Gradle

Gradle

Build 자동화 툴

⇒ 프로그램의 컴파일, 테스트, 배포 문서화 등을 포함하는 넓은 개념의 Build

Groovy 기반 DSL(Domain-Specific Lang.)

⇒ 특정한 용도에 맞게 Groovy기반으로 각색한 언어

https://webfirewood.tistory.com/129

Build.gradle

- **plugins** vs **apply plugin(+buildscript)**
- sourceCompatibility
- implementation vs api vs compile
-

plugins가 상대적으로 최근에 나온 방법

5.6v 이후에는 buildscript + apply plugin 으로 적용 가능하던 대부분의 기능을 지원

https://plugins.gradle.org/search?term=org.springframework.boot 에서 공식 plugin 확인 가능

https://stackoverflow.com/questions/32352816/what-the-difference-in-applying-gradle-plugin

Build.gradle

- **plugins** vs **apply plugin(+buildscript)**
- sourceCompatibility
- implementation vs api vs compile
-

Subproject 에서 사용 방법

```
plugins {
    id 'com.example.hello' version '1.0.0' apply false
    id 'com.example.goodbye' version '1.0.0' apply false
}
subprojects {
    if (name.startsWith('hello')) {
        apply plugin: 'com.example.hello'
    }
}
```

https://doughman.tistory.com/19

Build.gradle

- **plugins** vs **apply plugin(+buildscript)**
- sourceCompatibility
- implementation vs api vs compile
-

plugins 고유 저장소 지정 방법

Setting.gradle

```
pluginManagement {
    plugins { }
    resolutionsStrategy { }
    repositories { }
}
rootProject.name = ‘plugin-management’
```

https://docs.gradle.org/current/userguide/plugins.html

Build.gradle

- plugins vs apply plugin(+buildscript)
- **sourceCompatibility**
- implementation vs api vs compile

- Java version 입력

targetCompatibility(Legacy Java 1.5 version 이하에 사용)

=> Version대로 컴파일한다는 의미가 아닌 해당 버전에 사용 가능한 라이브러리를 사용(라이브러리 빌드 실패)

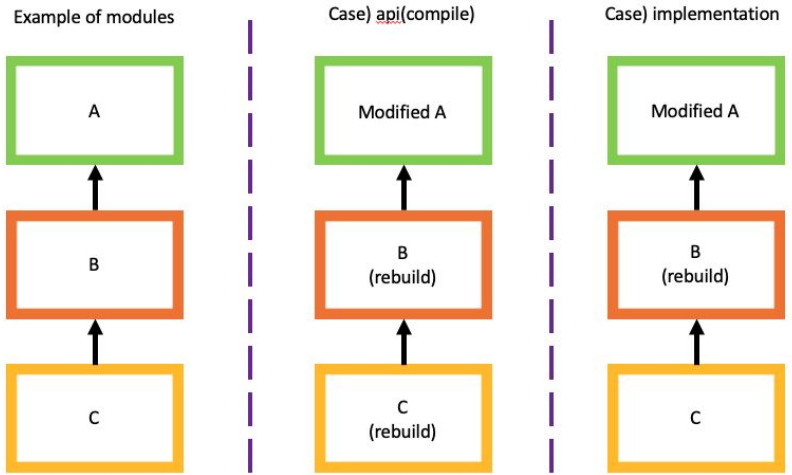
https://stackoverflow.com/questions/16654951/gradle-sourcecompatibility-vs-targetcompatibility

Build.gradle

- plugins vs apply plugin(+buildscript)
- sourceCompatibility
- **implementation** vs **api** vs **compile**

Library들의 노출 정도를 의미

compile(api) > implementation 순



https://jongmin92.github.io/2019/05/09/Gradle/gradle-api-vs-implementation/
https://bluayer.com/13

감사합니다.