

《Netty 权威指南》—— NIO客户端序列图

步骤一：打开SocketChannel，绑定客户端本地地址（可选，默认系统会随机分配一个可用的本地地址），

示例代码如下：

```
1 SocketChannel clientChannel = SocketChannel.open();
```

步骤二：设置SocketChannel为非阻塞模式，同时设置客户端连接的TCP参数，示例代码如下：

```
1 clientChannel.configureBlocking(false);
2 socket.setReuseAddress(true);
3 socket.setReceiveBufferSize(BUFFER_SIZE);
4 socket.setSendBufferSize(BUFFER_SIZE);
```

步骤三：异步连接服务端，示例代码如下：

```
1 boolean connected = clientChannel.connect(new InetSocketAddress("ip",port));
```

步骤四：判断是否连接成功，如果连接成功，则直接注册读状态位到多路复用器中，如果当前没有连接成功（异步连接，返回false，说明客户端已经发送sync包，服务端没有返回ack包，物理链路还没有建立），示例代码如下：

```
1 if (connected)
2 {
3     clientChannel.register(selector, SelectionKey.OP_READ, ioHandler);
4 }
5 else
6 {
7     clientChannel.register(selector, SelectionKey.OP_CONNECT, ioHandler);
8 }
```

步骤五：向Reactor线程的多路复用器注册OP_CONNECT状态位，监听服务端的TCP ACK应答，示例代码如下：

```
1 clientChannel.register(selector, SelectionKey.OP_CONNECT, ioHandler);
```

步骤六：创建Reactor线程，创建多路复用器并启动线程，代码如下：

```
1 Selector selector = Selector.open();
2 New Thread(new ReactorTask()).start();
```

步骤七：多路复用器在线程run方法的无限循环体内轮询准备就绪的Key，代码如下：

```
1 int num = selector.select();
2 Set selectedKeys = selector.selectedKeys();
3 Iterator it = selectedKeys.iterator();
4 while (it.hasNext()) {
5     if (key.isConnectable())
6         //handlerConnect();
7 }
```

步骤九：判断连接结果，如果连接成功，注册读事件到多路复用器，示例代码如下：

```
1 if (channel.finishConnect())
2     registerRead();
```

步骤十：注册读事件到多路复用器：

```
1 clientChannel.register(selector, SelectionKey.OP_READ, ioHandler);
```

步骤十一：异步读客户端请求消息到缓冲区，示例代码如下：

```
1 int readNumber = channel.read(receivedBuffer);
```

步骤十二：对ByteBuffer进行编解码，如果有半包消息接收缓冲区Reset，继续读取后续的报文，将解码成功的消息封装成Task，投递到业务线程池中，进行业务逻辑编排，示例代码如下：

```
01 Object message = null;
02 while(buffer.hasRemain())
03 {
```

```

04     byteBuffer.mark();
05     Object message = decode(byteBuffer);
06     if (message == null)
07     {
08         byteBuffer.reset();
09         break;
10     }
11     messageList.add(message );
12 }
13 if (!byteBuffer.hasRemain())
14 byteBuffer.clear();
15 else
16     byteBuffer.compact();
17 if (messageList != null & !messageList.isEmpty())
18 {
19     for(Object messageE : messageList)
20         handlerTask(messageE);
21 }

```

步骤十三：将POJO对象encode成ByteBuffer，调用SocketChannel的异步write接口，将消息异步发送给客户端，示例代码如下：

```
1 socketChannel.write(buffer);
```

通过序列图和关键代码的解说，相信大家对创建NIO客户端程序有了一个初步的了解，下面，就跟随着我们的脚步继续看看如果使用NIO改造之前的时间服务器客户端TimeClient。



NIO客户端创建序列图