Android 앱 빌드 속도를 높이는 10가지 방법

Jirawatee | 2019.09.20
I am a technology evangelist at LINE.

> *참고. 이번 블로그는 LINE DEVELOPER DAY 2018에서 Jirawat Karanwittayakarn 님이 발표한 'Build your Android app Faster and Smaller than ever' 세션을 기록한 내용을 각색하여 옮긴 글입니다(원문 기록 및 제공: logmi).*

들어가며

이번 글에서는 Android 앱 빌드 속도를 높일 수 있는 10가지 방법에 대해 알아보려고 합니다. 빌드 속도는 생산성 향상을 위해 매우 중요한 요소입니다. 저는 APK 하나를 생성하는 데에 8분 넘게 걸렸던 적이 있습니다. 각 프로젝트 상황에 따라 더 오래 걸리는 팀도 있을 텐데요. 그게 바로 우리 팀 얘기라면 이번 블로그를 꼭 읽고 제가 소개드리는 방법을 잘 활용하셨으면 좋겠습니다.

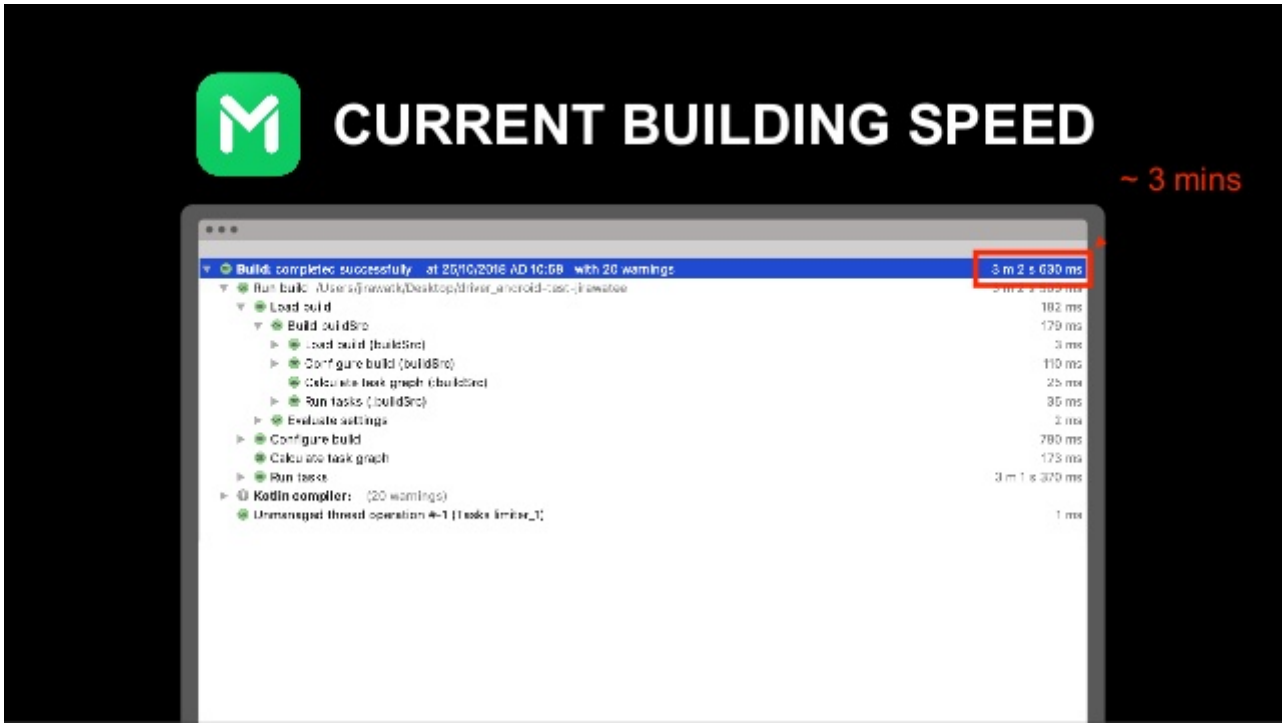
참고로, 이번 글의 내용은 각 프로젝트의 성격이나 환경에 따라 적용 방식이나 결과에 차이가 발생할 수 있다는 점을 염두에 두시기 바랍니다.

Android 앱 빌드 속도를 높이는 10가지 방법

그럼 LINE MAN 드라이버용 앱 빌드 속도를 높일 때 유용했던 10가지 방법을 소개하겠습니다.

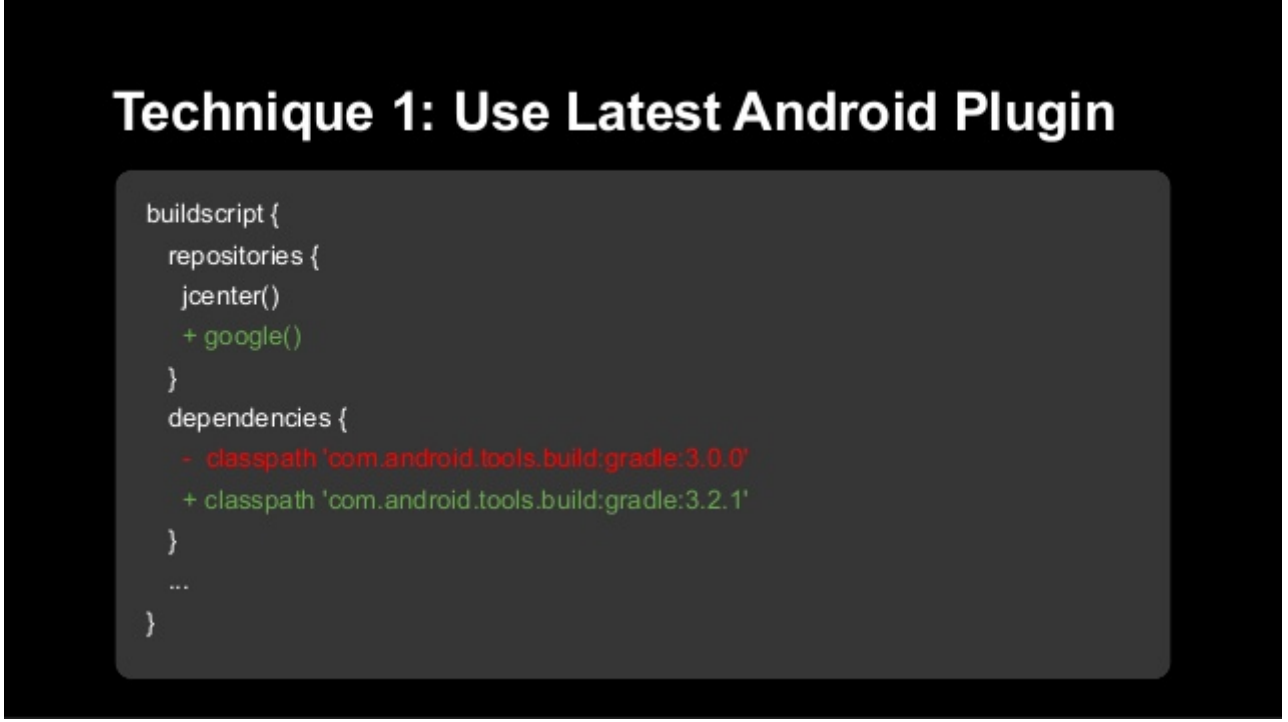
적용 전

각 팁을 적용하기 전에는 LINE MAN 앱을 Android Studio로 전체 빌드하는 데 3분이 걸렸습니다. 이 시간을 얼마나 단축할 수 있을까요?



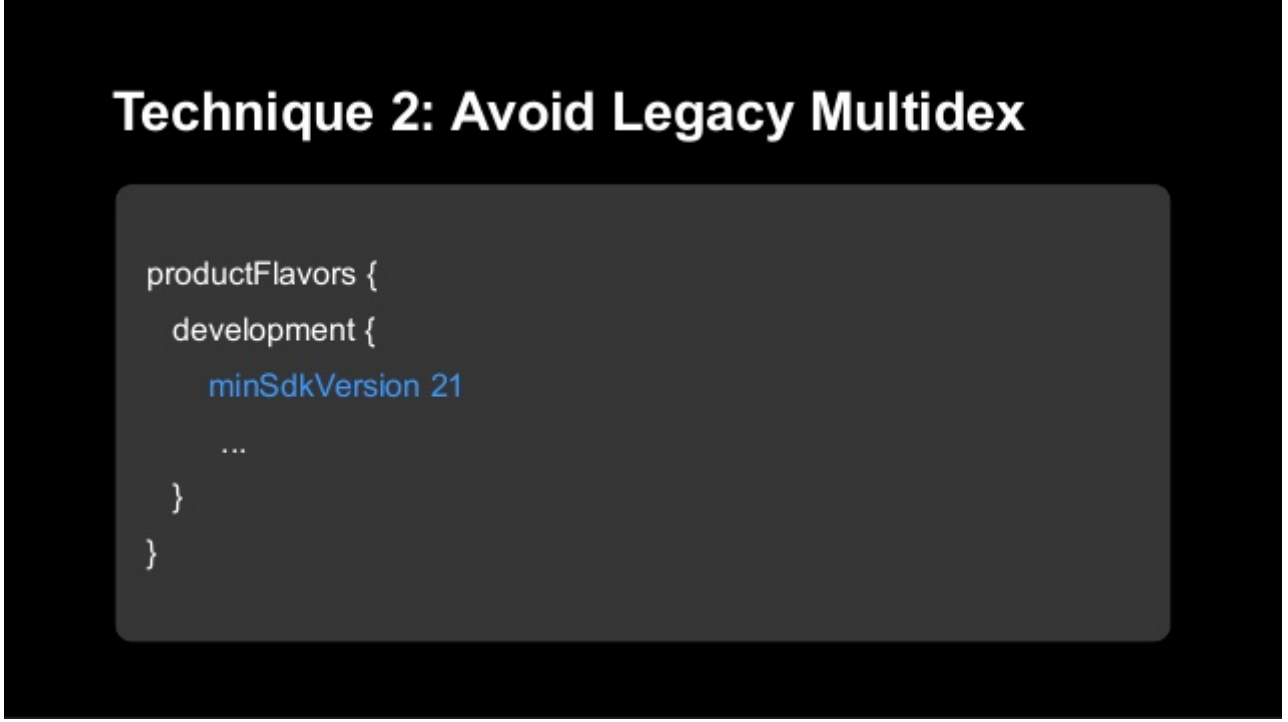
1. 최신 Android Gradle plugin 사용

최신 버전에는 많은 버그와 성능 관련 문제가 이미 해결되어 있습니다. 그러니 꼭 Android Studio를 업데이트한 뒤 사용하시기 바랍니다. 이렇게 업데이트를 하는 것만으로도 시간을 줄일 수 있습니다.



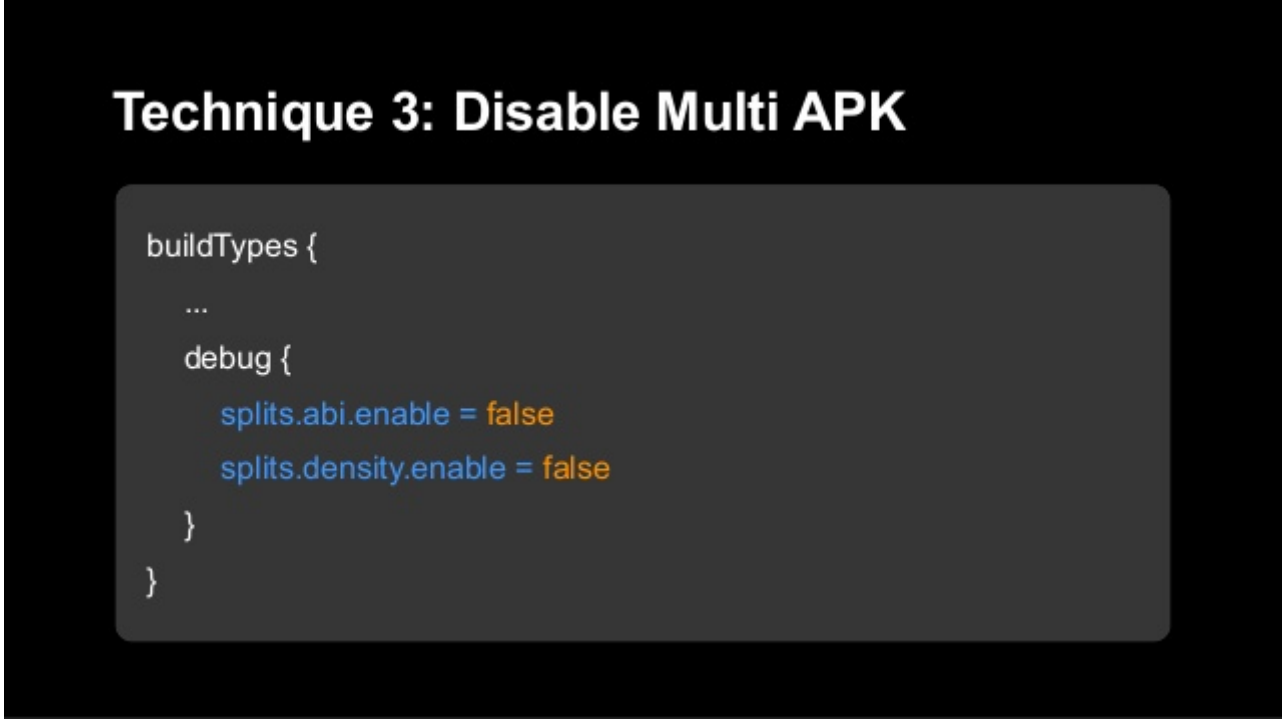
2. 레거시 multidex 사용 자제

앱이 메서드 수 제한(65,536개)을 초과하면 multidex를 사용해야 합니다. Android에서 multidex를 사용하면서 SDK의 최소 버전이 21 이하라면 레거시 multidex를 사용하는 셈이 됩니다. 레거시 버전을 사용하면 빌드 속도가 상당히 느려집니다. 이런 상황을 피하려면 먼저 `productFlavors` 를 정의해야 합니다. 아래 예시에서는 `development` 블럭을 사용했는데요. 이와 같이 SDK 버전 21 이상을 지정하면 15나 17 등의 레거시 버전 사용을 피할 수 있습니다. 다시 한번 강조하겠습니다. `minSdkVersion` 은 21 이상입니다.



3. 개발 시 여러 개의 APK 생성 설정 비활성화

앞서 말씀드린 내용과 관련이 있는데요. 화면 밀도나 ABI(Application Binary Interface)에 따라 여러 개의 APK를 빌드하는 작업에는 시간이 걸립니다. `debug` 블럭에 아래처럼 코드를 두 줄 추가하여 여러 개의 APK를 생성하는 설정을 비활성화하면 시간을 줄일 수 있습니다.

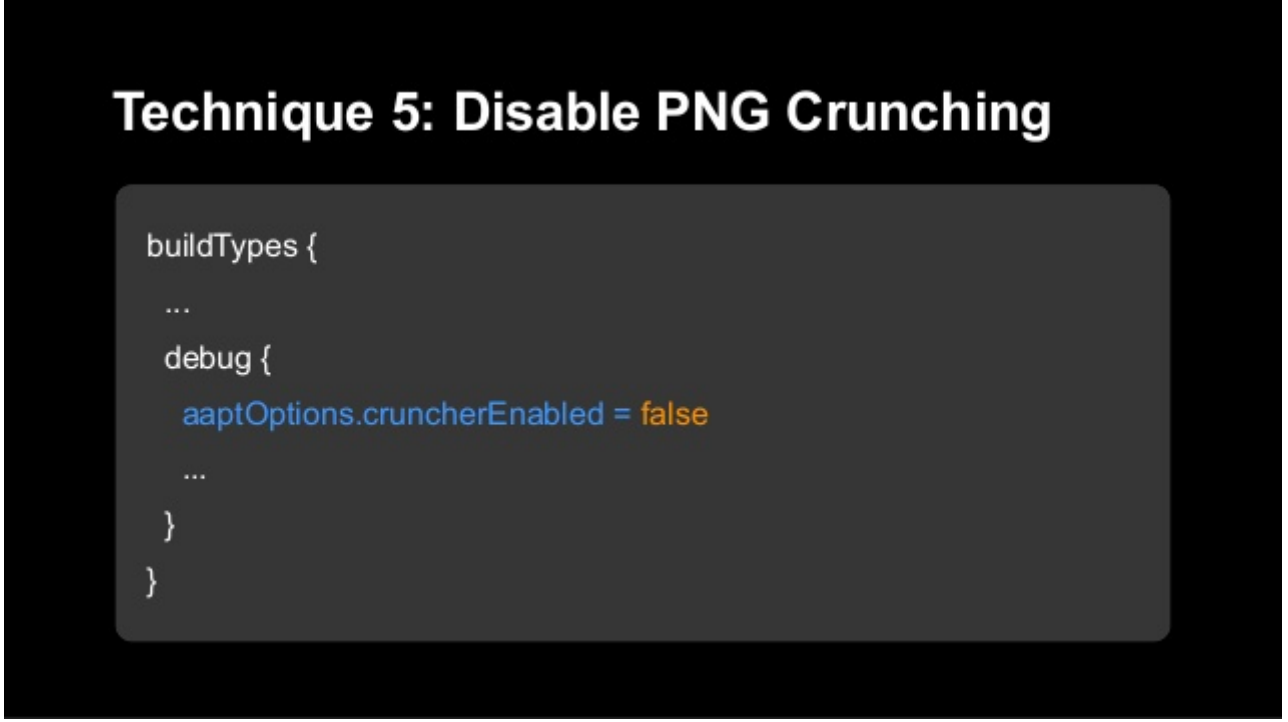


4. 개발 빌드에서 패키징 리소스 최소화

앱과 라이브러리에 사용할 수 있는 모든 언어와 화면 해상도는 빌드 시스템에 전부 기본적으로 포함되어 있습니다. 이 때문에 빌드하는 데 많은 시간이 걸리게 되는데요. 그런 것들이 다 필요하지는 않으니 `resConfig` 속성에서 언어와 화면을 한정해서 추가하여 그 리소스를 한 세트로 사용하세요.

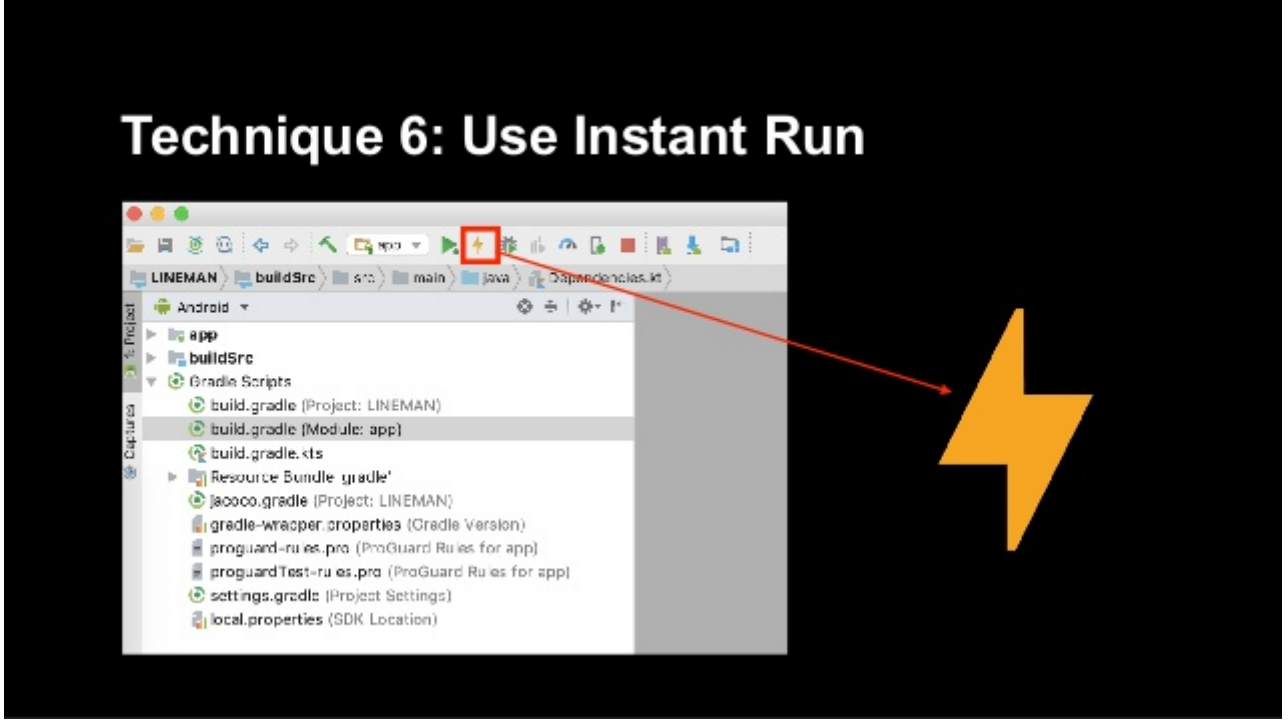
5. PNG 크런칭 비활성화

앱의 크기를 줄이기 위해 AAPT(Android Asset Packaging Tool)는 기본적으로 PNG를 크런치(crunch)하는데요. APK를 릴리스할 땐 좋지만 작업에 시간이 걸리기 때문에 개발 빌드에선 하지 않아도 됩니다. 개발할 땐 아래처럼 `false` 로 설정하세요.



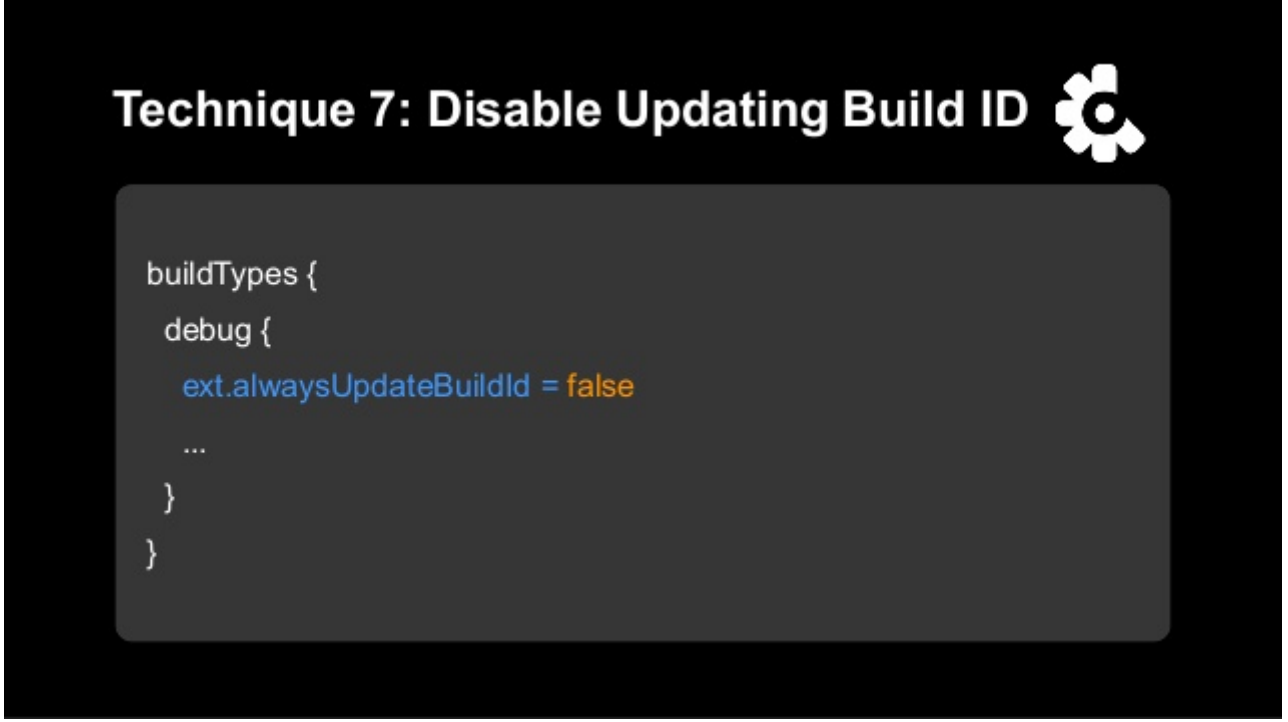
6. Instant Run 사용

Android Studio에는 기본적으로 초록색 화살표 모양의 **Run** 버튼이 있습니다. 이 버튼을 클릭하면 코드 스와핑이 일어나며 앱이 재시작됩니다. 매번 재시작해야 합니다. 이때 Instant Run을 사용하면 앱을 업데이트하는 데 걸리는 시간을 상당히 줄여줍니다(참고). 앱을 변경한 후 아래 이미지에서 보이는 번개 모양의 **Apply Changes** 버튼을 클릭하면 시스템은 라이브 프로세스로 즉시 변경사항을 푸시합니다. 이 경우 재시작할 필요가 없습니다.



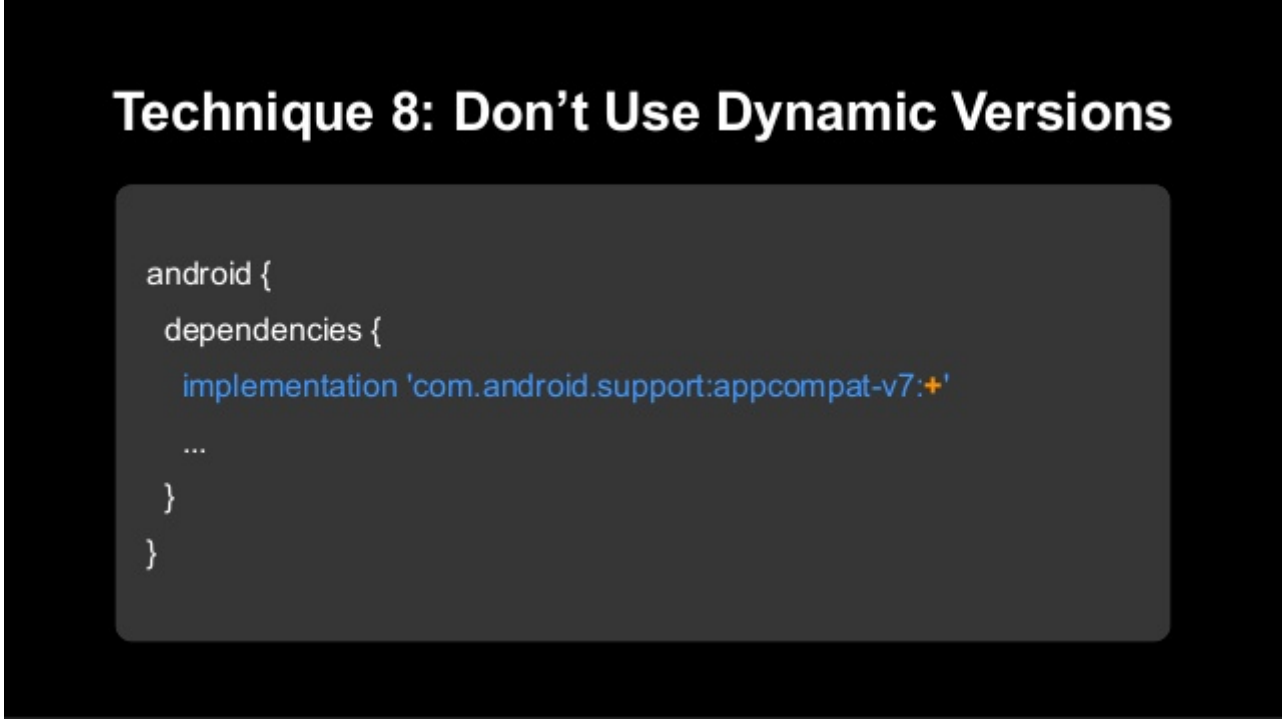
7. Crashlytics의 Build ID 업데이트 비활성화

Crashlytics는 세계 1위의 오류 보고 솔루션으로 많은 모바일 개발자들이 이미 사용하고 있고 LINE MAN에서도 역시 사용하고 있습니다. Crashlytics는 각 빌드별로 고유한 Build ID를 생성하는데요. 개발할 때 `debug` 블록에서 이 플래그를 `false` 로 설정하면 시간을 줄일 수 있습니다. 단, 이 설정은 개발에서만 사용하시기 바랍니다.



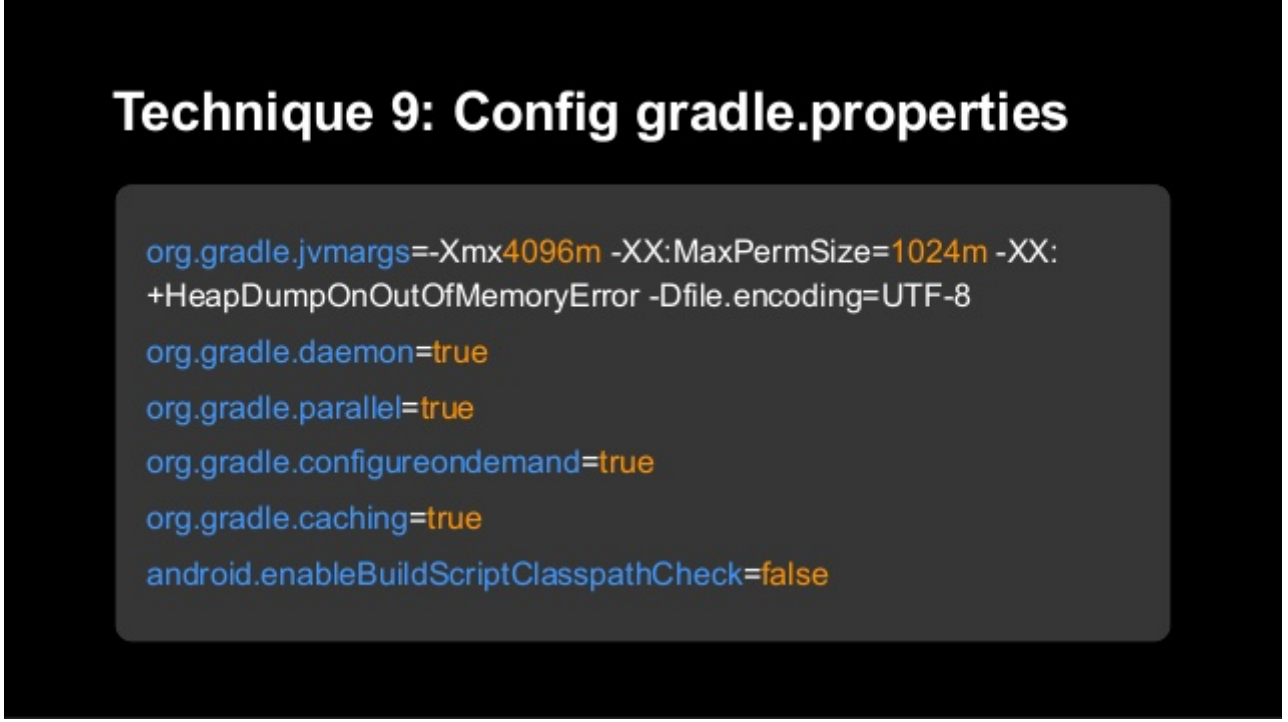
8. 종속성 동적 버전 사용 자제

Gradle은 맨 뒤에 '+'를 추가하면 자동으로 최신 버전으로 업데이트하게 되는데요. 이렇게 되면 예상치 못한 버전 업데이트가 발생해 개발하던 도중 크래시가 발생할 수 있고 Gradle이 업데이트를 확인하느라 빌드 속도가 느려질 수도 있습니다. 프로젝트에서 사용할 버전을 동적이 아닌 정적, 하드 코딩으로 지정하면 빌드 속도도 빨라지고 안전성도 높일 수 있어 추천합니다.



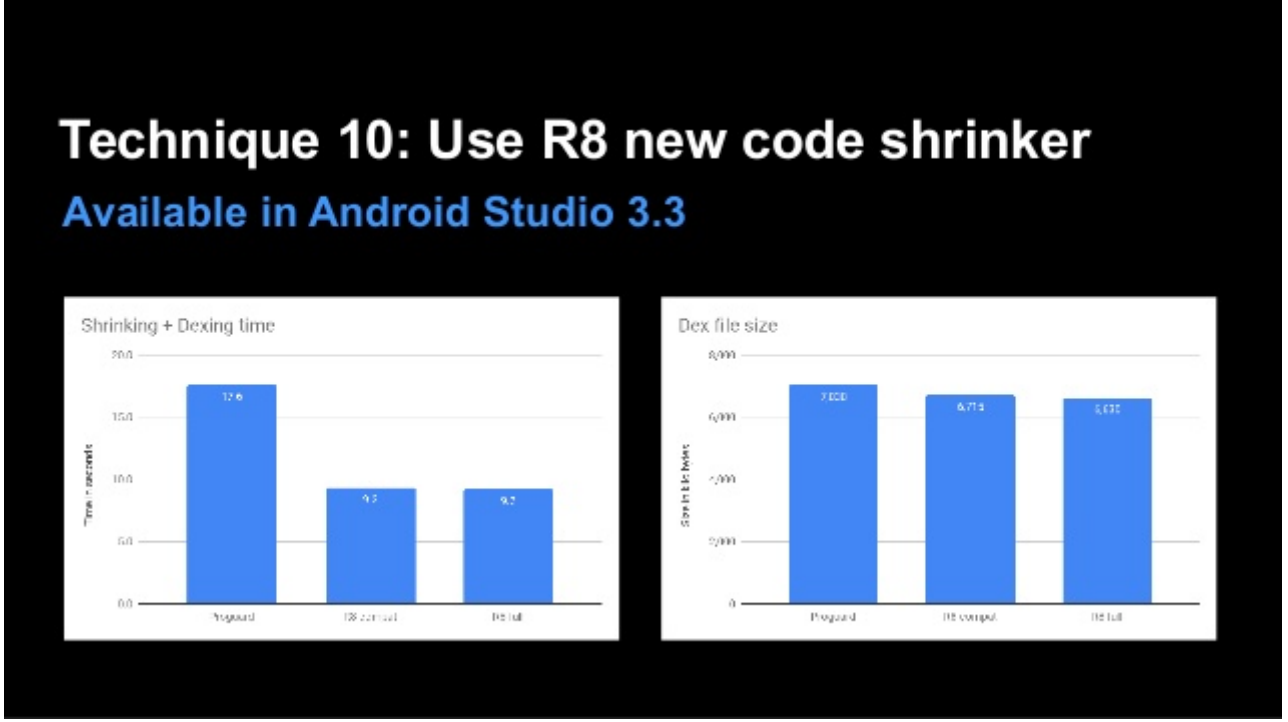
9. gradle.properties 설정

아래는 제 Android 개발 경험을 토대로 설정한 `gradle.properties` 인데요. 이렇게 설정하여 빌드 속도를 높일 수 있었습니다. 여러분도 참고하시면 좋을 것 같습니다.



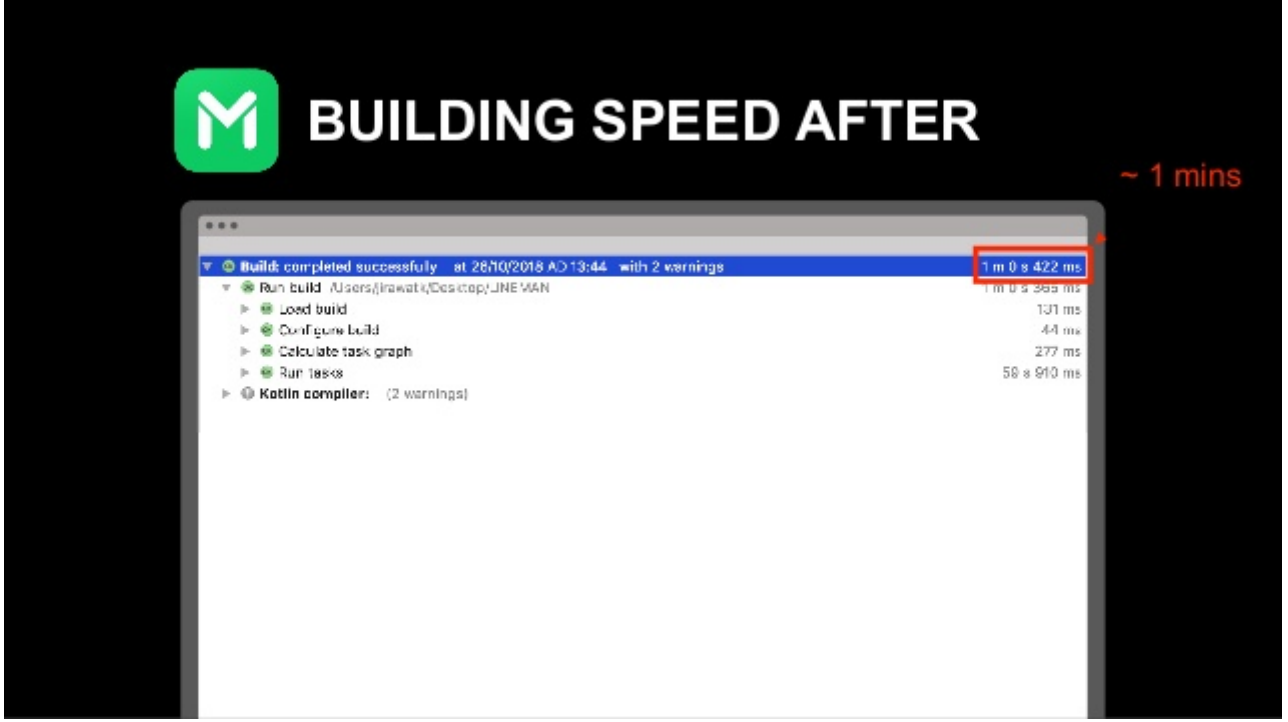
10. R8 사용

R8은 코드를 축소하기 위해 사용되며 Android Studio 3.3 베타 버전부터 사용할 수 있습니다(참고). 기존에 코드를 축소하기 위해 사용했던 Proguard 보다 더 빠른 속도로 더 작게 줄여주니 꼭 사용하시기 바랍니다.



적용 후

지금까지 소개한 기술을 모두 적용하고 빌드해 보니 전체 빌드가 약 1분 만에 실행되었습니다! 이제 3배 빠른 속도로 개발할 수 있게 되었습니다.



마치며

Android 앱의 크기를 줄이는 10가지 방법과 이번 글에서 소개드린 Android 앱 빌드 속도를 높일 수 있는 방법을 잘 활용하여 생산성을 올려서 더 작은 앱을, 더 빨리 개발하기를 바랍니다. 이 글과 함께 코딩을 즐겨보세요!

Related Post

Android 앱의 크기를 줄이는 10가지 방법
TRACKIT에서 딥링크를 사용하는 방법
AIR GO에 안드로이드 9 APK 서명 scheme v3 적용하기...
Kotlin으로 서버사이드 개발과 Clova Skill Award 도전!...