안드로이드를 위한 Gradle 레시피

# 추천사

우리에게 필요한 책이 나왔습니다. 한창 Head First Android Development(한빛미디어)을 집필하던 중에 구글은 공식 IDE를 새롭게 발표하였습니다. 당시 대다수의 개발자들은 이클립스에 Android Development Toolkit(ADT)를 설치하였습니다. 하지만 이제 구글은 안드로이드 개발자에게 제트 브레인사의 IntelliJ IDEA 기반인 안드로이드 스튜디오로 전환하기를 지속적으로 요청하고 있습니다.

우리는 이 같은 일들에 익숙합니다. 대부분의 기술 저자가 그렇지요. 소프트웨어 스타트업은 어떤 새롭고 빛나는 기술에 만족하지 않고 훨씬 새롭고 훨씬 더 빛나는 기술로 갈아타는 것에 익숙합니다. 이런 일은 항상 일어납니다. 샘플 코드를 다시 작성하고, 이미지를 업데이트하고 더 이상 사용하지 않는 기능을 제거하며 유용한 부분은 새롭게 받아들입니다. 이클립스에서 안드로이드 스튜디오로 전환하는 것이 남다른 이유는 새로운 IDE가 그 속에 강력한 엔진을 내장하고 있기 때문입니다.

안드로이드 스튜디오는 코드를 빌드하고 패키징하고 배포하는데 Gradle을 사용합니다. 이름은 알고 있었지만 우리 중 누구도 Gradle을 그전에 사용해보지는 않았습니다. 일종의 Maven 같이 생겼지만 기나긴 XML 설정 파일은 존재하지 않습니다. 대신 견고하고 간결한 스크립트 언어인 Groovy를 활용합니다.

우리는 책의 모든 스크린샷을 교체하고 이미 다 쓰여진 일곱 개가 넘는 챕터의 본문을 수정하고 집필의 남은 부분을 진행하였습니다. Gradle로 애플리케이션을 작성하는 것이 미묘하지만 상당 부분 기존의 방법과 다르다는 것을 곧 알 수 있었습니다. IDE에서만 할 수 있었던 대부분의 기능을 이제 명령창에서 실행할 수 있습니다. 이것은 빌드 파이프라인을 자동화할 수 있다는 것을 의미합니다. 몇 번의 키만 누르면 라이브러리의 다른 버전으로 교체하고 제품 특성(build flavors)도 적용할 수 있습니다. 모든 것이 코드로 이루어져 있기 때문에 소스 코드를 작성하듯이 빌드 스크립트를 작성할 수 있습니다.

Gradle을 배우는 것은 모든 안드로이드 개발자에게 매우 중요합니다. Java를 알거나 액티비티의 라이프사이클을 이해하는 것과 마찬가지입니다. 그러나 시행착오를 통해서만 Gradle을 배우게 되면 때로는 너무나 고통스럽습니다. 그래서 이 책의 존재가치가 생깁니다. 이 책에는 흔히 발생할 수 있는 빌드 문제에 관하여 풍부하고 유용한 팁들을 제공해줍니다. 테스트 시스템을 설정하거나 APK를 서명하여 빌드하거나 빌드 파이프라인(build pipelines)의 성능을 높이는데 도움을 받을 수 있습니다. 저자인 켄의 살아있는 필체와 실질적인 예제는 이 책을 두고 두고 다시 찾게 만듭니다. 이 책을 통하여 켄은 본인이 Groovy의 전문가일뿐만 아니라 Gradle에도 일가견이 있음을 여지없이 보여주었습니다.

Dawn and David Griffiths

Head First Android Development의 저자

2016년 4월 20일

# 서문

이 책에 관하여

이 책은 안드로이드 프로젝트에서 Gradle 빌드 시스템을 다루는 다양한 레시피(recipes)을 포함하고 있습니다. Gradle은 Java 세상에서 앱을 빌드하는 가장 인기있는 도구이며 C++ 같은 다른 언어로 그 영역을 확대하고 있습니다. 구글의 안드로이드 팀은 2013년 봄에 Gradle을 우선적인 빌드 시스템으로 선정하였고 이후 그 사용이 꾸준하게 성장하고 있습니다.

Gradle이 Groovy 생태계에서 탄생하였기 때문에 많은 안드로이드 개발자는 그것에 익숙하지 않을 수 있습니다. 하지만 Groovy는 기존 자바 개발자들이 쉽게 배울 수 있습니다. 이 책의 목적은 Gradle을 활용하여 안드로이드 앱에 관한 빌드 작업을 할 수 있도록 돕는데 있습니다.

모든 예제 코드는 안드로이드 SDK 버전 23 기반이며 에뮬레이터는 마시멜로(Android 6) 혹은 롤리팝(Android 5점대)입니다. 안드로이드 스튜디오의 버전은 2.0 혹은 2.1(베타)이며 포함되어 있는 Gradle은 2.10 혹은 이후 버전입니다.

전제 조건

안드로이드를 위한 Gradle 플러그인을 사용하려면 적어도 약간의 Java, Groovy, Gradle 과 안드로이드에 관한 지식을 필요로 합니다. 위 내용은 책의 전반에서 나오기 때문에 각각에 대해서는 자세하게 다루지 못할 수도 있습니다.

이 책의 내용은 주로 안드로이드 개발에 익숙한 개발자를 대상으로 합니다. 안드로이드 개발의 지식은 거의 제공하지 않으며 모든 예제의 전체 소스 코드는 이 책의 GitHub 저장소에서 확인할 수 있습니다. 안드로이드를 이해하고 있다함은 Java를 이해하고 있다는 것을 의미합니다. 그래서 그에 대한 내용도 기본적인 것은 알고 있어야 합니다.

Groovy 혹은 Gradle에 관해서는 전혀 모른다고 가정하였습니다. 하지만 부록 A에는 Groovy 문법과 활용에 관하여 간단하게 요약정리하였습니다. Groovy 개념은 책의 다양한 레시피를 거치면서 다시 나옵니다. 부록 B는 Gradle 의 기본적인 정보입니다. 하지만 본문에서는 레시피별로 Gradle을 주제별로 자세하게 다루고 있습니다.

이러한 한계점을 제외하면 이 책은 가능한 자체적으로 충분한 내용을 담으려고 디자인하였습니다. 필요한 경우 외부 레퍼런스(특히 문서 링크)가 제공됩니다.

이 책에는 안드로이드 스튜디오를 광범위하게 활용합니다. 안드로이드 스튜디오는 안드로이드 개발을 위한 공식 IDE입니다. 안드로이드 스튜디오는 Gradle을 위한 뷰와 도구들을 제공하고 자세한 내용은 각 레시피에서 소개하였습니다. 이 책이 안드로이드 스튜디오를 위한 책은 아니기 때문에 꼭 필요한 내용만 소개되어 있지만 이를 통해 IDE에 관하여 좀 더 익숙해지는 것도 좋을 것입니다.

이 책에서 사용된 용례

이 책에서는 다음과 같은 글씨체를 사용한다.

이탤릭

새로운 용어, URL, 이메일 주소, 파일 이름과 확장자

고정폭 글꼴

소스코드를 표시함. 본문의 단락안에서는 프로그램의 요소를 표시할 때 사용함. 예를 들어 변수, 함수 이름, 데이터베이스, 데이터 타입, 환경변수, 문장 과 예약어

고정폭 볼드 체

사용자가 글자 그대로 입력해야 하는 명령이나 그 외 텍스트

고정폭 이탤릭체

사용자가 입력한 값이나 문맥에 의해 결정되는 값으로 테스트의 내용이 교체되어야 하는 경우

<도마뱀> 팁이나 제안 사항을 나타낸다

<까마귀> 일반적인 메모

<전갈> 경고나 주의사항

예제 코드 활용에 관하여

이 책에서 제공되는 예제 코드나 실습과 같은 보충자료는 아래의 URL에서 다운로드 받을 수 있습니다. <https://github>

이 책은 당신의 업무를 돕기 위해 존재합니다. 이 책에서 제공되는 예제 코드는 일반적으로 당신의 프로그램이나 문서에서 사용할 수 있습니다. 코드의 상당 부분을 복사하는 경우가 아니라면 별도의 허가를 받지 않아도 됩니다. 예를 들어 이 책에서 코드의 몇 몇 부분을 사용하는 프로그램이라면 허가를 사전에 받지 않아도 됩니다. 오라일리 책에서 제공되는 예제들을 CD-ROM으로 판매하거나 배포하는 경우에는 반드시 허가를 받아야 합니다. 이 책에서 인용하였음을 표시하고 예제 코드를 인용하는 경우에는 허가를 받지 않아도 됩니다. 단, 예제 코드의 상당한 많은 부분을 당신의 프로젝트 문서에서 사용하는 경우에는 허가를 받아야 합니다.

출처 표기(attribution)을 해주시면 감사하겠지만 필수적인 것은 아닙니다. 출처 표기는 보통 아래와 같은 양식으로 제목, 저자, 출판사 그리고 ISBN 정보를 포함합니다. 예를 들어 “…..”

만약 정당한 사용이 아니거나 위에서 언급한 내용 외의 경우에는 언제든지 [permissons@oreilly.com](mailto:permissons@oreilly.com)으로 문의를 부탁드립니다.

Safari Books Online

<이부분은 제가 굳이 번역할 필요가 없이 다른 책에서 그대로 가져오면 될 것 같습니다>

여러분의 의견을 받습니다.

이 책에 관한 조언과 질문사항은 아래로 연락하시기 바랍니다.

<이 부분도 오라일리의 내용이 아니라 한빛 미디어의 내용이면 충분할 것 같습니다>

**감사의 글**

저자는 Gradle사의 Hans Dockter, Luke Daley, Rooz Mohazabbi 와 Cedric Champeau 등 여러 멤버들이 보여준 자애로운 도움과 원조에 감사합니다. 그들의 도움은 Gradle이 기술적으로나 그 회사 모두 전도 유망함을 보여주는 이유입니다.

나는 또한 구글의 안드로이드 스튜디오 팀과 안드로이드 플러그인 프로젝트의 리더인 Xavier Ducrohet에게도 감사의 마음을 전합니다. 그의 노력으로 IDE와 안드로이드 플러그인을 사용하는데 즐거웠습니다. 나는 또한 그와 그의 팀이 온라인 문서를 최신으로 충분히 업데이트하지 않음에도 감사합니다. 덕분에 이 책이 나올 수 있었습니다. <각주: 전적으로 농담입니다. 하지만 지금이라도 최신으로 업데이트 해준다면 누구도 마다하지 않을 것입니다>

No Fluff, Just Stuff 컨퍼런스의 정회원으로서 Gradle과 Android 관련 주제를 수년간 여러 번 발표할 있게 배려해준 Jay Zimmerman도 감사합니다. No Fluff 연사 커뮤니티를 통해서 좋은 친구들을 만날 수 있었습니다. 특히 Nate Schutta, Raju Gandhi, Venkat Subramaniam, Neal Ford, Dan Hinojosa, Brian Sletten, Michael Carducci 와 Craig Walls은 꼭 언급하고 싶지만 나머지 다른 이름도 십수명은 충분히 언급할 수 있습니다. 다음 No Fluff 컨퍼런스에서는 이 글을 읽고 여기에 언급되지 않은 사람에 대해서도 들을 것입니다.

이 시리즈의 이전 저자인 Matthew McCullough 와 Tim Berglund에게도 감사합니다. 두 분다 친절하고 도움을 많이 받았습니다. 이 시리즈에 이 책이 포함되어 영광입니다. (어떤 시리즈일까요??)

이 책의 리뷰어들은 책의 품질을 향상시키는데 큰 도움이 되었습니다. 특히 Andrew Reitz 와 James Harmon에게 감사합니다. 이책의 기술적인 통찰 뿐만 아니라 가독성을 높이는데 큰 도움이 되었습니다.

오라일리 출판사의 편집자인 Meghan Blanchette 와 Brian Foster도 감사합니다. Meghan은 이 책을 시작하는데 핵심적인 역할을 하였고 초창기에 편집을 담당하였습니다. 이후 Brian이 맡아 나머지 출판까지 이끌어 주었습니다. 오라일리 출판사의 다른 담당자들에게도 감사드립니다. 막연하게만 알고 있던 사실들을 구체화하여 최종적으로 출판하는데 큰 도움이 되었습니다.

비록 표면상으로는 경쟁도서인 Gradle for Android(Packt Publishing) 의 저자인 Kevin Pelgrims 에게도 감사합니다. 그의 책은 훌륭하고 나에게도 많은 가르침을 주었습니다. 나의 책이 레시피 기반으로 다른 관점을 취하고 있고 새롭고 최신의 내용을 다루고 있음에는 변함이 없지만 만약 그가 나와 같은 시도를 했다면 두 책다 모두 권할 수 밖에 없을 것입니다.

무엇보다도 부인인 Ginger와 내 아들 Xander에게도 감사합니다. 지난번 집필을 마치고 바로 이번책을 집필하게 되어 미안한 마음을 전합니다. 아마도 새로운 책을 시작할 때까지는 여유있는 시간을 함께 보내기를 약속합니다.

이 책을 읽어주셔서 감사합니다. 이 책이 당신에게 유용했으면 좋겠습니다. 내용의 오류나 빠진내용들은 모두 저의 책임입니다.

# 안드로이드를 위한 Gradle 기초

안드로이드 애플리케이션은 오픈소스인 Gradle 빌드 시스템으로 빌드합니다. Gradle은 쉽게 커스터마이즈할 수 있는 최신 API를 제공하고 자바 진영에서 널리 사용됩니다. 안드로이드를 위한 Gradle 플러그인은 안드로이드 애플리케이션에 특화된 많은 기능들을 제공합니다. 예를 들면 빌드 타입, 제품 특성(product flavors), 앱 서명에 관한 설정, 라이브러리 프로젝트 등이 있습니다.

이 책의 레시피들은 안드로이드 프로젝트에서 활용할 수 있는 다양한 Gradle의 기능들을 소개합니다. 안드로이드 스튜디오 IDE는 Gradle을 내부적으로 사용하기 때문에 몇몇 특별 레시피는 특히 그것에 대해 다룹니다.

바라건데 이 책의 레시피가 당신이 만드는 안드로이드 애플리케이션을 설정하고 빌드하는데 큰 도움이 되었으면 좋겠습니다.

레시피 1.1 안드로이드를 위한 Gradle 파일들

문제

새로운 안드로이드 애플리케이션을 만들었을 때 생성되는 Gradle 파일이 무엇인지 알고 싶다

해결

안드로이드 스튜디오에서 새로운 안드로이드 프로젝트를 만들고 새롭게 생성되는 settings.gradle, build.gradle, app/build.gradle 파일에 대해 알아보자

논의

안드로이드 스튜디오는 안드로이드 프로젝트를 위한 공식 IDE입니다. 안드로이드 스튜디오에서 새로운 안드로이드 프로젝트를 만들려면 프로젝트 시작 마법사를 실행합니다. [그림 1-1]

[그림 1-1] 안드로이드 스튜디오 퀵 스타트

마법사가 시작되면 프로젝트 이름과 도메인을 입력해야 합니다. [그림 1-2]와 같이 프로젝트 이름은 My Android App으로 도메인은 oreilly.com 으로 입력합니다. 그 다음 “Phone and Tablet” 옵션을 선택하고 blank activity를 추가합니다. 액티비티 이름은 기본값인 MainActivity로 합니다.

<까마귀> 액티비티의 이름과 타입은 Gradle 빌드 파일들에 영향을 주지 않습니다.

[그림 1-3]와 같이 프로젝트가 완성되었습니다. 안드로이드 뷰가 표시됩니다. Gradle 파일을 강조하였습니다.

[그림 1-2] 새 프로젝트 생성

[그림 1-3] 프로젝트 구조 (안드로이드 뷰)

[그림 1-4]는 프로젝트 구조를 프로젝트 뷰로 표시하고 있습니다.

[그림 1-4] 프로젝트 구조(프로젝트 뷰)

안드로이드 프로젝트는 멀티 프로젝트 구조입니다. (편집자주: 기존 이클립스의 경우 싱글프로젝트구조로 서로 다릅니다) settings.gradle 파일에는 포함되는 하위 프로젝트(이하 모듈)이 담겨있습니다. [예제 1-1]에는 기본적으로 생성되는 값이 나와 있습니다.

[예제 1-1] settings.gradle 파일

include 문장에는 app 모듈이 포함되어 있습니다. 만약 안드로이드 라이브러리 프로젝트를 추가한다면 settings.gradle 파일에 추가됩니다.

[예제 1-2]는 최상위 build.gradle 파일 (이하 프로젝트 build.gradle 파일)이 나와 있습니다.

[예제 1-2] 프로젝트 build.gradle 파일

Gradle 배포판에는 안드로이드 기능이 포함되어 있지 않기 때문에 구글에서는 안드로이드 플러그인을 제공하여 손쉽게 필요한 설정할 수 있습니다. 프로젝트 build.gradle 파일의 buildscript 블록에는 Gradle 배포판을 어디서 다운받을지 지정합니다.

기본값으로 플러그인은 jcenter에서 다운받습니다. 이는 Bintray사의 JCenter 저장소를 의미합니다. 다른 저장소도 지원됩니다. 특히 mavenCentral()은 기본 Maven 저장소를 의미합니다. Jcenter는 안전한 HTTPS 연결을 통해 CDN으로 제공됩니다. 속도도 상당히 빠릅니다. (각주: CDN은 Content Delivery Network의 약자로 콘텐츠를 효율적으로 전달하기 위해 여러 노드를 가진 네트워크에 데이터를 저장하여 제공하는 시스템을 말한다 <위키백과: <https://ko.wikipedia.org/wiki/%EC%BD%98%ED%85%90%EC%B8%A0_%EC%A0%84%EC%86%A1_%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC> )

allprojects 블록은 최상위 프로젝트와 하위 모듈이 공통적으로 활용할 내용을 지정합니다. jcenter()로 지정하였는데 이는 외부 라이브러리 참조시 우선적으로 JCenter를 참조하게 됩니다.

Gradle은 내장된 태스크 뿐만 아니라 사용자 정의 태스크도 만들 수 있습니다. 각 태스크는 단방향 비순환 그래프(DAG, directed acyclic graph)에 저장됩니다. clean 태스크는 프로젝트 build.gradle에 정의되어 있습니다. type: Delete 부분은 새로 정의한 태스크가 내장 태스크인 Delete를 상속했다는 의미입니다. clean 태스크는 루트 프로젝트 및 하위 모듈의 build 폴더를 제거합니다.

[예제 1-3]은 app 모듈의 build.gradle 파일을 보여줍니다.

[예제 1-3] app 모듈의 build.gradle 파일

apply 문장은 안드로이드 플러그인을 지정합니다. Gradle 플러그인은 안드로이드 환경을 위한 도메인 특화 언어(DSL)를 구현하였습니다. 자세한 내용은 레시피 1.2에서 다룹니다.

dependencies 블록은 세 줄로 이루어져 있습니다. 첫재 줄은 fileTree 의존성을 정의하는데 libs 폴더안에 있는 모든 jar 파일을 참조합니다. 두번째 줄은 JUnit 4.12을 다운로드하고 testCompile 단계에서 참조합니다. 이것은 /src/androidTest/java 폴더 혹은 /src/test/java/ 폴더에서 참조할 수 있습니다.

세번째 줄은 안드로이드 서포트 라이브러리인 appcompat-v7.jar 을 클래스패스에 추가하였습니다. 버전은 23.3.0 이고 v7은 안드로이드 버전 7부터 하위호환성을 제공한다는 의미입니다. 라이브러리 버전이 7이라는 의미는 아닙니다. 서포트 라이브러리는 compile 단계에 추가되어 프로젝트 전체에서 사용할 수 있습니다.

그외 (See Also)

레시피 6.2에는 필요한 문서 사이트 목록이 제공됩니다. 외부 라이브러리 참조에 관해서는 레시피 1.5를 참고하세요. 저장소에 관해서는 레시피 1.7을 참고하세요.

레시피 1.2 SDK 버전과 그외 기본값 설정하기

문제

최소 SDK 버전과 목적 SDK 버전을 설정하고 그외 속성값을 지정하고 싶다.

해결책

모듈 build.gradle에 있는 android 블록을 수정하라

논의

프로젝트 build.gradle 파일의 buildscript 블록은 안드로이드 플러그인 버전을 지정하고 모듈 build.gradle 파일의 android 블록은 [그림 1-4]와 같이 세부 내용을 정의한다.

[예제 1-4] 모듈 build.gradle 파일의 android 블록

일반적인 자바 프로젝트에서는 java 플러그인 사용해야 한다. 그러나 안드로이드 프로젝트는 대신 com.android.application 플러그인을 사용해야 한다.

<전갈> 자바 플러그인을 적용하면 빌드 오류가 발생한다.

android 블록의 compileSdkVersion 항목은 컴파일 SDK 버전을 buildToolsVersion 항목은 빌드툴 버전을 정의한다. 두 값 모두 가능한 최신 버전을 기입하는 것이 좋다. 기본적으로 하위 호환성이 지원하고 알려진 버그를 수정하였기 때문이다. (역자주: 항상 최신 버전을 그대로 따라갈 필요는 없다. 예를 들어 빌드툴 23.0.1의 경우 화면 레이아웃이 깨지는 버그가 발생하여 바로 23.0.2로 업데이트된 사례가 있다. 해당 버전은 더 이상 다운로드받을 수 없다)

android 블록안에 있는 defaultConfig 블록에는 몇가지 속성을 지정할 수 있다.

applicationId

애플리케이션의 패키지 이름을 정의한다. 구글 플레이 스토어에서 유일한 이름으로 구별되어야 한다. 한번 마켓에 올라가면 이 값은 더이상 변경되어서는 안된다. 값을 변경하게 되면 전혀 다른 앱이 되어 기존 사용자가 동일한 이름으로 최신 버전을 검색할 수 없다. Gradle로 넘어가기 전에는 AndroidManifest.xml 파일의 package 속성에 정의하였다. 이제 두 값은 서로 달라도 된다.

minSdkVersion

지원하는 최소 SDK 버전. 이 값보다 낮은 기기에서는 해당 앱을 검색할 수 없다.

targetSdkVersion

앱이 의도하는 목적 SDK 버전. 안드로이드 스튜디오에서는 이 값이 안드로이드 최신 버전보다 낮으면 경고를 표시한다. 원하는 버전을 지정하면 된다.

versionCode

앱의 버전을 나타내는 정수값. 업그레이드의 기준이 된다.

versionName

사용자에게 배포되는 앱의 버전을 나타내는 문자열. 보통 <major>.<minor>.<version>의 형식을 따른다. 예를 들면 5.0.1과 같다.

Gradle로 넘어가기 전에는 minSdkVersion과 buildToolsVersion 값은 AndroidManifest.xml 파일의 <uses-sdk> 태그의 속성으로 정의하였다. 이러한 방식은 이제 더 이상 지원(deprecated)하지 않으며 Gradle의 빌드 속성값으로 대체된다.

compileOptions 블록은 이 애플리케이션이 JDK 1.7을 사용하고 있음을 알려준다.

[그림 1-5]는 프로젝트 구조를 위의 속성값을 시각적으로 보여준다.

defaultConfig 블록은 [그림 1-6]에서 Flavors 탭에 표시되며 자세한 내용은 DSL 문서를 참고하면 된다.

[그림 1-5] 안드로이드 스튜디오의 Project Structure

[그림 1-6] defaultConfig 블록의 내용

그 외

android 블록의 나머지 내용(buildTypes, productFlavors 등)은 레시피 3.1, 3.2와 3.4를 참고하라. 문서의 링크는 레시피 6.2에서 제공된다.

================================================================