

Cross Platform Development

Lab # 5 Part A: Using PhoneGap for Cross-Platform Development (Android)

The objective of this lab is to build upon the work completed in previous labs that taught you how to create a BlackBerry WebWorks smartphone application. This lab will show you how to import that application into PhoneGap in order to deploy it across many platforms. In this lab, we'll be making changes to the configuration document and using the PhoneGap build service to compile applications. We will test this application in the Android Emulator and then learn how to sign the application for distribution. This lab also assumes that you have knowledge of basic HTML and CSS.

Transitioning WebWorks Application for the Android Platform

Step 1: Updating the config.xml file

- Open your config.xml (from Lab 4) in your favorite text editor.
- You will see that many elements mention BlackBerry devices, for PhoneGap we will change these to fit their standard.
- Change the application name between the name tags to something meaningful. Also add your name as the author.
- Where you find `xmlns:rim="http://www.blackberry.com/ns/widgets"` change it to:

```
xmlns:gap = "http://phonegap.com/ns/1.0"
```

- For Android version control, also add the version code after the above line but still inside the xml tag

```
versionCode="10"
```

- In order to disable the cursor on BlackBerry devices, after the author line add

```
<preference name="disable-cursor" value="true" />
```

- To allow network communication, add the line:

```
<feature name="http://api.phonegap.com/1.0/network"/>
```

- To set the device to always display the application in landscape mode, add this line:

```
<preference name="orientation" value="landscape" />
```

- Your completed index.xml file should look similar to this:

```
<?xml version="1.0" encoding="UTF-8"?>
<widget
  xmlns="http://www.w3.org/ns/widgets"
  xmlns:gap = "http://phonegap.com/ns/1.0"
```

```

version="1.0.0.0"
versionCode="10">

<name>Scholar App Lab 5</name>
<author href="http://cmer.uoguelph.ca/">CMER</author>

<preference name="disable-cursor" value="true" />
<feature name="http://api.phonegap.com/1.0/network"/>
<preference name="orientation" value="landscape" />
<access subdomains="true" uri="http://scholar.google.ca" />
<access subdomains="true" uri="*" />
<icon src="images/icon.png" />
<content src="index.html" />

</widget>

```

Step 2: Removing BlackBerry Specific Application code

- In your index.html (from lab 4), find the visitcmer button and change the onclick section to:

```
onclick="window.location.href='http://cmer.uoguelph.ca/'"
```

- In script.js look for the total cites function. For the ‘viewpapersbutton’ change the onclick to

```
html += "<br/><br/><div id='viewpapersbutton'><input type='button'
onclick=window.location.href='"+openpageurl+"' value='View Papers by the
Author'/></div>";
```

- That’s it! When working cross-platform, its best to adhere to standards and not use any device specific API’s. With PhoneGap, there are many API’s you can use in place of device specific ones, but with this app, they were not needed.

Step 3: Compiling and Signing your Android Application

When an application is submitted to any application market, it must first be signed with a digital certificate. We will first create a certificate and then compile our signed application.

- The steps to creating a signing key can be seen in detail at <http://developer.android.com/guide/publishing/app-signing.html> we will go through them to make your key
- We will be using the keytool that is part of your JDK. Open a command prompt and navigate to your Keytool location similar to

```
C:\Program Files (x86)\Java\jdk1.6.0_30\bin
```

- Enter the following command. This will begin the generation process to produce a file called my-release-key.keystore using an RSA algorithm that is valid for 10 000 days

keytool -genkey -v -keystore my-release-key.keystore -alias myKey -keyalg RSA -keysize 2048 -validity 10000

- You will be asked various questions and to provide two passwords which may or may not be different. Do not lose this password.

```

C:\Windows\system32\cmd.exe
C:\>keytool -genkey -v -keystore my-release-key.keystore -alias myKey -keyalg RSA
A -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: John Doe
What is the name of your organizational unit?
[Unknown]: CMER
What is the name of your organization?
[Unknown]: University of Guelph
What is the name of your City or Locality?
[Unknown]: Guelph
What is the name of your State or Province?
[Unknown]: Ontario
What is the two-letter country code for this unit?
[Unknown]: CA
Is CN=John Doe, OU=CMER, O=University of Guelph, L=Guelph, ST=Ontario, C=CA correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA1withRSA) with
a validity of 10,000 days
for: CN=John Doe, OU=CMER, O=University of Guelph, L=Guelph, ST=Ontario,
C=CA
Enter key password for <myKey>
<RETURN if same as keystore password>:
[Storing my-release-key.keystore]

```

- Now that you have your signing key go to <https://build.phonegap.com> and register a new account
- Once in the user interface, click your email address in the top right corner and select the signing tab. Under Android, select Add a Key.
- Create a title of your choosing and select your keystore file. Enter the alias we chose above, which was “myKey” and your password and click create. Set this key as default.

Android Key

title
Android Key

keystore file
Choose File my-release-key.keystore

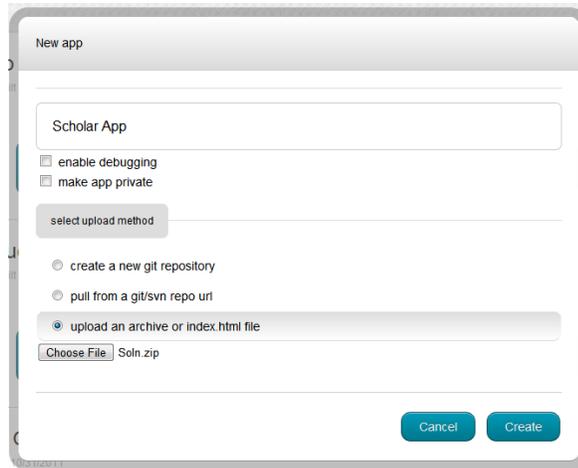
alias
myKey

key password
.....

keystore pw
.....

cancel create

- Now back on the main page, click the New App button. Set a name for your application and select upload an archive. Choose a .zip file of your application source code and click create. This Zip file should be created in the same manner you did for WebWorks.



- Your app will start building immediately and will allow you to download an Android APK file once complete.

Step 4: Setup the Testing Environment

- Please install the Android SDK starter package available at: <http://developer.android.com/sdk/index.html>. Additional instructions on how to setup and start using the SDK can be found here if needed but we will go through the steps: <http://developer.android.com/sdk/installing.html>.
- You will not need to setup Eclipse as we will only be using the SDK for the device emulator. Download and install the windows executable and when asked to start the SDK Manager, say yes.
- The SDK Manager will now ask you which parts of the SDK you would like to download to test on, we will select Android 4.0.3 and install all packages
- Once complete, in your start menu or the Android-SDK installation folder you will find the AVD Manager, open this program. This application allows us to create a virtualized Android device.
- In the AVD Manager, create a new device from the new button. Call it Android. Select the target as Android 4.0.3 and click create. Now click your new device and press the start button.

Step 5: Testing

- Once your Android Emulator has loaded, open a command prompt and navigate to the platform-tools directory of your Android SDK
- Confirm that your device is found by typing

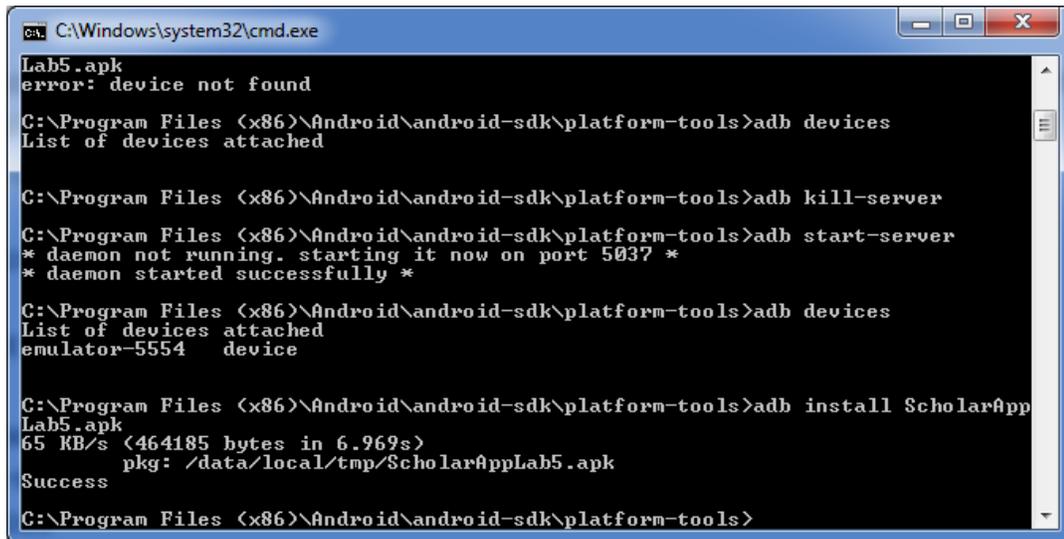
adb devices

If it is not attached, do the following:

```
adb kill-server
adb start-server
adb devices
```

Now that your device is listed, ensure you copy your apk file to the platform-tools directory and run the command:

```
adb install ScholAppLab5.apk
```



```
C:\Windows\system32\cmd.exe
Lab5.apk
error: device not found

C:\Program Files (x86)\Android\android-sdk\platform-tools>adb devices
List of devices attached

C:\Program Files (x86)\Android\android-sdk\platform-tools>adb kill-server

C:\Program Files (x86)\Android\android-sdk\platform-tools>adb start-server
* daemon not running. starting it now on port 5037 *
* daemon started successfully *

C:\Program Files (x86)\Android\android-sdk\platform-tools>adb devices
List of devices attached
emulator-5554    device

C:\Program Files (x86)\Android\android-sdk\platform-tools>adb install ScholAppLab5.apk
65 KB/s (464185 bytes in 6.969s)
Success
C:\Program Files (x86)\Android\android-sdk\platform-tools>
```

- Now test your completed application in the emulator. This apk can also be installed on any android device by simply downloading it to the device and opening the file. Additionally, it is ready to be submitted to the Android Market for others to download or purchase since you have signed it.
- If you find problems with the user interface on a certain devices, try going back into your code and optimizing the CSS to detect device resolution and aspect ratio, then display accordingly.
- The lab is now completed. You now have a fully functioning Android application. Part B of this lab will guide you through the process of building this application for iOS.