蚂蚁金服金融科技产品手册

扫一扫

产品版本： V20200101

文档版本： V20200101

蚂蚁金服金融科技文档

商标声明

及其他蚂蚁金服金融科技服务相关的商标均为蚂蚁金服金融科技所有。
本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁金服金融科技保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁金服金融科技授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁金服金融科技授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

目录

1 扫一扫简介

mPaaS 提供的扫一扫组件能快速扫描识别条码，得到相关信息。

扫一扫组件支持以下的码类型：

- 二维条形码（二维码）
- 一维条形码（条码）

二维条形码（二维码）

Gen0（普通二维码）：

Gen1（视觉码，将 Gen0 码与图片叠加）：

Gen2（在 Gen1 码基础上优化点阵的离散化，同时带隐藏码）：

Gen3（ visualead 自定义码 ）:

一维条形码（条码）

EAN8:

EAN13:

EAN14:

EAN18:

EAN128:

ISBN:

ISSN:

Code39:

Code128:

UPC-A:

UPC-E:

ITF-14:

2 产品优势

支付宝客户端中的扫一扫，在同等条件下，比起业界领先的同类产品，在码的识别速度、识别率等能力上占有优势。

识别速度

在同等距离同等光源的情况下，支付宝扫码的识别速度要高于同类产品的。

请复制以下地址在网页中访问视频:
http://idoc.alipay.net/image/upload_f59e1e3fcae88f9238bd473b0eba653a.mp4

模糊处理和数据评估矫正

下面列举一些业界领先的同类产品的相册识别不了，但支付宝客户端扫一扫能识别的图片。

- 这个图片同类产品摄像头可以识别，但是相册不能识别

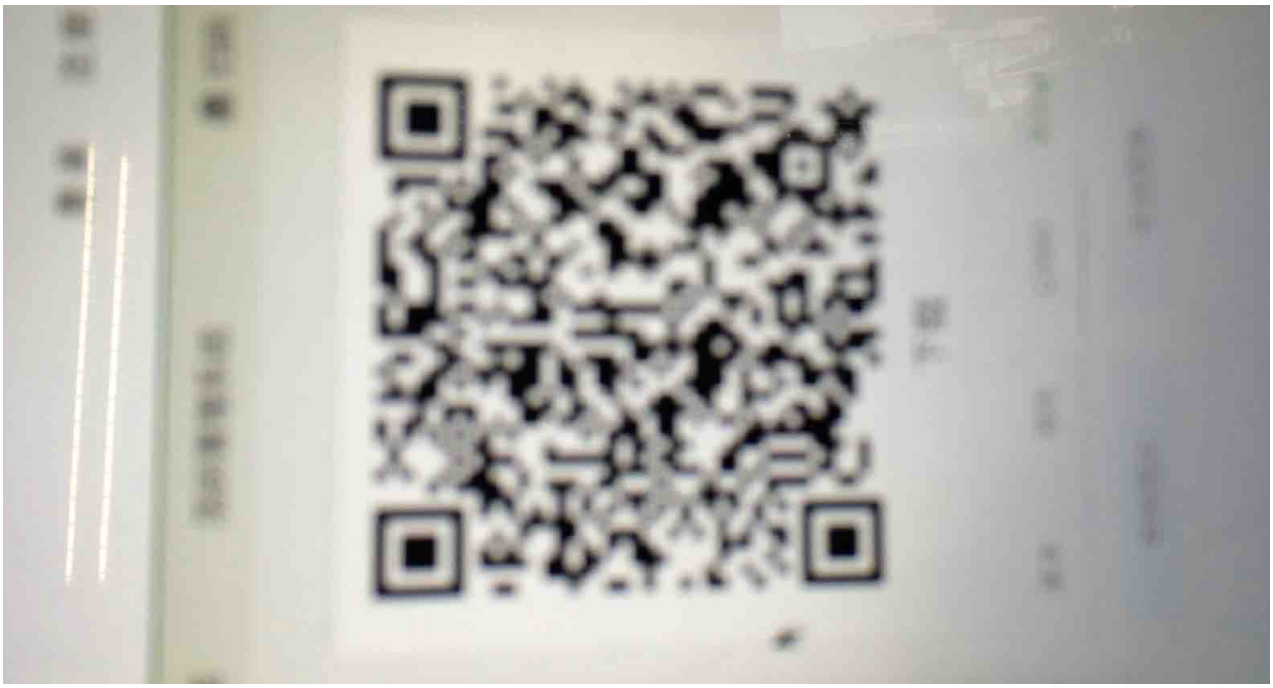
- 同类产品完全不能识别的码


用支付宝扫二维码，加我好友

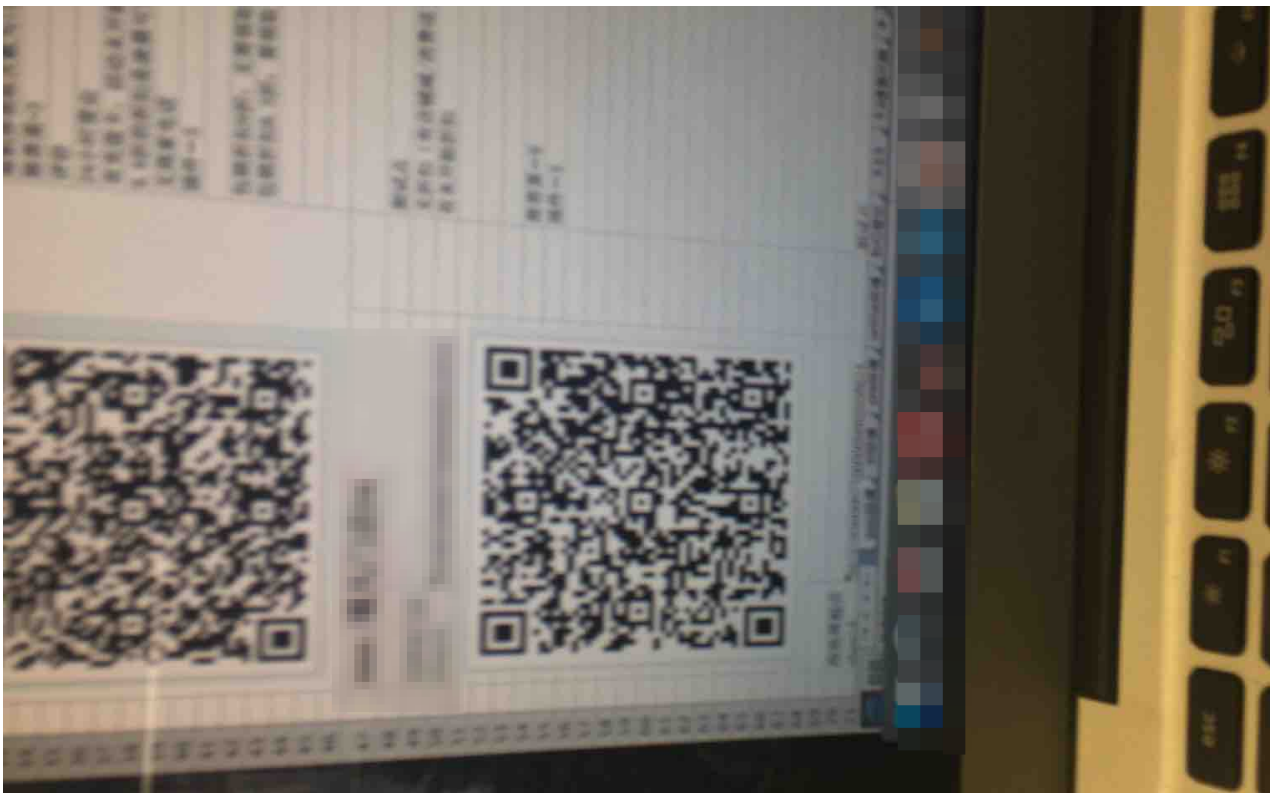

支付宝


手机淘宝扫一扫
新年大礼送到家


用支付宝扫二维码，加入送群




可付款哦！




支付宝
口碑

3 添加 Android SDK

扫一扫 SDK 是支付宝目前正在使用的识别二维码、条形码等功能的 SDK 。

前置条件

集成 mPaaS 提供的扫一扫功能，需满足以下前置条件：

- 已在 mPaaS 后台中创建了一个 App，参见 创建应用 。
- 已在创建的应用中，从 **代码配置** > **下载配置** 下载了配置，参见 快速开始 。

工程配置

AndroidManifest 配置

在 manifest 中配置 SDK 会用到的权限：

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

基于 mPaaS Android 框架的工程
添加依赖

使用 mPaaS 插件，分别在 Portal 和 Bundle 工程中添加 **扫码（SCAN）** 组件依赖。更多信息，请参考 管理组件依赖 > 增删组件依赖 。

使用扫一扫功能

调用标准 UI 的扫码服务：

```
ScanService service = LauncherApplicationAgent
.getInstance().getMicroApplicationContext()
.findServiceByInterface(ScanService.class.getName());

ScanRequest scanRequest = new ScanRequest();
scanRequest.setScanType(ScanRequest.ScanType.QRCODE);
// 设置扫码界面 title
scanRequest.setTitleText("标准扫码");
// 设置扫码窗口下提示文字
scanRequest.setViewText("提示文字");
// 设置打开手电筒提示文字，仅 10.1.60 及以上基线支持
scanRequest.setOpenTorchText("打开手电筒");
// 设置关闭手电筒提示文字，仅 10.1.60 及以上基线支持
scanRequest.setCloseTorchText("关闭手电筒");

service.scan(this, scanRequest, new ScanCallback() {
@Override
public void onScanResult(boolean isProcessed, final Intent result) {
if (!isProcessed) {
// 扫码界面点击物理返回键或左上角返回键
return;
}
// 注意：本回调是在子线程中执行
runOnUiThread(new Runnable() {
@Override
public void run() {
if (result == null || result.getData() == null) {
// 扫码失败
return;
}
// 扫码成功
String url = result.getData().toString();
}
});
}
});
```

自定义 UI 扫码请参考 代码示例 ，升级 SDK 版本时您可能需要适配以下变更：

- 10.1.60 版本：BQCScanCallback 类新增部分接口，您只需空实现这些接口即可。
- 10.1.20 版本：MaScanCallback 类接口变更如下：
  void onResultMa(MaScanResult maScanResult) 变更为 void onResultMa(MultiMaScanResult multiMaScanResult)
  您可以按照以下方式获取 MaScanResult：

```
MaScanResult maScanResult = multiMaScanResult.maScanResults[0];
```

原生工程

参考 代码示例 。

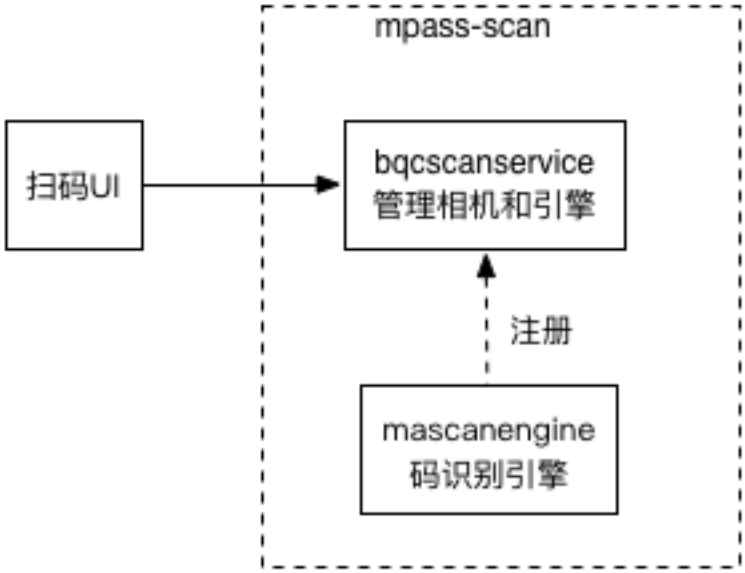
在 build.gradle 中配置扫一扫需要的依赖：

```
// 扫码必须的依赖
compile 'com.alipay.android.phone.mobilesdk:logging:2.0.0'
compile 'com.alipay.android.phone.scancode:fakeframework:2.0.1@aar'
compile 'com.alipay.android.phone.scancode:mascanengine-api:2.0.1@aar'
compile 'com.alipay.android.phone.scancode:bqcscanservice-api:2.0.1@aar'
```

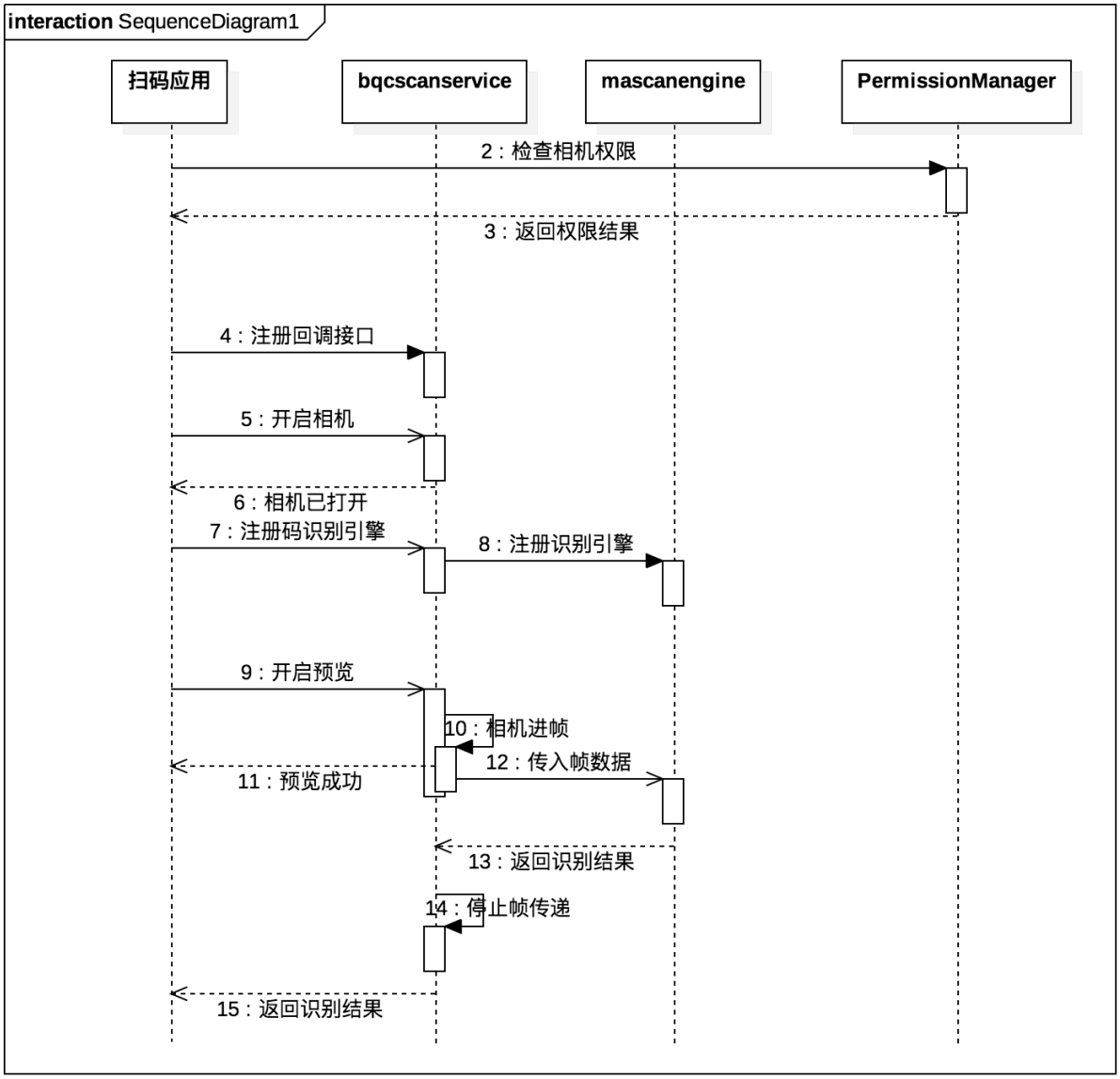
原理

下图为扫一扫 SDK 的基本结构。扫一扫的主要由 bqcscanservice（主要用于管理相机和识别引擎）和 mascanengine（码识别引擎）组成。

说明：mascanengine 的注册过程由扫码应用根据 bqcscanservice 对外暴露的注册接口实现。



下图为扫码应用的顺序图，下文会详细说明扫码应用的执行过程。



API 说明

初始化

```
LoggerFactory.init(getApplicationContext());
```

仅原生工程需要调用初始化方法。

通用 API

自定义 ui 时可根据需要参考以下 API：

说明：bqcscanservice 服务接口主要提供打开相机、进行预览、关闭预览、关闭相机、开关闪光灯、开关识别引擎等操作。

```
public abstract class BQCScanService extends ExternalService {
/** 初始化 */
public abstract void setup(Context ctx, BQCScanCallback callback);

/** 设置预览显示对象 */
public abstract void setDisplay(TextureView view);

/** 清理资源 */
public abstract void cleanup(long postcode);

/** 注册识别引擎 */
public abstract void regScanEngine(String type, Class<? extends BQCScanEngine> engine,
BQCScanEngine.EngineCallback engineCallback);

/** 设置Scan类型 */
public abstract boolean setScanType(String type);

/** 开始预览 */
public abstract void startPreview();

/** 停止预览 */
public abstract void stopPreview();

/** 是否正在预览 */
public abstract boolean isPreviewing();

/** 设置是否扫描 */
public abstract void setScanEnable(boolean enable);

/** 获取扫描状态 */
public abstract boolean isScanEnable();

/** 设置预览图像上的识别区域(目前只对扫码有效) */
public abstract void setScanRegion(Rect region);

/** 设置闪光灯 */
public abstract void setTorch(boolean on);

/** 闪光灯是否打开 */
public abstract boolean isTorchOn();

/** 获取当前zoom参数 */
public abstract int getCurrentZoom();

/** 设置当前zoom参数 */
public abstract void setZoom(int zoom);

/** 获取最大zoom参数 */
public abstract int getMaxZoom();

/** 设置相机参数 */
public abstract void setCameraParam(String key, Object value);

/** 获取相机参数 */
public abstract Object getCameraParam(String key);
```

```
/** 设置Camera id，可用来切换相机 */
public abstract void setCameraId(int id);

/** 获取预览对应的Camera */
public abstract Camera getCamera();

/** 当预览texture可用时 */
public abstract void onSurfaceAvailable();

/** 设置引擎参数 */
public abstract void setEngineParameters(Map<String, Object> parameters);

/** 获取预览旋转角度 */
public abstract int getCameraDisplayOrientation();
}
```

bqcscanservice 状态回调接口 BQCScanCallback。

```
public interface BQCScanCallback {

/** 相机参数设置后回调 */
void onParametersSetted(long postcode);

/** textureview准备好后回调 */
void onSurfaceAvaliable();

/** 相机开启后回调*/
void onCameraOpened();

/**预览画面开始后回调 */
void onPreviewFrameShow();

/** 出错时回调 */
void onError(BQCScanError error);

/** 相机自动对焦后回调 */
void onCameraAutoFocus(boolean success);

/** 环境监测后回调，已废弃 */
void onOuterEnvDetected(boolean shouldShow);
}
```

mascanengine 识别引擎结果回调 MaScanCallback。

```
public interface MaScanCallback extends BQCScanEngine.EngineCallback{
/**
* 扫码检测结果
*
* @param result
*/
void onResultMa(MaScanResult result);
}
```

4 添加 iOS SDK

扫一扫 SDK 是支付宝目前正在使用的识别二维码、条形码等功能的 SDK 。

前置条件

为了集成 mPaaS 提供的扫一扫功能，您需要先执行以下步骤完成基础配置：

- 安装开发者工具
- 在控制台创建应用
- 创建基于 mPaaS 框架的新工程

添加 SDK

使用 mPaaS 插件，添加 **扫码** 模块到工程中，操作方法参见 mPaaS 插件 > 编辑模块 。

使用 SDK

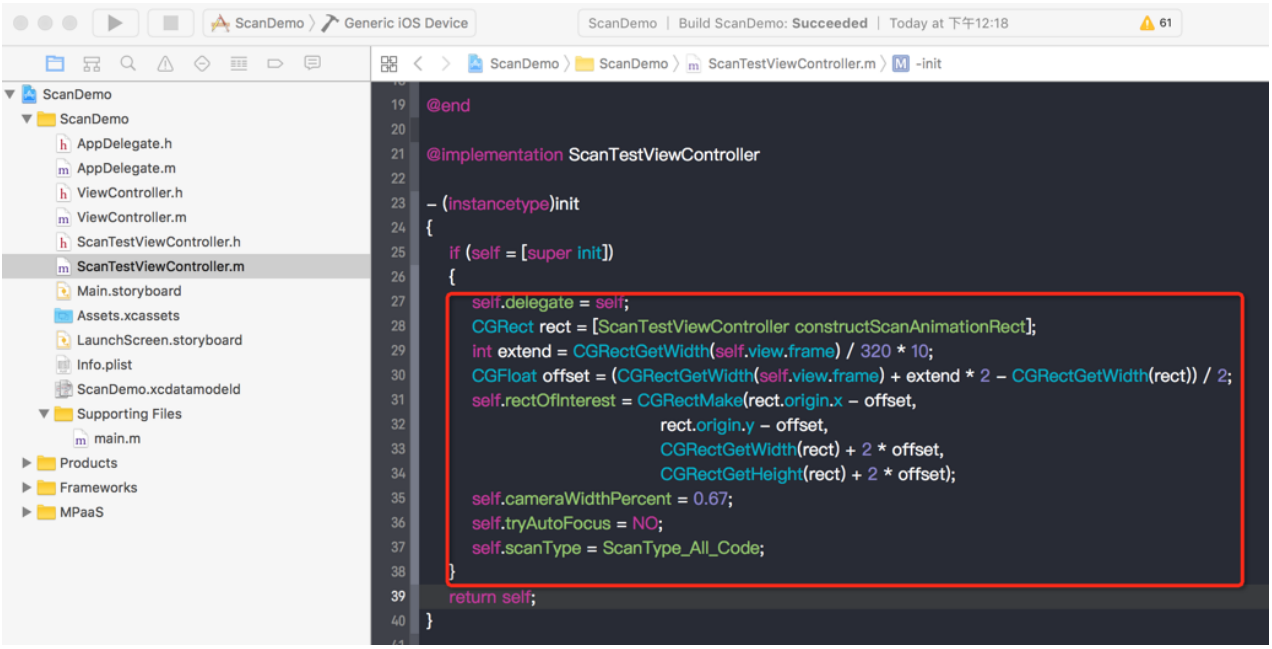
配置工程

在当前工程的 Build Phases > Link Binary With Libraries 中添加系统库：ImageIO.framework、AssetsLibrary.framework、AVFoundation.framework 和 libz.tbd。

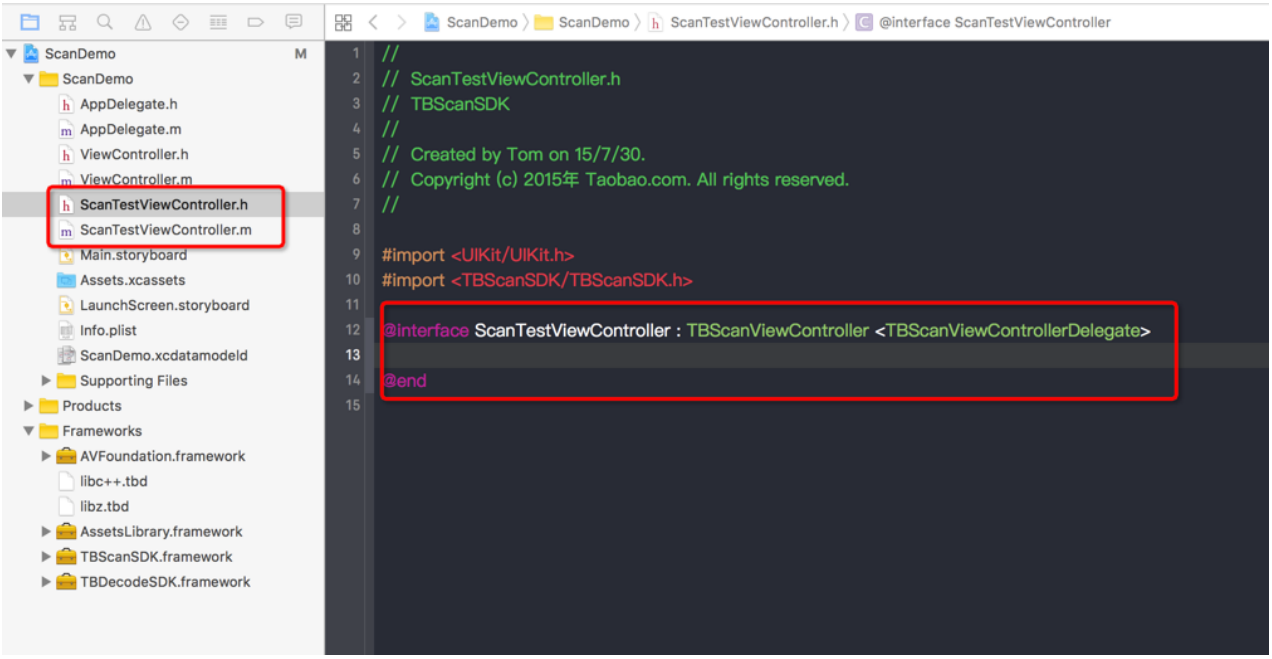
代码示例

参考 代码示例 ，获取具体代码。

设置扫码界面

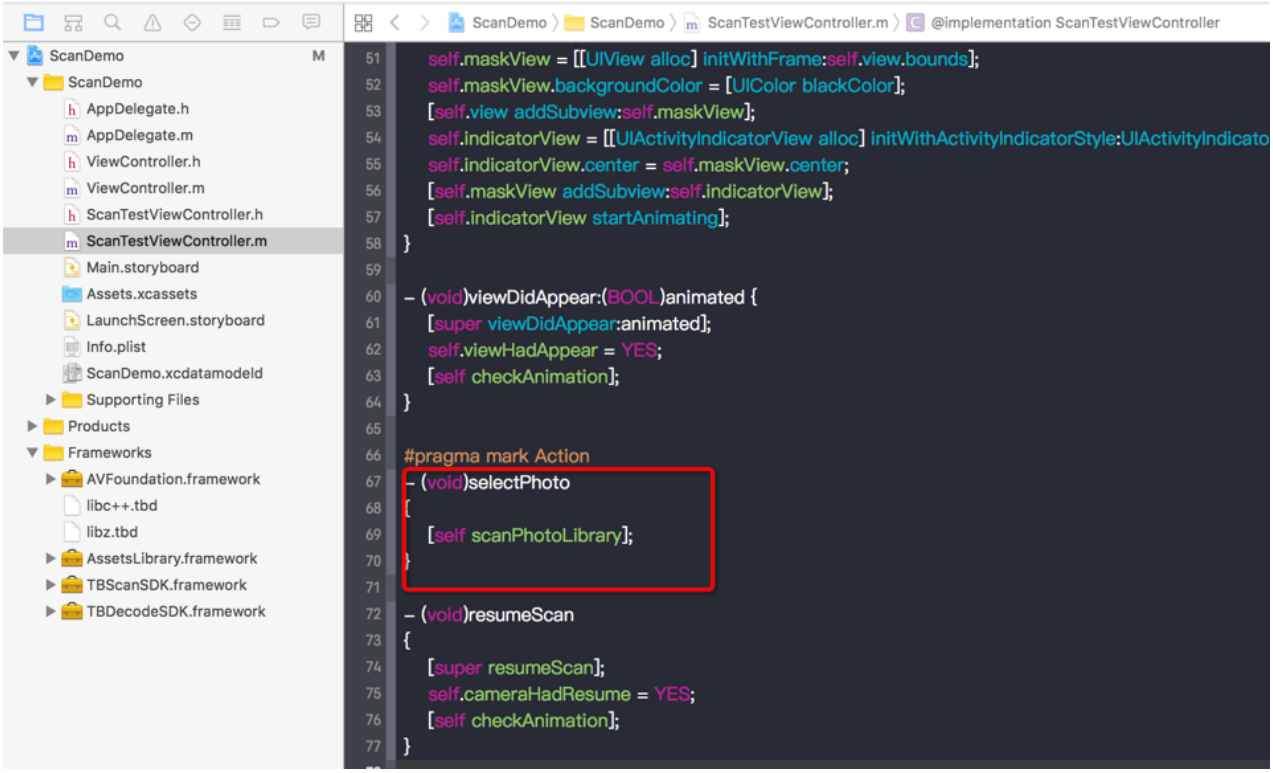


唤起扫码界面



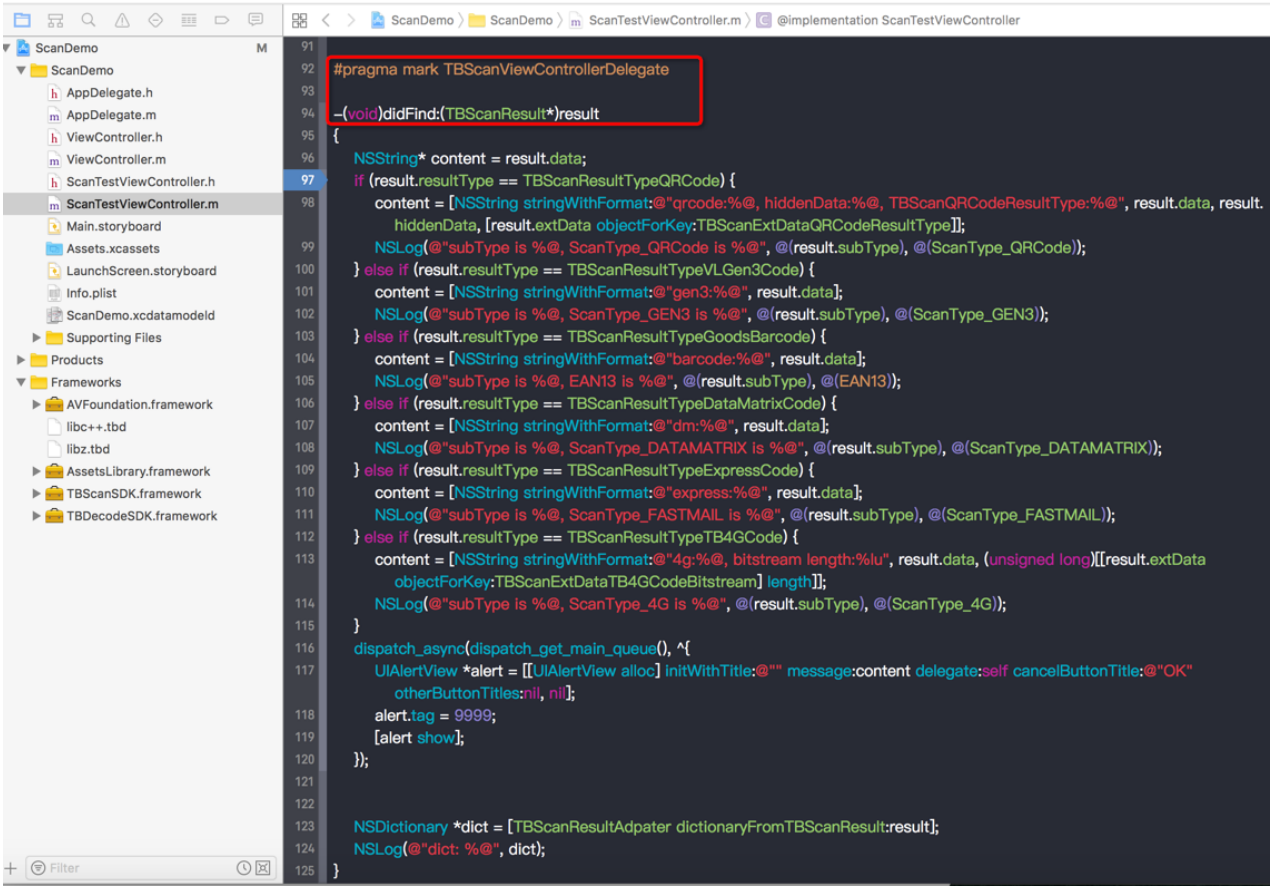
扫描本地相册

扫一扫 功能支持识别本地相册中的图片。



处理扫描结果

通过实现 TBScanViewControllerDelegate 代理中的方法，对扫描结果进行处理。



接口方法

```
@interface TBScanViewController : UIViewController <UIAlertViewDelegate>

- (instancetype)init;
- (instancetype)initWithAnimationRect:(CGRect)animationRect
delegate:(id<TBScanViewControllerDelegate>)delegate;

//停止扫描并关闭摄像头，一般情况下不要直接调用该方法。
-(void)exitScan;

//暂停扫码。
-(void)pauseScan;

//继续扫码。
-(void)resumeScan;

//暂停扫码，并停止相机(预览停止)。
-(void)pauseCaptureSession;

//继续扫码，并恢复相机。
-(void)resumeCaptureSession;

//扫描本地图库，会调起本地相册。
-(void)scanPhotoLibrary;

//从相册中选择图片的流程。
-(void)scanPhotoImage:(UIImage*)image;

//点击闪光灯，之后会调用代理接口 -(void)setTorchState:(TorchState)bState 来返回闪光灯状态。在该代理接口中根据参数 bState，来判断当前闪光灯的状态，从而实现对应的 UI。
-(void)onTorch;

//返回闪光灯的状态。
-(TorchState)torchMode;

@property (nonatomic,weak) id<TBScanViewControllerDelegate> delegate;
//动画框大小。
@property (nonatomic,assign) CGRect animationRect;
/**
*
* @discussion rectOfInterest 设置底层解码时的裁剪区域，若不设置，SDK 会默认设置一个裁剪区域。
*
*/
@property (nonatomic,assign) CGRect rectOfInterest;
/**
*
* @discussion 支持本次识别的码类型，默认为 ScanType_All_Code。scanType 是一个整型，用该值来确定要支持的扫码类型，可以用或操作符来支持多个类型，详细见 ScanType 定义。
* 推荐使用最小类型，扫码类型越小识别速度越快，比如只支持条形码则使用 ScanType_Barcode，只支持 QR 码则使用 ScanType_QRCode，只支持条形码和 QR 码则使用 ScanType_Barcode|ScanType_QRCode。请谨慎使用 ScanType_All_Code 类型，该类型中包含一些不常用的扫码类型。
*
*/
@property (nonatomic,assign) ScanType scanType;
```

```
/**
*
* @discussion cameraWidthPercent 设置屏幕中显示摄像头内容的百分比，设值有效范围大约为 0.4~1.0，默认为 0.74。
*
*/
@property (nonatomic,assign) CGFloat cameraWidthPercent;
/**
*
* @discussion bPlaySound 设置识别成功后是否播放声音，YES 为播放，NO 为不播放，默认为 YES。
*
*/
@property (nonatomic,assign) BOOL bPlaySound;

//是否支持自动根据屏幕内疑似二维码的 size 做放大缩小，默认为 YES。
@property (nonatomic,assign,setter=setBAutoZoomEnable:) BOOL bAutoZoomEnable;

//是否支持手势缩放，默认为 YES。
@property (nonatomic,assign) BOOL bGestureEnable;

//提示打开闪光灯的阈值，有默认值。
@property (nonatomic,assign) CGFloat flashOnBrightnessThreshold;

//提示关闭闪光灯的阈值，有默认值。
@property (nonatomic,assign) CGFloat flashOffBrightnessThreshold;

////////////////////////////自定义UIAlertView的文案//////////////////////////////////
// 无摄像头权限时的 AlertView 的 title。
@property (nonatomic,strong) NSString *cameraPermissionDeniedTitle;

// 无摄像头权限时的 AlertView 的提示信息。
@property (nonatomic,strong) NSString *cameraPermissionDeniedMsg;

// 无摄像头权限时的 AlertView 取消按钮的 title。
@property (nonatomic,strong) NSString *cameraPermissionDeniedCancelTitle;

// 相册识别失败时的 AlertView 的提示信息。
@property (nonatomic,strong) NSString *scanPhotoFailedMsg;

// 相册识别失败时 AlertView 取消按钮的 title。
@property (nonatomic,strong) NSString *scanPhotoFailedCancelTitle;
@end

@interface TBScanViewController(pad)
/**
*
* @discussion 方法仅对 iPad 有效。对 iPad 相册取图弹出框有定制要求的可以使用该接口来替换通用的相册取图 scanPhotoLibrary 接口。
*
* @param rect 指定相册取图弹出框的区域。
*
* @param arrowDirections 指定相册取图弹出框的箭头方向。
*
* @param arrowColor 指定相册取图弹出框的背景色，nil 为白色。
*
*/
-(void)scanPhotoLibrary:(CGRect)rect permittedArrowDirections:(UIPopoverArrowDirection)arrowDirections
```

```
with:(UIColor*)arrowColor;
@end

@interface TBScanViewController(plugin)
//针对预留 ScanType 可以注册自定义的 Plugin。
-(void)registerPlugin:(TBScanPlugin*)inPlugin withType:(ScanType)type;
@end

@interface TBScanViewController(CaptureManager)
/**
*
* @discussion 判断摄像头是否支持该 SessionPreset。
*
*/
-(BOOL)canSetSession:(NSString *)sessionPreset;

// 判断当前是否正在使用前置摄像头。
-(BOOL)isUsingFrontCamera;
@end
```

回调方法

```
@protocol TBScanViewControllerDelegate <NSObject>
@required

//识别出来的码值，可能多码。相机识别的码值分线程返回，相册识别的码值在主线程中返回。
-(void)didFind:(NSArray<TBScanResult*>*)resultArray;

@optional
//摄像头没有访问权限的回调，当摄像头没有访问权限时会调用该方法。
-(void)cameraPermissionDenied;
/**
*
* @discussion 码值识别过滤。扫描成功后回调 didFind: 之前接入方可以通过这个接口决定是否处理此次扫描结果。
*
* @param result 扫描成功的结果，具体参见 TBScanResult 类。
*
* @return 返回 YES 会暂停扫描然后回调 didFind；返回 NO 则忽略扫描识别的码值，继续扫描。
*
*/
-(BOOL)canHandleScanResult:(TBScanResult*)result;

//摄像头启动后的回调。
-(void)cameraDidStart;

//摄像头启动失败后的回调。
-(void)cameraStartFail;

//由于收到内存警告而释放摄像头的回调。
-(void)releaseByMemoryWarning;
/**
*
* @discussion 闪光灯初始化、状态变更都会调用这个方法，可以在这里实现闪光灯的 UI。
*
```

```
* @param bState 闪光灯当前的状态。
*
*/
-(void)setTorchState:(TorchState)bState;

////////////////////////////////埋点接口////////////////////////////////
//用接入方自己的埋点接口实现。
-(void)userTrack:(NSString*)name;

//用接入方的埋点接口实现，可以上传数据。
-(void)userTrack:(NSString*)name args:(NSDictionary*)data;

//用于研究主线程切换耗时。
-(void)userAlipayLog:(NSString*)name;

////////////////////////////////iPad相关回调////////////////////////////
//iPad 识别本地相册图片失败时会调用该方法。
-(void)scanPadPhotoFailed。

//popover 消失通知。
-(void)dismissPopover;

////////////////////////////////定制////////////////////////////
//自定义界面，添加在现有界面之上。
-(void)buildContainerView:(UIView*)containerView;

//自定义相册，后续流程由外部负责。
-(void)buildCustomAlbum;

//自定义相机权限框。
-(void)buildCustomCameraPermissionAlert;
@end
```

5 iOS 代码示例

说明：本 Demo 适用于 iOS 客户端。

前置条件

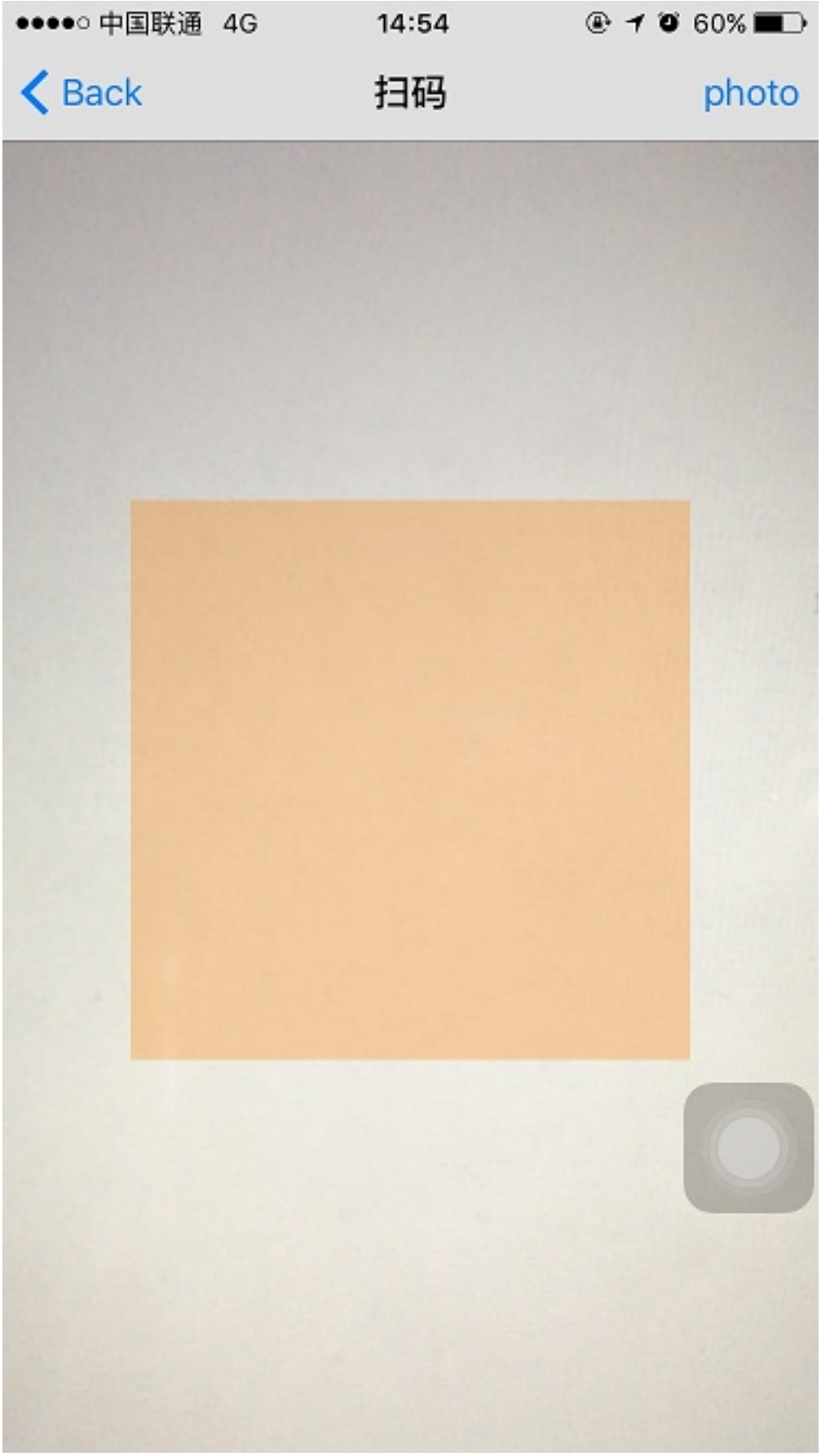
使用本 Demo 前，确保您的应用客户端已接入 mPaaS 扫码功能。客户端集成扫码功能的详细介绍请参考 扫码 > iOS 客户端编程 。

下载代码

点击这里 ，找到所需的代码示例并下载到本地。

快速开始

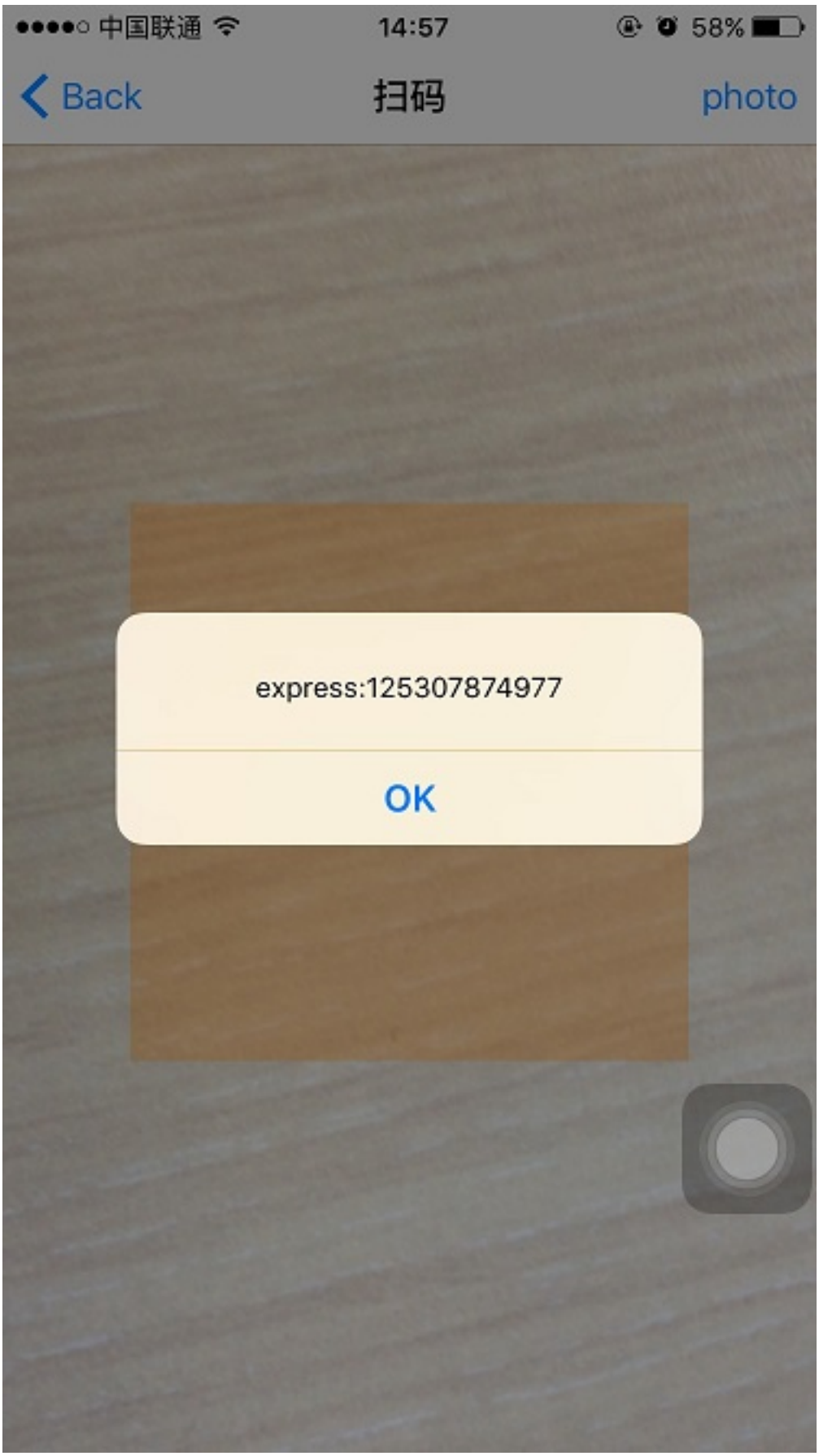
运行程序，点击 **扫码** 按钮唤起扫码界面。


●●●●○ 中国联通 4G
14:54
60%
Back
扫码
photo

扫描二维码或条形码，得到识别结果，支持的码类型可参考 简介 > 码类型 。


●●●●○ 中国联通 4G
14:55
59%
Back
扫码
photo
目前我们支付宝钱包扫码支持
支持的二维码类型
Gen0（普通二维码）：
qrcode:http://ma.taobao.com,
hiddenData:(null),
TBScanQRCodeResultType:(null)
OK
Gen1（视觉码，将Gen0码与

点击 photo 按钮唤起本地相册，选择相应码进行识别。


●●●●○ 中国联通
14:57
58%
Back
扫码
photo
express:125307874977
OK


蚂蚁金服
ANT FINANCIAL
金融科技
FINANCIAL TECHNOLOGY