



I'm not robot



reCAPTCHA

Continue

Exclude classes from jacoco report gradle

After searching Google, reading the Gradle docs and going through older StackOverflow posts, I found this answer on the Official gradle forums! jacocoTestReport { afterEvaluate { classDirectories.setFrom(files(classDirectories.files.collect { fileTree(dir: it, exclude: 'com/blah/**') }) } } Source: For older gradle versions < 5.x may need to use classDirectories = files(classDirectories.files.collect { instead of classDirectories.setFrom Solution to my build.gradle for Java/Groovy projects: apply plugin: 'java' apply plugin: 'jacoco' jacocoTestReport { reports { xml { enabled true // coversalls plugin depends on xml format report } html { enabled true } } afterEvaluate { classDirectories = files(classDirectories.files.collect { fileTree(dir: it, exclude: ['codeeval/**', 'crackingthecode/part3knowledgebased/**', '**/Chapter7ObjectOrientedDesign/**', '**/Chapter11Testing/**', '**/Chapter12SystemDesignAndMemoryLimits/**', 'projecteuler/**'] }) } } As you can see, I was successfully able to add more to exclude: in order to filter a few packages. Source: Custom tasks for other projects such as Android: apply plugin: 'jacoco' task jacocoReport(type: JacocoReport) { reports { xml { enabled true // coversalls plugin depends on xml format report } html { enabled true } } afterEvaluate { classDirectories = files(classDirectories.files.collect { fileTree(dir: it, exclude: ['codeeval/**', 'crackingthecode/part3knowledgebased/**', '**/Chapter7ObjectOrientedDesign/**', '**/Chapter11Testing/**', '**/Chapter12SystemDesignAndMemoryLimits/**', 'projecteuler/**'] }) } } Source: I want to use Jacoco in a way so that it excludes a Sample.java class from the overall coverage. To achieve that I have included within prepare-agent goal in maven pom.xml Jacoco plugin: org.jacoco:jacoco-maven-plugin 0.7.1.201405082137 default-prepare-agent prepare-agent Surefire plugin: org.apache.maven.plugins:maven-surefire-plugin 2.4.3 **/*Sample.java properties section: -Dfile.encoding=ISO-8859-1 2 This feature requires the use of JaCoCo version 0.6.3 or higher. Code coverage requirements can be specified for a project as a whole, for individual files, and for particular JaCoCo-specific types of coverage, e.g., lines covered or branches covered. The following example describes the syntax. Example 5. Configuring violation rules tasks.jacocoTestCoverageVerification { violationRules { rule { limit { minimum = "0.5".toBigDecimal() } } rule { isEnabled = false element = "CLASS" includes = listOf("org.gradle.*") limit { counter = "LINE" value = "TOTALCOUNT" maximum = "0.3".toBigDecimal() } } } } jacocoTestCoverageVerification { violationRules { rule { limit { minimum = 0.5 } } rule { enabled = false element = "CLASS" includes = ["org.gradle.*"] limit { counter = "LINE" value = "TOTALCOUNT" maximum = 0.3 } } } } The JacocoCoverageVerification task is not a task dependency of the check task provided by the Java plugin. There is a good reason for it. The task is currently not incremental as it doesn't declare any outputs. Any violation of the declared rules would automatically result in a failed build when executing the check task. This behavior might not be desirable for all users. Future versions of Gradle might change the behavior. I have multi-module gradle build. I want to generate aggregated view of Code Coverage across modules, subprojects { apply plugin: 'java' dependencies { compile 'org.jacoco:org.jacoco.ant:0.7.5.201505241946' } jacocoTestReport { additionalSourceDirs = files(sourceSets.main.allSource.srcDirs) sourceDirectories = files(sourceSets.main.allSource.srcDirs) //classDirectories = files(sourceSets.main.output) afterEvaluate { classDirectories = files(classDirectories.files.collect { fileTree(dir: it, exclude: 'net/xxxxx/test/core') }) //this task is excluding the core package classes } reports { html.enabled = true xml.enabled = false csv.enabled = false html.destination = "\${buildDir}/jacocoHtml" } } } task jacocoRootReport(type: org.gradle.testing.jacoco.tasks.JacocoReport) { dependsOn = subprojects.test additionalSourceDirs = files(subprojects.sourceSets.main.allSource.srcDirs) sourceDirectories = files(subprojects.sourceSets.main.allSource.srcDirs) classDirectories = files(subprojects.sourceSets.main.output) executionData = files(subprojects.jacocoTestReport.executionData) reports { html.enabled = true xml.enabled = false csv.enabled = false html.destination = "\${buildDir}/jacocoHtml" } } //however Root Report is not excluding the same. Please let me know how do I exclude classes from Root/Aggregated report.