

TEST AUTOMATION TRAINING

Aaron Evans
@fijiaaron
aaron@one-shore.com

RUNNING TESTS IN PARALLEL WITH

MAVEN SUREFIRE PLUGIN

SIMPLE TEST

I'm going to show you how to run automated tests in parallel using the Maven Surefire plugin.

This will allow you to run tests faster and get feedback from every checkin without waiting hours for a full test run to complete.

I have a simple Selenium test here written in Java and using the JUnit test runner.

It checks the current temperature using Google weather and then verifies that my result is for the the right city and the temperature is displayed in Celsius as requested.

```
mvn clean test -Dtest=SimpleTest
```

PARAMETERIZED TEST

I'm just going to duplicate it and create a parameterized test that will check the weather in several cities.

```
mvn clean test -Dtest=ParameterizedTest
```

Let's see how long this takes.

UI tests are inherently slow.

When you automate tests with Selenium you have to start the browser, wait for pages to load, and perform actions like typing and clicking buttons.

There are some things you can do to improve this but still it's going to be slow. And when you have hundreds of tests it can really add up.

PARALLELIZATION

It's important to get test results as fast as possible. The faster the feedback you provide, the more often the tests can be run and any issues you find can be dealt with right away.

The best way to get through hundreds of tests quickly is to run them in parallel.

If you have 100 tests that take 1 minute each, it will take more than an hour and a half to finish the full suite.

But if you can run them all at the same time, its only going to take a minute. Now you can get results in a reasonable timeframe and thoroughly test your system with every checkin.

MAVEN SUREFIRE PLUGIN

These tests are written in Java and use maven to execute them.

Maven is a common build tool that handles dependencies and compiling your project. It also has a task for running tests.

The maven surefire plugin can run junit (or testng) tests in parallel by executing multiple threads.

[surefire documentation page]

<https://maven.apache.org/surefire/maven-surefire-plugin/>

CONFIGURING PARALLEL TESTS WITH MAVEN SUREFIRE PLUGIN

Let's look at our maven pom.xml configuration file to see how that works

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>3.0.0-M1</version>
  <configuration>
    <parallel>all</parallel>
    <threadCount>10</threadCount>
    <redirectTestOutputToFile>>false</redirectTestOutputToFile>
  </configuration>
</plugin>
```

Include the maven surefire plugin under build > plugins

Add <configuration> <parallel>. This can be set to classes, methods, or both.

<threadCount> says we want to run 10 threads in parallel.

<redirectTestOutputToFile> just says to print standard output on the command line.

RUNNING TESTS IN PARALLEL

Let's go ahead and run our tests again, this time in parallel.
Same way as before, with 'mvn test'

```
mvn clean test -Dtest=ParameterizedTest
```

And now let's watch them run in parallel and see how long it takes.

Already done.

It's that simple.

The maven surefire plugin allows your tests to run in parallel resulting in faster total execution time, which means you can run your tests more often and catch bugs quicker.