Queens University Belfast

Android Studio

CSC3054 / CSC7054

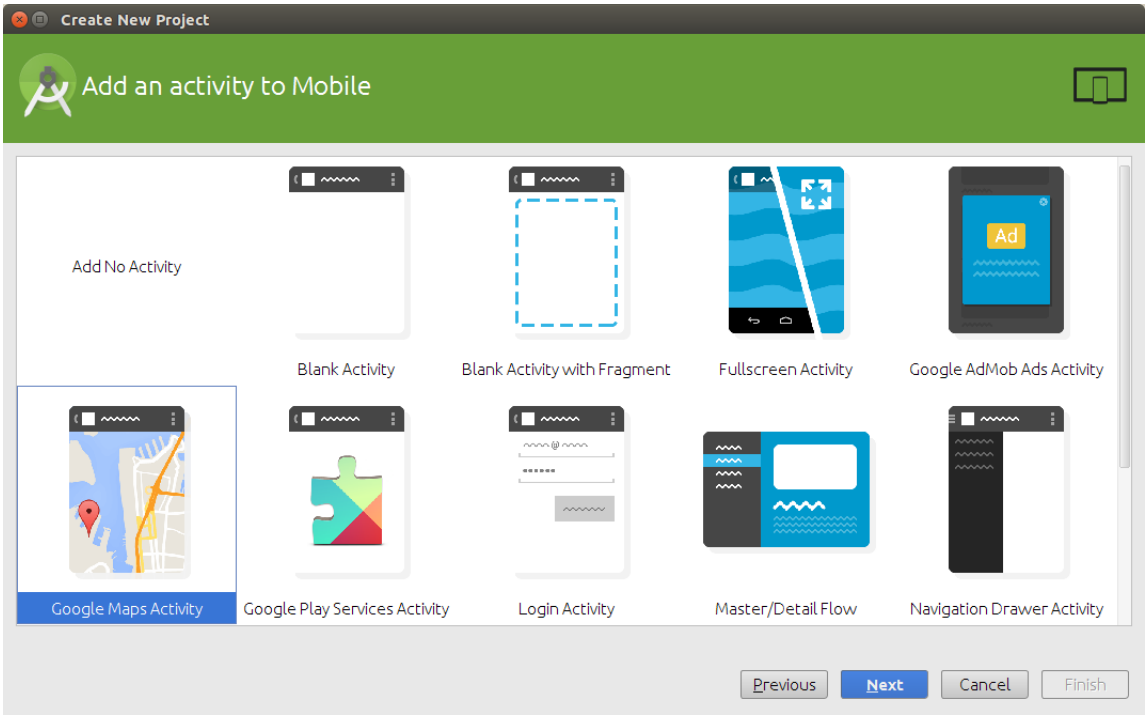
Update to Google Maps

Queens University Belfast

Android Studio is Generating Deprecated Maps Code

Android Studio is generating code that uses the deprecated method `GoogleMap.getMap()`. This tutorial will demonstrate how to change the generated code to use `GoogleMap.getMapAsync()` instead.

Use the `"Create New Project"` wizard in Android Studio to create a `Google Maps Activity`.



The wizard generated this code:

```
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity {

    private GoogleMap mMap; //Might be null if Google Play services APK is not available.

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        setUpMapIfNeeded();
    }
```

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

    /**
     * Sets up the map if it is possible to do so (i.e., the Google Play services APK is
correctly installed) and the map has not already been instantiated.. This will ensure
that we only ever call {@link #setUpMap()} once when {@link #mMap} is not null. If it
isn't installed {@link SupportMapFragment} (and {@link
com.google.android.gms.maps.MapView MapView}) will show a prompt for the user to
install/update the Google Play services APK on their device. A user can return to this
FragmentActivity after following the prompt and correctly installing/updating/enabling
the Google Play services. Since the FragmentActivity may not have been completely
destroyed during this process (it is likely that it would only be stopped or paused),
{@link #onCreate(Bundle)} may not be called again so we should call this method in {@link
#onResume()} to guarantee that it will be called.
     */
    private void setUpMapIfNeeded() {
        // Do a null check to confirm that we have not already instantiated the map.
        if (mMap == null) {
            // Try to obtain the map from the SupportMapFragment.
            mMap = ((SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map))
                    .getMap();
            // Check if we were successful in obtaining the map.
            if (mMap != null) {
                setUpMap();
            }
        }
    }

    /**
     * This is where we can add markers or lines, add listeners or move the camera. In
this case, we just add a marker near Africa. This should only be called once and when we
are sure that {@link #mMap} is not null.
     */
    private void setUpMap() {
        mMap.addMarker(new MarkerOptions().position(new LatLng(0, 0)).title("Marker"));
    }
}
```

The Problem

This code uses `GoogleMap.getMap()` method which has been deprecated since Google Play Services version 6.5, released in December 2014. This generated code is overly complex to support polling for a `GoogleMap` instance which may not be ready. What happens if `getMap()` returns `null` in `setUpMapIfNeeded()`? Additional retry logic is required in order for this this code to be correct.

The Solution

Google recognized that their API suffered this weakness and provided the `getMapAsync()` as a replacement. If this is used instead, the method `setUpMapIfNeeded()` can be removed.

By using `SupportMapFragment` instead of `MapFragment`, lower version devices are able to be supported. The `MapActivity` should extend to the `FragmentActivity` class, which contains the support library to support lower versions of Android. Please not the `support repository` from the SDK Manager must be downloaded.

The `GoogleMap` object

The `GoogleMap` object helps us to use the Google Maps API effectively and to implement many features in our application.

Some features are as follows:

- It can be used to add markers, overlays , and so on
- It can be used to add controls, such as pan and zoom

Using the callback method

A callback function is a function that is provided to another piece of code, allowing it to be called by this code.

The `GoogleMap` object can be obtained from `MapFragment` or directly, however it may return `null`, which may result in the application crashing.

A callback will handle this scenario seamlessly and to enable a callback on the `MapFragment`:

- The `getMapAsync()` method must be called on the `MapFragment`.
- The activity must implement the `OnMapReadyCallback` interface.
- The `onMapReady()` method must be used to handle the callback which will have a `GoogleMap` object as a parameter.
- Both `getMapAsync()` and the callback will be called on the main thread.

```
// The android.support.v4.app package provides classes to support backward compatibility.

import android.support.v4.app.FragmentActivity;


import android.os.Bundle;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
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;
```

```
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        ((SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map))
                .getMapAsync(this);
    }

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

    /**
     * This is where we can add markers or lines, add listeners or move the camera. In
this case, we just add a marker near Africa.<p/>This should only be called once. */
    private void setUpMap() {
        mMap.addMarker(new MarkerOptions().position(new LatLng(0, 0)).title("Marker"));
        try {
            mMap.setMyLocationEnabled(true);
        }
        catch(SecurityException e)
        {
            e.printStackTrace();
        }
    }
}
```

In the code above, `MapsActivity` implements the `OnMapReadyCallback` interface.

The `getMapAsync()` method is used to enable the callback on the main thread.

The callback method is defined in `onMapReady()` which will be called when the map is ready. This method will also has the `GoogleMap` object as the parameter.

The `setMyLocationEnabled()` method is used to enable the location button on `MapFragment`, and this button can be used to set the users current location in maps using GPS.

Hopefully Google will update the wizard in Android Studio as well as their code samples so people will stop using the deprecated `getMap()` and start using `getMapAsync()` instead.

Things to Note (from google developers):

GoogleMap.OnMapLoadedCallback

public static interface GoogleMap.OnMapLoadedCallback

com.google.android.gms.maps.GoogleMap.OnMapLoadedCallback

Callback interface for when the map has finished rendering. This occurs after all tiles required to render the map have been fetched, and all labeling is complete. This event will not fire if the map never loads due to connectivity issues, or if the map is continuously changing and never completes loading due to the user constantly interacting with the map.

public abstract void onMapLoaded ()

Called when the map has finished rendering. This will only be called once. You must request another callback if you want to be notified again.

OnMapReadyCallback

public interface OnMapReadyCallback

com.google.android.gms.maps.OnMapReadyCallback

Callback interface for when the map is ready to be used. Once an instance of this interface is set on a `MapFragment` or `MapView` object, the `onMapReady(GoogleMap)` method is triggered when the map is ready to be used and provides a non-null instance of `GoogleMap`.

If Google Play services is not installed on the device, the user will be prompted to install it, and the `onMapReady(GoogleMap)` method will only be triggered when the user has installed it and returned to the app.

public abstract void onMapReady (GoogleMap googleMap)

Called when the map is ready to be used.

Parameters

`googleMap` A non-null instance of a `GoogleMap` associated with the `MapFragment` or `MapView` that defines the callback.