# ProtoBuf

## 基础类型

|  |  |  |
| --- | --- | --- |
| proto类型 | java类型 | 备注 |
| double | double |  |
| float | float |  |
| bool | boolean |  |
| int32 | int | 可变长编码。负数不够高效——使用sint32。 |
| int64 | long | 可变长编码。负数不够高效——使用sint64。 |
| uint32 | int[1] |  |
| uint64 | long[1] |  |
| sint32 | int | 可变长编码。有符号的整型值。编码时比通常的int32高效。 |
| sint64 | long | 可变长编码。有符号的整型值。编码时比通常的int64高效。 |
| fixed32 | int[1] | 4个字节。如果数值总是比256大的话，比uint32高效。 |
| fixed64 | long[1] | 8个字节。如果数值总是比256大的话，比uint64高效。 |
| sfixed32 | int | 4个字节。 |
| sfixed64 | long | 8个字节。 |
| string | String | 一个字符串必须是UTF-8编码或者7-bit |
| bytes | ByteString | 可能包含任意顺序的字节数据 |

## 标识符

标识符必须大于0；enum必须从0开始，否则报错，因为有默认值。

[1,15]之内的标识号在编码的时候会占用一个字节。[16,2047]之内的标识号则占用2个字节。

最小的标识号可以从1开始，最大到2^29 - 1, or 536,870,911。不可以使用其中的[19000－19999]的标识号， Protobuf协议实现中对这些进行了预留。如果非要在.proto文件中使用这些预留标识号，编译时就会报警。最小的标识号可以从1开始，最大到2^29 - 1, or 536,870,911。不可以使用其中的[19000－19999]的标识号， Protobuf协议实现中对这些进行了预留。如果非要在.proto文件中使用这些预留标识号，编译时就会报警。

## 保留标识符（reserved）

reserved 2,15,9 to 11; reserved "foo", "bar";

只能同时预定标识号或者变量名。

## Proto3版本更新了什么

移除了required（必须有值），default（支持设定默认值，现为系统默认值）

## .proto文件生成了什么？

编译器为每一个消息类型生成了一个.java文件，以及一个特殊的Builder类（该类是用来创建消息类接口的）。

## import 和 import public

import public可以使用import文件下已import的文件。

## oneof

oneof test\_oneof { string one1 = 19; string one2 = 20;}

只有一个值能被赋值，getTestOneofCase可以得到被设置的域名。

# 如何存储

## int32

标识符<<<3|0

while (true) {

//~0x7F .111 1000 0000:用来判断是否有大于等于7位置  
 if ((value & ~0x7F) == 0) {  
 buffer[position++] = (byte) value;  
 return;  
 } else {

//先取小于等于7位置数据  
 buffer[position++] = (byte) ((value & 0x7F) | 0x80);  
 value >>>= 7;  
 }  
}

先存低位，再存高位，单字节的第8位来表示是否有下一位数据。

## sint

(n << 1) ^ (n >> 31)

正数乘以2，负数乘以2-1。

uint

fixed

sfixed

## double

标识符<<<3|1

## float

uint