



23장. 지도 활용

23.1 지도 출력

23.1.1. 구글 지도를 위한 설정

play-service 라이브러리 준비

```
implementation 'com.google.android.gms:play-services:12.0.1'
```

API 키 준비

개발자 PC에서 SHA-1(Secure Hash Algorithm 1) 지문

```
keytool -list -v -keystore debug.keystore -alias androiddebugkey -storepass android -keypass android
```

구글 개발자 콘솔

<https://console.developers.google.com>

AndroidManifest.xml 파일에 추가

API 키

이름

test8-23

키 제한사항

키 제한사항을 통해 이 키를 사용할 수 있는 웹사이트, IP 주소 또는 앱을 지정할 수 있습니다. 자세히 알아보기

☐ 없음

☐ HTTP 리퍼러(웹사이트)

☐ IP 주소(웹 서버, 크론 작업 등)

☒ Android 앱

☐ iOS 앱

Android 앱의 사용량 제한 (선택사항)

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

AndroidManifest.xml 파일에서 패키지 이름을 가져온 후 다음 명령어를 사용하여 지문을 가져오세요.

```
$ keytool -list -v -keystore mystore.keystore
```

패키지 이름

SHA-1 인증서 지문

com.example.test8_23

6A:64:41:3D:10:8C:ED:3B:9A:F3:4C:7F:6E:9A:AF:B1:39:AC:D5:DB

+ 패키지 이름 및 지문 추가

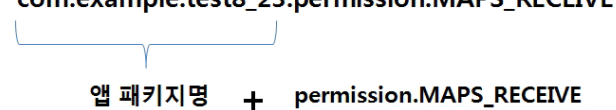
```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="~~~~~" />
```

23.1 지도 출력

- AndroidManifest.xml 파일 준비
- <permission> 태그로 퍼미션을 하나 정의

```
<permission
    android:name="com.example.test8_23.permission.MAPS_RECEIVE"
    android:protectionLevel="signature" />
<uses-permission android:name="com.example.test8_23.permission.MAPS_RECEIVE" />
```

com.example.test8_23.permission.MAPS_RECEIVE



앱 패키지명 + permission.MAPS_RECEIVE

- 필요한 여러 <uses-permission>을 선언

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

- play-service 라이브러리의 버전

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

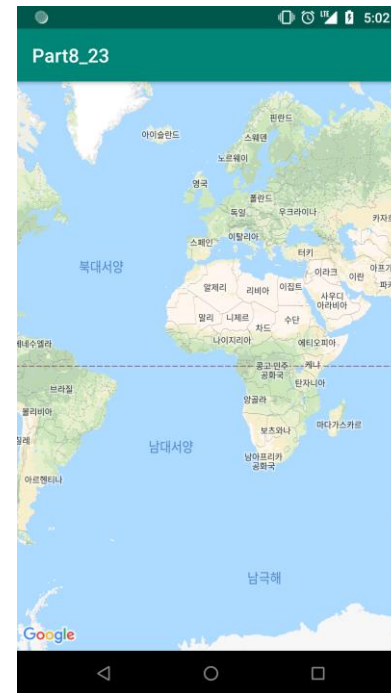
23.1 지도 출력

23.1.2. 구글 지도 화면 출력

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mission1_map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.google.android.gms.maps.SupportMapFragment"/>
```

Step by Step 23-1 – 지도 출력까지의 작업

- 모듈 생성
- play-service 라이브러리 추가
- SHA-1 지문 획득
- 구글 개발자 콘솔 작업
- AndroidManifest.xml 파일 작업
- activity_main.xml 작성
- 실행



23.2 구글 지도 제어

23.2.1. 지도 중심 위치 이동

- GoogleMap 객체

```
public class MainActivity extends AppCompatActivity implements OnMapReadyCallback{
    GoogleMap map;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        //...
        ((SupportMapFragment) getSupportFragmentManager().
findFragmentById(R.id.lab1_map)).getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        map=googleMap;
    }
}
```

- 지도의 중심 위치 이동

```
LatLng latLng = new LatLng(37.566643, 126.978279);
CameraPosition position = new CameraPosition.Builder()
    .target(latLng).zoom(16f).build();
map.moveCamera(CameraUpdateFactory.newCameraPosition(position));
```

23.2 구글 지도 제어

23.2.2. GoogleMap 이벤트 처리

OnMapClickListener: 지도의 특정 위치를 클릭

- OnMapLongClickListener: 지도의 특정 위치를 롱클릭
- OnCameraMoveListener: 지도의 중심 위치가 변경되거나 확대 수준이 변경
- OnCameraIdleListener: 지도의 중심 위치, 확대 수준 변경이 완료된 순간

```
map.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
    public void onMapClick(LatLng latLng) {}
});
map.setOnMapLongClickListener(new GoogleMap.OnMapLongClickListener() {
    public void onMapLongClick(LatLng latLng) {}
});
map.setOnCameraMoveListener(new GoogleMap.OnCameraMoveListener() {
    public void onCameraMove() {
        int zoom=(int)map.getCameraPosition().zoom;
        String center=map.getCameraPosition().target.latitude+":"+
            map.getCameraPosition().target.longitude;
    }
});
map.setOnCameraIdleListener(new GoogleMap.OnCameraIdleListener() {
    public void onCameraIdle() {
        int zoom=(int)map.getCameraPosition().zoom;
        String center=map.getCameraPosition().target.latitude+":"+
            map.getCameraPosition().target.longitude;
        //...    }
});
```

23.2 구글 지도 제어

23.2.3. 현재 위치 표시

- 마커 보이기
- 지도에 표시되는 이미지

```
MarkerOptions markerOptions = new MarkerOptions();  
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.ic_marker));  
markerOptions.position(latLng);  
markerOptions.title("서울시청");  
markerOptions.snippet("Tel:02-120");
```



addMarker() 함수로 지도 위에 마커를 출력

마커 이벤트 처리

- OnMarkerClickListener: 마커를 클릭한 순간의 이벤트
- OnInfoWindowClickListener: 마커의 정보 창을 클릭한 순간의 이벤트

```
map.setOnMarkerClickListener(new GoogleMap.OnMarkerClickListener() {  
    public boolean onMarkerClick(Marker marker) {  
        if(marker==marker1){  
            return false;  
        }  
    });  
map.setOnInfoWindowClickListener(new GoogleMap.OnInfoWindowClickListener() {  
    public void onInfoWindowClick(Marker marker) {}  
});
```


23.2 구글 지도 제어

23.2.4. 주소와 위경도 변환

Geocoder는 위경도를 주소 문자열로 혹은 주소를 위경도 값으로 변형시키기 위한 프로그램

- Geocoding: 주소를 위경도로 변환
- Reverse Geocoding: 위경도를 주소로 변환

23.2 구글 지도 제어

- Geocoding : 주소를 위경도로 변환

```
public void run() {
    Geocoder geocoder = new Geocoder(MainActivity.this);

    List<Address> addresses = null;
    String addressText = "";
    try {
        addresses = geocoder.getFromLocation(latLng.latitude,
            latLng.longitude, 1);

        if (addresses != null && addresses.size() > 0) {
            Address address = addresses.get(0);
            addressText = address.getAdminArea()+"
            address.getAddressLine(0) : address.getLocality()+" "
            "+(address.getMaxAddressLineIndex() > 0 ?

            String txt=address.getSubLocality();

            if(txt != null)
                addressText += txt+ " ";

            addressText += address.getThoroughfare() + " "
                + address.getSubThoroughfare();
            //...
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

23.2 구글 지도 제어

- 결과 주소 문자열은 Address 타입
- getAdminArea(): 서울특별시
- getMaxAddressLineIndex(): 0
- getAddressLine(0): 대한민국 서울특별시 중구 명동 35
- getLocality(): 중구
- getSubLocality(): null
- getThoroughfare(): 명동
- getSubThoroughfare(): 35
- Reverse Geocoding : 위경도를 주소로 변환

```
public void run() {  
    Geocoder geocoder = new Geocoder(MainActivity.this);  
    try{  
        List<Address> results = geocoder.getFromLocationName(address, 1);  
        Address resultAddress=results.get(0);  
        LatLng latLng=new LatLng(resultAddress.getLatitude(), resultAddress.getLongitude());  
  
        //...  
    }catch (Exception e){  
        e.printStackTrace();  
    }  
}
```

Step by Step 23-2 - GoogleMap

- 파일 복사
- MainActivity 작성
- 실행

