Getting Started with rSkybox

Welcome to rSkybox! In just a few minutes time we'll have rSkybox integrated into your iOS application and you will be easily tracking errors and crashes!

Lets get started...

What you will need before you being
- rSkybox Application Id: the application id string you received when you registered for rSkybox
- rSkybox Basic Authentication Token: the token you received when you registered

Adding files to your project

- Find the folder titled 'rSkybox Helpers' and add it to your iOS project. You can do this by dragging the entire folder into your project in Xcode.

- After you drag the folder into your project, you will see a screen for 'options for adding these files'. Make sure the following boxes are checked: "Copy items into destination group's folder (if needed)" and "Add to targets - YourTargetName"

- Using the same drag and drop method, add the rSkybox.h and rSkybox.m files to your project, making sure the same to check boxes are checked when prompted

Adding code to rSkybox.m file

- Navigate to the file rSkybox.m and find the '//TODO - rSkybox import' at the top of the file, and replace it with an import of your projects AppDelegate.h file (example: #import AppDelegate.h)

- In the same file, find the '//TODO - rSkybox ids near the top of the file. Replace the applicationId string with the Application Id you received when you registered for rSkybox. Replace the basicAuthToken string with the Authorization Token you received when registering for rSkybox.

- In the same file, find the '//TODO - rSkybox' in the +(void)addEventToSession:(NSString *) event() method. Uncomment the line below the TODO and replace the 'MyAppDelegate' words with the name of your own app delegate

- Again in the same file, find all 4 locations of the '//TODO - rSkybox userId'.
Replace the word 'USERID' in the line below the TODO with a unique identifier for this user. (if you store a token or id for each user, use that). If not, you can randomly generate a unique string for each user and use this instead. You will need to store this string on the device, as it will be used when you report logs, crashes, or feedback with rSkybox

Adding code to your app delegate

- Add the following code to your app delegate .h file:

```
@property (nonatomic, strong) NSString *appActions;
```

```
@property (nonatomic, strong) NSString *appActionsTime;
@property (nonatomic, strong) NSString *crashSummary;
@property (nonatomic, strong) NSString *crashUserName;
@property (nonatomic, strong) NSDate *crashDetectDate;
@property (nonatomic, strong) NSData *crashStackData;
@property (nonatomic, strong) NSString *crashInstanceUrl;

-(void)saveUserInfo;
-(void)handleCrashReport;
```

- Add the following code to your app delegate .m file:

```
#import <CrashReporter/CrashReporter.h>
#import "UIDevice-Hardware.h"
#import "rSkybox.h"
@synthesize appActions, appActionsTime, crashSummary, crashUserName, crashStackData,
crashDetectDate, crashInstanceUrl;
```

- Add this init() method to your app delegate .m file (if you already have an init() method, then copy and past the code inside this method into your already existing method).

```
- (id) init {

  [rSkybox initiateSession];

  NSUserDefaults *prefs = [NSUserDefaults standardUserDefaults];
  self.appActions = [prefs valueForKey:@"appActions"];
  self.appActionsTime = [prefs valueForKey:@"appActionsTime"];

      @try {

            if ([self.appActions length] > 0) {
                //Set the trace session array
                NSMutableArray *tmpTraceArray = [NSMutableArray arrayWithArray:[self.appActions
componentsSeparatedByString:@","]];

                NSMutableArray *tmpTraceTimeArray = [NSMutableArray arrayWithArray:[self.appActionsTime
componentsSeparatedByString:@","]];

                NSMutableArray *tmpDateArray = [NSMutableArray array];

                for (int i = 0; i < [tmpTraceTimeArray count]; i++) {
                    NSString *tmpTime = [tmpTraceTimeArray objectAtIndex:i];

                    NSDateFormatter *dateFormatter = [[NSDateFormatter alloc] init];
                    [dateFormatter setDateFormat:@"YYYY-MM-dd hh:mm:ss.SSS"];
                    NSDate *theDate = [dateFormatter dateFromString:tmpTime];

                    [tmpDateArray addObject:theDate];
                }

                [rSkybox setSavedArray:tmpTraceArray :tmpDateArray];

            }
```

```
        }
        @catch (NSException *exception) {

        }

    return self;
}
```

- Again in your app delegate .m file, add the following code to your application’s didFinishLaunching() or didFinishLaunchingWithOptions() method:

```
PLCrashReporter *crashReporter = [PLCrashReporter sharedReporter];
NSError *error;

    /* Check if we previously crashed */
    if ([crashReporter hasPendingCrashReport]) {

        [self handleCrashReport];
    }

    if (![crashReporter enableCrashReporterAndReturnError: &error]){
        //NSLog(@"*************************Warning: Could not enable crash reporter: %@", error);
    }
```

- In the same file, add this code to the method:

```
-(void)applicationDidBecomeActive:(UIApplication *)application{

    [self performSelectorInBackground:@selector(createEndUser) withObject:nil];
```

- Add the following full new methods to your application delegate .m file

```
- (void) handleCrashReport {

    PLCrashReporter *crashReporter = [PLCrashReporter sharedReporter];
    NSData *crashData;
    NSError *error;
    self.crashDetectDate = [NSDate date];
    self.crashStackData = nil;
    //self.crashUserName = mainDelegate.token;

    /* Try loading the crash report */
    bool isNil = false;
    crashData = [crashReporter loadPendingCrashReportDataAndReturnError: &error];
    if (crashData == nil) {
        //NSLog(@"Could not load crash report: %@", error);
        isNil = true;
    }

    if (!isNil) {

        PLCrashReport *report = [[PLCrashReport alloc] initWithData: crashData error: &error];
        bool thisIsNil = false;
        if (report == nil) {
            self.crashSummary = @"Could not parse crash report";
            [self performSelectorInBackground:@selector(sendCrashDetect) withObject:nil];
            thisIsNil = true;
        }
        if (!thisIsNil) {

            @try {


                NSString *platform = [[UIDevice currentDevice] platformString];

                self.crashSummary = [NSString stringWithFormat:@"Crashed with signal=%@, app
version=%@, os version=%@, hardware=%@", report.signalInfo.name,
report.applicationInfo.applicationVersion, report.systemInfo.operatingSystemVersion, platform];

                self.crashStackData = [crashReporter loadPendingCrashReportDataAndReturnError: &error];

                [self performSelectorInBackground:@selector(sendCrashDetect) withObject:nil];
            }
            @catch (NSException *exception) {

            }


        }else{
            [crashReporter purgePendingCrashReport];
            return;
        }

    }else{
        [crashReporter purgePendingCrashReport];
        return;
    }




}

-(void)sendCrashDetect {

    @autoreleasepool {
        // send crash detect to GAE


        [rSkybox sendCrashDetect:self.crashSummary theStackData:self.crashStackData];


        PLCrashReporter *crashReporter = [PLCrashReporter sharedReporter];

        [crashReporter purgePendingCrashReport];

    }


}


-(void)saveUserInfo{

     NSUserDefaults *prefs = [NSUserDefaults standardUserDefaults];
            [prefs setValue:self.appActions forKey:@"appActions"];
     [prefs setValue:self.appActionsTime forKey:@"appActionsTime"];
            [prefs synchronize];
```

```
}



-(void)createEndUser{
    @autoreleasepool {
        [rSkybox createEndUser];
    }
}
```

Adding the Crash Reporter Framework

Add the CrashReporter.framework folder to your project by dragging the folder into your project in Xcode. Make sure the checkboxes are checked that will copy the files into the destination folder, and to add to your projects main target

If you are running on the simulator, add libCrashReporter-iphonesumulator.a file to project by dragging it into the project in Xcode
If running on device, add libCrashReporter-iphoneos.a file to project by dragging it into the project in Xcode

In Xcode, select your project, then select your main project target, click on the 'Build Phases' tab, and expand the "Link Binary With Libraries" section. Make sure the .a file you just added is above the CrashReporter.framework in this section (every time you switch from the simulator to the iOS .a file, you will have to check and make sure it is above the CrashReporter.framework listed in this section).

Build your project, and you are ready to go!

Using rSkybox-
-Crash Detection is up and running, you will see all crashes reported on your rSkybox web page
-Anywhere you want a log, use the [rSkybox sendClientLog()] method;
If you want to use the audio feedback feature, use the [rSkybox sendFeedback()] method;

App Actions
App Actions are a very powerful tool included in rSkybox that allow you to see the last 20 (or more) actions your users took before the device crashed, or the log was reported. To add these actions to your project, just include this line of code:

```
[rSkybox addEventToSession:@"YOUR EVENT"]
```

An example would be [rSkybox addEventToSession:@"User Logged In"]. If you add these events throughout your project, whenever a crash or log is reported, you will also see the last actions the user was going through before the problem occurred. This can prove to be invaluable information when trying to find the cause of the issue.

Before Adding app Actions or rSkybox methods to any of your classes, you must first include the following import to the top of the class:

```
#import “rSkybox.h”
```