Smarter Testing with Spock

Peter Niederwieser

Principal Engineer, Gradleware

Spock is...

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

- Expressive, elegant, fun!

How to Get 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-1.7/spock-example
- Code for This Presentation
  https://github.com/spockframework/spock-uberconf-2011

Who is Using Spock?

Gradle

Geb

GPars

Grails

Grails Plugin Collective

Spring*

Tapestry*

Spock

Who is Using Spock? (2)

be2

eHarmony

CTI Digital

Smarter Ecommerce

BSkyB

bemoko

Energized Work

IntelliGrape Software

SystemsForge

Software Projects

Gennemtænkt IT

State-based Testing

- AAA: Arrange - Act - Assert
- Given - When - Then

DEMO

Full demo project available here:

https://github.com/spockframework/spock-uberconf-2011

Recap: State-based Testing

- Blocks
  given, when, then, where
  setup, cleanup, and
- Fixture methods
  setup(), cleanup(), setupSpec(), cleanupSpec()
- instance and @Shared variables
- old() and thrown()
- Data-driven tests

Interaction-based Testing

- Mocking and Stubbing
- Interactions between SUT and collaborators

DEMO

Full demo project available here:

https://github.com/spockframework/spock-uberconf-2011

Recap: Interaction-based Testing

Creating:

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

Mocking:

```
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:

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

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:

```
(_.._) * _._(_) >> _
```

Extensions

- Based on Listeners and Interceptors
- Annotation-Driven Extensions
- Global Extensions

Built-in Extensions

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

@Rule Extension

```
class JUnitRules extends Specification {
    @Rule name = new TestName()

    def "retrieve test name at runtime"() {
        println "running '$name.methodName'"
        expect: 1 + 1 == 2
    }
}
```

@Stepwise Extension

```
@Stepwise
class StepwiseSpec extends Specification {
    def "step 1"() {
        expect: true
    }

    def "step 2"() {
        expect: false
    }

    def "step 3"() {
        expect: true
    }
}
```

Optional Extensions

- spock-grails
- spock-spring
- spock-guice
- spock-tapestry
- spock-unitils

- spock-griffon
- spock-arquillian
- spock-extensions
  http://github.com/robfletcher/spock-extensions

Grails Extension

http://www.grails.org/plugin/spock

```
grails install-plugin spock 0.5-SNAPSHOT
grails install-plugin spock 0.5-groovy-1.7-SNAPSHOT

grails test-app
grails test-app integration:spock 'C*'
```

Grails Extension (2)

```
class MyUnitSpec extends UnitSpec {
    def "domain mocking"() {
        setup:
        mockDomain(Person)

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

        then:
        Person.findByName(name) != null

        where:
        name = "bill"
    }
}
```

Spring Extension

```
class InjectionExamples extends Specification {
    @Autowired
    IService1 byType

    @Resource
    IService1 byName

    @Autowired
    ApplicationContext context
}
```

Writing Custom Extensions

DEMO

Full demo project available here:

https://github.com/spockframework/spock-uberconf-2011

Configuring Spock

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

```
runner {
    filterStackTrace false
    include Fast
    exclude Slow
    optimizeRunOrder true
}

@Fast
class MyFastSpec extends Specification {
    def "I’m really fast!"() { expect: true }

    @Slow
    def "sorry but I’m slow"() { expect: false }
}
```

Tool Integration

- IDEs: Eclipse, IDEA
- Build Tools: Ant, Maven, Gradle
- CI Servers: Hudson/Jenkins, Bamboo, TeamCity
- Runs everywhere JUnit and Groovy run!

Spock Under The Hood

DSL implemented as Groovy compiler plugin (AST transform)

Before:

```
expect: sum(a, b) == c
```

After:

```
def recorder = new ValueRecorder()
runtime.verify(
    record(4, record(sum(2, record(0, a), record(1, b)))
            == record(3, c)
    )
)
```

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-1.7/spock-example
- Code for This Presentation
  https://github.com/spockframework/spock-uberconf-2011