

LengthFieldBasedFrameDecoder

笔记本: <Inbox>

创建时间: 1/25/2019 10:05

更新时间: 1/25/2019 10:07

作者: wangqinlinmail

标签: Netty

LengthFieldBasedFrameDecoder是netty解决拆包粘包问题的一个重要的类，主要结构就是header+body结构。我们只需要传入正确的参数就可以发送和接收正确的数据，那么重点就在于这几个参数的意义。下面我们就具体了解一下这几个参数的意义。先来看一下LengthFieldBasedFrameDecoder主要的构造方法：

```
public LengthFieldBasedFrameDecoder(  
    int maxFrameLength,  
    int lengthFieldOffset, int lengthFieldLength,  
    int lengthAdjustment, int initialBytesToStrip)
```

那么这几个重要的参数如下：

- maxFrameLength：最大帧长度。也就是可以接收的数据的最大长度。如果超过，此次数据会被丢弃。
- lengthFieldOffset：长度域偏移。就是说数据开始的几个字节可能不是表示数据长度，需要后移几个字节才是长度域。
- lengthFieldLength：长度域字节数。用几个字节来表示数据长度。
- lengthAdjustment：数据长度修正。因为长度域指定的长度可以使header+body的整个长度，也可以只是body的长度。如果表示header+body的整个长度，那么我们需要修正数据长度。
- initialBytesToStrip：跳过的字节数。如果你需要接收header+body的所有数据，此值就是0，如果你只想接收body数据，那么需要跳过header所占用的字节数。

下面我们根据几个例子的使用来具体说明这几个参数的使用。

需求1

长度域为2个字节，我们要求发送和接收的数据如下所示：

```
* 发送的数据 (14 bytes) 接收到数据 (14 bytes)  
* +-----+-----+ +-----+ +-----+  
* | Length | Actual Content |----->| Length | Actual Content |
```

```
* | 12 | "HELLO, WORLD" | | 12 | "HELLO, WORLD" |
* +-----+-----+ +-----+-----+
```

细心的你肯定发现了，长度域只是实际内容的长度，不包括长度域的长度。下面是参数的值：

- lengthFieldOffset=0: 开始的2个字节就是长度域，所以不需要长度域偏移。
- lengthFieldLength=2: 长度域2个字节。
- lengthAdjustment=0: 数据长度修正为0，因为长度域只包含数据的长度，所以不需要修正。
- initialBytesToStrip=0: 发送和接收的数据完全一致，所以不需要跳过任何字节。

需求2

长度域为2个字节，我们要求发送和接收的数据如下所示：

```
* 发送的数据 (14 bytes) 接收到数据 (12 bytes)
* +-----+-----+ +-----+
* | Length | Actual Content |---->| Actual Content |
* | 12 | "HELLO, WORLD" | | "HELLO, WORLD" |
* +-----+-----+ +-----+
```

参数值如下：

- lengthFieldOffset=0: 开始的2个字节就是长度域，所以不需要长度域偏移。
- lengthFieldLength=2: 长度域2个字节。
- lengthAdjustment=0: 数据长度修正为0，因为长度域只包含数据的长度，所以不需要修正。
- initialBytesToStrip=2: 我们发现接收的数据没有长度域的数据，所以要跳过长度域的2个字节。

需求3

长度域为2个字节，我们要求发送和接收的数据如下所示：

```
* BEFORE DECODE (14 bytes) AFTER DECODE (14 bytes)
* +-----+-----+ +-----+
* | Length | Actual Content |---->| Length | Actual Content |
* | 14 | "HELLO, WORLD" | | 14 | "HELLO, WORLD" |
* +-----+-----+ +-----+
```

细心的你肯定又发现了，长度域表示的长度是总长度 也就是header+body的总长度。参数如下：

- lengthFieldOffset=0: 开始的2个字节就是长度域，所以不需要长度域偏移。
- lengthFieldLength=2: 长度域2个字节。
- lengthAdjustment=-2: 因为长度域为总长度，所以我们需要修正数据长度，也就是减去2。
- initialBytesToStrip=0: 我们发现接收的数据没有长度域的数据，所以要跳过长度域的2个字节。

需求4

长度域为2个字节，我们要求发送和接收的数据如下所示：

```
* BEFORE DECODE (17 bytes) AFTER DECODE (17 bytes)
* +-----+-----+-----+-----+-----+-----+
* | meta | Length | Actual Content |----->| meta | Length | Actual
  Content |
* | 0xCAFE | 12 | "HELLO, WORLD" | | 0xCAFE | 12 | "HELLO, WORLD" |
* +-----+-----+-----+-----+-----+-----+
  -----+
```

我们发现，数据的结构有点变化，变成了 meta+header+body 的结构。meta 一般表示元数据，魔数等。我们定义这里 meta 有三个字节。参数如下：

- lengthFieldOffset=3: 开始的3个字节是 meta，然后才是长度域，所以长度域偏移为3。
- lengthFieldLength=2: 长度域2个字节。
- lengthAdjustment=0: 长度域指定的长度位数据长度，所以数据长度不需要修正。
- initialBytesToStrip=0: 发送和接收数据相同，不需要跳过数据。

需求5

长度域为2个字节，我们要求发送和接收的数据如下所示：

```
* BEFORE DECODE (17 bytes) AFTER DECODE (17 bytes)
* +-----+-----+-----+-----+-----+-----+
* | Length | meta | Actual Content |----->| Length | meta | Actual
  Content |
* | 12 | 0xCAFE | "HELLO, WORLD" | | 12 | 0xCAFE | "HELLO, WORLD" |
* +-----+-----+-----+-----+-----+-----+
  -----+
```

我们发现，数据的结构有点变化，变成了 header+meta+body 的结构。meta 一般表示元数据，魔数等。我们定义这里 meta 有三个字节。参数如下：

- lengthFieldOffset=0: 开始的2个字节就是长度域，所以不需要长度域偏移。
- lengthFieldLength=2: 长度域2个字节。
- lengthAdjustment=3: 我们需要把meta+body当做body处理，所以数据长度需要加3。
- initialBytesToStrip=0: 发送和接收数据相同，不需要跳过数据。

需求6

长度域为2个字节，我们要求发送和接收的数据如下所示：

```
* BEFORE DECODE (16 bytes) AFTER DECODE (13 bytes)
* +-----+-----+-----+-----+ +-----+-----+
-+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Co
  | ntent |
* | 0xCA | 0x000C | 0xFE | "HELLO, WORLD" | | 0xFE | "HELLO, WORLD"
  |
* +-----+-----+-----+-----+ +-----+-----+
-+
```

我们发现，数据的结构有点变化，变成了 hdr1+header+hdr2+body的结构。我们定义这里hdr1和hdr2都只有1个字节。参数如下：

- lengthFieldOffset=1: 开始的1个字节是长度域，所以需要设置长度域偏移为1。
- lengthFieldLength=2: 长度域2个字节。
- lengthAdjustment=1: 我们需要把hdr2+body当做body处理，所以数据长度需要加1。
- initialBytesToStrip=3: 接收数据不包括hdr1和长度域相同，所以需要跳过3个字节。