Migrating from JUnit 4

Esteban Herrera
JAVA ARCHITECT

@eh3rrera www.eherrera.net

Overview

Differences between JUnit 4 and 5

Running JUnit 4 tests in JUnit 5

Rule support in JUnit 5

Differences Between JUnit 4 and 5

org.junit.jupiter

All classes and annotations are now under this package

```
public class Test {
   @Test
   public void myTest() {
      ...
   }
}
```

Public Classes/Methods

```
class Test {
  @Test
  void myTest() {
    ...
  }
}
```

No More Public Classes/Methods

```
@BeforeClass
public static void setUpOnce() { /* ... */ }
@Before
public void setUp() { /* ... */ }
@After
public void teardown() { /* ... */ }
@AfterClass
public static void tearDownOnce() { /* ... */ }
```

Lifecycle Annotations

```
@BeforeClass
static void setUpOnce() { /* ... */ }
@Before
void setUp() { /* ... */ }
@After
void teardown() { /* ... */ }
@AfterClass
static void tearDownOnce() { /* ... */ }
```

Lifecycle Annotations

```
@BeforeAll
static void setUpOnce() { /* ... */ }
@BeforeEach
void setUp() { /* ... */ }
@AfterEach
void teardown() { /* ... */ }
@AfterAll
static void tearDownOnce() { /* ... */ }
```

Lifecycle Annotations

```
@BeforeAll
void setUpOnce() { /* ... */ }
@BeforeEach
void setUp() { /* ... */ }
@AfterEach
void teardown() { /* ... */ }
@AfterAll
void tearDownOnce() { /* ... */ }
```

Lifecycle Annotations

@TestInstance(Lifecycle.PER_CLASS)

```
@Test
@Ignore
public void myTest() {
   ...
}
```

Ignore Annotation

```
@Test
@Disable
void myTest() {
   ...
}
```

Disable Annotation

```
@Test
@Category({ Version1.class, Important.class })
void myTest() {
   ...
}
```

Categories

```
@Test
@Tag("Version1")
@Tag("Important")
void myTest() {
   ...
}
```

Tags

`org.junit.Assert`

Assertions Package

`org.junit.jupiter.api.Assertions`

Assertions Package

```
assertEquals("Error message", expected, actual)
```

Parameter Order

```
assertEquals(expected, actual, "Error message")
```

Parameter Order

```
assertEquals(expected, actual, () -> "Error message")
```

Lazy Strings

`org.junit.Assert.assertThat`

assertThat

~~`org.junit.Assert.assertThat`~~

assertThat Is Gone

Use it directly from Hamcrest

```
@Rule
ErrorCollector collector = new ErrorCollector();
@Test
public void myTest() {
  collector.checkThat("aa", equalTo("a"));
  collector.checkThat(1, equalTo(11));
}
```

Continue Test Execution After Failure

```
@Test
public void myTest() {
  collector.checkThat("aa", equalTo("a"));
  collector.checkThat(1, equalTo(11));
}
```

Continue Test Execution After Failure

```
@Test
void myTest() {
  collector.checkThat("aa", equalTo("a"));
  collector.checkThat(1, equalTo(11));
}
```

Continue Test Execution After Failure

```
@Test
void myTest() {
  assertAll(
  );
  collector.checkThat("aa", equalTo("a"));
  collector.checkThat(1, equalTo(11));
}
```

Continue Test Execution After Failure

```
@Test
void myTest() {
  assertAll(
    () -> assertThat("aa", equalTo("a")),
    () -> assertThat(1, equalTo(11))
  );
}
```

Continue Test Execution After Failure

Timeout in JUnit 4 (One Method Version)

```
@Test(timeout=1000)
public void testWithTimeout() {
   // ...
}
```

Timeout in JUnit 4 (All Methods Version)

```
@Rule
public Timeout globalTimeout = Timeout.seconds(10);

@Test
public void longTest() {
  // ...
}
```

Timeout in JUnit 5

```
@Test
public void longTest() {
  assertTimeout(ofSeconds(10), () -> {
    // ...
  }, "The longTest method takes more than 10 seconds");
}
```

Timeout in JUnit 5 (Preemptively Version)

```
@Test
public void longTest() {
  assertTimeoutPreemptively(ofSeconds(10), () -> {
    // ...
  }, "The longTest method takes more than 10 seconds, aborted");
}
```

Exception Testing in JUnit 4 (Try-catch Version)

```
@Test
public void catchTheException() {
  try {
    // Code that may throw an exception
    fail("Shouldn't get here");
  } catch(RuntimeException e) {
    // Assert something about the exception
  }
}
```

Exception Testing in JUnit 4 (Annotation Version)

```java
@Test(expected = RuntimeException.class)
public void annotationBasedApproach() {
  // Code that may throw an exception
}
```

Exception Testing in JUnit 4 (Rule Version)

```
@Rule
ExpectedException thrown = ExpectedException.none();

@Test
public void ruleBasedApproach() {
  thrown.expect(RuntimeException.class);
  thrown.expectMessage(containsString("..."));
  // Code that may throw an exception
}
```

Exception Testing in JUnit 5

```
@Test
void newAssertThrows() {
  assertThrows(RuntimeException.class, () -> {
    // Code that may throw an exception
  });
}
```

Exception Testing in JUnit 5

```
@Test
void newAssertThrows() {
  RuntimeException e = assertThrows(RuntimeException.class, () -> {
    // Code that may throw an exception
  });
  assertEquals("...", e.getMessage());
}
```

```
@RunWith(Suite.class)
public class MyTest {
  // ...
}
```

Extension Mode in Junit 4

@RunWith

```
@Rule
public final TemporaryFolder folder = new TemporaryFolder();


@ClassRule
public static final ExternalResource resource =
  new ExternalResource() {
      // ...
}
```

Extension Mode in Junit 4

@Rules

@ClassRule

```
class MyExtension implements BeforeTestExecutionCallback {
  // ...
}

@ExtendWith(MyExtension.class)
void myTest() { /* ... */ }
```

Extension Model in JUnit 5

@ExtendWith

New Features

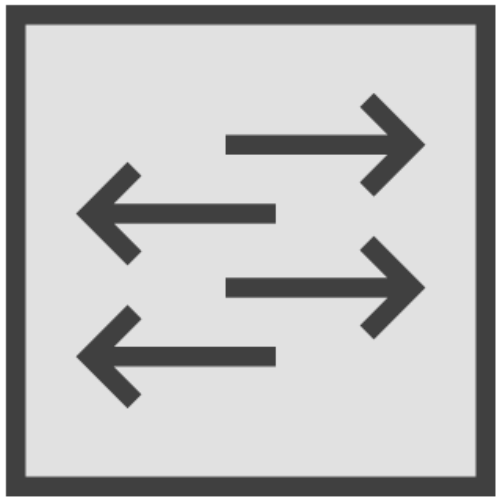
- Nested tests
- Custom display names
- Java 8 support
- Parameter injection
- Dynamic and parameterized tests
- Meta-annotations

Don’t migrate all your tests to JUnit 5 at once

Do it gradually

Running JUnit 4 Tests in JUnit 5

Compatibility



Backward compatibility

- JUnit Vintage engine

Forward compatibility

- JUnitPlatformRunner

Gradual migration to the Jupiter API

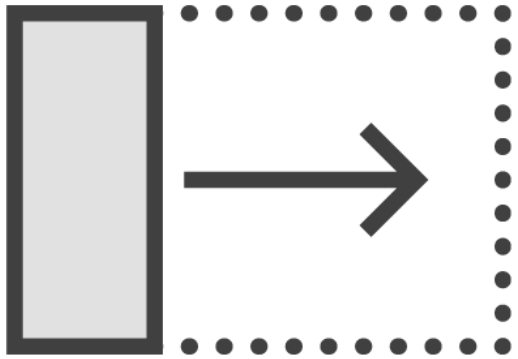
Demo

Running JUnit 4 tests in JUnit 5

- JUnit Vintage

Rule Support in JUnit 5

JUnit 4 Extension Mechanism



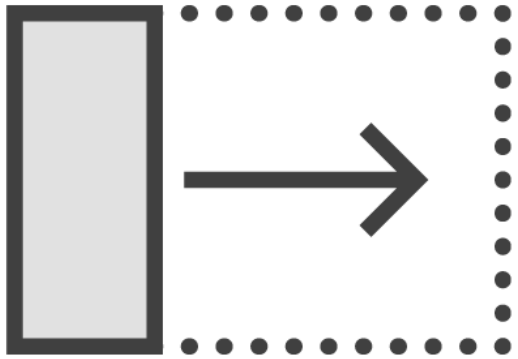
Runners

- @RunWith

Rules

- Public, non-static field
- Subtype of TestRule
- @Rule

In JUnit5...

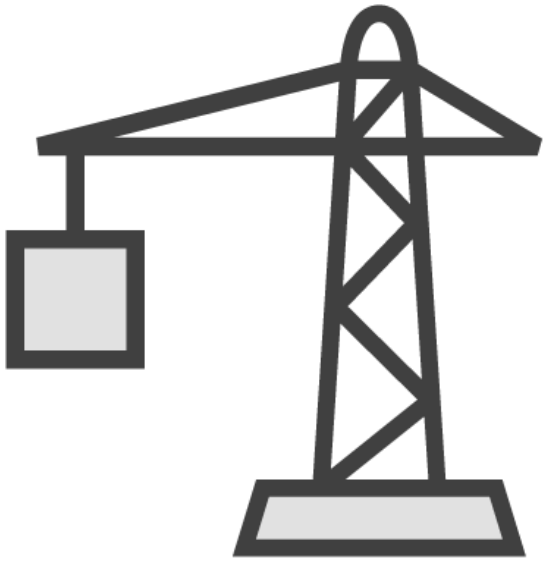


Extension points

Runners can be implemented as extensions

Limited Rule support

JUnit 5 Rule Support



ExternalResource

- TemporaryFolder

Verifier

- ErrorCollector

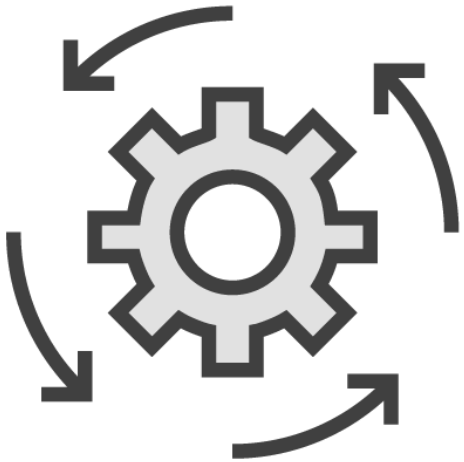
ExpectedException

```
@Target(value=TYPE)
@Retention(value=RUNTIME)
@API(status=EXPERIMENTAL, since="5.0")
@ExtendWith(value=ExternalResourceSupport.class)
@ExtendWith(value=VerifierSupport.class)
@ExtendWith(value=ExpectedExceptionSupport.class)
public @interface EnableRuleMigrationSupport { /* ... */ }
```

Annotation to Enable Support

@EnableRuleMigrationSupport

JUnit 5 Migration Support API



Group ID: org.junit.jupiter

Artifact ID: junit-jupiter-migrationsupport

Version: 5.0.1

Demo

Rule support in JUnit 5

- ErrorCollector

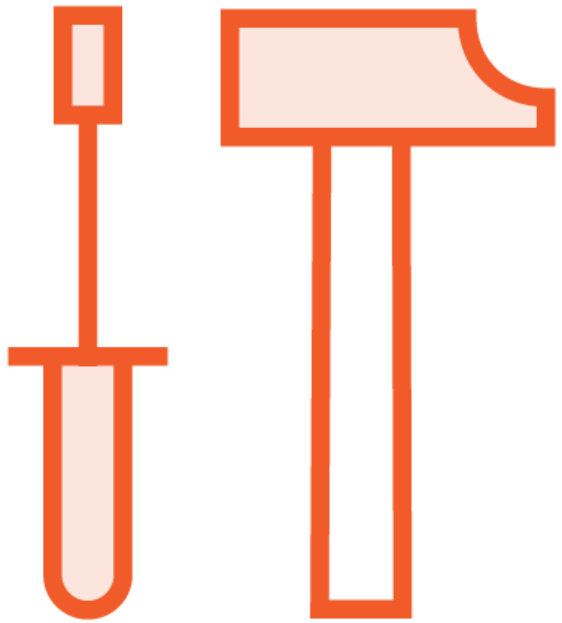
Summary

Differences between JUnit 4 and 5

Running JUnit 4 tests in JUnit 5

Rule support in JUnit 5

JUnit Is...



Thank you