

Viafoura Social Login and Android WebView Integration

VIAFOURA SOCIAL LOGIN WITHIN AN ANDROID WEBVIEW

Objective

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

When developing an app in Android that includes a WebView 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 Android WebView component provides a subset of the functionality of the Android Webkit browser. The WebView's behaviour deviates from regular browsers in the way that it manages popup windows. When a popup window is opened from a link in a page within a WebView, it is opened separately in the Android browser, and not within the same WebView. This breaks the continuity of the web page, and makes it impossible to return control to the page hosted in an app, in a WebView,

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 WebView component opens the contents of the popup in the separate browser app, the popup now has no opening window reference with which it can communicate it's successful login back to the source app.

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 Android WebView sample project to enable social login in your WebView project.

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

- 1.) Add the `com.viafoura.SocialLoginController` Activity class, and `social_login_controller.xml` files into the real project
- 2.) Add a reference to the `SocialLoginController` from the class that controls the `WebView`
- 3.) Update the `WebView` management code in your `WebView` Activity class to contain the following logic to initialize the `WebView`, and to assign it a `WebViewClient` that will delegate the Social Login process to the Viafoura Social Login Controller:

```
private android.webkit.WebView mWebView;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_web_view);

    mWebView = (WebView) findViewById(R.id.activity_web_view);

    // Enable Javascript
    WebSettings webSettings = mWebView.getSettings();
    webSettings.setJavaScriptEnabled(true);

    mWebView.setWebViewClient(new WebViewClient() {

        @Override
        public boolean shouldOverrideUrlLoading(WebView view, String url) {

            if (url.contains(com.viafoura.SocialLoginController.VIAFOURA_SOCIAL_LOGIN_URL)) {

                Log.i("shouldOverrideUrlLoading", "Caught social login request.  Instantiating
Social Login Controller with URL: " + url);

                Intent slcIntent = new Intent(getApplicationContext(),
SocialLoginController.class);
```

```

        slcIntent.putExtra(SocialLoginController.VIAFOURA_SOCIAL_LOGIN_INTENT_URL_KEY,
url);

        startActivityForResult(slcIntent, 1);

        return true;
    }
    else
    {
        return false;
    }
}

});

mWebView.loadUrl(TEST_SITE);
}

```

4.) Update the WebView management code in your WebView Activity class to contain the following logic to handle the return values from the Viafoura Social Login Controller:

```

// Listen for results.
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data){
    // See which child activity is calling us back.
    switch (requestCode) {

        case 1:

            switch (resultCode ) {

                case Activity.RESULT_OK:

                    String postCommand =
data.getExtras().getString(SocialLoginController.VIAFOURA_SOCIAL_LOGIN_INTENT_RESULT_KEY);

                    Log.i("onActivityResult", "Caught successful Social Login" + postCommand);

                    mWebView.loadUrl("javascript:" + postCommand, null);

                    break;

                default:

                    Log.i("onActivityResult", "Caught Social Login cancellation");

                    break;
            }
        }
    }
}

```

```
        }  
        break;  
    default:  
        break;  
    }  
}
```

The Viafoura WebView sample project includes a working sample of a WebView connecting to an example site. It includes the class and XML files needed to integrate into your project.

The example project illustrates how an Activity class that manages a WebView component should interact with the page hosting Viafoura. As shown in the code snippet above, the WebViewClient 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 Activity class to notify it of the successful login, so that the Viafoura Javascript components in the parent WebView recognize the logged in state, and completes the login process.

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

SocialLoginController.java

```
package com.viafoura;

/**
 * Created by demetree on 2014-07-07.
 */

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.webkit.WebView;
import android.webkit.WebSettings;
import android.webkit.WebViewClient;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.view.View;
import android.view.View.OnClickListener;
import android.content.Intent;
import java.net.URL;
import java.net.URLDecoder;
import java.util.Map;
import java.util.LinkedHashMap;

public class SocialLoginController extends Activity {

    public final static String VIAFOURA_SOCIAL_LOGIN_URL = "viafoura-login.hub.loginradius.com/RequestHandler.aspx";
    public final static String VIAFOURA_SOCIAL_LOGIN_INTENT_URL_KEY = "com.viafoura.SocialLoginController.url";
    public final static String VIAFOURA_SOCIAL_LOGIN_INTENT_RESULT_KEY = "com.viafoura.SocialLoginController.result";
    private final static String VIAFOURA_SOCIAL_LOGIN_SUCCEEDED_URL = "viafoura-login.hub.loginradius.com/success.aspx";

    private android.webkit.WebView mWebView;
    private Button mCloseButton;

    private String callbackURL;
    private String postMessageCommand;
    private boolean successfulSocialLogin;
    private boolean closing;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.social_login_controller);

successfulSocialLogin = false;
callbackURL = "";
postMessageCommand = "";
closing = false;

mWebView = (WebView) findViewById(R.id.activity_web_view);

// Enable Javascript
WebSettings webSettings = mWebView.getSettings();
webSettings.setJavaScriptEnabled(true);

//
// Get the URL to go to from the Intent
//
Bundle extras = getIntent().getExtras();
String urlString =
extras.getString(SocialLoginController.VIAFOURA_SOCIAL_LOGIN_INTENT_URL_KEY);

mWebView.setWebViewClient(new WebViewClient() {

    @Override
    public void onPageFinished(WebView view, String urlString) {

        Log.i("SocialLoginController.onPageFinished", urlString);

        if (successfulSocialLogin == true &&
            urlString.startsWith(callbackURL) &&
            closing == false) {

            closing = true;

            Intent resultIntent = new Intent();

            resultIntent.putExtra(VIAFOURA_SOCIAL_LOGIN_INTENT_RESULT_KEY,
postMessageCommand);

            setResult(Activity.RESULT_OK, resultIntent);

            Log.i("SocialLoginController.onPageFinished", "Closing Social Login Controller");

            finish();
        }
    }
});
```

```

        return;
    }

    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String urlString) {

        Log.i("SocialLoginController.shouldOverrideUrlLoading", urlString);

        //
        // 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.
        //
        if (urlString.contains(SocialLoginController.VIAFOURA_SOCIAL_LOGIN_SUCCEEDED_URL)) {

            Log.i("SocialLoginController.shouldOverrideUrlLoading", "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.
            //
            try {

                URL url = new URL(urlString);

                Map<String, String> query_pairs = new LinkedHashMap<String, String>();
                String query = url.getQuery();
                String[] pairs = query.split("&");
                for (String pair : pairs) {
                    int idx = pair.indexOf("=");
                    query_pairs.put(URLDecoder.decode(pair.substring(0, idx), "UTF-8"),
URLDecoder.decode(pair.substring(idx + 1), "UTF-8"));
                }

                // build the JavaScript post message containing the token.
                postMessageCommand = "window.postMessage('" + query_pairs.get("token") + "',
'*')";

                Log.i("SocialLoginController.shouldOverrideUrlLoading", "postMessageCommand
is: " + postMessageCommand);

            } catch (Exception ex) {

```

```
        callbackURL = "";
    }

    successfulSocialLogin = true;

}

// default result
return false;
}

});

//
// Look for the callback parameter when loading the social login URL. Keep it for later.
//
if (urlString.contains(com.viafoura.SocialLoginController.VIAFOURA_SOCIAL_LOGIN_URL)) {

    //
    // Read the call back url from the query string.
    //
    try {

        URL url = new URL(urlString);

        Map<String, String> query_pairs = new LinkedHashMap<String, String>();
        String query = url.getQuery();
        String[] pairs = query.split("&");
        for (String pair : pairs) {
            int idx = pair.indexOf("=");
            query_pairs.put(URLDecoder.decode(pair.substring(0, idx), "UTF-8"),
                URLDecoder.decode(pair.substring(idx + 1), "UTF-8"));
        }

        callbackURL = query_pairs.get("callback");

        Log.i("SocialLoginController.shouldOverrideUrlLoading", "Callback URL is: " +
callbackURL);
    } catch (Exception ex) {
        callbackURL = "";
    }
}

mCloseButton = (Button) findViewById(R.id.close_button);

mCloseButton.setOnClickListener(new OnClickListener() {
```

```
@Override
public void onClick(View arg0) {

    Intent resultIntent = new Intent();

    setResult(Activity.RESULT_CANCELED, resultIntent);

    finish();
}

});

// load the social login page.
mWebView.loadUrl(urlString);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.web_view, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}
```