

项目——搭建一个web服务器

建立一个服务器和客户端

你已经学习过了如何编写客户端，那么编写一个服务端需要什么呢？只需要一对**Channel**，一个是**ServerSocketChannel**，它需要等待客户端请求，还有一个是**SocketChannel**，用于客户端通信。如果有多个客户端，我们就需要多个通道。

```
//如何工作
//服务器创建一个ServerSocketChannel并绑定到一个特定端口
ServerSocketChannel serverChannel = ServerSocketChannel.open();
serverChannel.bind(new InetSocketAddress(5000));
//客户端创建一个连接到服务器应用的SocketChannel
SocketChannel svr = SocketChannel.open(new InetSocketAddress("190.165.1.103",
5000));
//服务器创建一个新SocketChannel与客户端通信
SocketChannel clientChannel = serverChannel.accept();
```

通过这样的操作，你可以实现一个简单的服务器

这里给出一个实现的简单网络服务器的例子：[Java网络编程](#)

你也可以自行寻找一些资料进行更深入的学习。

编写一个聊天客户端

你前面已经学习过如何编写客户端，现在就是把它投入实战的时候了，在这里我们只要求你实现一个只发送消息的版本，使之可以向服务器发送消息，下面是聊天服务器需要提供的主要功能结构：

```
public class SimpleChatClientA {
    public void go() {
        // call the setUpNetworking() method
    }
    private void setUpNetworking() {
        // open a SocketChannel to the server
        // make a PrintWriter and assign to writer instance variable
    }
    private void sendMessage() {
        // send it to the server using the writer (a PrintWriter)
    }
}
```

编写一个聊天服务器

```
public class SimpleChatServer {
    private final List<PrintWriter> clientWriters = new ArrayList<>();
    public static void main(String[] args) {
        new SimpleChatServer().go();
    }
    public void go() {
        //将服务器运行起来
    }
}
```

```
private void tellEveryone(String message) {  
    //将消息打印出来  
}  
public class ClientHandler implements Runnable {  
    //定义一个控制类  
}  
public void run() {  
    //运行函数  
}  
}
```

当你做完这些之后，你可以在终端将服务器开启。随后创建一个新的终端，在新的终端启动服务端，这个时候你的聊天服务就可以使用了，测试一下，看看有没有问题呢。

要求：将实现的代码和运行截图推送到GitHub仓库上，此处提交链接：

附加题

改进你的服务器

现在我们的服务器只具备接收消息的功能，，我们也许可以改进我们的服务器，使之可以发送和接收消息。

问题：我们如何从服务器得到消息

问题：我们应该何时从服务器得到消息

认真思考上面的问题，并想出合适的解决方案。

我们需要连续的运行服务器，检查来自服务器消息，并且不中断用户的交互，这意味着我们需要一个全新的栈，一个全新的线程。事实上，java已经为你提供了多线程支持：

```
Thread t = new Thread();  
t.start();
```

通过这样的操作，你就可以创建一个全新的线程，但是这样的线程在创建的时候就已经“死去”了。想要深入了解多线程的知识，这里我们提供了教程：[Java多线程](#)，[Java 多线程编程](#)

任务：现在你的任务是，改进聊天服务器，在向服务器发送消息的同时从服务器读取收到的消息。

要求：将实现的代码和运行截图推送到GitHub仓库上，此处提交链接：

出题人QQ：1727448271@qq.com