**1. Thrift数据类型及基本写法**

Thrift大小写敏感

基本数据类型：

bool 布尔值

byte 有符号字节

i16 16位有符号整数

i32 32位有符号整数

i64 64位有符号整数

double 64位浮点数

string 与编码无关的文本或二进制字符串

常量：

const i32 INT\_CONST = 1234;

类型别名：

typedef i32 Integer

结构体：

struct Work {

1: i32 num1 = 0,

2: i32 num2,

3: Operation op,

4: optional string comment,

}

枚举：

enum Operation {

ADD = 1,

SUBTRACT = 2,

MULTIPLY = 3,

DIVIDE = 4

}

枚举类型里不需要加序号

容器：

list<type>：映射为STL vector，Java ArrayList，Python list

set<type>：映射为STL set，Java HashSet，Python set

Map<type1,type2>：映射为STL map，Java HashMap，Python dictionary

异常：

exception InvalidOperation {

1: i32 whatOp,

2: string why

}

服务：

service <name> {

<returntype> <name>(<arguments>)

[throws (<exceptions>)]

...

}

service Calculator extends shared.SharedService {

void ping(),

i32 add(1:i32 num1, 2:i32 num2),

i32 calculate(1:i32 logid, 2:Work w) throws (1:InvalidOperation ouch),

oneway void zip()

}

这里函数参数要用数字进行编号，编号从1开始，“序号: 参数类型 参数名”

每个函数末尾加逗号，最后一个函数后面不需要

命名空间：

namespace cpp com.example.project

namespace java com.example.project

这里java对应相应的包的名称

引入其他.thrift文件：

include “tweet.thrift”

在使用tweet.thrift内部定义的变量时，注意加上tweet前缀，如：tweet.Tweet

注释：

单行注释：#或//

多行注释：/\*\* \*\*/

Thrift分三部分：

第一部分 :

核心的\*\*.thrift文件，

thrift命令生成的相应的java或c++代码文件，

服务端接口文件, 如\*\*Impl.java

第二部分 : 服务端代码\*\*Server.java

第三部分 : 客户端代码\*\*Client.java

Thrift代码 : (放在服务端)

.thrift文件格式 :

namespace java service.demo

service Hello {

string helloString(1:string para),

i32 helloInt(1:i32 para),

bool helloBoolean(1:bool para),

void helloVoid(),

string helloNull(),

}

服务端 :

创建TServerTransport/TServerSocket类

创建TProcessor类

由TServerTransport类和TProcessor类创建TServer/TThreadPoolServer类

客户端 :

创建TTransport/TSocket类

由TTransport类创建TProtocol/TBinaryProtocol类, 将结构化数据组装成message

由TProtocol类创建Client类

thrift生成对应语言文件语法, 这里java可替换为cpp/csharp/py等以生成不同的语言接口

$ thrift -gen <language> <Thrift filename>

$ thrift -gen java hello.thrift

服务端的TServerSocket类和客户端的TSocket类用于建立连接

RPC远程过程调用