**package** com.example.user.googlemapclient;

**import** android.app.Activity;

**import** android.app.AlertDialog;

**import** android.content.Context;

**import** android.content.DialogInterface;

**import** android.content.Intent;

**import** android.location.Location;

**import** android.location.LocationManager;

**import** android.support.v4.app.FragmentActivity;

**import** android.os.Bundle;

**import** android.widget.Toast;

**import** com.google.android.gms.common.ConnectionResult;

**import** com.google.android.gms.common.api.GoogleApiClient;

**import** com.google.android.gms.location.LocationListener;

**import** com.google.android.gms.location.LocationRequest;

**import** com.google.android.gms.location.LocationServices;

**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.CameraPosition;

**import** com.google.android.gms.maps.model.LatLng;

**import** com.google.android.gms.maps.model.Marker;

**import** com.google.android.gms.maps.model.MarkerOptions;

**public class** MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback,GoogleApiClient.ConnectionCallbacks,GoogleApiClient.OnConnectionFailedListener,LocationListener {

**private** GoogleMap **mMap**;

LocationManager **locationManager**;

**private** GoogleApiClient **googleApiClient**;

**private** LocationRequest **locationRequest**;

**private** Location **currentLocation**;

**private** Marker **currentMarker**,**itemMarker**;

@Override

**protected void** onCreate(Bundle savedInstanceState) **throws** SecurityException {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_maps***);

*// Obtain the SupportMapFragment and get notified when the map is ready to be used.*

SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()

.findFragmentById(R.id.***map***);

mapFragment.getMapAsync(**this**);

configGoogleApiClient();

configLocationRequest();

**locationManager** = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);

**currentLocation** = **locationManager**.getLastKnownLocation(LocationManager.***NETWORK\_PROVIDER***);

}

**private synchronized void** configGoogleApiClient() {

**googleApiClient** = **new** GoogleApiClient.Builder(**this**)

.addConnectionCallbacks(**this**)

.addOnConnectionFailedListener(**this**)

.addApi(LocationServices.***API***)

.build();

}

**private void** configLocationRequest() {

**locationRequest** = **new** LocationRequest();

**locationRequest**.setInterval(1000);

**locationRequest**.setFastestInterval(1000);

**locationRequest**.setPriority(LocationRequest.***PRIORITY\_HIGH\_ACCURACY***);

}

*/\*\**

*\* Manipulates the map once available.*

*\* This callback is triggered when the map is ready to be used.*

*\* This is where we can add markers or lines, add listeners or move the camera. In this case,*

*\* we just add a marker near Sydney, Australia.*

*\* If Google Play services is not installed on the device, the user will be prompted to install*

*\* it inside the SupportMapFragment. This method will only be triggered once the user has*

*\* installed Google Play services and returned to the app.*

*\*/*

@Override

**public void** onMapReady(GoogleMap googleMap) {

*//mMap = googleMap;*

**double** latitude = **currentLocation**.getLatitude();

**double** longitude = **currentLocation**.getLongitude();

LatLng latlng = **new** LatLng(latitude,longitude);

**mMap**.addMarker(**new** MarkerOptions().position(latlng).title(**"here.."**));

**mMap**.moveCamera(CameraUpdateFactory.*newLatLng*(latlng));

*// Add a marker in Sydney and move the camera*

*//LatLng sydney = new LatLng(-34, 151);*

*//mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));*

*//mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));*

}

@Override

**public void** onConnected(Bundle bundle) {

LocationServices.***FusedLocationApi***.requestLocationUpdates(**googleApiClient**, **locationRequest**, (LocationListener) MapsActivity.**this**);

}

@Override

**public void** onConnectionSuspended(**int** i) {

}

@Override

**public void** onConnectionFailed(ConnectionResult connectionResult) {

**int** errorCode = connectionResult.getErrorCode();

**if** (errorCode == ConnectionResult.***SERVICE\_MISSING***) {

Toast.*makeText*(**this**, **"google\_play\_service\_missing"**,Toast.***LENGTH\_LONG***).show();

}

}

@Override

**public void** onLocationChanged(Location location) {

**currentLocation** = location;

LatLng latLng = **new** LatLng(

location.getLatitude(), location.getLongitude());

**if** (**currentMarker** == **null**) {

**currentMarker** = **mMap**.addMarker(**new** MarkerOptions().position(latLng));

}

**else** {

**currentMarker**.setPosition(latLng);

}

moveMap(latLng);

}

**private void** moveMap(LatLng place) {

CameraPosition cameraPosition =

**new** CameraPosition.Builder()

.target(place)

.zoom(17)

.build();

**mMap**.animateCamera(CameraUpdateFactory.*newCameraPosition*(cameraPosition));

}

@Override

**protected void** onResume() {

**super**.onResume();

setUpMapIfNeeded();

*// 連線到Google API用戶端*

**if** (!**googleApiClient**.isConnected() && **currentMarker** != **null**) {

**googleApiClient**.connect();

}

}

@Override

**protected void** onPause() {

**super**.onPause();

*// 移除位置請求服務*

**if** (**googleApiClient**.isConnected()) {

LocationServices.***FusedLocationApi***.removeLocationUpdates(

**googleApiClient**, **this**);

}

}

@Override

**protected void** onStop() {

**super**.onStop();

*// 移除Google API用戶端連線*

**if** (**googleApiClient**.isConnected()) {

**googleApiClient**.disconnect();

}

}

**private void** addMarker(LatLng place, String title, String snippet) {

MarkerOptions markerOptions = **new** MarkerOptions();

markerOptions.position(place)

.title(title)

.snippet(snippet);

**mMap**.addMarker(markerOptions);

}

**private void** setUpMapIfNeeded() {

**if** (**mMap** == **null**) {

**mMap** = ((SupportMapFragment) getSupportFragmentManager().

findFragmentById(R.id.***map***)).getMap();

**if** (**mMap** != **null**) {

processController();

}

}

}

*// 移除地圖設定方法*

**private void** setUpMap() {

*// 建立位置的座標物件*

LatLng place = **new** LatLng(25.033408, 121.564099);

*// 移動地圖*

moveMap(place);

*// 加入地圖標記*

addMarker(place, **"Hello!"**, **" Google Maps v2!"**);

}

**private void** processController() {

*// 對話框按鈕事件*

**final** DialogInterface.OnClickListener listener =

**new** DialogInterface.OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) {

**switch** (which) {

*// 更新位置資訊*

**case** DialogInterface.***BUTTON\_POSITIVE***:

*// 連線到Google API用戶端*

**if** (!**googleApiClient**.isConnected()) {

**googleApiClient**.connect();

}

**break**;

*// 清除位置資訊*

**case** DialogInterface.***BUTTON\_NEUTRAL***:

Intent result = **new** Intent();

result.putExtra(**"lat"**, 0);

result.putExtra(**"lng"**, 0);

setResult(Activity.***RESULT\_OK***, result);

finish();

**break**;

*// 取消*

**case** DialogInterface.***BUTTON\_NEGATIVE***:

**break**;

}

}

};

*// 標記訊息框點擊事件*

**mMap**.setOnInfoWindowClickListener(**new** GoogleMap.OnInfoWindowClickListener() {

@Override

**public void** onInfoWindowClick(Marker marker) {

*// 如果是記事儲存的標記*

**if** (marker.equals(**itemMarker**)) {

AlertDialog.Builder ab = **new** AlertDialog.Builder(MapsActivity.**this**);

ab.setTitle(**"title\_update\_location"**)

.setMessage(**"title\_update\_location"**)

.setCancelable(**true**);

ab.setPositiveButton(**"update"**, listener);

ab.setNeutralButton(**"clear"**, listener);

ab.setNegativeButton(android.R.string.***cancel***, listener);

ab.show();

}

}

});

*// 標記點擊事件*

**mMap**.setOnMarkerClickListener(**new** GoogleMap.OnMarkerClickListener() {

@Override

**public boolean** onMarkerClick(Marker marker) {

*// 如果是目前位置標記*

**if** (marker.equals(**currentMarker**)) {

AlertDialog.Builder ab = **new** AlertDialog.Builder(MapsActivity.**this**);

ab.setTitle(**"title\_current\_location"**)

.setMessage(**"title\_current\_location"**)

.setCancelable(**true**);

ab.setPositiveButton(android.R.string.***ok***, **new** DialogInterface.OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) {

Intent result = **new** Intent();

result.putExtra(**"lat"**, **currentLocation**.getLatitude());

result.putExtra(**"lng"**, **currentLocation**.getLongitude());

setResult(Activity.***RESULT\_OK***, result);

finish();

}

});

ab.setNegativeButton(android.R.string.***cancel***, **null**);

ab.show();

**return true**;

}

**return false**;

}

});

}

}