



# Junit 5

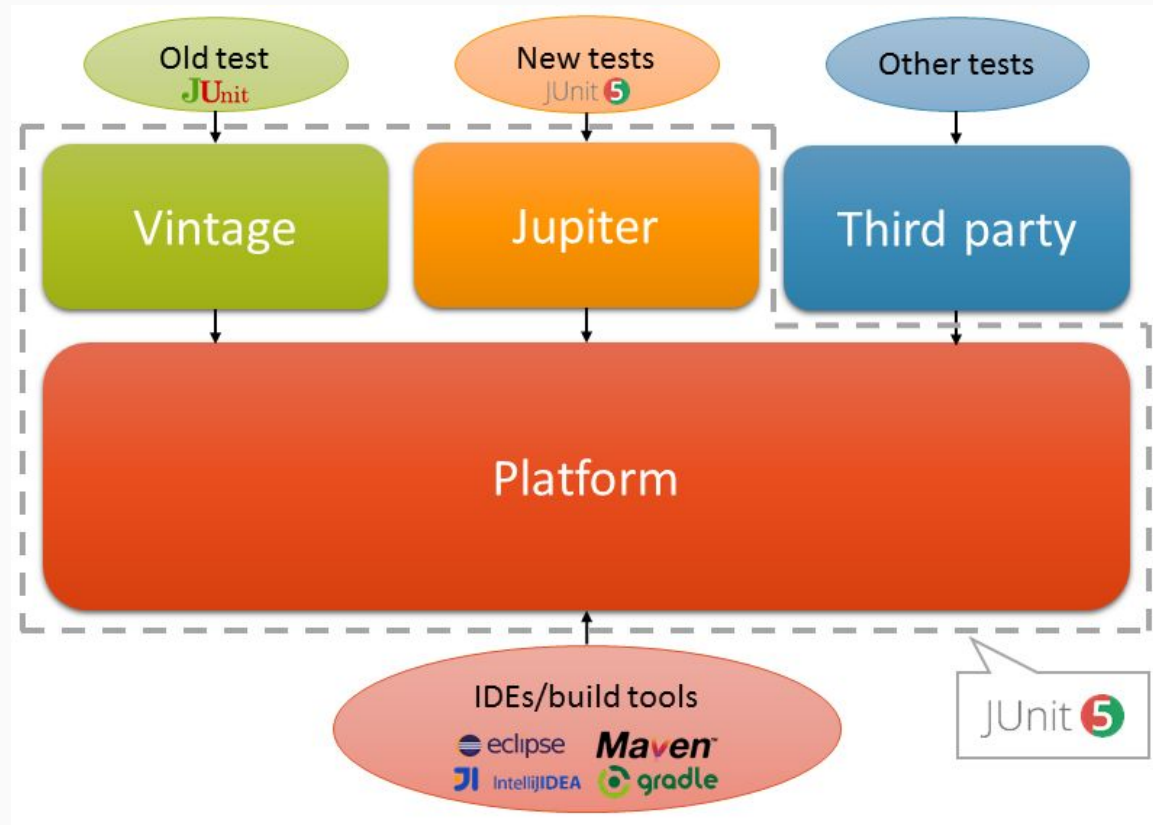
Getting Started

## We Cover

- Architecture
- Installation
- First test
- The process



# Architecture



JUnit 5 = JUnit Platform + [JUnit Jupiter] + [JUnit Vintage]

## JUnit Platform

- Foundation for launching testing frameworks on the JVM
- Define TestEngine API
- Providing Console Launcher

## JUnit Jupiter

- Programming Model
- Extension Model
- JUnit default test engine

## JUnit Vintage

- Test Engine for JUnit 3 and JUnit 4

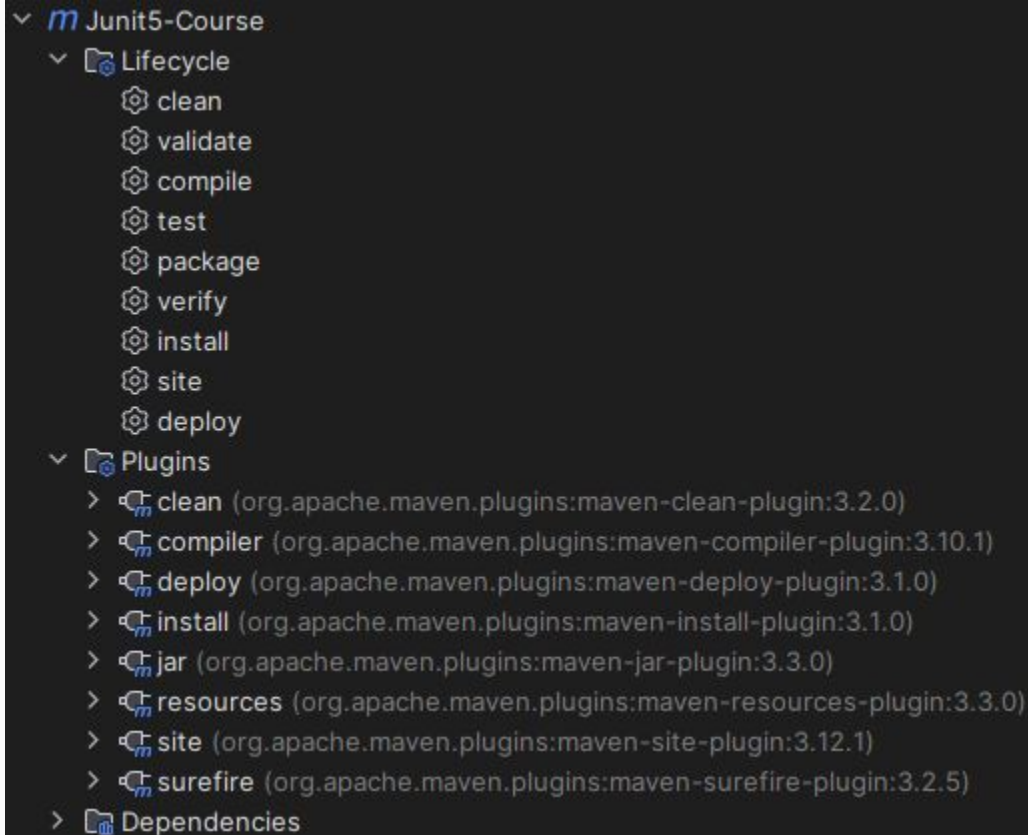


## Junit 5 pom.xml

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.junit</groupId>
      <artifactId>junit-bom</artifactId>
      <version>5.10.2</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>

<dependencies>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

# Maven Lifecycle



# Unit Test plugins

## Surefire

```
1. <project>
2.   [...]
3.   <build>
4.     <pluginManagement>
5.       <plugins>
6.         <plugin>
7.           <groupId>org.apache.maven.plugins</groupId>
8.           <artifactId>maven-surefire-plugin</artifactId>
9.           <version>3.2.5</version>
10.        </plugin>
11.      </plugins>
12.    </pluginManagement>
13.  </build>
14.  [...]
15. </project>
```

## Failsafe

```
1. <project>
2.   [...]
3.   <build>
4.     <plugins>
5.       <plugin>
6.         <groupId>org.apache.maven.plugins</groupId>
7.         <artifactId>maven-failsafe-plugin</artifactId>
8.         <version>3.2.5</version>
9.         <executions>
10.          <execution>
11.            <goals>
12.              <goal>integration-test</goal>
13.              <goal>verify</goal>
14.            </goals>
15.          </execution>
16.        </executions>
17.      </plugin>
18.    </plugins>
19.  </build>
20.  [...]
21. </project>
```

# First Test - Introduction

```
public static int divide(int a, int b) {  
    return a / b;  
}
```

```
.....  
  
public static void main(String[] args) {  
    int result = NumericalUtilities.divide(10, 2);  
    if (result != 5) {  
        throw new RuntimeException("Division result is not correct");  
    }  
    System.out.println("Division result is: " + result);  
}
```



10 / 2



9 / 2



2 / 0



9 / 4.5



# First Test

```
public static int divide(int a, int b) {  
    return a / b;  
}
```

```
@Test  
public void testDivide() {  
    int result = divide(10, 2);  
    Assertions.assertEquals(5, result);  
}
```



10 / 2



9 / 2



2 / 0



9 / 4.5

# First Test

```
public static double divide(int a, int b) {  
    return a / b;  
}
```

```
@Test  
public void testDivide() {  
    double result = divide(9, 2);  
    Assertions.assertEquals(4.5, result);  
}
```



10 / 2



9 / 2



2 / 0



9 / 4.5

# First Test

```
public static <T extends Number> double divide(T numerator, T denominator) throws IllegalArgumentException
{
    if (denominator.doubleValue() == 0) {
        throw new IllegalArgumentException("Cannot divide by zero");
    }

    return numerator.doubleValue() / denominator.doubleValue();
}
```

```
@Test
public void testDivide() {
    double result = divide(9, 4.5);
    Assertions.assertEquals(2, result);
}
```



10 / 2



9 / 2



2 / 0



9 / 4.5

# Surefire / Failsafe Mechanism

- Surefire / Failsafe attached to maven lifecycle
- Junit 5 engine starts
- Junit 5 execution engine starts
- Discovery process (class with Test / IT)
- List the classes according to filter and order
- In each class - list the tests according to filter and order
- List relevant extensions and executes them
- Instantiate each test class when it's time and execute the tests



# Time to add a notch