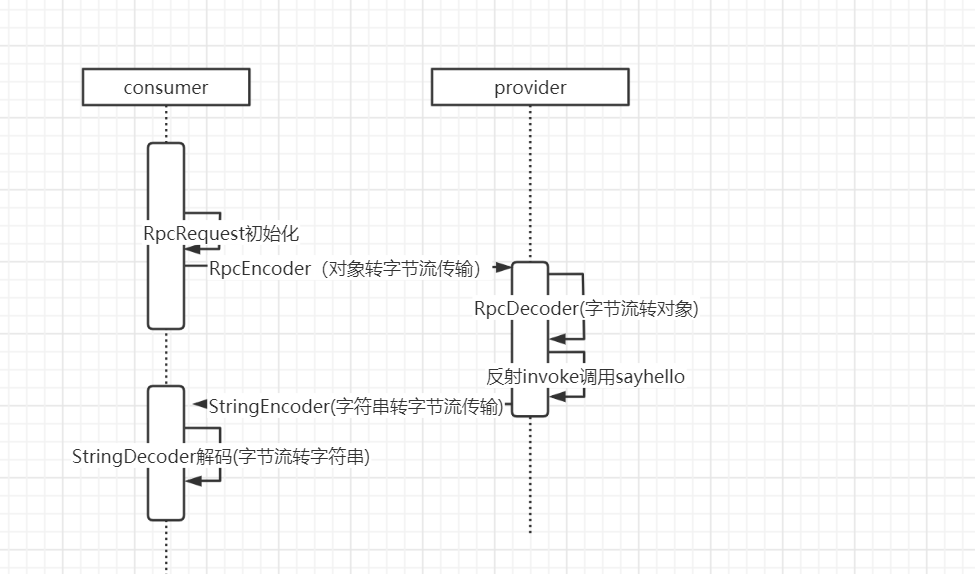
**在基于Netty的自定义RPC的案例基础上，进行改造(说明文档)**

# 序列化

首先对象要在网络上传输必须序列化，项目中采用fastjson序列化工具。同时对象解析的时候需要把网络传过来的字节流反序列化成对象

# 思路分析



# 总结

1. Consumer 组件构造 请求参数对象 RpcRequest，同时把反射调用的参数设置到参数对象中
2. Consumer 对象使用fastjson序列化机制,把RpcRequest对象序列化成字节流发送给服务端
3. 服务端接收到字节流用户fastjson反序列化机制进行反序列化把字节流转成RpcRequest对象
4. 服务端根据RpcRequest参数中是属性进行反射调用IuserService的sayHello方法并返回success，通过StringEncoder转成字节流传给客户端
5. 客户端接受到返回用StringDecoder把自己转成字符，打印到控制台