

iOS GlobalIQ SDK (v1.0.9) User Guide

Installation

CocoaPods

1. Specify the SDK name and local path in the podfile:

```
pod 'AAIGlobalIQ', :path => '/your-local-path-to/AAIGlobalIQSDK/'
```

2. Install the dependencies in your project:

```
pod install
```

3. Add camera usage description in Info.plist as bellow. Ignore this step if you have added those.

```
<key>NSCameraUsageDescription</key>  
<string>Use the camera to detect the card</string>
```

4. Refer to demo project to see how to use the SDK.

Manually

1. Add `AAIGlobalIQSDK.xcframework` to your project and set embed as "Embed & Sign".
2. Choose "TARGETS -> General" add the following system libraries and frameworks in the `Frameworks, Libraries, and Embedded Content` section:

- o `libc++.tbd`
- o `libz.tbd`
- o `AVFoundation.framework`
- o `CoreGraphics.framework`
- o `MediaPlayer.framework`
- o `SystemConfiguration.framework`
- o `Accelerate.framework`

3. Add camera usage description in `Info.plist` as bellow. Ignore this step if you have added those.

```
<key>NSCameraUsageDescription</key>  
<string>Use the camera to detect the card</string>
```

4. Import SDK and initialize with `regin` and `cardType` and `cardSide`:

```

@import AAIGlobalIQASDK;

AAIGlobalIQAConfig *config = [AAIGlobalIQAConfig initWithRegion:@"ID"
cardType:AAIIQACardTypeIDCard cardSide:AAIIQACardSideFront];
config.delegate = self;
[AAIGlobalIQASDK initWithConfig:config];

```

5. Configure SDK license and show SDK page:

```

NSString *result = [AAIGlobalIQASDK setLicenseAndCheck:@"your-license-content"];
if ([result isEqualToString:@"SUCCESS"]) {
    [AAIGlobalIQASDK startWithRootVC:self];
} else {
    NSLog(@"%@", result);
}

```

6. Get detection results.

```

- (void)iqaOnDetectionComplete:(AAIGlobalIQAResult *)result
{
    if (result.success) {
        UIImage *cardImg = result.cardImg;
        NSString *idvid = result.IDVID;
        NSString *transactionId = result.transactionId;
        NSString *pictureType = result.pictureType;

        if (result.ocrResult) {
            NSDictionary *ocrResult = result.ocrResult;
        }

        if (result.idForgeryResult) {
            NSDictionary *idForgeryResult = result.idForgeryResult;
        }
    } else {
        NSString *errorCode = result.errorCode;
        if (result.errorMsg) {
            NSString *errorMsg = result.errorMsg;
        }

        if (result.transactionId) {
            NSString *transactionId = result.transactionId;
        }
    }
}

```

Run demo project

1. Navigate to the directory of `AAIGlobalIQASDKObjCDemo` or `AAIGlobalIQASDKSwiftDemo` project and install the dependencies:

```
pod install
```

2. Open xcworkspace file in Xcode.
3. Specify your license, and configure `region` and `cardType` and `cardSide`:
4. Run.

Customizable UI

1. UI Customization.

Currently, the SDK only supports limited UI customization:

```
AAIGlobalIQAConfig *config = [AAIGlobalIQAConfig initWithRegion:@"ID"
cardType:AAIIQACardTypeIDCard cardSide:AAIIQACardSideFront];
// e.g.
config.uiConfig.flipCameraBtnVisible = NO;
config.uiConfig.lightBtnVisible = NO;

// Whether the show the timer label in the upper right corner of the view. Default
is YES.
config.uiConfig.timerLabelVisible = NO;

config.uiConfig.navBarTitleTextColor = UIColor.redColor;
config.uiConfig.navBarBackgroundColor = UIColor.whiteColor;
config.uiConfig.statusBarStyle = UIStatusBarStyleDefault;
config.uiConfig.tipIconVisible = NO;
config.uiConfig.pageBackgroundColor = UIColor.redColor;
config.uiConfig.viewfinderColor = UIColor.greenColor;
config.uiConfig.primaryTextColor = UIColor.yellowColor;
config.uiConfig.toolButtonThemeColor = UIColor.blueColor;
config.uiConfig.orientation = AAIQAOrientationPortrait;
config.uiConfig.takePhotoTipTextColor = UIColor.whiteColor;
config.uiConfig.takePhotoTipBackgroundColor = UIColor.blueColor;
config.uiConfig.previewPhotoTipBackgroundColor = UIColor.blueColor;
config.uiConfig.previewPhotoTipTextColor = UIColor.whiteColor;
config.detectionTimeoutInterval = 20;
config.returnEmptyOfOCR = NO;
config.languageLprojName = @"id";

// Set the left and right margin on x-axis of cameraView when device in portrait
mode.
config.uiConfig.cameraViewMarginLRInPortraitMode = @(12);
```

```

// The duration of the alert view displayed when SDK are about to enter the photo
mode,
// the default is 3s, if set to 0, the alert view will not be displayed.
config.takePhotoAlertViewTimeoutInterval = 0;

// If you want to add custom UI on the SDK page, just register the custom class to
the SDK.
/*
// You can customize the scan page UI by inheriting `AAIIQAScanController` class,
then register your custom class to the sdk. See demo project for more detail.
[config registerClass:[CustomScanController class]
forModule:AAIIQAModuleTypeScanController];
*/

/*
// You can customize the preview the photos taken page UI by inheriting
`AAIIQAPreviewImgController` class,
// then register your custom class to the sdk. See demo project for more detail.
[config registerClass:[CustomPreviewImgController class]
forModule:AAIIQAModuleTypePreviewImgController];
*/

/*
// You can customize the camera overlay view UI of `AAIIQAScanController` page by
inheriting `AAIIQAOverlayView` class,
// then register your custom class to the sdk. See demo project for more detail.
[config registerClass:[CustomOverlayView class]
forModule:AAIIQAModuleTypeOverlayViewOfScanVC];
*/

/*
// You can customize the camera overlay view UI of `AAIIQAPreviewImgController`
page by inheriting `AAIIQAOverlayView` class,
// then register your custom class to the sdk. See demo project for more detail.
[config registerClass:[CustomOverlayView class]
forModule:AAIIQAModuleTypeOverlayViewOfPreviewImgVC];
*/

/* imageName list:
  advance_iqa_scan: An image to show the scan animation.
  advance_iqa_tip_capture: The icon in the countdown pop-up view after the scan
times out.
  iqa_back: Icon of navigation bar back button.
  iqa_land_confirm: Confirm button icon in landscape take photo mode.
  iqa_land_retake: Retake button icon in landscape take photo mode.
  iqa_light_off: Icon of flashlight off button.
  iqa_light_on: Icon of flashlight on button.
  iqa_scan_processing: The icon displayed at the bottom of the vertical screen
(local string key is: "hold_phone").

```

```

    iqa_scan_successfully: The icon displayed at the bottom of the vertical screen
(local string key is: "iqa_scan_successfully").
    iqa_scan_warning: The icon displayed at the bottom of the vertical screen (local
string key is: "iqa_card_poor_quality").
    iqa_take_photo_tip: The tip icon displayed at the bottom of the viewfinder view
(local string key is: "iqa_take_photo_tips").
    iqa_take_photo: Icon of take photo button.
    iqa_transform_camera: Icon of switch camera button.
*/
config.uiConfig.loadImage = ^UIImage * _Nonnull(NSString * _Nonnull imgName,
UIImage * _Nonnull img) {
    if ([imgName isEqualToString:@"iqa_scan_successfully"]) {
        return [UIImage imageNamed:@"ok"];
    }
    return img;
};

/*
For all available keys, see
"AAIGlobalIQASDK.framework/AIIQALanguageString.bundle/en.lproj/Localizable.strings
"
*/
config.uiConfig.loadLocalizedString = ^NSString * _Nonnull(NSString * _Nonnull key,
NSString * _Nonnull value, NSString * _Nonnull language) {
    if ([key isEqualToString:@"iqa_top_desc"]) {
        if ([language isEqualToString:@"en"]) {
            return @"My test string";
        }
    }

    return value;
};

```

2. Resources File Customization.

Currently, this SDK supports English (en), Simplified Chinese (zh-Hans), Indonesian (id).

The SDK will use the corresponding resource file according to the current language of the device. If this device language is not in the above languages, the SDK will use English by default.

If you want to support only a fixed language, You can delete other irrelevant `.lproj` folders of `AAIIQALanguageString.bundle` and keep only the `.lproj` folder you need.

The resource files corresponding to these languages are in the following bundles:

Name	Description	Customizable description
AAIQALanguageString	Muti-language bundle	Can be modified directly
AAIQAAudio.bundle	Audio bundle	Use your audio file to replace it. NOTE: the audio file name cannot be changed

3. Error code list.

Name	Description
USER_GIVE_UP	User tapped the back button
DEVICE_NOT_SUPPORT	This device is not supported
CAMERA_PERMISSION_DENIED	Permission to access the camera is not authorized
NETWORK_REQUEST_FAILED	Network request failed
CAMERA_OPEN_FAILED	Failed to open camera
MODEL_ERROR	Load model failed
Other error code	See document

Release Logs

v1.0.1 (2022.2.17)

1. Support customize icon and navigationbar background color.

v1.0.2 (2022.3.18)

1. Minor update.

v1.0.3 (2022.3.21)

1. Support loading specified language lproj file:

```
// Specifies the language lproj file to be loaded by the SDK. e.g. "en" or "id" or "zh-Hans".
config.languageLprojName = @"id";
```

v1.0.4 (2022.4.19)

1. Support passport card type.

v1.0.5 (2022.6.6)

1. Set the result to always succeed in take photo mode.

v1.0.6 (2022.6.7)

1. Add `pictureType` property for `AAIGlobalIQAResult`.

v1.0.7 (2022.6.27)

1. Adjust the UI.
2. Support customize the background color and text color of the bottom tip view in take photo mode or preview photo mode.

```
// e.g.
config.uiConfig.takePhotoTipTextColor = UIColor.whiteColor;
config.uiConfig.takePhotoTipBackgroundColor = UIColor.blueColor;
config.uiConfig.previewPhotoTipBackgroundColor = UIColor.blueColor;
config.uiConfig.previewPhotoTipTextColor = UIColor.whiteColor;
```

3. Support customize the localized string.

```
// e.g.
config.uiConfig.loadLocalizedString = ^NSString * _Nonnull(NSString * _Nonnull key,
NSString * _Nonnull value, NSString * _Nonnull language) {
    if ([key isEqualToString:@"iqa_top_desc"]) {
        if ([language isEqualToString:@"en"]) {
            return @"My test string";
        }
    }

    return value;
};
```

v1.0.8 (2022.7.5)

1. Adjust the UI.
2. Support hide the timeLabel of scan mode.

```
// e.g.
config.uiConfig.timerLabelVisible = NO;
```

3. Support adjust the margin left and right of cameraView when device in portrait mode.

```
// e.g.
config.uiConfig.cameraViewMarginLRInPortraitMode = @(12);
```

4. Support customize the duration of the alert view displayed when SDK are about to enter the photo mode.

```
// e.g.  
config.takePhotoAlertViewTimeoutInterval = 0;
```

5. Support adding your custom UI by subclassing specified classes. See demo project for more detail.

v1.0.9 (2022.8.5)

1. Minor update.