链滴

7.1 .Netty 初认识 -- 编解码 -Java 序列化

作者：289306290
原文链接：https://ld246.com/article/1583140910951
来源网站：链滴

场景：

客户端发送产品订购信息给服务端，服务端收到请求信息进行简单的验证，然后给客户端返回对应的息。

订阅请求体信息如下:

字段名称	字段类型	备注
subReqID	整型	订购编号
userName	字符串	用户名
productName	字符串	订购的产名称
phoneNumber	字符串	订购者话号码
address	字符串	订购者的家庭住址

响应体信息如下:

字段名称	字段类型	备注
subReqId	整型	订购编号
respCode	整型	订购结果:0标识功
desc	字符串	可选的详细描述信息

SubScribeReq.java

```
package club.wujingjian.com.wujingjian.netty.serial.java.vo;

import java.io.Serializable;

public class SubscribeReq implements Serializable {

    private static final long serialVersionUID = -6751748286672176249L;
    private int subReqID;

    private String userName;

    private String productName;

    private String phoneNumber;

    private String address;

    public int getSubReqID() {
        return subReqID;
    }

    public void setSubReqID(int subReqID) {
        this.subReqID = subReqID;
    }
```

```
    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public String getProductName() {
        return productName;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    @Override
    public String toString() {
        return "SubscribeReq{" +
                "subReqID=" + subReqID +
                ", userName='" + userName + '\'' +
                ", productName='" + productName + '\'' +
                ", phoneNumber='" + phoneNumber + '\'' +
                ", address='" + address + '\'' +
                '}';
    }
}
```

SubScribeResp.java

```
package club.wujingjian.com.wujingjian.netty.serial.java.vo;

import java.io.Serializable;

public class SubscribeResp implements Serializable {
    private static final long serialVersionUID = 5777787033239807714L;
```

```
    private int subReqId;

    private int respCode;

    private String desc;

    public int getSubReqId() {
        return subReqId;
    }

    public void setSubReqId(int subReqId) {
        this.subReqId = subReqId;
    }

    public int getRespCode() {
        return respCode;
    }

    public void setRespCode(int respCode) {
        this.respCode = respCode;
    }

    public String getDesc() {
        return desc;
    }

    public void setDesc(String desc) {
        this.desc = desc;
    }

    @Override
    public String toString() {
        return "SubscribeResp{" +
                "subReqId=" + subReqId +
                ", respCode=" + respCode +
                ", desc='" + desc + '\'' +
                '}';
    }
}
```

示例:

SubReqServer.java

```
package club.wujingjian.com.wujingjian.netty.serial.java.server;

import io.netty.bootstrap.ServerBootstrap;
import io.netty.channel.ChannelFuture;
import io.netty.channel.ChannelInitializer;
import io.netty.channel.ChannelOption;
import io.netty.channel.EventLoopGroup;
import io.netty.channel.nio.NioEventLoopGroup;
```

```
import io.netty.channel.socket.SocketChannel;
import io.netty.channel.socket.nio.NioServerSocketChannel;
import io.netty.handler.codec.serialization.ClassResolvers;
import io.netty.handler.codec.serialization.ObjectDecoder;
import io.netty.handler.codec.serialization.ObjectEncoder;
import io.netty.handler.logging.LogLevel;
import io.netty.handler.logging.LoggingHandler;


public class SubReqServer {

    public void bind(int port) throws  Exception {
        EventLoopGroup bossGroup = new NioEventLoopGroup();
        EventLoopGroup workerGroup = new NioEventLoopGroup();
        try {
            ServerBootstrap b = new ServerBootstrap();
            b.group(bossGroup,workerGroup)
                    .channel(NioServerSocketChannel.class)
                    .option(ChannelOption.SO_BACKLOG, 100)
                    .handler(new LoggingHandler(LogLevel.INFO))
                    .childHandler(new ChannelInitializer<SocketChannel>() {
                        @Override
                        protected void initChannel(SocketChannel socketChannel) throws Exception {
                            socketChannel.pipeline().addLast(new ObjectDecoder(1024 * 1024, ClassReso
vers.weakCachingConcurrentResolver(this.getClass().getClassLoader())));
                            socketChannel.pipeline().addLast(new ObjectEncoder());
                            socketChannel.pipeline().addLast(new SubReqServerHandler());
                        }
                    });
            ChannelFuture f = b.bind(port).sync();
            f.channel().closeFuture().sync();
        } finally {
            bossGroup.shutdownGracefully();
            workerGroup.shutdownGracefully();
        }
    }

    public static void main(String[] args) throws Exception {
        int port = 8080;
        if (args != null && args.length > 0) {
            port = Integer.parseInt(args[0]);
        }
        new SubReqServer().bind(port);

    }
}
```

SubReqServerHandler.java

```
package club.wujingjian.com.wujingjian.netty.serial.java.server;

import club.wujingjian.com.wujingjian.netty.serial.java.vo.SubscribeReq;
import club.wujingjian.com.wujingjian.netty.serial.java.vo.SubscribeResp;
```

```
import io.netty.channel.ChannelHandler;
import io.netty.channel.ChannelHandlerAdapter;
import io.netty.channel.ChannelHandlerContext;

@ChannelHandler.Sharable
public class SubReqServerHandler extends ChannelHandlerAdapter {

    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
        SubscribeReq req = (SubscribeReq) msg;
        if ("wujingjian".equalsIgnoreCase(req.getUserName())) {
            System.out.println("Service accept client subscribe req :[" + req.toString() + "]");
            ctx.writeAndFlush(resp(req.getSubReqID()));
        }
    }

    private SubscribeResp resp(int subReqID) {
        SubscribeResp resp = new SubscribeResp();
        resp.setSubReqId(subReqID);
        resp.setRespCode(0);
        resp.setDesc("Netty book order succeed, 3 days later, sent to the designated address");
        return resp;
    }

    @Override
    public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) throws Excepti
n {
        cause.printStackTrace();
        ctx.close();
    }
}
```

SubReqClient.java

```
package club.wujingjian.com.wujingjian.netty.serial.java.client;

import io.netty.bootstrap.Bootstrap;
import io.netty.channel.ChannelFuture;
import io.netty.channel.ChannelInitializer;
import io.netty.channel.ChannelOption;
import io.netty.channel.EventLoopGroup;
import io.netty.channel.nio.NioEventLoopGroup;
import io.netty.channel.socket.SocketChannel;
import io.netty.channel.socket.nio.NioSocketChannel;
import io.netty.handler.codec.serialization.ClassResolvers;
import io.netty.handler.codec.serialization.ObjectDecoder;
import io.netty.handler.codec.serialization.ObjectEncoder;

public class SubReqClient {
    public void connect(int port,String host) throws Exception {
        EventLoopGroup group = new NioEventLoopGroup();
        try {
            Bootstrap b = new Bootstrap();
```

```
            b.group(group)
                    .channel(NioSocketChannel.class)
                    .option(ChannelOption.TCP_NODELAY,true)
//                    .option(ChannelOption.SO_BACKLOG,100)
                    .handler(new ChannelInitializer<SocketChannel>() {
                        @Override
                        protected void initChannel(SocketChannel socketChannel) throws Exception {
                            socketChannel.pipeline().addLast(new ObjectDecoder(1024, ClassResolvers.c
cheDisabled(this.getClass().getClassLoader())));
                            socketChannel.pipeline().addLast(new ObjectEncoder());
                            socketChannel.pipeline().addLast(new SubReqClientHandler());
                        }
                    });
            ChannelFuture f = b.connect(host, port).sync();

            f.channel().closeFuture().sync();
        } finally {
            group.shutdownGracefully();
        }
    }

    public static void main(String[] args) throws Exception {
        int port = 8080;
        if (args != null && args.length > 0) {
            port = Integer.parseInt(args[0]);
        }
        new SubReqClient().connect(port, "127.0.0.1");

    }

}
```

SubReqClientHandler.java

```
package club.wujingjian.com.wujingjian.netty.serial.java.client;

import club.wujingjian.com.wujingjian.netty.serial.java.vo.SubscribeReq;
import io.netty.channel.ChannelHandlerAdapter;
import io.netty.channel.ChannelHandlerContext;

public class SubReqClientHandler extends ChannelHandlerAdapter {

    public SubReqClientHandler(){}

    @Override
    public void channelActive(ChannelHandlerContext ctx) throws Exception {
        for (int i = 0; i < 10; i++) {
            ctx.write(subReq(i));
        }
        ctx.flush();
    }

    private SubscribeReq subReq(int i) {
```

```
        SubscribeReq req = new SubscribeReq();
        req.setAddress("北京市东城区秘密办公点");
        req.setPhoneNumber("17311112223");
        req.setProductName("Netty 权威指南");
        req.setSubReqID(i);
        req.setUserName("wujingjian");
        return req;
    }

    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
        System.out.println("Receive server response :[" + msg +" ]");
    }

    @Override
    public void channelReadComplete(ChannelHandlerContext ctx) throws Exception {
        ctx.flush();
    }

    @Override
    public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) throws Excepti
n {
        cause.printStackTrace();;
        ctx.close();
    }
}
```

服务端输出:

```
SubReqServer   SubReqClient
三月 02, 2020 5:08:59 下午 io.netty.handler.logging.LoggingHandler channelActive
信息: [id: 0x71c1508f, /0:0:0:0:0:0:0:0:8080] ACTIVE
三月 02, 2020 5:09:04 下午 io.netty.handler.logging.LoggingHandler channelRead
信息: [id: 0x71c1508f, /0:0:0:0:0:0:0:0:8080] RECEIVED: [id: 0xd2204196, /127.0.0.1:64330 => /127.0.0.1:8080]
Service accept client subscribe req :[SubscribeReq{subReqID=0, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=1, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=2, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=3, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=4, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=5, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=6, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=7, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=8, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
Service accept client subscribe req :[SubscribeReq{subReqID=9, userName='wujingjian', productName='Netty 权威指南', phoneNumber='17311112223', address='北京市东城区秘密办公点'}]
```

客户端输出:

```
Run:  SubReqServer   SubReqClient
/Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home/bin/java ...
Receive server response :[SubscribeResp{subReqId=0, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=1, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=2, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=3, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=4, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=5, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=6, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=7, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=8, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
Receive server response :[SubscribeResp{subReqId=9, respCode=0, desc='Netty book order succeed, 3 days later, sent to the designated address'} ]
```