

Viafoura Social Login and iOS UIWebView Integration

VIAFOURA SOCIAL LOGIN WITHIN AN IOS UIWEBVIEW

Objective

The objective of this document is to illustrate how to use a UIWebView component to access a page integrating with Viafoura in order to use the social login capability Viafoura provides.

When developing an app in iOS that includes a UIWebView component, a few additional steps are needed to ensure that the social login functionality that Viafoura provides functions properly.

This document describes what to do, why it needs to be done, and includes source code examples.

Why

The iOS UIWebView component provides a subset of the functionality of the iOS Safari browser, and provides a means to navigate to and control a web page. The UIWebView's behaviour deviates from regular browsers in the way that it manages popup windows.

The UIWebView component's management of popup windows is not the same as typical browsers are in that a popup window is loaded in place of the current document. Rather than opening a popup in a separate window, the UIWebView replaces the current page with the popup content, while not retaining an 'opener' link to the original page,

Viafoura's social login process requires a popup window to manage the authentication process with the appropriate social network. Once the social network has authorized a user's credentials, the popup window is closed, and the originating site will now be logged into Viafoura. Because the UIWebView component replaced it's originating page with the contents of the popup, the popup now has no opening window reference with which it can communicate it's successful login.

The ViafouraSocialLoginController replaces this functionality, which allows the social login process to work seamlessly.

How

This section explains how to use the code described in this document, and included in the Viafoura iOS UIWebView sample project to enable social login in your UIWebView project.

The following steps must be taken to allow the Viafoura social login process to work properly:

- 1.) Add the three ViafouraSocialLoginController files included in the sample project into the real project
- 2.) Add a reference to the ViafouraSocialLoginController.h from the file that controls the UIWebView
- 3.) Add this code to the module file that controls the UIWebView:

```
- (BOOL)webView:(UIWebView *)webViewInput shouldStartLoadWithRequest:(NSURLRequest *)request
navigationType:(UIWebViewNavigationType)navigationType;
{
    NSString *url = request.URL.absoluteString;

    //
    // Bring up the Viafoura Social Login controller to handle the social login process
    //
    if ([url rangeOfString:VIAFOURA_SOCIAL_LOGIN_URL].location != NSNotFound)
    {
        NSLog(@"Caught social login click. Sending request to Viafoura Social Login Controller.");

        socialLoginController = [[ViafouraSocialLoginViewController alloc] initWithURLRequest:request
andParentWebView:webViewInput];

        socialLoginController.delegate = self;

        //
        // Display the social login view controller
        //
        [self.view addSubview:socialLoginController.view];

        //
        // Stop processing this request - because it is being done by the social login controller.
        //
        return NO;
    }

    return YES;
}
```

The Viafoura UIWebView sample project includes a working sample of a UIWebView connecting to an example site. The project uses the three Viafoura Social Login management files that you need to include in your project.

The files to include in your project are:

1. ViafouraSocialLoginController.h
2. ViafouraSocialLoginController.m
3. ViafouraSocialLoginController.xib

The example project illustrates how a View Controller that manages a UIWebView component should interact with the page hosting Viafoura. As shown in the code snippet above, the ViewController will watch for a URL request for a Viafoura social login. If that is detected, then the Viafoura Social Login Controller performs all of the work necessary to manage the social login process.

The Viafoura Social Login Controller replicates the behaviour of the popup process in a regular web browser, and watches for a URL request that indicates a successful social login. Once that has been detected, it sends a JavaScript message to the parent UIWebView to notify if of the successful login, so that the Viafoura Javascript components in the parent UIWebView recognize the logged in state, and complete the login process.

For reference, the ViafouraSocalLoginController logic is included in the next section of this document.

ViafouraSocialLoginController.h

```
//
//  ViafouraSocialLoginViewController.h
//  Viafoura_UIWebview
//
//  Created by Demetrios Kallergis on 2014-06-29.
//  Copyright (c) 2014 Viafoura. All rights reserved.
//

#import <UIKit/UIKit.h>

@protocol ViafouraSocialLoginControllerDelegate <NSObject>

@optional
-(void) socialLoginSuccessful;
-(void) socialLoginCancelled;
@end

#define VIAFOURA_SOCIAL_LOGIN_URL @"viafoura-login.hub.loginradius.com/RequestHandler.aspx"

@interface ViafouraSocialLoginViewController : UIViewController <UIWebViewDelegate>
{
    IBOutlet UIButton *closeButton;
    IBOutlet UIWebView *socialLoginWebView;

    UIWebView *parentWebView;

    NSURLRequest *initialRequest;
    NSString *postMessageCommand;

    NSString *callback;

    BOOL successfulSocialLogin;
    BOOL takeDownPopup;

    id<ViafouraSocialLoginControllerDelegate> delegate;
}

@property (nonatomic, retain) id <ViafouraSocialLoginControllerDelegate> delegate;

-(IBAction) cancelSocialLogin: (id) sender;
-(id) initWithURLRequest:(NSURLRequest *) request andParentWebView:(UIWebView *) webView;

@end
```

ViafouraSocialLoginController.m

```
//  
// ViafouraSocialLoginViewController.m  
// Viafoura_UIWebview  
//  
// Created by Demetrios Kallergis on 2014-06-29.  
// Copyright (c) 2014 Viafoura. All rights reserved.  
//  
  
#import "ViafouraSocialLoginViewController.h"  
  
#define VIAFOURA_SOCIAL_LOGIN_SUCCEEDED_URL @"viafoura-login.hub.loginradius.com/success.aspx"  
  
@interface ViafouraSocialLoginViewController ()  
  
@end  
  
@implementation ViafouraSocialLoginViewController  
  
@synthesize delegate = delegate;  
  
-(id) initWithURLRequest:(NSURLRequest *) request andParentWebView:(UIWebView *) webView;  
{  
    self = [super initWithNibName:@"ViafouraSocialLoginViewController" bundle:nil];  
    if (self) {  
        initialRequest = request;  
        parentWebView = webView;  
    }  
}
```

```
        successfulSocialLogin = NO;

        takeDownPopup = NO;

        return self;
    }

    - (void)viewDidLoad
    {
        [super viewDidLoad];

        socialLoginWebView.scalesPageToFit = YES;

        [socialLoginWebView loadRequest:initialRequest];
    }

    -(IBAction) cancelSocialLogin: (id) sender
    {
        if (delegate != nil &&
            [delegate respondsToSelector:@selector(socialLoginCancelled)])
        {
            [delegate socialLoginCancelled];
        }

        [self.view removeFromSuperview];
        [self removeFromParentViewController];
    }
```

```
#pragma mark UIWebView Delegate Functions
```

```
- (BOOL)webView:(UIWebView *)webViewInput shouldStartLoadWithRequest:(NSURLRequest *)request  
navigationType:(UIWebViewNavigationType)navigationType;
```

```
{  
  
    NSString *url = request.URL.absoluteString;  
  
    NSLog(@"Popup Controller Url: %@", url);  
  
  
    //  
    // Look for the callback parameter when loading the social login URL. Keep it for later.  
    //  
    if ([url rangeOfString:VIAFOURA_SOCIAL_LOGIN_URL].location != NSNotFound )  
    {  
        //  
        // Read the call back url from the query string.  
        //  
        NSArray *params = [[[request.URL] query] componentsSeparatedByString:@"&"];  
        NSMutableDictionary *queryStringDictionary = [[NSMutableDictionary alloc] init];  
  
        for (NSString *keyValuePair in params)  
        {  
            NSArray *pairComponents = [keyValuePair componentsSeparatedByString:@"="];  
            NSString *key = [pairComponents objectAtIndex:0];  
            NSString *value = [pairComponents objectAtIndex:1];  
  
            [queryStringDictionary setObject:value forKey:key];  
        }  
    }  
}
```

```
        self->callback = [queryStringDictionary objectForKey:@"callback"];

    }

    //

    // Otherwise, check if this requests indicates that the social login proess has succeeded.  If it
    has succeeded,

    // then pull out the token parameter, and use it to build a JavaScript command to later inject
    into the parent UIWebView.

    //

    else if ([url rangeOfString:VIAFOURA_SOCIAL_LOGIN_SUCCEEDED_URL].location != NSNotFound )
    {

        NSLog(@"Caught successful social login.");

        //

        // Watch for a successful login, and then flip the logged in flag to true

        //

        //

        // grab the token from the query string.

        //

        NSArray *params = [[[request URL] query] componentsSeparatedByString:@"&"];
        NSMutableDictionary *queryStringDictionary = [[NSMutableDictionary alloc] init];

        for (NSString *keyValuePair in params)
        {

            NSArray *pairComponents = [keyValuePair componentsSeparatedByString:@"="];

            NSString *key = [pairComponents objectAtIndex:0];

            NSString *value = [pairComponents objectAtIndex:1];

            [queryStringDictionary setObject:value forKey:key];
```

```
    }

    self->postMessageCommand = [NSString stringWithFormat:@"window.postMessage('%@', '*');",
                               [queryStringDictionary objectForKey:@"token"]];

    NSLog(self->postMessageCommand);

    successfulSocialLogin = YES;
}

//
// Otherwise, check if the request being placed is for the callback server we stored earlier, and
make
// sure that we're successfully logged in. If both are true, then set the command to take down
// the social login controller once the request completes.
//
else if ([url rangeOfString:callback].location != NSNotFound &&
         successfulSocialLogin == YES)
{
    //
    // next, we want the popup taken down, if the callback loads successfully.
    //
    takeDownPopup = YES;
}

return YES;
}
```

```
- (void)webViewDidFinishLoad:(UIWebView *)webViewInput;
{
    //
    // If the callback request completes successfully, then inject the postMessage into the parent
    UIWebView,
    // then taken down the social login controller.
    //
    if (successfulSocialLogin == YES &&
        takeDownPopup == YES)
    {
        NSLog(@"About to inject postMessage to parent UIWebView");

        [parentWebView stringByEvaluatingJavaScriptFromString:self->postMessageCommand];

        if (delegate != nil &&
            [delegate respondsToSelector:@selector(socialLoginSuccessful)])
        {
            [delegate socialLoginSuccessful];
        }

        [self.view removeFromSuperview];
        [self removeFromParentViewController];
    }
}

@end
```