

R1 Effects SDK Quick Start (ios version)

The R1 Effects SDK allows user's to edit photos using filters, borders, cropping, stickers, and text effects.

Contents

- [Prerequisites](#)
- [Setup](#)
- [Basic Usage](#)

Prerequisites

- XCode 4.6 with iOS 6.0 SDK
- Deployment target of 5.0 or greater

Quick Start

Setup

In order to use the SDK in an existing app, you must do the following:

- **Project setup**
Make sure you're running the latest version of Xcode and Apple's LLVM compiler.
- **Unzip**
Unzip the R1PhotoEffectsSDK.zip to a convient location
- **Drag**
Drag the entire Unzipped folder into the project navigator of your XCode project. Ensure that `Copy Items into destination group's folder (if needed)` is selected
Ensure that `Create groups for any added folder` is selected Ensure that the proper target for you app is selected.
- **Link against libraries**
Check your target's "Link Binary With Libraries" build phase. Make sure your app is being linked against

```
R1PhotoEffectsSDK.a
```

Link against the following libraries and frameworks:

```
Foundation.framework
UIKit.framework
R1PhotoEffectsSDK.a
SystemConfiguration.framework
OpenGL.framework
QuartzCore.framework
AVFoundation.framework
CoreMedia.framework
CoreVideo.framework
CoreGraphics.framework
CoreText.framework
CoreTelephony.framework
MobileCoreServices.framework
StoreKit.framework
AdSupport.framework (optional)
```

- **Add linker flags**
Update your target's (or project's) build settings to include the following "Other Linker Flags:" (under the "Linking" group)

```
-ObjC -all_load
```

- **Import headers**
Include the following line to your project's prefix.pch file

```
#import "R1PhotoEffectsSDK.h"
```

Basic Usage

First, enable the SDK in your Application Delegate's -application:didFinishLaunchingWithOptions: method. Use the key assigned to your app.

```
//Enable the sdk inside your existing didFinishLaunching method
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
{
    [[R1PhotoEffectsSDK sharedManager] enableWithClientID:@"asf232jnj98asfadfdf"];
    //your existing code...
}
}
```

To present the photo editor, create an instance of `R1PhotoEffectsViewController`, set it's delegate, assign a `UIImage` the user would like to edit and present it modally. Implement the 2 required callback methods to handle the user canceling and finishing editing the image.

```
- (IBAction)buttonPressed
{
    //pick an existing UIImage
    UIImage *pickedImage = self.imageView.image

    //push the view controller
    [self dismissViewControllerAnimated:YES completion:nil];
    UIViewController *vc = [[R1PhotoEffectsSDK sharedManager] photoEffectsControllerForImage:self.imageView.image delegate:(id<R1PhotoEffectsEditingV
        [self presentViewController:vc animated:YES completion:nil];
}

- (void)photoEffectsEditingViewController:(R1PhotoEffectsEditingViewController *)controller didFinishWithImage:(UIImage *)image
{
    //Do something with the resulting UIImage
    //For example:
    self.imageView.image = image;
    //Dismiss the editor
    [self dismissViewControllerAnimated:YES completion:nil];
}

- (void)photoEffectsEditingViewControllerDidCancel:(R1PhotoEffectsEditingViewController *)controller
{
    [self dismissViewControllerAnimated:YES completion:nil];
}
```