

# ToiletPaper #122

## Remote Debugging of Maven (Surefire) Tests

Author: Jan Machnik / Software Engineer/ Office Stuttgart

### ✗ Problem

There is this phenomenon – all unit tests in IntelliJ's IDEA are positive, but Maven disagrees and displays failed tests in your build pipeline by saying "BUILD FAILURE, have fun searching".

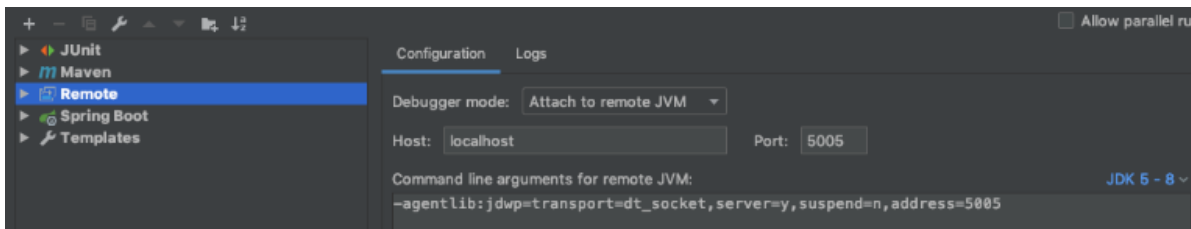
But how do you actually get the code debugged during build time? Simply setting a breakpoint and running the test again in IntelliJ's debug mode won't work. We need to run the test under the same conditions as Maven does.

### ✓ Solution

We can use the Maven Surefire plugin, which Maven uses during the test phase of the build lifecycle. With the plugin, it is possible to have Maven connecting a port with your IDE. But the remote debugger is not limited to local Maven projects. You can also debug Gradle projects, JARs, applications on other servers, and even Docker containers.

### ➔ Example

First, we need to create a remote run configuration in IntelliJ IDEA (works similarly in other IDEs like Eclipse, ...), set the port (Maven uses port 5005 by default), and set a few more command line parameters, as displayed in an exemplary configuration:



Then breakpoints can be set and the build process can be started again with the command line parameter - **Dmaven.surefire.debug**.

For a single test the command could look like this: **mvn package -Dtest=TestClass.java#testMethod -Dmaven.surefire.debug**

For all tests: **mvn package -Dmaven.surefire.debug**

Maven reports as soon as it arrives at the tests:

```
-----  
T E S T S  
-----
```

```
Listening for transport dt_socket at address: 5005
```

Now it's time to start the previously created run configuration in IntelliJ IDEA and hope that the breakpoints are set correctly ;). From this point on, we can step through the code as usual and find the cause of our error. In that sense, happy debugging!

### + Further Aspects

- [This link leads to the documentation of the Maven plugin](#)
- [General information about remote debugging](#)