

# 目录

前言	1.1
iOS逆向YouTube概览	1.2
逆向YouTube过程	1.3
砸壳ipa	1.3.1
安装ipa	1.3.1.1
静态分析	1.3.2
导出头文件	1.3.2.1
IDA分析代码逻辑	1.3.2.2
动态调试	1.3.3
分析界面元素	1.3.3.1
Reveal	1.3.3.1.1
Cycript	1.3.3.1.2
调试代码逻辑	1.3.3.2
Xcode+MonkeyDev	1.3.3.2.1
重复上述过程	1.3.4
期间涉及	1.3.5
恢复符号表	1.3.5.1
Charles抓包	1.3.5.2
逆向YouTube产出	1.4
YouTube的app内部逻辑	1.4.1
很多类的具体实现	1.4.1.1
返回数据	1.4.1.2
YouTube广告过滤相关	1.4.2
hook的类	1.4.2.1
IDA伪代码	1.4.2.2
函数调用逻辑	1.4.2.3
常见url	1.4.2.4
逆向YouTube总结	1.5
子教程	1.6
附录	1.7
参考资料	1.7.1

# iOS逆向开发：YouTube逆向

- 最新版本： v0.8
- 更新时间： 20221104

## 简介

以YouTube逆向介绍典型的iOS逆向的流程。先是YouTube逆向概述，再是详细介绍iOS的YouTube逆向的详细过程，包括砸壳ipa以及安装ipa、静态分析的导出头文件、IDA分析代码逻辑、动态调试的分析界面元素和MonkeyDev调试，其中界面分析用Reveal、Cycrypt；MonkeyDev调试典型逻辑是，猜测类和函数、写hook代码、加断点调试；期间涉及到恢复符号表、Charles抓包；然后介绍逆向的产出和相关总结；产出包括app内部逻辑和广告过滤相关内容；以及相关子教程。

## 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\\_youtube\\_reverse: iOS逆向开发：YouTube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逆向开发：YouTube逆向 book.crifan.org](#)
- [iOS逆向开发：YouTube逆向 crifan.github.io](#)

### 离线下载阅读

- [iOS逆向开发：YouTube逆向 PDF](#)
- [iOS逆向开发：YouTube逆向 ePUB](#)
- [iOS逆向开发：YouTube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。

## 更多其他电子书

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](#)

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 14:57:53

# iOS逆向YouTube概览

此处去以iOS逆向YouTube的app为例，介绍典型的iOS逆向的过程。

此处逆向YouTube的目的暂且定为，尝试搞懂YouTube广告过滤的可能性和底层涉及相关机制。

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：

2022-10-27 10:59:12

# 逆向YouTube过程

下面介绍，如何进行iOS逆向YouTube的过程。

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：  
2022-10-27 10:57:00

## 砸壳ipa

TODO:

- 把下面多个帖子内容整理过来
- 

在iPhone越狱后，接着去逆向iOS的YouTube。

先准备好砸壳后的ipa。

两种可能=来源：

- 别人的
  - 别人给的
  - 网上找的
    - 【部分解决】下载16.36.4旧版本YouTube的ipa
    - 【未解决】寻找和下载16.36.4或更低版本的YouTube的ipa
- 自己砸壳得到的

## 砸壳YouTube得到ipa

此处主要介绍后者：

自己从AppStore中安装YouTube，再砸壳出ipa

- 先去AppStore中安装YouTube
  - 【记录】国行越狱iPhone安装YouTube
  - 【无需解决】国内越狱iPhone如何翻墙上外网使用YouTube
- 再去砸壳
  - 【已解决】Mac中用frida-ios-dump去砸壳出YouTube的ipa文件
  - 【已解决】frida-ios-dump砸壳YouTube但是无法启动应用
- 安装ipa到iPhone，确保YouTube正常使用
  - 说明：动态调试才需要，一般静态分析，不需要
  - 涉及帖子
    - 【已解决】用Filza安装YouTube的ipa确认能正常运行
    - 【已解决】iPhone中查看已安装app的包名
    - 【记录】测试16.44.4旧版YouTube的ipa广告拦截是否也时间重置

## 其他相关

### 从AppStore中找历史版本

- 【未解决】Cydia中安装AppStore++寻找YouTube的16.36.4的旧版本
- 【未解决】试试CokePokes的原版AppStore++能否安装旧版YouTube
- 【已解决】安装CokePokes的原版AppStore++
- 【未解决】用Cydia致尚官方源安装的AppStore++安装旧版YouTube

- 【未解决】AppStore++用Manual Install Advanced去尝试安装旧版YouTube
- 【已解决】Cydia中安装应用降级AppStore++插件

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:  
2022-10-27 11:01:06

# 安装ipa

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：  
2022-10-27 10:52:10

# 静态分析

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：  
2022-10-27 10:52:34

## 导出头文件

TODO:

- 【记录】 class-dump 导出头文件: YouTube 的动态库二进制 Module\_Framework
  - 【未解决】 class-dump 导出 YouTube 头文件 报错: Unknown load command 0x00000032
  - 【已解决】 Mac 中用 class-dump 导出 YouTube 头文件
- 

用 `class-dump` 导出 YouTube 的头文件

此处 YouTube 有2个二进制：

- `YouTube`
- `Module_Framework`

所以要分别导出对应的头文件。

导出后的效果：

- 2.7万多个头文件

◦

- 3800多个头文件

◦

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:  
2022-10-27 11:27:04

# IDA分析代码逻辑

主要涉及，分别用IDA加载和分析，YouTube的2个二进制的文件：

- YouTube
- Module\_Framework

然后分析其代码逻辑。

其中最核心用法是，找到可以的要研究的类的函数后，去 Functions 窗口中，通过函数名搜索函数，找到对应汇编代码，再 F5 查看反编译后的伪代码，尝试搞懂代码的具体逻辑。

## 特殊：导出IDA全部伪代码

为了更好的，全局去搜一些类、变量、字符串等，用于iOS逆向研究YouTube的逻辑

所以考虑去导出全部的IDA伪代码：

TODO:

- 【未解决】IDA图形界面GUI模式使用ida-batch\_decompile批量导出全部YouTube的Module\_Framework全部伪代码
- 【未解决】用IDA的命令行文本模式使用ida-batch\_decompile批量导出全部YouTube的Module\_Framework全部伪代码
- 【未解决】用插件导出IDA的YouTube的Module\_Framework的全部反汇编的源码伪代码
- 【记录】IDA中用idat64的Batch Mode尝试反编译导出全部源代码：试试二进制小点的
- 【已解决】IDA中用idat64的Batch Mode尝试反编译导出YouTube的Module\_Framework全部代码伪代码

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：

2022-10-27 10:54:02

# 动态调试

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：  
2022-10-27 10:54:29

# 分析界面元素

动态调试 常会涉及：分析YouTube的界面元素

常用工具：

- [Reveal](#)
- [Cycrypt](#)

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：  
2022-10-27 10:51:21

# Reveal

TODO:

- Reveal
  - 【已解决】Mac中下载和安装Reveal
  - 【已解决】MonkeyDev中如何使用Reveal调试YouTube广告页面元素
  - 【已解决】Reveal报错：The app is linked against an older version of the Reveal library
  - 【记录】通过Reveal查看YouTube广告页面元素
  - 【已解决】Mac中Reveal无法启动
  - 【间接解决】Reveal加载页面始终无法完成无法刷新
  - 【记录】XCode和Reveal配合调试YouTube的广告相关逻辑
  - 【记录】通过Reveal查看页面元素找YouTube广告相关类
  - 【记录】用Reveal查看YouTube跳过广告相关页面的元素和类

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：  
2022-10-27 10:58:58

## Cycript

TODO:

- Cycript
  - 【已解决】用MonkeyDev中Cycript去调试YouTube的UI页面的元素
  - 【记录】Mac中用csrutil关闭SIP
  - 【未解决】install\_name\_tool
  - 【记录】Cycript输出的YouTube的当前页面的类
  - 【已解决】Mac中运行Cycript报错：dyld Library not loaded libruby.2.0.0.dylib

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：  
2022-10-27 10:53:22

# 调试代码逻辑

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:  
2022-10-27 10:57:23

## Xcode+MonkeyDev

MonkeyDev调试YouTube的代码逻辑：

TODO:

- 【记录】恢复自己Mac的iOS逆向开发环境
- 【已解决】Xcode调试越狱iPhone6中的YouTube
- 【已解决】XCode+MonkeyDev动态调试YouTube的ipa
- 【记录】研究YouTube广告拦截导致视频从头播放的原因：XCode+MonkeyDev动态调试
- 【记录】恢复iOS逆向Xcode调试YouTube的开发环境

期间涉及：写hook代码，加断点调试：

找到一些可疑的，需要调试确认是否被调用到的类，则就是：去写hook代码，加断点，去调试看看是否触发：

- 如果触发：说明推测和判断是对的
- 如果没触发：说明推测出错
  - 再去找其他可疑的类，再继续去写hook加断点去调试

涉及的过程大概是：

- 【未解决】研究iPhone中YouTube的app播放视频中间广告导致从头播放
- 【已解决】iOS的Theos的Logos的hook代码中获取当前YouTube类YTWatchController的属性值activeVideoID
- 【记录】XCode的lldb调试YouTube的类：YTWatchController
- 【已解决】XCode+MonkeyDev动态调试YouTube：让程序停止在出现广告的页面
- 【基本解决】XCode的lldb中输出YouTube类MDXSession实例的属性值currentVideoID
- 【记录】给MonkeyDev的YouTube的hook中加上通用log代码
- 【已解决】XCode的lldb调试输出YouTube类YTWatchController实例的属性值和调用函数
- 【记录】XCode+MonkeyDev动态调试YouTube：寻找和广告相关的类和逻辑
- 【记录】XCode中Pause暂停去找YouTube的广告相关的类
- 【记录】研究YouTube广告逻辑：MDXPlaybackController的didPressSkipAd
- 【记录】研究YouTube广告逻辑：IDA中搜skipAd
- 【未解决】YouTube动态调试：写hook代码监控视频播放时间被重置变成0
- 以及后续换其他思路
  - 【记录】从.googlevideo.com入手找YouTube广告视频相关逻辑
  - 【记录】研究YouTube函数：resetWatchTime
  - 【记录】XCode+MonkeyDev动态调试YouTube：寻找和广告相关的类和逻辑
  - 【未解决】研究YouTube逻辑：找到最初的广告视频请求
  - 【未解决】研究YouTube逻辑：找到最初广告视频请求的响应
  - 【已解决】研究YouTube逻辑：找到带ctier=A的response和error出错的地方
  - 【已解决】研究YouTube逻辑：ctier=A时的request和response的函数调用顺序
  - 【未解决】研究YouTube逻辑：寻找搞懂广告视频ID获取的大流程
  - 【已解决】研究YouTube逻辑：抓包iOS版YouTube的广告视频ID产生的大流程

2022-10-27 10:58:08

## 重复上述过程

对于普通的iOS逆向，就重复上述过程

对于此处，逆向YouTube来说，具体是：

- 静态分析
  - 研究和搜索头文件中的类，找出可疑的类
    - YouTube\_17.08.2\_headers
      - header\_ModuleFramework
      - headers\_YouTube

- 研究导出的字符串等资源中，是否有我们要的内容
  - staticAnalysis

```
└── YouTube_17.08.2
    ├── Module_Framework
    │   ├── Module_Framework_nm.txt
    │   ├── Module_Framework_otool_l.txt
    │   ├── Module_Framework_otool_oV.txt
    │   └── Module_Framework_strings.txt
    └── YouTube
        ├── YouTube_nm.txt
        ├── YouTube_otool_l.txt
        └── YouTube_otool_oV.txt
```

- 动态调试
  - 分析界面元素
    - 从界面元素入手，找到对应的类或父类
      - 作为后续深入研究的切入点：找类的相关头文件
  - 调试代码逻辑
    - 针对可疑的类和函数，写hook代码，加断点调试
  - 期间涉及
    - 加过滤条件
      - 比如找到了 http 的 request 和 response 的类，还要继续加过滤条件
        - 过滤出我们所关心的类，比如initplayback, videoplayback等url
          - <https://xxx.googlevideo.com/initplayback?xxx>
          - <https://xxx.googlevideo.com/videoplayback?xxx>
      - log太多了，如何优化减少log
        - TODO:
          - 【已解决】研究YouTube逻辑：log日志打印优化每隔几次才输出

## 期间涉及

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：  
2022-10-27 10:52:11

## 恢复符号表

为了动态调试 常会涉及：恢复符号表

TODO:

- 【已解决】 restore-symbol 恢复 YouTube 的 Block 符号表报错：Address not found in the image
- 【记录】 YouTube 恢复符号表后 Xcode 调试效果
- 【已解决】 给 YouTube 恢复符号表方便 Xcode 调试
- 【记录】 YouTube 恢复符号表恢复出部分无效的 symbol 符号

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：  
2022-10-27 10:59:36

# Charles抓包

期间涉及：iOS的app的抓包：

TODO:

- 【已解决】Mac用Charles抓包iPhone中YouTube视频过滤广告导致从头播放的数据包

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：

2022-10-27 10:53:21

# 逆向YouTube产出

此处整理出iOS逆向YouTube输出的相关结论。

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:  
2022-10-27 11:06:50

# YouTube的app内部逻辑

TODO:

- 网络请求
  - 【整理】 YouTube广告拦截从头播放视频：相关的网络请求和大概逻辑
- 网络协议
  - Cronet
    - 【整理】 Cronet相关源码和定义和具体实现
    - 【已解决】 YouTube中struct Cronet.UrlRequestCallback的具体定义实现
    - 【已解决】 iOS的YouTube中Cronet.UrlRequestCallback的OnFailed中cronet=10和quic协议关系
  - QUIC
    - 【整理】 YouTube相关： QUIC协议和源码
    - 【整理】 YouTube涉及到的Network、 quic等Chromium相关源码
    - 【整理】 YouTube涉及到Chromium中Cronet相关源码和文档
- YouTube视频格式
  - 【整理】 YouTube中视频相关信息： ISO Media file produced by Google Inc
  - 【未解决】 YouTube的response的数据视频格式： ftyp=dash
  - 【整理】 YouTube逻辑： 视频流格式代码
- HTTP协议
  - 【已解决】 研究YouTube视频内容请求： Transfer-Encoding chunked
- 【整理】 YouTube中Module\_Framework函数对应关系 20220804
- YouTube广告过滤规则
  - 圈X
    - 【记录】 研究YouTube逻辑： 验证最新圈X的规则能否拦截广告
    - 【记录】 pico中使用圈X最新规则的YouTube广告拦截效果
    - 【记录】 测试最新版过滤规则的iOS的YouTube的app中广告过滤效果
- 调试相关
  - 【记录】 Xcode+MonkeyDev动态调试YouTube： 广告页面相关类
  - 【记录】 调试YouTube广告页面： 调试相关数据

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：  
2022-10-27 11:10:35

# 很多类的具体实现

内部很多类的具体实现

## SRLRegistry\_sharedInstance

`SRLRegistry_sharedInstance.coffee`

```
(lldb) po 0x00000002839803c0
SRLRegistry: 0x2839803c0
Single bindings
{
    "ELMByteStore_API" = "SRLImplementation<0x283690d50, instanceType: ELMByteStore, lifetime: 0, scopeType: 0, configuration: None>";
    "ELMJSSModuleCache_API" = "SRLImplementation<0x2837f7d80, instanceType: ELMJSSModuleCache, lifetime: 0, scopeType: 0, configuration: None>";
    "ELMResourceLoader_API" = "SRLImplementation<0x283710a50, instanceType: ELMResourceLoader, lifetime: 0, scopeType: 0, configuration: None>";
    "ELMRuntimeConfigFactory_API" = "SRLImplementation<0x283762f40, instanceType: YTELMRuntimeConfigFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "ELMStore_API" = "SRLImplementation<0x283690cf0, instanceType: YTELMSStore, lifetime: 0, scopeType: 0, configuration: None>";
    "GHKY2021BugfixesService_API" = "SRLImplementation<0x28363c990, instanceType: GHKY2021Bugfixes, lifetime: 0, scopeType: 0, configuration: None>";
    "GIKAccountDeletionWipeoutManager_API" = "SRLImplementation<0x2837c9650, instanceType: GIKAccountDeletionWipeoutManagerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "GIKAccountSourceService_API" = "SRLImplementation<0x28376ca50, instanceType: GIKAccountSourceServiceImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "GIKSACountSourceService_API" = "SRLImplementation<0x28376cae0, instanceType: GIKSACountSourceServiceImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "GIKSABridgeService_API" = "SRLImplementation<0x28376cb10, instanceType: GIKSABridgeServiceImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "GIKSharedLibraryService_API" = "SRLImplementation<0x28376ca80, instanceType: GIKSharedLibraryServiceImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "HAMClock_API" = "SRLImplementation<0x283651f20, instanceType: HAMSystemClock, lifetime: 0, scopeType: 0, configuration: None>";
    "HAMDataLoadTaskObserver_API" = "SRLImplementation<0x2834811a0, instanceType: MLHAMDataLoadTaskObserverImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "HAMIIdleTimer_API" = "SRLImplementation<0x2834a3420, instanceType: HAMIIdleTimerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "HAMSystemNotifications_API" = "SRLImplementation<0x2834b3960, instanceType: YTSystemNotifications, lifetime: 0, scopeType: 0, configuration: None>";
    "IGDManager_API" = "SRLImplementation<0x283698030, instanceType: IGDManager, lifetime: 0, scopeType: 0, configuration: None>";
    "MDXConfig_API" = "SRLImplementation<0x283763660, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "MDXServices_API" = "SRLImplementation<0x283763810, instanceType: MDXRealServices, lifetime: 0, scopeType: 0, configuration: None>";
    "MLAVAssetDownloader_API" = "SRLImplementation<0x2836f2f70, instanceType: MLAVAssetDownloader, lifetime: 0, scopeType: 0, configuration: None>";
    "MLAudioSession_API" = "SRLImplementation<0x2834846c0, instanceType: MLAUDIOSESSION, lifetime: 0, scopeType: 0, configuration: None>";
```

```

    "MLCDM_API" = "SRLImplementation<0x2834a5e30, instanceType: MLWidevineCDMImpl, lifetime: 0,
    scopeType: 0, configuration: None>";
    "MLCacheProvider_API" = "SRLImplementation<0x2834b2460, instanceType: MLDefaultCacheProvide
r, lifetime: 0, scopeType: 0, configuration: None>";
    "MLCaptionConfig_API" = "SRLImplementation<0x2834a9aa0, instanceType: YTSettings, lifetime:
0, scopeType: 0, configuration: None>";
    "MLCaptionControllerFactory_API" = "SRLImplementation<0x2834a7d80, instanceType: MLDefaultC
aptionControllerFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "MLCaptionDisplayControllerFactory_API" = "SRLImplementation<0x2834a39f0, instanceType: YTG
LCaptionDisplayControllerFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "MLCaptionDisplaySettingsProvider_API" = "SRLImplementation<0x2834ac4e0, instanceType: MLDe
faultCaptionDisplaySettingsProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "MLCaptionService_API" = "SRLImplementation<0x283682100, instanceType: MLDefaultCaptionServ
ice, lifetime: 0, scopeType: 0, configuration: None>";
    "MLClientInfoFiller_API" = "SRLImplementation<0x28348a190, instanceType: MLClientInfoFiller
Impl, lifetime: 0, scopeType: 0, configuration: None>";
    "MLClockOffsetEstimator_API" = "SRLImplementation<0x2834b1440, instanceType: MLClockOffsetE
stimator, lifetime: 0, scopeType: 0, configuration: None>";
    "MLDataPlanController_API" = "SRLImplementation<0x2836098f0, instanceType: MLDataPlanContro
ller, lifetime: 0, scopeType: 0, configuration: None>";
    "MLDefaultViewportSizeProvider_API" = "SRLImplementation<0x283482e50, instanceType: MLSingl
etonViewportSizeProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "MLHAMAUDIOEngineProvider_API" = "SRLImplementation<0x283487690, instanceType: MLHAMAUDIOEn
gineProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "MLHAMDataLoaderProvider_API" = "SRLImplementation<0x283608f90, instanceType: MLHAMDataLoad
erProviderImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "MLInitialFormatSelectionConfig_API" = "SRLImplementation<0x28348a0d0, instanceType: YTSett
ings, lifetime: 0, scopeType: 0, configuration: None>";
    "MLInnerTubeContextDecorator_API" = "SRLImplementation<0x28360a400, instanceType: MLInnerTu
beContextDecorator, lifetime: 0, scopeType: 4, configuration: None>";
    "MLLocalAVAssetBundleManager_API" = "SRLImplementation<0x2836f2df0, instanceType: MLLocalAV
AssetBundleManager, lifetime: 0, scopeType: 0, configuration: None>";
    "MLMediaCapabilitiesProvider_API" = "SRLImplementation<0x28348a100, instanceType: MLMediaCa
pabilitiesProviderImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "MLNetworkStatsProvider_API" = "SRLImplementation<0x2835627f0, instanceType: MLNetworkStats
Provider, lifetime: 0, scopeType: 0, configuration: None>";
    "MLOfflineRepoFactory_API" = "SRLImplementation<0x2836f2ee0, instanceType: YTOfflineRepoFac
tory, lifetime: 0, scopeType: 0, configuration: None>";
    "MLOfflineTracklistCaptionProvider_API" = "SRLImplementation<0x2834a9ad0, instanceType: YTO
fflineVideoCaptionsProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "MLOfflineVideoDownloadController_API" = "SRLImplementation<0x2836f33f0, instanceType: MLOF
flineVideoDownloadControllerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "MLOnesieDataSaverConfig_API" = "SRLImplementation<0x283562910, instanceType: MLOnesieDataS
averConfigImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "MLOnesieFetcherService_API" = "SRLImplementation<0x2834b1560, instanceType: MLOnesieFetch
erService, lifetime: 0, scopeType: 0, configuration: None>";
    "MLOnesieHostProvider_API" = "SRLImplementation<0x283561380, instanceType: MLOnesieHostProv
ider, lifetime: 0, scopeType: 0, configuration: None>";
    "MLOnesieRequestFactory_API" = "SRLImplementation<0x2834b2e50, instanceType: MLOnesieReques
tFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "MLOnesieVideoDataProvider_API" = "SRLImplementation<0x2834a69a0, instanceType: MLOnesieVid
eoDataProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "MLPIPController_API" = "SRLImplementation<0x2835657a0, instanceType: MLPIPController, life
time: 0, scopeType: 0, configuration: None>";
    "MLPlayerPool_API" = "SRLImplementation<0x283483a80, instanceType: MLPlayerPoolImpl, lifeti
me: 0, scopeType: 0, configuration: None>";

```

```

    "MLPlayerViewFactory_API" = "SRLImplementation<0x283483ab0, instanceType: YTGLMediaPlayerViewFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "MLScreenCapturedStatus_API" = "SRLImplementation<0x2834a3630, instanceType: MLScreenCapturedStatusImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "MLStreamStoreConfig_API" = "SRLImplementation<0x283483b10, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "MLStreamStore_API" = "SRLImplementation<0x283483ae0, instanceType: MLStreamStore, lifetime: 0, scopeType: 0, configuration: None>";
    "MLTFLiteModelService_API" = "SRLImplementation<0x28360a8e0, instanceType: MLTFLiteModelService, lifetime: 0, scopeType: 0, configuration: None>";
    "MLTFLiteNetworkModel_API" = "SRLImplementation<0x283562700, instanceType: MLTFLiteNetworkModel, lifetime: 0, scopeType: 0, configuration: None>";
    "MLUserFeatures_API" = "SRLImplementation<0x2834b2df0, instanceType: MLUserFeaturesImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "MLVideoDownloadTaskDelegate_API" = "SRLImplementation<0x2836f2e20, instanceType: MLVideoDownloaderMediaDownloaderShim, lifetime: 0, scopeType: 0, configuration: None>";
    "MLVideoDownloaderDelegate_API" = "SRLImplementation<0x2836f2c40, instanceType: MLVideoDownloaderMediaDownloaderShim, lifetime: 0, scopeType: 0, configuration: None>";
    "MLVideoDownloaderMediaDownloaderShim_API" = "SRLImplementation<0x2836f2e50, instanceType: MLVideoDownloaderMediaDownloaderShim, lifetime: 0, scopeType: 0, configuration: None>";
    "MLVideoDownloader_API" = "SRLImplementation<0x2836f2d90, instanceType: MLVideoDownloader, lifetime: 0, scopeType: 0, configuration: None>";
    "PHTPhenotypeFlagsRegistryService_API" = "SRLImplementation<0x28363c9c0, instanceType: PHTPhenotypeFlagsRegistry, lifetime: 0, scopeType: 0, configuration: None>";
    "RCHServiceConfigurationProvider_API" = "SRLImplementation<0x283682af0, instanceType: YTRCHServiceConfigurationProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "RCHService_API" = "SRLImplementation<0x283682ac0, instanceType: RCHServiceImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "SSOConfigurationProvider_API" = "SRLImplementation<0x2837c9e60, instanceType: YTSSOServiceFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "SSOServiceContainer_API" = "SRLImplementation<0x28376d200, instanceType: SSOServiceContainerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "SSOService_API" = "SRLImplementation<0x2837c9e90, instanceType: SSOService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAccountLinkingMenuVisibilityHandler_API" = "SRLImplementation<0x282b9b7e0, instanceType: YTAccountLinkingMenuVisibilityHandler, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAccountScopedCommandRegistrar_API" = "SRLImplementation<0x283574600, instanceType: YTAccountScopedCommandRegistrar, lifetime: 0, scopeType: 1, configuration: None>";
    "YTAccountScopedCommandRouter_API" = "SRLImplementation<0x2835725b0, instanceType: YTAccountScopedCommandRouter, lifetime: 0, scopeType: 1, configuration: None>";
    "YTAccountScopedInnerTubeContextFactory_API" = "SRLImplementation<0x2834e9410, instanceType: YTAccountScopedInnerTubeContextFactory, lifetime: 0, scopeType: 1, configuration: None>";
    "YTAccountScopedInnerTubeRequestFactory_API" = "SRLImplementation<0x2834e91d0, instanceType: YTAccountScopedInnerTubeRequestFactory, lifetime: 0, scopeType: 1, configuration: None>";
    "YTAccountScopedInnerTubeRequestorFactory_API" = "SRLImplementation<0x2834e9260, instanceType: YTAccountScopedInnerTubeRequestorFactory, lifetime: 0, scopeType: 1, configuration: None>";
    "YTAccountScopedInnerTubeResponseCache_API" = "SRLImplementation<0x2834e91a0, instanceType: YTAccountScopedInnerTubeResponseCache, lifetime: 0, scopeType: 1, configuration: None>";
    "YTAccountScopedInnerTubeStackConfiguration_API" = "SRLImplementation<0x2834e92f0, instanceType: YTAccountScopedInnerTubeStackConfiguration, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAccountScopedVisitorDataRequestDecorator_API" = "SRLImplementation<0x2834e9230, instanceType: YTAccountScopedVisitorDataRequestDecorator, lifetime: 0, scopeType: 1, configuration: None>";
    "YTActiveStateDescriptorSubControllerProtocolFactory_API" = "SRLImplementation<0x28360b3c0, instanceType: YTCoreActiveStateDescriptorSubControllerFactory, lifetime: 0, scopeType: 0, conf

```

```

iguration: None>;
    "YTActiveVideoNotifier_API" = "SRLImplementation<0x28376a160, instanceType: YTActiveVideoNotifier, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdBreakFetcherAPI_API" = "SRLImplementation<0x282bbb0f0, instanceType: YTAdBreakRendererFetcher, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdCuepointAPI_API" = "SRLImplementation<0x2834879f0, instanceType: YTAdCuepointEventCenter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdElementsCellAPIImpl_API" = "SRLImplementation<0x2837014d0, instanceType: YTAdElementsCellAPIImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdElementsCellExternallyManagedSlotAdapter_API" = "SRLImplementation<0x28362a040, instanceType: YTAdElementsCellExternallyManagedSlotAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdInnerTubeConfig_API" = "SRLImplementation<0x283711680, instanceType: YTRealAdInnerTubeConfig, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdInterruptAPI_API" = "SRLImplementation<0x282b88240, instanceType: YTAdsPlaybackService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdIntervalManagerTriggerAdapter_API" = "SRLImplementation<0x282b887e0, instanceType: YTAdIntervalManagerTriggerAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdIntervalManager_API" = "SRLImplementation<0x282bb8390, instanceType: YTAdIntervalManager, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdLoggingAPI_API" = "SRLImplementation<0x2834ae5b0, instanceType: YTAdsPlaybackService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdMacroExpanderProvider_API" = "SRLImplementation<0x283650570, instanceType: YTAdMacroExpanderProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsControlFlowEventTriggerAdapter_API" = "SRLImplementation<0x28349be40, instanceType: YTAdsControlFlowEventTriggerAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsControlFlowIDGenerator_API" = "SRLImplementation<0x28362a010, instanceType: YTAdsControlFlowIDGeneratorImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsControlFlowManager_API" = "SRLImplementation<0x283487600, instanceType: YTAdsControlFlowManagerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsControlFlowObservable_API" = "SRLImplementation<0x28349a2e0, instanceType: YTAdsControlFlowManagerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsControlFlowOpportunityReceivedCommandAPIListener_API" = "SRLImplementation<0x283629ec0, instanceType: YTCommandDrivenExternallyManagedSlotOpportunityAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsControlFlowOpportunityReceivedCommandHandler_API" = "SRLImplementation<0x2836299e0, instanceType: YTAdsControlFlowOpportunityReceivedCommandHandler, lifetime: 0, scopeType: 1, configuration: None>";
    "YTAdsEventLoggingController_API" = "SRLImplementation<0x283629f20, instanceType: YTAdsEventLoggingControllerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsFrequencyCapProtocol_API" = "SRLImplementation<0x2834b0db0, instanceType: YTFrequencyCapTracker, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsPlaybackService_API" = "SRLImplementation<0x2834a2e50, instanceType: YTAdsPlaybackService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsPlayerServices_API" = "SRLImplementation<0x2834963a0, instanceType: YTRealAdsPlayerServices, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAdsSKOverlayService_API" = "SRLImplementation<0x28348f0f0, instanceType: YTAdsMainAppSKOOverlayServiceImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAgeCapabilitiesService_API" = "SRLImplementation<0x283763510, instanceType: YTAgeCapabilitiesServiceImpl, lifetime: 0, scopeType: 4, configuration: None>";
    "YTAppBrowseSwipeLeftCommandProvider_API" = "SRLImplementation<0x2836bf300, instanceType: YTReelCameraCommandCache, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppConfig_API" = "SRLImplementation<0x28357f1e0, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppFlowStateController_API" = "SRLImplementation<0x283681920, instanceType: YTAppFlowStateController, lifetime: 0, scopeType: 0, configuration: None>";

```

```

    "YTAppHomeElementsRegistry_API" = "SRLImplementation<0x2836ea0a0, instanceType: YTAppHomeElementsRegistry, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppMealbarPromoController_API" = "SRLImplementation<0x28369df20, instanceType: YTAppMealbarPromoController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppOfflineContentRepairController_API" = "SRLImplementation<0x2836fd380, instanceType: YTAppOfflineContentRepairController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppOfflineResumeController_API" = "SRLImplementation<0x28349e340, instanceType: YTAppOfflineResumeController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppOfflineResumeSyncController_API" = "SRLImplementation<0x28349e4f0, instanceType: YTAppOfflineResumeSyncController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppOfflineVideoController_API" = "SRLImplementation<0x283480840, instanceType: YTAppOfflineVideoController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppSettingsStore_API" = "SRLImplementation<0x2836feb80, instanceType: YTAppSettingsStore, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppStartupInstrumentationTransmitter_API" = "SRLImplementation<0x28356f0c0, instanceType: YTGECSITransmitter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppStartupReasonController_API" = "SRLImplementation<0x283689680, instanceType: (null), lifetime: 1, scopeType: 0, configuration: None>";
    "YTAppThemeSettingPromoController_API" = "SRLImplementation<0x2836cc600, instanceType: YTAppThemeSettingPromoController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppTransactionQueueController_API" = "SRLImplementation<0x28369e0d0, instanceType: YTAppTransactionQueueController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppUserDefaultsMigrator_API" = "SRLImplementation<0x283481c50, instanceType: YTAppUserDefaultsMigrator, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAppViewController_API" = "SRLImplementation<0x283690c60, instanceType: YTAppViewController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAttestationChallengeService_API" = "SRLImplementation<0x283763570, instanceType: YTAttestationChallengeService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTAudioSession_API" = "SRLImplementation<0x2837cb660, instanceType: MLAudioSession, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBackAsStackManager_API" = "SRLImplementation<0x283698660, instanceType: YTBackAsStackManager, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBackgroundTasksController_API" = "SRLImplementation<0x2836cadc0, instanceType: YTAppBackgroundTasksController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBackgroundabilityPolicy_API" = "SRLImplementation<0x2834849c0, instanceType: YTBackgroundabilityPolicy, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBedtimeReminderController_API" = "SRLImplementation<0x2836ad6e0, instanceType: YTBedtimeReminderController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBelowPlayerCompanionRenderingAPIImpl_API" = "SRLImplementation<0x2828ad500, instanceType: YTBelowPlayerCompanionRenderingAPIImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBelowPlayerImmersiveRenderingAPIImpl_API" = "SRLImplementation<0x2828ad530, instanceType: YTBelowPlayerImmersiveRenderingAPIImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBelowPlayerImmersiveSlotFulfillmentAdapterFactory_API" = "SRLImplementation<0x282bcc450, instanceType: YTBelowPlayerImmersiveSlotFulfillmentAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBelowPlayerSlotFulfillmentAdapterFactory_API" = "SRLImplementation<0x282bcc2a0, instanceType: YTBelowPlayerSlotFulfillmentAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBlobStorageBlocking_API" = "SRLImplementation<0x2836b23a0, instanceType: YTPINDiskCacheBlobStorage, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBlobStorage_API" = "SRLImplementation<0x28360a340, instanceType: YTPINDiskCacheBlobStorage, lifetime: 0, scopeType: 0, configuration: None>";
    "YTBrowseService_API" = "SRLImplementation<0x283714000, instanceType: YTAppBrowseService, lifetime: 0, scopeType: 4, configuration: None>";
    "YTCTTelephonyNetworkInfo_API" = "SRLImplementation<0x2837f8690, instanceType: YTCTTelephonyNetworkInfoImpl, lifetime: 0, scopeType: 0, configuration: None>";

```

```

    "YTCastSessionController_API" = "SRLImplementation<0x282b9af10, instanceType: YTNoOpCastSessionController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCellFactory_API" = "SRLImplementation<0x28361f030, instanceType: YTInnerTubeCellFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTClearcutController_API" = "SRLImplementation<0x2837c8930, instanceType: YTClearcutController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTClientEnvironment_API" = "SRLImplementation<0x2834ab090, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTClientEventIdController_API" = "SRLImplementation<0x2837fce10, instanceType: YTClientEventIdController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTClipController_API" = "SRLImplementation<0x282b82070, instanceType: YTClipController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTClipsElementsRegistry_API" = "SRLImplementation<0x282b820d0, instanceType: YTClipsElementsRegistry, lifetime: 0, scopeType: 0, configuration: None>";
    "YTClockEntityTransformationTrigger_API" = "SRLImplementation<0x2836f1fb0, instanceType: YTClockEntityTransformationTrigger, lifetime: 0, scopeType: 0, configuration: None>";
    "YTColdConfig_API" = "SRLImplementation<0x28379b5a0, instanceType: YTColdConfig, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCollectionScrollTrackerSamplerGate_API" = "SRLImplementation<0x2836c6e50, instanceType: YTCollectionScrollTrackerSamplerGate, lifetime: 0, scopeType: 0, configuration: None>";
    "YTColorStyle_API" = "SRLImplementation<0x283767570, instanceType: YTAppColorStyle, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCommandRegistrar_API" = "SRLImplementation<0x283681b60, instanceType: YTCommandRegistrar, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCommandRouter_API" = "SRLImplementation<0x283763630, instanceType: YTCommandRouter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCompanionAdSectionControllerProvider_API" = "SRLImplementation<0x282bfaa60, instanceType: YTCompanionAdSectionControllerProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "YTConfigService_API" = "SRLImplementation<0x28357edf0, instanceType: YTConfigService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTConnectionStatusController_API" = "SRLImplementation<0x283698930, instanceType: YTConnectionStatusController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTConsistencyTokenJarVolatileCache_API" = "SRLImplementation<0x283714240, instanceType: YTConsistencyTokenJarVolatileCache, lifetime: 0, scopeType: 0, configuration: None>";
    "YTContentPlaybackInfoAPI_API" = "SRLImplementation<0x283487ab0, instanceType: YTAdsPlaybackService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTContentVideoPlayerResponseSlotOpportunityAdapter_API" = "SRLImplementation<0x283487c30, instanceType: YTContentVideoPlayerResponseSlotOpportunityAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCoreDataDelayedEventQueue_API" = "SRLImplementation<0x2837ef630, instanceType: YTCoreDataDelayedEventQueue, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCrashConfig_API" = "SRLImplementation<0x283689500, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCrashContextController_API" = "SRLImplementation<0x283689560, instanceType: YTCrashContextController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCrashContextProtocol_API" = "SRLImplementation<0x2837fcba0, instanceType: YTCrashContextController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCrashControllerDelegate_API" = "SRLImplementation<0x2836895c0, instanceType: YTCrashController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCrashController_API" = "SRLImplementation<0x2836894a0, instanceType: YTCrashController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCrashRestoreVideoControllerConfig_API" = "SRLImplementation<0x2836842a0, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTCrashRestoreVideoController_API" = "SRLImplementation<0x283684270, instanceType: YTCrashRestoreVideoController, lifetime: 0, scopeType: 0, configuration: None>";
    "YT CuepointOpportunityAdapter_API" = "SRLImplementation<0x283487960, instanceType: YT Cuepoi

```

```

    ntOpportunityAdapter, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDAIAdsPlaybackLifecycleAPI_API" = "SRLImplementation<0x283487b40, instanceType: YTDAIAds
    PlaybackLifecycleController, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDAIContentPlaybackLifecycleAPI_API" = "SRLImplementation<0x283487840, instanceType: YTDA
    IContentPlaybackLifecycleController, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDAIContentPlaybackStatusAPI_API" = "SRLImplementation<0x283487a80, instanceType: YTDAICo
    ntentPlaybackStatusController, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDataFetcherService_API" = "SRLImplementation<0x2837f00c0, instanceType: YTDataFetcherSer
    vice, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDataOperationQueue_API" = "SRLImplementation<0x2837fc7e0, instanceType: YTInnerTubeOpera
    tionQueue, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDataServices_API" = "SRLImplementation<0x283694090, instanceType: YTRealDataServices, li
    fetime: 0, scopeType: 0, configuration: None>";
        "YTDefaultPlayerOverlayFactory_API" = "SRLImplementation<0x2834966d0, instanceType: YTDefau
    ltPlayerOverlayFactory, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDefaultSlotAdapterFactory_API" = "SRLImplementation<0x2834a6430, instanceType: YTDefault
    SlotAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDeferredAtrController_API" = "SRLImplementation<0x2836fd470, instanceType: YTDeferredAtr
    Controller, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDeferredAtrDataStore_API" = "SRLImplementation<0x2836fd650, instanceType: YTDeferredAtrD
    ataStore, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDelayedEventEntityStore_API" = "SRLImplementation<0x2837ef6f0, instanceType: YTDelayedEv
    entEntityStore, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDeviceClassificationURLDecorator_API" = "SRLImplementation<0x2837fc810, instanceType: YT
    DeviceClassificationURLDecorator, lifetime: 0, scopeType: 0, configuration: None>";
        "YTDiscoveryPlaybackTrackerLayoutEnteredOpportunityAdapter_API" = "SRLImplementation<0x2834
    9a3d0, instanceType: YTDiscoveryPlaybackTrackerLayoutEnteredOpportunityAdapter, lifetime: 0, sc
    opeType: 0, configuration: None>";
        "YTECatcherController_API" = "SRLImplementation<0x2837fc7b0, instanceType: YTECatcherContro
    ller, lifetime: 0, scopeType: 0, configuration: None>";
        "YTELMEntitiesProcessorRegistrar_API" = "SRLImplementation<0x283573420, instanceType: YTEL
    MEntitiesProcessorRegistrar, lifetime: 0, scopeType: 0, configuration: None>";
        "YTELMDFontLoader_API" = "SRLImplementation<0x283579080, instanceType: YTELMDFontLoader, life
    time: 0, scopeType: 0, configuration: None>";
        "YTELMSJSEnvironment_API" = "SRLImplementation<0x283710840, instanceType: YTELMSJSEnvironment
    , lifetime: 0, scopeType: 0, configuration: None>";
        "YTELMPerformanceAgentFactory_API" = "SRLImplementation<0x283710810, instanceType: YTELMPer
    formanceAgentFactory, lifetime: 0, scopeType: 0, configuration: None>";
        "YTELMQQueryEngineProvider_API" = "SRLImplementation<0x283573450, instanceType: YTELMQQueryEn
    gineProvider, lifetime: 0, scopeType: 0, configuration: None>";
        "YTELMRRegistry_API" = "SRLImplementation<0x2835728e0, instanceType: YTELMRRegistry, lifetime
    : 0, scopeType: 0, configuration: None>";
        "YTELMRResourceLoader_API" = "SRLImplementation<0x283491170, instanceType: YTELMRResourceLoad
    er, lifetime: 0, scopeType: 0, configuration: None>";
        "YTEditEffectAssetManager_API" = "SRLImplementation<0x28357f2a0, instanceType:YTEditEffectAs
    setManager, lifetime: 0, scopeType: 0, configuration: None>";
        "YTEditEffectAssetStore_API" = "SRLImplementation<0x28357f270, instanceType:YTEditEffectAs
    setStore, lifetime: 0, scopeType: 0, configuration: None>";
        "YTElementsCommandHandler_API" = "SRLImplementation<0x282b983c0, instanceType:YTElementsCo
    mmandHandler, lifetime: 0, scopeType: 1, configuration: None>";
        "YTElementsContextFactory_API" = "SRLImplementation<0x2835728b0, instanceType:YTELMContext
    Factory, lifetime: 0, scopeType: 0, configuration: None>";
        "YTElementsDispatcherFactory_API" = "SRLImplementation<0x283573240, instanceType:YTELMDisp
    atcherFactory, lifetime: 0, scopeType: 0, configuration: None>";
        "YTElementsLoggerFactory_API" = "SRLImplementation<0x283573270, instanceType:YTELMLoggerFa
    ctory, lifetime: 0, scopeType: 0, configuration: None>";

```

```

    "YTEElementsUpdateHandler_API" = "SRLImplementation<0x283710180, instanceType: YTElementUpdateHandler, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEngagementPanelControllerProvider_API" = "SRLImplementation<0x282bd0930, instanceType: YTEngagementPanelControllerProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEntitiesConfig_API" = "SRLImplementation<0x283718a20, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEntityTransformationController_API" = "SRLImplementation<0x28361ad30, instanceType: YTEntityTransformationController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEntityTransformationMissingEntityObserver_API" = "SRLImplementation<0x283603720, instanceType: YTEntityTransformationMissingEntityObserver, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEventLogger_API" = "SRLImplementation<0x2837cb1b0, instanceType: YTEventLoggingController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEventLoggingController_API" = "SRLImplementation<0x283713600, instanceType: YTEventLoggingController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEventLoggingDelayedEventController_API" = "SRLImplementation<0x2837ef5d0, instanceType: YTEventLoggingDelayedEventController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEventLoggingRetryController_API" = "SRLImplementation<0x2837fce70, instanceType: YTEventLoggingRetryController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEEventLoggingRetryDelayedEventController_API" = "SRLImplementation<0x283762d00, instanceType: YTEventLoggingRetryDelayedEventController, lifetime: 0, scopeType: 0, configuration: None>";
}

    "YTEEventLoggingService_API" = "SRLImplementation<0x2837fc1b0, instanceType: YTEventLoggingService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTEExternallyManagedSlotAdapterDelegate_API" = "SRLImplementation<0x283487c00, instanceType: YTAdsControlFlowManagerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTFeatureGatesClient_API" = "SRLImplementation<0x283763750, instanceType: YTFeatureGatesClient, lifetime: 0, scopeType: 4, configuration: None>";
    "YTFeedFilterChipBarControllerConfig_API" = "SRLImplementation<0x2836703c0, instanceType: YTFeedFilterChipBarController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTFeedbackCoordinator_API" = "SRLImplementation<0x28369f600, instanceType: YTFeedbackCoordinator, lifetime: 0, scopeType: 0, configuration: None>";
    "YTFetcherConfig_API" = "SRLImplementation<0x2837cb3f0, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTFlatFileLogsReader_API" = "SRLImplementation<0x2836aac40, instanceType: YTFlatFileLogsReader, lifetime: 0, scopeType: 0, configuration: None>";
    "YTFrequencyCapTracker_API" = "SRLImplementation<0x2834949f0, instanceType: YTFrequencyCapTracker, lifetime: 0, scopeType: 0, configuration: None>";
    "YT FulfillmentAdapterDelegate_API" = "SRLImplementation<0x2834a64c0, instanceType: YTAdsControlFlowManagerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YT FulfillmentAdapterFactoryManager_API" = "SRLImplementation<0x28349b630, instanceType: YT FulfillmentAdapterFactoryManager, lifetime: 0, scopeType: 0, configuration: None>";
    "YT FullscreenEngagementCompanionLayoutRenderingAdapterFactory_API" = "SRLImplementation<0x282b87c30, instanceType: YT FullscreenEngagementCompanionLayoutRenderingAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YT GELStreamzService_API" = "SRLImplementation<0x283682b20, instanceType: YT GELStreamzService, lifetime: 0, scopeType: 0, configuration: None>";
    "YT GLCardboardRenderer_API" = "SRLImplementation<0x283483870, instanceType: YT GLCardboardRenderer, lifetime: 0, scopeType: 0, configuration: None>";
    "YT GLMediaPlayerViewFactory_API" = "SRLImplementation<0x283483a20, instanceType: YT GLMediaPlayerViewFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YT GLSceneProtocol_API" = "SRLImplementation<0x283484a20, instanceType: YT GLScene, lifetime: 0, scopeType: 0, configuration: None>";
    "YT GaiaIdentityService_API" = "SRLImplementation<0x2837c8ea0, instanceType: YT GaiaIdentityService, lifetime: 0, scopeType: 0, configuration: None>";
    "YT GetLocationCommandHandlerClassObservable_API" = "SRLImplementation<0x282bb8ed0, instance

```

```

Type: YTGetLocationCommandHandlerClassObservable, lifetime: 0, scopeType: 0, configuration: None>;
    "YTGlobalConfig_API" = "SRLImplementation<0x28379b7b0, instanceType: YTGlobalConfig, lifetime: 0, scopeType: 0, configuration: None>";
    "YTGlobalConfigsApplier_API" = "SRLImplementation<0x2837fc780, instanceType: YTGlobalConfigApplier, lifetime: 0, scopeType: 0, configuration: None>";
    "YTGoogleConversionPingConfig_API" = "SRLImplementation<0x2836cc4e0, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTGuideServiceCoordinator_API" = "SRLImplementation<0x283692190, instanceType: YTAppGuideServiceCoordinator, lifetime: 0, scopeType: 0, configuration: None>";
    "YTGuideService_API" = "SRLImplementation<0x2836921c0, instanceType: YTGuideService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTHHeaderLogoController_API" = "SRLImplementation<0x2836c6490, instanceType: YTHHeaderLogoController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTHeartbeatService_API" = "SRLImplementation<0x2834adfe0, instanceType: YTHeartbeatService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTHintController_API" = "SRLImplementation<0x2836a4060, instanceType: YTHintController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTHintsConfig_API" = "SRLImplementation<0x2836a4090, instanceType: YTAppHintsConfig, lifetime: 0, scopeType: 0, configuration: None>";
    "YTHomeFeedGlobalMonitor_API" = "SRLImplementation<0x28360b3f0, instanceType: YTInlineMutedPlaybackStateController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTHomeOfflineController_API" = "SRLImplementation<0x28368cab0, instanceType: YTHomeOfflineController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTHotConfig_API" = "SRLImplementation<0x2837a5140, instanceType: YTHotConfig, lifetime: 0, scopeType: 0, configuration: None>";
    "YTIOSGuardSnapshotController_API" = "SRLImplementation<0x283704150, instanceType: YTIOSGuardSnapshotController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTIdentityController_API" = "SRLImplementation<0x2837f7c30, instanceType: YTIdentityController, lifetime: 0, scopeType: 4, configuration: None>";
    "YTIdentityEntityTransformer_API" = "SRLImplementation<0x28361a430, instanceType: YTIdentityEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>";
    "YTIdentityNotifier_API" = "SRLImplementation<0x2837c9890, instanceType: YTIdentityNotifier, lifetime: 0, scopeType: 4, configuration: None>";
    "YTIdentityProvider_API" = "SRLImplementation<0x2837c9830, instanceType: YTIdentityController, lifetime: 0, scopeType: 4, configuration: None>";
    "YTIdentityRestorer_API" = "SRLImplementation<0x2837c9590, instanceType: YTIdentityRestorer, lifetime: 0, scopeType: 4, configuration: None>";
    "YTIdentityRetriever_API" = "SRLImplementation<0x2837c9560, instanceType: YTIdentityRetriever, lifetime: 0, scopeType: 0, configuration: None>";
    "YTIdentityScopedPersistentEntityStoreProvider_API" = "SRLImplementation<0x2837130f0, instanceType: YTIdentityScopedPersistentEntityStoreProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "YTIdentityScopedPersistentEntityStore_API" = "SRLImplementation<0x283718a50, instanceType: YTIdentityScopedPersistentEntityStoreV2, lifetime: 0, scopeType: 1, configuration: None>";
    "YTIdentityStateLoggingController_API" = "SRLImplementation<0x2837656b0, instanceType: YTIdentityStateLoggingController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTIdentityStore_API" = "SRLImplementation<0x2837c9320, instanceType: YTIdentityStore, lifetime: 0, scopeType: 4, configuration: None>";
    "YTIdentityStrategy_API" = "SRLImplementation<0x2837c9110, instanceType: YTInlineIdentityStrategy, lifetime: 0, scopeType: 4, configuration: None>";
    "YTIdentityTransactionCoalescer_API" = "SRLImplementation<0x2837c9020, instanceType: YTIdentityTransactionCoalescer, lifetime: 0, scopeType: 4, configuration: None>";
    "YTIdentityTransactionFactory_API" = "SRLImplementation<0x2837c90e0, instanceType: YTIdentityTransactionFactory, lifetime: 0, scopeType: 4, configuration: None>";
    "YTImageFetcherService_API" = "SRLImplementation<0x2837cb300, instanceType: YTHotSwappableI

```

```

mageFetcherService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTImageOperationQueue_API" = "SRLImplementation<0x2836922e0, instanceType: YTImageOperatio
nQueue, lifetime: 0, scopeType: 0, configuration: None>";
    "YTImageSelectionStrategy_API" = "SRLImplementation<0x283692310, instanceType: YTBestMatchI
mageSelectionStrategy, lifetime: 0, scopeType: 0, configuration: None>";
    "YTImageServiceDataSource_API" = "SRLImplementation<0x283692370, instanceType: YTOffli
neLocalImageEntitySource, lifetime: 0, scopeType: 0, configuration: None>";
    "YTImageService_API" = "SRLImplementation<0x2836922b0, instanceType: YTImageService, lifeti
me: 0, scopeType: 0, configuration: None>";
    "YTImageStyle_API" = "SRLImplementation<0x2836bf180, instanceType: YTAppImageStyle, lifetim
e: 0, scopeType: 0, configuration: None>";
    "YTInMemoryEntityStoreObserverTransformationTrigger_API" = "SRLImplementation<0x2829f03f0,
instanceType: YTInMemoryEntityStoreObserverTransformationTrigger, lifetime: 0, scopeType: 0, co
nfiguration: None>";
    "YTInMemoryEntityStore_API" = "SRLImplementation<0x28376bf30, instanceType: YTInMemoryEntit
yStoreV2, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInPlayerDAIExternallyManagedSlotAdapter_API" = "SRLImplementation<0x283487ba0, instanceT
ype: YTInPlayerDAIExternallyManagedSlotAdapter, lifetime: 0, scopeType: 0, configuration: None>
";
    "YTInceptionControllerProtocol_API" = "SRLImplementation<0x2836eac70, instanceType: YTIncep
tionController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInfoCardControllerProviderProtocol_API" = "SRLImplementation<0x2834ace40, instanceType:
YTInfoCardControllerProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInlineMutedPlaybackStateController_API" = "SRLImplementation<0x28354bc60, instanceType:
YTInlineMutedPlaybackStateController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInlineMutedWatchEndpointMutationCommandHandler_API" = "SRLImplementation<0x282b80600, in
stanceType: YTInlineMutedWatchEndpointMutationCommandHandler, lifetime: 0, scopeType: 1, config
uration: None>";
    "YTInnerTubeConfig_API" = "SRLImplementation<0x2837cab20, instanceType: YTSettings, lifetim
e: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeContextFactory_API" = "SRLImplementation<0x2837fc210, instanceType: YTInnerTube
ContextFactory, lifetime: 0, scopeType: 4, configuration: None>";
    "YTInnerTubeControllerFactory_API" = "SRLImplementation<0x2836dc750, instanceType: YTAppInn
erTubeControllerFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeCoreDataContext_API" = "SRLImplementation<0x2836f47b0, instanceType: YTInnerTub
eCoreDataContext, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeFetcherService_API" = "SRLImplementation<0x2837cff90, instanceType: YTHotSwappa
bleFetcherServiceAndURLBuilder, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeOperationQueue_API" = "SRLImplementation<0x2837fc600, instanceType: YTInnerTube
OperationQueue, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeRequestFactory_API" = "SRLImplementation<0x2837fc1e0, instanceType: YTInnerTube
RequestFactory, lifetime: 0, scopeType: 4, configuration: None>";
    "YTInnerTubeRequestorFactory_API" = "SRLImplementation<0x2837fc4b0, instanceType: YTInnerTu
beRequestorFactory, lifetime: 0, scopeType: 4, configuration: None>";
    "YTInnerTubeResponseCache_API" = "SRLImplementation<0x2837fc6f0, instanceType: YTInnerTubeR
esponseCache, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeRetryTaskController_API" = "SRLImplementation<0x2836fd2f0, instanceType: YTInne
rTubeRetryTaskController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeSharedData_API" = "SRLImplementation<0x2837fc6c0, instanceType: YTSettings, lif
etime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeUIGlobals_API" = "SRLImplementation<0x2836c6a60, instanceType: YTLiveServices,
lifetime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeUIServiceMapper_API" = "SRLImplementation<0x2836dc780, instanceType: YTAppInner
TubeUIServiceMapper, lifetime: 0, scopeType: 0, configuration: None>";
    "YTInnerTubeURLBuilder_API" = "SRLImplementation<0x283765440, instanceType: YTHotSwappableF
etcherServiceAndURLBuilder, lifetime: 0, scopeType: 0, configuration: None>";

```

```

    "YTInteractionLoggingOnGELController_API" = "SRLImplementation<0x2836ad320, instanceType: YTInteractionLoggingOnGELController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLastActionController_API" = "SRLImplementation<0x2837657a0, instanceType: YTLastActionController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLastActionProvider_API" = "SRLImplementation<0x2837fcc90, instanceType: YTLastActionController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLatencyLogger_API" = "SRLImplementation<0x2836ff1b0, instanceType: YTLatencyLoggingController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLayoutFactory_API" = "SRLImplementation<0x28362a070, instanceType: YTLayoutFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLayoutRenderingAdapterFactoryManager_API" = "SRLImplementation<0x2834ae250, instanceType: YTLayoutRenderingAdapterFactoryManager, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLayoutSelfExitRequestedTriggerAdapter_API" = "SRLImplementation<0x282b88ab0, instanceType: YTLayoutSelfExitRequestedTriggerAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLayoutStyle_API" = "SRLImplementation<0x283767540, instanceType: YTAppLayoutStyle, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLiveMidrollCuepointOpportunityAdapter_API" = "SRLImplementation<0x283487990, instanceType: YTLiveMidrollCuepointOpportunityAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLiveStreamBreakTransitionTriggerAdapter_API" = "SRLImplementation<0x283487ae0, instanceType: YTLiveStreamBreakTransitionTriggerAdapter, lifetime: 0, scopeType: 0, configuration: None>";
}

    "YTLocationInfoMonitor_API" = "SRLImplementation<0x2837198c0, instanceType: YTLocationInfoMonitor, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLockScreenController_API" = "SRLImplementation<0x283487d20, instanceType: YTLockScreenController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLoggingCoreDataContext_API" = "SRLImplementation<0x2837ef720, instanceType: YTLoggingCoreDataContext, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLoggingDispatchQueue_API" = "SRLImplementation<0x2837fcf30, instanceType: YTLoggingDispatchQueue, lifetime: 0, scopeType: 0, configuration: None>";
    "YTLoggingIdentityProvider_API" = "SRLImplementation<0x2837ca7c0, instanceType: YTLoggingIdentityProvider, lifetime: 0, scopeType: 4, configuration: None>";
    "YTLoggingURLController_API" = "SRLImplementation<0x283650600, instanceType: YTLoggingURLController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTMRRegistry_API" = "SRLImplementation<0x2836cd440, instanceType: YTMRRegistry, lifetime: 0, scopeType: 0, configuration: None>";
    "YTMacroMarkersController_API" = "SRLImplementation<0x282b87ea0, instanceType: YTMacroMarkersController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTMacroMarkersTimedSyncController_API" = "SRLImplementation<0x282b87f90, instanceType: YTMacroMarkersTimedSyncController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTMainAppDiscoveryInPlayerRenderingAPI_API" = "SRLImplementation<0x282b829a0, instanceType: YTMainAppDiscoveryInPlayerRenderingAPI, lifetime: 0, scopeType: 0, configuration: None>";
    "YTMainAppEngagementPanelDataCollector_API" = "SRLImplementation<0x283681b30, instanceType: YTMainAppEngagementPanelDataCollector, lifetime: 0, scopeType: 0, configuration: None>";
    "YTMainAppPlayerOverlayDataCollector_API" = "SRLImplementation<0x282b86130, instanceType: YTMainAppPlayerOverlayDataCollector, lifetime: 0, scopeType: 0, configuration: None>";
    "YTMemoryWatchdog_API" = "SRLImplementation<0x28368a1c0, instanceType: YTMemoryWatchdog, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNavigationCommandHandler_API" = "SRLImplementation<0x2836d8ff0, instanceType: YTNavigationCommandHandler, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNetworkActiveController_API" = "SRLImplementation<0x283562820, instanceType: YTNetworkActiveControllerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNetworkBaselineLogger_API" = "SRLImplementation<0x2837fcbb0, instanceType: (null), lifetime: 1, scopeType: 0, configuration: None>";
    "YTNetworkEventLogger_API" = "SRLImplementation<0x2837fc960, instanceType: YTNetworkEventLogger, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNetworkHealthController_API" = "SRLImplementation<0x2836f4660, instanceType: (null), lif

```

```

etime: 1, scopeType: 0, configuration: None>";
    "YTNetworkHealthLogger_API" = "SRLImplementation<0x2837fc930, instanceType: (null), lifetime: 1, scopeType: 0, configuration: None>";
    "YTNetworkReceivedDelegate_API" = "SRLImplementation<0x2837fc690, instanceType: (null), lifetime: 1, scopeType: 0, configuration: None>";
    "YTNetworkRequestTracker_API" = "SRLImplementation<0x2837cb330, instanceType: YTNetworkRequestTrackerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNetworkRetryDelegate_API" = "SRLImplementation<0x2837fc420, instanceType: (null), lifetime: 1, scopeType: 0, configuration: None>";
    "YTNetworkStatusEntityTransformationTrigger_API" = "SRLImplementation<0x2836eab50, instanceType: YTNetworkStatusEntityTransformationTrigger, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNetworkUsageNotifier_API" = "SRLImplementation<0x28357eac0, instanceType: YTNetworkUsageNotifier, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNonCriticalStartupSchedulerConfig_API" = "SRLImplementation<0x28376fc60, instanceType: YTAppNonCriticalStartupSchedulerConfig, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNonCriticalStartupScheduler_API" = "SRLImplementation<0x28376fc00, instanceType: (null), lifetime: 1, scopeType: 0, configuration: None>";
    "YTNotificationAccountScopedInnerTubeContextDecorator_API" = "SRLImplementation<0x2836aa910, instanceType: YTNotificationAccountScopedInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>";
    "YTNotificationConfig_API" = "SRLImplementation<0x283682730, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNotificationInnerTubeContextDecorator_API" = "SRLImplementation<0x2836aa760, instanceType: YTNotificationInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>";
    "YTNotificationOSSettingEntityController_API" = "SRLImplementation<0x2834803c0, instanceType: YTNotificationOSSettingEntityController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNotificationRegistrationManager_API" = "SRLImplementation<0x283682760, instanceType: YTNotificationRegistrationManager, lifetime: 1, scopeType: 0, configuration: None>";
    "YTNotificationService_API" = "SRLImplementation<0x283682790, instanceType: YTNotificationService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNthThumbnailLoadInstrumentation_API" = "SRLImplementation<0x283684f60, instanceType: YTThumbnailLoadInstrumentation, lifetime: 0, scopeType: 0, configuration: None>";
    "YTNthThumbnailLoadTransmitter_API" = "SRLImplementation<0x28356e190, instanceType: YTAppStartupInstrumentationThumbnailLoadTransmitter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOcclusionMonitoringController_API" = "SRLImplementation<0x2836ead90, instanceType: YTInlineMutedPlaybackStateController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineAdController_API" = "SRLImplementation<0x28369f660, instanceType: YTOfflineAdController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineAdFrequencyCapProvider_API" = "SRLImplementation<0x283494c30, instanceType: YTOfflineAdFrequencyCoordinator, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineAdParser_API" = "SRLImplementation<0x28369f690, instanceType: YTOfflineAdParser, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineAdPlaylistController_API" = "SRLImplementation<0x283682580, instanceType: YTOfflineAdPlaylistController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineAdProvider_API" = "SRLImplementation<0x282bbb000, instanceType: YTOfflineAdPlaylistController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineAppStateEventCenter_API" = "SRLImplementation<0x2836fd320, instanceType: YTOfflineAppStateEventCenter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineCacheSynchronizer_API" = "SRLImplementation<0x2836f4870, instanceType: YTOfflineCacheSynchronizer, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineClientStateLogger_API" = "SRLImplementation<0x2836fed90, instanceType: YTOfflineClientStateLogger, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineClientStateProvider_API" = "SRLImplementation<0x2836fedc0, instanceType: YTOfflineClientStateProvider, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineConfig_API" = "SRLImplementation<0x283682280, instanceType: YTSettings, lifetime:

```

```

    0, scopeType: 0, configuration: None>";
    "YTOfflineContentController_API" = "SRLImplementation<0x28349d9e0, instanceType: YTOfflineContentController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineCoordinator_API" = "SRLImplementation<0x283685560, instanceType: YTOfflineCoordinator, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineDataService_API" = "SRLImplementation<0x2836821f0, instanceType: YTOfflineDataService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineEntityGenerationController_API" = "SRLImplementation<0x2836feb20, instanceType: YTOfflineEntityGenerationController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineImageController_API" = "SRLImplementation<0x2836822b0, instanceType: YTOfflineImageController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineModeProvider_API" = "SRLImplementation<0x28369f630, instanceType: YTOfflineCoordinator, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineNotificationController_API" = "SRLImplementation<0x28356ff30, instanceType: YTOfflineNotificationController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineOrchestrationAutoRefreshController_API" = "SRLImplementation<0x2836fed00, instanceType: YTOfflineOrchestrationAutoRefreshController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineOrchestrationController_API" = "SRLImplementation<0x2836fedf0, instanceType: YTOfflineOrchestrationController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflinePlaybackPositionSyncService_API" = "SRLImplementation<0x28349e460, instanceType: YTOfflinePlaybackPositionSyncService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflinePlaylistController_API" = "SRLImplementation<0x283682520, instanceType: YTOfflinePlaylistController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflinePlaylistMonitor_API" = "SRLImplementation<0x282b9b5d0, instanceType: YTNoOpOfflinePlaylistMonitor, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflinePlaylistStore_API" = "SRLImplementation<0x2836f48d0, instanceType: YTOfflinePlaylistStore, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineRefreshEntityQueryTable_API" = "SRLImplementation<0x2836fec10, instanceType: YTOfflineRefreshEntityQueryTable, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineServiceController_API" = "SRLImplementation<0x28369dec0, instanceType: YTAppOfflineServiceController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineSettingsServiceSwitch_API" = "SRLImplementation<0x2836825e0, instanceType: YTOfflineSettingsServiceSwitch, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineSettingsService_API" = "SRLImplementation<0x283682610, instanceType: YTOfflineSettingsService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineStoreAPI_API" = "SRLImplementation<0x282bbaf0, instanceType: YTOfflineStoreAPIImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineTracklistCaptionController_API" = "SRLImplementation<0x2836820d0, instanceType: YTOfflineTracklistCaptionController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineTransferStatusLogger_API" = "SRLImplementation<0x2836f3060, instanceType: YTOfflineTransferStatusLogger, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineVideoController_API" = "SRLImplementation<0x2836821c0, instanceType: YTOfflineVideoController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineVideoDownloadScheduler_API" = "SRLImplementation<0x2836f3330, instanceType: YTOfflineVideoDownloadScheduler, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineVideoDownloader_API" = "SRLImplementation<0x2836f3240, instanceType: YTOfflineVideoDownloader, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineVideoListController_API" = "SRLImplementation<0x283682550, instanceType: YTOfflineVideoListController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineVideoRepairWorker_API" = "SRLImplementation<0x2834809c0, instanceType: YTOfflineVideoRepairWorker, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineVideoStore_API" = "SRLImplementation<0x2836fd260, instanceType: YTOfflineVideoStore, lifetime: 0, scopeType: 0, configuration: None>";
    "YTOfflineWatchNextServiceAdapter_API" = "SRLImplementation<0x282badf50, instanceType: YTApplianceOfflineWatchNextServiceAdapter, lifetime: 0, scopeType: 0, configuration: None>";

```

```

    "YTOrganicWatchNextResponseOpportunityAdapter_API" = "SRLImplementation<0x2834876f0, instanceType: YTOrganicWatchNextResponseOpportunityAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPageStyleController_API" = "SRLImplementation<0x2837103f0, instanceType: YTPageStyleController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPersistentToInMemoryEntityStoreProjectionController_API" = "SRLImplementation<0x2836e9740, instanceType: YTPersistentToInMemoryEntityStoreProjectionController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPersonalizedSuggestionsCache_API" = "SRLImplementation<0x283698270, instanceType: YTPersonalizedSuggestionsCache, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPhotoUploadController_API" = "SRLImplementation<0x283692250, instanceType: YTPhotoUploadController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPingControllerConfig_API" = "SRLImplementation<0x2836fd6b0, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPingService_API" = "SRLImplementation<0x28368c0c0, instanceType: YTPingService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPingStore_API" = "SRLImplementation<0x28368c060, instanceType: YTPingStore, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPingingEndpointCommandHandler_API" = "SRLImplementation<0x283651a0, instanceType: YTPingingEndpointCommandHandler, lifetime: 0, scopeType: 1, configuration: None>";
    "YTPivotBarProvider_API" = "SRLImplementation<0x283614ed0, instanceType: YTAppViewController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayAsYouBrowseSettingController_API" = "SRLImplementation<0x2836eadc0, instanceType: YTPlayAsYouBrowseSettingController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlaybackCoordinator_API" = "SRLImplementation<0x283483060, instanceType: MDXPlaybackCoordinator, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlaybackLockScreenController_API" = "SRLImplementation<0x283487cf0, instanceType: YTPlaybackLockScreenController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlaybackRequestFactory_API" = "SRLImplementation<0x2834b0d80, instanceType: YTPlaybackRequestFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlaybackRouteButtonControllerProvider_API" = "SRLImplementation<0x2836c21c0, instanceType: MDXRealServices, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlaybackTrackingLayoutRenderingAdapterFactory_API" = "SRLImplementation<0x2834ae580, instanceType: YTPlaybackTrackingLayoutRenderingAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlaybackTrackingSlotFulfillmentAdapterFactory_API" = "SRLImplementation<0x2834a64f0, instanceType: YTPlaybackTrackingSlotFulfillmentAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerAccountScopedInnerTubeContextDecorator_API" = "SRLImplementation<0x2834adfb0, instanceType: YTPlayerAccountScopedInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>";
    "YTPlayerAdEventTriggerAdapter_API" = "SRLImplementation<0x283487930, instanceType: YTPlayerAdEventTriggerAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerAdsProvider_API" = "SRLImplementation<0x283487720, instanceType: YTRealAdsPlayerServices, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerBytesLayoutEnteredAndFullscreenOpportunityAdapter_API" = "SRLImplementation<0x28349b990, instanceType: YTPlayerBytesLayoutEnteredAndFullscreenOpportunityAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerBytesLayoutEnteredOpportunityAdapter_API" = "SRLImplementation<0x28349bb40, instanceType: YTPlayerBytesLayoutEnteredOpportunityAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerBytesLayoutRenderingAdapterFactory_API" = "SRLImplementation<0x282b88210, instanceType: YTPlayerBytesLayoutRenderingAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";

```

```

    "YTPlayerBytesSlotAdapterFactory_API" = "SRLImplementation<0x282bbb180, instanceType: YTPlayerBytesSlotAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerBytesSlotFulfillmentAdapterFactory_API" = "SRLImplementation<0x282bbaf10, instanceType: YTPlayerBytesSlotFulfillmentAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerBytesSlotRegistrationAPI_API" = "SRLImplementation<0x283487c90, instanceType: YTAdaptivePlaybackService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerInnerTubeContextDecorator_API" = "SRLImplementation<0x2834adef0, instanceType: YTPlayerInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>";
    "YTPlayerOfflineFallbackConfig_API" = "SRLImplementation<0x2834897d0, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerPrebufferController_API" = "SRLImplementation<0x2834b0bd0, instanceType: YTPlayerPrebufferController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerPrefetchService_API" = "SRLImplementation<0x283497360, instanceType: YTPlayerPrefetchService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerRequestFactory_API" = "SRLImplementation<0x2834b0de0, instanceType: YTPlayerRequestFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerScrubLoggingService_API" = "SRLImplementation<0x28348f960, instanceType: YTPlayerScrubLoggingService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerServiceConfig_API" = "SRLImplementation<0x2834887b0, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerService_API" = "SRLImplementation<0x2834b1fb0, instanceType: YTPlayerService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPlayerViewControllerConfig_API" = "SRLImplementation<0x283483030, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPreviousQuery_API" = "SRLImplementation<0x283698480, instanceType: YTPreviousQuery, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPriorityLoader_API" = "SRLImplementation<0x28376bcc0, instanceType: YTPriorityLoader, lifetime: 0, scopeType: 0, configuration: None>";
    "YTProfileCardController_API" = "SRLImplementation<0x282b80b40, instanceType: YTProfileCardController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPromosheetController_API" = "SRLImplementation<0x283690ff0, instanceType: YTPromosheetController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPushNotificationController_API" = "SRLImplementation<0x283682700, instanceType: YTPushNotificationController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTPushNotificationService_API" = "SRLImplementation<0x2837d3bd0, instanceType: YTPushNotificationService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTQueueConfig_API" = "SRLImplementation<0x282b99ad0, instanceType: YTDefaultQueueConfig, lifetime: 0, scopeType: 0, configuration: None>";
    "YTReachabilityController_API" = "SRLImplementation<0x2837fc240, instanceType: YTReachabilityController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTReelImageFetcher_API" = "SRLImplementation<0x2836f4030, instanceType: YTReelImageFetcher, lifetime: 0, scopeType: 0, configuration: None>";
    "YTRenderingConfig_API" = "SRLImplementation<0x2836c6fd0, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTResponderRoot_API" = "SRLImplementation<0x283762dc0, instanceType: YTAppViewController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTRestoreController_API" = "SRLImplementation<0x2836842d0, instanceType: YTRestoreController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTRetroactiveSignInController_API" = "SRLImplementation<0x28357e580, instanceType: YTRetroactiveSignInController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSRLAccountSynchronizer_API" = "SRLImplementation<0x283480270, instanceType: YTSRLAccountSynchronizer, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSSOConfig_API" = "SRLImplementation<0x2837c9e00, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTScrollFocusGlobalStateMonitor_API" = "SRLImplementation<0x28348c7b0, instanceType: YTInl

```

```

    ineMutedPlaybackStateController, lifetime: 0, scopeType: 0, configuration: None>;
    "YTScrollFocusGlobalStateProtocol_API" = "SRLImplementation<0x283608690, instanceType: YTIn
lineMutedPlaybackStateController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSeamlessSignIn_API" = "SRLImplementation<0x28360c210, instanceType: YTSeamlessSignIn, li
fetime: 0, scopeType: 0, configuration: None>";
    "YTSearchHistoryService_API" = "SRLImplementation<0x2836984e0, instanceType: YTSearchHistor
yService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSearchLatencyCache_API" = "SRLImplementation<0x283698630, instanceType: YTSearchLatencyC
ache, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSearchService_API" = "SRLImplementation<0x2834eab20, instanceType: YTAppSearchService, 1
ifetime: 0, scopeType: 0, configuration: None>";
    "YTSearchboxStatsTracker_API" = "SRLImplementation<0x28369de00, instanceType: YTSearchboxSt
atsTracker, lifetime: 0, scopeType: 0, configuration: None>";
    "YTServices_API" = "SRLImplementation<0x283694060, instanceType: YTLiveServices, lifetime:
0, scopeType: 0, configuration: None>";
    "YTSettings_API" = "SRLImplementation<0x28379bfc0, instanceType: YTSettings, lifetime: 0, s
copeType: 0, configuration: None>";
    "YTShareObserverController_API" = "SRLImplementation<0x282b997d0, instanceType: YTShareObse
verController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTShareServices_API" = "SRLImplementation<0x282bd4390, instanceType: YTRealShareServices,
lifetime: 0, scopeType: 0, configuration: None>";
    "YTShortsStartupCoordinator_API" = "SRLImplementation<0x2836f4180, instanceType: YTShortsSt
artupCoordinator, lifetime: 0, scopeType: 0, configuration: None>";
    "YTShowContentPillNotifier_API" = "SRLImplementation<0x2836c6f70, instanceType: YTShowConte
ntPillCommandHandlerClassObservable, lifetime: 0, scopeType: 0, configuration: None>";
    "YTShowTooltipCommandHandler_API" = "SRLImplementation<0x28291dbc0, instanceType: YTShowToo
ltipCommandHandler, lifetime: 0, scopeType: 1, configuration: None>";
    "YTShowTransientPlayerScrimOverlayCommandHandlerClassDelegation_API" = "SRLImplementation<0
x282b80870, instanceType: YTShowTransientPlayerScrimOverlayCommandHandlerClassDelegation, lifet
ime: 0, scopeType: 0, configuration: None>";
    "YTSignedOutLocalSearchHistory_API" = "SRLImplementation<0x283482c10, instanceType: YTSigne
dOutLocalSearchHistory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSingleVideoControllerFactory_API" = "SRLImplementation<0x2834b0c60, instanceType: YTSing
leVideoControllerFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSingleVideoOfflineLoggingController_API" = "SRLImplementation<0x2834a72d0, instanceType:
YTSingleVideoOfflineLoggingController__657306411_GeneratedFactoryImpl, lifetime: 1, scopeType:
0, configuration: None>";
    "YTSiriShortcutManager_API" = "SRLImplementation<0x2836fd800, instanceType: YTSiriShortcutM
anager, lifetime: 0, scopeType: 0, configuration: None>";
    "YT.SkipButtonClickedTriggerAdapter_API" = "SRLImplementation<0x282b88360, instanceType: YT
skipButtonClickedTriggerAdapter, lifetime: 0, scopeType: 0, configuration: None>";
    "YT.SlimStatusBarController_API" = "SRLImplementation<0x283698900, instanceType: YT.SlimStatu
sBarController, lifetime: 0, scopeType: 0, configuration: None>";
    "YT.SlotAdapterFactoryManager_API" = "SRLImplementation<0x2834a6460, instanceType: YT.SlotAda
pterFactoryManager, lifetime: 0, scopeType: 0, configuration: None>";
    "YT.SlotFactory_API" = "SRLImplementation<0x283629fe0, instanceType: YT.SlotFactory, lifetime
: 0, scopeType: 0, configuration: None>";
    "YT.SlotOpportunityAdapterDelegate_API" = "SRLImplementation<0x28349a250, instanceType: YTAd
APTERFlowManagerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YT.SmartDownloadsSettingsObserver_API" = "SRLImplementation<0x2836feac0, instanceType: YTSm
artDownloadsSettingsObserver, lifetime: 0, scopeType: 0, configuration: None>";
    "YT.StallDetector_API" = "SRLImplementation<0x283684090, instanceType: YT.StallDetector, life
time: 0, scopeType: 0, configuration: None>";
    "YT.StallMinidumpGenerator_API" = "SRLImplementation<0x2837f7090, instanceType: YT.StallMinid
umpGenerator, lifetime: 0, scopeType: 0, configuration: None>";
    "YT.StateTagNotifications_API" = "SRLImplementation<0x2837fc750, instanceType: YT.StateTagNot
ifications>";

```

```

ifications, lifetime: 0, scopeType: 0, configuration: None>;
    "YTStorageController_API" = "SRLImplementation<0x28369e160, instanceType: YTStorageController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTStoryboardControllerFactory_API" = "SRLImplementation<0x2834ac9f0, instanceType: YTStoryboardControllerFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSubscriptionService_API" = "SRLImplementation<0x2836da4c0, instanceType: YTSubscriptionService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSuggestConfig_API" = "SRLImplementation<0x283698390, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSuggestParser_API" = "SRLImplementation<0x283482100, instanceType: YTAppSuggestParser, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSuggestRequestFactory_API" = "SRLImplementation<0x283698420, instanceType: YTAppSuggestRequestFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSuggestService_API" = "SRLImplementation<0x2836983f0, instanceType: YTService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSystemHealthCapturerRegistry_API" = "SRLImplementation<0x2837cb180, instanceType: YTSystemHealthCapturerRegistry, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSystemHealthContext_API" = "SRLImplementation<0x2837cadf0, instanceType: YTSystemHealthContext, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSystemHealthController_API" = "SRLImplementation<0x2837ca730, instanceType: YTSystemHealthController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSystemHealthTransmitterRegistry_API" = "SRLImplementation<0x2837ca160, instanceType: YTSystemHealthTransmitterRegistry, lifetime: 0, scopeType: 0, configuration: None>";
    "YTSystemNotifications_API" = "SRLImplementation<0x2837cb090, instanceType: YTSystemNotifications, lifetime: 0, scopeType: 0, configuration: None>";
    "YTTerminationEventPingController_API" = "SRLImplementation<0x2836d9050, instanceType: YTTerminationEventPingController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTThemeEntityManager_API" = "SRLImplementation<0x283690cc0, instanceType: YTThemeEntityManager, lifetime: 0, scopeType: 0, configuration: None>";
    "YTThumbnailLatencyLogger_API" = "SRLImplementation<0x2836929d0, instanceType: YTCSIThumbnailLatencyLogger, lifetime: 0, scopeType: 0, configuration: None>";
    "YTToastController_API" = "SRLImplementation<0x282baf9c0, instanceType: YTMainAppToastController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTTooltipController_API" = "SRLImplementation<0x2836b2670, instanceType: YTTooltipController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTTopViewControllerProvider_API" = "SRLImplementation<0x2836db540, instanceType: YTAppViewController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTTransactionQueueController_API" = "SRLImplementation<0x28369e100, instanceType: YTTransactionQueueController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTTriggerAdapterDelegate_API" = "SRLImplementation<0x28349b9f0, instanceType: YTAdsControlFlowManagerImpl, lifetime: 0, scopeType: 0, configuration: None>";
    "YTTriggerAdapterFactory_API" = "SRLImplementation<0x283499b90, instanceType: YTTriggerAdapterFactory, lifetime: 0, scopeType: 0, configuration: None>";
    "YTTextStyle_API" = "SRLImplementation<0x283693a80, instanceType: YTDefaultTextStyle, lifetime: 0, scopeType: 0, configuration: None>";
    "YTUnicornTimelyBlockingController_API" = "SRLImplementation<0x2836aed90, instanceType: YTUnicornTimelyBlockingController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTUnpluggedResolveLocationService_API" = "SRLImplementation<0x283713cf0, instanceType: YTUnpluggedResolveLocationService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTUpdatedMetadataService_API" = "SRLImplementation<0x282b862e0, instanceType: YTUpdatedMetadataService, lifetime: 0, scopeType: 0, configuration: None>";
    "YTUpgradeController_API" = "SRLImplementation<0x28357f120, instanceType: YTUpgradeController, lifetime: 0, scopeType: 0, configuration: None>";
    "YTUploadAppSpecificRequirementsStore_API" = "SRLImplementation<0x28362be40, instanceType: YTUploadAppSpecificRequirementsStore, lifetime: 0, scopeType: 0, configuration: None>";
    "YTUploadConfig_API" = "SRLImplementation<0x28349e610, instanceType: YTSettings, lifetime:

```

```

0, scopeType: 0, configuration: None>;
    "YTUploadCoreDataContext_API" = "SRLImplementation<0x28349ea60, instanceType: YTUploadCoreDataContext, lifetime: 0, scopeType: 0, configuration: None>;
    "YTUploadEngine_API" = "SRLImplementation<0x283480ae0, instanceType: YTUploadEngine, lifetime: 0, scopeType: 0, configuration: None>;
    "YTUploadProgressProvider_API" = "SRLImplementation<0x283480300, instanceType: YTUploadProgressProvider, lifetime: 0, scopeType: 0, configuration: None>;
    "YTUploadRequirementsStore_API" = "SRLImplementation<0x28362b300, instanceType: YTUploadRequirementsStore, lifetime: 0, scopeType: 0, configuration: None>;
    "YTUploadTaskController_API" = "SRLImplementation<0x283480b10, instanceType: YTUploadTaskController, lifetime: 0, scopeType: 0, configuration: None>;
    "YTUploadTaskStore_API" = "SRLImplementation<0x28349eca0, instanceType: YTUploadTaskStore, lifetime: 0, scopeType: 0, configuration: None>;
    "YTUploadUserSpecificRequirementsStore_API" = "SRLImplementation<0x283484060, instanceType: YTUploadUserSpecificRequirementsStore, lifetime: 0, scopeType: 0, configuration: None>;
    "YTUserDefaultsKeysProvider_API" = "SRLImplementation<0x2837b0630, instanceType: YTYouTubeUserDefaultKeysProvider, lifetime: 0, scopeType: 0, configuration: None>;
    "YTUserDefaults_API" = "SRLImplementation<0x2837b05a0, instanceType: YTYouTubeUserDefaults, lifetime: 0, scopeType: 4, configuration: None>;
    "YTVASTParser_API" = "SRLImplementation<0x283681bf0, instanceType: YTVASTParser, lifetime: 0, scopeType: 0, configuration: None>;
    "YTVELoggingAPIProvider_API" = "SRLImplementation<0x282b81ad0, instanceType: YTVELoggingAPIProvider, lifetime: 0, scopeType: 0, configuration: None>;
    "YTVMAPParser_API" = "SRLImplementation<0x283681f20, instanceType: YTVMAPParser, lifetime: 0, scopeType: 0, configuration: None>;
    "YTSSControllerFactory_API" = "SRLImplementation<0x2834ab000, instanceType: YTSSControllerFactory, lifetime: 0, scopeType: 0, configuration: None>;
    "YTVideoPlayerOverlayConfig_API" = "SRLImplementation<0x2836a4150, instanceType: YTSettings, lifetime: 0, scopeType: 0, configuration: None>;
    "YTVideoQualitySwitchControllerFactory_API" = "SRLImplementation<0x283561770, instanceType: YTVideoQualitySwitchControllerFactory, lifetime: 0, scopeType: 0, configuration: None>;
    "YTVisitorDataRequestDecorator_API" = "SRLImplementation<0x2837fc990, instanceType: YTVisitorDataRequestDecorator, lifetime: 0, scopeType: 4, configuration: None>;
    "YTVisitorDataStore_API" = "SRLImplementation<0x2837fc900, instanceType: YTIdentityController, lifetime: 0, scopeType: 4, configuration: None>;
    "YTVoiceSearchCache_API" = "SRLImplementation<0x2834f60a0, instanceType: YTVoiceSearchCache, lifetime: 0, scopeType: 0, configuration: None>;
    "YTVoiceSearchService_API" = "SRLImplementation<0x28369de60, instanceType: YTVoiceSearchService, lifetime: 0, scopeType: 0, configuration: None>;
    "YTWatchBreakController_API" = "SRLImplementation<0x2836a1aa0, instanceType: YTWatchBreakController, lifetime: 0, scopeType: 0, configuration: None>;
    "YTWatchEventCenter_API" = "SRLImplementation<0x2836a4390, instanceType: YTWatchEventCenter, lifetime: 0, scopeType: 0, configuration: None>;
    "YTWatchHistoryService_API" = "SRLImplementation<0x2834e97d0, instanceType: YTWatchHistoryService, lifetime: 0, scopeType: 1, configuration: None>;
    "YTWatchHistoryStatusNotifier_API" = "SRLImplementation<0x28349e100, instanceType: YTWatchHistoryStatusNotifier, lifetime: 0, scopeType: 0, configuration: None>;
    "YTWatchNextResponseFetcherAPI_API" = "SRLImplementation<0x282bbafa0, instanceType: YTAdBreakRendererFetcher, lifetime: 0, scopeType: 0, configuration: None>;
    "YTWatchNextService_API" = "SRLImplementation<0x282b9b090, instanceType: YTAppWatchNextService, lifetime: 0, scopeType: 0, configuration: None>;
    "YTWorkerController_API" = "SRLImplementation<0x28368a220, instanceType: YTWorkerController, lifetime: 0, scopeType: 0, configuration: None>;
    "YTWorkerScheduler_API" = "SRLImplementation<0x28357e880, instanceType: YTWorkerScheduler, lifetime: 0, scopeType: 0, configuration: None>;
    "YTYouTubeUserDefaults_API" = "SRLImplementation<0x2837f7f00, instanceType: YTYouTubeUserDefaults

```

```

        faults, lifetime: 0, scopeType: 4, configuration: None>";
    }

Multi bindings
{
    "ELMMissingResourceHandler_API" = "{(\n    SRLImplementation<0x2834e94a0, instanceType: YTAccountScopedAdsInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>,\n    SRLImplementation<0x2834e9710, instanceType: YTDataPushInnerTubeContextDecorator, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x2834e92c0, instanceType: YTELMAccountScopedInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>,\n    SRLImplementation<0x2834e82a0, instanceType: MLAccountScopedInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>,\n    SRLImplementation<0x2834e9e30, instanceType: YTAccountScopedClientStoreInfoInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>,\n    SRLImplementation<0x2834e9da0, instanceType: YTNotificationAccountScopedInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>,\n    SRLImplementation<0x2834e9890, instanceType: YTLocationAccountScopedInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>,\n    SRLImplementation<0x2834e9c80, instanceType: YTPlayerAccountScopedInnerTubeContextDecorator, lifetime: 0, scopeType: 1, configuration: None>\n)}";
    "YTAppBrowseServiceRequestDecorator_API" = "{(\n    SRLImplementation<0x283713030, instanceType: YTDownloadsBrowseParamsBrowseRequestDecorator, lifetime: 0, scopeType: 0, configuration: None>\n)}";
    "YTAppBrowseServiceResponseProvider_API" = "{(\n    SRLImplementation<0x2836b2640, instanceType: YTAppBrowseServiceLibraryResponseProvider, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x283689f80, instanceType: YTAppBrowseServicePlaylistResponseProvider, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x2836b0780, instanceType: YTAppBrowseServiceHomeResponseProvider, lifetime: 0, scopeType: 0, configuration: None>\n)}";
    "YTBackgroundAppRefreshTask_API" = "{(\n    SRLImplementation<0x2836cadf0, instanceType: YTAppBackgroundAppRefreshTask, lifetime: 0, scopeType: 0, configuration: None\n)}";
    "YTBackgroundProcessingTask_API" = "{(\n    SRLImplementation<0x2836cae50, instanceType: YTAppBackgroundProcessingTask, lifetime: 0, scopeType: 0, configuration: None\n)}";
    "YTElementsCommandExtensionRegistryProvider_API" = "{(\n    SRLImplementation<0x283572bb0, instanceType: YTCommentElementsCommandRegistryProvider, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x283572b80, instanceType: YTAdsElementsCommandExtensionRegistryProvider, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x283572b50, instanceType: YTAccountElementsCommandExtensionRegistryProvider, lifetime: 0, scopeType: 0, configuration: None\n)}";
    "YTElementsNodeClassRegistryProvider_API" = "{(\n    SRLImplementation<0x283572dc0, instanceType: YTAdsElementsNodeClassRegistryProvider, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x283572df0, instanceType: YTCommentElementsNodeClassRegistryProvider, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x283572e20, instanceType: YTGuessElementsNodeClassRegistryProvider, lifetime: 0, scopeType: 0, configuration: None\n)}";
    "YTEntityMergerProvider_API" = "{(\n    SRLImplementation<0x283718ed0, instanceType: YTMarkersEngagementPanelSyncEntityMergerProvider, lifetime: 0, scopeType: 0, configuration: None\n)}";
    "YTEntityStoreQueryTable_API" = "{(\n    SRLImplementation<0x283718c00, instanceType: YTTransferEntityQueryTable, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x283718ba0, instanceType: YTOfflineActionWrapperEntityQueryTable, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x283718c30, instanceType: YTOfflineRefreshEntityQueryTable, lifetime: 0, scopeType: 0, configuration: None\n)}";
    "YTEntityTransformationTriggerProtocol_API" = "{(\n    SRLImplementation<0x28361d020, instanceType: YTNetworkStatusEntityTransformationTrigger, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x28361d050, instanceType: YTPersistentEntityStoreObserverTransformationTrigger, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x28361

```

```

d0b0, instanceType: YTInMemoryEntityStoreObserverTransformationTrigger, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x28361ea60, instanceType: YTClockEntityTransformationTrigger, lifetime: 0, scopeType: 0, configuration: None>\n    }\";
    "YTEntityTransformerProtocol_API" = "{(\n      SRLImplementation<0x28361a250, instanceType: YTMainDownloadedVideoEntityToDownloadStatusEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x28361adc0, instanceType: YTMainDownloadsListEntityToSmartDownloadsHeaderEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x28361a580, instanceType: YTMainDownloadedVideoEntityToDownloadsPageVideoEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x28361ae50, instanceType: YTMainRecommendedDownloadVideoEntityToDownloadsPageRecommendedVideoEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x2836ee130, instanceType: YTMainVideoEntityToDownloadStatusEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x2836eb8a0, instanceType: YTMainVideoEntityToDownloadsPageVideoEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x28361a220, instanceType: YTMainDownloadedVideoEntityToDownloadedVideoWithContextEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x28361ab50, instanceType: YTMainDownloadedPlaylistEntityToDownloadStatusEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x283619e90, instanceType: YTMainDownloadedPlaylistEntityToDownloadsPagePlaylistEntityTransformer, lifetime: 0, scopeType: 0, configuration: None>\n    )\";
    "YTInnerTubeCellControllerRegistrationProvider_API" = "{(\n      SRLImplementation<0x2836036f0, instanceType: YTReelInnerTubeCellControllerRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>\n    )\";
    "YTInnerTubeCellRegistrationProvider_API" = "{(\n      SRLImplementation<0x28361f150, instanceType: YTReelInnerTubeCellRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>\n    )\";
    "YTInnerTubeCommandRegistrationProvider_API" = "{(\n      SRLImplementation<0x283681b90, instanceType: YTFlowCommandRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x283681bc0, instanceType: YTReelInnerTubeCommandRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>\n    )\";
    "YTInnerTubeContextDecorator_API" = "{(\n      SRLImplementation<0x283714960, instanceType: YTELMIInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>,\n      SRLImplementation<0x283714690, instanceType: YTClientStoreInfoInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>,\n      SRLImplementation<0x283714a50, instanceType: MLInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>,\n      SRLImplementation<0x283714930, instanceType: YTLocationInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>,\n      SRLImplementation<0x2837148d0, instanceType: YTDataPushInnerTubeContextDecorator, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x283714a80, instanceType: YTNotificationInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>,\n      SRLImplementation<0x283714b10, instanceType: YTPlayerInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>,\n      SRLImplementation<0x283714630, instanceType: YTAdsInnerTubeContextDecorator, lifetime: 0, scopeType: 4, configuration: None>\n    )\";
    "YTInnerTubeControllerRegistrationProvider_API" = "{(\n      SRLImplementation<0x2836dcea0, instanceType: YTReelShortsInnerTubeControllerRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x2836dce70, instanceType: YTARCameraInnerTubeControllerRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>\n    )\";
    "YTInnerTubeSectionControllerMatchingRegistrationProvider_API" = "{(\n      SRLImplementation<0x2836c0120, instanceType: YTReelInnerTubeSectionControllerMatchingRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>\n    )\";
    "YTInnerTubeTopControllerRegistrationProvider_API" = "{(\n      SRLImplementation<0x2836dc7b0, instanceType: YTEditInnerTubeTopControllerRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x2836dc7e0, instanceType: YTLiveCreationInnerTubeTopControllerRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>,\n      SRLImplementation<0x2836dc810, instanceType: YTReelInnerTubeTopControllerRegistrationProvider, lifetime: 0, scopeType: 0, configuration: None>\n    )\";
    "YTInnerTubeViewControllerMatchingRegistrationProvider_API" = "{(\n    )\";

```

```
"YTMMainWindowObserver_API" = "({\n    SRLImplementation<0x28369f570, instanceType: YTLMWindowFrameObserver, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x28369f5a0, instanceType: YTLastActionController, lifetime: 0, scopeType: 0, configuration: None>\n})\";  
"YTModuleInitializer_API" = "({\n    SRLImplementation<0x2837cf120, instanceType: YTInnerTubeModuleInitializer, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x2837cf0c0, instanceType: YTAccountModuleInitializer, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x2837cf0f0, instanceType: YTFamilyManagementModuleInitializer, lifetime: 0, scopeType: 0, configuration: None>\n})\";  
"YTPlaybackContextDecorator_API" = "({\n    SRLImplementation<0x2834b0e10, instanceType: XYTPlaybackContextDecorator, lifetime: 0, scopeType: 0, configuration: None>\n})\";  
"YTUploadFlowFactory_API" = "({\n    SRLImplementation<0x28349e9d0, instanceType: YTUploadFeedbackOnlyFlowFactory, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x28349e6a0, instanceType: YTUploadShortsFlowFactory, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x28349e5e0, instanceType: YTUploadNormalFlowFactory, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x28349e910, instanceType: YTUploadArchiveReplacementFlowFactory, lifetime: 0, scopeType: 0, configuration: None>,\n    SRLImplementation<0x28349eac0, instanceType: YTUploadReelsFlowFactory, lifetime: 0, scopeType: 0, configuration: None>\n})\";  
}  
1
```

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:  
2022-10-27 11:02:46

## 返回数据

某个广告视频的内部返回数据：

hook代码：

```
- (void)setVideoDetails:(id)arg1{
    iosLogInfo(@"arg1=%@", arg1);
    %orig;
}
```

输出：

```
2022-03-28 20:36:41.954044+0800 YouTube[25775:2715494] hook_ youtubeDylib.xm YTWatchMiniBarView
Controller$setVideoDetails$: arg1= YTIVideoDetails 0x281b27f40 : {
    video_id: "LA6szWzus0g"
    title: "Get Chrome for iOS"
    length_seconds: 30
    channel_id: "UCL8ZULXASCc1I_oaOT0NaOQ"
    is_owner_viewing: false
    short_description: "Chrome is a fast, easy to use, and secure web browser for iOS. Get Chrome now. https://apps.apple.com/us/app/google-chrome/id535886823"
    is_crawlable: false
    thumbnail {
        thumbnails {
            url: "https://i.ytimg.com/vi/LA6szWzus0g/hqdefault.jpg?sqp=-oaymwEYCHgQWkhG8quKqQMMCAEV
AACIQnABwAEG&rs=AOn4CLBU26fMso0e3iR2comi4Uz-x5cj7Q"
            width: 120
            height: 90
        }
        thumbnails {
            url: "https://i.ytimg.com/vi/LA6szWzus0g/hqdefault.jpg?sqp=-oaymwEcCIACEJABSEbyq4qpAw4I
ARUAAIhCGAFwAcABBg==&rs=AOn4CLA2zPsvsB2aQxkZvfPFXMVhwdkpHQ"
            width: 256
            height: 144
        }
        thumbnails {
            url: "https://i.ytimg.com/vi/LA6szWzus0g/hqdefault.jpg?sqp=-oaymwEcCMACEQBSEbyq4qpAw4I
ARUAAIhCGAFwAcABBg==&rs=AOn4CLD0aI6fXD2ptZTQmDP1myZBfcg13Q"
            width: 320
            height: 180
        }
        thumbnails {
            url: "https://i.ytimg.com/vi_webp/LA6szWzus0g/hqdefault.webp"
            width: 480
            height: 360
        }
        thumbnails {
            url: "https://i.ytimg.com/vi_webp/LA6szWzus0g/sddefault.webp"
            width: 640
            height: 480
        }
    }
}
```

```
thumbnails {
    url: "https://i.ytimg.com/vi/LA6szWzus0g/hq720.jpg?sqp=-oaymwEcCK4FEIIDSEbyq4qpAw4IARUA
AIhCGAFwAcABBg==&rs=AOn4CLDvGWQtBt1ONoEcK933xt4Aypr_7w"
    width: 686
    height: 386
}
}
allow_ratings: true
view_count: "26531222"
author: "Google Chrome"
is_private: false
is_unplugged_corpus: false
is_live_content: false
}
```

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:  
2022-10-27 11:11:46

# YouTube广告过滤相关

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:  
2022-10-27 11:04:57

## hook的类

一些hook的核心的类

### youtubeReqResp.xm

- youtubeDylib/Logos/youtubeReqResp.xm

```
/*
File: youtubeReqResp.xm
Function: Youtube hook request and response related
*/

#import <UIKit/UIKit.h>
#import <objc/runtime.h>
#import "CrifanLibiOS.h"

#import "youtubeCommon.h"

/*-----*
 NSURLRequest
-----*/
hook NSURLRequest

+ (NSURLRequest *)requestWithURL:(NSURL *)URL{
    NSURLRequest *rwu = [orig;

    if (isYoutubeAdsVideo_current(URL)){
        iosLogInfo("isYoutubeAdsVideo_current: %@ -> rwu=%@", URL, rwu);
        gNoUse = 1;
    }

    return rwu;
}

- (NSURLRequest *)initWithURL:(NSURL *)URL{
    NSURLRequest *iwu = [orig;

    if (isYoutubeAdsVideo_current(URL)){
        iosLogInfo("isYoutubeAdsVideo_current: %@ -> iwu=%@", URL, iwu);
        gNoUse = 1;
    }

    return iwu;
}

+ (NSURLRequest *)requestWithURL:(NSURL *)URL cachePolicy:(NSURLRequestCachePolicy)cachePolicy timeoutInterval:(NSTimeInterval)timeoutInterval{
    NSURLRequest *rwu_uct = [orig;
```

```

iosLogInfo("URL=%@,cachePolicy=%lu,timeoutInterval=%f -> rwu_uct=%@", URL, (unsigned long)cachePolicy, timeoutInterval, rwu_uct);
if (is1SecondVideo(URL)){
    iosLogInfo("URL=%@", URL);
    gNoUse = 1;
}

if (isYoutubeAdsVideo_current(URL)){
    gNoUse = 1;
}

return rwu_uct;
}

- (NSURLRequest *)initWithURL:(NSURL *)URL cachePolicy:(NSURLRequestCachePolicy)cachePolicy timeoutInterval:(NSTimeInterval)timeoutInterval{
    NSURLRequest *iwu_uct = %orig;

    if (isYoutubeAdsVideo_current(URL)){
        iosLogInfo("isYoutubeAdsVideo_current: URL=%@,cachePolicy=%lu,timeoutInterval=%f -> iwu_uct=%@", URL, (unsigned long)cachePolicy, timeoutInterval, iwu_uct);
        gNoUse = 1;
    }

    return iwu_uct;
}
}

end

/*-----
hook NSURLConnection
-----*/

```

```

:hook NSURLConnection

- (void)setHTTPBody:(NSData *)newHttpBodyData {
    NSURL *reqUrl = [self URL];
    if (isYoutubeAdsVideo_current(reqUrl)){
        iosLogInfo("isYoutubeAdsVideo_current: reqUrl=%@", reqUrl);
        gNoUse = 1;
    }

    %orig;
}

end

/*-----
hook NSHTTPURLResponse
-----*/

```

```

:hook NSHTTPURLResponse

- (NSHTTPURLResponse *)initWithURL:(NSURL *)url
    statusCode:(NSInteger)statusCode

```

```

        HTTPVersion (NSString *)HTTPVersion
        headerFields (NSDictionary NSString *, NSString *> *)headerFields(
    NSHTTPURLResponse* newUrlResp = %orig;

    if (is1SecondVideo(url)){
        iosLogInfo("url=%@", url);
        gNoUse = 1;
    }

    if (isYoutubeAdsVideo_current(url)){
        iosLogInfo("isYoutubeAdsVideo_current: url=%@,statusCode=%d,HTTPVersion=%@,headerField
s=%@ -> newUrlResp=%@", url, statusCode, HTTPVersion, headerFields, newUrlResp);
        gNoUse = 1;

        if (isYoutubeAdsVideo_ctierA(url)){
            iosLogInfo("isYoutubeAdsVideo_ctierA: url=%@,statusCode=%d,HTTPVersion=%@,headerFi
elds=%@ -> newUrlResp=%@", url, statusCode, HTTPVersion, headerFields, newUrlResp);
            gNoUse = 1;
        }
    } else {
        // TODO: after test, change back to ads video filter
        iosLogInfo("url=%@", url);
        gNoUse = 1;
    }

    return newUrlResp;
}

(NSDictionary *)allHeaderFields{
    NSURL* curUrl = [self URL];
    if (isYoutubeAdsVideo_current(curUrl)){
//        iosLogInfo("isYoutubeAdsVideo: self=%@", self);
        gNoUse = 1;
    }

    NSDictionary allHeader = %orig;
    iosLogInfo("curUrl=%@ : allHeader=%@", curUrl, allHeader);
    return allHeader;
}

-(NSInteger)statusCode{
    NSURL* curUrl = [self URL];
    if (isYoutubeAdsVideo_current(curUrl)){
//        iosLogInfo("isYoutubeAdsVideo: self=%@", self);
        gNoUse = 1;
    }

    NSInteger respStatusCode = %orig;
    iosLogInfo("respStatusCode=%ld", respStatusCode);
    return respStatusCode;
}

@end

/*
-----
```

```

HAMDataLoadRequest
-----*/
hook HAMDataLoadRequest

- (id)initWithRangedURL:(id)rangeUrl timeoutInterval:(double)timeoutInterval HTTPMethod:(NSString *)HTTPMethod HTTPContentType:(NSString *)HTTPContentType HTTPBody:(id)HTTPBody idempotent:(BOOL)idempotent{
    iosLogInfo("rangeUrl=%@,timeoutInterval=%f,HTTPMethod=%@,HTTPContentType=%@,HTTPBody=%@,idempotent=%s", rangeUrl, timeoutInterval, HTTPMethod, HTTPContentType, HTTPBody, boolToStr(idempotent));
    gNoUse = 1;
    id newDataLoadReq = %orig;
    return newDataLoadReq;
}

- (id)initWithRangedURL:(id)rangeUrl{
    iosLogInfo("rangeUrl=%@", rangeUrl);
    gNoUse = 1;
    id newDataLoadReq = %orig;
    return newDataLoadReq;
}

- (id)initWithURL:(id)url{
    iosLogInfo("url=%@", url);
    gNoUse = 1;
    id newDataLoadReq = %orig;
    return newDataLoadReq;
}

end

/*-----
MLOnesieRequestFactory
-----*/
hook MLOnesieRequestFactory

- (id)onesieRequestForPlayerRequest:(id)request dataLoader:(id)dataLoader context:(id)context cryptor:(id)cryptor requestNumber:(long long)requestNumber error:(id *)error{
    iosLogInfo("request=%@,dataLoader=%@,context=%@,cryptor=%@,requestNumber=%lld,error=%p", request, dataLoader, context, cryptor, requestNumber, error);
    gNoUse = 1;
    id newReqFactory = %orig;
    iosLogInfo("-> newReqFactory=%@", newReqFactory);
    return newReqFactory;
}

- (void)onesieRequestForPlayerRequest:(id)request authorization:(id)authorization dataLoader:(id)dataLoader context:(id)context cryptor:(id)cryptor requestNumber:(long long)requestNumber completionHandler:(id)completionHandler{
    iosLogInfo("request=%@,authorization=%@,dataLoader=%@,context=%@,cryptor=%@,requestNumber=%lld,completionHandler=%@", request, authorization, dataLoader, context, cryptor, requestNumber, completionHandler);
    gNoUse = 1;
    %orig;
}

```

```

}

end

/*-----
YTIIInnerTubeContext
-----*/
hook YTIIInnerTubeContext

- (id)data{
    id ctxData = %orig;
    iosLogInfo("ctxData=%@", ctxData);
    return ctxData;
}

+ (id)descriptor{
    gNoUse = 1;
    id curDesc = %orig;
    iosLogInfo("curDesc=%@", curDesc);
    return curDesc;
}

-(YTIClientInfo *)remoteClient{
    gNoUse = 1;
    YTIClientInfo* newClientInfo = %orig;
    iosLogInfo("newClientInfo=%@", newClientInfo);
    return newClientInfo;
}

end

/*-----
YTPlayerService
-----*/
hook YTPlayerService

- (id)makePlaybackRequest (id)playbackRequest responseBlock (id)responseBlock errorBlock (id)errorBlock{
    iosLogInfo("playbackRequest=%@,responseBlock=%@,errorBlock=%@", playbackRequest, responseBlock, errorBlock);
    gNoUse = 1;
    id newPlayService = %orig;
    iosLogInfo("-> newPlayService=%@", newPlayService);
    return newPlayService;
}

end

/*-----
YTPlayerRequestFactory
-----*/
hook YTPlayerRequestFactory

```

```

- (id)requestForPlayerWithPlayerRequest (id)playerRequest URLRequestProperties (id)URLRequestProperties{
    iosLogInfo("playerRequest=%@,URLRequestProperties=%@", playerRequest, URLRequestProperties);

    gNoUse = 1;
    id newReqFactory = %orig;
    iosLogInfo("-> newReqFactory=%@", newReqFactory);
    return newReqFactory;
}

- (id)innerTubeRequestForPlayerWithVideoID:(id)videoID playlistID:(id)playlistID playlistIndex:(unsigned long long)playlistIndex playbackContext:(id)playbackContext forOffline:(_Bool)forOffline clickTrackingParams:(id)clickTrackingParams playerParams:(id)playerParams{
    iosLogInfo("videoID=%@,playlistID=%@,playlistIndex=%lld,playbackContext=%@,forOffline=%s,clickTrackingParams=%@,playerParams=%@", videoID, playlistID, playlistIndex, playbackContext, boolToStr(forOffline), clickTrackingParams, playerParams);
    gNoUse = 1;
    id newPlayerReqFactory = %orig;
    iosLogInfo("-> newPlayerReqFactory=%@", newPlayerReqFactory);
    return newPlayerReqFactory;
}

End

/*
 *-----*
 * YTPPlaybackDataLoader
 *-----*/
hook YTPPlaybackDataLoader

- (id)onlinePromise{
    iosLogInfo("%s", "");
    gNoUse = 1;
    id newPromise = %orig;
    iosLogInfo("-> newPromise=%@", newPromise);
    return newPromise;
}

- (id)createPlaybackRequest{
    iosLogInfo("%s", "");
    gNoUse = 1;
    id newReq = %orig;
    iosLogInfo("-> newReq=%@", newReq);
    return newReq;
}

End

/*
 *-----*
 * YTPPlaybackRequestFactory
 *-----*/
hook YTPPlaybackRequestFactory

- (id)playbackRequestForPlayerTransition:(id)playerTransition visibility:(int)visibility viewpo

```

```

rtSizeProvider (id)viewportSizeProvider mediaStickySettings (id)mediaStickySettings prefetchCon
text (id)prefetchContext{
    iosLogInfo("playerTransition=%@,visibility=%d,viewportSizeProvider=%@,mediaStickySettings=%
@,prefetchContext=%@", playerTransition, visibility, viewportSizeProvider, mediaStickySettings,
prefetchContext);
    gNoUse = 1;
    id newPlaybackReqFactory = %orig;
    iosLogInfo("-> newPlaybackReqFactory=%@", newPlaybackReqFactory);
    return newPlaybackReqFactory;
}

End

/*
GPBMessage
*/
hook GPBMessage

- (id)data{
    gNoUse = 1;
    id neGpbwData = %orig;
    iosLogInfo("-> neGpbwData=%@", neGpbwData);
    return neGpbwData;
}

End

/*
YTAccountScopedInnerTubeRequestFactory
*/
hook YTAccountScopedInnerTubeRequestFactory

- (id)requestForProtoRequest (id)protoRequest withService (long long)serviceNumber needsClickTr
ackingParams: (_Bool)needsClickTrackingParams{
    iosLogInfo("protoRequest=%@,serviceNumber=%lld,needsClickTrackingParams=%s", protoRequest,
serviceNumber, boolToStr(needsClickTrackingParams));
    if (ServiceNumber_AdBreak == serviceNumber){
        iosLogInfo("is ServiceNumber_AdBreak: %lld", serviceNumber);
        gNoUse = 1;
    }

    id newRequest = %orig;
    return newRequest;
}

End

/*
YTPlayerBytesSlotFulfillmentAdapter
*/
hook YTPlayerBytesSlotFulfillmentAdapter

```

```

- (void)fillSlot{
    iosLogInfo("%s", "");
    %orig;
}

End

/*
YTAdBreakMetadataProvider
-----*/
hook YTAdBreakMetadataProvider

+ (id)dataFromMetadata:(id)metadata{
    id data = %orig;
    iosLogInfo("metadata=%@ -> data=%@", metadata, data);
    return data;
}

End

/*
YTAdClientMetadataAbstractProvider
-----*/
hook YTAdClientMetadataAbstractProvider

+ (id)tagName{
    id curTagName = %orig;
    iosLogInfo("curTagName=%@", curTagName);
    return curTagName;
}

End

/*
YTAdClientMetadata
-----*/
hook YTAdClientMetadata

- (id)metadataProviderForProviderName:(id)providerName{
    id metadataProvider = %orig;
    iosLogInfo("providerName=%@ -> metadataProvider=%@", providerName, metadataProvider);
    return metadataProvider;
}

End

```



# IDA伪代码

一些核心的类的IDA伪代码

## HAMCronetDataLoadTask

- [HAMCronetDataLoadTask startWithDelegate:delegateQueue:]

- HAMCronetDataLoadTask\_startWithDelegate\_delegateQueue.coffee

```
void __cdecl -[HAMCronetDataLoadTask startWithDelegate delegateQueue ](HAMCronetDataLoadTask * s
elf, SEL a2, id inputDelegate, id inputDispatchQueue)
{
    struct objc_object * dispatchQueue; // x21
    struct objc_object * curDelegate; // x20
    HAMCronetDataLoadTask * curHAMCronetDataLoadTask; // x27
    struct objc_object * dispatchQueue_; // x23
    double absoluteTime; // d0
    __int64 CronetDataLoadTask; // x19
    NSURLRequest * request; // x28
    double timeoutInterval; // d8
    NSDictionary * userInfo; // x26
    struct objc_object * clock; // x24
    Cronet_Engine * engine; // ST30_8
    id request_; // x20
    id clock_; // x22
    HAMCronetDataLoadTask * curHAMCronetDataLoadTask_; // x25
    OS_dispatch_queue * v18; // x27
    __int64 v19; // x0
    Cronet.UrlRequestCallback * cronetUrlRequestCallback; // x21
    id clock_; // x0
    CronetDataLoadTask * CronetDataLoadTask_; // x22
    double clockAbsoluteTime; // d0
    struct objc_object * v24; // x0
    HAMDataLoadTaskObserver * MLHAMDataLoadTaskObserverImpl; // x0
    struct objc_object * v26; // x8
    Cronet_Executor * Cronet_ExecutorPtr; // x0
    void * Cronet.UrlRequestParamsPtr; // x19
    struct objc_object * v29; // x0
    NSString * httpMethod; // x28
    NSString * requestMethod_1; // x0
    char * requestMethodStr; // x0
    struct objc_object * v33; // x0
    id allHTTPHeaderFields; // x21
    struct objc_object * v35; // x0
    id HTTPBody; // x21
    Cronet.UploadDataProvider * Cronet.UploadDataProviderPtr; // x0
    struct objc_object * v38; // x0
    id v39; // x0
    struct objc_object * requestUrl; // x24
    struct objc_object * v41; // x0
    struct objc_object * requestUrlNSString; // x26
```

```

Cronet_Executor v43; // x5
int Cronet_RESULT_InitWithParams; // w26
OS_dispatch_queue v45; // x0
void (*fastcall *blockInvokeFunc)(__int64); // x8
struct objc_object inputDelegate_1; // x19
struct objc_object userInfo_1; // x23
struct objc_object dispatchQueue_2; // x24
int Cronet_RESULT_Start; // w0
id dispatchQueue_1; // [xsp+8h] [xbp-F8h]
id userInfo_; // [xsp+10h] [xbp-F0h]
id inputDelegate_; // [xsp+18h] [xbp-E8h]
BOOL idempotent; // [xsp+2Ch] [xbp-D4h]
struct objc_object object; // [xsp+38h] [xbp-C8h]
void **stackBlock; // [xsp+40h] [xbp-C0h]
InternalMetadata v57; // [xsp+48h] [xbp-B8h]
HasBits<1> v58[2]; // [xsp+50h] [xbp-B0h]
void *v59; // [xsp+58h] [xbp-A8h]
CronetDataLoadTask v60; // [xsp+60h] [xbp-A0h]
int v61; // [xsp+68h] [xbp-98h]
void *globalBlock; // [xsp+70h] [xbp-90h]
__int64 blockFlags; // [xsp+78h] [xbp-88h]
void (*fastcall blockInvoke)(__int64, struct objc_object *, struct objc_object *); // [xsp+
80h] [xbp-80h]
void *blockDescriptor; // [xsp+88h] [xbp-78h]
void *v66; // [xsp+90h] [xbp-70h]

dispatchQueue = inputDispatchQueue;
curDelegate = inputDelegate;
curHAMCronetDataLoadTask = self;
inputDelegate_ = objc_retain(inputDelegate);
dispatchQueue_ = objc_retain(dispatchQueue);
objc_msgSend_3E84298((void *)curHAMCronetDataLoadTask->_clock, "absoluteTime");
curHAMCronetDataLoadTask->_startTime = absoluteTime;
CronetDataLoadTask = (__int64)curHAMCronetDataLoadTask->_task;
objc_storeStrong((id *)curHAMCronetDataLoadTask->_task.delegate_, curDelegate);
request = curHAMCronetDataLoadTask->_request;
timeoutInterval = curHAMCronetDataLoadTask->_timeoutInterval;
idempotent = curHAMCronetDataLoadTask->_idempotent;
userInfo = curHAMCronetDataLoadTask->_userInfo;
clock = (struct objc_object *)curHAMCronetDataLoadTask->_clock;
engine = curHAMCronetDataLoadTask->_engine;
request_ = objc_retain((id)curHAMCronetDataLoadTask->_request);
userInfo_ = objc_retain((id)userInfo);
clock_ = objc_retain(clock);
curHAMCronetDataLoadTask_ = (HAMCronetDataLoadTask *)objc_retain((id)curHAMCronetDataLoadTask)

dispatchQueue_1 = objc_retain(dispatchQueue_);
objc_storeStrong((id *)curHAMCronetDataLoadTask->_task.parentTask_, (id)curHAMCronetDataLo
dTask);
objc_storeStrong((id *)curHAMCronetDataLoadTask->_task.URLRequest_, (id)request);
curHAMCronetDataLoadTask->_task.timeoutInterval_ = timeoutInterval;
objc_storeStrong((id *)curHAMCronetDataLoadTask->_task.userInfo_, (id)userInfo);
objc_storeStrong((id *)curHAMCronetDataLoadTask->_task.clock_, clock_);
v18 = &curHAMCronetDataLoadTask->_task.queue_;
objc_storeStrong((id *)curHAMCronetDataLoadTask->_task.queue_, dispatchQueue);
curHAMCronetDataLoadTask->_task.request_ = (Cronet_UrlRequest *)sub_3DF6D08(v19);

```

```

cronetUrlRequestCallback = (Cronet.UrlRequestCallback *)mallocAndSetResponseCallbacks_3DF91D8(
    (__int64)Cronet.UrlRequestCallback_OnRedirectReceived_194C9D8,
    (__int64)Cronet.UrlRequestCallback_OnResponseStarted_194CCE4,
    (__int64)Cronet.UrlRequestCallback_OnReadCompleted_194CFE8,
    (__int64)Cronet.UrlRequestCallback_OnSucceeded_194D248,
    (__int64)Cronet.UrlRequestCallback_OnFailed_194D310,
    (__int64)Cronet.UrlRequestCallback_OnCanceled_194D5C0);
setOffset8Value_3DF9198((__int64)cronetUrlRequestCallback, CronetDataLoadTask);
curHAMCronetDataLoadTask_>_task.callback_ = cronetUrlRequestCallback;
object = clock_;
clock__ = clock_;
CronetDataLoadTask_ = (CronetDataLoadTask *)CronetDataLoadTask;
objc_msgSend_3E84298(clock__, "absoluteTime");
curHAMCronetDataLoadTask_>_task.absoluteTime_ = clockAbsoluteTime;
v24 = (struct objc_object *)InjectOptionalHAMDataLoadTaskObserver();
MLHAMDataLoadTaskObserverImpl = (HAMDataLoadTaskObserver *)objc_retainAutoreleasedReturnValue
_3E842EC(v24);
v26 = (struct objc_object *)curHAMCronetDataLoadTask_>_task.observer_;
curHAMCronetDataLoadTask_>_task.observer_ = MLHAMDataLoadTaskObserverImpl;
objc_release_3E842BC(v26);
objc_msgSend_3E84298(
    (void *)curHAMCronetDataLoadTask_>_task.observer_,
    "dataLoadTaskDidStart:",
    curHAMCronetDataLoadTask_);
Cronet_ExecutorPtr = (Cronet.Executor *)Cronet.Executor_CreateWith_3DF9004(Cronet.Executor_ExecuteFunc_194C27C);
curHAMCronetDataLoadTask_>_task.executor_ = Cronet_ExecutorPtr;
setOffset8Value_3DF8FF8((__int64)Cronet_ExecutorPtr, CronetDataLoadTask);
Cronet.UrlRequestParamsPtr = (void *)Cronet.UrlRequestParams_Create_3DFA39C();
v29 = (struct objc_object *)objc_msgSend_3E84298(request_, "HTTPMethod");
httpMethod = (NSString *)objc_retainAutoreleasedReturnValue_3E842EC(v29);
if ( httpMethod )
{
    requestMethod_1 = (NSString *)objc_retainAutorelease_3E842D4();
    requestMethodStr = (char *)objc_msgSend_3E84298(requestMethod_1, "UTF8String");
    Cronet.UrlRequestParams_http_method_set_3DFA3D0((__int64)Cronet.UrlRequestParamsPtr, (__int
64)requestMethodStr);
}
if ( idempotent )
    Cronet.UrlRequestParams_idempotency_set_3DFA45C((__int64)Cronet.UrlRequestParamsPtr, 1);//
1=Cronet.UrlRequestParams_IDEMPOTENCY_IDEMPOTENT
v33 = (struct objc_object *)objc_msgSend_3E84298(request_, "allHTTPHeaderFields");
allHTTPHeaderFields = objc_retainAutoreleasedReturnValue_3E842EC(v33);
globalBlock = _NSConcreteGlobalBlock;
blockFlags = 0xD0800000LL;
blockInvoke = cornetAddHeader_194C2FC;
blockDescriptor = (void **)unk_4EF9AF8;
v66 = Cronet.UrlRequestParamsPtr;
objc_msgSend_3E84298(allHTTPHeaderFields, "enumerateKeysAndObjectsUsingBlock:", &globalBlock);

```

```

objc_release_3E842BC(allHTTPHeaderFields);
v35 = (struct objc_object *)objc_msgSend_3E84298(request_, "HTTPBody");
HTTPBody = objc_retainAutoreleasedReturnValue_3E842EC(v35);
if (objc_msgSend_3E84298(HTTPBody, "length") )
{
    Cronet_UploadDataProviderPtr = (Cronet_UploadDataProvider *)Cronet_UploadDataProvider_CreateWith_3DF92E0(
        (__int64)Cronet_UploadDataProvider_GetLengthFunc_194C3C4,
        (__int64)Cronet_UploadDataProvider_ReadFunc_194C418,
        (__int64)Cronet_UploadDataProvider_RewindFunc_194C4F4,
        (__int64)Cronet_UploadDataProvider_CloseFunc_194C51C);
    curHAMCronetDataLoadTask_>_task.uploadDataProvider_ = Cronet_UploadDataProviderPtr;
    setOffset8Value_3DF92A0((__int64)Cronet_UploadDataProviderPtr, (__int64)CronetDataLoadTask_);
}
Cronet_UrlRequestParams_upload_data_provider_set_3DFA3EC(
    __int64)Cronet_UrlRequestParamsPtr,
    (__int64)curHAMCronetDataLoadTask_>_task.uploadDataProvider_);
Cronet_UrlRequestParams_upload_data_provider_executor_set_3DFA3F4(
    __int64)Cronet_UrlRequestParamsPtr,
    (__int64)curHAMCronetDataLoadTask_>_task.executor_);
}
v38 = (struct objc_object *)objc_msgSend_3E84298(request_, "URL");
v39 = objc_retainAutoreleasedReturnValue_3E842EC(v38);
requestUrl = v39;
v41 = (struct objc_object *)objc_msgSend_3E84298(v39, "absoluteString");
objc_retainAutoreleasedReturnValue_3E842EC(v41);
requestUrlNSString = (struct objc_object *)objc_retainAutorelease_3E842D4();
objc_msgSend_3E84298(requestUrlNSString, "cStringUsingEncoding:", 4LL);
objc_release_3E842BC(requestUrlNSString);
objc_release_3E842BC(requestUrl);
v43 = curHAMCronetDataLoadTask_>_task.executor_;
Cronet_RESULT_InitWithParams = Cronet_UrlRequest_InitWithParams_3DF931C((__int64)curHAMCronetDataLoadTask_>_task.request_);
Cronet_UrlRequestParams_Destroy_3DFA3B8(Cronet_UrlRequestParamsPtr);
if (Cronet_RESULT_InitWithParams)
{
    v45 = v18;
    stackBlock = _NSConcreteStackBlock;
    blockInvokeFunc = InitWithParamsErrorCallback_194C520;
    userInfo_1 = userInfo_;
    inputDelegate_1 = inputDelegate_;
    dispatchQueue_2 = dispatchQueue_1;
LABEL_11:
    v57.var0 = 0xC0000000LL;
    *(_QWORD *)v58[0].var0 = blockInvokeFunc;
    v59 = unk_4EF9B18;
    v60 = CronetDataLoadTask_;
    v61 = Cronet_RESULT_InitWithParams;
    dispatch_async(v45, &stackBlock);
    goto LABEL_12;
}

```

```
tryResetTimeoutTimer_194C704(CronetDataLoadTask_);
Cronet_RESULT_Start = callCronetUrlRequestStart_3DF9338((__int64)curHAMCronetDataLoadTask_ >_
task.request_);
userInfo_1 = userInfo_;
inputDelegate_1 = inputDelegate_;
dispatchQueue_2 = dispatchQueue_1;
if ( Cronet_RESULT_Start )
{
    Cronet_RESULT_InitWithParams = Cronet_RESULT_Start;
    v45 = v18;
    stackBlock = _NSConcreteStackBlock;
    blockInvokeFunc = startErrorCallback_194C7FC;
    goto LABEL_11;
}
LABEL_12:
objc_release_3E842BC(HTTPBody);
objc_release_3E842BC((id)httpMethod);
objc_release_3E842BC(dispatchQueue_2);
objc_release_3E842BC((id)curHAMCronetDataLoadTask_);
objc_release_3E842BC(object);
objc_release_3E842BC(userInfo_1);
objc_release_3E842BC(request_);
objc_release_3E842BC(dispatchQueue_2);
objc_release_3E842BC(inputDelegate_1);
}
```

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:  
2022-10-27 11:04:29

## 函数调用逻辑

广告过滤内部调用逻辑，涉及到多个函数多个参数：

### disable\_ads\_filter\_logic

`disable_ads_filter_logic.coffee`

禁用广告过滤时，多次调试

目前的发现的逻辑是：

【1】即便是 oad 的 initplayback，正常返回response，也会返回错误

(1) case 1:

请求：

```
NSURLRequest requestWithURL cachePolicy timeoutInterval
    URL https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&oeis=1&c=IOS
&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1&osnz
=1&msp=1&odeak=1&odepv=1&osfc=1&id=be5e83ddc65db9b&initcwndbps=1127500&mt=1657589100&ack=1&cpn
=j3M40XDCGhcG1Jgj&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
    cachePolicy 0
    timeoutInterval 60.000000
```

响应：

```
NSHTTPURLResponse initWithURL statusCode HTTPVersion headerFields:
    url https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&oeis=1&c=IOS
&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1&osnz
=1&msp=1&odeak=1&odepv=1&osfc=1&id=be5e83ddc65db9b&initcwndbps=1127500&mt=1657589100&ack=1&cpn
=j3M40XDCGhcG1Jgj&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
    statusCode 200
    HTTPVersion (null)
    headerFields {
        "alt-svc" = "h3=:443"; ma=2592000, h3-29=:443; ma=2592000, h3-Q050=:443; ma=2592000, h3-Q046=:443; ma=2592000, h3-Q043=:443; ma=2592000, quic=:443; ma=2592000; v="46,43"";
        "cache-control" = "public, max-age=86400";
        "client-protocol" = quic;
        "content-type" = "application/vnd.yt-ump";
        "cross-origin-resource-policy" = "cross-origin";
        date = "Tue, 12 Jul 2022 01:52:01 GMT";
        expires = "Wed, 13 Jul 2022 01:52:01 GMT";
        server = "gvs 1.0";
        vary = Origin;
        "x-content-type-options" = nosniff;
    }
```

但还是会报错：

```

NSError errorWithDomain code userInfo
    errDomain com.google.ios.hamplayer
    code 6000
    userInfo{
        HAMErrorDetails = "cronet=10|internal=-356|quic=85";
        HAMErrorURLRequest = "<NSMutableURLRequest: 0x282833ae0"
    {
        URL: https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&oeis=1&c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1&osnz=1&msp=1&odeak=1&odepv=1&osfc=1&id=be5e83dddc65db9b&initcwndbps=1127500&mt=1657589100&ack=1&cpn=j3M40XDCGhcG1Jgj&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
    }";
        HAMRequestNumberLoadTaskKey = 1;
    }
}

```

以及：

```

NSError ml_mediaErrorWithCode underlyingError:
    arg1 708
    underlyingError
        Error Domain com.google.ios.hamplayer
        Code 6000 "(null)"
        UserInfo{
            HAMErrorDetails cronet 10|internal -356 quic 85
            HAMRequestNumberLoadTaskKey 1
            HAMErrorURLRequest = <NSMutableURLRequest: 0x282833ae0>
    {
        URL: https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&oeis=1&c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1&osnz=1&msp=1&odeak=1&odepv=1&osfc=1&id=be5e83dddc65db9b&initcwndbps=1127500&mt=1657589100&ack=1&cpn=j3M40XDCGhcG1Jgj&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140 } } -> mewc=Error Domain=YTMediaError Code=708 "(null)" UserInfo={NSUnderlyingError=0x283bda940 {Error Domain=com.google.ios.hamplayer Code=6000 "(null)" UserInfo={HAMErrorDetails=cronet=10|internal -356|quic=85, HAMRequestNumberLoadTaskKey=1, HAMErrorURLRequest=<NSMutableURLRequest: 0x282833ae0> { URL: https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&oeis=1&c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1&osnz=1&msp=1&odeak=1&odepv=1&osfc=1&id=be5e83dddc65db9b&initcwndbps=1127500&mt=1657589100&ack=1&cpn=j3M40XDCGhcG1Jgj&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140 }}}
    }
}

```

(2) case 2:

```

NSURLRequest requestWithURL cachePolicy timeoutInterval:
    URL https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1&osnz=1&msp=1&odeak=1&odepv=1&osfc=1&id=38d955056e41c6f3&initcwndbps=1195000&mt=1657590782&ack=1&cpn=SKSZKIy71ZZ2pxor&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
    cachePolicy 0
    timeoutInterval 60.000000

```

响应：

```

NSHTTPURLResponse initWithURL statusCode HTTPVersion headerFields:
    url https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1&osnz

```

```

=1&msp=1&odeak=1&odepv=1&osfc=1&id=38d955056e41c6f3&initcwndbps=1195000&mt=1657590782&ack=1&cpn
=SKSZKIy71ZZ2pxor&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
  statusCode 200
  HTTPVersion (null)
  headerFields {
    "alt-svc" = "h3=:443"; ma=2592000,h3-29=:443"; ma=2592000,h3-Q050=:443"; ma=2592000,h3-Q046=:443"; ma=2592000,h3-Q043=:443"; ma=2592000,quic=:443"; ma=2592000; v="46,43""";
    "cache-control" = "public, max-age=86400";
    "client-protocol" = quic;
    "content-type" = "application/vnd.yt-ump";
    "cross-origin-resource-policy" = "cross-origin";
    date = "Tue, 12 Jul 2022 02:00:07 GMT";
    expires = "Wed, 13 Jul 2022 02:00:07 GMT";
    server = "gvs 1.0";
    vary = Origin;
    "x-content-type-options" = nosniff;
  }
}

```

也还是会报错：

```

NSError errorWithDomain code userInfo
  errDomain com.google.ios.hamplayer
  code 6000
  userInfo {
    HAMErrorDetails = "cronet=10|internal=-356|quic=85";
    HAMErrorURLRequest = " NSMutableURLRequest 0x2828fc1b0 ";
    URL: https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&oeis=1
    &c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=
    1&osnz=1&msp=1&odeak=1&odepv=1&osfc=1&id=38d955056e41c6f3&initcwndbps=1195000&mt=1657590782&ack
    =1&cpn=SKSZKIy71ZZ2pxor&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
  ";
    HAMRequestNumberLoadTaskKey = 1;
  }
}

```

以及：

```

NSError HAMErrorWithCode userInfo
  reqUrl=https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&oeis=1&c=
  IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=
  1&osnz=1&msp=1&odeak=1&odepv=1&osfc=1&id=38d955056e41c6f3&initcwndbps=1195000&mt=1657590782&ack=1&
  cpn=SKSZKIy71ZZ2pxor&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
}

```

以及：

```

NSError ml_mediaErrorWithCode underlyingError:
  arg1 708
  underlyingError
    Error Domain com.google.ios.hamplayer
    Code 6000 "(null)"
    UserInfo {
      HAMErrorDetails cronet 10|internal=-356|quic 85
      HAMRequestNumberLoadTaskKey 1
      HAMErrorURLRequest = NSMutableURLRequest 0x2828fc1b0 ;
      URL: https://rr2---sn-phvo-jpqe.googlevideo.com/initplayback?source=youtube&orc=1&oeis=
      1&c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr
    }
}

```

```
=1&osnz=1&msp=1&odeak=1&odepv=1&osfc=1&id=38d955056e41c6f3&initcwndbps=1195000&mt=1657590782&ac
k=1&cfn=SKSZKIy71ZZ2pxor&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
}
}
```

【2】即使是 ctier L 的 videoplayback，在前几次请求 (rn 1、rn 2 等) 时，正常返回 response，也会报错

(1) rn 1 的 ctier L

`NSURLRequest requestWithURL cachePolicy timeoutInterval:`

```
URL 

也还是会报错：


```

`NSError errorWithDomain code userInfo`

```
reqUrl 

以及：


```

`NSError HAMErrorWithCode userInfo`

```
reqUrl 

\(2\) rn 2 的 ctier L


```

正常响应，但也报错：

`NSHTTPURLResponse initWithURL statusCode HTTPVersion headerFields:`

```
url 67
```

```

mv1,pl,initcwndbps&lsig=AG3C_xAwRAIgTJgONnp0_iio87Dj5f9Dc1H8TzxX_zPCs1clwp4poqACIDPhiXjWBAuAs9G
AhBSD11lyHobe7EfWyHSV4Eoixwhh&cfn=aFN2DnMpoKgorDiB&rn=2
  statusCode 200
  HTTPVersion (null)
  headerFields {
    "alt-svc" = "h3=:443"; ma=2592000,h3-29=:443"; ma=2592000,h3-Q050=:443"; ma=2592000,h3-Q046=:443"; ma=2592000,h3-Q043=:443"; ma=2592000,quic=:443"; ma=2592000; v="46,43""";
    "cache-control" = "private, max-age=21041";
    "client-protocol" = quic;
    "content-type" = "application/vnd.yt-ump";
    "cross-origin-resource-policy" = "cross-origin";
    date = "Tue, 12 Jul 2022 02:14:24 GMT";
    expires = "Tue, 12 Jul 2022 02:14:24 GMT";
    server = "gvs 1.0";
    vary = Origin;
    "x-content-type-options" = nosniff;
  }
}

```

但也会报错：

```
NSError errorWithDomain code userInfo
```

【3】当 rn 2 或 rn 3 时的 ctier=L 就正常了，不报错了，能正常返回数据了

(1) rn 3 的 ctier=L

```

NSURLRequest requestWithURL cachePolicy timeoutInterval:
  URL https://rr2---sn-phvo-jpq1.googlevideo.com/videoplayback?expire=1657613405&ei=_dfMYqc2IuG
ajQTGv7PYAw&ip=162.253.45.206&id=o-AIjeweHrLqp_Pz60zv20T5JprNQBR0HmN6o8txqvBehU&source=youtube&
requiressl=yes&mh=GE&mm=31,29&mn=sn-phvo-jpq1,sn-vgqsrn1k&ms=au,rdu&mv=m&mvi=2&p=22&ctier=L&in
itcwndbps=1213750&svpuc=1&sabr=1&mt=1657591502&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IO
S&spars=expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=AOq0QJ8wRAIgVsPJepnU04V9FcfXV3
OqVZtNkVp5xDaw-jII0VaPqpUCID1hXsWyAMG-_A41aRCm2HF5eqtfDTt5IJs_ATne2Jfz&lsparams=mh,mm,mn,ms,mv,
mv1,pl,initcwndbps&lsig=AG3C_xAwRAIgTJgONnp0_iio87Dj5f9Dc1H8TzxX_zPCs1clwp4poqACIDPhiXjWBAuAs9G
AhBSD11lyHobe7EfWyHSV4Eoixwhh&cfn=aFN2DnMpoKgorDiB&rn=3
  cachePolicy 0
  timeoutInterval 60.000000

```

响应：

```

NSHTTPURLResponse initWithURL statusCode HTTPVersion headerFields
  url https://rr2---sn-phvo-jpq1.googlevideo.com/videoplayback?expire=1657613405&ei=_dfMYqc2IuG
ajQTGv7PYAw&ip=162.253.45.206&id=o-AIjeweHrLqp_Pz60zv20T5JprNQBR0HmN6o8txqvBehU&source=youtube&
requiressl=yes&mh=GE&mm=31,29&mn=sn-phvo-jpq1,sn-vgqsrn1k&ms=au,rdu&mv=m&mvi=2&p=22&ctier=L&in
itcwndbps=1213750&svpuc=1&sabr=1&mt=1657591502&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IO
S&spars=expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=AOq0QJ8wRAIgVsPJepnU04V9FcfXV3
OqVZtNkVp5xDaw-jII0VaPqpUCID1hXsWyAMG-_A41aRCm2HF5eqtfDTt5IJs_ATne2Jfz&lsparams=mh,mm,mn,ms,mv,
mv1,pl,initcwndbps&lsig=AG3C_xAwRAIgTJgONnp0_iio87Dj5f9Dc1H8TzxX_zPCs1clwp4poqACIDPhiXjWBAuAs9G
AhBSD11lyHobe7EfWyHSV4Eoixwhh&cfn=aFN2DnMpoKgorDiB&rn=3
  statusCode 200
  HTTPVersion (null)
  headerFields {
    "alt-svc" = "h3=:443"; ma=2592000,h3-29=:443"; ma=2592000,h3-Q050=:443"; ma=2592000,h3-Q046=:443"; ma=2592000,h3-Q043=:443"; ma=2592000,quic=:443"; ma=2592000; v="46,43""";
  }
}

```

```

    "cache-control" = "private, max-age=20802";
    "client-protocol" = quic;
    "content-type" = "application/vnd.yt-ump";
    "cross-origin-resource-policy" = "cross-origin";
    date = "Tue, 12 Jul 2022 02:18:23 GMT";
    expires = "Tue, 12 Jul 2022 02:18:23 GMT";
    server = "gvs 1.0";
    vary = Origin;
    "x-content-type-options" = nosniff;
}

```

(2) rn 2 的 ctier=L, 且能正常返回视频数据chunks

```

MLServerABRLoader dataLoadTask:didReceiveResponse
    arg1 <MLServerABRDataTask: 0x285a0bb70>
    didReceiveResponse = NSHTTPURLResponse 0x280dae440
    {
        URL: https://rr1---sn-phvo-jpql.googlevideo.com/videoplayback?expire=1657615602&ei=kuDMYr
        OabpHDigT5y4LYDw&ip=162.253.45.206&id=o-AMkZ1BzQ6p7kCJW94A6DtyMGG3HfqwKg1KnZ-fNzwp2U&source=youtube&requiressl=yes&mh=Mk&mm=31,29&mn=sn-phvo-jpql,sn-vgqsknsk&ms=au,rdu&mv=m&mvi=1&pcm2cms=yes&p1=22&ctier=L&initcwndbps=1266250&svpuc=1&sabr=1&mt=1657593666&fvip=4&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=A0q0QJ8wRQIgAqEYZ1R2jQz3cDKbLMWJKMhZa_JyTS1vtd_9064WLDkCIQdfL32Gx3XA9C4XUvNdi1XvM8TQjC1SjAv2N0B2bt-f1w%3D%3D&lsparams=mh,mm,mn,ms,mv,mvi,pcm2cms,p1,initcwndbps&lsig=AG3C_xAwRgIhAJ8F9p0M1D7yBXsonsAKJ95tAp
        ty4njmPKzuAFqoM81WA1EA1Fcqzn3LXLCxMMe_WxKUJ0d4L9uLrZ7oP_BtbspIFBY%3D&cpn=tuURUWDAKHN5FNX8&rn=2
    }
    {
        Status Code: 200
        Headers {
            "Cache-Control" = (
                "private, max-age=20896"
            );
            "Content-Type" = (
                "application/vnd.yt-ump"
            );
            Date = (
                "Tue, 12 Jul 2022 02:53:26 GMT"
            );
            Expires = (
                "Tue, 12 Jul 2022 02:53:26 GMT"
            );
            Server = (
                "gvs 1.0"
            );
            Vary = (
                Origin
            );
            "alt-svc" = (
                "h3=\":443\"; ma=2592000, h3-29=\":443\"; ma=2592000, h3-Q050=\":443\"; ma=2592000, h3-Q046=\":443\"; ma=2592000, h3-Q043=\":443\"; ma=2592000, quic=\":443\"; ma=2592000; v=\"46,43\""
            );
            "client-protocol" = (
                quic
            );
            "cross-origin-resource-policy" = (
                "cross-origin"
            );
        }
    }
}

```

```

    );
    "x-content-type-options" =     (
        nosniff
    );
}
}

MLServerABRLoader dataLoadTask:didReceiveData:
    arg1 <MLServerABRDataTask: 0x285a0bb70>
    didReceiveData {length = 65536, bytes = 0x235908d0 fa0610d0 fa0618e0 d4033a4b ... 4286a906 e5
0ad07b }

```

不过后续此处 `rn 2` 的 `ctier L` 也报错了：

```

NSError errorWithDomain code userInfo
    https://rr1---sn-phvo-jpq1.googlevideo.com/videoplayback?expire=1657615602&ei=kuDMYrOaBpHDigT
5y4LYDw&ip=162.253.45.206&id=o-AMkZ1BzQ6p7kCJW94A6DtyMGG3HfqwKg1KnZ-fNzwp2U&source=youtube&requ
iressl=yes&mh=Mk&mm=31,29&mn=sn-phvo-jpq1,sn-vgqsknsk&ms=au,rdu&mv=m&mv1=1&pcm2cms=yes&p1=22&ct
ier=L&initcwndbps=1266250&svpuc=1&sabr=1&mt=1657593666&fvip=4&keepalive=yes&fexp=24001373,24007
246&c=IOS&sparams=expire,e1,ip,id,source,requiressl,ctier,svpuc,sabr&sig=A0q0QJ8wRQIgAqEYZ1R2jQ
z3cDKbLMWJKMhZa_JyTS1vtd_9064WLDkCIQDFL32Gx3XA9C4XUvNdixVm8TQjC1SjAv2N0B2bt-f1w%3D%3D&lsparms
=mh,mm,mn,ms,mv,mv1,pcm2cms,p1,initcwndbps&lsig=AG3C_xAwRgIhAJ8F9p0M1D7yBXsonsAKJ95tApty4njmPKz
uAFqoM81WA1EA1Fcqzn3LXLCxMMe_WxKUJ0d4L9uLrZ7oP_BtbspIFBY%3D&cpn=tuURUWDACKHN5FNX8&rn=2

```

## youtubeCallSequence

- 内部多个函数的调用过程：最初是从播放时间变成0开始分析，慢慢找到的函数调用逻辑
  - 其中最核心的几个函数：
    - 请求request
      - `NSURLRequest requestWithURL:cachePolicy:timeoutInterval:`
      - `NSURLRequest initWithURL:cachePolicy:timeoutInterval:`
    - 响应response
      - `NSHTTPURLResponse initWithURL:statusCode:HTTPVersion:headerFields:`
    - 其他
      - Cronet
        - `-[HAMCronetDataTask startWithDelegate:delegateQueue:]`

全部内容详见：

`youtubeCallSequence.coffee`

最后更新 20220823

【第一个广告视频ID的产生来源：`googlevideo.com initplayback`】

最早的 `initplayback` 的 `request` (的 `http body` 的 `data`)的来源：

```

YTPPlaybackDataLoader onlinePromise
    YTPPlaybackDataLoader createPlaybackRequest
        YTPPlaybackRequestFactory playbackRequestForPlayerTransition visibility viewportSizeProvider mediaStickySettings prefetchContext
            YTPPlaybackRequestFactory playbackRequestForNavigationEndpoint autoplay autonav flungFromMDX expiredForSeconds visibility viewportSizeProvider mediaStickySettings playerRequestDecorator prefetchContext latencyLogger watchEndpointLoggingContext
                YTPPlaybackRequest initWithProtoRequest URLRequestProperties CPN QOEController latencyLogger
                    参数
                        URLRequestProperties<{
                            MLOnesieRequestContext = "<MLOnesieRequestContext: 0x28240cd80>";
                            inline = 1;
                        }
                    }
                    YTPPlayerRequestFactory innerTubeRequestForPlayerWithVideoID playlistID playlistIndex playbackContext forOffline clickTrackingParams playerParams
                        参数
                            videoID DpJ7BL6RdFM
                            playlistID (null)
                            playlistIndex 0
                            playbackContext <YTIPPlaybackContext 0x282fbc480>: {
                                content_playback_context {
                                    time_since_last_ad_seconds 0
                                    last_milliseconds 1834558
                                    autoplays_since_last_ad 0
                                    conn 3
                                    vis 5
                                    fling false
                                    autoplay true
                                    adsense_client_params: "video_format=22&sdkv=i.17.08&output=xml"
                                    _vast2"
                                    autonav true
                                }
                            }
                            forOffline False
                            clickTrackingParams (length = 26, bytes = 0x08001084 67221308 95b584e3
95ddf902 ... d494091d 560607d6 )
                            playerParams CkkIDxAGQhd3LU1FWTViSkRJUE5vOWtQaU1tdH1Ba4oDKSADKAIwBzgFSh
MI1raF45Xd QIVg YoBR2IZAuZUgYIBT .....TE1UWV0mZ3Y9W0dPT0dMRV9WSUVXQUJJTE1UWV0Y0A8YAfAEAfgEAQ 3
D 3D

                    返回值:
                        <YTIPPlayerRequest 0x2838233a0>: {
                            context {
                                client {
                                    hl "zh-CN"
                                    gl "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"
                                    raw_device_id "FC1C0EA4-13EB-4B2E-AB49-32609F28DC8B"
                                    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: "CMvakJgGENLcrQUQjPKtBRC1760FEMaFrQUQ
mNSTBRCFta4FEKedrgUQgYauBRDXka0FEOK5rgUQx7GtBRChqq4FEJ0vrgUQhqGuBRCBxa0FEI61rQUQkJmuBRCvs64FEOW
4rgUQz4WuBRDxuq0FEIjprQUQj5SuBRDBvK4FEIy9rgUQxZquBRDtjq0FELytrgUQjZSuBRDqna0FEL22rgUQqM-oFxCcva
4FEJyxrgUQy6KuBRD0x60FEIH4rQUQuIuuBRDUg64FEKmqrgUQnv2tBRCFka4FELTPqBcQibGuBRD3iK4FEOK8rgUQ0_GtB
RDVVK4FEIG4rgUQpbCuBRCboK4FENm7rQUQ4a2uBRCX-60FEMWFrgUQt8utBRDrma0FEJaarQUQzeCtBRDr5K0FEPy6rgUQ
6pesBRDX9a0FEO-3ngUaMkFDS0c1cG5XU0dZOFRhMFRnNkJ5Vj1ieWtLUGRUVmxQUkxCWHhkRktXc1J5XzVNNmxRIjJBQ0t
HNXBuV1NHWThUYTBUzzZCeVY5Yn1rS1BkVFZsUFJMQ1h4ZEZLV3JSeV81TTzsUSoYQ0FNU0RBMERzTX1xQWhVRGdvdTVEQT
09"
            cold_hash_data: "CPnFk5gGEhQxMTU5NDYyMDIxMjM1NDQwNDMxOB
jL2pCYBii0z6gXKOK8rgUoibGuBSj3iK4FKJugrgUogbiuBS1lsK4FKN18rgUo_0_GtBS1B-K0FKMuIrgUonLGuBSiFka4FK
J79rQouIuuBS1pqq4FKPTHrQuo1IOuBSj8uq4FKNf1rQuo77euBSjq16wFKJf7rQuo4a2uBSjZu60FKLfLrQuo1pqtBSjr
5K0FKMWFrqUo65mtBSjN4K0FKJ0vrgUoh6quBSjXka0FKOK5rgUox7gtBSi0pa0FKK-zrgUokJmuBS1Bxa0FKIahrgUo0ty
tBSinna4FKMaFrQuogYauBSi1760FKIw1rgUojPKtBSjqna0FKI2UrgUovK2uBSi19tq4FKJjUrQuoxZquBSjtiq0FKKjPqB
conL2uBSjxuq0FKIjprQuoJ5SuBSj1uK4FKIy9rgUoz4WuBSjBvK4FMjJBQ0tHNXBuV1NHWThUYTBUzzZCeVY5Yn1rS1BkV
FZsUFJMQ1h4ZEZLV3JSeV81TTzsUToyQUNLRzVwbldTR1k4VGEwVGc2Qn1W0WJ5a0tQZFRwbFBSTEJYeGRGS1dyUn1fNU02
bFFCGBTVNEQTBEc015cUFoVURnb3U1REE9PQ%3D%3D"
            hot_hash_data: "CPnFk5gGEhQxNjEzNDg0MTcwMTQ4MTYyODk3Nxj
5xZOYBijJ-f0SKPne_RIo1P_9EiiRvv0SKIpt_RIoY639Eijc4P0SKOK__RIO0979Ei17_v0SKILG_RIopdD9Eijizv0SKN
LL_RIopqr9EiiFrP0SKKXv_RIonv_8EijI8f0SKL3K_RIope39EijU-f0SKIrg_RIo4az9EiibzP0SKnfM_Riomcb9Eiilq
P0SKM7A_RIo10T8EiifgP4SKLn1_Bio89_9Eijc_v0SKMvs_RIoYtj9EiikxP0SKNGf_RIoYvD9Eiiz-f0SKML2_RiovL79
EijGsv0SKNCy_RIo3JP9EiiqtP0SK004_RIo4oZ9Eiir1v0SKOPj_RIo-Pn9Eiibgv0SKKbw_RIoqar9Ei193v0SKMTs_RI
yMkFDS0c1cG5XU0dZOFRhMFRnNkJ5Vj1ieWtLUGRUVmxQUkxCWHhkRktXc1J5XzVNNmxR0jJBQ0tHNXBuV1NHWThUYTBUzz
ZCeVY5Yn1rS1BkVFZsUFJMQ1h4ZEZLV3JSeV81TTzsUUICq0FNU0VRMEJvdGY2R1pzcEZRRGR6OE1NbK13RQ%3D%3D"
        }
        screen_density_float: 2
        utc_offset_minutes: 480
        user_interface_theme: USER_INTERFACE_THEME_DARK
        time_zone: "Asia/Shanghai"
        eml_template_context: " \313\334\314\373\331\206\352\310\00
1 \206\327\321\205\205\213\346\323\361\001 \212\221\203\351\370\266\265\3168 \270\361\327\226\2
50\315\324\301) \221\307\376\333\240\306\304\357\370\001 \204\255\254\350\361\275\307\370\203\0
01 \365\360\247\340\223\316\264\322\016 \263\301\327\347\206\371\271\266. \320\301\200\233\321\
362\214\245\027 \220\364\314\240\332\360\264\300\314\001 \270\365\242\255\321\241\372\326y \360
\322\224\360\247\344\261\322\033 \310\335\301\224\261\262\275\372> \346\261\374\316\345\221\323
\350\321\001 \217\375\340\345\313\225\361\264% \206\370\205\214\332\344\367\272\247\001 \343\23
7\335\242\271\344\250\261\023 \333\236\251\262\323\323\306\322\200\001 \201\311\332\214\371\363
\217\252\267\001 \223\240\255\221\227\300\300\301\267\001 \332\271\246\252\206\257\225\273~ \21
4\327\321\232\303\213\204\363\251\001 \250\321\220\302\333\240\347\222\327\001 \307\306\342\230
\345\316\241\330~ \233\311\261\225\335\357\225\226\030 \214\222\262\254\377\315\326\267\365\001
\207\251\305\212\325\337\205\206\345\001 \252\207\217\352\236\264\232\265r"
        memory_total_kbytes: 2099249
        notification_permission_info {
            notifications_setting: NOTIFICATIONS_SETTING_ENABLED
            last_device_opt_in_change_time_ago_sec: 5707596
        }
        client_store_info {
            ios_store_country: "CHN"
        }
    }
    user {

```

```

        }
        request {
        }
        click_tracking {
            click_tracking_params: "\010\000\020\204g\" \023\010\225\265
\204\343\225\335\371\002\025=\324\224\t\035V\006\007\326"
        }
        ad_signals_info {
            params {
                key: "ms"
                value: "zNCgkMGJRpQCc6aMdpe0Gv6ez_sOHNtGlWj7lg6EjWr707Z
qIEwzYOOSSfzRnPhx3kY3I33makVsgrtFOzKammmpQqrf9Jq_eK18MyBd1-obQx_U8oanWw-_NlQf71191piuk2G3xzWMnX
Xln426o9M5jcbqwVPf5Nf8opchuQBPNPYwEbDXj6uTpkZ1471PL100BZpF5xWpjI14zuD3co0YMnBQHEcZv3AKQlp0UEksZ
diohp69Qk01ZoUPZrZrjty7kzuy6swSY7C5Baex2ZA11fj7Qkd_oxnqRxR1sMQuxaBF5CjfVUE5DMAQyxDg3aC3iCqE83h7
TJTnSRhOIQ"
            }
        }
    }
    video_id: "DpJ7BL6RdFM"
    playback_context {
        content_playback_context {
            time_since_last_ad_seconds: 0
            last_milliseconds: 1834558
            autoplays_since_last_ad: 0
            conn: 3
            vis: 5
            fling: false
            autoplay: true
            adsense_client_params: "video_format=22&sdkv=1.17.08&output
=xml_vast2"
            autonav: true
            mdx_context { mdx_context {} }
        }
    }
    racy_check_ok: false
    for_offline: false
    params: "CkkIDxAGQhd3LU1FWTV1SkRJUE5vOWtQaU1tdH1Ba4oDKSADKAIwBzgFSh
MI1raF45Xd-QIVg-YoBR2IZAuZUgYIBTABWAFOAXAJEpkaEpYaCAESkRoSDXByb21vdGVkVm1kZW9K_xkIARL6GSKpCwgTE
p0LaHR0cHM6Ly93d3cuZ29vZ2x1 ... ... tDTE1DS19NU10mYWRfbXQ9W0FEX01UXSZhY3Z3PVtWSUVXQUJJTE1UWV0mZ
3Y9W0dPT0dMRV9WSUVXQUJJTE1UWV0Y0A8YAfAEAfgEAQ%3D%3D"
    cpn: "Fmq_1vPSmpZkjLS0"
    override_muted_at_start: false
    caption_params {
        device_captions_on: false
    }
    video_quality_setting_params {
        video_quality_setting_mobile_network: VIDEO_QUALITY_SETTING_UNK
    }
    NOWN
        video_quality_setting_wifi: VIDEO_QUALITY_SETTING_UNKNOWN
        video_quality_last_written_ms: 0
    }
}

YTPlayerRequest init
GPBMessage

```

注：

`GPBMessage` ~ General Protocol Buffer Message

调用 `HAMDataLoadRequest` 时 傳入了 POST的application/x protobuf的NSData:

```
Module_Framework`-[YTTimedAction startTimerWithFireDate timeInterval repeats runLoopMode:]_block + 40
    Module_Framework`-[YTTimedAction triggerAction] + 124
        YouTube`-[YTInlineMutedPlaybackContainerController startPlaybackImmediately] + 324
            YouTube`-[YTInlineMutedPlaybackWatchController loadVideoAndAttachView] + 256
                Module_Framework`-[YTPlayerViewController prepareToLoadWithPlayerTransition expectedLayout:] + 276
                    Module_Framework`-[YTLocalPlaybackController prepareToLoadWithPlayerTransition expectedLayout:] + 556
                        Module_Framework`-[YTPublicDataLoader startLoading] + 1060
                            Module_Framework`-[YTPublicDataLoader onlinePromise] + 332
                                Module_Framework`-[YTPlayerService makePlaybackRequest responseBlock errorBlock] + 524
                                    Module_Framework`-[YTPlayerServiceTask sendRequest] + 248
                                        Module_Framework`-[YTInnerTubeRequestor makeInnerTubeRequest withParser completionBlock retryEnabled timeoutInterval enableLatencyTicks mutableSharedData] + 28
                                            Module_Framework`-[YTInnerTubeRequestor makeInnerTubeRequest withParser completionBlock retryEnabled timeoutInterval enableLatencyTicks mutableSharedData nilIdentityIsSignedOut:] + 36
                                                Module_Framework`-[YTBaseInnerTubeService makePOSTRequest withParser completionBlock retryEnabled timeoutInterval enableLatencyTicks mutableSharedData nilIdentityIsSignedOut:] + 540
                                                    Module_Framework`-[YTBaseInnerTubeService performHTTPRequest service withIdentity dataBlock errorBlock retryEnabled nilIdentityIsSignedOut:] + 628
                                                        Module_Framework`-[YTVISitorDataRequest Decorator decorateRequest withCompletionBlock] + 260
                                                            sub_186AB30 = Module_Framework`__1
                                                                ldb_unnamed_symbol14785$$Module_Framework + 608
                                                                    Module_Framework`-[YTBaseNetworkService performHTTPRequest withAuthorizer responseBlock errorBlock retryConfig] + 184
                                                                        Module_Framework`-[YTBaseNetworkService performHTTPRequest withAuthorizer backgroundCompletionBlock retryConfig] + 928
                                                                            Module_Framework`-[MLOnesieUMPFetcherTask resume] + 208
                                                                                Module_Framework`-[MLOnesieRequestFactory onesieRequestForPlayerRequest authorization dataLoader context cryptor requestNumber completionHandler] + 432
                                                                                    Module_Framework`-[MLOnesieRequestFactory onesieRequestForPlayerRequest authorization dataLoader context cryptor requestNumber completionHandler]_block + 64
                                                                                        Module_Framework`-[MLOnesieRequestFactory onesieRequestForPlayerRequest dataLoader context cryptor requestNumber error] + 5628
                                                                                            参数:
```

```

req
uest <NSMutableURLRequest 0x282e83150> {
    URL = "https://youtubei.googleapis.com/youtubei/v1/player?key=AIZaSyB-63vPrdThhKuerbB2N_17Kwwcx
j6yUAc&id=r17kWuxKXKs&t=DA5C7286-F10C-4EBC-942E-96B8CA551C57"
}

dat
aLoader <HAMCronetDataLoader 0x283c621f0>
con
text <MLOnesieRequestContext 0x2809ae3a0>
cry
ptor <MLOnesiePlayerServiceCryptor: 0x283c68930>
req
uestNumber = 1
err
or 0x16d760c08

HAMData
LoadRequest initWithRangedURL timeoutInterval HTTPMethod HTTPContentType HTTPBody idempotent
参数
:

rangeUrl = HAMRangedURL: 0x281963320>

    URL = "https://rr1---sn-hp57kndr.googlevideo.com/initplayback?source=youtube&orc=1&oeis=1&
c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1
&osnz=1&rbqsm=fr&msp=1&odeak=1&odepv=1&osfc=1&id=fd97bcf151a359c8&initcwndbps=1645000&mt=166115
6982&pxtags=Cg4KAnR4EggyNDE5NzI3NQ&rxtags=Cg4KAnR4EggyNDE5NzI3NQ,Cg4KAnR4EggyNDE5NzI3Ng,Cg4KAnR
4EggyNDE5NzI3Nw&ack=1&cpn=AuykYLjoeyBPVKWO&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&p
ai=139,140"

timeoutInterval 60
HTTPMethod = "POST"
HTTPContentType "application/x-protobuf"
HTTPBody = 7647 bytes
idempotent true

```

`googlevideo.com initplayback` 的调用顺序:

```

[case 1]

MLOnesieUMPFetcherTask onRequestFactoryCompletionWithRequest error:
参数
    request <HAMDataLoadRequest: 0x282027040>
        _idempotent (BOOL): YES
        _rangedURL (HAMRangedURL): <HAMRangedURL: 0x283831120>
        _timeoutInterval (double): 60
        _HTTPMethod (NSString): @"POST"
        _HTTPContentType (NSString): @"application/x-protobuf"
        _HTTPBody (NSData): <NSConcreteData 0x283831140>

```

```

    error (null)

    startRequestTask_104238C = -[MLOnesieUMPFetcherTask onRequestFactoryCompletionWithRequest e
rror ]_block + 64
        -[HAMCronetDataLoader taskWithRequest userInfo] + 140
            -[HAMDataLoadRequest buildURLRequest] + 140
                NSURLRequest requestWithURL cachePolicy timeoutInterval
                    参数
                        URL "https://rr5---sn-p5qddn76.googlevideo.com/initplayback?source=yout
ube&orc=1&oeis=1&c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses
=1&ofpcc=1&osbr=1&osnz=1&rbqsm=fr&msp=1&odeak=1&odepv=1&osfc=1&id=a52feece777301a2&initcwndbps=
1786250&mtn=1660899670&pxtags=Cg4KAnR4EggyNDE5NzI3NQ&rxtags=Cg4KAnR4EggyNDE5NzI3NQ,Cg4KAnR4EggyN
DE5NzI3Ng,Cg4KAnR4EggyNDE5NzI3Nw&ack=1&cpn=Bu494mABvg3R6sCr&rn=1&opr=1&pvi=160,133,134,135,136,
137,298,299,264&pai=139,140"
                            cachePolicy NSURLRequestUseProtocolCachePolicy
                            timeoutInterval 60

[case 2]

callPrewarmConnection_10470B8
    startTask_1048AA0 = __lldb_unnamed_symbol12576$$Module_Framework + 72
        -[HAMCronetDataLoader taskWithRequest userInfo] + 140
            -[HAMDataLoadRequest buildURLRequest] + 380
                NSURLRequest requestWithURL cachePolicy timeoutInterval
                    参数
                        URL "https://rr2---sn-p5qs7nsr.googlevideo.com/initplayback?source=yout
ube&orc=1&oeis=1&c=IOS&oss=1&oda=1&oad=5500&ovd=5500&oaad=11000&oavd=11000&ocs=700&oputc=1&oses
=1&ofpcc=1&osbr=1&osnz=1&rbqsm=fr&msp=1&odeak=1&odepv=1&osfc=1&id=8bfb111f4c684ce5&initcwndbps=
1683750&mtn=1660916238&pxtags=Cg4KAnR4EggyNDE5NzI3NQ&rxtags=Cg4KAnR4EggyNDE5NzI3NQ,Cg4KAnR4EggyN
DE5NzI3Ng,Cg4KAnR4EggyNDE5NzI3Nw&cfn=YCXS_oBo1TcIotC0&pai=0&pvi=0&owc=1"
                            cachePolicy NSURLRequestUseProtocolCachePolicy
                            timeoutInterval 60

```

MLServerABRLoader dataLoadTask didReceiveData 的后续response的数据的parse过程

当 禁用广告过滤 正常加载广告时的流程:

(1) MLServerABRLoader -> HAMMP4InitializationChunk

```

[HAMCronetDataLoadTask startWithDelegate delegateQueue ]_block_block + 36
    -[MLServerABRDataTask dataLoadTask didReceiveData] + 292
        MLServerABRLoader dataLoadTask didReceiveData:
            serverABRUMPPParserParseResponseData_FE2F90 = __lldb_unnamed_symbol12192$$Module_Fr
amework + 136
                parserParseData_32CAA84 = __lldb_unnamed_symbol184370$$Module_Framework + 160
                    extractMediaHeaderBytes_32CAC60
                callParseData_32CAE80 = __lldb_unnamed_symbol184372$$Module_Framework + 52
                    sub_FE4C80 = __lldb_unnamed_symbol12224$$Module_Framework
                    BaseUMPParser_parseData_FE37D4 = __lldb_unnamed_symbol12209$$Modul
e_Framework + 172
                        citer 的 videoplayback
                            for youtube media: ServerABRUMPPParser + 16 => checkMediaHe
aderUmpBadPartid_107DD4C = __lldb_unnamed_symbol12801$$Module_Framework
                                MLQOEError QOEWithErrorCode details:

```

```

    参数:
        code_ump_badpartid
        details { partid = 0xA; }

    其他情况, 比如 正常流程 或 oad 的 initplayback
        for youtube::media::OnesieUMPParser + 504 => checkMediaHeader
erUmpBadPartid_1046494 = __lldb_unnamed_symbol12555$$Module_Framework
                           toCallProcessMediaHeader_FE2EB0 = __lldb_unnamed_symbol
112191$$Module_Framework + 108
                           validateMediaHeader_FE72BC = __lldb_unnamed_symbol
12318$$Module_Framework + 144
                           parseAndSaveChunk_107D850 = __lldb_unnamed_symbol
bol12797$$Module_Framework + 64
                           - [MLServerABRLoader chunkForMediaHeader] +
324
                           + [MLMP4ChunkFactory initializationChunk
WithURL:] + 72
                           HAMMP4InitializationChunk initWithF
ormat rangedURL preferTrackTimescale

    参数: case 1
        arg1 0x0000000282ca6b80
        arg2 0x000000028116c800
        arg3 false

(2) MLOnesieUMPFetcherTask -> HAMMP4InitializationChunk

[HACronetDataLoadTask startWithDelegate:delegateQueue:]_block_block + 36
    - [MLOnesieUMPFetcherTask dataLoadTask didReceiveData:] + 196
        __lldb_unnamed_symbol12192$$Module_Framework + 136
        __lldb_unnamed_symbol184370$$Module_Framework + 160
        __lldb_unnamed_symbol184372$$Module_Framework + 52
        __lldb_unnamed_symbol12209$$Module_Framework + 312
        sub_1045904 = __lldb_unnamed_symbol12549$$Module_Framework + 168
        __lldb_unnamed_symbol12191$$Module_Framework + 108
        processOnesieMediaHeader_10482D8 = __lldb_unnamed_symbol12575$-
$Module_Framework + 1120
                           - [MLMP4ChunkFactory initializationChunkWithURL] + 72
                           HAMMP4InitializationChunk initWithFormat rangedURL pref
erTrackTimescale

(3) MLServerABRDataTask -> HAMChunk

[HACronetDataLoadTask startWithDelegate:delegateQueue:]_block_block + 36
    - [MLServerABRDataTask dataLoadTask didReceiveData:] + 292
        __lldb_unnamed_symbol12192$$Module_Framework + 136
        __lldb_unnamed_symbol184370$$Module_Framework + 160
        __lldb_unnamed_symbol184372$$Module_Framework + 52
        __lldb_unnamed_symbol12209$$Module_Framework + 336
        sub_107DAB0 = __lldb_unnamed_symbol12800$$Module_Framework + 232
                           HAMChunk loadDidCompleteWithError

(4) MLOnesieChunkLoadTask -> HAMChunk

[MLOnesieChunkLoadTask initWithRange:chunks:dataBuffer:cache:delegate:queue:latencyLogger]_bl
ock_block + 100
    - [HAMBaseChunkLoadTask didReceiveData:error] + 656

```

```

- [HAMBaseChunkLoadTask currentChunkDidComplete] + 256
  - [HAMMediaChunk loadDidCompleteWithError:] + 48
    HAMchunk loadDidCompleteWithError

(5) -> MLServerABRLoader recordInvalidFormatQOEErrorWithFormatID extraDetails:

[HAMCronetDataLoadTask startWithDelegate:delegateQueue:]_block_block + 36
  - [MLServerABRDataTask dataLoadTask didReceiveData] + 292
    __lldb_unnamed_symbol12192$$Module_Framework + 136
      __lldb_unnamed_symbol184370$$Module_Framework + 160
        __lldb_unnamed_symbol184372$$Module_Framework + 52
          __lldb_unnamed_symbol12209$$Module_Framework + 624
            chekAndCallProcessMediaHeader_FE43F4 = __lldb_unnamed_symbol12214$$Module_Framework + 276
              __lldb_unnamed_symbol12191$$Module_Framework + 108
                void youtube::media::BaseUMPParser::AccumulateDataAndParseProto(
video_streaming::FormatInitializationMetadata (unsigned int, absl::string_view, video_streaming::UMPPartID, std::unique_ptr<function<void (video_streaming::FormatInitializationMetadata)>>)::'lambda'(absl::string_view):operator()(absl::string_view) const + 436
                  sub_FE61AC = __lldb_unnamed_symbol12275$$Module_Framework + 160
                    callDidParseFormatInitializationMetadata_107E5FC = __lldb_unnamed_symbol12805$$Module_Framework + 44
                      - [MLServerABRLoader didParseFormatInitializationMetadata] + 268
                        MLServerABRLoader recordInvalidFormatQOEErrorWithFormatID extraDetails:
                          参数
                          formatID = "140_1629449114196897"
                          extraDetails = "init_metadata_badid"

```

## 相关函数

[callParseData\_32CAE80]

核心代码逻辑:

```

return (*(&curServerABRUMPParserOffset0x20Value + 0x30LL))();
// call: sub_FE4C80 = __lldb_unnamed_symbol1224:
//       x0 = curServerABRUMPParserOffset0x20Value
//       x1 = curServerABRUMPParserOffset0x40Ptr
//       x2 = remainBufferSizePtr
//       x3 = mediaHeaderBytesPtr

```

[sub\_FE4C80]

核心逻辑:

```
return BaseUMPParser_parseData_FE37D4(*(_QWORD *)(&a1 + 8), &a2, &a3, &a4);
```

[BaseUMPParser\_parseData\_FE37D4]

核心逻辑:

```
result = ((__int64 (*)())(&_QWORD(result + 0x70LL))());
// 实际调试 = 出错流程? : checkMediaHeaderUmpBadPartid_107DD4C
```

[chekAndCallProcessMediaHeader\_FE43F4]

核心逻辑:

```

    toCallProcessMediaHeader_FE2EB0(v10, v9, v8, v7, (__int64)&v21);

[ toCallProcessMediaHeader_FE2EB0 ]
核心逻辑:
    result = (*(__int64 (**)(void))(*(_QWORD *)v9 + 0x30LL))();
    // for MLServerABRDataTask(MLServerABRLoader) is: validateMediaHeader_FE72BC
    // for MLOnesieUMPFetcherTask is: processOnesieMediaHeader_10482D8

[ validateMediaHeader_FE72BC ]
核心逻辑:
    (*void __fastcall **)(__int64, char *)(*_QWORD *v2 + 32LL)(v2, &v10); // = parseAndSaveChunk_107D850

[ parseAndSaveChunk_107D850 ]
核心逻辑:
    v7 = (struct objc_object *)objc_msgSend_3E84298(curMLServerABRLoader_, "chunkForMediaHeader:", mediaHeader);

[youtube::media::BaseUMPParser::AccumulateDataAndParseProto video_streaming::FormatInitializationMetadata](unsigned int, absl::string_view, video_streaming::UMPPartID, std::unique_ptr<function<void ()>(video_streaming::FormatInitializationMetadata))::[lambda(absl::string_view)]
核心逻辑:
    (*void (**)(void))(*(_QWORD *)v20 + 0x30LL))(); // sub_FE61AC

[ sub_FE61AC ]
核心逻辑:
    (*void (**)(void))(*(_QWORD **)(v3 + 8) + 0x50LL))(); // callDidParseFormatInitializationMetadata_107E5FC

```

-----

ctier 重发请求 死循环的调用逻辑:

```

[MLServerABRLoader continueLoading]
    shouldSendRequest = True
        - [MLServerABRDataTaskForRequest createDataLoadTaskForRequest] + 128
            - [MLServerABRDataTaskForHTTPBody isPrimaryHostProbing] + 424
                - [HAMCronetDataLoader taskWithRequest userInfo] + 140
                    - [HAMDataLoadRequest buildURLRequest] + 140
                        NSURLRequest requestWithURL cachePolicy timeoutInterval

```

```

                - [MLServerABRDataTask startWithDelegate delegateQueue] + 88
                    HAMCronetDataLoadTask startWithDelegate delegateQueue
                        Cronet_UrlRequestCallback_OnResponseStarted_194CCE4 = __lldb_unnamed_symbol156
76$$Module_Framework
                        generateNewHttpResponse_194D6D8 = __lldb_unnamed_symbol15681$$Module_Framework
work + 376 = sub_194D6D8
                        NSHTTPURLResponse initWithURL statusCode HTTPVersion headerFields ->

```

创建新的 ctier 的 response

```

                - [MLServerABRDataTask dataLoadTask didReceiveResponse] + 424
                    - [MLServerABRLoader dataLoadTask didReceiveResponse] -> 当 ctier 报
错时
                        NSError HAMErrorWithURLResponse userInfo request
                        NSError HAMErrorWithCode userInfo

```

```

Cronet_UrlRequestCallback_OnSucceeded_194D248 = __lldb_unnamed_symbol15678$$Module_Framework
MLServerABRDataTask dataLoadTask didCompleteWithError:
    MLServerABRLoader dataLoadTask didCompleteWithError:
        MLServerABRLoader updateLoadRetryPolicyOnLoadCompletionWithError:
            MLDefaultLoadRetryPolicy didCompleteLoadWithError:
                error == NULL : MLDefaultLoadRetryPolicy cancelRetryAndResetBackoff
retryID++
error == NULL : MLDefaultLoadRetryPolicy scheduleRetryWithBlock:
    canRetry == True callExecuteRetryBlockWithRetryID_FF3070 = -[MLDefaultLoadRetryPolicy scheduleRetryWithBlock ]_block + 44
        -[MLDefaultLoadRetryPolicy executeRetryBlockWithRetryID ] + 68
callMLServerABRLoaderContinueLoading_107B5A8
    -[MLServerABRLoader continueLoading] + 924
        -[MLServerABRDataTask startWithDelegate:
delegateQueue] + 88
HAMCronetDataTask startWithDelegate:
delegateQueue -> 进入 ctier 的死循环 重复请求

canRetry == False callMLServerABRLoaderFailWithError_107B5B8
MLServerABRLoader failWithError:

```

#### 相关函数详解：

```

MLServerABRLoader dataLoadTask didCompleteWithError:
if ( curServerABRUMPPParser )
    parseResponseError_107E6B0(curServerABRUMPPParser, inputError_1);

if ( !inputError_1
    && curMLServerABRLoader->_didReceiveChunkForCurrentRequest
    && curMLServerABRLoader->_didReceiveNextRequestPolicyForCurrentRequest )
{
    defaultBackOffTimeMs = curMLServerABRLoader->_defaultBackOffTimeMs;
    if ( defaultBackOffTimeMs >= 1 )
    {
        ...
    }
}

objc_msgSend_3E84298(curMLServerABRLoader, "updateLoadRetryPolicyOnLoadCompletionWithError:"
, inputError_3);

MLServerABRLoader updateLoadRetryPolicyOnLoadCompletionWithError:
参数:
case 1
arg1 {
    Error Domain com.google.ios.hamplayer

```

```

        Code = 10000 "(null)"
        UserInfo {
            HAMRequestNumberLoadTaskKey = 1,
            HAMErrorURLRequest <NSMutableURLRequest 0x280ab2470> {
                URL: https://rr1---sn-8xgp1vo-p5ql.googlevideo.com/videoplayback?expire=1654546559&ei=HwyeYrH5DJWN_9EPouqpqA8&ip=70.110.26.104&id=o-ABMzsKNB0GJ_BmboM7YSytnAHB7ASC_Y7G
                C6j0056wJe&source=youtube&requiressl=yes&mh=j1&mm=31,29&mn=sn-8xgp1vo-p5ql,sn-p5qlsndz&ms=au,rd
                u&mv=m&mv1=1&pcm2cms=yes&p1=20&ctier=L&initcwndbps=1490000&svpuc=1&sabr=1&mt=1654524545&fvip=2&
                keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,svpu
                c,sabr&sig=A0q0QJ8wRgIhAPnD2xwm1J-b9yvruQe1aozyNm_9xaCQvtdIM7n32vA5AiEA4mVXjIUgf7425J8X1wfLdVhM
                yD1ZHftRhRE1jHxtWZo%3D&lsparms=mh,mm,mn,ms,mv,mv1,pcm2cms,p1,initcwndbps&lsig=AG3C_xAwRQIgd3oo
                Iz7fpF6JTB11opldSL1WIpx4L5FaBEqhdEZBLwCIQChh18q1sc8T2XEIwlQBI809cF_DI2FPVLjVCNRhRPCQ%3D%3D&cp
                n=sjOhS6zpDL-OE-pP&rnr=1
            }
        }
    }

    case 2:
        arg1 domain: "com.google.ios.hamplayer" - code 6009

    case 3:
        Error Domain com.google.ios.hamplayer
        Code 6009 "(null)"
        UserInfo = {
            HAMErrorHTTPURLResponse <NSHTTPURLResponse 0x28095cd60> {
                URL: https://rr1---sn-8xgp1vo-p5ql.googlevideo.com/videoplayback?expire=165
                4546559&ei=HwyeYrH5DJWN_9EPouqpqA8&ip=70.110.26.104&id=o-ABMzsKNB0GJ_BmboM7YSytnAHB7ASC_Y7GC6j0
                056wJe&source=youtube&requiressl=yes&mh=j1&mm=31,29&mn=sn-8xgp1vo-p5ql,sn-p5qlsndz&ms=au,rd
                u&mv=m&mv1=1&pcm2cms=yes&p1=20&ctier=L&initcwndbps=1490000&svpuc=1&sabr=1&mt=1654524545&fvip=2&keep
                alive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,svpuc,sa
                br&sig=A0q0QJ8wRgIhAPnD2xwm1J-b9yvruQe1aozyNm_9xaCQvtdIM7n32vA5AiEA4mVXjIUgf7425J8X1wfLdVhMyD1Z
                HftRhRE1jHxtWZo%3D&lsparms=mh,mm,mn,ms,mv,mv1,pcm2cms,p1,initcwndbps&lsig=AG3C_xAwRQIgd3ooIz7f
                pF6JTB11opldSL1WIpx4L5FaBEqhdEZBLwCIQChh18q1sc8T2XEIwlQBI809cF_DI2FPVLjVCNRhRPCQ%3D%3D&cpn=sj
                OhS6zpDL-OE-pP&rnr=2
            }
        }
    }

    Status Code: 302,
    Headers {
        Connection = close
        Content Length = 0
        Location = "https://rr1---sn-8xgp1vo-p5ql.googlevideo.com/videoplay
        back?expire=1654546559&ei=HwyeYrH5DJWN_9EPouqpqA8&ip=70.110.26.104&id=o-ABMzsKNB0GJ_BmboM7YSytn
        AHB7ASC_Y7GC6j0056wJe&source=youtube&requiressl=yes&mh=j1&mm=31,29&mn=sn-8xgp1vo-p5ql,sn-p5qlsn
        dz&ms=au,rd&u&mv=m&mv1=1&pcm2cms=yes&p1=20&ctier=A&initcwndbps=1490000&svpuc=1&sabr=1&mt=1654524
        545&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl
        ,ctier,svpuc,sabr&sig=A0q0QJ8wRgIhAPnD2xwm1J-b9yvruQe1aozyNm_9xaCQvtdIM7n32vA5AiEA4mVXjIUgf7425
        J8X1wfLdVhMyD1ZHftRhRE1jHxtWZo%3D&lsparms=mh,mm,mn,ms,mv,mv1,pcm2cms,p1,initcwndbps&lsig=AG3C_
        xAwRQIgd3ooIz7fpF6JTB11opldSL1WIpx4L5FaBEqhdEZBLwCIQChh18q1sc8T2XEIwlQBI809cF_DI2FPVLjVCNRhRP
        CQ%3D%3D&cpn=sjOhS6zpDL-OE-pP&rnr=2"
        Server = "atomos-tproxy 1.0"
    }
}

HAMRequestNumberLoadTaskKey = 2,
HAMErrorURLRequest <NSMutableURLRequest 0x280ab1f50> {
    URL: https://rr1---sn-8xgp1vo-p5ql.googlevideo.com/videoplayback?expire=165
    4546559&ei=HwyeYrH5DJWN_9EPouqpqA8&ip=70.110.26.104&id=o-ABMzsKNB0GJ_BmboM7YSytnAHB7ASC_Y7GC6j0
}

```

```
056wJe&source=youtube&requiressl=yes&mh=j1&mm=31,29&mn=sn-8xgp1vo-p5ql,sn-p5qlsndz&ms=au,rd&mv=m&mv1=1&pcm2cms=yes&p1=20&ctier=L&initcwndbps=1490000&svpuc=1&sabr=1&mt=1654524545&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,e1,ip,id,source,requiressl,ctier,svpuc,sabr&sig=A0q0QJ8wRgIhAPnD2xwm1J-b9yvruQe1aozyNm_9xaCQvtdIM7n32vA5AiEA4mVXjIUgf7425J8X1wfLdVhMyD1ZHFtRhRE1jHxtWZo%3D&lsparams=mh,mm,mn,ms,mv1,pcm2cms,p1,initcwndbps&lsig=AG3C_xAwRQIgd3ooIz7fpF6JTB11opl1SL1WIPTx4L5FaBEqhdEZBLwCIQChh18q1sc8T2XEIwlQBI809cF_D12FPVLjVCNRHhRPCQ%3D%3D&cfn=sjOhS6zpDL-OE-pP&rn=2
    }
}
}
```

核心代码逻辑：

```
objc_msgSend_3E84298((void *)curMLServerABRLoader->_loadRetryPolicy, "didCompleteLoadWithError:", error_);
if (error_)
{
    if ((unsigned int)objc_msgSend_3E84298((void *)curMLServerABRLoader->_loadRetryPolicy, "canRetry"))
    {
        blockInvoke = callMLServerABRLoaderContinueLoading_107B5A8;

        objc_msgSend_3E84298((void *)loadRetryPolicy, "scheduleRetryWithBlock:", &StackBlock);
    }
    else
    {
        blockInvoke1 = callMLServerABRLoaderFailWithError_107B5B8;
    }
}
}
```

调用到：

```
MLDefaultLoadRetryPolicy canRetry

MLDefaultLoadRetryPolicy didCompleteLoadWithError

MLDefaultLoadRetryPolicy scheduleRetryWithBlock

callMLServerABRLoaderContinueLoading_107B5A8

callMLServerABRLoaderFailWithError_107B5B8
```

MLDefaultLoadRetryPolicy didCompleteLoadWithError

参数：

```
case 1: error NULL
case 2: NSError(
    Error Domain com.google.ios.hamplayer
    Code 10000 "(null)"
    UserInfo {
        HAMRequestNumberLoadTaskKey_1,
        HAMErrorURLRequest=NSMutableURLRequest: 0x2838e9480>{
            URL: https://rr4---sn-a5msenes.googlevideo.com/videoplayback?expire
=1654443385&ei=GxmcYoryCa-Rsf1P1Oy_mAs&ip=104.172.170.169&id=o-AnzDZ3h7N5rZAl0UCdQGuYMBEb80Uu_e
4KPc7xaAyG1K&source=youtube&requiressl=yes&mh=_h&mm=31,29&mn=sn-a5msenes,sn-a5mekn6k&ms=au,rd&mv=m&mv1=4&p1=17&ctier=L&initcwndbps=1842500&svpuc=1&sabr=1&mt=1654421335&fvip=1&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,e1,ip,id,source,requiressl,ctier,svpuc,sabr&sig=A0q0QJ8wRQIgLJQmm-EOXj9xFYjCBH1gn2RiukszRseDFdGPROqj8ACIQCdoic9p3f1B3FJOu2lqys31IE4I-wBZKNPR-hjkf
```

```
0HHA%3D%3D&lsparams=mh,mm,mn,ms,mv,mvi,pl,initcwndbps&lsig=AG3C_xAwRAIgULlnjQq0nkVEN6oLDPNPfvxU
rcF1uFeiOfQiUhNE9RMCIgbuOCN12TVJp2VegKsJVujYtW_j53A1zYzznjVe3xp3&cpn=q6v1kI3_nWlzP4Qd&rn=1
    }
}
}

核心代码逻辑:
if ( inputError )
{
    if ( objc_msgSend_3E84298(inputError, "code") != &zero_2710 && objc_msgSend_3E84298(
inputError_, "code") != algn0x20_1779 )
    {
        objc_msgSend_3E84298((void *)curMLDefaultLoadRetryPolicy->_errors, "addObject:",
inputError_);
        objc_msgSend_3E84298(curMLDefaultLoadRetryPolicy->_backoffCalculator, "updateIn
terval");
    }
    else
    {
        objc_msgSend_3E84298((void *)curMLDefaultLoadRetryPolicy->_errors, "removeAllObject
s");
        objc_msgSend_3E84298(curMLDefaultLoadRetryPolicy->_backoffCalculator, "resetInterva
l");
        objc_msgSend_3E84298(curMLDefaultLoadRetryPolicy, "cancelRetryAndResetBackoff:", 0LL
);
    }
}
}

调用到:
MLDefaultLoadRetryPolicy cancelRetryAndResetBackoff
```

**MLDefaultLoadRetryPolicy cancelRetryAndResetBackoff**

核心逻辑:

```
++self->_retryID;
curMLDefaultLoadRetryPolicy->_retryBlock = 0LL;
```

**MLDefaultLoadRetryPolicy scheduleRetryWithBlock**

核心逻辑:

```
curMLDefaultLoadRetryPolicy->_retryBlock = newRetryBlock_2;
if ( !curRetryBlock )
{
    retryID = curMLDefaultLoadRetryPolicy->_retryID;
    objc_msgSend_3E84298(curMLDefaultLoadRetryPolicy->_backoffCalculator, "interval");
    dispatchTime_backoffInterval = dispatch_time(0LL, (signed __int64)(backoffInterval *
1000000000.0));

    blockInvoke = callExecuteRetryBlockWithRetryID_FF3070;
}
```

调用到:

```
callExecuteRetryBlockWithRetryID_FF3070
```

**Cronet.UrlRequestCallback\_OnResponseStarted\_194CCE4 = \_\_lldb\_unnamed\_symbol15676\$\$Module\_Frame
work + 80 = sub\_194CCE4**

代码核心逻辑:

```

if ( !curCronetDataLoadTask->taskCancelled_ )
{
    newNSHTTPURLResponse = generateNewHttpReponse_194D6D8(responseInfo_);

    objc_msgSend_3E84298(
        (void *)curCronetDataLoadTask->delegate_,
        "dataLoadTask:didReceiveResponse:",
        curCronetDataLoadTask->parentTask_,
        curCronetDataLoadTask->response_
    );

    if ( (signed int)getResponseStatusCode_3DFA210(responseInfo_) < 400 )
    {
        curCronetDataLoadTask->bufferData_ = malloc(curCronetDataLoadTask->bufferData
Size_);
    }

    tryResetTimeoutTimer_194C704(curCronetDataLoadTask_);

    cronetUrlRequestReadResult = callCronetUrlRequestRead_3DF9370(request_);

    if ( (_DWORD)cronetUrlRequestReadResult ) // read error
    {
        ...
    }
    else
    {
        newHamError = objc_msgSend_3E84298(
           _OBJC_CLASS__NSError,
            "HAMErrorWithURLResponse:userInfo:request:",
            curCronetDataLoadTask->response_,
            curCronetDataLoadTask->userInfo_,
            curCronetDataLoadTask->URLRequest_
        );

        objc_msgSend_3E84298(
            (void *)curCronetDataLoadTask->delegate_,
            "dataLoadTask:didCompleteWithError:",
            curCronetDataLoadTask->parentTask_,
            newHamError_
        );
        objc_msgSend_3E84298(
            (void *)curCronetDataLoadTask->observer_,
            "dataLoadTask:didCompleteWithError:",
            curCronetDataLoadTask->parentTask_,
            newHamError_
        );
    }

    objc_msgSend_3E84298(curCronetDataLoadTask->timeoutTimer_, "invalidate");
}
}

```

新生成 ctier L 的 NSMutableURLRequest 请求的函数调用顺序:

```
[HAMPlayerInternal playerLoop] + 1736
  - [HAMBaseTrackRenderer prepareAndReturnError] + 48
    - [HAMSBARAudioTrackRenderer internalPrepareAndReturnError] + 40
      - [HAMQueueingRootSampleBufferSource prepareAndReturnError] + 704
        - [HMDASHSampleBufferSource prepareAndReturnError] + 100
          - [MLServerABRController prepareAndReturnError] + 40
            - [MLServerABRLoader continueLoading] + 840
              - [MLServerABRDataLoader createDataLoadTaskForRequest] + 128
                - [MLServerABRDataLoader dataLoadTaskForHTTPBody isPrimaryHostPr
obing] + 424
                  - [HACronetDataLoader taskWithRequest userInfo] + 140
                    - [HAMDataLoadRequest buildURLRequest] + 140
                      NSURLRequest requestWithURL cachePolicy timeoutInte
rval

```

某次参数:

```
URL = "https://rr4---sn-nx57ynsz.googlevideo.com/videoplayback?expire=1655759112&ei=p4ywYoCpO-GMsfIP9c-j-A0&ip=72.174.144.110&id=o-AOtW_cgK5DpGS1TFzlnNQ5jG-qYf6Yb0-W_J4o6Z7Bg0&source=youtube&requiressl=yes&mh=mR&mm=31,29&mn=sn-nx57ynsz,sn-nx5s7n7y&ms=au,rdu&mv=m&mvi=4&pl=19&ctier=L&initcwndbps=1712500&svpuc=1&sabr=1&mt=1655737257&fvip=1&keepalive=yes&fexp=24001373,24007246,24208264&c=IOS&sparams=expire,ei,ip,id,source,requ
uiressl,ctier,svpuc,sabr&sig=AQq0QJ8wRAIgdJbzVTf5BpT1a0Qd-15luRpD3uxu9oqoIBFYuNPFrw0CIHogAdP_kVZFI1gE2yvIbih1AY5458ViOpGC77ywCY1Y&lsp

```

cachePolicy = NSURLRequestUseProtocolCachePolicy

```
timeoutInterval 60
CFNetwork`__lldb_unnamed_symbol1231$$CFNetwork + 64

```

NSURLRequest initWithURL cachePolicy timeoutInterval:

某次参数:

```
URL = "https://rr4---sn-nx57ynsz.googlevideo.com/videoplayback?expire=1655759112&ei=p4ywYoCpO-GMsfIP9c-j-A0&ip=72.174.144.110&id=o-AOtW_cgK5DpGS1TFzlnNQ5jG-qYf6Yb0-W_J4o6Z7Bg0&source=youtube&requiressl=yes&mh=mR&mm=31,29&mn=sn-nx57ynsz,sn-nx5s7n7y&ms=au,rdu&mv=m&mvi=4&pl=19&ctier=L&initcwndbps=1712500&svpuc=1&sabr=1&mt=1655737257&fvip=1&keepalive=yes&fexp=24001373,24007246,24208264&c=IOS&sparams=expire,ei,ip,id,source,requ
uiressl,ctier,svpuc,sabr&sig=AQq0QJ8wRAIgdJbzVTf5BpT1a0Qd-15luRpD3uxu9oqoIBFYuNPFrw0CIHogAdP_kVZFI1gE2yvIbih1AY5458ViOpGC77ywCY1Y&lsp

```

cachePolicy = NSURLRequestUseProtocolCachePolicy

```
timeoutInterval 60

```

触发 HACronetDataLoadTask startWithDelegate delegateQueue 的调用顺序:

```
[第一次 run ctier L]
[HAMPlayerInternal playerLoop] + 1736
  - [HAMBaseTrackRenderer prepareAndReturnError] + 48

```

```

- [HAMSBARAudioTrackRenderer internalPrepareAndReturnError] + 40
  - [HAMQueueingRootSampleBufferSource prepareAndReturnError] + 704
    - [HMDASHSampleBufferSource prepareAndReturnError] + 100
      - [MLServerABRController prepareAndReturnError] + 40
        _logos_method$_ungrouped$MLServerABRLoader$continueLoading(self 0x00000
0012e689210, _cmd "continueLoading") at youtubeCronet.xm 289:5
          - [MLServerABRLoader continueLoading] + 924
            - [MLServerABRDataTask startWithDelegate delegateQueue] + 88

  HAMCronetDataTask startWithDelegate delegateQueue:
  某次的参数:
    arg1 delegate <MLServerABRDataTask 0x284243690>

    arg2 delegateQueue = OS_dispatch_queue_serial: MLHAM
Player

此时:
reqUrl
rn 1: https://rr1---sn-phvo-jpql.googlevideo.co
m/videoplayback?expire=1658155941&ei=RR_VYqa6GJ71NsC6gIgE&ip=162.253.45.206&id=o-AGLA8-7MEt9wII
kvKGany12Vu6HmM4t-R27oa-wU3eyG&source=youtube&requiressl=yes&mh=Mk&mm=31,29&mn=sn-phvo-jpql,sn-
vgqsrnsd&ms=au,rdm&mv=m&mv1=1&pl=22&ctier=L&initcwndbps=1168750&svpuc=1&sabr=1&mt=1658133649&fv
ip=5&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier
,svpuc,sabr&sig=AQoQ0QJ8wRQIhAMoq1fU-fdprDjIkRanRHLMJLg-mISEmE1CC7_OqigFVAiBpC6IpBGXN_mWIzoPn8nG
2Ah4WNEIauk5j3_y4uUJooQ%3D%3D&lsparams=mh,mm,mn,ms,mv1,pl,initcwndbps&lsig=AG3C_xAwRQIgaeU1m
BeqHILR9D1Kwm0nihNijdMhZgAFvJjbZKW1FrUCIQDFRyVh1aMUMdMDCNfmmOPgasbsu5dLSDy3JFFsKF1Kw%3D%3D&cpn
=2oxmwItyh1T1zWxv&rn=1

[第二次 rn 2 ctiler L]

- [MLDefaultLoadRetryPolicy scheduleRetryWithBlock] _block + 44
  - [MLDefaultLoadRetryPolicy executeRetryBlockWithRetryID] + 68
    - [MLServerABRLoader continueLoading] + 924
      - [MLServerABRDataTask startWithDelegate delegateQueue] + 88

  HAMCronetDataTask startWithDelegate delegateQueue:
  某次的参数:
    arg1 delegate <MLServerABRDataTask 0x286198f00>
    arg2 delegateQueue = OS_dispatch_queue_serial: MLHAMPlayer[0x280218180] =
{ xref = 34, ref = 5, sref = 2, target = com.apple.root.user.initiated qos.overcommit[0x11050d
080], width = 0x1, state = 0x006000a500001801, enqueued, dirty, max qos 5, draining on 0x1803,
in barrier }

此时:
reqUrl
rn 2时: https://rr2---sn-phvo-jpql.googlevideo.com/videoplayback?ex
pire=1658305786&ei=mWjXYuCLOOucir4P3c6A1AQ&ip=162.253.45.206&id=o-AMnr7yIbmMhFTmywfJSN8wAIx4A4Q
oq0hGO91N7dCzIp&source=youtube&requiressl=yes&mh=hx&mm=31,29&mn=sn-phvo-jpql,sn-vggsknek&ms=au,
rdm&mv=m&mv1=2&pcm2cms=yes&pl=22&ctier=L&initcwndbps=1031250&svpuc=1&sabr=1&mt=1658283891&fvip=
1&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,sv
puc,sabr&sig=AQoQ0QJ8wRgIhANNsZzr74YwYCAd-wykDLNs7C6jNLkd1TSuiVTH1I2v0A1EA4JnoG-7SCvNPKqt91FwCbP
DyHgfX8NtTULRDqp6HnRM%3D&lsparams=mh,mm,mn,ms,mv1,pcm2cms,pl,initcwndbps&lsig=AG3C_xAwRQIgXZ
KQOPCBuuixe3Bdk0qXJbgIxJqr_7LpKHE7xQ1OnQCIQDq4WnBDfATQNkxbpOWAQ_RM4NEThcpb0aqUFqbjUC1vg%3D%3D&
cpn=aCkM2ux19mPWYM5x&rn=2

[第三次 rn 2 rn 3 ctiler A]
  调用顺序同上
  但是URL略有不同
  此时:
```

```

reqUrl
    rn 3时: https://rr2---sn-phvo-jpq1.googlevideo.com/videoplayback?ex-
pire=1658305786&ei=mWjXYuCLOOucir4P3c6AiAQ&ip=162.253.45.206&id=o-AMnr7yIbmMhFTmywfJSN8wAIx4A4Q
oq0hGO91N7dCzIp&source=youtube&requiressl=yes&mh=hx&mm=31,29&mn=sn-phvo-jpq1,sn-vgqsknek&ms=au,
rdu&mv=m&mv1=2&pcm2cms=yes&pl=22&ctier=A&initcwndbps=1031250&svpuc=1&sabr=1&mt=1658283891&fvip=
1&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,sv-
puc,sabr&sig=AOq0QJ8wRgIhANNsZzr74YwYCad-wykDLNs7C6jNLkd1TSuiVTH1I2v0AiEA4JnoG-7SCvNPKqt91FwCbP
DyHgfX8NtTULRDqp6HnRM%3D&lsparms=mh,mm,mn,ms,mv,mv1,pcm2cms,pl,initcwndbps&lsig=AG3C_xAwRQIgXZ
KQOPCBuuyxe3Bdk0qXJbgIxJqr_7LpKHE7xQ1OnQCIQDq4WnBDfATQNkxbpOWAQ_RM4NEThcpb0aqUFqbjUC1vg%3D%3D&
cfn=aCkM2ux19mPWYM5x&rn=2&rn=3

```

ctier L 的 请求 `NSMutableURLRequest` 也返回 `NSError` 出错 的函数调用顺序

```

[HAMPlayerInternal playerLoop] + 1736
  - [HAMBaseTrackRenderer prepareAndReturnError] + 48
    - [HAMPixelBufferVideoTrackRenderer internalPrepareAndReturnError] + 40
      - [HAMVideoDecoderSampleBufferSource prepareAndReturnError] + 80
        - [HAMQueueingRootSampleBufferSource prepareAndReturnError] + 632
          - [HAMDASHSampleBufferSource seekToTime] + 140
            - [MLServerABRController seekToTime] + 36
              - [MLServerABRLoader seekDidOccur] + 148
                - lldb_unnamed_symbol15667$Module_Framework + 112
                  NSError HAMErrorWithCode userInfo
                    某次的参数:
                      arg1 10000
                      arg2 {
                        HAMErrorURLRequest = "<NSMutableURLRequest:
0x28302c0b0> { URL: https://rr2---sn-nx5s7ne1.googlevideo.com/videoplayback?expire=1655754561&
ei=4HqwYt29JNSDkwbAt5vQDg&ip=72.174.144.110&id=o-AJjQ1PwnnZAb4c4XmDyophxk9bxSybJHNgVtg0xPnY5B&s
ource=youtube&requiressl=yes&mh=Vb&mm=31,29&mn=sn-nx5s7ne1,sn-nx57ynsl&ms=au,rdu&mv=m&mv1=2&pl=
19&ctier=L&initcwndbps=1721250&svpuc=1&sabr=1&mt=1655732690&fvip=2&keepalive=yes&fexp=24001373,
24007246,24208264&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=AOq0QJ8w
RQIhAIoI_-R65dCbFS1ymVXG3fMpNeJ05LUNC-t_O1IRl6aJAiBy8tnuzG_Gag61Wwu-Pk1DayQd1MvY4t2MVxvfYVA5mg%
3D%3D&lsparms=mh,mm,mn,ms,mv,mv1,pl,initcwndbps&lsig=AG3C_xAwRQIhAPgKcNRbTnALWCxhsV7zPQ48EmZyL
8VzBjUOibM1y6SfAiAMJA2YCB4AUu1HaS9MuE1aWCkpY5JLkHyY5yooEGQU9Q%3D%3D&cfn=uDxkagKFInGD-plw&rn=1 }"
;
                      HAMRequestNumberLoadTaskKey = 1;
}
                    NSError errorWithDomain code userInfo
                    某次的参数:
                      arg1 "com.google.ios.hamplayer"
                      arg2 10000
                      arg3 {
                        HAMErrorURLRequest = "<NSMutableURLRequest:
0x28302c0b0> { URL: https://rr2---sn-nx5s7ne1.googlevideo.com/videoplayback?expire=1655754561&
ei=4HqwYt29JNSDkwbAt5vQDg&ip=72.174.144.110&id=o-AJjQ1PwnnZAb4c4XmDyophxk9bxSybJHNgVtg0xPnY5B&s
ource=youtube&requiressl=yes&mh=Vb&mm=31,29&mn=sn-nx5s7ne1,sn-nx57ynsl&ms=au,rdu&mv=m&mv1=2&pl=
19&ctier=L&initcwndbps=1721250&svpuc=1&sabr=1&mt=1655732690&fvip=2&keepalive=yes&fexp=24001373,
24007246,24208264&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=AOq0QJ8w
RQIhAIoI_-R65dCbFS1ymVXG3fMpNeJ05LUNC-t_O1IRl6aJAiBy8tnuzG_Gag61Wwu-Pk1DayQd1MvY4t2MVxvfYVA5mg%
3D%3D&lsparms=mh,mm,mn,ms,mv,mv1,pl,initcwndbps&lsig=AG3C_xAwRQIhAPgKcNRbTnALWCxhsV7zPQ48EmZyL
8VzBjUOibM1y6SfAiAMJA2YCB4AUu1HaS9MuE1aWCkpY5JLkHyY5yooEGQU9Q%3D%3D&cfn=uDxkagKFInGD-plw&rn=1 }"
;
                      HAMRequestNumberLoadTaskKey = 1;

```

说明：下面是 `ctier A` 的 `NSURLRequest` 的调用顺序

[case 1]

```

- [MLDefaultLoadRetryPolicy scheduleRetryWithBlock ]
  - [MLDefaultLoadRetryPolicy scheduleRetryWithBlock:]_block + 44
    - [MLDefaultLoadRetryPolicy executeRetryBlockWithRetryID:] + 68
      某次的参数: arg1 3

      MLServerABRLoader continueLoading
        - [MLServerABRLoader continueLoading] + 840
          - [MLServerABRDataLoader createDataTaskForRequest] + 128
            - [MLServerABRDataLoader dataLoadTaskForHTTPBody isPrimaryHostProbing] +
              424
              - [HAMCronetDataLoader taskWithRequest userInfo:] + 140
                - [HAMDataLoadRequest buildURLRequest] + 140
                  NSURLConnection requestWithURL:cachePolicy:timeoutInterval
                  某次的参数:
                    URL = "https://rr1---sn-bvvbax4pcxg-najl.googlevideo.com/videoplayback?expire=1654284469&ei=VAyaYvi5G4eHsfIPqoK1kAg&ip=136.36.116.116&id=o-AD4uH0_eMoAD1cTG1Fs1Uw5dhdYtelrEPN9konw5oPZb&source=youtube&requiressl=yes&mh=_u&mm=31,29&mn=sn-bvvbax4pcxg-najl,sn-o097zns1&ms=au,rdu&mv=m&mv_i=1&pl=20&ctier=A&initcwndbps=1786250&svpuc=1&sabr=1&mt=1654262462&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=A0q0QJ8wRQIgRHLJbeJI4tvvoD9pV1KThMgoBW8xgE27L2nGFFXaX6YCIQC6SEoO尼ui3-50_gkauy6nXCQZN1Es1omMAH7ZoPfyQX3D%3D&lsparams=mh,mm,mn,ms,mv,mvi,pl,initcwndbps&lsig=AG3C_xAwRQIhAKMDmxqaOMD0YMFUgG8otkDKKVPLHSBRuD3yKVpd-DAiB73wqU9tdTRN5zW-yIL3kXDxwNuQ7L4ua7n00tb6YnFAX3D%3D&cpn=WPioc6dOPwVsC718&rn=2&rn=5"
                    cachePolicy NSURLConnectionUseProtocolCachePolicy
                    timeoutInterval 60

      某次的返回值:
        NSMutableURLRequest: 0x280ab2460 {
          URL: https://rr1---sn-8xgp1vo-p5ql.googlevideo.com/videoplayback?expire=1654546559&ei=HwyeYrH5DJWN_9EPouqpqA8&ip=70.110.26.104&id=o-ABMZsKNB0GJ_BmboM7YSytAHB7ASC_Y7GC6j0056wJe&source=youtube&requiressl=yes&mh=j1&mm=31,29&mn=sn-8xgp1vo-p5ql,sn-p5qlsndz&ms=au,rdu&mv=m&mv_i=1&pcm2cms=yes&pl=20&ctier=L&initcwndbps=1490000&svpuc=1&sabr=1&mt=1654524545&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=A0q0QJ8wRgIhAPnD2xwm1j-b9yvruQe1aozyNm_9xaCQvtdIM7n32vA5AiEA4mVxjIugf7425j8X1wfLdVhMyDiZHftRhRE1jHxtWZo%3D&lsparams=mh,mm,mn,ms,mv,mvi,pcm2cms,pl,initcwndbps&lsig=AG3C_xAwRQIgd3ooIz7fpF6JTB11op1dSL1WIpx4L5FaBEqhdEZBLwCIQChh18q1sc8T2XEIwlQBI809cF_D12FPVLjVCNRHhRPCQ%3D%3D&cpn=sjOhS6zpDL-0E-pP&rn=1
        }

        CFNetwork`__lldb_unnamed_symbol1231$$CFNetwork + 64

        NSURLConnection initWithURL cachePolicy timeoutInterval

        参数 case 1:
          URL = "https://rr1---sn-bvvbax4pcxg-najl.googlevideo.com/videoplayback?expire=1654284469&ei=VAyaYvi5G4eHsfIPqoK1kAg&ip=136.36.116.116&id=o-AD4uH0_eMoAD1cTG1Fs1Uw5dhdYtelrEPN9konw5oPZb&source=youtube&requiressl=yes&mh=_u&mm=31,29&mn=sn-bvvbax4pcxg-najl,sn-o097zns1&ms=au,rdu&mv=m&mv_i=1&pl=20&ctier=A&initcwndbps=1786250&svpuc=1&sabr=1&mt=1654262462&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=A0q0QJ8wRQIgRHLJbeJI4tvvoD9pV1KThMgoBW8xgE27L2nGFFXaX6YCIQC6SEoO尼ui3-50_gkauy6nXCQZN1Es1omMAH7ZoPfyQX3D%3D&lsparams=mh,mm,mn,ms,mv,mvi,pl,initcwndbps&lsig=AG3C_xAwRQIhAKMDmxqaOMD0YMFUgG8otkDKKVPLHSBRuD3yKVpd-DAiB73wqU9tdTRN5zW-yIL3kXDxwNuQ7L4ua7n00tb6YnFAX3D%3D&cpn=WPioc6dOPwVsC718&rn=2&rn=5"
            cachePolicy NSURLConnectionUseProtocolCachePolicy
            timeoutInterval 60
  }
}

```

```
r=1&mt=1654262462&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiresl,ctier,svpuc,sabr&sig=AOq0QJ8wRQIgRHLJbeJI4tvvoD9pV1KThMgoBW8xgE27L2nGFFXaX6YCIQC6SEoNiui3-50_gkauy6nXCQZN1Es1omMAH7ZoPfyQ%3D%3D&lsparms=mh,mm,mn,ms,mv,mvi,pl,initcwndbps&lslsigs=AG3C_xAwRQIhAKMDmxqaOMD0YMFUgG8otkDKKVPLHSBRuD3yKVpd-DAiB73wqU9tdTRN5zW-yIL3kXDxwNuQ7L4ua7n00tb6YnFA%3D%3D&cfn=WPloc6dOPwVsC718&rn=2&rn=5"
cachePolicy=NSURLRequestUseProtocolCachePolicy
icy
timeoutInterval 60

参数 case 2:
URL="https://rr1---sn-8xgp1vo-p5ql.googlevideo.com/videoplayback?expire=1654546559&ei=HwyeYrH5DJWN_9EPouqpqA&ip=70.110.26.104&id=o-ABMZsKNB0GJ_BmboM7YSytnAHB7ASC_Y7GC6j0056wJe&source=youtube&requiresl=yes&mh=j&mm=31,29&mn=sn-8xgp1vo-p5ql,sn-p5qlsndz&ms=au,rdu&mv=m&mvi=1&pcm2cms=yes&pl=20&ctier=A&initcwndbps=1490000&svpuc=1&sabr=1&mt=1654524545&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiresl,ctier,svpuc,sabr&sig=AOq0QJ8wRgIhAPnD2xwm1J-b9yvrue1aozyNm_9xaCQvtdIM7n32vA5AiEA4mVxjIUgf7425J8X1wfLdVhMyDizHftRRE1jHxtWZo%3D&lsparms=mh,mm,mn,ms,mv,mvi,pcm2cms,pl,initcwndbps&lslsigs=AG3C_xAwRQIgd3ooIz7fpF6JTB11opldSL1WIpx4L5FaBEqhdEZBLwCIQChh18q1sc8T2XEIwlQBI809cFDI2FPVLjVCNRHhRPCQ%3D%3D&cfn=sjOhS6zpDL-OE-pP&rn=2&rn=5"
cachePolicy=NSURLRequestUseProtocolCachePolicy
icy
timeoutInterval 60

return NSMutableURLRequest: 0x28063c790 {
    URL: https://rr1---sn-bvvbax4pcxg-naj1.googlevideo.com/videoplayback?expire=1654284469&ei=VAyaYv15G4eHsfIPqoK1kAg&ip=136.36.116.116&id=o-AD4uH0_eMoAD1cTG1Fs1Uw5dhdytelrEPN9konw5oPzb&source=youtube&requiresl=yes&mh=_u&mm=31,29&mn=sn-bvvbax4pcxg-naj1,sn-o097zns1&ms=au,rdu&mv=m&mvi=1&pl=20&ctier=A&initcwndbps=1786250&svpuc=1&sabr=1&mt=1654262462&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,ei,ip,id,source,requiresl,ctier,svpuc,sabr&sig=AOq0QJ8wRQIgRHLJbeJI4tvvoD9pV1KThMgoBW8xgE27L2nGFFXaX6YCIQC6SEoNiui3-50_gkauy6nXCQZN1Es1omMAH7ZoPfyQ%3D%3D&lsparms=mh,mm,mn,ms,mv,mvi,pl,initcwndbps&lslsigs=AG3C_xAwRQIhAKMDmxqaOMD0YMFUgG8otkDKKVPLHSBRuD3yKVpd-DAiB73wqU9tdTRN5zW-yIL3kXDxwNuQ7L4ua7n00tb6YnFA%3D%3D&cfn=WPloc6dOPwVsC718&rn=2&rn=5
}

[case 2]

[HAMPlayerInternal playerLoop] + 1736
- [HAMBaseTrackRenderer prepareAndReturnError] + 48
- [HAMSBARAudioTrackRenderer internalPrepareAndReturnError] + 40
[HAMQueueingRootSampleBufferSource prepareAndReturnError] + 704
- [HAMDASHSampleBufferSource prepareAndReturnError] + 100
- [MLServerABRController prepareAndReturnError] + 40
- [MLServerABRLoader continueLoading] + 840
- [MLServerABRDataLoader createDataLoadTaskForRequest] + 128
- [MLServerABRDataLoader dataLoadTaskForHTTPBody isPrimaryHostPr
obing] + 424
- [HACronetDataLoader taskWithRequest userInfo] + 140
- [HADataLoadRequest buildURLRequest] + 140
    NSURLRequest requestWithURL cachePolicy timeoutInte
rval
CFNetwork`__lldb_unnamed_symbol1231$$CFNetwork +
64
    NSURLRequest initWithURL cachePolicy timeout
tInterval:
```

说明：下面是 `ctier A` 的 `response = NSHTTPURLResponse` 的调用顺序

```

HAMCronetDataLoadTask startWithDelegate delegateQueue: ]_block_block + 36 = sub_194D978
    __lldb_unnamed_symbol110205$$Module_Framework + 40 = sub_38FCE24
    __lldb_unnamed_symbol171000$$Module_Framework + 28 = sub_3DF8048
    __lldb_unnamed_symbol170908$$Module_Framework + 52 = sub_3DF60E8
        Cronet_UrlRequestCallback_OnResponseStarted_194CCE4 = __lldb_unnamed_symbol156
76$$Module_Framework + 80 = sub_194CCE4
            generateNewHttpResponse_194D6D8 = __lldb_unnamed_symbol15681$$Module_Frame
work + 376 = sub_194D6D8
                NSHTTPURLResponse initWithURL statusCode HTTPVersion headerFields:
                    某次参数:
                        url "https://rr1---sn-8xgp1vo-p5ql.googlevideo.com/videoplaybac
k?expire=1654546559&ei=HwyeYrH5DJWN_9EPouqpqA8&ip=70.110.26.104&id=o-ABMzsKNB0GJ_BmboM7YSytAH
7ASC_Y7GC6j0056wJe&source=youtube&requiresl=yes&mh=j1&mm=31,29&mn=sn-8xgp1vo-p5ql,sn-p5qlsndz&
ms=au,rdu&mv=m&mvi=1&pcm2cms=yes&pl=20&ctier=A&initcwndbps=1490000&svpuc=1&sabr=1&mt=1654524545
&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,e1,ip,id,source,requiresl,ct
ier,svpuc,sabr&sig=A0q0QJ8wRgIhAPnD2xwm1J-b9yvruQe1aozyNm_9xaCQvtdIM7n32vA5AiEA4mVXjIUgf7425J8X
lwfLdVhMyDiZHFtRhRE1jHxtWZo%3D&lsparams=mh,mm,mn,ms,mv,mi,pcm2cms,pl,initcwndbps&lsig=AG3C_xAw
RQIgd3ooIz7fpF6JTB11op1dSL1WIpx4L5FaBEqhdEZBLwCIQChh18q1sc8T2XEIwlQBI809cF_DI2FPVLjVCNRhRPCQ%
3D%3D&cpn=sjOhS6zpDL-OE-pP&rn=2&rn=3",
                        statusCode 200,
                        HTTPVersion 0x0000000000000000,
                        headerFields 4 key value pairs
                    某次的返回值:
                        NSHTTPURLResponse 0x280ee1d00 <
                            URL https://rr1---sn-8xgp1vo-p5ql.googlevideo.com/videoplaybac
k?expire=1654546559&ei=HwyeYrH5DJWN_9EPouqpqA8&ip=70.110.26.104&id=o-ABMzsKNB0GJ_BmboM7YSyt
nAHB7ASC_Y7GC6j0056wJe&source=youtube&requiresl=yes&mh=j1&mm=31,29&mn=sn-8xgp1vo-p5ql,sn-p5qls
ndz&ms=au,rdu&mv=m&mvi=1&pcm2cms=yes&pl=20&ctier=A&initcwndbps=1490000&svpuc=1&sabr=1&mt=165452
4545&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,e1,ip,id,source,requiresl,ct
ier,svpuc,sabr&sig=A0q0QJ8wRgIhAPnD2xwm1J-b9yvruQe1aozyNm_9xaCQvtdIM7n32vA5AiEA4mVXjIUgf742
5J8XlwfLdVhMyDiZHFtRhRE1jHxtWZo%3D&lsparams=mh,mm,mn,ms,mv,mi,pcm2cms,pl,initcwndbps&lsig=AG3C
_xAwRQIgd3ooIz7fpF6JTB11op1dSL1WIpx4L5FaBEqhdEZBLwCIQChh18q1sc8T2XEIwlQBI809cF_DI2FPVLjVCNRhR
PCQ%3D%3D&cpn=sjOhS6zpDL-OE-pP&rn=2&rn=3
                            }
                            {
                                Status Code 200,
                                Headers {
                                    Connection = "keep-alive"
                                    "Content-Length" = 0
                                    "Content-Type" = "application/vnd.yt-ump"
                                    Server = "atomos-tproxy 1.0"
                                }
                            }
                        
```

触发了 `HAMCronetDataLoadTask startWithDelegate delegateQueue:` 当第一次 `ctier A` 的 `url` 时的调用顺序：

```

- [MLDefaultLoadRetryPolicy scheduleRetryWithBlock ]_block + 44
- [MLDefaultLoadRetryPolicy executeRetryBlockWithRetryID ] + 68
    某次参数:
        arg1 2
        - [MLServerABRLoader continueLoading] + 924
            - [MLServerABRDataTask startWithDelegate delegateQueue ] + 88
                某次的参数:
                    arg1 = MLServerABRDataTask 0x281226d00
                    arg2 = OS_dispatch_queue_serial MLHAMPlayer[0x282d34f00] = { xref = 33, ref
= 6, sref = 2, target = com.apple.root.user initiated qos overcommit[0x10c651080], width = 0x1,
state = 0x006000a50001a401, enqueued, dirty, max qos 5, draining on 0x1a403, in barrier}

```

下面是 ctier A 的 response = NSHTTPURLResponse 在生成了之后，被调用到 statusCode 时的调用顺序

```

- [MLPlayerItemEventCenter networkRequestDidReceiveResponseWithEvent ] + 204
- [MLQOEController networkRequestDidReceiveResponseWithEvent ] + 272
    NSHTTPURLResponse statusCode
        return 200;

```

说明：下面是 ctier A 的response后的，始终retry重发请求的相关的调用逻辑

[case 1]

```

cancelCronetDataLoadTask_194BF8
    reportDataTaskError_194DA94 = sub_194DA94 = __lldb_unnamed_symbol15689$$Module_Framework + 40
        - [MLServerABRDataTask dataLoadTask didCompleteWithError ] + 368
            - [MLServerABRLoader dataLoadTask didCompleteWithError ] + 260
                参数
                    case 1:
                        arg1 0x0000000281aaa490
                        arg2 domain: "com.google.ios.hamplayer" - code 10000
                    case 2:
                        arg1 0x0000000280597a50
                        arg2 domain: "com.google.ios.hamplayer" - code 6009

                MLServerABRLoader updateLoadRetryPolicyOnLoadCompletionWithError

```

[case 2]

```

[HAMCronetDataLoadTask startWithDelegate delegateQueue ]_block_block + 36
    __lldb_unnamed_symbol110205$$Module_Framework + 40
    __lldb_unnamed_symbol171023$$Module_Framework + 32
    __lldb_unnamed_symbol170906$$Module_Framework + 84
        Cronet.UrlRequestCallback_OnRedirectReceived_194C9D8 = __lldb_unnamed_symbol15
675$$Module_Framework + 620
            - [MLServerABRDataTask dataLoadTask didCompleteWithError ] + 368
                - [MLServerABRLoader dataLoadTask didCompleteWithError ] + 260
                    MLServerABRLoader updateLoadRetryPolicyOnLoadCompletionWithError

```

[case 3]

找到 死循环 多次重试的调用关系了：

```
callHAMTimerFire_199CFB8 = -[HAMTimer initWithQueue startNanoseconds intervalNanoseconds leeway
Nanoseconds repeats block ]_block + 36
    -[HAMTimer fire] + 52
        callMLOnesieConnectionOpenerOnPingTimer_1039FB8 = sub_1039FB8 = -[MLOnesieConnectionOpener
scheduleNextRequestUsingStartDelay ]_block + 36
            -[MLOnesieConnectionOpener onPingTimer] + 180
                -[HACronetDataLoadTask startWithDelegate delegateQueue ] + 932
                    tryResetTimeoutTimer_194C704 = sub_194C704 = __lldb_unnamed_symbol15674$$Module_Framework + 200
                        -[HAMTimer timerWithQueue startNanoseconds leewayNanoseconds block ] +
104
                            -[HAMTimer initWithQueue startNanoseconds intervalNanoseconds leeway
Nanoseconds repeats block ]
```

.....

```
[HAMTimer initWithQueue startNanoseconds intervalNanoseconds leewayNanoseconds repeats block ]
    -[HAMTimer initWithQueue startNanoseconds intervalNanoseconds leewayNanoseconds repeats blo
ck ]_block + 36
        -[HAMTimer fire] + 52
            callMLOnesieConnectionOpenerOnPingTimer_1039FB8 = sub_1039FB8 = -[MLOnesieConnect
ionOpener scheduleNextRequestUsingStartDelay:]_block + 36
                -[MLOnesieConnectionOpener onPingTimer] + 180
                    -[HACronetDataLoadTask startWithDelegate delegateQueue:] + 648
某次参数：
    arg1 = MLOnesieConnectionOpener 0x2829dde00
    delegateQueue = OS_dispatch_queue_serial queue_serial[0x283868800]
= { xref = 8, ref = 5, sref = 2, target = com.apple.root.default qos overcommit[0x104664f80], w
idth = 0x1, state = 0x006000a500012805, enqueued, dirty, max qos 5, draining on 0x12807, in bar
rier}>
```

核心代码

```
sub_3DF9004(sub_194C27C);
```

```
Cronet_Executor_ExecuteFunc_194C27C = sub_194C27C = __lldb_unnamed_sym
bol15668
    -[HACronetDataLoadTask startWithDelegate delegateQueue ]_block_blo
ck + 36 = sub_194D978
        __lldb_unnamed_symbol110205$$Module_Framework + 40 = sub_38FCE
24
        __lldb_unnamed_symbol171000$$Module_Framework + 28 = sub_3
DF8048
        __lldb_unnamed_symbol170908$$Module_Framework + 52 = s
ub_3DF60E8
        Cronet.UrlRequestCallback_OnResponseStarted_194CCE4
= sub_194CCE4 = __lldb_unnamed_symbol15676$$Module_Framework + 124
            -[MLServerABRDataTask dataLoadTask didRecei
veResponse:] + 424
                -[MLServerABRLoader dataLoadTask didReceive
Response:]
```

某次的参数：

```
arg1 = MLServerABRDataTask 0x28
```

```

06a7060>,
arg2 = NSHTTPURLResponse 0x280ee1d00
> {
    URL: https://rr1---sn-8xgp1
vo-p5ql.googlevideo.com/videoplayback?expire=1654546559&ei=HwyeYrH5DJWN_9EPouqpqA8&ip=70.110.26
.104&id=o-ABMZsKNB0GJ_BmboM7YSytnAHB7ASC_Y7GC6j0056wJe&source=youtube&requiressl=yes&mh=j1&mm=3
1,29&mn=sn-8xgp1vo-p5ql,sn-p5qlsndz&ms=au,rdu&mv=m&mvi=1&pcm2cms=yes&pl=20&ctier=A&initcwndbps=
1490000&svpuc=1&sabr=1&mt=1654524545&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=
expire,ei,ip,id,source,requiresl,ctier,svpuc,sabr&sig=A0q0QJ8wRgIhAPnD2xwm1J-b9yvruQe1aozyNm_9
xaCQvtdIM7n32vA5AiEA4mVXjIUgf7425J8X1wfLdVhMyDizHftRhRE1jHxtWZo%3D&lsparms=mh,mm,mn,ms,mv,mvi,
pcm2cms,pl,initcwndbps&lsig=AG3C_xAwRQIgd3ooIz7fpF6JTB11opldSL1WIpx4L5FaBEqhdEZBLwCIQChh18q1sc
8T2XEIwlQBI809cF_DI2FPVLjVCNRHhRPCQ%3D%3D&cpn=sjOhS6zpDL-OE-pP&rn=2&rnr=3
}
{
    Status Code: 200, Headers {
        Connection = "keep-alive"
        "Content-Length" = 0
        "Content-Type" = "application/vnd.yt-ump"
        Server = "atomos-tproxy
1.0"
    }
}

NSError HAMErrorWithURLRequest userInfo
request:
某次的参数:
response <NSHTTPURLResponse 0x
281464760>,
userInfo {
    HAMRequestNumberLoadTaskKey
= 2;
},
request <NSMutableURLRequest 0
x2810f1090>

v22 = objc_msgSend((void *)v7, "HAM
ErrorWithCode:userInfo:", v20 6000, v14 userInfoDict);
NSError HAMErrorWithCode userInfo
fo:
某次的参数:
hamErrCode 6000,
userInfo dict

return (id)objc_msgSend((vo
id *)self, "errorWithDomain:code:userInfo:", CFSTR("com.google.ios.hamplayer"), a3, a4);

某次的返回值:
Error Domain com.google.
ios.hamplayer
Code 6000 "(null)"
UserInfo {
    HAMErrorHTTPURLRespon
onse <NSHTTPURLResponse 0x281ceda60> {
    URL: https
}

```

```

//rr5---sn-q4flrnee.googlevideo.com/videoplayback?expire=1654110110&ei=PmOXYpHwG-6Rlu8P0L09yAQ&
ip=72.179.224.229&id=o-ANJ-ePUG7jWZMQ_icM7pjIuiw8IEgTemHHlekQ1ItK_&source=youtube&requiressl=y
es&mh=aT&mm=31,29&mn=sn-q4flrnee,sn-q4f16ns7&ms=au,rdu&mv=m&mv1=5&p1=21&ctier=A&initcwndbps=148
0000&svpuc=1&sabr=1&mt=1654088228&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=exp
ire,ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=AOq0QJ8wRAIgTokhx1KrvIZN4PZHmq14CCoCk-_Bdg1_
_WTIrlx1Y7ICIGr2dSn6txZUrrMdum-VwuQ8TbwSK-HTPVVZB28aPwO-&lsparams=mh,mm,mn,ms,mv,mvi,p1,initc
wndbps&lsig=AG3C_xAwRQIgGW-eIbWvW751rbGlodLzWTMVtDL7Kt1xEW-gRO1_6_kCIQDwf7k8v2jZdCDAY9KK1Cf_k_T4
IsKDol2nm-49RdhBfQ%3D%3D&cfn=F98L8S2i9bcH06us&rn=2&rn=3
}

{
    Status Code: 200
    Headers {
        Connect
        "Content-Type": "text/html; charset=UTF-8"
        Server: "atomos-tproxy/1.0"
    }
},
HAMRequestNumberLo
dTaskKey=3,
HAMErrorURLRequest <
NSMutableURLRequest: 0x281886580> {
    URL https://rr5---sn-q4flrnee.googlevideo.com/videoplayback?expire=1654110110&ei=PmOXYpHwG-6Rlu8P0L09yAQ&ip=7
2.179.224.229&id=o-ANJ-ePUG7jWZMQ_icM7pjIuiw8IEgTemHHlekQ1ItK_&source=youtube&requiressl=yes&m
h=aT&mm=31,29&mn=sn-q4flrnee,sn-q4f16ns7&ms=au,rdu&mv=m&mv1=5&p1=21&ctier=A&initcwndbps=1480000
&svpuc=1&sabr=1&mt=1654088228&fvip=2&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire,
ei,ip,id,source,requiressl,ctier,svpuc,sabr&sig=AOq0QJ8wRAIgTokhx1KrvIZN4PZHmq14CCoCk-_Bdg1_WTI
Irlx1Y7ICIGr2dSn6txZUrrMdum-VwuQ8TbwSK-HTPVVZB28aPwO-&lsparams=mh,mm,mn,ms,mv,mvi,p1,initcwndbp
s&lsig=AG3C_xAwRQIgGW-eIbWvW751rbGlodLzWTMVtDL7Kt1xEW-gRO1_6_kCIQDwf7k8v2jZdCDAY9KK1Cf_k_T4IsKD
ol2nm-49RdhBfQ%3D%3D&cfn=F98L8S2i9bcH06us&rn=2&rn=3
}
}
}

```

当去掉广告过滤时，调用到 `MLServerABRLoader onesieDataLoader didCompleteChunksWithError:` 的顺序：

```

[MLOneisieChunkLoadTask initWithRange chunks dataBuffer cache delegate queue latencyLogger]
    sub_10380F8 =-[MLOneisieChunkLoadTask initWithRange chunks dataBuffer cache delegate queue
latencyLogger]_block
    sub_10381F8 = [MLOneisieChunkLoadTask initWithRange chunks dataBuffer cache delegate qu
e latencyLogger]_block_block + 344 =
        [HAMBaseChunkLoadTask didCompleteWithError:] + 412
        -[MLServerABROneisieDataLoader chunks didCompleteLoadingWithError:] + 92
        MLServerABRLoader oneisieDataLoader didCompleteChunksWithError
参数举例：
    [case 1]:
        arg1 0x0000000282b42000,
        arg2 1 element,
        arg3 domain: "com.google.ios.hamplayer" - code: 10000
    [case 2]:

```

```

    arg1 <MLServerABRLoader> 0x282bd5e80>,
    arg2 <__NSSingleObjectArrayI 0x2807d1510>(
        <HAMMP4InitializationChunk> 0x282bc3b80>
    )
    arg3 NULL

```

下面是：MLServerABRLoader initWithQueue 的调用顺序

```

[GPCPromise resolveChild ]_block + 128
    -[YTVideoResolver registerCompletionBlocksOnPlaybackDataLoader]_block + 64
        -[YTVideoResolver didLoadVideoWithPlaybackData:] + 108
            -[YTVideoResolver changeToState:] + 92
                -[YTLocalPlaybackController videoResolver didChangeFromState toState:] + 488
                    -[YTLocalPlaybackController startPlaybackWithVideoResolver internallyPrefetchedVideoController:] + 2184
                        YTAdsControlFlowPlaybackCoordinator startPrerollAdBreak
                            -[YTAdsControlFlowPlaybackCoordinator startPrerollAdBreak] + 52
                                -[YTAdsControlFlowPlaybackCoordinator doneWithPlayerBytesSlotWithBreakType:] + 80
                                    -[YTLocalPlaybackController adsPlaybackCoordinator didFinishBreakWithBreakType:] + 68
                                        -[YTSingleVideoSequencer activateContentSequence] + 224
                                            YTSingleVideoSequencer activateVideoController(reloadContext:) + 248
                                                -[YTSingleVideoController initializeAndLoadWithViewportSizeProvider playerVisibility reloadContext:] + 860
                                                    -[MLHAMQueuePlayer load] + 220
                                                        -[MLHAMPlayerItem load] + 3088

MLServerABRLoader initWithQueue data
    aLoader playerEventCenter playerItemEventCenter streamingData mediaCommonConfig hamplayerConfig
    onesieVideoData QOEController loadRetryPolicy isLiveSource livePlayerConfig windowedLiveConfig
    metadata

说明: 某次调试时参数:
    arg1 <OS_dispatch_queue_serializer>
    dataLoader = HAMCronetDataLoader
    playerEventCenter = (null),
    playerItemEventCenter = MLPlayerItemEventCenter
    streamingData = MLStreamingData
    mediaCommonConfig = YTIMediaCommonConfig
    dynamic_readahead_config {
        max_read_ahead_medi
        min_read_ahead_medi
        read_ahead_growth_r
    }
    a_time_ms: 8000
    a_time_ms: 15000
    ate_ms 1000
}

```

```

    media_ustreamer_request
_config {
    video_playback_use_
ump true
    video_playback_ustr
eamer_config: "\n\317\003\n\326\002\010\000\020\200\005\030\350\002%\372~\252>\000\000\200?5=\n\227?h\001r9\n\022mfs2_cmfs_v3_1_043\022\037\n\035\n\014device_model\022\r\n\013\n\tiphone9,1\030\000 \001r\026\n\022mfs2_cmfs_v3_1_043\030\000x\350\002\200\001\001\250\001\001\265\001\366(\334?\240\002\350\002\332\002\\020\260\352\001\030\250F \240\234\001(\33060\230uH\001P\001X\001h\001p\210\020\001\364\003\270\001\001\300\001\001\340\001\003\370\001\001\210\002\001\220\002\001\230\002\014\240\002\001\250\002\001\260\002\001\270\002\001\300\002\001\310\002\001\320\002\002\340\002\001\350\002\002\200\003\002\220\003\001\370\265\221\345\014\001\372\002-\010\014\020\030\0302 2-\000\000pB5\000\000\214B@\001H\001P\nX\ne\000\000\200@h\300pp\001\210\001\001\315\001\000\000\200?\202\003\000\210\003\001\220\003\001\240\003\001\250\003\001\260\003\003\310\003\001\320\003\001\330\003\001\340\003\220N\270\004\001\312\004\035\n\024\010\300>\020\230u\030\350\007%\000\000\000(\000\000@\001\020\340\324\003\030\320\017\360\004\001\370\004\001\030\001 \0012\014\010\211\001\020\257\267\220\271\236\212\367\0022\014\010\210\001\020\314\203\356\267\236\212\367\0022\014\010\207\001\020\343\206\301\271\236\212\367\0022\014\010\206\001\020\366\305\356\267\236\212\367\0022\014\010\205\001\020\306\327\220\271\236\212\367\0022\014\010\240\001\020\236\312\314\270\236\212\367\0022\014\010\213\001\020\217\354\253\267\236\212\367\0022\014\010\214\001\020\266\312\253\267\236\212\367\002:\000\022L\000\362\330\357\3010E\0021\000\230\215 '\203G\270\247?\202\342\310\337E\255\271;\354\266\367\210\342\0247\233~\206\36121\177\363\200\002 ^\262\221\336\225\324\217B\213H\327r\217\352\327]\335\253E*Y_\210\216\tI\336\027bA\274\177\032\002e1"
}
server_readahead_config
{
    enable true
    next_request_policy
{
    target_audio_re
adahead_ms: 120000
    target_video_re
adahead_ms: 120000
}
}
use_server_driven_abr
true
{
    me_ms 0
    enable_hostFallback
    primary_probing_delay_ms 5000
    max_failure_attempt
    s_beforeFallback 2
}
},
hamplayerConfig <YTHamplay
erConfig 0x110ca4440>: {
    enable_on_cellular true
    qos_class IOS_QOS_CLASS
    decode_qos_class IOS_Q
}

```

```

OS_CLASS_USER_INITIATED

    video_track_renderer {
        hamplayer_pixel_buf
        min_frame_queue
        target_frame_qu
        max_frame_fall_
        queue_more_fram
    }

    stall_predictor {
        hamplayer_default_s
        max_buffer_read
        min_buffer_read
        bandwidth_tweak
        bandwidth_tweak
    }

    live {
        enable_accurate_seek
        accurate_seek_retry
        enable_accurate_see
        enable_ss_dai_emsg_
        enable_ss_dai_updat
        fix_post_live_head_
        finish_live_playbac
    }

    async_decode: true
    network_stats_sampler_c
    onfig {
        minimum_sample_size
        minimum_sample_dura
        target_sample_durat
        ion_ms 400
    }

    decode_teardown_on_deco
    decode_wait_for_frames_

```

```

before_teardown true
true
inate true
2000
: 500
10000
1000
}
decode_separate_queues
decode_no_wait_for_term
resync_policy_config {
target_readahead_ms:
min_required_time_ms
max_elapsed_time_ms:
seek_tolerance_ms
}
audio_abr_config {
downshift_scalar 3
downshift_constant
0
upshift_scalar 6
upshift_constant 0
max_downshift_readahead
min_upshift_readahead
upshift_replace_medium
buffer_target_readahead
buffer_max_size_bytes
buffer_trim_behind
buffer_trim_ahead
max_chunks_per_request
est 0
sync_readahead 10
stun_duration 0
max_consecutive_error
ors 0
min_upshift_replace
_low_memory_warn_buffer_size
_low_memory_critical
_low_memory_buffer_size
size_cool_down_ms 0
disable_hdr_in_low_power_mode: true
observe_network_activity_controller false
}
video_abr_config {
downshift_scalar 1
}

```

```

downshift_constant

48000
upshift_scalar 1
upshift_constant 4

8000
oversend_factor 1.

18
max_downshift_reada

head_ms 30000
min_upshift_readahe

ad_ms 0
upshift_replace_med

ia false
local_max_bitrate_r

eadahead 15
buffer_target_reada

head 8
buffer_max_size_byt

es 751619276
buffer_trim_behind

60
buffer_trim_ahead

120
max_chunks_per_requ

est 0
min_readahead_for_a

verage_bitrate 10000
load_extra_formats

false
sync_readahead 10
stun_duration 10
use_high_replicatio

n_formats_while_stunned true
max_consecutive_err

ors 4
min_upshift_replace

_chunks_readahead_ms 5000
max_media_seconds_p

er_request 11
low_memory_warn_buf

fer_size_bytes 751619276
low_memory_critical

_buffer_size_bytes 751619276
low_memory_buffer_s

ize_cool_down_ms 0
disable_hdr_in_low_

power_mode true
hpq_oversend_factor

1.72
observe_network_act

ive_controller false
}

true
enable_airplay_audio t

chunk_loader_config {
    enable_fallback_host
}

```

```

    : true
    max_failure_attempt
    s_beforeFallback 3
    primary_probing_del
    ay 5000
    treat_no_media_as_c
    cancellation true
}
sbar_audio_track_render
er {
    samples_per_buffer
10
    flush_on_return_fro
    m_background true
    trim_priming false
    feed_media_data tr
ue
    handle_automatic_f1
    ush true
    renderer_max_retry_
count 2
    use_all_tracks_for_
playability_status false
}
use_multiple_periods_pl
ayer true
load_retry_config {
    max_non_network_err
}
render_view_type HAMPL
server_abr_config {
    fallback_on_error
    fallback_disable_se
enable_local_streams
    log_partial_chunk_e
true
    max_failure_attemp
rver_abr_duration_seconds 5
    seek_through_medial
    : false
    continue_loading_ti
viction false
    ts 5
    continue_loading_ti
    log_partial_chunk_e
    max_failure_attemp
    ib_when_receiving_partial_chunk false
    cover_chunk_discont
mer_interval_ms 1000
    seek_through_medial
    continue_loading_ti
    continue_loading_ti
    max_failure_attemp
    mer_leeway_ms 100
    cover_chunk_discont
inuity true
    keep_stall_for_live
    _when_close_to_seekable_end false
    allow_multiple_serv
er_seek true
    use_reload_context_
}

```

```

for_fallback true
}

use_new_sample_buffer_s
stop_track_renderer_bef
log_debug_details_on_lo
deprecate_stall_predict
player_loop_timer_inter
player_loop_timer_leewa
offline_config {
    downloader_apply_a
    downloader_apply_ab
}
select_default_track_fo
use_reload_context_requ
},
onesieVideoData (null),
QOEController=MLQOEPingControl
ler 0x115df0ef0,
retryPolicy 0x28385b160,
loadRetryPolicy=MLDefaultLoadR
isLiveSource False,
livePlayerConfig (null),
windowedLiveConfig (null),
metadataRelay=MLHAMManifestles
sMetadataCollector 0x281551f80,
delegate <MLHAMPlayerItem: 0x12
2203240>,
latencyLogger=YTWatchLatencyTi
ckLogger 0x2801ea200,
watchEndpointUstreamerConfig (n
ull),
periodID 0,
firstRequestNumber 1,
cache (null)

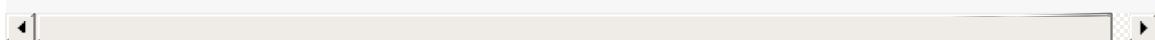
```

## 常见url

### initplayback

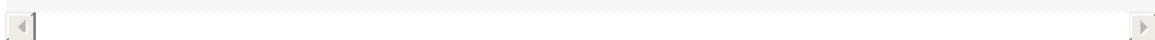
- initplayback
  - 发起（正常视频或广告视频的）视频数据请求的

```
https://rr3---sn-vgqsknek.googlevideo.com/initplayback?source youtube&orc=1&oeis=1&c=IOS
&oss=1&oda=1&load=5500&ovd=5500&ooad=11000&oavd=11000&ocs=700&oputc=1&oses=1&ofpcc=1&osbr=1&osnz=1&msp=1&odeak=1&odepv=1&osfc=1&id=553003fe2eb8727c&initcwndbps=1680000&mt=1655903589&ack=1&cpn=jcLGzlc6c8LsaadH&rn=1&opr=1&pvi=160,133,134,135,136,137,298,299,264&pai=139,140
```



- redirector: 没啥用

```
https://redirector.googlevideo.com/initplayback?source youtube&itag=0&pvi=0&pai=0&owc=y&es&cmo=rnp%3Drr&id=0000000000002409
```



### videoplayback

- videoplayback
  - ctier=L

```
https://rr2---sn-a5meknsy.googlevideo.com/videoplayback?expire=1657110332&ei=3CrFYrS3B8
&q1kwbDk6m4Dg&ip=76.170.156.167&id=o-AN990KkuGRjk0tpS0zvkbCippQdwVTosh4W040ICinJ&source=
youtube&requiressl=yes&hcs=ir,&mh=v4&mm=31,26&mn=sn-a5meknsy,sn-q4flrnez&ms=au,onr&mv=m
&mvi=2&pl=17&rmhost=rr3---sn-a5meknsy.googlevideo.com,&ctier=L&initcwndbps=2142500&svpuc=
1&sabr=1&mt=1657088457&fvip=5&keepalive=yes&fexp=24001373,24007246&c=IOS&sparams=expire
,&ei,&ip,&id,&source,&requiressl,&ctier,&svpuc,&sabr,&sig=A0q0QJ8wRQIhAPXZb_XYMsCV_fh05opM85fJbw
UAdMIXGhBCYe3yDH16AiBAwpSbmd53tjQZadmgq3rLdyX-Nd5447nkXP6NBPXNHQ%3D%3D&lsparams=hcs,mh,
mm,mn,ms,mv,pl,rmhost,initcwndbps&lsig=AG3C_xAwRQIhAP22VWdWbIIxrvp91Cm3NMY8QQ4TS3Ur
FwJFxyxZH7C2AiAhnv2jgK4Nj0SwXmhuddVSWSRvX8iGYRtWZJsfBG03XQ%3D%3D&cpn=5ofB6P60RRZhJUE8&rn=
7&probe=1
```



- ctier/L

```
https://r2---sn-q4flrnez.googlevideo.com/videoplayback/expire/1657108318/ei/_iLFYofvH72
QsfIPy56rkA0/ip/76.170.156.167/id/o-AIPVOknUnDee3HugNutm3ybodjkv230I_U-FbZP3qt_k/source
/youtube/requiressl/yes/mh/sw/mm/31,26/mn/sn-a5meknsz,sn-q4flrnez/ms/au,onr/mv/m/mv1/5/
pl/17/ctier/L/initcwndbps/2056250/svpuc/1/sabr/1/mt/1657086303/fvip/2/keepalive/yes/fex
p/24001373,24007246/c/IOS/sparams/expire,ei,ip,id,source,requiressl,ctier,svpuc,sabr/si
g/A0q0QJ8wRgIhALUv6hCXG_Dq8gwMkYfOxUAch6KUBSZBcf6CikscQYp1AiEAkcLG_OxZHTcNs4tzkR109Awqc
VwYXQ374k3kyGOPZ-0 /lsparams/mh,mm,mn,ms,mv,pl,initcwndbps/lsig=AG3C_xAwRAIgH8N-m_K
KOvrdVXq7KWBgjV27TWKPn48MQjYw9qWlP1ACIFk-KqOBGsauP4fBIHDVfLC6w78LrlgHmuKzWXwKgVIt/cpn/v
TddL1Whp7fI_1Ze/fallback_count/1/?rn=6
```



## ad\_break

- ad\_break
  - iOS

```
https://youtubei.googleapis.com/youtubei/v1/player/ad_break?key AIzaSyB-63vPrdThhKuerbB  
2N_17Kwwcxj6yUAc
```



- web

```
https://www.youtube.com/youtubei/v1/player/ad_break?key AIzaSyAO_FJ2S1qU8Q4STEHLGCilw_Y  
9_11qcW8&prettyPrint=false
```



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：  
2022-10-27 11:10:24

# 逆向YouTube总结

iOS逆向YouTube的话：

- 普通的类的逆向，相对不难
  - 毕竟针对于防破解做的很弱
    - 估计和欧美对于知识产权保护的比较严，有关系
- 有些底层的类，想要搞懂逻辑，不太容易
  - 比如：
    - Cronet 的相关调用
      - -[HAMCronetDataLoadTask startWithDelegate:delegateQueue:]
- 以及涉及到protobuf
  - 普通的类，容易通过 GPBMessage 找出原始字段定义
  - 但部分关键的类，还是被保护起来了：做成了lite版的protobuf类
    - 比如：
      - OnesieRequestProto
      - 等等
    - 只能很艰难的，通过反推 MessageLite 的 vtable 定义，再加上底层调试，才或许有机会，找出真正字段的定义（暂未来得及完成）

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：

2022-10-27 11:17:43

## 子教程

- iOS逆向YouTube
  - Protobuf相关
    - [iOS逆向YouTube: protobuf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 14:56:23

## 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：  
2022-10-27 10:54:52

## 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:  
2022-10-27 10:51:59