**Email Composer plug-in documentation**

**Objective:**

Email Composer Phonegap application allows any agent to compose email using the existing email application in his device.

This plugin is compatible with PhoneGap 2.8.1.

**Getting Started :**

Before you start to look at the functionality of twitter plug-in make sure that you have following softwares installed:

* android-sdk
* phonegap-2.8.1
* sencha touch 2.0.1.1.

Apart from above mentioned prerequisites, we also must have at least one email application already installed on to the device.

**Part 1: Creating StandAlone plugin In Eclipse:**

**Steps for Email plugin functionality**

1. Create a phone gap project in the eclipse of the android sdk.For that we must have android sdk installed.

You can download the android sdk (ADT BUNDLE) form the following link:

<http://developer.android.com/sdk/index.html>

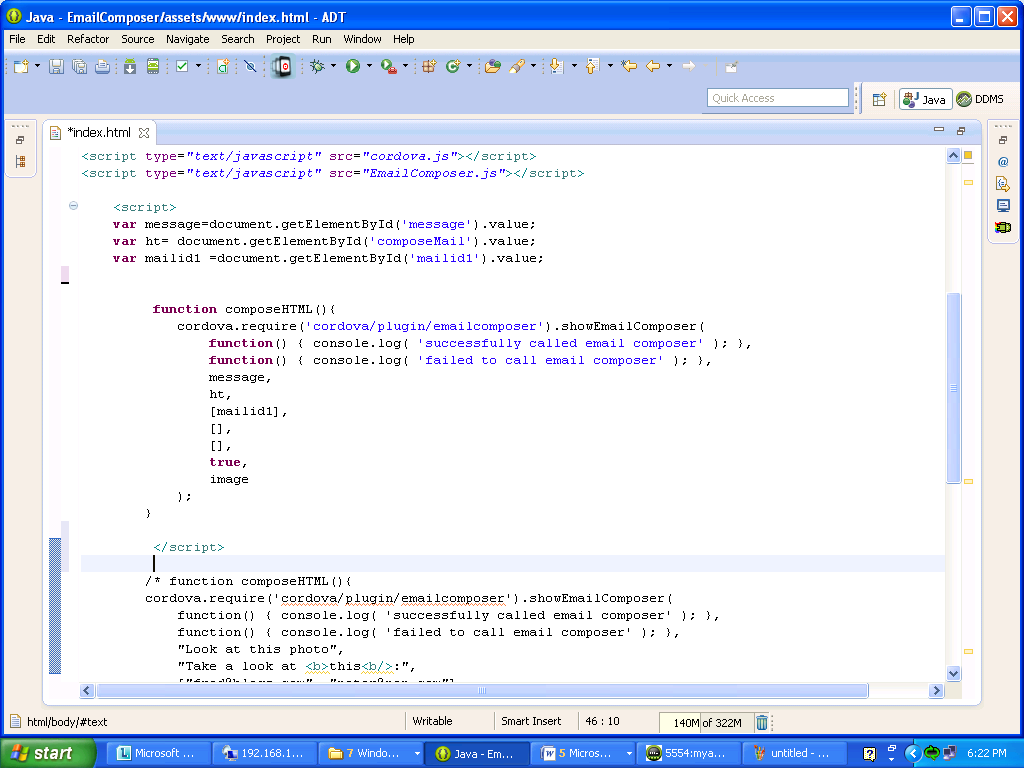
1. **To install the plugin, move www/EmailComposer.js to your project's www folder and include a reference to it in your html file after phonegap.js.**  
     
       <script type="text/javascript" charset="utf-8" src="phonegap.js"></script>  
       <script type="text/javascript" charset="utf-8" src="EmailComposer.js"></script>
2. Create a package(directory) within your project called "src/com/phonegap/plugin/email" and move EmailComposer.java into it.
3. In the assets folder,  there exist a folder named as www,  move the **EmailComposer.js**file in there.

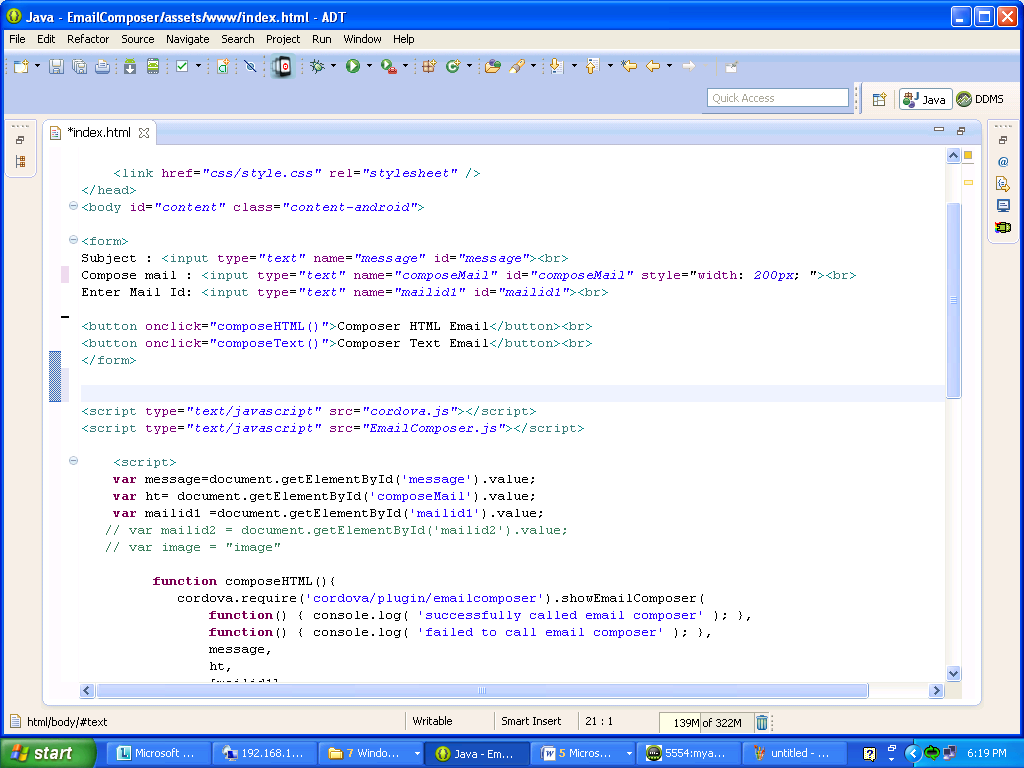
5. In your res/xml/plugins.xml file add the following line:  
  
    <plugin name="Email Composer" value="com.phonegap.plugin.email.EmailComposer"/>

6. **## Using the plugin ##**

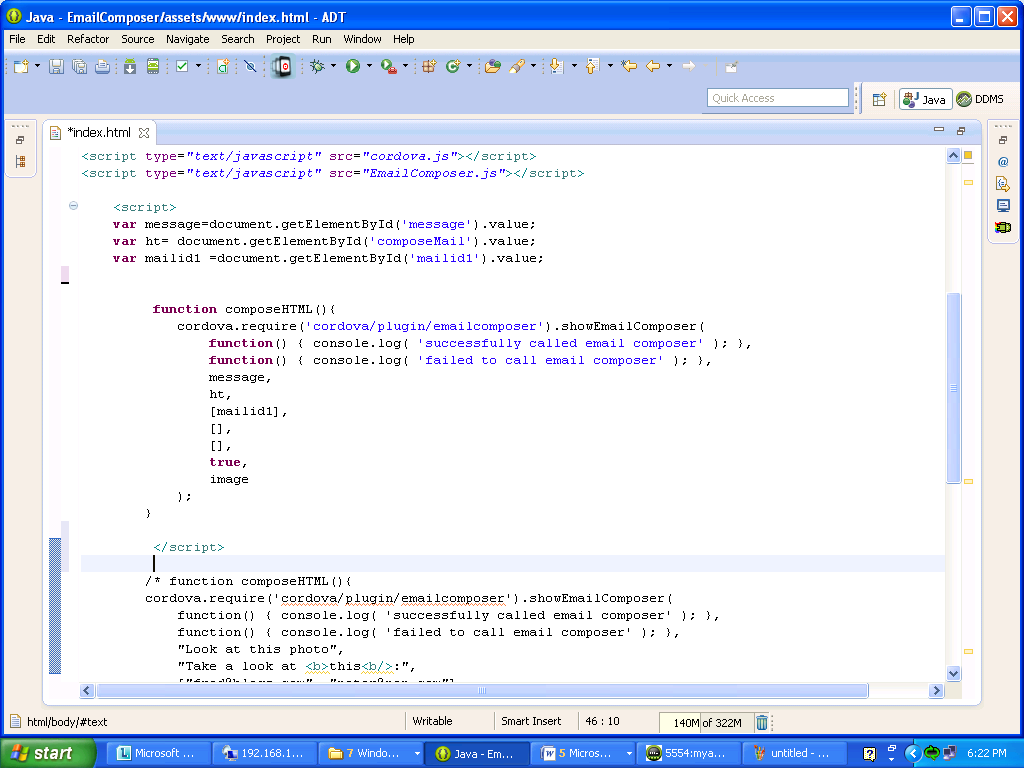
You can call the methods in Twitter.java using following available methods from script tag of your index.html  
  
The plugin creates the object `'cordova/plugin/emailcomposer'. To use, call one of the following, available methods:

**Index.html:**

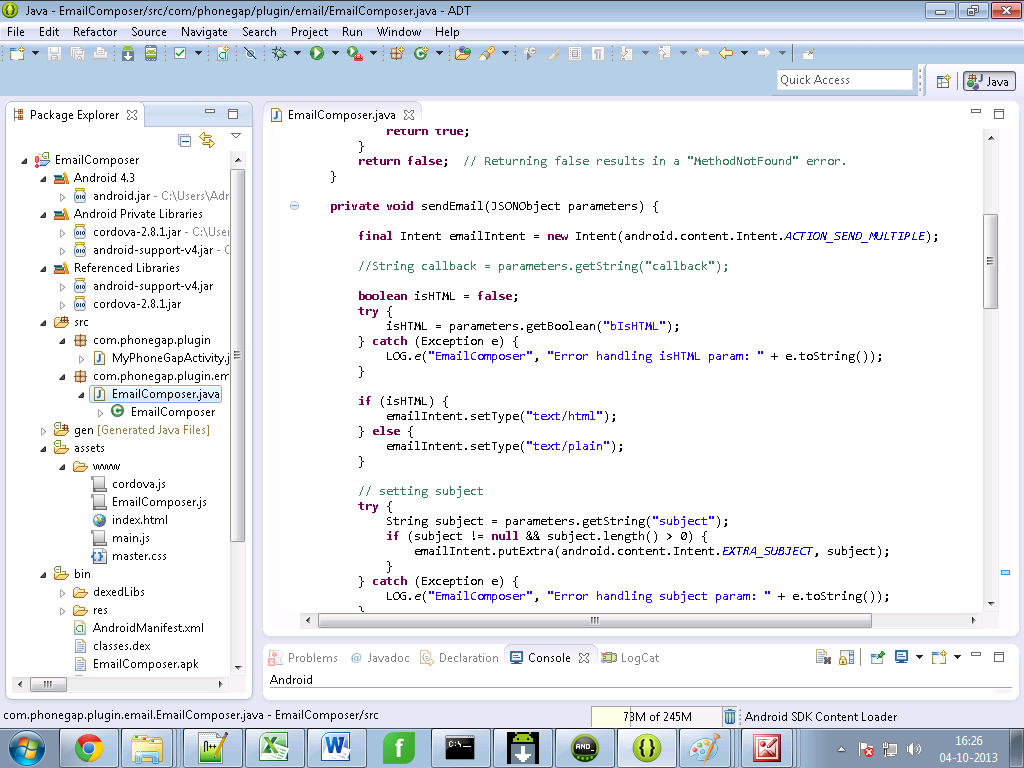
****

****

Call the method in EmailComposer.js:



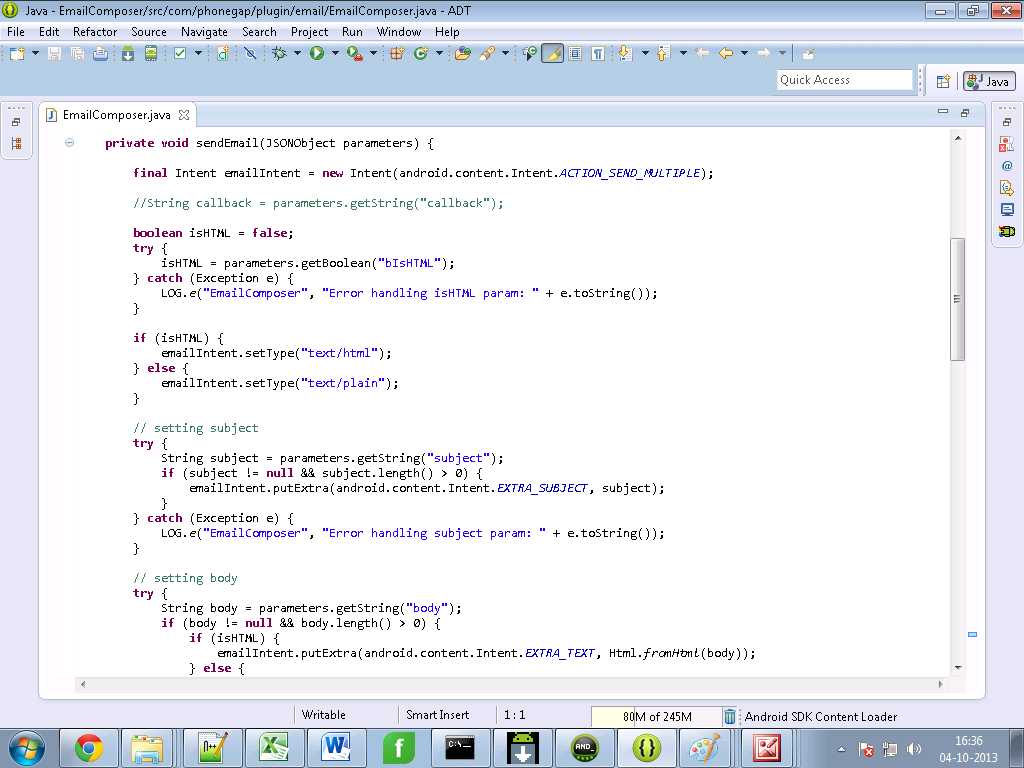
**Directory structure:**



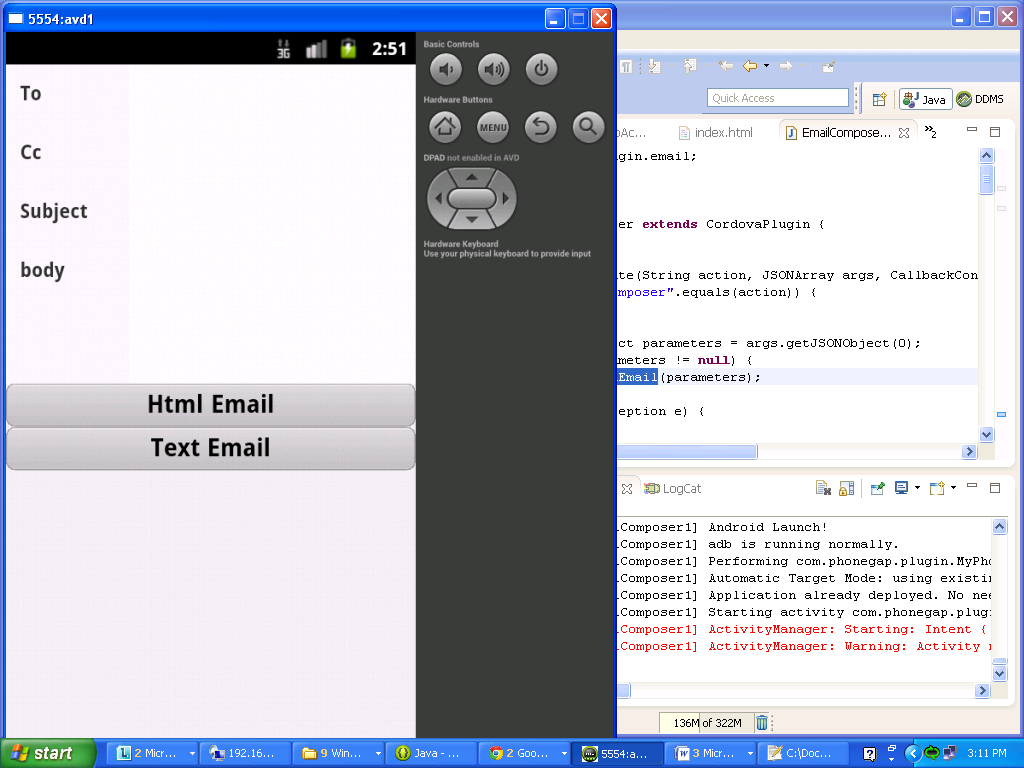
7. In the same www folder a file named as index .html exists which helps in rendering the view.

**Methods in EmailComposer.java:**

8. Screen shot showing the content of sendEmail() function .This function sends the message by setting the receiver(to) , subject(defines title of the message)and the body of the message in the default compose page of the existing email application of the phone.



9. You can now run the project.



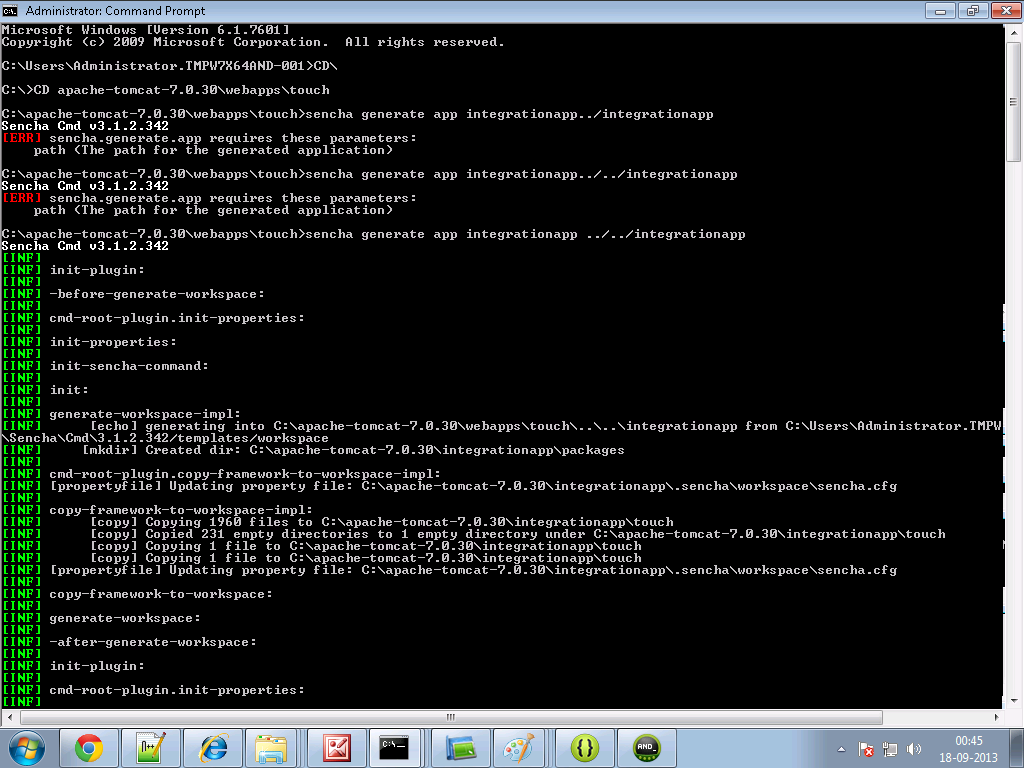
**Part2.**

**Creating Plugin in sencha and Integrating Email Plugin with Eclipse code**

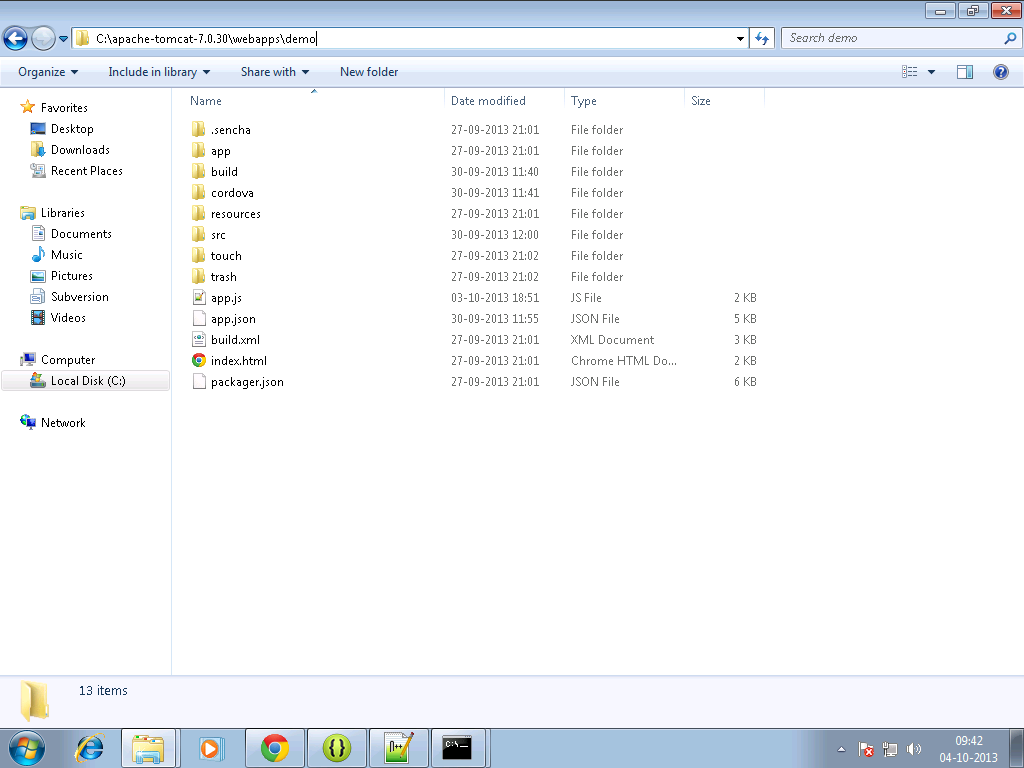
**Steps for email plugin functionality**

1. **create a new application with the help of following command in cmd:**

**C:\apache-tomcat-7.0.30\webapps\touch>** **sencha generate app <appname> ../<appname>**



1. **Create a folder and name it as cordova in your sencha <appname> directory. For example**



**3.Copy the cordova.js ,cordova.js\_android, cordova.js\_ios, cordova.js\_plugins and**

**EmailComposer.js in cordova folder.**

**4.Make an entry in the js tag of app.json file residing in your<appname> directory as follows:**

js": [  
  {  
            "path": "cordova/cordova.js",  
            "update": "delta"  
        },

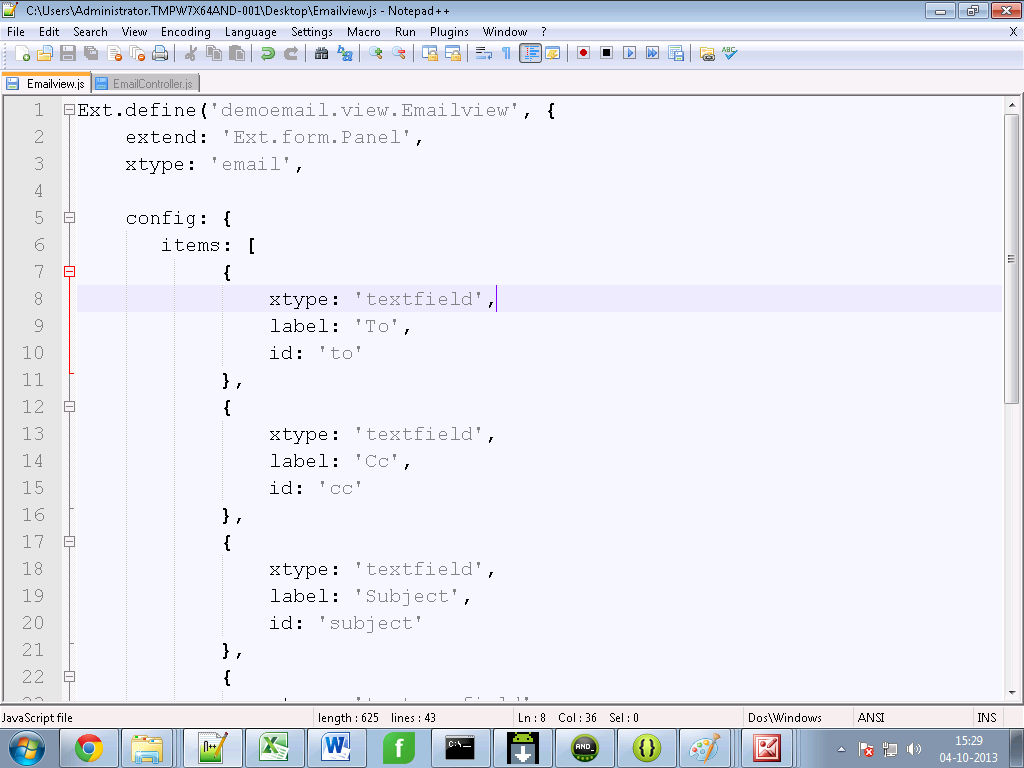
{  
            "path": "cordova/EmailComposer.js",  
            "update": "delta"

}

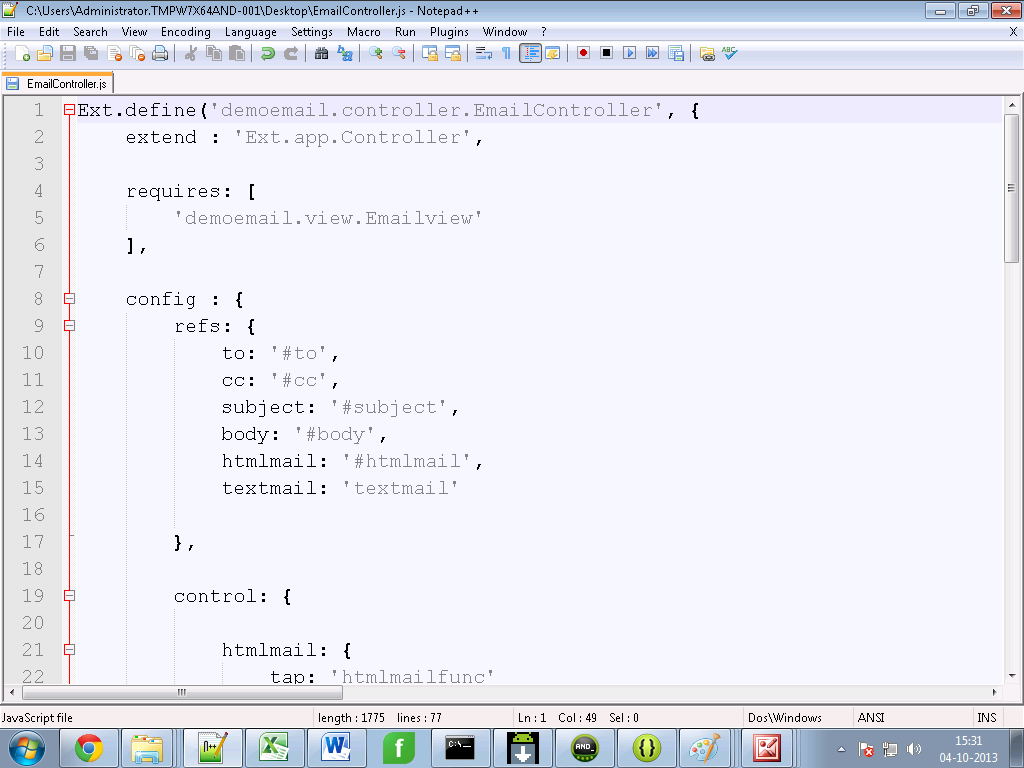
]

**5. Write the sencha code for view and controller in your sencha <appname>.**

**Emailview.js**

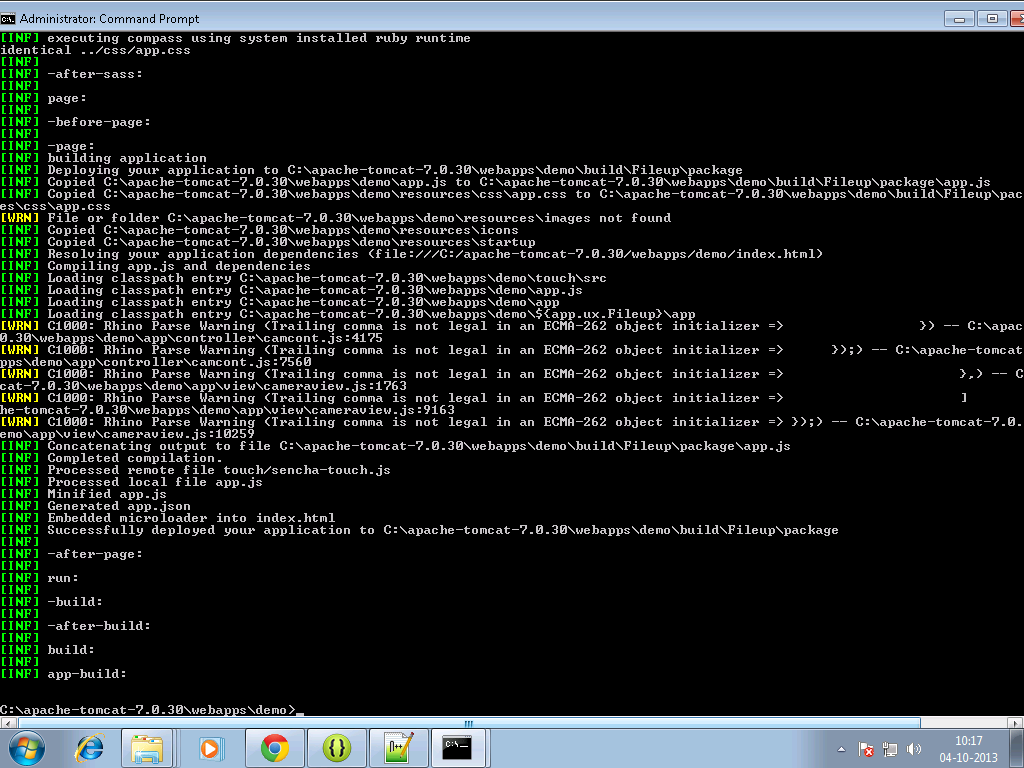


**EmailController.js**



**6.** **In command prompt go to your sencha <appname> directory first, then Build the package of your sencha app by typing the following command :**

**Sencha app build package**

****

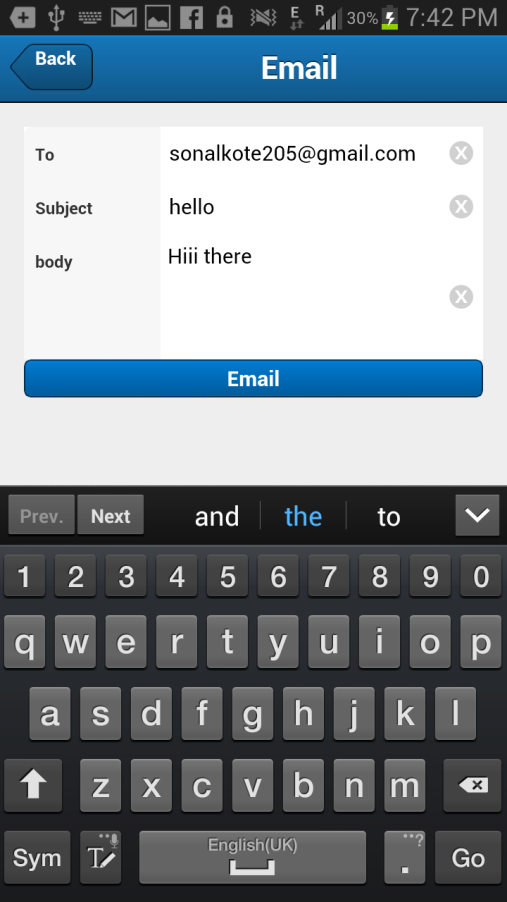
**7. Move to your sench<app> directory ,there a build folder will get created.**

**8. Copy files from package folder under <sencha appname>/build/ sencha < appname> to your www folder of android project for example:**

**C:\temp\TempAndroid\assets\www**

**9. build it - > apk will be in bin folder**

**Output on Device:**

****

