# Netty+Spring Boot仿微信全栈开发高性能后台及客户端

# Netty

是一个提供了易于使用的api的客户端/服务器框架

并发高 – NIO 非阻塞IO

传输快 零拷贝

Io流-> 缓冲区 -> 堆 原始的需要拷贝

<https://netty.io/wiki/user-guide-for-4.x.html>

## 阻塞和非阻塞

线程访问资源，该资源是否准备就绪的一种处理方式

线程等到这个资源处理完毕返回一个结果 阻塞

线程不会持续等待处理完毕，会去请求别的资源 这就是非阻塞

## 同步和异步

是访问数据的一种机制

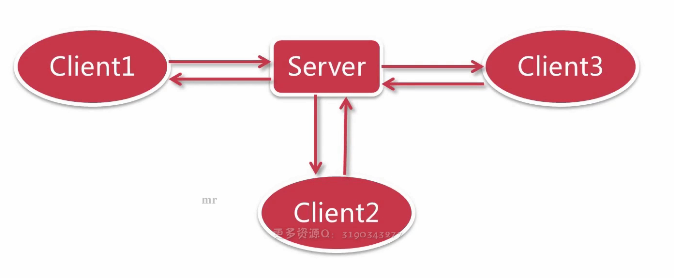
同步：主动请求，等待io操作完了会通知

异步：主动请求可以发起其他请求，异步通知

## BIO

同步阻塞IO Block Io

Io在进行读写，这个线程是被阻塞，无法进行其他的操作 并发能力很低 资源间通信慢 依赖贷款网速



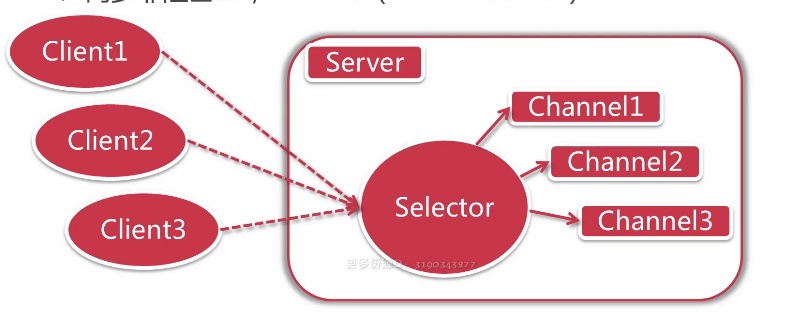
一应一答 客户端逐渐增多，频繁创建线程，销毁线程 很差劲 改良之后，就是线程池 伪异步io 这个bio就像是上厕所找不到位置一直等着

## NIO

同步非阻塞io

去上厕所，发现没坑了。就开始玩手机，时不时看看有没有坑，不是光等着。

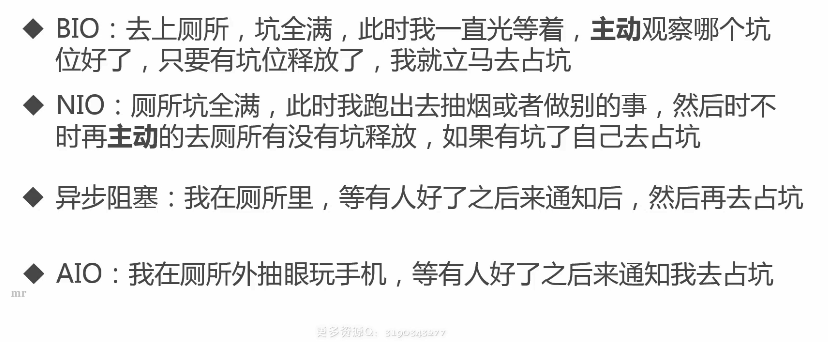
Selector 选择器，轮询

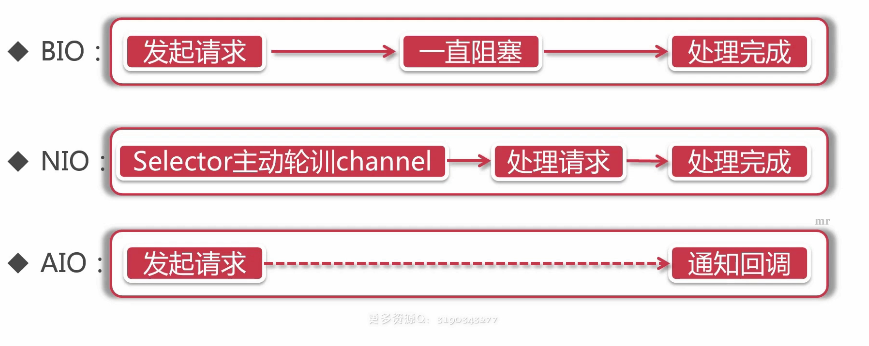


## AIO

异步非阻塞io 相比下面不等着，在做自己的事情

异步阻塞io 几乎用不到 比如上厕所，没位置了。就等着。就让哪个人好了告诉我我再去。这种方式非常傻





## Netty

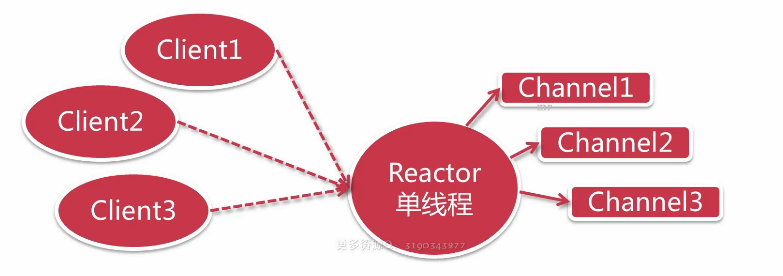
Java 的开源框架 网络服务器客户端框架

使用netty的优点 java开源框架 原始nio有bug存在。

## Reactor线程模型

提供了三种线程模型

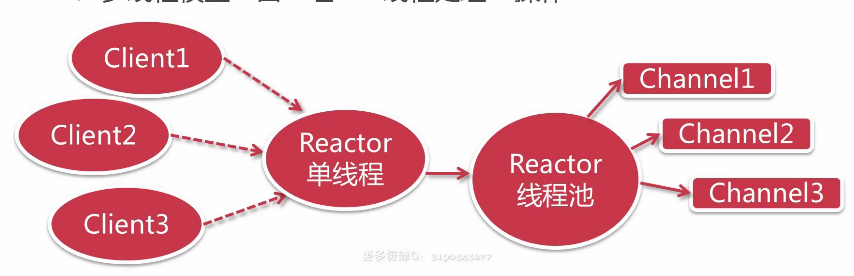
单线程模型：所有io操作都由同一个nio线程处理的



独立处理所有io的操作 完成所有的操作

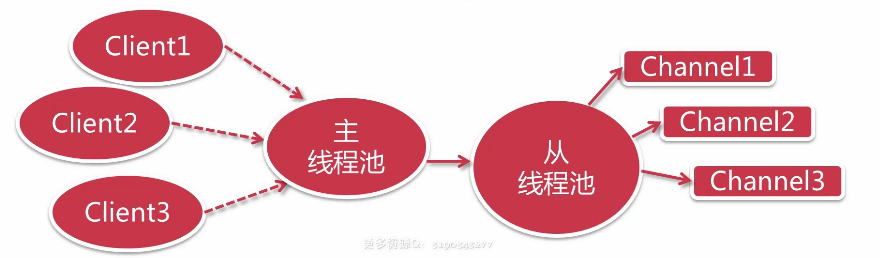
适合场景：高并发、海量数据复杂的场景

多线程：一组nio线程处理io操作



重点概念：线程池 等待队列

主从线程模型：一组线程池接受请求，一组线程池，处理io



<https://blog.csdn.net/quxing10086/article/details/80296245>

# java仿微信全栈开发高性能后台+移动客户端

## hello netty步骤：

这些东西多看官方文档

构建一组主从线程组 高效

定义服务器启动类

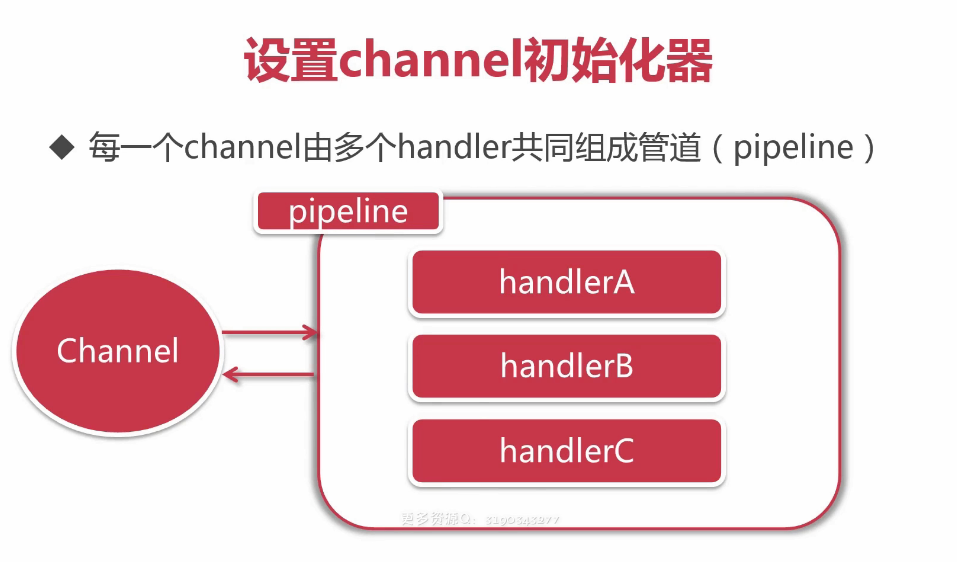
为服务器设置Channel

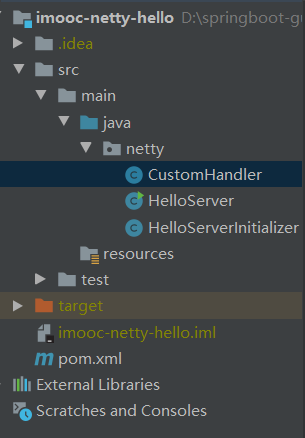
设置处理从线程池的助手类初始化器

监听启动和关闭服务器

这里要注意netty5已经被废弃了 开发过程用的是4

新建maven工程 添加netty依赖

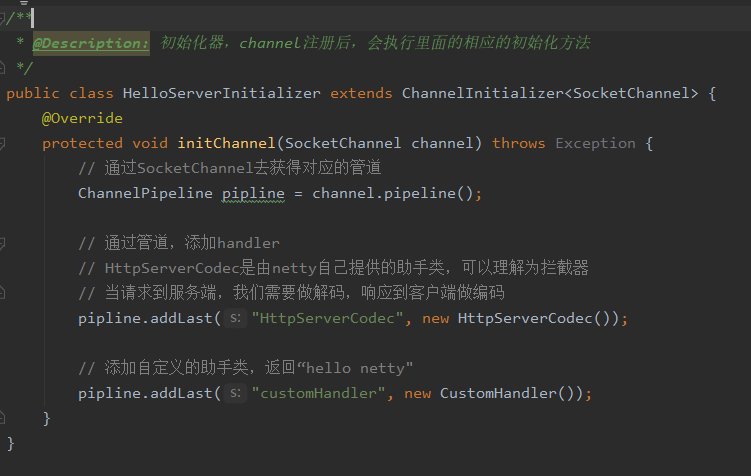




### 先写helloserver

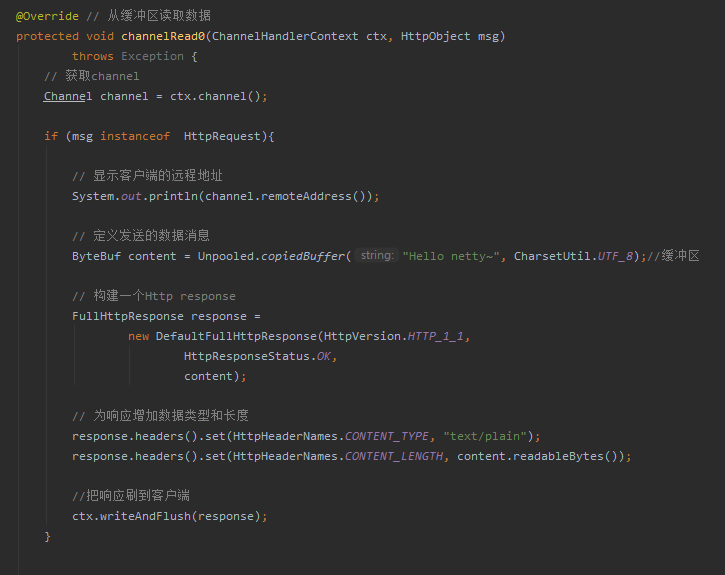


### 编写初始化器



### 子处理器





## 聊天小练习

### 实时通信

Ajax轮训

Long pull

Websocket



前端采用hbuilder来开发

# 使用mui与h5+构建移动端

# 使用springboot整合netty构建后台

# 用户注册登陆个人信息

# 发现页面与通讯录相关功能开发

# 聊天页面开发

# 课程总结