# 文字识别 客户端 SDK 接入文档





### 【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】



# 🥎 腾讯云

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

客户端 SDK 接入文档

客户端 SDK 接入概览

SDK 更新日志

IOS 文字识别客户端 SDK 更新日志

Android 文字识别客户端 SDK 更新日志

Harmony NEXT 文字识别客户端 SDK 更新日志

- 一分钟快速体验
- 一分钟跑通 Demo

跑通 Demo (iOS)

跑通 Demo (Android)

跑通 Demo (Harmony)

一分钟集成 SDK

快速集成(iOS)

快速集成(Android)

快速集成 (Harmony)

客户端 API 文档

iOS API 概览

Android API 概览

Harmony API 概览



# 客户端 SDK 接入文档客户端 SDK 接入概览

最近更新时间: 2025-06-04 17:24:02

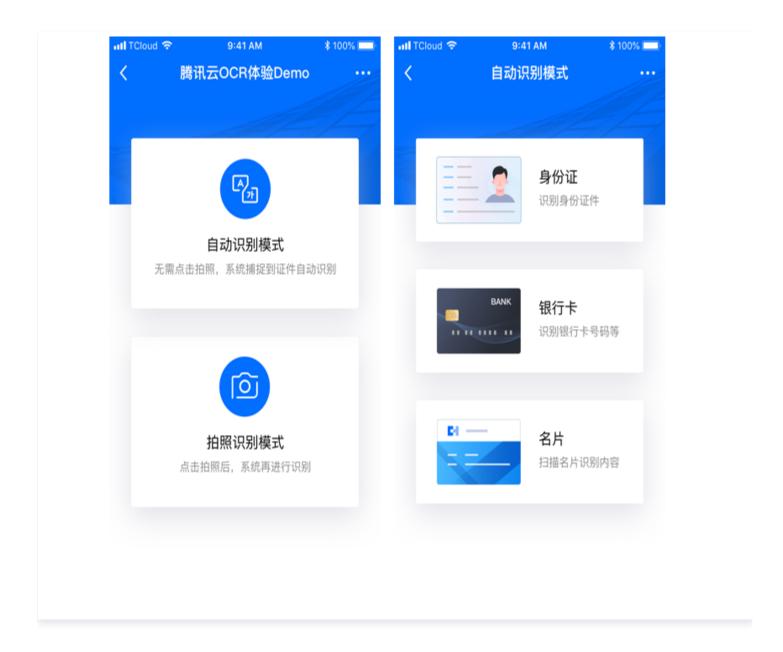
# SDK 接入概览

### SDK 说明

文字识别 OCR 客户端 SDK 给您提供了一种方便、快捷、灵活的方式,将腾讯云文字识别能力集成到您的客户端,以使用腾讯云快速、准确的文字识别服务。

目前腾讯云文字识别客户端 SDK 支持的能力:

- 身份证识别
- 银行卡识别
- 名片识别
- 车辆 VIN 码识别
- 车牌识别
- 行驶证识别
- 驾驶证识别
- 中国香港身份证识别
- 港澳台通行证识别
- 护照识别(港澳台地区及境外护照)



后续将逐步加入更多文字识别能力,敬请期待!

# 快速体验

目前腾讯云提供了 Android 端的文字识别体验 App,一分钟快速体验 即可体验文字识别客户端 SDK 能力。

# SDK 接入

文字识别客户端 SDK 目前支持 Android、iOS、Harmony 三个平台。前往 文字识别控制台 即可下载 SDK。







# SDK 更新日志 IOS 文字识别客户端 SDK 更新日志

最近更新时间: 2024-08-23 19:45:21

### SDK 固定下载地址

#### SDK 下载

### 更新内容

v1.1.0.17 更新内容(2024年7月17日): 修复大陆身份证双面录入时界面显示异常问题。

#### v1.1.0.16 更新内容(2024年7月1日):

- 修复卡证参数问题。
- 识别范围优化。
- 新增转场动画控制字段 isForbidAnima。

v1.1.0.15 更新内容(2024年6月25日): 新增澳门身份证卡证类型(IDCardOCR\_MACAU)。

v1.1.0.14 更新内容(2024年5月12日):

新增配置字段 minCropSize,支持设置最小裁剪范围。

### v1.1.0.13 更新内容(2024年3月12日):

- 修复切换增强身份证识别不生效问题。
- 优化识别完成后内存释放问题。
- 增加图片压缩配置字段 maxCompressLength。
- 身份证正反面提示文字修复。
- 调整 IDCardOCR\_FRONT 类型为国徽面,IDCardOCR\_BACK 类型为人像面。

#### v1.1.0.12 更新内容(2024年3月10日):

- 1. 修复裁剪页面按钮未支持多语言问题。
- 2. 修复切换模式 toast 未支持多语言问题。
- 3. 修改弹出切换裁剪页面为全屏。
- 4. 修改扫描也没 tips Label 约束,增加宽度。
- 5. validCount 稳定帧数默认值改为10帧。



v1.1.0.11 更新内容(2023年10月27日): 稳定帧、模糊阈值逻辑优化。

v1.1.0.10 更新内容(2023年7月10日): 增加权限弹窗配置。

v1.1.0.9 更新内容(2023年6月21日): 修复已知问题。

v1.1.0.8 更新内容(2023年6月12日):增加支持仅相册模式开关。 修复已知问题。

v1.1.0.7 更新内容(2023年6月8日): 增加关闭裁剪功能配置开关。

v1.1.0.6 更新内容(2023年4月13日): 修复已知问题。

v1.1.0.5 更新内容(2023年3月24日): 增加配置项: 默认相机缩放倍数。

v1.1.0.4 更新内容(2023年1月13日):

- 增加控制弹窗样式字段。
- 优化隐藏按钮逻辑。
- 修复已知问。

### v1.1.0.3 更新内容(2023年1月29日):

- 修复xcode14报警告问题。
- 修复已知问题。



# Android 文字识别客户端 SDK 更新日志

最近更新时间: 2024-08-23 19:45:21

### SDK 固定下载地址

SDK 下载

### 更新内容

v2.0.0.19(2024年7月11日) 海外版本优化国内身份证识别

v2.0.0.18(2024年6月25日) 修复已知问题

v2.0.0.17(2024年5月27日) 调整 SDK 竖屏下的预览大小

v2.0.0.16(2024年2月29日) 支持图片采集最小大小调整

v2.0.0.15(2024年2月26日) 解决已知问题

v2.0.0.12(2024年1月2日)

修复在 targetSdkVersion 34下部分手机无法选择相册的问题

v2.0.0.11(2023年6月19日)

新增 showCustomPermissionDialog 配置项

v2.0.0.10(2023年6月19日)

新增 supportOnlyAlbumMode 配置项



# Harmony NEXT 文字识别客户端 SDK 更新日志

最近更新时间: 2024-10-23 14:50:11

### SDK 固定下载地址

SDK 下载

## 更新内容

v3.0.2.31 (2024年9月11日): 修正对接文档中有关认证描述及部分名称错误

v3.0.0.14(2024年7月10日): 初次发布



# 一分钟快速体验

最近更新时间: 2024-08-23 16:57:38

目前腾讯云提供了 Android 端的文字识别体验 App,扫码即可体验文字识别客户端 SDK 能力。



Android 端文字识别 App 提供以下能力的体验:

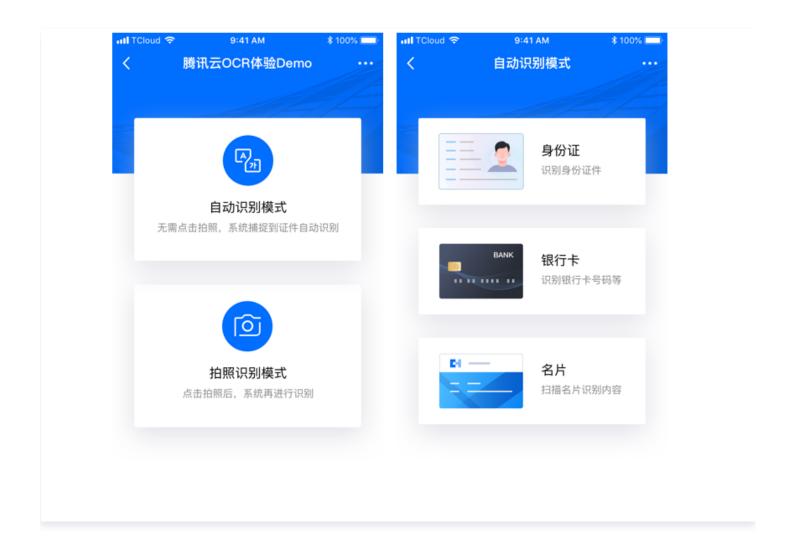
- 身份证识别
- 银行卡识别
- 名片识别
- 车辆 VIN 码识别
- 车牌识别
- 行驶证识别
- 驾驶证识别

### 并且提供两种识别模式:

- 自动识别
- 拍照识别

Android 端文字识别体验 App 页面













# 一分钟跑通 Demo 跑通 Demo(iOS)

最近更新时间: 2025-03-19 09:44:22

本文主要介绍如何快速运行腾讯云 文字识别 OCR Demo。

### 环境要求

- 兼容 iOS11.0 及以上版本的 iOS 系统
- 开发工具使用 Xcode11.0 或以上版本,推荐使用 Xcode 11.5
- 建议使用 iPhone 真机体验

### 前提条件

- 您已 注册腾讯云 账号, 并完成 实名认证 。
- 进入 文字识别控制台,即可开通相应服务,并在 账号中心 获取 API 密钥。
- 前往文字识别客户端 SDK 接入页面 下载对应 SDK,并将 SDK 的 demo 文件下载到本地。



# 操作步骤

# 步骤1: 导入工程

进入下载的 Demo 文件夹,双击使用 Xcode 打开项目工程。



Podfile	昨天 下午 2:39
Podfile.lock	昨天 下午 2:39
▶ <b>i</b> Pods	昨天 下午3:27
► TencentOcrSDKDemo	前天 下午 4:56
TencentOcrSDKDemo.xcodeproj	昨天 下午3:28
TencentOcrSDKDemo.xcworkspace	前天 上午 9:42

## 步骤2: 若使用真机调试,配置您的苹果开发者账号

TARGETS > 应该是Signing&Capabilities > Signing 设置

### 步骤3:修改配置信息

修改项目中的 SecretId SecretKey 值,填入您在腾讯云控制台的 API 密钥。

### **△** 注意

本文提到的将 secretId、secretKey 直接配置到客户端中的方式,很容易被反编译逆向破解,一旦您的密钥泄密,攻击者可以使用您的 secretId、secretKey 信息进行 OCR 识别请求,给您造成损失。因此该方法**仅适用于本地跑通 Demo 和功能调试。** 

我们建议您将 secretId、secretKey 配置到服务器端,同时客户端使用临时密钥兑换的方式进行 OCR 识别。具体流程可以参考 临时密钥兑换流程 。

# 步骤4:编译运行

完成配置之后,选择您的运行机器(模拟器&真机)。(tips:真机才能体验摄像头拍照识别功能的)。



# 跑通 Demo (Android)

最近更新时间: 2024-12-04 14:26:53

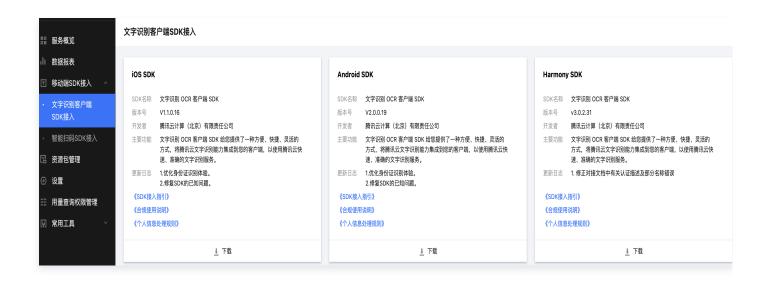
本文主要介绍如何快速运行腾讯云 文字识别 OCR Demo。

### 环境要求

- 最低兼容 Android 4.4 (SDK API Level 19),建议使用 Android 5.0 (SDK API Level 21)及以上版本。
- Android Studio 4.0 及以上版本
- App 要求 Android 4.4 及以上的设备

### 前提条件

- 您已 注册腾讯云 账号, 并完成 实名认证 。
- 进入 文字识别控制台,即可开通相应服务,并在 账号中心 获取 API 密钥。
- 前往文字识别客户端 SDK 接入页面 下载对应 SDK,并将 SDK 的 demo 文件下载到本地。

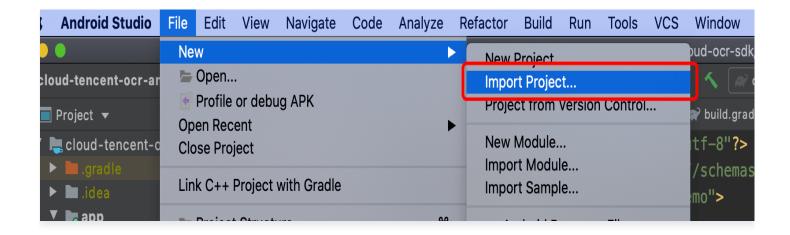


# 操作步骤

步骤1: 导入工程

打开 Android Studio,选择导入工程选项。





随后选择 demo 项目,将 demo 项目导入打开。

▼ ■ cloud-tencent-ocr-android	今天 上午 9:39
▶ <b>app</b>	今天 上午 9:43
▶ 🛅 build	今天 上午 9:43

### 步骤2: 修改配置信息

打开文件 SecretPamera 类,将在后台获取到的 secretId、secretKey 更新为您的信息。

```
/**

* 密钥配置信息

*/
public class SecretPamera {
   public final static String secretId = "您的 secretId";
   public final static String secretKey = "您的 secretKey";
}
```

### △ 注意

本文提到的将 secretId、secretKey 直接配置到客户端中的方式,很容易被反编译逆向破解,一旦您的密钥泄密,攻击者可以使用您的 secretId、secretKey 信息进行 OCR 识别请求,给您造成损失。因此该方法**仅适用于本地跑通 Demo 和功能调试**。

我们建议您将 secretId、secretKey 配置到服务器端,同时客户端使用临时密钥兑换的方式进行 OCR 识别。具体流程可以参考 临时密钥兑换流程 。



# 步骤3:编译运行

完成配置之后可以单击 Android Studio 的运行按钮,体验 Demo。



# 跑通 Demo (Harmony)

最近更新时间: 2024-12-04 14:26:53

本文主要介绍如何快速运行腾讯云 文字识别 OCR Demo。

## 环境要求

- 最低兼容手机系统 API level 11。
- DevEco 5.0.3.及以上版本(建议使用最新的开发工具开发)。

# 前提条件

- 您已 注册腾讯云 账号,并完成 实名认证。
- 进入 文字识别控制台,即可开通相应服务,并在 账号中心 获取 API 密钥。
- 前往文字识别客户端 SDK 接入页面 下载对应 SDK,并将 SDK 的 demo 文件下载到本地。



# 操作步骤

# 步骤1: 打开 Demo 工程

使用 DevEco 开发工具导入 SDK 交付包内的 Demo 源码。

# 步骤2:修改配置信息

打开文件 OcrCardSelectPage.ets,将在后台获取到的 secretId、secretKey 更新为您的信息。

```
let config: OcrSDKConfig = new OcrSDKConfigBuilder()
    .setSecretID(secretId)
    .setSecretKey(secretKey)
```



### **企 注意**

本文提到的将 secretId、secretKey 直接配置到客户端中的方式,很容易被反编译逆向破解,一旦您的密钥泄密,攻击者可以使用您的 secretId、secretKey 信息进行 OCR 识别请求,给您造成损失。因此该方法**仅适用于本地跑通 Demo 和功能调试**。

我们建议您将 secretId、secretKey 配置到服务器端,同时客户端使用临时密钥兑换的方式进行 OCR 识别。具体流程可以参考 临时密钥兑换流程 。

# 步骤3:编译运行

完成配置之后可以单击 DevEco 的运行按钮,体验 Demo。



# 一分钟集成 SDK 快速集成(iOS)

最近更新时间: 2025-01-13 11:22:02

# 开发准备

- 1. 注册腾讯云账号,单击进入 文字识别控制台,即可开通相应服务。
- 2. 在 账号中心 获取 API 密钥。
- 3. 前往文字识别客户端 SDK 接入页面 下载对应 SDK。



# iOS 端 OCR SDK 接入流程

# iOS 端 OCR SDK 介绍

SDK 中包含了以下 framework 库以及资源文件:

- OcrSDKKit.framework OCR 对外接口、页面设置及网络请求库
- YTImageRefiner.framework 图片解析
- tiny\_opencv2.framework opencv 库
- tnn.framework 底层深度学习库
- OcrSDK.bundle 资源文件
- Photos.framework 相册资源库

### 环境依赖

- 当前 iOS OCR 识别 SDK 版本适用于 iOS 11.0 及以上的版本
- 开发工具使用 xcode11 或以上版本集成开发

### 接入步骤



将 ocr Framework、系统 Framework 库以及 bundle 文件都添加至项目中。

```
├── OcrSDKKit.framework
├── YTImageRefiner.framework
├── tiny_opencv2.framework
└── tnn.framework
//系统库
├── Accelerate.framework
├── Photos.framework
├── PhotosUI.framework
└── CoreML.framework
```

```
//资源文件

L—— OcrSDK.bundle
```

- 2. 添加编译选项
  - 将**调用 SDK的ViewController** 设置为 **Objective-C++Source** 或者更改后缀为.mm(sdk 内部使用了 Objective-C++ 语法)
  - Other Linker Flags添加为 -ObjC
- 3. 权限设置

OCR SDK 需要手机网络、 摄像头、访问相册的使用权限,请添加对应的权限声明。

```
<key>Privacy - Camera Usage Description</key>
<string>OCR识别需要开启您的摄像头权限,用于识别</string>
<key>Privacy - Photo Library Usage Description</key>
<string>OCR识别需要您开启相册权限,浏览您的照片</string>
<key>Prevent limited photos access alert</key>
<Boolean>YES</Boolean>
```

# SDK 接口说明

### SDK 初始化

客户初始化 OCR SDK

```
#import <OcrsDKKit/OcrsDKKit.h>
#import <OcrsDKKit/OcrsDKConfig.h>

static NSString* const SECRET_ID = @""; // SECRET_ID 信息
static NSString* const SECRET_KEY = @""; // SECRET_KEY 信息
```



```
* OCR 配置类:

* ocrModeType: 检测类型 OCR_DETECT_MANUAL 手动拍摄;

OCR_DETECT_AUTO_MANUAL 自动识别卡片

*/

OcrSDKConfig *ocrSDKConfig = [[OcrSDKConfig alloc] init];

ocrSDKConfig.ocrModeType = _ocrModel;

/// SDKKIt 加载 OCR 配置信息

/// @param secretId Secret id

/// @param secretKey Secret key

/// @param ocrConfig ocr 配置类

[[OcrSDKKit sharedInstance] loadSDKConfigWithSecretId:nil

withSecretKey:nil withConfig:ocrSdkConfig];
```

### 进入 OCR 主页面

```
OCR UI 配置类:
CustomConfiqUI *customConfiqUI = [[CustomConfiqUI alloc] init];
/// 启动 SDK 模块,运行带有 UI 界面的功能识别模块
/// @param OcrType 识别模式
/// @param customConfigUI UI 配置对象
/// @param onProcessSucceed 成功回调 block
/// @param onProcessFailed 失败回调 block
[[OcrSDKKit sharedInstance] startProcessOcr:IDCardOCR_BACK
withSDKUIConfig:customConfigUI withProcessSucceedBlock: ^(id _Nonnull
resultInfo, UIImage *resultImage, id _Nonnull reserved) {
   ///resultInfo 识别成功信息(json)
 ///resultImage 识别成功后截取的图片
} withProcessFailedBlock: ^(NSError * _Nonnull error, id _Nullable
reserved) {
 ///error 错误信息
 ///reserved 一般会回传 requestid, 定位错误
```

### 更新临时密钥

OCR SDK 支持使用临时密钥接口,使用临时密钥的好处主要有以下两点,第一将固定密钥与终端分离可以增加安全性;第二因为兑换临时密钥是您完全可控的行为,因此您可以根据自定义规则来控制最终用户的接口访问权限。因



### 此建议您使用临时密钥的方式,具体可以参考文档(临时密钥文档与流程链接)

```
/// @param tmpSecretId 临时 SecretId
/// @param tmpSecretKey 临时密钥信息
/// @param token 临时兑换 token
[[OcrSDKKit sharedInstance] updateFederationToken:tmpSecretId
withTempSecretKey:tmpSecretKey withToken:token];
```

### SDK 资源释放

```
/// 清理 SDK 资源
[OcrSDKKit clearInstance];
```

### 目前 ocr SDK 支持七种类型的识别模式如下表所示。

OcrType 类型	代表含义
OcrType.IDCardOCR_FRONT	身份证人像面识别模式
OcrType.IDCardOCR_BACK	身份证国徽面识别模式
OcrType.BankCardOCR	银行卡正面识别模式
OcrType.BusinessCardOCR	名片卡正面识别模式
OcrType.MLIdCardOCR	马来西亚身份证识别模式
OcrType.LicensePlateOCR	汽车车牌识别模式
OcrType.VinOCR	汽车 VIN 码识别模式
OcrType.VehicleLicenseOCR_FRONT	行驶证主页识别模式
OcrType.VehicleLicenseOCR_BACK	行驶证副页识别模式
OcrType.DriverLicenseOCR_FRONT	驾驶证主页识别模式
OcrType.DriverLicenseOCR_BACK	驾驶证副页识别模式
OcrType.VinOCR2	汽车VIN码识别模式(高精度版)
OcrType.HKIDCardoCR_03	中国香港身份证03版本识别模式
OcrType.IDCardOCR_HK18	中国香港身份证18版本识别模式
OcrType.PermitOCR	港澳台通行证识别模式



OcrType.MLIDPassportocR	国际护照识别模式
OcrType.HmtResidentPermitOCR	港澳台居住证

### 常见错误

- 1. 当提示 **requsetConfigDict is nil**,检查下是不是在进入 SDK 时,执行了 [OcrSDKKit cleanInstance] 把密钥和配置设置清除了。
- 2. SDK 页面依托于 UIWindow,所以需要在 AppDelegate.h 中添加 @property (nonatomic, strong) UIWindow \* window;。
- 3. 当出现进入 SDK 黑屏,添加设置Other Linker Flags添加 -ObjC。打印日志 Application tried to push a nil view controller on target....,原因是 self.storyboard 等于 nil,可以参考 demo,在调用 SDK 页面的 ViewController 手动加载 xib 页面,然后调用 SDK 进入识别页面。
- 4. Building for iOS Simulator, but the linked and embedded framework 'OcrSDKKit .framework' was built for iOS + iOS Simulator.

解决方法是: Buil Settings --> Build Options --> Validate Workspace 改为Yes



# 快速集成(Android)

最近更新时间: 2025-01-13 11:22:03

# 开发准备

- 1. 注册腾讯云账号,单击进入 文字识别控制台,即可开通相应服务。
- 2. 在 账号中心 获取 API 密钥。
- 3. 前往文字识别客户端 SDK 接入页面 下载对应 SDK。



# Android 端 OCR SDK 接入流程

# Android 端OCR SDK 介绍

SDK 提供的文件为 OCR\_Android\_SDK\_V1.0.9,该文件封装了 OCR 识别终端能力。目前包括了身份证识别、银行卡识别、名片识别、车辆 Vin 码识别、车牌识别、驾驶证识别以及行驶证识别。

# 环境依赖

当前 Android 端 OCR 识别 SDK 适用于 API 19 (Android 4.4) 及以上版本。

# 接入步骤

- 1. 将 OcrSDK-public-v2.0.0.1-release.aar、OcrSDK-common-model-v1.0.0-release.aar 和 公共库 tencent-ai-sdk-aicamera-1.0.18-release.aar、tencent-ai-sdk-common-1.1.27-release.aar添加到您工程目录的 libs 目录下。
- 2. 在您工程的 build.gradle 中进行如下配置:

```
dependencies {
// 依赖腾讯云的 OcrSDK 的 aar
implementation files('libs/OcrSDK-common-model-v1.0.0-release.aar')
```



```
implementation files('libs/OcrSDK-public-v2.0.0.1-release.aar')
implementation files('libs/tencent-ai-sdk-aicamera-1.0.18-
release.aar')
implementation files('libs/tencent-ai-sdk-common-1.1.28-
release.aar')
// OCR SDK 返回实体对象需要的依赖
implementation 'com.google.code.gson:gson:2.8.5'
}
```

3. 同时需要在 AndroidManifest.xml 文件中进行必要的权限声明

```
<!--摄像头使用权限-->
<uses-feature android:name="android.hardware.camera" />
<uses-permission
    android:name="android.permission.CAMERA"
    android:required="true" />
<!--文件存储使用权限[可选]-->
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<!--网络访问权限-->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.permission.INTERNET" />
<uses-permission android:name="android.permission.permission.permission.permission.permission.permission.permission.permission.permission.permission.permission.permission.permission.permission.permission.permission.permission.permissi
```

对于需要兼容 Android 6.0 以上的用户,以上权限除了需要在 AndroidManifest.xml 文件中声明权以外,还 需使用代码动态申请权限。

# SDK 接口使用说明

#### SDK 初始化:

使用 OCR SDK 之前需要进行初始化,您可以按照您的需求设置默认值。



```
// 初始化 SDK
OcrSDKKit.getInstance().initWithConfig(this.getApplicationContext(),
configBuilder);
```

### 目前 OCR SDK 支持的业务模式:

OcrModeType 类型	代表含义
OCR_DETECT_MANUAL	手动拍摄模式
OCR_DETECT_AUTO_MANUAL	自动识别模式(tips:20s后提示,是否切换到手动拍摄)

### 更新临时密钥

OCR SDK 支持使用临时密钥接口,使用临时密钥的好处主要有以下两点,第一将固定密钥与终端分离可以增加安全性;第二因为兑换临时密钥是您完全可控的行为,因此您可以根据自定义规则来控制最终用户的接口访问权限。因此建议您使用临时密钥的方式,具体可以参考文档(<mark>临时密钥文档与流程链接</mark>)

```
OcrSDKKit.getInstance().updateFederationToken(tmpSecretId, tmpSecretKey, token);
```

### OCR 识别(返回 Json 字符串):

当您需要使用 OCR 识别的功能的时候,您可以直接调用识别接口,进行 OCR 业务识别。

```
// 启动 ocr 识别,识别类型为身份证正面
OcrSDKKit.getInstance().startProcessOcr(MainActivity.this,
OcrType.IDCardOCR_FRONT, customConfigUi, new ISDKKitResultListener() {
    @Override
    public void onProcessSucceed(String response, String srcBase64Image,
String requestId) {
        popTip(response, "Succeed"); // 展示 ocr 识别结果
    }

    @Override
    public void onProcessFailed(String errorCode, String message, String requestId) {
        popTip(message, errorCode); // 展示 ocr 识别错误信息
    }
});
```

### OCR 识别(返回对象实体类):



当您需要使用 OCR 识别功能,同时需要直接获取实体对象而非 JsonString 时,可以使用此方法。

```
OcrSDKKit.getInstance().startProcessOcrResultEntity(OcrTypeIdCardActivit y.this,
OcrType.IDCardOCR_FRONT, null, IdCardOcrResult.class,
new ISdkOcrEntityResultListener<IdCardOcrResult>() {
     @Override
     public void onProcessSucceed(IdCardOcrResult

idCardOcrResult, String base64Str) {
        Log.e(TAG, "IdCardOcrResult:" +

idCardOcrResult.toString()); // OCR 识别成功 IdCardOcrResult

}

@Override
     public void onProcessFailed(String errorCode, String
message, String requestId) {
        Log.e(TAG, "errorCode:" + errorCode + " message:" +

message); // OCR 识别失败 IdCardOcrResult
     }
});
```

### 目前 OCR SDK 支持几种类型的识别模式如下表所示,以及对应的实体类返回结果。

OcrType 类型	代表含义	对应结果实体类
OcrType.IDCardOCR_FRO	身份证人像面识别模式	IdCardOcrResult
OcrType.IDCardOCR_BAC K	身份证国徽面识别模式	IdCardOcrResult
OcrType.BankCardOCR	银行卡正面识别模式	BankCardOcrResul t
OcrType.BusinessCardOC R	名片卡正面识别模式	BusinessCardOcrR esult
OcrType.VinOCR	车辆的 VIN 识别模式	VinOcrResult
OcrType.LicensePlateOCR	车辆的车牌识别模式	CarLicensePlateRe sult
OcrType.DriverLicenseOC R_FRONT	驾驶证主页识别模式	DriverLicenseCard Result



OcrType.DriverLicenseOC R_BACK	驾驶证副页识别模式	DriverLicenseCard Result
OcrType.VehicleLicenseO CR_FRONT	行驶证主页识别模式	VehicleLicenseCar dResult
OcrType.VehicleLicenseO CR_BACK	行驶证副页识别模式	VehicleLicenseCar dResult
OcrType.GENERAL_VIN	车辆的VIN码通用识别模式(主要推荐拍 照模式使用)	VinOcrResult
OcrType.IDCardOCR_HK0	香港身份证03版本识别模式	HKIDCardOcrResul t
OcrType.IDCardOCR_HK1	香港身份证18版本识别模式	HKIDCardOcrResul t
OcrType.Exit_Entry_HK_ Macao_Card	港澳台通行证识别模式	PermitOcrResult
OcrType.MLID_PASSPOR T	国际护照识别模式	MLIDPassportOcrR esult
OcrType.HMT_RESIDENT _PERMIT_OCR	港澳台居住证	HmtResidentPermi tOcrResult

### SDK 版本号获取:

OCR SDK 提供了直接获取 SDK 版本号的接口,您可以调用此接口获取。

```
OcrSDKKit.getInstance().getVersion()
```

### SDK 资源释放:

在您 App 退出使用或者需要重启加载 OCR 功能的时候,可以调用 SDK 资源释放接口。

```
@Override
protected void onDestroy() {
   if (mDialog != null) {
      mDialog.dismiss();
   }
   // 释放 OCR SDK 的资源
   OcrSDKKit.getInstance().release();
   super.onDestroy();
```



```
}
```

## 混淆规则配置

如果您的应用开启了混淆功能,为确保 SDK 的正常使用,请把以下部分添加到您的混淆文件中。

```
#保留自定义的 OcrSDKKit 类和类成员不被混淆
-keep class com.tencent.ocr.sdk.** {*;}
#第三方 jar 包不被混淆
-keep class com.tencent.youtu.** {*;}
#公共库相关内容不混淆
-keep class com.tencent.could.** {*;}
```

### 常见问题

1. 如同时集成其余 SDK,出现 More than one file was found with OS independent path 'lib/armeabi−v7a/libc++\_shared.so'. 的问题。

主要是 OCR SDK 和其余 SDK 都添加了 libc++\_shared.so 这个库,解决办法可以在 build.gradle 中添加如下配置:

2. 如果集成方使用了 AndResGuard 的混淆工具,可以添加混淆配置:

```
// for OCR SDK
"R.string.ocr_*",
"R.string.rst_*",
"R.string.net_*",
"R.string.msg_*",
```

3. 集成 OCR SDK 后如果出现 Invoke-customs are only supported starting with Android O (-- min-api 26) 错误?

需要在 build.gradle 中添加如下配置:



```
// java 版本支持1.8

compileOptions {
   sourceCompatibility JavaVersion.VERSION_1_8
   targetCompatibility JavaVersion.VERSION_1_8
}
```

4. 如果使用了 AutoSize 组件,同时又使用了 OCR 的横屏模式时,横屏模式出现页面元素大小异常的问题。主要原因是 AutoSize 默认设置为竖屏情况下的宽高基准,可以在 Application 里注册下面的回调,实现横竖屏基准自适应:

```
* AutoSize随界面横竖自适应方法,可在Application中注册
* 以design_width_in_dp 360, design_height_in_dp 640 为例(客户可以修改成自
己定义的)
     @Override
     public void onAdaptBefore(Object target, Activity activity) {
         // 设置当前屏幕的大小
         // 获取当前Activity对应的屏幕方向
         // 如果是横屏,调整对应的屏幕基准
         if (orientation == Configuration.ORIENTATION_LANDSCAPE) {
```



```
AutoSizeConfig.getInstance().setDesignWidthInDp(DESIGN_HEIGHT_DP);

AutoSizeConfig.getInstance().setDesignHeightInDp(DESIGN_WIDTH_DP);

} else {
    // 如果是竖屏,调整对应的屏幕基准

AutoSizeConfig.getInstance().setDesignWidthInDp(DESIGN_WIDTH_DP);

AutoSizeConfig.getInstance().setDesignHeightInDp(DESIGN_HEIGHT_DP);

}

@Override
    public void onAdaptAfter(Object target, Activity activity) {

}

});

});
```

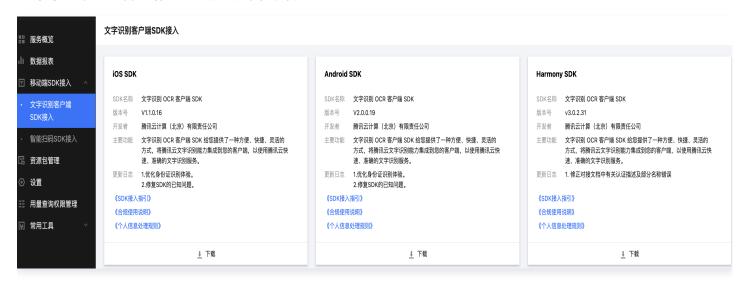


# 快速集成 (Harmony)

最近更新时间: 2025-01-13 11:22:03

# 开发准备

- 1. 注册腾讯云账号,单击进入 文字识别控制台,即可开通相应服务。
- 2. 在 账号中心 获取 API 密钥。
- 3. 前往文字识别客户端 SDK 接入页面 下载对应 SDK。



# Harmony 端 OCR SDK 接入流程

# Harmony 端OCR SDK 介绍

SDK 提供的文件为 OCR\_Harmony\_Public\_SDK\_3.x.x.x.har(具体版本号以 SDK 交付件为准),该文件封装了 OCR 识别终端能力。目前包括了身份证识别、银行卡识别、名片识别、车辆 Vin 码识别、车牌识别、驾驶证识别以及行驶证识别。

# 环境依赖

当前 Harmony 端 OCR 识别 SDK 适用于 API 11 (Harmony Next 4.0) 及以上版本。

# 接入步骤

1. 将OCR\_Harmony\_Public\_SDK\_3.x.x.x.har添加到您工程目录的 libs 目录下。



### 2. 添加模型文件

### △ 注意:

模型文件在交付包内sdk/models下,模型文件包含通用模型和身份证专用模型(身份证专用模型可以检测出身份证正反面),不可更改文件名,按需选择合适的模型。

3. 在您工程的 oh-package.json5 中进行配置修改,具体可以参考下面内容:

```
{
    "license": "",
    "author": "",
    "name": "HuiyanHarmonyDemo",
    "description": "Please describe the basic information.",
    "main": "",
    "version": "1.0.0",
    "dependencies": {
        // 添加依赖
        "OcrPublicSDK": "file:./libs/OCR_Harmony_Public_SDK_3.x.x.x.har",
    }
}
```

# SDK 接口使用说明



### SDK 初始化:

使用 OCR SDK 之前需要进行初始化。

```
OcrSDKKit.getInstance().init(getContext());
```

### 目前 OCR SDK 支持的业务模式:

OcrModeType 类型	代表含义
OCR_DETECT_MANUAL	手动拍摄模式
OCR_DETECT_AUTO_MANUAL	自动识别模式(tips:20s后提示,是否切换到手动拍摄)

### 更新临时密钥

OCR SDK 支持使用临时密钥接口,使用临时密钥的好处主要有以下两点,第一将固定密钥与终端分离可以增加安全性;第二因为兑换临时密钥是您完全可控的行为,因此您可以根据自定义规则来控制最终用户的接口访问权限。因此建议您使用临时密钥的方式,具体可以参考文档(<mark>临时密钥文档与流程链接</mark>)

```
let fetch = new FetchToken()
fetch.fetchToken(false) // 返回临时的secretID, secretKey, token
```

### OCR 识别(返回 Json 字符串):

当您需要使用 OCR 识别的功能的时候,您可以直接调用识别接口,进行 OCR 业务识别 。

```
// 启动 ocr 识别,识别类型为身份证正面

let config: OcrSDKConfig = new OcrSDKConfigBuilder()
    .setOcrMode(this.ocrMode)
    .setAuto2ManualTimeout(this.timeout)
    .setCardType(ocrCardType)
    .setOcrUIConfig(uiConfig)
    .setSecretID(val[0])
    .setSecretKey(val[1])
    .setToken(val[2])
    .build();
OcrSDKKit.getInstance().startOcrByConfig(config, {
    onSuccess: (response: string, requestID: string, imgBase64: string) => {
        // 成功
    },
    onFail: (errorCode: number, errMsg: string) => {
```



```
// 失败
});
```

### SDK 资源释放:

在您 App 退出使用或者需要重启加载 OCR 功能的时候,可以调用 SDK 资源释放接口。

OcrSDKKit.getInstance().release();



# 客户端 API 文档 iOS API 概览

最近更新时间: 2024-11-25 09:24:03

iOS 端文字识别 SDK 主要涉及的类有 OcrSDKKit、OcrSDKConfig、CustomConfigUI、OcrCommmDef,下面对其支持的 API 做出详细说明。

#### **OcrSDKKit**

OcrSDKKit 是文字识别 OCR 的对外接口类,文字识别功能主要集中在这个类中。

API	功能描述
sharedInstance()	创建 OcrSDKKit 的单例
clearInstance()	主动释放资源
getVersion()	获取当前 SDK 的版本号信息
loadSDKConfig	初始化 SDK 配置信息
updateFederationToken()	主动更新临时密钥
startProcessOcr()	启动 OCR 识别

### sharedInstance()

- + (nonnull instancetype) sharedInstance;
- 功能描述:创建 OcrSDKKit 的单例。
- 返回结果: OcrSDKKit 的单例对象。

### clearInstance()

/// **清理** SDK **资源** 

• 功能描述:

主动释放资源的方法,可在退出应用或者需要清理资源的时候调用。



#### getVersion()

```
- (NSString *_Nonnull)getVersion;
```

• 功能描述:

获取 SDK 当前的版本号。

• 返回结果:

当前 SDK 的版本信息。

### loadSDKConfig()

```
/// SDKKIt 加载 OCR 配置信息 这里使用的密钥为固定密钥,当使用临时密钥时,secretId secretKey 填入 nil 空值
/// @param secretId Secret id
/// @param secretKey Secret key
/// @param ocrConfig ocr 配置类
- (int)loadSDKConfigWithSecretId:(NSString *)secretId withSecretKey:
(NSString *)secretKey withConfig:(OcrSDKConfig *)ocrSDKConfig;
```

• 功能描述: 初始化 SDK 信息。

• 传入参数:

参数类型	参数名称	参数含义
OcrSDKConfig	ocrSDKConfig	SDK 配置参数的实体类
OcrSDKUIConfi g	ocrSDKUIConfig	SDK UI 配置参数的实体类
NSString	secretId	SecretId 密钥
NSString	secretKey	SecretKey 密钥

## updateFederationToken()

```
/// @param tmpSecretId 临时 SecretId
/// @param tmpSecretKey 临时密钥信息
/// @param token 临时兑换 token
- (void)updateFederationToken:(NSString *_Nonnull) tmpSecretId
withTempSecretKey:(NSString *_Nullable)tmpSecretKey withToken:(NSString
*_Nonnull)token;
```



• 功能描述:

主动更新临时密钥信息,在您与服务器兑换得到临时密钥之后主动调用设置。

• 传入参数:

参数类型	参数名称	参数含义
NSString	tmpSecretId	临时密钥 SecretId
NSString	tmpSecretKey	临时密钥 SecretKey
NSString	token	兑换后的临时访问 token

#### startProcessOcr()

```
/// 启动 SDK 模块,运行带有 UI 界面的功能识别模块
/// @param ocrType 识别模式
/// @param CustomConfigUI ocrUI 配置类 当传入 nil 时使用默认配置
/// @param onProcessSucceed 成功回调 block
/// @param onProcessFailed 失败回调 block
- (void)startProcessOcr:(int)ocrType withSDKUIConfig:(CustomConfigUI
*)customConfigUI withProcessSucceedBlock:(OcrSDKKitProcessSucceedBlock
_Nonnull)onProcessSucceed withProcessFailedBlock:
(OcrSDKKitProcessFailedBlock _Nonnull)onProcessFailed;
```

• 功能描述:

启动 OCR 识别

• 传入参数:

参数类型	参数名称	参数含义
OcrType	ocrType	启动的 OCR 识别类型
CustomConfigUI	customConfigUI	ocrUI 配置类
OcrSDKKitProcess SucceedBlock	onProcessSucceed	识别成功的回调
OcrSDKKitProcess FailedBlock	onProcessFailed	识别失败的回调

### **OcrSDKConfig**

OcrSDKConfig 是在 OCR 初始化时需要传入的 SDK 的配置信息实体类。 支持参数及其默认值如下:



类型	名称	含义	默认值
OcrType	ocrType	默认识别类型	IDCardOCR_FRONT ,IDCardOCR_BACK 均代表 id_card
OcrModeTy pe	ocrModeTyp e	识别模式类型: OCR_DETECT_MANUAL 代表手动拍摄模式, OCR_DETECT_AUTO_MAN UAL 代码自动捕获模式 20s未检测提示是否切换至手动拍摄	OCR_DETECT_MAN UAL 默认值
long	auto_timeout _ms	自动捕捉模式下自动切换至手动拍 照模式的超时时间(5000ms <auto_timeout_ms<a>&lt;180000ms)</auto_timeout_ms<a>	20000ms
BOOL	cropldCard	开启身份证照片裁剪(去掉证件外 多余的边缘、自动矫正拍摄角度) 开关	NO
BOOL	cropPortrait	开启人像照片裁剪(自动抠取身份 证头像区域)	NO
BOOL	copyWarn	开启复印件告警	NO
BOOL	borderCheck Warn	开启边框和框内遮挡告警	NO
BOOL	reshootWarn	开启翻拍告警	NO
BOOL	detectPsWar n	开启 PS 检测告警	NO
BOOL	templdWarn	开启临时身份证告警	NO
BOOL	invalidDateW arn	开启身份证有效日期不合法告警	NO
BOOL	quality	开启图片质量分数(评价图片的模 糊程度)	NO
BOOL	multiCardDet ect	是否开启多卡证检测	NO
BOOL	reflectWarn	是否开启反光告警	NO



NSString	retImageTyp e	图像预处理,检测图片倾斜的角 度,将原本倾斜的图片围绕中心点 转正,最终输出一张正的名片抠 图。	空
BOOL	retlmage	马来西亚身份证接口是否返回图片	NO
BOOL	retBorderCutI mage	是否返回预处理(精确剪裁对齐) 后的 <b>银行卡 图片数据(仅</b> BankCardOCR类型下生效)	NO
BOOL	retCardNoIm age	是否返回银行卡卡号的切图图片数据(仅BankCardOCR类型下生效)	NO
BOOL	enableCopyC heck	银行卡复印件检测开关,如果输入的图片是银行卡复印件图片则返回告警(仅BankCardOCR类型下生效)	NO
BOOL	enableResho otCheck	银行卡翻拍检测开关,如果输入的 图片是银行卡翻拍图片则返回告警 (仅BankCardOCR类型下生效)	NO
BOOL	enableBorder Check	银行卡边框遮挡检测开关,如果输入的图片是银行卡边框被遮挡则返回告警(仅BankCardOCR类型下生效)	NO
BOOL	enableQuality Value	是否返回银行卡图片质量分数(图片质量分数是评价一个图片的模糊程度的标准)(仅BankCardOCR类型下生效)	NO
BOOL	detectFake	香港身份证是否鉴伪(仅 HKIDCardOCR_03、 HKIDCardOCR_18生效)	NO
BOOL	returnHeadI mage	香港身份证是否返回人像照片(仅 HKIDCardOCR_03、 HKIDCardOCR_18下生效)	NO

## CustomConfigUI

CustomConfigUI 是在启动 SDK 模块时需要传入的 SDK 的 UI 配置信息实体类。 支持参数及其默认值如下:



类型	名称	含义	默认值
BOOL	isShowTips	是否显示中间提示语	YES
NSString	tipsShowText	中间提示语内容(限制15个中文 字)	"请避免识别内容折角、遮 挡和反光"
NSString	remindDialog Text	dialog 模式转换提示文字	"未能识别证件,是否切换 模式拍照上传?"
UIColor	remindConfir mColor	dialog 模式转换提示,按钮颜色	RGBA: 510611
UIColor	cardFrameCo lor	卡片框选中颜色	RGBA: 510611
Ullmage	takePicturel mage	拍照按钮图标 80×80	默认图标
Ullmage	lightONImage	打开手电筒按钮图标 40x40	默认图标
Ullmage	lightOFFImag e	关闭手电筒按钮图标40x40	默认图标
Ullmage	albumImage	相册按钮图标40×40	默认图标
BOOL	isShowAlbum Btn	是否显示相册按钮	YES
BOOL	isHorizontal	是否横屏显示	NO
float	blurThreshol d	光线模糊的最低范围	0.5
int	validCount	稳定帧的帧数	4
OCR_Langu ageType	languageTyp e	设置当前使用语言	OCR_DEFAULT
OcrCamera Zoom	zoomType	设置默认的缩放倍数	OCR_ZOOM_1X
BOOL	openClipImag e	设置打开图片裁剪	YES
NSString	ocrSDKBundl ePath	设置OCRSDK bundle文件路 径,默认从main bundle下读取	nil



## OcrType

OcrType 是一个枚举类型,列举了当前文字识别 OCR 的 SDK 所支持业务类型的种类,大致如下:

OcrType 类型	代表含义
OcrType.IDCardOCR_FRONT	身份证人像面识别模式
OcrType.IDCardOCR_BACK	身份证国徽面识别模式
OcrType.BankCardOCR	银行卡正面识别模式
OcrType.BusinessCardOCR	名片卡正面识别模式
OcrType.MLIdCardOCR	马来西亚身份证识别模式
OcrType.LicensePlateOCR	汽车车牌识别模式
OcrType.VinOCR	汽车VIN码识别模式
OcrType.VehicleLicenseOCR_FRONT	行驶证主页识别模式
OcrType.VehicleLicenseOCR_BACK	行驶证副页识别模式
OcrType.DriverLicenseOCR_FRONT	驾驶证主页识别模式
OcrType.DriverLicenseOCR_BACK	驾驶证副页识别模式
OcrType.VinOCR2	汽车VIN码识别模式(高精度版)
OcrType.HKIDCardoCR_03	香港身份证03版本识别模式
OcrType.IDCardOCR_HK18	香港身份证18版本识别模式
OcrType.PermitOCR	港澳台通行证识别模式
OcrType.MLIDPassportocR	国际护照识别模式
OcrType.HmtResidentPermitOCR	港澳台居住证

## OcrModeType

OcrModeType 是一个枚举类型,列举了卡片识别模式。

OcrModeType 类型	代表含义
OCR_DETECT_MANUAL	手动拍摄模式

版权所有:腾讯云计算(北京)有限责任公司 第44 共95页



OCR_DETECT_AUTO_MANUAL	自动识别模式(tips: 20s后提示 是否切换 到手动拍摄)
------------------------	------------------------------------

## OCR\_LanguageType

OCR\_LanguageType是一个枚举类型,列举了语言设置。

类型	代表含义
OCR_DEFAULT	跟随系统
OCR_ZH_HANS	中文简体
OCR_ZH_HANT	中文繁体
OCR_EN	英文

### **OcrCameraZoom**

OcrCameraZoom是一个枚举类型,列举了相机默认缩放倍数。

类型	代表含义
OCR_ZOOM_1X	1倍
OCR_ZOOM_1_5X	1.5倍
OCR_ZOOM_2X	2倍

### 识别结果回调

文字识别 OCR 识别结果的回调类,用于接收识别结果以及错误异常。

```
///SDKKit 处理成功回调接口
///@param resultInfo 会根据不同的工作模式返回对应下的成功信息(一般都是网络回包
json 字段)
///@param reserved 预留位
typedef void (^OcrSDKKitProcessSucceedBlock)(id _Nonnull resultInfo,
UIImage *resultImage,id _Nonnull reserved);

/// SDKKIt 处理失败回调接口
/// @param error 处理过程中触发的异常错误
/// @param reserved 预留位
///tips
```



```
typedef void (^OcrSDKKitProcessFailedBlock)(NSError *_Nonnull error, id
_Nullable reserved);
```

#### ① 说明

用户取消文字识别退出会在 OcrSDKKitProcessFailedBlock 回调

- domain: "OcrSdk.UserCancelOcr" code: 200101
- NSLocalizedDescription: "用户主动停止文字识别"

身份证正面请求返回 resultInfo 结果示例:

```
"Name": "李明",
"Sex": "男",
"Nation": "汉",
"Birth": "1987/1/1",
"Address": "北京市石景山区高新技术园腾讯大楼",
"IdNum": "440524*******010014",
"Authority": "",
"ValidDate": "",
"AdvancedInfo": "{}",
"RequestId": "ab2c132e-9e1c-43d3-b0ef-9b4d80f00330"
}
```

身份证反面请求返回 resultInfo 结果示例:

```
"Name": "",
"Sex": "",
"Nation": "",
"Birth": "",
"Address": "",
"IdNum": "",
"Authority": "赵县公安局",
"ValidDate": "2010.07.21-2020.07.21",
"AdvancedInfo": "{}",
"RequestId": "0d394478-6d4d-48fc-8b19-552415bf46de"
}
```

银行卡请求返回 resultInfo 结果示例:



```
{
    "CardNo": "62257600888888888",
    "BankInfo": "招商银行(03080000)",
    "ValidDate": "08/2022",
    "RequestId": "46ab2e62-11e3-4d04-9fab-0abe18e7c927"
}
```

#### 名片请求结果返回 resultInfo 结果示例:

```
"Name": "姓名",
"Value": "艾米"
"Name": "职位",
"Value": "视觉设计师"
"Name": "部门",
"Value": "社交平台部"
"Name": "公司",
"Value": "Tencent腾讯"
"Name": "地址",
"Value": "深圳市南山区高新技术园科技中一路腾讯大厦"
"Name": "邮箱",
"Name": "手机",
```



```
},
{
    "Name": "微信",
    "Value": "amy001"
}
],
"RetImageBase64": "",
"RequestId": "98f8fcbf-933a-4e95-ac48-6f1a9308fs6h"
}
```

#### 马来西亚身份证请求结果返回 resultInfo 结果示例:

```
"Response": {
    "Name": "KAVIN ONG KHI MN",
    "ID": "710716-08-6085",
    "Address": "NO 11 PERSIARN PERAJRIT 4 TAMA PERAK 31400 IPOH ERAK",
    "Sex": "LEAKI",
    "Birthday": "",
    "Warn": [],
    "Image": "",
    "AdvancedInfo": "{\"ID\":{\"Confidence\":\"1.0000\"},\"Name\":
{\"Confidence\":\"0.9996\"},\"Address\":
{\"Confidence\":\"0.9997\"},\"Sex\":{\"Confidence\":\"0.9999\"}}",
    "Type": "MyKad",
    "RequestId": "c969da05-54e3-4d0a-a55d-b3ef90d4ebf5"
}
```

#### 车牌识别请求结果返回 resultInfo 结果示例:

```
{
    "Response": {
        "Number": "京NOL9U8",
        "Confidence": 99,
        "Rect": {
            "X": 217,
            "Y": 233,
            "Width": 170,
            "Height": 21
        },
```



```
"RequestId": "210103d3-db06-4691-abe0-c0853aae606b"
}
```

车辆 VIN 码识请求结果返回 resultInfo 结果示例:

```
{
    "Response": {
        "Vin": "LBV2B25G2E5069977",
        "RequestId": "c59d9002-6c8c-426d-b57f-a8837dee2c7c"
    }
}
```

行驶证主页和副页请求结果返回 resultInfo 结果示例:

```
"PlateNo": "沪AA1234",
"VehicleType": "小型轿车",
"Owner": "李明",
"Address": "上海市徐汇区田林路397号腾云大厦6F",
"UseCharacter": "非营运",
"Model": "别克牌SGM7151LAAA",
"Seal": "上海市公安局交通警寨总队"
```

驾驶证主页和副页请求结果返回 resultInfo 结果示例:



```
"Name": "李明",
"Sex": "男",
"Nationality": "中国",
"Address": "上海市徐汇区田林路397号腾云大厦6F",
"IssuingAuthority": "上海市公安局交通警察总队",
```

对于返回的错误码以及错误信息,可以参考 错误码。

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated.

Please check your signature is correct."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```



## Android API 概览

最近更新时间: 2025-04-10 10:20:22

Android 端文字识别 SDK 主要涉及的类有 OcrSDKKit、OcrSDKConfig、OcrType、OcrModeType、ISDKKitResultListener、ISdkOcrEntityResultListener 以及 OCR 识别返回实体类,下面对其支持的 API 做出详细说明。

#### **OcrSDKKit**

OcrSDKKit 是文字识别 OCR 的对外接口类,文字识别功能主要集中在这个类中。

API	功能描述
getInstance()	创建 OcrSDKKit 的单例
release()	主动释放资源
getVersion()	获取当前 SDK 的版本号信息
initWithConfig()	初始化 SDK 配置信息
updateFederationToken()	主动更新临时密钥
startProcessOcr()	启动 OCR 识别,返回 JsonString
startProcessOcrResultEntity()	启动 OCR 识别,返回不同模式对应实体对象

### getInstance()

public static OcrSDKKit getInstance(

- 功能描述:
  - 创建 OcrSDKKit 的单例。
- 返回结果:
  - OcrSDKKit 的单例对象。

#### release()

public void release()

• 功能描述:

主动释放资源的方法,可在退出应用或者需要清理资源的时候调用。



### getVersion()

public final String getVersion()

• 功能描述:

获取 SDK 当前的版本号。

• 返回结果:

当前 SDK 的版本信息。

### initWithConfig()

public void initWithConfig(Context context, OcrSDKConfig config)

• 功能描述:

初始化 SDK 信息。

• 传入参数:

参数类型	参数名称	参数含义
Context	context	应用的上下文信息
OcrSDKConfig	config	SDK 配置参数的实体类

### updateFederationToken()

public void updateFederationToken(final String tmpSecretId, final String tmpSecretKey, final String token)

• 功能描述:

主动更新临时密钥信息,在您与服务器兑换得到临时密钥之后主动调用设置。

• 传入参数:

参数类型	参数名称	参数含义
String	tmpSecretId	临时密钥 SecretId
String	tmpSecretKey	临时密钥 SecretKey
String	token	兑换后的临时访问 token

### startProcessOcr()



public void startProcessOcr(Activity activity, OcrType ocrType,
CustomConfigUi customConfigUi, ISDKKitResultListener resultListener)

#### • 功能描述:

启动默认界面进行 OCR 识别,可进行部分 UI 元素的自定义配置。(如 CustomConfigUi 传入 null,可使用默认 UI 配置)

传入参数:

参数类型	参数名称	参数含义
Activity	activity	启动 OCR 默认界面的当前界面 Activity 对象
OcrType	ocrType	启动的 OCR 识别类型
CustomConfigUi	customConfigUi	启动默认界面时候传入的界面配置参数,如需完全使用默认配置,可传入 null
ISDKKitResultListe ner	resultListener	用于接收 OCR 识别结果的回调对象

#### startProcessOcrResultEntity()

public <T extends OcrResult> void startProcessOcrResultEntity(Activity
activity, OcrType ocrType,CustomConfigUi customConfigUi, Class<T>
entity,ISdkOcrEntityResultListener<T> ocrEntityResultListener)

#### • 功能描述:

启动默认界面进行 OCR 识别,可进行部分 UI 元素的自定义配置。(如 CustomConfigUi 传入 null,可使用默认 UI 配置),根据指定不同的识别模式返回不同识别结果类型实体。

• 传入参数:

参数类型	参数名称	参数含义
Activity	activity	启动 OCR 默认界面的当前界面 Activity 对象
OcrType	ocrType	启动的 OCR 识别类型
CustomConfigUi	customConfigUi	启动默认界面时候传入的界面配置参数,如需完全使用默认配置,可传入 null
Class < T >	entity	OCR 识别结果的实体类



ISdkOcrEntityResu ItListener	ocrEntityResultListe ner	用于接收 OCR 识别结果的回调对象
< T > extends OcrResult	<t></t>	OCR 识别结果 OcrResult 的子类型

OCR 识别结果的实体类,OCR 识别结果 OcrResult 的子类型。

## **OcrSDKConfig**

OcrSDKConfig 是在 OCR 初始化时需要传入的 SDK 的配置信息实体类,采用构建者 builder 的方式进行参数配置。

#### 支持参数及其默认值如下:

类型	名称	含义	默认值
OcrT ype	ocrType	默认识别类型	IDCardOCR_FRONT, IDCardOCR_BACK均代 表 id_card
int	cardType	身份证模式时正反面0正,1反	0正面
int	modeType	识别模式类型: 0代表手动拍摄模式,1代码自动捕获模式,2代表自动+手动模式(先使用自动超时后转为手动拍照模式)	2代表自动 + 手动模式
int	autoTimeo ut	自动捕获超时(毫秒单位,最少设置5秒, 内部上限30秒)	20000毫秒
String	secretId	请求使用的密钥信息(如果使用固定密钥 模式,可传入固定密钥)	空
String	secretKey	请求使用的密钥信息(如果使用固定密钥 模式,可传入固定密钥)	空
String	tempToke n	请求使用的临时 token 信息	空
boole an	cropldCar d	开启身份证照片裁剪(去掉证件外多余的 边缘、自动矫正拍摄角度)开关	false
boole an	cropPortra it	开启人像照片裁剪(自动抠取身份证头像 区域)	false
boole an	copyWarn	开启复印件告警	false



boole an	borderChe ckWarn	开启边框和框内遮挡告警	false
boole an	reshootWa rn	开启翻拍告警	false
boole an	detectPsW arn	开启 PS 检测告警	false
boole an	templdWar n	开启临时身份证告警	false
boole an	invalidDate Warn	开启身份证有效日期不合法告警	false
boole an	quality	开启图片质量分数(评价图片的模糊程 度)	false
String	retImageT ype	图像预处理,检测图片倾斜的角度,将原本倾斜的图片围绕中心点转正,最终输出 一张正的名片抠图。	空
boole an	canCancel Waiting	允许在识别过程中(出现loading提示 框),点击back主动取消	false
boole an	isOpenClip Image	选择图片时是否启用采集功能	true
boole an	retBorder CutImage	银行卡专用,是否返回预处理(精确剪裁 对齐)后的银行卡图片数据	false
boole an	retCardNoI mage	银行卡专用,是否返回卡号的切图图片数据,默认false。	false
boole an	enableQua lityValue	银行卡专用,是否返回图片质量分数(图 片质量分数是评价一个图片的模糊程度的 标准)	false

## OcrType

OcrType 是一个枚举类型,列举了当前文字识别 OCR 的 SDK 所支持业务类型的种类,大致如下:

OcrType类型	代表含义	对应结果实体类
OcrType.IDCardOCR_FRO	身份证人像面识别模式	IdCardOcrResult

版权所有: 腾讯云计算(北京)有限责任公司 第55 共95页



OcrType.IDCardOCR_BAC	身份证国徽面识别模式	IdCardOcrResult
OcrType.BankCardOCR	银行卡正面识别模式	BankCardOcrResul t
OcrType.BusinessCardOC R	名片卡正面识别模式	BusinessCardOcrR esult
OcrType.VinOCR	车辆的 VIN 识别模式	VinOcrResult
OcrType.LicensePlateOCR	车辆的车牌识别模式	CarLicensePlateRe sult
OcrType.DriverLicenseOC R_FRONT	驾驶证主页识别模式	DriverLicenseCard Result
OcrType.DriverLicenseOC R_BACK	驾驶证副页识别模式	DriverLicenseCard Result
OcrType.VehicleLicenseO CR_FRONT	行驶证主页识别模式	VehicleLicenseCar dResult
OcrType.VehicleLicenseO CR_BACK	行驶证副页识别模式	VehicleLicenseCar dResult
OcrType.GENERAL_VIN	车辆的VIN码通用识别模式(主要推荐拍 照模式使用)	VinOcrResult
OcrType.IDCardOCR_HK0	香港身份证03版本识别模式	HKIDCardOcrResul t
OcrType.IDCardOCR_HK1	香港身份证18版本识别模式	HKIDCardOcrResul t
OcrType.Exit_Entry_HK_ Macao_Card	港澳台通行证识别模式	PermitOcrResult
OcrType.MLID_PASSPOR T	国际护照识别模式	MLIDPassportOcr Result
OcrType.HMT_RESIDENT _PERMIT_OCR	港澳台居住证	HmtResidentPermi tOcrResult

## OcrModeType

OcrModeType 是一个枚举类型,列举了卡片识别模式。



OcrModeType 类型	代表含义
OCR_DETECT_MANUAL	手动拍摄模式
OCR_DETECT_AUTO_MA NUAL	自动识别模式(tips:20s后提示 是否切换到手动拍摄)

## LanguageStyle

LanguageStyle 是一个枚举类型,列举了支持的语言类型。

LanguageStyle 类型	代表含义
AUTO	跟随系统语言
ENGLISH	英文
SIMPLIFIED_CHINESE	简体中文

## CustomConfigUi

此为用户自定义 UI 的配置类,当前支持配置的属性如下表所示,可以通过 javabean set 的方式设置。

类型	名称	含义
boo lea n	isShowTitleBar	设置是否显示默认界面的标题栏
Stri ng	titleBarText	设置标题栏显示的文字内容
int	titleColor	设置标题栏背景颜色(0xFFFFFF类型)
Stri ng	remindDialogText	设置提醒 dialog 信息文字内容
int	remindDialogTextC olor	设置提醒dialog信息文字颜色(0xFFFFFF类型)
int	remindDialogTextS ize	设置提醒dialog信息文字大小
int	remindDialogConfir mColor	设置提醒 dialog 中确认按钮(切换模式)的颜色(0xFFFFFF类型)



Stri ng	remindDialogConfir mText	设置提醒dialog中确认按钮的文字内容
int	remindDialogCanc elColor	设置提醒dialog中取消按钮的颜色(0xFFFFFF类型)
Stri ng	remindDialogCanc elText	设置提醒dialog中取消按钮的文字内容
boo lea n	remindDialogShow Title	是否显示提示dialog的title
int	cardFrameColor	设置卡片预览框四角选中时的颜色(OxFFFFFF类型)
int	successRemindTe xtColor	设置自定捕捉或拍摄成功后,提示的文字颜色(0xFFFFFF类型)
Int	statusBarColor	设置状态栏背景颜色(0xFFFFFF类型)
boo lea n	isShowStatusBar	设置默认界面是否显示状态栏,默认true
int	imageSelectResId	设置默认界面中图库中图片选择的图标资源 id
int	lightImageOnResId	设置默认界面打开闪光灯的图标资源 id
int	lightImageOffResId	设置默认界面关闭闪光灯的图标资源 id
int	takePicturesResId	设置默认界面主动拍照按钮的图标资源 id
int	backActionIconRe sld	设置默认界面返回按钮的图标资源id
boo lea n	isRemoveAlbum	设置默认界面是否显示相册选取按钮,默认为显示
boo lea n	isRemoveFlash	设置默认界面是否显示闪光灯点击按钮,默认显示
boo lea n	isLandscape	设置默认界面为横屏模式,默认false



boo lea n	isShowTips	设置默认界面是否显示tips框,默认true
Stri ng	showTipsText	设置提醒tips的文字内容,默认"请避免识别内容折角、遮挡和反 光"(不超过15字)
boo lea n	isShowStatusBar	设置默认界面是否显示状态栏,默认true
int	showTipsTextColo r	设置提醒tips的文字颜色(0xFFFFFF类型)
int	showTipsTextSize	设置提醒tips的文字大小
boo lea n	isShowTipsIcon	设置是否显示提醒tips上的icon图标
boo lea n	isShowTipsBackgr ound	置是否显示提醒tips的背景框
int	portraitLayoutResI d	竖屏布局的资源id,具体使用方法参考自定义UI布局
int	landscapeLayoutR esId	横屏布局的资源id,具体使用方法参考自定义UI布局
boo lea n	useDeepColorStat usBarlcon	是否使用深色的状态栏图标与文字(配合浅色状态栏颜色使用),默认 false
Stri ng	bottomTipsContext	预览框自带下方tips的内容的动态设置( <b>"\n",可以主动换行</b> )

### 自定义 UI 布局

OCR SDK支持自定义UI布局,SDK会提供默认的竖屏布局文件demo\_ocr\_detect\_fragment.xml和横屏布局文件demo\_ocr\_detect\_hor\_fragment.xml,集成方可以在此基础上进行UI的修改。不要删除或者修改默认的view与view对应的id,

相关资源可以在SDK提供的安装包的res文件夹内获取

下面提供了一份竖屏的默认布局文件:

<?xml version="1.0" encoding="utf-8"?>



```
相机的预览View -->
    整体的遮照View以及识别框
<!-- 自定义的Tips显示界面
<!-- 底部三个按钮对应的界面 -->
```



```
相册选择按钮
   <!-- 进行拍照按钮
   <!-- 闪光灯按钮 -->
<!-- 默认返回按钮对应的View -->
<!-- 上层的title布局
```



```
</RelativeLayout>
```

如果添加在默认布局上添加了 view,有需要对 view 进行控制的话,可以调用参考下面方式进行注册:

### 修改 CameraMaskView

CameraMaskView 是主体的遮罩组件,提供使用属性配置的方式进行修改,直接在 xml 里进行操作即可目前已 支持的属性如下表:

属性	类型	含义
txy_mask_color	color reference	外部蒙层的默认颜色
txy_tip_text	string reference	默认的提示文本
txy_tip_color	color reference	上方提示文字的颜色(默认)
txy_tip_light_color	color reference	上方提示文字的颜色(正确提醒)
txy_tip_error_color	color reference	上方提示文字的颜色(异常提醒)
txy_tip_size	integer dimension	上方提示文字的大小
txy_tip_margin	integer dimension	上方提示文字距离选择框的距离



txy_tip_textStyle	enum[normal、 bold、italic、 bold_italic]	上方提示的字体样式
txy_tip_show	boolean	是否显示上方提示,默认显示
txy_line_color	color reference	选择框的边线颜色(默认)
txy_line_light_color	color reference	选择框的边线颜色(正确提醒)
txy_line_error_color	color reference	选择框的边线颜色(异常提醒)
txy_line_width	integer dimension	选择框边线的宽度
txy_line_length	integer dimension	选择框边线的长度
txy_line_padding	integer dimension	选择框边线距离中心透明框的距离
txy_line_margin	integer dimension	选择框边线距离手机屏幕的距离
txy_position_flag	enum[center、 margin]	选择框的位置,center代表居中,margin代表 顶部
txy_mask_margin	integer dimension	txy_position_flag = margin,选择框的距离 顶部的位置
txy_mask_view_use _type	enum[portrait、 landscape]	portrait代表适配竖屏,landscape适配横屏
txy_rect_height_wei ght_hor	integer	仅适用于横屏,中间高度所占屏幕的比例
txy_front_head_icon	reference	正面人像图的替换图@drawable/head
txy_back_emblem_ic on	reference	背面徽章图的替换图@drawable/emblem
txy_bottom_tips_text Style	enum[normal、 bold、italic、 bold_italic]	下方提示的字体样式
txy_bottom_tips_txt	string reference	下方提示的默认文本内容, <b>("\n",可以主动换</b> 行)
txy_bottom_tip_size	integer dimension	下方提示的字体大小
txy_bottom_tip_color	color reference	下方提示的颜色



txy_bottom_tip_marg in_top	integer dimension	下方提示距离边线框上方的距离
txy_bottom_tip_sho w	boolean	下方提示是否显示,默认不显示
txy_is_show_scanlin e	boolean	是否显示内置扫描线,默认不显示
txy_scanline_animat or_time	integer	扫描线完成一次扫描动画需要的时间( <b>单位</b> <b>ms</b> ),默认1500ms
txy_scanline_start_c olor	color reference	扫描线动画区域的起始颜色,0~40%的区域为此颜色
txy_scanline_mid_co	color reference	扫描线动画区域的过渡颜色,40%~80%为起始颜色与此颜色的渐变效果
txy_scanline_end_co	color reference	扫描线动画区域的终止颜色,80%~100%为过渡颜色与此颜色的渐变效果
txy_scanline_start_c olor_size_range	integer	从0到这一位置的百分比都是起始颜色,默认40 (百分比)
txy_scanline_start_t o_mid_color_size_ra nge	integer	txy_scanline_start_color_size_range到 这个位置为起始到过渡色的渐变区间,默认80 (百分比)

```
<!-- demo里的参考扫描线的颜色 -->
<!-- 起始颜色 -->
<color name="txy_scanline_start_color">#00000000</color>
<!-- 过渡颜色 -->
<color name="txy_scanline_mid_color">#22F77C61</color>
<!-- 终止颜色 -->
<color name="txy_scanline_mid_color">#EEF77C61</color>
<color name="txy_scanline_end_color">#EEF77C61</color>
```

如果上面和 CustomConfigUi 传入的属性产生冲突时,CustomConfigUi 配置的优先级更高。

### 自定义 Tips 和部分 UI 文字

SDK 会提供 **demo\_tips\_string.xml**,如果需要修改 tips 提示,可以保持 key 不变,直接修改对应的 value。同时最后把 demo\_tips\_string.xml 放到依赖 ocr sdk 的 module 的资源下进行打包,Android studio 会自动完成资源替换、覆盖。目前支持的替换内容如下:

```
<?xml version="1.0" encoding="utf-8"?>
```



```
<!-- 默认dialog上的信息
   <string name="txy_ocr_cancel">取消</string>
   <string name="txy ocr ok">确定</string>
   <string name="txy_ocr_change_mode">切换模式</string>
   <string name="txy_ocr_tips">提示</string>
   <string name="txy_ocr_change_mode_info">未能识别内容,是否切换模式拍照上
传?</string>
   <string name="txy_ocr_loading">数据加载中</string>
   <string name="txy_ocr_back">返回</string>
   <string name="txy_ocr_title">标题</string>
   <string name="txy_ocr_setting">设置</string>
   <string name="txy_ocr_permission_info">您已拒绝或禁用权限,请手动授予允许
访问</string>
   <string name="txy_ocr_change_mode_manual">已切换至手动拍摄</string>
   <string name="txt_user_cancel_ocr">用户主动停止文字识别</string>
   <string name="txt_inner_sdk_ocr">文字识别内部错误,请重试</string>
   <string name="txt_select_image_error">选择图片发生异常</string>
   <string name="txy_auto_mode_time_out_error">未自动捕获到照片</string>
   <string name="txy_ocr_tip_default_txt">请避免识别内容折角、遮挡和反光
   <!-- OCR识别里的文字信息
   <string name="txy_ocr_tip_please">请将</string>
   <string name="txy_ocr_tip_id_card">身份证</string>
   <string name="txy_ocr_tip_bank_card">银行卡</string>
   <string name="txy_ocr_tip_business_card">名片</string>
   <string name="txy_ocr_tip_card">卡证</string>
   <string name="txy_ocr_tip_car_card">车牌</string>
   <string name="txy_ocr_tip_car_vin">车辆VIN码</string>
   <string name="txy_ocr_tip_driver_license">驾驶证</string>
   <string name="txy_ocr_tip_vehicle_license">行驶证</string>
   <string name="txy_ocr_tip_hold_in_kuang">置于此框内</string>
   <string name="txy_ocr_txt_identify">识别</string>
   <string name="txy_ocr_hk_idcard_03">香港身份证03</string>
   <string name="txy_ocr_hk_idcard_18">香港身份证18</string>
   <string name="txy_ocr_exit_entry_hk_macao_card">港澳通行证</string>
   <!-- 错误信息
   <string name="txy_ocr_error_result_txt">ocr识别请求失败</string>
   <!-- 错误信息
   <string name="txy_ai_camera_open_error">打开相机失败</string>
   <string name="txy_ai_camera_close_error">关闭相机失败</string>
   <string name="txy_ai_camera_preview_error">预览相机失败</string>
```



#### **IdCardOcrResult**

身份证 OCR 识别的结果实体对象

类型	名称	含义
String	name	姓名
String	sex	性别
String	nation	民族
String	birth	出生日期
String	address	住址
String	idNum	身份证号
String	authority	发证机关
String	validDate	有效日期
String	requestId	请求 id
String	advancedInfo	扩展字段内容 参考官网

### **BankCardOcrResult**

银行卡 OCR 识别的结果实体对象

类型	名称	含义	
----	----	----	--



String	bankInfo	银行信息
String	cardNo	银行卡号
String	validDate	有效日期
String	requestId	请求id
String	cardType	银行卡类型
String	cardName	银行卡名称
String	borderCutImage	切片图片数据,取不到为 null
String	cardNoImage	卡号图片数据,取不到为 null

#### **BusinessCardOcrResult**

名片 OCR 识别的结果实体对象,主要包含一系列的 BusinessCardItems 对象。

类型	名称	含义
List< BusinessCardIt ems >	businessCardInfos	名片元素列表
String	requestId	请求 id

### **BusinessCardItems**

名片 OCR 识别的基本元素,包括名称与信息。

类型	名称	含义
String	name	子项名称
String	value	子项内容

### **VinOcrResult**

车辆的 VIN 的识别的结果实体对象

类型	名称	含义
String	vin	车辆的 VIN 码

## CarLicensePlateResult

版权所有: 腾讯云计算 ( 北京 ) 有限责任公司 第67 共95页



类型	名称	含义
String	number	车牌的字符串信息
int	confidence	本次识别的置信度
OcrRect	rect	文本行在原图片中的像素坐标框

### **OcrRect**

#### 矩形坐标

类型	名称	含义
int	х	左上角x的坐标
int	У	左上角y的坐标
int	width	宽度
int	height	高度

## **DriverLicenseCardResult**

#### 驾驶证识别的结果实体类

类型	名称	含义
String	name	姓名
String	sex	性别
String	nationality	国籍
String	address	地址
String	dateOfBirth	出生日期
String	dateOfFirstIssue	初次领证日期
String	classType	准驾车类型
String	startDate	有效期开始的时间
String	endDate	有效期截止日期
String	cardCode	证件号



String	archivesCode	档案编号
String	record	记录
String	issuingAuthority	发证单位
List< Integer >	recognizeWarnCode	识别的警告码列表
List< String >	recognizeWarnMsg	识别的警告码说明

## VehicleLicenseCardResult

#### 行驶证识别的结果实体类

类型	名称	含义
VehicleFrontI nfo	vehicleFrontInfo	行驶证的主页信息
VehicleBackl nfo	vehicleBackInfo	行驶证的副页信息
List< Integer >	recognizeWarnCode	识别的 Code 告警码列表和释义
List< String >	recognizeWarnMsg	识别的警告码说明

## VehicleFrontInfo

#### 行驶证的主页的识别结果实体类

类型	名称	含义
String	plateNo	号牌号码
String	vehicleType	车辆类型
String	owner	所有人
String	address	住址
String	useCharacter	使用性质
String	model	品牌型号
String	vin	车辆识别代号

版权所有:腾讯云计算(北京)有限责任公司 第69 共95页



String	engineNo	发动机编号
String	registerDate	注册日期
String	IssueDate	发证日期
String	seal	印章

## VehicleBackInfo

#### 行驶证副页的识别结果实体类

类型	名称	含义
String	plateNo	号牌号码
String	fileNo	档案编号
String	allowNum	核定人数
String	totalMass	总质量
String	curbWeight	整备质量
String	loadQuality	核定载的质量
String	externalSize	外廓尺寸
String	marks	备注
String	record	检验记录
String	totalQuasiMass	总牵引质量

### **HKIDCardOcrResult**

#### 香港身份证03与18版的共同识别结果实体类

类型	名称	含义
String	cnName	中文姓名
String	enName	英文姓名
String	telexCode	中文姓名对应电码
String	sex	性别:"男M"或"女F"

版权所有:腾讯云计算(北京)有限责任公司 第70 共95页



String	birthday	出生日期
String	permanent	永久性居民身份证。 0: 非永久; 1: 永久; −1: 未知。
String	idNum	身份证号码
String	symbol	证件符号,出生日期下的符号,例如"***AZ"
String	firstlssueDat e	首次签发日期
String	currentIssue Date	最近领用日期
String	fakeDetectR esult	真假判断。 0:无法判断(图像模糊、不完整、反光、过暗等导致无法判断); 1:假; 2:真。 注意:此字段可能返回 null,表示取不到有效值。
List <int eger&gt;</int 	warningCod e	多重告警码,当身份证是翻拍、复印、PS件时返回对应告警码。 -9102:证照复印件告警 -9103:证照翻拍告警 -9104:证照PS告警 -9105:证照防伪告警

## **PermitOcrResult**

#### 港澳通行证的识别结果实体类

类型	名称	含义
String	name	姓名
String	englishName	英文姓名
String	number	证件号
String	sex	性别
String	validDate	有效期限
String	issueAuthority	签发机关



String	issueAddress	签发地点
String	birthday	出生日期

## MLIDPassportOcrResult

#### 国际护照识别结果对应的实体类

类型	名称	含义
String	id	护照ID
String	name	姓名
String	issuingCountry	发行国
String	nationality	国籍
String	dateOfBirth	出生日期
String	sex	性别(F女,M男)
String	dateOfExpiration	有效期
List <integer></integer>	warn	告警码 -9103 证照翻拍告警 -9102 证照复印件告警 -9106 证件遮挡告警
String	image	证件图片
String	advancedInfo	扩展字段: { ID:{ Confidence:0.9999 }, Name:{ Confidence:0.9996 } }
String	codeSet	最下方第一行 MRZ Code 序列
String	codeCrc	最下方第二行 MRZ Code 序列

## **HmtResidentPermitOcrResult**



#### 港澳台居住证识别结果的实体类

类型	名称	含义
String	name	证件姓名
String	sex	性别
String	birth	出生日期
String	address	地址
String	idCardNo	身份证号
int	cardType	0−正面 1−反面
String	validDate	证件有效期限
String	authority	签发机关
String	visaNum	签发次数
String	passNo	通行证号码

### **ISDKKitResultListener**

文字识别 OCR 识别结果的回调类,用于接收识别结果以及错误异常。

```
/**

* OCR识别结果的回调类

*/
public interface ISDKKitResultListener {

/**

* orc识别成功结果

*

* @param response 识别结果Json信息

* @param ocrProcessResult Ocr的识别结果,附带的额外信息

*/

void onProcessSucceed(String response, OcrProcessResult
ocrProcessResult);

/**

* orc识别异常

* orc识别异常
```



```
* @param errorCode 错误码

* @param message 异常信息

* @param ocrProcessResult Ocr的识别结果,附带的额外信息

*/

void onProcessFailed(String errorCode, String message,

OcrProcessResult ocrProcessResult);

}
```

OcrProcessResult 是识别结果额外的带的内容。

#### **OcrProcessResult**

OCR识别完成后,额外携带的数据内容。

类型	名称	含义	默认 值
Stri ng	requestId	本次识别请求对应的requestId(如果没有进行网络请求,就失败 了为空)	空
Stri ng	imageBase6 4Str	本次识别所使用的图片base64	空

#### 身份证正面请求返回 response 结果示例:

```
"Name": "李明",
"Sex": "男",
"Nation": "汉",
"Birth": "1987/1/1",
"Address": "北京市石景山区高新技术园腾讯大楼",
"IdNum": "440524********010014",
"Authority": "",
"ValidDate": "",
"AdvancedInfo": "{}",
"RequestId": "ab2c132e-9e1c-43d3-b0ef-9b4d80f00330"
}
```

#### 身份证反面请求返回 response 结果示例:

```
{
    "Name": "",
    "Sex": "",
```



```
"Nation": "",
"Birth": "",
"Address": "",
"IdNum": "",
"Authority": "赵县公安局",
"ValidDate": "2010.07.21-2020.07.21",
"AdvancedInfo": "{}",
"RequestId": "0d394478-6d4d-48fc-8b19-552415bf46de"
}
```

#### 银行卡请求返回 response 结果示例:

#### 名片请求结果返回 response 结果示例:

```
{
    "BusinessCardInfos": [
    {
        "Name": "姓名",
        "Value": "艾米"
    },
    {
        "Name": "职位",
        "Value": "视觉设计师"
    },
    {
        "Name": "部门",
        "Value": "社交平台部"
    },
}
```



```
"Name": "公司",
"Value": "Tencent腾讯"
"Name": "地址",
"Value": "深圳市南山区高新技术园科技中一路腾讯大厦"
"Name": "邮箱",
"Name": "手机",
"Name": "微信",
```

#### 车辆的 VIN 返回 response 结果示例:

```
{
   "Response": {
     "Vin": "LBV2B25G2E5069977",
     "RequestId": "c59d9002-6c8c-426d-b57f-a8837dee2c7c"
   }
}
```

#### 车辆的车牌返回 response 结果示例:

```
{
"Response": {
```



```
"Number": "京NOL9U8",

"Confidence": 99,

"Rect": {
    "X": 217,
    "Y": 233,
    "Width": 170,
    "Height": 21
    },

"RequestId": "210103d3-db06-4691-abe0-c0853aae606b"
}
```

驾驶证识别的response结果示例:

行驶证识别主页识别的response结果示例:

```
{
"FrontInfo": {
"PlateNo": "沪AA1234",
```



```
"VehicleType": "小型轿车",

"Owner": "李明",

"Address": "上海市徐汇区田林路397号腾云大厦6F",

"UseCharacter": "非营运",

"Model": "别克牌SGM7151LAAA",

"Vin": "ABCDEFGH123456789",

"EngineNo": "8B54321",

"RegisterDate": "2011-10-10",

"IssueDate": "",

"Seal": "上海市公安局交通警察总队"
},

"BackInfo": null,

"RecognizeWarnCode": [
    -9106
],

"RecognizeWarnMsg": [
    "WARN_DRIVER_LICENSE_PS_CARD"
],

"RequestId": "820916b4-b391-40a8-9203-7ae87e3f1954"
}
```

#### 行驶证识别副页识别的response结果示例:

```
"FrontInfo": null,
"BackInfo": {
    "PlateNo": "院AC4L50",
    "FileNo": "A00-00476807",
    "AllowNum": "5人",
    "TotalMass": "1837kg",
    "CurbWeight": "1431kg",
    "LoadQuality": "--",
    "ExternalSize": "4620×1780×1498mm",
    "TotalQuasiMass": "--",
    "Marks": "",
    "Record": "检验有效期至2015年11月皖A(4s)"
},
"RecognizeWarnCode": [
    -9106
],
    "RecognizeWarnMsg": [
    "WARN_DRIVER_LICENSE_PS_CARD"
],
```



```
"RequestId": "e2ebfaa0-19d3-4d2b-9ed8-7c7c21eb2b74"
}
```

港澳台居住证识别的response结果示例:

```
"Response": {
    "Name": "李优优",
    "Sex": "女",
    "Birth": "1997/6/2",
    "Address": "上海市徐汇区古美路1528号A1栋腾讯优图1号",
    "IdCardNo": "830000199706020042",
    "CardType": 0,
    "ValidDate": "",
    "Authority": "",
    "VisaNum": "0",
    "PassNo": "000",
    "RequestId": "f72e7048-f1e0-42f3-b0bf-f8730d48bb5c"
}
```

中国香港身份证识别的response结果示例:



```
}
}
```

港澳台通行证识别的response结果示例:

```
"Response": {
    "Name": "李明",
    "EnglishName": "LIMING",
    "Number": "C000000000",
    "Sex": "男",
    "ValidDate": "2018.10.09-2028.10.08",
    "IssueAuthority": "公安部出入境管理局",
    "IssueAddress": "广东",
    "Birthday": "1981.08.03",
    "RequestId": "3090debe-3662-4ef1-8784-6ef2fb59f75e"
  }
}
```

国际护照识别的response结果示例:

```
"Response": {
    "ID": "E6918C",
    "Name": "LIM HEG CHUN STEE",
    "IssuingCountry": "SGP",
    "Nationality": "SGP",
    "DateOfBirth": "",
    "Sex": "M",
    "DateOfExpiration": "230414",
    "Warn": [],
    "Image": "",
    "AdvancedInfo": "{\"IssuingCountry\":
{\"Confidence\":\"0.9500\"},\"Name\":{\"Confidence\":\"0.9500\"},\"ID\":
{\"Confidence\":\"0.9500\"},\"Sex\":
{\"Confidence\":\"0.9500\"},\"DateOfExpiration\":
{\"Confidence\":\"0.9500\"},\"DateOfExpiration\":
{\"Confidence\":\"0.9500\"},\"DateOfExpiration\":
{\"Confidence\":\"0.9500\"},\"DateOfExpiration\":
{\"Confidence\":\"0.9500\"},\"RequestId": "0ee989d3-d064-45ec-bccb-63f5064247b4"
}
```



```
}
```

对于返回的错误码以及错误信息,可以参考 错误码。

# **ISdkOcrEntityResultListener**

文字识别 OCR 识别结果的回调类,用于接收识别结果的实体类以及错误异常。

OcrProcessResult 是识别结果额外带的内容。

### 错误码

#### 公共错误码

错误码	说明
UnsupportedOperation	操作不支持。
ResourceInUse	资源被占用。
InternalError	内部错误。
RequestLimitExceeded	请求的次数超过了频率限制。



AuthFailure.SecretIdNotF ound	密钥不存在。 请在控制台检查密钥是否已被删除或者禁用,如状态正 常,请检查密钥是否填写正确,注意前后不得有空格。
LimitExceeded	超过配额限制。
NoSuchVersion	接口版本不存在。
ResourceNotFound	资源不存在。
AuthFailure.SignatureFail ure	签名错误。 签名计算错误,请对照调用方式中的签名方法文档检查签名 计算过程。
AuthFailure.SignatureExp ire	签名过期。Timestamp 和服务器时间相差不得超过五分钟,请检查本 地时间是否和标准时间同步。
UnsupportedRegion	接口不支持所传地域。
UnauthorizedOperation	未授权操作。
InvalidParameter	参数错误。
ResourceUnavailable	资源不可用。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.Unauthorized Operation	请求未授权。请参考 CAM 文档对鉴权的说明。
AuthFailure.InvalidSecretI	密钥非法(不是云 API 密钥类型)。
AuthFailure.TokenFailure	token 错误。
DryRunOperation	DryRun 操作,代表请求将会是成功的,只是多传了 DryRun 参数。
FailedOperation	操作失败。
UnknownParameter	未知参数错误。
UnsupportedProtocol	HTTP(S)请求协议错误,只支持 GET 和 POST 请求。
InvalidParameterValue	参数取值错误。
InvalidAction	接口不存在。
MissingParameter	缺少参数错误。
ResourceInsufficient	资源不足。



### 业务错误码

错误码	说明	
FailedOperation.ArrearsE rror	账号已欠费。	
FailedOperation.CountLi mitError	今日次数达到限制。	
FailedOperation.DetectFa iled	检测失败。	
FailedOperation.DownLo adError	文件下载失败。	
FailedOperation.EmptyIm ageError	图片内容为空。	
FailedOperation.EngineR ecognizeTimeout	引擎识别超时。	
FailedOperation.ldCardInf olllegal	身份证信息不合法(身份证号、姓名字段校验非法等)。	
FailedOperation.lmageBl ur	图片模糊。	
FailedOperation.ImageDe codeFailed	图片解码失败。	
FailedOperation.ImageNo BusinessCard	照片未检测到名片。	
FailedOperation.ImageNo IdCard	图片中未检测到身份证。	
FailedOperation.ImageNo Text	图片中未检测到文本。	
FailedOperation.ImageSiz eTooLarge	图片尺寸过大,请参考输出参数中关于图片大小限制的说明。	
FailedOperation.InvoiceM ismatch	发票数据不一致。	
FailedOperation.Languag eNotSupport	输入的 Language 不支持。	



FailedOperation.MultiCar dError	照片中存在多张卡。	
FailedOperation.OcrFaile d	OCR 识别失败。	
FailedOperation.UnKnow Error	未知错误。	
FailedOperation.UnOpen Error	服务未开通。	
InvalidParameter.ConfigF ormatError	Config 不是有效的 JSON 格式。	
InvalidParameter.Enginel mageDecodeFailed	图片解码失败。	
InvalidParameter.InvalidG TINError	无效的 GTIN。	
InvalidParameterValue.In validParameterValueLimit	参数值错误。	
LimitExceeded.TooLarge FileError	文件内容太大。	
ResourceNotFound.NoIn voice	发票不存在。	
ResourcesSoldOut.Charg eStatusException	计费状态异常。	

# SDK 本地错误码

错误码	说明	
OcrSdk.UserCancelOcr	用户主动停止文字识别	
OcrSdk.InnerOcrError	文字识别内部错误,请重试	
OcrSdk.CallInitFirst	SDK需要先调用init	
FailedOperation.OcrFaile d	服务识别OCR结果异常	
OcrSdk.InnerOcrErrorClip	SDK内部选图裁剪逻辑失败	



Error	
OcrSdk.PermissionError	用户拒绝权限的回调
OcrSdk.PermissionError_ GoSetting	用户拒绝权限的回调并且主动前往Setting界面



# Harmony API 概览

最近更新时间: 2024-08-23 15:56:11

Harmony 端OCR SDK 主要涉及如下几个类: API 接口类 OcrSDKKit、参数配置类 OcrSDKConfig、结果回调类 OcrResultCallback。

#### **OcrSDKKit**

OcrSDKKit 为OCR SDK 的对外接口类,主要逻辑也都是调用此类完成。

API	功能描述	
init()	初始化接口	
release()	资源释放接口	
startOcrByConfig()	启动OCR识别	
setOcrEventListener()	设置本次OCR识别的事件回调	
stopOcr()	停止OCR识别	
getVersion()	获得当前 SDK 的版本号	

### init()

public init(context: Context

功能介绍: OCR SDK 的初始化接口。

传入参数:

参数类型	参数名称	参数含义
Context	context	App 的上下文信息

# release()

public release (

功能介绍: OCR SDK 资源释放的接口。

# startOcrByConfig()



public startOcrByConfig(config: OcrSDKConfig, cb: OcrResultCallback)

功能介绍: OCR SDK 启动接口。

传入参数:

参数类型	参数名称	参数含义
OcrSDKConfig	config	SDK启动配置项
OcrResultCallback	cb	结果回调类

### setOcrEventListener()

public setOcrEventListener(callback: OcrEventCallback)

功能介绍:设置本次 OCR 识别的事件回调

传入参数:

参数类型	参数名称	参数含义
OcrEventCallback	callba ck	事件对应的回调信息

# stopOcr()

public stopOcr()

功能介绍: 停止 OCR 识别。

### getVersion()

public getVersion(): string

功能介绍:获取当前版本 OCR SDK 的版本号。

# **OcrSDKConfig**

OcrSDKConfig 是在启动 OCR SDK 时的配置实体类,主要包含了以下属性。

版权所有:腾讯云计算(北京)有限责任公司 第87 共95页



OcrMode	ocrMode	卡片识别模式	OCR_DETECT_ AUTO_MANUAL
number	auto2Ma nualTime out	自动捕获超时(毫秒单位,最少设置5秒,内 部上限30秒)	20000毫秒
OcrCardTyp e	cardType	卡片识别类型	COMMON_OCR
OcrUIConfig	ocrUICon fig	自定义UI配置	默认值
string	secretID	请求使用的密钥信息(如果使用固定密钥模 式,可传入固定密钥)	空
string	secretKe y	请求使用的密钥信息(如果使用固定密钥模 式,可传入固定密钥)	空
string	token	请求使用的临时 token 信息	空
boolean	cropldCa rd	开启身份证照片裁剪(去掉证件外多余的边 缘、自动矫正拍摄角度)开关	false
boolean	cropPortr ait	人像照片裁剪(自动抠取身份证头像区域)	false
boolean	copyWar n	开启复印件告警	false
boolean	borderCh eckWarn	开启边框和框内遮挡告警	false
boolean	reshootW arn	开启翻拍告警	false
boolean	detectPs Warn	开启 PS 检测告警	false
boolean	templdW arn	开启临时身份证告警	false
boolean	invalidDa teWarn	开启身份证有效日期不合法告警	false
boolean	quality	开启图片质量分数(评价图片的模糊程度)	false



boolean	multiCard Detect	是否开启正反面同框识别(仅支持二代身份证 正反页同框识别或临时身份证正反页同框识 别)	false
boolean	reflectWa rn	是否开启反光检测	false
boolean	retImage Type	图像预处理,检测图片倾斜的角度,将原本倾 斜的图片围绕中心点转正,最终输出一张正的 名片抠图。	false
boolean	retBorder CutImage	银行卡专用,是否返回预处理(精确剪裁对 齐)后的银行卡图片数据	false
boolean	retCardN olmage	银行卡专用,是否返回卡号的切图图片数据, 默认false。	false
boolean	enableQu alityValu e	银行卡专用,是否返回图片质量分数(图片质量分数是评价一个图片的模糊程度的标准)	false

# OcrCardType

### OCR 支持的卡片识别类型

OcrCardType 类型	含义
IDCardOCR_FRONT	身份证人像面识别模式
IDCardOCR_BACK	身份证国徽面识别模式
BankCardOCR	银行卡正面识别模式
BusinessCardOCR	名片卡正面识别模式
LicensePlateOCR	车辆的车牌识别模式
VinOCR	车辆的 VIN 识别模式
VehicleLicenseOCR_F RONT	行驶证主页识别模式
VehicleLicenseOCR_B ACK	行驶证副页识别模式
DriverLicenseOCR_FR ONT	驾驶证主页识别模式



DriverLicenseOCR_BA	驾驶证副页识别模式
IDCardOCR_HK03	香港身份证03版本识别模式
IDCardOCR_HK18	香港身份证18版本识别模式
Exit_Entry_HK_Macao _Card	港澳台通行证识别模式
HMT_RESIDENT_PER MIT_OCR	港澳台居住证
MLID_PASSPORT	国际护照识别模式

### **OcrMode**

OcrMode 是一个枚举类型,列举了卡片识别模式。

OcrModeType 类型	代表含义
OCR_DETECT_MANUAL	手动拍摄模式
OCR_DETECT_AUTO_MA NUAL	自动识别模式(tips:20s后提示 是否切换到手动拍摄)
OCR_DETECT_AUTO	自动识别模式

# **OcrResultCallback**

#### OCR结果的回调类

```
export interface OcrResultCallback {
    /**
    * 成功的回调
    *
    * @param response, 服务端返回的信息
    * @param requestID, 服务端的requestID
    * @param imageBase64Str SDK最优图jpg的base64编码
    */
    onSuccess: (response: string, requestID: string, imageBase64Str:
string) => void;
    /**
    * 失败的回调
    *
    * @param errorCode 错误码
```



```
* @param errMsg 错误详细信息

*/

onFail: (errorCode: number, errMsg: string) => void;
}
```

### **OcrEventCallback**

用来监听 OCR 过程中的事件的回调

```
export interface OcrEventCallback {
    /**

    * 额外的参数透传

    * @param type 事件类型

    * @param params

    */
    onOcrEvent: (type: OcrEventType, param: string) => void;
}
```

# **OcrEventType**

对应的事件类型与含义:

事件枚举	事件含义
EVENT_DETECT_RECT_INFO	检测框信息(检测框左上角的坐标以及检测框宽高, params形如:"{ x: 0, y: 0, w: 0, h: 0}")
EVENT_TIME_OUT	超时事件回调(倒计时启动、暂停、结束,params形如:start、pause、end)
EVENT_AUTO_TO_MANUAL	自动切手动时的事件回调(params无意义,可不关 注)

# **OcrUIConfig**

OCR 检测页面自定义UI的参数配置

类型	名称	含义	默认值
boolean	showLightIcon	是否显示闪光灯按钮	true
boolean	showAlbumIcon	是否显示相册选图按钮	true



boolean	showFixTips	是否显示'请避免识别内容折 角'提示	true
boolean	showRectCorne rLine	是否显示四个角的线条	true
boolean	showIDCardLog o	是否显示人头和国徽的轮廓 (大陆身份证)	true
boolean	keepDetectView	检测到最佳帧后,是否销毁检 测页面(需配合自定义动画效 果使用)	false
number	lightIconSize	闪光灯图标的宽度,高度自适 应	50
number	takePictureIcon Size	拍照图标的宽度,高度自适应	80
number	albumlconSize	相册图标的宽度,高度自适应	50
string	detectNormalCo lor	常规反馈Tips的颜色	#ffffff
string	detectCorrectC olor	正常反馈Tips颜色	#006eff
string	detectErrorColo r	异常反馈Tips颜色	#e94b2c
number	rectCornerLine Weight	检测框四角线条的粗细	3
number	rectCornerLineL en	检测框四角线条的长度	30
string	maskViewColor Style	检测页面的蒙版颜色样式	rgba(80, 80, 80, 0.8)
number	rectMarginTop	检测框距离顶部的距离	50
number	rectHorizontalM argin	检测框距离屏幕的横向距离	5
number	tipsMarginTop	tips提示距离顶部的距离	20
number	fixTipsMarginTo p	fix tips提示距离顶部的距离	350



string	switchModeDial ogTitleColor	超时dialog 标题的颜色	#000000
string	switchModeDial ogContentColor	超时dialog 内容的颜色	#000000
string	switchModeDial ogCancelColor	超时dialog 取消按钮的颜色	#000000
string	switchModeDial ogConfirmColor	超时dialog 确认按钮的颜色	#006eff
number	switchModeDial ogTitleFontSize	超时dialog 标题的字体大小	16
number	switchModeDial ogContentFontS ize	超时dialog 内容的字体大小	14
number	switchModeDial ogCancelFontSi ze	超时dialog 取消按钮的字体大 小	16
number	switchModeDial ogConfirmFontS ize	超时dialog 确认按钮的字体大 小	16
boolean	switchModeDial ogConfirmOnLef t	超时dialog 确认按钮是否显示 在左侧	false
(param: CommonUIMod el) => void	commonCustom UI	基础自定义UI(详情参见自定义 UI说明文档)	(param: CommonUIModel ): void => {}
(param: AnimationUIMo del) => void	animUl	自定义动画UI(详情参见自定义 UI说明文档)	(param: AnimationUIMod el): void => {}

# 自定义 UI

# 自定义 UI 显示

OCR SDK 支持自定义 UI,可通过 OcrUIConfig 配置。使用方式如下:

```
// 1. 自定义配置
let uiConfig = new OcrUIConfig()
```



#### 新增 UI 组件

这里主要介绍如何在核身页面增加自定义的 UI 组件以及如何绑定事件。使用方式如下:

```
/**

* 自定义UI

* 資param $$

*/

@Builder

MyCustomUI($$: CustomUIModel) {

Column() {

Button('自定义的Button').onClick(() => {

    // 自定义Uim应事件

    router.back();

}).margin({

    bottom: '3%'

}).backgroundColor(Color.Blue)

}.backgroundColor(Color.Red)

.margin({

    bottom: '3%'

})

// 1. 自定义配置

let uiConfig = new OcrUIConfig();

// 传入自定义布局

uiConfig.commonCustomUI = this.MyCustomUI;
```



### 自定义提示

您若需要自定义 OCR 页面的提示语,可以按照下面的方式实现(OCR SDK 的交付件中会提供一个翻译文件 ocr\_custom\_string.json,该文件包含了所有可进行自定义的文字)实现:

- 1. 打开名为 <module\_name> 的 module(集成 OCR SDK 的 module)
- 2. 将 ocr\_custom\_string.json 添加到对应的语言文件夹下 (<module\_name>/src/main/resources/zh\_CN/element/)
- 3. 修改 ocr\_custom\_string.json 中需要自定义的文字内容
- 4. 打包以后, 您修改的内容将会覆盖 SDK 原有的提示文本。