



链滴

6.Netty 初认识 -- 粘包拆包 LineBasedFrameDecoder

作者: [289306290](#)

原文链接: <https://ld246.com/article/1582885236240>

来源网站: [链滴](#)

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

SomeServer.java

```
package club.wujingjian.study;

import io.netty.bootstrap.ServerBootstrap;
import io.netty.channel.ChannelFuture;
import io.netty.channel.ChannelInitializer;
import io.netty.channel.nio.NioEventLoopGroup;
import io.netty.channel.socket.SocketChannel;
import io.netty.channel.socket.nio.NioServerSocketChannel;
import io.netty.handler.codec.LineBasedFrameDecoder;
import io.netty.handler.codec.string.StringDecoder;

public class SomeServer {
    public static void main(String[] args) throws InterruptedException {
        NioEventLoopGroup parentGroup = new NioEventLoopGroup();
        NioEventLoopGroup childGroup = new NioEventLoopGroup();

        try{
            ServerBootstrap bootstrap = new ServerBootstrap();
            bootstrap.group(parentGroup,childGroup)
                .channel(NioServerSocketChannel.class)
                .childHandler(new ChannelInitializer<SocketChannel>() {
                    @Override
                    protected void initChannel(SocketChannel ch) throws Exception {
                        //StringDecoder()字符串解码器,将Channel中的ByteBuf数据解码为String
                        //StringEncoder()字符串编码器,将要写入Channel中的String编码为ByteBuf
                        ch.pipeline()
                            .addLast(new LineBasedFrameDecoder(5120)) //5kb
                            .addLast(new StringDecoder())
                            .addLast(new SomeServerHandler());
                    }
                });
            ChannelFuture future = bootstrap.bind(8888).sync();
            System.out.println("服务器已启动");
            future.channel().closeFuture().sync();
        }finally {
            parentGroup.shutdownGracefully();
            childGroup.shutdownGracefully();
        }
    }
}
```

SomeServerHandler.java

```
package club.wujingjian.study;

import io.netty.channel.ChannelHandlerContext;
import io.netty.channel.SimpleChannelInboundHandler;

public class SomeServerHandler extends SimpleChannelInboundHandler<String> {
```

```

private int counter ;
@Override
protected void channelRead0(ChannelHandlerContext ctx, String msg) throws Exception {
    System.out.println("server端接收到的第[" + ++counter + "]" + msg);
}

@Override
public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) throws Exception {
    ctx.close();
}
}

```

SomeClient.java

```

package club.wujingjian.study.client;

import io.netty.bootstrap.Bootstrap;
import io.netty.channel.ChannelFuture;
import io.netty.channel.ChannelInitializer;
import io.netty.channel.nio.NioEventLoopGroup;
import io.netty.channel.socket.SocketChannel;
import io.netty.channel.socket.nio.NioSocketChannel;
import io.netty.handler.codec.string.StringEncoder;

public class SomeClient {

    public static void main(String[] args) throws InterruptedException {
        NioEventLoopGroup group = new NioEventLoopGroup();
        try {
            Bootstrap bootstrap = new Bootstrap();
            bootstrap.group(group)
                .channel(NioSocketChannel.class)
                .handler(new ChannelInitializer<SocketChannel>() {
                    @Override
                    protected void initChannel(SocketChannel ch) throws Exception {
                        ch.pipeline().addLast(new StringEncoder())
                            .addLast(new SomeClientHandler());
                    }
                });
            ChannelFuture future = bootstrap.connect("localhost", 8888).sync();
            future.channel().closeFuture().sync();
        } finally {
            group.shutdownGracefully();
        }
    }
}

```

SomeClientHandler.java

```

package club.wujingjian.study.client;

```

```

import io.netty.buffer.ByteBuf;
import io.netty.buffer.Unpooled;
import io.netty.channel.ChannelHandlerContext;
import io.netty.channel.ChannelInboundHandlerAdapter;

public class SomeClientHandler extends ChannelInboundHandlerAdapter {

    private String msg =
        "from server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的" +
        "应数据!ca18dff5-d312-452b-afe0-0fe6374657be" +
        "from server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的" +
        "应数据!ca18dff5-d312-452b-afe0-0fe6374657be" +
        "from server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的" +
        "应数据!ca18dff5-d312-452b-afe0-0fe6374657be" +
        "from server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的" +
        "应数据!ca18dff5-d312-452b-afe0-0fe6374657be" +
        "from server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的" +
        "应数据!ca18dff5-d312-452b-afe0-0fe6374657be" +
        "from server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的" +
        "应数据!ca18dff5-d312-452b-afe0-0fe6374657be" +
        System.getProperty("line.separator");

    @Override
    public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) throws Exception {
        System.out.println("客户端异常" + ctx.close());
    }

    //当Channel被激活会触发该方法的执行
    @Override
    public void channelActive(ChannelHandlerContext ctx) throws Exception {
        // ctx.writeAndFlush("1.from client: 客户端的channel被激活了,begin talking" + msg);
        // ctx.writeAndFlush("1.from client: 客户端的channel被激活了,begin talking" + msg);
        byte[] bytes = msg.getBytes();
        ByteBuf bufer = null;
        for (int i = 0; i < 2; i++) {
            //申请缓存空间
            bufer = Unpooled.buffer(bytes.length);
            // 将数据写入到缓存
            bufer.writeBytes(bytes);
            //将缓存中的数据写入到Channel
            ctx.writeAndFlush(bufer);
        }
    }
}

```

服务端会输出内容:

```
SomeServer (2) x SomeClient (1) x
/Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home/bin/java ...
服务器已启动
server端接收到的第[1]from server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657be
server端接收到的第[2]from server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657befrom server:来自于客户端localhost/127.0.0.1:8888 的消息:from server:这里是客户端的响应数据!ca18dff5-d312-452b-afe0-0fe6374657be
```