环境搭建-Java

工具

被章耿添加，被章耿最后更新于九月 02, 2016

简介：

http://www.grpc.io/

一个基于HTTP2 和 Protobuf 的RPC 实现。

这里看到了RPC最重要的两部分内容，传输协议和序列化协议。

这两个都是支持跨语言的。

准备

1.下载protobuf编译工具

https://github.com/google/protobuf/releases/tag/v3.0.0 拉到最下面，根据操作系统下载，例如 protoc-3.0.0-linux-x86_64.zip

解压到 **$protoc3**

2. 下载 grpc-java源码

https://github.com/grpc/grpc-java

3. 用grpc-java 编译grpc代码生成插件

试试吧，不报错还好，报错要吐。 https://github.com/grpc/grpc-java/tree/master/compiler

太麻烦，还老出错，下载官方编译好的吧。https://repo1.maven.org/maven2/io/grpc/protoc-gen-grpc-java/1.0.0/ 记得chmod +x。

Protobuf文档编写

语法：https://developers.google.com/protocol-buffers/docs/proto3

具体不多说，看里面的例子吧。

支持常见的数据类型，3.0支持map，还支持非必填字段等。

例如：

dto.proto

```
syntax = "proto3";

package com.jd.yunfan.testgrpc.dto;

option java_multiple_files = true;
option java_outer_classname = "HelloServiceDto";

message PingRequest {
    string in=1;
}

message PingResponse {
    string out=1;
}

message QueryParameter {
    int32 ageStart = 1;
    int32 ageEnd = 2;
}

message Person {
    int32 age = 1;
    string name = 2;
    bool sex=3;
    double salary=4;
    int32 childrenCount=5;
}

message PersonList {
    repeated Person items = 1;
}
```

再例如接口

```
syntax = "proto3";

import "dto.proto";

package com.jd.yunfan.testgrpc.service;

option java_multiple_files = true;
option java_outer_classname = "HelloService";

service HelloService {
    rpc Ping (com.jd.yunfan.testgrpc.dto.PingRequest) returns (com.jd.yunfan.testgrpc.dto.PingResponse) {}

    rpc getPersonList (com.jd.yunfan.testgrpc.dto.QueryParameter) returns
(com.jd.yunfan.testgrpc.dto.PersonList) {}
}
```

都比较好理解吧。有人问为什么要分开写？是为了下面生成代码的时候，对象和接口隔离。

继续下一步吧。

生成代码

用到我们刚才准备的工具了。

例如我的

生成protobuf代码，参考命令如下：

```
#!/usr/bin/env bash
PROTOC3="/home/zhanggeng/workspace/project_yunfan/protobuf"
PROJECT_HOME="/home/zhanggeng/workspace/project_yunfan/testgrpc/"
echo "gen dto"
${PROTOC3}/bin/protoc \
    -I=${PROJECT_HOME}/src/main/resources/ \
    --java_out=${PROJECT_HOME}/src/main/java \
    ${PROJECT_HOME}/src/main/resources/dto.proto
echo "gen service"
${PROTOC3}/bin/protoc \
    -I=${PROJECT_HOME}/src/main/resources/ \
    --java_out=${PROJECT_HOME}/src/main/java \
    ${PROJECT_HOME}/src/main/resources/service.proto
echo "gen grpc service"
${PROTOC3}/bin/protoc \
     --plugin=protoc-gen-grpc-java=/home/zhanggeng/workspace/project_yunfan/protobuf/bin/protoc-gen-grpc-
java-1.0.0-linux-x86_64.exe \
     --grpc-java_out=${PROJECT_HOME}/src/main/java \
     -I=${PROJECT_HOME}/src/main/resources/ \
      ${PROJECT_HOME}/src/main/resources/service.proto
echo "over!"
```

就会在src/main/java根据你的文件生成指定路径的 java dto文件和service文件

再生成 grpc的相关代码

编写服务端和客户端

demo 地址：

http://source.jd.com/app/testgrpc-java

无