

JUnit 5 Fundamentals

INTRODUCING JUNIT 5



Esteban Herrera

JAVA ARCHITECT

@eh3rrera www.eherrera.net



What are the most
important or popular
Java frameworks?



“Never in the field of software development was so much owed by so many to so few lines of code.”

Martin Fowler

<http://bit.ly/mfowlerquote>



Overview



Introducing JUnit 5

Writing tests

Creating dynamic and parameterized tests

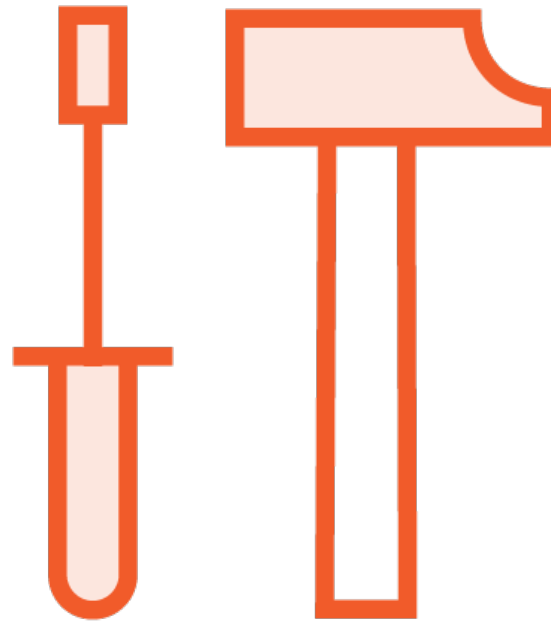
Extending JUnit

Integrating JUnit

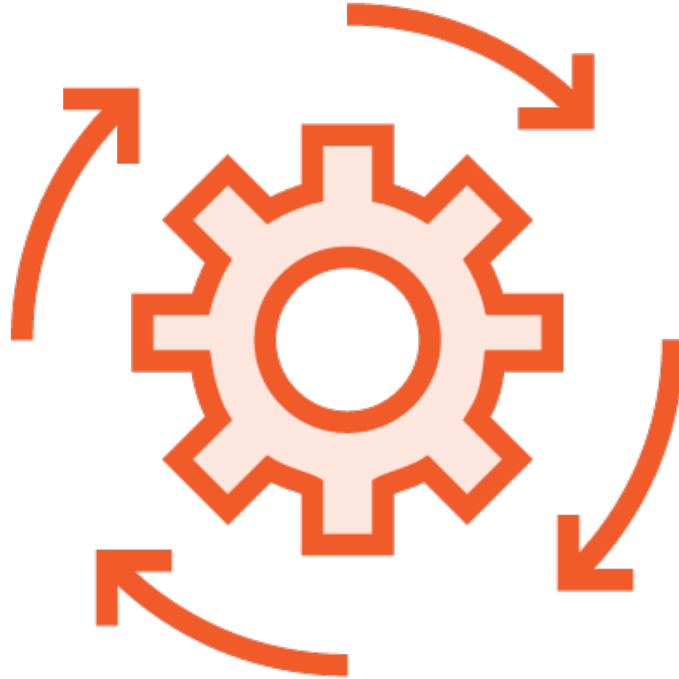
Migrating from JUnit 4



JUnit Is...



Testing Is Feedback



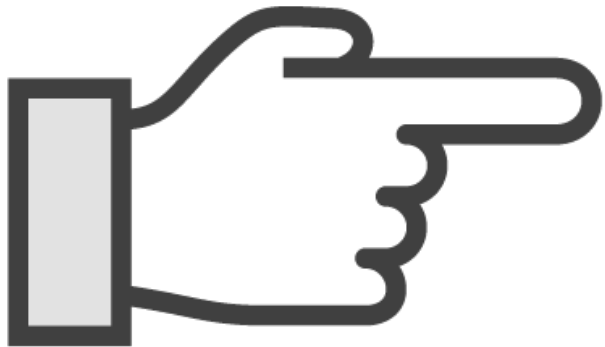
“Programs should be written for people to read, and only incidentally for machines to execute.”

Structure and Interpretation of Computer Programs
by Abelson and Sussman.

<http://bit.ly/sicppreface>



Tests Must Be...



Easy to understand

Easy to read

Easy to modify

Introduction to Testing in Java

by Richard Warburton

The easiest and most pleasant way to get started with unit testing, JUnit, and Test Driven Development (TDD) that you could imagine.

 Start Course



Bookmark



Add to Channel



Live mentoring

<http://bit.ly/introductiontestingjava>



Test-Driven Development Practices in Java

by Mike Nolan

This course covers Test-Driven Development (TDD) practices, and tools supporting TDD on the Java Platform.

 Start Course



Bookmark



Add to Channel



Live mentoring

<http://bit.ly/tddjavaps>



Questions?



Types of Tests



According to the Knowledge of the System



Black box



White box



Types of Testing

Acceptance

Integration

System

Load

Regression

Unit

Stress

Security

Recovery

Usability



Types of Testing

Unit



Unit Tests



Test a piece of code



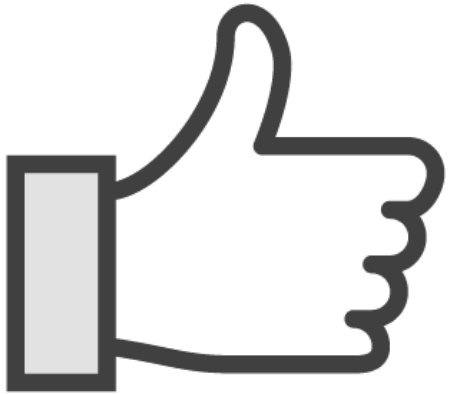

```
public BigDecimal calculateCommission() {  
    ...  
}
```



```
class Sale {  
    ...  
    public BigDecimal calculateCommission() {  
        ...  
    }  
    ...  
}
```



A Good Unit Test Should Be



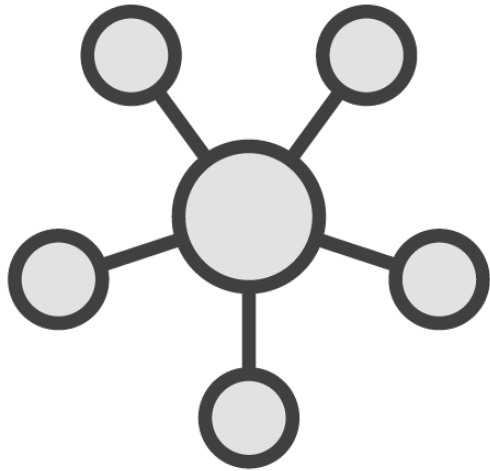
Automated
Repeatable
Fast



```
class Sale {  
    private Webservice commissionWS;  
    ...  
    public BigDecimal calculateCommission() {  
        double commissionPercentage = commissionWS.getPercentage();  
        ...  
    }  
    ...  
}
```



Integration Tests



Test how components work together

```
void testCalculateCommission() {  
    Sale sale = new Sale();  
    sale.setCommissionWS(new FakeWebService());  
  
    // Test calculateCommission() method ...  
}
```



Why Unit Tests Are Important



Benefits



Reduce debugging time

Serve as documentation

Help to improve the design



Would It Be...



Hard?

Time consuming?



Well...



Yes, but not impossible



What If Our Unit Tests Were...

Easy to run

Fast

Complete

Up to date



Courage



What Do You Think Is Better?

**Terrible designed system
Good suite of tests**

**Good designed system
Terrible suite of tests**



What's JUnit?



“JUnit is a simple, open source framework to write and run repeatable tests.”

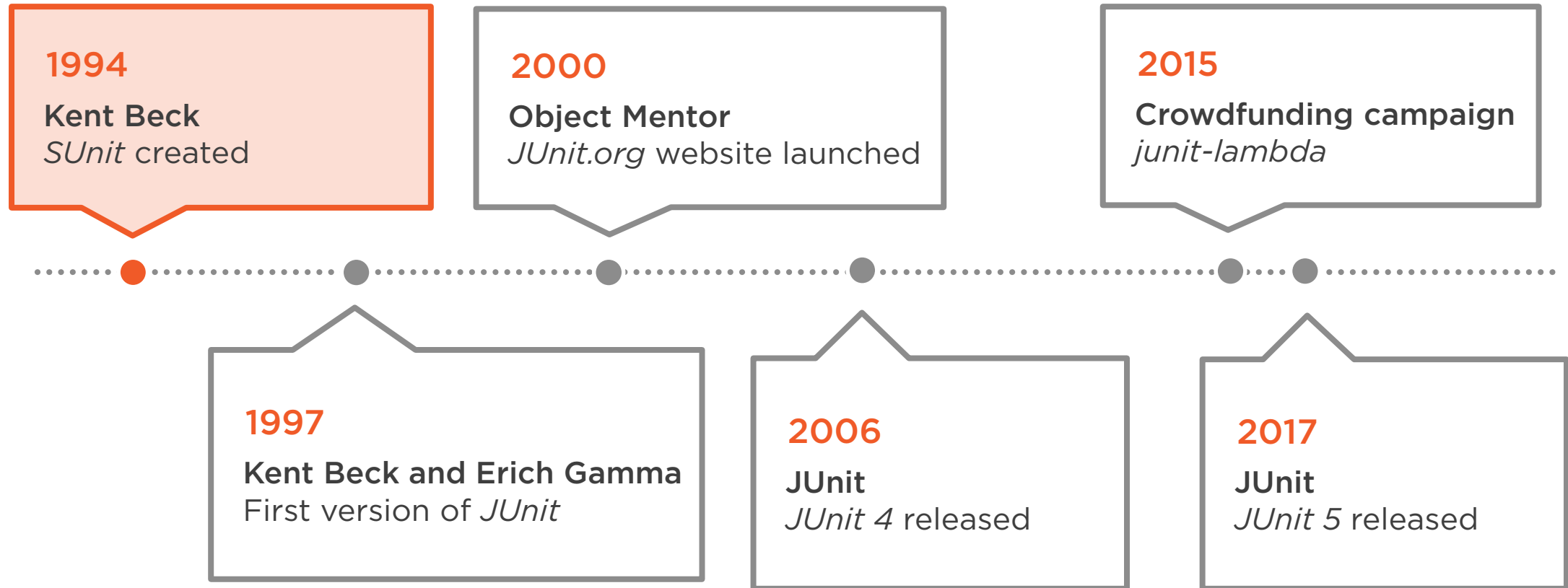
junit.org



How JUnit Works



JUnit Timeline



JUnit 5

JUnit Platform

JUnit Jupiter

JUnit Vintage



JUnit 5 Architecture



The Problem

junit.jar

A single JAR file

Used by everyone

No flexible API



Tools provided great support



But locked development in



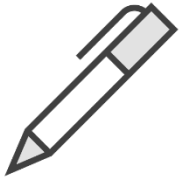
“The success of JUnit as a platform prevents the development of JUnit as a tool.”

Johannes Link

<http://bit.ly/interviewjlink>



Separation of Concerns



An API to write tests



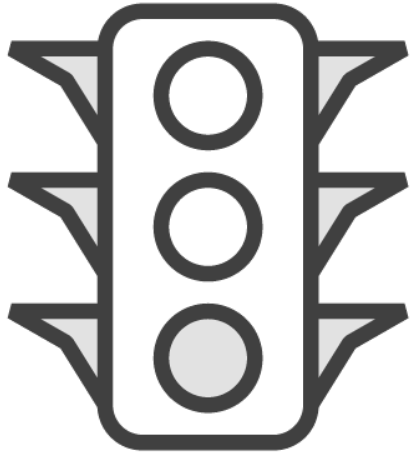
A mechanism to discover and run tests



An API to run tests (for tools)



Design Goals



Decouple tests

Preference for extension points

Support Java 8 features

JUnit 5

JUnit Platform

JUnit Jupiter

JUnit Vintage



JUnit Platform

junit-platform-console

junit-platform-engine

junit-platform-
launcher

junit-platform-runner

junit-platform-gradle-
plugin

junit-platform-
surefire-provider



JUnit Jupiter

junit-jupiter-api

junit-jupiter-engine

junit-jupiter-params

junit-jupiter-migrationsupport



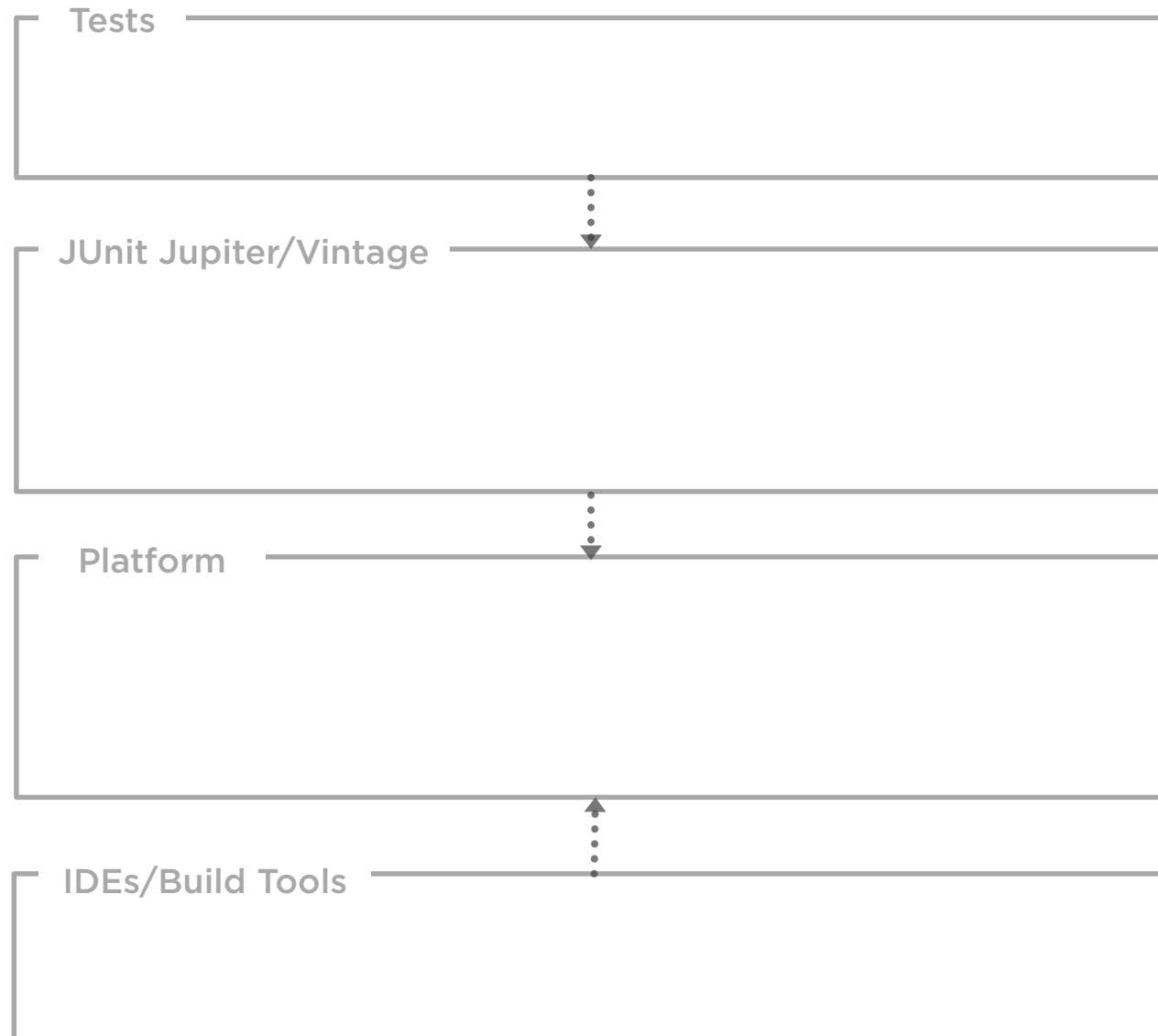
JUnit Vintage

junit-vintage-engine

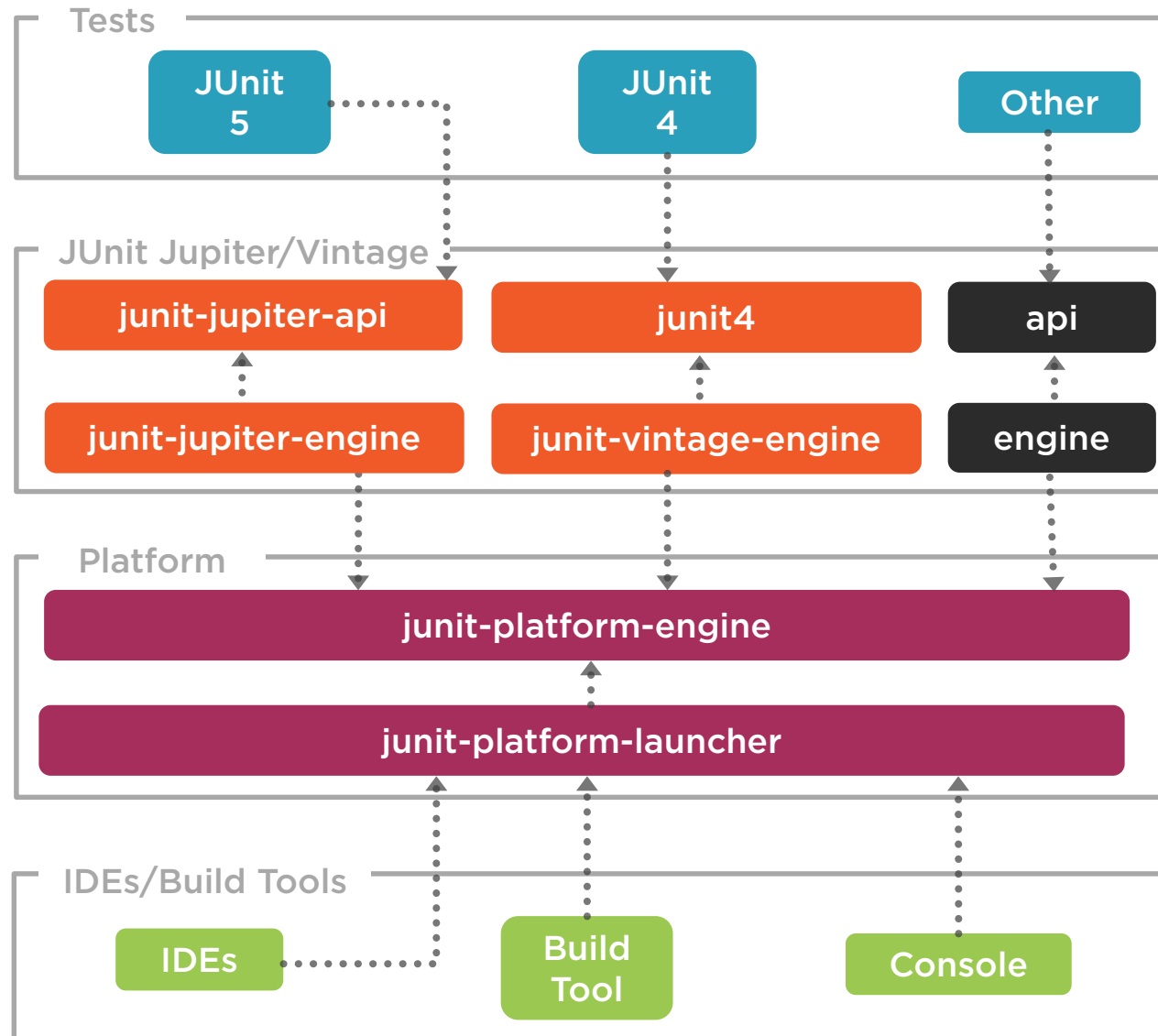
JUnit 3/4



Architecture



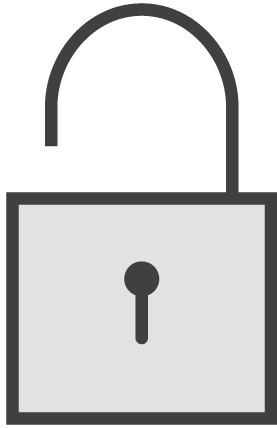
Architecture



The JUnit platform is
available to everybody



Opening up the Platform Means



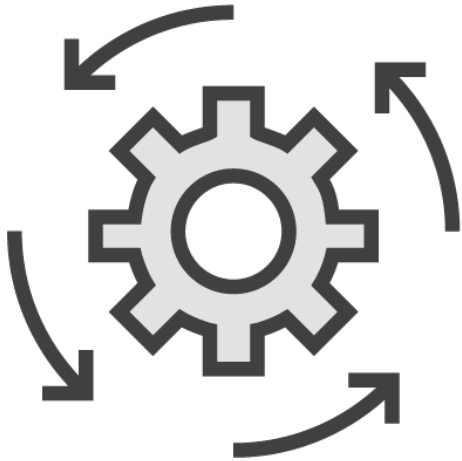
Full tool support

- Other frameworks
- New frameworks

IDEs and Build Tool Support



JUnit 5 Test API



Group ID: org.junit.jupiter

Artifact ID: junit-jupiter-api

Version: 5.0.1

First Test

```
import org.junit.jupiter.api.Test;
```

```
class HelloWorldTest {
```

```
    @Test
```

```
    void firstTest() {
```

```
        System.out.println("First test");
```

```
    }
```

```
}
```



IDE Support

IntelliJ IDEA

Eclipse



Build Tool Support

Gradle

Maven



Setting up JUnit with Gradle



Demo



Setting up JUnit - Gradle



Setting up JUnit with Maven



Demo



Setting up JUnit - Maven



IDEs Without Support



Two Options



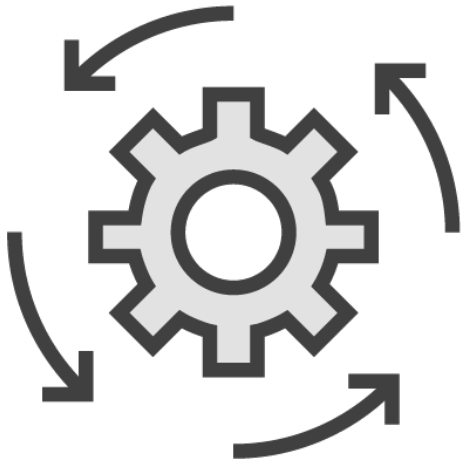
Command line

JUnit 4 runner

- JUnitPlatform



JUnit 5 Platform Runner

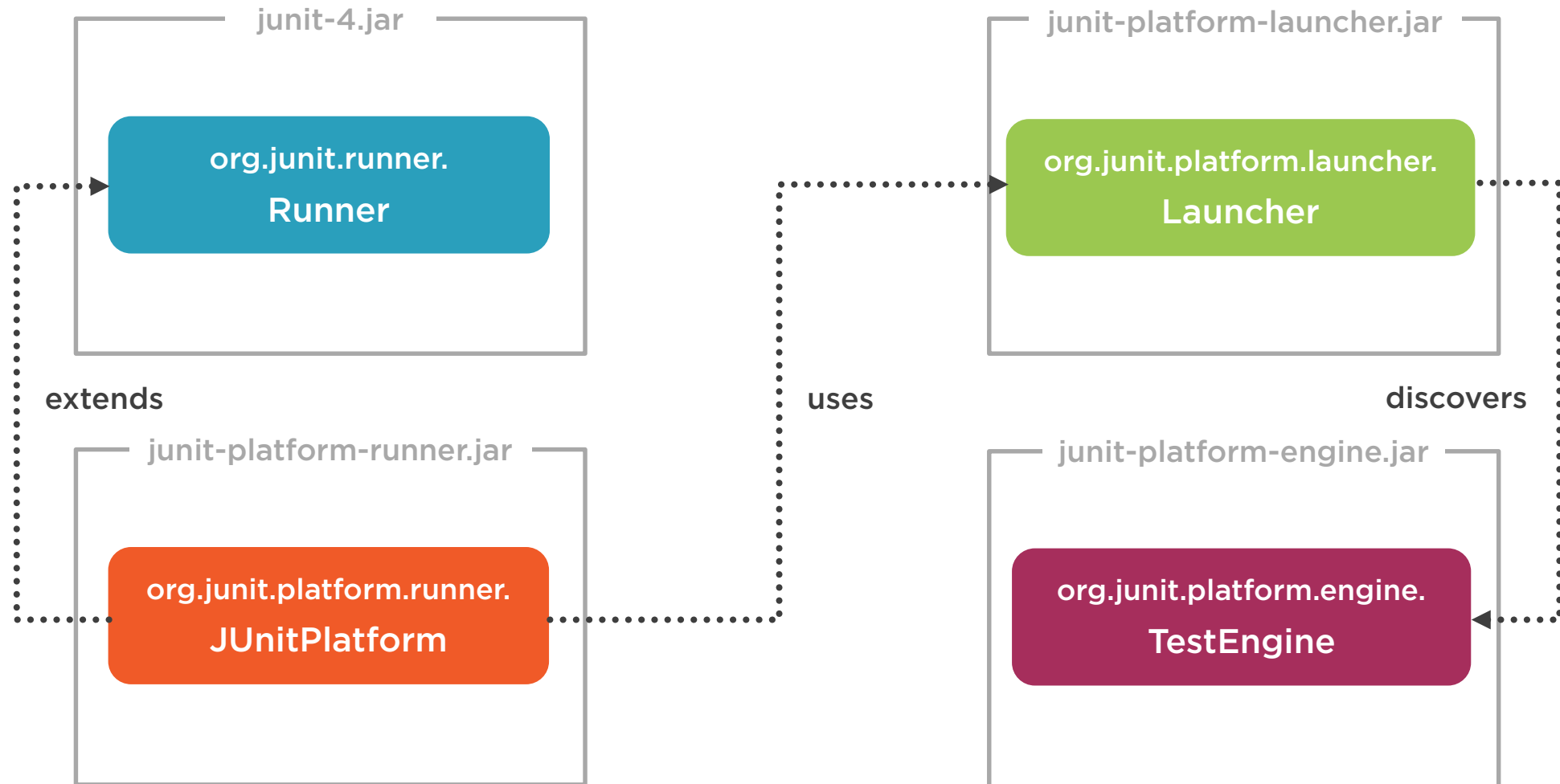


Group ID: org.junit.platform

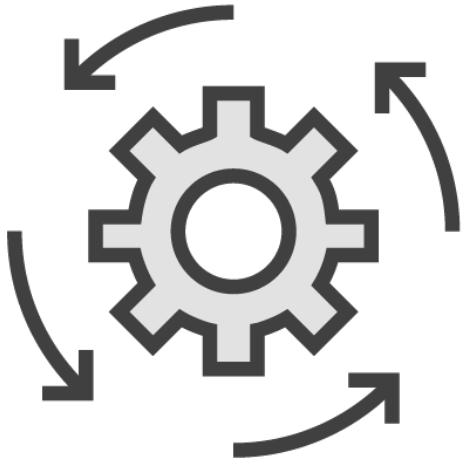
Artifact ID: junit-platform-runner

Version: 1.0.1

How JUnitPlatform Works



JUnit 5 Engine



Group ID: org.junit.jupiter

Artifact ID: junit-jupiter-engine

Version: 5.0.1

Demo



Setting up JUnit - NetBeans



Create a Test Suite

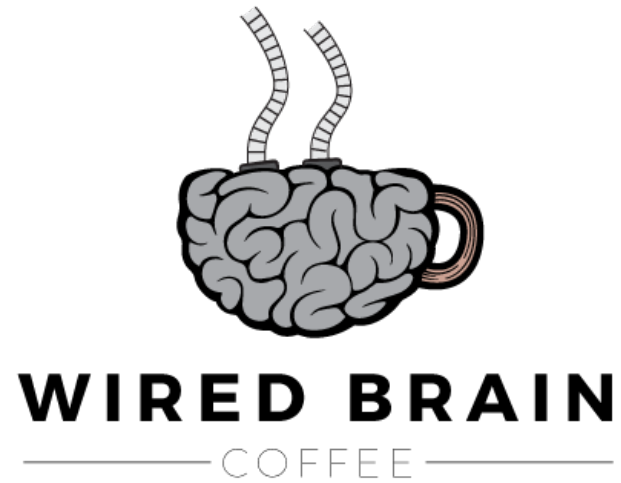
```
import org.junit.platform.runner.JUnitPlatform;
import org.junit.platform.runner.SelectPackages;
import org.junit.runner.RunWith;

@RunWith(JUnitPlatform.class)
@SelectPackages( {"my.package" })
public class TestWithJUnit5 { }
```



Course Scenario





Loyalty Program

- Every order generates points
- Rewards
 - Conversion
 - Discount
 - Gift



Summary



Types of tests

Why unit tests are important

What's JUnit?

- How it works
- History

JUnit 5 architecture

IDE and build tool support

- IntelliJ IDEA and Eclipse
- Gradle and Maven
- IDEs without support

