

# Agile Acceptance Tests

bdd, cucumber, cuke4duke, groovy,  
selenium2, webdriver

# Behaviour Driven Development (BDD)

## **Feature**

In order to ...

As a ...

I want ...

## **Scenario**

Given ...

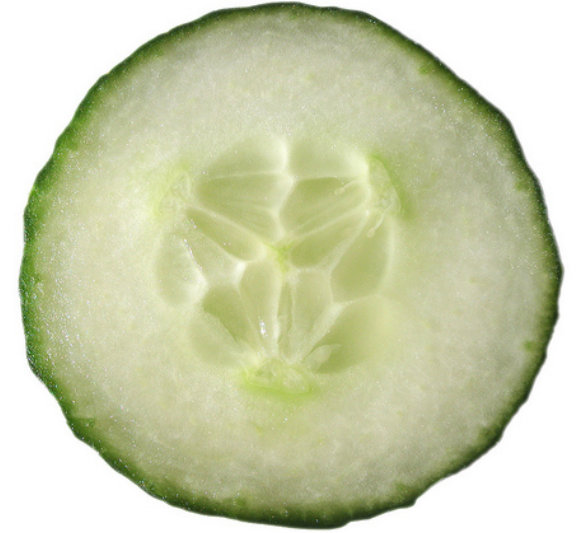
When ...

Then ...

This is just a small part of BDD directly related to acceptance testing.

# Cucumber

BDD testing framework written in Ruby



**Cucumber + JRuby + Cuke4Duke**

=

**BDD testing framework for the JVM**

Languages: **Groovy**, Java, Scala, Javascript, Ioke, Ruby

Frameworks: Spring, Guice, WebDriver, Maven, Ant

# Kick Start

```
$ git clone http://github.com/aslakhellesoy/cuke4duke.git
```

```
$ cd examples/groovy-webdriver
```

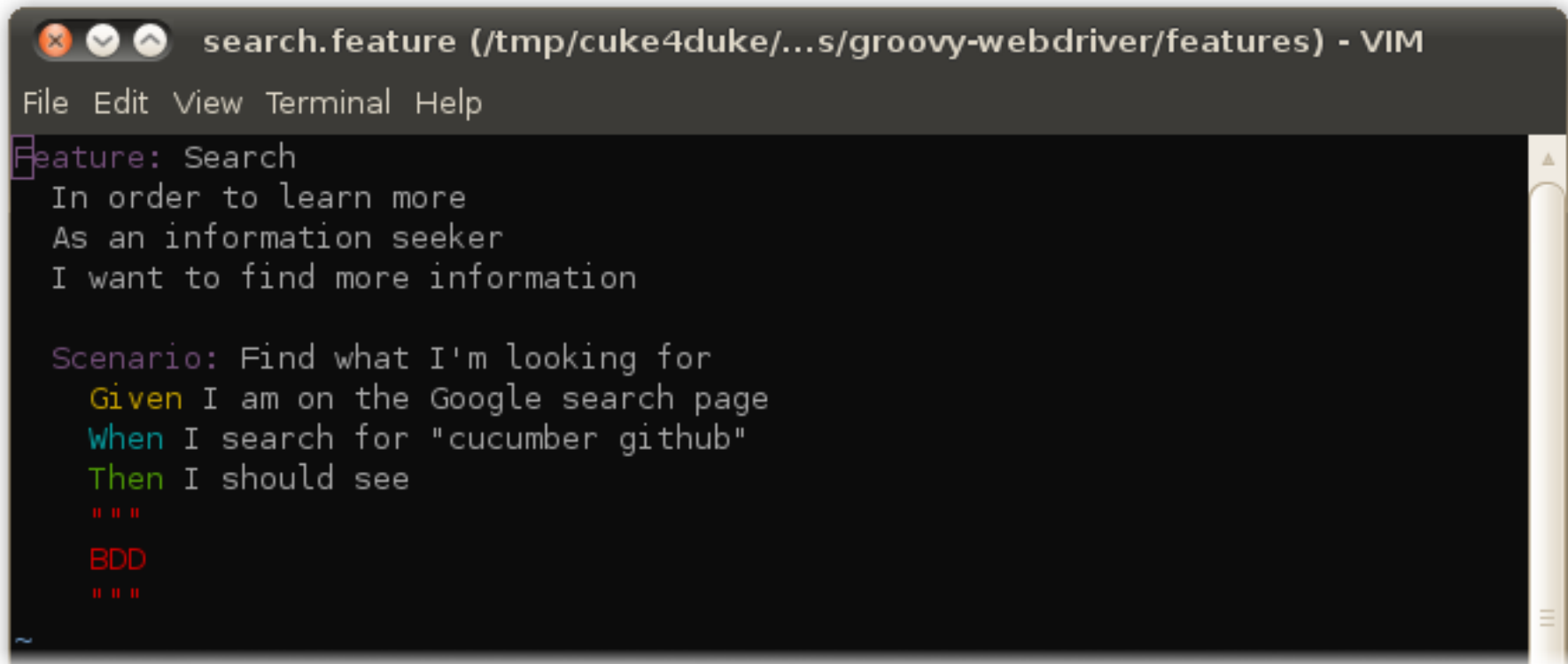
```
$ mvn integration-test
```

The first time you run a version of Cucumber you need to install the Ruby gems:

```
$ mvn -Dcucumber.installGems=true cuke4duke:cucumber
```

Warnings: Bleeding edge. First run will download a lot of dependencies.

# Syntax - Gherkin



```
search.feature (/tmp/cuke4duke/...s/groovy-webdriver/features) - VIM
File Edit View Terminal Help
Feature: Search
  In order to learn more
  As an information seeker
  I want to find more information

  Scenario: Find what I'm looking for
    Given I am on the Google search page
    When I search for "cucumber github"
    Then I should see
      """
      BDD
      """
```

## Steps

**Given** - Set up the context

**When** - Perform an action

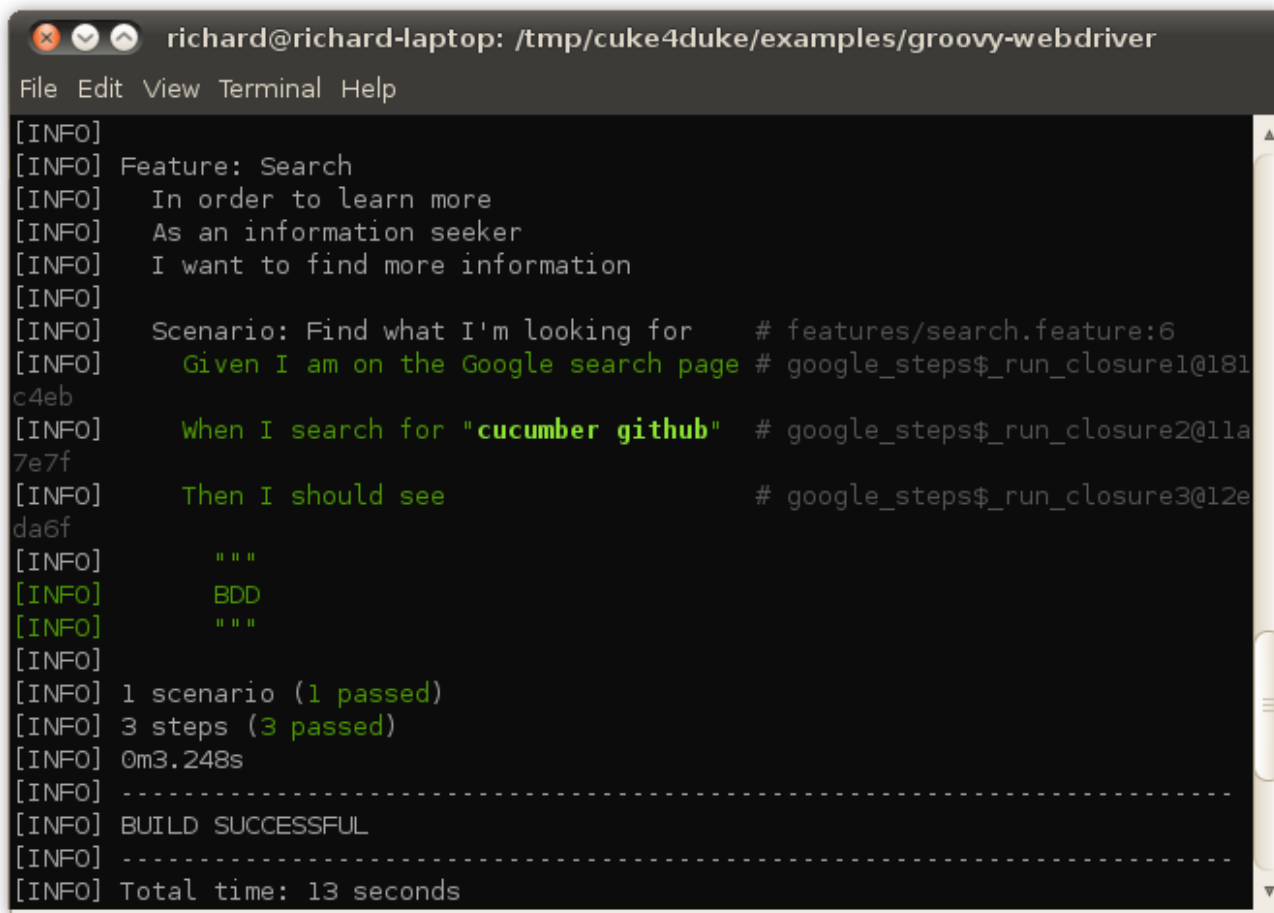
**Then** - Verify outcome

Can use **And** to chain additional steps of the same type

# Running Cucumber

**\$ mvn integration-test (Maven phase)**

**\$ mvn cuke4duke:cucumber (Maven goal)**

A terminal window titled "richard@richard-laptop: /tmp/cuke4duke/examples/groovy-webdriver" showing the output of a Cucumber test run. The output is displayed in a dark-themed terminal with green text for step names and white text for other information. The test passes successfully.

```
richard@richard-laptop: /tmp/cuke4duke/examples/groovy-webdriver
File Edit View Terminal Help
[INFO]
[INFO] Feature: Search
[INFO]   In order to learn more
[INFO]   As an information seeker
[INFO]   I want to find more information
[INFO]
[INFO] Scenario: Find what I'm looking for # features/search.feature:6
[INFO]   Given I am on the Google search page # google_steps$_run_closure1@181
c4eb
[INFO]   When I search for "cucumber github" # google_steps$_run_closure2@11a
7e7f
[INFO]   Then I should see # google_steps$_run_closure3@12e
da6f
[INFO]     ""
[INFO]     BDD
[INFO]     ""
[INFO]
[INFO] 1 scenario (1 passed)
[INFO] 3 steps (3 passed)
[INFO] 0m3.248s
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 13 seconds
```

# Automation - Step Definitions

```
google_steps.groovy (/tmp/cuke4...r/features/step_definitions) - VIM
File Edit View Terminal Help
import org.openqa.selenium.By
import static org.junit.Assert.*
import static org.junit.matchers.JUnitMatchers.*

this.metaClass.mixin(cuke4duke.GroovyDsl)

Given(~"I am on the Google search page") { ->
    browser.get("http://google.com/")
}

When(~"I search for \"(.*)\"") { String query ->
    searchField = browser.findElement(By.name("q"))
    searchField.sendKeys(query)
    // WebDriver will find the containing form for us from the searchField element
    searchField.submit()
}

Then(~"I should see") { String text ->
    assertThat(browser.getPageSource(), containsString(text))
}

~
~
"features/step_definitions/google_steps.groovy" 21L, 579C 1,1 All
```

# Where does browser come from?



```
env.groovy (/tmp/cuke4duke/exam...-webdriver/features/support) - VIM
File Edit View Terminal Help
//import org.openqa.selenium.firefox.FirefoxDriver
import org.openqa.selenium.htmlunit.HtmlUnitDriver

this.metaClass.mixin(cuke4duke.GroovyDsl)

Before() {
    //browser = new FirefoxDriver()
    browser = new HtmlUnitDriver()
}

After() {
    browser.close()
    browser.quit()
}
~
~
~
~
~
~
~
~
~
~
"features/support/env.groovy" 14L, 274C 1,1 All
```

# GroovyDsl

```
Given(~'repeated \d+ times') { times ->  
  // check something appears x times  
}
```

Same applies for **When, Then, And**

```
Before('tag') {  
  // run code before a scenario  
}
```

```
After('tag') {  
  // run code after a scenario  
}
```

```
World() {  
  // see http://bit.ly/davWpW for more details  
}
```

# Selenium 2 - WebDriver

Selenium 2 is the melding of Selenium and the WebDriver API

```
browser = new FirefoxDriver() // Or IE etc
```

```
browser.get('http://example.com')
```

```
browser.findElementByClassName('navigation')
```

```
browser.findElementsByCssSelector('#results li')
```

**findElementByXXX** returns a WebElement for first match

**findElementsByXXX** return a collection of matches

```
element.text // getText() in Java
```

```
element.click()
```

```
element.sendKeys('type something')
```

```
element.value
```

```
element.selected
```

```
element.getAttribute('href')
```

# Groovyisms - Collections

```
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
</ul>
```

```
def elements = browser.findElementsByTagName('li')
elements.text
> ['1', '2', '3']
```

```
// Check the numbers increase
elements.eachWithIndex { element, i ->
  assertThat(element.text, is(i + ''))
}
```

# When? Why?

- Acceptance tests are written up front
- Defines the scenarios you need to code
- Facilitates discussion on different scenarios
- Once green you are done
- Leads to no unnecessary code (YAGNI)
- Acceptance criteria automatically becomes regression suite

Amber > Red > Green  
Pending > Failing > Passing

# How?

- Acceptance tests are first class artifacts
- Keep code DRY by reusing steps
- Refactor steps to keep them clean
- Focus on **actions** not **tasks**

## **Good**

**Given I am signed in as an administrator**  
**When I view the list of signed in users**  
**Then I should see ...**

**Bad** (unless you are trying to test sign in)

**Given I am on the home page**  
**And I click 'Sign in'**  
**And I fill in username with 'admin'**  
**And I fill in password with 'secret'**  
**And I press 'Sign in'**

# Links

<http://cukes.info/>

<http://wiki.github.com/aslakhellesoy/cuke4duke/>

<http://github.com/tpope/vim-cucumber>

<http://selenium.googlecode.com/svn/trunk/docs/api/java/index.html>

<http://www.rapaul.com/tag/bdd/>

Cucumber photo (creative commons) <http://www.flickr.com/photos/vizzual-dot-com/2738586453/>

Thanks!

Richard Paul

<http://rapaul.com>

@rapaul