목차

[1.node.js , npm 설치 1](#_Toc125722552)

[1.1.개요 1](#_Toc125722553)

[1.2.설치 1](#_Toc125722554)

[**1.2.1.다운로드** 1](#_Toc125722555)

[1.2.2. 설치 1](#_Toc125722556)

[1.2.3. 설치확인 1](#_Toc125722557)

[2.안드로이드 스튜디오 설치 1](#_Toc125722558)

[2.1.개요 1](#_Toc125722559)

[2.**2.설치** 1](#_Toc125722560)

[**2.2.1.다운로드** 1](#_Toc125722561)

[**2.2.2.설치** 1](#_Toc125722562)

[**2.2.3. Android SDK 설정 (안드로이드 스튜디오 에물레이터 설정)** 1](#_Toc125722563)

[**2.2.4.환경변수 설정(안드로이드)** 1](#_Toc125722564)

[**3.Open JAVA SDK 1.8 설치** 1](#_Toc125722565)

[**3.1.개요** 1](#_Toc125722566)

[**3.2.다운로드** 1](#_Toc125722567)

[3.2.설치 1](#_Toc125722568)

[**3.2.1.환경변수 설정(JAVA)** 1](#_Toc125722569)

[**4.Gradle 설치** 1](#_Toc125722570)

[**4.1.개요** 1](#_Toc125722571)

[**4.2.설치** 1](#_Toc125722572)

[**4.2.1.다운로드** 1](#_Toc125722573)

[**4.2.2. Gradle 설치** 1](#_Toc125722574)

[4.2.3. 환경 변수 설정 1](#_Toc125722575)

[**5.Quasar FrameWork** 1](#_Toc125722576)

[**5.1. Quasar FrameWork API Document** 1](#_Toc125722577)

[**5.2. Quasar FrameWork 프로젝트 생성** 1](#_Toc125722578)

[**5.2.1. 프로젝트 workspace 생성** 1](#_Toc125722579)

[**6. Cordova Framework** 1](#_Toc125722580)

[**6.1. 개요** 1](#_Toc125722581)

[**6.2.설치** 1](#_Toc125722582)

# 1.node.js , npm 설치

## 1.1.개요

* [**npm(node package manager)**](https://www.npmjs.com/)은 자바스크립트 패키지 매니저이다. **Node.js**에서 사용할 수 있는 모듈들을 패키지화 하여 모아둔 저장소 역할과 패키지 설치 및 관리를 위한 **CLI(Command line interface)**를 제공한다. 자신이 작성한 패키지를 공개할 수도 있고 필요한 패키지를 검색하여 재사용할 수도 있다

.

* Nodoe.js설치후 npm 패키지확인

## 1.2.설치

### **1.2.1.다운로드**

* 링크URL : <https://nodejs.org/ko/>

**텍스트이(가) 표시된 사진

자동 생성된 설명**

### 1.2.2. 설치

**텍스트이(가) 표시된 사진

자동 생성된 설명**

### 1.2.3. 설치확인

* CMD창을 열어 설치확인
* **nodoe -v** (노드설치 버전확인)
* **npm -v** (npm 설치버전확인)

****

# 2.안드로이드 스튜디오 설치

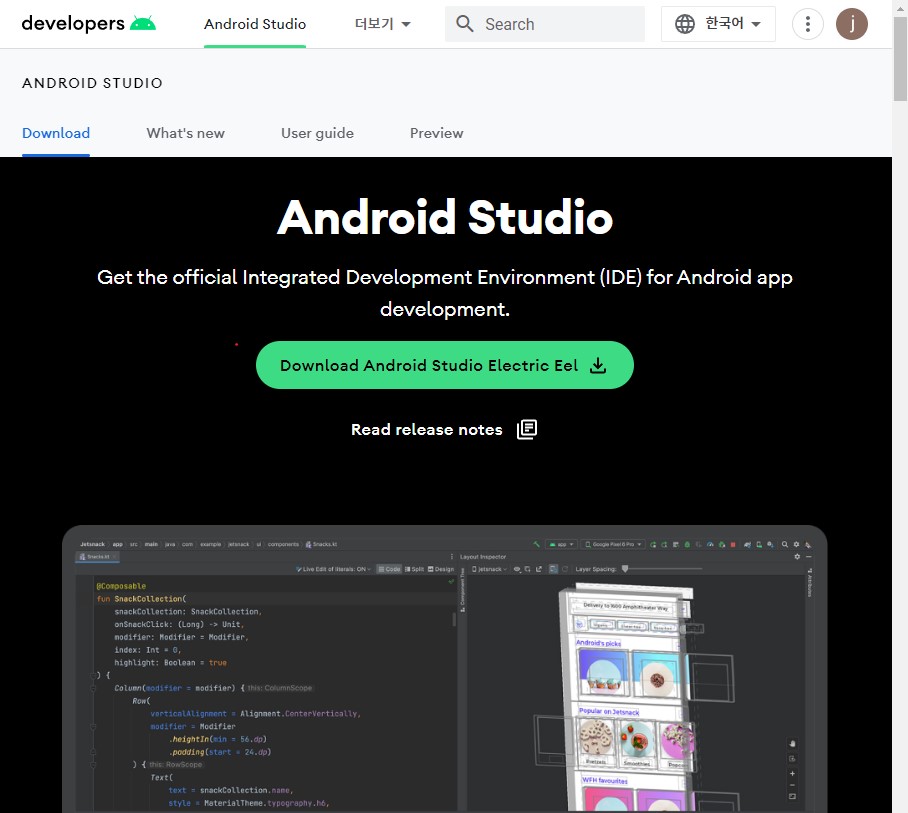
## 2.1.개요

* 이용약관 동의를 체크하고 진행하면 exe파일이 다운받아진다. 이것도 실행해서 설치해주자.
* **Android SDK** 따로 설치 없이 개발 내용을 에물레이터로 확인하거나 **apk**를 생성하여 실기(휴대폰,타블릿)에서 확인이 가능하다.

## 2.**2.설치**

### **2.2.1.다운로드**

* 링크URL**:** [Download Android Studio & App Tools - Android Developers](https://developer.android.com/studio?hl=ko)



### **2.2.2.설치**

**1. 2.**  
텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

3. 4.  
**텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명  
5. 6.**

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

### **2.2.3. Android SDK 설정 (안드로이드 스튜디오 에물레이터 설정)**

* 개발결과물을 보여줄 에물레이터를 생성한다
* 안드로이 스튜디오 실행

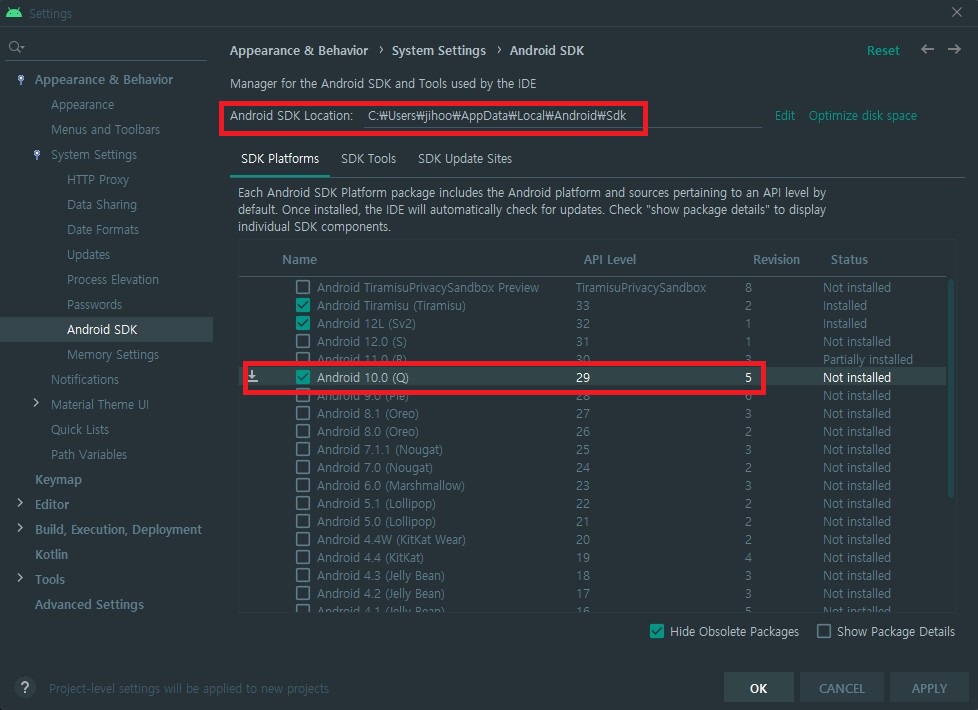
1.우측 메뉴를 클릭 **SDK Manager** 메뉴를 선택

텍스트이(가) 표시된 사진

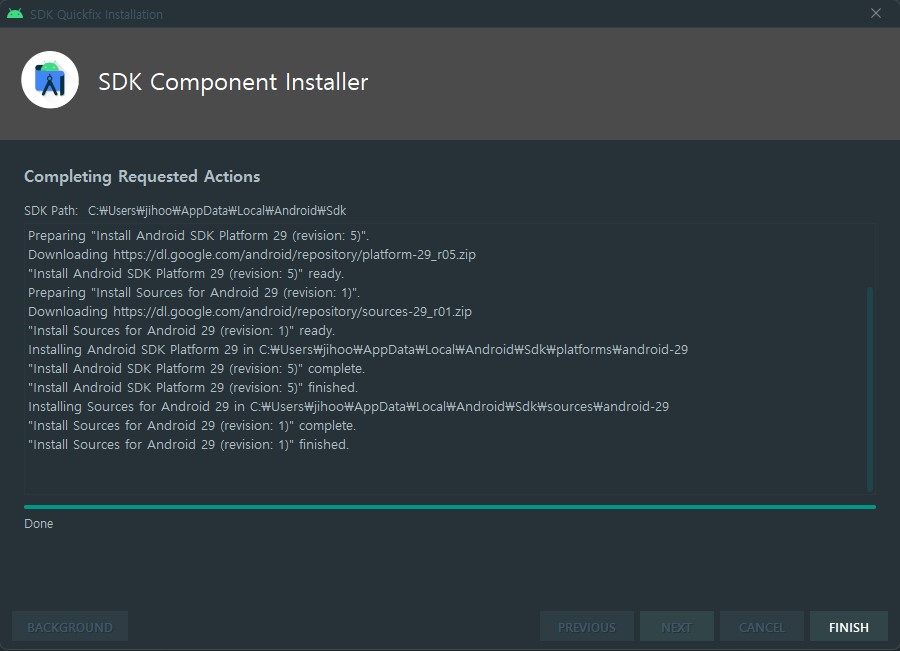
자동 생성된 설명

**2.**

* **Android SDK Location** 확인 위치변경 가능하나 디폴트로 놓고 사용
* **SDK Platform** 탭에서 하단 **API Level 30**을 선택 후 **OK 버튼클릭**

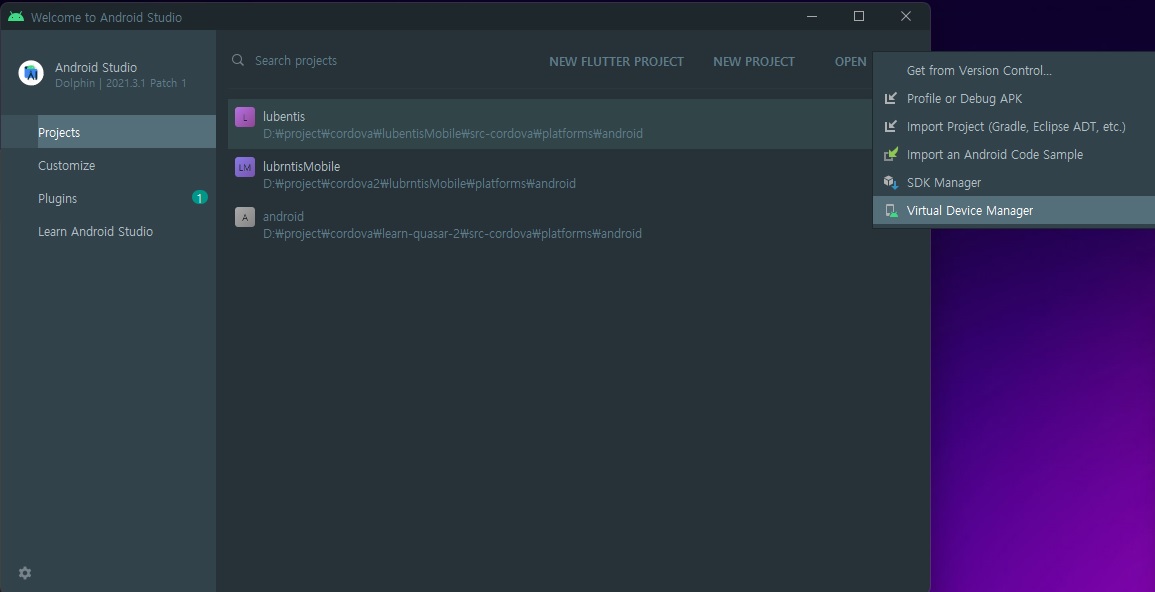
****

**3. Android SDK 다운로드 완료 후 FINISH 버튼클릭**

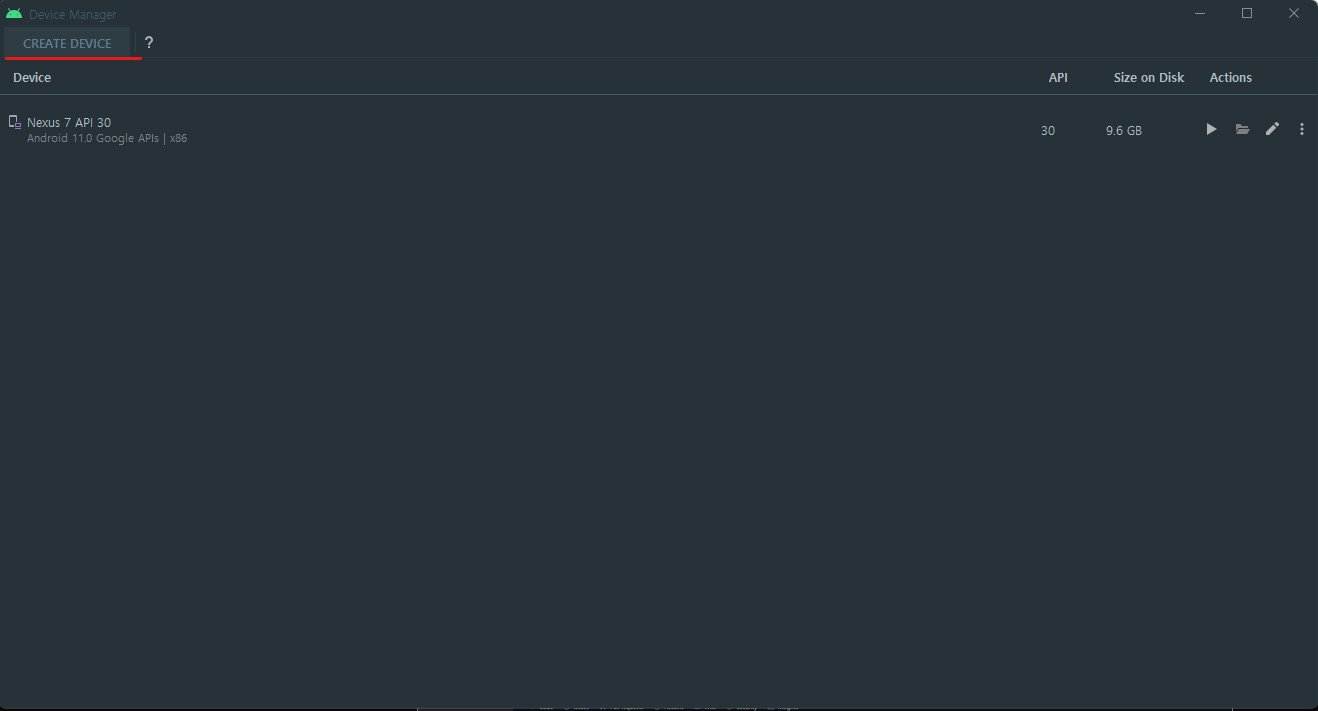
****

**4. 에물레이터 생성**

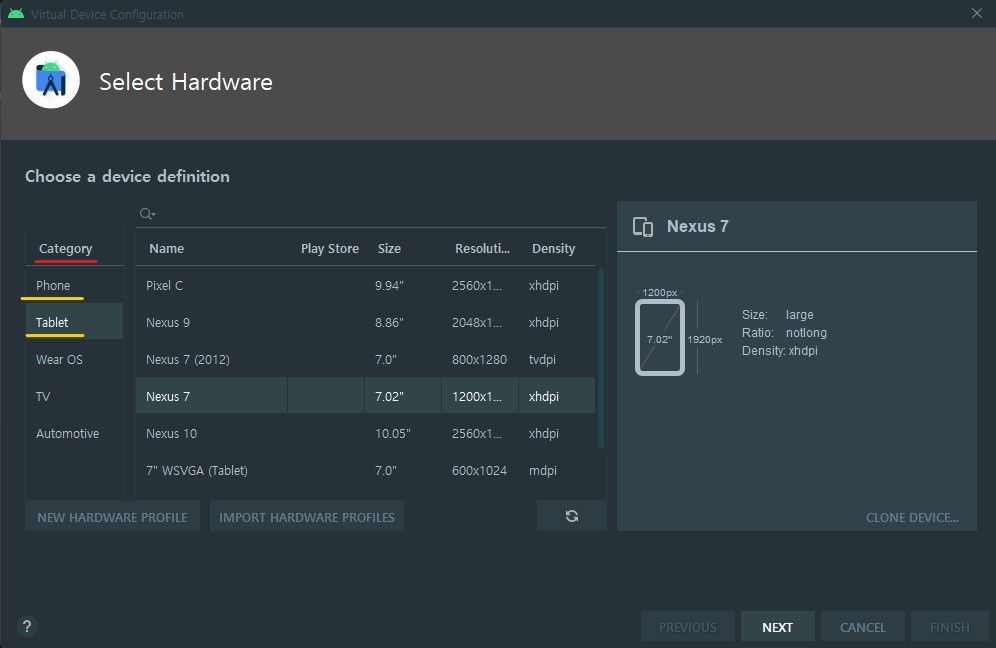
* **Android SDK 가 다운완료 되면 그에 따른 에물레이터를 생성한다.**
* **Virtual Device Manager 메뉴를 클릭한다.**

****

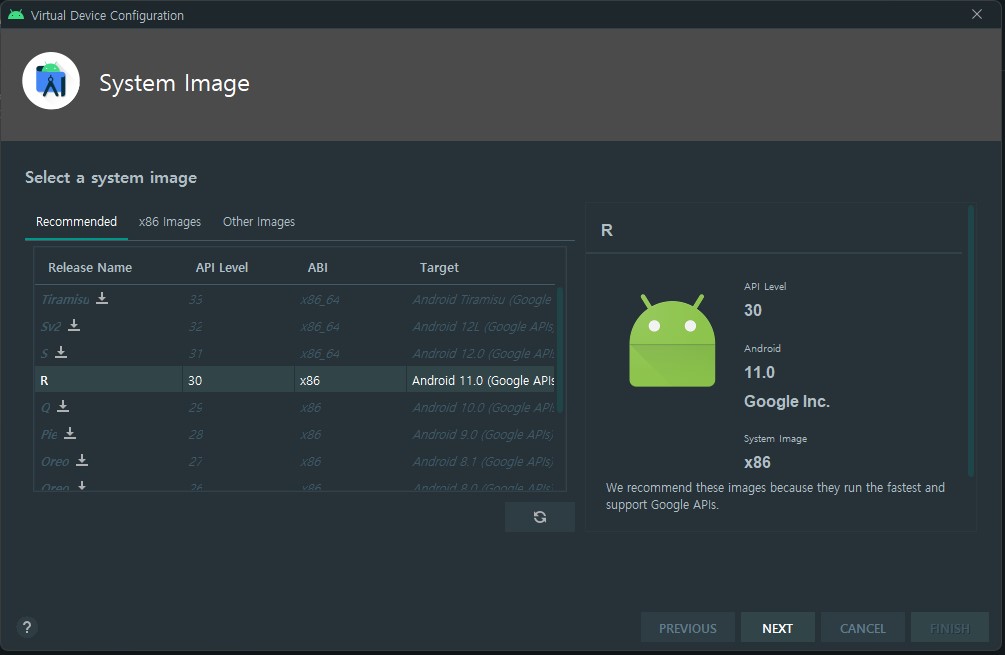
* CREATE DEVICE 를 클릭한다.



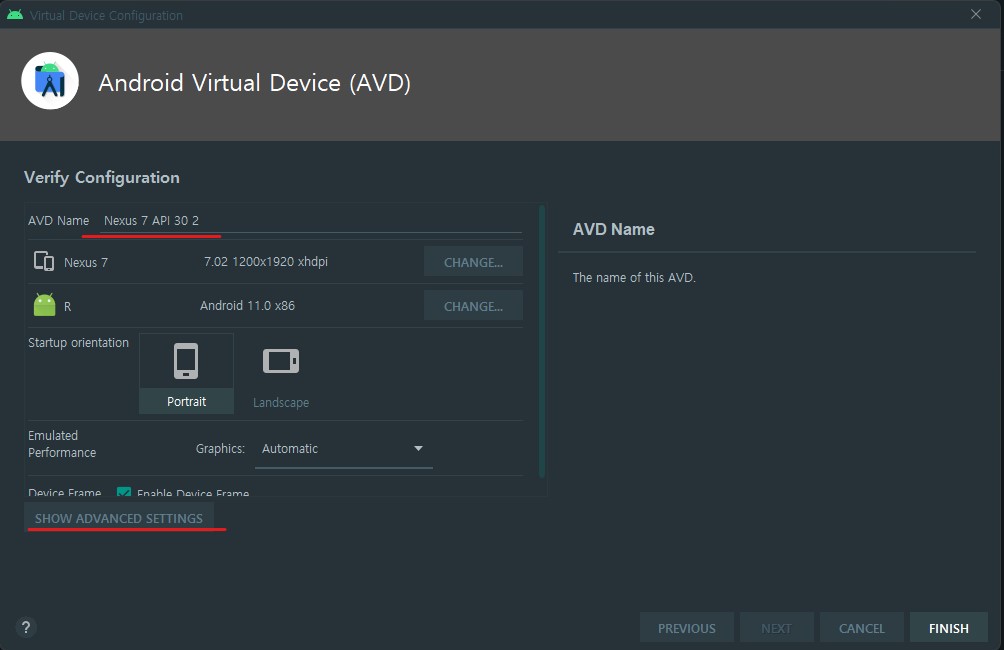
* Categgory 에서 테스트에 이용할 device를 선택 후 NEXT 버튼을 클릭
* 예제) Nexus 7를 선택



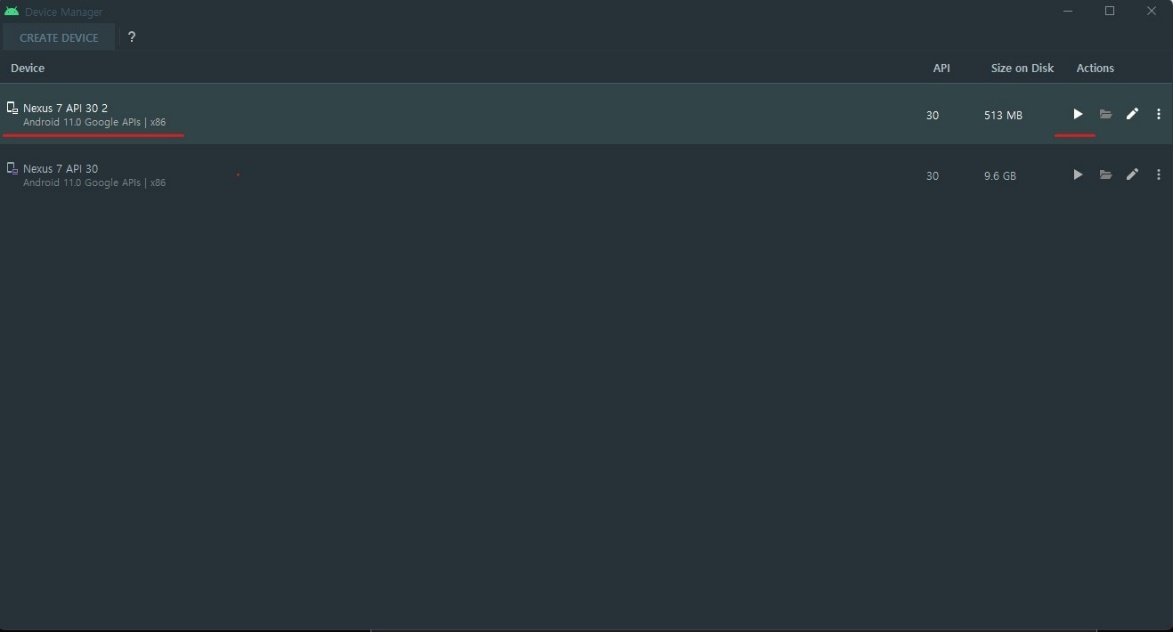
* **NEXT** 버튼 클릭

****

* **AVD Name** 기기 이름(변경가능)
* **SHOW ADVANCED SETTING** 를 클릭하여 에물에 화상 카메라를 실기 카메라와 연결 가능하다. 설치 후 수정이 가능 하니 여기서는 생략하도록 한다.
* **FINISH** 버튼을 클릭한다.

****

* Nexus 7 API 30 2 에물레이터가 생성 좌측에 플레이 버튼을 클릭

****

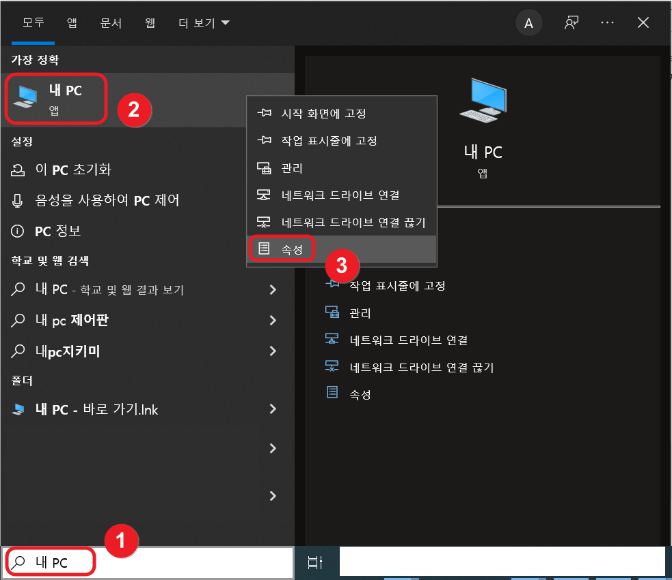
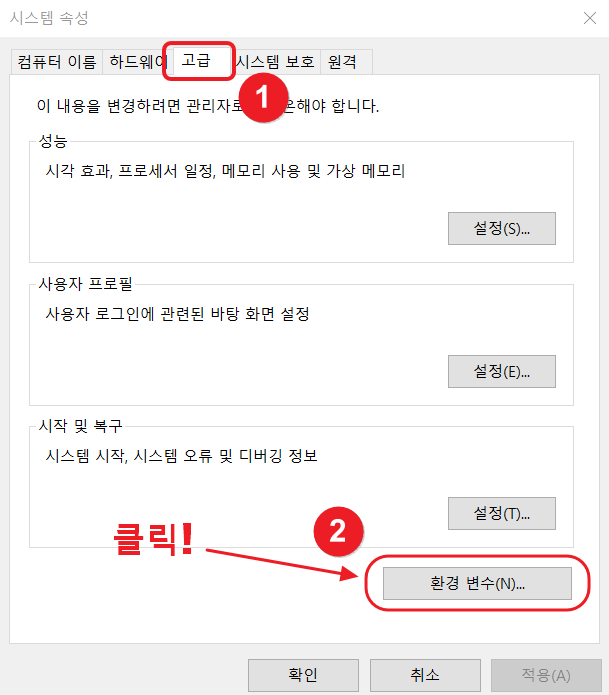
* 에물레이터가 실행된다.

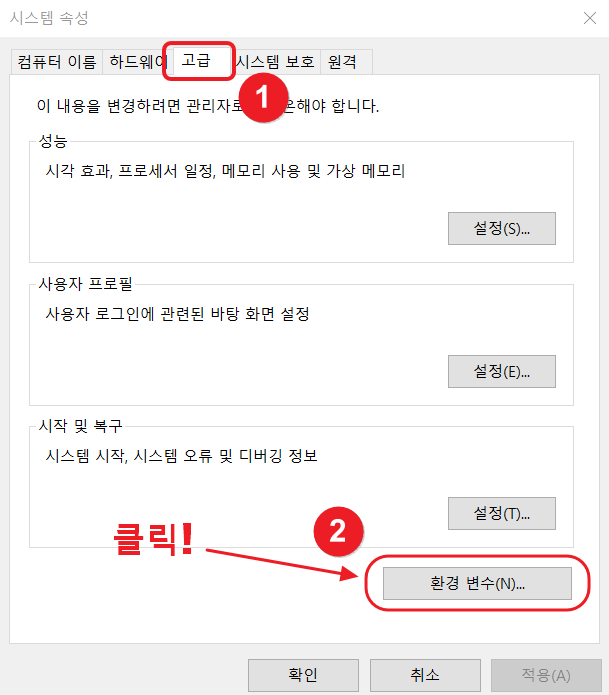
텍스트, 모니터, 전자기기, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

### **2.2.4.환경변수 설정(안드로이드)**

* 작업표시줄 검색창에 내 PC를 검색한 후 **내 PC 아이콘 우클릭 -> 속성**으로 들어갑니다.
* 설정창에서 스크롤을 내리면 관련 설정이라는 곳이 있습니다. 여기서**고급 시스템 설정을 클릭**합니다.
* 시스템 속성의 고급 탭에서 **환경 변수를 클릭**합니다.

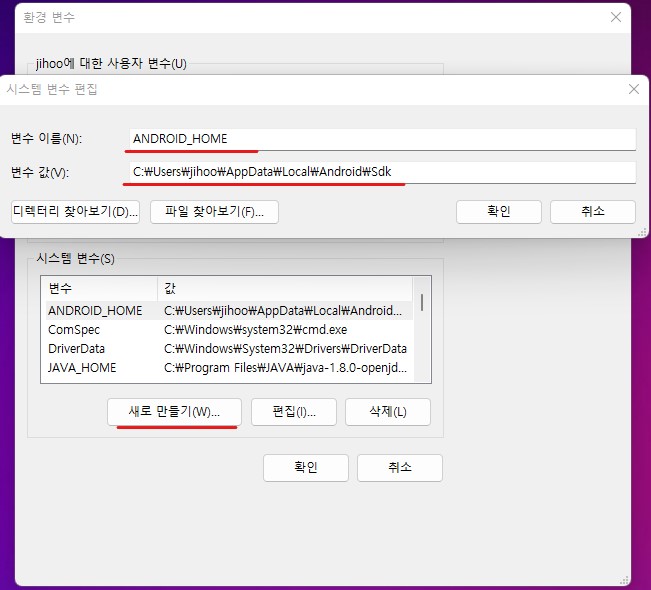
 



* 새로 만들기 버튼 클릭
* 변수이름 **ANDROID\_HOME** 입력
* 변수 값의 **Android SDK의 디렉토리 값을 넣어준다.**  확인 위치는 **1.2.3. Android SDK 설정 (안드로이드 스튜디오 에물레이터 설정)** 화면을 참고

**(Android SDK Location 확인 가능**)

**보통 \사용자\PC로그인사용자\AppData\Local\Android\Sdk** 확인 가능 [링크확인](#_2.2.3._Android_SDK)



* **시스템 변수 Path 클릭 후 편집 버튼 클릭**
* **새로 만들기 클릭 후**

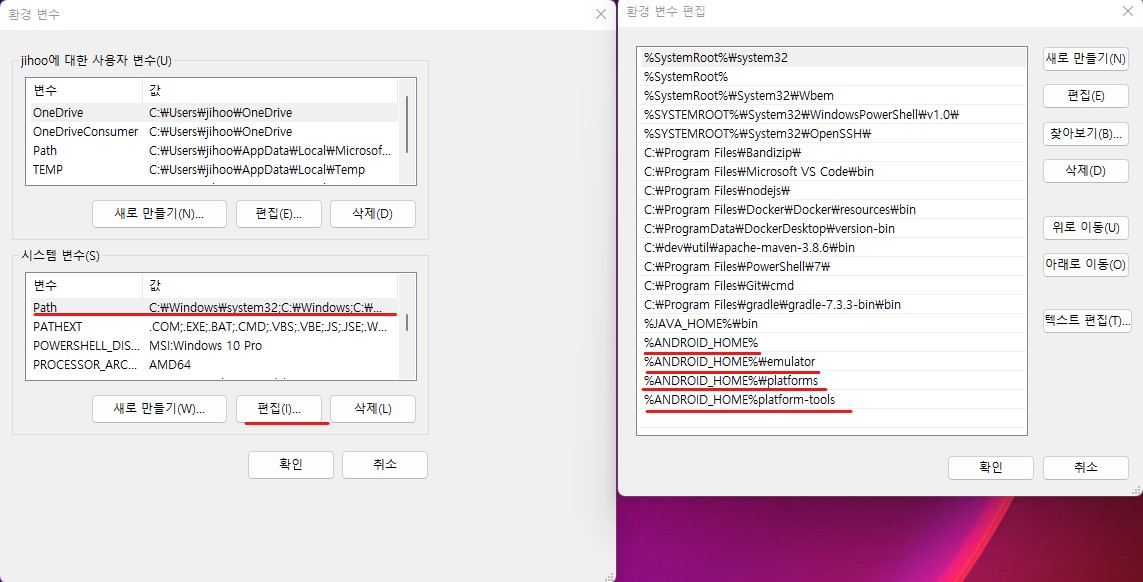
**%ANDRODIO\_HOME%**

**%ANDROID\_HOME%\emulator**

**%ANDRODIO\_HOME%\platforms**

**%ANDRODIO\_HOME%\platform-tools**

**등록**

****

# **3.Open JAVA SDK 1.8 설치**

## **3.1.개요**

## **3.2.다운로드**

* 링크 URL: [**https://github.com/ojdkbuild/ojdkbuild**](https://github.com/ojdkbuild/ojdkbuild)

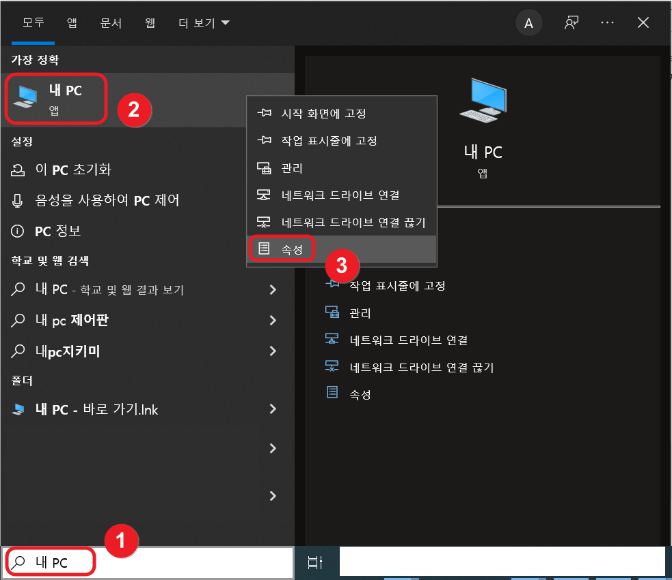
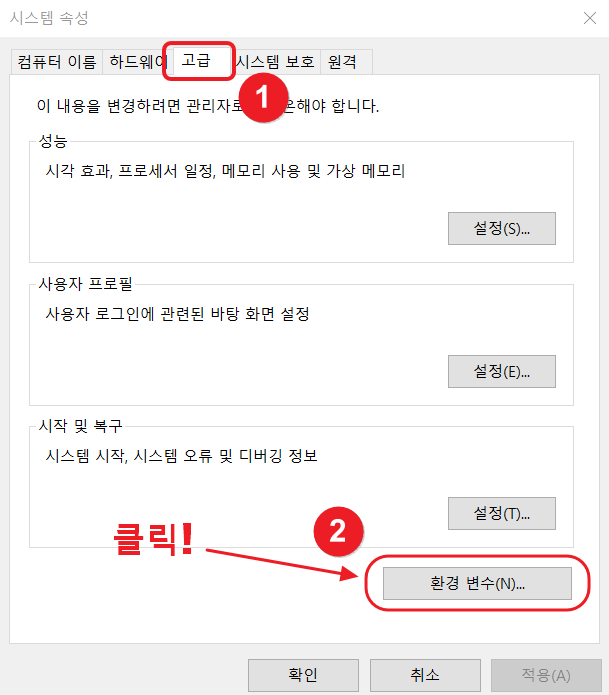
텍스트, 스크린샷, 모니터, 검은색이(가) 표시된 사진

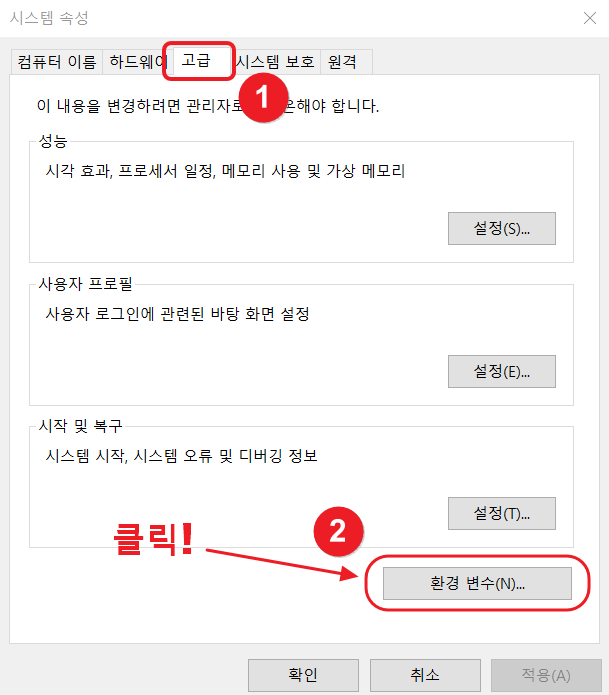
자동 생성된 설명

## 3.2.설치

### **3.2.1.환경변수 설정(JAVA)**

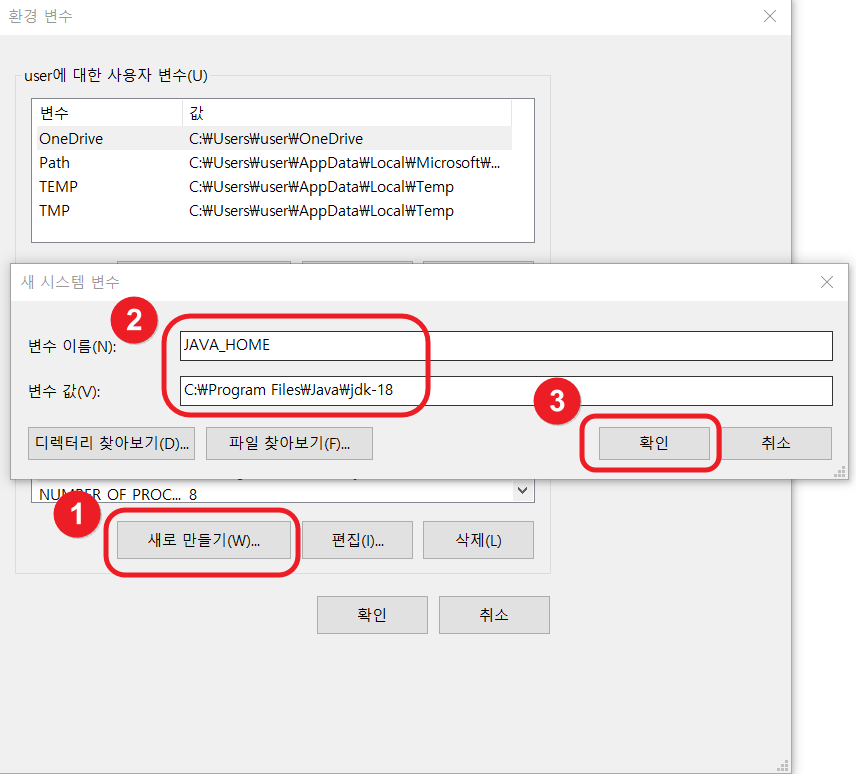
* 작업표시줄 검색창에 내 PC를 검색한 후 **내 PC 아이콘 우클릭 -> 속성**으로 들어갑니다.
* 설정창에서 스크롤을 내리면 관련 설정이라는 곳이 있습니다. 여기서**고급 시스템 설정을 클릭**합니다.
* 시스템 속성의 고급 탭에서 **환경 변수를 클릭**합니다.



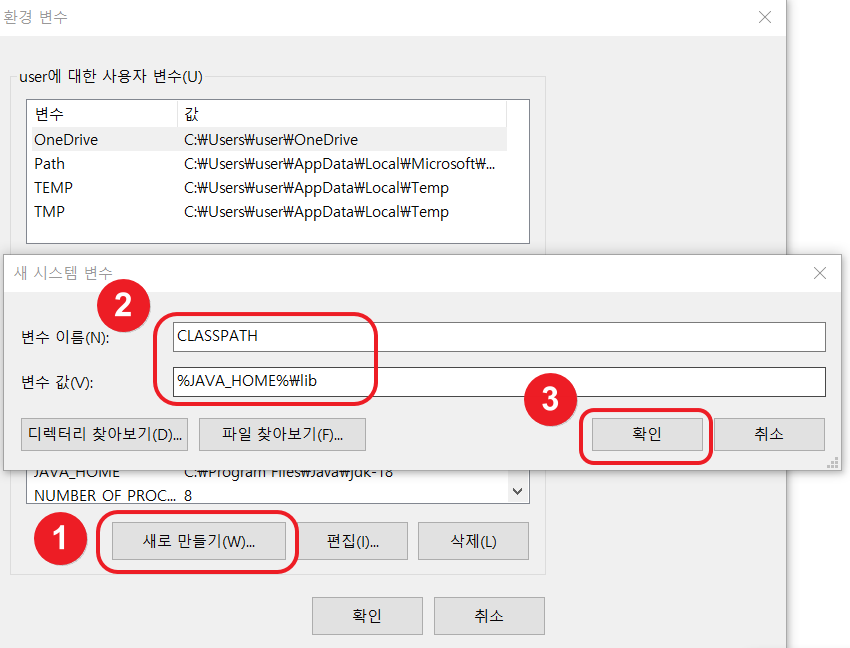
* 환경 변수에서 **새로 만들기** 버튼을 클릭한 후 아래 정보를 넣고 확인 버튼을 클릭
* **변수 이름 :**JAVA\_HOME

**변수 값 :**JDK가 설치된 폴더의 경로

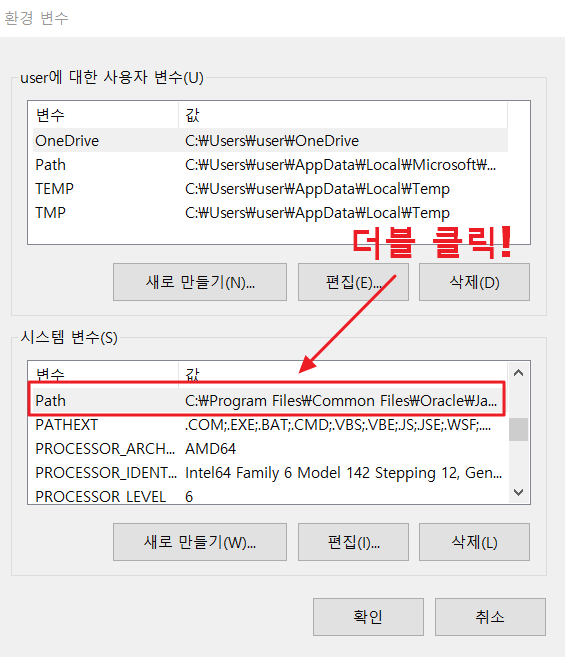


* 환경 변수에서 **새로 만들기** 버튼을 클릭한 후 아래 정보를 넣고 확인 버튼을 클릭합니다.
* **변수 이름 :**CLASSPATH

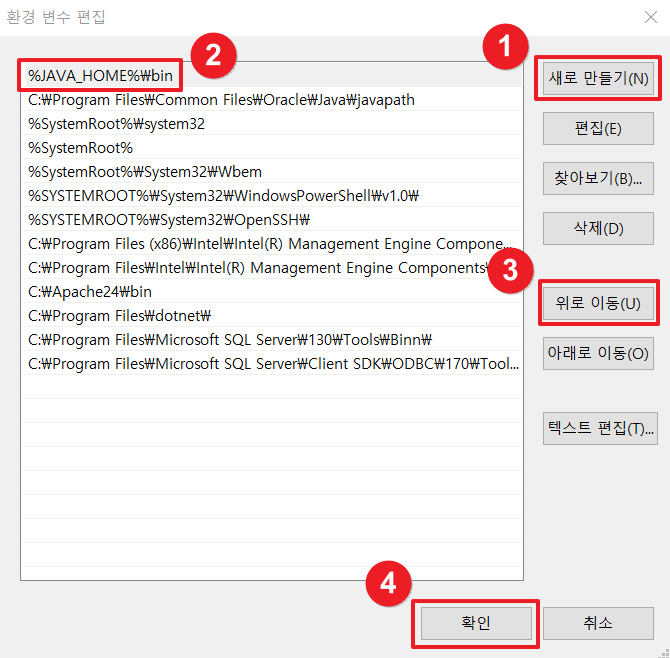
**변수 값 :** %JAVA\_HOME%\lib



* 시스템 변수 리스트에서 **Path를 더블클릭하여 편집**창으로 들어갑니다.

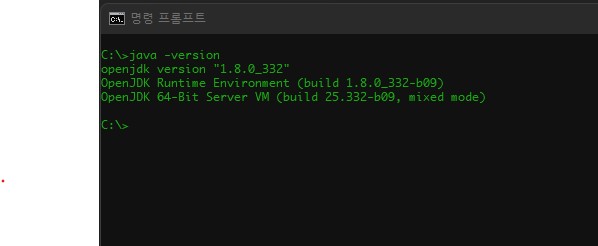


* 좌측의 **새로 만들기** 버튼을 클릭한 후 **%JAVA\_HOME%\bin**을 입력하고 위로 이동 버튼을 클릭하여 **가장 위로** 올려주고 확인 버튼을 클릭합니다.
* %JAVA\_HOME%\bin을 가장 위로 올려주는 이유는 명령 라인에서 명령어를 찾을 때, Path 환경변수에 등록된 순서대로찾기 때문입니다. 그렇기 때문에 %JAVA\_HOME%\bin
* 보다 먼저 등록된 폴더에 java 명령어가 있다면 다른 버전의 java 명령어가 사용될 수 있고 새로 설치한 java가 제대로 인식이 안되는 경우가 발생할 수 있기 때문에 가장 위로 올려줍니다.



자바 환경 변수 정상 확인하기

* cmd 창을 켠 후 java -version 명령어를 입력하여 설치된 자바의 버전이 나온다면 정상입니다.

****

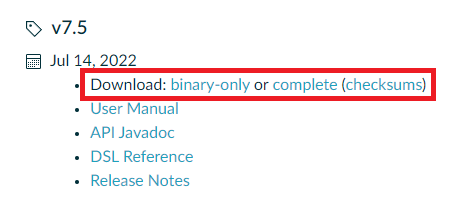
# **4.Gradle 설치**

## **4.1.개요**

## **4.2.설치**

### **4.2.1.다운로드**

* <https://gradle.org/releases/> 에서 원하는 버전에 binary-only 또는 complete을 선택하여 다운로드



### **4.2.2. Gradle 설치**

* 디렉토리(C:\gradle\)를 만들고 다운로드한 압축 파일을 풀어주세요.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

### 4.2.3. 환경 변수 설정

* 시스템 환경 변수 편집 > 환경 변수 > 시스템 변수 > Path를 선택한 후 C:\gradle\gradle-7.5\bin을 추가

텍스트이(가) 표시된 사진

자동 생성된 설명

* CMD 창에서 "gradle -v" 명령어를 입력하여 정상적으로 설치되었는지 확인

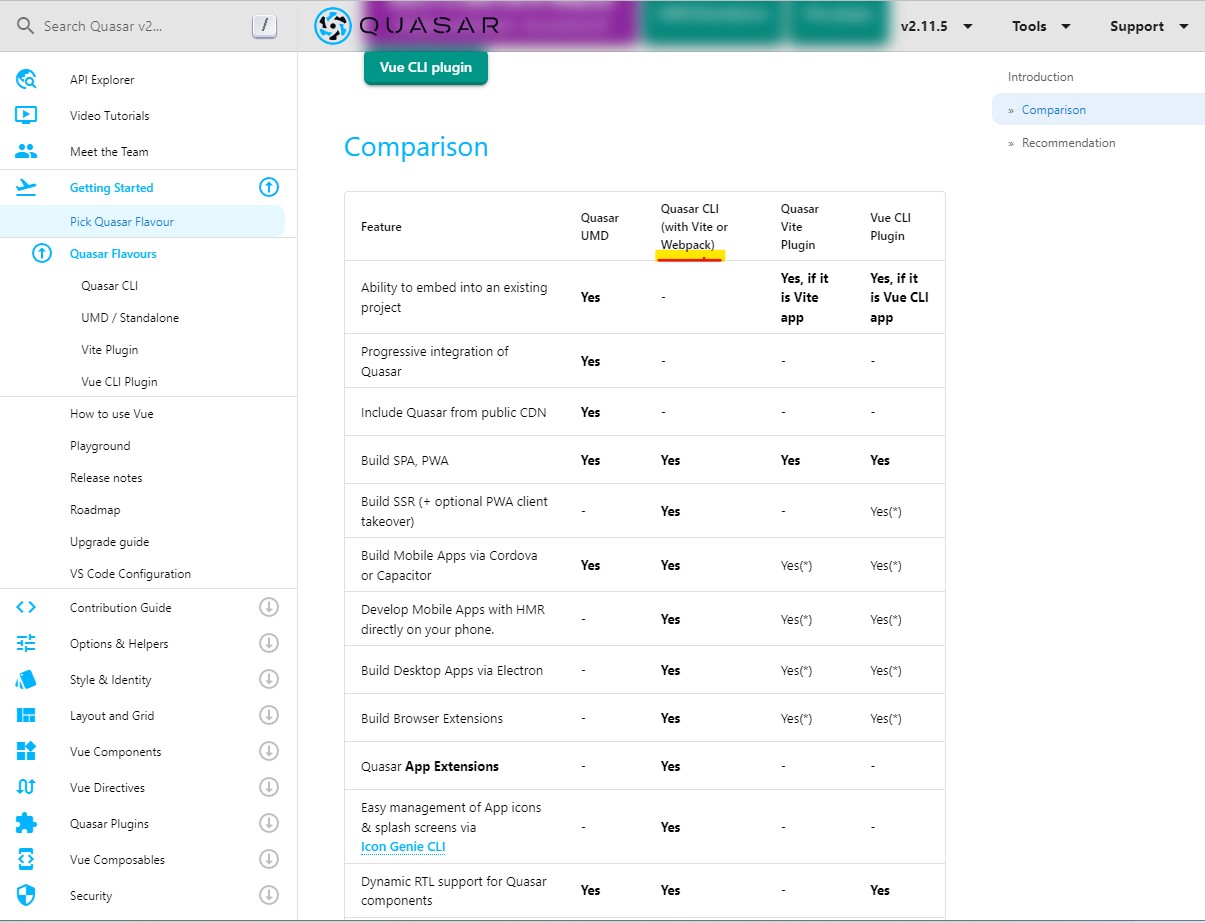
텍스트이(가) 표시된 사진

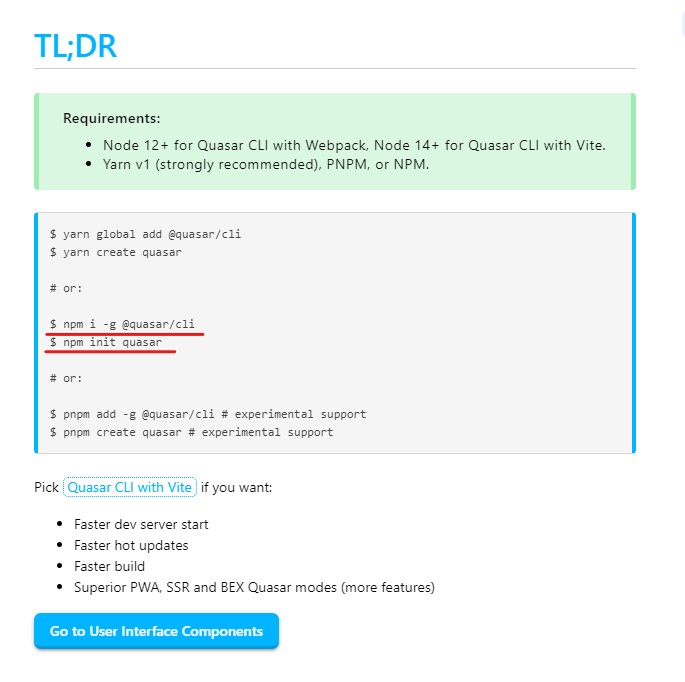
자동 생성된 설명

# **5.Quasar FrameWork**

## **5.1. Quasar FrameWork API Document**

* + - * 링크 : [**https://quasar.dev/**](https://quasar.dev/)
      * 설치 방법 링크:[**https://quasar.dev/start/**](https://quasar.dev/start/)
      * **Quasar CLI (with Vite or Webpack) 설치**





## **5.2. Quasar FrameWork 프로젝트 생성**

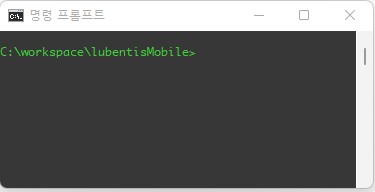
## **5.2.1. 프로젝트 workspace 생성**

**해당부분을 동영상으로 제작 한번 시청요망**

링크: <https://www.youtube.com/watch?v=33sJdtQG09g&list=PLlaP-jSd-nK8ce9LigEM6H-1vZ-EuJ85A&index=3>

* + - * + C:\workspace\lubentisMobile 디렉토리 생성
        + 명령 프롬프트(CMD) 실행 후

**cd \workspace\lubentisMobile**



* + - * + Quassar cli를 npm을 이용하여 global 로 설치해준다**. (필수 설치)**

**npm i -g @quasar/cli**

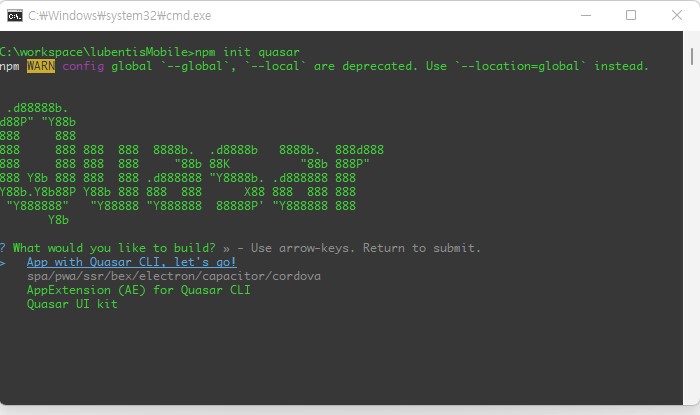
텍스트, 스크린샷, 전자기기, 디스플레이이(가) 표시된 사진

자동 생성된 설명

* + - * + **해당 디렉토리에서**

**npm init quasar**

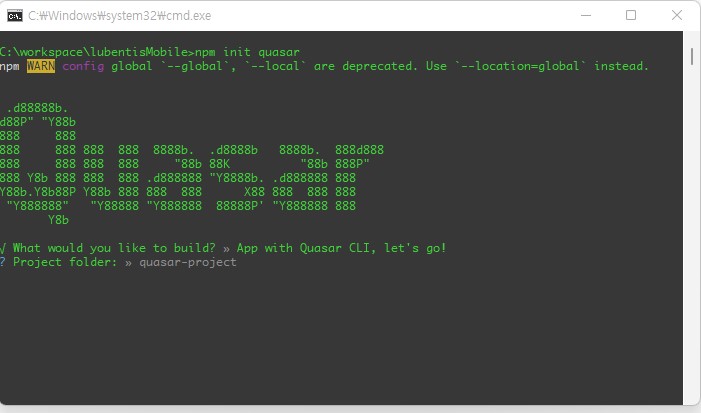
**App with Quasar CLI, let`s go! <- 선택ㄴ**

****

* + - * + **Project folder: >> quasar-project**

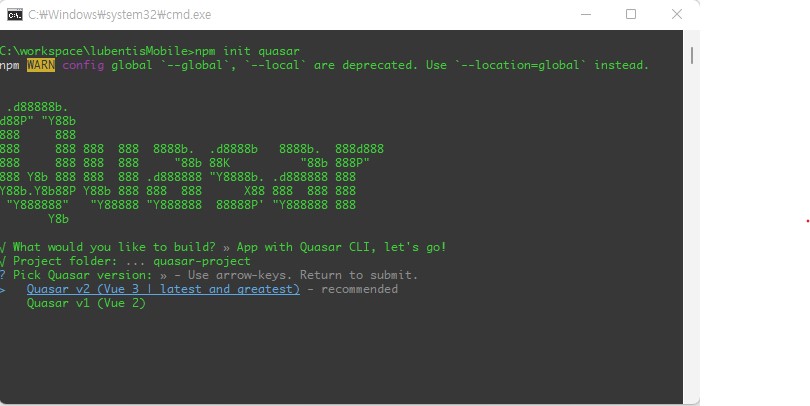
Quasar 프로젝트의 디렉토리를 생성한다.

예제)기본프로젝트로 선택

****

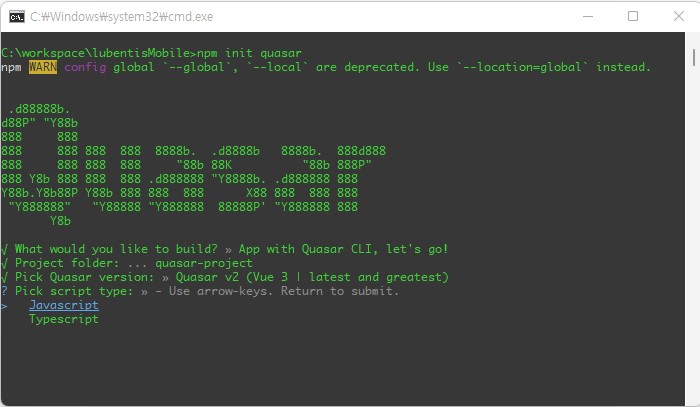
* + - * + **Quasar 프로젝트**의 사용할 **Quasar**과 연관 **Vue**버전을 선택한다.

예제) **Quasar v2(Vue 3 | latest and aretest) 선택**

****

* + - * + **Quasar 프로젝트**의 사용할 **스크립트 언어**를 선택한다**.**

예제) **Javascript** 선택

****

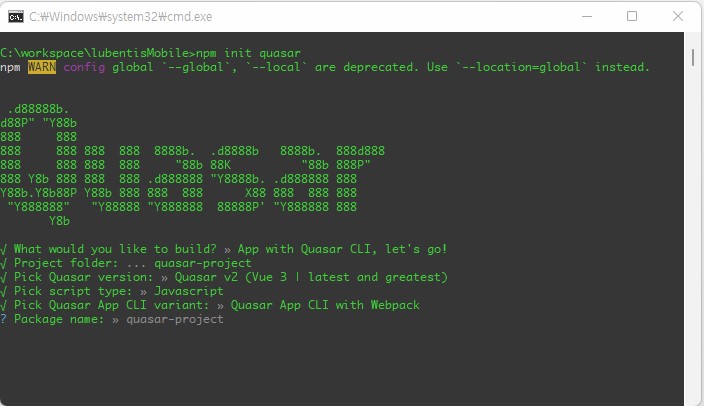
* + - * + **Quasar 프로젝트**의 사용할 자바스크립트 모듈러를 선택한다.

예제) **Quasar App CLI with Vite선택**

****

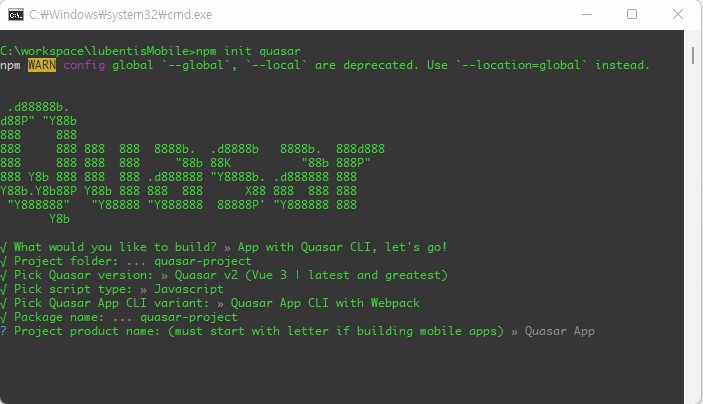
* + - * + **Quasar 프로젝트**의 **Package name을 지정한다.**

예제) 디렉토리와 동일한 **quassar-projects**로 지정한다.

****

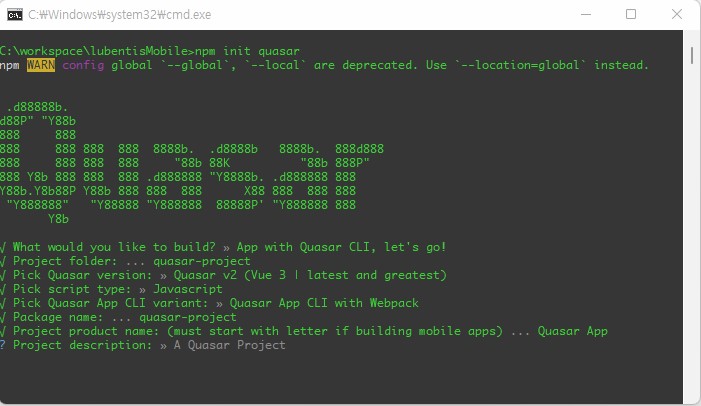
* + - * + **Quasar 프로젝트**의 **Project product name**을 지정한다. 만약 모바일로 빌드시 해당 name은 **App Name**이 된다.

예제) Quasar App으로 지정(기본)

****

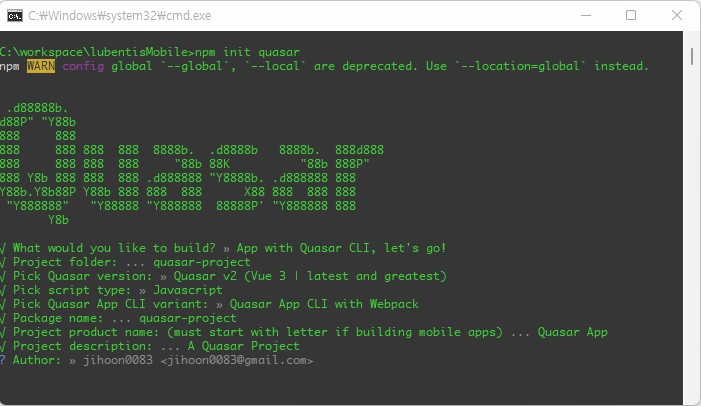
* + - * + **Quasar 프로젝트**의 **설명을 기술한다.**

예제) Project description:>> A Quasar Project (기본지정 사용)

****

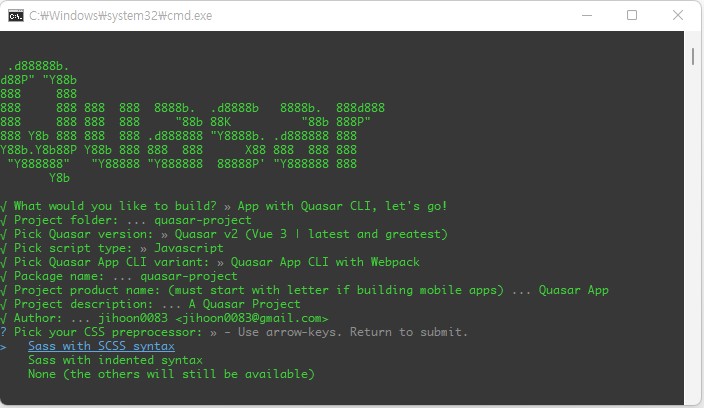
* + - * + **Quasar 프로젝트**의 Author를 기술한다.

예제) 본인기준 default(기본지정)을 사용

****

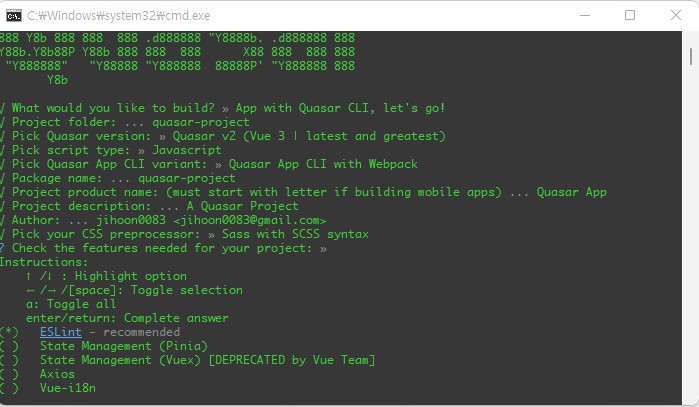
* + - * + **Quasar 프로젝트**의 사용할 **CSS Type**을 선택한다.

예제) Sass with SCSS syntax 지정

****

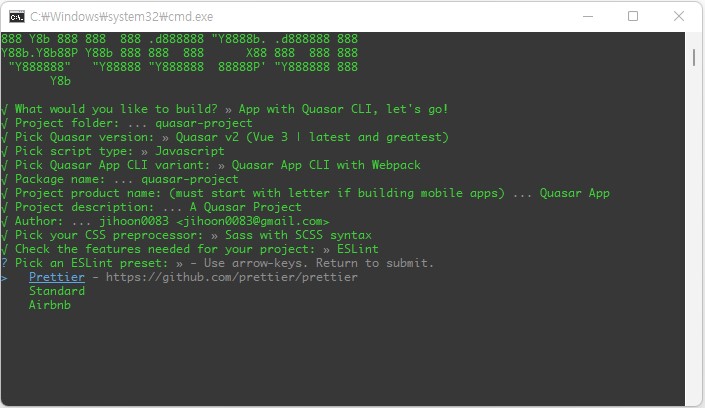
* + - * + **Quasar 프로젝트**의 정적코드 분석도구를 선택한다.

ESLint – recommended 만 선택 ( 다중선택 선택 )

****

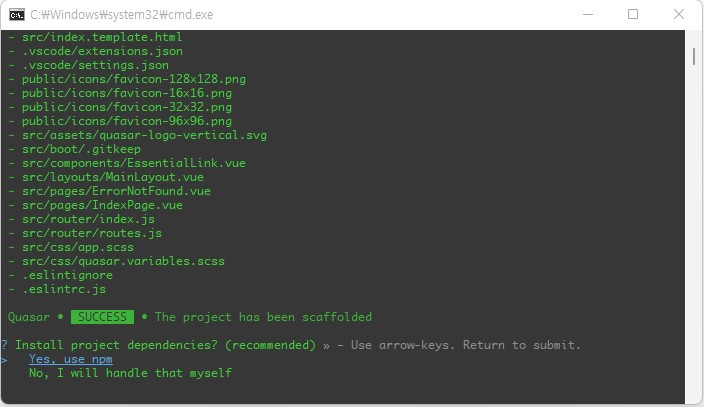
* + - * + **Quasar 프로젝트**의 사용할 코드 포맷터를 선택한다.

Pretter 선택하거나 Standard를 선택

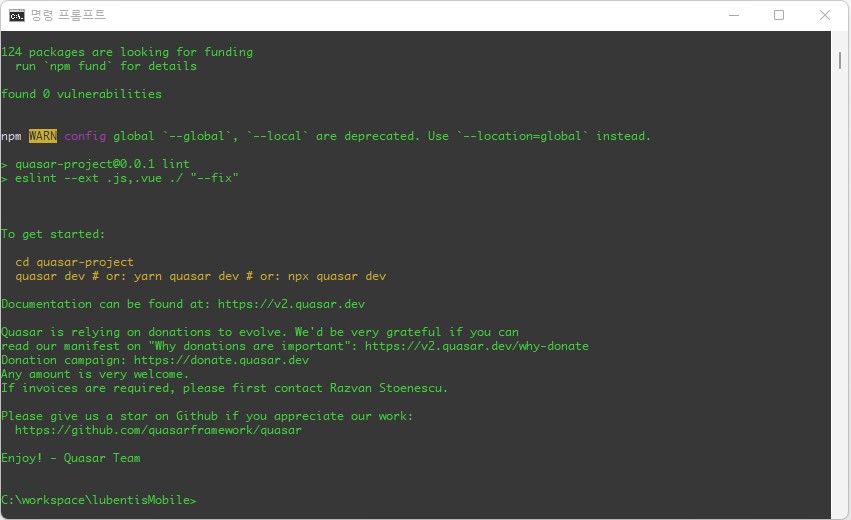
****

* + - * + **Quasar 프로젝트**의 사용할 의존성을 선택

예제 ) Yse, use npm 선택

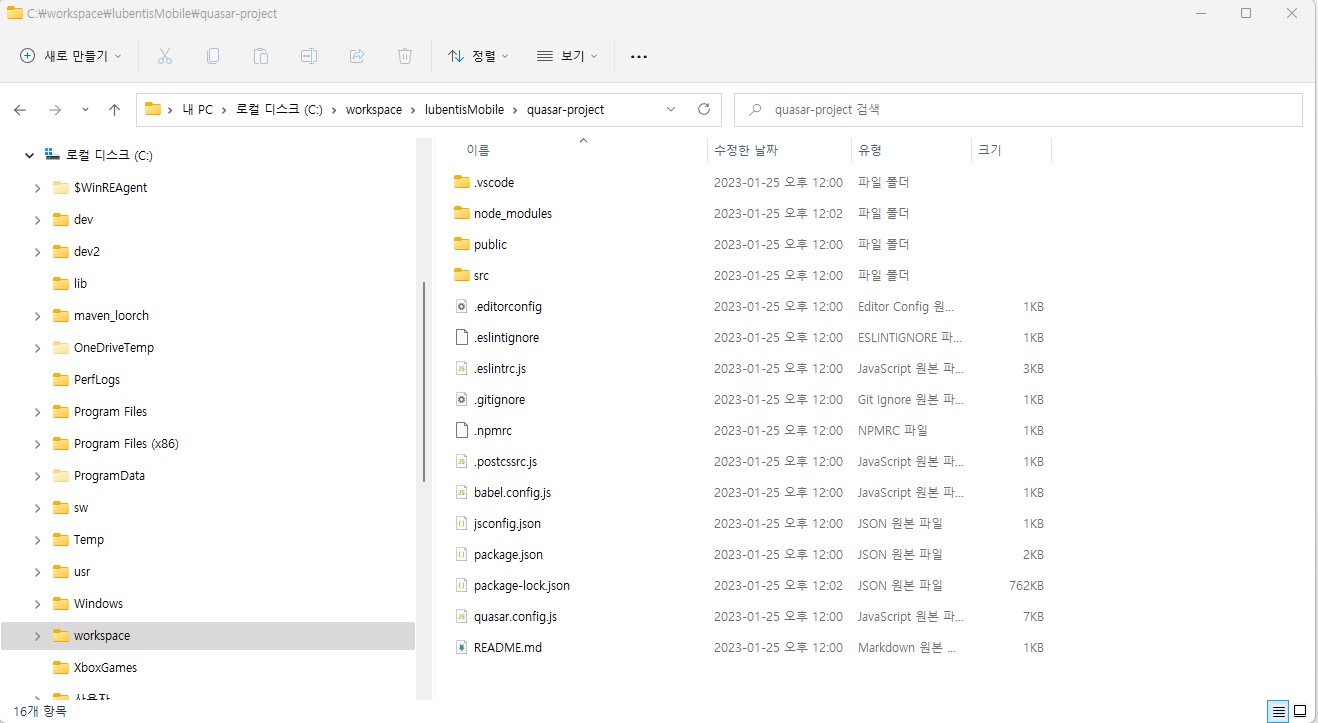
****

* + - * + **Quasar 프로젝트** 설치완료

****

* + - * + 설치가 완료되면 quasar-project 디렉토리가 생성

**Quasar 프로젝트의 root 디렉토리가 된다.**

**Sㄴ**

* + - * + 해당화면은 비주얼 스튜디오 코드를 사용하여 **Quasar 프로젝트 C:\workspace\lubentisMobile\quasar-project\package.json** 파일을 오픈

**“scripts”: <- 부분에**

**“dev”: ”quasar dev”,**

**“build”: “quasar build”**

추가

**텍스트이(가) 표시된 사진

자동 생성된 설명**

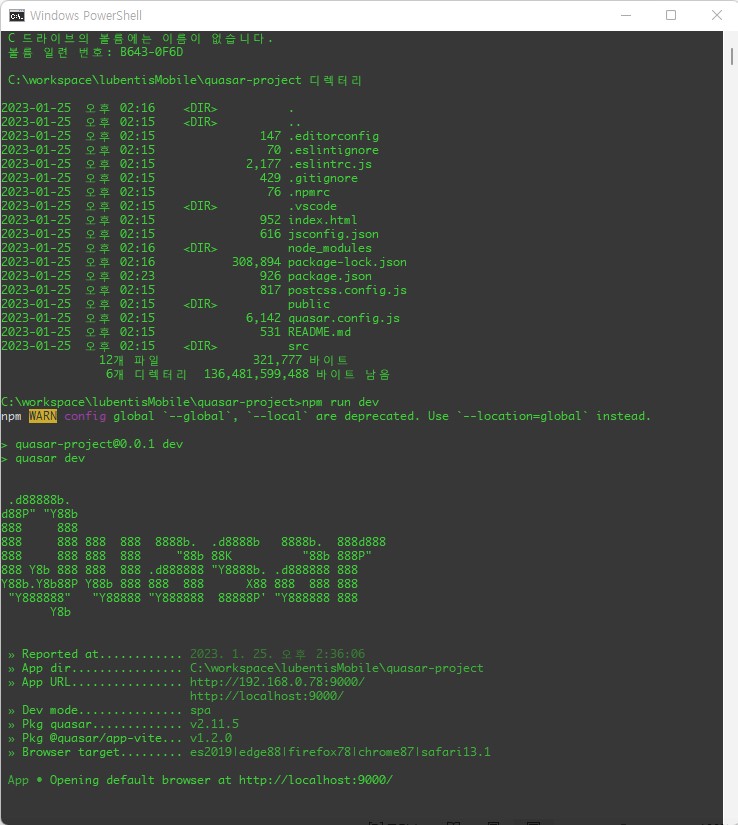
* + - * + 명령프럼프트 실행(CMD)

**cd \workspace\lubentisMobile\quasar-project**

**C:\workspace\lubentisMobile\quasar-project** 해당위치에서

**npm run dev** 명령어 실행

[**http://localhost:9000**](http://localhost:9000) **으로 접속**

****

**포트의 경우 자동 부여 만약 고정하고 싶다면**

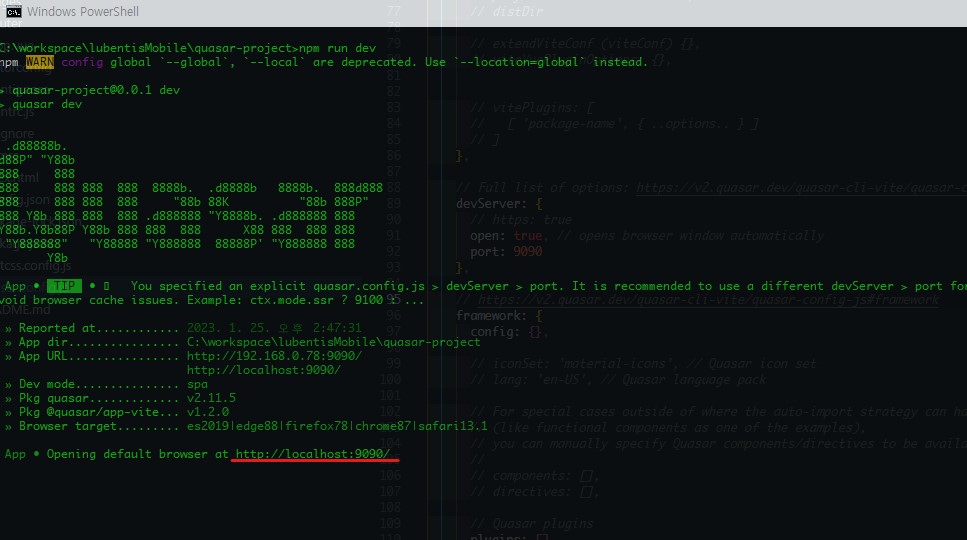
**C:\workspace\lubentisMobile\quasar-project\quasar.config.json 파일오픈**

**devServer:{**

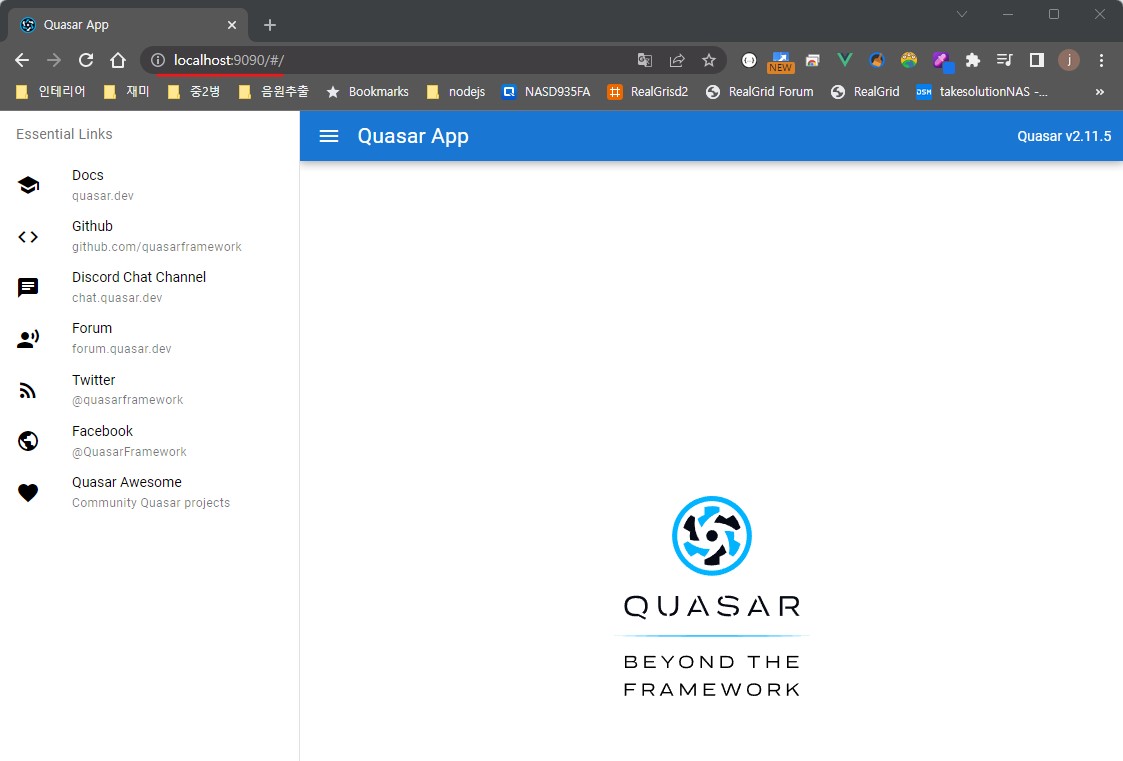
**}**

부분에 포트추가

**Port: 9090**

**s**

[**http://localhost:9000**](http://localhost:9000) **접속 해당화면이 나오면 우선설치 완료**

****

# **6. Cordova Framework**

## **6.1. 개요**

* **Node.js** 플랫폼을 기반으로 하는 하이브리드 앱 개발 프레임워크
* **크로스플랫 폼(안드로이드, IOS, Window Phone, Fire OS) 지원**
* 아파치 코르도바 사이트 링크 : [**https://cordova.apache.org/**](https://cordova.apache.org/)
* Quasar Framework와 동시사용 동영상 강좌 : [**https://www.youtube.com/watch?v=aay091vGPoQ&list=PLlWoe5hcgrk4qQVIBxDA3d-5ZRfYuITxb&index=16**](https://www.youtube.com/watch?v=aay091vGPoQ&list=PLlWoe5hcgrk4qQVIBxDA3d-5ZRfYuITxb&index=16)

해당강좌를 기반으로 해당 사항을 기술한다.

## **6.2.설치**

* Quasar Framework을 npm 이용하여 global 로 설치해준다**. (필수 설치)**

**npm install -g cordova**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

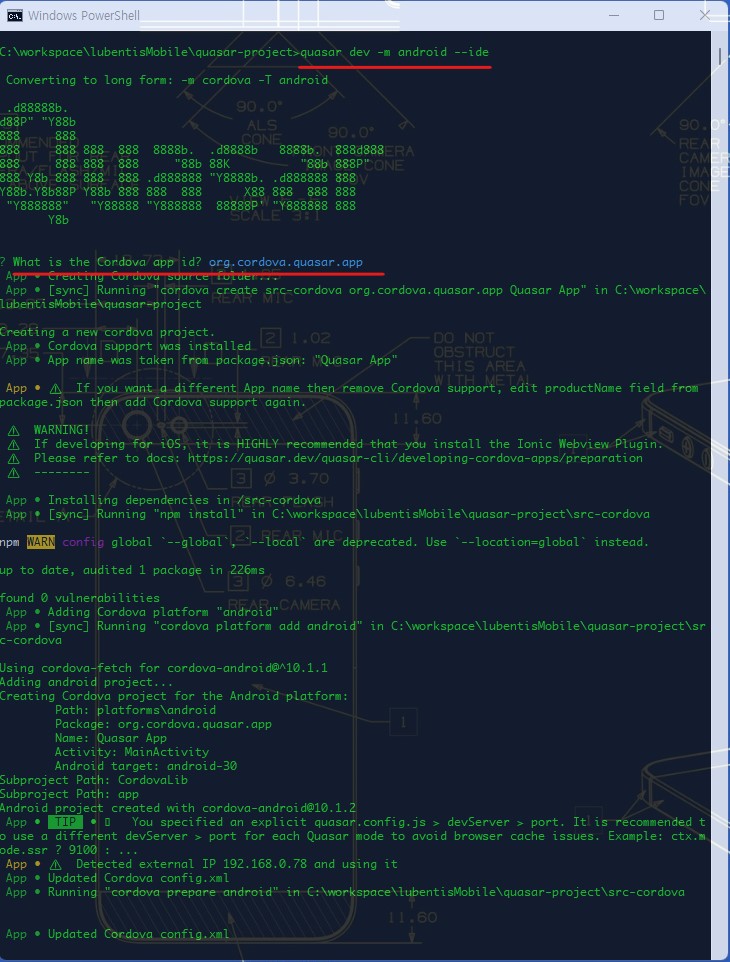
* 해당 퀘이사 프로젝트 폴더로 이동

예제) **C:\workspace\lubentisMobile\quasar-project**

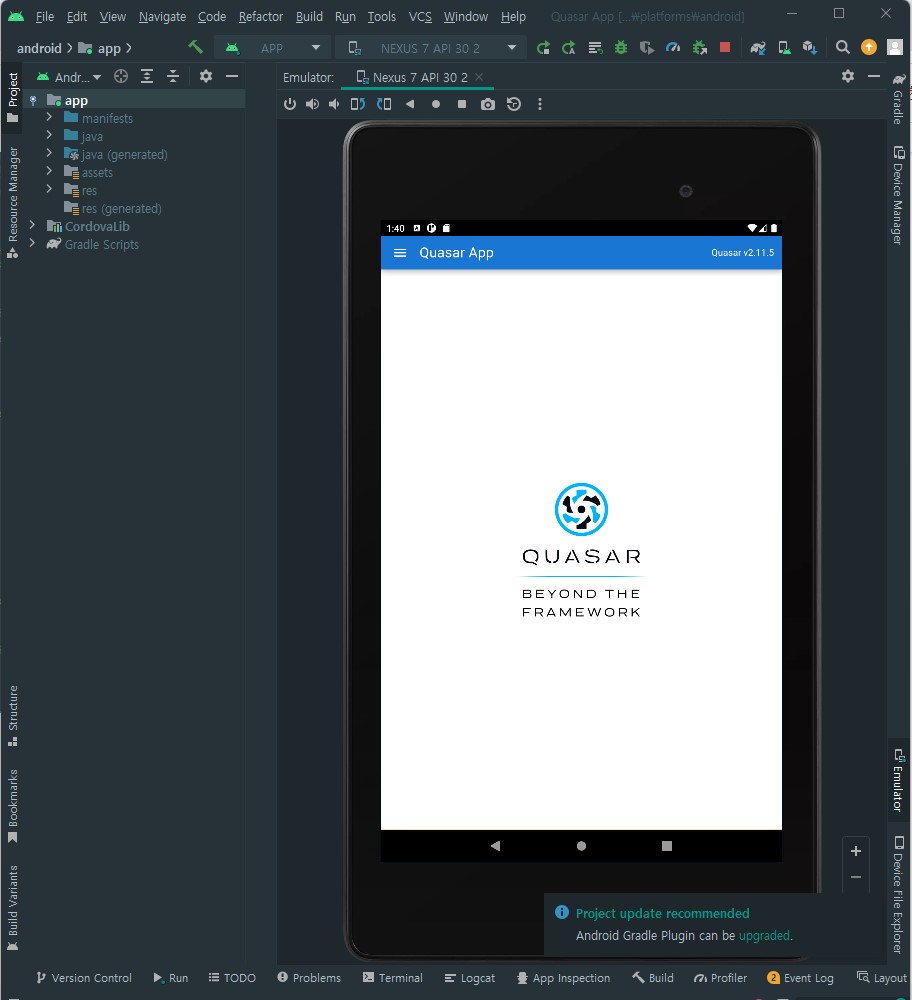
해당 예제는 안드로이드 기반으로 작성

**quasar dev -m android --ide** 실행(현재 쿼이사 프로젝트를 모바일 안드로이드기반에 에물레이터에 실행)

중간 **app id**를 묻는데 기본 **org.cordova.quasar.apps (수정)**

****

현재 퀘이사 프로젝트화면이 안드로이드 스튜디오에 있는 에물레이터에 표시



* **기본폴더 구조**

퀘이사 프로젝트 폴더 아래 **src-cordova** 코르도바 프로젝트 기본폴더 생성확인

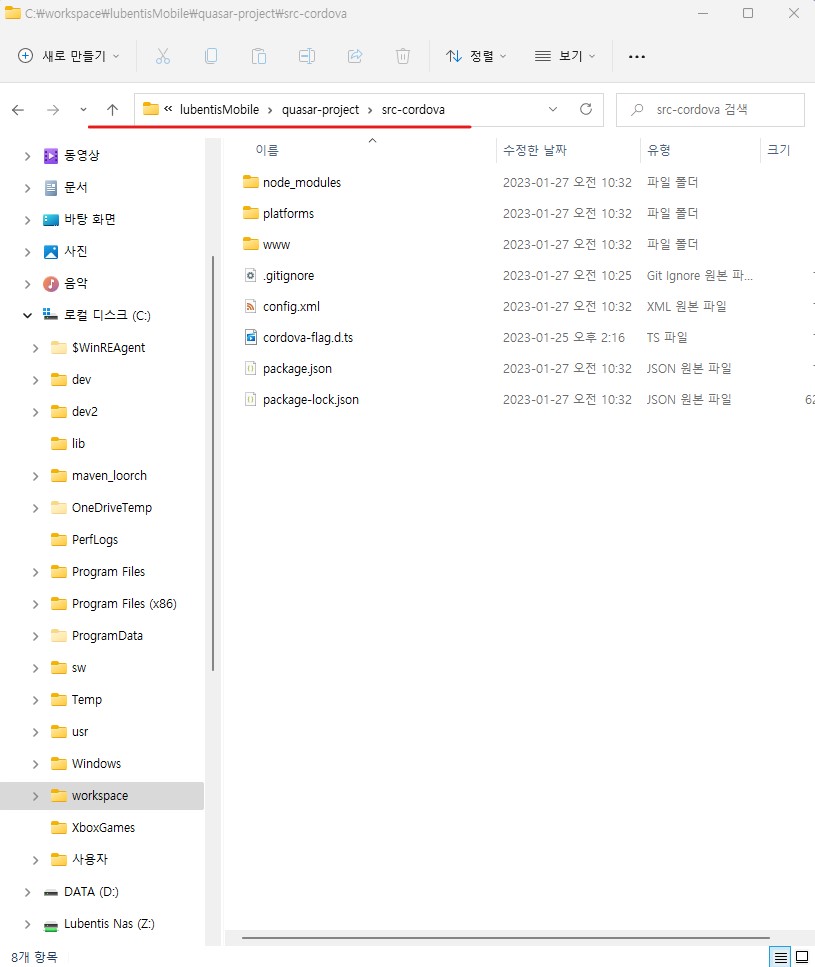
예제)

퀘이사 프로젝트 기본폴더 : **C:\workspace\lubentisMobile\quasar-project**

코르도바 프로젝트 기본폴더 : **C:\workspace\lubentisMobile\quasar-project\src-cordova**

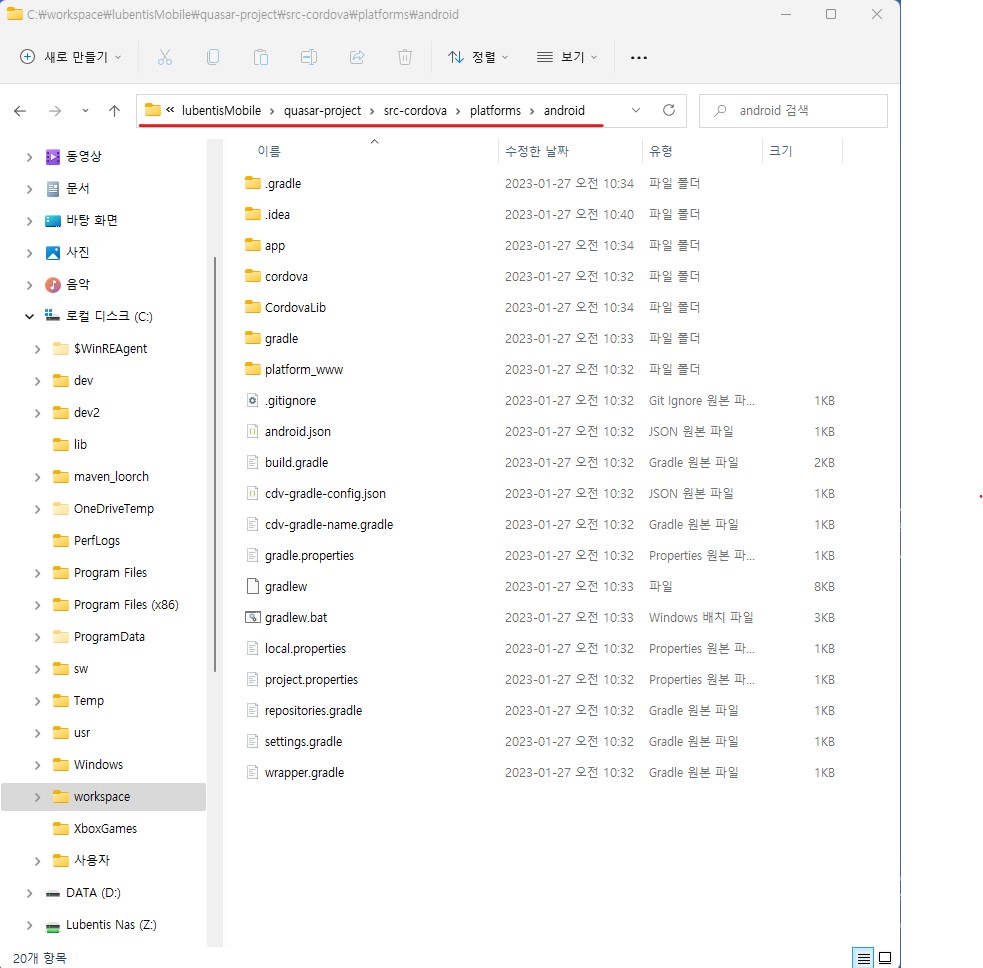
**( 주의 : 모바일 코르도바플러그인 설치시 코르도바 프로젝트 기본폴더기준으로 설치)**





**C:\workspace\lubentisMobile\quasar-project\src-cordova\platforms\android**

안드로이드 컨버전 파일위치

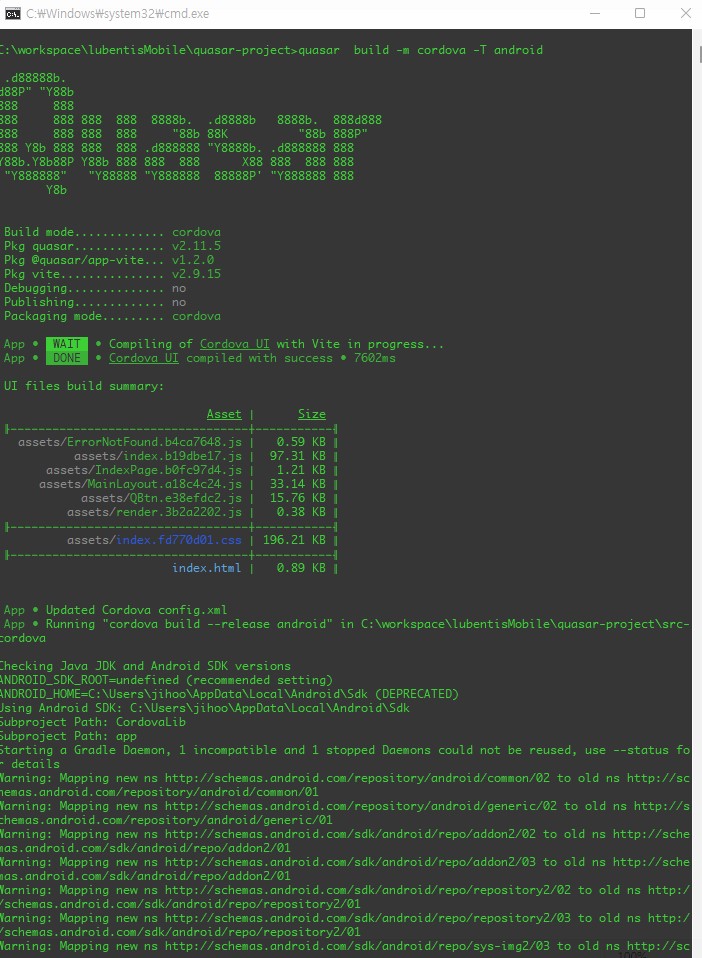


* 안드로이드 **apk** 파일생성 (실기 테스트 용)

현재 퀘이사 프로젝트 실제 기기에서 돌아가도록 배포 파일생성

퀘이사 프로젝트 기본폴더에서 ( 예제: **C:\workspace\lubentisMobile\quasar-project** )

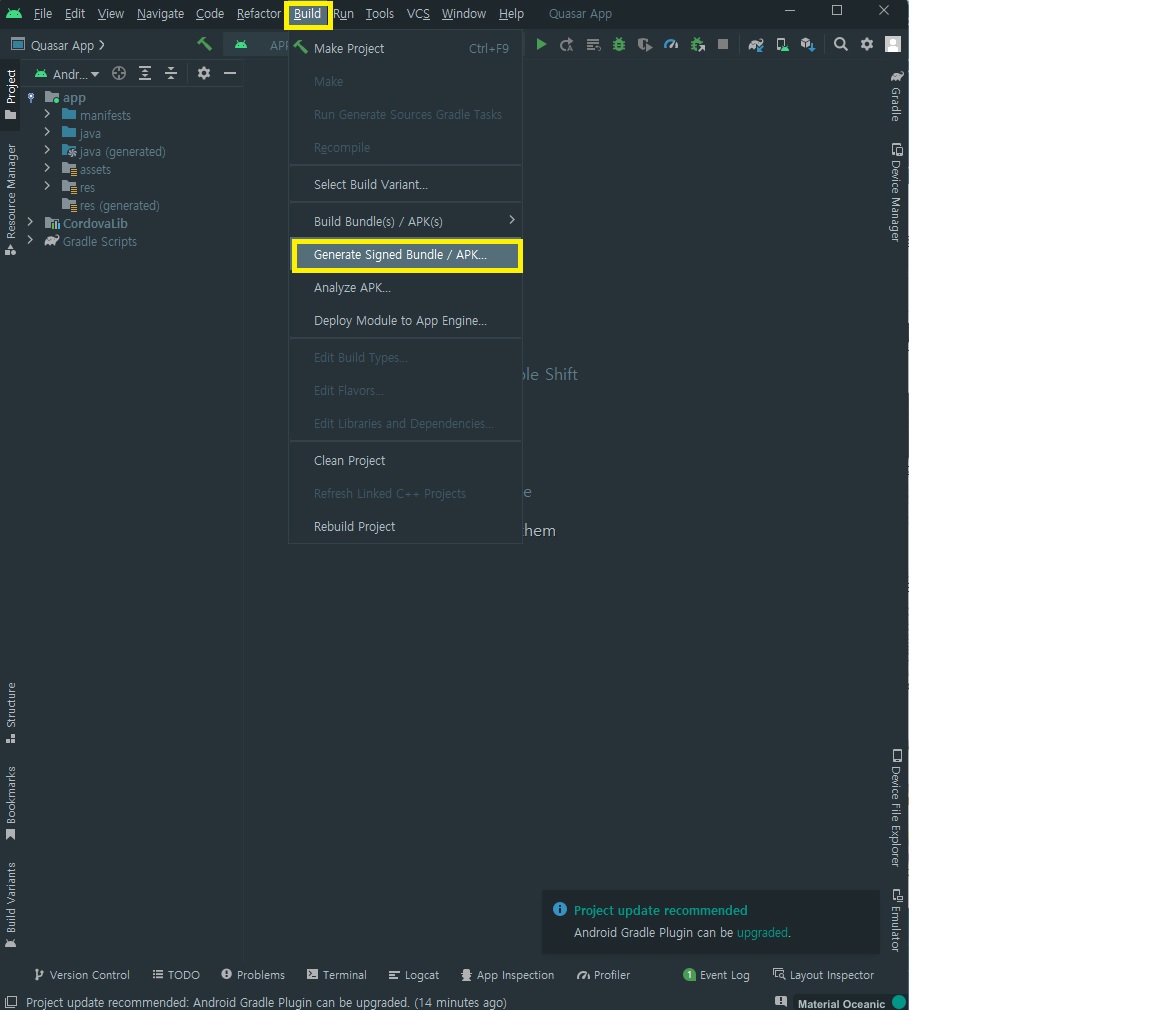
**quasar build -m cordova -T android 실행**



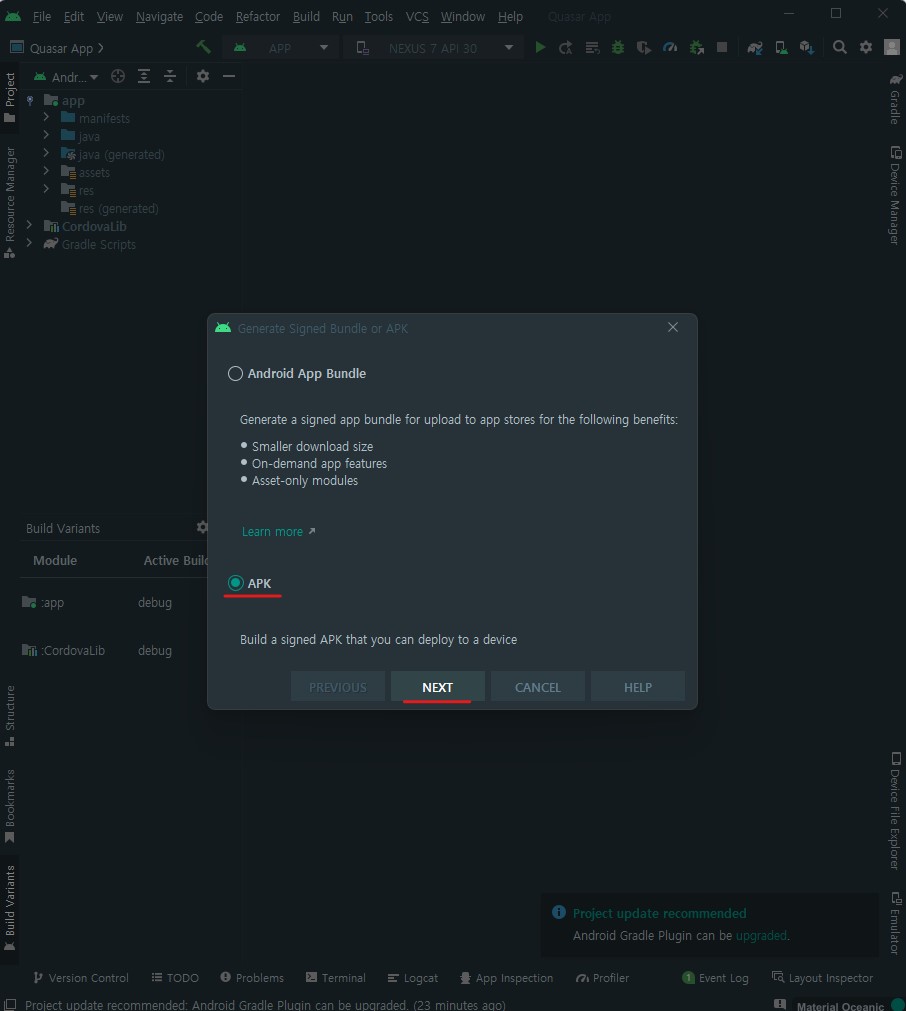
안드로이드 스튜디오가 닫혀 있으면 실행 해당 프로젝트 선택

안드로이드 스튜디오에서 절대 컴파일하지 말 것 ( 컴파일 할 경우 **quasar dev -m android --ide 실행으로 파일 원복**)

Build > Generate Signed Bundle / APK



APK 선택 후 다음버튼 클릭( NEXTS )



안드로이드 앱을 플레이스토어에 올리려면 배포(만들 때)할 때 자신의 코드 서명 인증서로 서명을 해야 함.

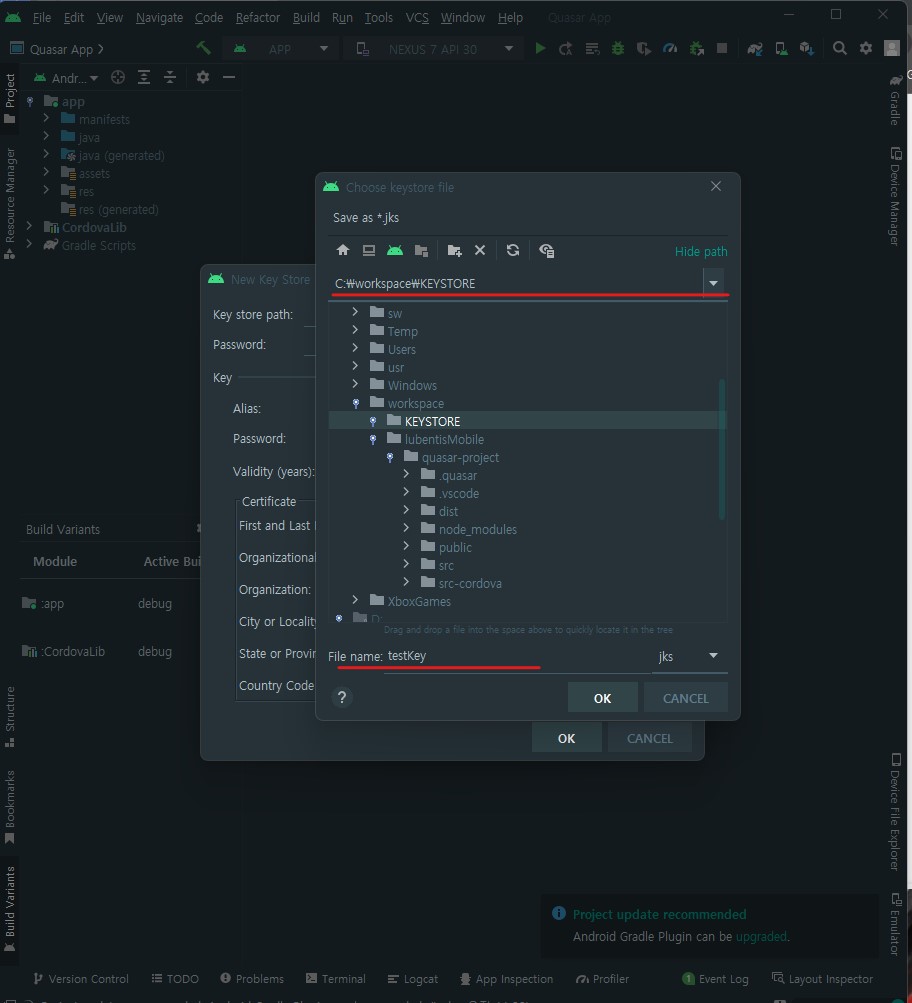
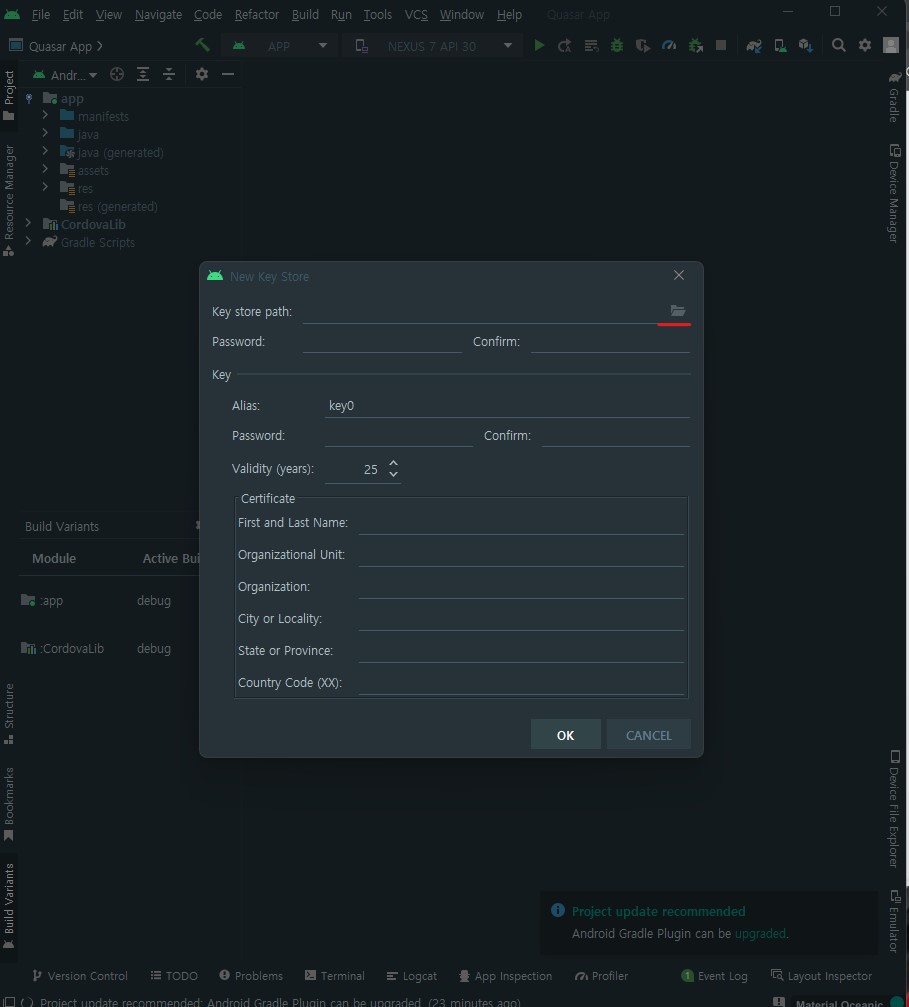
이후 이 서명이 동일해야 플레이스토어에서 앱 업데이트 가능

스튜디오에서 만들어 앱에 서명하는 방법

**New Key Store** 창에서 **Key store path**의 위치 아이콘을 클릭



자신의 프로젝트 경로 혹은 원하는 키 저장소 위치를 선택하고 file name에 식별할 수 있는 이름을 입력 후 OK버튼을 클릭



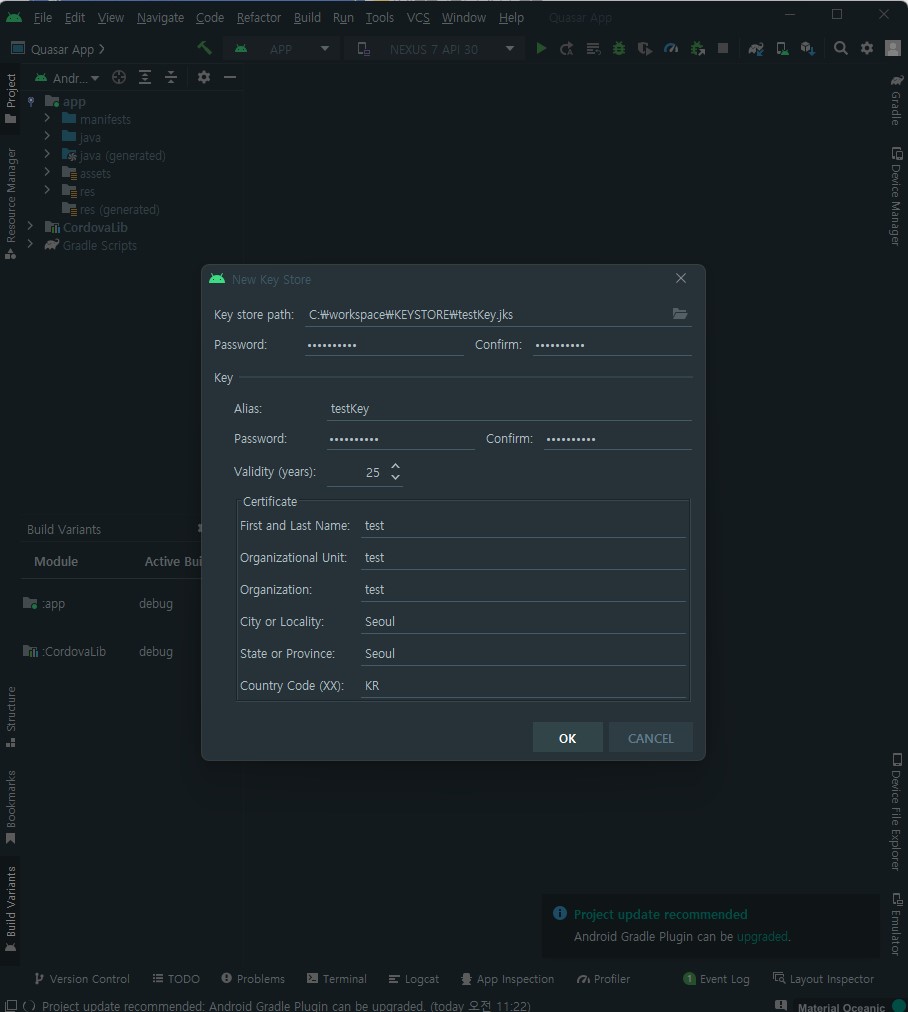
new key store창에서 키정보를 아래와 같이 입력 후 OK 버튼클릭

Alias: 키를 식별할 수 있는 이름을 입력

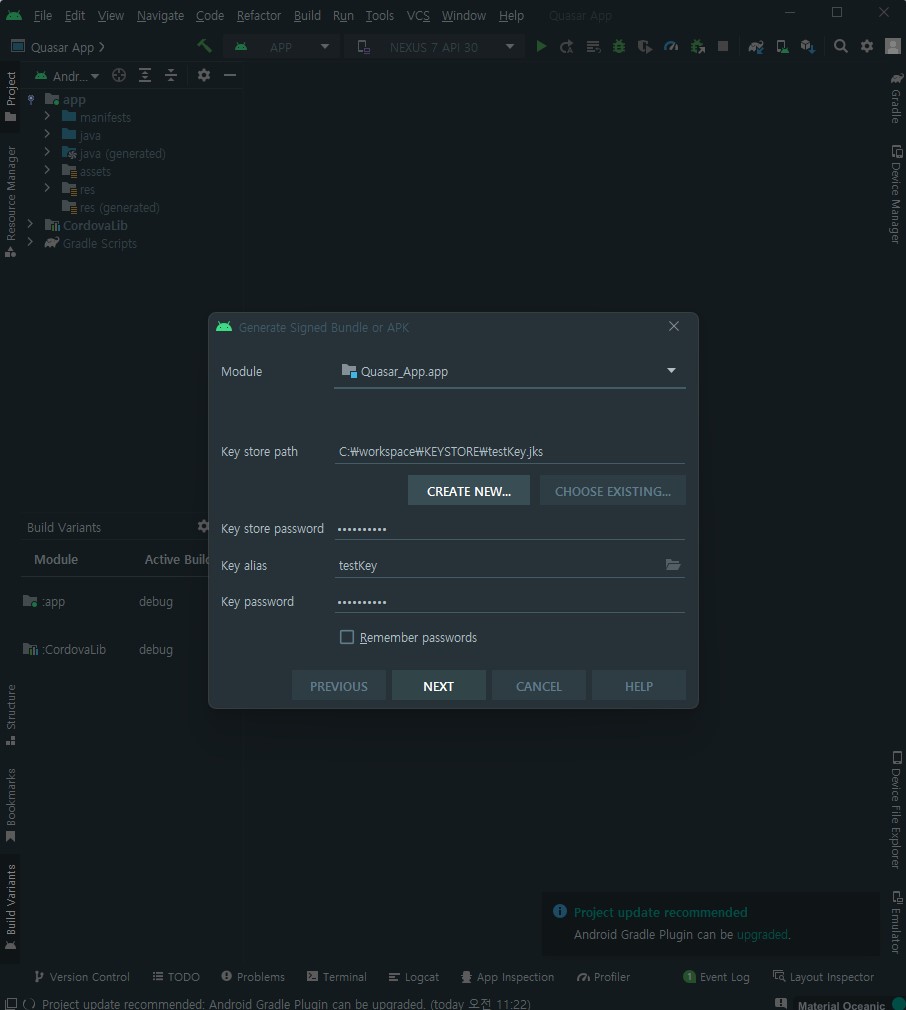
Password: 키 암호를 입력합니다. 이 암호는 키 저장소 암호와 다르게 설정하는 것이 권고.

Validity (years): 키가 유효하게 유지되는 기간(년)을 설정

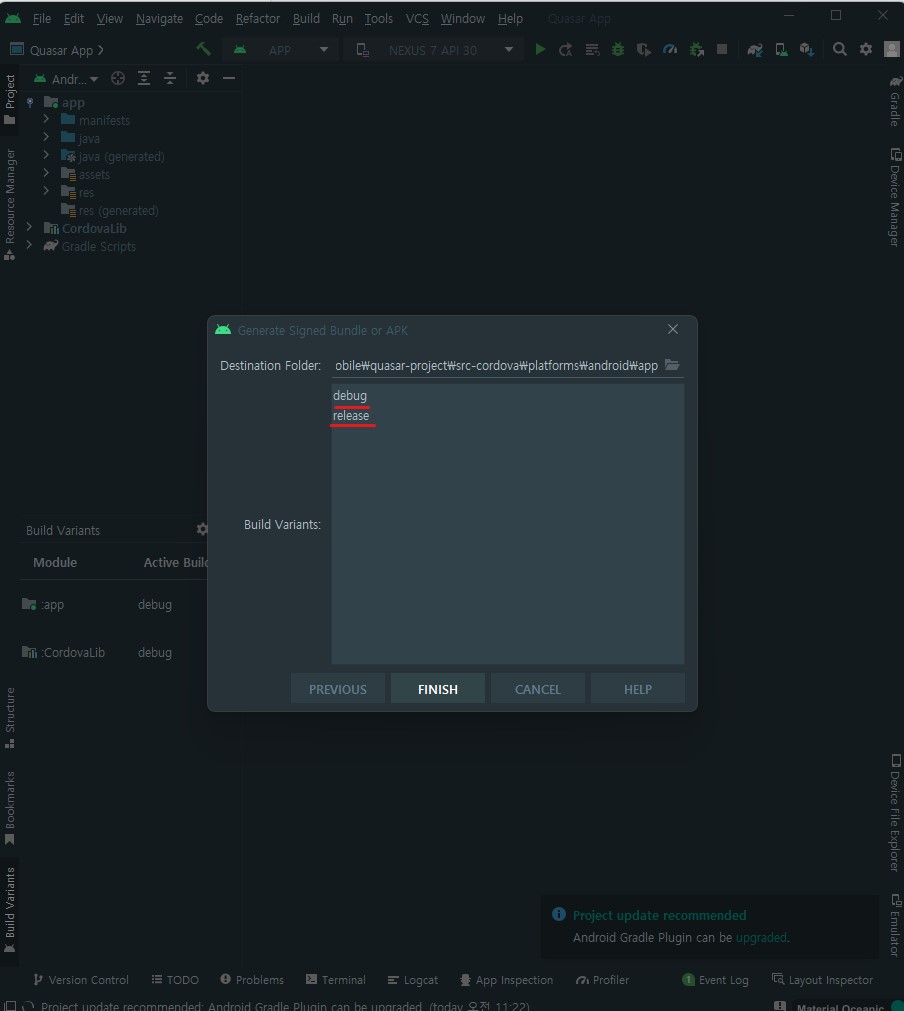
키가 최소 25년 동안 유효하게 유지되어야 앱 수명 동안 같은 키로 앱 업데이트에 서명할 수 있음



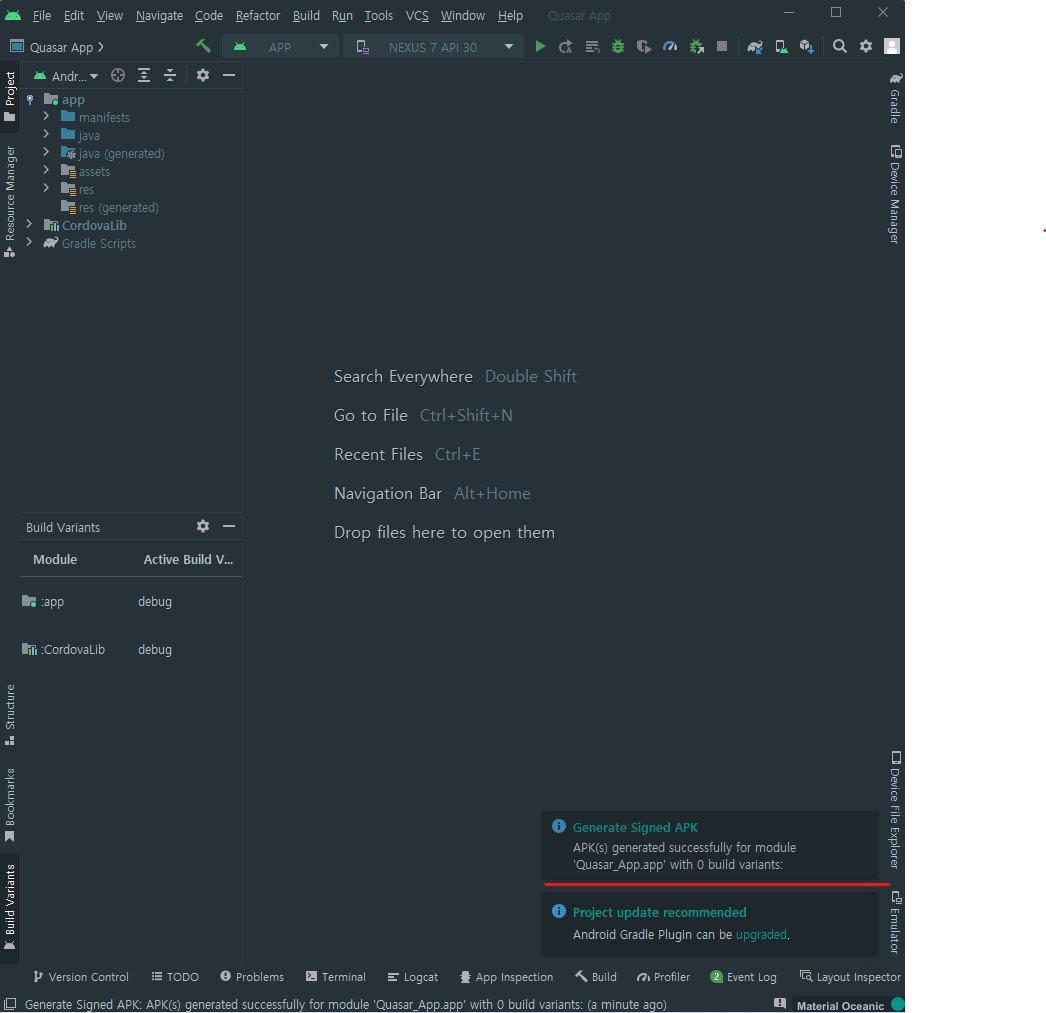
NEXT (다음) 버튼 클릭



debug / release 중 release 선택하고 FINISH 버튼 클릭



APK 컴파일 완료



**APK** 파일위치

**C:\workspace\lubentisMobile\quasar-project\src-cordova\platforms\android\app\release**

**app-release.apk** 파일 확인

