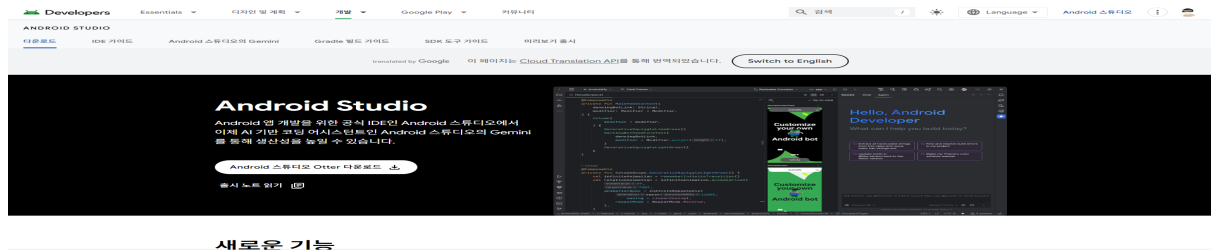


# 사용자 맞춤 앱 제작 매뉴얼

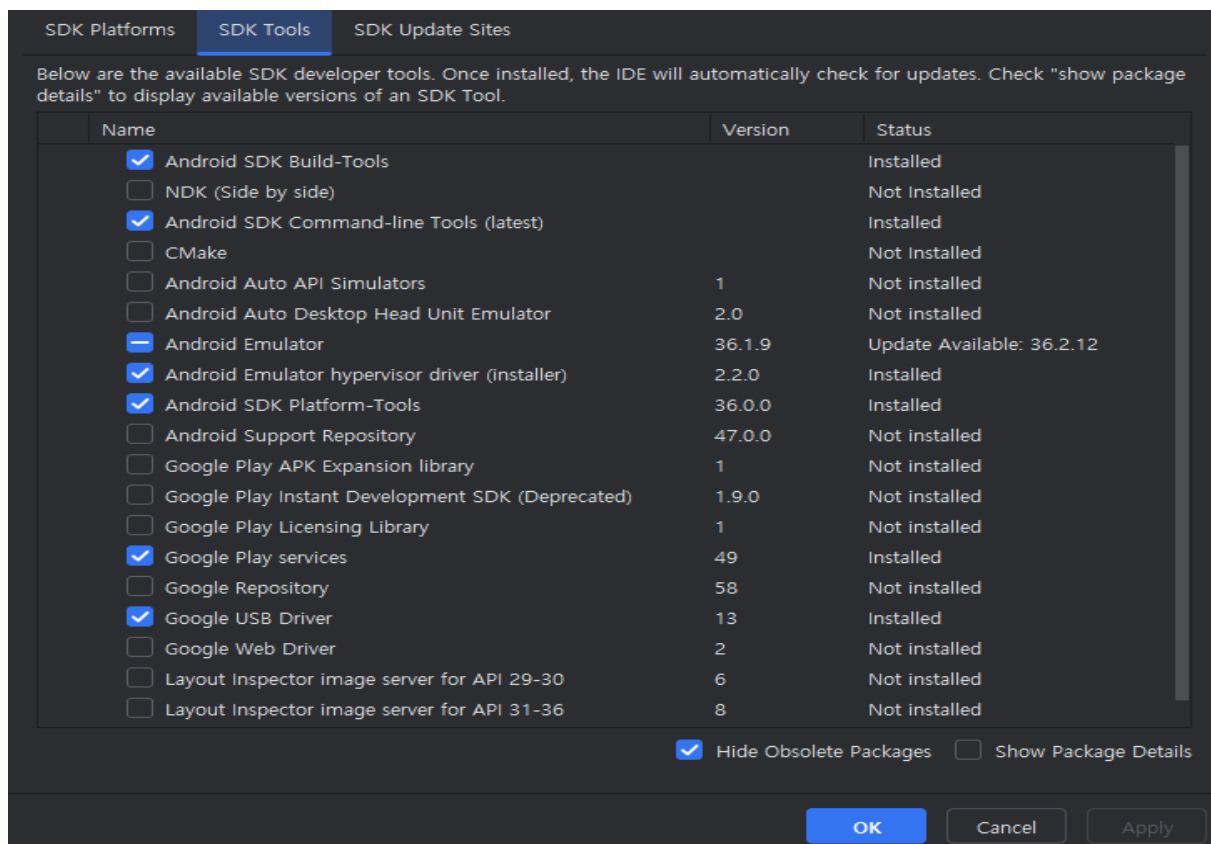
## 1. Android Studio 설치하기.

### 1-1. <https://developer.android.com/studio?hl=ko>



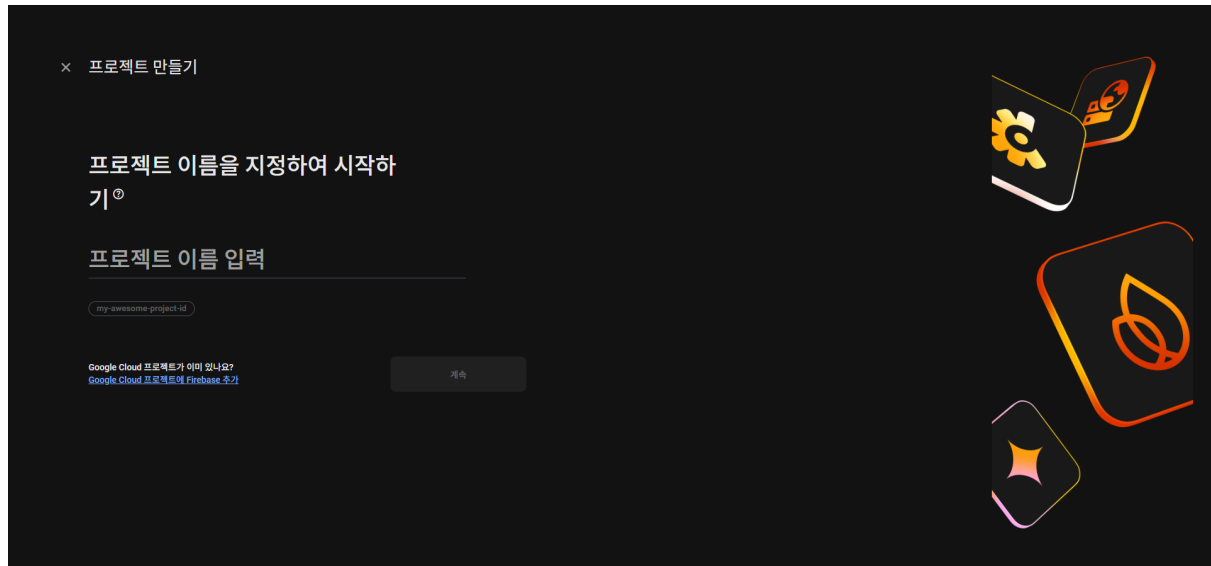
1-2. Android Studio 설정으로 들어가서 [Languages & Frameworks]에 [Android SDK]를 클릭을 한 후 Android 16.0 ("Baklava"), Android 15.0 ("VanillaIceCream"), Android 14.0 ("UpsideDownCake")를 클릭 후 Apply(적용)을 시키면 된다.

1-3. 그리고 SDK Tools는 아래 사진과 같이 선택 후 Apply 하면 된다.

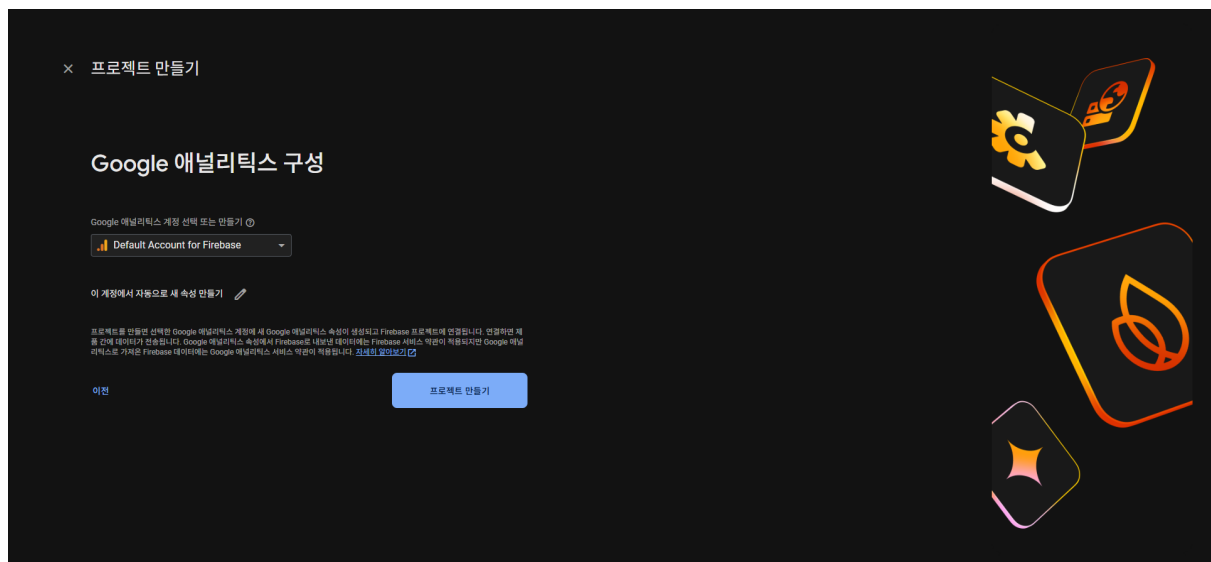


## 2. Firebase 세팅하기.

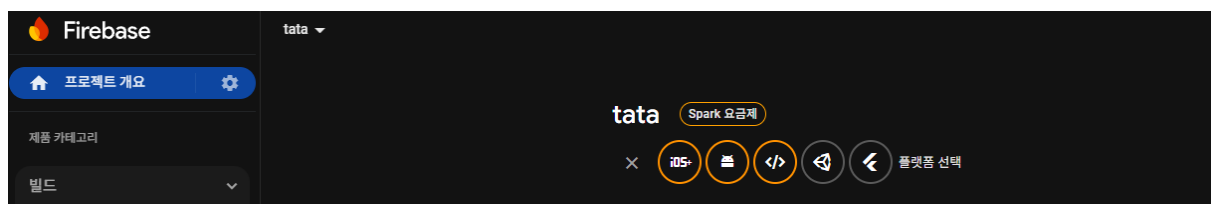
### 2-1. 프로젝트명 정하기



### 2-2. Google 애널리틱스 구성하기. (Default Account for Firebase)



### 2-3. [+앱 추가]를 선택 후 안드로이드 클릭.



## 2-4. Android 패키지 이름

(일반적으로 패키지 이름은 앱 수준 **build.gradle** 파일의 **applicationId** 이다.)

1 앱 등록

Android 패키지 이름 ?  
com.company.appname

앱 닉네임(선택사항) ?  
내 Android 앱

## 2-5. 구성 파일 다운로드 후 추가하기.

2 구성 파일 다운로드 후 추가

Android 스튜디오 안내(아래 참조) | [Unity](#) [C++](#)

📄 google-services.json 다운로드

Android 스튜디오에서 프로젝트 뷰로 전환하여 프로젝트 루트 디렉터리를 확인하세요.

다운로드한 google-services.json 파일을 모듈(앱 수준) 루트 디렉터리로 이동하세요.

  
google-services.json

다음

Project

MyApplication [My Application]

- .gradle
- .idea
- app
  - libs
  - src
  - .gitignore
  - build.gradle.kts
  - google-services.json
  - proguard-rules.pro
- gradle

## 2-6. Firebase SDK 추가하기.

**3** **Firebase SDK 추가** Gradle 안내 Unity C++

★ 여전히 `buildscript` 문법을 사용하여 플러그인을 관리하고 있나요? 해당 문법을 사용하여 [Firebase 플러그인을 추가](#) 하는 방법을 알아보세요.

1. Firebase SDK가 `google-services.json` 구성 값에 액세스할 수 있도록 하려면 Google 서비스 Gradle 플러그인이 필요합니다.

☒ Kotlin DSL(`build.gradle.kts`) ☐ Groovy(`build.gradle`)

프로젝트 수준의 `build.gradle.kts` 파일에 플러그인을 종속 항목으로 추가합니다.

루트 수준(프로젝트 수준) Gradle 파일(`<project>/build.gradle.kts`):

```
plugins {  
    // ...  
  
    // Add the dependency for the Google services Gradle plugin  
    id("com.google.gms.google-services") version "4.4.4" apply false  
}
```

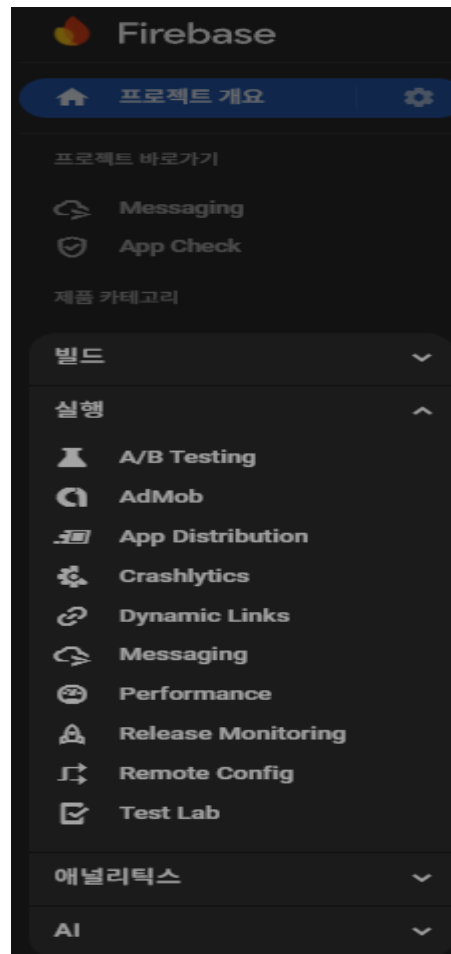
2. 그런 다음 모듈(앱 수준) `build.gradle.kts` 파일에서 `google-services` 플러그인과 앱에서 사용할 Firebase SDK를 모두 추가합니다.

모듈(앱 수준) Gradle 파일(`<project>/<app-module>/build.gradle.kts`):

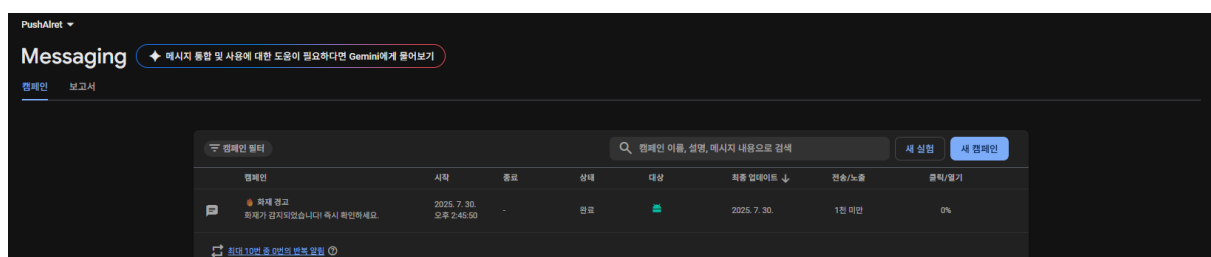
```
plugins {  
    id("com.android.application")  
    // Add the Google services Gradle plugin  
    id("com.google.gms.google-services")  
    ...  
}  
  
dependencies {  
    // Import the Firebase BoM  
    implementation(platform("com.google.firebase:firebase-bom:34.5.0"))  
  
    // TODO: Add the dependencies for Firebase products you want to use  
    // When using the BoM, don't specify versions in Firebase dependencies  
    implementation("com.google.firebase:firebase-analytics")  
  
    // Add the dependencies for any other desired Firebase products  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```

Firebase Android BoM을 사용하면 앱에서 항상 호환되는 Firebase 라이브러리 버전을 사용합니다. [자세히 알아보기](#)

2-7. 그 이후 실행에서 Messaging으로 들어가기.



2-8. [새 캠페인] 으로 들어가기



## 2-9. [Firebase 알림 메시지] 선택

### Firebase 메시지 온보딩

메시지 유형 및 플랫폼

☐

**Firebase 알림 메시지**  
알림 메시지를 보내 사용자에게 프로모션 또는 새로운 기능을 즉시 알려세요.  
사용자가 앱 외부에 있어도 알림 메시지가 수신됩니다.

☐

**Firebase 인앱 메시지**  
인앱 메시지를 보내 앱의 활성 사용자가 구독하거나, 동영상을 보거나, 레벨을 달성하거나, 상품을 구매하도록 유도할 수 있습니다. **사용자는 앱 내부에서만 인앱 메시지를 받을 수 있습니다.**

취소

만들기

## 2-10. 알림 제목 및 텍스트 입력

1 알림

알림 제목 ②

제목 입력(선택 사항)

알림 텍스트

알림 텍스트 입력

알림 이미지(선택사항) ②

예: <https://yourapp.com/image.png>

알림 이름(선택사항) ②

이름 입력(선택사항)

기기 미리보기

이 미리보기는 휴대기기에서 메시지가 어떻게 표시될지 대략적으로 보여줍니다. 실제 메시지 렌더링은 기기에 따라 달라집니다. 정확한 결과를 보려면 실제 기기 테스트하세요.

테스트 메시지 전송

초기 상태

펼쳐보기

Android

Apple

## 2-11. 타겟 정하기.

The screenshot shows a dark-themed interface for setting up a campaign. On the left, a vertical progress bar has five steps: 1. 알림 (Notification), 2. 타겟 (Target), 3. 예약 (Reservation), 4. 주요 이벤트(선택사항) (Main Event (Optional)), and 5. 추가 옵션(선택사항) (Additional Options (Optional)). Step 2, '타겟', is currently active. The main content area for step 2 includes a '사용자 세그먼트' (User Segment) dropdown menu, a '주제' (Topic) dropdown menu, and a '사용자 타겟팅 조건...' (User Targeting Conditions...) section. This section contains two input fields: '앱' (App) and '앱 선택' (App Selection), followed by a '및' (And) button. Below these fields are two buttons: '다른 앱 타겟팅' (Other App Targeting) and '다음' (Next). At the bottom right of the interface, there are two buttons: '초안으로 저장' (Save as Draft) and '검토' (Review).

## 2-12. 맞춤 데이터로 키와 값을 작성. (안해도 된다.)

The screenshot shows the '추가 옵션(선택사항)' (Additional Options (Optional)) step in the campaign setup interface. The progress bar on the left shows step 5 is active. The main content area includes a message '모든 입력란이 선택사항입니다.' (All input fields are optional.). Below this is a section for 'Android 알림 채널 ②' (Android Notification Channel ②) with a text input field. The next section is '맞춤 데이터 ②' (Custom Data ②), which contains two input fields labeled '키' (Key) and '값' (Value). Below these is a section for '알림음' (Notification Sound) with a dropdown menu currently set to '사용 중지됨' (Muted). The final section is '만료 ②' (Expiration ②), which contains two dropdown menus: one for the number '4' and another for the unit '주' (Week).

### 3. Android Studio CODE 입력

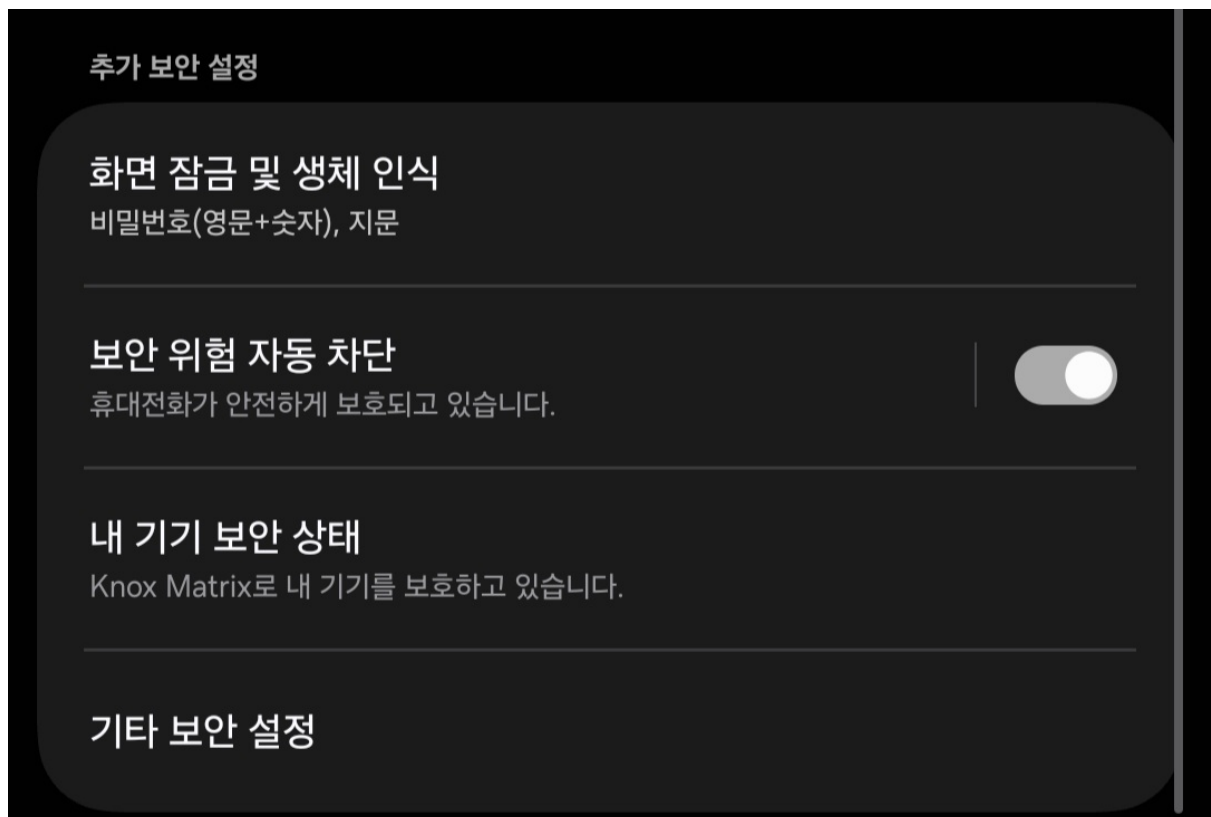
- 아래에 있는 Git 링크에 들어가서 코드를 입력하기.

[https://github.com/elecage/DU\\_AT/tree/e9d6739e88eb2484f7eec1432c82eef1935bbbee/AuditoryAssist/App](https://github.com/elecage/DU_AT/tree/e9d6739e88eb2484f7eec1432c82eef1935bbbee/AuditoryAssist/App)

- 해당 src.zip 파일은 그대로 복붙하면 된다.
- 해당 gradle.zip 파일은 gradle 폴더에 그대로 복사

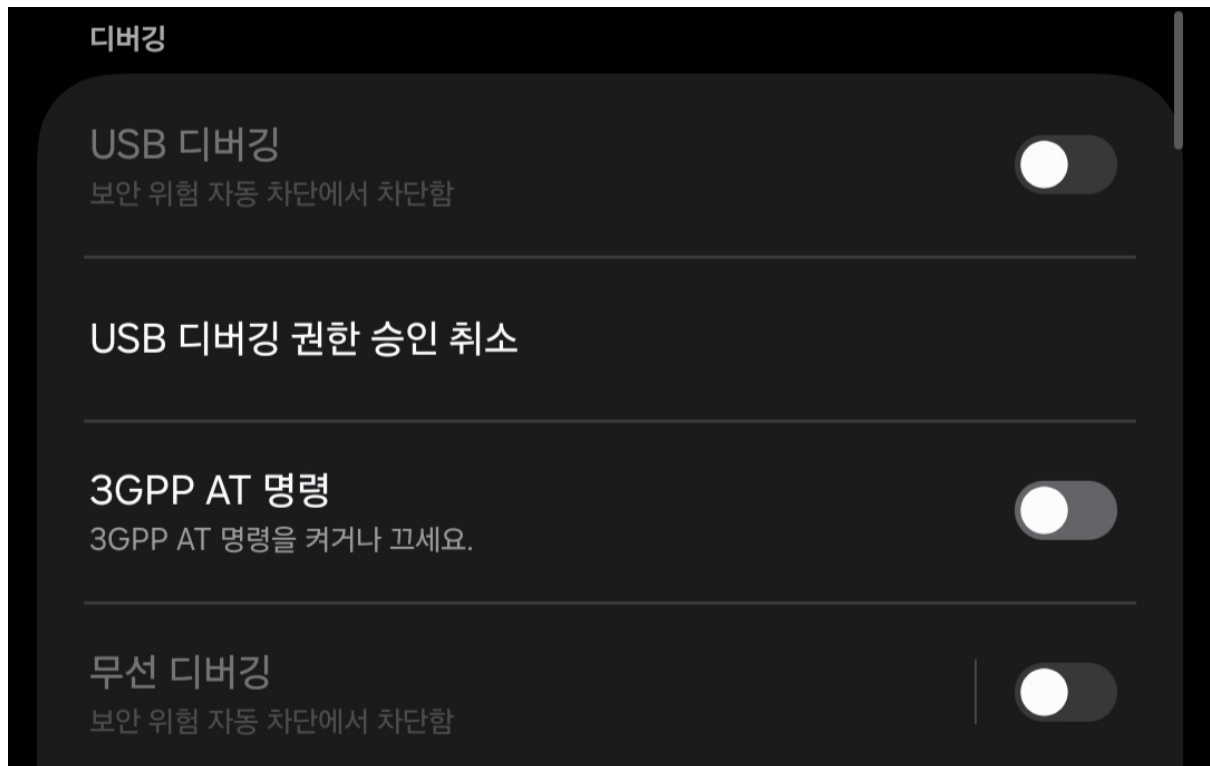
### 4. 핸드폰 설정.

4-1. 설정에서 [보안]으로 이동한 후 [보안 위험 자동 차단]을 켜킨다.





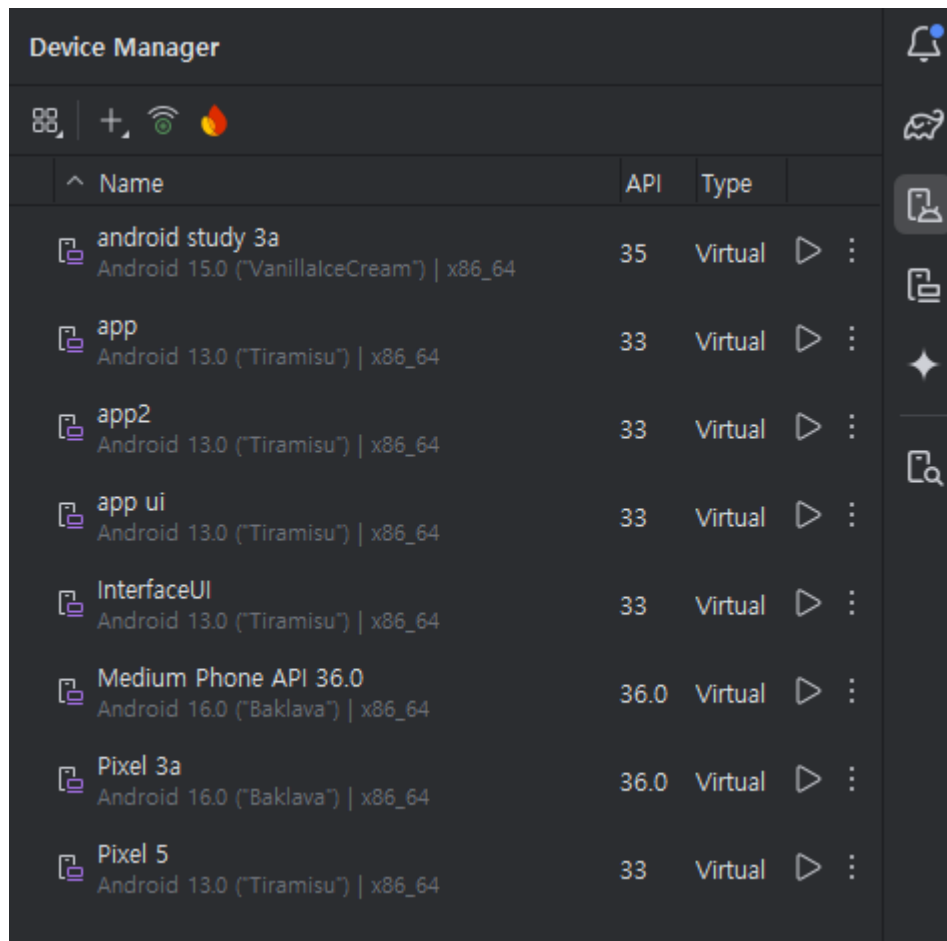
4-2. [보안 위험 자동 차단]을 켜 후 [개발자 옵션]으로 가서 [USB 디버깅]을 켜면 된다.



4-3. 그 이후 PC와 핸드폰을 USB로 연결하고, 권한 승인을 하고나서 실행을 시키면 핸드폰에 자동으로 앱이 깔린다.

## 5. 테스트 에뮬레이터 설정하는 방법.

5-1. 오른쪽에서 Device Manager로 들어간다.



5-2. [+] 버튼을 누르고 [Create Virtual Device]를 선택한다.

5-3. 그리고 [pixel 5 API 30+]를 선택한다.

Pixel 6	31+	1080	2400	420 dpi
Pixel 5	30+	1080	2340	440 dpi
Pixel 4a	30+	1080	2340	440 dpi

5-4. [API 33 "Tiramisu", Android 13.0]을 설정 후 Google ApIs Intel x86\_64 Atom System Image를 선택하고 Finish를 선택하면 끝.

