Creating Dynamic and Parameterized Tests

Esteban Herrera
JAVA ARCHITECT

@eh3rrera   www.eherrera.net

Overview

Dynamic Tests

- @TestFactory
- Sources

Parameterized Tests

- Setup
- Argument sources
- Argument conversion

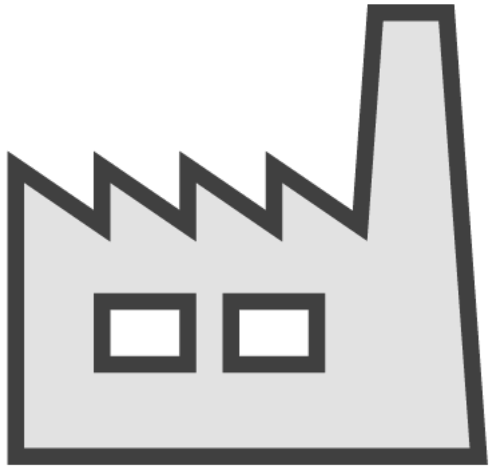
Dynamic Tests

```java
@Test
void testRewardProgram() {
  // ...
  assertEquals(type, reward.getType());
}
```

```java
@Test
void testRewardProgram() {
  // ...
  List<TestData> list = createTestData();
  for(TestData data : list) {
    // ...
    assertEquals(type, reward.getType());
  }
}
```

```java
@RepeatedTest(10)
void testRewardProgram(RepetitionInfo repetitionInfo) {

  // ...

  TestData data = list.get(
       repetitionInfo.getCurrentRepetition() - 1
  );

  // ...

  assertEquals(type, reward.getType());

}
```

Dynamic Tests



@TestFactory

- Factory of tests

No private or static methods

Experimental API

Sources

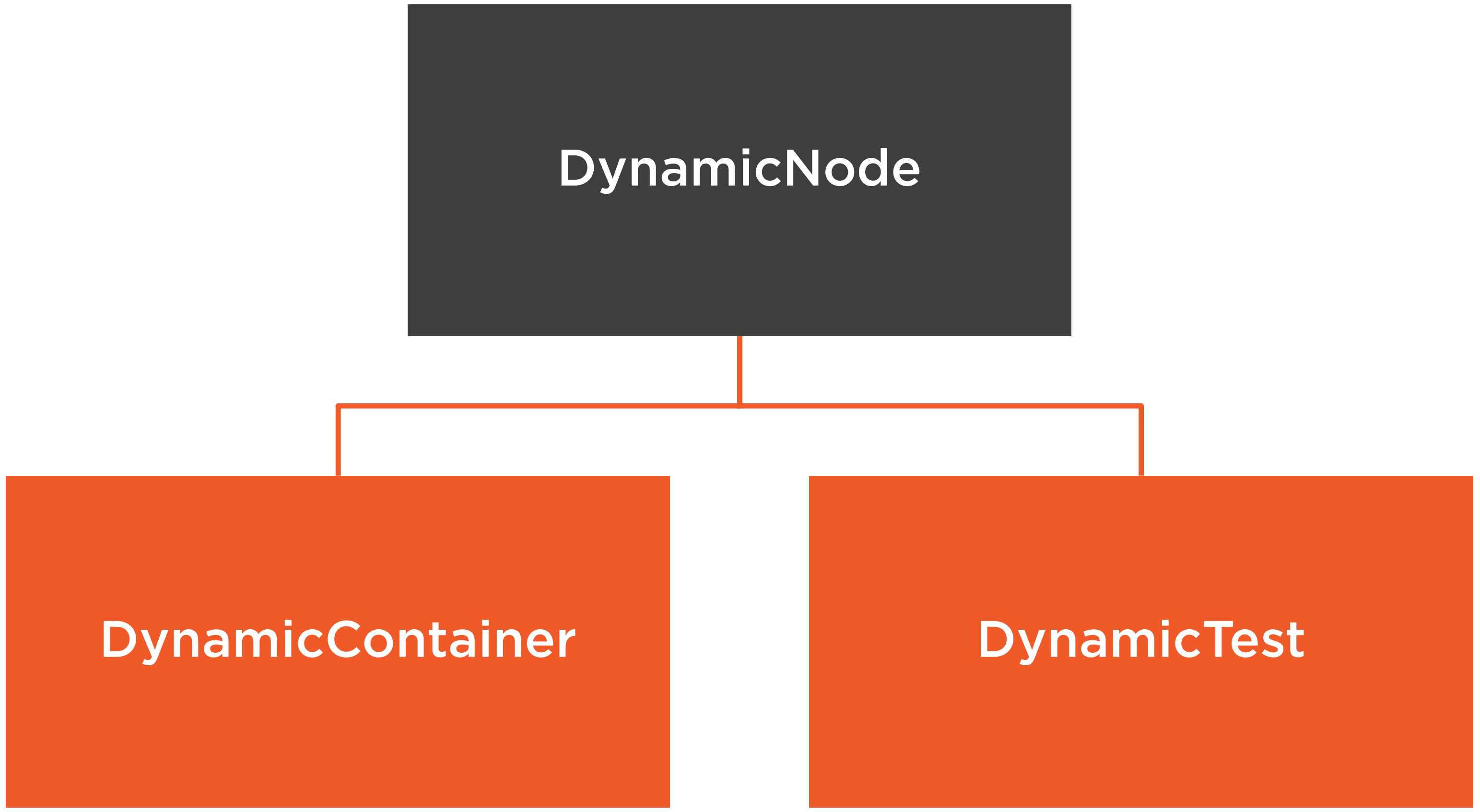
Collection

Iterable

Iterator

Stream

DynamicNode Subclasses



DynamicContainer

Display name

Iterable (or Stream)

- DynamicNodes
  - DynamicContainer
  - DynamicTest

DynamicTest

Display name

Executable

- Functional interface

@BeforeEach and **@AfterEach** methods are not executed for each dynamic test

Demo

Dynamic Tests

- Lifecycle
- DynamicTest
- DynamicContainer

Parameterized Tests

Parameterized Tests

@ParameterizedTest

- Like regular test
- Declare at least one source

Experimental API

Dependency

Group ID: org.junit.jupiter

Artifact ID: junit-jupiter-params

Version: 5.0.1

```
{index}

{arguments}

{0}, {1}, ...
```

Custom Display Name

@ParameterizedTest placeholders

Parameter Injection

@ParameterizedTest parameters

- Can’t be injected into lifecycle methods

Test information parameters

- TestInfo
- TestReporter
- After parameters injected to the test

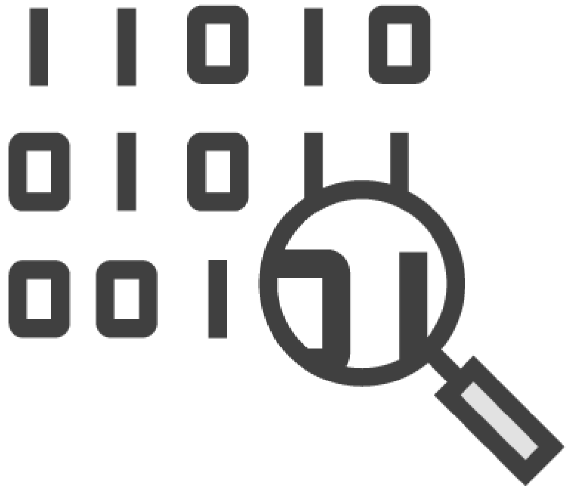
Demo

Parameterized Tests

- Setup
- Lifecycle
- Custom display name
- Inject test information parameters

Argument Sources

Sources Rules



At least one source

Provide values for all parameters

One execution for each group of arguments

@ValueSource

Arrays of type:

- String
- int
- long
- double

For single parameter methods

@EnumSource

Values of an enum

Optional parameters:

- names
- mode

For single parameter methods

@MethodSource

Refer one or more methods

For tests with a single parameter:

- Return a Stream of parameter type
- Return a Stream of primitive types

For tests with multiple parameters:

- Return a Stream, Iterable, Iterator, or array of type Arguments

The methods used:

- Must be static
    - Unless you're using @TestInstance(Lifecycle.PER_CLASS)
- And optionally private

@CsvSource

Comma-separated String literals

Parameter:

- Delimiter

Uses a single quote (') as quote character

@CsvFileSource

CSV files from classpath

Parameters:

- Encoding
- Line separator
- Delimiter

Each line results in one invocation

Uses a double quote (") as quote character

@ArgumentsSource

For custom sources

Use an ArgumentsProvider implementation

```java
interface ArgumentsProvider {
    Stream<? extends Arguments>
        provideArguments(ExtensionContext context)
            throws Exception;
}
```

@ArgumentsSource

Demo

Argument Sources

Argument Conversion

Sources

@ValueSource

@CsvSource

@CsvFileSource

Implicit Conversion

String to:

- Primitive values
- Enum
- java.time classes

```
interface ArgumentConverter {
    Object convert(Object source, ParameterContext context)
            throws ArgumentConversionException;
}
```

Custom Converter

@ConvertWith

```java
abstract class SimpleArgumentConverter implements ArgumentConverter {
   protected abstract Object convert(Object source,
                                     Class<?> targetType)
                              throws ArgumentConversionException;
   // ...
}
```

SimpleArgumentConverter

Demo

Custom Converter

Summary

Dynamic Tests

- @TestFactory
- No support for lifecycle methods
- Collection, Iterable, Iterator, Stream
- DynamicContainer and DynamicTest

Parameterized Tests

- Support for lifecycle methods
- Dependency junit-jupiter-params
- Multiple sources
- Custom display name
- Custom converters