# Introduction

1. What is Eva?

**Evature** develops an Expert Virtual Agent (EVA), enabling free-text search for online travel.   Utilizing innovative algorithms to process search requests, EVA understands the users' inputs and converts them to structured search queries with very high precision.  EVA significantly improves conversion rates, revenues and user satisfaction.

2. What this document is for?

This document describes how to install and use Eva SDK for iOS.

# Project Setup and Installation

## Download & Install the SDK

1. Download the SDK:

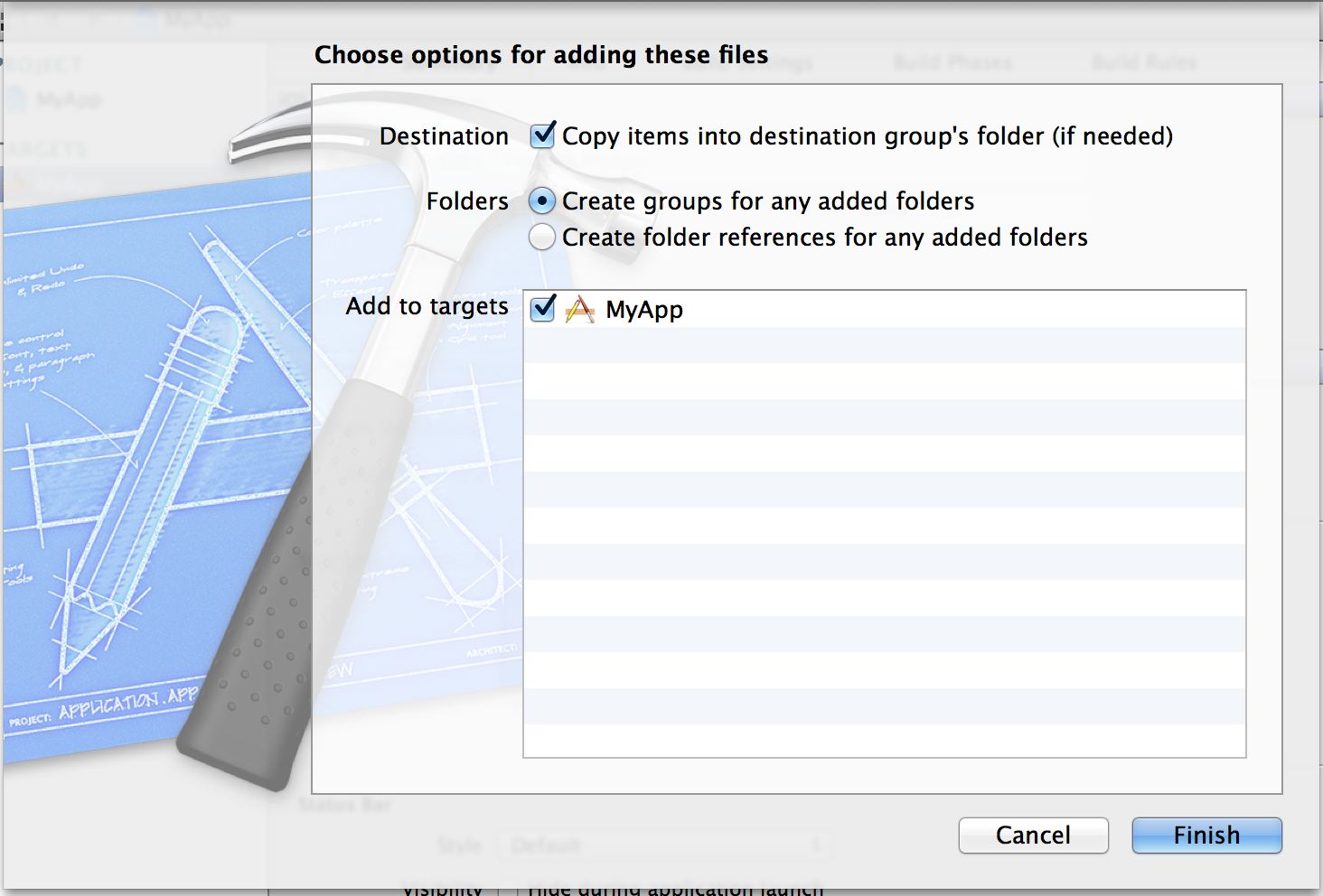
<https://github.com/evature/ios/tree/master/EvaTest/EvaFramework>

Or from the dump file (EvaTest.zip):

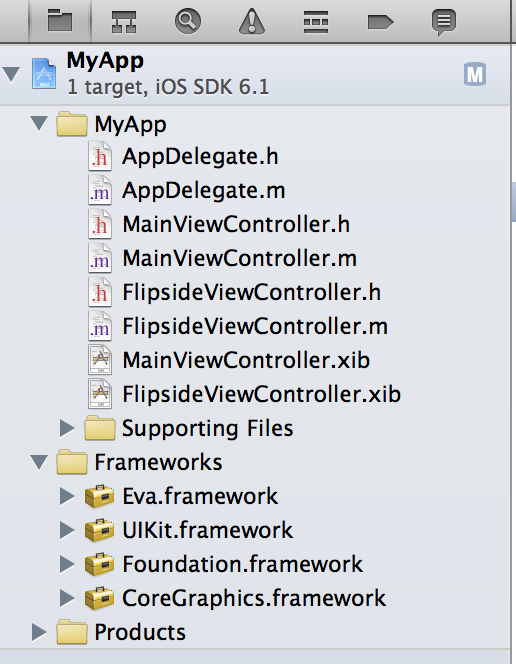
/EvaTest/EvaFramework/

2. Make sure you are using the latest version of xCode (4.6+) and targeting iOS5.0 or higher.

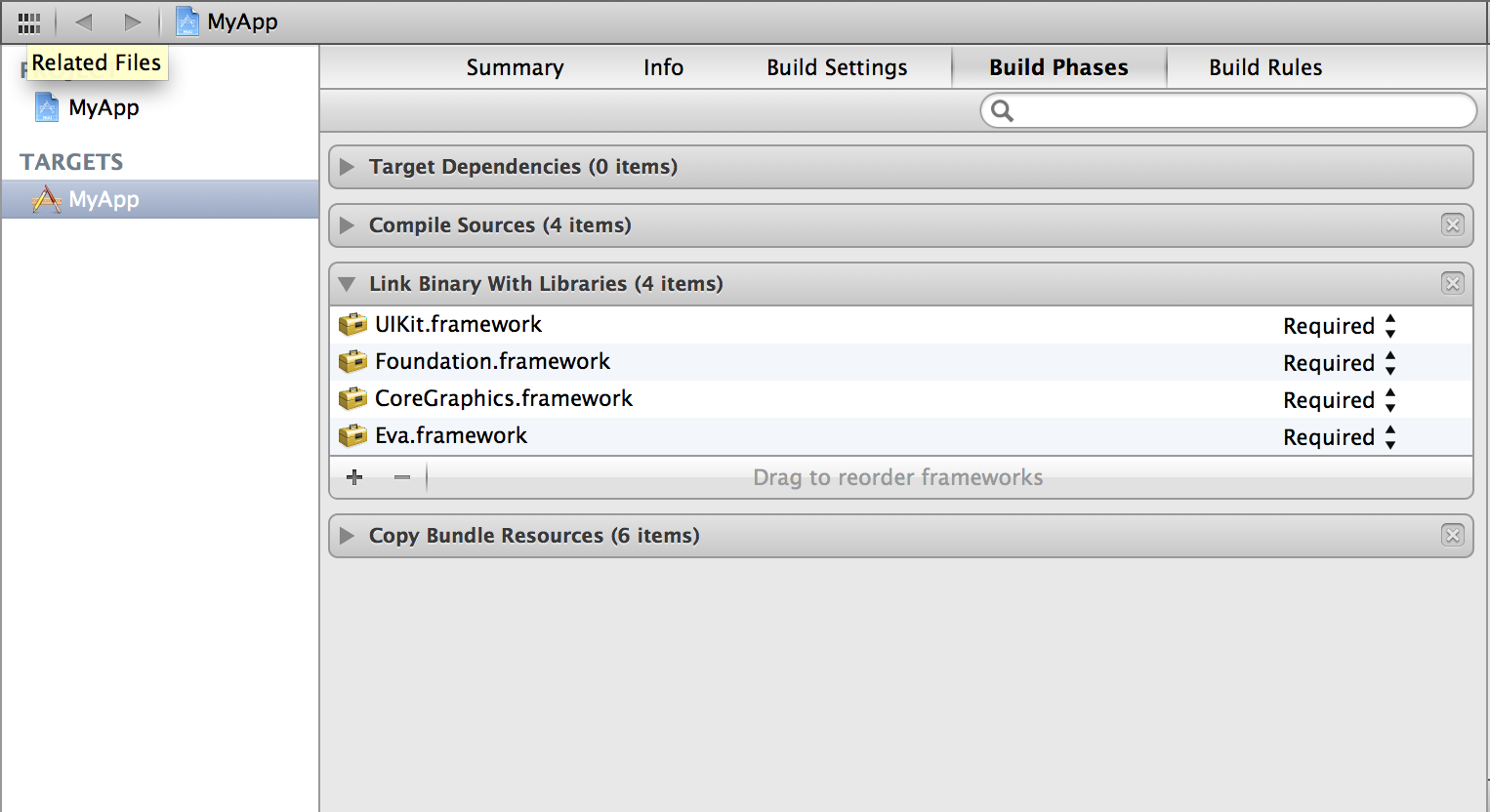
3. Drag Eva.framework file to your xCode project folder target (Make sure the “Copy items to destination’s group folder is checked).



4. It should look like this (Eva.framework is added):



5. Click on the Targets -> Your app name -> and then “build phases” tab, and then expand “Link binary with libraries”



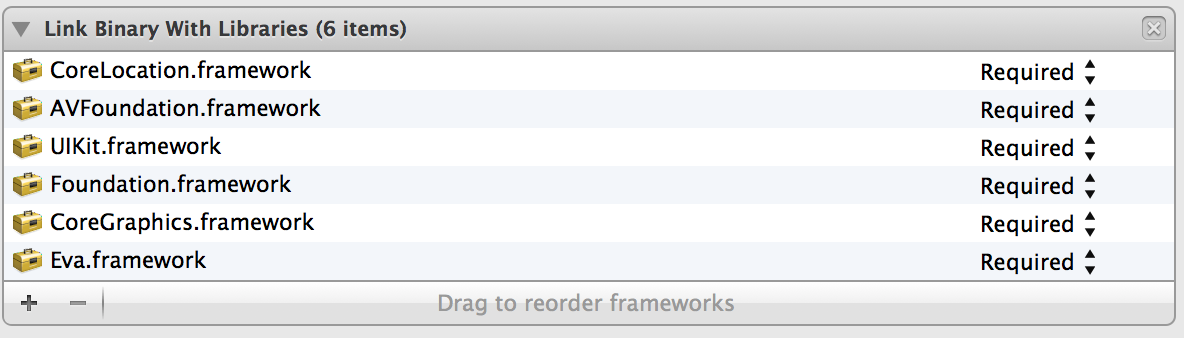
6. Click the ‘+’ button in that section (Bottom left)

7. Add the following libraries:

CoreLocation.framework

AVFoundation.framework

8. Now it should look like (Eva, CoreLocation and AVFoundation frameworks added):



# Code Integration

## UI Views Integration

### Single View Project

If you are using Eva at only one view inside your app please do the following (If not, see **Multiple Views Project** section):

**a.** Go to “YourViewController.h” (Where you want to integrate Eva) and add this line:

#import <Eva/Eva.h>

Add EvaDelegate to the Controller delegates and add evaModule object, File should look like that:

#import <Eva/Eva.h>

@interface YourViewController : UIViewController <FlipsideViewControllerDelegate,EvaDelegate>{

Eva \*evaModule;

}

@property(nonatomic,retain) Eva \*evaModule;

@end

**b.** Sythesize evaModule:

@implementation YourViewController

@synthesize evaModule;

**c.** Allocate evaModule, set the delegate and your keys to viewDidLoad:

- (void)viewDidLoad

{

[super viewDidLoad];

evaModule = [[Eva alloc] init];

evaModule.delegate = self;

// Initialize Eva keys //

[evaModule setAPIkey:@"YOUR-API\_KEY"

withSiteCode:@"YOUR-SITE-CODE"];

}

Jump to “**Use Eva and** **Handling With Results”** section (page 7).

### Multiple Views Project

If you want to use Eva on more than one view inside your app please do the following:

**a.** Go to your “AppDelegate.m” file, and add this line:

#import <Eva/Eva.h>

Add the following lines to

-(BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions{

[[Eva sharedInstance] setAPIkey:@"YOUR-API-KEY" withSiteCode:@"YOUR-SITE-CODE"];

}

**b.** Go to “YourViewController.h” (Where you want to integrate Eva) and import Eva also:

#import <Eva/Eva.h>

Add EvaDelegate to the Controller delegates, File should look like that:

#import <Eva/Eva.h>

@interface YourViewController : UIViewController < EvaDelegate>{

}

@end

**c.** Now, set the delegate in viewWillAppear:

-(void)viewWillAppear:(BOOL)animated{

[Eva sharedInstance].delegate = self; // Setting the delegate to this view //

// The delegate initiation is here for it to be set-up every time this view is called //

}

Do **b.** to **c.** to every view you want to integrate Eva.

**IMPORTANT**: In case you are using multiple views with Eva, as shown in this section, you should replace **evaModule** with **[Eva sharedInstance]** on the code that would be shown on the rest of this document.

## Use Eva and Handling With Results

1. Implement the two delegates (a must):

#pragma mark - Eva Delegate

- (void)evaDidReceiveData:(NSData \*)dataFromServer{

NSString\* dataStr = [[NSString alloc] initWithData:dataFromServer encoding:NSASCIIStringEncoding];

NSLog(@"Received data from Eva %@", dataStr);

}

- (void)evaDidFailWithError:(NSError \*)error{

NSLog(@"Got error from Eva");

}

1. To start recording a new session call:

[evaModule startRecord:TRUE];

If you want to continue the previous session should you call:

[evaModule startRecord:FALSE];

This would keep the previous session number.

1. When you want to stop the record, please call:

[evaModule stopRecord];

Record would stop automatically after 8 seconds of talking time or when silent is detected.

Expect to get a delegate call just after that.

1. All configurable parameters are optional. Those can be set, For example:

evaModule.home = @"paris";

evaModule.version = @"v1.0";

(In case you are using multiple views with Eva, as shown in the **Multiple   
Views Project** section, it is recommended to do all those settings in the **AppDelegate.m** file just before the **setApiKey:withSiteCode** function )

## Advanced API Parameters

1. If you want to implement mic activity level inside your app, On either the **Single View Project** section or **Multiple Views Project** section, you should call:

[evaModule setAPIkey:@"YOUR-API\_KEY"

withSiteCode:@"YOUR-SITE-CODE"

withMicLevel:TRUE];

Instead of:

[evaModule setAPIkey:@"YOUR-API\_KEY"

withSiteCode:@"YOUR-SITE-CODE"];

1. In case you set withMicLevel to TRUE you **must** implement:

-(void)evaMicLevelCallbackAverage: (float)averagePower andPeak: (float)peakPower;

-(void)evaMicStopRecording;

averagePower and peakPower are in decibels, evaMicLevelCallbackAverage:andPeak would be called for each mic sample. Check out EvaTest project for animation example.

1. In case you want to change default timeout for recording (default is 8 sec) you can use:

[evaModule setAPIkey:apiKeyString withSiteCode:siteCodeString withMicLevel:TRUE withRecordingTimeout:8.0f];

This is exactly same as on paragraph 1, but you can change 8 sec timeout to any value you wish in seconds, Pay attention to send a float value.

# CHECK OUT THE DEMO PROJECT (EvaTest).