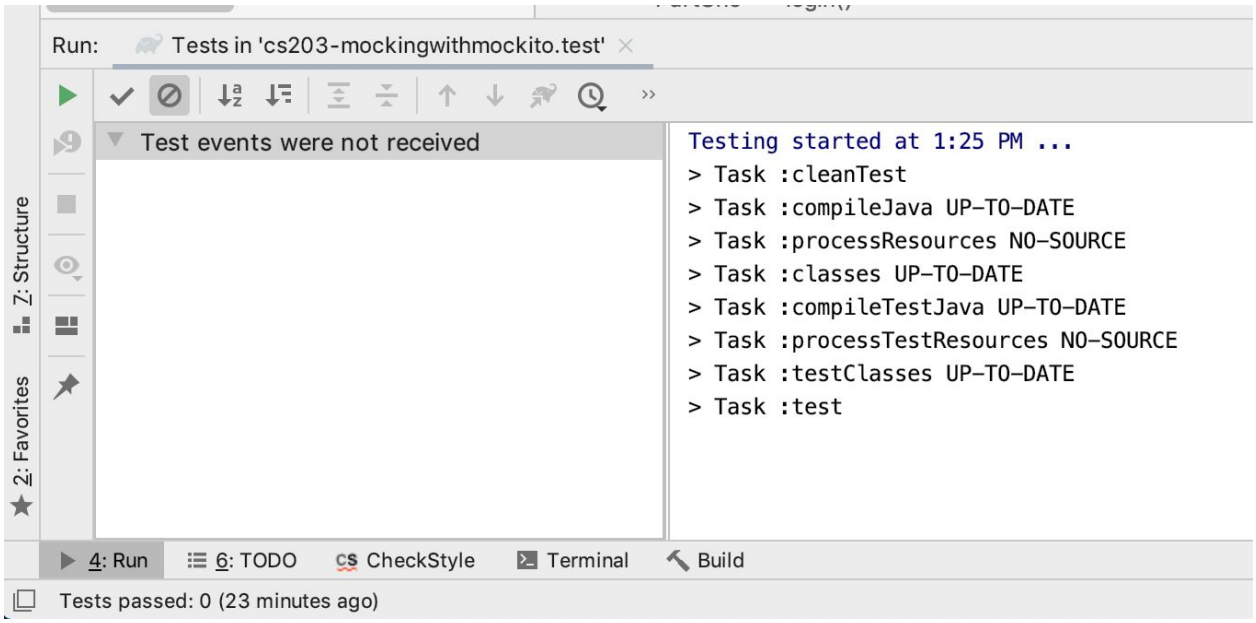
Setup and Test Running Issues in Intellij

We have observed various setup and test running issues with Intellij that may or may not occur for you, depending on which versions of Intellij, Java, and dependency management frameworks (Gradle and Maven) you are using. Please review this document if you encounter issues such as incorrect Java target versions, or test not found errors. Some of the issues occur only in Gradle projects or only in Maven projects, while others may occur with either framework or when you are not using a framework. If you do not find a solution here, try a Google search. Most problems you encounter will also have been encountered by others and solutions can usually be found after a few minutes of searching.

Issues Covered in This Document

Test Events Were Not Received

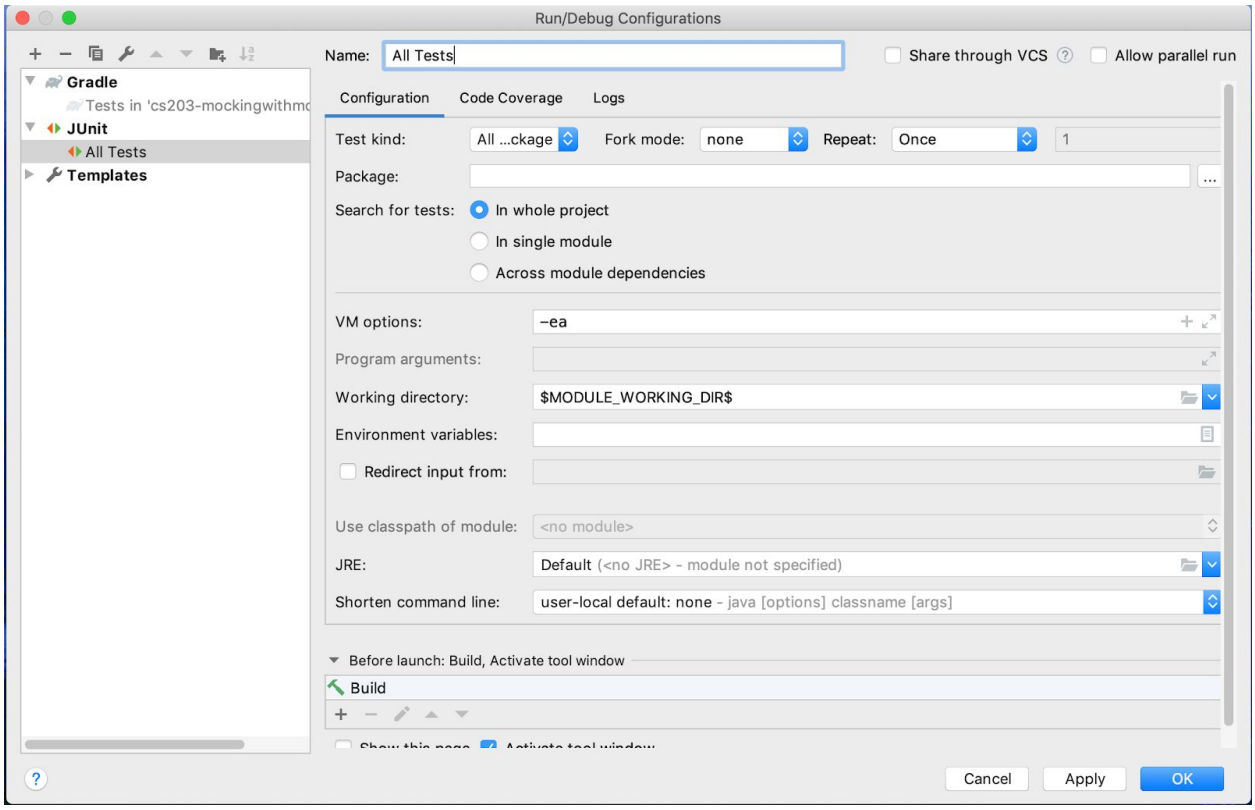
While attempting to run unit tests in a Gradle project, you may encounter a “Test events were not received” error.



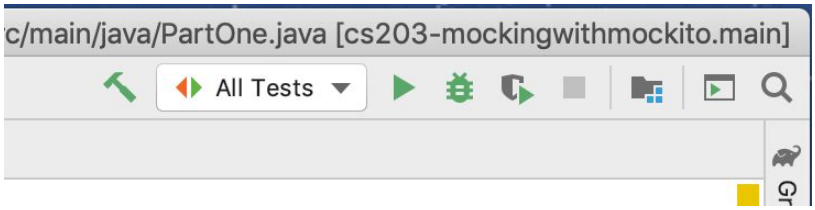
This seems to be a bug in the Gradle integration of recent versions of Intellij.

A workaround for this issue is to create a run configuration for your tests that uses the JUnit test runner instead of the Gradle test runner. Create and use a JUnit run configuration by completing the following steps:

1. Navigate to “Run > Edit Configurations…” in the top toolbar, click on the “+” in the top left corner to add a new configuration and select “JUnit”. Give it a name that describes which tests you are trying to run.
2. Specify the “Test Kind”, “Package” (if relevant based on the Test Kind) you select, and the “Search for Tests” value. For example, to create a configuration to run all tests in the default package (meaning all tests for which you have not specified a package), you would select “All in package” for “Test kind”, leave the “Package” blank and specify “In whole project” for “Search for tests:” as shown below:

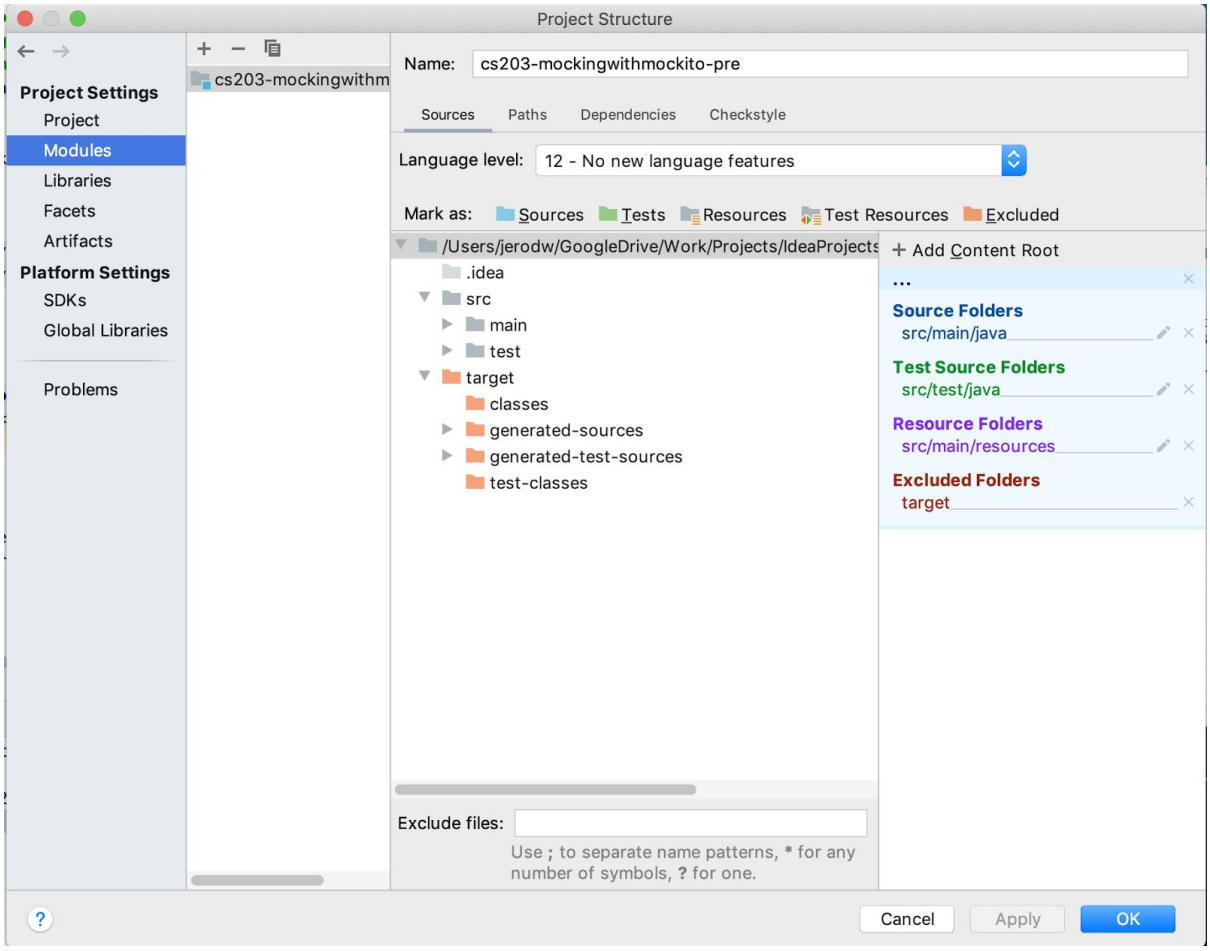


3. Select OK.
4. Even though you have created a JUnit run configuration, Intellij may still try to use a Gradle configuration when you run your tests. Ensure that the configuration you just created runs by selecting the run configuration you created in the drop-down near the upper right corner and selecting one of the run icons to the right, as shown below:



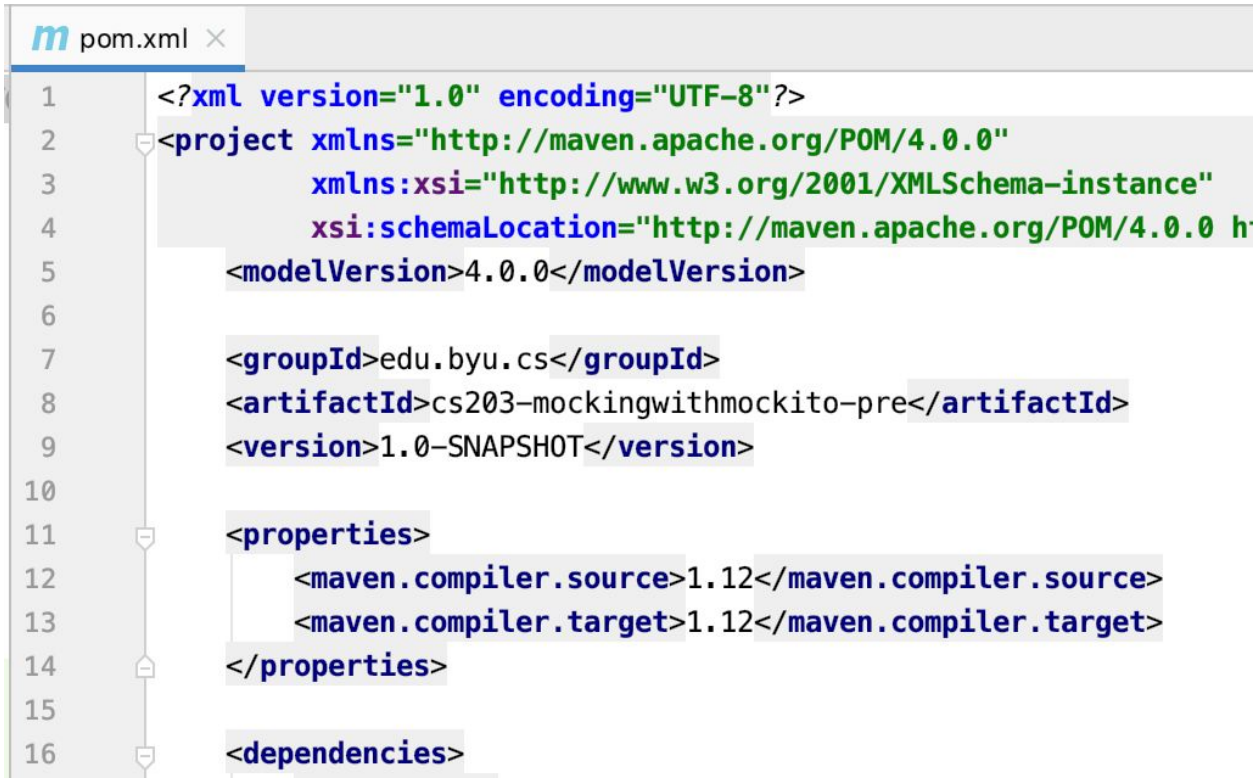
Intellij Reverts to An Earlier Java Version After you Specify a Latter One

This problem may occur with Maven projects. To specify a later version of Java for your project than the default your Intellij installation uses, open the “Project Structure” dialog (found by selecting “File > Project Structure…”), and then select the desired version by selecting “modules” then your project’s specific module (projects can have multiple modules but in this class our projects will only have one), make sure the ‘Sources’ tab is selected, and then setting the desired version as the “Language Level”.



Sometimes if your project is a Maven project, your language setting will revert back to a prior language level, which you will notice when you start getting compile errors for using later version syntax. Do the following to solve this problem:

1. Open your pom.xml file and add “maven.compiler.source” and “maven.compiler.target” properties that specify your desired Java version as shown below:

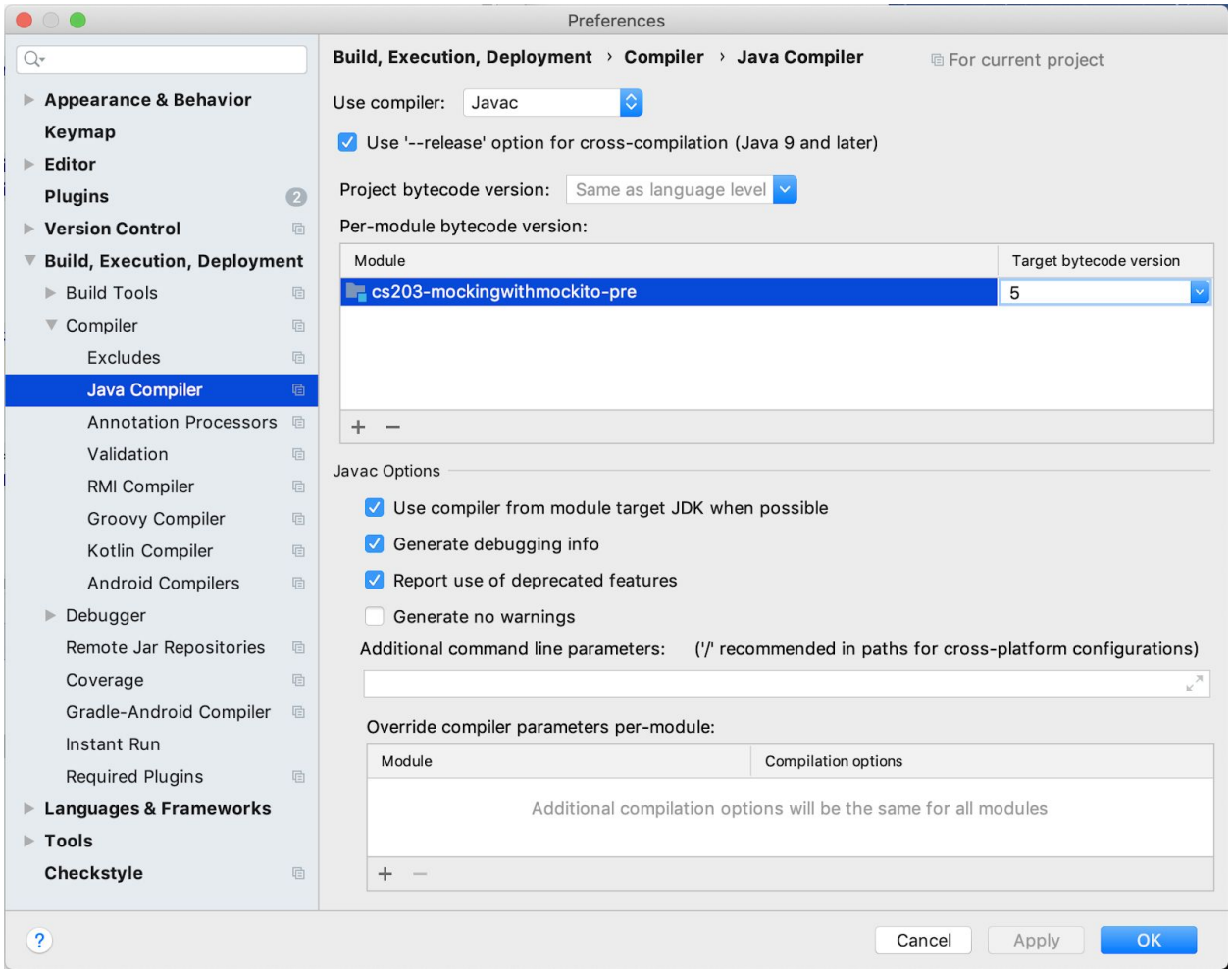


2. Open the project structure dialog one more time and set the “Language Level” to the same version used in your pom.xml properties if it isn’t already set to that level (after setting the properties in pom.xml, this will be the last time you have to do this--Intellij will no longer change it).
3. Follow the instructions for the “Error:java: error: release version 5 not supported” issue to set the target bytecode version to the same Java version used in the previous two steps.

Error:java: error: release version 5 not supported

If you see the following error: “Error:java: error: release version 5 not supported”, your project settings are indicating that your code should be compiled to Java version 5 (which seems to be the default on even some recent versions of Intellij) even if you have specified that your project should be compiled using the rules of a later version of Java. To fix this problem, you must set the Java target to a later version by doing the following:

1. Open Intellij preferences (found from “IntelliJ IDEA > Preferences…” on Mac) and select “Build, Execution, Deployment > Compiler > Java Compiler’.



2. In the table in the middle of the dialog, select the version you want to use in the “Target bytecode version” dropdown.
3. Select OK.