[socket 1](#_Toc13274)

[NioSocketConnector 2](#_Toc11538)

[IoSession 2](#_Toc25767)

[IoHandlerAdapter 2](#_Toc25505)

mina是一个java框架，我理解的作用是简化socket编程。

# socket

软件编程中，HTTP协议强大，使用最多，或者说几乎都在用它。经常让我觉得，网络通信就这么一种协议，其他都是异类。

但强如HTTP也是在socket基础上做的封装。

那什么是socket？

一个底层网络通信协议。

那什么是协议？

一个A，一个B，用来约束A和B行为的规范就叫协议。

所以再详细解释下，socket做为一种通信协议，要求请求方如何发送数据，如何接收数据，如何验签；服务方如何接收数据，如何发送数据，如何验签。

再详细一点，我就没那个本事了。。。。。

至于为什么明明有HTTP，还要用socket，这个我回答不上来，只能说某些场景下，确实socket比较方便吧，毕竟底层，更利于传输，就是写代码的人头疼点。目前，我接触到的，与硬件设备通信基本上都是走socket。

那mina要怎么简化这个过程呢？

/\*

\* 占坑，原生java如何socket编程的代码还没见过。

\*/

第一，使用一个connector建立连接。

第二，获取一个session，并利用这个session向对方发送数据。

第三，定义一个handler，意思就是收到对方发来的数据后，要怎么处理。

我打赌你没用过mina肯定以为过程是这样的：

// 下面都是假代码

Connector conn = new Connector(参数，参数，参数);

Session session = conn.getSession();

Handler handler = session.writeMessage(参数，参数，参数);

如果你不是这么想的，少年，至少你比我强。

其实是这样的（下面只是一种写法，应该还有其它的，还没研究）

Response response;

{

Handler handler = new Handler();

Connector conn = new Connector(handler，参数，参数，参数);

Session session = conn.getSession();

session.writeMessage(参数，参数，参数);

// 上面代码执行完后，这里可以直接用response

response.XXX();

}

先定义好Handler，然后注册给Connector，接着获取Session，并发送消息，Session会捕获对方返回的数据，并自动调用handler，然后解析数据，并按某种定义好的规则封装到Response对象里，当然这个Response类也是要自己定义的。

# NioSocketConnector

# IoSession

# IoHandlerAdapter