Smarter Testing With Spock

Peter Niederwieser
Principal Engineer, Gradleware
What we’ll talk about

- Spock?!  
- State Based Testing  
- Data Driven Testing  
- Interaction Based Testing  
- Spock Extensions  
- More Cool Stuff
SPOCK?!
Spock is...

- A developer testing framework...
- for Groovy and Java applications...
- based on Groovy...
- fully compatible with JUnit...
- but going beyond!
Spock Can...

- Reduce the lines of test code
- Make tests more readable
- Turn tests into specifications
- Be extended in powerful ways
- Bring back the fun to testing!
Getting Started

- Homepage
  http://spockframework.org

- Source Code
  https://github.com/spockframework/spock

- Spock Web Console
  http://meet.spockframework.org

- Spock Example Project
  http://downloads.spockframework.org
  https://github.com/spockframework/spock/tree/groovy-1.8/spock-example

- Slides and Code for this Presentation
  https://github.com/spockframework/smarter-testing-with-spock
Who’s Using Spock?

Gradle  GPars

Grails  Geb

Grails Plugin Collective

Apache Tapestry

Griffon  Spock

Spring?
Who’s Using Spock? (2)

be2
CTI Digital
Energized Work
SystemsForge

eHarmony
Smarter Ecommerce
IntelliGrape Software

BSkyB
bemoko

BSkyB
bemoko

Gennemtænkt IT
State Based Testing
State Based Testing

- Classical Unit Testing
  - Arrange
  - Act
  - Assert
- Given-When-Then
State Based Testing

- Classical Unit Testing
  - Arrange
  - Act
  - Assert
- Given-When-Then

Show me the code!
Recap: State Based Testing

- Blocks
  setup: cleanup: expect: given: when: then: where: and:

- Fixture Methods
  setup() cleanup() setupSpec() cleanupSpec()

- Instance and @Shared fields

- old() and thrown()
Data Driven Testing
Data Driven Testing

- Test the same behavior...
- with varying data!
Data Driven Testing

- Test the same behavior...
- with varying data!

Show me the code!
Recap: Data Driven Testing

- where: block
- Data tables
- External data sources
- @Unroll
Interaction
Based Testing
Interaction Based Testing

- Design and test how your objects communicate
- Mocking frameworks to the rescue
- Spock comes with its own mocking framework
Interaction Based Testing

- Design and test how your objects communicate
- Mocking frameworks to the rescue
- Spock comes with its own mocking framework

Show me the code!
Recap: Interaction Based Testing

Creating

```python
def sub = Mock(Subscriber)
Subscriber sub = Mock()
```

Mocking

```python
1 * sub.receive("msg")
(1..3) * sub.receive(_)
(1.._) * sub.receive(_ as String)
1 * sub.receive(!null)
1 * sub.receive({it.contains("m")})
1 * _./rec.*/("msg")
```
Recap: Interaction Based Testing (2)

**Stubbing**

// now returns status code
String receive(String msg) { ... }

sub.receive(_) >> "ok"
sub.receive(_) >>> ["ok", "ok", "fail"]
sub.receive(_) >>> { msg -> msg.size() > 3 ? "ok" : "fail" }

**Mocking and Stubbing**

3 * sub.receive(_) >>> ["ok", "ok", "fail"]

**Impressing your friends**

(._._.) * _._.(_.*) >> _
Spock
Extensions
Spock Extensions

- Listeners
- Interceptors
- Annotation-driven extensions
- Global extensions
Built-in Extensions

- `@Ignore`
- `@IgnoreRest`
- `@FailsWith`
- `@Timeout`
- `@AutoCleanup`
- `@Stepwise`
- `@RevertMetaClass`
- `@Rule`
Built-in Extensions

- `@Ignore`
- `@IgnoreRest`
- `@FailsWith`
- `@Timeout`
- `@AutoCleanup`
- `@Stepwise`
- `@RevertMetaClass`
- `@Rule`

Show me the code!
External Extensions

- spock-grails
- spock-spring
- spock-guice
- spock-tapestry
- spock-unitils
- spock-griffon
- spock-arquillian

Grails Extension

- http://grails.org/plugin/spock
- https://github.com/spockframework/spock-grails

- grails install plugin spock 0.6-SNAPSHOT
- grails test-app
- grails test-app integration:spock ‘C*’
class MyUnitSpec extends UnitSpec {

def "domain mocking"() {
  setup:
  mockDomain(Person)

  when:
  new Person(name: name).save()

  then:
  Person.findByName(name) != null

  where:
  name = "bill"
}
}
class InjectionExamples extends Specification {
    @Autowired
    IService1 byType

    @Resource
    IService1 byName

    @Autowired
    ApplicationContext context
}
Other Cool Stuff
Configuring Spock

~/.spock/SpockConfig.groovy, or on class path, or with -Dspock.configuration

```groovy
runner {
  filterStackTrace false
  include Fast
  exclude Slow
  optimizeRunOrder true
}
@Fast
class MyFastSpec extends Specification {
  def "I'm fast as hell!"() { expect: true }
}
@Slow
class MySlowSpec extends Specification {
  def "sorry, can't keep up..."() {
    expect: false
  }
}
```
Tooling

- Eclipse, IDEA
- Ant, Maven, Gradle
- Jenkins, Bamboo, TeamCity
- Spock runs everywhere JUnit and Groovy run!
You write...

```plaintext
a = 1; b = 2; c = 4
expect: sum(a, b) == c
```

Spock generates...

```python
def rec = new ValueRecorder()
verifier.expect(
    rec.record(rec.record(sum(rec.record(a),
                           rec.record(b))) == rec.record(c)))
```

You see...

```
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum(a, b) == c</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>false</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Q&A

- Homepage
  http://spockframework.org

- Source Code
  https://github.com/spockframework/spock

- Spock Web Console
  http://meet.spockframework.org

- Spock Example Project
  http://downloads.spockframework.org
  https://github.com/spockframework/spock/tree/groovy-1.8/spock-example

- Slides and Code for this Presentation
  https://github.com/spockframework/smarter-testing-with-spock