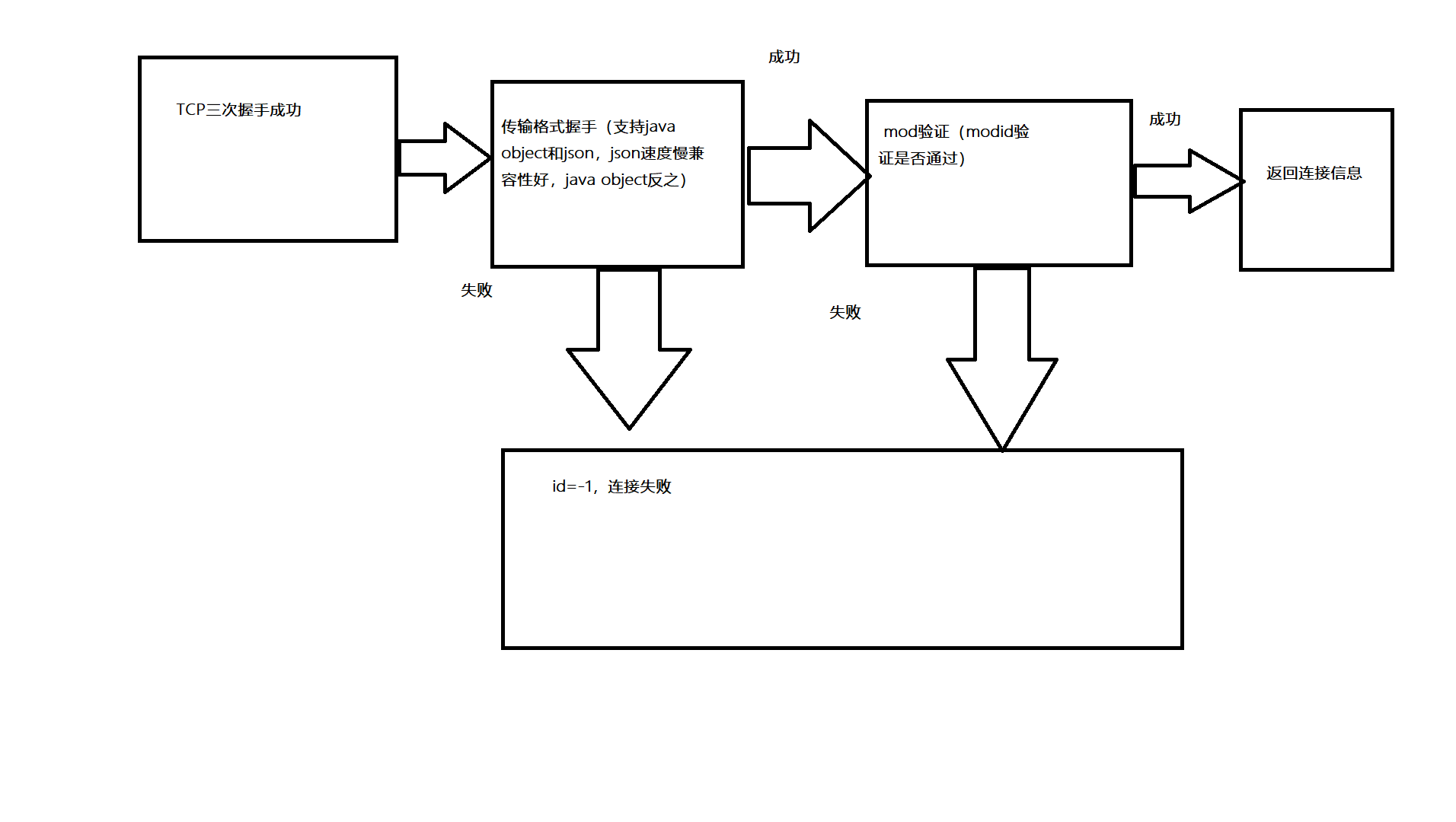
请先更新库到最新版本

这个api最重要的两个类当属Client和Server，他们均为TCP协议的，C/S的设计模式使得不能用浏览器访问这里的Server。Client在连接Server的时候会进行握手，如果握手流程中需要有改动的地方，给我提交个PR就好了。我看情况接受（如果一些核心部分被改的不能用了不会接受）握手有个流程（可以用事件api写，但是我懒，而且事件api是非常后面的东西了）：



然而，有两个格式可能引起了你的注意：RECONNECT和FINISH，这不是格式，而是两个特殊的值。FINISH是格式不兼容的时候发送的，为了避免握手带来的开销，RECONNECT被用于断线重连（连接重置的时候socket和握手信息仍然保留，重连通过连接id确定客户端，id为客户端数组下标）

sync子包下的ClientSideSynchronizeImpl是未完成的Impl，其内部类ServerSideSynchronizeImpl同为未完成的Impl（别问我为啥写内部类，方便，且ServerSideSynchronizeImpl是一个客户端一个的，且不应该调用父类构造器，故不宜用继承，未完成：网络传输出现异常时的操作未完成，我用TODO注释写了）

然而比Impl还重要的东西是event。这个api可以说是为解决不兼容json而生。EventContent类是万恶之源，所有的事件都要继承EventContent，sendExec为事件被发送时被执行的，recvExec为接收时执行。

EventContent类有个getSourceClient的static方法，返回的是接收到事件的客户端（用其携带的id属性表示，id==c.msg.id，即EventContent的id属性和握手信息的id一致，在一个连接中，服务器和客户端持有一个相同的id）

所有自定义的事件都要求继承EventContent类，没有eid（eid在注册事件的时候被赋值，用到了反射）的事件是不允许被json格式传输的。其中一个抽象类叫做OtherEvent，也继承自EventContent，这个抽象类给创建事件带来了一点方便。First，事件被允许携带内容（content属性），为了支持json格式，这边需要实现一个名为initContent(Map)的方法，这个方法允许通过一个Map来构建content的内容，并返回这个对象（没必要赋值给content，在读取的时候会给你赋值）。Second，注册事件必须调用Events.reg(Class<? extends EventContent,int)方法或者Events的构造代码块增加相应语句来实现

说到json格式，就不得不提json的转换方式了。这边没用到别的库，用的是我~~最喜欢的~~java原生的nashorn。简单的把json转换成javascript object，然后把javascript object返回，在java里面表示为ScriptObjectMirror（其实就是Map）。如果你觉得用gson api更靠谱，可以替换（把JavaScript.parse等方法替换成对应的操作就可以，如果不是Map，则把Map改成对应参数类型），这样一来，你可以自由调整json格式的转换

Copyleft(Ɔ) CraftGame Studio.