

# Integrating JUnit 5

---



**Esteban Herrera**

JAVA ARCHITECT

@eh3rrera [www.eherrera.net](http://www.eherrera.net)



# Overview



Running tests from the console

Running tests with Gradle

Running tests with Maven

Include/exclude tests with tags

Code coverage

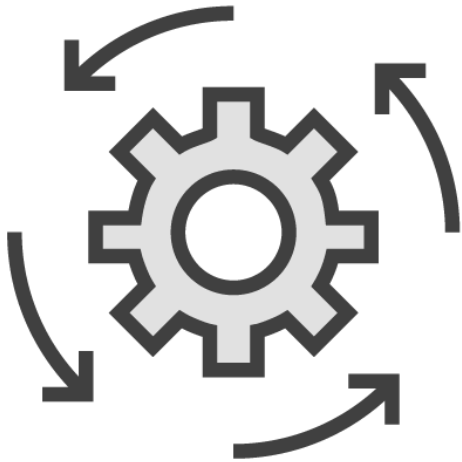


# Running Tests from the Console

---



# JUnit 5 Console Launcher



**Group ID:** org.junit.platform

**Artifact ID:** junit-platform-console-standalone

**Version:** 1.0.1

<http://bit.ly/mavenjunitconsole>



```
java -jar junit-platform-console-standalone-1.0.1.jar \  
    --cp ${PATH_TO_COMPILED_CLASSES} \  
    --scan-classpath
```

## Run Tests



# Demo



## Running tests from the console



# Running Tests with Gradle

---



# Enabling the JUnit Gradle Plugin

```
buildscript {  
    repositories {  
        mavenCentral()  
    }  
    dependencies {  
        classpath 'org.junit.platform:junit-platform-gradle-plugin:1.0.1'  
    }  
}  
  
apply plugin: 'org.junit.platform.gradle.plugin'
```





# Configuring the JUnit Gradle Plugin

...

```
junitPlatform {
```

```
}
```



# Configuring the JUnit Gradle Plugin

...

```
junitPlatform {
```

```
    logManager 'org.apache.logging.log4j.jul.LogManager'
```

```
}
```



# Configuring the JUnit Gradle Plugin

...

```
junitPlatform {  
    logManager 'org.apache.logging.log4j.jul.LogManager'  
    reportsDir file('build/test-results/junit-platform')  
  
}
```



# Configuring the JUnit Gradle Plugin

...

```
junitPlatform {  
    logManager 'org.apache.logging.log4j.jul.LogManager'  
    reportsDir file('build/test-results/junit-platform')  
    enableStandardTestTask true  
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    logManager 'org.apache.logging.log4j.jul.LogManager'  
    reportsDir file('build/test-results/junit-platform')  
    enableStandardTestTask true  
    configurationParameter 'junit.jupiter.testinstance.lifecycle.default', 'per_class'  
  
}
```



# Configuring the JUnit Gradle Plugin

```
...
junitPlatform {
    logManager 'org.apache.logging.log4j.jul.LogManager'
    reportsDir file('build/test-results/junit-platform')
    enableStandardTestTask true
    configurationParameter 'junit.jupiter.testinstance.lifecycle.default', 'per_class'
    configurationParameters([
        'junit.jupiter.testinstance.lifecycle.default': 'per_class',
        'junit.jupiter.conditions.deactivate': '*'
    ])
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
    selectors {  
  
  
    }  
}
```



# Configuring the JUnit Gradle Plugin

...

```
junitPlatform {
```

...

```
selectors {
```

```
    uri 'file:///data.txt'
```

```
    file 'data.json'
```

```
    directory 'resources/data'
```

```
    aPackage 'com.wbc.rewards'
```

```
    aClass 'com.wbc.rewards.PointsTest'
```

```
    method 'com.wbc.sale.SaleTest#test(java.lang.String[])'
```

```
    resource '/com/wbc/program.properties'
```

```
}
```

```
}
```





# Configuring the JUnit Gradle Plugin

```
...
junitPlatform {
    ...
    selectors {
        uris 'file:///data.txt', 'file:///data2.txt'
        files 'data.json', 'data2.json'
        directories 'resources/data', 'resources/data2'
        packages 'com.wbc.rewards', 'com.wbc.customer'
        classes 'com.wbc.rewards.PointsTest', 'com.wbc.rewards.PointsTest'
        methods 'com.wbc.sale.SaleTest#test(java.lang.String[])', 'com.wbc.Main#main'
        resources '/com/wbc/program.properties', '/com/wbc/points.properties'
    }
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
  
    filters {  
  
  
    }  
  
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
  
    filters {  
        engines {  
            include 'junit-jupiter', 'junit-vintage'  
        }  
    }  
}
```



# Configuring the JUnit Gradle Plugin

```
...
junitPlatform {
    ...

    filters {
        engines {
            exclude 'junit-jupiter', 'junit-vintage'
        }
    }
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
  
    filters {  
        tags {  
            include 'test-ci', 'integration'  
        }  
    }  
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
  
    filters {  
        tags {  
            exclude 'test-ci', 'integration'  
        }  
    }  
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
  
    filters {  
        packages {  
            include 'com.wbc.rewards', 'com.wbc.customer'  
        }  
    }  
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
  
    filters {  
        packages {  
            exclude 'com.wbc.rewards', 'com.wbc.customer'  
        }  
    }  
}
```





# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
  
    filters {  
        includeClassNamePattern '.*Tests'  
    }  
}
```



# Configuring the JUnit Gradle Plugin

```
...  
junitPlatform {  
    ...  
  
    filters {  
        includeClassNamePattern '.*Tests'  
        includeClassNamePatterns '.*Test', '.*Case'  
    }  
}
```



# Demo



## Running tests with Gradle



# Running Tests with Maven

---



# Configuring the Maven Surefire Plugin

```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>2.19.1</version>
      <dependencies>
        <dependency>
          <groupId>org.junit.platform</groupId>
          <artifactId>junit-platform-surefire-provider</artifactId>
          <version>1.0.1</version>
        </dependency>
      </dependencies>
    </plugin>
  </plugins>
</build>
```



# Configuring the Maven Surefire Plugin

```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>2.19.1</version>
      <dependencies>
        ...
      </dependencies>

    </plugin>
  </plugins>
</build>
```



# Configuring the Maven Surefire Plugin

```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>2.19.1</version>
      <dependencies>
        ...
      </dependencies>
      <configuration>

      </configuration>
    </plugin>
  </plugins>
</build>
```



# Configuring the Maven Surefire Plugin

```
<configuration>
```

```
</configuration>
```





# Configuring the Maven Surefire Plugin

```
<configuration>  
  <includes>  
    <include>**/Test*.java</include>  
    <include>**/*Test.java</include>  
  </includes>  
</configuration>
```



# Default Inclusions

```
"**/Test*.java"
```

```
"**/*Test.java"
```

```
"**/*Tests.java"
```

```
"**/*TestCase.java"
```



# Configuring the Maven Surefire Plugin

```
<configuration>
  <includes>
    <include>**/Test*.java</include>
    <include>**/*Test.java</include>
  </includes>
</configuration>
```



# Configuring the Maven Surefire Plugin

```
<configuration>  
  <excludes>  
    <exclude>**/Test*.java</exclude>  
    <exclude>**/*Test.java</exclude>  
  </excludes>  
</configuration>
```



# Configuring the Maven Surefire Plugin

```
<configuration>  
  <properties>  
    <includeTags>integration</includeTags>  
    <excludeTags>slow, test-ci</excludeTags>  
  </properties>  
</configuration>
```



# Configuring the Maven Surefire Plugin

```
<configuration>
```

```
  <properties>
```

```
    </properties>
```

```
  </configuration>
```



# Configuring the Maven Surefire Plugin

```
<configuration>  
  <properties>  
    <configurationParameters>  
  
      </configurationParameters>  
    </properties>  
  </configuration>
```



# Configuring the Maven Surefire Plugin

```
<configuration>
  <properties>
    <configurationParameters>
      junit.jupiter.conditions.deactivate=*
      junit.jupiter.testinstance.lifecycle.default=per_class
    </configurationParameters>
  </properties>
</configuration>
```





# Demo



## Running tests with Maven



# Include/exclude Tests with Tags

---



# Tags

```
import org.junit.jupiter.api.Tag;  
  
...  
@Tag("v1")  
@Tag("reward")  
class TestRewards {  
    @Test  
    @Tag("fast")  
    void testRewardProgram() { /* ... */ }  
}
```



# Interfaces

```
@Tag("error")
@ExtendWith({ExceptionHandler.class})
public interface ErrorHandler {
    ...
}
```

---

```
class TestRewardProgram implements ErrorHandler {
    ...
}
```



# Meta-annotations

```
@Target({ElementType.TYPE, ElementType.METHOD})
```

```
@Retention(RetentionPolicy.RUNTIME)
```

```
@Test
```

```
@Tag("error")
```

```
@ExtendWith({ExceptionHandler.class})
```

```
public @interface TestWithErrorHandler() { }
```

---

```
@TestWithErrorHandler
```

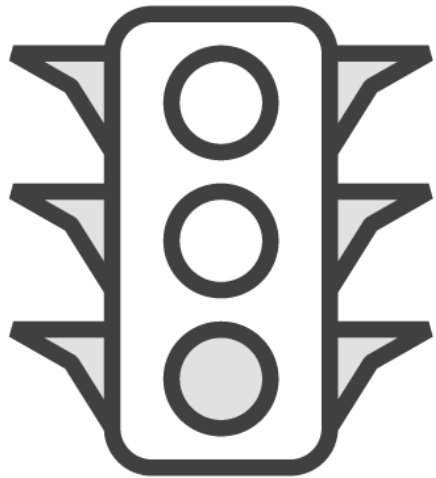
```
void testRewardProgram() {
```

```
    ...
```

```
}
```



# Syntax Rules for Tags



**A tag must not be null or blank**

**A *trimmed* tag must not contain:**

- Whitespace
- ISO control characters
- Any of the following *reserved characters*: , ( ) & | !

# Configuring the JUnit Gradle Plugin

...

```
junitPlatform {
```

...

```
  filters {
```

```
    tags {
```

```
      include 'slow'
```

```
      exclude 'test-ci', 'integration'
```

```
    }
```

```
  }
```

```
}
```



# Configuring the Maven Surefire Plugin

```
<configuration>  
  <properties>  
    <includeTags>integration</includeTags>  
    <excludeTags>slow, test-ci</excludeTags>  
  </properties>  
</configuration>
```





# Demo



## Tags

- Gradle
- Maven



# Code Coverage

---



# Code coverage

Measure used to describe the degree to which the code of your program is covered by your tests



# Demo



## Code Coverage

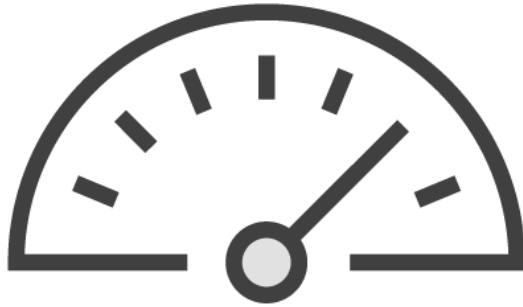
- JaCoCo
  - Maven
  - Gradle



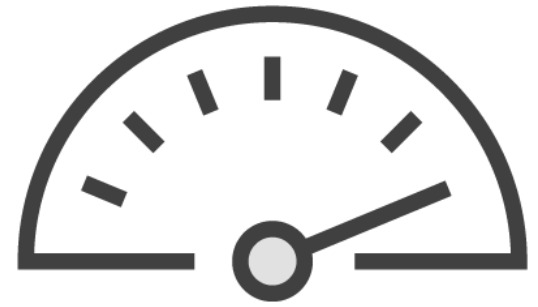
# What Percentage Should Be Our Goal?



60%?



80%?



100%?



# What Percentage Should Be Our Goal?



100%



100% code coverage  
doesn't mean your code  
works perfectly



# Just Remember



**100% must be the goal**

**But it only means how much code is tested**

**It doesn't necessarily mean it's well-tested**



# Mutation Testing



## **Small modifications**

AND instead of OR

## **If the unit test fails**

It works fine

## **If the test succeeds**

It is not testing the right thing

**The result is the percentage of failing tests**



# Summary



Running tests from the console

Running tests with Gradle

Running tests with Maven

Include/exclude tests with tags

Code coverage

