netty tcp □□ □□

```
    <dependencies>
        <dependency>
            <groupId>io.netty</groupId>
            <artifactId>netty-all</artifactId>
            <version>4.1.68.Final</version>
        </dependency>
        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-api</artifactId>
            <version>1.7.30</version>
        </dependency>
        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-simple</artifactId>
            <version>1.7.30</version>
        </dependency>
    </dependencies>
```

```
public class NettyServer {

    private final int port;

    public NettyServer(int port) {
        this.port = port;
    }

    public void start() throws InterruptedException {
        EventLoopGroup bossGroup = new NioEventLoopGroup(1);
        EventLoopGroup workerGroup = new NioEventLoopGroup();

        try {
            ServerBootstrap b = new ServerBootstrap();
            b.group(bossGroup, workerGroup)
                    .channel(NioServerSocketChannel.class)
                    .handler(new LoggingHandler(LogLevel.INFO))
```

```
                .childHandler(new ChannelInitializer<SocketChannel>() {
                    @Override
                    public void initChannel(SocketChannel ch) {
                        ChannelPipeline p = ch.pipeline();
                        p.addLast(new LoggingHandler(LogLevel.INFO));
                        p.addLast(new NettyServerHandler());
                    }
                });

            ChannelFuture f = b.bind(port).sync();
            f.channel().closeFuture().sync();
        } finally {
            bossGroup.shutdownGracefully();
            workerGroup.shutdownGracefully();
        }
    }

    public static void main(String[] args) throws InterruptedException {
        int port = 6601; // 포트를 원하는 값으로 설정
        new NettyServer(port).start();
    }
}
```

```
public class NettyServerHandler extends ChannelInboundHandlerAdapter {

    private static final Logger logger = LoggerFactory.getLogger(NettyServerHandler.class);

    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) {
        // 수신된 메시지를 로깅
        logger.info("Received message: {}", msg);
        // 여기서 원하는 처리를 할 수 있습니다
    }

    @Override
    public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) {
        logger.error("Error occurred", cause);
        ctx.close();
    }
}
```

Revision #1
Created 26 July 2024 07:35:23 by Yeonwoo Kim
Updated 26 July 2024 07:36:13 by Yeonwoo Kim