**一、创建服务端**

package com.qiqi.netty;

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.string.StringDecoder;

import io.netty.handler.codec.string.StringEncoder;

import java.net.InetSocketAddress;

public class HelloWorldServer {

private int port;

public HelloWorldServer(int port) {

this.port = port;

}

public void start(){

EventLoopGroup bossGroup = new NioEventLoopGroup(1);

EventLoopGroup workerGroup = new NioEventLoopGroup();

try {

ServerBootstrap sbs = new ServerBootstrap().group(bossGroup,workerGroup).channel(NioServerSocketChannel.class).localAddress(new InetSocketAddress(port))

.childHandler(new ChannelInitializer<SocketChannel>() {

protected void initChannel(SocketChannel ch) throws Exception {

// ch.pipeline().addLast("framer", new DelimiterBasedFrameDecoder(8192, Delimiters.lineDelimiter()));

ch.pipeline().addLast("decoder", new StringDecoder());

ch.pipeline().addLast("encoder", new StringEncoder());

ch.pipeline().addLast(new HelloWorldServerHandler());

};

}).option(ChannelOption.SO\_BACKLOG, 128)

.childOption(ChannelOption.SO\_KEEPALIVE, true);

// 绑定端口，开始接收进来的连接

ChannelFuture future = sbs.bind(port).sync();

System.out.println("Server start listen at " + port );

future.channel().closeFuture().sync();

} catch (Exception e) {

bossGroup.shutdownGracefully();

workerGroup.shutdownGracefully();

}

}

public static void main(String[] args) throws Exception {

int port;

if (args.length > 0) {

port = Integer.parseInt(args[0]);

} else {

port = 8080;

}

new HelloWorldServer(port).start();

}

}

**二、创建服务接收实现该类**

package com.qiqi.netty;

import io.netty.channel.ChannelHandlerContext;

import io.netty.channel.ChannelInboundHandlerAdapter;

public class HelloWorldServerHandler extends ChannelInboundHandlerAdapter{

@Override

public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {

System.out.println("server channelRead..");

System.out.println(ctx.channel().remoteAddress()+"->Server :"+ msg.toString());

ctx.write("server write"+msg);

ctx.flush();

}

@Override

public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) throws Exception {

cause.printStackTrace();

ctx.close();

}

}

**三、创建客户端发送消息**

package com.qiqi.netty;

import io.netty.bootstrap.Bootstrap;

import io.netty.channel.ChannelFuture;

import io.netty.channel.ChannelInitializer;

import io.netty.channel.ChannelOption;

import io.netty.channel.ChannelPipeline;

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.string.StringDecoder;

import io.netty.handler.codec.string.StringEncoder;

public class HelloWorldClient {

static final String HOST = System.getProperty("host", "127.0.0.1");

static final int PORT = Integer.parseInt(System.getProperty("port", "8080"));

static final int SIZE = Integer.parseInt(System.getProperty("size", "256"));

public static void main(String[] args) throws Exception {

// Configure the client.

EventLoopGroup group = new NioEventLoopGroup();

try {

Bootstrap b = new Bootstrap();

b.group(group)

.channel(NioSocketChannel.class)

.option(ChannelOption.TCP\_NODELAY, true)

.handler(new ChannelInitializer<SocketChannel>() {

@Override

public void initChannel(SocketChannel ch) throws Exception {

ChannelPipeline p = ch.pipeline();

p.addLast("decoder", new StringDecoder());

p.addLast("encoder", new StringEncoder());

p.addLast(new HelloWorldClientHandler());

}

});

ChannelFuture future = b.connect(HOST, PORT).sync();

future.channel().writeAndFlush("我是客户端，我发送给服务端一条信息");

future.channel().closeFuture().sync();

} finally {

group.shutdownGracefully();

}

}

}

**四、创建客户端接收消息的实现该接口**

package com.qiqi.netty;

import io.netty.channel.ChannelHandlerContext;

import io.netty.channel.ChannelInboundHandlerAdapter;

public class HelloWorldClientHandler extends ChannelInboundHandlerAdapter{

@Override

public void channelActive(ChannelHandlerContext ctx) {

System.out.println("HelloWorldClientHandler Active");

}

@Override

public void channelRead(ChannelHandlerContext ctx, Object msg) {

System.out.println("HelloWorldClientHandler read Message:"+msg);

}

@Override

public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) {

cause.printStackTrace();

ctx.close();

}

}