Cucumber on the JVM with Groovy

cucumber, cuke4duke, groovy, geb

Richard Paul

2011-03-24

What is Groovy

- Dynamic language for the JVM
- Inspired by Python, Ruby, Smalltalk
- Integrates closely with Java
- Superset of Java syntax

http://groovy.codehaus.org/

Why Groovy

- Leverage existing Java API knowledge
- Integrate closely with production Java code
- Expressive language & libraries
- Just a 5MB jar on the classpath

Groovy Support in Cucumber

- Cuke4Duke
- Uses JRuby behind the scenes
  - Pure Java coming later this year
- Groovy DSL to automate scenarios

https://github.com/aslakhellesoy/cuke4duke

Feature

```
Scenario: Regular numbers
    Given I have entered 3
    And I have entered 2
    When I press divide
    Then the result should be 1.5
```

https://github.com/aslakhellesoy/cuke4duke/tree/master/examples/groovy

Given

```
Given(~'I have entered (.*)') { number ->
    calculator.push number
}
```

```
// Implicit coercion to integer
Given(~'I have entered (.*)') { int number ->
    assert number.class == Integer.class
    calculator.push number
}
```

When

```
When(~'I press (\\w+)') { operatorName ->
    result = calculator."$operatorName"()
}
```

```
// Can use slashy strings to avoid escaping
When(/I press (\w+)/) { operatorName ->
    result = calculator."$operatorName"()
}
```

Then

```
Then(~'the result should be (.*)') {
        double result ->
    assert expected == result
}


// == compares equality
// not same instance as in Java
```

Power Assert

```
def now = new Date()
def old = Date.parse('yyyyMMdd', '20100101')
assert now.date == old.date
```

```
Assertion failed:

assert now.date == old.date
       |   |    |  |   |
       |   19   |  |   1
       |        |  Fri Jan 01 00:00:00 GMT 2010
       |        false
       Sat Mar 19 13:18:42 GMT 2011
```

Multiline Strings

```
Given some text
"""
Line 1
Line 2
"""
```

```
Given(~'some text') { body ->
    body.eachLine {
        println it
    }
}
```

```
=> Line 1
=> Line 2
```

Tables

```
Given I have the following foods
   |name  |healthy|
   |Orange|Yes    |
   |Chips |No     |
When I count the number of healthy items
Then I have 1 healthy item
```

Tables

```
Given(~'I have the following foods') { table ->
    basket = new Basket()
    table.hashes().each {
        def item = new Food(
            name:    it.name,  // it.get('name')
            healthy: it.healthy == 'Yes')
        basket.add(item)
    }
}
When(~'I count the healthy items') {
    numberOfHealthy = basket.numberOfHealthy
}
Then(~'I have (.+) healthy items') { int count ->
    assert numberOfHealthy == count
}
```

Tables

```
class Basket {
    private items = []
    void add(item) {
        items << item
    }
    int getNumberOfHealthy() {
        items.findAll {
            it.healthy
        }.size()
    }
}

class Food {
    def name
    def healthy
}
```

Organising Step Definitions

Cucumber will read in any .groovy files within step_definitions

All steps are then available to any scenarios

State can be shared between steps by setting to script binding

```
When(~'I set a variable to the binding') {
    x = 1
}
Then(~'the other step can access variable') {
    assert x == 1
}
```

Before/After

```
Before {
  // Initialise something
}

Before('@tagname') {
  // Initialise only for features/scenarios
  // tagged with @tagname
}
Before('~@tagname') {} // not tagged

After {
  // Clean up something
}
```

World

Allow simple access to methods from within step definitions

```
World {
    def world = new Object()
    world.metaClass.mixin Math
    world
}
```

```
When(~'we take the square root') {
    sqrt(4)    // calls Math.sqrt(4)
}
```

Browser Automation with Geb

```
Given(~'I am on the Wikipedia homepage') {
    go()
}

When(~'I search for "(.+)"'') { query ->
    $('#searchInput').value(query)
    $('.searchButton').click()
}

Then(~'I am shown the "(.+)" article') { article ->
    assert $('h1').text() == article
}
```

env.groovy for Geb

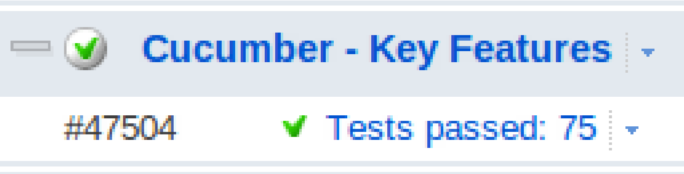
```
import geb.Browser
this.metaClass.mixin(cuke4duke.GroovyDsl)

World {
    new Browser('http://wikipedia.org')
}

After {
    clearCookies()
}
```

Build Tools

Integration with

- Maven
- Ant

Cucumber can write reports in JUnit format for CI reports



Cucumber - Key Features

#47504 ✓ Tests passed: 75

Discussion/Questions

Thanks!

http://rapaul.com **@rapaul**