4 月	

开源组织进行服务器端软件开发的重要依据就是针对 Flash Media Server 的通信协议的反向工程, 早期 Flash Media Server 的协议一直都是以 RTMP 为主, 因此 Red5 一直以来也是如此。而 Flash Media Server 在 4.0 时引入的 P2P 模式是基于 RTMFP 协议进行工作, 因此 Red5 并不支持此方式的通信。

1.3 两者的对比

由于支持 Flash Media Server 与 Red5 的开发团队的性质差异（一方为商业软件研发，一方为开源社区模式），因此两者在今天已经有了一些比较显著的差异，具体细节如表 1-3 所示。

表 1-3 Flash Media Server 与 Red5 一些比较显著的差异

产品名称	多媒体格式	协议类型	新特性
Flash Media Server	FLV、MP3、H264、WAV、MP4	RTMFP、RTMP、HTTP	P2P 通信方式、IP 广播、负载均衡
Red5	FLV、MP3、H264、WAV、MP4	RTMP、HTTP、MRTMP	暂无

如果一定需要在两者之间划一条分界线的话，我们可以认为 Red5 等价于一个 Flash Media Server 2.0 到 3.0 之间的中间版本。

1.4 Red5 的设计与架构

如前所述，Red5 是为了实现 Flash Media Server 的语音视频和通信功能而产生的开源项目，Red5 的组成部分如图 1-3 所示。

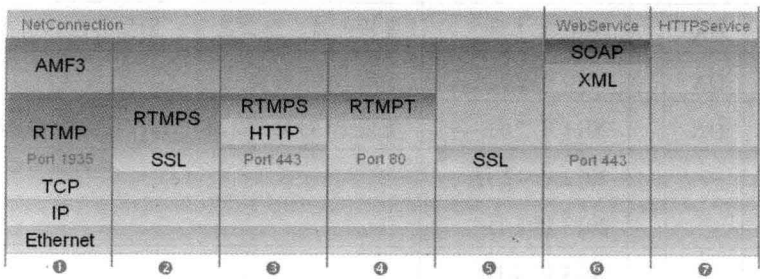


图 1-3

由上图可知，Red5 包含了 HTTP 通信服务、RTMP 通信服务等部分，因此其架构主要由 Java 实现的 Jetty6、Mina 和 Spring 组成，如图 1-4 所示。

其中，Jetty 负责 HTTP 部分的通信和 Java Servlet 的承载和调度，Mina 负责 RTMP 协议的通信和实现，Spring 则负责将两个部分有效地整合在一起实现有机调度。

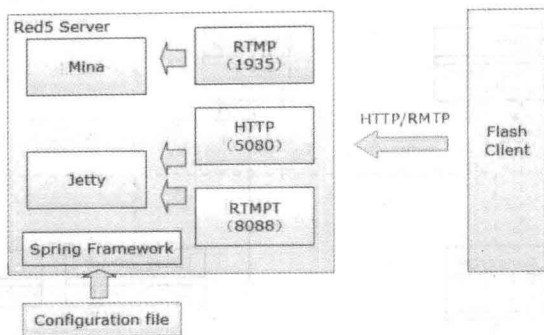


图 1-4

1.4.1 Jetty 的架构

Jetty 的架构简单描述如图 1-5 所示。

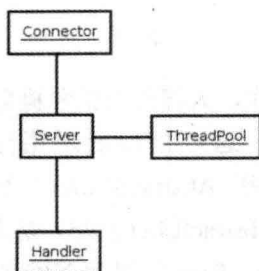


图 1-5

Jetty server 是由一组接受 http 连接的 Connectors 和一组处理来自连接请求并作出响应的 Handlers 构成的，其工作是通过取自线程池中的线程来完成的。



虽然 Jetty 的 Request/Response 是从 Servlet API 传递过来，但是要获得全功能的 Servlet API 必须配置适当的 Handler。比如，在 Request 中的 Session API 只有在 Request 传递给了 Session Handler 之后才可用。Servlet 概念本身是由 ServletHandler 实现的，因此，如果 Servlet 不是必须的且 Servlet API 很少使用，我们可以让 Jetty 通过简单的 Connectors 和 Handlers 来构建连接实现，而不需要 Servlet。

Jetty 的配置工作就是构建一个 Connectors 和 Handlers 的网络，并提供它们单独的配置。因为 Jetty 组件就是简单的 POJO，这种组件的配置可以通过多种技术实现，如下。

- 在代码中实现，可以查看 Jetty 官方提供的 org.mortbay.jetty.example 包中的例子。
- 使用 jetty.xml 格式的依赖注入风格。
- 使用依赖注入框架：Spring 或 Xbean。
- 使用 Deployers：WebAppDeployer 或 ContextDeployer 等。

Jetty 的工作模式如图 1-6 所示。

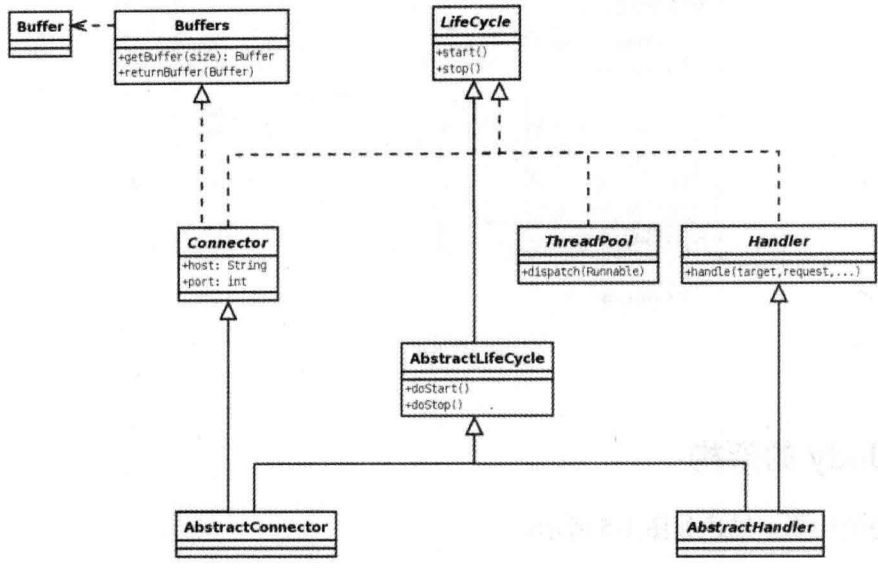


图 1-6

Jetty 的实现遵循一些标准的模式，大部分的抽象概念是通过接口捕获（Captured）的，比如 Connector、Handler、Buffer。这些接口的通用处理通过抽象类来实现，比如：AbstractConnector、AbstractHandler 和 AbstractBuffer。另外，大部分的 Jetty 组件是通过 LifeCycle 接口呈现的，抽象实现（AbstractLifeCycle）是大部分 Jetty 组件的基础。

Jetty 提供了自己的建立在 String、Byte 数组和 NIO 缓冲之上的 IO Buffer 的抽象，这样可以得到更的可移植性，并且也隐藏了一些 NIO 层和它的高级特性的复杂性。

(1) Connectors

Connectors 的实现如图 1-7 所示。

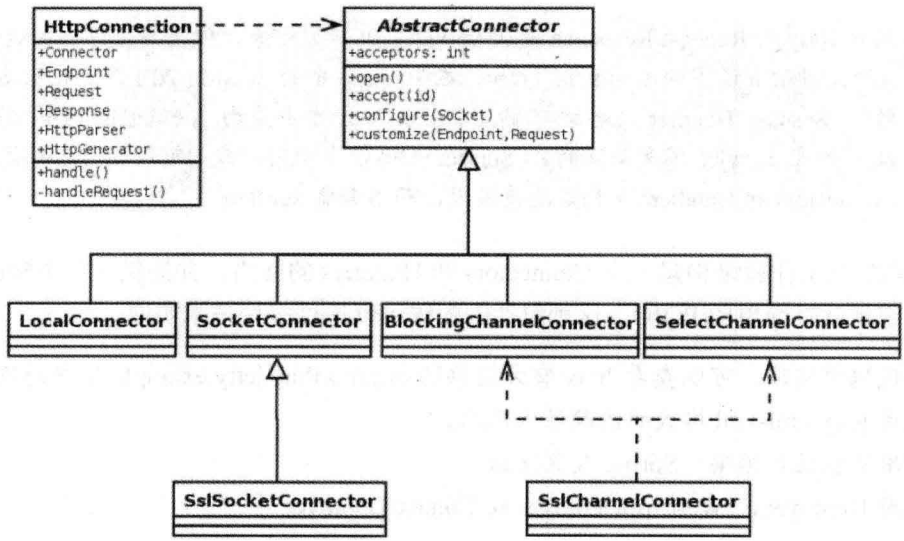


图 1-7

Connectors 进行协议的处理：接受连接、分析请求、生成响应。基于使用的协议、调度模型、IO API 的不同，有多种可用的 Connector，如下所示。

- SocketConnector: 用于繁忙的连接或是 NIO 不可用的场合。
- BlockingChannelConnector: 用于繁忙的连接，并且 NIO 可用。
- SelectChannelConnector: 用于大部分时间空闲的连接或者 Ajax 请求的异步处理。
- SslSocketConnector: 没有使用 NIO 的 SSL 连接器。
- SslSelectChannelConnector: 使用非阻塞 NIO 支持的 SSL 连接器。
- AJPConnector: AJP 协议支持，用于来自 apache mod_jk 或者 mod_proxy_ajp 的连接。

(2) Handlers

Handlers 的实现如图 1-8 所示。

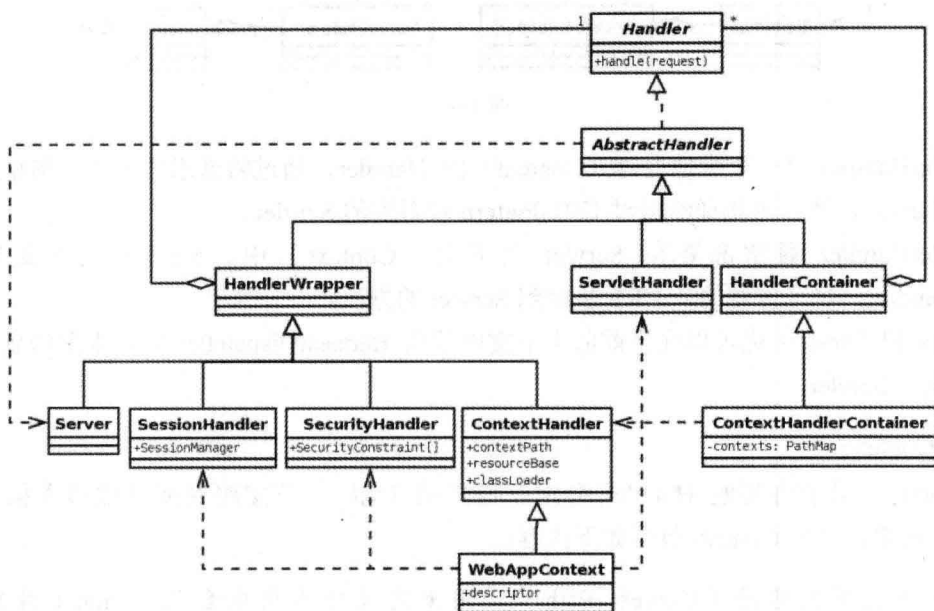


图 1-8

Handler 是用来处理接收到的请求的组件。Handler 的核心 API 是 Handle 方法，如下：

```

public void handle (String target, HttpServletRequest request, HttpServletResponse response, int ispatch) throws IOException, ServletException;
  
```

该方法需要实现对请求的处理，然后将请求传递给另一个 Handler（或 Servlet）。一般而言，有如下三种方式的 Handler。

- 协调处理器：将请求传递给其他的 Handler（如 HandlerCollection、ContextHandlerCollection）。
- 过滤处理器：对请求进行包装后再传递给其他的 Handler（如 HandlerWrapper、ContextHandler、SessionHandler）。
- 生成处理器：生成内容（如 ResourceHandler、ServletHandler）。