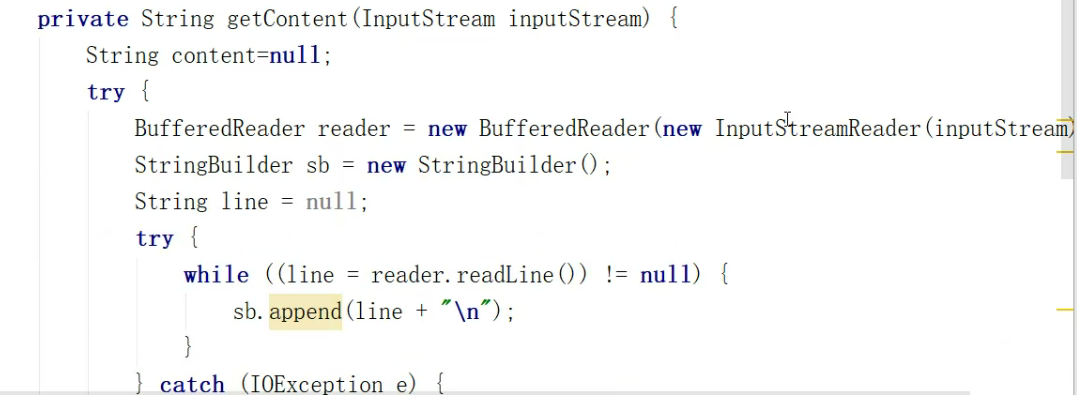
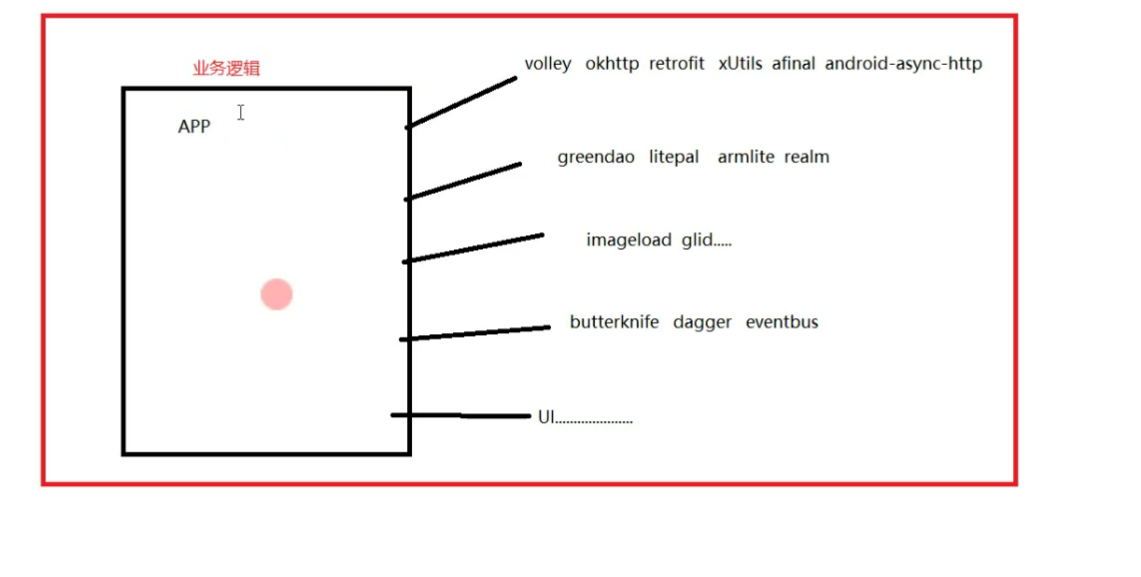
OKHTTP框架设计

#### 常识

1. 常用解析方式：JSON、XML
2. 大公司做框架必须写文档。
3. 服务器秒并发数可达到10W以上
4. 马爸爸公司的Json库
5. 习惯使用单例
6. 数据结构：
7. 泛型
8. Java中 mXXX ： 变量 IXXXX： 接口
9. 最通用的数据结构： byte数组
10. get请求不需要参数，POST请求一般有表单
11. 网络文本编码：UTF-8
12. 写来写去也就是二十几个编程模式
13. 数据解析完全可以用Json来完成（对象转JSON，Json转对象）
14. 流转字符串
15. 架构必须了解 1泛型 2反射 3注解 4设计模式

#### 需求

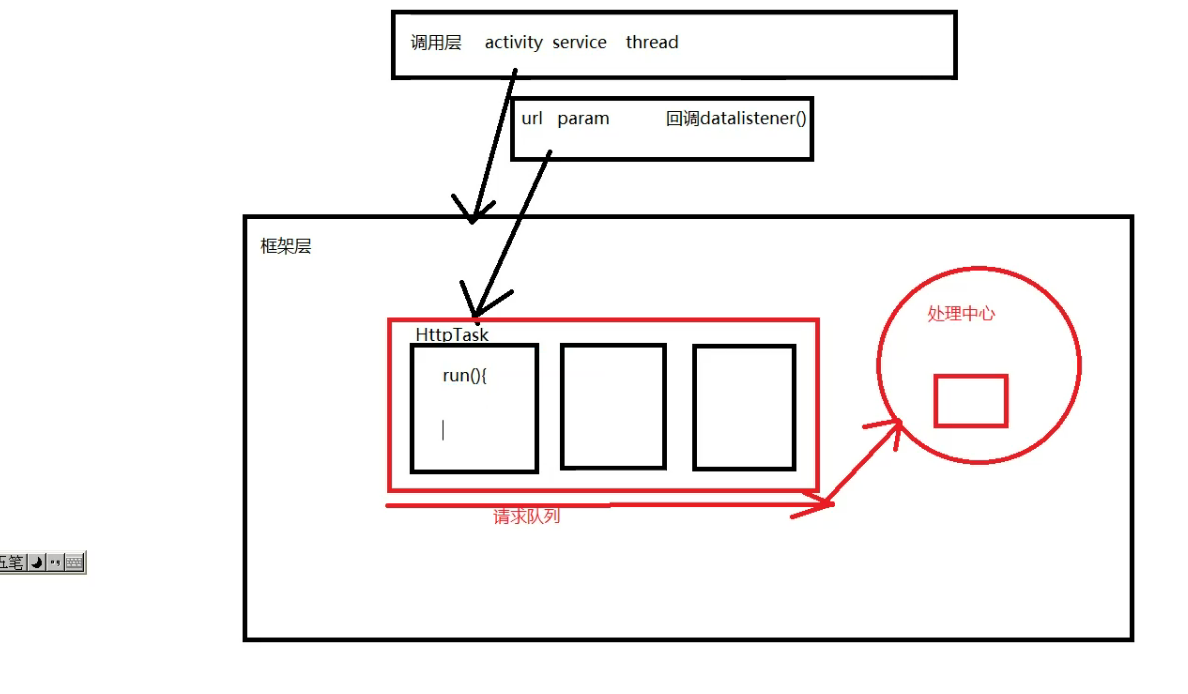
1. 调用层不需要关心数据解析
2. 框架内完成网络访问等耗时操作在子线程进行，框架完成线程切换
3. 高并发数HTTP连接 （200~1000秒发）

#### 一次HTTP任务

URL

RequestParams

callback



HttpRequest – Objects

Queue – Collection

Processors – ThreadPool

线程池排队超时 -> 线程池会丢弃线程

->> 不能丢弃用户的请求 🡪 重新放回队列

#### 队列要求Queue

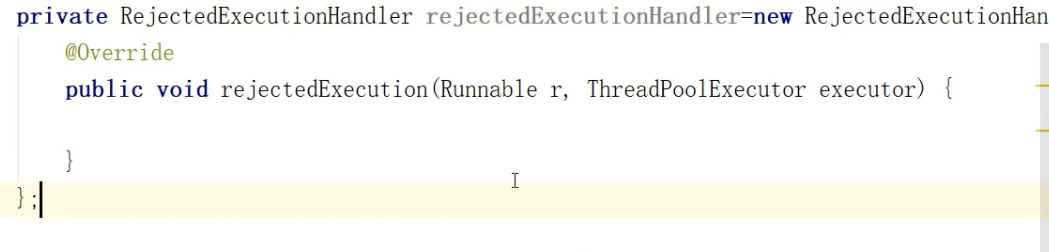
1. 阻塞式
2. 插入动作频繁
3. 容量无限【不考虑数组】

**结论：**



#### 线程池 ThreadPoolManager ThreadPoolExecutor

拒绝策略【 Google的框架 】



**实现循环执行**

**Thread.execute(**

**new Runnable(){**

**public void Run(){**

**while(true){**

**Runnable r = threadpool.take();**

**if(r!=null) threadexecutor.execute(r);**

**}**

**}**

**}**

**)**

#### HttpTask



**接口**



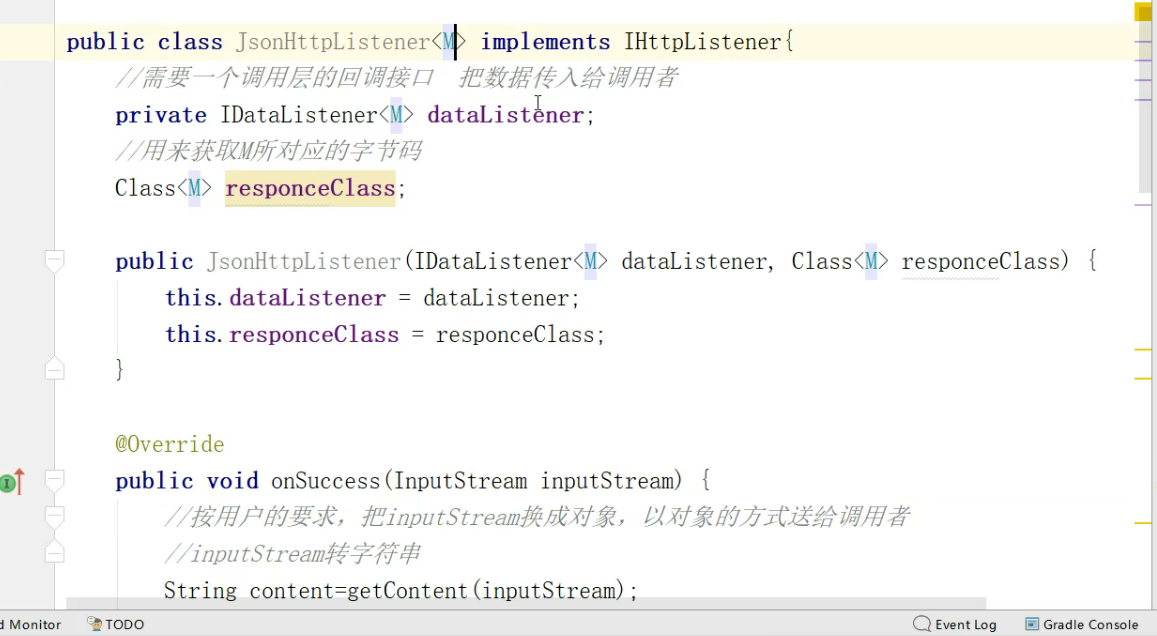
**实现**



**Json的妙用**



回调接口



#### 封装【黑盒子】

