

Kotlin Lint Rules

Created by Paul Ridding, last modified on Jan 18, 2021

When building Android applications using Kotlin we should integrate a linter into the process that ensures that all code follows the same style. We will use KTLint as the linter and to enable it we need to do the following:

- Create a `ktlint.gradle` file in the base directory. Add the following content:

```
repositories {  
    jcenter()  
}  
  
configurations {  
    ktlint  
}  
  
dependencies {  
    ktlint "com.pinterest:ktlint:0.36.0"  
}  
  
task ktlint(type: JavaExec, group: "verification") {  
    description = "Check Kotlin code style."  
    classpath = configurations.ktlint  
    main = "com.pinterest.ktlint.Main"  
    args "src/**/*.kt"  
}  
  
check.dependsOn ktlint  
  
task ktlintFormat(type: JavaExec, group: "formatting") {  
    description = "Fix Kotlin code style deviations."  
    classpath = configurations.ktlint  
    main = "com.pinterest.ktlint.Main"  
    args "-F", "src/**/*.kt"  
}  
  
preBuild.dependsOn ktlint
```

- Reference the `ktlint.gradle` file from the main `build.gradle`:

```
apply from : '../ktlint.gradle'
```

- This makes two new tasks available, you can use these from the command line using 'gradle ktlint' and 'gradle ktlintFormat'
 - ktlint - This verifies the format of your code files.
 - ktlintFormat - This fixes any issues (when it can fix it automatically) and then verifies the format.

By adding the `preBuild` dependency we can ensure that the `ktlint` call will be the first command run during the build process. If the lint rules are not followed then the build will fail.

No labels