

什么是protobuf

Protocol Buffers 是一种轻便高效的结构化数据存储格式，可以用于结构化数据串行化，或者说序列化。它很适合做数据存储或 RPC 数据交换格式。可用于通讯协议、数据存储等领域的语言无关、平台无关、可扩展的序列化结构数据格式。

centos下安装protobuf

获取源码包

```
wget https://github.com/google/protobuf/archive/v3.5.0.tar.gz
```

解压缩并进入源码目录

```
tar -zxvf v3.5.0.tar.gz
```

```
cd protobuf-3.5.0
```

生成configure文件

```
./autogen.sh
```

在执行./autogen.sh过程中可能会因缺乏unzip依赖库而报错，要解决此错误：在linux系统可以通过yum -y install zip unzip

在执行./autogen.sh过程中可能会因缺乏automake依赖库而报错：autoreconf: failed to run aclocal: No such file or directory，要解决此错误，在linux系统可以通过sudo yum install automake

在执行./autogen.sh过程中可能会出现下面错误：

```
configure.ac:30: error: possibly undefined macro: AC_PROG_LIBTOOL
```

If this token and others are legitimate, please use m4_pattern_allow.

See the Autoconf documentation.

原因：缺少一个工具：sudo yum install libtool

编译安装

```
./configure
```

```
make
```

```
make check
make install
```

如何使用protobuf

在golang中安装protobuf相关的库

```
go get -u github.com/golang/protobuf/{protoc-gen-go,proto}
```

编写.proto文件

```
$GOPATH/src/test/protobuf/pb/user.proto
```

```
syntax = "proto3";
package pb;

message user {
    int32 id = 1;
    string name = 2;
}

message multi_user {
    repeated user users = 1;
}
```

基于.proto文件生成数据操作代码

```
protoc --go_out=. user.proto
```

执行命令完成，在与user.proto文件同级的目录下生成了一个user.pb.go文件

数据序列化演示

```
$GOPATH/src/test/protobuf/main.go
```

```
package main

import (
    "log"
    "test/protobuf/pb"

    "github.com/golang/protobuf/proto"
)
```

```
func main() {
    user1 := pb.User{
        Id:  *proto.Int32(1),
        Name: *proto.String("Mike"),
    }

    user2 := pb.User{
        Id:  2,
        Name: "John",
    }

    users := pb.MultiUser{
        Users: []*pb.User{&user1, &user2},
    }

    // 序列化数据
    data, err := proto.Marshal(&users)
    if err != nil {
        log.Fatalln("Marshal data error: ", err)
    }
    println(users.Users[0].GetName()) // output: Mike

    // 对已序列化的数据进行反序列化
    var target pb.MultiUser
    err = proto.Unmarshal(data, &target)
    if err != nil {
        log.Fatalln("Unmarshal data error: ", err)
    }
    println(target.GetUsers()[1].Name) // output: John
}
```