



Test unitaire avec Scala

Scala Enthusiastic Lausanne

Agenda

- Valeurs du Framework
- Code a Tester
- Frameworks Sélectionnés
- Tests d'intégration
- Atelier

Valeurs du Framework

- Intégration avec Maven (Build et Rapport)
- Intégration avec IntelliJ/Eclipse
- Lisibilité du code

Code à Tester

Test de la collection "Stack[Int]" (mutable)

A Tester:

"push" / "pop"	Ordre OK
"sum"	Sommation OK
"pop" d'une stack vide	"NoSuchElementException"

Framework Sélectionné

Choix des framework les plus courant

- jUnit
- TestNG
- ScalaTest
- Specs

jUnit

- Ajout la dépendance jUnit à Maven

```
<dependency>  
  <groupId>junit</groupId>  
  <artifactId>junit</artifactId>  
  <version>4.8.2</version>  
</dependency>
```

- Ajout annotation @org.junit.Test

```
@Test  
def testSum() = {
```

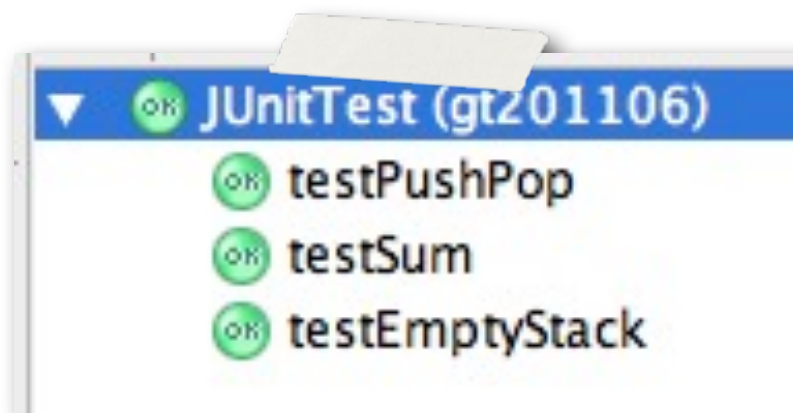
jUnit

```
@Test
def testPushPop() = {

    val stack = new Stack[Int]
    stack.push(1)
    stack.push(2)

    assertEquals(stack.pop(), 2)
    assertEquals(stack.pop(), 1)
}
```


jUnit



TestNG

- Ajout la dépendance TestNG à Maven

```
<dependency>  
  <groupId>org.testng</groupId>  
  <artifactId>testng</artifactId>  
  <version>5.14</version>  
  <scope>test</scope>  
</dependency>
```

- Ajout annotation `@org.testng.annotations.Test`

```
@Test  
def testSum() = {
```

- Nom de la classe doit respecter `*Test` (Maven)

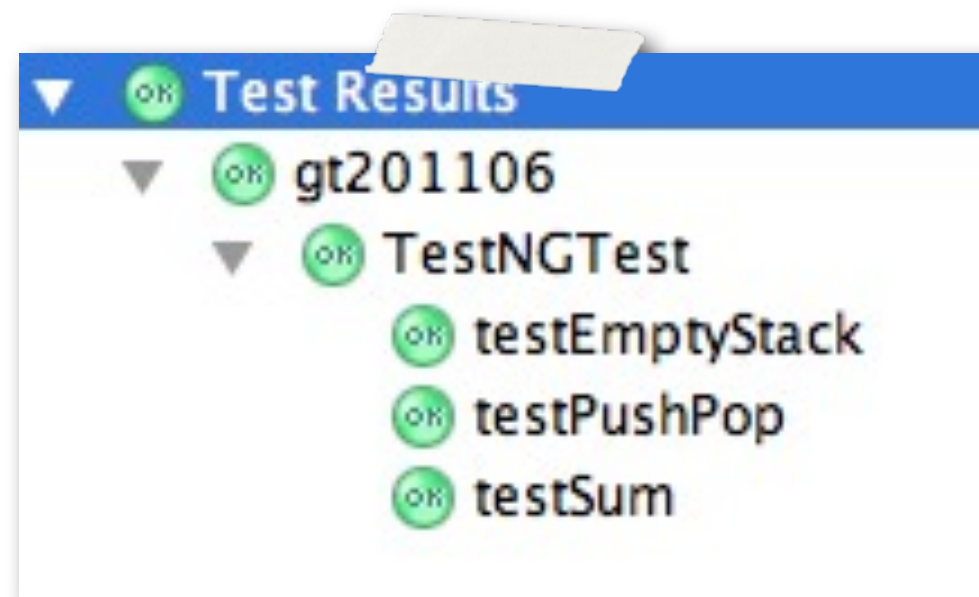
TestNG

```
@Test
def testPushPop() = {

    val stack = new Stack[Int]
    stack.push(1)
    stack.push(2)

    assertEquals(stack.pop(), 2)
    assertEquals(stack.pop(), 1)
}
```

TestNG



ScalaTest (Spec)

- Ajout la dépendance ScalaTest à Maven

```
<dependency>  
  <groupId>org.scalatest</groupId>  
  <artifactId>scalatest</artifactId>  
  <version>1.3</version>  
  <scope>test</scope>  
</dependency>
```

- Extension de Spec et ShouldMatcher

```
class ScalaTestSpecTest extends Spec with ShouldMatchers {
```

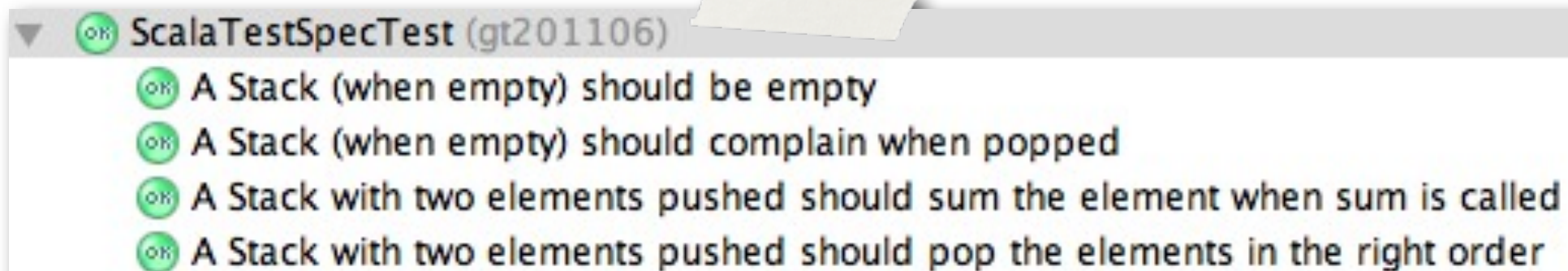
- Pour que Maven tient compte des tests

```
@RunWith(classOf[JUnitRunner])
```

ScalaTest (Spec)

```
describe("with two elements pushed") {  
  
  val stack = new Stack[Int]  
  stack.push(1)  
  stack.push(2)  
  
  it("should sum the element when sum is called") {  
    stack.sum should be(3)  
  }  
}
```

ScalaTest (Spec)

- 
- ScalaTestSpecTest (gt201106)
- OK A Stack (when empty) should be empty
 - OK A Stack (when empty) should complain when popped
 - OK A Stack with two elements pushed should sum the element when sum is called
 - OK A Stack with two elements pushed should pop the elements in the right order

ScalaTest (WordSpec)

- Ajout la dépendance ScalaTest à Maven

```
<dependency>  
  <groupId>org.scalatest</groupId>  
  <artifactId>scalatest</artifactId>  
  <version>1.3</version>  
  <scope>test</scope>  
</dependency>
```

- Extension de WordSpec et ShouldMatcher

```
class ScalaTestWordSpecTest extends WordSpec with ShouldMatchers {
```

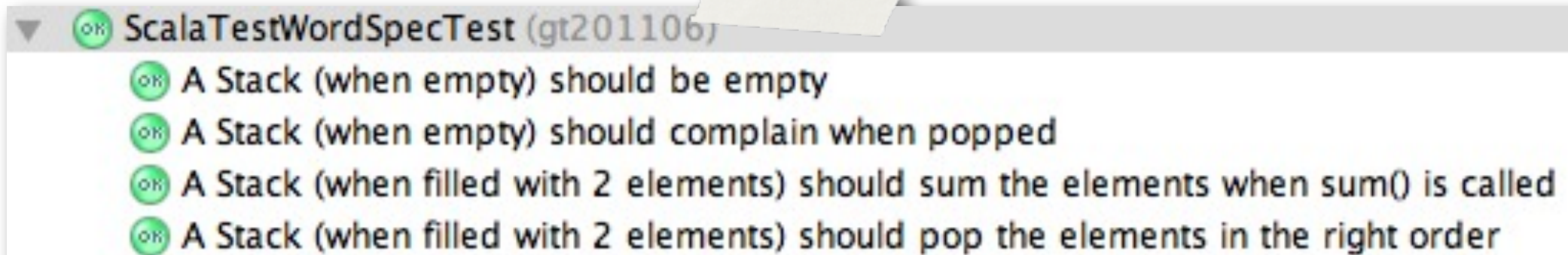
- Pour que Maven tient compte des tests

```
@RunWith(classOf[JUnitRunner])
```

ScalaTest (WordSpec)

```
"A Stack" when {  
  val stack = new Stack[Int]  
  
  "filled with 2 elements" should {  
    stack.push(1)  
    stack.push(2)  
  
    "sum the elements when sum() is called" in {  
      stack.sum should be(3)  
    }  
  
    "pop the elements in the right order" in {  
      stack.pop should be(2)  
      stack.pop should be(1)  
    }  
  }  
}
```

ScalaTest (WordSpec)

A screenshot of a ScalaTest WordSpecTest result. The title bar shows a green 'OK' icon and the text 'ScalaTestWordSpecTest (gt201106)'. Below the title bar, there are four test cases, each preceded by a green 'OK' icon. The test cases are: 'A Stack (when empty) should be empty', 'A Stack (when empty) should complain when popped', 'A Stack (when filled with 2 elements) should sum the elements when sum() is called', and 'A Stack (when filled with 2 elements) should pop the elements in the right order'.

ScalaTestWordSpecTest (gt201106)

- OK A Stack (when empty) should be empty
- OK A Stack (when empty) should complain when popped
- OK A Stack (when filled with 2 elements) should sum the elements when sum() is called
- OK A Stack (when filled with 2 elements) should pop the elements in the right order

ScalaTest (FlatSpec)

- Ajout la dépendance ScalaTest à Maven

```
<dependency>  
  <groupId>org.scalatest</groupId>  
  <artifactId>scalatest</artifactId>  
  <version>1.3</version>  
  <scope>test</scope>  
</dependency>
```

- Extension de FlatSpec et ShouldMatcher

```
class ScalaTestFlatSpecTest extends FlatSpec with ShouldMatchers {
```

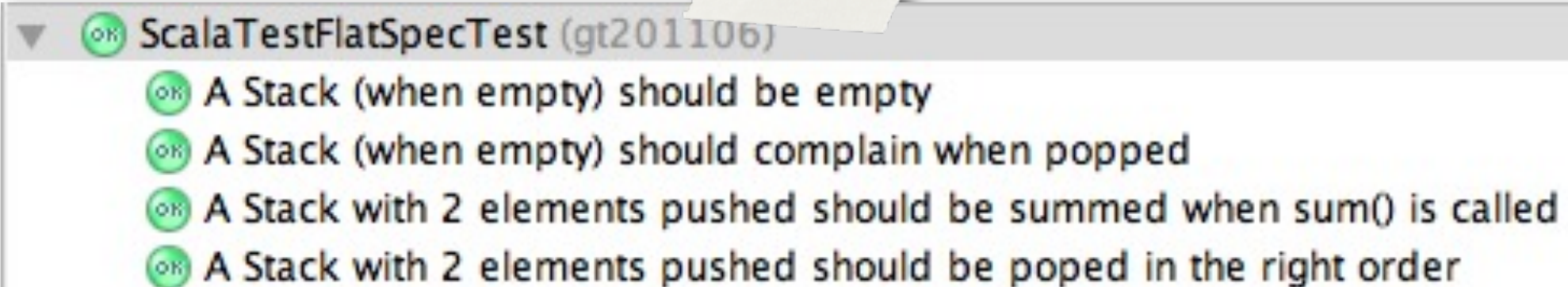
- Pour que Maven tient compte des tests

```
@RunWith(classOf[JUnitRunner])
```

ScalaTest (FlatSpec)

```
"A Stack with 2 elements pushed" should "be summed when sum() is called" in {  
  stack.push(1)  
  stack.push(2)  
  stack.sum should be(3)  
}  
  
it should "be popped in the right order" in {  
  stack.pop() should be(2)  
  stack.pop() should be(1)  
}
```

ScalaTest (FlatSpec)

- 
- A screenshot of a test runner interface. At the top, there is a header bar with a dropdown arrow and a green circle containing 'OK'. Below this, the text 'ScalaTestFlatSpecTest (gt201106)' is displayed. A small white rectangular sticker is placed over the top right corner of the test results box. Below the header, there is a list of four test cases, each preceded by a green circle containing 'OK'.
- OK ScalaTestFlatSpecTest (gt201106)
 - OK A Stack (when empty) should be empty
 - OK A Stack (when empty) should complain when popped
 - OK A Stack with 2 elements pushed should be summed when sum() is called
 - OK A Stack with 2 elements pushed should be popped in the right order

Specs

- Ajout la dépendance Specs à Maven

```
<dependency>  
  <groupId>org.scala-tools.testing</groupId>  
  <artifactId>specs_2.8.1</artifactId>  
  <version>1.6.7</version>  
  <scope>test</scope>  
</dependency>
```

- Extension de SpecificationWithJUnit

```
class SpecsTest extends SpecificationWithJUnit("Stack") {
```








- Pour que Maven tient compte des tests

```
@RunWith(classOf [JUnitRunner])
```

Specs

```
"A Stack with 2 elements pushed" should {  
  val stack = new Stack[Int]  
  stack.push(1)  
  stack.push(2)  
  
  "be summed when sum() is called" in {  
    stack.sum mustEqual 3  
  }  
  
  "be popped in the right order" in {  
    stack.pop mustEqual 2  
    stack.pop mustEqual 1  
  }  
}
```

Specs

- ▼  SpecsTest (gt201106)
 - ▼  A new stack should
 -  Throw an Exception when pop is called (org.specs.runner.ExampleTestCase)
 -  be empty (org.specs.runner.ExampleTestCase)
 - ▼  A Stack with 2 elements pushed should
 -  be summed when sum() is called (org.specs.runner.ExampleTestCase)
 -  be popped in the right order (org.specs.runner.ExampleTestCase)

Tests d'intégration

- SodaTest

Spreadsheet-Driven Integration Testing

<https://github.com/GrahamLea/SodaTest>

Atelier

Coquille du projet:
<https://github.com/ceki/segl>

ScalaTest :
<http://www.scalatest.org/>

Specs :
<http://code.google.com/p/specs/>