

Create Android Native Menu in Phonegap

TechRounder PDF Edition

Live article: <https://www.techrounder.com/phonegap/create-android-native-menu-in-phonegap/>

By Vipin PG | Published April 26, 2013 | Updated December 18, 2018 | Format: Article | 2 min read

In brief

Here is the step to create native android menu in phonegap.

Here is the step to create native android menu in phonegap.

Step 1 : Add the code in your activity class file 'public boolean onCreateOptionsMenu(Menu menu) { // Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.activity_main, menu); return true; }'

It looks like as follows 'public class MainActivity extends DroidGap { @Override public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); super.loadUrl("file:///android_asset/www/index.html"); } @Override public boolean onCreateOptionsMenu(Menu menu) { // Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.activity_main, menu); return true; } }'

Step 2 : Now create an activity_main.xml file in "res/menu/" folder

Add the following code in it.

Step 3 : Now define the strings in the strings.xml file at "res/values/" location

Step 4 : Also define the ids and string in R.java file at "gen/com.example.testapp/", normally the ids and all other variables will be automatically added to R.java file, so you only needed to add it manually if shows error.

```
'public final class R { public static final class attr { } public static final class drawable { public static final int ic_launcher=0x7f020000; } public static final class id { public static final int menu_exit=0x7f080003; public static final int menu_home=0x7f080001; public static final int menu_settings=0x7f080002; } public static final class layout { public static final int activity_main=0x7f030000; } public static final class menu { public static final int activity_main=0x7f070000; } public static final class string { public static final int app_name=0x7f050000; public static final int hello_world=0x7f050001; public static final int menu_exit=0x7f050005; public static final int menu_home=0x7f050004; public static final int menu_settings=0x7f050002; } public static final int really_quit=0x7f050007; } }
```

Step 5: Now add code in your main activity so that the menu action to be done when clicking them.

```
'@Override public boolean onOptionsItemSelected(MenuItem item) { // Handle item selection switch (item.getItemId()) { case R.id.menu_home: ////code to be executed return true; case R.id.menu_settings: ////code to be executed return true; case R.id.menu_exit: ////code to be executed return true; default: return super.onOptionsItemSelected(item); } }
```

Now it looks like this

```
'public class MainActivity extends DroidGap { @Override public void onCreate(Bundle
savedInstanceState) { super.onCreate(savedInstanceState);
super.loadUrl("file:///android_asset/www/index.html"); } @Override public boolean
onCreateOptionsMenu(Menu menu) { // Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.activity_main, menu); return true; }'

public boolean onOptionsItemSelected(MenuItem item) { // Handle item selection switch (item.getItemId())
{ case R.id.menu_home: ////code to be executed return true; case R.id.menu_settings: ////code to be
executed return true; case R.id.menu_exit: ////code to be executed return true; default: return
super.onOptionsItemSelected(item); } }
```

Now it works !!!