**Barcode scanner plug-in documentation**

**Objective:**

Barcode Scanner Phonegap application allows any agent to scan the barcode and get the code embedded in it.

This plugin is compatible with PhoneGap 2.8.1.

**Getting Started :**

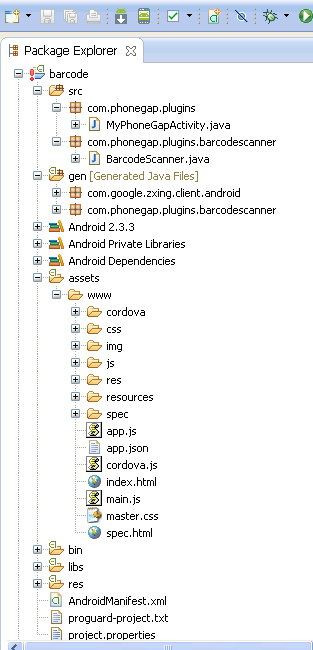
Before you start to look at the functionality of twitter plug-in make sure that you have following softwares installed:

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

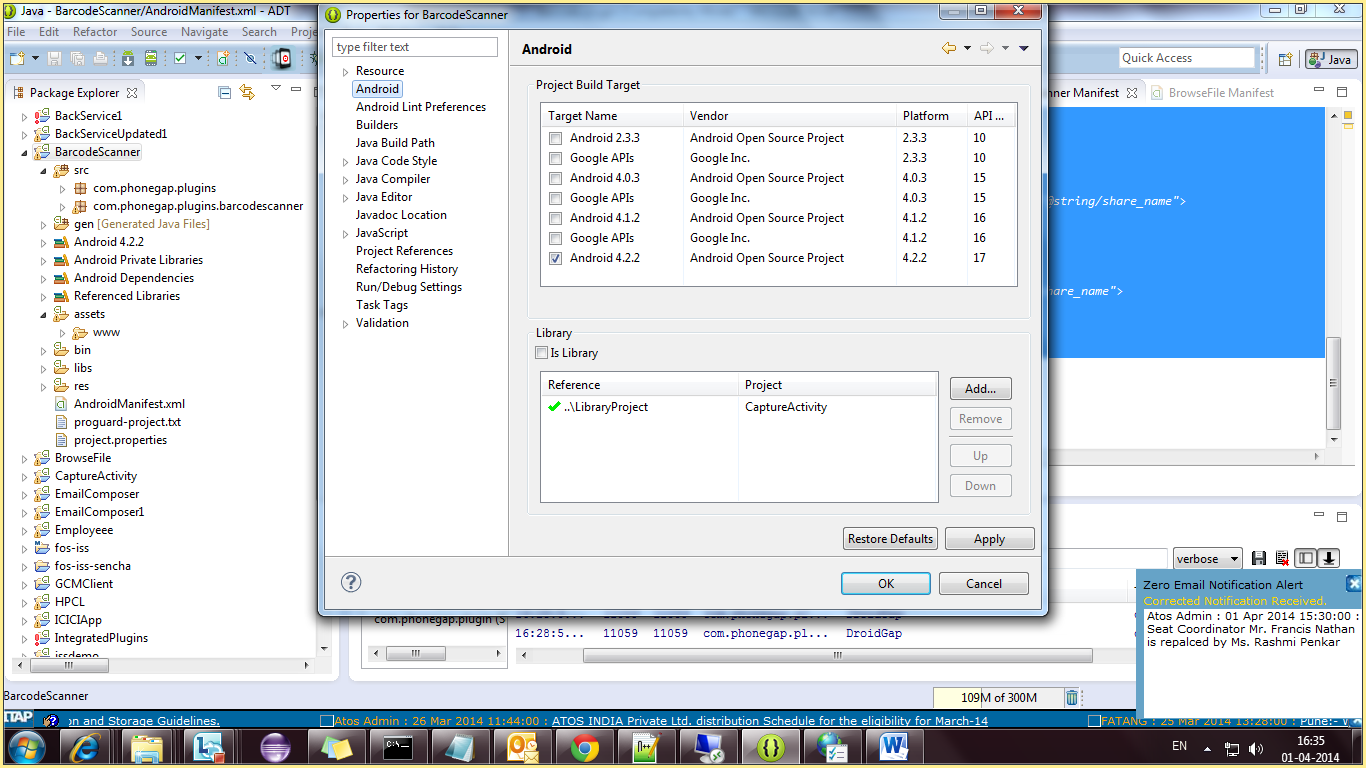
**Part 1: Creating plugin In Eclipse:**

**Steps for barcode scanner plug-in functionality**

1. Add the 'LibraryProject' into Eclipse. File -> New Android Project -> create project from existing source.



1. In the new project you've just added to Eclipse go to the project properties. Select the Android section and at the bottom of the dialog check the "Is Library" checkbox.
2. In your application go into the project properties. In the Android section under library click the Add button and select the library you created in step 2.



1. To install the plugin, move barcodescanner.js to your project's www folder and include a reference to it in your html files.
2. Create a folder called 'com/phonegap/plugins/barcodescanner' within your project's src folder.
3. And copy the BarcodeScanner.java file into that new folder.

mkdir /src/com/phonegap/plugins/barcodescanner

cp ./src/com/phonegap/plugins/barcodescanner/BarcodeScanner.java /src/com/phonegap/plugins/barcodescanner

1. In your res/xml/config.xml file add the following line:

<plugin name="BarcodeScanner" value="com.phonegap.plugins.barcodescanner.BarcodeScanner"/>

1. And make sure you have the following permissions in your AndroidManifest.xml file:

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

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

1. Add the following activity to android manifest

<activity android:name=*"com.google.zxing.client.android.CaptureActivity"*

android:screenOrientation=*"landscape"*

android:clearTaskOnLaunch=*"true"*

android:configChanges=*"orientation|keyboardHidden"*

android:theme=*"@android:style/Theme.NoTitleBar.Fullscreen"*

android:windowSoftInputMode=*"stateAlwaysHidden"*

android:exported=*"false"*>

<intent-filter>

<action android:name=*"com.phonegap.plugins.barcodescanner.SCAN"*/>

<category android:name=*"android.intent.category.DEFAULT"*/>

</intent-filter>

</activity>

<activity android:name=*"com.google.zxing.client.android.encode.EncodeActivity"* android:label=*"@string/share\_name"*>

<intent-filter>

<action android:name=*"com.phonegap.plugins.barcodescanner.ENCODE"*/>

<category android:name=*"android.intent.category.DEFAULT"*/>

</intent-filter>

</activity>

<activity android:name=*"com.google.zxing.client.android.HelpActivity"* android:label=*"@string/share\_name"*>

<intent-filter>

<action android:name=*"android.intent.action.VIEW"*/>

<category android:name=*"android.intent.category.DEFAULT"*/>

</intent-filter>

</activity>

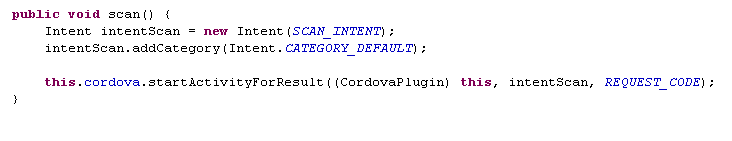
**Using the plugin**

The plugin creates the object window.plugins.barcodeScanner with the method scan(success, fail). The following barcode types are currently supported:

* QR\_CODE
* DATA\_MATRIX
* UPC\_E
* UPC\_A
* EAN\_8
* EAN\_13
* CODE\_128
* CODE\_39
* CODE\_93
* CODABAR
* ITF
* RSS14
* PDF417
* RSS\_EXPANDED

success and fail are callback functions. Success is passed an object with data, type and cancelled properties. Data is the text representation of the barcode data, type is the type of barcode detected and cancelled is whether or not the user cancelled the scan.

A full example could be:

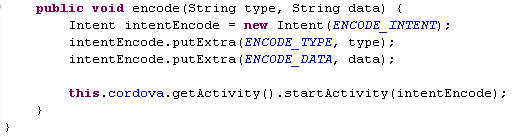
****

**Encoding a Barcode**

The plugin creates the object window.plugins.barcodeScanner with the method encode(type, data, success, fail). Supported encoding types:

* TEXT\_TYPE
* EMAIL\_TYPE
* PHONE\_TYPE
* SMS\_TYPE

A full example could be:



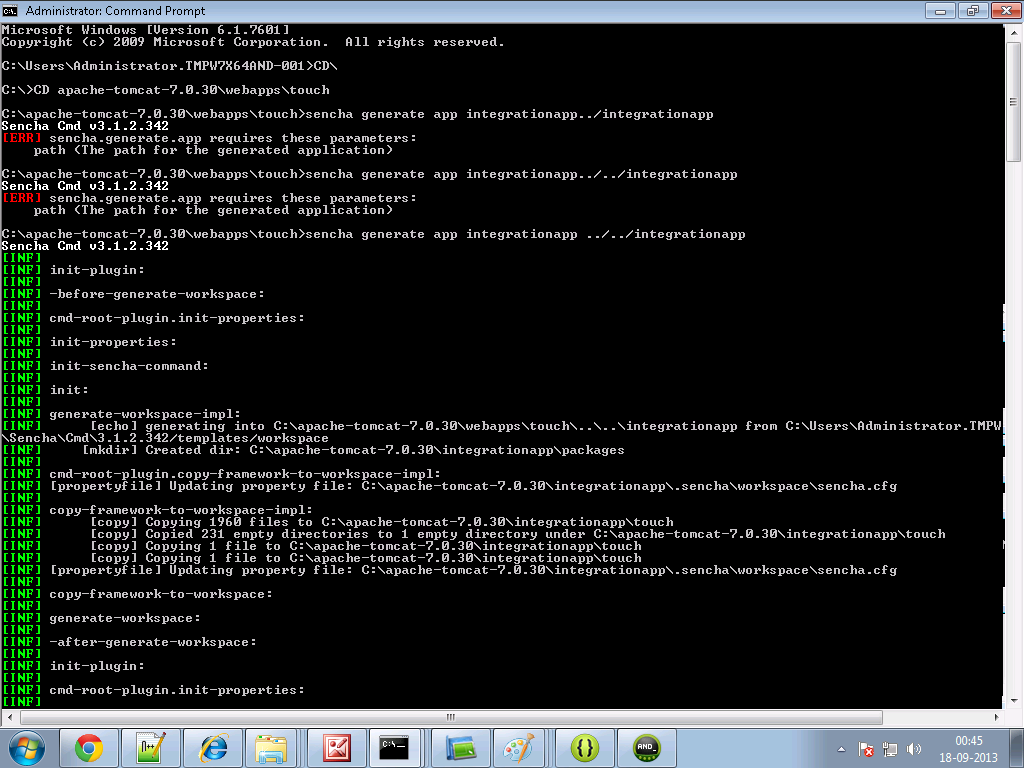
**Part2.**

**Integrating Barcode Plugin with sencha**

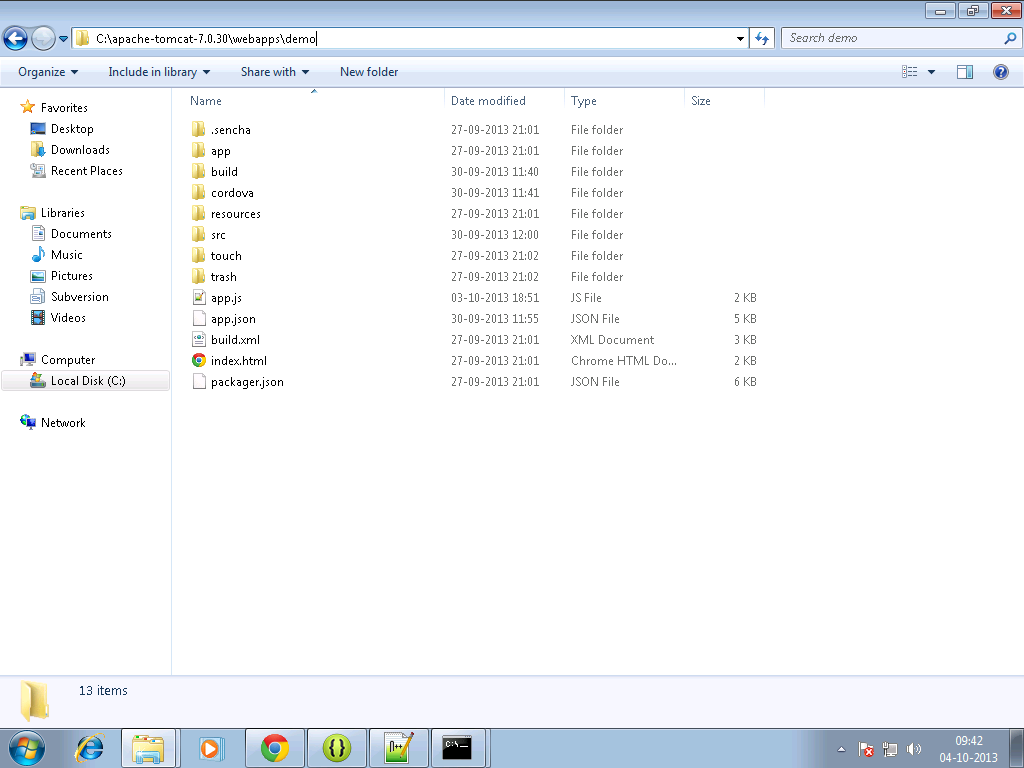
**Steps for barcode plugin functionality**

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**



1. **Copy the cordova.js ,cordova.js\_android, cordova.js\_ios and cordova.js\_plugins in them.**
2. **Make an entry in the js tag of app.json file residing in your<appname> directory as follows:**

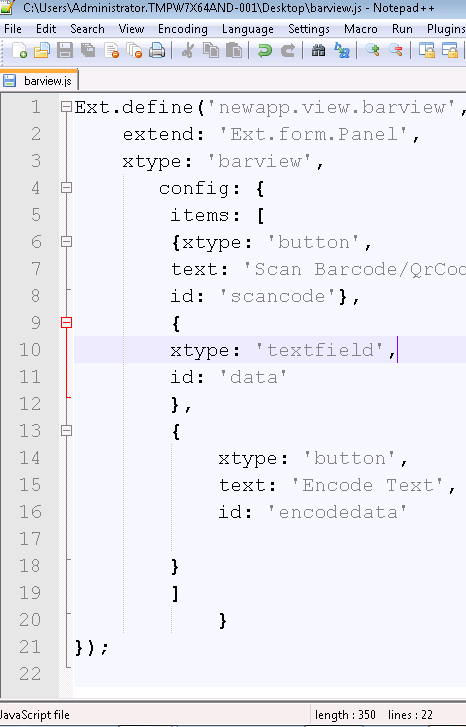
js": [  
  {  
            "path": "cordova/cordova.js",  
            "update": "delta"  
        },

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

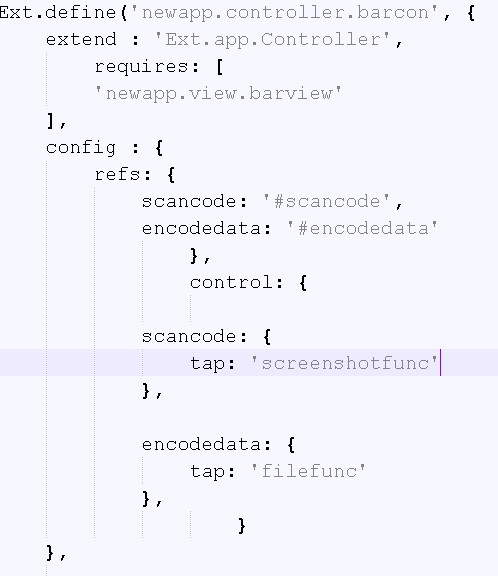
]

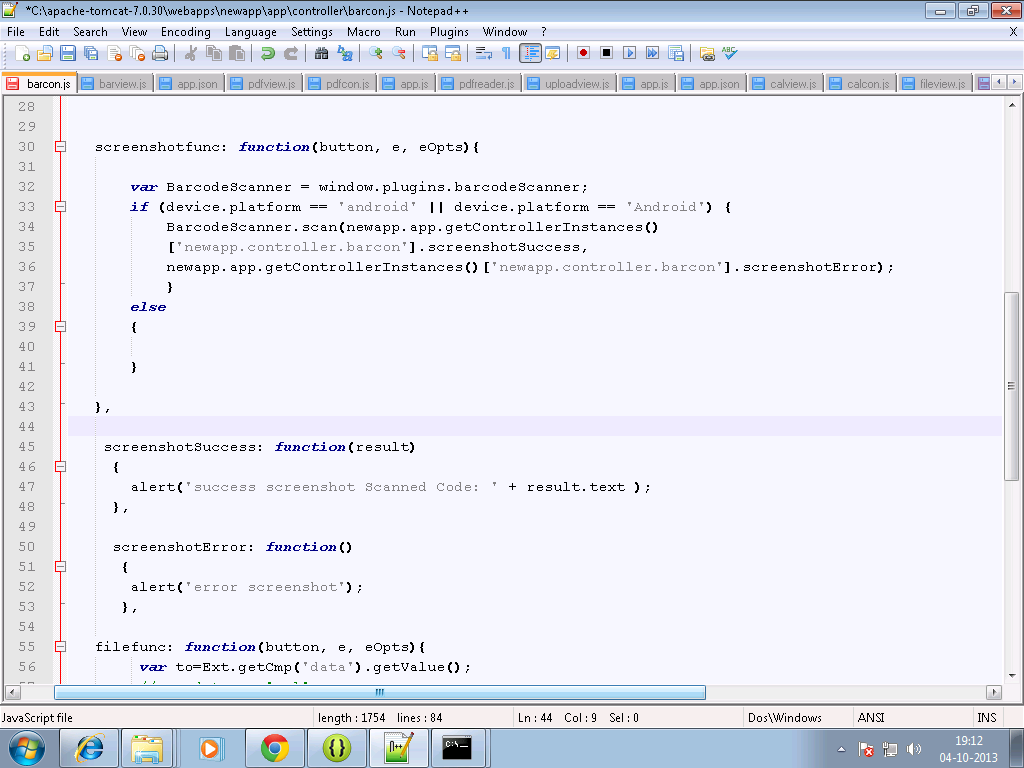
1. **Write the sencha code for view and controller in your sencha <appname>.**

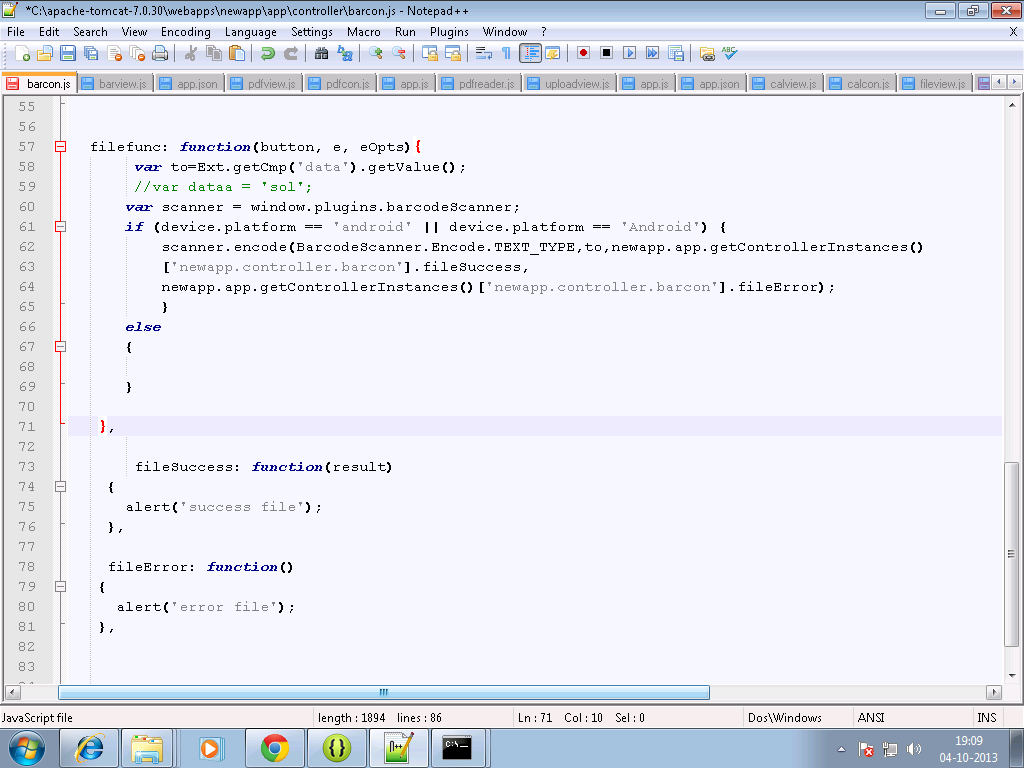
**barview.js**



**Barcon.js:**

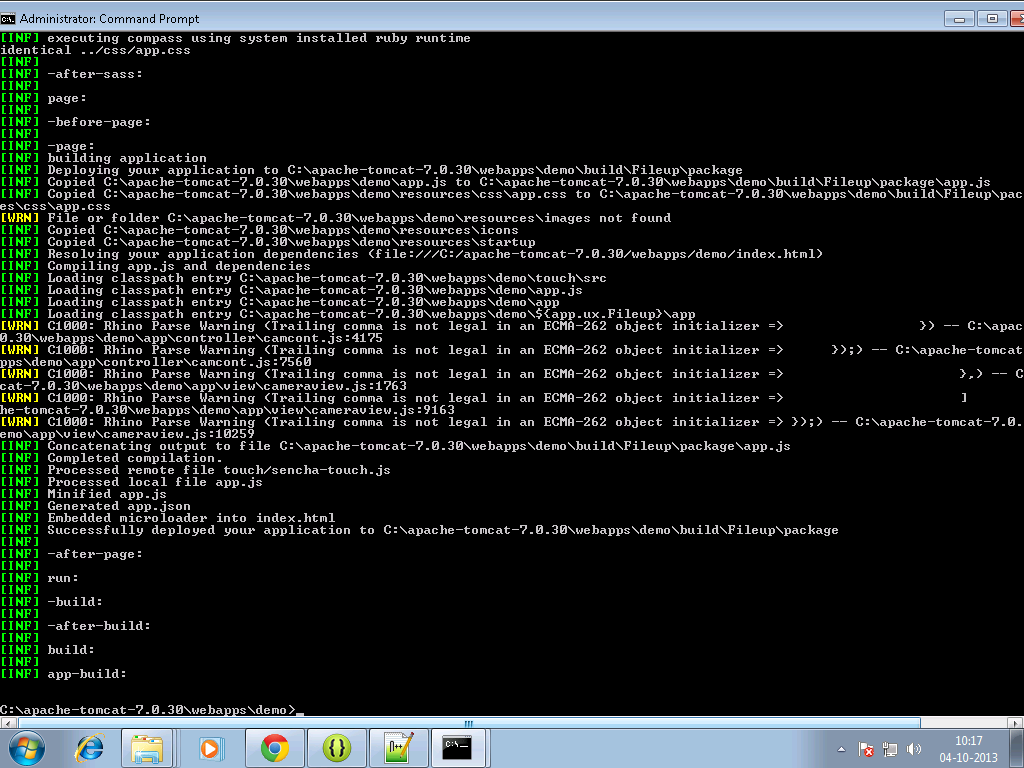






**6.** **In command prompt go to your sencha <appname> directory first, then Build the package of your sencha app by typing the following command :**

**Sencha app build package**



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

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

Copy files from **C:\ Desktop\barcode plugin\pluginDemo\build\pluginDemo\package in sencha**

**to**

**C:\temp\BarcodeScanner\assets\www in eclipse adt**

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

Output on Device:

