

目录

前言	1.1
Protobuf概览	1.2
Protobuf正向	1.3
Protobuf逆向	1.4
普通的iOS类	1.4.1
GPBMessage	1.4.1.1
GPBDescriptor	1.4.1.2
GPBFieldDescriptor	1.4.1.3
lite的C++类	1.4.2
MessageLite	1.4.2.1
MessageLite定义	1.4.2.1.1
YouTube中Protobuf	1.4.3
普通的iOS类	1.4.3.1
YTIPlayerRequest	1.4.3.1.1
YTIAdBreakRequest	1.4.3.1.2
YTIIInnerTubeContext	1.4.3.1.3
YTIClientInfo	1.4.3.1.4
YTIOnesieHotConfig	1.4.3.1.5
lite的C++类	1.4.3.2
OnesieRequestProto	1.4.3.2.1
附录	1.5
参考资料	1.5.1

iOS逆向YouTube: protobuf逆向

- 最新版本: v0.9
- 更新时间: 20221110

简介

整理iOS逆向YouTube期间涉及到的Protobuf的逆向。先是概览；再是Protobuf的正向相关知识；然后是Protobuf逆向内容，包括普通的iOS的类、lite精简版的C++的类、YouTube中Protobuf的内容；其中，普通的iOS的类有GPBMessage、GPBDescriptor、GPBFieldDescriptor；lite精简版C++的类有MessageLite及其定义；YouTube中的普通的iOS类的Protobuf类有：YTIPlayerRequest、YTIAdbreakRequest、YTINnertubeContext、YTIClientInfo、YTIONesieHotConfig，以及lite的C++类有OnesieRequestProto；

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- [crifan/ios_re_protobuf_reverse: iOS逆向YouTube: protobuf逆向](#)

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- [iOS逆向YouTube: protobuf逆向 book.crifan.org](#)
- [iOS逆向YouTube: protobuf逆向 crifan.github.io](#)

离线下载阅读

- [iOS逆向YouTube: protobuf逆向 PDF](#)
- [iOS逆向YouTube: protobuf逆向 ePUB](#)
- [iOS逆向YouTube: protobuf逆向 Mobi](#)

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。如发现有侵权，请通过邮箱联系我 `admin 艾特 crifan.com`，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

其他

作者的其他电子书

本人 `crifan` 还写了其他 150+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

关于作者

关于作者更多介绍，详见：

[关于CrifanLi李茂 – 在路上](#)

crifan.org，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：

2022-11-10 10:35:57

Protobuf概览

在iOS逆向YouTube期间，遇到了 Protobuf 的逆向。

其中涉及到，了解Protobuf的正向内容，和相关逆向内容。

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：

2022-11-04 14:45:51

Protobuf正向

TODO:

- 【整理】 Chromium相关: Protobuf源码和文档
 - 【整理】 Protocol Buffers中.proto文件写法即protocobuf的语法
 - 【已解决】 protobuf中字段规则类型: required和optional
-

- **protobuf**
 - `protobuf = Protocol Buffers`
 - 概述
 - `Protocol buffers` 是Google用于序列化结构化数据的语言中立、平台中立、可扩展机制
 - `Protocol buffers` are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data
 - 特点
 - 比 `JSON`、`XML` 等格式，占用体积更小，更快，更简单
 - 是什么：是一系列的组合 = 包含三部分
 - 定义语言：`definition language`
 - 自己写的：`.proto` 文件
 - 代码 = `code`
 - 需要专门的`protocol compiler` (对应命令行工具是：`protoc`)去编译生成对应语言的代码
 - 用于操作数据的代码
 - 运行时：`language-specific protobuf runtime libraries`
 - 序列化：`serialization format for data`
 - 写入文件
 - 或：用网络传输
 - 相关名词
 - `encode` = 编码 = `serialize` = 序列化
 - 把原始数据编码成protobuf的二进制数据
 - `decode` = 解码 = `deserialize` = 反序列化 = `parse` = 解析
 - 把protobuf的二进制数据解码出原始的数据定义
 - 以对应的语言的类的形式保存
 - =解码出来是对的某种语言的某个protobuf的类
 - 支持多种语言
 - `proto2`
 - `Java`
 - `Python`
 - `Objective-C`
 - `C++`
 - `PHP`
 - `proto3`
 - `Kotlin`
 - `Dart`

- Go
- Ruby
- C#

- 支持场景

- 临时短期的网络传输
- 长期的数据保存

- 特点

- 支持扩展extended=extensions

- Protocol buffers can be extended with new information without invalidating existing data or requiring code to be updated
- In proto2, messages can allow extensions to define fields outside of the message, itself. For example, the protobuf library's internal message schema allows extensions for custom, usage-specific options.

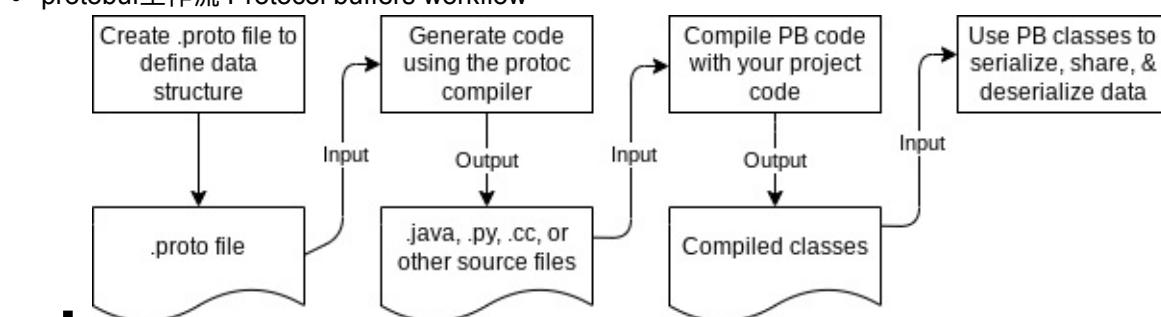
- 兼容性好

- 向后兼容性很好
- 向前兼容性也很好
 - 无缝的支持:
 - 字段的变化
 - 新增字段
 - 删除字段

- 使用Protocol Buffers的好处

- 体积小=占用空间少=数据存储很紧凑
- 解析速度快
- 支持多种语言
- 有各种经过优化的功能（通过编译器生成的类去实现的）

- protobuf工作流 Protocol buffers workflow



- `proto compiler = protoc`

- 输入: `.proto`
- 输出: 对应的特定语言的代码
 - 对每个字段和方法, 都有一个accessor访问器
 - 用于在数据结构 和 二进制数据 之间解析和转换
- 编译生成的文件

- C++ : `.h` 和 `.cc`
- Java : `.java`
- Kotlin : `.kt`
- Python : 生成内容是一个模块module, 放在 `.proto` 文件中
- Go : `.pb.go`
- Ruby : `.rb`

- Objective-C : pbobjc.h 和 pbobjc.m
- C# : .cs
- Dart : .pb.dart

◦ 相关

- google内部常用到的一些protobuf定义
 - timestamp
 - protobuf/timestamp.proto at main · protocolbuffers/protobuf (github.com)
 - status
 - googleapis/status.proto at master · googleapis/googleapis (github.com)

◦ 文档和代码

- 详见：
 - 【整理】 Chromium相关： Protobuf源码和文档

protobuf举例

.proto 定义：

```
message Person {
  required string name = 1;
  required int32 id = 2;
  optional string email = 3;
}
```

-》 序列化： Java 写入文件：

```
Person john = Person.newBuilder()
  .setId(1234)
  .setName("John Doe")
  .setEmail("jdoe@example.com")
  .build();
output = new FileOutputStream(args[0]);
john.writeTo(output);
```

-》 反序列化： C++解析

```
Person john;
fstream input(argv[1],
    ios::in | ios::binary);
john.ParseFromIstream( input );
id = john.id();
name = john.name();
email = john.email();
```

更多例子：

- examples - external/github.com/google/protobuf - Git at Google (googlesource.com)

Protobuf逆向

TODO:

- 【记录】Protobuf的full和lite的Message正向和逆向机制对比

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-11-04 14:50:13

普通的iOS类

TODO:

- 【未解决】Protobuf的iOS的ObjC版的lite的库即runtime运行时
- 【未解决】iOS的ObjC的protobuf的sample示例的正向编译和逆向尝试
- 【已解决】protobuf逆向：已有data如何获取子属性子字段的数据
- 【已解决】protobuf逆向：iOS端无需data直接解析protobuf类的字段定义

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-04 15:43:08

GPBMessage

TODO:

- 【未解决】研究YouTube逻辑: GPBMessage的数据
- 【已解决】研究YouTube逻辑: GPBMessage
- 【已解决】研究YouTube逻辑: 关于GPBMessage子类中protobuf属性字段和顺序的相关理解
- 【未解决】研究YouTube逻辑: GPBMessage的parseFromData

GPBMessage 类的头文件定义

header_ModuleFramework/GPBMessage.h

```
//
//     Generated by class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48).
//
//     class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2015 by Steve Nygard.
//

#import <objc/NSObject.h>

#import <Module_Framework/IGListDiffable-Protocol.h>
#import <Module_Framework/NSCopying-Protocol.h>
#import <Module_Framework/NSSecureCoding-Protocol.h>
#import <Module_Framework/YTVisitable-Protocol.h>

@class GPBExtensionDescriptor, GPBFieldDescriptor, GPBUncKnownFieldSet, NSMutableDictionary, NSString, YTICClientYouTubeData;

@interface GPBMessage : NSObject <IGListDiffable, YTVisitable, NSSecureCoding, NSCopying>
{
    GPBUncKnownFieldSet *unknownFields_;
    NSMutableDictionary *extensionMap_;
    NSMutableDictionary *autoCreatedExtensionMap_;
    GPBMessage *autoCreator_;
    GPBFieldDescriptor *autoCreatorField_;
    GPBExtensionDescriptor *autoCreatorExtension_;
    // Error parsing type: A@, name: readOnlySemaphore_
    struct GPBMessage_Storage *messageStorage_;
}

+ (_Bool)accessInstanceVariablesDirectly;
+ (_Bool)supportsSecureCoding;
+ (_Bool)resolveClassMethod:(SEL)arg1;
+ (_Bool)resolveInstanceMethod:(SEL)arg1;
+ (id)parseDelimitedFromCodedInputStream:(id)arg1 extensionRegistry:(id)arg2 error:(id *)arg3;
+ (id)parseFromCodedInputStream:(id)arg1 extensionRegistry:(id)arg2 error:(id *)arg3;
+ (id)parseFromData:(id)arg1 extensionRegistry:(id)arg2 error:(id *)arg3;
+ (id)parseFromData:(id)arg1 error:(id *)arg2;
+ (id)message;
```

```

+ (id)descriptor;
+ (id)alloc;
+ (id)allocWithZone:(struct _NSZone *)arg1;
+ (void)initialize;
+ (id)goog_parseFromData (id)arg1 error:(id *)arg2;
+ (id)parseFromData:(id)arg1;
+ (id)deserializeFromString:(id)arg1;
+ (_Bool)yt_failOrPopulateError:(id *)arg1 code:(unsigned long long)arg2 reason:(id)arg3;
+ (_Bool)mergeUnknownField:(int)arg1 source:(id)arg2 destination:(id)arg3 mergedUnknownFieldSet:(id)arg4 mask:(id)arg5 options:(id)arg6 error:(id *)arg7;
+ (id)unknownFieldsSetOnSource:(id)arg1 destination:(id)arg2 mask:(id)arg3;
+ (_Bool)yt_mergeUnknownFieldsFromSource:(id)arg1 destination:(id)arg2 mask:(id)arg3 options:(id)arg4 error:(id *)arg5;
+ (void)yt_overwriteExtension:(id)arg1 source:(id)arg2 destination:(id)arg3;
+ (_Bool)yt_mergeExtension:(id)arg1 source:(id)arg2 destination:(id)arg3 mask:(id)arg4 options:(id)arg5 error:(id *)arg6;
+ (id)yt_arrayOfExtensionsSetOnSource:(id)arg1 destination:(id)arg2 mask:(id)arg3;
+ (_Bool)yt_mergeExtensionsFromSource:(id)arg1 destination:(id)arg2 mask:(id)arg3 options:(id)arg4 error:(id *)arg5;
+ (void)yt_overwriteField:(id)arg1 source:(id)arg2 destination:(id)arg3 options:(id)arg4;
+ (_Bool)yt_mergeField:(id)arg1 source:(id)arg2 destination:(id)arg3 mask:(id)arg4 options:(id)arg5 error:(id *)arg6;
+ (void)yt_clearRepeatedField:(id)arg1 message:(id)arg2;
+ (void)yt_appendRepeatedField:(id)arg1 source:(id)arg2 destination:(id)arg3;
+ (id)yt_arrayOfFieldsSetOnSource:(id)arg1 destination:(id)arg2 mask:(id)arg3;
+ (_Bool)yt_mergeFieldsFromSource:(id)arg1 destination:(id)arg2 mask:(id)arg3 options:(id)arg4 error:(id *)arg5;
+ (_Bool)yt_mergeSource:(id)arg1 destination:(id)arg2 mask:(id)arg3 options:(id)arg4 error:(id *)arg5;
- (void)encodeWithCoder:(id)arg1;
- (id)initWithCoder:(id)arg1;
- (unsigned long long)serializedSize;
@property(nonatomic, copy) NSString *description;
@property(nonatomic) unsigned long long hash;
- (_Bool)isEqual:(id)arg1;
- (void)mergeFrom:(id)arg1;
- (void)mergeFromCodedInputStream:(id)arg1 extensionRegistry:(id)arg2;
- (void)addUnknownMapEntry:(int)arg1 value:(id)arg2;
- (_Bool)parseUnknownField:(id)arg1 extensionRegistry:(id)arg2 tag:(unsigned int)arg3;
- (void)parseMessageSet:(id)arg1 extensionRegistry:(id)arg2;
@property(nonatomic, copy, nonatomic) GPBUncalculatedFieldSet *unknownFields;
- (void)mergeDelimitedFromCodedInputStream:(id)arg1 extensionRegistry:(id)arg2;
- (void)mergeFromData:(id)arg1 extensionRegistry:(id)arg2;
- (void)clearExtension:(id)arg1;
- (void)setExtension:(id)arg1 index:(unsigned long long)arg2 value:(id)arg3;
- (void)addExtension:(id)arg1 value:(id)arg2;
- (void)setExtension:(id)arg1 value:(id)arg2;
- (void)writeExtensionsToCodedOutputStream:(id)arg1 range:(struct GPBExtensionRange)arg2 sortedExtensions:(id)arg3;
- (id)extensionsCurrentlySet;
- (_Bool)hasExtension:(id)arg1;
- (id)getExistingExtension:(id)arg1;
- (id)getExtension:(id)arg1;
- (void)writeField:(id)arg1 toCodedOutputStream:(id)arg2;
- (void)writeDelimitedToCodedOutputStream:(id)arg1;
- (void)writeDelimitedToOutputStream:(id)arg1;

```

```

- (void)writeToCodedOutputStream:(id)arg1;
- (void)writeToOutputStream:(id)arg1;
- (id)delimitedData;
- (id)data;
- (id)descriptor;
@property(nonatomic, readonly, getter=isInitialized) _Bool initialized;
- (void)internalClear:(_Bool)arg1;
- (void)clear;
- (id)copyWithZone:(struct _NSZone *)arg1;
- (void)copyFieldsInto:(id)arg1 zone:(struct _NSZone *)arg2 descriptor:(id)arg3;
- (void)dealloc;
- (id)initWithCodedInputStream:(id)arg1 extensionRegistry:(id)arg2 error:(id *)arg3;
- (id)initWithData:(id)arg1 extensionRegistry:(id)arg2 error:(id *)arg3;
- (id)initWithData:(id)arg1 error:(id *)arg2;
- (id)init;
- (void)yt_setLightweightProxyButton:(id)arg1;
- (id)yt_lightweightProxyButton;
- (id)yt_sectionReloadContinuation;
- (id)sectionNextContinuation;
- (id)sectionContentsForCellFactory:(id)arg1;
- (void)yt_lightweightSetVEType:(int)arg1 trackableDataElementObject:(id)arg2;
- (void)reportOneTimeVisibilityUpdateWithParentResponder:(id)arg1;
- (id)targetIDField;
- (id)tooltipTargetID;
- (_Bool)isEqualToString:(id)arg1;
- (id)diffIdentifier;
- (void)logging_enumerateFieldDescriptorsUsingBlock:(CDUnknownBlockType)arg1;
- (id)logging_messageForFieldNumber:(unsigned long long)arg1 messageClass:(Class)arg2;
- (id)yt_trackingParams;
- (id)yt_loggingDirectives;
- (id)yt_visualElement;
@property(nonatomic, setter=yt_setShouldAttachChildProtos) _Bool yt_shouldAttachChildProtos;
@property(nonatomic, retain, nonatomic, setter=yt_setDataElement) GPBMessage *yt_dataElement;
@property(nonatomic, retain, nonatomic, setter=yt_setYouTubeData) YTIClientYouTubeData *yt_youtubeData;
@property(nonatomic, setter=yt_setVECounter) int yt_veCounter;
@property(nonatomic, setter=yt_setVEType) int yt_veType;
- (id)sortedExtensionsCurrentlySet;
- (void)acceptExtension:(id)arg1 withVisitor:(id)arg2;
- (void)acceptVisitor:(id)arg1;
- (void)visitExtension:(id)arg1 withBlock:(CDUnknownBlockType)arg2 endBlock:(CDUnknownBlockType)
arg3 stop:(_Bool *)arg4;
- (void)visitMessageTreeInPreorderWithBlock:(CDUnknownBlockType)arg1 endBlock:(CDUnknownBlockType)
arg2 stop:(_Bool *)arg3;
- (void)visitMessageTreeInPreorderWithBlock:(CDUnknownBlockType)arg1 endBlock:(CDUnknownBlockType)
arg2;
- (void)visitMessageTreeInPreorderWithBlock:(CDUnknownBlockType)arg1;
- (void)enumerateFieldDescriptorsUsingBlock:(CDUnknownBlockType)arg1;
- (id)mutableCopy;
- (void)forEachMessageField:(CDUnknownBlockType)arg1;
- (id)firstValue;
- (id)messageForFieldNumber:(unsigned long long)arg1;
- (id)messageForFieldNumber:(unsigned long long)arg1 messageClass:(Class)arg2;
- (id)firstSubmessage;
- (id)serializedString;
- (id)yt_messageWithMask:(id)arg1 options:(id)arg2 error:(id *)arg3;
- (id)yt_messageByMergingFromSource:(id)arg1 mask:(id)arg2 options:(id)arg3 error:(id *)arg4;

```

```
- (_Bool)yt_mergeFromSource (id)arg1 mask:(id)arg2 options:(id)arg3 error:(id *)arg4;

// Remaining properties
@property(nonatomic, copy) NSString *debugDescription;
@property(nonatomic) Class superclass;

@end
```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-11-10 10:31:11

GPBDescriptor

TODO:

- 【未解决】研究YouTube逻辑: GPBDescriptor

GPBDescriptor 类的头文件定义

header_ModuleFramework/GPBDescriptor.h

```
//
//     Generated by class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48).
//
//     class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2015 by Steve Nygard.
//

#import <objc/NSObject.h>

#import <Module_Framework/NSCopying_Protocol.h>

@class GPBFileDescriptor, NSArray, NSString;

@interface GPBDescriptor : NSObject <NSCopying>
{
    NSArray *fields_;
    NSArray *oneofs_;
    unsigned int storageSize_;
    Class messageClass_;
    GPBFileDescriptor *file_;
    _Bool wireType_;
    unsigned int extensionRangesCount_;
    const struct GPBExtensionRange *extensionRanges_;
}

+ (id)allocDescriptorForClass:(Class)arg1 rootClass:(Class)arg2 file (id)arg3 fields:(void *)arg4 fieldCount:(unsigned int)arg5 storageSize:(unsigned int)arg6 flags:(unsigned int)arg7;
@property(readonly, nonatomic, getter=isWireFormat) _Bool wireType; // @synthesize wireType=wireFormat_;
@property(readonly, nonatomic) GPBFileDescriptor *file; // @synthesize file=file_;
@property(readonly, nonatomic) unsigned int extensionRangesCount; // @synthesize extensionRangesCount=extensionRangesCount_;
@property(readonly, nonatomic) const struct GPBExtensionRange *extensionRanges; // @synthesize extensionRanges=extensionRanges_;
@property(readonly, nonatomic) NSArray *oneofs; // @synthesize oneofs=oneofs_;
@property(readonly, nonatomic) NSArray *fields; // @synthesize fields=fields_;
@property(readonly, nonatomic) Class messageClass; // @synthesize messageClass=messageClass_;
- (id)oneofWithName:(id)arg1;
- (id)fieldWithName:(id)arg1;
- (id)fieldWithNumber:(unsigned int)arg1;
- (id)copyWithZone:(struct _NSZone *)arg1;
@property(readonly) NSString *fullName;
```

```
@property(nonatomic) GPBDescriptor *containingType;
@property(nonatomic, copy, nonatomic) NSString *name;
- (void)setupMessageClassNameSuffix:(id)arg1;
- (void)setupContainingMessageClassName:(const char *)arg1;
- (void)setupContainingMessageClass:(Class)arg1;
- (void)setupExtensionRanges:(const struct GPBExtensionRange *)arg1 count:(int)arg2;
- (void)setupExtraTextInfo:(const char *)arg1;
- (void)setupOneofs:(const char **)arg1 count:(unsigned int)arg2 firstHasIndex:(int)arg3;
- (void)dealloc;
- (id)initWithClass:(Class)arg1 file:(id)arg2 fields:(id)arg3 storageSize:(unsigned int)arg4 willReformat:(_Bool)arg5;

@end
```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-10 10:31:27

GPBFieldDescriptor

TODO:

- 【已解决】研究YouTube逻辑: GPBFieldDescriptor

GPBFieldDescriptor 类的头文件定义

header_ModuleFramework/GPBFieldDescriptor.h

```
//
//     Generated by class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48).
//
//     class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2015 by Steve Nygard.
//

#import <objc/NSObject.h>

@class GPBEnumDescriptor, GPBOneofDescriptor, NSString;

@interface GPBFieldDescriptor : NSObject
{
    struct GPBMessageFieldDescription *description_;
    GPBOneofDescriptor *containingOneof_;
    SEL getSel_;
    SEL setSel_;
    SEL hasOrCountSel_;
    SEL setHasSel_;
    CDUnion_88782d86 defaultValue_;
    Class msgClass_;
    union {
        GPBEnumDescriptor *enumDescriptor_;
        CDUnknownFunctionPointerType enumVerifier_;
    } enumHandling_;
}

@property(nonatomic) GPBOneofDescriptor *containingOneof; // @synthesize containingOneof=containingOneof_;
@property(nonatomic) Class msgClass; // @synthesize msgClass=msgClass_;
- (id)textFormatName;
@property(nonatomic) CDUnion_88782d86 defaultValue;
@property(nonatomic) GPBEnumDescriptor *enumDescriptor;
- (_Bool)isValidEnumValue (int)arg1;
@property(nonatomic, getter=isPackable) _Bool packable;
@property(nonatomic) unsigned char mapKeyDataType;
@property(nonatomic) unsigned char fieldType;
@property(nonatomic, getter=isOptional) _Bool optional;
@property(nonatomic, getter=isRequired) _Bool required;
@property(nonatomic, copy, nonatomic) NSString *name;
@property(nonatomic) unsigned int number;
@property(nonatomic) _Bool hasDefaultValue;
}
```

```
@property(nonatomic) unsigned char dataType;
- (void)dealloc;
- (id)initWithFieldDescription:(void *)arg1 includesDefault:(_Bool)arg2 usesClassRefs:(_Bool)arg3 proto3OptionalKnown:(_Bool)arg4 syntax:(unsigned char)arg5;
- (id)init;

@end
```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-11-10 10:31:45

lite的C++类

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-04 15:43:02

MessageLite

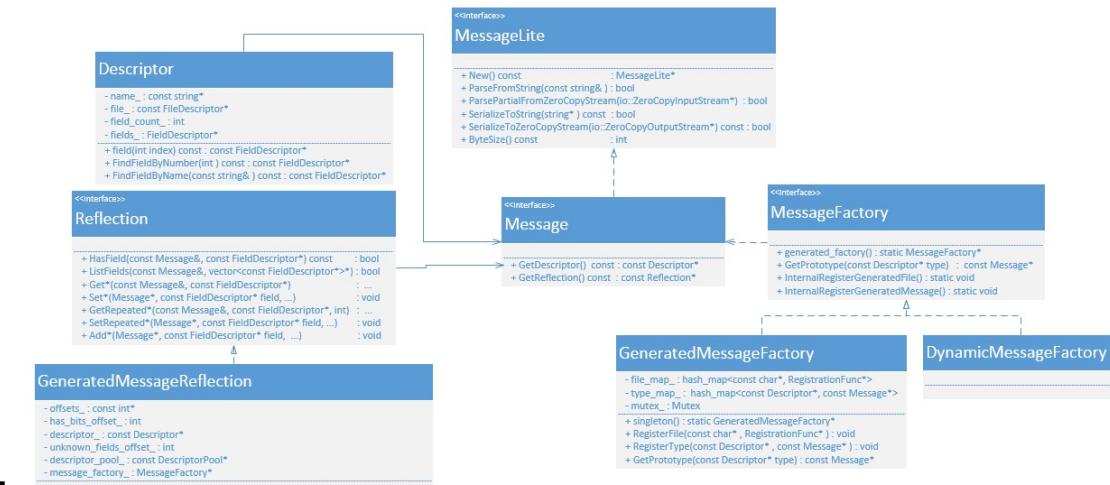
背景

之前在逆向 YouTube 期间遇到过：

```
__const 000000004F7BF60 ; public video_streaming : OnesieRequestProto ;
__const 000000004F7BF60 ; public /* offset 0x0 */ proto2 MessageLite
```

结论

- Protobuf 的 C++ 的类是：Message 和 MessageLite
 - MessageLite 是轻量级= lite 版的 Message
 - Lite 版==编译为 LITE_RUNTIME 版本=runtime是lite版
 - 只有：Encoding = 编码 和 serializing = 序列化
 - 没有：reflection 和 descriptors
 - MessageLite类的继承关系图



LITE_RUNTIME

LITE_RUNTIME 的定义

<https://github.com/alexeyxo/protobuf-objc.git>

中有定义：

- src/compiler/google/protobuf/descriptor.proto

```
// Generated classes can be optimized for speed or code size.
enum OptimizeMode {
    SPEED = 1;           // Generate complete code for parsing, serialization,
                        // etc.
    CODE_SIZE = 2;        // Use ReflectionOps to implement these methods.
```

```

    LITE_RUNTIME = 3; // Generate code using MessageLite and the lite runtime.
}
optional OptimimizeMode optimize_for = 9 [default SPEED];

```

以及相关部分：

- src/compiler/google/protobuf/descriptor.pb.h

```

// nested types ----

typedef FileOptions_OptimizeMode OptimimizeMode;
static const OptimimizeMode SPEED = FileOptions_OptimizeMode_SPEED;
static const OptimimizeMode CODE_SIZE = FileOptions_OptimizeMode_CODE_SIZE;
static const OptimizeMode LITE_RUNTIME = FileOptions_OptimizeMode_LITE_RUNTIME;
static inline bool OptimimizeMode_IsValid(int value) {
    return FileOptions_OptimizeMode_IsValid(value);
}
static const OptimimizeMode OptimimizeMode_MIN =
    FileOptions_OptimizeMode_OptimizeMode_MIN;
static const OptimizeMode OptimizeMode_MAX =
    FileOptions_OptimizeMode_OptimizeMode_MAX;
static const int OptimizeMode_ARRAYSIZE =
    FileOptions_OptimizeMode_OptimizeMode_ARRAYSIZE;
static inline const ::google::protobuf::EnumDescriptor*
OptimizeMode_descriptor() {
    return FileOptions_OptimizeMode_descriptor();
}
static inline const std::string* OptimimizeMode_Name(OptimizeMode value) {
    return FileOptions_OptimizeMode_Name(value);
}
static inline bool OptimimizeMode_Parse(const std::string* name,
    OptimzeMode value) {
    return FileOptions_OptimizeMode_Parse(name, value);
}

```

- src/compiler/google/protobuf/descriptor.pb.cc

```

bool FileOptions::MergePartialFromCodedStream(
    ::google::protobuf::io::CodedInputStream* input) {
#define DO_(EXPRESSION) if (!(EXPRESSION)) goto failure
    ::google::protobuf::uint32 tag;
    // @@protoc_insertion_point(parse_start:google.protobuf.FileOptions)
    for (;;) {
        std::pair<::google::protobuf::uint32, bool> p = input->ReadTagWithCutoff(16383);
        tag = p.first;
        if (!p.second) goto handle_unusual;
        switch (::google::protobuf::internal::WireFormatLite::GetTagFieldNumber(tag)) {
            // optional string java_package = 1;
            case 1: {
                if (tag == 10) {
                    DO_(<::google::protobuf::internal::WireFormatLite::ReadString(
                        input, this->mutable_java_package())));
                    ::google::protobuf::internal::WireFormat::VerifyUTF8StringNamedField(
                        this->java_package().data(), this->java_package().length(),

```

```

    : google::protobuf::internal::WireFormat::PARSE,
    "java_package");
} else {
    goto handle_unusual;
}
if (input >ExpectTag(66)) goto parse_java_outer_classname;
break;
}

// optional .google.protobuf.FileOptions.OptimizeMode optimize_for = 9 [default = SPEED];
case 9: {
    if (tag == 72) {
        parse_optimize_for
        int value;
        DO_(::google::protobuf::internal::WireFormatLite::ReadPrimitive<
            int, ::google::protobuf::internal::WireFormatLite::TYPE_ENUM>(
            input, &value));
        if (!::google::protobuf::FileOptions_OptimizeMode_IsValid(value)) {
            set_optimize_for(static_cast<::google::protobuf::FileOptions_OptimizeMode>(value));
        }
    } else {
        mutable_unknown_fields()->AddVarint(9, value);
    }
} else {
    goto handle_unusual;
}
if (input >ExpectTag(80)) goto parse_java_multiple_files;
break;
}

```

LITE_RUNTIME 的用法

在 .proto 文件中加上定义：

```
option optimize_for = LITE_RUNTIME;
```

就表示：

- 底层实现是用： MessageLite
 - 而不是普通的： Message
- 且不会用到任何有关于 descriptors 或 reflection
- 适用于场景：资源受限
 - 此处YouTube估计是故意的，以便于你很难逆向找到protobuf的定义
 - 注：如果只是想要省空间，则应该改用：
 - optimize_for = CODE_SIZE

Proto2MessageLite

另外，之前下载的

```
google/protobuf/protobuf-refs_heads_master/java/lite.md
```

中也有很多：`Proto3MessageLite`

以及可以搜到很多相关的：

```
java core src test java com google protobuf Proto2ExtensionLookupSchemaTest.java \
java core src test java com google protobuf Proto2LiteSchemaTest.java \
java core src test java com google protobuf Proto2MessageFactory.java \
java core src test java com google protobuf Proto2MessageInfoFactory.java \
java core src test java com google protobuf Proto2MessageLiteFactory.java \
java core src test java com google protobuf Proto2SchemaTest.java \
java core src test java com google protobuf Proto2UnknownEnumValueTest.java \
java core src test java com google protobuf Proto3LiteSchemaTest.java \
java core src test java com google protobuf Proto3MessageFactory.java \
java core src test java com google protobuf Proto3MessageInfoFactory.java \
java core src test java com google protobuf Proto3MessageLiteFactory.java \
java core src test java com google protobuf Proto3MessageLiteInfoFactory.java \
java core src test java com google protobuf Proto3SchemaTest.java
```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-10 10:35:51

MessageLite 的定义

对于 MessageLite 的定义，目前找到了几个地方：

C++ 版 MessageLite

C++ 版 MessageLite 的代码定义

- [protobuf/src/google/protobuf/message_lite.h - chromium/src/third_party - Git at Google](#)

```
// Protocol Buffers - Google's data interchange format
// Copyright 2008 Google Inc. All rights reserved.
// https://developers.google.com/protocol-buffers/
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//     * Redistributions of source code must retain the above copyright
//       notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
//       copyright notice, this list of conditions and the following disclaimer
//       in the documentation and/or other materials provided with the
//       distribution.
//     * Neither the name of Google Inc. nor the names of its
//       contributors may be used to endorse or promote products derived from
//       this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Authors: wink@google.com (Wink Saville),
//          kenton@google.com (Kenton Varda)
// Based on original Protocol Buffers design by
// Sanjay Ghemawat, Jeff Dean, and others.
//
// Defines MessageLite, the abstract interface implemented by all (lite
// and non-lite) protocol message objects.
#ifndef GOOGLE_PROTOBUF_MESSAGE_LITE_H_
#define GOOGLE_PROTOBUF_MESSAGE_LITE_H_
#include <climits>
#include <string>
#include <google/protobuf/stubs/common.h>
```

```

#include <google/protobuf/stubs/logging.h>
#include <google/protobuf/io/coded_stream.h>
#include <google/protobuf/arena.h>
#include <google/protobuf/stubs/once.h>
#include <google/protobuf/port.h>
#include <google/protobuf/stubs/strutil.h>
#include <google/protobuf/explicitly_constructed.h>
#include <google/protobuf/metadata_lite.h>
#include <google/protobuf/stubs/hash.h> // TODO(b/211442718): cleanup
// clang-format off
#include <google/protobuf/port_def.inc>
// clang-format on
#ifndef SWIG
#error "You cannot SWIG proto headers"
#endif
namespace google {
namespace protobuf {
template <typename T>
class RepeatedPtrField;
class FastReflectionMessageMutator;
class FastReflectionStringSetter;
class Reflection;
namespace io {
class CodedInputStream;
class CodedOutputStream;
class ZeroCopyInputStream;
class ZeroCopyOutputStream;
} // namespace io
namespace internal {
class SwapFieldHelper;
// See parse_context.h for explanation
class ParseContext;
class ExtensionSet;
class LazyField;
class RepeatedPtrFieldBase;
class TcParser;
class WireFormatLite;
class WeakFieldMap;
template <typename Type>
class GenericTypeHandler; // defined in repeated_field.h
// We compute sizes as size_t but cache them as int. This function converts a
// computed size to a cached size. Since we don't proceed with serialization
// if the total size was > INT_MAX, it is not important what this function
// returns for inputs > INT_MAX. However this case should not error or
// GOOGLE_CHECK-fail, because the full size_t resolution is still returned from
// ByteSizeLong() and checked against INT_MAX; we can catch the overflow
// there.
inline int ToCachedSize(size_t size) { return static_cast<int>(size); }
// We mainly calculate sizes in terms of size_t, but some functions that
// compute sizes return "int". These int sizes are expected to always be
// positive. This function is more efficient than casting an int to size_t
// directly on 64-bit platforms because it avoids making the compiler emit a
// sign extending instruction, which we don't want and don't want to pay for.
inline size_t FromIntSize(int size) {
    // Convert to unsigned before widening so sign extension is not necessary.
    return static_cast<unsigned int>(size);
}
}
}

```

```

}

// For cases where a legacy function returns an integer size. We GOOGLE_DCHECK()
// that the conversion will fit within an integer; if this is false then we
// are losing information.
inline int ToIntSize(size_t size) {
    GOOGLE_DCHECK_LE(size, static_cast<size_t>(INT_MAX));
    return static_cast<int>(size);
}

// Default empty string object. Don't use this directly. Instead, call
// GetEmptyString() to get the reference. This empty string is aligned with a
// minimum alignment of 8 bytes to match the requirement of ArenaStringPtr.
PROTOBUF_EXPORT extern ExplicitlyConstructedArenaString
    fixed_address_empty_string;
PROTOBUF_EXPORT constexpr const std::string& GetEmptyStringAlreadyInited() {
    return fixed_address_empty_string.get();
}

PROTOBUF_EXPORT size_t StringSpaceUsedExcludingSelfLong(const std::string str);
} // namespace internal

// Interface to light weight protocol messages.

// This interface is implemented by all protocol message objects. Non-lite
// messages additionally implement the Message interface, which is a
// subclass of MessageLite. Use MessageLite instead when you only need
// the subset of features which it supports -- namely, nothing that uses
// descriptors or reflection. You can instruct the protocol compiler
// to generate classes which implement only MessageLite, not the full
// Message interface, by adding the following line to the .proto file:
//
//   option optimize_for = LITE_RUNTIME;
//
// This is particularly useful on resource-constrained systems where
// the full protocol buffers runtime library is too big.
//
// Note that on non-constrained systems (e.g. servers) when you need
// to link in lots of protocol definitions, a better way to reduce
// total code footprint is to use optimize_for = CODE_SIZE. This
// will make the generated code smaller while still supporting all the
// same features (at the expense of speed). optimize_for = LITE_RUNTIME
// is best when you only have a small number of message types linked
// into your binary, in which case the size of the protocol buffers
// runtime itself is the biggest problem.
//
// Users must not derive from this class. Only the protocol compiler and
// the internal library are allowed to create subclasses.
class PROTOBUF_EXPORT MessageLite {
public:
    constexpr MessageLite() {}
    virtual ~MessageLite() = default;
    // Basic Operations -----
    // Get the name of this message type, e.g. "foo.bar.BazProto".
    virtual std::string GetTypeName() const = 0;
    // Construct a new instance of the same type. Ownership is passed to the
    // caller.
    MessageLite* New() const { return New(nullptr); }
    // Construct a new instance on the arena. Ownership is passed to the caller
    // if arena is a nullptr.

```

```

virtual MessageLite* New(Arena* arena) const = 0;
// Returns user-owned arena; nullptr if it's message owned.
Arena* GetArena() const { return _internal_metadata_.user_arena(); }
// Clear all fields of the message and set them to their default values.
// Clear() avoids freeing memory, assuming that any memory allocated
// to hold parts of the message will be needed again to hold the next
// message. If you actually want to free the memory used by a Message,
// you must delete it.
virtual void Clear() = 0;
// Quickly check if all required fields have values set.
virtual bool IsInitialized() const = 0;
// This is not implemented for Lite messages -- it just returns "(cannot
// determine missing fields for lite message)". However, it is implemented
// for full messages. See message.h.
virtual std::string InitializationErrorString() const;
// If |other| is the exact same class as this, calls MergeFrom(). Otherwise,
// results are undefined (probably crash).
virtual void CheckTypeAndMergeFrom(const MessageLite* other) = 0;
// These methods return a human-readable summary of the message. Note that
// since the MessageLite interface does not support reflection, there is very
// little information that these methods can provide. They are shadowed by
// methods of the same name on the Message interface which provide much more
// information. The methods here are intended primarily to facilitate code
// reuse for logic that needs to interoperate with both full and lite protos.
//
// The format of the returned string is subject to change, so please do not
// assume it will remain stable over time.
std::string DebugString() const;
std::string ShortDebugString() const { return DebugString(); }
// MessageLite::DebugString is already Utf8 Safe. This is to add compatibility
// with Message.
std::string Utf8DebugString() const { return DebugString(); }
// Parsing -----
// Methods for parsing in protocol buffer format. Most of these are
// just simple wrappers around MergeFromCodedStream(). Clear() will be
// called before merging the input.
// Fill the message with a protocol buffer parsed from the given input
// stream. Returns false on a read error or if the input is in the wrong
// format. A successful return does not indicate the entire input is
// consumed, ensure you call ConsumedEntireMessage() to check that if
// applicable.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromCodedStream(
    io::CodedInputStream input);
// Like ParseFromCodedStream(), but accepts messages that are missing
// required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromCodedStream(
    io::CodedInputStream input);
// Read a protocol buffer from the given zero-copy input stream. If
// successful, the entire input will be consumed.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromZeroCopyStream(
    io::ZeroCopyInputStream input);
// Like ParseFromZeroCopyStream(), but accepts messages that are missing
// required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromZeroCopyStream(
    io::ZeroCopyInputStream input);
// Parse a protocol buffer from a file descriptor. If successful, the entire

```

```

// input will be consumed.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromFileDescriptor(
    int file_descriptor);
// Like ParseFromFileDescriptor(), but accepts messages that are missing
// required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromFileDescriptor(
    int file_descriptor);
// Parse a protocol buffer from a C++ istream. If successful, the entire
// input will be consumed.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromIstream(std::istream* input);
// Like ParseFromIstream(), but accepts messages that are missing
// required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromIstream(
    std::istream* input);
// Read a protocol buffer from the given zero-copy input stream, expecting
// the message to be exactly "size" bytes long. If successful, exactly
// this many bytes will have been consumed from the input.
bool MergePartialFromBoundedZeroCopyStream(io::ZeroCopyInputStream* input,
                                            int size);
// Like ParseFromBoundedZeroCopyStream(), but accepts messages that are
// missing required fields.
bool MergeFromBoundedZeroCopyStream(io::ZeroCopyInputStream* input, int size);
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromBoundedZeroCopyStream(
    io::ZeroCopyInputStream* input, int size);
// Like ParseFromBoundedZeroCopyStream(), but accepts messages that are
// missing required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromBoundedZeroCopyStream(
    io::ZeroCopyInputStream* input, int size);
// Parses a protocol buffer contained in a string. Returns true on success.
// This function takes a string in the (non-human-readable) binary wire
// format, matching the encoding output by MessageLite::SerializeToString().
// If you'd like to convert a human-readable string into a protocol buffer
// object, see google::protobuf::TextFormat::ParseFromString().
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromString(ConstStringParam data);
// Like ParseFromString(), but accepts messages that are missing
// required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromString(
    ConstStringParam data);
// Parse a protocol buffer contained in an array of bytes.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromArray(const void* data,
                                                    int size);
// Like ParseFromArray(), but accepts messages that are missing
// required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromArray(const void* data,
                                                            int size);
// Reads a protocol buffer from the stream and merges it into this
// Message. Singular fields read from the what is
// already in the Message and repeated fields are appended to those
// already present.
//
// It is the responsibility of the caller to call input->LastTagWas()
// (for groups) or input->ConsumedEntireMessage() (for non-groups) after
// this returns to verify that the message's end was delimited correctly.
//
// ParseFromCodedStream() is implemented as Clear() followed by
// MergeFromCodedStream().

```

```

bool MergeFromCodedStream(io::CodedInputStream* input);
// Like MergeFromCodedStream(), but succeeds even if required fields are
// missing in the input.
//
// MergeFromCodedStream() is just implemented as MergePartialFromCodedStream()
// followed by IsInitialized().
bool MergePartialFromCodedStream(io::CodedInputStream* input);
// Merge a protocol buffer contained in a string.
bool MergeFromString(ConstStringParam data);
// Serialization -----
// Methods for serializing in protocol buffer format. Most of these
// are just simple wrappers around ByteSize() and SerializeWithCachedSizes().
// Write a protocol buffer of this message to the given output. Returns
// false on a write error. If the message is missing required fields,
// this may GOOGLE_CHECK-fail.
bool SerializeToCodedStream(io::CodedOutputStream* output) const;
// Like SerializeToCodedStream(), but allows missing required fields.
bool SerializePartialToCodedStream(io::CodedOutputStream* output) const;
// Write the message to the given zero-copy output stream. All required
// fields must be set.
bool SerializeToZeroCopyStream(io::ZeroCopyOutputStream* output) const;
// Like SerializeToZeroCopyStream(), but allows missing required fields.
bool SerializePartialToZeroCopyStream(io::ZeroCopyOutputStream* output) const;
// Serialize the message and store it in the given string. All required
// fields must be set.
bool SerializeToString(std::string* output) const;
// Like SerializeToString(), but allows missing required fields.
bool SerializePartialToString(std::string* output) const;
// Serialize the message and store it in the given byte array. All required
// fields must be set.
bool SerializeToArray(void* data, int size) const;
// Like SerializeToArray(), but allows missing required fields.
bool SerializePartialToArray(void* data, int size) const;
// Make a string encoding the message. Is equivalent to calling
// SerializeToString() on a string and using that. Returns the empty
// string if SerializeToString() would have returned an error.
// Note: If you intend to generate many such strings, you may
// reduce heap fragmentation by instead re-using the same string
// object with calls to SerializeToString().
std::string SerializeAsString() const;
// Like SerializeAsString(), but allows missing required fields.
std::string SerializePartialAsString() const;
// Serialize the message and write it to the given file descriptor. All
// required fields must be set.
bool SerializeToFileDescriptor(int file_descriptor) const;
// Like SerializeToFileDescriptor(), but allows missing required fields.
bool SerializePartialToFileDescriptor(int file_descriptor) const;
// Serialize the message and write it to the given C++ ostream. All
// required fields must be set.
bool SerializeToOstream(std::ostream* output) const;
// Like SerializeToOstream(), but allows missing required fields.
bool SerializePartialToOstream(std::ostream* output) const;
// Like SerializeToString(), but appends to the data to the string's
// existing contents. All required fields must be set.
bool AppendToString(std::string* output) const;
// Like AppendToString(), but allows missing required fields.

```

```

bool AppendPartialToString(std::string output) const;
// Computes the serialized size of the message. This recursively calls
// ByteSizeLong() on all embedded messages.
//
// ByteSizeLong() is generally linear in the number of fields defined for the
// proto.
virtual size_t ByteSizeLong() const = 0;
// Legacy ByteSize() API.
PROTOBUF_DEPRECATED_MSG("Please use ByteSizeLong() instead")
int ByteSize() const { return internal::ToIntSize(ByteSizeLong()); }
// Serializes the message without recomputing the size. The message must not
// have changed since the last call to ByteSize(), and the value returned by
// ByteSize must be non-negative. Otherwise the results are undefined.
void SerializeWithCachedSizes(io::CodedOutputStream* output) const {
    output->SetCur(_InternalSerialize(output->Cur(), output->EpsCopy()));
}

// Functions below here are not part of the public interface. It isn't
// enforced, but they should be treated as private, and will be private
// at some future time. Unfortunately the implementation of the "friend"
// keyword in GCC is broken at the moment, but we expect it will be fixed.
// Like SerializeWithCachedSizes, but writes directly to *target, returning
// a pointer to the byte immediately after the last byte written. "target"
// must point at a byte array of at least ByteSize() bytes. Whether to use
// deterministic serialization, e.g., maps in sorted order, is determined by
// CodedOutputStream::IsDefaultSerializationDeterministic().
uint8_t* SerializeWithCachedSizesToArray(uint8_t* target) const;
// Returns the result of the last call to ByteSize(). An embedded message's
// size is needed both to serialize it (because embedded messages are
// length-delimited) and to compute the outer message's size. Caching
// the size avoids computing it multiple times.
//
// ByteSize() does not automatically use the cached size when available
// because this would require invalidating it every time the message was
// modified, which would be too hard and expensive. (E.g. if a deeply-nested
// sub-message is changed, all of its parents' cached sizes would need to be
// invalidated, which is too much work for an otherwise inlined setter
// method.)
virtual int GetCachedSize() const = 0;
virtual const char* _InternalParse(const char* /*ptr*/,
                                  internal::ParseContext* /*ctx*/) {
    return nullptr;
}
virtual void OnDemandRegisterArenaDtor(Arena* /*arena*/) {}

protected:
    template <typename T>
    static T* CreateMaybeMessage(Arena* arena) {
        return Arena::CreateMaybeMessage<T>(arena);
    }

    inline explicit MessageLite(Arena* arena, bool is_message_owned = false)
        : _internal_metadata_(arena, is_message_owned) {}

    // Returns the arena, if any, that directly owns this message and its internal
    // memory (Arena::Own is different in that the arena doesn't directly own the
    // internal memory). This method is used in proto's implementation for
    // swapping, moving and setting allocated, for deciding whether the ownership
    // of this message or its internal memory could be changed.
    Arena* GetOwningArena() const { return _internal_metadata_.owning_arena(); }

```

```

// Returns the arena, used for allocating internal objects(e.g., child
// messages, etc), or owning incoming objects (e.g., set allocated).
Arena* GetArenaForAllocation() const { return _internal_metadata_.arena(); }
internal::InternalMetadata _internal_metadata_;
public:
    enum ParseFlags {
        kMerge = 0,
        kParse = 1,
        kMergePartial = 2,
        kParsePartial = 3,
        kMergeWithAliasing = 4,
        kParseWithAliasing = 5,
        kMergePartialWithAliasing = 6,
        kParsePartialWithAliasing = 7
    };
    template <ParseFlags flags, typename T>
    bool ParseFrom(const T* input);
    // Fast path when conditions match (ie. non-deterministic)
    // uint8_t* _InternalSerialize(uint8_t* ptr) const;
    virtual uint8_t* _InternalSerialize(
        uint8_t* ptr, io::EpsCopyOutputStream* stream) const = 0;
    // Identical to IsInitialized() except that it logs an error message.
    bool IsInitializedWithErrors() const {
        if (IsInitialized()) return true;
        LogInitializationErrorMessage();
        return false;
    }
private:
    friend class FastReflectionMessageMutator;
    friend class FastReflectionStringSetter;
    friend class Message;
    friend class Reflection;
    friend class internal::ExtensionSet;
    friend class internal::LazyField;
    friend class internal::SwapFieldHelper;
    friend class internal::TcParser;
    friend class internal::WeakFieldMap;
    friend class internal::WireFormatlite;
    template <typename Type>
    friend class Arena::InternalHelper;
    template <typename Type>
    friend class internal::GenericTypeHandler;
    void LogInitializationErrorMessage() const;
    bool MergeFromImpl(io::CodedInputStream* input, ParseFlags parse_flags);
    GOOGLE_DISALLOW_EVIL_CONSTRUCTORS(MessageLite);
};

namespace internal {
template <bool alias>
bool MergeFromImpl(StringPiece input, MessageLite* msg,
                  MessageLite::ParseFlags parse_flags);
extern template bool MergeFromImpl<false>(StringPiece input,
                                             MessageLite* msg,
                                             MessageLite::ParseFlags parse_flags);
extern template bool MergeFromImpl<true>(StringPiece input,
                                           MessageLite* msg,
                                           MessageLite::ParseFlags parse_flags);
}

```

```

template <bool alias>
bool MergeFromImpl(io::ZeroCopyInputStream* input, MessageLite* msg,
                   MessageLite::ParseFlags parse_flags);
extern template bool MergeFromImpl<false>(io::ZeroCopyInputStream* input,
                                             MessageLite* msg,
                                             MessageLite::ParseFlags parse_flags);
extern template bool MergeFromImpl<true>(io::ZeroCopyInputStream* input,
                                            MessageLite* msg,
                                            MessageLite::ParseFlags parse_flags);

struct BoundedZCIS {
    io::ZeroCopyInputStream* zcis;
    int limit;
};

template <bool alias>
bool MergeFromImpl(BoundedZCIS input, MessageLite* msg,
                   MessageLite::ParseFlags parse_flags);
extern template bool MergeFromImpl<false>(BoundedZCIS input, MessageLite* msg,
                                             MessageLite::ParseFlags parse_flags);
extern template bool MergeFromImpl<true>(BoundedZCIS input, MessageLite* msg,
                                            MessageLite::ParseFlags parse_flags);

template <typename I>
struct SourceWrapper;
template <bool alias, typename I>
bool MergeFromImpl(const SourceWrapper<I>& input, MessageLite* msg,
                   MessageLite::ParseFlags parse_flags) {
    return input.template MergeInto<alias>(msg, parse_flags);
}

} // namespace internal
template MessageLite::ParseFlags flags, typename I>
bool MessageLite::ParseFrom(const T& input) {
    if (flags & kParse) Clear();
    constexpr bool alias = (flags & kMergeWithAliasing) != 0;
    return internal::MergeFromImpl<alias>(input, this, flags);
}

// =====
// Shutdown support.
// Shut down the entire protocol buffers library, deleting all static-duration
// objects allocated by the library or by generated .pb.cc files.
//
// There are two reasons you might want to call this:
// * You use a draconian definition of "memory leak" in which you expect
//   every single malloc() to have a corresponding free(), even for objects
//   which live until program exit.
// * You are writing a dynamically-loaded library which needs to clean up
//   after itself when the library is unloaded.
//
// It is safe to call this multiple times. However, it is not safe to use
// any other part of the protocol buffers library after
// ShutdownProtobufLibrary() has been called. Furthermore this call is not
// thread safe, user needs to synchronize multiple calls.

PROTOBUF_EXPORT void ShutdownProtobufLibrary();

namespace internal {
// Register a function to be called when ShutdownProtocolBuffers() is called.
PROTOBUF_EXPORT void OnShutdown(void (*func)());
// Run an arbitrary function on an arg
PROTOBUF_EXPORT void OnShutdownRun(void (*f)(const void*), const void* arg);

```

```

template <typename T>
T* OnShutdownDelete(T* p) {
    OnShutdownRun([](const void* pp) { delete static_cast<const T*>(pp); }, p);
    return p;
}
} // namespace internal
} // namespace protobuf
} // namespace google
#include <google/protobuf/port_undef.inc>
#endif // GOOGLE_PROTOBUF_MESSAGE_LITE_H_

```

C++ 版 MessageLite 的文档说明

- [message_lite.h | Protocol Buffers | Google Developers](#)

```

message_lite.h

bookmark_border
#include <google/protobuf/message_lite.h>
namespace google::protobuf

Defines MessageLite, the abstract interface implemented by all (lite and non lite) protocol message objects.

Classes in this file
MessageLite
Interface to light weight protocol messages.
File Members
These definitions are not part of any class.
void ShutdownProtobufLibrary()
Shut down the entire protocol buffers library, deleting all static duration objects allocated by the library or by generated .pb.cc files. more...
void protobuf::ShutdownProtobufLibrary()
Shut down the entire protocol buffers library, deleting all static duration objects allocated by the library or by generated .pb.cc files.

There are two reasons you might want to call this

You use a draconian definition of "memory leak" in which you expect every single malloc() to have a corresponding free(), even for objects which live until program exit.
You are writing a dynamically loaded library which needs to clean up after itself when the library is unloaded.
It is safe to call this multiple times. However, it is not safe to use any other part of the protocol buffers library after ShutdownProtobufLibrary() has been called. Furthermore this call is not thread safe, user needs to synchronize multiple calls.

class MessageLite
#include <google/protobuf/message_lite.h>
namespace google::protobuf

Interface to light weight protocol messages.

This interface is implemented by all protocol message objects. Non lite messages additionally implement the Message interface, which is a subclass of MessageLite. Use MessageLite instead whe

```

If you only need the subset of features which it supports – namely, nothing that uses descriptors or reflection. You can instruct the protocol compiler to generate classes which implement only `MessageLite`, not the full `Message` interface, by adding the following line to the `.proto` file

```
option optimize_for = LITE_RUNTIME;
```

This is particularly useful on resource constrained systems where the full protocol buffers runtime library is too big.

Note that on non constrained systems (e.g. servers) when you need to link in lots of protocol definitions, a better way to reduce total code footprint is to use `optimize_for = CODE_SIZE`. This will make the generated code smaller while still supporting all the same features (at the expense of speed). `optimize_for = LITE_RUNTIME` is best when you only have a small number of message types linked into your binary, in which case the size of the protocol buffers runtime itself is the biggest problem.

Users must not derive from this class. Only the protocol compiler and the internal library are allowed to create subclasses.

Known subclasses

`Message`

Members

```
enum ParseFlags
```

more...

```
protected internal ::InternalMetadata _internal_metadata_
```

```
constexpr MessageLite()
```

```
virtual ~MessageLite()
```

```
template bool ParseFrom(const T & input)
```

```
virtual uint8 * _InternalSerialize(uint8 * ptr, io::EpsCopyOutputStream * stream) const = 0
```

Fast path when conditions match (ie. more...)

```
bool IsInitializedWithErrors() const
```

Identical to `IsInitialized()` except that it logs an error message.

```
protected template static T * CreateMaybeMessage(Arena * arena)
```

```
protected explicit MessageLite(Arena * arena)
```

```
protected Arena * GetOwningArena() const
```

Returns the arena, if any, that directly owns this message and its internal memory (`Arena::Own` is different in that the arena doesn't directly own the internal memory). more...

```
protected Arena * GetArenaForAllocation() const
```

Returns the arena, used for allocating internal objects (e.g., child messages, etc), or owning incoming objects (e.g., set allocated).

Basic Operations

```
virtual std::string GetTypeName() const = 0
```

Get the name of this message type, e.g. "foo.bar.BazProto".

```
virtual MessageLite * New() const = 0
```

Construct a new instance of the same type. more...

```
virtual MessageLite * New(Arena * arena) const
```

Construct a new instance on the arena. more...

```
Arena * GetArena() const
```

Same as `GetOwningArena`.

```
void * GetMaybeArenaPointer() const
```

Get a pointer that may be equal to this message's arena, or may not be. more...

```
virtual void Clear() = 0
```

Clear all fields of the message and set them to their default values. more...

```
virtual bool IsInitialized() const = 0
```

Quickly check if all required fields have values set.

```

virtual std::string InitializationErrorMessage() const
This is not implemented for Lite messages - it just returns "(cannot determine missing fields for lite message)". more...
virtual void CheckTypeAndMergeFrom(const MessageLite & other) = 0
If |other| is the exact same class as this, calls MergeFrom(). more...
std::string DebugString() const
These methods return a human readable summary of the message. more...
std::string ShortDebugString() const
std::string Utf8DebugString() const
MessageLite::DebugString is already Utf8 Safe. more...
Parsing
Methods for parsing in protocol buffer format.

```

Most of these are just simple wrappers around MergeFromCodedStream(). Clear() will be called before merging the input.

```

PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromCodedStream(io::CodedInputStream * input)
Fill the message with a protocol buffer parsed from the given input stream. more...
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromCodedStream(io::CodedInputStream * input)
Like ParseFromCodedStream(), but accepts messages that are missing required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromZeroCopyStream(io::ZeroCopyInputStream * input)
Read a protocol buffer from the given zero copy input stream. more...
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromZeroCopyStream(io::ZeroCopyInputStream * input)
Like ParseFromZeroCopyStream(), but accepts messages that are missing required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromFileDescriptor(int file_descriptor)
Parse a protocol buffer from a file descriptor. more...
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromFileDescriptor(int file_descriptor)
Like ParseFromFileDescriptor(), but accepts messages that are missing required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromIstream(std::istream * input)
Parse a protocol buffer from a C++ istream. more...
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromIstream(std::istream * input)
Like ParseFromIstream(), but accepts messages that are missing required fields.
bool MergePartialFromBoundedZeroCopyStream(io::ZeroCopyInputStream * input, int size)
Read a protocol buffer from the given zero copy input stream, expecting the message to be exactly "size" bytes long. more...
bool MergeFromBoundedZeroCopyStream(io::ZeroCopyInputStream * input, int size)
Like ParseFromBoundedZeroCopyStream(), but accepts messages that are missing required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromBoundedZeroCopyStream(io::ZeroCopyInputStream * input, int size)
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromBoundedZeroCopyStream(io::ZeroCopyInputStream * input, int size)
Like ParseFromBoundedZeroCopyStream(), but accepts messages that are missing required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromString(ConstStringParam data)
Parses a protocol buffer contained in a string. more...
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromString(ConstStringParam data)
Like ParseFromString(), but accepts messages that are missing required fields.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParseFromArray(const void * data, int size)
Parse a protocol buffer contained in an array of bytes.
PROTOBUF_ATTRIBUTE_REINITIALIZES bool ParsePartialFromArray(const void * data, int size)
Like ParseFromArray(), but accepts messages that are missing required fields.
bool MergeFromCodedStream(io::CodedInputStream * input)
Reads a protocol buffer from the stream and merges it into this Message. more...
bool MergePartialFromCodedStream(io::CodedInputStream * input)
Like MergeFromCodedStream(), but succeeds even if required fields are missing in the input. more...
...

```

```

bool MergeFromString(ConstStringParam data)
Merge a protocol buffer contained in a string.
Serialization
Methods for serializing in protocol buffer format.

Most of these are just simple wrappers around ByteSize() and SerializeWithCachedSizes().

bool SerializeToCodedStream(io::CodedOutputStream * output) const
Write a protocol buffer of this message to the given output. more...
bool SerializePartialToCodedStream(io::CodedOutputStream * output) const
Like SerializeToCodedStream(), but allows missing required fields.
bool SerializeToZeroCopyStream(io::ZeroCopyOutputStream * output) const
Write the message to the given zero copy output stream. more...
bool SerializePartialToZeroCopyStream(io::ZeroCopyOutputStream * output) const
Like SerializeToZeroCopyStream(), but allows missing required fields.
bool SerializeToString(std::string * output) const
Serialize the message and store it in the given string. more...
bool SerializePartialToString(std::string * output) const
Like SerializeToString(), but allows missing required fields.
bool SerializeToArray(void * data, int size) const
Serialize the message and store it in the given byte array. more...
bool SerializePartialToArray(void * data, int size) const
Like SerializeToArray(), but allows missing required fields.
std::string SerializeAsString() const
Make a string encoding the message. more...
std::string SerializePartialAsString() const
Like SerializeAsString(), but allows missing required fields.
bool SerializeToFileDescriptor(int file_descriptor) const
Serialize the message and write it to the given file descriptor. more...
bool SerializePartialToFileDescriptor(int file_descriptor) const
Like SerializeToFileDescriptor(), but allows missing required fields.
bool SerializeToOstream(std::ostream * output) const
Serialize the message and write it to the given C++ ostream. more...
bool SerializePartialToOstream(std::ostream * output) const
Like SerializeToOstream(), but allows missing required fields.
bool AppendToString(std::string * output) const
Like SerializeToString(), but appends to the data to the string's existing contents. more...
bool AppendPartialToString(std::string * output) const
Like AppendToString(), but allows missing required fields.
virtual size_t ByteSizeLong() const = 0
Computes the serialized size of the message. more...
int ByteSize() const
Legacy ByteSize() API.
void SerializeWithCachedSizes(io::CodedOutputStream * output) const
Serializes the message without recomputing the size. more...
uint8 * SerializeWithCachedSizesToArray(uint8 * target) const
Like SerializeWithCachedSizes, but writes directly to target, returning a pointer to the byte
immediately after the last byte written. more...
virtual int GetCachedSize() const = 0
Returns the result of the last call to ByteSize(). more...
virtual const char * _InternalParse(const char *, internal::ParseContext * )
enum MessageLite::ParseFlags {
    kMerge = 0,
    kParse = 1,
    kMergePartial = 2,
    kParsePartial = 3,
}

```

```

kMergeWithAliasing = 4,
kParseWithAliasing = 5,
kMergePartialWithAliasing = 6,
kParsePartialWithAliasing = 7
}

kMerge kParse kMergePartial kParsePartial kMergeWithAliasing kParseWithAliasing kMergePartialWithAliasing kParsePartialWithAliasing virtual uint8 * MessageLite::InternalSerialize(
    uint8 * ptr,
    io::EpsCopyOutputStream * stream) const = 0
Fast path when conditions match (ie.

non deterministic) uint8 * InternalSerialize(uint8 * ptr) const;

protected Arena * MessageLite::GetOwningArena() const
Returns the arena, if any, that directly owns this message and its internal memory (Arena::Own is different in that the arena doesn't directly own the internal memory).

This method is used in proto's implementation for swapping, moving and setting allocated, for deciding whether the ownership of this message or its internal memory could be changed.

virtual MessageLite * MessageLite::New() const = 0
Construct a new instance of the same type.

Ownership is passed to the caller.

virtual MessageLite * MessageLite::New(
    Arena * arena) const
Construct a new instance on the arena.

Ownership is passed to the caller if arena is a NULL. Default implementation for backwards compatibility.

void * MessageLite::GetMaybeArenaPointer() const
Get a pointer that may be equal to this message's arena, or may not be.

If the value returned by this method is equal to some arena pointer, then this message is on that arena; however, if this message is on some arena, this method may or may not return that arena's pointer. As a tradeoff, this method may be more efficient than GetArena(). The intent is to allow underlying representations that use e.g. tagged pointers to sometimes store the arena pointer directly, and sometimes in a more indirect way, and allow a fastpath comparison against the arena pointer when it's easy to obtain.

virtual void MessageLite::Clear() = 0
Clear all fields of the message and set them to their default values.

Clear() avoids freeing memory, assuming that any memory allocated to hold parts of the message will be needed again to hold the next message. If you actually want to free the memory used by a Message, you must delete it.

virtual std::string MessageLite::InitializationErrorString() const
This is not implemented for Lite messages - it just returns "(cannot determine missing fields for lite message)".

However, it is implemented for full messages. See message.h.

virtual void MessageLite::CheckTypeAndMergeFrom(

```

```
const MessageLite & other) = 0
If |other| is the exact same class as this, calls MergeFrom().
```

Otherwise, results are undefined (probably crash).

```
std::string MessageLite::DebugString() const
```

These methods return a human readable summary of the message.

Note that since the MessageLite interface does not support reflection, there is very little information that these methods can provide. They are shadowed by methods of the same name on the Message interface which provide much more information. The methods here are intended primarily to facilitate code reuse for logic that needs to interoperate with both full and lite protos.

The format of the returned string is subject to change, so please do not assume it will remain stable over time.

```
std::string MessageLite::Utf8DebugString() const
```

MessageLite::DebugString is already Utf8 Safe.

This is to add compatibility with Message.

```
PROTOBUF_ATTRIBUTE_REINITIALIZES bool
```

```
MessageLite::ParseFromCodedStream(
    io::CodedInputStream * input)
```

Fill the message with a protocol buffer parsed from the given input stream.

Returns false on a read error or if the input is in the wrong format. A successful return does not indicate the entire input is consumed, ensure you call ConsumedEntireMessage() to check that if applicable.

```
PROTOBUF_ATTRIBUTE_REINITIALIZES bool
```

```
MessageLite::ParseFromZeroCopyStream(
    io::ZeroCopyInputStream * input)
```

Read a protocol buffer from the given zero copy input stream.

If successful, the entire input will be consumed.

```
PROTOBUF_ATTRIBUTE_REINITIALIZES bool
```

```
MessageLite::ParseFromFileDescriptor(
    int file_descriptor)
```

Parse a protocol buffer from a file descriptor.

If successful, the entire input will be consumed.

```
PROTOBUF_ATTRIBUTE_REINITIALIZES bool
```

```
MessageLite::ParseFromIstream(
    std::istream * input)
```

Parse a protocol buffer from a C++ istream.

If successful, the entire input will be consumed.

```
bool MessageLite::MergePartialFromBoundedZeroCopyStream(
```

```
    io::ZeroCopyInputStream * input,
    int size)
```

Read a protocol buffer from the given zero copy input stream, expecting the message to be exactly "size" bytes long.

If successful, exactly this many bytes will have been consumed from the input.

```
PROTOBUF_ATTRIBUTE_REINITIALIZES bool
MessageLite::ParseFromString(
    const string& data)
Parses a protocol buffer contained in a string.
```

Returns true on success. This function takes a string in the (non human readable) binary wire format, matching the encoding output by `MessageLite::SerializeToString()`. If you'd like to convert a human readable string into a protocol buffer object, see `google::protobuf::TextFormat::ParseFromString()`.

```
bool MessageLite::MergeFromCodedStream(
    io::CodedInputStream* input)
Reads a protocol buffer from the stream and merges it into this Message.
```

Singular fields read from the what is already in the Message and repeated fields are appended to those already present.

It is the responsibility of the caller to call `input->LastTagWas()` (for groups) or `input->ConsumedEntireMessage()` (for non groups) after this returns to verify that the message's end was delimited correctly.

`ParseFromCodedStream()` is implemented as `Clear()` followed by `MergeFromCodedStream()`.

```
bool MessageLite::MergePartialFromCodedStream(
    io::CodedInputStream* input)
Like MergeFromCodedStream(), but succeeds even if required fields are missing in the input.
```

`MergeFromCodedStream()` is just implemented as `MergePartialFromCodedStream()` followed by `IsInitialized()`.

```
bool MessageLite::SerializeToCodedStream(
    io::CodedOutputStream* output) const
Write a protocol buffer of this message to the given output.
```

Returns false on a write error. If the message is missing required fields, this may GOOGLE_CHECK fail.

```
bool MessageLite::SerializeToZeroCopyStream(
    io::ZeroCopyOutputStream* output) const
Write the message to the given zero copy output stream.
```

All required fields must be set.

```
bool MessageLite::SerializeToString(
    std::string* output) const
Serialize the message and store it in the given string.
```

All required fields must be set.

```
bool MessageLite::SerializeToArray(
    void* data,
    int size) const
Serialize the message and store it in the given byte array.
```

All required fields must be set.

```
std::string MessageLite::SerializeAsString() const
Make a string encoding the message.
```

Is equivalent to calling `SerializeToString()` on a string and using that. Returns the empty string if `SerializeToString()` would have returned an error. Note If you intend to generate many such strings, you may reduce heap fragmentation by instead reusing the same string object with calls to `SerializeToString()`.

```
bool MessageLite::SerializeToFileDescriptor(
    int file_descriptor) const
Serialize the message and write it to the given file descriptor.
```

All required fields must be set.

```
bool MessageLite::SerializeToOstream(
    std::ostream * output) const
Serialize the message and write it to the given C++ ostream.
```

All required fields must be set.

```
bool MessageLite::AppendToString(
    std::string * output) const
Like SerializeToString(), but appends to the data to the string's existing contents.
```

All required fields must be set.

```
virtual size_t MessageLite::ByteSizeLong() const = 0
Computes the serialized size of the message.
```

This recursively calls `ByteSizeLong()` on all embedded messages.

`ByteSizeLong()` is generally linear in the number of fields defined for the proto.

```
void MessageLite::SerializeWithCachedSizes(
    io::CodedOutputStream * output) const
Serializes the message without recomputing the size.
```

The message must not have changed since the last call to `ByteSize()`, and the value returned by `ByteSize` must be non negative. Otherwise the results are undefined.

```
uint8 * MessageLite::SerializeWithCachedSizesToArray(
    uint8 * target) const
Like SerializeWithCachedSizes, but writes directly to target, returning a pointer to the byte immediately after the last byte written.
```

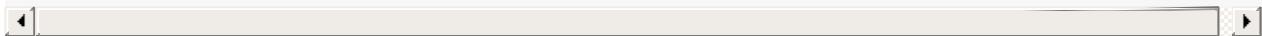
"`target`" must point at a byte array of at least `ByteSize()` bytes. Whether to use deterministic serialization, e.g., maps in sorted order, is determined by `CodedOutputStream::IsDefaultSerializationDeterministic()`.

```
virtual int MessageLite::GetCachedSize() const = 0
Returns the result of the last call to ByteSize().
```

An embedded message's size is needed both to serialize it (because embedded messages are length-

delimited) and to compute the outer message's size. Caching the size avoids computing it multiple times.

`ByteSize()` does not automatically use the cached size when available because this would require invalidating it every time the message was modified, which would be too hard and expensive. (E.g. if a deeply nested sub message is changed, all of its parents' cached sizes would need to be invalidated, which is too much work for an otherwise inlined setter method.)



其中可以看到有枚举定义：

```
enum MessageLite::ParseFlags {
    kMerge = 0,
    kParse = 1,
    kMergePartial = 2,
    kParsePartial = 3,
    kMergeWithAliasing = 4,
    kParseWithAliasing = 5,
    kMergePartialWithAliasing = 6,
    kParsePartialWithAliasing = 7
}
```

Java 版 MessageLite

Java 版 MessageLite 的代码定义

- java/core/src/main/java/com/google/protobuf/MessageLite.java - external/github.com/google/protobuf
- Git at Google (googlesource.com)

```
// Protocol Buffers - Google's data interchange format
// Copyright 2008 Google Inc. All rights reserved.
// https://developers.google.com/protocol-buffers/
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//     * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
//     * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
```

```

// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// TODO(kenton): Use generics? E.g. Builder<BuilderType extends Builder>, then
// mergeFrom*() could return BuilderType for better type-safety.

package           ;
import      IOException;
import      InputStream;
import      OutputStream;
/***
 * Abstract interface implemented by Protocol Message objects.
 *
 * <p>This interface is implemented by all protocol message objects. Non-lite messages additionally
 * implement the Message interface, which is a subclass of MessageLite. Use MessageLite instead
 * when
 * you only need the subset of features which it supports -- namely, nothing that uses descriptors
 * or reflection. You can instruct the protocol compiler to generate classes which implement only
 * MessageLite, not the full Message interface, by adding the follow line to the .proto file:
 *
 * <pre>
 *   option optimize_for = LITE_RUNTIME;
 * </pre>
 *
 * <p>This is particularly useful on resource-constrained systems where the full protocol buffers
 * runtime library is too big.
 *
 * <p>Note that on non-constrained systems (e.g. servers) when you need to link in lots of protocol
 * definitions, a better way to reduce total code footprint is to use {@code optimize_for =
 * CODE_SIZE}. This will make the generated code smaller while still supporting all the same
 * features (at the expense of speed). {@code optimize_for = LITE_RUNTIME} is best when you only
 * have a small number of message types linked into your binary, in which case the size of the
 * protocol buffers runtime itself is the biggest problem.
 *
 * @author kenton@google.com Kenton Varda
 */
public interface MessageLite extends MessageLiteOnBuilder {
    /**
     * Serializes the message and writes it to {@code output}. This does not flush or close the
     * stream.
     */
    void writeTo(CodedOutputStream output) throws IOException;
    /**
     * Get the number of bytes required to encode this message. The result is only computed on the
     * first call and memoized after that.
     *
     * If this message requires more than Integer.MAX_VALUE bytes to encode, the return value will

```

```

1   * be smaller than the actual number of bytes required and might be negative.
  */
int getSerializedSize();
/** Gets the parser for a message of the same type as this message. */
Parser<? extends MessageLite> getParserForType();
// -----
// Convenience methods.
/**
 * Serializes the message to a {@code ByteString} and returns it. This is just a trivial wrapper
 * around {@link #writeTo(CodedOutputStream)}.
 *
 * If this message requires more than Integer.MAX_VALUE bytes to encode, the behavior is
 * unpredictable. It may throw a runtime exception or truncate or slice the data.
 */
ByteString toByteString();
/**
 * Serializes the message to a {@code byte} array and returns it. This is just a trivial wrapper
 * around {@link #writeTo(CodedOutputStream)}.
 *
 * If this message requires more than Integer.MAX_VALUE bytes to encode, the behavior is
 * unpredictable. It may throw a runtime exception or truncate or slice the data.
 */
byte[] toByteArray();
/**
 * Serializes the message and writes it to {@code output}. This is just a trivial wrapper around
 * {@link #writeTo(CodedOutputStream)}. This does not flush or close the stream.
 *
 * <p>NOTE: Protocol Buffers are not self-delimiting. Therefore, if you write any more data to the
 * stream after the message, you must somehow ensure that the parser on the receiving end does not
 * interpret this as being part of the protocol message. This can be done e.g. by writing the size
 * of the message before the data, then making sure to limit the input to that size on the
 * receiving end (e.g. by wrapping the InputStream in one which limits the input). Alternatively,
 * just use {@link #writeDelimitedTo(OutputStream)}.
 */
void writeTo(OutputStream output) throws IOException;
/**
 * Like {@link #writeTo(OutputStream)}, but writes the size of the message as a varint before
 * writing the data. This allows more data to be written to the stream after the message without
 * the need to delimit the message data yourself. Use {@link
 * Builder#mergeDelimitedFrom(InputStream)} (or the static method {@code
 * YourMessageType.parseDelimitedFrom(InputStream)}) to parse messages written by this method
 *
 */
void writeDelimitedTo(OutputStream output) throws IOException;
// -----
// Builders
/** Constructs a new builder for a message of the same type as this message. */

```

```

Builder newBuilderForType();
/**
 * Constructs a builder initialized with the current message. Use this to derive a new message
 *
 * from the current one.
 */
Builder toBuilder();
/** Abstract interface implemented by Protocol Message builders. */
interface Builder extends MessageLiteOrBuilder, Cloneable {
    /** Resets all fields to their default values. */
    Builder clear();
    /**
     * Constructs the message based on the state of the Builder. Subsequent changes to the Builder
     * will not affect the returned message.
     *
     * @throws UninitializedMessageException The message is missing one or more required fields
     *      (i.e. {@link #isInitialized()} returns false). Use {@link #buildPartial()} to bypass
     *      this
     *      check.
     */
    MessageLite build();
    /**
     * Like {@link #build()}, but does not throw an exception if the message is missing required
     * fields. Instead, a partial message is returned. Subsequent changes to the Builder will not
     * affect the returned message.
     */
    MessageLite buildPartial();
    /**
     * Clones the Builder.
     *
     * @see Object#clone()
     */
    Builder clone();
    /**
     * Parses a message of this type from the input and merges it with this message.
     *
     * <p>Warning: This does not verify that all required fields are present in the input message.
     * If you call {@link #build()} without setting all required fields, it will throw an {@link
     * UninitializedMessageException}, which is a {@code RuntimeException} and thus might not be
     * caught. There are a few good ways to deal with this:
     *
     * <ul>
     *   <li>Call {@link #isInitialized()} to verify that all required fields are set before
     *       building.
     *   <li>Use {@code buildPartial()} to build, which ignores missing required fields.
     * </ul>
     *
     * <p>Note: The caller should call {@link CodedInputStream#checkLastTagWas(int)} after calling
     * this to verify that the last tag seen was the appropriate end-group tag, or zero for EOF
     */
}

```

```

        */
    Builder mergeFrom(CodedInputStream input) throws IOException;
    /**
     * Like {@link Builder#mergeFrom(CodedInputStream)}, but also parses extensions. The extensions
     * that you want to be able to parse must be registered in {@code extensionRegistry}. Extensions
     * not in the registry will be treated as unknown fields.
    */
    Builder mergeFrom(CodedInputStream input, ExtensionRegistryLite extensionRegistry)
        throws IOException;
    // -----
    // Convenience methods.
    /**
     * Parse {@code data} as a message of this type and merge it with the message being built.
This
     * is just a small wrapper around {@link #mergeFrom(CodedInputStream)}.
    *
    * @return this
    */
    Builder mergeFrom(ByteString data) throws InvalidProtocolBufferException;
    /**
     * Parse {@code data} as a message of this type and merge it with the message being built.
This
     * is just a small wrapper around {@link #mergeFrom(CodedInputStream,ExtensionRegistryLite)}.
    *
    * @return this
    */
    Builder mergeFrom(ByteString data, ExtensionRegistryLite extensionRegistry)
        throws InvalidProtocolBufferException;
    /**
     * Parse {@code data} as a message of this type and merge it with the message being built.
This
     * is just a small wrapper around {@link #mergeFrom(CodedInputStream)}.
    *
    * @return this
    */
    Builder mergeFrom(byte[] data) throws InvalidProtocolBufferException;
    /**
     * Parse {@code data} as a message of this type and merge it with the message being built.
This
     * is just a small wrapper around {@link #mergeFrom(CodedInputStream)}.
    *
    * @return this
    */
    Builder mergeFrom(byte[] data, int off, int len) throws InvalidProtocolBufferException;
    /**
     * Parse {@code data} as a message of this type and merge it with the message being built.
This
     * is just a small wrapper around {@link #mergeFrom(CodedInputStream,ExtensionRegistryLite)}.
    *
    * @return this
    */

```

```

Builder mergeFrom(byte[] data, ExtensionRegistryLite extensionRegistry)
    throws InvalidProtocolBufferException;
/**
 * Parse {@code data} as a message of this type and merge it with the message being built.
This
 * is just a small wrapper around {@link #mergeFrom(CodedInputStream,ExtensionRegistryLite)}
}.
*
* @return this
*/
Builder mergeFrom(byte[] data, int off, int len, ExtensionRegistryLite extensionRegistry)
    throws InvalidProtocolBufferException;
/**
 * Parse a message of this type from {@code input} and merge it with the message being built.
* This is just a small wrapper around {@link #mergeFrom(CodedInputStream)}. Note that this
 * method always reads the <i>entire</i> input (unless it throws an exception). If you want
it
 * to stop earlier, you will need to wrap your input in some wrapper stream that limits reading.
 * Or, use {@link MessageLite#writeDelimitedTo(OutputStream)} to write your message and {@link
}
 * #mergeDelimitedFrom(InputStream)} to read it.
*
* <p>Despite usually reading the entire input, this does not close the stream.
*
* @return this
*/
Builder mergeFrom(InputStream input) throws IOException;
/**
 * Parse a message of this type from {@code input} and merge it with the message being built.
* This is just a small wrapper around {@link
 * #mergeFrom(CodedInputStream,ExtensionRegistryLite)}.
*
* @return this
*/
Builder mergeFrom(InputStream input, ExtensionRegistryLite extensionRegistry)
    throws IOException;
/**
 * Merge {@code other} into the message being built. {@code other} must have the exact same
type
 * as {@code this} (i.e. {@code getClass().equals(getDefaultInstanceForType().getClass())})
.
*
* <p>Merging occurs as follows. For each field:
*
* * For singular primitive fields, if the field is set in {@code other}, then {@code other}
's
 * value overwrites the value in this message.
*
* * For singular message fields, if the field is set in {@code other}, it is merged into the
he
 * corresponding sub-message of this message using the same merging rules.
*
* * For repeated fields, the elements in {@code other} are concatenated with the elements

```

```

in
    * this message. * For oneof groups, if the other message has one of the fields set, the gr
oup
    * of this message is cleared and replaced by the field of the other message, so that the o
neof
    * constraint is preserved.
    *
    * <p>This is equivalent to the {@code Message::MergeFrom} method in C++.
    */
    Builder mergeFrom(MessageLite other);
    /**
     * Like {@link #mergeFrom(InputStream)}, but does not read until EOF. Instead, the size of
the
     * message (encoded as a varint) is read first, then the message data. Use {@link
     * MessageLite#writeDelimitedTo(OutputStream)} to write messages in this format.
     *
     * @return True if successful, or false if the stream is at EOF when the method starts. Any
     *         other error (including reaching EOF during parsing) will cause an exception to be th
rown.
    */
    boolean mergeDelimitedFrom(InputStream input) throws IOException;
    /**
     * Like {@link #mergeDelimitedFrom(InputStream)} but supporting extensions. */
    boolean mergeDelimitedFrom(InputStream input, ExtensionRegistryLite extensionRegistry)
        throws IOException;
}
}
}

```

-> 对应的函数定义：

- public interface MessageLite extends MessageLiteOrBuilder {
 - void writeTo(CodedOutputStream output) throws IOException;
 - int getSerializedSize();
 - Parser<? extends MessageLite> getParserForType();
 - ByteString toByteString();
 - byte[] toByteArray();
 - void writeTo(OutputStream output) throws IOException;
 - void writeDelimitedTo(OutputStream output) throws IOException;
 - Builders
 - Builder newBuilderForType();
 - Builder toBuilder();
 - interface Builder extends MessageLiteOrBuilder, Cloneable {
 - Builder clear();
 - MessageLite build();
 - MessageLite buildPartial();
 - Builder clone();
 - Builder mergeFrom(CodedInputStream input) throws IOException;
 - Builder mergeFrom(CodedInputStream input, ExtensionRegistryLite
extensionRegistry) throws IOException;
 - Builder mergeFrom(ByteString data) throws InvalidProtocolBufferException;
 - Builder mergeFrom(ByteString data, ExtensionRegistryLite extensionRegistry)

```

        throws InvalidProtocolBufferException;
    ■ Builder mergeFrom(byte[] data) throws InvalidProtocolBufferException;
    ■ Builder mergeFrom(byte[] data, int off, int len) throws
        InvalidProtocolBufferException;
    ■ Builder mergeFrom(byte[] data, ExtensionRegistryLite extensionRegistry) throws
        InvalidProtocolBufferException;
    ■ Builder mergeFrom(byte[] data, int off, int len, ExtensionRegistryLite
        extensionRegistry) throws InvalidProtocolBufferException;
    ■ Builder mergeFrom(InputStream input) throws IOException;
    ■ Builder mergeFrom(InputStream input, ExtensionRegistryLite extensionRegistry)
        throws IOException;
    ■ Builder mergeFrom(MessageLite other);
    ■ boolean mergeDelimitedFrom(InputStream input) throws IOException;
    ■ boolean mergeDelimitedFrom(InputStream input, ExtensionRegistryLite
        extensionRegistry) throws IOException;

```

Java 版 MessageLite 的文档说明

- com.google.protobuf Interface MessageLite
 - <https://developers.google.com/protocol-buffers/docs/reference/java/com/google/protobuf/MessageLite>
 - <https://developers.google.com/protocol-buffers/docs/reference/java>
 - 搜索并点击左边的: `MessageLite` , 即可查看文档

```

com.google.protobuf
Interface MessageLite
All Superinterfaces
MessageLiteOrBuilder
All Known Subinterfaces
Message
All Known Implementing Classes:
AbstractMessage, AbstractMessageLite, Any, Api, BoolValue, BytesValue, DescriptorProtos.DescriptorProto, DescriptorProtos.DescriptorProto.ExtensionRange, DescriptorProtos.DescriptorProto.ReservedRange, DescriptorProtos.EnumDescriptorProto, DescriptorProtos.EnumDescriptorProto.EnumReservedRange, DescriptorProtos.EnumOptions, DescriptorProtos.EnumValueDescriptorProto, DescriptorProtos.EnumValueOptions, DescriptorProtos.ExtensionRangeOptions, DescriptorProtos.FieldDescriptorProto, DescriptorProtos.FieldOptions, DescriptorProtos.FileDescriptorProto, DescriptorProtos.FileDescriptorSet, DescriptorProtos.FileOptions, DescriptorProtos.GeneratedCodeInfo, DescriptorProtos.GeneratedCodeInfo.Annotation, DescriptorProtos.MessageOptions, DescriptorProtos.MethodDescriptorProto, DescriptorProtos.MethodOptions, DescriptorProtos.OneofDescriptorProto, DescriptorProtos.OneofOptions, DescriptorProtos.ServiceDescriptorProto, DescriptorProtos.ServiceOptions, DescriptorProtos.SourceCodeInfo, DescriptorProtos.SourceCodeInfo.Location, DescriptorProtos.UninterpretedOption, DescriptorProtos.UninterpretedOption.NamePart, DoubleValue, Duration, DynamicMessage, Empty, Enum, EnumValue, Field, FieldMask, FloatValue, com.google.protobuf.GeneratedMessageV3, com.google.protobuf.GeneratedMessageV3.ExtendableMessage, Int32Value, Int64Value, ListValue, Method, Mixin, Option, PluginProtos.CodeGeneratorRequest, PluginProtos.CodeGeneratorResponse, PluginProtos.CodeGeneratorResponse.File, PluginProtos.Version, SourceContext, StringValue, Struct, Timestamp, Type, UInt32Value, UInt64Value, Value

public interface MessageLite

```

```
extends MessageLiteOrBuilder
Abstract interface implemented by Protocol Message objects.
This interface is implemented by all protocol message objects. Non lite messages additionally implement the Message interface, which is a subclass of MessageLite. Use MessageLite instead when you only need the subset of features which it supports -- namely, nothing that uses descriptors or reflection. You can instruct the protocol compiler to generate classes which implement only MessageLite, not the full Message interface, by adding the follow line to the .proto file
```

```
option optimize_for = LITE_RUNTIME;
```

This is particularly useful on resource constrained systems where the full protocol buffers runtime library is too big.

Note that on non constrained systems (e.g. servers) when you need to link in lots of protocol definitions, a better way to reduce total code footprint is to use `optimize_for = CODE_SIZE`. This will make the generated code smaller while still supporting all the same features (at the expense of speed). `optimize_for = LITE_RUNTIME` is best when you only have a small number of message types linked into your binary, in which case the size of the protocol buffers runtime itself is the biggest problem.

Nested Class Summary

Nested Classes

Modifier and Type Interface and Description

```
static interface MessageLite.Builder
```

Abstract interface implemented by Protocol Message builders.

Method Summary

All Methods

Instance Methods

Abstract Methods

Modifier and Type Method and Description

```
Parser<? extends MessageLite> getParserForType()
```

Gets the parser for a message of the same type as this message.

```
int getSerializedSize()
```

Get the number of bytes required to encode this message.

```
MessageLite.Builder newBuilderForType()
```

Constructs a new builder for a message of the same type as this message.

```
MessageLite.Builder toBuilder()
```

Constructs a builder initialized with the current message.

```
byte[] toByteArray()
```

Serializes the message to a byte array and returns it.

```
ByteString toByteString()
```

Serializes the message to a ByteString and returns it.

```
void writeDelimitedTo(java.io.OutputStream output)
```

Like `writeTo(OutputStream)`, but writes the size of the message as a varint before writing the data.

```
void writeTo(CodedOutputStream output)
```

Serializes the message and writes it to output.

```
void writeTo(java.io.OutputStream output)
```

Serializes the message and writes it to output.

Methods inherited from interface com.google.protobuf.MessageLiteOrBuilder

```
getDefaultValueForType(), isInitialized
```

Method Detail

writeTo

```
void writeTo(CodedOutputStream output)
```

```
throws java.io.IOException
```

Serializes the message and writes it to output. This does not flush or close the stream.

Throws

```
java.io.IOException
```

```

getSerializedSize
int getSerializedSize()
Get the number of bytes required to encode this message. The result is only computed on the first call and memoized after that.

getParserForType
Parser < extends MessageLite > getParserForType()
Gets the parser for a message of the same type as this message.

toByteString
ByteString toByteString()
Serializes the message to a ByteString and returns it. This is just a trivial wrapper around writeTo(CodedOutputStream).

toByteArray
byte[] toByteArray()
Serializes the message to a byte array and returns it. This is just a trivial wrapper around writeTo(CodedOutputStream).

writeTo
void writeTo(java.io.OutputStream output)
    throws java.io.IOException
Serializes the message and writes it to output. This is just a trivial wrapper around writeTo(CodedOutputStream). This does not flush or close the stream.

NOTE Protocol Buffers are not self delimiting. Therefore, if you write any more data to the stream after the message, you must somehow ensure that the parser on the receiving end does not interpret this as being part of the protocol message. This can be done e.g. by writing the size of the message before the data, then making sure to limit the input to that size on the receiving end (e.g. by wrapping the InputStream in one which limits the input). Alternatively, just use writeDelimitedTo(OutputStream).

Throws
java.io.IOException
writeDelimitedTo
void writeDelimitedTo(java.io.OutputStream output)
    throws java.io.IOException
Like writeTo(OutputStream), but writes the size of the message as a varint before writing the data. This allows more data to be written to the stream after the message without the need to delimit the message data yourself. Use MessageLite.Builder.mergeDelimitedFrom(InputStream) (or the static method YourMessageType.parseDelimitedFrom(InputStream)) to parse messages written by this method.

Throws
java.io.IOException
newBuilderForType
MessageLite.Builder newBuilderForType()
Constructs a new builder for a message of the same type as this message.

toBuilder
MessageLite.Builder toBuilder()
Constructs a builder initialized with the current message. Use this to derive a new message from the current one.

```

YouTube中Protobuf

TODO:

- 【已解决】研究YouTube逻辑: protobuf中搜索内容对应的可能的字符串编码
- 【未解决】研究YouTube逻辑: protobuf的逆向

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-11-04 14:53:25

普通的iOS类

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-04 15:44:28

YTIPPlayerRequest

TODO:

- 【未解决】研究YouTube逻辑：YTIPPlayerRequest的descriptor
- 【未解决】研究YouTube逻辑：YTIPPlayerRequest的context的protobuf的number不一致
- 【已解决】研究YouTube逻辑：通过GPBFieldDescriptor调试出YTIPPlayerRequest的protobuf的属性字段定义

YTIPPlayerRequest 的 protobuf 的字段定义

推导出 YTIPPlayerRequest 的 protobuf 的字段定义，大概类似于：

```
message YTIPPlayerRequest {
    YTIPPlayerContext context = 1;
    NSString *videoId = 2;
    _Bool contentCheckOk = 3;
    YTIPPlaybackContext playbackContext = 4;
    _Bool racyCheckOk = 5;
    NSString *id_p = 6;
    NSString *t = 7;
    _Bool forOffline = 8;
    NSString *playlistId = 9;
    int playlistIndex = 10;
    unsigned int startTimeSecs = 11;
    NSString *params = 12;
    ??? = 13;
    NSData *offlineSharingWrappedKey = 14;
    GPBInt32Array installedSharingServiceIdsArray = 15;
    YTIPPlayerAttestationRequestData *attestationRequest = 16;
    NSString *referringApp = 17;
    NSString *referrer = 18;
    NSString *serializedThirdPartyEmbedConfig = 19;
    _Bool proxiedByOnesie = 20;
    ??? = 21;
    NSString *hostAppToken = 22;
    NSString *cpn = 23;
    ??? = 24;
    _Bool overrideMutedAtStart = 25;
    YTIPPlayerRequestCaptionParams *captionParams = 26;
    ??? = 27;
    YTIPPlayerRequestVideoQualitySettingParams *videoQualitySettingParams = 28;
}
```

YTIPPlayerRequest 类的头文件定义

header_ModuleFramework/YTIPPlayerRequest.h

```

// Generated by class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48).
// class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2015 by Steve Nygard.
//

#import <Module_Framework/GBMessage.h>

#import <Module_Framework/YTInnerTubeResponseCacheKeyProtocol-Protocol.h>

@class GPBInt32Array, NSData, NSString, YTITubeContext, YTIPPlaybackContext, YTIPPlayerAttestationRequestData, YTIPPlayerRequestCaptionParams, YTIPPlayerRequestVideoQualitySettingParams, YTIPPlayerRequest;

@interface YTIPPlayerRequest : GBMessage <YTITubeResponseCacheKeyProtocol>
{
}

+ (id)descriptor;
- (id)innerTubeResponseCacheKeyWithService:(long long)arg1 coldConfig:(id)arg2;
- (_Bool)autoplay;
- (unsigned long long)autoplaysSinceLastAd;

// Remaining properties
@property(retain, nonatomic) YTIPPlayerAttestationRequestData *attestationRequest; // @dynamic attestationRequest;
@property(retain, nonatomic) YTIPPlayerRequestCaptionParams *captionParams; // @dynamic captionParams;
@property(nonatomic) _Bool contentCheckOk; // @dynamic contentCheckOk;
@property(retain, nonatomic) YTITubeContext *context; // @dynamic context;
@property(copy, nonatomic) NSString *cpn; // @dynamic cpn;
@property(nonatomic, copy) NSString *debugDescription;
@property(nonatomic, copy) NSString *description;
@property(nonatomic) _Bool forOffline; // @dynamic forOffline;
@property(nonatomic) _Bool hasAttestationRequest; // @dynamic hasAttestationRequest;
@property(nonatomic) _Bool hasCaptionParams; // @dynamic hasCaptionParams;
@property(nonatomic) _Bool hasContentCheckOk; // @dynamic hasContentCheckOk;
@property(nonatomic) _Bool hasContext; // @dynamic hasContext;
@property(nonatomic) _Bool hasCpn; // @dynamic hasCpn;
@property(nonatomic) _Bool hasForOffline; // @dynamic hasForOffline;
@property(nonatomic) _Bool hasHostAppToken; // @dynamic hasHostAppToken;
@property(nonatomic) _Bool hasId_p; // @dynamic hasId_p;
@property(nonatomic) _Bool hasOfflineSharingWrappedKey; // @dynamic hasOfflineSharingWrappedKey;

@property(nonatomic) _Bool hasOverrideMutedAtStart; // @dynamic hasOverrideMutedAtStart;
@property(nonatomic) _Bool hasParams; // @dynamic hasParams;
@property(nonatomic) _Bool hasPlaybackContext; // @dynamic hasPlaybackContext;
@property(nonatomic) _Bool hasPlaylistId; // @dynamic hasPlaylistId;
@property(nonatomic) _Bool hasPlaylistIndex; // @dynamic hasPlaylistIndex;
@property(nonatomic) _Bool hasProxiedByOnesie; // @dynamic hasProxiedByOnesie;
@property(nonatomic) _Bool hasRacyCheckOk; // @dynamic hasRacyCheckOk;
@property(nonatomic) _Bool hasReferrer; // @dynamic hasReferrer;
@property(nonatomic) _Bool hasReferringApp; // @dynamic hasReferringApp;
@property(nonatomic) _Bool hasSerializedThirdPartyEmbedConfig; // @dynamic hasSerializedThirdPartyEmbedConfig;
@property(nonatomic) _Bool hasStartTimeSecs; // @dynamic hasStartTimeSecs;

```

```

@property(nonatomic) _Bool hasT; // @dynamic hasT;
@property(nonatomic) _Bool hasVideoId; // @dynamic hasVideoId;
@property(nonatomic) _Bool hasVideoQualitySettingParams; // @dynamic hasVideoQualitySettingParams;
@property(nonatomic) unsigned long long hash;
@property(copy, nonatomic) NSString *hostAppToken; // @dynamic hostAppToken;
@property(copy, nonatomic) NSString *id_p; // @dynamic id_p;
@property(retain, nonatomic) GPBInt32Array installedSharingServiceIdsArray; // @dynamic installedSharingServiceIdsArray;
@property(nonatomic) unsigned long long installedSharingServiceIdsArray_Count; // @dynamic installedSharingServiceIdsArray_Count;
@property(copy, nonatomic) NSData *offlineSharingWrappedKey; // @dynamic offlineSharingWrappedKey;
@property(nonatomic) _Bool overrideMutedAtStart; // @dynamic overrideMutedAtStart;
@property(copy, nonatomic) NSString *params; // @dynamic params;
@property(retain, nonatomic) YTIPPlaybackContext *playbackContext; // @dynamic playbackContext;
@property(copy, nonatomic) NSString *playlistId; // @dynamic playlistId;
@property(nonatomic) int playlistIndex; // @dynamic playlistIndex;
@property(nonatomic) _Bool proxiedByOnesie; // @dynamic proxiedByOnesie;
@property(nonatomic) _Bool racyCheckOk; // @dynamic racyCheckOk;
@property(copy, nonatomic) NSString *referrer; // @dynamic referrer;
@property(copy, nonatomic) NSString *referringApp; // @dynamic referringApp;
@property(copy, nonatomic) NSString *serializedThirdPartyEmbedConfig; // @dynamic serializedThirdPartyEmbedConfig;
@property(nonatomic) unsigned int startTimeSecs; // @dynamic startTimeSecs;
@property(nonatomic) Class superclass;
@property(copy, nonatomic) NSString *t; // @dynamic t;
@property(copy, nonatomic) NSString *videoId; // @dynamic videoId;
@property(retain, nonatomic) YTIPPlayerRequestVideoQualitySettingParams *videoQualitySettingParams; // @dynamic videoQualitySettingParams;

@end

```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-11-10 10:18:23

YTIAdBreakRequest

TODO:

- 【已解决】研究YouTube逻辑：从data解析出YTIAdBreakRequest所有的字段属性的值
- 【未解决】研究YouTube逻辑：获取YTIAdBreakRequest所有的字段的定义即name和number映射关系
- 【已解决】研究YouTube逻辑：protobuf类YTIAdBreakRequest

从data解码反序列化出YTIAdBreakRequest的protobuf的json字符串数据

想要从protobuf的类YTIAdBreakRequest序列化后的NSData数据data，去解析出：

对应的所有属性的值，包括嵌套的子属性的值

思路是：

YTIAdBreakRequest 的基类=父类是 GPBMessage，其中有Class级别的函数：

```
+ (id)parseFromData:(id)arg1;
```

所以可以直接拿来解析。

具体写法：

```
po [objc_getClass("YTIAdBreakRequest") parseFromData: newHttpBodyData]
```

即可得到：

整个protobuf类YTIAdBreakRequest的所有属性的值，包括嵌套的属性。

举例：

- 输入：二进制的数据

```
(lldb) po newHttpBodyData
0ab3160a 9b120a05 7a682d43 4e120243 4e520243 4e620541 70706c65 6a096950 686f6e65 392c3180 0105
8a01 0731372e 30382e32 92010369 4f539a01 0c31332e 332e312e 31374435 30a802f7 02b0029b 05c80202
f00201b8 03f702c0 039b05e8 0303f203 830e0ab8 04434d61 5a714a6f 47454f72 4b726755 51714d2d 6f467
844 55673634 46455065 49726755 5134726d 75425244 53334b30 46454f71 64725155 516d6361 75425244 4
66d7134 46454d61 46725155 516c3943 75425243 44304b34 4645496a 70725155 51754975 75425244 746971
30 46454b58 76725155 51326275 74425243 476f6134 46454f62 4e726755 516c4d2d 75425244 726d6130 46
45497a 79725155 51703532 75425244 586b6130 46454a6a 55725155 51786232 75425243 43746134 46454f5
4 4b726755 51676357 74425244 667a7134 46454d65 78725155 51324c79 75425244 58396130 46454e76 4b7
26755 51373869 75425244 716c3677 4645496d 78726755 51676269 75425244 377a7134 46454a61 61725155
516d7236 75425244 316c7134 46454947 47726755 51305f47 74425243 626f4b34 46454b48 39725155 516f
4c6d 75425244 4e344b30 46454a50 51726755 516e7632 74425244 72354b30 46454961 31726755 51367271
74425243 47793634 46454f54 4e726755 51394d65 74425243 6c734b34 46476a4a 42533342 6c5a4768 35545
```

456 30566b4e 31515739 32633031 54556d52 75616a64 69624638 30526d68 454e6d70 4d576e5a 55593245 3
 5525852 534f4768 47566b52 52515349 79515574 775a5752 6f655530 3164465a 44645546 76646e4e 4e5531
 4a 6b626d6f 33596d78 664e455a 6f52445a 71544670 3256474e 684f5556 30556a68 6f526c5a 45555545 71
 47454e 4254564e 45515442 454e6b30 74634546 6f565551 3464444e 59524545 39505125 33442533 441adc0
 4 434a5077 73706f47 45684d79 4f544134 4d544977 4e44677a 4e546b78 4e546b33 4e545530 474d615a 714
 a6f47 4b4a6e47 7267556f 7a654374 42536947 74613446 4b4f544b 7267556f 6a504b74 42536a30 78363046
 4b4b5777 7267556f 35733275 42536a5a 75363046 4b4a7567 7267556f 6f4c6d75 42536a72 6d613046 4b4a
 5450 7267556f 67394375 42536961 76713446 4b4c694c 7267556f 6c394375 42536949 36613046 4b4e6938
 7267556f 36703274 42536a31 6c713446 4b4e5344 7267556f 36706573 42536942 754b3446 4b4f724b 72675
 56f 39346975 4253696f 7a366758 4b496d78 7267556f 78623275 42536a6b 7a613446 4b4e6552 7251556f 6
 8737575 4253696e 6e613446 4b4a5051 7267556f 36727174 42536943 74613446 4b494846 7251556f 6f6632
 74 42536947 6f613446 4b4e5078 7251556f 37597174 42536965 5f613046 4b4a6a55 7251556f 6c707174 42
 536a47 68613046 4b494747 7267556f 70652d74 42536a53 334b3046 4b4e6631 7251556f 78374774 42536a4
 6 6d713446 4b50764f 7267556f 37386975 42536a62 79713446 4b4e5f4f 7267556f 34726d75 42536a72 354
 b3046 4d6a4a42 5333426c 5a476835 54545630 566b4e31 51573932 63303154 556d5275 616a6469 62463830
 526d6845 4e6d704d 576e5a55 59324535 52585253 4f476847 566b5252 51546f79 51557477 5a57526f 6555
 3031 64465a44 64554676 646e4e4e 55314a6b 626d6f33 596d7866 4e455a6f 52445a71 54467032 56474e68
 4f555630 556a686f 526c5a45 55554643 47454e42 54564e45 51544245 4e6b3074 6345466f 56555134 64444
 e59 52454539 50512533 44253344 2ae60443 4a507773 706f4745 6851784e 4449774f 544d354e 7a41784e 6
 a417a4f 4467334e 5445784d 52695438 4c4b6142 69697969 5034534b 4c7a4b5f 52496f33 347a2d45 696941
 67 7634534b 4f32455f 68496f38 395f3945 696a4c72 6630534b 4c364a5f 68496f31 594c2d45 69695369 76
 34534b 4972675f 52496f33 6f332d45 696a4a68 5034534b 4b54455f 52496f6e 34662d45 69694138 5f30534
 b 4d482d5f 52496f70 64443945 69695535 5077534b 4f4b5f5f 52496f67 594c3945 6969506a 5034534b 4e5
 0655f 52496f6c 50443945 696a4b68 5f34534b 4d574f5f 68496f35 59442d45 696a4a2d 6630534b 4e47665f
 52496f35 344c2d45 696a3533 7630534b 4f69435f 68496f33 4f443945 696a6872 5030534b 4e4c4c5f 5249
 6f6e 765f3845 696a626b 5f30534b 4e794b5f 68496f77 49502d45 69693933 7630534b 5032735f 52496f79
 2d7a3945 696a6267 5034534b 4b66385f 52496f71 72543945 696a4773 7630534b 4b6d715f 52496f6d 63623
 945 69693876 7630534b 4a764d5f 52496f2d 49762d45 696a4f77 5030534b 4965435f 68496f6a 34582d45 6
 9693539 6677534b 4e664d5f 52496f68 617a3945 696a4b32 5030534d 6a4a4253 33426c5a 47683554 545630
 56 6b4e3151 57393263 30315455 6d527561 6a646962 46383052 6d68454e 6d704d57 6e5a5559 32453552 58
 52534f 47684756 6b525251 546f7951 5574775a 57526f65 55303164 465a4464 55467664 6e4e4e55 314a6b6
 2 6d6f3359 6d78664e 455a6f52 445a7154 46703256 474e684f 55563055 6a686f52 6c5a4555 55464348 454
 e4254 564e4655 54424362 33526d4e 6b5a6164 33424755 55526b65 6a684a54 57356a64 30552533 448d0400
 00004098 04e003f0 04018205 0d417369 612f5368 616e6768 6169a205 800320e5 89ffcfd9 ab8cfb92 0120
 83f8 d594f8d8 aed25620 d5c1ccf7 9387d7fb 5520be84 878c8afe f0c4e701 20d9dd90 cccc848f f3800120
 dade94fc aaacb990 5920c6ba 9597b4f9 c9b36120 86f2abaa 94cfaeed 2020c5c5 9488cddf ff879a01 20e2a
 1c0 b0e6df97 f22f209d a1cd4588 d4fcf07e 20fd8ec3 a2f8c5c3 d00320d9 a899e6ad 94dbf91f 2095a5f6 c
 d83f3ff 9a1420dc ed99fc2 cfde93da 0120b498 bdc8c3a4 afbf1620 bddb83b1 f5dec6dd a20120c2 e7f8d8
 90 f0cbca01 209ef581 908fd7bd bef70120 c0c688a7 fed0b3c1 3920a1ce fedfe18e a0e2bc01 20d286fb 98
 a2dfbd 970120f5 9a87c58d 92ced72d 20849ea7 c3be8fe4 db1620ed b4d3d3cb fee8dd1d 20c1cf86 ddb6ed9
 8 f14b20e4 fbcc81c1 8d9d955e 2085dde5 94bcb799 fdb40120 fee8cccf 8a8df480 2c20a182 e5f2e282 81b
 80d20 848ad3ec 8ddcdc9f 252094ac f5959abb b6c28501 20e7ae92 eee6f1ea b6c90120 b7dbd39a f0f385e5
 81012080 cf84abe7 bd9cb666 20a0af89 f7f0d4b3 edc10120 868099e5 d8a7b3ec d801f805 b1908001 8a060
 608 0110bd90 589a0605 0a034348 4e1a002a a101d201 9d010a97 01415041 52386e75 42734548 57666268 4
 d635f37 71627344 5a697352 68544f33 65745849 526a4b42 327a7673 516c396f 5f41764c 58647245 733169
 39 31417663 574a6a4a 646d4254 43613245 5a47765a 78677052 67525547 50535961 6b4a6166 73367067 4e
 38796a 746e3658 7354466b 50567865 6c5f474b 70306b65 33676245 582d6738 437a3057 72776d5f 6a66673
 5 46376a59 33523530 10d8044a e0020add 020a026d 7312d602 6e4b496e 4f44696b 50625735 6e5a4437 2d6
 f7256 7773434e 44566759 536b6148 36454758 776d5f50 6c414b4b 47535659 51564854 6377335f 58655049
 72456278 42545272 6d563355 55485457 462d4d52 4c446d64 35493950 67377631 5561336f 324d6a2d 42376
 242 6e457a4b 5064666e 46344471 4f585a77 41695f30 5a512d76 79556348 526b586d 7444484f 38625166 7
 4465452 56746c79 596f6a75 74554439 634d4b66 47723741 2d416675 7558304c 755f7a78 2d35724e 6a4c63
 72 306a4a6b 31705266 6c4c794e 7a7a4262 5f782d71 5f6b3442 4b367438 346f6f6c 5779624c 43567463 6a
 536774 6a7a4d34 50485139 33307448 7752795f 4a34466a 31525f7a 4b696b4a 59507a6c 544a5051 4e52504
 9 6b43716e 766d7339 4b4e3875 4e704839 7a775130 6d756746 75564e50 4f476168 4f436255 62585077 4d4
 3626d 5f667878 335f5935 54427631 68346e7a 6e613850 6e436c70 6367620a 0a084341 45534167 674312f7

```
02414d75 79323937 6d4f7158 3242787a 4f724b79 676b4973 59644448 6b4c6947 5f417972 53305975 6334
486e 6c73596d 6a373557 42484c62 484e374a 5f496744 48647862 476a4452 4e326c71 46395476 35444456
57514a69 35437a38 5553525f 7a524a53 7345392d 4e50594b 51353671 5f305776 6e4f646f 724e326b 31414
346 774c3545 6654584e 69767268 634e4e30 31687965 33766168 524a7833 774d364b 4a465534 5f4c6364 6
b647863 3363764a 6b537478 33302d63 5a59716d 746a6451 7a7a4d6c 6b4f5771 664b5a79 5a734e36 56526e
2d 612d5a67 64623370 7137726a 62414c55 486e7839 5f56686d 42304362 58344a6e 44616f68 4f644b4d 71
306278 54594f72 46394b73 3274704a 7964454f 3030534f 496a7270 705f4563 53693578 62434e65 64525147
58393667 6f414635 63443777 766b4146 46544475 3166414a 506e484e 46705335 68375f48 72553730 596d
446d 7a717647 724c564c 76347a45 30625f62 766c7043 76516853 43443731 44496c39 32356341 18a0d119
20012a48 0a462097 032889e5 18300038 03400048 00580062 2d766964 656f5f66 6f726d61 743d3232 26736
46b 763d692e 31372e30 38266f75 74707574 3d786d6c 5f766173 7432e801 00e80203 32104b4b 61304879 4
471544d 30533961 3641>
```

- 解析

```
po [objc_getClass("YTIAdBreakRequest") parseFromData: newHttpBodyData]
```

- 输出：json格式的Protobuf数据

```
(lldb) po [objc_getClass("YTIAdBreakRequest") parseFromData: newHttpBodyData]
2022-10-17 11:43:58.552972+0800 YouTube[21038:2204921] hook_ youtubeReqResp.xm YTINnnerTubeCont
ext$descriptor curDesc=<GPBDescriptor 0x2803e7a40>
2022-10-17 11:43:58.553697+0800 YouTube[21038:2204921] hook_ youtubeReqResp.xm YTINnnerTubeCont
ext$descriptor curDesc=<GPBDescriptor 0x2803e7a40>
2022-10-17 11:43:58.713094+0800 YouTube[21038:2204921] hook_ youtubeReqResp.xm YTINnnerTubeCont
ext$descriptor curDesc=<GPBDescriptor 0x2803e7a40>
<YTIAdBreakRequest 0x286120d20> {
    context {
        client {
            hl: "zh-CN"
            gl: "CN"
            carrier_geo: "CN"
            device_make: "Apple"
            device_model: "iPhone9,1"
            client_name: IOS
            client_version: "17.08.2"
            os_name: "iOS"
            os_version: "13.3.1.17D50"
            screen_width_points: 375
            screen_height_points: 667
            screen_pixel_density: 2
            client_form_factor: SMALL_FORM_FACTOR
            window_width_points: 375
            window_height_points: 667
            connection_type: CONN_WIFI
            config_info {
                cold_config_data: "CMaZqJoGEOrKrgUQqm-oFxDUg64FEPeIrgUQ4rmuBRDS3K0FEoQdrQUQmcauBRDFmq
4FEMaFrQUQ19CuBRCD0K4FEIjprQUQuIuuBRDt1q0FEKXvrQUQ2butBRCGao4FEObNrgUQ1M-uBRDrma0FEIz
yruQOp52uBRDXka0FEIjUrQUQxb2uBRCCta4FEOtKrgUQgcWtBRDfq4FEMexrQUQ2LyuBRDX9a0FENvKrgUQ781u
BRDq16wFEImxrguQgbiuBRD7zq4FEJaarQUQmr6uBRD11q4FEIGGrgUQ0_GtBRCboK4FEKH9rQUQoLmuBRDN4K0F
EJPQrgUQnv2tBRDr5K0FEIa1rgUQ6rqtBRCGy64FEOTNrgUQ9MetBRC1sK4FGjJBS3B1ZGh5TTV0Vkn1QW92c01T
UmRuajd1bF80RmhENmpMwnZUY2E5RXRSOGhGVkRRQSIyQutwZWroeU01dFZDdUFvdnNNU1Jkbmo3Ymx
fNEzoRDZqTFp2VGNhOUV0Ujh0R1ZEUEEqGENBTvNEQTBNk0tcEFoVUQ4dDNYREE9PQ%3D%3D"
```

```

    cold_hash_data: "CJPwspoGEhMyOTA4MTIwNDgzNTkxNTk3NTU0GMaZqJ0GKJnGrgUozeCtBSiGta4FKOTK
rgUojPKtBSj0x60FKKwrgUo5s2uBSjZu60FKJugrgUooLmuBSjrma0FKJTPngUog9CuBSiavq4FKLlLrgUo19CuBSiI6a0
FKN18rgUo6p2tBSj1lq4FKNSDrgUo6pesBS1Bu4FKOrKrgUo94iuBSioz6gXKImxrgUoxb2uBSjkza4FKNeRrQuohsuuBS
inna4FKJPQrgUo6rqtBSiCta4FKIHFrQuoof2tBSiGoa4FKNPxrQuo7YqtBSie_a0FKJjUrQuolpqtBSjGha0FKIGGrgUop
e-tBSjs3K0FKNF1rQUox7GtBSjFmq4FKPvOrgUo78iuBSjbqf4FKN_OrgUo4rmuBSjrr5K0FMjjBS3B1ZGh5TTV0Vkn1QW92
c01TUmRuajdibF80RmhENmpMwnZUY2E5RXRSOGhGVkRRQToyQutwZWroeU01dFZDdUFVdnNNU1Jkbmo3YmxnfNEZoRDZqTFp
2VGNhOUV0Ujh0r1ZEUUFCGENBTVNEQTBNk0tcEf0vuQ4dDNYREE9PQ%3D%3D"
    hot_hash_data: "CJPwspoGEhQxNDIwOTM5NzAxNjAzODg3NTExMRiT8LKabiiyiP4SKLzK_RIo34z-EiiAg
v4SK02E_hIo89_9EijLrf0SKL6J_hIo1YL-EiiSiv4SKIRg_RIo3o3-EijJhP4SKKTE_RIon4f-EiiA8_0SKMH_-RiopdD9
EiiU5PwSKOK_RIogYL9EiiPjP4SKNPe_RIo1PD9EijKh_4SKMWO_hIo5YD-EijJ-f0SKNGf_RIo54L-Eij53v0SKOic_hi
o3OD9EijhrP0SKNLL_Rionv_8Eijbk_0SKNyK_hIowIP-Eii93v0SKP2s_RIoy-z9EijbgP4SKKf8_RIoqrT9EijGsv0SKK
mq_Riomcb9Ei18vv0SKJvM_RIo-Iv-Eij0wP0SKIEc_hIoj4X-Eii59fwSKNFm_RIohaz9EijK2P0SMjJBs3B1ZGh5TTV0V
kn1QW92c01TUmRuajdibF80RmhENmpMwnZUY2E5RXRSOGhGVkRRQToyQutwZWroeU01dFZDdUFVdnNNU1Jkbmo3YmxnfNEZo
RDZqTFp2VGNhOUV0Ujh0r1ZEUUFCHENBTNFUTBCb3RmNkZad3BGUURkejhJTW5jd0U%3D"
}
screen_density_float 2
utc_offset_minutes 480
user_interface_theme USER_INTERFACE_THEME_LIGHT
time_zone "Asia/Shanghai"
eml_template_context " \345\211\377\317\331\253\214\373\222\001 \203\370\325\224\370\3
30\256\322V \325\301\314\367\223\207\327\373U \276\204\207\214\212\376\360\304\347\001 \331\335
\220\314\204\217\363\200\001 \332\336\224\374\252\254\271\220Y \306\272\225\227\264\371\311
\263a \206\362\253\252\224\317\256\355 \305\305\224\210\315\337\377\207\232\001 \342\241\300\2
60\346\337\227\362/ \235\241\315\245\210\324\374\360~ \375\216\303\242\370\305\303\320\003 \331
\250\231\346\255\224\333\371\037 \225\245\366\315\203\363\377\232\024 \334\355\231\374\342\317\
336\223\332\001 \264\230\275\310\303\244\257\277\026 \275\333\203\261\365\336\306\335\242\001 \
302\347\370\330\220\360\313\312\001 \236\365\201\220\217\327\275\276\367\001 \300\306\210\247\3
76\320\263\3019 \241\316\376\337\341\216\240\342\274\001 \322\206\373\230\242\337\275\227\001 \
365\232\207\305\215\222\316\327- \204\236\247\303\276\217\344\333\026 \355\264\323\323\313\376\
350\335\035 \301\317\206\335\266\355\230\361K \344\373\314\201\301\215\235\225^ \205\335\345\22
4\274\267\231\375\264\001 \376\350\314\317\212\215\364\200, \241\202\345\362\342\202\201\270\r
\204\212\323\354\215\334\334\237% \224\254\365\225\232\273\266\302\205\001 \347\256\222\356\346
\361\352\266\311\001 \267\333\323\232\360\363\205\345\201\001 \200\317\204\253\347\275\234\266f
\240\257\211\367\360\324\263\355\301\001 \206\200\231\345\330\247\263\354\330\001"
memory_total_kbytes 2099249
notification_permission_info {
    notifications_setting: NOTIFICATIONS_SETTING_ENABLED
    last_device_opt_in_change_time_ago_sec 1443901
}
client_store_info {
    ios_store_country "CHN"
}
user {}
request {
    consistency_token_jars {
        encrypted_token_jar_contents "APAR8nuBsEHWFbhMc_7qbsDZisRhT03etXIRjKB2zvsQl9o_AvLXdr
Es1191AvcWJjJdmBTCa2EZGvZxgpRgRUGPSYakJafs6pgN8yjtn6XsTFkPVxe1_GKp0ke3gbEX-g8Cz0Wrwm_jfg5F7jY3R
50"
        expiration_seconds 600
    }
}
ad_signals_info {
    params {
        key: "ms"
}

```

```

        value: "nKInOD1kPbW5nZD7-orVwsCNDVgYSkaH6EGXwm_P1AKKGSVYQVHTcw3_XePIrEbxBTRrmV3UUHTWF
-MRLDmd5I9Pg7v1Ua3o2Mj-B7bBnEzKPdfnF4DqOXzwAi_0ZQ-vyUcHRkXmtDHO8bQftFTRVtlyYojutUD9cMKfGr7A-Afu
uX0Lu_zx-5rNjLcr0jJk1pRf1LyNzzBb_x-q_k4BK6t84oolWybLCVtcjSgtjzM4PHQ930tHwRy_J4Fj1R_zKikJYPz1TJP
QRPIkCqnvms9Kn8uNpH9zwQ0mugFuVNPOGahOCbUbXPwMCbm_fxx3_Y5TBv1h4nzna8PnClpcg"
    }
}

active_players {
    player_context_params: "CAESAaggC"
}
}

params: "AMuy297m0qX2BxzOrKygkIsYdDHkLiG_AyrS0Yuc4HnlsYmj75WBHLbHN7J_IgDHdxbGjDRN2lqF9TV5DD
VWQJ15Cz8USR_zRJSsE9-NPYKQ56q_0Wvn0dorN2k1ACFwL5EfTXN1vrhcNN01hye3vahRJx3wM6KJFU4_Lcdkdx3cvJks
tx30-cZYqmtjdQzzM1k0WqfKZyZsN6VRn-a-Zgdb3pq7rjbALUHnx9_VhmB0CbX4JnDaoh0dKMq0bxTYOrF9Ks2tpJydE00
0SOIjrpp_EcS15xbCNedRQGX96goAF5cD7wvkAFFTDu1fAJPnHNFpS5h7_HrU70YmDmzqvGrLVLv4zE0b_bvlpCvQhSCD71
DI1925cA"
break_position_ms: 420000
break_index: 1
override_playback_context {
    content_playback_context {
        time_since_last_ad_seconds: 407
        last_milliseconds: 406153
        autoplays_since_last_ad: 0
        conn: 3
        vis: 0
        fling: false
        autoplay: false
        adsense_client_params: "video_format=22&sdkv=1.17.08&output=xml_vast2"
        autonav: false
        autonav_state: STATE_OFF
    }
}
client_playback_nonce: "KKa0HyDqTM0S9a6A"
}

```

YTIAdBreakRequest 类的头文件定义

header_ModuleFramework/YTIAdBreakRequest.h

```

//
//  Generated by class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48).
//
//  class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2015 by Steve Nygard.
//


#import <Module_Framework/GPBMessage.h>

@class NSString, YTIInnerTubeContext, YTIPPlaybackContext;

@interface YTIAdBreakRequest : GPBMessage
{
}

+ (id)descriptor;

```

```

// Remaining properties
@property(nonatomic) int adBlock; // @dynamic adBlock;
@property(nonatomic) int autonavState; // @dynamic autonavState;
@property(nonatomic) int breakIndex; // @dynamic breakIndex;
@property(nonatomic) unsigned long long breakLengthMs; // @dynamic breakLengthMs;
@property(nonatomic) long long breakPositionMs; // @dynamic breakPositionMs;
@property(copy, nonatomic) NSString *clientPlaybackNonce; // @dynamic clientPlaybackNonce;
@property(copy, nonatomic) NSString *clientSideAdTag; // @dynamic clientSideAdTag;
@property(retain, nonatomic) YTINnerTubeContext *context; // @dynamic context;
@property(nonatomic) long long currentMediaTimeMs; // @dynamic currentMediaTimeMs;
@property(nonatomic) long long driftFromHeadMs; // @dynamic driftFromHeadMs;
@property(copy, nonatomic) NSString *encodedAdSafetyReason; // @dynamic encodedAdSafetyReason;
@property(copy, nonatomic) NSString *encodedParentEventId; // @dynamic encodedParentEventId;
@property(nonatomic) _Bool hasAdBlock; // @dynamic hasAdBlock;
@property(nonatomic) _Bool hasAutonavState; // @dynamic hasAutonavState;
@property(nonatomic) _Bool hasBreakIndex; // @dynamic hasBreakIndex;
@property(nonatomic) _Bool hasBreakLengthMs; // @dynamic hasBreakLengthMs;
@property(nonatomic) _Bool hasBreakPositionMs; // @dynamic hasBreakPositionMs;
@property(nonatomic) _Bool hasClientPlaybackNonce; // @dynamic hasClientPlaybackNonce;
@property(nonatomic) _Bool hasClientSideAdTag; // @dynamic hasClientSideAdTag;
@property(nonatomic) _Bool hasContext; // @dynamic hasContext;
@property(nonatomic) _Bool hasCurrentMediaTimeMs; // @dynamic hasCurrentMediaTimeMs;
@property(nonatomic) _Bool hasDriftFromHeadMs; // @dynamic hasDriftFromHeadMs;
@property(nonatomic) _Bool hasEncodedAdSafetyReason; // @dynamic hasEncodedAdSafetyReason;
@property(nonatomic) _Bool hasEncodedParentEventId; // @dynamic hasEncodedParentEventId;
@property(nonatomic) _Bool hasIsProxyAdTagRequest; // @dynamic hasIsProxyAdTagRequest;
@property(nonatomic) _Bool hasLiveTargetingParams; // @dynamic hasLiveTargetingParams;
@property(nonatomic) _Bool hasOverridePlaybackContext; // @dynamic hasOverridePlaybackContext;
@property(nonatomic) _Bool hasParams; // @dynamic hasParams;
@property(nonatomic) _Bool hasPlayerHeight; // @dynamic hasPlayerHeight;
@property(nonatomic) _Bool hasPlayerWidth; // @dynamic hasPlayerWidth;
@property(nonatomic) _Bool hasProxyAdTag; // @dynamic hasProxyAdTag;
@property(nonatomic) _Bool hasProxyAdTagChecksum; // @dynamic hasProxyAdTagChecksum;
@property(nonatomic) _Bool hasTopLevelDomain; // @dynamic hasTopLevelDomain;
@property(nonatomic) _Bool hasVideoId; // @dynamic hasVideoId;
@property(nonatomic) _Bool isProxyAdTagRequest; // @dynamic isProxyAdTagRequest;
@property(copy, nonatomic) NSString *liveTargetingParams; // @dynamic liveTargetingParams;
@property(retain, nonatomic) YTIPPlaybackContext *overridePlaybackContext; // @dynamic overridePlaybackContext;
@property(copy, nonatomic) NSString *params; // @dynamic params;
@property(nonatomic) int playerHeight; // @dynamic playerHeight;
@property(nonatomic) int playerWidth; // @dynamic playerWidth;
@property(copy, nonatomic) NSString *proxyAdTag; // @dynamic proxyAdTag;
@property(copy, nonatomic) NSString *proxyAdTagChecksum; // @dynamic proxyAdTagChecksum;
@property(copy, nonatomic) NSString *topLevelDomain; // @dynamic topLevelDomain;
@property(copy, nonatomic) NSString *videoId; // @dynamic videoId;

@end

```

YTIIInnerTubeContext

TODO:

- 【未解决】研究YouTube逻辑：从OnesieRequestProto子属性类型YTIIInnerTubeContext找到生成data的机制

YTIIInnerTubeContext 类的头文件定义

header_ModuleFramework/YTIIInnerTubeContext.h

```
//
//     Generated by class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48).
//
//     class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2015 by Steve Nygard.
//

#import <Module_Framework/GPBMessag.h>

@class NSMutableArray, NSString, YTIAAdSignalsInfo, YTICapabilityInfo, YTIClickTrackingInfo, YTIClientInfo, YTIEperimentalData, YTIRquestInfo, YTIThirdPartyInfo, YTIUserInfo;

@interface YTIIInnerTubeContext : GPBMessag
{
}

+ (id)descriptor;

// Remaining properties
@property(retain, nonatomic) NSMutableArray *activePlayersArray; // @dynamic activePlayersArray;

@property(nonatomic) unsigned long long activePlayersArray_Count; // @dynamic activePlayersArray_Count;
@property(retain, nonatomic) YTIAAdSignalsInfo *adSignalsInfo; // @dynamic adSignalsInfo;
@property(retain, nonatomic) YTICapabilityInfo *capabilities; // @dynamic capabilities;
@property(retain, nonatomic) YTIClickTrackingInfo *clickTracking; // @dynamic clickTracking;
@property(retain, nonatomic) YTIClientInfo *client; // @dynamic client;
@property(copy, nonatomic) NSString *clientScreenNonce; // @dynamic clientScreenNonce;
@property(retain, nonatomic) YTIEperimentalData *experimentalData; // @dynamic experimentalData;
@property(nonatomic) _Bool hasAdSignalsInfo; // @dynamic hasAdSignalsInfo;
@property(nonatomic) _Bool hasCapabilities; // @dynamic hasCapabilities;
@property(nonatomic) _Bool hasClickTracking; // @dynamic hasClickTracking;
@property(nonatomic) _Bool hasClient; // @dynamic hasClient;
@property(nonatomic) _Bool hasClientScreenNonce; // @dynamic hasClientScreenNonce;
@property(nonatomic) _Bool hasExperimentalData; // @dynamic hasExperimentalData;
@property(nonatomic) _Bool hasRemoteClient; // @dynamic hasRemoteClient;
@property(nonatomic) _Bool hasRequest; // @dynamic hasRequest;
@property(nonatomic) _Bool hasThirdParty; // @dynamic hasThirdParty;
@property(nonatomic) _Bool hasUser; // @dynamic hasUser;
@property(retain, nonatomic) YTIClientInfo *remoteClient; // @dynamic remoteClient;
@property(retain, nonatomic) YTIRquestInfo *request; // @dynamic request;
```

```
@property(nonatomic) YTIThirdPartyInfo *thirdParty; // @dynamic thirdParty;
@property(nonatomic) YT IUserInfo *user; // @dynamic user;

@end
```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:

2022-11-10 10:19:37

YTIClientInfo

- 【未解决】研究YouTube逻辑：搞懂protobuf类YTIClientInfo的字段定义

YTIClientInfo 类的头文件定义

header_ModuleFramework/YTIClientInfo.h

```
//
//     Generated by class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48).
//
//     class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2015 by Steve Nygard.
//

#import <Module_Framework/GPBMessag.h>

@class AddPREFIX_Video_Partner_Rightsmgmt_ContentAvailability_UserViewData, GPBInt32Array, NSData, NSMutableArray, NSString, YTIClientStoreInfo, YTICConfigGroupsClientInfo, YTICountryLocationInfo, YTIHomeGroupInfo, YTIKidsAppInfo, YTILocationInfo, YTIMainAppWebInfo, YTIMobileDataPlanInfo, YTIMusicAppInfo, YTINotificationPermissionInfo, YTISpacecastClientInfo, YTITvAppInfo, YTIUnpluggedAppInfo, YTIUnpluggedLocationInfo;

@interface YTIClientInfo : GPBMessag
{
}

+ (id)descriptor;

// Remaining properties
@property(copy, nonatomic) NSString *acceptLanguage; // @dynamic acceptLanguage;
@property(copy, nonatomic) NSString *acceptRegion; // @dynamic acceptRegion;
@property(nonatomic) int androidSdkVersion; // @dynamic androidSdkVersion;
@property(nonatomic) _Bool animatedWebpSupport; // @dynamic animatedWebpSupport;
@property(nonatomic) int applicationState; // @dynamic applicationState;
@property(copy, nonatomic) NSString *browserName; // @dynamic browserName;
@property(copy, nonatomic) NSString *browserVersion; // @dynamic browserVersion;
@property(nonatomic) int cameraType; // @dynamic cameraType;
@property(copy, nonatomic) NSString *carrierGeo; // @dynamic carrierGeo;
@property(copy, nonatomic) NSString *chipset; // @dynamic chipset;
@property(nonatomic) int clientFormFactor; // @dynamic clientFormFactor;
@property(nonatomic) int clientName; // @dynamic clientName;
@property(nonatomic) int clientScreen; // @dynamic clientScreen;
@property(retain, nonatomic) YTIClientStoreInfo *clientStoreInfo; // @dynamic clientStoreInfo;
@property(nonatomic) _Bool clientValid; // @dynamic clientValid;
@property(copy, nonatomic) NSString *clientVersion; // @dynamic clientVersion;
@property(copy, nonatomic) NSData *coldAppBundleConfigData; // @dynamic coldAppBundleConfigData;

@property(copy, nonatomic) NSString *configData; // @dynamic configData;
@property(retain, nonatomic) YTICConfigGroupsClientInfo *configInfo; // @dynamic configInfo;
@property(nonatomic) int connectionType; // @dynamic connectionType;
@property(copy, nonatomic) NSString *contentSizeCategory; // @dynamic contentSizeCategory;
```

```

@property(retain, nonatomic) YTICountryLocationInfo *countryLocationInfo; // @dynamic countryLocationInfo;
@property(nonatomic) _Bool crackedH1; // @dynamic crackedH1;
@property(copy, nonatomic) NSString debugDeviceIdOverride; // @dynamic debugDeviceIdOverride;
@property(copy, nonatomic) NSString deviceBrand; // @dynamic deviceBrand;
@property(copy, nonatomic) NSString deviceId; // @dynamic deviceId;
@property(copy, nonatomic) NSString deviceMake; // @dynamic deviceMake;
@property(copy, nonatomic) NSString deviceModel; // @dynamic deviceModel;
@property(nonatomic) int effectiveConnectionType; // @dynamic effectiveConnectionType;
@property(copy, nonatomic) NSData emlTemplateContext; // @dynamic emlTemplateContext;
@property(retain, nonatomic) GPBInt32Array experimentIdsArray; // @dynamic experimentIdsArray;
@property(readonly, nonatomic) unsigned long long experimentIdsArray_Count; // @dynamic experimentIdsArray_Count;
@property(copy, nonatomic) NSString experimentsToken; // @dynamic experimentsToken;
@property(copy, nonatomic) NSString firmwareVersion; // @dynamic firmwareVersion;
@property(retain, nonatomic) GPBInt32Array firstTimeSignInExperimentIdsArray; // @dynamic firstTimeSignInExperimentIdsArray;
@property(readonly, nonatomic) unsigned long long firstTimeSignInExperimentIdsArray_Count; // @dynamic firstTimeSignInExperimentIdsArray_Count;
@property(nonatomic) float fontScale; // @dynamic fontScale;
@property(copy, nonatomic) NSString forwardedFor; // @dynamic forwardedFor;
@property(copy, nonatomic) NSString geo; // @dynamic geo;
@property(copy, nonatomic) NSString gfeFrontlineInfo; // @dynamic gfeFrontlineInfo;
@property(copy, nonatomic) NSString gl; // @dynamic gl;
@property(nonatomic) int gmscoreVersionCode; // @dynamic gmscoreVersionCode;
@property(nonatomic) _Bool hasAcceptLanguage; // @dynamic hasAcceptLanguage;
@property(nonatomic) _Bool hasAcceptRegion; // @dynamic hasAcceptRegion;
@property(nonatomic) _Bool hasAndroidSdkVersion; // @dynamic hasAndroidSdkVersion;
@property(nonatomic) _Bool hasAnimatedWebpSupport; // @dynamic hasAnimatedWebpSupport;
@property(nonatomic) _Bool hasApplicationState; // @dynamic hasApplicationState;
@property(nonatomic) _Bool hasBrowserName; // @dynamic hasBrowserName;
@property(nonatomic) _Bool hasBrowserVersion; // @dynamic hasBrowserVersion;
@property(nonatomic) _Bool hasCameraType; // @dynamic hasCameraType;
@property(nonatomic) _Bool hasCarrierGeo; // @dynamic hasCarrierGeo;
@property(nonatomic) _Bool hasChipset; // @dynamic hasChipset;
@property(nonatomic) _Bool hasClientFormFactor; // @dynamic hasClientFormFactor;
@property(nonatomic) _Bool hasClientName; // @dynamic hasClientName;
@property(nonatomic) _Bool hasClientScreen; // @dynamic hasClientScreen;
@property(nonatomic) _Bool hasClientStoreInfo; // @dynamic hasClientStoreInfo;
@property(nonatomic) _Bool hasClientValid; // @dynamic hasClientValid;
@property(nonatomic) _Bool hasClientVersion; // @dynamic hasClientVersion;
@property(nonatomic) _Bool hasColdAppBundleConfigData; // @dynamic hasColdAppBundleConfigData;
@property(nonatomic) _Bool hasConfigData; // @dynamic hasConfigData;
@property(nonatomic) _Bool hasConfigInfo; // @dynamic hasConfigInfo;
@property(nonatomic) _Bool hasConnectionType; // @dynamic hasConnectionType;
@property(nonatomic) _Bool hasCountryLocationInfo; // @dynamic hasCountryLocationInfo;
@property(nonatomic) _Bool hasCrackedH1; // @dynamic hasCrackedH1;
@property(nonatomic) _Bool hasDebugDeviceIdOverride; // @dynamic hasDebugDeviceIdOverride;
@property(nonatomic) _Bool hasDeviceBrand; // @dynamic hasDeviceBrand;
@property(nonatomic) _Bool hasDeviceId; // @dynamic hasDeviceId;
@property(nonatomic) _Bool hasDeviceMake; // @dynamic hasDeviceMake;
@property(nonatomic) _Bool hasDeviceModel; // @dynamic hasDeviceModel;
@property(nonatomic) _Bool hasEffectiveConnectionType; // @dynamic hasEffectiveConnectionType;
@property(nonatomic) _Bool hasEmlTemplateContext; // @dynamic hasEmlTemplateContext;
@property(nonatomic) _Bool hasExperimentsToken; // @dynamic hasExperimentsToken;
@property(nonatomic) _Bool hasFirmwareVersion; // @dynamic hasFirmwareVersion;

```

```

@property(nonatomic) _Bool hasForwardedFor; // @dynamic hasForwardedFor;
@property(nonatomic) _Bool hasGeo; // @dynamic hasGeo;
@property(nonatomic) _Bool hasGfeFrontlineInfo; // @dynamic hasGfeFrontlineInfo;
@property(nonatomic) _Bool hasGl; // @dynamic hasGl;
@property(nonatomic) _Bool hasGmscoreVersionCode; // @dynamic hasGmscoreVersionCode;
@property(nonatomic) _Bool hasH1; // @dynamic hasH1;
@property(nonatomic) _Bool hasHomeGroupInfo; // @dynamic hasHomeGroupInfo;
@property(nonatomic) _Bool hasInternalGeo; // @dynamic hasInternalGeo;
@property(nonatomic) _Bool hasInternalGeoIp; // @dynamic hasInternalGeoIp;
@property(nonatomic) _Bool hasIsInternal; // @dynamic hasIsInternal;
@property(nonatomic) _Bool hasKidsAppInfo; // @dynamic hasKidsAppInfo;
@property(nonatomic) _Bool hasLocationInfo; // @dynamic hasLocationInfo;
@property(nonatomic) _Bool hasLocationPlayabilityToken; // @dynamic hasLocationPlayabilityToken;

@property(nonatomic) _Bool hasMainAppWebInfo; // @dynamic hasMainAppWebInfo;
@property(nonatomic) _Bool hasMemoryTotalKbytes; // @dynamic hasMemoryTotalKbytes;
@property(nonatomic) _Bool hasMobileDataPlanInfo; // @dynamic hasMobileDataPlanInfo;
@property(nonatomic) _Bool hasMusicAppInfo; // @dynamic hasMusicAppInfo;
@property(nonatomic) _Bool hasMwebUi; // @dynamic hasMwebUi;
@property(nonatomic) _Bool hasNewVisitorCookie; // @dynamic hasNewVisitorCookie;
@property(nonatomic) _Bool hasNotificationPermissionInfo; // @dynamic hasNotificationPermission
Info;
@property(nonatomic) _Bool hasOriginalURL; // @dynamic hasOriginalURL;
@property(nonatomic) _Bool hasOsName; // @dynamic hasOsName;
@property(nonatomic) _Bool hasOsVersion; // @dynamic hasOsVersion;
@property(nonatomic) _Bool hasPlatform; // @dynamic hasPlatform;
@property(nonatomic) _Bool hasPlatformDetail; // @dynamic hasPlatformDetail;
@property(nonatomic) _Bool hasPlayerType; // @dynamic hasPlayerType;
@property(nonatomic) _Bool hasProjectId; // @dynamic hasProjectId;
@property(nonatomic) _Bool hasRawDeviceId; // @dynamic hasRawDeviceId;
@property(nonatomic) _Bool hasReleaseYear; // @dynamic hasReleaseYear;
@property(nonatomic) _Bool hasRemoteHost; // @dynamic hasRemoteHost;
@property(nonatomic) _Bool hasRmPlatform; // @dynamic hasRmPlatform;
@property(nonatomic) _Bool hasScreenDensityFloat; // @dynamic hasScreenDensityFloat;
@property(nonatomic) _Bool hasScreenHeightInches; // @dynamic hasScreenHeightInches;
@property(nonatomic) _Bool hasScreenHeightPoints; // @dynamic hasScreenHeightPoints;
@property(nonatomic) _Bool hasScreenPixelDensity; // @dynamic hasScreenPixelDensity;
@property(nonatomic) _Bool hasScreenWidthInches; // @dynamic hasScreenWidthInches;
@property(nonatomic) _Bool hasScreenWidthPoints; // @dynamic hasScreenWidthPoints;
@property(nonatomic) _Bool hasSpacecastClientInfo; // @dynamic hasSpacecastClientInfo;
@property(nonatomic) _Bool hasSpacecastToken; // @dynamic hasSpacecastToken;
@property(nonatomic) _Bool hasTheme; // @dynamic hasTheme;
@property(nonatomic) _Bool hasTimeZone; // @dynamic hasTimeZone;
@property(nonatomic) _Bool hasTvAppInfo; // @dynamic hasTvAppInfo;
@property(nonatomic) _Bool hasUnpluggedAppInfo; // @dynamic hasUnpluggedAppInfo;
@property(nonatomic) _Bool hasUnpluggedLocationInfo; // @dynamic hasUnpluggedLocationInfo;
@property(nonatomic) _Bool hasUserAgent; // @dynamic hasUserAgent;
@property(nonatomic) _Bool hasUserInterfaceTheme; // @dynamic hasUserInterfaceTheme;
@property(nonatomic) _Bool hasUserData; // @dynamic hasUserData;
@property(nonatomic) _Bool hasUtcOffsetMinutes; // @dynamic hasUtcOffsetMinutes;
@property(nonatomic) _Bool hasVisitorData; // @dynamic hasVisitorData;
@property(nonatomic) _Bool hasWebpSupport; // @dynamic hasWebpSupport;
@property(nonatomic) _Bool hasWindowHeightPoints; // @dynamic hasWindowHeightPoints;
@property(nonatomic) _Bool hasWindowWidthPoints; // @dynamic hasWindowWidthPoints;
@property(nonatomic) _Bool hasYtRestrictHeader; // @dynamic hasYtRestrictHeader;
@property(nonatomic) _Bool hasYtSafetyModeHeader; // @dynamic hasYtSafetyModeHeader;

```

```

@property(nonatomic) NSMutableArray *heterodyneIdsArray; // @dynamic heterodyneIdsArray;

@property(nonatomic) unsigned long long heterodyneIdsArray_Count; // @dynamic heterodyneIdsArray_Count;
@property(nonatomic) NSString *h1; // @dynamic h1;
@property(nonatomic) YTIBaseInfo *homeGroupInfo; // @dynamic homeGroupInfo;
@property(nonatomic) GPBInt32Array *internalClientExperimentIdsArray; // @dynamic internalClientExperimentIdsArray;
@property(nonatomic) unsigned long long internalClientExperimentIdsArray_Count; // @dynamic internalClientExperimentIdsArray_Count;
@property(nonatomic) GPBInt32Array *internalExperimentIdsArray; // @dynamic internalExperimentIdsArray;
@property(nonatomic) unsigned long long internalExperimentIdsArray_Count; // @dynamic internalExperimentIdsArray_Count;
@property(nonatomic) NSString *internalGeo; // @dynamic internalGeo;
@property(nonatomic) NSString *internalGeoIp; // @dynamic internalGeoIp;
@property(nonatomic) _Bool isInternal; // @dynamic isInternal;
@property(nonatomic) YTIBaseInfo *kidsAppInfo; // @dynamic kidsAppInfo;
@property(nonatomic) YTILocationInfo *locationInfo; // @dynamic locationInfo;
@property(nonatomic) NSString *locationPlayabilityToken; // @dynamic locationPlayabilityToken;
@property(nonatomic) YTIMainAppWebInfo *mainAppWebInfo; // @dynamic mainAppWebInfo;
@property(nonatomic) long long memoryTotalKbytes; // @dynamic memoryTotalKbytes;
@property(nonatomic) YTIMobileDataPlanInfo *mobileDataPlanInfo; // @dynamic mobileDataPlanInfo;
@property(nonatomic) YTIMusicAppInfo *musicAppInfo; // @dynamic musicAppInfo;
@property(nonatomic) int mwebUi; // @dynamic mwebUi;
@property(nonatomic) _Bool newVisitorCookie; // @dynamic newVisitorCookie;
@property(nonatomic) YTINotificationPermissionInfo *notificationPermissionInfo; // @dynamic notificationPermissionInfo;
@property(nonatomic) NSString *originalURL; // @dynamic originalURL;
@property(nonatomic) NSString *osName; // @dynamic osName;
@property(nonatomic) NSString *osVersion; // @dynamic osVersion;
@property(nonatomic) int platform; // @dynamic platform;
@property(nonatomic) int platformDetail; // @dynamic platformDetail;
@property(nonatomic) NSMutableArray *playerDatapushBuildIdsArray; // @dynamic playerDatapushBuildIdsArray;
@property(nonatomic) unsigned long long playerDatapushBuildIdsArray_Count; // @dynamic playerDatapushBuildIdsArray_Count;
@property(nonatomic) int playerType; // @dynamic playerType;
@property(nonatomic) NSString *projectId; // @dynamic projectId;
@property(nonatomic) NSString *rawDeviceId; // @dynamic rawDeviceId;
@property(nonatomic) int releaseYear; // @dynamic releaseYear;
@property(nonatomic) NSString *remoteHost; // @dynamic remoteHost;
@property(nonatomic) int rmPlatform; // @dynamic rmPlatform;
@property(nonatomic) float screenDensityFloat; // @dynamic screenDensityFloat;
@property(nonatomic) float screenHeightInches; // @dynamic screenHeightInches;
@property(nonatomic) int screenHeightPoints; // @dynamic screenHeightPoints;
@property(nonatomic) int screenPixelDensity; // @dynamic screenPixelDensity;
@property(nonatomic) float screenWidthInches; // @dynamic screenWidthInches;
@property(nonatomic) int screenWidthPoints; // @dynamic screenWidthPoints;
@property(nonatomic) YTISpacecastClientInfo *spacecastClientInfo; // @dynamic spacecastClientInfo;
@property(nonatomic) NSString *spacecastToken; // @dynamic spacecastToken;
@property(nonatomic) NSMutableArray *srsDatapushBuildIdsArray; // @dynamic srsDatapushBuildIdsArray;

```

```
@property(nonatomic) unsigned long long srsDatapushBuildIdsArray_Count; // @dynamic s  
rsDatapushBuildIdsArray_Count;  
@property(nonatomic) int textSizeAdjustmentOneOfCase; // @dynamic textSizeAdjustmen  
tOneOfCase;  
@property(nonatomic) int theme; // @dynamic theme;  
@property(copy, nonatomic) NSString *timeZone; // @dynamic timeZone;  
@property(retain, nonatomic) YTITvAppInfo *tvAppInfo; // @dynamic tvAppInfo;  
@property(retain, nonatomic) YTIUnpluggedAppInfo *unpluggedAppInfo; // @dynamic unpluggedAppInf  
o;  
@property(retain, nonatomic) YTIUnpluggedLocationInfo *unpluggedLocationInfo; // @dynamic unpla  
ggedLocationInfo;  
@property(copy, nonatomic) NSString *userAgent; // @dynamic userAgent;  
@property(nonatomic) int userInterfaceTheme; // @dynamic userInterfaceTheme;  
@property(retain, nonatomic) AddPREFIX_Video_Partner_Rightsmgmt_ContentAvailability_UserViewDat  
a *userViewData; // @dynamic userViewData;  
@property(nonatomic) int utcOffsetMinutes; // @dynamic utcOffsetMinutes;  
@property(copy, nonatomic) NSString *visitorData; // @dynamic visitorData;  
@property(nonatomic) _Bool webpSupport; // @dynamic webpSupport;  
@property(nonatomic) int windowHeightPoints; // @dynamic windowHeightPoints;  
@property(nonatomic) int windowWidthPoints; // @dynamic windowWidthPoints;  
@property(copy, nonatomic) NSString *ytRestrictHeader; // @dynamic ytRestrictHeader;  
@property(copy, nonatomic) NSString *ytSafetyModeHeader; // @dynamic ytSafetyModeHeader;  
  
@end
```



crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-10 10:20:31

YTIOnesieHotConfig

YTIOnesieHotConfig 类的头文件定义

header_ModuleFramework/YTIOnesieHotConfig.h

```

// Generated by class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48).
// class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2015 by Steve Nygard.

#import <Module_Framework/GPBMessag.h>

@class GPBInt32Array, NSData, NSString, YTIIosOnesieHotConfig, YTIOnesieConnectionPrewarmConfig;

@interface YTIOnesieHotConfig : GPBMessag
{
}

+ (id)descriptor;

// Remaining properties
@property(retain, nonatomic) GPBInt32Array audioItagWhitelistArray; // @dynamic audioItagWhite
listArray;
@property(nonatomic) unsigned long long audioItagWhitelistArray_Count; // @dynamic au
dioItagWhitelistArray_Count;
@property(copy, nonatomic) NSString * baseURL; // @dynamic baseURL;
@property(copy, nonatomic) NSData * clientKey; // @dynamic clientKey;
@property(nonatomic) _Bool disableFallbackToInnertube; // @dynamic disableFallbackToInnertube;
@property(nonatomic) _Bool disableHostReplacement; // @dynamic disableHostReplacement;
@property(nonatomic) _Bool enableUserFeatures; // @dynamic enableUserFeatures;
@property(nonatomic) _Bool enableZeroRtt; // @dynamic enableZeroRtt;
@property(copy, nonatomic) NSData * encryptedClientKey; // @dynamic encryptedClientKey;
@property(copy, nonatomic) NSString * fallbackHostname; // @dynamic fallbackHostname;
@property(copy, nonatomic) NSString * fallbackURLParams; // @dynamic fallbackURLParams;
@property(nonatomic) _Bool hasBaseUrl; // @dynamic hasBaseUrl;
@property(nonatomic) _Bool hasClientKey; // @dynamic hasClientKey;
@property(nonatomic) _Bool hasDisableFallbackToInnertube; // @dynamic hasDisableFallbackToInner
tube;
@property(nonatomic) _Bool hasDisableHostReplacement; // @dynamic hasDisableHostReplacement;
@property(nonatomic) _Bool hasEnableUserFeatures; // @dynamic hasEnableUserFeatures;
@property(nonatomic) _Bool hasEnableZeroRtt; // @dynamic hasEnableZeroRtt;
@property(nonatomic) _Bool hasEncryptedClientKey; // @dynamic hasEncryptedClientKey;
@property(nonatomic) _Bool hasFallbackHostname; // @dynamic hasFallbackHostname;
@property(nonatomic) _Bool hasFallbackURLParams; // @dynamic hasFallbackURLParams;
@property(nonatomic) _Bool hasIosConfig; // @dynamic hasIosConfig;
@property(nonatomic) _Bool hasKeyExpiresInSeconds; // @dynamic hasKeyExpiresInSeconds;
@property(nonatomic) _Bool hasLiveOnesieTtlMs; // @dynamic hasLiveOnesieTtlMs;
@property(nonatomic) _Bool hasMaxRetryTimeoutMs; // @dynamic hasMaxRetryTimeoutMs;
@property(nonatomic) _Bool hasOnesieRequestProtoInPostBody; // @dynamic hasOnesieRequestProtoIn

```

```

PostBody;
@property(nonatomic) _Bool hasOnesieUstreamerConfig; // @dynamic hasOnesieUstreamerConfig;
@property(nonatomic) _Bool hasPlayerServiceResponseTimeoutMs; // @dynamic hasPlayerServiceResponseTimeoutMs;
@property(nonatomic) _Bool hasPrewarmConfig; // @dynamic hasPrewarmConfig;
@property(nonatomic) _Bool hasRespectPersistedAbrCap; // @dynamic hasRespectPersistedAbrCap;
@property(nonatomic) _Bool hasRetryEnabled; // @dynamic hasRetryEnabled;
@property(nonatomic) _Bool hasReverseProxyConfig; // @dynamic hasReverseProxyConfig;
@property(nonatomic) _Bool hasSendClientInfoToUstreamer; // @dynamic hasSendClientInfoToUstreamer;
@property(nonatomic) _Bool hasSendMediaCapabilities; // @dynamic hasSendMediaCapabilities;
@property(nonatomic) _Bool hasServerFormatSelection; // @dynamic hasServerFormatSelection;
@property(nonatomic) _Bool hasUseClientAbrStateProto; // @dynamic hasUseClientAbrStateProto;
@property(nonatomic) _Bool hasUseHotConfigToCreateOnesieRequest; // @dynamic hasUseHotConfigToCreateOnesieRequest;
@property(nonatomic) _Bool hasUseHotConfigWithMissingPlaybackOnesieConfig; // @dynamic hasUseHotConfigWithMissingPlaybackOnesieConfig;
@property(nonatomic) _Bool hasUseLastMediaPartFlag; // @dynamic hasUseLastMediaPartFlag;
@property(nonatomic) _Bool hasUseUmp; // @dynamic hasUseUmp;
@property(nonatomic) _Bool hasUseUmpUniqueContentValue; // @dynamic hasUseUmpUniqueContentValue;

@property(retain, nonatomic) YTIIosOnesieHotConfig *iosConfig; // @dynamic iosConfig;
@property(nonatomic) long long keyExpiresInSeconds; // @dynamic keyExpiresInSeconds;
@property(nonatomic) int liveOnesieTtlMs; // @dynamic liveOnesieTtlMs;
@property(nonatomic) long long maxRetryTimeoutMs; // @dynamic maxRetryTimeoutMs;
@property(nonatomic) _Bool onesieRequestProtoInPostBody; // @dynamic onesieRequestProtoInPostBody;
@property(copy, nonatomic) NSData *onesieUstreamerConfig; // @dynamic onesieUstreamerConfig;
@property(nonatomic) int playerServiceResponseTimeoutMs; // @dynamic playerServiceResponseTimeoutMs;
@property(retain, nonatomic) YTIOOnesieConnectionPrewarmConfig *prewarmConfig; // @dynamic prewarmConfig;
@property(nonatomic) _Bool respectPersistedAbrCap; // @dynamic respectPersistedAbrCap;
@property(nonatomic) _Bool retryEnabled; // @dynamic retryEnabled;
@property(copy, nonatomic) NSString *reverseProxyConfig; // @dynamic reverseProxyConfig;
@property(nonatomic) _Bool sendClientInfoToUstreamer; // @dynamic sendClientInfoToUstreamer;
@property(nonatomic) _Bool sendMediaCapabilities; // @dynamic sendMediaCapabilities;
@property(nonatomic) _Bool serverFormatSelection; // @dynamic serverFormatSelection;
@property(nonatomic) _Bool useClientAbrStateProto; // @dynamic useClientAbrStateProto;
@property(nonatomic) _Bool useHotConfigToCreateOnesieRequest; // @dynamic useHotConfigToCreateOnesieRequest;
@property(nonatomic) _Bool useHotConfigWithMissingPlaybackOnesieConfig; // @dynamic useHotConfigWithMissingPlaybackOnesieConfig;
@property(nonatomic) _Bool useLastMediaPartFlag; // @dynamic useLastMediaPartFlag;
@property(nonatomic) _Bool useUmp; // @dynamic useUmp;
@property(nonatomic) _Bool useUmpUniqueContentValue; // @dynamic useUmpUniqueContentValue;
@property(retain, nonatomic) GPBInt32Array *videoItagWhitelistArray; // @dynamic videoItagWhitelistArray;
@property(readonly, nonatomic) unsigned long long videoItagWhitelistArray_Count; // @dynamic videoItagWhitelistArray_Count;

@end

```

2022-11-10 10:22:52

lite的C++类

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-04 15:44:48

OnesieRequestProto

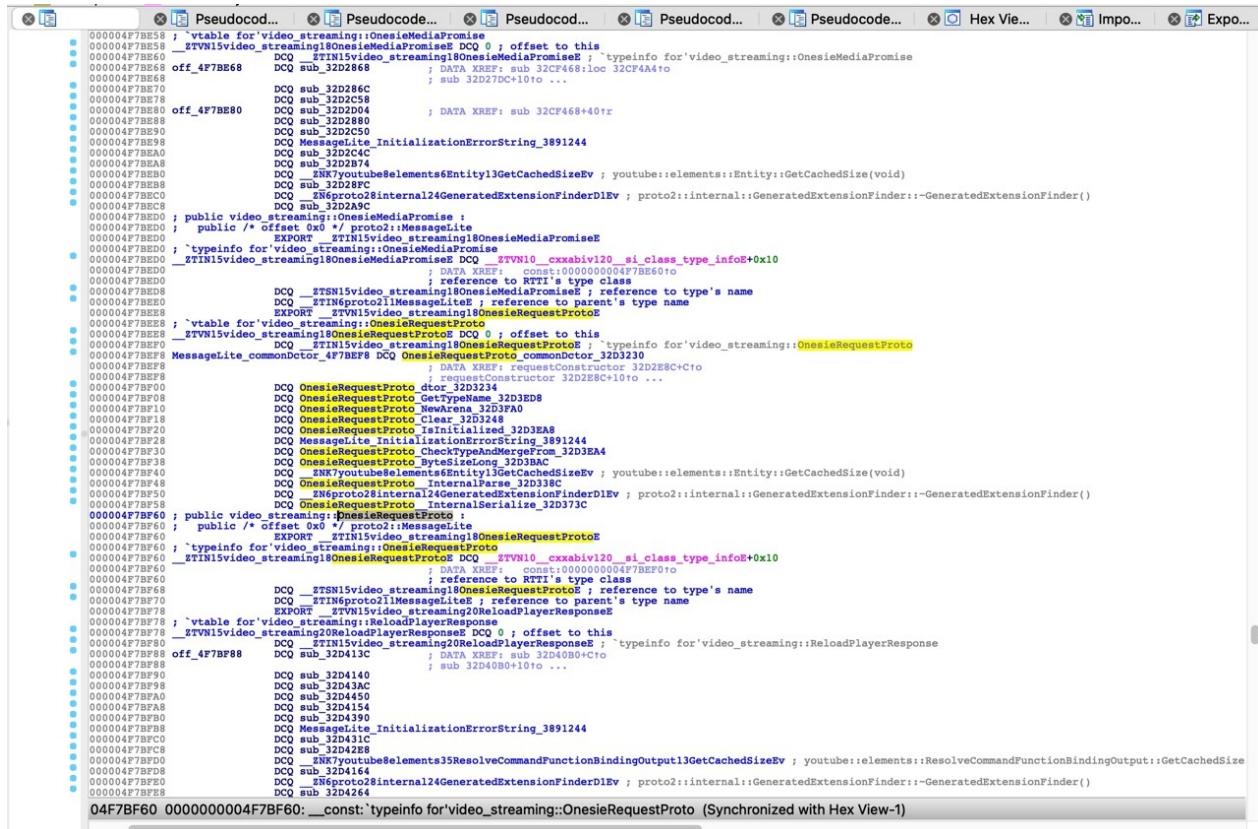
TODO:

- 【未解决】研究YouTube逻辑: lite版protobuf类OnesieRequestProto如何搞清楚属性字段定义
- 【未解决】研究YouTube逻辑: OnesieRequestProto相关

OnesieRequestProto的vtable定义

经过逆向，目前的理解是：

IDA中改名：



核心定义：

```

000004F7BEE8 ; `vtable for'video_streaming::OnesieRequestProto
__const 000000004F7BEE8 __ZTVN15video_streaming18OnesieRequestProtoE DCQ 0 ; offset to this
__const 000000004F7BEF0 DCQ __ZTIN15video_streaming18OnesieRequestProtoE ; `typeinfo for'video_streaming::OnesieRequestProto
__const 000000004F7BEF8 MessageLite_commonDctor_4F7BEF8 DCQ OnesieRequestProto_commonDctor_32D
3230
__const 000000004F7BEF8 ; DATA XREF: requestConstructo
r_32D2E8C Cto
__const 000000004F7BEF8 ; requestConstructo_32D2E8C 10
__const 000000004F7BF0 DCQ OnesieRequestProto_dtor_32D3234

```

```

__const 000000004F7BF08          DCQ OnesieRequestProto_GetTypeName_32D3ED8
__const 000000004F7BF10         DCQ OnesieRequestProto_NewArena_32D3FA0
__const 000000004F7BF18         DCQ OnesieRequestProto_Clear_32D3248
__const 000000004F7BF20         DCQ OnesieRequestProto_IsInitialized_32D3EA8
__const 000000004F7BF28         DCQ MessageLite_InitializationErrorString_3891244
__const 000000004F7BF30         DCQ OnesieRequestProto_CheckTypeAndMergeFrom_32D3EA4
__const 000000004F7BF38         DCQ OnesieRequestProto_ByteSizeLong_32D3BAC
__const 000000004F7BF40         DCQ __ZNK7youtube8elements6Entity13GetCachedSizeEv ; y
outube::elements::Entity::GetCachedSize(void)
__const 000000004F7BF48         DCQ OnesieRequestProto__InternalParse_32D338C
__const 000000004F7BF50         DCQ __ZN6proto28internal124GeneratedExtensionFinderD1Ev
; proto2::internal::GeneratedExtensionFinder::GeneratedExtensionFinder()
__const 000000004F7BF58         DCQ OnesieRequestProto__InternalSerialize_32D373C
__const 000000004F7BF60 ; public video_streaming::OnesieRequestProto :

```

OnesieRequestProto的vtable定义：

- vtable OnesieRequestProto
 - +0x08 = OnesieRequestProto_typeinfo
 - +0x10 = MessageLite_commonDctor_4F7BEF8 = OnesieRequestProto_commonDctor_32D3230
 - 某种 deconstructor ?
 - +0x18 = OnesieRequestProto_dtor_32D3234
 - virtual ~MessageLite() = default;
 - +0x20 = OnesieRequestProto_GetTypeName_32D3ED8
 - virtual std::string GetTypeName() const = 0;
 - +0x28 = OnesieRequestProto_NewArena_32D3FA0
 - virtual MessageLite New(Arena arena) const = 0;
 - +0x30 = OnesieRequestProto_Clear_32D3248
 - virtual void Clear() = 0;
 - +0x38 = OnesieRequestProto_IsInitialized_32D3EA8
 - virtual bool IsInitialized() const = 0;
 - +0x40 = MessageLite_InitializationErrorString_3891244
 - virtual std::string InitializationErrorString() const;
 - +0x48 = OnesieRequestProto_CheckTypeAndMergeFrom_32D3EA4
 - virtual void CheckTypeAndMergeFrom(const MessageLite& other) = 0;
 - +0x50 = OnesieRequestProto_ByteSizeLong_32D3BAC
 - virtual size_t ByteSizeLong() const = 0;
 - +0x58 = youtube::elements::Entity::GetCachedSize(void)
 - virtual int GetCachedSize() const = 0;
 - +0x60 = OnesieRequestProto__InternalParse_32D338C
 - virtual const char _InternalParse(const char /ptr/, internal::ParseContext /ctx*)
 - +0x68 = proto2::internal::GeneratedExtensionFinder::~GeneratedExtensionFinder()
 - virtual void OnDemandRegisterArenaDtor(Arena /arena*) {}
 - +0x70 = OnesieRequestProto__InternalSerialize_32D373C
 - virtual uint8_t _InternalSerialize(uint8_t ptr, io::EpsCopyOutputStream* stream) const = 0;

附录

下面列出相关参考资料。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-04 14:44:04

参考资料

- 【整理】Protobuf中的MessageLite
- 【整理】Protocol Buffers即protobuf总结
- 【未解决】研究YouTube逻辑：OnesieRequestProto类的具体vtable函数实现
- 【已解决】研究YouTube逻辑：从NSData直接解析出protobuf的类YTIAdBreakRequest
-
- java/core/src/main/java/com/google/protobuf/MessageLite.java - external/github.com/google/protobuf - Git at Google (googlesource.com)
- protobuf/src/google/protobuf/message_lite.h - chromium/src/third_party - Git at Google (googlesource.com)
-
- [message_lite.h | Protocol Buffers | Google Developers](https://message_lite.h)
- third_party/protobuf/src/google/protobuf/message_lite.cc - chromium/src - Git at Google (googlesource.com)
- PB协议报错 it is missing required fields: (cannot determine missing fields for lite message) _军说网事的博客-CSDN博客
- examples - external/github.com/google/protobuf - Git at Google (googlesource.com)
- [原创]protobuf还原——从ida中还原.proto文件-软件逆向-看雪论坛-安全社区|安全招聘|bbs.pediy.com
- Interface MessageLite | Java client library | Google Cloud
- Protobuf-Message相关类_felix_xw的博客-CSDN博客_message proto
- [protobuf::Any Message vs MessageLite interface \(google.com\)](https://protobuf::Any Message vs MessageLite interface (google.com))
-

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-10 10:35:47