Using MapBox V10 in a Java based Android Applications

Last updated: Apr-08-2023

by: Adi Barda
email: scenemax3d@gmail.com
website: www.scenemax3d.com

Table Of Contents

Official MapBox Java samples

https://github.com/mapbox/mapbox-maps-android/blob/main/app/src/public/java/com/mapbox/maps/testapp/examples/java/RuntimeStylingJavaActivity.java

https://github.com/mapbox/mapbox-maps-android/blob/main/app/src/main/java/com/mapbox/maps/testapp/examples/java/DSLStylingJavaActivity.java

General comments

`_mapView`	your MapView object
LatLng	a simple (Google Maps) object holding latitude & longitude data - you can totally avoid using it in your apps
ptMng	PointAnnotationManager object

Load a map style from a resource file:

```
public void onMapReady() {

    _mapView = _mapBoxViewFragment.getMap();
    String style = loadStyle(this.ctx, "Data/IHM.json");
    _mapView.getMapboxMap().loadStyleJson(style, (s) -> {
        addGeneralSymbolSource(s);
        addGeneralSymbolLayer(s);

        AnnotationPlugin annotationApi = AnnotationPluginImplKt.getAnnotations(_mapView);
        ptMng = PointAnnotationManagerKt.createPointAnnotationManager(annotationApi, new AnnotationConfig());
    });

}

public static String loadStyle(Context ctx, String res) {

    String data = "";

    try {
        InputStream stream = ctx.getAssets().open(res);
        int size = stream.available();
        byte[] buffer = new byte[size];
        stream.read(buffer);
        stream.close();
        data = new String(buffer);
    } catch (IOException e) {
        // Handle exceptions here
    }

    return data;

}
```

Add a polyline to the map

```
public Object addPolyline(List<LatLng> latlngs, int color, int width, int pattern) {

    Style style = _mapView.getMapboxMap().getStyle();

    List<Point> points = new ArrayList<>();
    for (LatLng pt:latlngs) {
        points.add(Point.fromLngLat(pt.longitude,pt.latitude));
    }

    LineString line = LineString.fromLngLats(points);
    Feature feature = Feature.fromGeometry(line);
    //FeatureCollection featureCollection = FeatureCollection.fromFeatures(Collections.singletonList(feature));
    GeoJsonSource polygon = (GeoJsonSource) SourceUtils.getSource(style,"track");
    if(polygon!=null) {
        polygon.feature(feature);
    } else {
        polygon = new GeoJsonSource.Builder("track")
                //.featureCollection(featureCollection)
                .feature(feature)
                .generateId(true)
                .build();

        SourceUtils.addSource(style, polygon);

        LineLayer lineLayer = new LineLayer("track-layer", "track");
        lineLayer.lineColor("#00FF00")
                .lineOpacity(0.7)
                .lineWidth(8.0);

        LayerUtils.addLayer(style,lineLayer);
    }

    return null;
}
```

Getting the PointAnnotationManager of the map

```
AnnotationPlugin annotationApi = AnnotationPluginImplKt.getAnnotations(_mapView);
ptMng = PointAnnotationManagerKt.createPointAnnotationManager(annotationApi, new AnnotationConfig());
```

Add a marker with icon from resource ID

```
public IMapMarker addMyLocationMarker(LatLng location) {

   Point point = Point.fromLngLat(location.longitude, location.latitude);
   PointAnnotationOptions pointAnnotationOptions = new PointAnnotationOptions()
           .withPoint(point)
           .withIconImage(BitmapFactory.decodeResource(ctx.getResources(), R.drawable.pointer3_70x96))
           .withIconSize(1);
   PointAnnotation pointAnnotation = ptMng.create(pointAnnotationOptions);
   MapBoxMarker marker = new MapBoxMarker(pointAnnotation, ptMng);

   return marker;
}
```

Add a marker with icon from bitmap

```
public IMapMarker addMarker(LatLng location, String markerText, Bitmap icon, String snippet) {
   Point point = Point.fromLngLat(location.longitude, location.latitude);
   PointAnnotationOptions pointAnnotationOptions = new PointAnnotationOptions()
           .withPoint(point)
           .withIconImage(icon)
           .withIconSize(1);
   PointAnnotation pointAnnotation = ptMng.create(pointAnnotationOptions);
   MapBoxMarker marker = new MapBoxMarker(pointAnnotation, ptMng);

   return marker;
```

```
}
```

Get camera position

```
public LatLng getCameraPosition() {
   if(_mapView==null) return null;
   Point p = this._mapView.getMapboxMap().getCameraState().getCenter();
   return new LatLng(p.latitude(), p.longitude());
}
```

Animate camera using flyTo() method

```
public void setPosition(LatLng l) {
   if(_mapView==null) return;
   CameraAnimationsUtils.flyTo(_mapView.getMapboxMap(),
           new CameraOptions.Builder().center(Point.fromLngLat(l.longitude,l.latitude)).build(),
           new MapAnimationOptions.Builder().duration(500).build());

}
```

Observe map long-click event

```
GesturesUtils.addOnMapLongClickListener(_mapView.getMapboxMap(),point -> {

   return true;
});
```

Observe map location change

```
_mapView.getMapboxMap().addOnCameraChangeListener(new OnCameraChangeListener() {
   @Override
   public void onCameraChanged(@NotNull CameraChangedEventData cameraChangedEventData) {
```

```
    }
});
```

Get current map visible bounds

```
CoordinateBounds bb = _mapView.getMapboxMap().coordinateBoundsForCamera(
        new CameraOptions.Builder().center(_mapView.getMapboxMap().getCameraState().getCenter())
                .zoom(_mapView.getMapboxMap().getCameraState().getZoom())
                .build());
```

Move layer below another layer

```
style.moveStyleLayer("track-layer", new LayerPosition(null,"markers",null));
```

Marker operations

```
package com.abware.watchdog_client;
import com.google.android.gms.maps.model.LatLng;
import com.mapbox.geojson.Point;
import com.mapbox.maps.plugin.annotation.generated.PointAnnotation;
import com.mapbox.maps.plugin.annotation.generated.PointAnnotationManager;
public class MapBoxMarker extends MapMarker<PointAnnotation> {
  private PointAnnotationManager mng;
  public MapBoxMarker(PointAnnotation marker, PointAnnotationManager mng) {
     super(marker);
     this.mng = mng;
  }
  @Override
  public void setPosition(LatLng val) {
     this.getMarker().setPoint(Point.fromLngLat(val.longitude,val.latitude));
     this.mng.update(this.getMarker());
  }
```

```
  @Override
  public void setRotation(float val) {
     this.getMarker().setIconRotate((double)val);
     this.mng.update(this.getMarker());
  }

  @Override
  public void setRotation(float val, boolean redrawMap) {
     this.getMarker().setIconRotate((double)val);
     this.mng.update(this.getMarker());
  }
  @Override
  public LatLng getPosition() {
     Point pt = this.getMarker().getPoint();
     return new LatLng(pt.latitude(), pt.longitude());
  }
}
```

Other functions

```
@Override
public float getTilt() {
   if(_mapView==null) return 0;
   return (float)this._mapView.getMapboxMap().getCameraState().getPitch();
}

@Override
public float getBearing() {
   if(_mapView==null) return 0;
   return (float)this._mapView.getMapboxMap().getCameraState().getBearing();
}

@Override
```

```
public float getZoom() {
   if(_mapView==null) return 0;
   return (float)this._mapView.getMapboxMap().getCameraState().getZoom();
}

@Override
public void setPosition(LatLng l) {
   if(_mapView==null) return;
   this._mapView.getMapboxMap().setCamera(new
CameraOptions.Builder().center(Point.fromLngLat(l.longitude,l.latitude)).build());
}

@Override
public void setBearing(float bearing) {
   if(_mapView==null) return;
   this._mapView.getMapboxMap().setCamera(new CameraOptions.Builder().bearing((double) bearing).build());
}

@Override
public void setZoom(float zoom) {
   if(_mapView==null) return;
   this._mapView.getMapboxMap().setCamera(new CameraOptions.Builder().zoom((double) zoom).build());
}

@Override
public LatLng getPosition() {
   if(_mapView==null) return null;
   Point p = this._mapView.getMapboxMap().getCameraState().getCenter();
   return new LatLng(p.latitude(), p.longitude());
}
```

A sample build.gradle file

```
apply plugin: 'com.android.application'

allprojects {
```

```
}

android {
    compileSdkVersion 31
    //buildToolsVersion '27.0.3'
    useLibrary 'org.apache.http.legacy'

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    packagingOptions {
        pickFirst '**/*.so'
    }

    defaultConfig {
        applicationId "com.abware.watchdog_client"
        minSdkVersion 26
        targetSdkVersion 31
        versionCode 98
        versionName '3.2'
        multiDexEnabled true
    }


    buildTypes {
        release {
            minifyEnabled false
        }
    }

    dexOptions {
        javaMaxHeapSize "4g"
    }

    lintOptions {
        checkReleaseBuilds false
```

```
    }

}

dependencies {

    implementation 'com.mapbox.maps:android:10.10.0'
    implementation 'com.mapbox.plugin:maps-annotation:10.11.0'

    implementation 'com.google.maps.android:android-maps-utils:0.5'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
}
```

A sample settings.gradle file

```
include ':app'

dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        maven { url "https://oss.sonatype.org/content/repositories/snapshots/" }
        maven { url "https://maven.google.com" }
        google()
        mavenCentral()
        jcenter()
        maven { url "https://jitpack.io" }
        maven {
            url 'https://api.mapbox.com/downloads/v2/releases/maven'
            authentication {
                basic(BasicAuthentication)
            }
            credentials {
                // Do not change the username below.
                // This should always be `mapbox` (not your username).
                username = "mapbox"
                // Use the secret token you stored in gradle.properties as the password
                password = "your_mapbox_password_goes_here"
```

```
            }
        }
    }
}
```