

# ImportNew

- [首页](#)
- [所有文章](#)
- [资讯](#)
- [Web](#)
- [架构](#)
- [基础技术](#)
- [书籍](#)
- [教程](#)
- [Java小组](#)
- [工具资源](#)

- 导航条 - ▼

## Java NIO系列教程（2）：Channel

2016/03/29 | 分类：[教程](#) | [0 条评论](#) | 标签：[Chanel](#), [Java NIO](#)

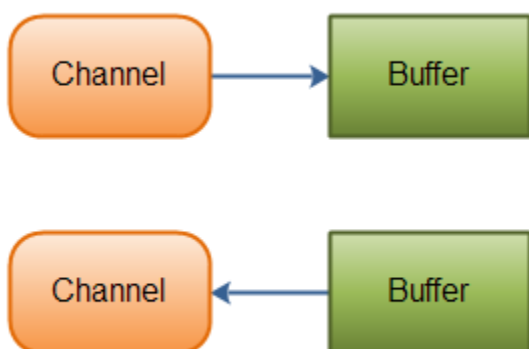
分享到：

27 译文出处：[airu](#) 原文出处：[Jakob Jenkov](#)

Java NIO的通道类似流，但又有些不同：

- 既可以从通道中读取数据，又可以写数据到通道。但流的读写通常是单向的。
- 通道可以异步地读写。
- 通道中的数据总是要先读到一个Buffer，或者总是要从一个Buffer中写入。

正如上面所说，从通道读取数据到缓冲区，从缓冲区写入数据到通道。如下图所示：



### Channel的实现

这些是Java NIO中最重要的通道的实现：

- `FileChannel`

- DatagramChannel
- SocketChannel
- ServerSocketChannel

FileChannel 从文件中读写数据。

DatagramChannel 能通过UDP读写网络中的数据。

SocketChannel 能通过TCP读写网络中的数据。

ServerSocketChannel可以监听新进来的TCP连接，像Web服务器那样。对每一个新进来的连接都会创建一个SocketChannel。

## 基本的 Channel 示例

下面是一个使用FileChannel读取数据到Buffer中的示例：

```
1 RandomAccessFile aFile = new RandomAccessFile("data/nio-data.txt", "rw");
2 FileChannel inChannel = aFile.getChannel();
3
4 ByteBuffer buf = ByteBuffer.allocate(48);
5
6 int bytesRead = inChannel.read(buf);
7 while (bytesRead != -1) {
8
9     System.out.println("Read " + bytesRead);
10    buf.flip();
11
12    while(buf.hasRemaining()){
13        System.out.print((char) buf.get());
14    }
15
16    buf.clear();
17    bytesRead = inChannel.read(buf);
18 }
19 aFile.close();
```



注意 buf.flip() 的调用，首先读取数据到Buffer，然后反转Buffer,接着再从Buffer中读取数据。下一节会深入讲解Buffer的更多细节。

## 本系列：

- [Java NIO系列教程（1）：Java NIO 概述](#)
- [Java NIO系列教程（2）：Channel](#)



## 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（8）：SocketChannel](#)
- [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](#)

## 发表评论

### Comment form

Name\*

邮箱\*

网站 (请以 http://开头)

评论内容\*



(\*) 表示必填项

[提交评论](#)

还没有评论。

[« Java NIO系列教程（1）：Java NIO 概述](#)

[Java NIO系列教程（3）：Buffer »](#)

Search for:



- [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](mailto:ImportNew.com@gmail.com)

新浪微博： [@ImportNew](#)

推荐微信号



反馈建议： [ImportNew.com@gmail.com](mailto: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

© 2017 ImportNew

