Mobile Application Development

By,

Prof. Himanshu H Patel,

Prof. Hiten M Sadani

U. V. Patel College of Engineering, Ganpat University

Google Maps

- integrate google maps in our application.
- You can show any location on the map
- You can show different routes on the map
- You can also customize the map according to your choices.
- It is different than Intent(google map).

Google Maps

```xml
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" />
```

Google Maps

- **Activity** is an application component that gives a user interface where the user can interact.
- **The fragment** is a part of an activity, which contributes its own UI to that activity. ... but using multiple fragments in a single activity we can create multi-pane UI.

Customizing Google Map

- Adding Marker
- You can place a maker with some text over it displaying your location on the map
- Val mehsana: LatLng = LatLng(21 , 57);
- Val mh: Marker = googleMap.addMarker(new MarkerOptions().position(mehsana).title(“This is my city")));

Customizing Google Map

Changing Map Type

There are four different types of map and each give a different view of the map.

Customizing Google Map

- Changing Map Type
- googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
- googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
- googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
- googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);

Customizing Google Map

Types of Google Map

There are four different types of Google map are available in map API. Each of them has different view of the map. These types are Normal, Hybrid, Satellite and Terrain.

- googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
- googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
- googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
- googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);

Using Kotlin syntax, we will use the above Google map types as:

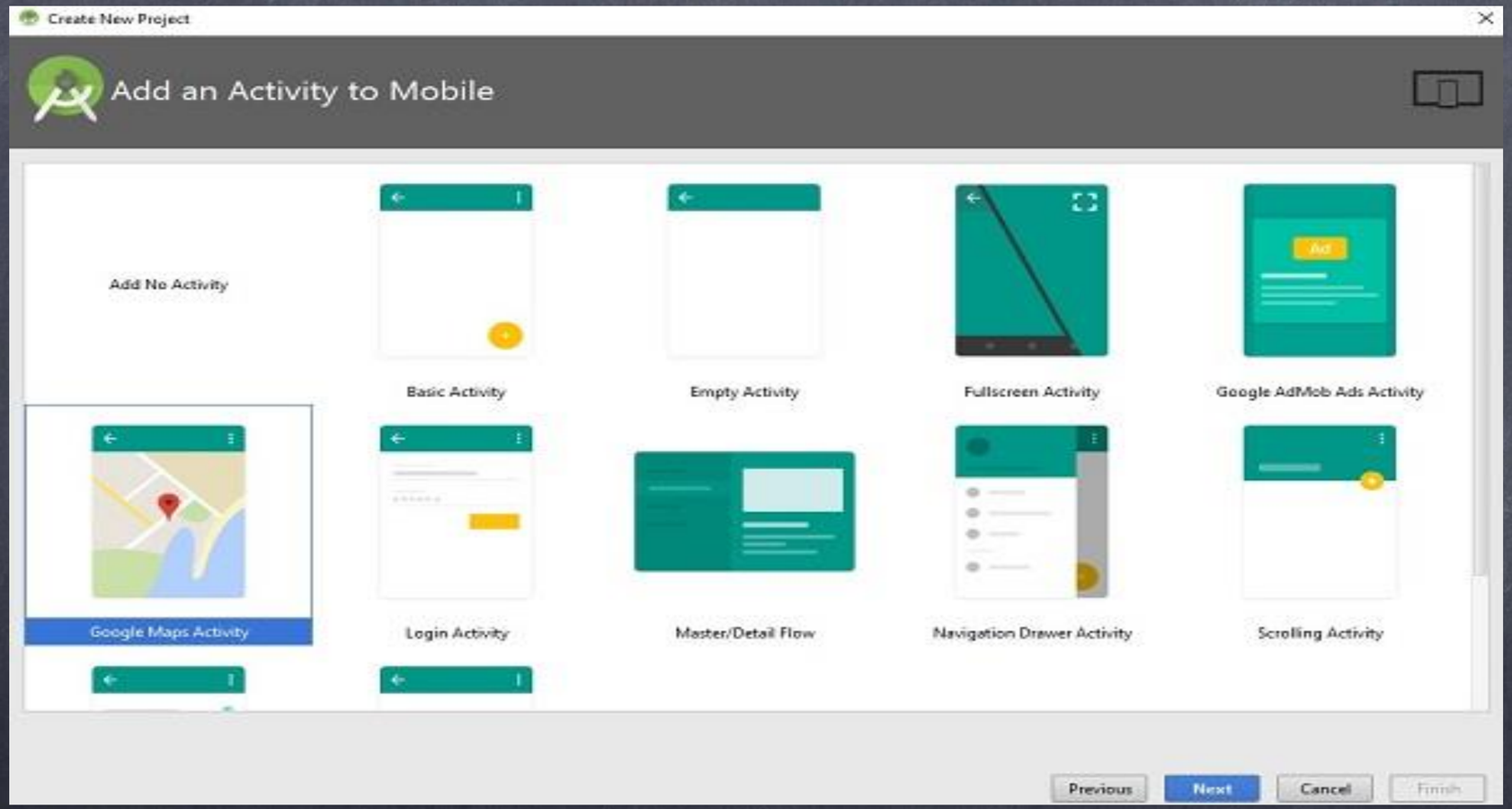
- googleMap.mapType = MAP_TYPE_NONE
- googleMap.mapType = MAP_TYPE_HYBRID
- googleMap.mapType = MAP_TYPE_SATELLITE
- googleMap.mapType = MAP_TYPE_TERRAIN

Customizing Google Map

- Enable/Disable zoom
- enable or disable the zoom gestures in the map by calling
  the **setZoomControlsEnabled(boolean)** method.
- googleMap.getUiSettings().setZoomGesturesEnable
  d(true);

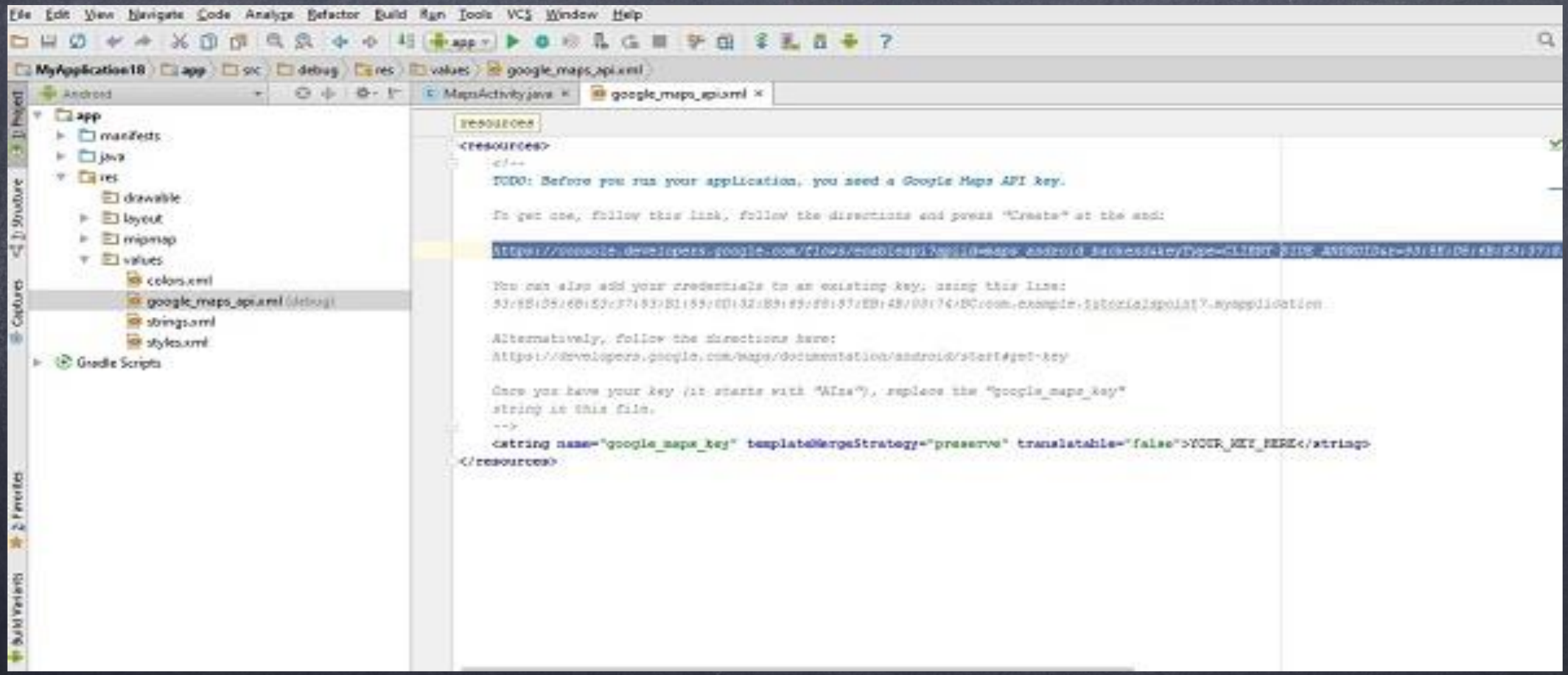
Customizing Google Map

S1: select Google Maps Activity.



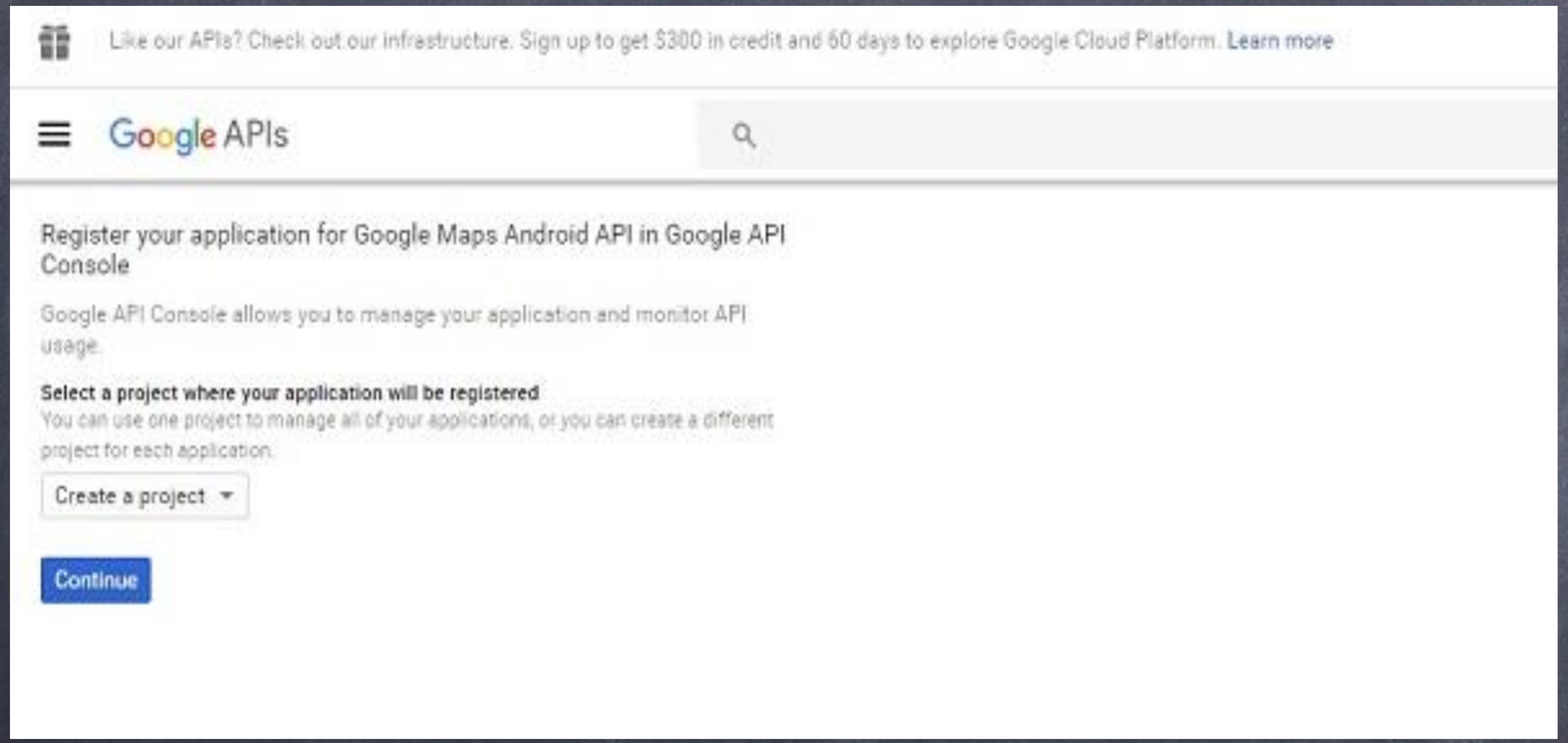
Customizing Google Map

S2: open google_maps_api.xml file
Copy url and login with gmail account



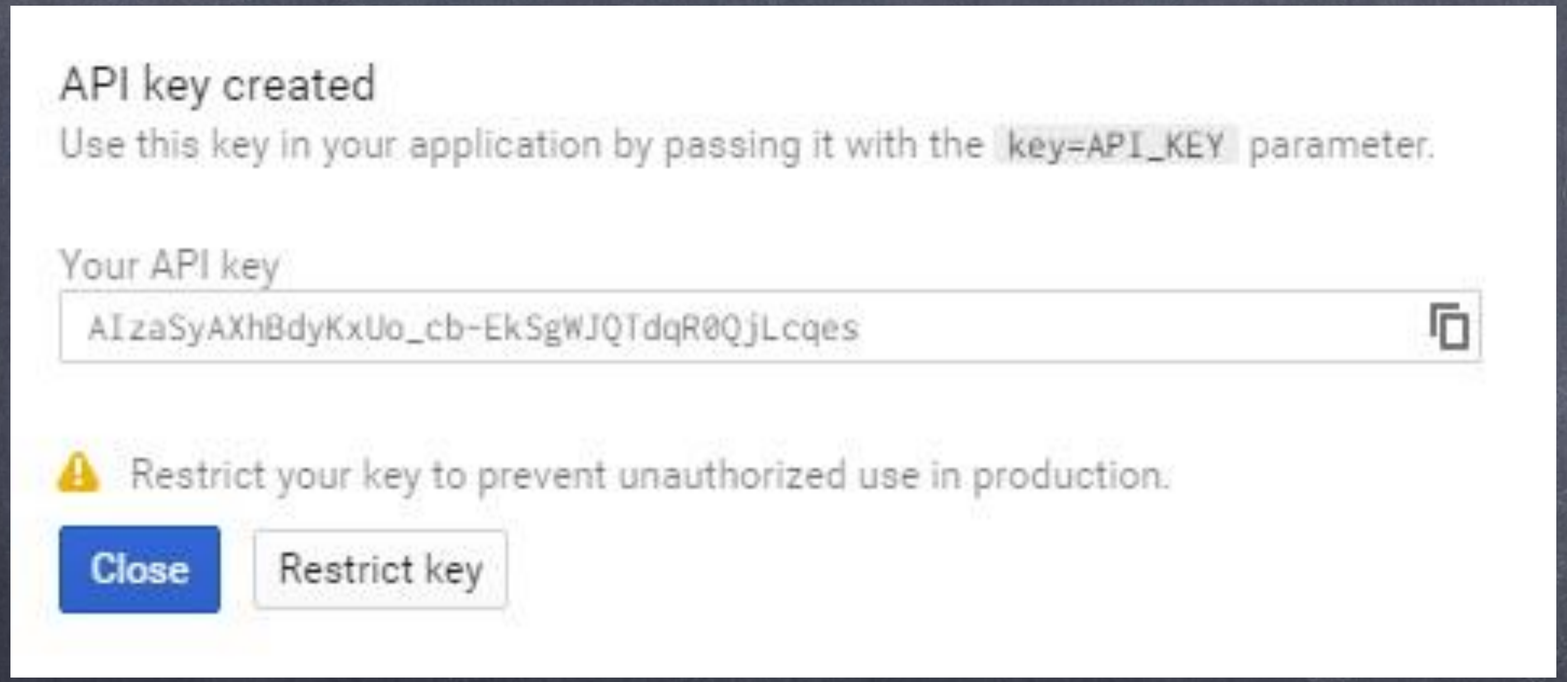
Customizing Google Map

S3: Create a project



Customizing Google Map

S4: create a API key,copy it and past it in your project xml file



Customizing Google Map

S5: Dangerous Permission and Non Dangerous permission

```
     */
    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap

        // Add a marker in Sydney and move the camera
        val sydney = LatLng(-34.0, 151.0)
        mMap.addMarker(MarkerOptions().position(sydney).title("Marker in Sydney"))
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney))
    }
}
```

Customizing Google Map

S5: Dangerous Permission and Non Dangerous permission

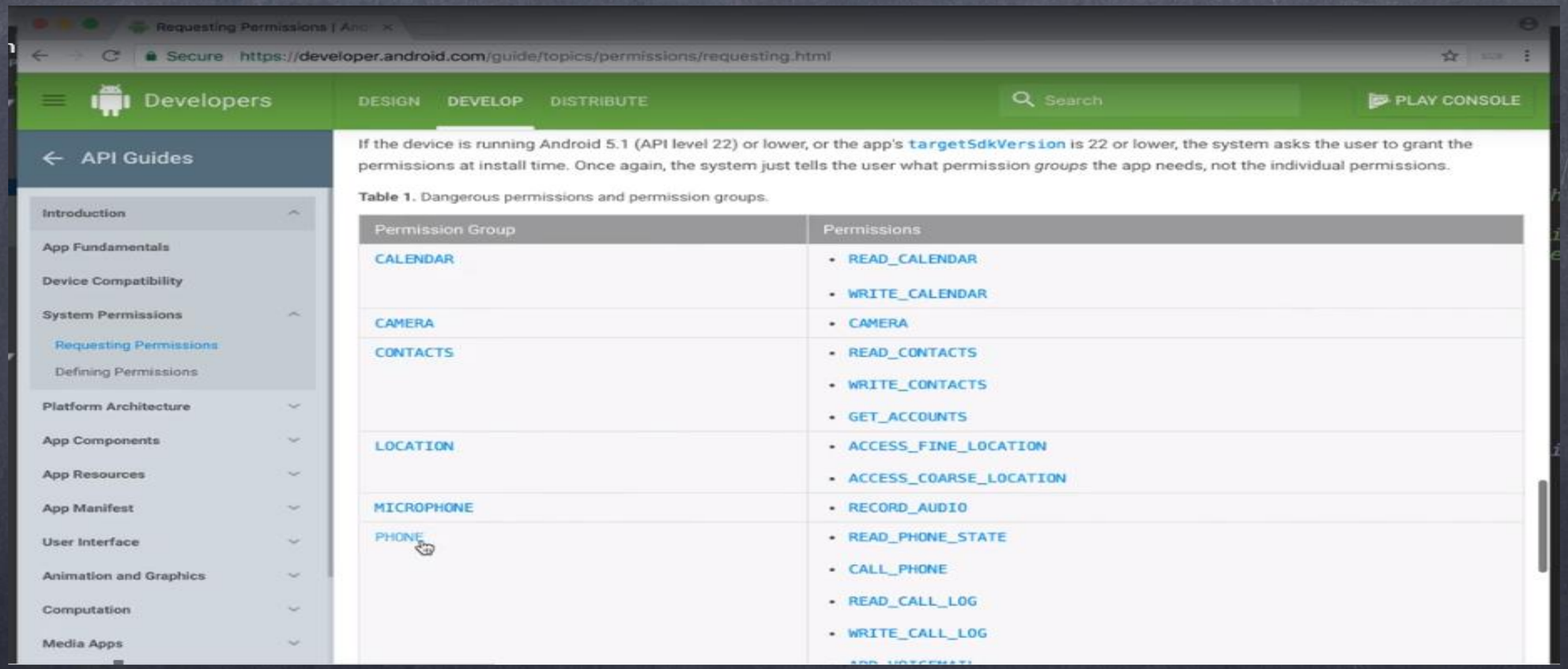
```
-->

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

Customizing Google Map

S5: Dangerous Permission and Non Dangerous permission



Customizing Google Map

S5: Dangerous Permission and Non Dangerous permission

```
var ACCESS_CODE=123
// dangerous permission
fun checkPermission(){
    // check for SDK versio of phone Marshmallow
    if(Build.VERSION.SDK_INT>=23){
      //check for self permission
        if(ActivityCompat.checkSelfPermission( context: this,
                android.Manifest.permission.ACCESS_FINE_LOCATION)!=PackageManager.PERMISSION_GRANTED){
                //request for permission
                  requestPermissions(arrayOf(android.Manifest.permission.ACCESS_FINE_LOCATION),ACCESS_CODE)
                  return;
            }
    }
    //if not then directly get the user location
        GetUserLocation()
}
```

Customizing Google Map

S5: Dangerous Permission and Non Dangerous permission

```
 // this function will call when someone call requestPermission
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {


    when(requestCode){
        ACCESS_CODE->{
            if (grantResults[0]==PackageManager.PERMISSION_GRANTED){
                GetUserLocation()
            }else{
                Toast.makeText( context: this, text: "We cannot access to your location",Toast.LENGTH_LONG).show()
            }
        }
    }
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
}
```

Customizing Google Map

S5: Dangerous Permission and Non Dangerous permission

```
 // this function will call when someone call requestPermission
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {


    when(requestCode){
        ACCESS_CODE->{
            if (grantResults[0]==PackageManager.PERMISSION_GRANTED){
                GetUserLocation()
            }else{
                Toast.makeText( context: this, text: "We cannot access to your location",Toast.LENGTH_LONG).show()
            }
        }
    }
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
}
```

Customizing Google Map

Get the User Location

```
fun GetUserLocation(){
    Toast.makeText( context: this, text: "User Location access On",Toast.LENGTH_LONG).show()
    var myLocation=MylocationListner()
    var locationManager=getSystemService(Context.LOCATION_SERVICE) as LocationManager
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, minTimeMs: 3, minDistanceM: 3f,myLocation)
    var mythread=myThread()
    myThread().start()

}
```

Customizing Google Map

Get the User Location

```
var location:Location?=null
//Get User location
inner class  MylocationListner:LocationListener{
    constructor(){
        location= Location( provider: "Start")
        location!!.longitude=0.0
        location!!.latitude=0.0
    }
    override fun onLocationChanged(p0: Location) {
        location=p0
    }
```

Customizing Google Map

Get the User Location

```
//get location and pass it to UI
inner class myThread: Thread {
    constructor():super(){
    }
    override fun run() {
        while (true){
            try {
                runOnUiThread() {
                    mMap!!.clear()
                    val sydney = LatLng(location!!.Latitude, location!!.Longitude)
                    mMap.addMarker(
                        MarkerOptions().position(sydney)
                            .title("Me")
                            .snippet("here is my location")
                            .icon(BitmapDescriptorFactory.fromResource(R.drawable.mario))
                    )
                    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, 14f))
                }
                Thread.sleep( millis: 1000)
            }catch (ex:Exception){

            }
```