

Python 调用*_pb2.py

目录

前言	2
1 下载	2
1.1 下载地址	2
1.2 下载	2
2 安装	3
2.1 配置安装路径	3
3 修改配置文件	3
4 检查	4
5 proto 转成 py 文件	4
6 python 调用*_pb2.py 文件	4

前言

protobuf3 只保留 repeated 标记数组类型, optional 和 required 都被去掉了。

1 下载

1.1 下载地址

<https://github.com/protocolbuffers/protobuf/releases/tag/v3.9.2>

<https://github.com/protocolbuffers/protobuf/releases/tag/v2.6.0>

1.2 下载

wget

<https://github.com/protocolbuffers/protobuf/releases/download/v3.8.0/protobuf-all-3.8.0.tar.gz>

https://github.com/protocolbuffers/protobuf/releases/download/v3.9.1/protoc-3.9.1-linux-x86_64.zip

wget

<https://github.com/protocolbuffers/protobuf/releases/download/v2.6.0/protobuf-2.6.0.tar.gz>

python 安装 protobuf, 用来支持调用 proto 转成 py 后的*_pb2.py 文件,
protobuf2 只支持 3.19.0 以下版本。

pip install protobuf==3.18.0

2 安装

2.1 配置安装路径

没有 configure 文件, 执行./autogen.sh

```
./configure --prefix=/usr1/exec_env/protobuf-3.9.2
```

编译&安装: make && make install

执行 ldconfig 更新动态链接库

3 修改配置文件

使用vim /etc/profile

```
# protobuf 路径
```

```
PROTOBUF_PATH="/usr1/exec_env/protobuf-2.6.0"
```

```
PATH=$PATH:${PYTHON_HOME}/bin:/usr/sbin/shutdown:${PROTOBUF_PATH}/bin/
```

(动态库搜索路径) 程序加载运行期间查找动态链接库时指定除了系统默认路径之外的其他路径

```
export
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${PROTOBUF_PATH}/lib/
```

(静态库搜索路径) 程序编译期间查找动态链接库时指定查找共享库的路径

```
export LIBRARY_PATH=$LIBRARY_PATH:${PROTOBUF_PATH}/lib/
```

c程序头文件搜索路径

```
export
```

```
C_INCLUDE_PATH=$C_INCLUDE_PATH:${PROTOBUF_PATH}/include/
```

c++程序头文件搜索路径

```
export
```

```
CPLUS_INCLUDE_PATH=$CPLUS_INCLUDE_PATH:${PROTOBUF_PATH}/i
```

```
nclude/  
# pkg-config 路径  
export PKG_CONFIG_PATH=${PROTOBUF_PATH}/lib/pkgconfig/
```

4 检查

```
protoc -version
```

5 proto 转成 py 文件

.proto 文件转成 pg2.py, 使用命令

```
protoc person.proto --python_out=./ 或  
protoc --python_out=./ person.proto
```

6 python 调用*_pb2.py 文件

示例:

```
syntax = "proto2";  
package proto_py;  
  
message Proto{  
    required uint32 p_age = 2;  
    required string p_name = 3;  
}  
  
message One{  
    required uint32 id = 1;  
    required Proto proto = 2;  
}  
  
message Two{  
    required uint32 x = 1;  
    required One one = 2;  
}  
  
message Three{  
    required uint32 y = 1;  
    required Two two = 2;  
}
```

```
from proto_pb2 import *

proto = Proto()
proto.p_age = 20
proto.p_name = "张三"
print("person")
print(proto)
print(proto.SerializeToString())
```

```
one = One()
one.id = 3
operson = one.proto
operson.p_age = 30
operson.p_name = "李四"
print("one")
print(one)
```

```
two = Two()
two.x = 1920
tone = two.one
tone.id = 39
tone.proto.p_age = 40
tone.proto.p_name = "王二"
# two.one.MergeFrom(one)
print("two")
print(two)
```

```
three = Three()
three.y = 1080
ttwo = three.two
...
three.two.MergeFrom(two)
print("three")
print(three)
```