

iOS 点播 Core SDK

SDK 地址

最新版 [change log](#)

适用于iOS8.0+。

功能简介

iOS 点播核心 SDK 提供对点播播放器的播放管理操作（不含 UI），功能包括：

- 视频播放功能：提供视频ID、请求url的token等参数开始播放,也支持视频加密播放
- 播放控制方法：播放、暂停、停止、跳转至指定时间、变换视频清晰度和倍速播放
- 视频播放代理：提供响应播放事件的代理方法，用于捕获和处理播放中的错误
- 获取视频信息：当前播放状态、视频时长、视频缓存时长
- 视频下载功能：需要视频ID、请求url的token等参数

SDK 集成

- Podfile 中设置 source

```
1. source 'https://github.com/CocoaPods/Specs.git'
```



```
2. source 'http://git2.baijiashilian.com/open-  
ios/specs.git'
```

- Podfile 中引入 `BJPlayerManagerCore`

```
1. pod 'BJPlayerManagerCore', '~> 1.0'  
2.  
3. script_phase \  
4. :name => '[BJVideoPlayer] Embed Frameworks',  
5. :script =>  
    'Pods/BJVideoPlayer/BJVideoPlayer/EmbedFramework  
  
6. :execution_position => :after_compile  
7.  
8. script_phase \  
9. :name => '[BJLiveBase] Clear Archs From  
    Frameworks',  
10. :script =>  
    'Pods/BJLiveBase/script/ClearArchsFromFrameworks  
    "BJYIJKMediaFramework.framework",  
11. :execution_position => :after_compile
```

- 配置ATS. 需要在info.plist里面增加
NSAllowsArbitraryLoads = true

1. 引入头文件

```
1. #import  
    <BJPlayerManagerCore/BJPlayerManagerCore.h>
```

2. 全局设置

- 使用下载功能和播放下载的时候, 需要在使用前设置下载的
rootPath, 或者可以在下面的方法中设置下载的rootpath

```
1. - (BOOL)application:(UIApplication *)application
    didFinishLaunchingWithOptions:(NSDictionary
    *)launchOptions;
```

SDK 使用

1. 设置专属域名前缀

- 1.4.5 之后的版本支持设置专属域名前缀，需要在创建 `BJPlayerManager` 实例之前设置。例如专属域名为 `demo123.at.baijiayun.com`，则前缀为 `demo123`，参考 [专属域名说明](#)。

```
1. // 以demo123.at.baijiayun.com为例，domainPrefix
   = @"demo123"
2. NSString *domainPrefix = @"yourDomainPrefix";
3. [[PMAAppConfig sharedInstance]
   setPrivateDomainPrefix:domainPrefix];
```

2. 播放管理类初始化

播放器管理建议使用 `BJPlayerManager`，不建议使用 `BJPlayerManager` 中的 `player`

- 1.3.x 版本

```
1. BJPlayerManager *playManger =
   [[BJPlayerManager alloc] init];
2. playManger.delegate = self; // 设置代理
```

- 1.4.x 版本开始提供视频加密播放

```
1. BJPlayerManagerType playType =  
    BJPlayerManagerType_AVPlayer;  
2. BJPlayerManager *playManger =  
    [[BJPlayerManager managerWithType: playType];  
3. playManager.delegate = self; // 设置代理
```

3. 视频播放

1. 设置视频播放源

- 获取视频信息, 设置视频播放源都会清空 BJPlayerManager 的视频信息.

```
1. //在线点播播放  
2. [self.playerManager setVideoID:vid token:token  
    completion:^(BOOL(PMVideoInfoModel * _Nonnull  
        result, NSError * _Nonnull error) {  
3.     return shouldAutoPlay; // block的返回值设置是  
        否自动播放  
4.     }];  
5.  
6. //本地视频播放  
7. [self.playerManager setVideoPath:path  
    definition:definitionType  
    completion:^(BOOL(PMVideoInfoModel * _Nonnull  
        result, NSError * _Nonnull error) {  
8.     return shouldAutoPlay; // block的返回值设置是  
        否自动播放  
9.     }];
```

2. 获取视频播放View, 设置frame

```
1. playerManager.player.view.frame =  
    CGRectMake(0, 0, SCREEN_WIDTH,
```

```
SCREEN_WIDTH*9/16);
2. [self.view
   addSubview:playerManager.player.view];//添加到
   目标视图
```

3. 视频播放流程控制 BJPMControlProtocol

```
1. [playerManager play];// 播放
2. [playerManager pause];// 暂停
3. [playerManager stop];// 停止播放
4. [playerManager seek:time];// 进度跳转
5.
6. /* 播放倍率 支持 [0.5, 2.0] 范围内的倍速设置, 该范围
   之外的倍速将不做任何响应 */
7. [playerManager changeRate:rate]; // 切换倍速
8. [playerManager changeDefinition:
   (PMVideoDefinitionType)ddefineType]; // 切换清晰度
```

4. 视频播放代理方法 BJPMProtocol

```
1. @required
2. // 播放过程中出错(具体错误码参考下节)
3. - (void)videoplayer:(BJPlayerManager
   *)playerManager throwPlayError:(NSError *)error;
4.
5. @optional
6.
7. // 视频播放完成的回调方法
8. - (void)videoDidFinishPlayInVideoPlayer:
   (BJPlayerManager *)playerManager;
9.
10. // 播放器暂停的回调方法
```

```
11. - (void)videoPlayPauseInVideoPlayer:
    (BJPlayerManager *)playerManager;
```

5. 视频信息获取

```
1.
2. [self.playerManager getVideoInfoWithVid:vid
   token:token completion:^(void(^)
   (PMVideoInfoModel *_Nullable videoInfo, NSError
   *_Nullable error))completion {
3. // PMVideoInfoModel 为视频信息
4. }];
```

6. 播放中的视频属性提供

- 具体使用方法可以参考UI SDK的使用

```
1. // 存放用户自定义消息
2. @property (nonatomic, strong, nullable) NSString
   *userInfo;
3.
4. // 是否开启记忆播放
5. @property (nonatomic, assign) BOOL
   playTimeRecordEnabled;
6.
7. // 设置初始化播放时间, 如果设置了开启记忆播放, 再设置这个值无效
8. @property (nonatomic, assign) NSTimeInterval
   initialPlaybackTime;
9.
10. // 当前播放状态
11. @property (nonatomic, readonly) PMPlayState
   playState;
12.
13. // 视频的时长
```

```
14. @property (nonatomic, readonly) NSTimeInterval
    duration;
15.
16. // 视频缓存时长
17. @property (nonatomic, readonly) NSTimeInterval
    cacheDuration;
18.
19. // 当前播放倍速
20. @property (nonatomic, readonly) CGFloat
    playRate;
21.
22. // 正片的当前的播放时间
23. @property (nonatomic, readonly) NSTimeInterval
    currentTime;
24.
25. // 播放器实例, 建议只读取属性, 不改变属性以及调用方法
26. @property (readonly, nullable)
    PKMoviePlayerController *player;
27.
28. // 播放信息
29. @property (nonatomic, readonly, nullable)
    PMVideoInfoModel *videoInfoModel;
30.
31. // 当前播放清晰度
32. @property (nonatomic, readonly, nullable)
    PMVideoDefinitionInfoModel
    *currDefinitionInfoModel;
33.
34. // 当前播放的CDN
35. @property (nonatomic, readonly, nullable)
    PMVideoCDNInfoModel *currCDNInfoModel;
```

7. 播放通知

1. `PMPlayStateChangeNotification`; // 播放状态改变通知
2. `PMPlayerCreateNotification`; // 播放器实例被创建, 即将播放视频
3. `PMPlayerWillPlayNotification`; // 获取到播放信息后的通知
4. `PMPlayerFileInvalidNotification`; // 本地播放地址无效的通知
5. `PMPlayerFileNotExistNotification`; // 本地播放文件不存在的播通知
6. `PMPlayerWillSeekNotification`; // 即将跳转到某一时刻
7. `PMPlayerWillChangeDefinitionNotification`; // 即将切换清晰度
8. `PMPlayerGetPlayInfoFailedNotification`; // 获取播放信息失败

4. 错误码(状态码)

1. `BJPMEErrorCodeLoading` = 1000, // 视频加载中
2. `BJPMEErrorCodeLoadingEnd` = 1001, // 视频加载完成
3. `BJPMEErrorCodeParse` = 1002, // 视频播放错误
4. `BJPMEErrorCodeNetwork` = 1003, // 网络错误, 没有网络或是未知网络
5. `BJPMEErrorCodeWWAN` = 1004, // 非WIFI环境
6. `BJPMEErrorCodeWIFI` = 1005, // wifi
7. `BJPMEErrorCodeServer` = 1006, // 请求服务端返回错误
8. `BJPMEErrorCodeApp` = 1007, // app端调用错误

```
9.    BJPMErrorCodeDownloadInvalid    = 1010,    //
      视频的url失效
10.   BJPMErrorCodeMissFile           = 1011,    // 下载的文件丢失
```

5. 记忆播放

- 设置 BJPlayerManager 的 playTimeRecordEnabled 属性开关记忆播放。开启后记录 vid 对应的播放进度，每 5 秒【更新】一次播放进度，seek 时、停止播放时立即【更新】一次；
- 如果 视频总时长 减去 当前进度 小于 N 秒，则视为播放结束，只移除记录、不添加，N 默认 5 秒，可自定义，参考 BJPlayerManager 的 ignorableRemainingTimeInterval 属性；
- 调用 clearPlayTimeRecords 方法清除播放纪录；
- 最多存 100 条，超过 100 条时顶掉最早的；
- 打开了记忆播放，再设置视频初始播放时间无效。

6. 视频下载 PMDownloadManager

1.初始设置

- 注意:下载前需要先设置根路径, 然后调用单例方法, 否则代码抛出异常
- 文件保存目录参考[iOS官方文件存储文档](#).

```
1. /* 下载根路径, folder需要是一个 存在的 文件夹, 如果不存在, 需要上层创建 */
2. NSString *path = @"xxx/folder";
3. [PMDownloadManager
   downloadManagerWithRootPath:path];
```

2.下载方法

```

1. //点播下载
2. [[PMDownloadManager downloadManager]
   addDownloadWithVid:vid token:token
   definionArray:definitionArray
   showFileName:showFileName];
3.
4. //回放下载
5. [[PMDownloadManager downloadManager]
   addDownloadWithClass:classID
   seesionID:sessionID token:token
   definionArray:defiArr showFileName:showname
   creatTime:@"creatTime"];

```

需要注意的是,由于我们使用了ID+token的视频下载方式,因此是存在token过期的现象,发生token失效或者下载url失效的问题时,会向上层抛出BJPMErrroCodeDownloadInvalid (1010)的错误码,此时需要上层重新获取token,并调用下面的方法来重新设置token

```

1. /**
2.  下载中的任务发生了下载的url失效的错误,需要调用此
   方法,内部重新请求下载地址
3.
4. @param downloader 下载的任务
5. @param token 新的token
6. */
7. - (void)resetDownloadWithDownloader:
   (PMDownloader *)downloader token:(NSString
   *)token;

```

3.主要属性

1. `/** 下载事件相关的代理 */`
2. `@property (nonatomic, weak )`
`id<PMDownloadDelegate> downloadDelegate;`
- 3.
4. `//下载过程中的数据 和 已经完成的数据`
5. `@property (nonatomic, readonly) NSArray`
`<PMDownloader *> *downloadingList;`
6. `@property (nonatomic, readonly) NSArray`
`<PMDownloadModel *> *finishedList;`

4. 下载代理方法

- 下载代理请使用 `PMDownloadManager` 的代理方法, 不能直接使用单个下载 `PMDownloader` 的代理方法.

1. 代理方法:
2. `- (void)startDownload:(PMDownloader`
`*)downloader;//开始下载`
3. `- (void)updateProgress:(PMDownloader`
`*)downloader;//获取下载进度`
4. `- (void)finishedDownload:(PMDownloader`
`*)downloader;//下载完成`
- 5.
6. `/**`
7. 下载时的错误回调, 包括开始下载之前和下载过程中的错误
- 8.
9. `@param downloader` 下载过程中的下载器实例, 可能为空
10. `@param beforeDownloadModel` 下载开始前的错误 model, 可能为空
11. `*/`
12. `- (void)downloadFail:(nullable PMDownloader`
`*)downloader beforeDownloadError:(nullable`

```
PMBeforeDownloadModel  
*)beforeDownloadModel;
```

5.分账户下载

- 设置rootpath时, 需要传入绝对路径, 文件保存目录参考[iOS官方文件存储文档](#).
- 由于rootpath为绝对路径, 所以就需要在APP每次使用前设置, 在- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions 里面更新路径.
- 分账户下载的rootpath是需要跟账户的唯一id(或者其他ID)相关联. 每次登陆一个新账户也需要清除上个账户的信息, 或者在退出账户时清理.

```
1. [PMDownloadManager  
downloadManagerDestory];
```

- 登录新账户时,也需要重新设置新账户的rootpath.

6.下载中的错误回调方法处理

1. 错误回调:

```
1. -(void)downloadFail:(nullable PMDownloader  
*)downloader beforeDownloadError:(nullable  
PMBeforeDownloadModel  
*)beforeDownloadModel;
```

2. 触发该回调的case:

1. - 下载需要2个步骤:
2. 1.由vid/(或者classid/sessionid)和token去服务端获取下载地址

3. 2.开始下载
- 4.
5. - 有2种情况的error,
6. a. beforeDownloadModel是开始下载之前的信息, 具体打印beforeDownloadModel.error;
7. b. 下载过程中的错误信息, 打印
downloader.downloadModel.error
- 8.
9. - beforeDownloadModel
10. 1. rootPath为空
11. 2. 获取下载地址时服务端返回的错误信息
12. 3. 文件已经下载完成
13. 4. 正在下载
14. 5. resume的时候, 会先检测当前下载的url是否有效, 具体打印beforeDownloadModel.urlCheckState
15. 6. 检测url, 发现下载的url已经失效
16. - downloader.downloadModel.error
17. 1. 下载过程中出现的错误

7. 片头片尾以及广告播放

- 可以在后台设置片头片尾广告以及片头片尾, 片头片尾广告是图片广告, 片头片尾可以上传视频, 目前ijk播放器不支持。

1. // 设置是否需要片头片尾, 默认不需要
2. [PMAAppConfig sharedInstance].isNeedAD = isNeedAD;
3. // 设置是否自动播放图片广告, 默认自动播放
4. [PMAAppConfig sharedInstance].notAutoStartImageAD = notAutoStartImageAD;
5. // 播放图片广告
6. [self.playerManager startAD];
7. // 暂停图片广告
8. [self.playerManager pauseAD];

8. 视频加密

- 视频加密需要使用1.4.x之后的SDK版本
- 只有ijk播放器支持播放加密视频
- 播放, 下载加密视频

```
1. self.playerManager = [BJPlayerManager
    managerWithType:BJPlayerManagerType_IJKPlayer];
2.
3. // 在线视频
4. BOOL needEncrypt = YES;
5. @YPWeakObj(self);
6. [self.playerManager setVideoID:vid token:token
    encrypted:needEncrypt
    completion:^(BOOL(PMVideoInfoModel * _Nonnull
    result, NSError * _Nonnull error) {
7.     @YPStrongObj(self);
8.     // 视频的详细信息
9.     self.videoInfoModel = result;
10.    return shouldAutoPlay;
11. }]);
12.
13. // 本地视频
14. @YPWeakObj(self);
15. [self.playerManager setVideoPath:path
    definition:definitionType
    completion:^(BOOL(PMVideoInfoModel * _Nonnull
    result, NSError * _Nonnull error) {
16. @YPStrongObj(self);
17.     // 视频的详细信息
18.     self.videoInfoModel = result;
19.     return shouldAutoPlay;
20. }]);
```

```
21.  
22. // 下载加密视频  
23. [[PMDownloadManager downloadManager]  
    addDownloadWithVid:vid token:token  
    definitionArray:array showFileName:showFileName  
    need_encrypt:needEncrypt];
```

9. 纯音频

- 目前纯音频需要使用1.4.5之后的SDK版本
- 点播需要在后台配置转码纯音频才有纯音频
- 点播在线播放的默认清晰度可以在后台配置清晰度列表, SDK根据配置的清晰度优先级列表来匹配第一个清晰度, 如果后台配置了纯音频最高优先级, 就可以进入回放和点播的时候直接播放纯音频, 否则就通过改变清晰度的方式播放纯音频。
- 视频清晰度 PMVideoDefinitionType 增加纯音频枚举

```
1.  
2. typedef NS_ENUM(NSInteger,  
    PMVideoDefinitionType){  
3.     DT_Audio    = -2, // 音频  
4.     DT_Unknown  = -1, // 未知  
5.     DT_LOW      = 0,  // 标清  
6.     DT_HIGH     = 1,  // 高清  
7.     DT_SUPPERHD = 2,  // 超清  
8.     DT_720p     = 3,  // 720p  
9.     DT_1080p    = 4,  // 1080p  
10. };
```

- 播放纯音频

```
1. // 播放音频
```

2. `[[self.playerManager changeDefinition:DT_Audio];`

- 下载纯音频, 同一个视频只能下载纯音频或其他的一种清晰度, 按照传入的清晰度列表匹配第一个满足条件的清晰度

```
1. self.preferredDefinitionList =  
   @[@(-2),@(0),@(1),@(2),@(3),@(4)];  
2.  
3. [[PMDownloadManager downloadManager]  
   addDownloadWithVid:vid token:token  
   definionArray:self.preferredDefinitionList  
   showFileName:showFileName];
```

点播Core git 地址

常见问题

1. app每次杀死重启之后,就找不到下载的finishedlist和downinglist?

- 确认设置下载目录rootpath时,需要在 `- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions` 方法中设置的.

2. 下载时经常提示下载失败

- SDK提供的下载方法是需要传递 `视频id+token`, 或者 `回放ID+sessionID+token` 的, 用户可以在百家云的后台设置token的有效期, 所以是存在token失效的问题的.
- 在downloadManager的delegate回调方法中监听到 `BJPMErrroCodeDownloadInvalid` 错误码时, 代表当前

url失效了,需要再次获取新的token,然后调用我们提供的 `resetDownload...` 方法重新设置token.

3. 对于下载和播放的回调方法中抛出来的error的具体信息?

- 通过 `NSString *errMsg = [error.userInfo objectForKey:NSLocalizedStringKey]`; 一般都可以获取到详细的错误信息,可以帮助进一步定位错误.

4. 播放在线视频/下载视频时偶现提示1006错误码,提示服务端的错误

- 出现1006的错误码时,大部分是token失效/token校验不通过的情况,建议播放视频下载时总是获取最新的token. 另外下载过程中也需要注意 `BJPMEErrorCodeDownloadInvalid` 的错误码,及时更新token.

5. token验证失败,或者返回1010的错误码

- 除了以上原因,传token参数的时候,请务必确认一下token里面不能包含空格,否则播放或者下载视频的时候,将有可能提示token验证失败.