

Easy embedding Google Map

ပထမဦးဆုံး ပြောချင်တာက ကျွန်တော့်ရဲ့ ဒီဆောင်းပါးလေးဟာ ကျွန်တော့်လို တကယ့်ကို အခြေခံသမားတွေ အတွက်ပဲ ရည်ရွယ်ပါတယ်။ နားလည် တတ်ကျွမ်းပြီးသား ဆရာများအနေနဲ့ မပြည့်စုံ မတိကျမှုတွေ၊ အားနည်းချက်တွေကို သီးခံပေးပြီး ဒီထက် ပိုကောင်းတဲ့ နည်းလမ်းများကို ညွှန်ပြပေးကြပါလို့ မေတ္တာရပ်ခံပါတယ်။

လိုအပ်ချက် တစ်ခုကြောင့် မြေပုံကို Software တွေထဲမှာ embedded လုပ်ပြီး သုံးဖို့ အလွယ်ဆုံး နည်းကို ကျွန်တော် လိုက်ရှာနေခဲ့ပါတယ်။ Google နဲ့ အခြား Map API တွေ အများကြီး ရှိမယ် ထင်ပါတယ်။ ဒါပေမယ့် ကျွန်တော့်ဖုန်းမှာက Google playstore service အဆင်မပြေတာနဲ့ ဖုန်းအားလုံးအတွက် လိုအပ်ချက် အနည်းဆုံး နည်းလမ်းကို ရှာခဲ့ရတာပါ။

ကျွန်တော် သုံးခဲ့တဲ့ နည်းလမ်းကတော့ Android နဲ့ Windows app တွေမှာ WebView ကိုသုံးပြီး အဲဒီနေရာမှာ ကိုယ်လိုချင်တဲ့ မြေပုံပေါ်စေတဲ့ URL ကို load လုပ်ပြီး ဖော်ပြစေတာပဲ ဖြစ်ပါတယ်။

Simple Map without Marker

ပထမဆုံး သုံးခဲ့တဲ့ နည်းကတော့ အောက်ပါ အတိုင်းပဲ ဖြစ်ပါတယ် -

<https://www.google.com.mm/maps/@22.0311769,96.4686856,21z?hl=en%22>

ဆိုတဲ့ URL ကို latitude, longitude နဲ့ zoom တွေ ကိုယ်လိုချင်တဲ့ တန်ဖိုးတွေနဲ့ အစားထိုးပြီး သုံးတဲ့ နည်းပါပဲ။

အဲဒီမှာ 22.0311769 ဆိုတာက latitude တန်ဖိုး၊ 96.4686856 ကတော့ longitude နဲ့ 21 ကတော့ Zoom တန်ဖိုးတွေ ဖြစ်ပါတယ်။ အဲဒီတန်ဖိုးတွေကို သက်ဆိုင်ရာ programming တွေထဲမှာ String variable တွေနဲ့ ကုဒ်မှာ အစားထိုးပြီး URL ကို စိတ်ကြိုက် ပြင်ဆင်နိုင်ပါတယ်။ နမူနာကုဒ်တွေကိုတော့ Android နဲ့ ရေးပြထားပါတယ်။

ဥပမာ-

```
String lat=22.0311769;
```

```
String lng=96.4686856;
```

```
String zoom=25;
```

```
String URL = "https://www.google.com.mm/maps/@\"+lat+\", \"\"+lng+\", \"\"+zoom+\"z?hl=en%22\"";
```

Latitude နဲ့ Longitude တန်ဖိုးတွေကို Android မှာ GPS ကနေတဆင့် ရယူနိုင်ပါသေးတယ်။ အဲဒီ URL ကို Webview မှာ load လုပ်ခြင်းဖြင့် မြေပုံပေါ်လာမှာ ဖြစ်ပါတယ်။

```
webView1=(WebView)findViewById(R.id.webView2);
```

```
webView1.getSettings().setLoadsImagesAutomatically(true);
```

```
webView1.getSettings().setJavaScriptEnabled(true);
```

```
webView1.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
```

```
wv1.loadUrl(url);
```

UI မှာ ကိုယ်ဖန်တီးထားတဲ့ WebView ပေါ်မှာ မြေပုံကို ပြစေချင်ရင် နောက်ထပ် ကုဒ်တွေ ထပ်ထည့်ပေးရပါသေးတယ်။

`wv1.setWebViewClient(new MyBrowser());` ဆိုတာကို ထည့်ပေးရပြီး `MyBrowser()` class တစ်ခုလည်း ထည့်ပေးရပါဦးမယ်။

```
private class MyBrowser extends WebViewClient {  
    @Override  
    public boolean shouldOverrideUrlLoading(WebView view, String url) {  
        view.loadUrl(url);  
        return true;  
    }  
}
```

ဒါပေမယ့် ဒီနည်းမှာ အားနည်းချက်က Markers တွေ တင်ပြီး နေရာကို ညွှန်ပြခြင်း မပြုနိုင်တဲ့အတွက် နောက်တစ်နည်းကို စဉ်းစားခဲ့ရပါတယ်။

Simple Map with Markers

နောက်ဆုံး အောက်ပါ လိပ်စာက Google Developers ကနည်းကို အသုံးပြုဖို့ ဆုံးဖြတ်လိုက်ပါတယ်။

<https://developers.google.com/maps/documentation/javascript/examples/marker-simple>

အောက်က ကုဒ်တွေကတော့ အထက်ပါလင့်က နမူနာကို အခြေခံပြီး မိတ်ဆွေ တစ်ယောက်ရဲ့ အကူအညီနဲ့ ရေးခဲ့ပါတယ်။ `map2.php` လို့ အမည်ပေးလိုက်ပါတယ်။

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#map {
```

```
width: 500px;
```

```
height: 400px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div id="map"></div>
```

```
<?php
```

```
$name=addslashes($_GET['name']);
```

```
$address=$_GET['address'];
```

```
$lat=$_GET['lat'];
```

```
$lng=$_GET['lng'];
```

```
?>
```

```
<script>
```

```
function initMap() {
```

```
    var myLatLng = {lat: <?php echo $lat; ?> , lng: <?php echo $lng; ?>};
```

```
    var mapDiv = document.getElementById('map');
```

```
    var map = new google.maps.Map(mapDiv, {
```

```
        center: myLatLng,
```

```
        zoom: 15
```

```
    });
```

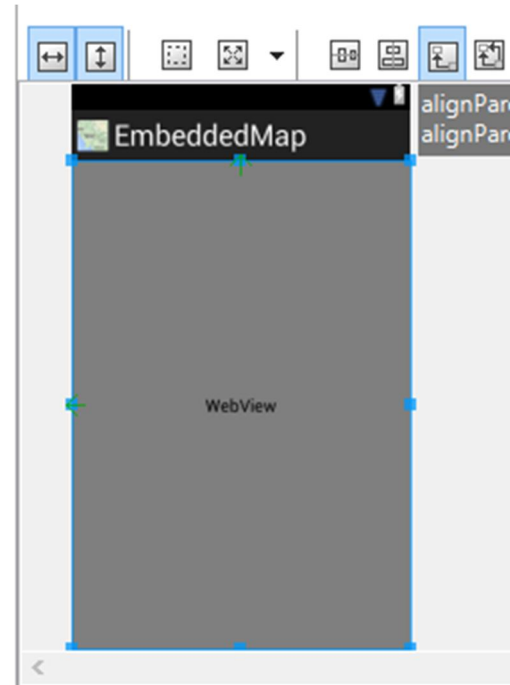
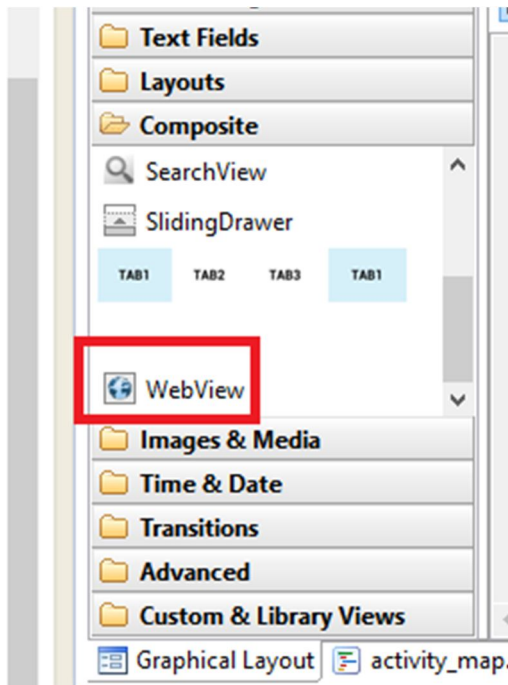
```
    var marker = new google.maps.Marker({
```

```
        position: myLatLng,
```

```
map: map,  
  
title: '<?php echo $name; ?>'  
  
});  
  
}  
  
</script>  
  
<script src="https://maps.googleapis.com/maps/api/js?callback=initMap"  
  
async defer> </script>  
  
</body>  
  
</html>
```

အဲဒီ PHP ဖိုင်ကို အင်တာနက်မှ web location တစ်နေရာမှာ သွားသိမ်းထားလိုက်ပါတယ်။
<http://mmgreenhackers.com/esdb/map2.php> ဆိုတဲ့ နမူနာနေရာလေးမှာပါ။

အဲဒီမြေပုံကို android က webview တစ်ခုပေါ်မှာ ပြသပေးဖို့ UI မှာ webview တစ်ခုကို ထည့်ပါတယ်။ webView1 ဆိုကြပါစို့။ UI ကို အောက်ပါအတိုင်း ရိုးရိုးရှင်းရှင်းပဲ တည်ဆောက်ပါမယ်။



activity_map.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
```

```
<WebView
    android:id="@+id/webView1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true" />
```

```
</RelativeLayout>
```

အင်တာနက်သုံးဖို့ permission ထည့်ပါမယ်။ အောက်က နှစ်ခုကတော့ GPS သုံးဖို့ လိုရမယ်ရ ထည့်ထားတာပါ။

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

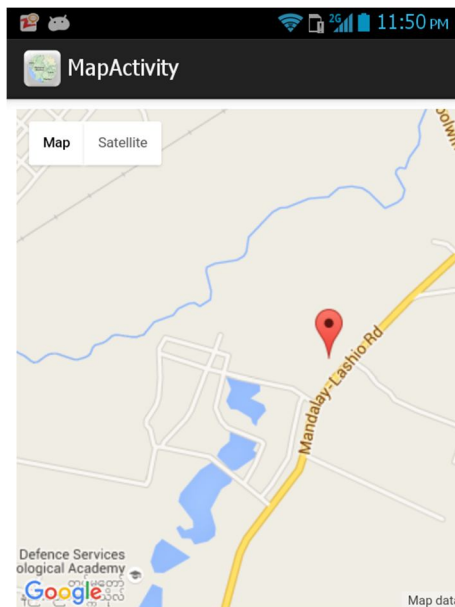
အောက်ပါကုန်တွေကို သုံးပါမယ်။ အသေးစိတ်ကို project တစ်ခုလုံး download လုပ်ပြီး လေ့လာကြည့်စေလိုပါတယ်။

```

public void showMap2(String name, String address, String lat, String lng)
{
    String url = http://mmgreenhackers.com/esdb/map2.php?name=
                + name + "&address=" + address + "&lat=" + lat + "&lng=" + lng;
    wv1 = (WebView) findViewById(R.id.webView1);
    wv1.setWebViewClient(new MyBrowser());
    wv1.getSettings().setLoadsImagesAutomatically(true);
    wv1.getSettings().setJavaScriptEnabled(true);
    wv1.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
    wv1.loadUrl(url);
}

```

အောက်ပါအတိုင်း မြေပုံကို Marker နဲ့ တကွ မြင်ရမှာ ဖြစ်ပါတယ် ခင်ဗျာ..



ကုန်တွေကို အောက်ပါ လိပ်စာမှာ MIT opensource license နဲ့ ရယူအသုံးပြုနိုင်ပါတယ် ခင်ဗျာ။

<https://github.com/AungWinnHtut/EmbeddedMap>

Dr. အောင်ဝင်းထွဋ်

<http://www.engineer4myanmar.com>