

腾讯特效 SDK

旧版文档



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

旧版文档

美颜参数表

Android (3.3.0及以前)

iOS (3.3.0及以前)

一分钟集成腾讯特效

iOS

Android

一分钟集成直播

iOS (3.3.0及以前)

Android (3.3.0及以前)

一分钟集成 TRTC

iOS (3.3.0及以前)

Android (3.3.0及以前)

一分钟集成短视频

iOS

Android

鉴权结果及错误码

Android

旧版文档

美颜参数表

Android（3.3.0及以前）

最近更新时间：2024-01-30 17:37:11

美颜

美白-自然

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_WHITEN	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	30	30
	innerMinValue	0	0
	innerMaxValue	1	1.3

美白-粉白(V3.2.0.4新增)

字段		取值
category		Category.BEAUTY
id		null
resPath		null

effKey		BeautyConstant.BEAUTY_WHITEN2	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	30	30
	innerMinValue	0	0
	innerMaxValue	1	1.3

美白-冷白(V3.2.0.4新增)

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_WHITEN3	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	30	30
	innerMinValue	0	0
	innerMaxValue	1	1.3

磨皮

字段		取值	
category		Category.BEAUTY	

id		null
resPath		null
effKey		BeautyConstant.BEAUTY_SMOOTH
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	50
	innerMinValue	0
	innerMaxValue	1

红润

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_ROSY
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	20
	innerMinValue	0
	innerMaxValue	1

画面调整

对比度

字段	取值
----	----

category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_CONTRAST
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

饱和度

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_SATURATION
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

清晰度

字段	取值
----	----

category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_CLEAR
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

锐化(V3.2.0.4新增)

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.SMOOTH_SHA RPEN
effValue(XmagicPropertyVal ues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultVal ue	0
	innerMinValue	0
	innerMaxValue	1

微整形

大眼

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_ENLARGE_EYE
effValue(XmagicProperty Values)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	20
	innerMinValue	0
	innerMaxValue	1

瘦脸-自然

字段		取值
category		Category.BEAUTY
id		BeautyConstant.BEAUTY_FACE_NATURE_ID
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_NATURE
effValue(XmagicProperty Values)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	30

	innerMinValue	0
	innerMaxValue	1

瘦脸-女神

字段		取值
category		Category.BEAUTY
id		BeautyConstant.BEAUTY_FACE_FEMALE_GOD_ID
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_GODNESS
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

瘦脸-英俊

字段		取值
category		Category.BEAUTY
id		BeautyConstant.BEAUTY_FACE_MALE_GOD_ID
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_MALE_GOD

effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

V 脸

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_FACE_V	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	30	30
	innerMinValue	0	0
	innerMaxValue	1	1.3

收下颌（V2.6.0新增）

字段	取值
category	Category.BEAUTY
id	null

resPath		null
effKey		BeautyConstant.BEAUTY_FACE_JAW
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

窄脸

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_THIN
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

短脸

字段		取值
category		Category.BEAUTY
id		null

resPath		null	
effKey		BeautyConstant.BEAUTY_FACE_S HORT	
effValue(XmagicPropertyV alues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultVa lue	0	0
	innerMinValue	0	0
	innerMaxValue	1	1.3

脸型

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_B ASIC
effValue(XmagicPropertyV alues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultVa lue	0
	innerMinValue	0
	innerMaxValue	1

瘦脸骨

字段	取值
category	Category.BEAUTY

id		null
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_THIN_C HEEKBONE
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

下巴

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_THIN_CHIN
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

额头

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_FOR EHEAD
effValue(XmagicProperty Values)	displayMinValu e	-100
	displayMaxVal ue	100
	displayDefault Value	0
	innerMinValue	-1
	innerMaxValue	1

亮眼

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_EYE_LIG HTEN	
effValue(XmagicPropertyV alues)	参数	普通模式	增强模式
	displayMinValu e	0	0
	displayMaxValu e	100	100

	displayDefaultV alue	30	30
	innerMinValue	0	0
	innerMaxValue	1	1.5

眼距

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_EYE_DIS TANCE	
effValue(XmagicPropertyV alues)	参数	普通模式	增强模式
	displayMinValu e	-100	-100
	displayMaxValu e	100	100
	displayDefaultV alue	0	0
	innerMinValue	-1	-1.3
	innerMaxValue	1	1.3

眼角

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_EYE_AN GLE

effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

眼宽 (V3.0.0新增)

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_EYE_WIDTH
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

眼高 (V3.0.0新增)

字段		取值
category		Category.BEAUTY
id		null

resPath		null
effKey		BeautyConstant.BEAUTY_EYE_HEIGHT
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

眉毛角度（V3.0.0新增）

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_EYEBROW_ANGLE
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

眉毛距离（V3.0.0新增）

字段	取值
----	----

category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_EYEBROW_DISTANCE
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

眉毛高度（V3.0.0新增）

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_EYEBROW_HEIGHT
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

眉毛长度（V3.0.0新增）

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_EYEBROW_LENGTH
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

眉毛粗细（V3.0.0新增）

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_EYEBROW_THICKNESS
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0

	innerMinValue	-1
	innerMaxValue	1

眉峰（V3.0.0新增）

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_EYEBROW_RIDGE
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

瘦鼻

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_NOSE_THIN
effValue(XmagicPropertyValues)	displayMinValue	0

	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

鼻翼

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_NOSE_WING
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

鼻子位置

字段		取值
category		Category.BEAUTY
id		null
resPath		null

effKey		BeautyConstant.BEAUTY_NOSE_H EIGHT	
effValue(XmagicPropertyV alues)	参数	普通模式	增强模式
	displayMinValu e	-100	-100
	displayMaxValu e	100	100
	displayDefaultV alue	0	0
	innerMinValue	-1	-1.3
	innerMaxValue	1	1.3

鼻梁(V3.3.0.2新增)

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_NOSE_B RIDGE_WIDTH	
effValue(XmagicPropertyV alues)	参数	普通模式	增强模式
	displayMinValu e	-100	-100
	displayMaxValu e	100	100
	displayDefaultV alue	0	0
	innerMinValue	-1	-1.3
	innerMaxValue	1	1.3

山根(V3.3.0.2新增)

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_NASION	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	-100	-100
	displayMaxValue	100	100
	displayDefaultValue	0	0
	innerMinValue	-1	-1.3
	innerMaxValue	1	1.3

白牙

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_TOOTH_WHITEN
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0

	innerMaxValue	1
--	---------------	---

祛皱

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_FACE_REMOVE_Wrinkle	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	0	0
	innerMinValue	0	0
	innerMaxValue	1	1.3

祛法令纹

字段		取值	
category		Category.BEAUTY	
id		null	
resPath		null	
effKey		BeautyConstant.BEAUTY_FACE_REMOVE_LAW_LINE	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0

	displayMaxValue	100	100
	displayDefaultValue	0	0
	innerMinValue	0	0
	innerMaxValue	1	1.3

祛眼袋

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_FACE_REMOVE_EYE_BAGS
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

嘴型

字段		取值
category		Category.BEAUTY
id		null
resPath		null

effKey		BeautyConstant.BEAUTY_MOUTH_SIZE
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

嘴唇厚度

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_MOUTH_HEIGHT
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

嘴唇宽度（V3.0.0新增）

字段		取值
category		Category.BEAUTY

id		null
resPath		null
effKey		BeautyConstant.BEAUTY_MOUTH_WIDTHH
effValue(XmagicProperty Values)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

嘴唇位置（V3.0.0新增）

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_MOUTH_POSITION
effValue(XmagicProperty Values)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

微笑唇（V3.0.0新增）

字段		取值
category		Category.BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BEAUTY_SMILE_FACE
effValue(XmagicPropertyValues)	displayMinValue	-100
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	-1
	innerMaxValue	1

单点妆容

口红-复古红

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_LIPS_LIPS_MASK	
resPath		/images/beauty/lips_fuguhong.png	
effKey		BeautyConstant.BEAUTY_MOUTH_LIPSTICK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100

	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.3

口红-蜜桃色

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_LIPS_LIPS_MASK	
resPath		/images/beauty/lips_mitaose.png	
effKey		BeautyConstant.BEAUTY_MOUTH_LIPSTICK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.3

口红-珊瑚橘

字段		取值	
category		Category.BEAUTY	

id		XmagicConstant.BeautyConstant.BEAUTY_LIPS_LIPS_MASK	
resPath		/images/beauty/lips_shanhuju.png	
effKey		BeautyConstant.BEAUTY_MOUTH_LIPSTICK	
effValue(XmagicPropertyValues)	参数	普通模式	增模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.3

口红-温柔粉

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_LIPS_LIPS_MASK	
resPath		/images/beauty/lips_wenroufen.png	
effKey		BeautyConstant.BEAUTY_MOUTH_LIPSTICK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100

	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.3

口红-活力橙

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_LIPS_LIPS_MASK	
resPath		/images/beauty/lips_huolicheng.png	
effKey		BeautyConstant.BEAUTY_MOUTH_LIPSTICK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.3

腮红-简约

字段	取值
category	Category.BEAUTY

id		XmagicConstant.BeautyConstant.BEAUTY_MAKEUP_MULTIPLY_MULTIPLY_MASK	
resPath		/images/beauty/saihong_jianyue.png	
effKey		BeautyConstant.BEAUTY_FACE_RED_CHECK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.8

腮红-盛夏

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_MAKEUP_MULTIPLY_MULTIPLY_MASK	
resPath		/images/beauty/saihong_shengxia.png	
effKey		BeautyConstant.BEAUTY_FACE_RED_CHECK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100

	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.8

腮红-害羞

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_MAKEUP_MULTIPLY_MULTIPLY_MASK	
resPath		/images/beauty/saihong_haixiu.png	
effKey		BeautyConstant.BEAUTY_FACE_RED_CHEEK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.8

腮红-成熟

字段	取值
category	Category.BEAUTY

id		XmagicConstant.BeautyConstant.BEAUTY_MAKEUP_MULTIPLY_MULTIPLY_MASK	
resPath		/images/beauty/saihong_chengshu.png	
effKey		BeautyConstant.BEAUTY_FACE_RED_CHECK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.8

腮红-雀斑

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_MAKEUP_MULTIPLY_MULTIPLY_MASK	
resPath		/images/beauty/saihong_queban.png	
effKey		BeautyConstant.BEAUTY_FACE_RED_CHECK	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100

	displayDefault tValue	50	50
	innerMinValu e	0	0
	innerMaxVal ue	1	1.8

眼影（V3.1.0新增）

字段		取值
category		Category.BEAUTY
id		beauty.eyesMakeup.mask.eyeSha dow
resPath		/images/beauty/eyes_makeup_eye _shadow_0-albatross.png //此处填 写的是资源路径，更多眼影效果可参考 demo
effKey		BeautyConstant.BEAUTY_FACE_ MAKEUP_EYE_SHADOW
effValue(XmagicPropertyV alues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultVal ue	50
	innerMinValue	0
	innerMaxValue	1

眼线（V3.1.0新增）

字段		取值
category		Category.BEAUTY
id		beauty.eyesMakeup.mask.eyeLine r

resPath		/images/beauty/eyes_makeup_eye_liner_0.png //此处填写的是资源路径，更多眼线效果可参考demo
effKey		BeautyConstant.BEAUTY_FACE_MAKEUP_EYE_LINER
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	50
	innerMinValue	0
	innerMaxValue	1

睫毛（V3.1.0新增）

字段		取值
category		Category.BEAUTY
id		beauty.eyesMakeup.mask.eyelash
resPath		/images/beauty/eyes_makeup_eyelash_0.png //此处填写的是资源路径，更多睫毛效果可参考demo
effKey		BeautyConstant.BEAUTY_FACE_MAKEUP_EYELASH
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	50
	innerMinValue	0
	innerMaxValue	1

眉毛（V3.1.0新增）

字段		取值
category		Category.BEAUTY
id		beauty.eyesMakeup.mask.eyebrow
resPath		/images/beauty/eyes_makeup_eyebrow_0.png //此处填写的是资源路径，更多眉毛效果可参考demo
effKey		BeautyConstant.BEAUTY_FACE_MAKEUP_EYEBROW
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	50
	innerMinValue	0
	innerMaxValue	1

美瞳（V3.3.0.2新增）

字段		取值
category		Category.BEAUTY
id		beauty.eyesMakeup.mask.eyeball
resPath		/images/beauty/eyes_makeup_eyeball_0.png //此处填写的是资源路径，更多美瞳效果可参考demo
effKey		BeautyConstant.BEAUTY_FACE_MAKEUP_EYEBALL
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	50

	ue	
	innerMinValue	0
	innerMaxValue	1

立体-自然

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_SOFTLIGHT_SOFTLIGHT_MASK	
resPath		/images/beauty/liti_ziran.png	
effKey		BeautyConstant.BEAUTY_FACE_SOFTLIGHT	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.3

立体-俊朗

字段		取值
category		Category.BEAUTY
id		XmagicConstant.BeautyConstant.BEAUTY_SOFTLIGHT_SOFTLIGHT_MASK
resPath		/images/beauty/liti_junlang.png

effKey		BeautyConstant.BEAUTY_FACE_SOFTLIGHT	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.3

立体-光芒

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_SOFTLIGHT_SOFTLIGHT_MASK	
resPath		/images/beauty/liti_guangmang.png	
effKey		BeautyConstant.BEAUTY_FACE_SOFTLIGHT	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0

	innerMaxValue	1	1.3
--	---------------	---	-----

立体-清新

字段		取值	
category		Category.BEAUTY	
id		XmagicConstant.BeautyConstant.BEAUTY_SOFTLIGHT_SOFTLIGHT_MASK	
resPath		/images/beauty/liti_qingxin.png	
effKey		BeautyConstant.BEAUTY_FACE_SOFTLIGHT	
effValue(XmagicPropertyValues)	参数	普通模式	增强模式
	displayMinValue	0	0
	displayMaxValue	100	100
	displayDefaultValue	50	50
	innerMinValue	0	0
	innerMaxValue	1	1.3

美体

一键瘦身

字段	取值
category	Category.BODY_BEAUTY
id	null
resPath	null

effKey		BeautyConstant.BODY_AUTOTHIN_BODY_STRENGTH
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

长腿

字段		取值
category		Category.BODY_BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BODY_LEG_STRETCH
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

瘦腿

字段	取值
----	----

category		Category.BODY_BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BODY_SLIM_LEG_STRENGTH
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

瘦腰

字段		取值
category		Category.BODY_BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BODY_WAIST_STRENGTH
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0

	innerMaxValue	1
--	---------------	---

瘦肩

字段		取值
category		Category.BODY_BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BODY_THIN_SHOULDER_STRENGTH
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

小头

字段		取值
category		Category.BODY_BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BODY_SLIM_HEAD_STRENGTH
effValue(XmagicPropertyValues)	displayMinValue	0

	displayMaxVal ue	100
	displayDefault Value	0
	innerMinValue	0
	innerMaxValu e	1

丰胸（V3.0.0新增）

字段		取值
category		Category.BODY_BEAUTY
id		null
resPath		null
effKey		BeautyConstant.BODY_ENLARGE_CHE ST_STRENGTH
effValue(XmagicProperty Values)	displayMinVal ue	0
	displayMaxVal ue	100
	displayDefault Value	0
	innerMinValue	0
	innerMaxValu e	1

滤镜

无

字段		取值
category		Category.LUT
id		XmagicProperty.ID_NONE

resPath		null
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	
	innerMinValue	
	innerMaxValue	

白哲

字段		取值
category		Category.LUT
id		n_baixi.png
resPath		xxxxxxx/xmagic/light_material/lut/n_baixi.png
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

其他

字段		取值
category		Category.LUT
id		滤镜图片资源的名称（参考 Demo）

resPath		滤镜图片资源的全路径（参考 Demo）
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	0
	innerMinValue	0
	innerMaxValue	1

动效

无

字段		取值
category		Category.MOTION
id		XmagicProperty.ID_NONE
resPath		xxxxxx /light_assets/template.json（此处需要参考 Demo，填写全路径）
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	
	innerMinValue	
	innerMaxValue	

可爱涂鸦

字段		取值
category		Category.MOTION
id		video_keaituya
resPath		xxxxxx/xmagic/MotionRes/2dMotionRes/video_keaituya
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	
	innerMinValue	
	innerMaxValue	

动效素材名称

字段		取值
category		Category.MOTION
id		素材文件夹名称（参考 Demo）
resPath		素材文件夹全路径（参考 Demo）
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	
	innerMinValue	
	innerMaxValue	

美妆

无

字段		取值
category		Category.MAKEUP
id		XmagicProperty.ID_NONE
resPath		xxxxxx/light_assets/template.json（此处需要参考 Demo，填写全路径）
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	
	innerMinValue	
	innerMaxValue	

女团妆

字段		取值
category		Category.MAKEUP
id		video_nvtuanzhuang
resPath		xxxxxx/xmagic/MotionRes/makeupRes/video_nvtuanzhuang
effKey		XmagicConstant.MakeUpEffKey.MAKEUP_EFF_KEY
effValue(XmagicPropertyValues)	displayMinValue	0

	displayMaxValue	100
	displayDefaultValue	60
	innerMinValue	0
	innerMaxValue	1

美妆素材名称

字段		取值
category		Category.MAKEUP
id		素材文件夹名称（参考 Demo）
resPath		素材文件夹全路径（参考 Demo）
effKey		XmagicConstant.MakeUpEffKey.MAKEUP_EFF_KEY
effValue(XmagicPropertyValues)	displayMinValue	0
	displayMaxValue	100
	displayDefaultValue	60
	innerMinValue	0
	innerMaxValue	1

分割

无

字段	取值
category	Category.SEGMENTATION

id		XmagicProperty.ID_NONE
resPath		xxxxxx/light_assets/template.json（此处需要参考 Demo，填写全路径）
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	
	innerMinValue	
	innerMaxValue	

背景模糊-弱

字段		取值
category		Category.SEGMENTATION
id		video_segmentation_blur_45
resPath		xxxxxx/xmagic/MotionRes/segmentMotionRes/video_segmentation_blur_45
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	
	innerMinValue	

	innerMaxValue	
--	---------------	--

分割素材名称

字段		取值
category		Category.SEGMENTATION
id		素材文件夹名称（参考 Demo）
resPath		素材文件夹全路径（参考 Demo）
effKey		null
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	
	innerMinValue	
	innerMaxValue	

自定义分割

字段		取值
category		Category.SEGMENTATION
id		XmagicConstant.SegmentationId.CUSTOM_SEG_ID
resPath		自定义分割素材的全路径
effKey		图片本地路径/视频本地路径
effValue(XmagicPropertyValues)	displayMinValue	null
	displayMaxValue	
	displayDefaultValue	

	innerMinValue	
	innerMaxValue	

绿幕分割

字段		取值
category		Category.SEGMENTATION
id		XmagicConstant.SegmentationId.GREEN_SCREEN_SEG_ID
resPath		绿幕分割素材的全路径
effKey		图片本地路径/视频本地路径
effValue(XmagicPropertyValues)	hashMapProperty	key="keyColor", value="#00FF00", 绿幕颜色，可选，默认为绿色，RGB字符串

iOS（3.3.0及以前）

最近更新时间：2023-11-02 17:22:42

美颜

美白-自然

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_WHITEN
propertyValue	30
extraInfo	nil

美白-粉白(V3.2.0.1新增)

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_WHITEN2
propertyValue	30
extraInfo	nil

美白-冷白(V3.2.0.1新增)

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_WHITEN3
propertyValue	30
extraInfo	nil

磨皮

字段	取值
----	----

propertyType	Category_BEAUTY
propertyName	BEAUTY_SMOOTH
propertyValue	30
extraInfo	nil

红润

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_ROSY
propertyValue	30
extraInfo	nil

对比度

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_CONTRAST
propertyValue	30
extraInfo	nil

饱和度

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_SATURATION
propertyValue	30
extraInfo	nil

清晰度

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_CLEAR
propertyValue	30
extraInfo	nil

锐化(V3.2.0.1新增)

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_SHAPE
propertyValue	30
extraInfo	nil

大眼

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_ENLARGE_EYE
propertyValue	30
extraInfo	nil

瘦脸/自然

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_NATURE
propertyValue	30
extraInfo	@{"reshape.basicFaceSubType":@"nature"}

瘦脸/女神

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_GODNESS
propertyValue	30
extraInfo	@{"reshape.basicFaceSubType":@"femaleGod"}

瘦脸/英俊

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_MALE_GOD
propertyValue	30
extraInfo	@{"reshape.basicFaceSubType":@"maleGod"}

V 脸

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_V
propertyValue	30
extraInfo	nil

收下颌（V2.6.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_JAW
propertyValue	30
extraInfo	nil

窄脸

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_THIN
propertyValue	30
extraInfo	nil

短脸

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_SHORT
propertyValue	30
extraInfo	nil

脸型

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_BASIC
propertyValue	30
extraInfo	nil

口红/复古红

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_MOUTH_LIPSTICK

propertyValue	30
extraInfo	@{"beauty.lips.lipsMask":@"images/beauty/lips_fuguhong.png", @"beauty.lips.lipsType":@"2"}

口红/蜜桃色

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_MOUTH_LIPSTICK
propertyValue	30
extraInfo	@{"beauty.lips.lipsMask":@"images/beauty/lips_mitaose.png", @"beauty.lips.lipsType":@"2"}

口红/珊瑚橘

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_MOUTH_LIPSTICK
propertyValue	30
extraInfo	@{"beauty.lips.lipsMask":@"images/beauty/lips_shanhuju.png", @"beauty.lips.lipsType":@"2"}

口红/温柔粉

字段	取值
propertyType	Category_BEAUTY

propertyName	BEAUTY_MOUTH_LIPSTICK
propertyValue	30
extraInfo	@{"beauty.lips.lipsMask":@"images/beauty/lips_wenroufen.png", @"beauty.lips.lipsType":@"2"}

口红/活力橙

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_MOUTH_LIPSTICK
propertyValue	30
extraInfo	@{"beauty.lips.lipsMask":@"images/beauty/lips_huolicheng.png", @"beauty.lips.lipsType":@"2"}

腮红/简约

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_RED_CHEEK
propertyValue	30
extraInfo	@{"beauty.makeupMultiply.multiplyMask":@"images/beauty/saihong_jian yue.png"}

腮红/盛夏

字段	取值
----	----

propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_RED_CHEEK
propertyValue	30
extraInfo	@{"beauty.makeupMultiply.multiplyMask":@"images/beauty/saihong_shengxia.png"}

腮红/害羞

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_RED_CHEEK
propertyValue	30
extraInfo	@{"beauty.makeupMultiply.multiplyMask":@"images/beauty/saihong_haixiu.png"}

腮红/成熟

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_RED_CHEEK
propertyValue	30
extraInfo	@{"beauty.makeupMultiply.multiplyMask":@"images/beauty/saihong_chengshu.png"}

腮红/雀斑

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_RED_CHEEK
propertyValue	30
extraInfo	@{"beauty.makeupMultiply.multiplyMask":@"images/beauty/saihong_queban.png"}

立体/自然

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_SOFTLIGHT
propertyValue	30
extraInfo	@{"beauty.softLight.softLightMask":@"images/beauty/liti_ziran.png"}

立体/俊朗

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_SOFTLIGHT
propertyValue	30
extraInfo	@{"beauty.softLight.softLightMask":@"images/beauty/liti_junlang.png"}

立体/光芒

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_SOFTLIGHT
propertyValue	30
extraInfo	@{"beauty.softLight.softLightMask":@"images/beauty/liti_guangmang.png"}

立体/清新

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_SOFTLIGHT
propertyValue	30
extraInfo	@{"beauty.softLight.softLightMask":@"images/beauty/liti_qingxin.png"}

瘦脸骨

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_THIN_CHEEKBONE
propertyValue	30
extraInfo	nil

下巴

字段	取值
propertyType	Category_BEAUTY

propertyName	BEAUTY_FACE_THIN_CHIN
propertyValue	30
extraInfo	nil

额头

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_FOREHEAD
propertyValue	30
extraInfo	nil

亮眼

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYE_LIGHTEN
propertyValue	30
extraInfo	nil

眼距

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYE_DISTANCE
propertyValue	30
extraInfo	nil

眼角

字段	取值
----	----

propertyType	Category_BEAUTY
propertyName	BEAUTY_EYE_ANGLE
propertyValue	30
extraInfo	nil

眼宽（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYE_WIDTH
propertyValue	30
extraInfo	nil

眼高（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYE_HEIGHT
propertyValue	30
extraInfo	nil

瘦鼻

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_NOSE_THIN
propertyValue	30
extraInfo	nil

鼻翼

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_NOSE_WING
propertyValue	30
extraInfo	nil

鼻子位置

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_NOSE_HEIGHT
propertyValue	30
extraInfo	nil

鼻梁(V3.3.0.1新增)

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_NOSE_BRIDGE_WIDTH
propertyValue	30
extraInfo	nil

山根(V3.3.0.1新增)

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_NASION
propertyValue	30
extraInfo	nil

白牙

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_TOOTH_WHITEN
propertyValue	30
extraInfo	nil

祛皱

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_REMOVE_WRINKLE
propertyValue	30
extraInfo	nil

祛法令纹

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_REMOVE_LAW_LINE
propertyValue	30
extraInfo	nil

祛眼袋

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_REMOVE_EYE_BAGS
propertyValue	30
extraInfo	nil

嘴型

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_MOUTH_SIZE
propertyValue	30
extraInfo	nil

嘴唇厚度

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_MOUTH_HEIGHT
propertyValue	30
extraInfo	nil

嘴唇宽度（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_MOUTH_WIDTH
propertyValue	30
extraInfo	nil

嘴唇位置（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_MOUTH_POSITION
propertyValue	30
extraInfo	nil

微笑唇（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_SMILE_FACE
propertyValue	30
extraInfo	nil

眉毛角度（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYEBROW_ANGLE
propertyValue	30
extraInfo	nil

眉毛距离（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYEBROW_DISTANCE
propertyValue	30
extraInfo	nil

眉毛高度（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYEBROW_HEIGHT
propertyValue	30
extraInfo	nil

眉毛长度（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYEBROW_LENGTH
propertyValue	30
extraInfo	nil

眉毛粗细（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYEBROW_THICKNESS
propertyValue	30
extraInfo	nil

眉峰（V3.0.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_EYEBROW_RIDGE
propertyValue	30
extraInfo	nil

眼影（V3.1.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_EYE_SHADOW
propertyValue	30

extraInfo	<pre>@{ @"beauty.eyesMakeup.mask.eyeShadow" : @"/images/beauty/eyes_makeup_eye_shadow_0-albatross.png" }</pre> 说明：传入字典，字典的 key 是@"beauty.eyesMakeup.mask.eyeShadow",value 是效果图片的相对路径@"/images/beauty/eyes_makeup_eye_shadow_0-albatross.png"。具体美颜效果参考 demo
-----------	---

眼线（V3.1.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_EYE_LINER
propertyValue	30
extraInfo	<pre>@{ @"beauty.eyesMakeup.mask.eyeLiner" : @"/images/beauty/eyes_makeup_eye_liner_0.png" }</pre> 说明：传入字典，字典的 key 是@"beauty.eyesMakeup.mask.eyeLiner",value 是效果图片的相对路径@"/images/beauty/eyes_makeup_eye_liner_0.png"。具体美颜效果参考 demo

睫毛（V3.1.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_EYELASH
propertyValue	30
extraInfo	<pre>@{ @"beauty.eyesMakeup.mask.eyelash" : @"/images/beauty/eyes_makeup_eyelash_0.png" }</pre>

说明：传入字典，字典的 key 是@"beauty.eyesMakeup.mask.eyelash",value 是效果图片的相对路径@"/images/beauty/eyes_makeup_eyelash_0.png"。具体美颜效果参考 demo

眉毛（V3.1.0新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_EYEBROW
propertyValue	30
extraInfo	<pre>@{ @"beauty.eyesMakeup.mask.eyebrow" : @"/images/beauty/eyes_makeup_eyebrow_0.png" }</pre> 说明：传入字典，字典的 key 是@"beauty.eyesMakeup.mask.eyebrow",value 是效果图片的相对路径@"/images/beauty/eyes_makeup_eyebrow_0.png"。具体美颜效果参考 demo

美瞳（V3.3.0.1新增）

字段	取值
propertyType	Category_BEAUTY
propertyName	BEAUTY_FACE_EYEBALL
propertyValue	30
extraInfo	<pre>@{ @"beauty.eyesMakeup.mask.eyeball" : @"/images/beauty/eyes_makeup_eyeball_0.png" }</pre> 说明：传入字典，字典的 key 是@"beauty.eyesMakeup.mask.eyeball",value 是效果图片的相对路径@"/images/beauty/eyes_makeup_eyeball_0.png"。具体美颜效果参考 demo

美体

一键瘦身

字段	取值
propertyType	Category_BODY_BEAUTY
propertyName	BODY_AUTOTHIN_BODY_STRENGTH
propertyValue	30
extraInfo	nil

长腿

字段	取值
propertyType	Category_BODY_BEAUTY
propertyName	BODY_LEG_STRETCH
propertyValue	30
extraInfo	nil

瘦腿

字段	取值
propertyType	Category_BODY_BEAUTY
propertyName	BODY_SLIM_LEG_STRENGTH
propertyValue	30
extraInfo	nil

瘦腰

字段	取值
propertyType	Category_BODY_BEAUTY
propertyName	BODY_WAIST_STRENGTH
propertyValue	30

extraInfo	nil
-----------	-----

瘦肩

字段	取值
propertyType	Category_BODY_BEAUTY
propertyName	BODY_THIN_SHOULDER_STRENGTH
propertyValue	30
extraInfo	nil

丰胸（V3.0.0新增）

字段	取值
propertyType	Category_BODY_BEAUTY
propertyName	BODY_ENLARGE_CHEST_STRENGTH
propertyValue	30
extraInfo	nil

小头

字段	取值
propertyType	Category_BODY_BEAUTY
propertyName	BODY_SLIM_HEAD_STRENGTH
propertyValue	30
extraInfo	nil

滤镜

无

字段	取值
propertyType	Category_LUT

propertyName	XmagicProperty_ID_NONE
propertyValue	30
extraInfo	nil

白皙（例子）

字段	取值
propertyType	Category_LUT
propertyName	lut.bundle/n_baixi.png
propertyValue	30
extraInfo	nil

其他

字段	取值
property Type	Category_LUT
property Name	滤镜图片资源的路径（参考 Demo，如果资源在 bundle 中，则可以使用相对路径，否则使用全路径）
property Value	30
extraInfo	nil

动效

无

字段	取值
propertyType	Category_MOTION
propertyName	XmagicProperty_ID_NONE
propertyValue	xxxxxx/2dMotionRes.bundle
extraInfo	nil

可爱涂鸦

字段	取值
propertyType	Category_MOTION
propertyName	video_keaituya
propertyValue	xxxxxxx/2dMotionRes.bundle
extraInfo	nil

动效素材名称

字段	取值
propertyType	Category_MOTION
propertyName	素材文件夹名称（参考 Demo）
propertyValue	素材文件夹全路径（此处需要参考 Demo，填写全路径）
extraInfo	nil

美妆

美妆时，先设置 motion，后设置 makeup，具体请参考 Demo。

无

字段	取值	
propertyType	Category_MOTION	Category_MAKEUP
propertyName	XmagicProperty_ID_NONE	BeautyConstant_MAKEUP
propertyValue	xxxxxxx/makeupMotionRes.bundle	0
extraInfo	nil	nil

女团妆

字段	取值	
propertyType	Category_MOTION	Category_MAKEUP
propertyName	video_nvtuanzhuang	BeautyConstant_MAKEUP

propertyValue	xxxxxxx/makeupMotionRes.bundle	30
extraInfo	nil	nil

美妆素材名称

字段	取值	
propertyType	Category_MOTION	Category_MAKEUP
propertyName	素材文件夹名称（参考 Demo）	BeautyConstant_MAKEUP
propertyValue	素材文件夹全路径（参考 Demo，填写全路径）	30
extraInfo	nil	nil

分割

无

字段	取值
propertyType	Category_MOTION
propertyName	XmagicProperty_ID_NONE
propertyValue	xxxxxxx/segmentMotionRes.bundle
extraInfo	nil

背景模糊-弱

字段	取值
propertyType	Category_MOTION
propertyName	video_segmentation_blur_45
propertyValue	xxxxxxx/segmentMotionRes.bundle
extraInfo	nil

分割素材名称

字段	取值
----	----

propertyType	Category_MOTION
propertyName	素材文件夹名称（参考 Demo）
propertyValue	素材文件夹全路径（此处需要参考 Demo，填写全路径）
extraInfo	nil

自定义分割

字段	取值
property Type	Category_MOTION
property Name	video_empty_segmentation
property Value	自定义分割素材的全路径
extraInfo	- bgName: 图片或者视频的全路径 - bgType: 如果是图片，则传入0，如果是视频，则传入1 - timeOffset: 如果是图片，则传入0，如果是视频，则传入视频时长 <pre>@{"bgName":bgName, @"bgType":bgType, @"timeOffset": timeOffset}</pre> （参考 Demo）

绿幕分割

字段	取值
property Type	Category_MOTION
property Name	video_greenscreen
property Value	绿幕分割素材的全路径
extraInfo	- bgName: 图片或者视频的全路径 - bgType: 如果是图片，则传入0，如果是视频，则传入1 - timeOffset: 如果是图片，则传入0，如果是视频，则传入视频时长 - keyColor: 绿幕颜色，可选，默认为绿色#00FF00

```
@{@"bgName":bgName, @"bgType":bgType, @"timeOffset": timeOffset,  
@"keyColor":@"#00FF00"}
```

一分钟集成腾讯特效

iOS

最近更新时间：2023-08-28 17:53:42

集成准备

开发者环境要求

- 开发工具 XCode 11 及以上：App Store 或单击 [下载地址](#)。
- 建议运行环境：
 - 设备要求：iPhone 5 及以上；iPhone 6 及以下前置摄像头最多支持到 720p，不支持 1080p。
 - 系统要求：iOS 10.0 及以上。

C/C++层开发环境

XCode 默认 C++ 环境。

类型	依赖库
系统依赖库	● AVFoundation ● Accelerate ● AssetsLibrary ● CoreML ● JavaScriptCore ● CoreFoundation ● MetalPerformanceShaders ● CoreTelephony ● libc++.tbd
自带的库	● YTCommon（鉴权静态库） ● XMagic（美颜静态库） ● libpag（视频解码动态库）

导入资源

资源

- 必需资源包：`LightCore.bundle`
- 分割功能包：`LightSegmentPlugin.bundle`

- 手势功能包: `LightHandPlugin.bundle`
- 3D 功能包: `Light3DPlugin.bundle`

导入方法

- 方法一: 加入到工程资源中即可。
- 方法二: 如果需要指定路径 `initWithRenderSize:assetsDict: (XMagic)` 可通过这里的 `assetsDict` 配置每个资源路径。

配置权限

在 Info.plist 文件中添加相应权限的说明, 否则程序在 iOS 10 系统上会出现崩溃。请在 Privacy – Camera Usage Description 中开启相机权限, 允许 App 使用相机。

集成步骤

步骤一: 签名准备

framework 签名可以直接在 General-->Masonry.framework 和 libpag.framework 选 Embed & Sign。

步骤二: 鉴权

1. 申请授权, 得到 LicenseURL 和 LicenseKEY, 请参见 [License 指引](#)。

⚠ 注意

不需要把 License 文件下载下来放到本地工程里。

2. 在工程 AppDelegate 的 didFinishLaunchingWithOptions 中添加如下代码, 触发 license 下载, 避免在使用前才临时去下载。其中, LicenseURL 和 LicenseKey 是控制台绑定 License 时生成的授权信息。

```
[TESign setKeyUrl:LicenseKey url:LicenseURL];
```

3. 然后在真正要使用美颜功能时, 再去做鉴权。

⚠ 注意

部分代码与 Demo 工程中的代码有差异, 请以本文档描述为准。

```
licenseObserver = [[NSNotificationCenter defaultCenter]
addObserverForName:@"kTESignLicenceLoadNotification" object:nil queue:
[NSOperationQueue mainQueue] usingBlock:^(NSNotification * _Nonnull
note) {
```

```
NSLog(@"test --%@",note.object);
if(note.object != nil){
    int code = [[note.object objectForKey:@"result"] intValue];
    NSLog(@"code --%d",code);
    if(code == 0){
        NSString* lic = [TESign getLicenceInfo];
        if(lic == nil){
            lic = @"";
        }
        int authRet = 0;
        NSError* error = noErr;
        NSData * getJsonData = [lic
dataUsingEncoding:NSUTF8StringEncoding];
        NSDictionary * getDict = [NSJSONSerialization
JSONObjectWithData:getJsonData options:NSJSONReadingMutableContainers
error:&error];
        authRet = [XMagicAuthManager initAuthByString:[getDict
objectForKey:@"TELicense"] withSecretKey:@""];
        NSLog(@"errorCode: %@", authRet: %d", error, authRet);
        if(authRet!=0){
            NSLog(@"licenseInfo: %@", lic);
            UIAlertView * alert = [[UIAlertView alloc]
initWithTitle:@"提醒" message:@"授权失败" delegate:nil
cancelButtonTitle:@"取消" otherButtonTitles: nil];
            [alert show];
            return;
        }
    }
}

[[NSNotificationCenter defaultCenter]
removeObserver:licenseObserver];

}];
[TESign setKeyUrl:@"Key" url:@"LicenseURL"];
```

步骤三：加载 SDK: XMagic.framework

使用腾讯特效 SDK 生命周期大致如下:

1. 加载美颜相关资源。

```
NSDictionary *assetsDict = @{@"core_name":@"LightCore.bundle",
@"root_path":[[NSBundle mainBundle] bundlePath]
```

```
};
```

2. 初始化腾讯特效 SDK。

```
initWithRenderSize:assetsDict: (XMagic)  
self.beautyKit = [[XMagic alloc] initWithRenderSize:previewSize  
assetsDict:assetsDict];
```

3. 腾讯特效 SDK 处理每帧数据并返回相应处理结果。

```
process: (XMagic)
```

```
// 在摄像头回调传入帧数据  
- (void)captureOutput:(AVCaptureOutput *)captureOutput  
didOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer fromConnection:  
(AVCaptureConnection *)connection;  
  
// 获取原始数据，处理每帧的渲染信息  
- (void)mycaptureOutput:(AVCaptureOutput *)captureOutput  
didOutputSampleBuffer:(CMSampleBufferRef)inputSampleBuffer  
fromConnection:(AVCaptureConnection *)connection originImageProcess:  
(BOOL)originImageProcess;  
  
// 使用CPU处理数据  
- (YTProcessOutput*)processDataWithCpuFuc:  
(CMSampleBufferRef)inputSampleBuffer;  
  
// 使用GPU处理数据  
- (YTProcessOutput*)processDataWithGpuFuc:  
(CMSampleBufferRef)inputSampleBuffer;  
  
// 腾讯特效 SDK处理数据接口  
/// @param input 输入处理数据信息  
/// @return 输出处理后的数据信息  
- (YTProcessOutput* _Nonnull)process:(YTProcessInput * _Nonnull)input;
```

4. 释放腾讯特效 SDK。

```
deinit (XMagic)  
// 在需要释放SDK资源的地方调用
```

```
[self.beautyKit deinit]
```

❗ 说明

完成上述步骤后，用户即可根据自己的实际需求控制展示时机以及其他设备相关环境。

常见问题

问题1：编译报错：“unexpected service error: build aborted due to an internal error: unable to write manifest to-xxxx-manifest.xcbuild': mkdir(/data, S_IRWXU | S_IRWXG | S_IRWXO): Read-only file system (30):” ？

1. 前往 **File > Project settings > Build System** 选择 **Legacy Build System**。
2. Xcode 13.0++ 需要在 **File > Workspace Settings** 勾选 **Do not show a diagnostic issue about build system deprecation**。

问题2：iOS 导入资源运行后报错：“Xcode 12.X 版本编译提示 Building for iOS Simulator, but the linked and embedded framework '.framework'...” ？

将 **Buil Settings > Build Options > Validate Workspace** 改为 **Yes**，再单击运行。

❗ 说明

Validate Workspace 改为 Yes 之后编译完成，再改回 No，也可以正常运行，所以这里有这个问题注意下即可。

问题3：滤镜设置没反应？

检查下设置的值是否正确，范围为 0-100，可能值太小了效果不明显。

问题4：iOS demo编译，生成 dSYM 时报错？

```
PhaseScriptExecution CMake\ PostBuild\ Rules
build/XMagicDemo.build/Debug-iphonios/XMagicDemo.build/Script-
81731F743E244CF2B089C1BF.sh
    cd /Users/zhenli/Downloads/xmagic_s106
    /bin/sh -c
/Users/zhenli/Downloads/xmagic_s106/build/XMagicDemo.build/Debug-
iphonios/XMagicDemo.build/Script-81731F743E244CF2B089C1BF.sh

Command /bin/sh failed with exit code 1
```

- 问题原因: `libpag.framework`和`Masonry.framework` 重签名失败。

- 解决方法:

1.1 打开 `demo/copy_framework.sh` 。

1.2 用下述命令查看本机 `cmake` 的路径, 将 `$(which cmake)` 改为本地 `cmake` 绝对路径。

```
which cmake
```

1.3 用自己的签名替换所有 `Apple Development: .....` 。

Android

最近更新时间：2023-08-28 17:53:42

集成准备

文件准备：

文件类型	说明
xmagic-xxx.aar	SDK，必选
../assets/	算法模型、素材资源包，必选
../jniLibs	so 库，必选

导入资源

资源

- 添加上述文件准备的全部 `.aar` 文件到 app 工程 `libs` 目录下。
- 将 SDK 包内的 `assets/` 目录下的全部资源拷贝到 `../src/main/assets` 目录下。
- 将 `jniLibs` 文件夹拷贝到工程的 `../src/main/jniLibs` 目录下。

导入方法

打开 app 模块的 `build.gradle` 添加依赖引用：

```
android{
    ...
    defaultConfig {
        applicationId "修改成与授权lic绑定的appID"
        ....
    }
    packagingOptions {
        pickFirst '**/libc++_shared.so'
    }
}

dependencies{
    ...
    compile fileTree(dir: 'libs', include: ['*.jar', '*.aar'])//添加 *.aar
}
```

包体大小瘦身：动态下载 assets、so、动效资源指引

为了减少包大小，您可以将 SDK 所需的 assets 资源、so 库、以及动效资源 MotionRes（部分基础版 SDK 无动效资源）改为联网下载。在下载成功后，将上述文件的路径设置给 SDK。

我们建议您复用 Demo 的下载逻辑，当然，也可以使用您已有的下载服务。动态下载的详细指引，请参见 [SDK 包体瘦身（Android）](#)。

整体流程

步骤一：鉴权

⚠ 注意

部分代码与 Demo 工程中的代码有差异，请以本文档描述为准。

1. 申请授权，得到 License URL 和 License KEY，请参见 [License 指引](#)。

⚠ 注意

不需要把 License 文件下载下来放到本地工程里。

2. 在 Application 的 onCreate 或相关业务模块的初始化代码中设置 URL 和 KEY，触发 license 下载，避免在使用前才临时去下载。

```
LicenceCheck.getInstance().setXMagicLicense(context, URL, KEY);
```

3. 然后在真正要使用美颜功能时（例如 Demo 的 LaunchActivity.java），再去做鉴权：

```
private void auth() {
    LicenceCheck.getInstance().setListener(new
    LicenceCheck.LicenceCheckListener() {
        @Override
        public void onLicenceLoaded(int result, String reason) {
            //在2.4.0版本，如果无需下载，或者下载失败，不会回调这个方法。（后续版本
            会补齐）
            //如果有下载，且下载成功，会回调。result为LicenceCheck.ERROR_OK表示
            下载下来的license文件是有效的
            if (result == LicenceCheck.ERROR_OK) {
                checkAuth(context);
            }
        }
    });
    //再次触发下载（因为有可能之前在onCreate那里触发下载没有成功）
```

```
LicenceCheck.getInstance().setXMagicLicense(context, URL, KEY);

checkAuth(context);
}

private boolean authorized = false;
private synchronized void checkAuth(Context context) {
    Log.d(TAG, "checkAuth: authorized=" + authorized);
    if (authorized) {
        return;
    }
    LicenceCheck mLicenceCheck = LicenceCheck.getInstance();
    String licenseInfo = mLicenceCheck.getBase64Licence();
    if (TextUtils.isEmpty(licenseInfo)) {
        licenseInfo = mLicenceCheck.getLicensePathBase64();
    }
    if (TextUtils.isEmpty(licenseInfo)) {
        Log.d(TAG, "licenseInfo is empty");
        authorized = false;
    } else {
        Auth.AuthResult result = Auth.authByBase64(context, licenseInfo,
            "");

        String msg = Json.toJsonStr(result);
        Log.d(TAG, "isSucceed=" + result.isSucceed);
        Log.d(TAG, "msg=" + msg);
        authorized = result.isSucceed;
    }

    if (authorized) {
        //TODO 鉴权成功，在这里通知UI刷新、执行下一步操作之类的事情
    }
}
```

❗ 说明

如果鉴权失败，您可参考 [查看鉴权结果及错误码说明](#) 定位失败原因。

步骤二：加载腾讯特效 SDK xmagic-xxx.aar

使用腾讯特效 SDK 生命周期大致如下：

1. 构造美颜 UI 数据，可参考 Demo 工程的

XmagicResParser.java, XmagicPropertyData.java, XmagicUIState.java 代码。

2. 预览布局中添加 GLSurfaceView。

```
<android.opengl.GLSurfaceView
    android:id="@+id/camera_gl_surface_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

3. （可选）快速实现相机。

将 Demo 工程中的 `com.tencent.demo.camera` 目录拷贝到工程中。利用 `PreviewMgr` 类快速实现相机功能。详细实现可参考 Demo 工程的 `MainActivity.java`。

```
//初始化相机
mPreviewMgr = new PreviewMgr();
//将布局的GLSurfaceView示例传入相机工具类
mPreviewMgr.onCreate(mGlSurfaceView);
//注册预览纹理数据回调函数
mPreviewMgr.setCustomTextureProcessor((textureId, textureWidth,
    textureHeight) -> {
    if (mXmagicApi == null) {
        return textureId;
    }
    //调用美颜sdk进行渲染
    int outTexture = mXmagicApi.process(textureId, textureWidth,
    textureHeight);
    return outTexture;
});

//在Activity中的onResume方法中启动相机
mPreviewMgr.onResume(this, 1280, 720);
```

4. 初始化美颜 SDK，建议放在 Activity 的 `onResume()` 方法中。

```
mXmagicApi = new XmagicApi(this, XmagicResParser.getResPath(),
    new XmagicApi.OnXmagicPropertyErrorListener());
```

● 参数

参数	含义
Context context	上下文

String resDir	资源文件目录，V1版本固定写法
OnXmagicPropertyErrorListener errorListener	回调函数实现类

- 返回

错误码含义对照表：

错误码	含义
-1	未知错误
-100	3D 引擎资源初始化失败
-200	不支持 GAN 素材
-300	设备不支持此素材组件
-400	模板 JSON 内容为空
-500	SDK版本过低
-600	不支持分割
-700	不支持 OpenGL
-800	不支持脚本
5000	分割背景图片分辨率超过2160*3840
5001	分割背景图片所需内存不足
5002	分割背景视频解析失败
5003	分割背景视频超过200秒
5004	分割背景视频格式不支持

5. 美颜 SDK 处理每帧数据并返回相应处理结果。

```
int outTexture = mXmagicApi.process(textureId, textureWidth, textureHeight);
```

6. 用于更新指定类型的美颜特效数值。

```
// 可用的入参属性可以从 XmagicResParser.parseRes() 获得
```

```
mXmagicApi.updateProperty(XmagicProperty<?> p);
```

7. 释放美颜 SDK，建议与 Activity 的 `onPause()` 生命周期绑定。

```
//在 Activity 的 onPause 时调用，需要在 OpenGL 线程调用  
mXmagicApi.onPause();
```

8. 添加素材提示语回调函数（方法回调有可能运行在子线程）。

```
mXmagicApi.setTipsListener(new XmagicTipsListener() {  
    final XmagicToast mToast = new XmagicToast();  
    @Override  
    public void tipsNeedShow(String tips, String tipsIcon, int type, int  
duration) {  
        mToast.show(MainActivity.this, tips, duration);  
    }  
  
    @Override  
    public void tipsNeedHide(String tips, String tipsIcon, int type) {  
        mToast.dismiss();  
    }  
});
```

完成上述步骤后，用户即可根据自己的实际需求控制展示时机以及其他设备相关环境。

一分钟集成直播 iOS（3.3.0及以前）

最近更新时间：2025-06-05 10:29:42

集成准备

1. 下载并解压 [Demo 包](#)。
2. 将 Demo工程中的 xmagic 模块（bundle，XmagicIconRes，Xmagic 文件夹）导入到实际项目工程中。
3. 如果使用的 XMagic SDK 版本在2.5.0之前，导入 SDK 目录中的 `libpag.framework`、`Masonry.framework`、`XMagic.framework`、`YTCommonXMagic.framework`。如果使用的 XMagic SDK 版本在2.5.1及以后，导入 SDK 目录中的 `libpag.framework`、`Masonry.framework`、`XMagic.framework`、`YTCommonXMagic.framework`、`Audio2Exp.framework`、`TEFFmpeg.framework` (version3.0.0以后，改名为：`TECodec.framework`)。
4. framework 签名 **General--> Masonry.framework 和 libpag.framework 选 Embed & Sign。**
`YTCommonXMagic.framework` 在版本2.5.1之前选 **Do Not Embed**，在版本2.5.1及以后选 **Embed & Sign**。
5. 将 Bundle ID 修改成与签发的授权一致。

开发者环境要求

- 开发工具 XCode 14 及以上：App Store 或单击 [下载地址](#)。
- 建议运行环境：
 - 设备要求：iPhone 5 及以上；iPhone 6 及以下前置摄像头最多支持到 720p，不支持 1080p。
 - 系统要求：iOS 10.0 及以上。

C/C++层开发环境

XCode 默认 C++ 环境。

类型	依赖库
系统依赖库	● Accelerate ● AssetsLibrary ● AVFoundation ● CoreMedia ● CoreFoundation ● CoreML ● Foundation

	• JavaScriptCore • libc++.tbd • libz.b • libresolv.tbd • libsqlite3.0.tbd • MetalPerformanceShaders • MetalKit • MobileCoreServices • OpneAL • OpneGLES • ReplayKit • SystemConfiguration • UIKit
自带的库	• YTCommon（鉴权静态库） • XMagic（美颜静态库） • libpag（视频解码动态库） • Masonry（控件布局库） • TXLiteAVSDK_Professional • TXFFmpeg(version3.0.0以后，改名为：TECodec.framework) • TXSoundTouch • Audio2Exp（xmagic sdk version在2.5.1及以后的版本才有） • TEFFmpeg（xmagic sdk version在2.5.1及以后的版本才有）

SDK 接口集成

- [步骤一](#) 和 [步骤二](#) 可参考 Demo 工程中，`ThirdBeautyViewController` 类 `viewDidLoad`，`buildBeautySDK` 方法；`AppDelegate` 类的 `application` 方法进行了Xmagic鉴权。
- [步骤四](#) 至 [步骤七](#) 可参考 Demo 工程的 `ThirdBeautyViewController`，`BeautyView` 类相关示例代码。

步骤一：初始化授权

1. 首先在工程 `AppDelegate` 的 `didFinishLaunchingWithOptions` 中添加如下鉴权代码，其中 `LicenseURL`，`LicenseKey` 为腾讯云官网申请到授权信息，请参见 [License 指引](#)：

```
[TXLiveBase setLicenceURL:LicenseURL key:LicenseKey];
```

2. Xmagic 鉴权：在相关业务模块的初始化代码中设置 URL 和 KEY，触发 License 下载，避免在使用前才临时去下载。也可以在 `AppDelegate` 的 `didFinishLaunchingWithOptions` 方法里触发下载。其中

`LicenseURL` 和 `LicenseKey` 是控制台绑定 License 时生成的授权信息。SDK版本在2.5.1以前，`TELicenseCheck.h` 在 `XMagic.framework` 里面；SDK版本在2.5.1及以后，`TELicenseCheck.h` 在 `YTCommonXMagic.framework` 里面。

```
[TELicenseCheck setTELicense:LicenseURL key:LicenseKey
completion:^(NSInteger authresult, NSString * _Nonnull errorMsg) {
    if (authresult == TELicenseCheckOk) {
        NSLog(@"鉴权成功");
    } else {
        NSLog(@"鉴权失败");
    }
}];
```

鉴权 errorCode 说明:

错误码	说明
0	成功。Success
-1	输入参数无效，例如 URL 或 KEY 为空
-3	下载环节失败，请检查网络设置
-4	从本地读取的 TE 授权信息为空，可能是 IO 失败引起
-5	读取 VCUBE TEMP License文件内容为空，可能是 IO 失败引起
-6	v_cube.license 文件 JSON 字段不对。请联系腾讯云团队处理
-7	签名校验失败。请联系腾讯云团队处理
-8	解密失败。请联系腾讯云团队处理
-9	TELicense 字段里的 JSON 字段不对。请联系腾讯云团队处理
-10	从网络解析的 TE 授权信息为空。请联系腾讯云团队处理
-11	把TE授权信息写到本地文件时失败，可能是 IO 失败引起
-12	下载失败，解析本地 asset 也失败
-13	鉴权失败
其他	请联系腾讯云团队处理

步骤二：设置 SDK 素材资源路径

```
- (void)buildBeautySDK:(int)width and:(int)height texture:
(unsigned)textureID {
    NSDictionary *assetsDict = @{@"core_name":@"LightCore.bundle",
                                  @"root_path":[[NSBundle mainBundle]
bundlePath]};
    // 初始化SDK: width和height分别是texture的宽高
    self.xMagicKit = [[XMagic alloc]
initWithRenderSize:CGSizeMake(width,height) assetsDict:assetsDict];
}
```

步骤三：添加日志和事件监听

```
// Register log
[self.beautyKit registerSDKEventListener:self];
[self.beautyKit registerLoggerListener:self
withDefaultLevel:YT_SDK_ERROR_LEVEL];
```

步骤四：配置美颜各种效果

```
- (int)configPropertyWithType:(NSString *_Nonnull)propertyType withName:
(NSString *_Nonnull)propertyName withData:
(NSString *_Nonnull)propertyValue withExtraInfo:(id _Nullable)extraInfo;
```

步骤五：进行渲染处理

在视频帧回调接口，构造 YTProcessInput 传入到 SDK 内做渲染处理，可参考 Demo 中的 ThirdBeautyViewController。

```
[self.xMagicKit process:inputCPU withOrigin:YtLightImageOriginTopLeft
withOrientation:YtLightCameraRotation0]
```

步骤六：暂停/恢复 SDK

```
[self.beautyKit onPause];
[self.beautyKit onResume];
```

步骤七：布局中添加 SDK 美颜面板

```
UIEdgeInsets gSafeInset;
```

```
#if __IPHONE_11_0 && __IPHONE_OS_VERSION_MAX_ALLOWED >= __IPHONE_11_0
if (gSafeInset.bottom > 0) {
}
if (@available(iOS 11.0, *)) {
    gSafeInset = [UIApplication
sharedApplication].keyWindow.safeAreaInsets;
} else
#endif
{
    gSafeInset = UIEdgeInsetsZero;
}

dispatch_async(dispatch_get_main_queue(), ^{
    //美颜选项界面
    _vBeauty = [[BeautyView alloc] init];
    [self.view addSubview:_vBeauty];
    [_vBeauty mas_makeConstraints:^(MASConstraintMaker *make) {
        make.width.mas_equalTo(self.view);
        make.centerX.mas_equalTo(self.view);
        make.height.mas_equalTo(254);
        if (gSafeInset.bottom > 0.0) { // 适配全面屏
            make.bottom.mas_equalTo(self.view.mas_bottom).mas_offset(0);
        } else {
            make.bottom.mas_equalTo(self.view.mas_bottom).mas_offset(-10);
        }
    }];
    _vBeauty.hidden = YES;
});
```


Android（3.3.0及以前）

最近更新时间：2024-01-30 17:37:11

步骤一：解压 Demo 工程

1. 下载集成了腾讯特效 TE 的 [MLVB Demo](#) 工程。本 Demo 基于腾讯特效 SDK S1-04 套餐构建。
2. 替换资源。由于本 Demo 工程使用的 SDK 套餐未必与您实际的套餐一致，因此要将本 Demo 中的相关 SDK 文件替换为您实际使用的套餐的 SDK 文件。具体操作如下：
 - 删除 xmagic 模块中 libs 目录下的 .aar 文件，将 SDK 中 libs 目录下的 .aar 文件拷贝进 xmagic 模块中 libs 目录下。
 - 删除 xmagic 模块中 assets 目录下的所有文件，将 SDK 中的 assets/ 目录下的全部资源拷贝到 xmagic 模块 ../src/main/assets 目录下，如果 SDK 包中的 MotionRes 文件夹内有资源，将此文件夹也拷贝到 ../src/main/assets 目录下。
 - 删除 xmagic 模块中 jniLibs 目录下的所有 .so 文件，在 SDK 包内的 jniLibs 中找到对应的 .so 文件（由于 SDK 中 jniLibs 文件夹下的 arm64-v8a 和 armeabi-v7a 的 .so 文件在压缩包中，所以需要先解压），拷贝到 xmagic 模块中的 ../src/main/jniLibs 目录下。
3. 将 Demo 工程中的 xmagic 模块引到实际项工程中。

步骤二：打开 app 模块的 build.gradle

1. 将 applicationId 修改成与申请的测试授权一致的包名。
2. 添加 gson 依赖设置。

```
configurations {  
    all*.exclude group: 'com.google.code.gson'  
}
```

步骤三：SDK 接口集成

可参考 Demo 工程的 ThirdBeautyActivity 类。

1. 授权：

```
//鉴权注意事项及错误码详情，请参考  
https://cloud.tencent.com/document/product/616/65891#.E6.AD.A5.E9.AA.A  
4.E4.B8.80.EF.BC.9A.E9.89.B4.E6.9D.83  
XMagicImpl.checkAuth((errorCode, msg) -> {  
    if (errorCode == TELicenseCheck.ERROR_OK) {  
        showLoadResourceView();  
    } else {
```

```
TXCLog.e(TAG, "鉴权失败, 请检查鉴权url和key" + errorCode + "  
" + msg);  
    }  
});
```

2. 初始化素材:

```
private void showLoadResourceView() {  
    if (XmagicLoadAssetsView.isCopiedRes) {  
        XmagicResParser.parseRes(getApplicationContext());  
        initXMagic();  
    } else {  
        XmagicLoadAssetsView loadAssetsView = new  
XmagicLoadAssetsView(this);  
        loadAssetsView.setOnAssetsLoadFinishListener(() -> {  
            XmagicResParser.parseRes(getApplicationContext());  
            initXMagic();  
        });  
    }  
}
```

3. 启动推流设置:

```
String userId = String.valueOf(new Random().nextInt(10000));  
String pushUrl = AddressUtils.generatePushUrl(streamId, userId, 0);  
mLivePusher = new V2TXLivePusherImpl(this,  
V2TXLiveDef.V2TXLiveMode.TXLiveMode_RTC);  
mLivePusher.enableCustomVideoProcess(true,  
V2TXLivePixelFormatTexture2D, V2TXLiveBufferTypeTexture);  
mLivePusher.setObserver(new V2TXLivePusherObserver() {  
    @Override  
    public void onGLContextCreated() {  
    }  
  
    @Override  
    public int onProcessVideoFrame(V2TXLiveDef.V2TXLiveVideoFrame  
srcFrame, V2TXLiveDef.V2TXLiveVideoFrame dstFrame) {  
        if (mXMagic != null) {  
            dstFrame.texture.textureId =  
mXMagic.process(srcFrame.texture.textureId, srcFrame.width,  
srcFrame.height);  
        }  
    }  
});
```

```
        return srcFrame.texture.textureId;
    }

    @Override
    public void onGLContextDestroyed() {
        if (mXMagic != null) {
            mXMagic.onDestroy();
        }
    }
});
mLivePusher.setRenderView(mPushRenderView);
mLivePusher.startCamera(true);
int ret = mLivePusher.startPush(pushUrl);
mLivePusher.startMicrophone();
```

4. 将 textureId 传入到 SDK 内做渲染处理:

在 V2TXLivePusherObserver 接口的

```
onProcessVideoFrame(V2TXLiveDef.V2TXLiveVideoFrame srcFrame,
V2TXLiveDef.V2TXLiveVideoFrame dstFrame)
```

方法中添加如下代码。

```
if (mXMagic != null) {
    dstFrame.texture.textureId =
mXMagic.process(srcFrame.texture.textureId,
srcFrame.width,srcFrame.height);
}
return srcFrame.texture.textureId;
```

5. 暂停/销毁 SDK:

`onPause()` 用于暂停美颜效果，可以在 Activity/Fragment 生命周期方法中执行，`onDestroy` 方法需要在 GL 线程调用（可以在 `onTextureDestroyed` 方法中调用 `XMagicImpl` 对象的 `onDestroy()`），更多使用请参考 Demo。

```
mXMagic.onPause(); //暂停，与Activity的onPause方法绑定
mXMagic.onDestroy(); // //销毁，需要在GL线程中调用
```

6. 布局中添加 SDK 美颜面板:

```
<RelativeLayout
    android:id="@+id/livepusher_bp_beauty_annel"
    android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"  
android:layout_above="@+id/ll_edit_info" />
```

7. 初始化面板:

```
private void initXMagic() {  
    if (mXMagic == null) {  
        mXMagic = new XMagicImpl(this, mBeautyPanelView);  
    } else {  
        mXMagic.onResume();  
    }  
}
```

具体操作请参见 Demo 程的 `ThirdBeautyActivity.initXMagic()` 法。

一分钟集成 TRTC iOS（3.3.0及以前）

最近更新时间：2025-05-27 16:29:32

集成准备

1. 下载并解压 [Demo 包](#)，将 Demo 工程中的 xmagic 模块（bundle，XmagicIconRes，Xmagic 文件夹）导入到实际项目工程中。
2. 如果使用的 XMagic SDK 版本在2.5.0之前，导入 SDK 目录中的 `libpag.framework`、`Masonry.framework`、`XMagic.framework`、`YTCommonXMagic.framework`。如果使用的 XMagic SDK 版本在2.5.1及以后，导入 SDK 目录中的 `libpag.framework`、`Masonry.framework`、`XMagic.framework`、`YTCommonXMagic.framework`、`Audio2Exp.framework`、`TEFFmpeg.framework`（version3.0.0以后，改名为：`TECodec.framework`）。
3. framework 签名 General--> Masonry.framework 和 libpag.framework 选 Embed & Sign。YTCommonXMagic.framework 在版本2.5.1之前选 Do Not Embed，在版本2.5.1及以后选 Embed & Sign。
4. 将 Bundle ID 修改成与申请的测试授权一致。

开发者环境要求

- 开发工具 XCode 14 及以上：App Store 或单击 [下载地址](#)。
- 建议运行环境：
 - 设备要求：iPhone 5 及以上；iPhone 6 及以下前置摄像头最多支持到 720p，不支持 1080p。
 - 系统要求：iOS 10.0 及以上。

C/C++层开发环境

XCode 默认 C++ 环境。

类型	依赖库
系统依赖库	● Accelerate ● AssetsLibrary ● AVFoundation ● CoreMedia ● CoreFoundation ● CoreML ● Foundation ● JavaScriptCore

	• libc++.tbd • libz.b • libresolv.tbd • libsqlite3.0.tbd • MetalPerformanceShaders • MetalKit • MobileCoreServices • OpneAL • OpneGLES • Security • ReplayKit • SystemConfiguration • UIKit
自带的库	• YTCommon（鉴权静态库） • XMagic（美颜静态库） • libpag（视频解码动态库） • Masonry（控件布局库） • TXLiteAVSDK_Professional • TXFFmpeg(version3.0.0以后，改名为：TECodec.framework) • TXSoundTouch • Audio2Exp（xmagic sdk version在2.5.1及以后的版本才有） • TEFFmpeg（xmagic sdk version在2.5.1及以后的版本才有）

SDK 接口集成

objective-c

- [步骤一](#) 和 [步骤二](#) 可参考 Demo 工程中，ThirdBeautyViewController 类 viewDidLoad，buildBeautySDK 方法；AppDelegate类的application方法进行了Xmagic鉴权。
- [步骤四](#) 至 [步骤七](#) 可参考 Demo 工程的 `ThirdBeautyViewController`，`BeautyView` 类相关示例代码。

swift

- [步骤一](#) 可参考 Demo 工程中，ThirdBeautyEnterViewController 类viewDidLoad中的TELicenseCheck.setTELicense。

- **步骤二** 至 **步骤七** 可参考 Demo 工程的 ThirdBeautyViewController，BeautyView 类相关示例代码。

❗ 说明

在 swift 项目中集成 XMagic SDK 时，如果 XMagic SDK 的版本在version3.0.0.3及以上版本时，支持模块引用。

```
import XMagic
```

如果 XMagic SDK 的版本在version3.0.0.3以前，需要以桥接文件的方式引入头文件，具体步骤是：

1. 创建桥接头文件。例如 `***-Bridging-Header.h`，并添加如下代码`#import "XMagic.h"`。
2. 配置工程 BuildSetting 的 Objective-c Bridging header 选项。设置桥接文件的路径并添加到 Objective-c Bridging header 中（如：`$(SRCROOT)/SwiftCallOC/***-Bridging-Header.h`，根据项目具体路径确定），编译运行即可。

步骤一：初始化授权

XMagic 鉴权：在相关业务模块的初始化代码中设置 URL 和 KEY，触发 License 下载，避免在使用前才临时去下载。也可以在 AppDelegate 的 `didFinishLaunchingWithOptions` 方法里触发下载。其中，`LicenseURL` 和 `LicenseKey` 是控制台绑定 License 时生成的授权信息。SDK 版本在2.5.1以前，`TELICENSECheck.h` 在 `XMagic.framework` 里面；SDK 版本在2.5.1及以后，`TELICENSECheck.h` 在 `YTCommonXMagic.framework` 里面。

objective-c

```
[TELICENSECheck setTELICENSE:LicenseURL key:LicenseKey
completion:^(NSInteger authresult, NSString * _Nonnull errorMsg) {
    if (authresult == TELICENSECheckOk) {
        NSLog(@"success");
    } else {
        NSLog(@"failed");
    }
}];
```

swift

```
TELicenseCheck.setTELicense(LicenseURL, key: LicenseKey) {  
    authresult, errorMsg in  
        if authresult == 0{  
            print("success")  
        }else{  
            print("failed")  
        }  
    }  
}
```

鉴权 errorCode 说明:

错误码	说明
0	成功。Success
-1	输入参数无效, 例如 URL 或 KEY 为空
-3	下载环节失败, 请检查网络设置
-4	从本地读取的 TE 授权信息为空, 可能是 IO 失败引起
-5	读取 VCUBE TEMP License文件内容为空, 可能是 IO 失败引起
-6	v_cube.license 文件 JSON 字段不对。请联系腾讯云团队处理
-7	签名校验失败。请联系腾讯云团队处理
-8	解密失败。请联系腾讯云团队处理
-9	TELicense 字段里的 JSON 字段不对。请联系腾讯云团队处理
-10	从网络解析的 TE 授权信息为空。请联系腾讯云团队处理
-11	把TE授权信息写到本地文件时失败, 可能是 IO 失败引起
-12	下载失败, 解析本地 asset 也失败
-13	鉴权失败
其他	请联系腾讯云团队处理

步骤二：设置 SDK 素材资源路径

objective-c

```
- (void)buildBeautySDK:(int)width and:(int)height texture:
(unsigned)textureID {
    NSDictionary *assetsDict = @{@"core_name":@"LightCore.bundle",
                                   @"root_path":[[NSBundle mainBundle]
bundlePath]};
    // 初始化SDK: width和height分别是texture的宽高
    self.xMagicKit = [[XMagic alloc]
initWithRenderSize:CGSizeMake(width,height) assetsDict:assetsDict];
}
```

swift

```
///初始化SDK
func buildBeautySDK(width:UInt32,height:UInt32,texture:GLuint) {
    let assetsDict:[String:String] =
["core_name":"LightCore.bundle", "root_path":Bundle.main.bundlePath]
    let size = CGSize(width: CGFloat(width), height:
CGFloat(height))
    xMagic = XMagic.init(renderSize: size, assetsDict:
assetsDict)
}
```

步骤三：添加日志和事件监听

objective-c

```
// Register log
[self.xMagicKit registerSDKEventListener:self];
```

```
[self.xMagicKit registerLoggerListener:self
withDefaultLevel:YT_SDK_ERROR_LEVEL];
```

swift

```
xMagic?.register(self)
xMagic?.registerLoggerListener(self, withDefaultLevel:
YtSDKLogLevel.YT_SDK_DEBUG_LEVEL)
```

步骤四：配置美颜各种效果（详细美颜效果配置请参考 [美颜参数说明](#) 和 Demo）

```
// @brief 配置美颜各种效果
// @param propertyType 效果类型 字符串: beauty, lut, motion
// @param propertyName 效果名称
// @param propertyValue 效果数值
// @param extraInfo 预留扩展, 附加额外配置dict
// @return 成功返回0, 失败返回其他
- (int)configPropertyWithType:(NSString *_Nonnull)propertyType withName:
(NSString *_Nonnull)propertyName withData:
(NSString *_Nonnull)propertyValue withExtraInfo:(id _Nullable)extraInfo;
```

步骤五：进行渲染处理

TRTC SDK 设置第三方美颜的视频数据回调：设置该回调之后，TRTC SDK 会把采集到的视频帧通过您设置的 delegate 回调出来，用于第三方美颜组件进行二次处理。

objective-c

```
[self.trtcCloud setLocalVideoProcessDelegete:self
pixelFormat:TRTCVideoPixelFormat_Texture_2D
bufferType:TRTCVideoBufferType_Texture];
```

swift

```
trtcCloud.setLocalVideoProcessDelegete(self, pixelFormat:
TRTCVideoPixelFormat._Texture_2D, bufferType:.texture)
```

在视频帧回调接口 `onProcessVideoFrame:(TRTCVideoFrame *_Nonnull)srcFrame dstFrame:(TRTCVideoFrame *_Nonnull)dstFrame` 中，构造 `YTProcessInput` 传入到 SDK 内做渲染处理，可参考 Demo 中的 `ThirdBeautyViewController`。

objective-c

```
#pragma mark - TRTCVideoFrameDelegate
- (uint32_t)onProcessVideoFrame:(TRTCVideoFrame *_Nonnull)srcFrame
dstFrame:(TRTCVideoFrame *_Nonnull)dstFrame {
    if (srcFrame.width != _renderSize.width || srcFrame.height !=
_renderSize.height) {
        _renderSize = CGSizeMake(srcFrame.width, srcFrame.height);
        if (!_xMagicKit) {
            [self buildBeautySDK:srcFrame.width
and:srcFrame.height];
        } else {
            [_xMagicKit setRenderSize:_renderSize];
        }
    }
    YTProcessInput *input = [[YTProcessInput alloc] init];
    input.textureData = [[YTTextureData alloc] init];
    input.textureData.texture = srcFrame.textureId;
    input.textureData.textureWidth = srcFrame.width;
    input.textureData.textureHeight = srcFrame.height;
    input.dataType = kYTTextureData;
    YTProcessOutput *output = [self.xMagicKit process:input
withOrigin:YtLightImageOriginTopLeft
withOrientation:YtLightCameraRotation0];
    dstFrame.textureId = output.textureData.texture;
    return 0;
}
```

```
}
```

swift

```
#pragma mark - TRTCVideoFrameDelegate

func onProcessVideoFrame(_ srcFrame: TRTCVideoFrame, dstFrame:
TRTCVideoFrame) -> UInt32 {
    if xMagic == nil {
        buildBeautySDK(width: srcFrame.width, height:
srcFrame.height, texture: srcFrame.textureId)
    }
    if xMagic != nil && (heightF != srcFrame.height || widthF !=
srcFrame.width) {
        widthF = srcFrame.width
        heightF = srcFrame.height
        let rendersize = CGSize(width: CGFloat(srcFrame.width),
height: CGFloat(srcFrame.height))
        xMagic?.setRenderSize(rendersize)
    }
    let input = YTProcessInput.init()
    input.textureData = YTTextureData.init()
    input.textureData?.texture = Int32(srcFrame.textureId)
    input.textureData?.textureWidth = Int32(srcFrame.width)
    input.textureData?.textureHeight = Int32(srcFrame.height)
    input.dataType = kYTTextureData
    let output = xMagic?.process(input, with:
YtLightImageOrigin(rawValue: 0)!, with:
YtLightDeviceCameraOrientation(rawValue: 0)!)
    dstFrame.textureId = GLuint((output?.textureData?.texture)!)
    return 0
}
```

步骤六：暂停/恢复/销毁 SDK

objective-c

```
//暂停SDK
[self.xMagicKit onPause];
//恢复SDK
[self.xMagicKit onResume];
//销毁SDK
[self.xMagicKit clearListeners];
[self.xMagicKit deinit];
self.xMagicKit = nil;
```

swift

```
//暂停SDK
xMagic?.onPause()
//恢复SDK
xMagic?.onResume()
//销毁SDK
xMagic?.clearListeners()
xMagic?.deinit()
xMagic = nil
```

步骤七：布局中添加 SDK 美颜面板

objective-c

```
UIEdgeInsets gSafeInset;
#if __IPHONE_11_0 && __IPHONE_OS_VERSION_MAX_ALLOWED >=
__IPHONE_11_0
if(gSafeInset.bottom > 0){
}
if (@available(iOS 11.0, *)) {
    gSafeInset = [UIApplication
sharedApplication].keyWindow.safeAreaInsets;
} else
#endif
```

```
{
    gSafeInset = UIEdgeInsetsZero;
}

dispatch_async(dispatch_get_main_queue(), ^{
    //美颜选项界面
    self.beautyContainer = [[BeautyView alloc] init];
    [self.view addSubview:self.beautyContainer];
    [self.beautyContainer mas_makeConstraints:^(MASConstraintMaker
*make) {
        make.width.mas_equalTo(self.view);
        make.centerX.mas_equalTo(self.view);
        make.height.mas_equalTo(254);
        if(gSafeInset.bottom > 0.0){ // 适配全面屏

make.bottom.mas_equalTo(self.view.mas_bottom).mas_offset(0);
            } else {

make.bottom.mas_equalTo(self.view.mas_bottom).mas_offset(-10);
            }
        }
    }];
});
```

swift

```
//初始化xmagic以后，把xmagic对象传给beautyView
beautyView.beautyKitRef = xMagic
view.addSubview(beautyView)
beautyView.snp.makeConstraints { make in
    make.height.equalTo(200)
    make.bottom.equalTo(switchCamButton.snp.top).offset(10)
    make.left.right.equalTo(view)
}
```

Android（3.3.0及以前）

最近更新时间：2024-01-30 17:37:11

步骤一：解压 Demo 工程

1. 下载集成了腾讯特效 TE 的 **TRTC Demo** 工程。本 Demo 基于腾讯特效 SDK S1-04 套餐构建。
2. 替换资源。由于本 Demo 工程使用的 SDK 套餐未必与您实际的套餐一致，因此要将本 Demo 中的相关 SDK 文件替换为您实际使用的套餐的 SDK 文件。具体操作如下：
 - 删除 xmagic 模块中 libs 目录下的 `.aar` 文件，将 SDK 中 libs 目录下的 `.aar` 文件拷贝进 xmagic 模块中 libs 目录下。
 - 删除 xmagic 模块中 assets 目录下的所有文件，将 SDK 中的 `assets/` 目录下的全部资源拷贝到 xmagic 模块 `../src/main/assets` 目录下，如果 SDK 包中的 MotionRes 文件夹内有资源，将此文件夹也拷贝到 `../src/main/assets` 目录下。
 - 删除 xmagic 模块中 jniLibs 目录下的所有 `.so` 文件，在 SDK 包内的 jniLibs 中找到对应的 `.so` 文件（由于 SDK 中 jniLibs 文件夹下的 arm64-v8a 和 armeabi-v7a 的 `.so` 文件在压缩包中，所以需要先解压），拷贝到 xmagic 模块中的 `../src/main/jniLibs` 目录下。
3. 将 Demo 工程中的 xmagic 模块引到实际项工程中。

步骤二：打开 app 模块的 build.gradle

1. 将 applicationId 修改成与申请的测试授权一致的包名。
2. 添加 gson 依赖设置。

```
configurations{
    all*.exclude group:'com.google.code.gson'
}
```

步骤三：SDK 接口集成

可参考 Demo 工程的 ThirdBeautyActivity 类。

1. 授权：

```
//鉴权注意事项及错误码详情，请参考
https://cloud.tencent.com/document/product/616/65891#.E6.AD.A5.E9.AA.A
4.E4.B8.80.EF.BC.9A.E9.89.B4.E6.9D.83
XMagicImpl.checkAuth((errorCode, msg) -> {
    if (errorCode == TELicenseCheck.ERROR_OK) {
        showLoadResourceView();
    } else {
```

```
TXCLog.e(TAG, "鉴权失败, 请检查鉴权url和key" + errorCode  
+ " " + msg);  
    }  
});
```

2. 初始化素材:

```
private void showLoadResourceView() {  
    if (XmagicLoadAssetsView.isCopyedRes) {  
        XmagicResParser.parseRes(getApplicationContext());  
        initXMagic();  
    } else {  
        loadAssetsView = new XmagicLoadAssetsView(this);  
        loadAssetsView.setOnAssetsLoadFinishListener(() -> {  
            XmagicResParser.parseRes(getApplicationContext());  
            initXMagic();  
        });  
    }  
}
```

3. 开启推流设置:

```
mTRTCCloud.setLocalVideoProcessListener(TRTCCLoudDef.TRTC_VIDEO_PIXEL_  
FORMAT_Texture_2D, TRTCCLoudDef.TRTC_VIDEO_BUFFER_TYPE_TEXTURE, new  
TRTCCLoudListener.TRTCVideoFrameListener() {  
    @Override  
    public void onGLContextCreated() {  
    }  
    @Override  
    public int onProcessVideoFrame(TRTCCLoudDef.TRTCVideoFrame srcFrame,  
TRTCCLoudDef.TRTCVideoFrame dstFrame) {  
    }  
    @Override  
    public void onGLContextDestory() {  
    }  
});
```

4. 将 textureId 传入到 SDK 内做渲染处理: 在 TRTCVideoFrameListener 接口的


```
onProcessVideoFrame(TRTCCloudDef. TRTCVideoFrame srcFrame,  
TRTCCloudDef. TRTCVideoFrame dstFrame)
```

方法内添加如下代码：

```
dstFrame.texture.textureId = mXMagic.process(srcFrame.texture.textureI  
d, srcFrame.width, srcFrame.height);
```

process 方法以及美颜 SDK 的各 API 的详细说明见 [API 文档](#)。

5. 暂停/关闭 SDK:

onPause() 用于暂停美颜效果，可以在 Activity/Fragment 生命周期方法中执行，onDestroy 方法需要在 GL 线程调用（可以在 onTextureDestroyed 方法中调用 XMagicImpl 对象的 `onDestroy()` ），更多使用请参考 Demo。

```
mXMagic.onPause(); //暂停，与Activity的onPause方法绑定  
mXMagic.onDestroy(); //销毁，需要在GL线程中调用
```

6. 布局中添加 SDK 美颜面板:

```
<RelativeLayout  
    android:layout_above="@+id/ll_edit_info"  
    android:id="@+id/livepusher_bp_beauty_annel"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

7. 初始化面板:

```
private void initXMagic() {  
    if (mXMagic == null) {  
        mXMagic = new XMagicImpl(this, mBeautyPanelView);  
    }else {  
        mXMagic.onResume();  
    }  
}
```

具体操作请参见 Demo 程的 `ThirdBeautyActivity.initXMagic()` 法。

一分钟集成短视频 iOS

最近更新时间：2025-06-05 10:29:42

集成准备

1. 下载并解压 [Demo 包](#)，将 Demo 工程中的 xmagic 模块（bundle，XmagicIconRes 两个文件夹下面的文件，Record>View 文件夹下面的文件）导入到实际项目工程中。
2. 导入 lib 目录中的 libpag.framework、Masonry.framework、XMagic.framework 和 YTCommonXMagic.framework。
3. framework 签名 General>Masonry.framework 和 libpag.framework 选 Embed & Sign。
4. 将 Bundle ID 修改成与申请的测试授权一致。

SDK 接口集成

- [步骤一](#) 和 [步骤二](#) 可参考 Demo 工程中，UGCKitRecordViewController 类 viewDidLoad，buildBeautySDK 方法。
- [步骤四](#) 至 [步骤七](#) 可参考 Demo 工程的 UGCKitRecordViewController，BeautyView 类相关实例代码。

步骤一：初始化授权

在工程 AppDelegate 的 didFinishLaunchingWithOptions 中添加如下代码，其中 LicenseURL，LicenseKey 为腾讯云官网申请到授权信息，请参见 [License 指引](#)：

```
[TXUGCBase setLicenceURL:LicenseURL key:LicenseKey];
```

授权代码可参考 Demo 中 UGCKitRecordViewController 类 viewDidLoad 中的授权代码：

```
NSString *licenseInfo = [TXUGCBase getLicenceInfo];
NSData *jsonData = [licenseInfo dataUsingEncoding:NSUTF8StringEncoding];
NSError *err = nil;
NSDictionary *dic = [NSJSONSerialization JSONObjectWithData:jsonData
options:NSJSONReadingMutableContainers error:&err];
NSString *xmagicLicBase64Str = [dic objectForKey:@"TELicense"];

//初始化 xmagic 授权
int authRet = [XMagicAuthManager initAuthByString:xmagicLicBase64Str
withSecretKey:@""]; // withSecretKey 为空字符串，不需要填写内容
```

```
NSLog(@"xmagic auth ret : %i", authRet);  
NSLog(@"xmagic auth version : %@", [XMagicAuthManager getVersion]);
```

步骤二：设置 SDK 素材资源路径

```
CGSize previewSize = [self  
getPreviewSizeByResolution:self.currentPreviewResolution];  
NSString *beautyConfigPath =  
[NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,  
NSUserDomainMask, YES) lastObject];  
beautyConfigPath = [beautyConfigPath  
stringByAppendingPathComponent:@"beauty_config.json"];  
NSFileManager *localFileManager=[[NSFileManager alloc] init];  
BOOL isDir = YES;  
NSDictionary * beautyConfigJson = @{};  
if ([localFileManager fileExistsAtPath:beautyConfigPath  
isDirectory:&isDir] && !isDir) {  
    NSString *beautyConfigJsonStr = [NSString  
stringWithContentsOfFile:beautyConfigPath encoding:NSUTF8StringEncoding  
error:nil];  
    NSError *jsonError;  
    NSData *objectData = [beautyConfigJsonStr  
dataUsingEncoding:NSUTF8StringEncoding];  
    beautyConfigJson = [NSJSONSerialization  
JSONObjectWithData:objectData  
options:NSJSONReadingMutableContainers  
error:&jsonError];  
}  
NSDictionary *assetsDict = @{@"core_name":@"LightCore.bundle",  
                             @"root_path":[[NSBundle mainBundle]  
bundlePath],  
                             @"tnn_"  
                             @"beauty_config":beautyConfigJson  
};  
// Init beauty kit  
self.beautyKit = [[XMagic alloc] initWithRenderSize:previewSize  
assetsDict:assetsDict];
```

步骤三：添加日志和事件监听

```
// Register log
```

```
[self.beautyKit registerSDKEventListener:self];
[self.beautyKit registerLoggerListener:self
withDefaultLevel:YT_SDK_ERROR_LEVEL];
```

步骤四：配置美颜各种效果

```
- (int)configPropertyWithType:(NSString *_Nonnull)propertyType withName:
(NSString *_Nonnull)propertyName withData:
(NSString*_Nonnull)propertyValue withExtraInfo:(id _Nullable)extraInfo;
```

步骤五：进行渲染处理

在短视频预处理帧回调接口，构造 YTProcessInput 将 textureId 传入到 SDK 内做渲染处理。

```
[self.xMagicKit process:inputCPU withOrigin:YtLightImageOriginTopLeft
withOrientation:YtLightCameraRotation0]
```

步骤六：暂停/恢复 SDK

```
[self.beautyKit onPause];
[self.beautyKit onResume];
```

步骤七：布局中添加 SDK 美颜面板

```
UIEdgeInsets gSafeInset;
#if __IPHONE_11_0 && __IPHONE_OS_VERSION_MAX_ALLOWED >= __IPHONE_11_0
if(gSafeInset.bottom > 0){
}
if (@available(iOS 11.0, *)) {
    gSafeInset = [UIApplication
sharedApplication].keyWindow.safeAreaInsets;
} else
#endif
{
    gSafeInset = UIEdgeInsetsZero;
}

dispatch_async(dispatch_get_main_queue(), ^{
    //美颜选项界面
    _vBeauty = [[BeautyView alloc] init];
```

```
[self.view addSubview:_vBeauty];
[_vBeauty mas_makeConstraints:^(MASConstraintMaker *make) {
    make.width.mas_equalTo(self.view);
    make.centerX.mas_equalTo(self.view);
    make.height.mas_equalTo(254);
    if(gSafeInset.bottom > 0.0){ // 适配全面屏
        make.bottom.mas_equalTo(self.view.mas_bottom).mas_offset(0);
    } else {

make.bottom.mas_equalTo(self.view.mas_bottom).mas_offset(-10);
    }
}];
_vBeauty.hidden = YES;
});
```

Android

最近更新时间：2023-05-25 15:44:00

步骤一：解压 Demo 工程

1. 下载集成了腾讯特效 TE 的 [UGSV Demo](#) 工程。
2. 将 Demo 工程中的 xmagic 模块引到实际项目中。

步骤二：授权

⚠ 注意

部分代码与 Demo 工程中的代码有差异，请以本文档描述为准。

1. 申请 License URL 和 License KEY，请参见 [License 指引](#)。

⚠ 注意

不需要把 License 文件下载下来放到本地工程里。

2. 在 TCAApplication 的 onCreate 方法中调用下面这个方法，触发 License 下载，避免在使用前才临时去下载。

```
import com.tencent.xmagic.license.LicenceCheck;
LicenceCheck.getInstance().setXMagicLicense(context, URL, KEY);
```

3. 然后在真正要使用美颜功能时（例如 XMagicImpl.java 中），再去做鉴权：

```
private void auth() {
    LicenceCheck.getInstance().setListener(new
LicenceCheck.LicenceCheckListener() {
        @Override
        public void onLicenceLoaded(int result, String reason) {
            //在2.4.0版本，如果无需下载，或者下载失败，不会回调这个方法。（后续版本会补齐）
            //如果有下载，且下载成功，会回调。result为LicenceCheck.ERROR_OK表示下载下来的license文件是有效的
            if (result == LicenceCheck.ERROR_OK) {
                checkAuth(context);
            }
        }
    });
}
```

```
//再次触发下载（因为有可能之前在onCreate那里触发下载没有成功）
LicenceCheck.getInstance().setXMagicLicense(context, URL, KEY);

checkAuth(context);
}

private boolean authorized = false;
private synchronized void checkAuth(Context context) {
    Log.d(TAG, "checkAuth: authorized=" + authorized);
    if (authorized) {
        return;
    }
    LicenceCheck mLicenceCheck = LicenceCheck.getInstance();
    String licenseInfo = mLicenceCheck.getBase64Licence();
    if (TextUtils.isEmpty(licenseInfo)) {
        licenseInfo = mLicenceCheck.getLicensePathBase64();
    }
    if (TextUtils.isEmpty(licenseInfo)) {
        Log.d(TAG, "licenseInfo is empty");
        authorized = false;
    } else {
        Auth.AuthResult result = Auth.authByBase64(context,
            licenseInfo, "");

        String msg = Json.toJsonStr(result);
        Log.d(TAG, "isSucceed=" + result.isSucceed);
        Log.d(TAG, "msg=" + msg);
        authorized = result.isSucceed;
    }

    if (authorized) {
        //TODO 鉴权成功，在这里通知UI刷新、执行下一步操作之类的事情
    }
}
```

❗ 说明

如果鉴权失败，您可参考 [查看鉴权结果及错误码说明](#) 定位失败原因。

步骤三：打开 app 模块的 build.gradle

1. 将 applicationId 修改成与申请的测试授权 致的包名。
2. 添加 gson 依赖设置。

```
configurations{
    all*.exclude group:'com.google.code.gson'
}
```

步骤四：SDK 接口集成

可参考 Demo 程的 TCVideoRecoredActivity 类。

1. 授权：

```
XMagicImpl.initAuth(getApplicationContext());
```

2. 初始化素材：

```
XmagicLoadAssetsView loadAssetsView = new XmagicLoadAssetsView(this);
loadAssetsView.setOnAssetsLoadFinishListener(new
XmagicLoadAssetsView.OnAssetsLoadFinishListener() {
    @Override
    public void onAssetsLoadFinish() {
        XmagicResParser.parseRes();
        XmagicUIState.initDatas(XmagicResParser.getProperties());
        initXMagic();
    }
});
```

3. 开启推流设置：

```
TXUGCRecord instance = TXUGCRecord.getInstance(this);
instance.setVideoProcessListener(new
TXUGCRecord.VideoCustomProcessListener() {
    @Override
    public int onTextureCustomProcess(int textureId, int width, int
height) {
        if (isPause == 0 && mXMagic !=null) {
            return mXMagic.process(textureId, width, height);
        }
        return 0;
    }
    @Override
    public void onDetectFacePoints(float[] floats) {
    }
    @Override
```



```
public void onTextureDestroyed() {  
    ...  
}  
});
```

4. 将 textureId 传入到 SDK 内做渲染处理:

在 VideoCustomProcessListener 接口的

onTextureCustomProcess(int textureId, int width, int height) 方法内添加如下代码。

```
if (isPause == 0 && mXMagic != null) {  
    return mXMagic.process(textureId, width, height);  
}  
return 0;
```

5. 暂停/销毁 SDK:

⚠ 注意

当调用 onPause 方法后, 需要调用 onDestroy 方法销毁, 如果需要再次使用, 则需要重新创建 mXMagic 对象。

```
mXMagic.onPause(); // 暂停, 与 Activity 的 onPause 方法绑定  
mXMagic.onDestroy(); // 销毁, 与 Activity 的 onDestroy 方法绑定
```

6. 布局中添加 SDK 美颜面板:

```
<include  
    layout="@layout/xmagic_panel"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:visibility="gone"/>
```

7. 初始化面板与美颜设置回调接口:

具体操作请参见 Demo 程的 TCVideoRecoredActivity.initXMagic(); 法。

鉴权结果及错误码

Android

最近更新时间：2023-08-28 17:53:43

Auth.AuthResult

授权返回结果实体类。

属性	含义
<code>public final boolean isSucceed;</code>	是否授权成功
<code>public final AuthResultSucceed authResultSucceed;</code>	授权成功详情
<code>public final AuthResultFail authResultFail;</code>	授权失败详情
<code>public final AuthInfo authInfo;</code>	授权详情信息，一般供排查问题时使用

Auth.AuthResultSucceed

授权成功详情实体类。

属性	含义
<code>public final Map<Integer, String> features = new HashMap<Integer, String>();</code>	已授权的能力： <能力ID, 能力名称>
<code>public final long endTime;</code>	授权结束时间, UNIX 时间戳, 单位是秒
<code>public final String endTimeStr;</code>	授权结束时间格式化后的字符串, 方便阅读

Auth.AuthResultFail

授权失败详情实体类。

属性	含义
<code>public final int code;</code>	错误码

```
public final String msg;
```

错误提示信息

错误码

如果 ret 授权返回0，则已授权成功。否则请按照返回值来排查问题：

错误码	原因	处理建议
3004/3005	无效授权	检查授权文件的文件名/路径是否正确，比如 license 和 licence 的拼写方式是否统一
3015	Bundle Id / Package Name 不匹配	检查您的 App 使用的 Bundle Id / Package Name 和申请的是否一致，检查是否使用了正确的授权文件
3018	授权文件已过期	需要向腾讯云申请续期
其他	内部错误	请将错误码上报给腾讯云支持团队处理