

# Android

## 003 Lint

# 1. Lint 란?

## Android Code Scanning Tool

가. app을 실행하거나 테스트 케이스를 작성하지 않고도, 코드의 **구조적 품질 문제**를 쉽게 파악해서 **교정**하는 것을 도와주는 코드 스캐닝 도구

나. 각각의 문제는, 설명 메시지와 심각도 수준을 보고하기 때문에, 조치해야 할 심각한 **개선사항의 우선 순위**를 정할 수 있다

다. 프로젝트와 관련이 없는 문제를 무시하거나, 또는 심각도 수준을 높이도록 **심각도 수준을 구성**할 수 있다.

라. **명령/줄 인터페이스**를 갖고 있어서 이 자동화된 테스트 프로세스에 **쉽게 통합**시킬 수 있다

마. **정확성, 보안, 성능, 사용성, 접근성, 국제화 측면에서 잠재적 버그 및 최적화** 개선을 위하여 안드로이드 프로젝트 소스 파일을 체크한다.

## 2. Lint 검사항목

기본적인 검사 항목 카테고리

Correctness

Usability

Security

Accessibility

Performance

I18n

각 검사는 고유 ID가 있으며, 각 검사는 아래와 같은 요소로 구성된다

( Lint check list : <http://tools.android.com/tips/lint-checks> )

ID

항목요약

우선순위

심각도

카테고리

### TooDeepLayout

**Summary:** Layout hierarchy is too deep

**Priority:** 1 / 10

**Severity:** Warning

**Category:** Performance

Layouts with too much nesting is bad for perform..

### 3. Lint Severity (심각도)

검사 후 검출된 항목은 심각도에 따라 다음 세 단계로 나뉜다

#### Error

적절한 조치를 취하지 않으면 심각한 문제를 발생시킬 수 있는 항목  
여기에 해당하는 항목이 하나라도 있을 경우 애플리케이션 바이너리 생성시 오류 발생

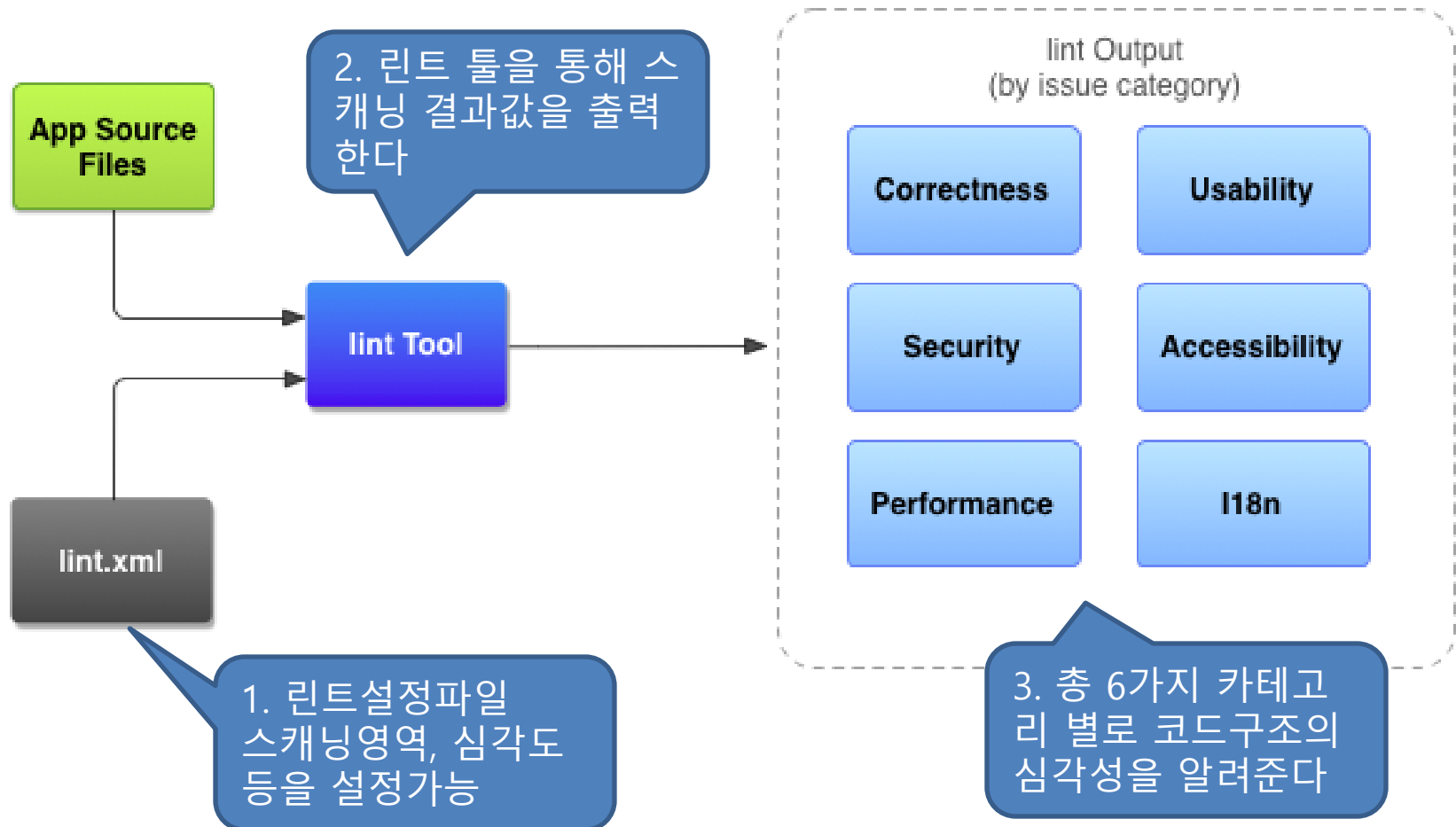
#### Warning

시스템에 심각한 영향은 미치지 않으나 예상치 못한 동작의 원인이 되는 항목

#### Informational

심각도가 매우 낮거나 단순 참고용으로 볼 항목

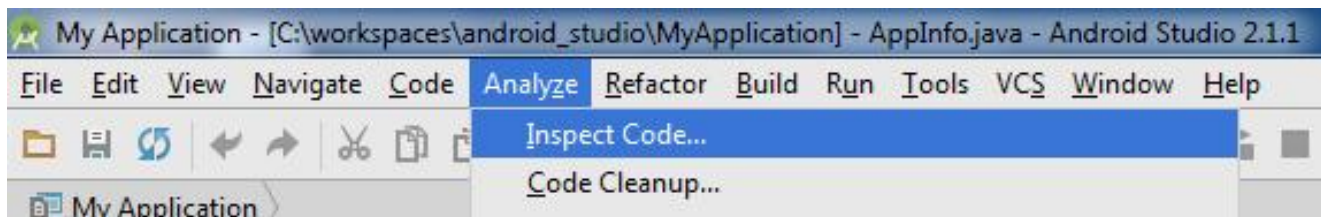
## 4. Lint Workflow



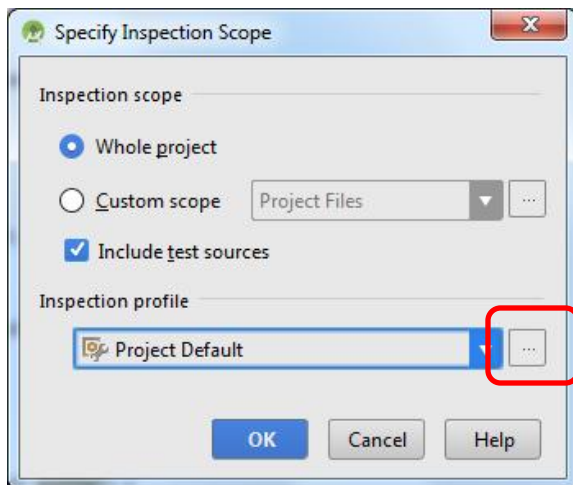
## 5.1 Lint 설정

- Android Studio에서 빌드시에 Lint 검사를 같이 수행할 수 있다.

1. 상단메뉴 > Analyze > Inspect Code



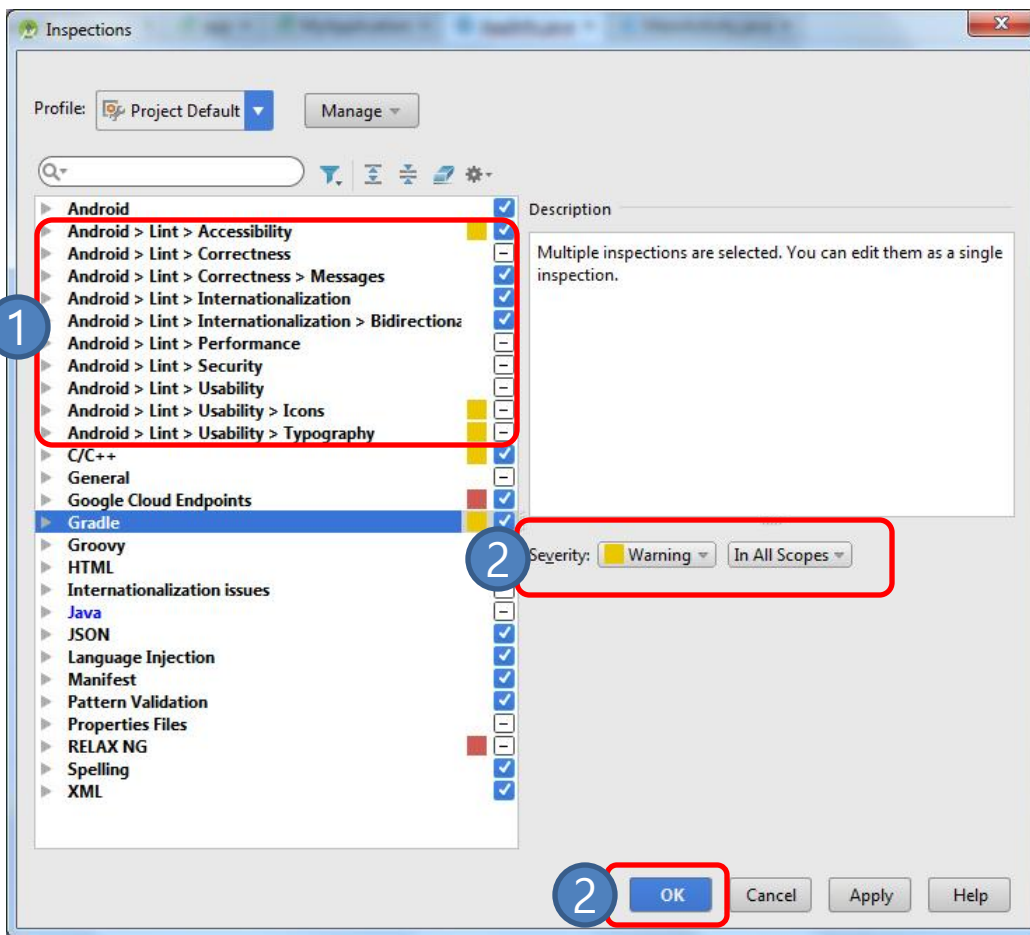
2. 팝업되는 설정창 우측하단 [ ... ] 버튼 클릭



클릭

## 5.2 Lint 설정

### 3. 검사항목 설정창



- 각 항목을 클릭한 후 설정

① Lint의 경우

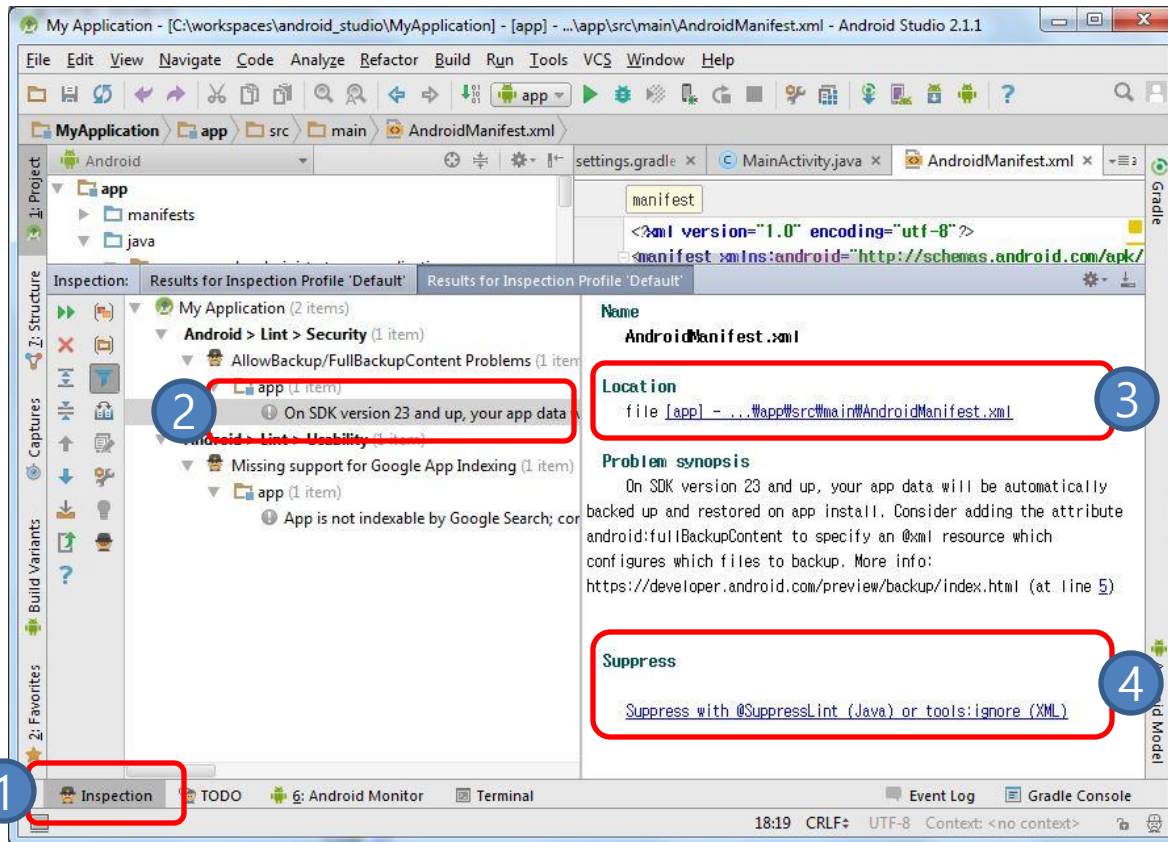
Android > Lint 아래에 항목들을 수정할 수 있으며, 하부항목의 이름에 따라, 접근성, 보안, 성능등의 카테고리리를 확인할 수 있다.

② 각 항목 클릭후 우측에 나타나는 Severity(심각도) 및 Scope(검사영역)를 프로젝트에 맞게 임의로 변경할 수도 있다

③ 설정 후 OK 를 클릭한다

## 5.3 Lint 설정 - Inspection Tab

### 4. 일반적으로 화면 하단에 Inspection Tab이 나타난다

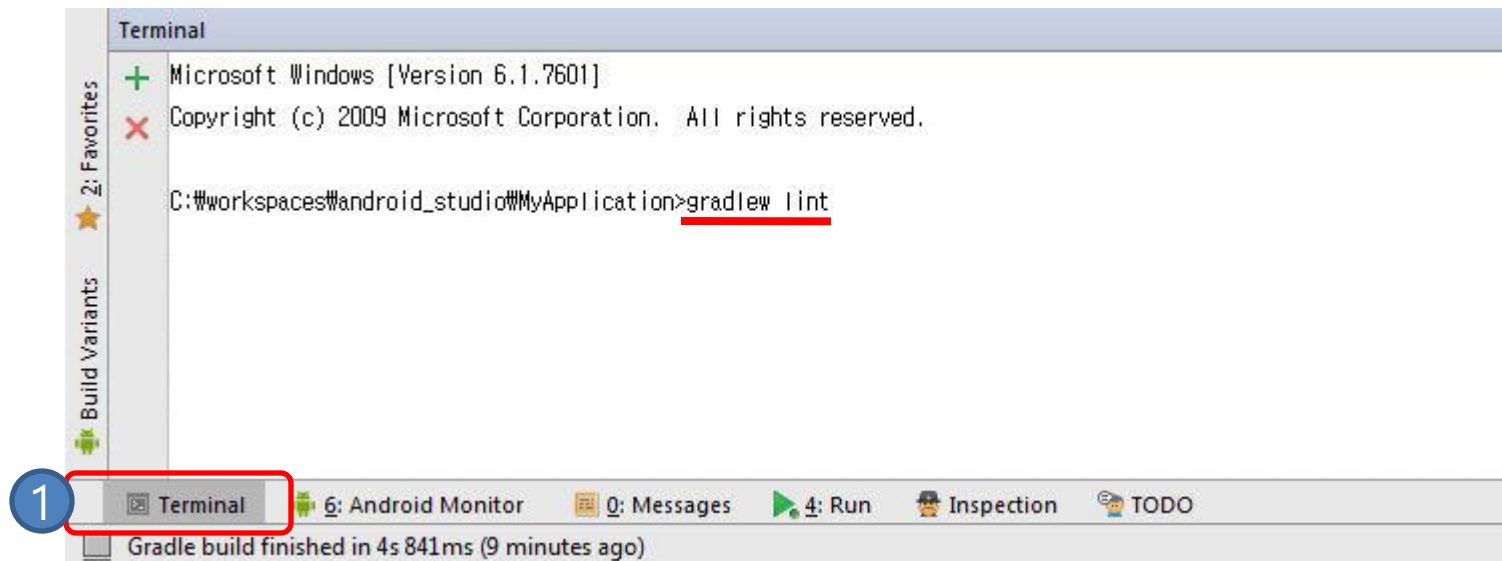




## 6.1. Gradle Lint 실행 - Terminal

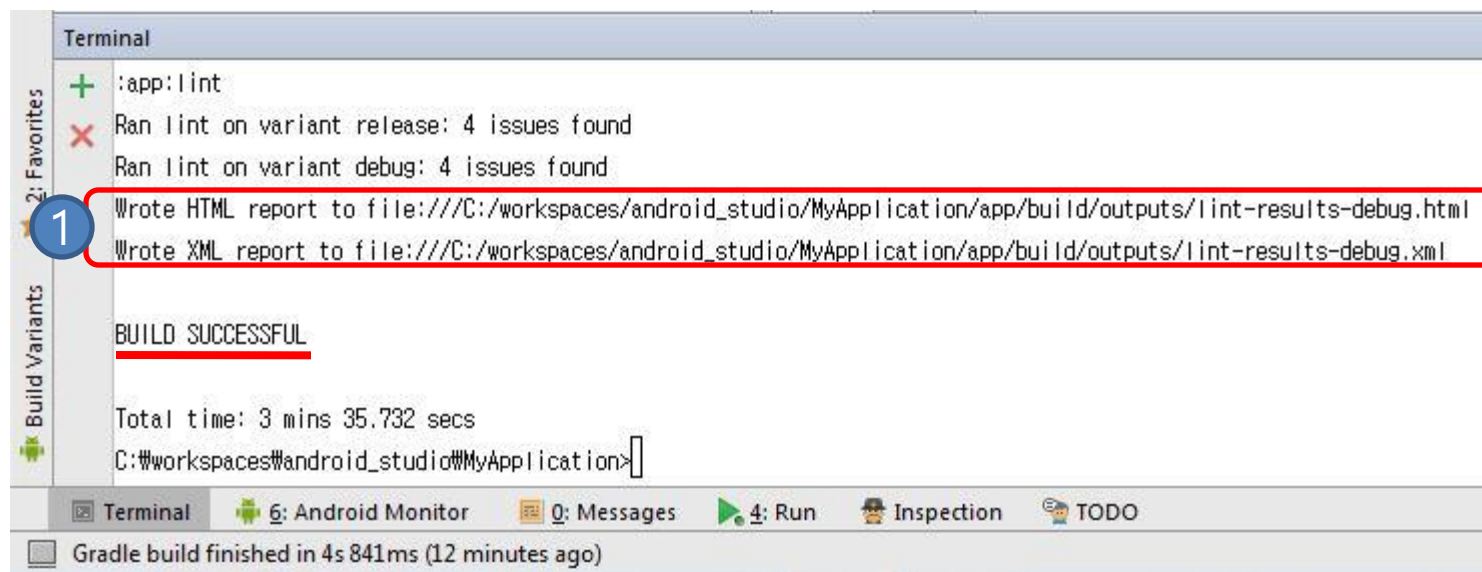
1. 하단 ① Terminal 탭에서 아래와 같이 입력한다.  
탭의 순서는 아래 그림과 다를 수도 있다

C:\프로젝트경로> **gradlew lint**



## 6.2. Lint 실행결과 - Terminal

2. 아래와 같이 성공 메시지가 출력된다  
프로젝트 경로아래 > app > build > outputs 폴더내에 ①검사결과 파일이 생성된다



```
Terminal
+ :app:lint
x Ran lint on variant release: 4 issues found
Ran lint on variant debug: 4 issues found
1 Wrote HTML report to file:///C:/workspaces/android_studio/MyApplication/app/build/outputs/lint-results-debug.html
Wrote XML report to file:///C:/workspaces/android_studio/MyApplication/app/build/outputs/lint-results-debug.xml

BUILD SUCCESSFUL

Total time: 3 mins 35.732 secs
C:\workspaces\android_studio\MyApplication>
```

Terminal 6: Android Monitor 0: Messages 4: Run Inspection TODO

Gradle build finished in 4s 841ms (12 minutes ago)

## 6.3. Lint 실행결과 확인 HTML

3. 프로젝트 > app > build > outputs > lint-results-debug.html 파일을 웹브라우저에서 열어본다. 아래와 같은 화면을 확인할 수 있다

### Lint Report

---

Check performed at Mon Aug 08 20:53:53 KST 2016.  
0 errors and 4 warnings found:

---

**Correctness**

1	⚠ GradleDependency: Obsolete Gradle Dependency
---	--

---

**Security**

1	⚠ AllowBackup: AllowBackup/FullBackupContent Problems
---	---

---

**Usability**

1	⚠ GoogleAppIndexingWarning: Missing support for Google App Indexing
---	---

---

**Internationalization**

1	⚠ HardcodedText: Hardcoded text
---	---------------------------------

---

**Disabled Checks (15)**

## 7. 검사설정 변경

- 이렇게 lint 를 통해 코드 내역을 검사했다고 하더라도 실제로 프로젝트에 영향을 미치지 않거나 개발자가 의도적으로 작성한 코드들도 존재 할 수 있다. 이럴 경우 검사설정을 변경하여 lint 검사내역에서 제외할 수 있다
- 검사설정 변경을 위해서 아래의 방법들을 사용할 수 있다

source code /  
resource

.java, .xml 등의 소스파일에 @애너테이션형태의 소스코드 직접 추가

build.gradle

gradle 스크립트 내에 lintOptions 추가

lint.xml

린트 설정파일에 <lint> 항목 직접 작성

## 7.1. lint 설정 - 소스코드

source code / resource

.java, .xml 등의 소스파일에 @애너테이션형태의 소스코드 직접 추가

- java 코드

```
@SuppressWarnings("검사항목") // SuppressLint 애너테이션 사용
public void foo( ) {
}

@SuppressWarnings({"검사항목1", "검사항목2"}) // 복수개의 Lint 검사 제외시
public void bar( ) {
}
```

- resource

```
<?xml version="1.0" encoding="utf-8"?>
<resources xmlns:tools="http://schemas.android.com/tools">
<string name="foo" tools:ignore="검사항목">Foo</string>
<string name="foo" tools:ignore="검사항목1,검사항목2">Foo</string>
</resources>
```

## 7.2. lint 설정 - gradle 스크립트

build.gradle

gradle 스크립트 내에 lintOptions 추가

- build.gradle 파일

```
android {
    lintOptions {
        disable 'RtlHardcoded','NewApi'
    }
}
```

- 지정 옵션

옵션	설정값	내용	설정 예시
abortOnError	boolean	에러 발생 시 빌드를 중단	abortOnError false
disable	String[ ]	검사에서 제외	disable 'NewApi','CheckResult'
enable	String[ ]	검사에 포함	enable 'MissingTranslation'
check	String[ ]	입력된 항목만 검사	check 'UnusedAttribute','NewApi'
lintConfig	File	Lint 설정파일 경로	lintConfig file("lint.xml")

## 7.3. lint 설정 - lint.xml

lint.xml

린트 설정파일에 <lint> 항목 직접 작성  
\* gradle 파일이 있는 곳에 생성

```
<?xml version="1.0" encoding="UTF-8"?>
<lint>
  <issue id="RtlHardcoded" severity="ignore" />
  <issue id="StopShip" severity="error" />

  <issue id="MissingTranslation">
    <ignore path="src/main/res/values/strings.xml" />
  </issue>

  <issue id="NewApi">
    <ignore path="src/./MainActivity.java" />
  </issue>
</lint>
```