Truepic Inc.

SDK Overview

The Truepic SDK is not a vanilla API with functions that applications pass information to and directly receive results from. Instead, the Truepic SDK presents a living camera interface that takes over your mobile device screen, controlled by your users while they take verified photos and videos. This means..

Information is passed asynchronously

Most information about the photos and videos taken within the Truepic Camera View is returned to the application asynchronously via the operating system platform's standard notifications system.

The application doesn't directly control the Truepic Camera View

The only direct control the application will have over the Truepic Camera View is choosing when to open it, and setting configuration values (such as types of media it will allow) before it opens. The user will control operation of the camera, including the photos and videos that are taken and when the Truepic Camera View is closed.

Notifications continue even when the Truepic Camera View is closed

Even after the Truepic SDK camera view is closed, it continues to update the application as newly captured photos and videos progress through the various verification states in the Capture Lifecycle.

The Capture Lifecycle

1. **The user takes a photo or video.**
2. **If Camera verification is disabled, mark capture as failing verification.**

   The camera verification state is clearly displayed in the camera view at all times while capturing photos and videos. If it becomes disabled, most often it's caused by the current location being inaccurate or no network connection. But verification will also disable if the application is running on a rooted/jailbroken device, or the SDK detects software that isn't allowed, such as a debugger, or GPS simulation.

3. **If Camera verification is enabled, perform preliminary (metadata) verification.**

   This is performed by sending image metadata from the capture to Truepic servers for analysis. If verification isn't completed within 15 seconds, or fails, the capture is marked as failing verification.

4. **If preliminary verification succeeds, start capture upload.**
5. **If upload fails, retry.**

   The SDK retries uploads as many times as necessary, and without any time limit, to try to ensure all uploads complete. Even so, in rare circumstances uploads can fail permanently.

6. **When upload succeeds, server performs final verification.**

   Once an upload succeeds, all photo/video and final verification information are available from the server API.

Note: Any capture that fails verification or upload is always saved to the user's camera roll.

Integrating the SDK

Integrating the SDK on both the Android and iOS platforms has three basic steps.

1. Contact Truepic to get a Truepic API Key and Application Key for your application.
2. Integrate the Truepic SDK library into your application project
3. Update your project's settings to support Truepic SDK features and requirements.

The SDKs along with platform specific documentation on integrating and implementing them can be found at (note: this is a private repo, accessing it requires a github account, and contacting Truepic to enable access permissions for your account)

https://github.com/TRUEPIC/TruepicSDK

Once you have integrated the SDK into your application project, your next step is to implement the SDK's required feature-set.

Implementing Truepic SDK

TruePic SDK features are implemented on Android and iOS in a similar manner.

1. **Initialize the Truepic SDK**

When your application is first launched, initialize the SDK by passing it your Truepic API Key and Application Key. This is also a good time to set specific display options for the Camera View.

2. Register to receive Truepic SDK messages

SDK messages are sent for key SDK events, such as when the Truepic Camera View has been closed, or when uploads or verifications have failed. To receive SDK message your application will need to register for them, and provide handlers for the event messages your application wants to handle.

3. Optional: Provide UI for previously captured photos and videos

Typically your app will implement a gallery view of recent photos and videos taken by the user. Or you might just want to display recent capture uploads and their current upload status.

For captures, the SDK API keeps a local database of "Gallery Items", storing each captured image/video locally along with detailed status information. This allows the application to display captures (and their status) while they are in the Capture Lifecycle. Whenever a gallery item changes state during it's Capture Lifecycle an update message is sent to the application.

Note that the gallery db is only temporary storage. Once images are verified they are available from the Truepic server, via the SDK fetch call that returns all images for the current account.

4. Open the Truepic Camera View

Your application decides when to open the Camera View. If you want to set any of the available camera options, such as turning off the front facing camera, or opening in video mode, you'll need to set them before opening.

5. Manage local capture storage

The SDK's gallery database keeps local copies of all captures, which over time can use a large amount of disk space. The SDK doesn't know how long your application wants to keep local images, so your application needs to clear the directory whenever appropriate. The SDK API allows you to clear the entire directory, or just delete specific items, such as only photos, only videos, or matching a specific gallery item state.

The gallery can be cleared whenever and as often as necessary. It's even safe to delete a gallery item while it's in the Capture Lifecycle, as verification and uploads are managed separately. Once gallery items successfully complete upload, they will be permanently available from the TruePic server, along with their verification status details.

But we recommend only deleting gallery items that have completed upload because the gallery database is the only way to access items and their images until their uploads completes. Deleting them before completion will also end update messages for their status changes. Finally it means deleting any gallery items failing verification or upload is permanent (though they will still be saved to the device's camera roll).