

INTERNATIONAL MULTIDISCIPLINARY
SCIENTIFIC GEOCONFERENCES

17-26 June, 2014, Bulgaria

14th GeoConference on
INFORMATICS, GEOINFORMATICS
AND REMOTE SENSING

Conference Proceedings

Volume I

INFORMATICS
GEOINFORMATICS

SGEM


DISCLAIMER

This book contains abstracts and complete papers approved by the Conference Review Committee. Authors are responsible for the content and accuracy.

Opinions expressed may not necessarily reflect the position of the International Scientific Council of SGEM.

Information in the SGEM 2014 Conference Proceedings is subject to change without notice. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of the International Scientific Council of SGEM.

Copyright © SGEM2014

All Rights Reserved by the International Multidisciplinary Scientific GeoConference SGEM

Published by STEF92 Technology Ltd., 1 “Andrey Lyapchev” Blvd., 1797 Sofia, Bulgaria

Total print: 5000

ISBN 978-619-7105-10-0

ISSN 1314-2704

DOI: 10.5593/sgem2014B21

**INTERNATIONAL MULTIDISCIPLINARY SCIENTIFIC GEOCONFERENCE SGEM
Secretariat Bureau**

Phone: +359 2 975 3982

Fax: +359 2 874 1088

E-mails: sgem@sgem.org | sgem@stef92.com

URL: www.sgem.org

CONFERENCE ORGANIZERS

- BULGARIAN ACADEMY OF SCIENCES
- ACADEMY OF SCIENCES OF THE CZECH REPUBLIC
- LATVIAN ACADEMY OF SCIENCES
- POLISH ACADEMY OF SCIENCES
- RUSSIAN ACADEMY OF SCIENCES
- SERBIAN ACADEMY OF SCIENCES AND ARTS
- SLOVAK ACADEMY OF SCIENCES
- NATIONAL ACADEMY OF SCIENCES OF UKRAINE
- INSTITUTE OF WATER PROBLEM AND HYDROPOWER OF NAS KR
- NATIONAL ACADEMY OF SCIENCES OF ARMENIA
- SCIENCE COUNCIL OF JAPAN
- THE WORLD ACADEMY OF SCIENCES (TWAS)
- EUROPEAN ACADEMY OF SCIENCES, ARTS AND LETTERS
- ACADEMY OF SCIENCES OF MOLDOVA
- MONTENEGRIN ACADEMY OF SCIENCES AND ARTS
- CROATIAN ACADEMY OF SCIENCES AND ARTS, CROATIA
- GEORGIAN NATIONAL ACADEMY OF SCIENCES
- ACADEMY OF FINE ARTS AND DESIGN IN BRATISLAVA
- TURKISH ACADEMY OF SCIENCES
- BULGARIAN INDUSTRIAL ASSOCIATION
- BULGARIAN MINISTRY OF ENVIRONMENT AND WATER

HONORED ORGANIZER



BULGARIAN ACADEMY OF SCIENCES

EXCLUSIVE SUPPORTING PARTNER



INTERNATIONAL SCIENTIFIC COMMITTEE

Informatics, Geoinformatics and Remote Sensing

- PROF. ING. ALEŠ ČEPEK, CSC., CZECH REPUBLIC
- PROF. G. BARTHA, HUNGARY
- PROF. DR. DAMIR MEDAK, CROATIA
- PROF. PETER REINARTZ, GERMANY
- PROF. DR. JÓZSEF ÁDAM, HUNGARY
- PROF. RUI MIGUEL MARQUES MOURA, PORTUGAL
- PROF. DR. ISIK YILMAZ, TURKEY
- PROF. DR. ING. KAREL PAVELKA, CZECH REPUBLIC
- PROF. DR. MARCEL MOJZES, SLOVAKIA

- ASSOC. PROF. DR MILAN HOREMUZ, SWEDEN
- DR. TIBERIU RUS, ROMANIA
- DR. MARKO KREVS, SLOVENIA

10. COMPARISON OF DATABASE MIRRORS TECHNOLOGY OF THE SELECTED DATABASE SYSTEMS FOR USE IN FAULT-TOLERANT INFORMATION SYSTEMS SOLUTIONS , Dr. Lukas Otte, Dr. Roman Danel, Dr. Vladislav Vancura, Dr. Michal Repka, Dr. Marketa Rolcikova, VSB-Technical University of Ostrava, Czech Republic	73
11. COMPARISON OF THE RESULTS OF LAMINAR AND TURBULENT FLOW OF METHANE IN POROUS MEDIA , Pavel Stasa, Ph.D., Assoc. Prof. Vladimir Kebo, Assoc. Prof. Oldrich Kodym, VSB-Technical University of Ostrava, Czech Republic	81
12. COMPETENCY-BASED MANAGEMENT OF INFORMATION FLOWS IN CONSTRUCTION PROJECTS , Assoc. Prof. Peter Mesaros, Ph.D., Ing. Tomas Mandicak, Assoc. Prof. Margita Mesarosova, Ph.D., Research Institute of ICT in Construction Sector Ltd. - Vyskumny ustav stavebnej informatiky s.r.o. , Slovakia.....	89
✓ 13. CONNECTING AIR MOBILE AND DESKTOP APPLICATIONS WITH SOCIAL NETWORKS , M.Sc Dalibor Serafimovski, Prof. Dr. Kosta Mitreski, University Goce Delcev, FYR of Macedonia.....	97
14. DESIGNING A WORK ENVIRONMENT USING AN ERGONOMIC APPLICATION MODEL OF THE PROCESS OF LATHING TWO-WHEEL ASSEMBLIES , Assoc. Prof. Michaela Balazikova, Ph.D., Ing. Michal Hovanec, Ph.D., Ing. Stefan Erdelyi, Technical University of Kosice, Slovakia.....	105
15. DEVELOPMENT OF ALGORITHM FOR LOCALIZATION OF PRODUCTION MARKINGS WITH THE USE OF ANALYSIS OF THE COLOR DATA ON DIGITAL IMAGES , Prof. Alexey Orlov, Alexandr Astafiev, Murom Institute of Vladimir State University, Russia.....	113
16. DEVELOPMENT OF LABORATORY WORK FOR REMOTE ACCESS LABORATORY , Prof. Andrey Vladimirovich Ostroukh, Prof. Andrey Borisovich Nikolaev, Nataliya Evgenievna Surkova, Prof. Mikhail Nikolaevich Krasnyanskiy, Moscow Automobile and Road construction State Technical University, Russia	119
17. DIGITAL THERMAL SENSOR FOR FPGA LOCAL AREA BASED ON MONTE-CARLO METHOD , Assoc.Prof. Pavel Kustarev, Sergey Bikovsky, Alexander Antonov, University ITMO, Russia.....	127
18. DISPATCHING CENTRES AND THEIR INFORMATION SUPPORT , Dr. Lukas Otte, Dr. Vladislav Vancura, Dr. Roman Danel, VSB-Technical University of Ostrava, Czech Republic	135
19. DNA ENCRYPTION FOR DATA TRANSMISSION AND STORAGE SECURITY IN INDUSTRIAL ENVIRONMENTS , Assoc. Prof. Dr. Otilia Cangea, Assoc. Prof. Dr. Adrian Moise, Assist. Prof. Dr. Cristina Popescu, Petroleum-Gas University of Ploiesti, Romania	143

CONNECTING AIR MOBILE AND DESKTOP APPLICATIONS WITH SOCIAL NETWORKS

M.Sc Dalibor Serafimovski¹

Full Prof. Dr.Kosta Mitreski²

¹Faculty of Computer Science, University "Goce Delčev"-Štip, **R. Macedonia**

²Faculty of Computer Science and Engineering, University "Ss.Cyril and Methodius" Skopje, **R. Macedonia**

ABSTRACT

Connecting with social networks has become an essential part of the process of mobile and desktop application development. Lots of applications, especially the mobile applications are using different authentication methods to authenticate their users and store the user data into their database. The method of authentication using social networks has proved to be very effective in many different ways. The main benefit of using this method is time-saving and avoidance of the registration process. Users prefer to use their profiles on social networks for registration on a web-based system rather than creating a separate user account. This paper provides tips for connecting AIR mobile and desktop applications with Facebook using ActionScript Graph API. The paper is divided in two parts. The first part is about connecting mobile applications with Facebook using Facebook Mobile ActionScript API. In this part we are focused on giving clear guidance for the process of authentication using access token and it is elimination of potential barriers during the process of implementation. Also, very important to mention that this method is cross-platform oriented. That means that the mobile application will be suitable across many different devices and platforms. This also raises the need of analyzing the rendering of HTML content across mobile devices using ActionScript. The second part of the paper covers the desktop AIR applications and the connection with social networks on desktop systems which is less used method but just as effective as in mobile systems.

Keywords: Flex, Adobe AIR, social networks, authentication, user, profile

INTRODUCTION

The main goal of this paper is to give clear guidance in the process of connecting AIR mobile and desktop applications with Facebook social network. First of all we need to define the characteristics of the AIR runtime environment and Flex mobile and desktop applications. Adobe AIR is cross-platform runtime environment which allows developers to create applications for different operating systems, browsers and also different mobile platforms. Step by step in the paper we will review the process of connecting AIR desktop applications and mobile applications with the Facebook social network as a modern way of user authentication and attracting customers. The usage of the Facebook platform occupies a large scale today, everybody from people to companies are on this social platform trying to attract attention.

Especially mobile developers see the opportunity here to attract customers for their mobile applications. Lots of online stores, mobile and desktop games are using Facebook platform for authenticating their users. They easily found a way to develop

large database of users. This method is also acceptable for the client side because users doesn't want to create different user accounts for every service on the internet. For example, mobile applications are so popular nowadays, people are downloading large amount of apps for their cell phones and tablets every day, it is very irritating to create different user profiles for all those applications and games.

Therefore this method becomes very popular in programming world. Developers also see an opportunity here for free advertising of their product. We decided to present the use of this method for AIR applications just because this technology is cross-platform oriented and it is very useful for implementation inside mobile applications.

FIRST STEP - REGISTERING FACEBOOK APPLICATION

In this paper we will try to simplify the process and to pay attention only on the main points. The analysis of the possible obstacles during this process is also very important. The very first step is to go on the Facebook developers¹ portal and register a Facebook application to receive a unique Facebook identifier for your application. The next figure shows the Facebook developers portal.

Figure 1. Creating and registering Facebook application on the Facebook developers portal. Inside the red rectangle is marked the unique application identifier.

Application ID is necessary for communication between Facebook and your application. GraphAPI is the primary way for getting data in and out of Facebook's social graph. According to the references, the Facebook GraphAPI is a HTTP based API that developers are using to access Facebook profiles query data and do all the needed tasks for their applications. Here is an example of using HTTP *GET* request to retrieve data and information about the current user:

Example code snippet:

```
FB.api (
  "/me",
```

¹<https://developers.facebook.com>

```
function (response) {
    if (response && !response.error) {
    }    });
```

CONNECTING FLEX MOBILE APPLICATION WITH FACEBOOK PLATFORM

After generating the Facebook application on the Facebook developers portal, now we are ready to start. First of all, for the purposes of this paper we will use the Flex technology combined with MXML and ActionScript to create the necessary visual elements[2]. The next example demonstrates the creation of the main MXML file for the application. At the beginning we are using “Creation Complete” handler to call the *init()* function and start the initialization with Facebook.

Example code snippet:

```
<?xml version="1.0" encoding="utf-8"?>
<s:View xmlns:fx="http://ns.adobe.com/mxml/2009"
        xmlns:s="library://ns.adobe.com/flex/spark" title="Login
with Facebook"
        currentState = "loginstate"
        creationComplete="init()"
        destructionPolicy = "never"
>
```

To be able to work, first of all we need to download the corresponding swc file and add the file inside Flash Builder to become available in our working environment. Inside the *init()* function, we need to call the static *init* method of the Facebook Mobile class. The next example code snippet demonstrates the creation of the *init()* function.

Example code snippet:

```
<fx:Script>
    <![CDATA[
        import com.facebook.graph.FacebookMobile
        import flash.geom.Rectangle;
        import flash.media.StageWebView;
        import spark.events.ViewNavigatorEvent;
        public const FACEBOOK_APP_ORIGIN:String = "";
        private const APP_ID:String = "2132270088*****";
        private var accessToken:String = "";
        private var stageweb:StageWebView;

        private function init():void
        {
            //Initialize Facebook Library
            FacebookMobile.init(APP_ID, InitCompleted);
            busyind.visible = true;
        }
    ]]>
```

The first thing at the beginning is to create two constants and two variables. The important parameters here are:

- FACEBOOK_APP_ORIGIN
- APP_ID – the unique identifier from your registered Facebook application
- AccessToken
- StageWebView

We are using the StageWebView class inside our mobile application as a preferred method for rendering HTML content within Flex mobile applications. The main function of the *init* method is to see if there is a shared object for this application. If not, it creates it. You will notice that the *init()* method has two required arguments, you have to pass the application ID and a callback function.

The first time the user accesses the application, an access token does not exist and login to Facebook is needed. To accomplish this, developers need to use *login()* method from the FacebookMobile class. This method displays the Facebook login page inside our mobile application. According to Facebook Mobile documentation here are some of the most important public methods of the FacebookMobile class:

- api – makes a new request on the Facebook Graph API
- getSession – a call to return the current user's session
- getImageURL – a method to load an image from Facebook
- login – opens a new login window for the current user
- logout – clears a user's local session
- postData – shortcut method to post data on Facebook
- requestExtendedPermissions – asks the current user for extended permissions
- uploadVideo – shortcut method to upload video on Facebook

CREATING WEBVIEW

The next step is creating webview using StageWebView class. The StageWebView class is used to render HTML content on devices where HTMLLoader class is not supported. This class does not provide interaction between the HTML content and ActionScript. According to the references, the StageWebView class uses the system web control provided by the operating system of the device. You need to create a *viewPort* property in order to control the size of the rendering area. You need to display the StageWebView object directly by attaching it to a stage using the *stage* property. According to the references here are some of the public methods of the StageWebView class:

- stagewebview – creates a stagewebview object
- loadString – loads and displays the specified HTML string
- loadURL – loads the page at the specified URL
- reload – reloads the current page
- stop – halts the current load operation
- historyBack – navigates to the previous page in the browsing history

It is of paramount importance to understand the functioning of this class. The StageWebView class acts like a browser inside the Flex mobile application, hence the user is given a wide range of opportunities.[4]The only problem here is the fact that StageWebView does not go together with spark visual elements. In order to be able to achieve our goal, we will focus on the defining of the position of the StageWebView

object inside the stage. The next step is to create *InitCompleted* function. Inside this function we will create an instance of the StageWebView class and define the position and the size of the rectangle using *viewPort* property.

The next code snippet demonstrates the structure of the *InitCompleted* function. This function is called after the initialization is completed.

Example code snippet:

```
public function InitCompleted(a:*,b:*):void
{
    var mystageweb:StageWebView = new StageWebView();
    mystageweb.stage = this.stage;
    mystageweb.viewPort = new Rectangle(0, navigator.actionBar.height, width, height);
    FacebookMobile.login(this.onLogin, this.stage, ["publish_stream", "user_photos"], mystageweb);
}
```

Defining the X and Y position of the object

The next and most important step is to create the *onLogin* function and get the access token, the user image and the name of the current logged user. The next code snippet demonstrates the structure of the *onLogin* function.

Example code snippet:

```
public function onLogin(success:*, b:*):void
{
    currentState = "loggedin"
    if (success)
    {
        accessToken = success.accessToken
        username.text = success.user.name;
        userImg.source=FacebookMobile.getImageUrl(success.uid,"large");
        setPostMessage(success.user);
        busyInd.visible = false;
    }
}
```

If you look at the *userImg* line, you can see that we make a call here and request a “large” facebook profile picture. Instead of “large” you can simple use “small” to get small facebook profile picture from the current logged user. With this final step, the Facebook login procedure is completed.[3]Figure 2 shows our application running on iOS 7 device without any errors. The next part of this paper covers the visual elements inside the mobile application and functions for posting on Facebook wall, dispose the web view and also the logout function. It is very important to mention here that we used two different states. The first is “loginstate” and the second is “loggedin” state. The

visual elements are placed in the second state. When the *onLogin* function is called, the current state changes as it is shown on the code snippet.

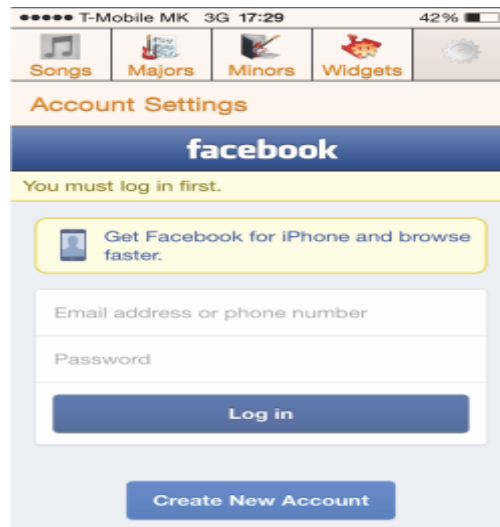


Figure 2. Facebook Login page running inside mobile application on iOS device

CREATING VISUAL ELEMENTS USING MXML

Companies and developers also use social networks for advertisement and promotion of their products. Facebook Graph API offers this opportunity to developers. Hence, there is a necessity to create functions within the application that will be able to distribute content directly from the application to the social network. Firstly, we need to create and sort the visual elements inside the mobile application. Having in mind that this is cross-platform oriented application, ordering of the visual elements takes specific part of the work. For creation of the visual elements we are using MXML language and Adobe Flash Builder [1]. The next code snippet demonstrates the creation of the needed visual elements for the application to be able to communicate with Facebook platform.

Example code snippet:

```
<s:layout>
    <s:VerticalLayout horizontalAlign="center" />
</s:layout>

<s:VGroup id="pstctn"
    includeInLayout="false"
    visible="false"
    horizontalAlign="center"
    width="100%" height="100%">

    <s:Label id="labelstatus" />
    <s:TextArea id="posttext"
        width="100%" height="25%" />

    <s:Button id="postbutton"
        label="Post"
        width="35%"
        click="postonwall(event)" icon="assets/icons/facebook.png" iconPlacement="left" />
    <s:Button id="logoutbutton"
        label="Logout"
        width="35%"
        click="onLogout()" iconPlacement="left" />
</s:VGroup>
```

ADDITIONAL FUNCTIONALITY

To build a fully functional application, you need to add other features in order to be able to communicate with Facebook platform. In order to be able to share content directly from Flex mobile application to Facebook it is necessary to build post method with all the required functions. And at the end, the final step would be adding logout button. For the logout method we recommend using the *logout()* method from FacebookMobile class and a simple method of hiding the visual elements on the stage and switching back to previous state. It is important to mention that, at the very beginning of the code definition, we've set the "destruction policy" property to "never" so that, the application will remember the last active state of this specific view.

Example code snippet:

```
private function thepostmessage(userObj:Object):void
{

    pstctn.includeInLayout = true;
    pstctn.visible = true;

    posttext.text = "http://dimdevelop.com";

}
```

We observe that it is very important for all functions to be properly connected in order the application to work as intended and not as it's coded. This next example code snippet demonstrates the method we used for posting on Facebook wall.

Example code snippet:

```
private function fbWallPost(e:Event):void
{
    busyInd.visible = true;
    var obj:Object = {};
    obj.message = posttext.text;
    obj.action_token = accessToken;
    FacebookMobile.api("/me/feed", PstStatus, obj, "POST");
}
private function FBinit(result:Object, fail:Object):void
{

    if(result)
    {
        accessToken = result.accessToken;
        thepostmessage(result.user);
    }

}
```

CONNECTING AIR DESKTOP APPLICATION WITH FACEBOOK PLATFORM

Connecting the AIR Desktop applications with Facebook platform also represents a new approach in desktop application development [2]. The process goes pretty much the same way; the only difference is the use of the StageWebView class. It is important to mention that for the Desktop applications we are using FacebookDesktop ActionScript API instead of FacebookMobile, also available for download on internet. As we said before, here we are not using StageWebView because the AIR desktop application acts like a browser and a simple function to call the Facebook page is enough. The next example code snippet demonstrates the difference.

Example code snippet:

```
public function afterInit(a:*,b:*) :void
{

    FacebookDesktop.login(this.fbLogin, ["publish_stream", "user_photos"] );

}
```

CONCLUSION

Linking mobile and desktop applications with social networks is an essential part of the modern development world. Using the new method of authentication is to facilitate the users and developers and also a great way to attract additional customers. Flex technology offers many different opportunities for developers and it is especially significant because it is cross-platform oriented. We hope that the approach and the method we described in this paper will be a good starting point for developers to try to make their applications more friendly and social.

REFERENCES

- [1]. Adobe Systems: Action Script 3.0 for Adobe Flash Professional, Classroom in a Book, Adobe Systems Incorporated and its licensors, The Official Training Workbook, 14-30. (2011)
- [2]. Yakov F. Viktor R. Anatole T.: Enterprise Development with FLEX, Adobe Developers Library, 63-97. (2010)
- [3]. Michael J. Williams: Facebook Graph API Development with Flash, Packt Publishing, 83-102. (2010)
- [4]. Steven P.: Flash Builder and Flash Catalyst, The New Workflow, Springer-Verlag New York, Inc., USA, 40-56. (2010)
- [5]. Rich T.: Developing iOS Applications With Flex 4.5 O'Reilly Media, Inc., USA, 47-75. (2011)
- [6]. John J. Maver, Cappy Popp: Essential Facebook Development, Pearson Education, Inc., USA, 83-107. (2010)
- [7]. Goldman J.: Facebook Cookbook, O'Reilly Media, Inc., USA, 125-140. (2009)
- [8]. Hasin H., Mark A. Bain: Learning Facebook Application Development, Packt Publishing, 5-42. (2008)