2017117061 송종근

맡은 역할 : 카카오맵 구현, 구글지도 구현, 카카오페이 구현

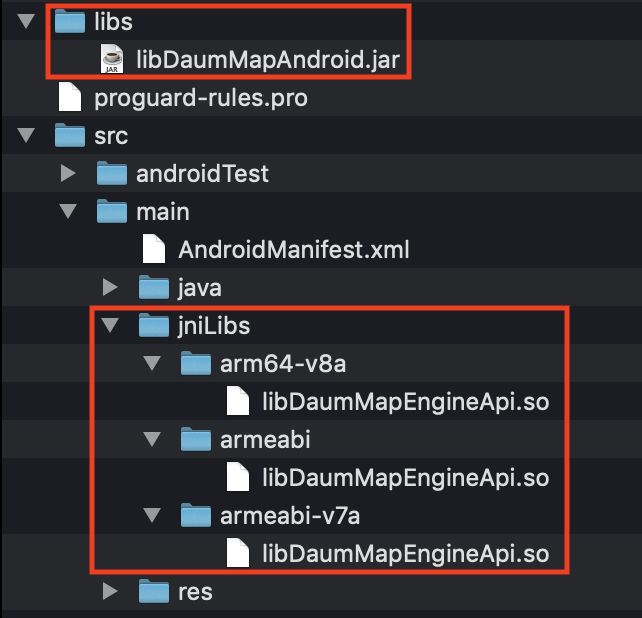
**카카오맵 구현** (카카오 개발자 페이지 공식 가이드 문서 참조)

1. 카카오 개발자 페이지에서 네이티브 앱 키 발급 및 해시 키 등록하기



위 그림의 코드를 아무것도 건드리지 않은 기본 메인 액티비티에서 onCreate() 외부에 넣은 후, onCreate 내부에 getHashKey()를 호출하면 해시키 획득 가능.

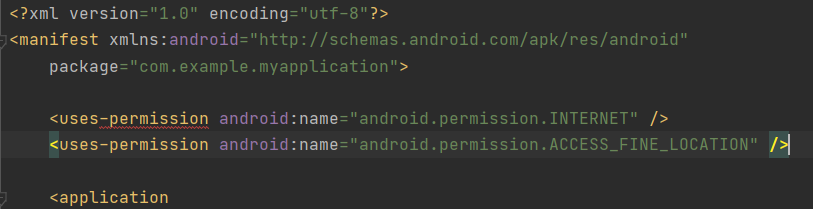
2. 라이브러리 파일 추가



카카오맵 api 가이드 문서에서 필요한 라이브러리를 다운받은 후 위의 형태와 같이 붙여넣기.

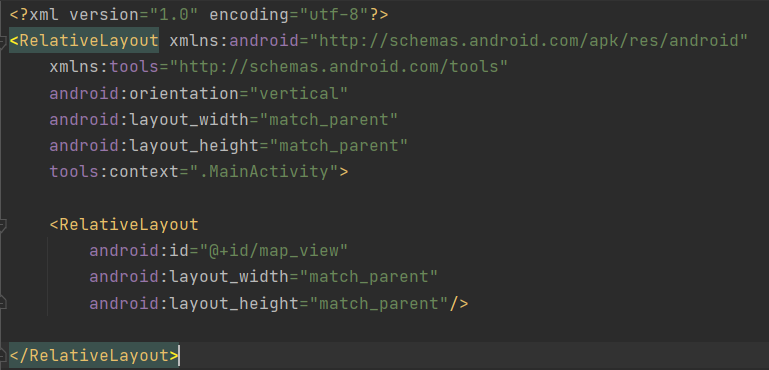


의존성 추가.



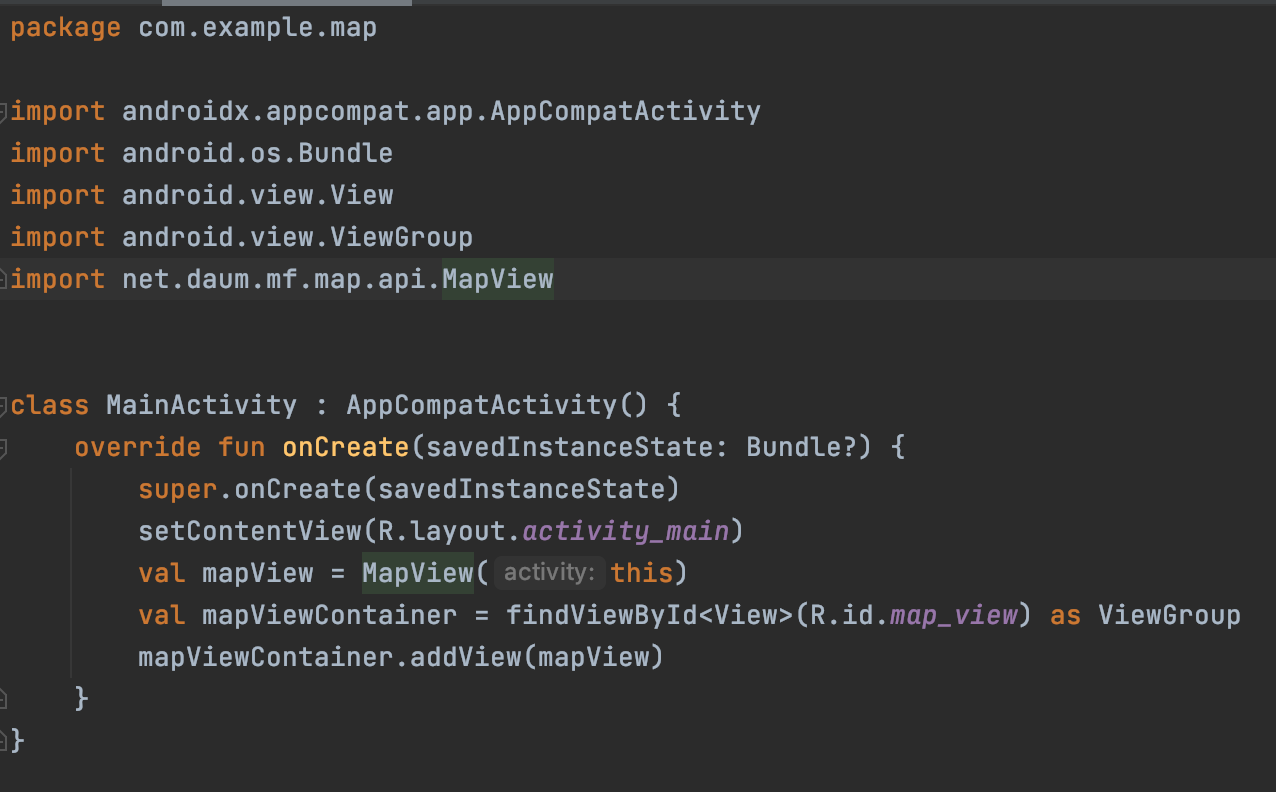
퍼미션 추가.

3. xml 파일 만들기



위와 같이 작성. Relative 레이아웃에 맵을 띄울 것임.

4. 코틀린 코드 작성



위와 같이 작성. xml파일에서 작성한 레이아웃에 맵을 연결시키는 코드임.

5. 빌드 결과 : 빌드시 앱이 정상적으로 켜지지 않고, 켜지는가 싶다가 바로 꺼짐. 원인 파악 불가로 구글지도를 사용하는 것으로 결정

**구글지도 구현**

1. API 사용을 위한 키 값 받아오기.

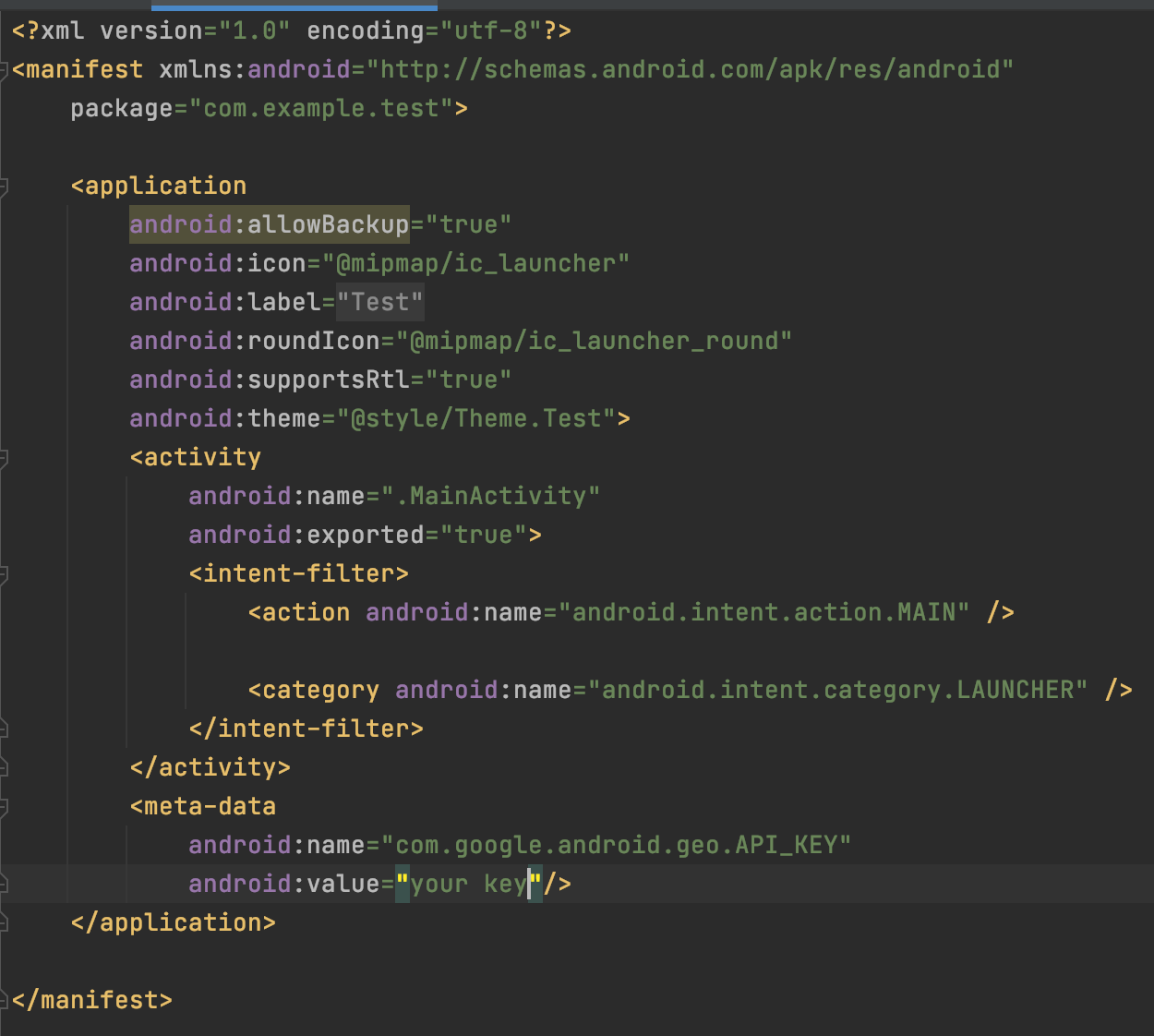
1-1. 구글 클라우드 플랫폼에서 로그인하고 프로젝트 만들기.

1-2. 검색창에서 Maps SDK for Android 검색 후 사용 클릭.

1-3. 좌상단 메뉴 클릭 후 API 및 서비스 > 사용자 인증정보에 들어가서 사용자 인증정보 만들기 클릭 후 API키 메뉴 눌러서 키 생성

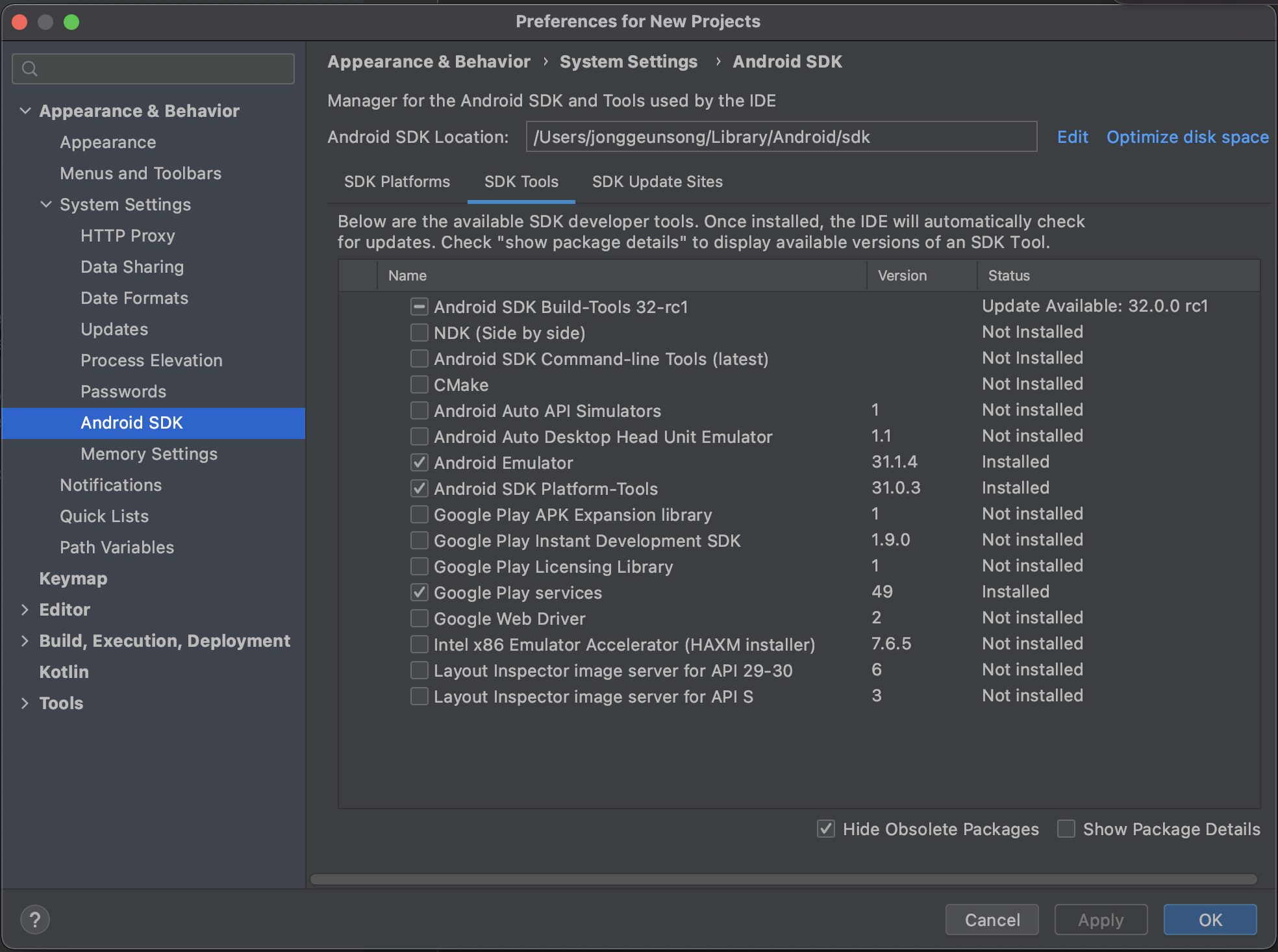
1-4. 키 제한에서 애플리케이션 제한사항 항목에서 Android 앱 클릭 후 API 키를 사용하고자하는 프로젝트의 SHA-1키 입력해넣기. 그리고 저장.

2. 메니페스트 파일 작성

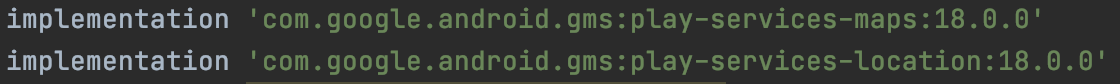


위와 같이 코드 작성. 메타데이타 태그에서 value에는 아까 발급받은 api 키 값 넣기.

3. 라이브러리 추가



안드로이드 스튜디오 Tools메뉴에서 SDK Manager에 들어가서 SDK tools를 누르고 Google Play servies 를 찾아 다운받기.

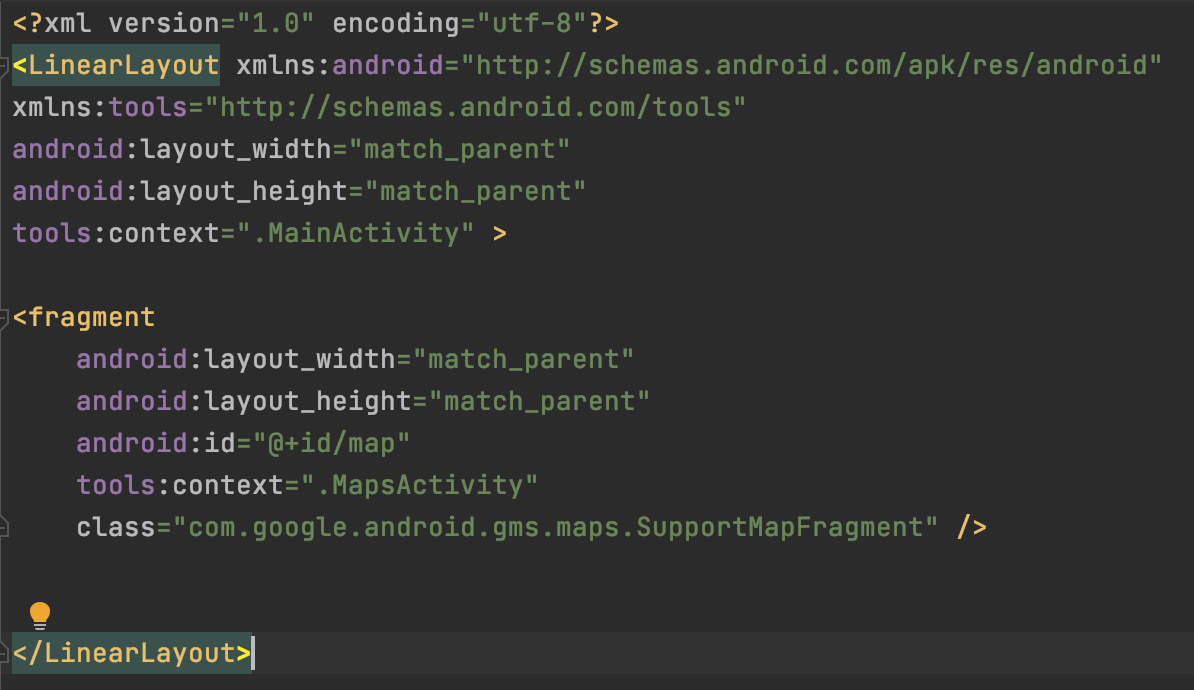


모듈수준 Gradle에서 위와 같이 의존성 추가.



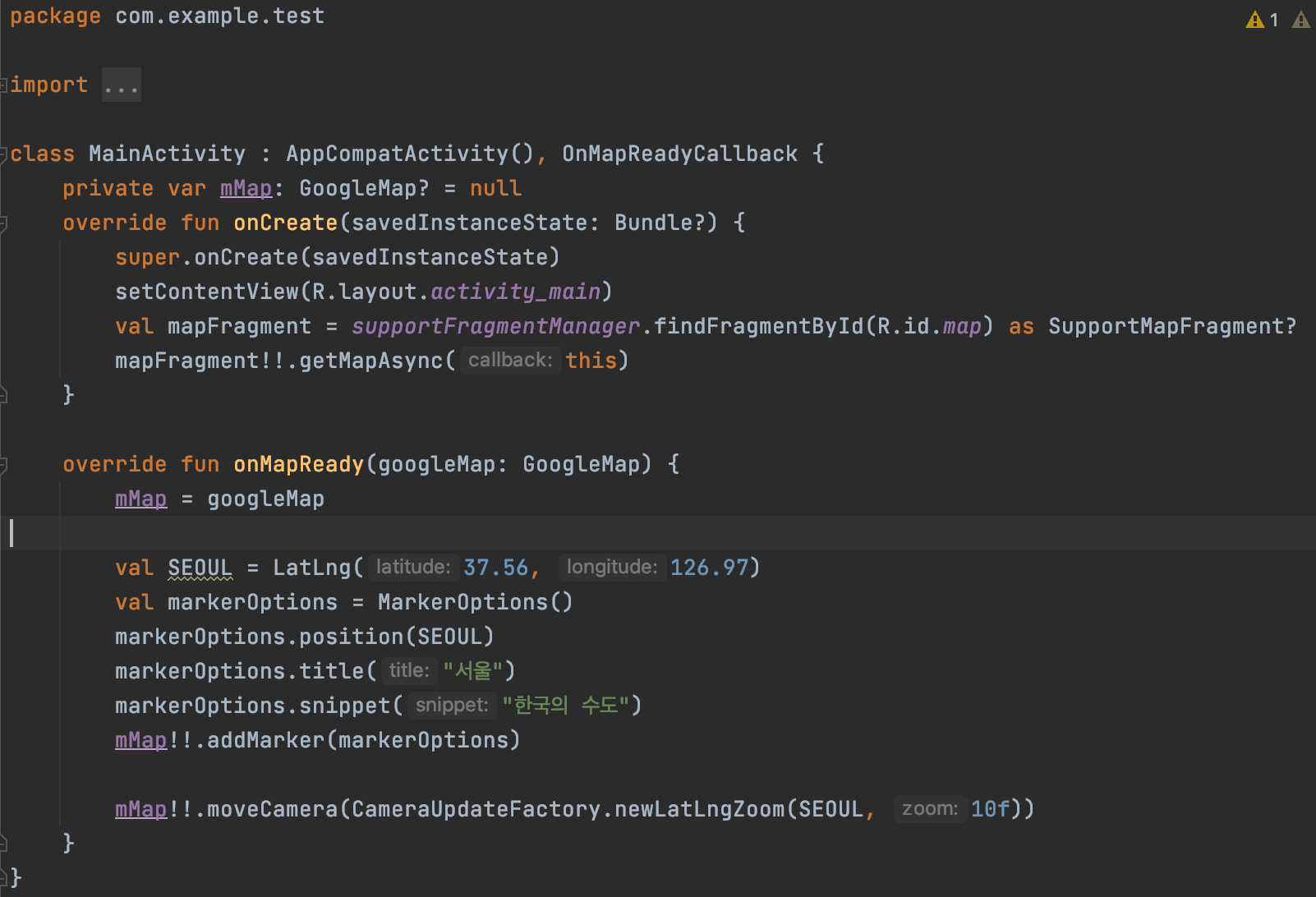
프로젝트수준 Gradle에서 위와 같이 의존성 추가.

4. xml 파일 만들기



위와 같이 코드 작성. Fragment에 지도 연결할 것임.

5. 코틀린 코드 작성



위와 같이 작성.

private var mMap: GoogleMap? = null : 구글 맵 인스턴스 생성

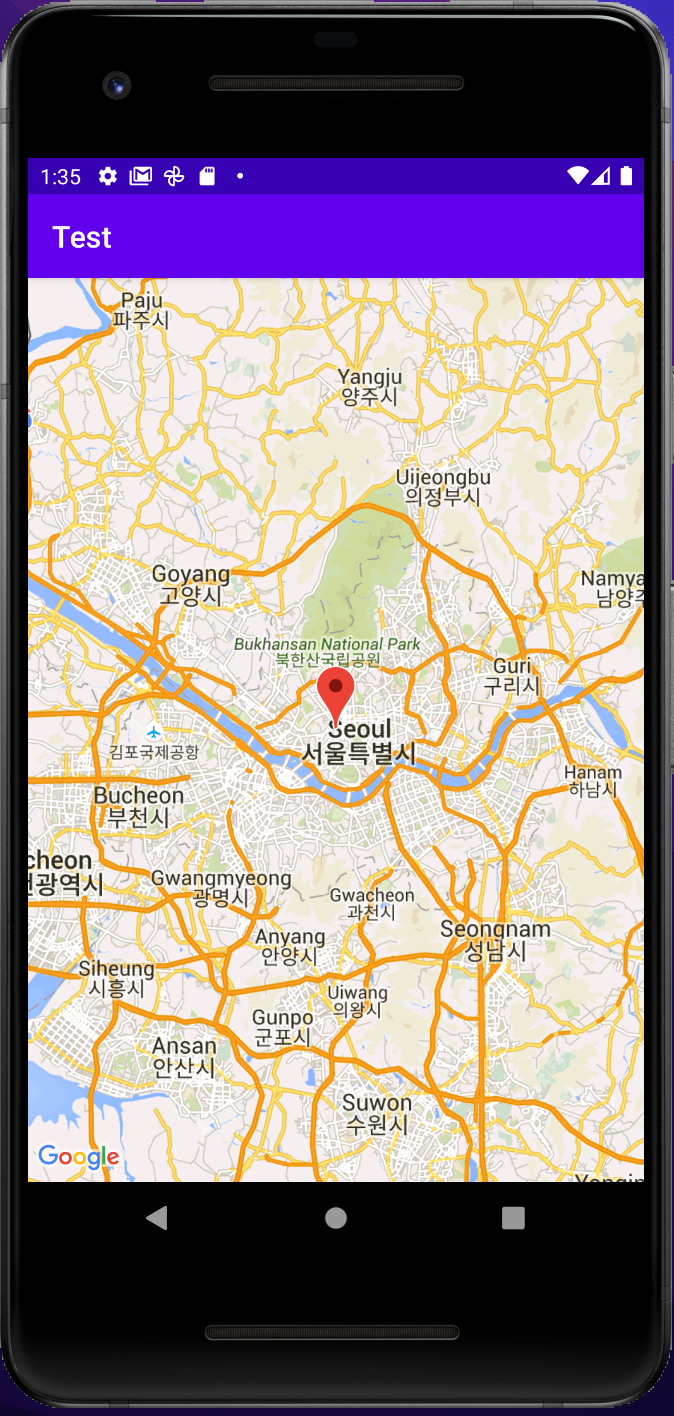
val mapFragment = *supportFragmentManager*.findFragmentById(R.id.*map*) as SupportMapFragment?

: xml에서의 fragment를 mapfragment로 해서 인스턴스 생성

mapFragment!!.getMapAsync(this) : mapfragment에 구글지도 연결

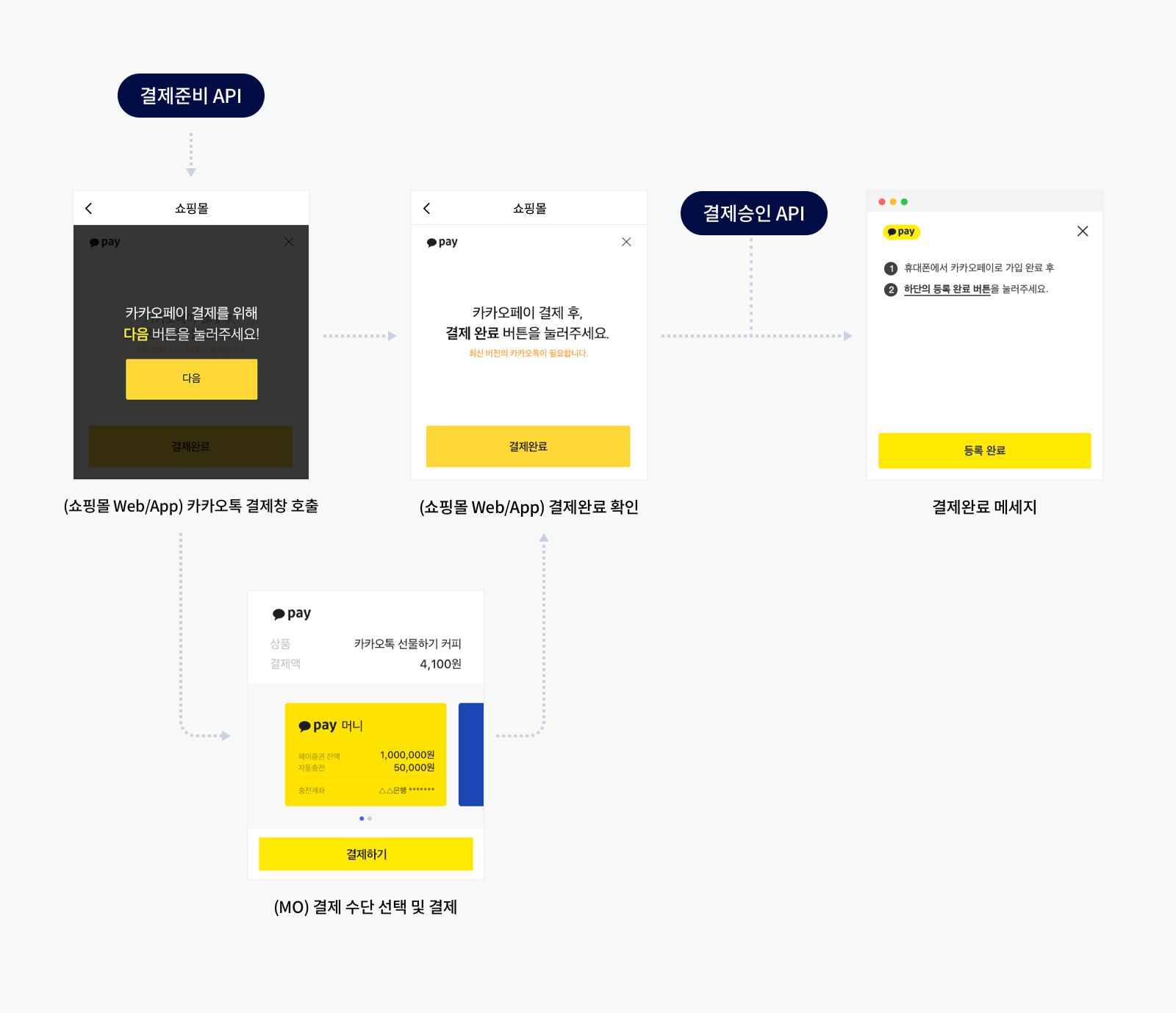
override fun onMapReady(googleMap: GoogleMap) : getMapAsync()에서 사용될 메소드. 구글지도를 생성하고 마커도 생성.

6. 빌드 결과



**카카오페이 구현**

1. 카카오페이 결제 프로세스(카카오 개발자 페이지 공식 가이드 문서 참조)



Step 1.결제 준비

서버에서 결제 준비 API를 호출합니다. 응답이 오면 요청한 결제와 TID를 매핑(Mapping)하여 저장하고, 추후 결제 승인 API 호출 및 대사 작업에 사용합니다. 보안을 위해 TID가 사용자에게 노출되지 않도록 주의합니다.

* 웹 브라우저(Web browser) : 응답 바디로 받은 결제 대기 화면 주소인 next\_redirect\_web\_url로 리다이렉트합니다.
* 네이티브 앱: 응답 바디로 받은 next\_redirect\_app\_url 값으로 결제 대기 화면 웹뷰를 띄웁니다.

Step 2.결제 수단 선택

사용자는 결제 준비 화면에서 결제 수단 선택 화면으로 이동하고, 결제 수단을 선택한 뒤 결제하기 버튼을 누릅니다.

Step 3.결제 대기

사용자는 결제 대기 화면으로 이동하고, 화면은 결제 요청 결과에 따라 다른 URL로 리다이렉트됩니다. 사용자가 결제를 취소한 경우, 보안을 위해 주문 상세 조회 API를 호출하고 결제 과정을 중단해야 합니다. 조회 시 상태 값이 사용자가 결제를 중단한 상태임을 나타내는 QUIT\_PAYMENT인 것을 확인하고 결제 중단 처리를 해야 합니다.

* 요청 결과별 리다이렉트 URL
  + 요청 성공: approval\_url로 리다이렉트
  + 요청 취소: cancel\_url로 리다이렉트
  + 요청 유효 시간(15분) 경과: fail\_url로 리다이렉트

approval\_url, cancel\_url, fail\_url은 카카오페이 API 요청 응답을 받아 처리할 주소입니다. 이 값들의 도메인(Domain)은 앱 정보에 등록된 웹 플랫폼 도메인과 일치해야 합니다.

Step 4.결제 승인

결제 대기 단계에서 결제 요청에 성공했다면 결제 승인 API를 호출합니다. 이때 결제 승인 요청을 인증하는 pg\_token을 전달해야 합니다. pg\_token은 사용자가 결제 수단을 선택하고 결제 버튼을 눌러 approval\_url로 리다이렉트될 때, 리다이렉트 요청의 approval\_url에 포함된 query string으로 전달 받습니다.

응답을 받으면 결제 결과를 저장하고 사용자에게 결제 완료 화면을 보여줍니다. 결제 승인 API의 동작에 따라 사용자도 결제 완료 메시지를 받습니다.

2. 구현부분

시간관계상 카카오페이 구현 보류