
Migration Guide Maps-Android

Table of Contents

Add a Map	3
Show User Location and Track	7
Add Marker	10
Custom Marker	12
Custom Info Window	14
Add a Polyline	16
Add a Polygon	17
Camera Animation	19
Map Click	21
Map Long Press	22

Add a Map

MapmyIndia:

The XML layout file

By default, the XML file that defines the app's layout is at `res/layout/activity_maps.xml`. It contains the following code:

```
<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/map"
    android:name="com.mapbox.mapboxsdk.maps.SupportMapFragment"/>
```

The maps activity Java file

By default, the Java file that defines the maps activity is named `MapsActivity.java`. It should contain the following code after your package name:

```
import com.mapbox.mapboxsdk.MapmyIndia;
import com.mapbox.mapboxsdk.camera.CameraUpdateFactory;
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.maps.MapView;
import com.mapbox.mapboxsdk.maps.MapboxMap;
import com.mapbox.mapboxsdk.maps.OnMapReadyCallback;
```

```
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
    .findFragmentById(R.id.map);
mapFragment.getMapAsync(this);
```

OR

```
mapView = (MapView) findViewById(R.id.map_view);
mapView.onCreate(savedInstanceState);
mapView.getMapAsync(this);
```

```
@Override
public void onMapReady(MapboxMap mapboxMap) {
    this.mMap = mapboxMap;
}

@Override
protected void onStart() {
    super.onStart();
    mapView.onStart();
}
```

```

}

@Override
protected void onStop() {
    super.onStop();
    mapView.onStop();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    mapView.onDestroy();
}

@Override
protected void onPause() {
    super.onPause();
    mapView.onPause();
}

@Override
protected void onResume() {
    super.onResume();
    mapView.onResume();
}

@Override
public void onLowMemory() {
    super.onLowMemory();
    mapView.onLowMemory();
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    mapView.onSaveInstanceState(outState);
}

```

Google:

The XML layout file

By default, the XML file that defines the app's layout is at `res/layout/activity_maps.xml`. It contains the following code:

```

<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>

```

The maps activity Java file

By default, the Java file that defines the maps activity is named `MapsActivity.java`. It should contain the following code after your package name:

```
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
```

```
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
    .findFragmentById(R.id.map);
mapFragment.getMapAsync(this);
```

OR

```
private static final String MAP_VIEW_BUNDLE_KEY = "MapViewBundleKey";
```

```
Bundle mapViewBundle = null;
if (savedInstanceState != null) {
    mapViewBundle = savedInstanceState.getBundle(MAP_VIEW_BUNDLE_KEY);
}
mapView = findViewById(R.id.map_view);
mapView.onCreate(mapViewBundle);
mapView.getMapAsync(this);
```

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
}
```

```
@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
```

```
Bundle mapViewBundle = outState.getBundle(MAP_VIEW_BUNDLE_KEY);
```

```
if (mapViewBundle == null) {
    mapViewBundle = new Bundle();
    outState.putBundle(MAP_VIEW_BUNDLE_KEY, mapViewBundle);
}
mapView.onSaveInstanceState(mapViewBundle);
}
```

```
@Override
protected void onResume() {
```

```
super.onResume();
mapView.onResume();
}

@Override
protected void onStart() {
    super.onStart();
    mapView.onStart();
}

@Override
protected void onStop() {
    super.onStop();
    mapView.onStop();
}

@Override
protected void onPause() {
    mapView.onPause();
    super.onPause();
}

@Override
protected void onDestroy() {
    mapView.onDestroy();
    super.onDestroy();
}

@Override
public void onLowMemory() {
    super.onLowMemory();
    mapView.onLowMemory();
}
```

Show User Location and Track

AndroidManifest.xml file

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication" >
    ...
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    ...
</manifest>
Or the fine location permission:
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication" >
    ...
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    ...
</manifest>
```

MapmyIndia:

Java file

It should contain the following code after your package name:

```
import com.mapbox.android.core.location.LocationEngine;
import com.mapbox.android.core.location.LocationEngineListener;
import com.mapbox.mapboxsdk.camera.CameraUpdateFactory;
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.location.LocationComponent;
import com.mapbox.mapboxsdk.location.LocationComponentOptions;
import com.mapbox.mapboxsdk.location.modes.CameraMode;
import com.mapbox.mapboxsdk.location.modes.RenderMode;
import com.mapbox.mapboxsdk.maps.MapboxMap;
```

```
LocationComponentOptions options = LocationComponentOptions.builder(this)
    .trackingGesturesManagement(true)
    .accuracyColor(ContextCompat.getColor(this, R.color.colorAccent))
    .build();

// Get an instance of the component
LocationComponent locationComponent = mapboxMap.getLocationComponent();

// Activate with options
locationComponent.activateLocationComponent(this, options);

// Enable to make component visible
```

```

locationComponent.setLocationComponentEnabled(true);
locationEngine = locationComponent.getLocationEngine();
locationEngine.addLocationEngineListener(this);

// Set the component's camera mode
locationComponent.setCameraMode(CameraMode.TRACKING);
locationComponent.setRenderMode(RenderMode.COMPASS);

@Override
public void onLocationChanged(Location location) {
    mapboxMap.animateCamera(CameraUpdateFactory.newLatLngZoom(
        new LatLng(location.getLatitude(), location.getLongitude()), 16));
    locationEngine.removeLocationEngineListener(this);
}

@Override
protected void onResume() {
    super.onResume();
    if (locationEngine != null) {
        locationEngine.removeLocationEngineListener(this);
        locationEngine.addLocationEngineListener(this);
    }
    isVisible = true;
}

@Override
protected void onPause() {
    super.onPause();
    if (locationEngine != null)
        locationEngine.removeLocationEngineListener(this);
    isVisible = false;
}

@Override
protected void onStop() {
    super.onStop();
    if (locationEngine != null) {
        locationEngine.removeLocationEngineListener(this);
        locationEngine.removeLocationUpdates();
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (locationEngine != null) {
        locationEngine.deactivate();
    }
}

```

Google:

Java file

It should contain the following code after your package name:

```
import com.google.android.gms.maps.GoogleMap;
```

```
@Override  
public void onMapReady(final GoogleMap mMap) {  
    this.mGoogleMap = mMap;  
    mGoogleMap.setMyLocationEnabled(true);  
}
```

Add Marker

MapmyIndia:

Java file

It should contain the following code after your package name:

```
import com.mapbox.mapboxsdk.annotations.MarkerOptions;
import com.mapbox.mapboxsdk.camera.CameraPosition;
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.maps.MapboxMap;
```

```
@Override
public void onMapReady(MapboxMap mapboxMap) {

    mapboxMap.addMarker(new MarkerOptions().position(new LatLng(
        25.321684, 82.987289)).title("Taj Mahal"));

    /* this is done for animating/moving camera to particular position */
    CameraPosition cameraPosition = new CameraPosition.Builder().target(new LatLng(
        25.321684, 82.987289)).zoom(10).tilt(0).build();
    mapboxMap.setCameraPosition(cameraPosition);
}
```

Google:

Java file

It should contain the following code after your package name:

```
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
```

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    // Add a marker in Sydney and move the camera

    LatLng latLng= new LatLng(
```

```
    25.321684, 82.987289);
    mMap.addMarker(new MarkerOptions().position(latLng).title("Taj Mahal"));
// Creating a marker
    MarkerOptions markerOptions = new MarkerOptions();
    // Setting the position for the marker
    markerOptions.position(latLng);
    // Setting the title for the marker.
    // This will be displayed on tapping the marker
    markerOptions.title(latLng.latitude + " : " + latLng.longitude);
    // Clears the previously touched position
    googleMap.clear();

    // Animating to the touched position
    googleMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
    // Placing a marker on the touched position
    googleMap.addMarker(markerOptions);
}
```

Custom Marker

MapmyIndia:

Java file

It should contain the following code after your package name:

```
import com.mapbox.mapboxsdk.annotations.Icon;
import com.mapbox.mapboxsdk.annotations.IconFactory;
import com.mapbox.mapboxsdk.annotations.MarkerOptions;
import com.mapbox.mapboxsdk.camera.CameraPosition;
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.maps.MapboxMap;
```

```
@Override
public void onMapReady(MapboxMap mapboxMap) {
    IconFactory iconFactory = IconFactory.getInstance(AddCustomMarkerActivity.this);
    Icon icon = iconFactory.fromResource(R.drawable.placeholder);
    mapboxMap.addMarker(new MarkerOptions().position(new LatLng(
        25.321684, 82.987289)).icon(icon));

    /* this is done for animating/moving camera to particular position */
    CameraPosition cameraPosition = new CameraPosition.Builder().target(new LatLng(
        25.321684, 82.987289)).zoom(14).tilt(0).build();
    mapboxMap.setCameraPosition(cameraPosition);
}
```

Google:

Java file

It should contain the following code after your package name:

```
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.BitmapDescriptor;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
```

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    // Add a marker in Sydney and move the camera
```

```
BitmapDescriptor icon = BitmapDescriptorFactory.fromResource(R.drawable.placeholder);
mMap.addMarker(new MarkerOptions().position(new LatLng(
    25.321684, 82.987289)).icon(icon));
CameraUpdate center=
    CameraUpdateFactory.newLatLng(new LatLng(40.76793169992044,
        -73.98180484771729));
CameraUpdate zoom=CameraUpdateFactory.zoomTo(15);
mMap.moveCamera(center);
mMap.animateCamera(zoom);
}
```

Custom Info Window

MapmyIndia:

```
MarkerOptions markerOptions = new MarkerOptions().position(point).icon(icon);
Marker marker = mapBoxMap.addMarker(markerOptions);
String tittle = "abc";
marker.setTitle(tittle);
mapboxMap.setInfoWindowAdapter(new MapboxMap.InfoWindowAdapter() {
    @Nullable
    @Override
    public View getInfoWindow(@NonNull Marker marker) {
        View view = (context).getLayoutInflater()
            .inflate(R.layout.layout, null);
        TextView text = (TextView)view.findViewById(R.id.text);
        text.setText(marker.getTitle());
        return view;
    }
});
```

Google:

```
LatLng latLng = new LatLng(25.321684, 82.987289);
MarkerOptions markerOptions = new MarkerOptions();
markerOptions.position(latLng)
    .title("")
    .snippet(latLng.latitude + " : " + latLng.longitude);

InfoData infoData = new InfoData();
infoData.setName("Demo");
infoData.setDetail("This is demo information");

CustomInfoWindowAdapter customInfoWindowAdapter = new CustomInfoWindowAdapter(this); // Custom Adapter
class

mapboxMap.setInfoWindowAdapter(new GoogleMap.InfoWindowAdapter() {
    @Override
    public View getInfoWindow(Marker marker) {
        return null;
    }

    @Override
    public View getInfoContents(Marker marker) {

        View view = GoogleInfoWindowActivity.this.getLayoutInflater().inflate(R.layout.custom_info_window_layout,
            null);

        TextView textName = view.findViewById(R.id.info_title);
        TextView textDetail = view.findViewById(R.id.info_details);

        InfoData infoData = (InfoData) marker.getTag();

        textName.setText(infoData.getName());
```

```
textDetail.setText(infoData.getDetail());  
return view;  
}  
});
```

```
Marker marker = mapboxMap.addMarker(markerOptions);  
marker.setTag(infoData);  
marker.showInfoWindow();
```

Add a Polyline

MapmyIndia:

Java file

It should contain the following code after your package name:

```
import com.mapbox.mapboxsdk.annotations.PolylineOptions;
import com.mapbox.mapboxsdk.camera.CameraUpdateFactory;
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.geometry.LatLngBounds;
import com.mapbox.mapboxsdk.maps.MapboxMap;
```

```
ArrayList<LatLng> listOfLatlang = new ArrayList<>();
@Override
public void onMapReady(MapboxMap mapboxMap) {
    listOfLatlang.add(new LatLng(28.705436, 77.100462));
    listOfLatlang.add(new LatLng(28.705191, 77.100784));
    listOfLatlang.add(new LatLng(28.704646, 77.101514));
    listOfLatlang.add(new LatLng(28.704194, 77.101171));
    listOfLatlang.add(new LatLng(28.704083, 77.101066));
    listOfLatlang.add(new LatLng(28.703900, 77.101318));
    mapboxMap.addPolyline(new
PolylineOptions().addAll(listOfLatlang).color(Color.parseColor("#3bb2d0")).width(4));

    LatLngBounds latLngBounds = new LatLngBounds.Builder().includes(listOfLatlang).build();
    mapboxMap.animateCamera(CameraUpdateFactory.newLatLngBounds(latLngBounds, 70));
}
```


Add a Polygon

MapmyIndia:

Java file

It should contain the following code after your package name:

```
import com.mapbox.mapboxsdk.annotations.PolylineOptions;
import com.mapbox.mapboxsdk.camera.CameraUpdateFactory;
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.geometry.LatLngBounds;
import com.mapbox.mapboxsdk.maps.MapboxMap;
```

```
ArrayList<LatLng> listOfLatLng = new ArrayList<>();

@Override
public void onMapReady(MapboxMap mapboxMap) {

    listOfLatLng.add(new LatLng(28.703900, 77.101318));
    listOfLatLng.add(new LatLng(28.703331, 77.102155));
    listOfLatLng.add(new LatLng(28.703905, 77.102761));
    listOfLatLng.add(new LatLng(28.704248, 77.102370));

    mapboxMap.addPolygon(new PolygonOptions().addAll(listOfLatLng).fillColor(Color.parseColor("#753bb2d0")));

    /* this is done for move camera focus to particular position */
    LatLngBounds latLngBounds = new LatLngBounds.Builder().includes(listOfLatLng).build();
    mapboxMap.animateCamera(CameraUpdateFactory.newLatLngBounds(latLngBounds, 70));
}
```

Google:

Java file

It should contain the following code after your package name:

```
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.LatLngBounds;
import com.google.android.gms.maps.model.PolygonOptions;
```

```
ArrayList<LatLng> listOfLatLng = new ArrayList<>();
```

```
@Override
```

```

public void onMapReady(GoogleMap googleMap) {

    listOfLatlang.add(new LatLng(28.703900, 77.101318));
    listOfLatlang.add(new LatLng(28.703331, 77.102155));
    listOfLatlang.add(new LatLng(28.703905, 77.102761));
    listOfLatlang.add(new LatLng(28.704248, 77.102370));
    //
    googleMap.addPolygon(new
PolygonOptions().addAll(listOfLatlang).fillColor(Color.parseColor("#753bb2d0")));
    /////* this is done for move camera focus to particular position */
    LatLngBounds.Builder builder = new LatLngBounds.Builder();
    for (LatLng listofLatlang : listOfLatlang) {
        builder.include(listofLatlang);
    }
    LatLngBounds latLngBounds= builder.build();
    googleMap.animateCamera(CameraUpdateFactory.newLatLngBounds(latLngBounds, 70));
}

```

Camera Animation

MapmyIndia:

Java file

It should contain the following code after your package name:

```
import com.mapbox.mapboxsdk.camera.CameraPosition;
import com.mapbox.mapboxsdk.camera.CameraUpdateFactory;
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.maps.MapboxMap;
```

```
@Override
public void onMapReady(MapboxMap mapboxMap) {
    this.mapboxMap = mapboxMap;
    CameraPosition cameraPosition = new CameraPosition.Builder().target(new LatLng(
        25.321684, 82.987289)).zoom(14).tilt(0).build();
    mapboxMap.setCameraPosition(cameraPosition);
}

public void moveCamera() {
    mapboxMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
        22.553147478403194,
        77.23388671875), 14));
}

public void easeCamera() {
    mapboxMap.easeCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
        28.704268, 77.103045), 14));
}

public void animateCamera() {
    mapboxMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(28.698791, 77.121243), 14));
}
```

Google:

Imports

```
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;
```

Overriden Method of OnMapReadyCallback

```
@Override
public void onMapReady(GoogleMap googleMap) {
    this.mMap = googleMap;
}
```

```
CameraUpdate center=
    CameraUpdateFactory.newLatLng(new LatLng(40.76793169992044,
        -73.98180484771729));
CameraUpdate zoom=CameraUpdateFactory.zoomTo(15);
mMap.moveCamera(center);
mMap.animateCamera(zoom);
}

public void moveCamera() {
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
        22.553147478403194,
        77.23388671875), 14));
}

public void animateCamera() {
    mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
        28.698791, 77.121243), 14));
}
```

Map Click

MapmyIndia:

Java file

It should contain the following code after your package name:

```
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.maps.MapboxMap;
```

```
@Override
public void onMapReady(MapboxMap mapboxMap) {
    mapboxMap.setOnMapClickListener(new MapboxMap.OnMapClickListener() {
        @Override
        public void onMapClick(@NonNull LatLng latLng) {
            Toast.makeText(MapClickActivity.this, latLng.toString(), Toast.LENGTH_LONG).show();
        }
    });
}
```

Google:

Java file

It should contain the following code after your package name:

```
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.LatLng;
```

```
@Override
public void onMapReady(GoogleMap googleMap) {

    googleMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
        @Override
        public void onMapClick(LatLng latLng) {

        }
    });
}
```

Map Long Press

MapmyIndia:

Java file

It should contain the following code after your package name:

```
import com.mapbox.mapboxsdk.geometry.LatLng;
import com.mapbox.mapboxsdk.maps.MapboxMap;
```

```
@Override
public void onMapReady(MapboxMap mapboxMap) {
    mapboxMap.setOnMapLongClickListener(new MapboxMap.OnMapLongClickListener() {
        @Override
        public void onMapLongClick(@NonNull LatLng latLng) {
            Toast.makeText(MapLongClickActivity.this, latLng.toString(), Toast.LENGTH_LONG).show();
        }
    });
}
```

Google:

Java file

It should contain the following code after your package name:

```
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.LatLng;
```

```
@Override
public void onMapReady(GoogleMap googleMap) {
    googleMap.setOnMapLongClickListener(new GoogleMap.OnMapLongClickListener() {
        @Override
        public void onMapLongClick(LatLng latLng) {
        }
    });
}
```