

In Class Lab: Making a google Maps Application

Start with default (set up key for app and all)

Puts you in maps_api.xml with instructions

```
<resources>
<!--
  TODO: Before you run your application, you need a Google Maps API key.

  To get one, follow this link, follow the directions and press "Create" at the end:

  https://console.developers.google.com/flows/enableapi?
  apiId=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=63:9E:62:C0:E0:B2:C2:0A:DB:23:80:D1:02:5F:E5:D6:09:9C:
  37:9A%3Bcom.library1.example.perkins.maps16

  You can also add your credentials to an existing key, using this line:
  63:9E:62:C0:E0:B2:C2:0A:DB:23:80:D1:02:5F:E5:D6:09:9C:37:9A;com.library1.example.perkins.maps16

  Alternatively, follow the directions here:
  https://developers.google.com/maps/documentation/android/start#get-key

  Once you have your key (it starts with "AIza"), replace the "google_maps_key"
  string in this file.
-->
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">YOUR_KEY_HERE</string>
</resources>
```

Then move to a fragment

In activity_maps.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"
    tools:context=".MapsActivity">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        android:theme="@style/ThemeOverlay.AppCompat.Dark">

        <Spinner
            android:id="@+id/spinner"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingRight="10dp" />
    </android.support.v7.widget.Toolbar>

    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        android:name="com.google.android.gms.maps.MapFragment"
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:layout_below="@id/toolbar" />
</RelativeLayout>
```

Alt-enter on this object to see decompiled definition

In strings.xml

```
<string-array name="map_types">
    <item>Normal</item>
    <item>Hybrid</item>
    <item>Satellite</item>
    <item>Terrain</item>
    <item>None</item>
</string-array>
```

Add a menu.xml! AND menu folder under res

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
```

Align appropriately
Space for tool bar at top the rest is a map
Get rid of line referencing mainactivity

```

        android:id="@+id/action_KP"
        app:showAsAction="always"
        android:title="KP"></item>

<item
    android:id="@+id/action_NZ"
    app:showAsAction="never"
    android:title="Christchurch"></item>
<item
    android:id="@+id/action_MT"
    app:showAsAction="never"
    android:title="Milford"></item>
<item
    android:id="@+id/action_RT"
    app:showAsAction="never"
    android:title="Routeburn"></item>

</menu>

in res/values/styles change from
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
to
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

```

In MainActivity

```
public class MapsActivity extends AppCompatActivity implements OnMapReadyCallback {
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    //toolbar
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    android.support.v7.app.ActionBar actionBar = getSupportActionBar();

    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    //NOTE I have to change from the default SupportMapFragment

    MapFragment mapFragment = (MapFragment) getFragmentManager()
        .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);
}

```

Get with fragment manager NOT default SupportMapFragment

And in onMapReady You want to set up the spinner after the map is loaded and ready otherwise the user will try to change the map type before its loaded

```

public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    mMap.addMarker(new MarkerOptions().position(KP_HOUSE).title("Marker KP"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(KP_HOUSE));

    setupSimpleSpinner();
}

```

```

Override and add
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu, menu);
    return true;
}

```

Hover over MainActivity and hit Ctrl-o

Hover over MainActivity and hit Ctrl-o

```
//handle navigate button
```

```
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {
```

```
        case R.id.action_KP:  
            goToKP();  
            break;
```

```
        case R.id.action_NZ:  
            goToNZ();  
            break;
```

```
        case R.id.action_MT:  
            goToMT();  
            break;
```

```
        case R.id.action_RT:  
            goToRT();  
            break;
```

```
        default:  
            break;
```

```
    }  
    return true;  
}
```

```
private static final LatLng KP_HOUSE = new LatLng(37.047291, -76.493837);
```

```
private static final LatLng CC_NZ = new LatLng(-43.5321, 172.6362);
```

```
private static final LatLng NZ_MT = new LatLng(-44.9083700, 167.9100500);
```

```
private static final LatLng NZ_RT = new LatLng(-44.7283600, 168.1800600);
```

```
private void goToKP() {
```

```
    CameraUpdate camera = CameraUpdateFactory.newLatLngZoom(KP_HOUSE, 15);
```

```
    mMap.addMarker(new MarkerOptions().position(CC_NZ).title("Keith and Lynns house"));
```

```
    mMap.animateCamera(camera);  
}
```

```
private void goToNZ() {
```

```
    CameraUpdate camera = CameraUpdateFactory.newLatLngZoom(CC_NZ, 15);
```

```
    mMap.addMarker(new MarkerOptions().position(CC_NZ).title("Christchurch NZ"));
```

```
    mMap.animateCamera(camera);  
}
```

```
private void goToMT() {
```

```
    CameraUpdate camera = CameraUpdateFactory.newLatLngZoom(NZ_MT, 15);
```

```
    mMap.addMarker(new MarkerOptions().position(NZ_MT).title("Milford Track NZ\nworlds best hike"));
```

```
    mMap.animateCamera(camera);  
}
```

```
private void goToRT() {
```

```
    CameraUpdate camera = CameraUpdateFactory.newLatLngZoom(NZ_RT, 15);
```

```
    mMap.addMarker(new MarkerOptions().position(NZ_RT).title("Routeburn Track NZ\nworlds best hike"));
```

```
    mMap.animateCamera(camera);  
}
```

```
test here you will need to go to res/values/styles and change the theme to
```

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

```
And finally the spinner
```

```
AdapterView.OnItemClickListener mySpinnerListener;
```

```
private void setupSimpleSpinner() {
```

```
    Spinner spinner = (Spinner) findViewById(R.id.spinner);
```

```
    // Create an ArrayAdapter using the string array and a default spinner layout
```

```
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
```

```
        R.array.map_types, android.R.layout.simple_spinner_item);
```

```
    // Specify the layout to use when the list of choices appears
```

```
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
```

```
    // Apply the adapter to the spinner
```

```
    spinner.setAdapter(adapter);
```

```
    //set listener
```

```
    mySpinnerListener = new AdapterView.OnItemClickListener() {
```

```
        @Override
```

```
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
```

Hover over and hit
alt-enter

```

        switch (position) {
            // Sets the map type
            case 0:
                mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
                break;
            case 1:
                mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
                break;
            case 2:
                mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
                break;
            case 3:
                mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
                break;
            case 4:
                mMap.setMapType(GoogleMap.MAP_TYPE_NONE);
                break;
            default:
                break;
        }
    }

    /**
     * Callback method to be invoked when the selection disappears from this
     * view. The selection can disappear for instance when touch is activated
     * or when the adapter becomes empty.
     *
     * @param parent The AdapterView that now contains no selected item.
     */
    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
};

//respond when spinner clicked
spinner.setOnItemClickListener(mySpinnerListener);
}

```