1 HelloWorld (If already done with CLI, then jump to section 2)

this lab session is about running phonegap for Android on Eclipse. We will be using phonegap 5.4 and its CLI, open this URL (or alternatively http://docs.phonegap.com/en/edge/).

We will not making use of eclipse for this lab session.

1.1 Set the JAVA_HOME and Android SDK path

Open a shell, and run

- export JAVA_HOME=/usr/jdk/latest
- export PATH=$PATH:/packages/mobserv/android-sdk/tools:/packages/mobserv/android-sdk/platform-tools/

Note that the env variable is only applied to the current shell, so if you open another shell, you need to export again the JAVA_HOME and PATH vars.

1.2 Create an Android Application Project

Create a directory for your phonegap applications, and create your app

- mkdir phonegap
- cd phonegap
- cordova create hello fr.eurecom.hello HelloWorld

1.3 Sanity check

- cd hello
- cordova platform add android (to test run: “phonegap cordova platform ls”)
- cordova build android : this executes the following two commands
    - cordova build prepare
    - cordova build compile
    - option: --debug for debug build or –release for release build
- open an android emulator
    - android avd, and then select /build the image, and launch it, you can also do this using IDE)
- cordova emulate android
    - you should see the layout
    - --stacktrace to see the trace of errors
    - --info or –debug : adjust the log verbosity

check that the device is ready: adb logcat explore the structure of the www/index.htm file.

1.4 Add permission to the Manifest file

Open an editor (gedit, emacs, kdevelop, or kate), and add the following permission to the manifest file located in platforms directory.

```
<uses-permission android:name="android.permission.VIBRATE"/>
```

1.5 Update the project

We will only modify the files inside the directory www. Open the following files using an editor:

- www/index.html and www/css/index.css

Add the following into www/index.html at the end <body> after the last script.

```
<script type="text/javascript" charset="utf-8">
//alert dialog dismissed
 function alertDismissed() {
          // do something
      }
var displayHello = function() {
var name = document.getElementById("firstname").value;
      navigator.notification.alert("Hello Dear " + name + "!",
                                                      alertDismissed,
                                                      "Hello",
                                                      "Done");
      }
</script>
<div id="txt">
<input  type="text" name="firstname" id="firstname" />
<a href="#" class="btn" onclick="displayHello(); return false;">Say Hello</a>
</div>
```

Add the following items to the end of your www/css/index.css

```
#txt {
    position:absolute;
    left:10%;
    top:10%;
    height:100px;
    width:250px;
    text-align:center;
    padding:0px;
    margin:0px;
}
input  {
 padding:3px;
 border:1px solid #dcdcdc;
 border-radius:5px;
 width:200px;
 box-shadow:1px 1px 2px #C0C0C0 inset;
 }
.btn {
       -webkit-box-shadow:inset 0px 1px 0px 0px #ffffff;
       box-shadow:inset 0px 1px 0px 0px #ffffff;
       background-color:transparent;
       -webkit-border-radius:6px;
       border-radius:6px;
       border:2px solid #dcdcdc;
       display:inline-block;
       color:#b341b3;
       font-size:12px;
       padding:6px 24px;
       text-decoration:none;
       text-shadow:1px 1px 0px #ffffff;
}
```

Run the project and check if the native-Android alert works as expected. You must see a notification by clicking "Say Hello".

- cordova emulate android

Check if the “notification” plugin is installed by running

- `cordova plugin list`

if not search for appropriate pluging using the following command:

- `cordova plugin search notification`
- `cordova plugin search vibration`

Then install

- `cordova plugin add cordova-plugin-dialogs (old cordova plugin add org.apache.cordova.dialogs)`
- `cordova plugin add cordova-plugin-vibration  (cordova plugin add org.apache.cordova.vibration)`

check the installation

- `cordova plugin list`

Check if the feature “notification” is enabled in config.xml and if not add the following

```
<feature name="Notification">
<param name="android-package" value="org.apache.cordova.dialogs.Notification" />
</feature>
<feature name="Vibration">
<param name="android-package" value="org.apache.cordova.vibration.Vibration" />
  </feature>
```

Test again. You should see the native Android dialog box appreaing, and saying hello !

Check all the available cordova CLI options

- Cordova help
- Or https://cordova.apache.org/docs/en/4.0.0/guide/cli/
- See https://cordova.apache.org/docs/en/3.0.0/cordova/notification/notification.html

The notification does not work because it violates the “Content-Security-Policy”. Change the policy as follows in the index.htm

```
<meta http-equiv="Content-Security-Policy" content="default-src *; script-src &apos;self&apos; &apos;unsafe-inline&apos; &apos;unsafe-eval&apos; *; style-src  &apos;self&apos; &apos;unsafe-inline&apos; &apos;unsafe-eval&apos; *">
```

And also instead of using href (i.e. <a href="#" class="btn" onclick="displayHello()">Say Hello</a> --cont>) , which potentially can access remote content, use a simple button as follows:

```
<button onclick="displayHello()">Say Hello</button>
```

1.6 Question

- How the native android dialog box is called
- How to use the phonegap dialog box?

2 IP localization (if section 2.2 is done, jump to the Lab 2)

Build an application that show the localization from your IP Address (in contrast to GPS-based), querying a free REST webservice.

Note: Although this can also be done without phonegap, through this example, one could incorporate such functionality within a more complex context-aware phonegap application.

In this application sample, the javascript code is isolated from the HTML code. Before sending a request to the Webservice, the application must check the connectivity.

In this example, the JavaScript code governs the display by dynamically injecting tags to the HTML DOM elements. See the comments inside the code.

2.1 Create the project with the help of CLI

In your phonegap home directory, **phonegap**, create a new project

- cordova create showiplocation fr.eurecom.showiplocation ShowIPLocation
- cd showiplocation

Overwrite the following files from /packages/mobserv/phonegap/showiplocation/www/ into your local www directory: css/ js/ index.html

If network connection errors happen, proceed as follows:

- check if the network information plugin is installed for your project. If not install it
- Add the following feature in your config.xml

```
  <feature name="NetworkStatus">
  <param name="android-package"
value="org.apache.cordova.networkinformation.NetworkManager" />
</feature>
```

- And the following permission into your platforms/android/AndroidManifest.xml

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Run and test if the raw information could be obtained. To see the logcat output, open a new terminal, and run “adb logcat”.

To start debugging, open the js/index.js, and follow the application flow and check when formatText function is called.

2.2 Question:

- Using a figure, explain how the application starts and interact with the files in www?
- Get and view the data in the following formats: raw, xml, and json
- Check the soap-based dictionary service. When a word is searched, the results are given in json. Extent the code to show all the available definitions.
- How to create a release version of the application?

Write up a short report comparing the following

- Compare raw, xml, and json formats
- Compare the RESTFull versus Soap based web service

3 Links:

- http://phonegap.com/developer
- https://developer.mozilla.org/en-US/docs/JavaScript
- http://www.w3.org/TR/REC-DOM-Level-1/
- https://developer.mozilla.org/en-US