# TPL Maps vs. Google Maps Flutter Integration Code Comparison

TPL Maps and Google Maps are two popular mapping platforms for Flutter developers. TPL Maps is a newer platform that offers a simpler and more lightweight API, while Google Maps is a more mature platform with a wider range of features.

In this document, we will perform a comprehensive feature-by-feature comparison between the provided TPL Maps Flutter integration code and a typical Google Maps Flutter integration code. We will also outline the missing features in the TPL Maps code and provide corresponding code snippets from Google Maps for reference.  
  
  
**Google Map structure**  
  
Please try to follow Google Map structure as the below features are needed directly, which TPL Maps lacks.

const GoogleMap({  
 Key? key**,** required this.initialCameraPosition**,** */// If no padding is specified default padding will be 0.*this.markers = const <Marker>{}**,** this.polygons = const <Polygon>{}**,** this.polylines = const <Polyline>{}**,** this.circles = const <Circle>{}**,** this.tileOverlays = const <TileOverlay>{},

this.onLongPress,

this.cloudMapId,this.onTap**,**})

**Polylines and its styling**

TPL Maps provides setup for polylines but lacks explicit polyline addition with customizable options.

Currently, there is no provision for customizing polyline color, width, and other styling options in TPL Maps.

Google Maps supports customization of polyline appearance.

// TPL Maps

TPLRoutingViewController tplRoutingViewController = TPLRoutingViewController(33.705349, 73.069788,33.698047971892045, 73.06930062598059 ,

(tplRoutingCallBack) => {

print(tplRoutingCallBack)

},);

tplRoutingViewController.getSearchItems();

controller.setUpPolyLine();

controller.addPolyLine(33.705349, 73.069788,33.698047971892045, 73.06930062598059);

Google Maps explicitly adds a polyline with customizable options, such as color, width, and start and end coordinates.

// Google Maps

Polyline(

  polylineId: PolylineId("examplePolyline"),

  points: polylineCoordinates,

  color: Colors.blue,

  width: 5.0,

);

**Circles**

TPL Maps includes the setup for circles but lacks explicit circle addition with customizable options.

// TPL Maps

controller.addCircle(33.705349, 73.069788, 10.0);

Google Maps explicitly adds a circle with customizable options, such as radius, fill color, and stroke color.

// Google Maps

Circle(

  circleId: CircleId("exampleCircle"),

  center: LatLng(37.7749, -122.4194),

  radius: 10.0,

  fillColor: Colors.red,

  strokeColor: Colors.black,

  strokeWidth: 2.0,

);

**Marker Info Windows & Customization**

There is no explicit code for displaying additional information on tapping a marker in TPL Maps.

Google Maps supports displaying additional information on tapping a marker and customization of marker icons and other properties.

// Google Maps

Marker(

  markerId: MarkerId("exampleMarker"),

  position: LatLng(37.7749, -122.4194),

  infoWindow: InfoWindow(

    title: 'Marker Title',

    snippet: 'Marker Description',

  ),

);

Marker(

  markerId: MarkerId("exampleMarker"),

  position: LatLng(37.7749, -122.4194),

  icon: BitmapDescriptor.fromAsset('assets/marker\_icon.png'),

);

TPL Maps adds markers but lacks options for customizing marker icons and other properties.

// TPL Maps

controller.addMarker(33.705349, 73.069788);

**Example**

From the below flutter code of TPL Maps for flutter integration taken from the flutter package of TPL Maps, the creation or routing of a polyline between two points is not shown on the screen nor there is any code to make changes to the color or its width is not possible nor the creation of the circle is not possible and not shown on the flutter app screen.

Dart

import 'dart:developer';

import 'package:flutter/material.dart';

import 'package:map\_comparison/second.dart';

import 'package:tplmapsflutterplugin/TplMapsView.dart';

void main() {

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Flutter Demo',

      theme: ThemeData(

        primarySwatch: Colors.blue,

      ),

      home: const MyHomePage(title: 'Flutter Demo Home Page'),

    );

  }

}

class MyHomePage extends StatefulWidget {

  const MyHomePage({Key? key, required this.title}) : super(key: key);

  final String title;

  @override

  State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

  late TplMapsViewController \_controller;

  // // Initial Selected Value

  // String dropdownvalue = 'Item 1';

  //

  // // List of items in our dropdown menu

  // var items = [

  //   'Item 1',

  //   'Item 2',

  //   'Item 3',

  //   'Item 4',

  //   'Item 5',

  // ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("TPL Maps"),

      ),

      body: Stack(

        children: [

          Container(

            child: TplMapsView(

              isShowBuildings: true,

              isZoomEnabled: true,

              showZoomControls: true,

              isTrafficEnabled: true,

              mapMode: MapMode.NIGHT,

              enablePOIs: true,

              setMyLocationEnabled: false,

              myLocationButtonEnabled: false,

              showsCompass: true,

              allGesturesEnabled: true,

              tplMapsViewCreatedCallback: \_callback,

              //tPlMapsViewMarkerCallBack: \_markerCallback,

            ),

          ),

          // Container(

          //     width: double.infinity,

          //     margin: EdgeInsets.fromLTRB(10, 10, 10, 20),

          //     child: Column(

          //       mainAxisAlignment: MainAxisAlignment.spaceBetween,

          //       children: [

          //         DropdownButton(

          //

          //           // Initial Value

          //           value: dropdownvalue,

          //

          //           // Down Arrow Icon

          //           icon: const Icon(Icons.keyboard\_arrow\_down),

          //

          //           // Array list of items

          //           items: items.map((String items) {

          //             return DropdownMenuItem(

          //               value: items,

          //               child: Text(items),

          //             );

          //           }).toList(),

          //           // After selecting the desired option,it will

          //           // change button value to selected value

          //           onChanged: (String? newValue) {

          //             setState(() {

          //               dropdownvalue = newValue!;

          //             });

          //           },

          //         ),

          //       ],

          //

          //     )

          // ),

          // Container(

          //   margin: EdgeInsets.fromLTRB(10, 50, 10, 0),

          //   width: double.infinity,

          //   height: 50,

          //   color: Colors.red,

          //   child: Text("Text on the Map", style: TextStyle(color: Colors.white , fontSize: 20),  textAlign: TextAlign.center,),

          // )

        ],

      ),

    );

  }

  void \_markerCallback(String callback){

    Navigator.push(

      context,

      MaterialPageRoute(builder: (context) => const SecondRoute()),

    );

    //Navigator.push(context, PageTransition(type: PageTransitionType.leftToRight, child: SecondRoute()));

  }

  void \_callback(TplMapsViewController controller) {

    controller.showBuildings(false);

    controller.showZoomControls(false);

    controller.setTrafficEnabled(false);

    controller.enablePOIs(true);

    // controller.setMyLocationEnabled(true);

    // controller.myLocationButtonEnabled(true);

    TPLRoutingViewController tplRoutingViewController =

    TPLRoutingViewController(33.705349, 73.069788, 33.698047971892045, 73.06930062598059 ,

        (tplRoutingCallBack) => {

        controller.setUpPolyLine(),

    // You will be get json list response

    log(tplRoutingCallBack)

        },);

    tplRoutingViewController.getSearchItems();

    controller.showsCompass(true);

    controller.showZoomControls(true);

    controller.setCameraPositionAnimated(33.698047971892045, 73.06930062598059,14.0);

    controller.addMarker(33.705349, 73.069788);

    controller.addMarker(33.698047971892045, 73.06930062598059);

    controller.addPolyLine(33.705349, 73.069788, 33.698047971892045, 73.06930062598059);

    controller.addCircle(33.705349, 73.069788 , 25.0);

    controller.setMapMode(MapMode.DEFAULT);

    bool isBuildingsEnabled = controller.isBuildingEnabled;

    print("isBuildingsEnabled: $isBuildingsEnabled");

    bool isTrafficEnabled = controller.isTrafficEnabled;

    print("isTrafficEnabled: $isTrafficEnabled");

    bool isPOIsEnabled = controller.isPOIsEnabled;

    print("isPOIsEnabled: $isPOIsEnabled");

    \_controller  = controller;

  }

  void addMarker(){

    \_controller.addMarker(33.705349, 73.069788);

  }

  void addPolyLine(){

    \_controller.addPolyLine(33.705349, 73.069788, 33.705349, 73.069788);

  }

  void addCircle(){

    \_controller.addCircle(33.705349, 73.069788 , 25.0);

  }

  void removeMarkers(){

    \_controller.removeAllMarker();

  }

  void removePolyline(){

    \_controller.removePolyline();

  }

  void removeAllCircles(){

    \_controller.removeAllCircles();

  }

  // Other methods

  void otherMethods(){

    //....

      \_controller.setZoomEnabled(true);

      \_controller.showBuildings(false);

      \_controller.showBuildings(false);

      \_controller.showZoomControls(true);

      \_controller.setTrafficEnabled(false);

      \_controller.enablePOIs(true);

      \_controller.setMyLocationEnabled(true);

      \_controller.myLocationButtonEnabled(true);

      \_controller.showsCompass(true);

      \_controller.setCameraPositionAnimated(33.69804797189, 73.0693006259, 14.0);

      \_controller.setMapMode(MapMode.DEFAULT);

      \_controller.isBuildingEnabled;

      \_controller.isTrafficEnabled;

      \_controller.isPOIsEnabled;

  }

  // Search

  void getSearchItemsbyName (){

    TPlSearchViewController tPlSearchViewController =

    TPlSearchViewController("Atrium Mall" , 24.8607 , 67.0011 , (retrieveItemsCallback) {

      // You will be get json list response

      log(retrieveItemsCallback);

    },);

    tPlSearchViewController.getReverseGeocoding();

  }

// Create Route between two points

  void getRouting(){

    TPLRoutingViewController tplRoutingViewController =

    TPLRoutingViewController(33.705349, 73.069788, 33.698047971892045, 73.06930062598059 , (tplRoutingCallBack) {

      log(tplRoutingCallBack);

      \_controller.setUpPolyLine();

    },);

    tplRoutingViewController.getSearchItems();

  }

}