ImportNew

Java NIO系列教程（8）：SocketChannel

2016/04/04 | 分类： 教程 | 0 条评论 | 标签： Java NIO, SocketChannel

 译文出处： 郑玉婷 原文出处：Jakob Jenkov

Java NIO中的SocketChannel是一个连接到TCP网络套接字的通道。可以通过以下2种方式创建SocketChannel：

1. 打开一个SocketChannel并连接到互联网上的某台服务器。
2. 一个新连接到达ServerSocketChannel时，会创建一个SocketChannel。

打开 SocketChannel

下面是SocketChannel的打开方式：

```
1 SocketChannel socketChannel = SocketChannel.open();
2 socketChannel.connect(new InetSocketAddress("http://jenkov.com", 80));
```

关闭 SocketChannel

当用完SocketChannel之后调用SocketChannel.close()关闭SocketChannel：

```
1 socketChannel.close();
```

从 SocketChannel 读取数据

要从SocketChannel中读取数据，调用一个read()的方法之一。以下是例子：

```
1 ByteBuffer buf = ByteBuffer.allocate(48);
2 int bytesRead = socketChannel.read(buf);
```

首先，分配一个Buffer。从SocketChannel读取到的数据将会放到这个Buffer中。

然后，调用SocketChannel.read()。该方法将数据从SocketChannel 读到Buffer中。read()方法返回的int值表示读了多少字节进Buffer里。如果返回的是-1，表示已经读到了流的末尾（连接关闭了）。

写入 SocketChannel

写数据到SocketChannel用的是SocketChannel.write()方法，该方法以一个Buffer作为参数。示例如下：

```
String newData = "New String to write to file..." + System.currentTimeMillis();

ByteBuffer buf = ByteBuffer.allocate(48);
buf.clear();
buf.put(newData.getBytes());

buf.flip();

while(buf.hasRemaining()) {
    channel.write(buf);
}
```

注意SocketChannel.write()方法的调用是在一个while循环中的。Write()方法无法保证能写多少字节到SocketChannel。所以，我们重复调用write()直到Buffer没有要写的字节为止。

非阻塞模式

可以设置 SocketChannel 为非阻塞模式（non-blocking mode）.设置之后，就可以在异步模式下调用connect(), read() 和write()了。

connect()

如果SocketChannel在非阻塞模式下，此时调用connect()，该方法可能在连接建立之前就返回了。为了确定连接是否建立，可以调用finishConnect()的方法。像这样：

```
socketChannel.configureBlocking(false);
socketChannel.connect(new InetSocketAddress("http://jenkov.com", 80));

while(! socketChannel.finishConnect() ){
    //wait, or do something else...
}
```

write()

非阻塞模式下，write()方法在尚未写出任何内容时可能就返回了。所以需要在循环中调用write()。前面已经有例子了，这里就不赘述了。

read()

非阻塞模式下,read()方法在尚未读取到任何数据时可能就返回了。所以需要关注它的int返回值，它会告诉你读取了多少字节。

非阻塞模式与选择器

非阻塞模式与选择器搭配会工作的更好，通过将一或多个SocketChannel注册到Selector，可以询问选择器哪个通道已经准备好了读取，写入等。Selector与SocketChannel的搭配使用会在后面详讲。

本系列：

- Java NIO系列教程（1）：Java NIO 概述
- Java NIO系列教程（2）：Channel
- Java NIO系列教程（3）：Buffer
- Java NIO系列教程（4）：Scatter/Gather
- Java NIO系列教程（5）：通道之间的数据传输
- Java NIO系列教程（6）：Selector
- Java NIO系列教程（7）：FileChannel
- Java NIO系列教程（8）：SocketChannel

15

相关文章

- 攻破JAVA NIO技术壁垒
- Java NIO系列教程（12）：Java NIO与IO
- Java NIO系列教程（11）：Pipe
- Java NIO系列教程（10）：Java NIO DatagramChannel
- Java NIO系列教程（9）：ServerSocketChannel
- Java NIO系列教程（7）：FileChannel
- Java NIO系列教程（6）：Selector
- Java NIO系列教程（5）：通道之间的数据传输
- Java NIO系列教程（4）：Scatter/Gather
- Java NIO系列教程（3）：Buffer

发表评论

Comment form

Name*

姓名

邮箱*

请填写邮箱

网站 (请以 http://开头)

请填写网站地址

评论内容*

请填写评论内容

(*) 表示必填项

提交评论

还没有评论。

« Java NIO系列教程（7）：FileChannel
Java NIO系列教程（9）：ServerSocketChannel »

Search for:

Search

Search

- 本周热门文章
- 本月热门
- 热门标签

0 记一次集群内无可用 http 服务问题...

1 Java 技术之垃圾回收机制

2 公司编程竞赛之最长路径问题

3 Java 中的十个"单行代码编程"（O...

4 Java 中 9 个处理 Exception ...

5 HttpClient 以及 Json 传递的...

6 浅析 Spring 中的事件驱动机制

7 浅析分布式下的事件驱动机制（ PubS...

8 探索各种随机函数（Java 环境...

9 Java 守护线程概述

最新评论

- Re: 攻破JAVA NIO技术壁垒
  Hi，请到伯乐在线的小组发帖提问，支持微信登录。链接是： http://group.jobbole.... 唐尤华
- Re: 攻破JAVA NIO技术壁垒
  TCP服务端的NIO写法 服务端怎么发送呢。原谅小白 菜鸟
- Re: 关于 Java 中的 double check ...
  volatile 可以避免指令重排啊。所以double check还是可以用的。 hipilee
- Re: Spring4 + Spring MVC + M...
  Hi，请到伯乐在线的小组发帖提问，支持微信登录。链接是： http://group.jobbole.... 唐尤华
- Re: Spring4 + Spring MVC + M...
  我的一直不太明白，spring的bean容器和springmvc的bean容器之间的关系。 hw_绝影
- Re: Spirng+SpringMVC+Maven+Myba...
  很好，按照步骤，已经成功。 莫凡
- Re: Spring中@Transactional事务...
  声明式事务可以用aop来实现,分别是jdk代理和cglib代理,基于接口和普通类.在同一个类中一个方... chengjiliang
- Re: 关于 Java 中的 double check ...
  在JDK1.5之后，用volatile关键字修饰_INSTANCE属性 就能避免因指令重排导致的对象... Byron

关于ImportNew

ImportNew 专注于 Java 技术分享。于2012年11月11日 11:11正式上线。是的，这是一个很特别的时刻 :)

ImportNew 由两个 Java 关键字 import 和 new 组成，意指：Java 开发者学习新知识的网站。import 可认为是学习和吸收， new 则可认为是新知识、新技术圈子和新朋友……

- 

联系我们

Email：ImportNew.com@gmail.com
新浪微博：@ImportNew
推荐微信号

反馈建议：ImportNew.com@gmail.com
广告与商务合作QQ：2302462408

推荐关注

小组 – 好的话题、有启发的回复、值得信赖的圈子
头条 – 写了文章？看干货？去头条！
相亲 – 为IT单身男女服务的征婚传播平台
资源 – 优秀的工具资源导航
翻译 – 活跃 & 专业的翻译小组
博客 – 国内外的精选博客文章
设计 – UI,网页，交互和用户体验
前端 – JavaScript, HTML5, CSS
安卓 – 专注Android技术分享
iOS – 专注iOS技术分享
Java – 专注Java技术分享
Python – 专注Python技术分享