

AI 핑퐁 TCP Gateway 서버 개발기 with Netty

AI 펭톡?



다룰 내용

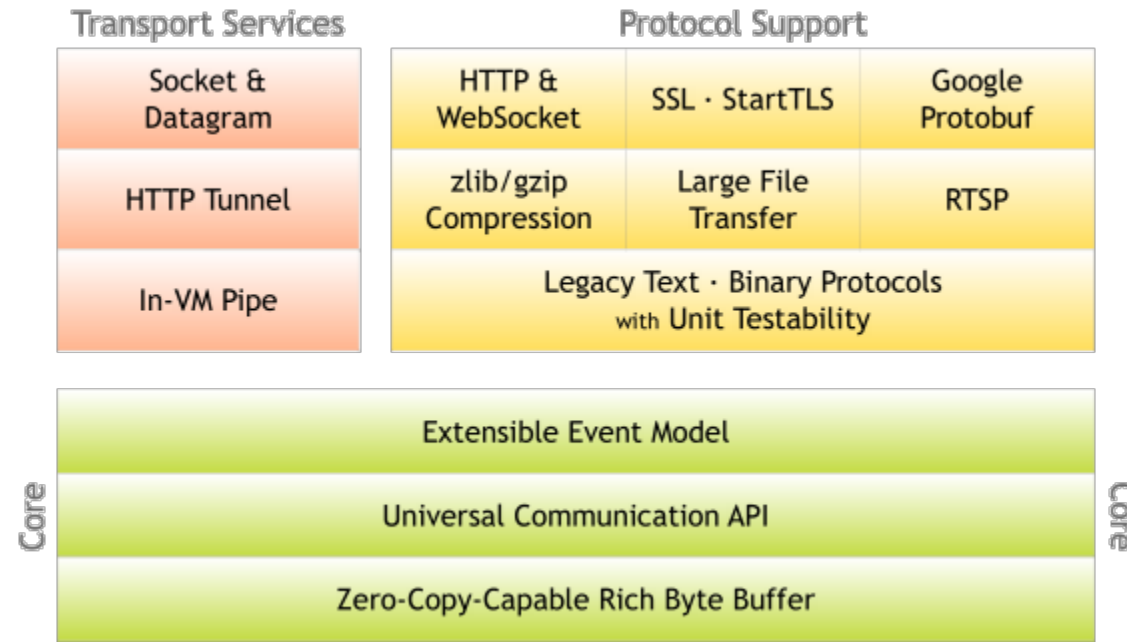
1. Netty가 뭐지?
2. 왜 Netty를 써야 할까?
3. Do it Netty

Netty가 뭐지?

Netty가 뭐지?



Netty가 뭐지?



Netty가 뭐지?



kakao

Pivotal

Google



LINE

NETFLIX



왜 Netty를 써야 할까?

왜 Netty를 써야 할까?

Java Socket - Packet

Type Unsigned 1Byte	Length Unsigned 2Byte	Data String(~65535Byte)
------------------------	--------------------------	----------------------------

- Type
 - 사용자: 1
 - 서버: 2
- Length: Data의 길이
- Data: 문자열
 - 서버는 사용자가 전송한 문자열 앞에 Echo를 붙여 응답한다.

왜 Netty를 써야 할까?

Java Socket – Bind & Accept

```
...
try (ServerSocket server = new ServerSocket()) {
    server.bind(new InetSocketAddress(10000));
    while (true) {
        Socket client = server.accept();
        new Thread(() -> {
            try (OutputStream sender = client.getOutputStream();
                InputStream receiver = client.getInputStream()) {
            ...

```

왜 Netty를 써야 할까?

Java Socket – Bind & Accept

```
...  
try (ServerSocket server = new ServerSocket()) {  
server.bind(new InetSocketAddress(10000));  
while (true) {  
    Socket client = server.accept();  
    new Thread(() -> {  
        try (OutputStream sender = client.getOutputStream();  
            InputStream receiver = client.getInputStream()) {  
            ...  
        }  
    }  
}
```

왜 Netty를 써야 할까?

Java Socket – Bind & Accept

```
...
try (ServerSocket server = new ServerSocket()) {
    server.bind(new InetSocketAddress(10000));
    while (true) {
        Socket client = server.accept();
        new Thread(() -> {
            try (OutputStream sender = client.getOutputStream();
                InputStream receiver = client.getInputStream()) {
...

```

왜 Netty를 써야 할까?

Java Socket – Bind & Accept

```
...
try (ServerSocket server = new ServerSocket()) {
    server.bind(new InetSocketAddress(10000));
    while (true) {
        Socket client = server.accept();
        new Thread(() -> {
            try (OutputStream sender = client.getOutputStream();
                InputStream receiver = client.getInputStream();) {
            ...

```

왜 Netty를 써야 할까?

Java Socket - Read

```
...  
byte[] bytes = new byte[3 + 65535];  
receiver.read(bytes, 0, 3);  
short type = (short) (bytes[0] & 0xff);  
int length = ((bytes[1] & 0xff) << 8) + (bytes[2] & 0xff);  
receiver.read(bytes, 3, length);  
String data = new String(bytes, 3, length);  
...
```

왜 Netty를 써야 할까?

Java Socket - Read

```
...  
byte[] bytes = new byte[3 + 65535];  
receiver.read(bytes, 0, 3);  
short type = (short) (bytes[0] & 0xff);  
int length = ((bytes[1] & 0xff) << 8) + (bytes[2] & 0xff);  
receiver.read(bytes, 3, length);  
String data = new String(bytes, 3, length);  
...
```

왜 Netty를 써야 할까?

Java Socket - Read

```
...  
byte[] bytes = new byte[3 + 65535];  
receiver.read(bytes, 0, 3);  
short type = (short) (bytes[0] & 0xff);  
int length = ((bytes[1] & 0xff) << 8) + (bytes[2] & 0xff);  
receiver.read(bytes, 3, length);  
String data = new String(bytes, 3, length);  
...
```


왜 Netty를 써야 할까?

Java Socket - Read

```
...  
byte[] bytes = new byte[3 + 65535];  
receiver.read(bytes, 0, 3);  
short type = (short) (bytes[0] & 0xff);  
int length = ((bytes[1] & 0xff) << 8) + (bytes[2] & 0xff);  
receiver.read(bytes, 3, length);  
String data = new String(bytes, 3, length);  
...
```

왜 Netty를 써야 할까?

Java Socket - Write

```
...
String sendData = "Echo - " + data;
byte[] sendBytes = new byte[3 + sendData.length()];
sendBytes[0] = (byte) (2 & 0xff);
sendBytes[1] = (byte) ((sendData.length() >> 8) & 0xff);
sendBytes[2] = (byte) (sendData.length() & 0xff);
System.arraycopy(sendData.getBytes(), 0, sendBytes, 3, sendData.length());
sender.write(sendBytes);
sender.flush();
...
```

왜 Netty를 써야 할까?

Java Socket - Write

```
...  
String sendData = "Echo - " + data;  
byte[] sendBytes = new byte[3 + sendData.length()];  
sendBytes[0] = (byte) (2 & 0xff);  
sendBytes[1] = (byte) ((sendData.length() >> 8) & 0xff);  
sendBytes[2] = (byte) (sendData.length() & 0xff);  
System.arraycopy(sendData.getBytes(), 0, sendBytes, 3, sendData.length());  
sender.write(sendBytes);  
sender.flush();  
...
```

왜 Netty를 써야 할까?

Java Socket - Write

```
...  
String sendData = "Echo - " + data;  
byte[] sendBytes = new byte[3 + sendData.length()];  
sendBytes[0] = (byte) (2 & 0xff);  
sendBytes[1] = (byte) ((sendData.length() >> 8) & 0xff);  
sendBytes[2] = (byte) (sendData.length() & 0xff);  
System.arraycopy(sendData.getBytes(), 0, sendBytes, 3, sendData.length());  
sender.write(sendBytes);  
sender.flush();  
...
```

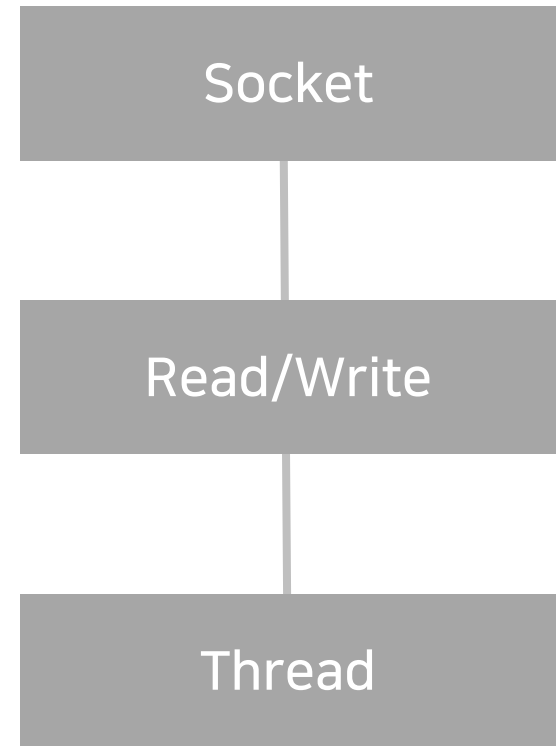
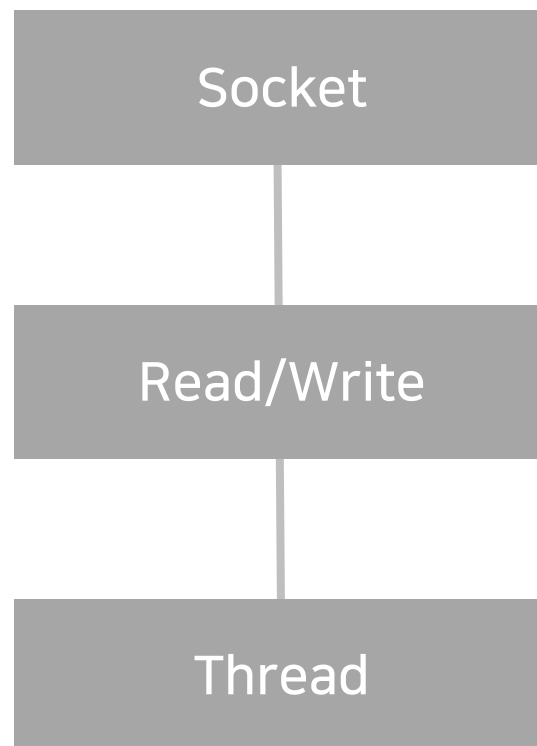
왜 Netty를 써야 할까?

Java Socket - Write

```
...  
String sendData = "Echo - " + data;  
byte[] sendBytes = new byte[3 + sendData.length()];  
sendBytes[0] = (byte) (2 & 0xff);  
sendBytes[1] = (byte) ((sendData.length() >> 8) & 0xff);  
sendBytes[2] = (byte) (sendData.length() & 0xff);  
System.arraycopy(sendData.getBytes(), 0, sendBytes, 3, sendData.length());  
sender.write(sendBytes);  
sender.flush();  
...
```

왜 Netty를 써야 할까?

스레드 부자



왜 Netty를 써야 할까?

불편한 Read & Write

```
...  
//Read  
int length = ((bytes[1] & 0xff) << 8) + (bytes[2] & 0xff);  
  
...  
//Write  
sendBytes[1] = (byte) ((sendData.length() >> 8) & 0xff);  
sendBytes[2] = (byte) (sendData.length() & 0xff);  
...
```

Do it Netty

The background of the slide is an abstract composition of overlapping, semi-transparent blue shapes. These shapes, which include rectangles and trapezoids, are arranged in a way that creates a sense of depth and movement, appearing to recede into the distance from the bottom-left towards the top-right. The colors range from a light, airy blue to a deep, vibrant royal blue.

Do it Netty

구성



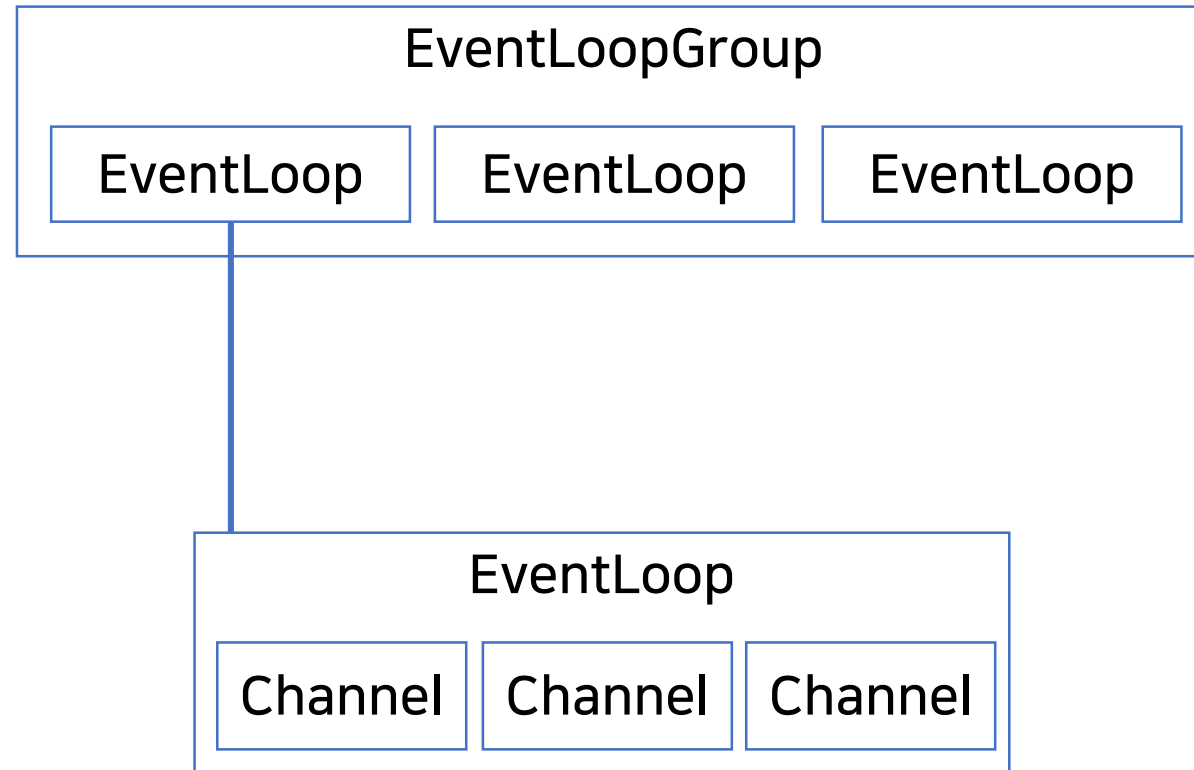
Do it Netty

Channel, EventLoop, EventLoopGroup

- Channel
 - I/O(bind, connect, read, write) 작업을 수행하는 요소 또는 네트워크 연결
- EventLoop
 - Channel의 I/O를 처리, 수명주기 동안 한 Thread에 바인딩
 - Channel은 생명주기 동안 하나의 EventLoop에 바인딩
 - EventLoop는 여러 Channel을 할당받아 처리할 수 있음
- EventLoopGroup
 - N 개의 EventLoop를 포함한 그룹

Do it Netty

Channel, EventLoop, EventLoopGroup



Do it Netty

ChannelHandler

- Channel의 I/O를 처리
- ChannelInboundHandler, ChannelOutboundHandler

Do it Netty

ChannelInboundHandler

- Channel의 입력 데이터를 처리
- Method list
 - channelRegistered
 - channelUnregistered
 - channelActive
 - channelInactive
 - channelReadComplete
 - channelRead
 - userEventTriggered
 - channelWritabilityChanged
 - exceptionCaught

Do it Netty

ChannelOutboundHandler

- Channel의 출력 데이터를 처리
- Method list
 - bind
 - connect
 - disconnect
 - close
 - deregister
 - read
 - write
 - flush

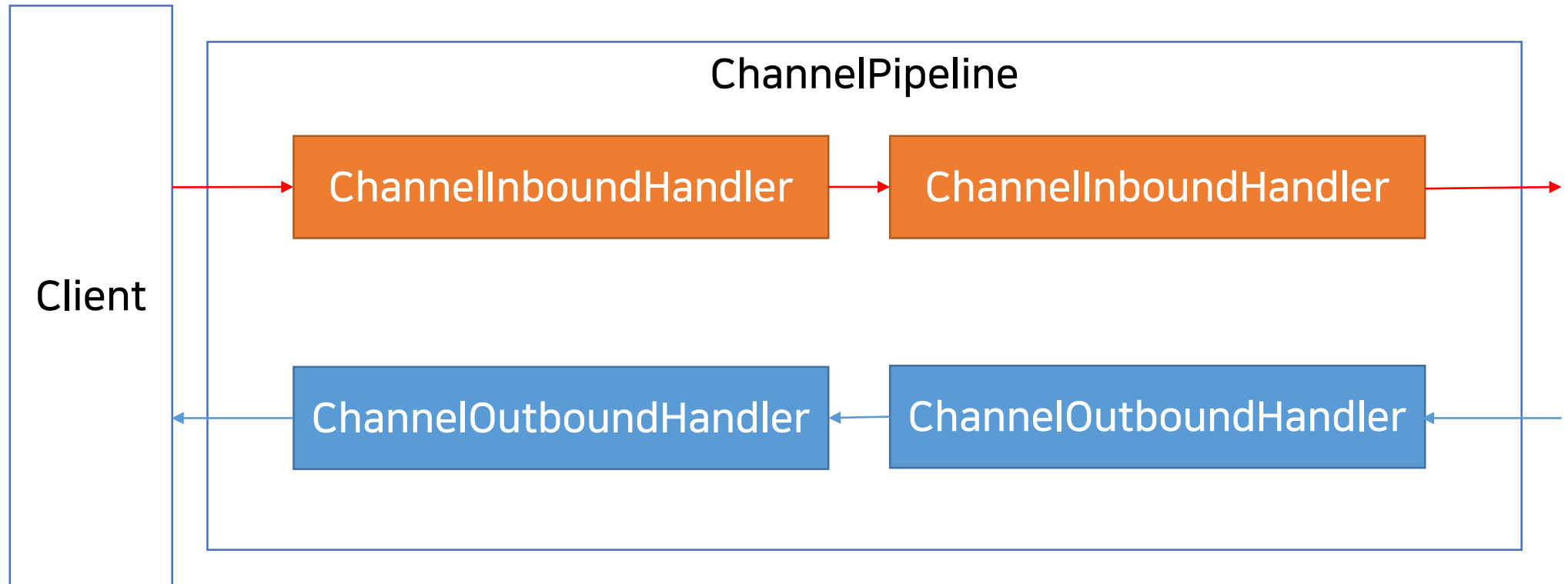
Do it Netty

ChannelPipeline, ChannelHandlerContext, ChannelInitializer

- ChannelPipeline
 - ChannelHandler 이벤트 체인 관리
 - Channel 생성 시 자동으로 할당
- ChannelHandlerContext
 - ChannelHandler와 ChannelPipeline 간 연결을 나타냄
 - ChannelHandler를 ChannelPipeline에 추가 시 할당
- ChannelInitializer
 - 여러 ChannelHandler를 ChannelPipeline에 할당하기 위한 클래스
 - Channel 생성 시 호출

Do it Netty

ChannelPipeline 흐름



Do it Netty

Bootstrap, ServerBootstrap

- 애플리케이션의 동작 및 설정을 담당하는 헬퍼 클래스
 - EventLoopGroup
 - 소켓 입출력 모드
 - EventHandler
 - 서버 소켓 옵션 설정
- Bootstrap = Client
- ServerBootstrap = Server

Do it Netty

Engine Server

```
...
EventLoopGroup parentGroup = new NioEventLoopGroup();
EventLoopGroup childGroup = new NioEventLoopGroup();
ServerBootstrap serverBootstrap = new ServerBootstrap();
serverBootstrap.group(parentGroup, childGroup)
    .channel(NioServerSocketChannel.class)
    .childHandler(new ChannelInitializer<SocketChannel>() {
        @Override
        protected void initChannel(SocketChannel ch) throws Exception {
            ch.pipeline().addLast(new IdleStateHandler(30, 0, 0))
                .addLast(new PacketDecoder())
                .addLast(new EngineInboundHandler())
                .addLast(new PacketEncoder());
        }
    });
serverBootstrap.bind(10000);
...
```

Do it Netty

Engine Server

```
...
EventLoopGroup parentGroup = new NioEventLoopGroup();
EventLoopGroup childGroup = new NioEventLoopGroup();
ServerBootstrap serverBootstrap = new ServerBootstrap();
serverBootstrap.group(parentGroup, childGroup)
    .channel(NioServerSocketChannel.class)
    .childHandler(new ChannelInitializer<SocketChannel>() {
        @Override
        protected void initChannel(SocketChannel ch) throws Exception {
            ch.pipeline().addLast(new IdleStateHandler(30, 0, 0))
                .addLast(new PacketDecoder())
                .addLast(new EngineInboundHandler())
                .addLast(new PacketEncoder());
        }
    });
serverBootstrap.bind(10000);
...
```

Do it Netty

Engine Server

```
...
EventLoopGroup parentGroup = new NioEventLoopGroup();
EventLoopGroup childGroup = new NioEventLoopGroup();
ServerBootstrap serverBootstrap = new ServerBootstrap();
serverBootstrap.group(parentGroup, childGroup)
.channel(NioServerSocketChannel.class)
.childHandler(new ChannelInitializer<SocketChannel>() {
    @Override
    protected void initChannel(SocketChannel ch) throws Exception {
        ch.pipeline().addLast(new IdleStateHandler(30, 0, 0))
        .addLast(new PacketDecoder())
        .addLast(new EngineInboundHandler())
        .addLast(new PacketEncoder());
    }
});
serverBootstrap.bind(10000);
...
```

Do it Netty

Engine Server

```
...
EventLoopGroup parentGroup = new NioEventLoopGroup();
EventLoopGroup childGroup = new NioEventLoopGroup();
ServerBootstrap serverBootstrap = new ServerBootstrap();
serverBootstrap.group(parentGroup, childGroup)
    .channel(NioServerSocketChannel.class)
    .childHandler(new ChannelInitializer<SocketChannel>() {
        @Override
        protected void initChannel(SocketChannel ch) throws Exception {
            ch.pipeline().addLast(new IdleStateHandler(30, 0, 0))
                .addLast(new PacketDecoder())
                .addLast(new EngineInboundHandler())
                .addLast(new PacketEncoder());
        }
    });
serverBootstrap.bind(10000);
...
```

Do it Netty

IdleStateHandler

- Channel의 I/O 수행 여부 감지
- 설정 시간이 지나면 IdleStateEvent 발생

Do it Netty

EngineInboundHandler

```
public class EngineInboundHandler extends ChannelInboundHandlerAdapter {
    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
        Packet readPacket = (Packet) msg;
        String writeData = "Echo - " + readPacket.getData();
        Packet writePacket = new Packet((short) 2, writeData.length(), writeData);
        ctx.pipeline().writeAndFlush(writePacket);
    }
    @Override
    public void userEventTriggered(ChannelHandlerContext ctx, Object evt) throws Exception {
        if (evt instanceof IdleStateEvent &&
            ((IdleStateEvent) evt).state() == IdleState.READER_IDLE) {
            ctx.close();
        }
    }
}
```

Do it Netty

EngineInboundHandler

```
public class EngineInboundHandler extends ChannelInboundHandlerAdapter {  
    @Override  
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {  
        Packet readPacket = (Packet) msg;  
        String writeData = "Echo - " + readPacket.getData();  
        Packet writePacket = new Packet((short) 2, writeData.length(), writeData);  
        ctx.pipeline().writeAndFlush(writePacket);  
    }  
    @Override  
    public void userEventTriggered(ChannelHandlerContext ctx, Object evt) throws Exception {  
        if (evt instanceof IdleStateEvent &&  
            ((IdleStateEvent) evt).state() == IdleState.READER_IDLE) {  
            ctx.close();  
        }  
    }  
}
```


Do it Netty

EngineInboundHandler

```
public class EngineInboundHandler extends ChannelInboundHandlerAdapter {  
    @Override  
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {  
        Packet readPacket = (Packet) msg;  
        String writeData = "Echo - " + readPacket.getData();  
        Packet writePacket = new Packet((short) 2, writeData.length(), writeData);  
        ctx.pipeline().writeAndFlush(writePacket);  
    }  
    @Override  
    public void userEventTriggered(ChannelHandlerContext ctx, Object evt) throws Exception {  
        if (evt instanceof IdleStateEvent &&  
            ((IdleStateEvent) evt).state() == IdleState.READER_IDLE) {  
            ctx.close();  
        }  
    }  
}
```

Do it Netty

EngineInboundHandler

```
public class EngineInboundHandler extends ChannelInboundHandlerAdapter {
    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
        Packet readPacket = (Packet) msg;
        String writeData = "Echo - " + readPacket.getData();
        Packet writePacket = new Packet((short) 2, writeData.length(), writeData);
        ctx.pipeline().writeAndFlush(writePacket);
    }
    @Override
    public void userEventTriggered(ChannelHandlerContext ctx, Object evt) throws Exception {
        if (evt instanceof IdleStateEvent &&
            ((IdleStateEvent) evt).state() == IdleState.READER_IDLE) {
            ctx.close();
        }
    }
}
```

Do it Netty

EngineInboundHandler

```
public class EngineInboundHandler extends ChannelInboundHandlerAdapter {
    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
        Packet readPacket = (Packet) msg;
        String writeData = "Echo - " + readPacket.getData();
        Packet writePacket = new Packet((short) 2, writeData.length(), writeData);
        ctx.pipeline().writeAndFlush(writePacket);
    }
    @Override
    public void userEventTriggered(ChannelHandlerContext ctx, Object evt) throws Exception {
        if (evt instanceof IdleStateEvent &&
            ((IdleStateEvent) evt).state() == IdleState.READER_IDLE) {
            ctx.close();
        }
    }
}
```

Do it Netty

ByteBuffer

- Java NIO ByteBuffer를 대체하는 데이터 처리 구현
 - readIndex, writeIndex 이용 데이터 접근 제어
 - 가변 사이즈
 - Pool 지원
 - Heap 영역 밖에 있는 direct buffer 지원
 - 다양한 read, write Method 지원
- 사용 후 release 필요

Do it Netty

Codec

- 네트워크상 데이터를 애플리케이션 내 원하는 데이터로 변환
 - Encoder
 - Decoder
- Http, WebSocket, Base64, Json, Protobuf ...

Do it Netty

Codec - ByteToMessageDecoder

```
public class PacketDecoder extends ByteToMessageDecoder {
    @Override
    protected void decode(ChannelHandlerContext ctx, ByteBuf in, List<Object> out) throws Exception {
        in.markReaderIndex();
        if (in.isReadable(3)) {
            short type = in.readUnsignedByte();
            int length = in.readUnsignedShort();
            if (in.isReadable(length)) {
                String data = in.readCharSequence(length, StandardCharsets.UTF_8).toString();
                Packet packet = new Packet(type, length, data);
                out.add(packet);
            } else {
                in.resetReaderIndex();
            }
        }
    }
}
```

Do it Netty

Codec - ByteToMessageDecoder

```
public class PacketDecoder extends ByteToMessageDecoder {
    @Override
    protected void decode(ChannelHandlerContext ctx, ByteBuf in, List<Object> out) throws Exception {
        in.markReaderIndex();
        if (in.isReadable(3)) {
            short type = in.readUnsignedByte();
            int length = in.readUnsignedShort();
            if (in.isReadable(length)) {
                String data = in.readCharSequence(length, StandardCharsets.UTF_8).toString();
                Packet packet = new Packet(type, length, data);
                out.add(packet);
            } else {
                in.resetReaderIndex();
            }
        }
    }
}
```

Do it Netty

Codec - ByteToMessageDecoder

```
public class PacketDecoder extends ByteToMessageDecoder {
    @Override
    protected void decode(ChannelHandlerContext ctx, ByteBuf in, List<Object> out) throws Exception {
        in.markReaderIndex();
        if (in.isReadable(3)) {
            short type = in.readUnsignedByte();
            int length = in.readUnsignedShort();
            if (in.isReadable(length)) {
                String data = in.readCharSequence(length, StandardCharsets.UTF_8).toString();
                Packet packet = new Packet(type, length, data);
                out.add(packet);
            } else {
                in.resetReaderIndex();
            }
        }
    }
}
```


Do it Netty

Codec - ByteToMessageDecoder

```
public class PacketDecoder extends ByteToMessageDecoder {
    @Override
    protected void decode(ChannelHandlerContext ctx, ByteBuf in, List<Object> out) throws Exception {
        in.markReaderIndex();
        if (in.isReadable(3)) {
            short type = in.readUnsignedByte();
            int length = in.readUnsignedShort();
            if (in.isReadable(length)) {
                String data = in.readCharSequence(length, StandardCharsets.UTF_8).toString();
                Packet packet = new Packet(type, length, data);
                out.add(packet);
            } else {
                in.resetReaderIndex();
            }
        }
    }
}
```

Do it Netty

Codec - MessageToByteEncoder

```
public class PacketEncoder extends MessageToByteEncoder<Packet> {  
    @Override  
    protected void encode(ChannelHandlerContext ctx, Packet msg, ByteBuf out) throws Exception {  
        out.writeByte(msg.getType());  
        out.writeShort(msg.getLength());  
        out.writeCharSequence(msg.getData(), StandardCharsets.UTF_8);  
    }  
}
```

Do it Netty

GatewayServer

```
.childHandler(new ChannelInitializer<SocketChannel>() {  
    @Override  
    protected void initChannel(SocketChannel ch) throws Exception {  
        ChannelPipeline pipeline = ch.pipeline();  
        pipeline  
            .addLast(new PacketDecoder())  
            .addLast(new GatewayInboundHandler())  
            .addLast(new PacketEncoder());  
    }  
});
```

Do it Netty

GatewayInboundHandler

```
public class GatewayInboundHandler extends ChannelInboundHandlerAdapter {
    private GatewayClient gatewayClient;
    @Override
    public void channelActive(ChannelHandlerContext ctx) throws Exception {
        gatewayClient = new GatewayClient(ctx);
    }
    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
        Packet packet = (Packet) msg;
        if (packet.getType() == (short) 1) gatewayClient.send(packet);
        else ctx.pipeline().writeAndFlush(packet);
    }
    @Override
    public void channelInactive(ChannelHandlerContext ctx) throws Exception {
        gatewayClient.disconnect();
    }
}
```

Do it Netty

GatewayInboundHandler

```
public class GatewayInboundHandler extends ChannelInboundHandlerAdapter {  
    private GatewayClient gatewayClient;  
    @Override  
    public void channelActive(ChannelHandlerContext ctx) throws Exception {  
        gatewayClient = new GatewayClient(ctx);  
    }  
    @Override  
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {  
        Packet packet = (Packet) msg;  
        if (packet.getType() == (short) 1) gatewayClient.send(packet);  
        else ctx.pipeline().writeAndFlush(packet);  
    }  
    @Override  
    public void channelInactive(ChannelHandlerContext ctx) throws Exception {  
        gatewayClient.disconnect();  
    }  
}
```

Do it Netty

GatewayInboundHandler

```
public class GatewayInboundHandler extends ChannelInboundHandlerAdapter {
    private GatewayClient gatewayClient;
    @Override
    public void channelActive(ChannelHandlerContext ctx) throws Exception {
        gatewayClient = new GatewayClient(ctx);
    }
    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
        Packet packet = (Packet) msg;
        if (packet.getType() == (short) 1) gatewayClient.send(packet);
        else ctx.pipeline().writeAndFlush(packet);
    }
    @Override
    public void channelInactive(ChannelHandlerContext ctx) throws Exception {
        gatewayClient.disconnect();
    }
}
```

Do it Netty

GatewayInboundHandler

```
public class GatewayInboundHandler extends ChannelInboundHandlerAdapter {
    private GatewayClient gatewayClient;
    @Override
    public void channelActive(ChannelHandlerContext ctx) throws Exception {
        gatewayClient = new GatewayClient(ctx);
    }
    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
        Packet packet = (Packet) msg;
        if (packet.getType() == (short) 1) gatewayClient.send(packet);
        else ctx.pipeline().writeAndFlush(packet);
    }
    @Override
    public void channelInactive(ChannelHandlerContext ctx) throws Exception {
        gatewayClient.disconnect();
    }
}
```

Do it Netty

GatewayClient

```
public class GatewayClient {
    private ChannelPipeline channelPipeline;
    public GatewayClient(ChannelHandlerContext serverCtx) {
        Bootstrap bootstrap = new Bootstrap();
        bootstrap.group(new NioEventLoopGroup()).channel(NioSocketChannel.class)
            .handler(new ChannelInitializer<SocketChannel>() {
                @Override
                protected void initChannel(SocketChannel ch) throws Exception {
                    channelPipeline = ch.pipeline();
                    channelPipeline.addLast(new PacketDecoder())
                        .addLast(new ChannelInboundHandlerAdapter() {
                            @Override
                            public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
                                serverCtx.pipeline().fireChannelRead(msg);
                            }
                        })
                    .addLast(new PacketEncoder());
                }
            })
            .connect("127.0.0.1", 10000);
    }
    ...
}
```


Do it Netty

GatewayClient

```
public class GatewayClient {
    private ChannelPipeline channelPipeline;
    public GatewayClient(ChannelHandlerContext serverCtx) {
        Bootstrap bootstrap = new Bootstrap();
        bootstrap.group(new NioEventLoopGroup()).channel(NioSocketChannel.class)
            .handler(new ChannelInitializer<SocketChannel>() {
                @Override
                protected void initChannel(SocketChannel ch) throws Exception {
                    channelPipeline = ch.pipeline();
                    channelPipeline.addLast(new PacketDecoder())
                        .addLast(new ChannelInboundHandlerAdapter() {
                            @Override
                            public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
                                serverCtx.pipeline().fireChannelRead(msg);
                            }
                        })
                    .addLast(new PacketEncoder());
                }
            }).connect("127.0.0.1", 10000);
    }
    ...
}
```

Do it Netty

GatewayClient

```
public class GatewayClient {
    private ChannelPipeline channelPipeline;
    public GatewayClient(ChannelHandlerContext serverCtx) {
        Bootstrap bootstrap = new Bootstrap();
        bootstrap.group(new NioEventLoopGroup()).channel(NioSocketChannel.class)
            .handler(new ChannelInitializer<SocketChannel>() {
                @Override
                protected void initChannel(SocketChannel ch) throws Exception {
                    channelPipeline = ch.pipeline();
                    channelPipeline.addLast(new PacketDecoder())
                        .addLast(new ChannelInboundHandlerAdapter() {
                            @Override
                            public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
                                serverCtx.pipeline().fireChannelRead(msg);
                            }
                        })
                    .addLast(new PacketEncoder());
                }
            }).connect("127.0.0.1", 10000);
    }
    ...
}
```

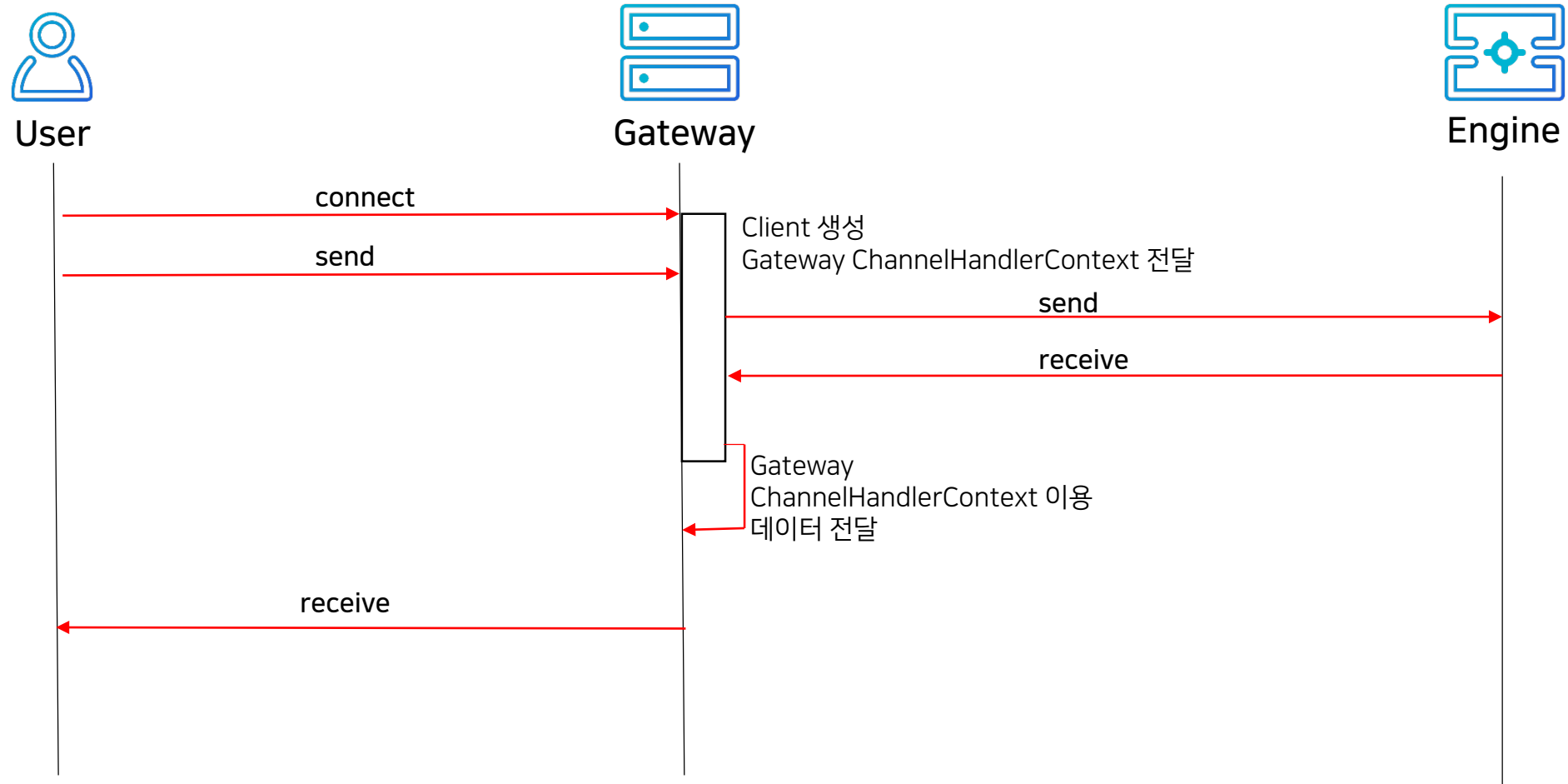
Do it Netty

GatewayClient

```
...  
    public void send(Packet packet) {  
        channelPipeline.writeAndFlush(packet);  
    }  
  
    public void disconnect() {  
        channelPipeline.disconnect();  
    }  
...
```

Do it Netty

Gateway - Engine 통신 정리



NHN FORWARD ▶▶▶