



锐动 iOS 带 UI 视频编辑 SDK



作者：RD iOS Team

日期：20017-08-01

1 编写目的

预期读者：

有视频编辑开发经验或者无经验的，打算或者正在使用“锐动 iOS 带 UI 视频编辑 SDK”的相关工程师。

- iOS 软件工程师。
- 产品经理。
- QA

2 名词解释

- 分辨率：用于计算机视频处理的图像，以水平和垂直方向上所能显示的像素数来表示分辨率。常见视频分辨率的有 1080P 即 1920x1080，720P 即 1080x720，640x480 等。
- 宽高比：视频分辨率的宽高比，常见的有 16:9，4:3，1:1。锐动视频编辑 SDK 对各宽高比的视频都支持编辑，导出的默认分辨率是 640x360，宽高比是 16:9。
- 帧率：每秒的帧数 (fps) 或者说帧率表示图形处理器处理场时每秒钟能够更新的次数。
- 码率：数据传输时单位时间传送的数据位数，一般我们用的单位是 kbps 即千位每秒。
- 素材：来自系统相机，其他 app，电脑，网络的照片，音乐，视频等。
- 视频分割：把视频分割为若干个片段，可用于从视频中截取一个或者多个精彩瞬间，或者删除不喜欢的片段，分割后的多个片段也可以用于重新调整顺序，视频合并等。
- 视频裁切：从视频画面中裁切出来仅需要的部分，锐动视频编辑 SDK 针对手机竖

屏拍摄的画面,支持动态调整裁切区域,帮助用户不同时间点选择自己需要的画面。

- 视频旋转: 对视频进行 90° , 180° , 270° 旋转, 一般用于矫正用户手机上录制的头像横置的视频。
- 视频镜像: 对视频左右镜像。
- 视频变速: 对视频播放速度调整, 实现慢镜头或者快镜头效果。
- 视频配乐: 选取本地或者网络音乐作为视频的背景音乐。
- 视频配音: 通过 microphone 对视频配音。
- 视频字幕: 使用文字标注视频。
- 视频滤镜: 调整视频的画面颜色效果。
- 视频特效: 在视频里面增加特效动画或声音
- 合并转场: 两个视频片段之间的衔接效果。
- 主题 (模版): 应用视频的整体效果方案。

3 XCode 集成步骤

3.1 运行环境

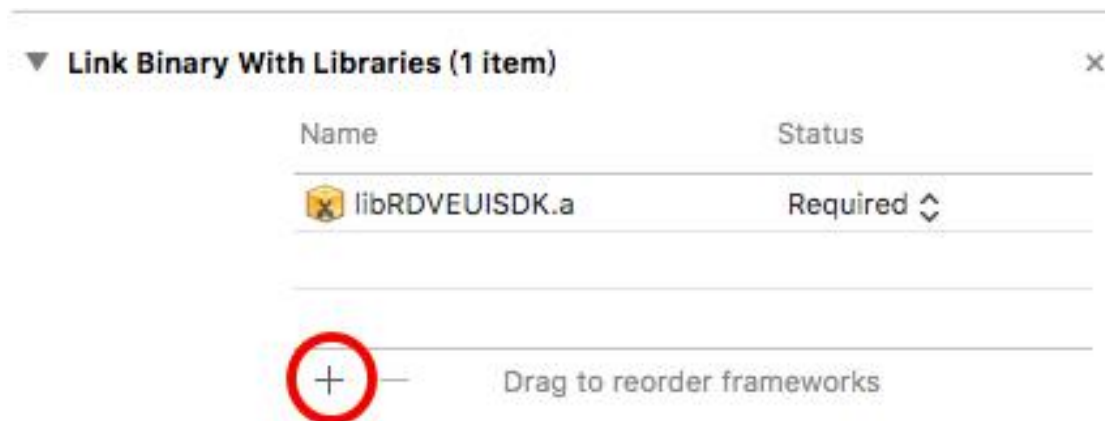
SDK 支持 iOS7.0 及以上版本的设备，兼容 iPhone、iPad、iPod touch

3.2 注册申请 AppKey 和 APPSecret

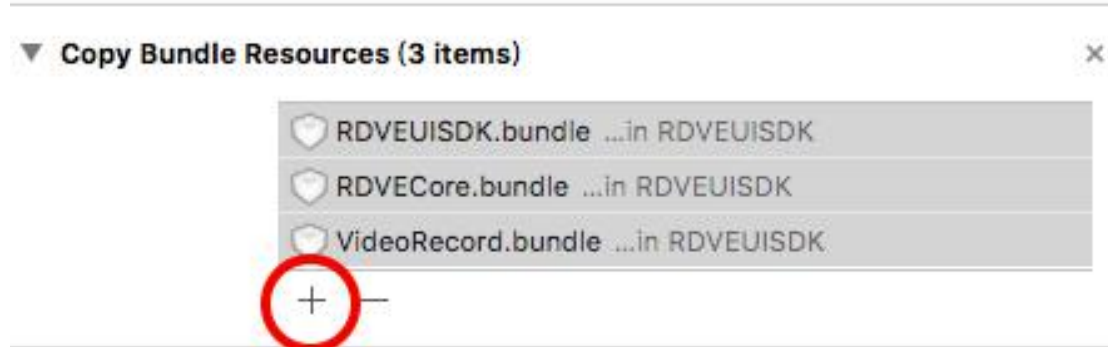
- 1、 打开 <http://www.rdsdk.com> ，在锐动官网首页选择登 录或注册，新用户先注册，老用户直接登录。
- 2、 登录完成后，进入视频云管理 点击“+新增”，填写您应 用的名称等，点击“提交”，应用创建完成。
- 3、 创建应用完成后会自动跳转至个人中心页面，此时会获 取到该应用的 AppKey 和 AppSecret。

3.3 下载并导入 SDK

3.3.1 添加静态库 libRDVEUISDK.a 和头文件

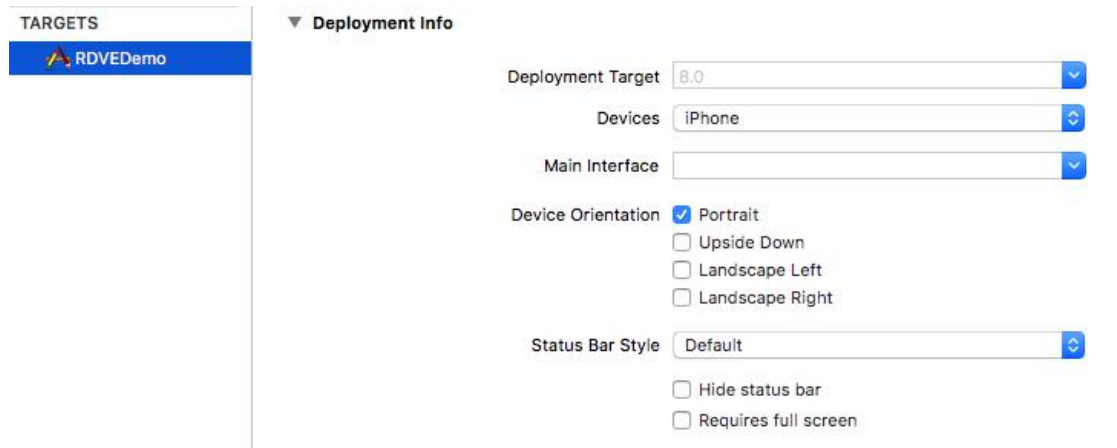


3.3.2 添加 RDVEUISDK.bundle 、 RDVECore.bundle 、 VideoRecord.bundle



3.3.3 工程设置

- TARGETS 的 Deployment Info 仅勾选竖屏 (Portrait) ， 横屏 (LandscapeLeft 、 LandscapeRight)不勾选:



- IOS10 适配

在 Info.plist 中 添加 NSCameraUsageDescription 、
NSMicrophoneUsageDescription 、 NSPhotoLibraryUsageDescription 、
NSAppleMusicUsageDescription。

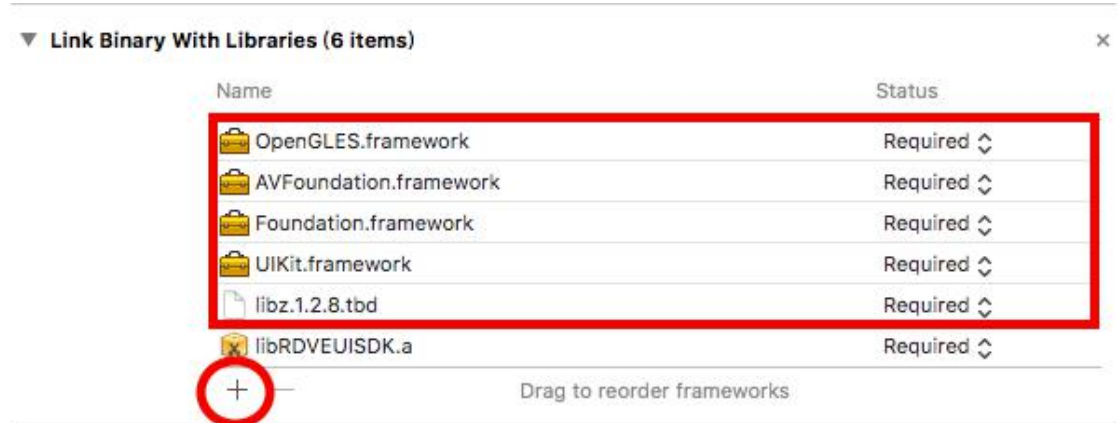
Privacy - Camera Usage Description	String
Privacy - Microphone Usage Description	String
Privacy - Media Library Usage Description	String
Privacy - Photo Library Usage Description	String

- HTTP 协议适配

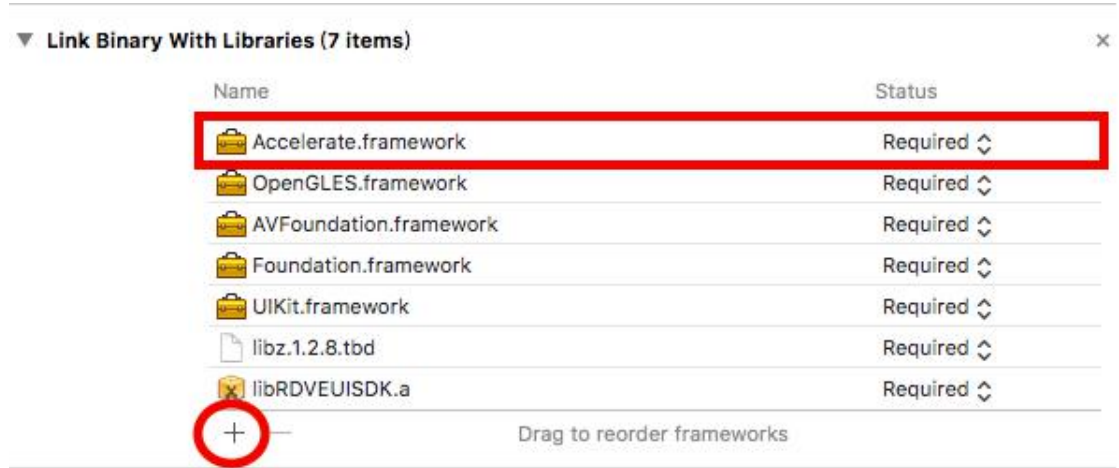
在 Info.plist 中 添加 App Transport Security Settings

▼ App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES

- 添加引用苹果官方库



如果录制时使用 faceU，还需要引用库 Accelerate.framework



- 设置 Other Linker Flags

在 TARGETS->Build Settings->Other Linker Flags 中加入-ObjC。

3.4 集成 SDK

3.4.1 引入头文件及初始化

```
#import "RDVEUISDK.h"
/** 初始化对象
 *
 * @param appkey          appkey description
 * @param appsecret       appsecret description
 * @param resultFailBlock 返回错误信息
 */
- (instancetype) initWithAPPKey:(NSString *)appkey
                      APPSecret:(NSString *)appsecret
                      resultFail:(RdVEFailBlock )resultFailBlock;
```

3.4.2 功能设置

```
/** 视频导出设置
 */
@property (nonatomic, strong) ExportConfiguration *exportConfiguration;

/** 编辑设置
 */
@property (nonatomic, strong) EditConfiguration * editConfiguration;

/** 拍摄功能设置
 */
@property (nonatomic, strong) CameraConfiguration * cameraConfiguration;
```

3.4.2.1 视频导出设置

```
/** 设置视频输出最大时长（单位是秒）不限制则传0，默认不限制
 */
@property (nonatomic,assign)long int    videoMaxDuration;

/** 是否禁用片尾
 */
@property (nonatomic,assign)bool    endPicDisabled;

/** 片尾显示的用户名
 */
@property (nonatomic,copy)NSString * _Nullable endPicUserName;

/** 片尾显示时长（不包括淡入时长）
 */
@property (nonatomic,assign)float    endPicDuration;

/** 片尾淡入时长
 */
@property (nonatomic,assign)float    endPicFadeDuration;

/** 片尾显示的图片路径
 */
@property (nonatomic,copy)NSString * _Nullable endPicImagepath;

/** 是否禁用水印
 */
@property (nonatomic,assign)bool    waterDisabled;

/** 文字水印
 */
@property (nonatomic,copy,nullable)NSString *waterText;

/** 文字水印颜色
 */
@property (nonatomic,copy,nullable)UIColor *waterTextColor;

/** 文字水印字体
 */
@property (nonatomic,copy,nullable)UIFont *waterTextFont;
```



```
/** 图片水印
 */
@property (nonatomic, strong, nullable) UIImage *waterImage;

/** 显示位置
 */
@property (nonatomic, assign) RDWATERPOSITION waterPosition;

/** 设置视频输出码率 (单位是兆, 默认是6M), 建议设置大于1M, 导出和压缩都生效
 */
@property (nonatomic, assign) float videoBitRate;

/** 压缩视频分辨率 (调用压缩视频接口生效)
 */
@property (nonatomic, assign) CGSize
condenseVideoResolutionRatio;
```

3.4.2.2 视频编辑设置

```
/** 向导模式 如果需要自己删除一些功能 则需启用此参数 default false
 */
@property (assign, nonatomic) bool enableWizard;

/** 编辑视频所支持的文件类型 (default all)
 */
@property (assign, nonatomic) SUPPORTFILETYPE supportFileType;
```

3.4.2.2.1 相册界面设置

```
/** 默认选中视频还是图片
 */
@property (nonatomic, assign) RDDEFAULTSELECTALBUM
defaultSelectAlbum;

/** 选择视频和图片的最大张数
 */
@property (nonatomic, assign) int mediaCountLimit;

/** 启用相册相机 (default true)
 */
@property (nonatomic, assign) bool enableAlbumCamera;
```

```
/** 点击相册界面相机按钮回调  
 */  
@property (nonatomic, copy, nullable)  
void(^clickAlbumCameraBlackBlock)();
```

3.4.2.2.2 截取界面设置

```
/** 截取时间模式  
 */  
@property (nonatomic, assign) TRIMMODE trimMode;  
  
/** 单个时间定长截取: 截取时间 默认15s  
 */  
@property (nonatomic, assign) float trimDuration_OneSpecifyTime;  
  
/** 两个时间定长截取: 偏小截取时间 默认12s  
 */  
@property (nonatomic, assign) float  
trimMinDuration_TwoSpecifyTime;  
  
/** 两个时间定长截取: 偏大截取时间 默认30s  
 */  
@property (nonatomic, assign) float  
trimMaxDuration_TwoSpecifyTime;  
  
/** 两个时间定长截取: 默认选中小值还是大值  
 */  
@property (nonatomic, assign) RDdefaultSelectCutMinOrMax  
defaultSelectMinOrMax;  
  
/** 定长截取时, 截取后视频分辨率类型 默认TRIMVIDEOTYPE_ORIGINAL  
 * 自由截取时, 始终为TRIMVIDEOTYPE_ORIGINAL, 该设置无效  
 */  
@property (nonatomic, assign) TRIMEXPORTVIDEOTYPE  
trimExportVideoType;
```

3.4.2.2.3 编辑界面设置

```
/** 截取 (default true)  
 */  
@property (nonatomic, assign) bool enableTrim;  
  
/** 分割 (default true)
```

```
*/
@property (nonatomic,assign) bool enableSplit;
/** 裁切 (default true)
*/
@property (nonatomic,assign) bool enableEdit;

/** 变速 (default true)
*/
@property (nonatomic,assign) bool enableSpeedcontrol;

/** 复制 (default true)
*/
@property (nonatomic,assign) bool enableCopy;

/** 调整顺序 (default true)
*/
@property (nonatomic,assign) bool enableSort;

/** 调整视频比例 (default true)
*/
@property (nonatomic,assign) bool enableProportion;

/** 调整图片显示时长 (default true)
*/
@property (nonatomic,assign) bool enableImageDurationControl;

/** 文字版 (default true)
*/
@property (nonatomic,assign) bool enableTextTitle;

/** 默认视频输出方式 (自动, 横屏, 1 : 1)
*/
@property (nonatomic,assign) RDPROPORTIONTYPE proportionType;

/** 倒放 (default true)
*/
@property (nonatomic,assign) bool enableReverseVideo;
```

3.4.2.2.4 高级编辑界面设置

```
/** Music网络资源地址 (需自己构建网络下载API)
*/
```

```
@property (nonatomic,copy)NSString * _Nullable
musicResourceURL;

/** cloudMusic网络资源地址 (需自己构建网络下载API) default nil
 */
@property (nonatomic,copy)NSString * _Nullable
cloudMusicResourceURL;

/** MV网络资源地址 (需自己构建网络下载API)
 */
@property (nonatomic,copy,nullable)NSString *mvResourceURL;

/** MV (default true)
 */
@property (nonatomic,assign) bool enableMV;

/** 配乐 (default true)
 */
@property (nonatomic,assign) bool enableMusic;

/** 本地音乐 (default true)
 */
@property (nonatomic,assign) bool enableLocalMusic;

/** 配音 (default true)
 */
@property (nonatomic,assign) bool enableDubbing;

/** 配音类型 (default 方式一(配音不放在配乐里面))
 */
@property (nonatomic,assign) RDDUBBINGTYPE dubbingType;

/** 字幕 (default true)
 */
@property (nonatomic,assign) bool enableSubtitle;

/** 滤镜 (default true)
 */
@property (nonatomic,assign) bool enableFilter;

/** 特效 (default true)
 */
@property (nonatomic,assign) bool enableEffect;
```

```
/** 片段编辑 (default true)
 */
@property (nonatomic,assign) bool enableFragmentedit;

/** 进入SDK界面是否需要动画 (default true)
 */
@property (nonatomic,assign) bool presentAnimated;

/** 退出SDK界面是否需要动画 (default true)
 */
@property (nonatomic,assign) bool dismissAnimated;

/** 字幕层次 (false 内层)
 */
@property (nonatomic,assign) bool captionsOuter;

/** 特效层次
 */
@property (nonatomic,assign) bool specialEffectOuter;
```

3.4.2.3 视频录制功能设置

```
/** 拍摄模式 default CameraModel_Onlyone
 */
@property (nonatomic,assign) CameraModelType
cameraModelType;
/** 是否在拍摄完成就保存到相册再返回 default false 该参数只有在
CameraModel_Onlyone 模式下才生效
 */
@property (nonatomic,assign) bool
cameraWriteToAlbum;

/** 前/后置摄像头 default AVCaptureDevicePositionFront
 */
@property (nonatomic,assign) AVCaptureDevicePosition
cameraCaptureDevicePosition;

/** 相机的方向 default RecordVideoOrientationAuto
 */
@property (nonatomic,assign) RecordVideoOrientation
cameraRecordOrientation;

/** 相机录制大小 (正方形录制, 非正方形录制)
```

```
*/
@property (nonatomic,assign) RecordVideoSizeType
cameraRecordSizeType;

/** 录制视频帧率 default 30
*/
@property (nonatomic,assign) int32_t
cameraFrameRate;

/** 录制视频码率 default 4000000
*/
@property (nonatomic,assign) int32_t
cameraBitRate;

/** 录制还是拍照 default RecordTypeVideo
*/
@property (nonatomic,assign) Record_Type
cameraRecord_Type;

/** 视频输出路径
*/
@property (nonatomic,copy,nullable) NSString
*cameraOutputPath;

/**录制的视频的大小 (如: {720,1280}) default CGSizeZero
*/
@property (nonatomic,assign) CGSize
cameraOutputSize;

/** 正方形录制时配置按钮的位置 (美颜, 延时, 摄像头切换 三个按钮的位置)
*/
@property (nonatomic,assign) CameraCollocationPositionType
cameraCollocationPosition;

/** 正方形录制最大时长(default 10 )
*/
@property (nonatomic,assign) float
cameraSquare_MaxVideoDuration;

/**非正方形录制最大时长 (default 0 , 不限制)
```

```
*/
@property (nonatomic,assign) float
cameraNotSquare_MaxVideoDuration;

/**录制最小时长 (default 0 , 不限制 正方形录制和长方形录制都生效)
*/
@property (nonatomic,assign) float
cameraMinVideoDuration;

/** 是否开启滤镜功能(只有在开启Faceu的时候该参数才生效)
*/
@property (nonatomic , assign) bool
enableFilter;

/**人脸道具贴纸
*/
@property (assign, nonatomic) bool
enableFaceU;

/**人脸道具贴纸证书
*/
@property (assign, nonatomic,nonnull) void
*faceUAuth;

/**人脸道具贴纸证书长度
*/
@property (assign, nonatomic)Float64
lenFaceUAuth;

/**是否启用网络下载faceUnity
*/
@property (assign, nonatomic)bool
enableNetFaceUnity;

/**人脸道具贴纸下载路径
*/
@property (copy, nonatomic,nonnull)NSString
*faceUURL;

/** 拍摄类型:可拍摄短视频MV(default true)
```

```
*/
@property (nonatomic,assign) bool
cameraMV;

/** 拍摄类型:可拍摄视频(default true)
*/
@property (nonatomic,assign) bool
cameraVideo;

/** 拍摄类型:可拍摄照片(default true)
*/
@property (nonatomic,assign) bool
cameraPhoto;

/** 短视频MV录制最小时长(default 3s )
*/
@property (nonatomic,assign) float
cameraMV_MinVideoDuration;

/** 短视频MV录制最大时长(default 15s )
*/
@property (nonatomic,assign) float
cameraMV_MaxVideoDuration;

/** 从相机进入相册
*/
@property (nonatomic,copy,nonnull)
void(^cameraEnterPhotoAlbumCallblackBlock)( );

/** 是否隐藏相机进入相册按钮
*/
@property (nonatomic,assign) bool
hiddenPhotoLib;

// faceU美颜设置
/**瘦脸 0.0~1.0    default 0.68
*/
@property (nonatomic , assign) float cheekThinning;

/**大眼 0.0~2.0    default 1.53
*/
@property (nonatomic , assign) float eyeEnlarging;
```



```
/**美白 0.0~1.0    default 0.48
 */
@property (nonatomic , assign) float colorLevel;

/**磨皮 1 2 3 4 5    default 3
 */
@property (nonatomic , assign) float blurLevel;

/** 瘦脸等级 0.0 ~ 1.0 默认1.0
 */
@property (nonatomic , assign) float faceShapeLevel;

/** 瘦脸类型 0.0 ~ 3.0 默认0.0
 */
@property (nonatomic , assign) float faceShape;

/*是否开启使用音乐录制。如果需要切换音乐请设置好音乐下载路径,不设置则跳转到本地音乐界面 (editConfiguration.cloudMusicResourceURL)
 */
@property (nonatomic, assign) BOOL enableUseMusic;

/*传入需要录制时播放的音乐
 */
@property (nonatomic, strong) RDMusicInfo * _Nullable musicInfo;
```

3.4.2.3.1 启用 faceU 说明

- 1、导入签权证书
- 2、导入美颜基础资源: face_beautification.bundle、v3.bundle

3.4.2.3.1.1 开启 faceU

```
rdVESDK.cameraConfiguration.enableFaceU = true;
```

3.4.2.3.1.2 启用签权证书

```
rdVESDK.cameraConfiguration.faceUAuth = &g_faceuAuth_package;
rdVESDK.cameraConfiguration.lenFaceUAuth = sizeof(g_faceuAuth_package);
```

3.4.2.3.1.3 从网络加载 faceUnity 道具资源

```
rdVESDK.cameraConfiguration.enableNetFaceUnity = true;
```

`rdVESDK.cameraConfiguration.faceUURL = @"http://faceUnity 道具资源下载地址";`

自定义获取 faceUnity 道具的 url 方法，具体参见《3.11.1 自定义 faceU 资源》。

3.5 录制视频

```
/** 录制视频 参数设置(cameraConfiguration)
 * @param source 源视图控制器
 * @param callbackBlock 完成录制回调(result : 0 表示MV 1表示视频)
 * @param imagebackBlock 拍照完成回调
 * @param cancelBlock 取消录制回调
 */

- (void)videoRecordAutoSizeWithSourceController: (UIViewController*)source
        callbackBlock: (RdVERecordCallbackBlock)callbackBlock
        imagebackBlock: (RdVECallbackBlock)imagebackBlock
        faileBlock: (RdVEFailBlock)failBlock
        cancel: (RdVETCancelBlock)cancelBlock;
```

3.6 编辑视频

```
/** 编辑短视频MV
 *
 * @param viewController 源控制器
 * @param urlAsset 视频源
 * @param clipTimeRange 时间段
 * @param crop crop范围 default : CGRectMake(0, 0, 1, 1)
 */
- (void)editVideoWithSuperController:(UIViewController *)viewController
        urlAsset:(AVURLAsset *)urlAsset
        clipTimeRange:(CMTimeRange )clipTimeRange
        crop:(CGRect)crop
        outputPath:(NSString *)outputVideoPath
        callback:(RdVECallbackBlock )callbackBlock
        cancel:(RdVETCancelBlock )cancelBlock;
```

```
/** 编辑视频(传入URL数组)
 *
 * @param viewController 源控制器
 * @param foldertype 缓存文件夹类型 (Documents、Library、Temp,None)
 * @param appAlbumCacheName 需扫描的缓存文件夹名称
 * @param urlsArray 视频/图片路径(NSMutableArray <NSURL*>)
如: 相册视频 assets-library://asset/asset...
 * @param outputPath 视频输出路径
 * @param callbackBlock 完成编辑回调
 * @param cancelBlock 取消编辑回调
 */
- (void)editVideoWithSuperController:(UIViewController *)viewController
    foldertype:(FolderType)foldertype
    appAlbumCacheName:(NSString *)appAlbumCacheName
    urlsArray:(NSMutableArray *)urlsArray
    outputPath:(NSString *)outputVideoPath
    callback:(RdVECallbackBlock )callbackBlock
    cancel:(RdVETCancelBlock )cancelBlock;
```

3.7 自定义相册或是调用系统相册

实现代理方法:

```
/**
 *如果需要自定义相册则需要实现此函数
 */
- (void)selectVideoAndImageResult:(UINavigationController *)nav
callbackBlock:(void (^)(NSMutableArray *lists))callbackBlock;

/**
 *如果需要自定义相册则需要实现此函数(添加视频)
 */
- (void)selectVideosResult:(UINavigationController *)nav
callbackBlock:(void (^)(NSMutableArray *lists))callbackBlock;

/**
 *如果需要自定义相册则需要实现此函数(添加图片)
 */
- (void)selectImagesResult:(UINavigationController *)nav
```

```
callbackBlock:(void (^)(NSMutableArray *lists))callbackBlock;
```

以系统相册为例：（需实现以下两个协议：UINavigationControllerDelegate, UIImagePickerControllerDelegate）

进入系统相册：

```
- (void)selectVideoAndImageResult:(UINavigationController *)nav callbackBlock:(void (^)(NSArray *))callbackBlock{
    _addFileType = addVideoAndImages;
    _editingSdk.addVideosAndImagesCallbackBlock = callbackBlock;
    if([self checkCameraISOpen]){
        UIImagePickerControllerSourceType sourceType = UIImagePickerControllerSourceTypeCamera;
        sourceType = UIImagePickerControllerSourceTypePhotoLibrary;//相册库

        UIImagePickerController *picker = [[UIImagePickerController alloc] init];
        picker.delegate = self;
        picker.allowsEditing = NO;
        picker.sourceType = sourceType;
        [nav presentViewController:picker animated:YES completion:nil];
    }
}

- (void)selectImagesResult:(UINavigationController *)nav callbackBlock:(void (^)(NSArray *))callbackBlock{
    _addFileType = addImage;
    _editingSdk.addImagesCallbackBlock = callbackBlock;
    if([self checkCameraISOpen]){
        UIImagePickerControllerSourceType sourceType = UIImagePickerControllerSourceTypeCamera;
        sourceType = UIImagePickerControllerSourceTypePhotoLibrary;//相册库
        //UIVideoEditorController
        UIImagePickerController *picker = [[UIImagePickerController alloc] init];
        picker.delegate = self;
        picker.allowsEditing = NO;
        picker.sourceType = sourceType;
        [nav presentViewController:picker animated:YES completion:nil];
    }
}
```

相册选择完成：

```
- (void)imagePickerController:(UIImagePickerController *)picker didFinishPickingMediaWithInfo:(NSDictionary *)info {
    [picker dismissViewControllerAnimated:YES completion:nil];

    NSMutableArray *lists = [[NSMutableArray alloc] init];
    NSURL *fileUrl = [info objectForKey:@"UIImagePickerControllerReferenceURL"];
    [lists addObject:fileUrl];

    if(_addFileType == addVideo){
        //添加视频
        _editingSdk.addVideosCallbackBlock(lists);//selectVideosResult 回调
    }else if(_addFileType == addImage){
        //添加图片
        _editingSdk.addImagesCallbackBlock(lists);//selectImageResult 回调
    }else{
        //添加视频和图片
        _editingSdk.addVideosAndImagesCallbackBlock(lists);//selectVideoAndImageResult 回调
    }
}
```

3.8 截取视频

```
/** 截取视频 通过传入视频对象 AVUrlAsset
 * @param viewController 源控制器
 * @param title 导航栏标题
 * @param backgroundColor 背景色
 * @param cancelButtonTitle 取消按钮文字
 * @param cancelButtonTitleColor 取消按钮标题颜色
 * @param cancelButtonTitleBackgroundColor 取消按钮背景色
```

```
* @param otherButtonTitle          完成按钮文字
* @param otherButtonTitleColor     完成按钮标题颜色
* @param otherButtonBackgroundColor 完成按钮背景色
* @param urlAsset                  数据源
* @param outputPath                视频输出路径
* @param callbackBlock             截取完成回调
* @param failback                  截取失败回调
* @param cancelBlock               取消截取回调
*/
- (void)trimVideoWithSuperController:(UIViewController *)viewController
    controllerTitle:(NSString *) title
    backgroundColor:(UIColor *) backgroundColor
    cancelButtonTitle:(NSString *) cancelButtonTitle
    cancelButtonTitleColor:(UIColor *) cancelButtonTitleColor
    cancelButtonBackgroundColor:(UIColor *) cancelButtonBackgroundColor
    otherButtonTitle:(NSString *) otherButtonTitle
    otherButtonTitleColor:(UIColor *) otherButtonTitleColor
    otherButtonBackgroundColor:(UIColor *) otherButtonBackgroundColor
    urlAsset:(AVURLAsset *) urlAsset
    outputPath:(NSString *) outputPath
    callbackBlock:(RdVE_TrimAssetCallbackBlock) callbackBlock
    failback:(RdVEFailBlock ) failback
    cancel:(RdVECcancelBlock ) cancelBlock;

/** 截取视频 通过传入路径 path
* @param viewController            源控制器
* @param title                    导航栏标题
* @param backgroundColor          背景色
* @param cancelButtonTitle        取消按钮文字
* @param cancelButtonTitleColor   取消按钮标题颜色
* @param cancelButtonBackgroundCo 取消按钮背景色
* @param otherButtonTitle        完成按钮文字
* @param otherButtonTitleColor    完成按钮标题颜色
* @param otherButtonBackgroundCo 完成按钮背景色
* @param assetPath                数据源
* @param outputPath              视频输出路径
* @param callbackBlock            截取完成回调
* @param failback                截取失败回调
* @param cancelBlock              取消截取回调
*/
- (void)trimVideoWithSuperController:(UIViewController *)viewController
    controllerTitle:(NSString *) title
    backgroundColor:(UIColor *) backgroundColor
    cancelButtonTitle:(NSString *) cancelButtonTitle
    cancelButtonTitleColor:(UIColor *) cancelButtonTitleColor
```

```
cancelButtonBackgroundColor:(UIColor *) cancelButtonTitle
otherButtonTitle:(NSString *) otherButtonTitle
otherButtonTitleColor:(UIColor *) otherButtonTitleColor
otherButtonBackgroundColor:(UIColor *) otherButtonBackgroundColor
assetPath:(NSString *) assetPath
outputPath:(NSString *) outputPath
callbackBlock:(RdVE_TrimVideoPathCallbackBlock)
callbackBlock
failback:(RdVEFailBlock ) failback
cancel:(RdVECancelButton ) cancelButton;

/**设置截取返回类型
 * 此方法点击自定义弹框的按钮时才会被调用
 */
- (void)trimVideoWithType:(RDCutVideoReturnType )type;
```

3.9 相册功能（选择视频或图片后返回选中资源）

```
/**选择视频或图片
 * @param viewController 源控制器
 * @param albumType 选择相册类型（仅选图片，仅选视频，两者皆可）
 * @param callbackBlock 相册选择完成返回一个NSURL数组 NSMutableArray
<NSURL *> *List
 * @param cancelButton 取消选择相册资源
 */
- (BOOL)onRdVEAlbumWithSuperController:(UIViewController *)viewController
albumType:(ALBUMTYPE)albumType
callback:(OnAlbumCallbackBlock) callbackBlock
cancelBlock:(RdVECancelButton) cancelButton;
```

3.10 视频压缩

```
/**获取视频信息 字典 width height fps bitrate
 *
 */
- (NSDictionary *) getVideoInformation:(AVURLAsset *)urlAsset;
```

```
/** 压缩视频 参数设置(ExportConfiguration)
 * @param size          压缩输出分辨率
 * @param bitrate       压缩输出比特率
 * @param progressBlock 压缩进度回调
 * @param callbackBlock 压缩完成回调
 * @param failBlock     压缩失败回调
 */
- (void)compressVideoAsset:(AVURLAsset *)urlAsset
    outputPath:(NSString *)outputPath
    startTime:(CMTime )startTime
    endTime:(CMTime )endTime
    outputSize:(CGSize) size
    outputBitrate:(float) bitrate
    superView:(UIViewController *)superView
    progressBlock:(void (^)(float progress))progressBlock
    callbackBlock:(void (^)(NSString
*videoPath))callbackBlock
    fail:(void (^)(NSError *error))failBlock;

/** 取消导出
 */
- (void) compressCancel;
```

3.11 TS/AVI 转 MP4

```
/** TS/AVI 转MP4
 *
 * @param sourcePath      源文件路径
 * @param outputPath     输出文件路径
 * @param keepAudio       是否需要音频
 * @param progressBlock   输出进度
 * @param completedBlock  转换完成的回调 [error: 转换失败的错误码]
 */
- (void)TSAVItoMP4:(NSString *)sourcePath
    outputPath:(NSString *)outputFilePath
    keepAudio:(bool)keepAudio
    progressBlock:(void (^)(float progress)) progressBlock
    completedBlock:(void (^)(NSError *error))completedBlock;
```

3.12 自定义资源

3.12.1 自定义 faceU 资源

获取人脸贴纸资源的 url 需满足以下定义：

请求格式：

URL	http://**
支持格式	JSON
HTTP 请求验证	POST/GET
请求次数限制	否

参数说明：

参数名	必选	类型及范围	说明
type	是	String	ios android

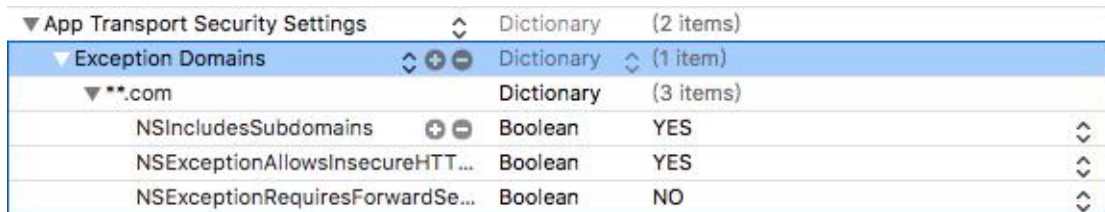
正常返回结果：

返回键	类型	返回值	说明
state	Boolean	请求状态	
code	Int	请求状态码	
message	String	请求状态文本信息	
result	Json	请求返回参数	
bundles	Array	贴纸列表	
name	String	贴纸名称	
img	String	贴纸图片	
url	String	贴纸链接	

```

{
  "state":true,
  "code":"000000",
  "message": "",
  "result":{
    "bundles":[
      {
        "name":"kitty1",
        "img":"http://**.com/Public/bundle/img/kitty1.png",
        "url":"http://**.com/Public/bundle/bun/kitty1.bundle",
      }
    ]
  }
}
  
```

设置访问 http 域名白名单:



App Transport Security Settings	Dictionary	(2 items)
Exception Domains	Dictionary	(1 item)
**.com	Dictionary	(3 items)
NSIncludesSubdomains	Boolean	YES
NSEnvironmentAllowsInsecureHTT...	Boolean	YES
NSEnvironmentRequiresForwardSe...	Boolean	NO

3.12.2 自定义 MV 资源

通过以下配置启用 MV，并设置获取 MV 资源的 url

```

rdVESDK.editConfiguration.enableMV = false;
rdVESDK.editConfiguration.mvResourceURL = @"http://**";
  
```

获取 MV 资源的 url 需满足以下定义:

请求格式:

URL	http://**
支持格式	JSON
HTTP 请求验证	POST/GET
请求次数限制	否

参数说明:

参数名	必选	类型及范围	说明
type	是	String	ios android

正常返回结果:

返回键	类型	返回值	说明
state	Boolean	请求状态	true
code	Int	请求状态码	200正常
message	String	请求状态文本信息	
result	Json	请求返回参数	
mvlist	Array	mv列表	
enname	String	mv英文名称	
img	String	mv图片	
name	String	mv名称	
url	String	mv链接	

```

{
  state = 1;
  code = 200;
  message = "";
  result = {
    mvlist = (
      {
        enname = fengjing;
        img = "http://**.com/dianbookupload/shortvideo/mv/fengjing.png";
        name = "风景";
        url = "http://**.com/dianbookupload/shortvideo/mv/fengjing.zip";
      }
    );
  };
};
    
```

3.12.3 自定义 Music 资源

通过以下配置启用自定义 music ,并设置获取 music 资源的 url

```
rdVESDK.editConfiguration.musicResourceURL= @"http:// **"; //若不设置 url 自动隐藏该功能
```

获取 Music 资源的 url 需满足以下定义:

请求格式:

URL	http://**
支持格式	JSON
HTTP 请求验证	POST/GET
请求次数限制	否

参数说明:

参数名	必选	类型及范围	说明
type	是	String	ios android

正常返回结果:

返回键	类型	返回值	说明
state	Boolean	请求状态	true
code	Int	请求状态码	200正常
message	String	请求状态文本信息	
result	Json	请求返回参数	
bgmusic	Array	music列表	
icon	String	music图片	
name	String	music名称	不能为null, 且唯一
url	String	music链接	

```

{
  state = 1;
  code = 200;
  message = "";
  result = {
    bgmusic = (
      {
        icon = "http://**.com/dianbookupload/shortvideo/bgmusic/saye.png";
        name = "\U6492\U91ce";
        url = "http://**.com/dianbookupload/shortvideo/bgmusic/saye.mp3";
      }
    );
  };
}
  
```

3.12.4 自定义云音乐资源

通过以下配置启用自定义云音乐, 并设置获取云音乐资源的 url

```
rdVESDK.editConfiguration.cloudMusicResourceURL= @"http://**"; //若不设置 url 自动隐藏该功能
```

获取云音乐资源的 url 需满足以下定义:

请求格式:

URL	http://**
支持格式	JSON
HTTP 请求验证	POST/GET
请求次数限制	否

参数说明:

参数名	必选	类型及范围	说明
type	是	String	ios android

正常返回结果:

返回键	类型	返回值	说明
state	Boolean	请求状态	true
code	Int	请求状态码	200正常
message	String	请求状态文本信息	
result	Json	请求返回参数	
bgmusic	Array	music总列表	
name	String	music分类名称	
icon	String	music分类图片	
musiclist	Array	music分类下的音乐列表	
name	String	music名称	不能为null, 且唯一
times	Int	music时长	单位: 秒 >0
url	String	music链接	

```

{
  state = 1;
  code = 200;
  message = "";
  result = {
    bgmusic = (
      {
        icon = "http://**.com/dianbookupload/shortvideo/cloudmusic/caiyi.png";
        musiclist = (
          {
            name = "Boys(摇滚合唱版)";
            times = 262;
            url = "http://**.com/dianbookupload/shortvideo/cloudmusic/boys.mp3";
          },
          name = "摇滚";
        );
      }
    );
  };
}
    
```

3.13 分享视频

开发中

3.14 wifi 分享视频

开发中