

java nio - channel

作者: [DASHU](#)

原文链接: <https://ld246.com/article/1365589884030>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在老的java io操作中，我们的io操作一般都是通过InputStream和OutputStream来实现。

但是在java nio中，我们的io操作，是通过channel来实现的。

channel可处于打开或关闭状态。所以，在不能确定channel是否处于打开状态的时候，可以使用Open()方法来判断。如果在channel关闭的时候进行io操作，则会抛出ClosedChannelException异常。

channel有几个用的比较多的子类：FileChannel、ServerSocketChannel和SocketChannel。

FileChannel

FileChannel是指向File文件的通道。

通常用得比较多的操作有：

- 对文件中绝对位置的字节进行读取或写入。
- 将文件中的某个区域直接映射到内存中；对于较大的文件，这通常比调用普通的 read 或 write 法更为高效。

通常FileChannel可以通过FileInputStream或者FileOutputStream的getChannel()方法来获取

FileChannel.map(FileChannel.MapMode mode, long position, long size)可以文件的指定位，映射到一个[Buffer](http://www.idashu.me/java-nio-buffer.html "java nio Buffer")里面。

FileChannel.read(ByteBuffer dst)读取信息。

FileChannel.write(ByteBuffer src)写入信息。

ServerSocketChannel

ServerSocketChannel通常用来侦听socket连接。

ServerSocketChannel对象一般由ServerSocketChannel.open()方法来获取。一般使用方法：

```
// 打开连接
ServerSocketChannel ssc = ServerSocketChannel.open();
// 设置为非阻塞
ssc.configureBlocking(false);
// 绑定ip、端口
ssc.socket().bind(new InetSocketAddress(port));
// 注册到selector中
ssc.register(selector, SelectionKey.OP_ACCEPT);
```

SocketChannel

SocketChannel其实就是代表某一个socket的channel。

SocketChannel对象通常可以有Socket.getChannel()方法或者ServerSocketChannel.accept()方法来获得。

前者代表从一个连接的Socket对象获取SocketChannel对象。

或者是ServerSocketChannel监听到socket连接，然后可以调用accept()方法来获取SocketChannel对象。

SocketChannel读写一般要配合Buffer，可以参考前一篇文章[Buffer](http://www.idashu.me/ava-nio-buffer.html "java nio Buffer")。