

# Google Map의 활용



# 목차

---

 Google Map 사용 준비

 지도 상의 위치 이동 처리

 지도 위에 그리기

- ◆ 마커 표시
- ◆ 이동 경로 표시

 지도 이벤트 처리

- ◆ 마커 클릭 이벤트
- ◆ 지도 클릭 이벤트
- ◆ 지도 롱클릭 이벤트

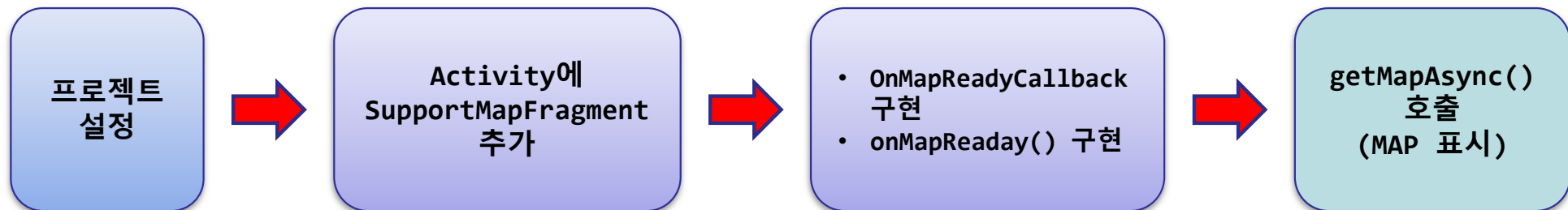
# 개요

## 앱 내부에서의 지도 사용

- ◆구글에서 제공하는 맵 위젯을 내장
- ◆네트워크를 통해 지도 정보 수신
- ◆Fragment로 Activity에 포함
  - SupportMapFragment

- Fragment
  - 앱 UI 의 재사용 가능 부분
  - Activity 내부를 구성하는 Activity의 하위 화면 단위
  - 하나의 Activity 안에 여러 개의 Fragment 사용 가능
  - 하나의 Fragment를 여러 Activity 에서 재활용 가능
  - 자신 만의 lifecycle을 갖고 있으나, Activity lifecycle에 종속

## 사용 절차



- [프로젝트 설정] 은 최초 1회만 설정
- 안드로이드 스튜디오에 필요 SDK 추가
- Google API 사용을 위한 API 정보 설정
- AndroidManifest 정보 설정

# Google Map 사용 절차

## AndroidStudio 에서의 Map 사용 준비

1. Google Play Services 준비
2. Google Map library 설정

## Google Open API 정보 설정

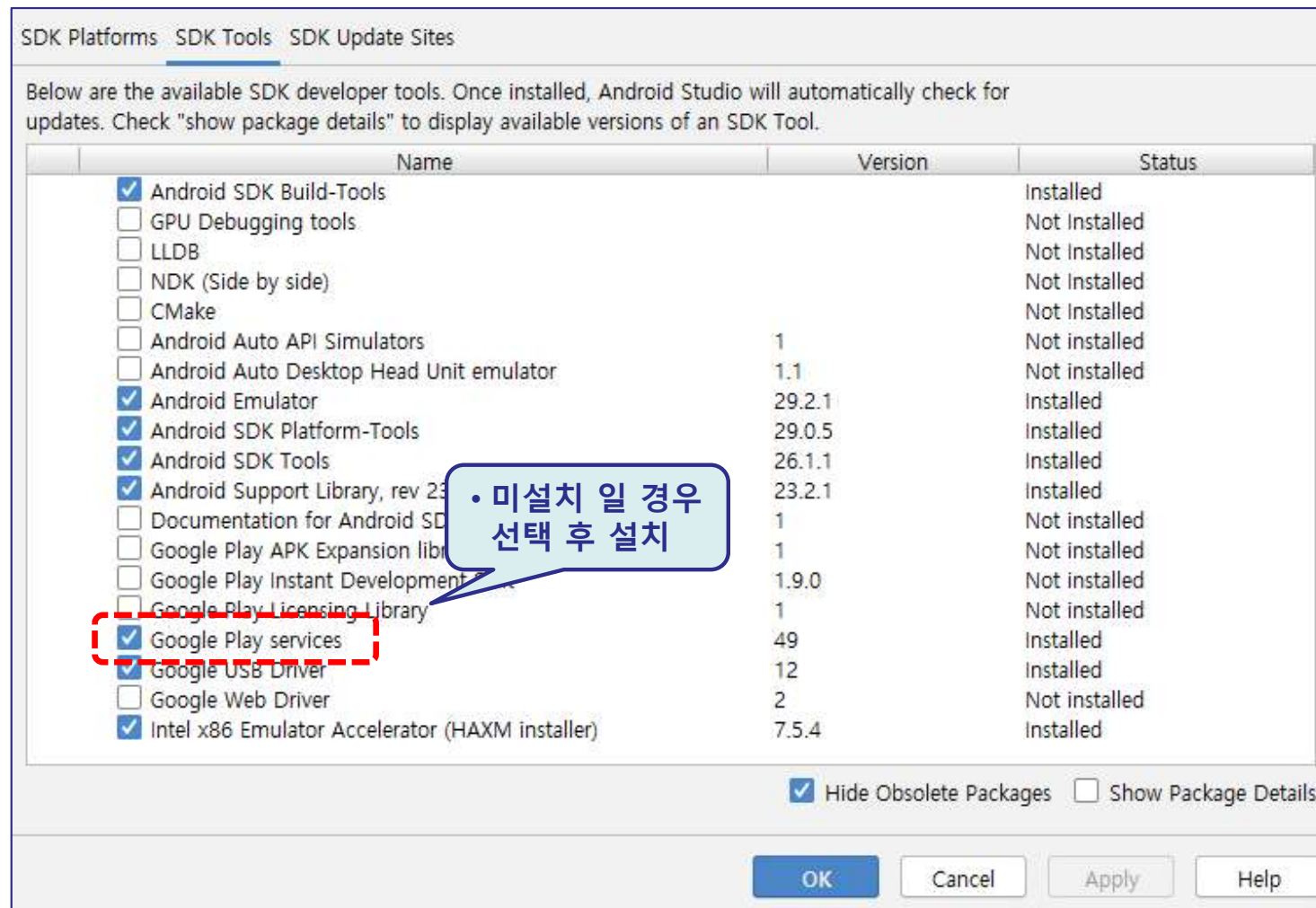
1. API Key 발급을 위한 인증키 준비 → Google 에 접속 전  
에 준비
2. Google Map API Key 발급
3. Android Manifest 설정
4. 레이아웃에 Google Map MapFragment 추가 → 하위 버  
전 호환을 위해 SupportMapFragment 사용

# AndroidStudio 에서의 Map 사용 준비 1

## Google Play Services SDK 설치

### ◆ Android SDK Manager 실행

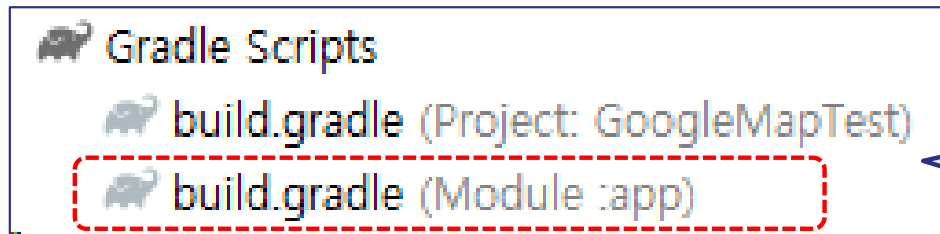
■ [Tools] → [Android] → [SDK Manager] → [SDK Tools]



# AndroidStudio 에서의 Map 사용 준비 2

## Google Map library 추가

- ◆ GooglePlay Service 의 일부로 포함되어 있으므로 GooglePlay Service 라이브러리를 프로젝트에 추가
- ◆ build.gradle (Module:app) 에서 확인



- compileSdk 31 이상
- minSdk 19 이상

• 현재 최신 버전은 18.2.0

```
// GoogleMap
implementation 'com.google.android.gms:play-services-maps:18.2.0'
```

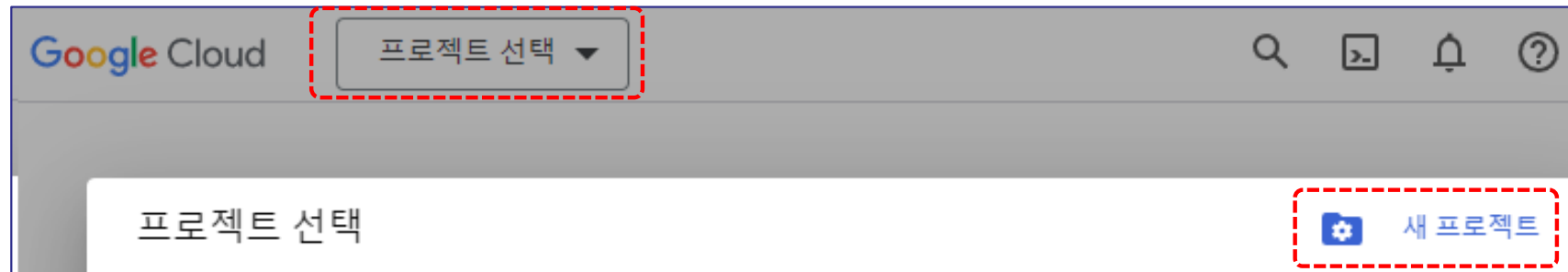
- play-services 중 Google Map만 사용할 경우
- 버전은 계속 변경될 수 있으므로 개발 사이트 참조

• 참조 링크: <https://developers.google.com/android/guides/setup>

# Google OpenAPI 사용 준비 1

## Google Map API Key 발급

- ◆ <https://console.cloud.google.com> 접속 → 구글 정보로 로그인  
→ 화면 상단의 [프로젝트 선택] → [새 프로젝트] 선택
  - 프로젝트 이름: My GoogleMap Test, 위치: 조직 없음



새 프로젝트

⚠ projects 할당량이 23개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제하세요. [자세히 알아보기](#)

[MANAGE QUOTAS](#)

프로젝트 이름 \*  
My GoogleMap Test

프로젝트 ID: enhanced-emblem-405407입니다. 나중에 변경할 수 없습니다. [수정](#)

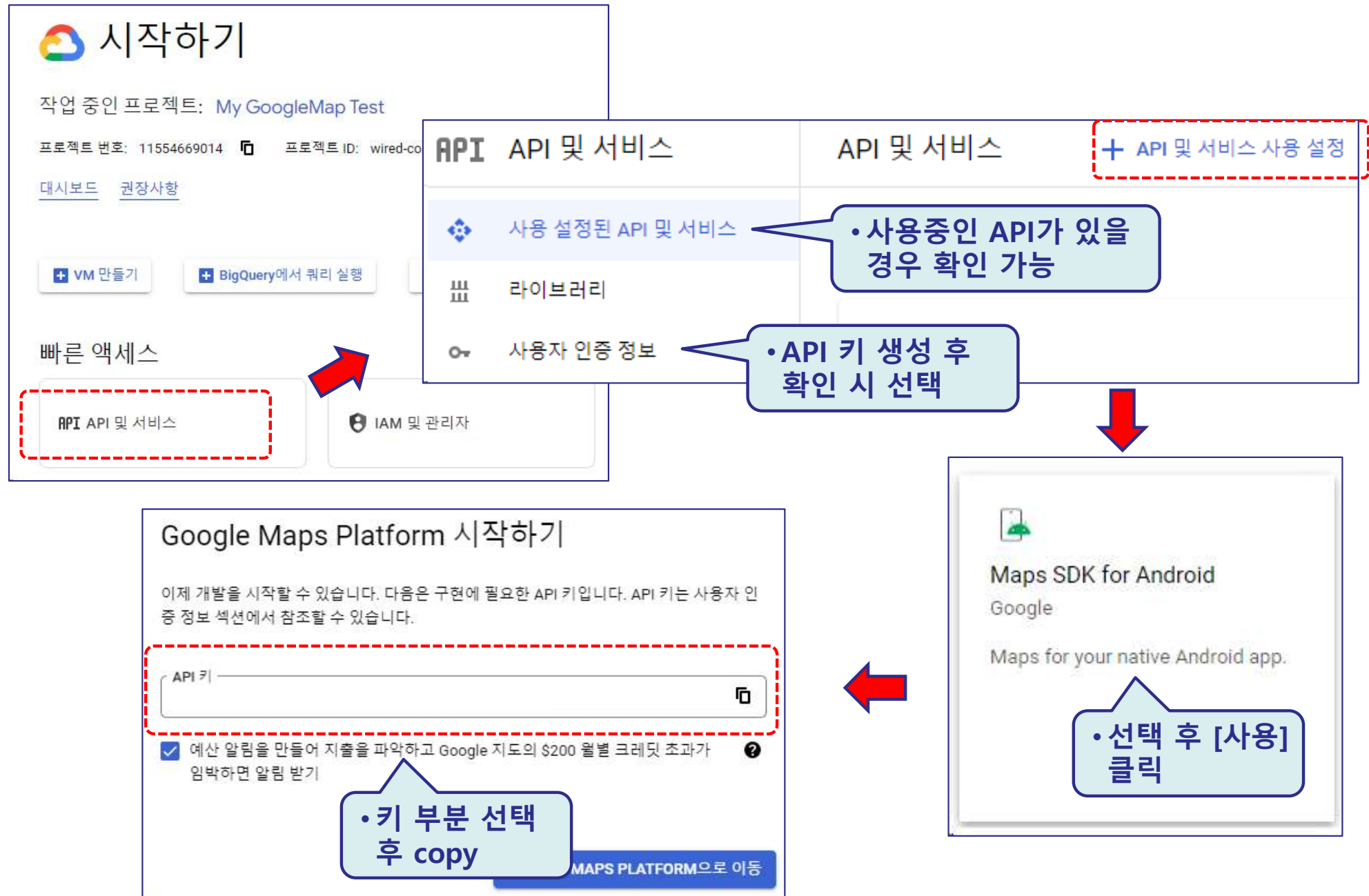
위치 \*  
📍 조직 없음 [찾아보기](#)

상위 조직 또는 폴더

[만들기](#) [취소](#)

• 프로젝트 이름 및  
위치 입력

# Google OpenAPI 사용 준비 2





# Google OpenAPI 사용 준비 3

## 발급받은 API 키의 사용 제한

- ◆ 제한하지 않을 경우 플랫폼(웹/모바일 등)에 상관 없이 API 키 사용 가능

### API 키 보호

원치 않는 악의적 사용으로부터 계정을 보호하기 위해 Google API 키를 제한하는 것이 좋습니다. [자세히 보기](#)

Select restriction type

- HTTP 리퍼러(웹사이트)
- IP 주소(웹 서버, 크론 작업 등)
- Android 앱**
- iOS 앱
- API 제한 없음

### API 키 보호

원치 않는 악의적 사용으로부터 계정을 보호하기 위해 Google Maps Platform API 키를 제한하는 것이 좋습니다. [자세히 보기](#)

Select restriction type  
Android 앱

Android 앱의 사용량을 제한하려면 패키지 이름과 SHA-1 서명 인증서 디지털 지문을 추가하세요.

패키지 이름 \*  
ddwu.com.mobile.googlemaptest

SHA-1 인증서 디지털 지문 \*

[나중에]

키 제한

• 앱 패키지명 기록

• Android 앱에서만 사용하고자 할 경우 정보 설정

• [나중에] 선택

• 개발환경의 디지털 지문 기록 (SHA-1)

# Google MAP 추가 1

## Android Manifest 설정

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

```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="GoogleMap Test"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.GoogleMapTest"
    tools:targetApi="31">
```

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="GoogleMap API key 값 기록" />
```

```
<activity
    android:name=".MainAc
```

• 필요 permission 추가  
(위치 정보 사용)

• 생성한 GoogleMap key값

• API 키를 보다 안전하게 지정해야 할 경우  
[https://developers.google.com/maps/documentation/android-sdk/config?hl=ko#step\\_3\\_add\\_your\\_api\\_key\\_to\\_the\\_project](https://developers.google.com/maps/documentation/android-sdk/config?hl=ko#step_3_add_your_api_key_to_the_project) 참조

# Google MAP 추가 2

Google Map Fragment를 Layout 에 배치 (입력)

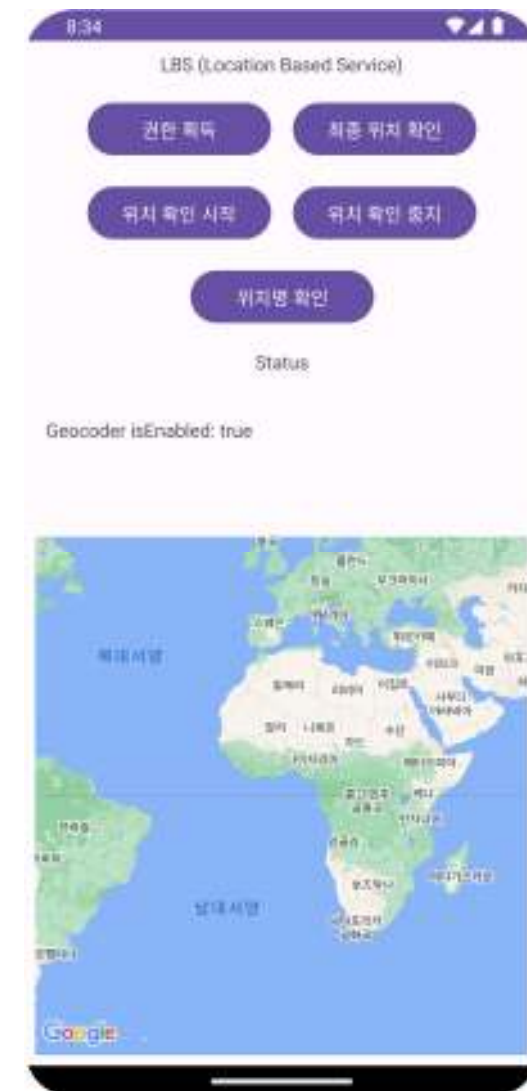
```
<fragment
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

• Resource ID

• 지도 Fragment

실행 화면

- ◆ 초기 레이아웃 디자인 시 지도를 확인할 수 없을 경우  
→ 레이아웃 매니저에서 경고 무시 선택
- ◆ 레이아웃의 다른 요소들과 배치



# 지도의 초기 상태 설정 (in XML)

## <Fragment> 에 값 설정의 예

- map 네임스페이스를 사용하기 위해 기록
- 상위 태그에 있을 경우 생략 가능

<fragment

xmlns:map="http://schemas.android.com/apk/res-auto"

android:id="@+id/map"

android:name="com.google.android.gms.maps.SupportMapFragment"

map:cameraBearing="112.5"

map:cameraTargetLat="37.606537"

map:cameraTargetLng="127.041758"

map:cameraZoom="13"

map:cameraTilt="50"

map:mapType="normal"

map:uiCompass="true"

map:uiRotateGestures="true"

map:uiScrollGestures="false"

map:uiTiltGestures="true"

map:uiZoomControls="true"

map:uiZoomGestures="true"

...

- SupportMapFragment 의 정적 속성 값 지정

- 지도의 초기 상태 값을 XML 로 지정
- 실행 중 코드에서 값 변경 가능

- 참고 링크:

<https://developers.google.com/maps/documentation/android-sdk/configure-map?hl=ko>

# 레이아웃의 지도를 코드로 가져오기

```
private lateinit var googleMap : GoogleMap
```

• 지도를 저장할  
멤버변수 선언

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(mainBinding.root)  
    ...
```

• map Fragment 객체 준비  
(SupportMapFragment)

```
val mapFragment: SupportMapFragment  
    = supportFragmentManager.findFragmentById(R.id.map)  
    as SupportMapFragment
```

```
mapFragment.getMapAsync (mapReadyCallback)
```

• map 정보 가져오기  
(Callback 호출)

```
}
```

• map 정보 가져오기  
완료 확인 Callback

```
val mapReadyCallback = object: OnMapReadyCallback {  
    override fun onMapReady(map: GoogleMap) {  
        googleMap = map  
        Log.d(TAG, "GoogleMap is ready")  
    }  
}
```

• map 정보 가져오기 완료 시  
멤버변수에 저장

• map 가져오기 후 초기에 수  
행하여야 할 작업이 있을 경  
우 해당내용 구현

•참고 링크:

[https://developers.google.com/maps/documentation/android-sdk/map#change\\_the\\_map\\_type](https://developers.google.com/maps/documentation/android-sdk/map#change_the_map_type)

# 지도 위치 이동하기

## 특정 위치로 이동하기

```
val targetLoc: LatLng = LatLng(37.606320, 127.041808)
// 바로 이동
googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(targetLoc, 17F))
// 애니메이션 효과를 보이며 이동
googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(targetLoc, 17F))
```

• 위도/경도/줌배율  
(float) 지정

## GPS 수신위치로 이동하기

- ◆ FusedLocationProviderClient와 LocationCallback 사용
- ◆ LocationCallback의 onLocationResult()에서 전달 받은 위치로 지도의 위치 변경
  - 전달받은 LocationResult 객체에서 위치 정보(longitude, latitude) 확인

# 위치 수신정보 추가 - LBS와 연결

```
// 현재 위치를 저장한 Location 변수
private lateinit var currentLoc : Location

val locCallback : LocationCallback = object : LocationCallback() {
    override fun onLocationResult(locResult: LocationResult) {
        currentLoc = locResult.locations.get(0) // 수신한 위치 중 첫번째 위치를 기록

        val targetLoc: LatLng = LatLng( currentLoc.latitude, currentLoc.longitude)

        // 애니메이션 효과를 보이며 이동
        googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(targetLoc, 17F))
    }
}

val locRequest = LocationRequest.Builder(5000)
    .setMinUpdateIntervalMillis(3000)
    .setPriority(Priority.PRIORITY_BALANCED_POWER_ACCURACY)
    .build()

private fun startLocUpdates() {
    fusedLocationClient.requestLocationUpdates(
        locRequest,      // LocationRequest 객체
        locCallback,      // LocationCallback 객체
        Looper.getMainLooper() // System 메시지 수신 Looper
    )
}
```

- 위치 정보를 수신할 때마다 해당 위치로 지도의 중심 변경

# 지도 위에 그리기 1 - Marker

```
var centerMarker : Marker? = null // 마커 보관이 필요할 경우 변수 선언
```

```
fun addMarker(targetLoc: LatLng) {
```

```
    val markerOptions: MarkerOptions = MarkerOptions() // 마커를 표현하는 Option 생성
    markerOptions.position(targetLoc) // 필수
        .title("마커 제목")
        .snippet("마커 말풍선")
        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED))
    // .icon(BitmapDescriptorFactory.fromResource(R.mipmap.android))
```

```
    centerMarker = googleMap.addMarker(markerOptions) // 지도에 마커 추가, 추가마커 반환
    centerMarker?.showInfoWindow() // 마커 터치 시 InfoWindow 표시
    centerMarker?.tag = "database_id" // 마커에 관련 정보(Object) 저장
```

• Marker 설정 및 추가

```
// 마커 클릭 이벤트 처리
```

```
googleMap.setOnMarkerClickListener { marker ->
    Toast.makeText(this, marker.tag.toString(), Toast.LENGTH_SHORT).show()
    false // true일 경우 이벤트처리 종료이므로 info window 미출력
}
```

```
// 마커 InfoWindow 클릭 이벤트 처리
```

```
googleMap.setOnInfoWindowClickListener { marker ->
    Toast.makeText(this, marker.title, Toast.LENGTH_SHORT).show()
}
```

```
}
```



# 지도 위에 그리기 2 - Marker

## 마커의 이동

- ◆ 위도/경도를 저장한 LatLng 객체 전달

```
addedMarker?.position = LatLng(37.606320, 127.041808)
// addedMarker 는 지도에 추가한 Marker 객체
```

- ◆ 새로운 LatLng 전달 시 해당 위치로 이동

## 마커의 삭제

```
addedMarker?.remove()
```

•참고 링크:

<https://developers.google.com/maps/documentation/android-sdk/marker?hl=ko>

# 지도 위에 그리기 3 - PolyLine

```
fun drawLine() {  
    // Line 을 표시하는 옵션 생성  
    val polylineOptions = PolylineOptions()  
        // 선 색상 지정, 여러 번 지정 가능  
        .addSpan( StyleSpan(Color.RED) )  
        // LatLng 로 선 지점 추가  
        .add( LatLng(37.604151, 127.042453) )  
        .add( LatLng(37.605347, 127.041207) )  
        .add( LatLng(37.606038, 127.041344) )  
        .add( LatLng(37.606220, 127.041674) )  
        .add( LatLng(37.606631, 127.041595) )  
        .add( LatLng(37.606823, 127.042380) )  
  
    // 지도에 Line 추가, 추가된 Line 반환  
    val line = googleMap.addPolyline(polylineOptions)  
    // line.remove() 로 선 삭제  
}
```

•참고 링크:

<https://developers.google.com/maps/documentation/android-sdk/marker?hl=ko>

# 지도 이벤트 처리

## 지도 클릭/롱클릭 이벤트

- ◆ 지도 상의 임의의 위치 클릭/롱클릭 이벤트 처리

```
// 지도의 특정 지점 클릭 이벤트 처리
googleMap.setOnMapClickListener { latLng : LatLng ->
    Toast.makeText(applicationContext,
        latLng.toString(), Toast.LENGTH_SHORT).show()
}

// 지도의 특정 지점 롱클릭 이벤트 처리
googleMap.setOnMapLongClickListener { latLng : LatLng ->
    Toast.makeText(applicationContext,
        latLng.toString(), Toast.LENGTH_SHORT).show()
}
```

# 실습 1

 앱을 실행시키면 가장 마지막으로 수신한 위치로 지도를 이동하여 마커 표시하기

- ◆ FusedLocationProviderClient 의 getLastLocation() 사용
  - 위치가 있을 경우 마지막 위치의 Location 객체 반환
  - 수신 위치가 없을 경우 null 반환
- ◆ 마지막으로 수신한 위치가 없을 경우 학교 위치에 마커 표시

## 실습 2

 [위치 확인 시작] 버튼을 누르면 GPS에서 수신한 위치로 마커 이동

- ◆수신간격은 3초, 5m
- ◆Fake GPS 앱에서 이동 시뮬레이션을 활성화한 후 테스트

 마커 이동 시 이동한 위치를 선으로 계속 연결하여 표시

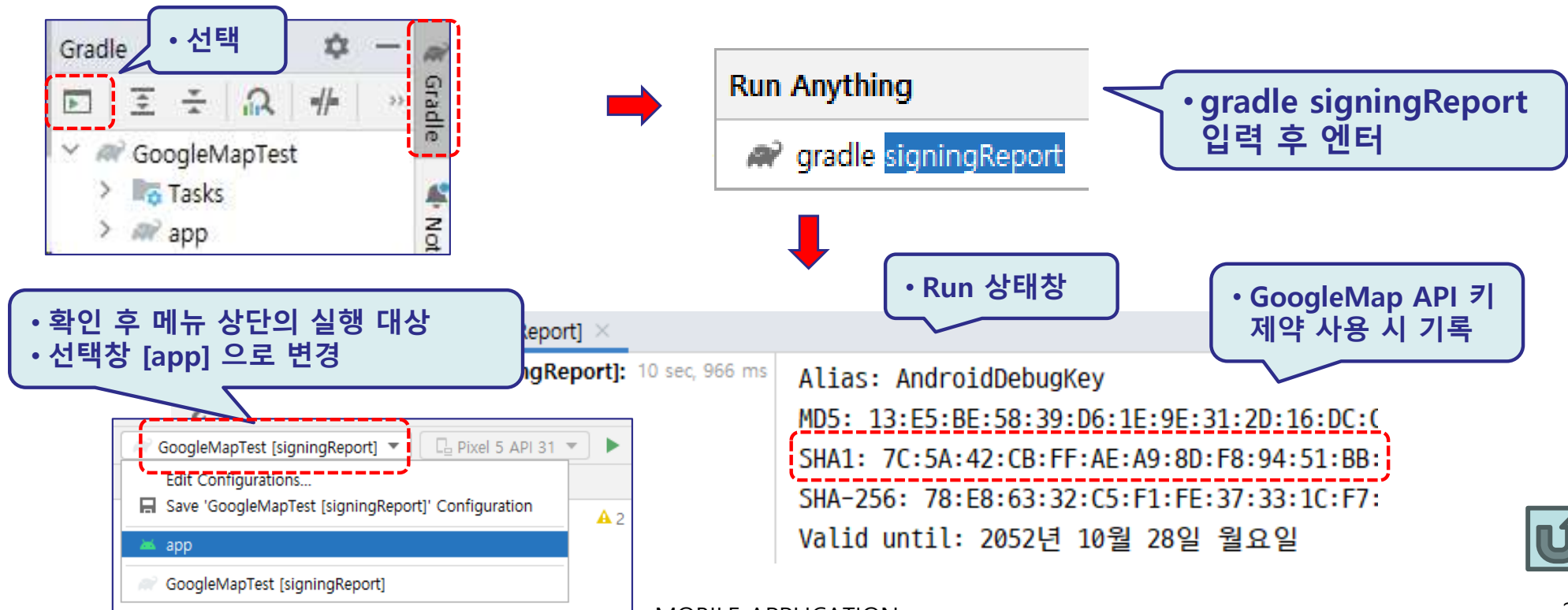
 지도를 롱클릭할 경우 새로운 마커 추가

- ◆새로운 마커 추가 시 snippet 에 추가 위치의 주소 기록 → geocoding 사용

# 참고 – API 인증 키 확인

## Google OpenAPI Key 발급을 위한 인증키

- ◆ Google 접속 전 개발 PC에 확인
- ◆ API 이용 대상을 제한하지 않을 경우 생략
- ◆ Google Map API 키의 API 이용 플랫폼 제한 설정 시에 필요 → 배포키 vs. 디버그용 키
  - Command line 상에서 keytool 명령어로 확인 가능
  - AndroidStudio 우측 [Gradle] 에서 컴파일 후 확인 가능



# 참고

---

## Fragment

- ◆ <https://developer.android.com/guide/fragments?hl=ko>

## Google Cloud 플랫폼 설정 (OpenAPI 사용)

- ◆ <https://developers.google.com/maps/documentation/android-sdk/cloud-setup?hl=ko>

## GoogleMap 추가

- ◆ <https://developers.google.com/maps/documentation/android-sdk/map?hl=ko>

## GoogleMap 에 그리기 (Marker, PolyLine 등)

- ◆ <https://developers.google.com/maps/documentation/android-sdk/marker?hl=ko>