Thrift 的各项传输协议和Socket方式及各种Server的特性。这里是0.9.3版本整理的。

**1、数据类型**

基本类型：  
bool：布尔值，true 或 false，对应 Java 的 boolean  
byte：8 位有符号整数，对应 Java 的 byte  
i16：16 位有符号整数，对应 Java 的 short  
i32：32 位有符号整数，对应 Java 的 int  
i64：64 位有符号整数，对应 Java 的 long  
double：64 位浮点数，对应 Java 的 double  
string：未知编码文本或二进制字符串，对应 Java 的 String  
结构体类型：  
struct：定义公共的对象，类似于 C 语言中的结构体定义，在 Java 中是一个 JavaBean  
容器类型：  
list：对应 Java 的 ArrayList  
set：对应 Java 的 HashSet  
map：对应 Java 的 HashMap  
异常类型：  
exception：对应 Java 的 Exception  
服务类型：service：对应服务的类

**2、支持的传输格式**

TBinaryProtocol   二进制格式  
TCompactProtocol   压缩格式  
TJSONProtocol     使用JSON的数据编码协议进行数据传输  
TSimpleJSONProtocol 提供JSON只写协议，生成的文件很容易通过脚本语言解析

**3、支持的数据传输方式**

  阻塞式socket  
  
TFramedTransport 以frame为单位传输，非阻塞式服务中使用  
TFileTransport   以文件形式进行传输  
  
TZlibTransport   使用zlib进行压缩。  
  
TNonblockingTransport 使用非阻塞方式，用于构建异步客户端。

**4、支持的服务模型[主要是服务端类型**

TSimpleServer   简单的单线程服务模型，常用于测试  
  
TThreadPoolServer 多线程服务模型使用标准的阻塞式IO,预先创建一组线程处理请求  
TNonblockingServer 多线程服务模型使用非阻塞IO(需要用TFramedTransport数据传输方式)  
  
ps:什么叫阻塞与非阻塞？  
阻塞IO：socket的阻塞意味着必须要做完IO包括错误才会返回  
非阻塞IO：无论操作是否完成都会立刻返回，需要通过其他方式来判断具体操作是否成功  
在网络阻塞严重的时候，网络层没有足够的内存来进行写操作，这时候就会出现写不成功的情况，  
阻塞情况下会尽可能(有可能被中断)等待到数据全部发送完毕， 对于非阻塞的情况就是一次写多少算多少,

没有中断的情况下也还是会出现write 到一部分的情况.

附录一个thrift文件格式demo：

1. namespace java com.zwz.server
2. service StockTest {
3. string server\_demo1(1:string para)
4. string server\_demo2(1:string para)
5. }