Behaviour Driven Development

using Cucumber JVM and Groovy

About Me

Marco Vermeulen

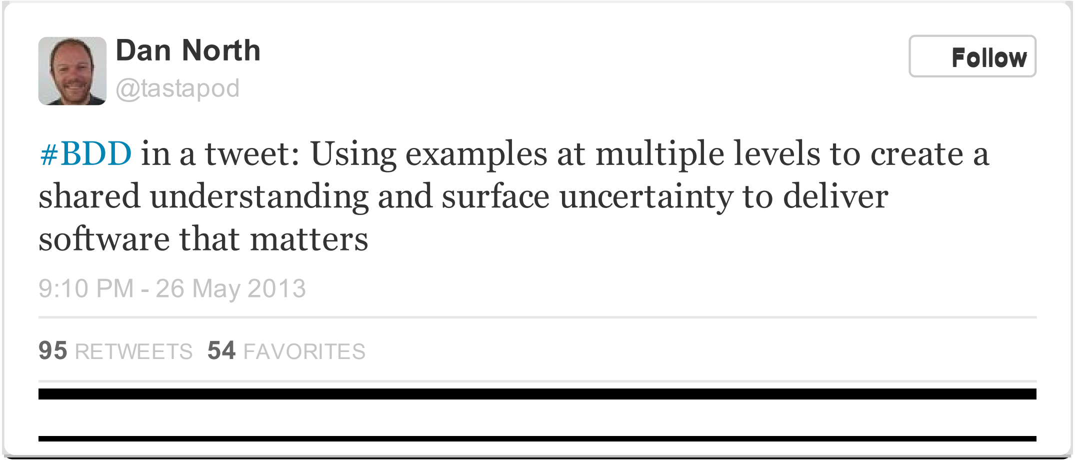
- Java Dev for past 10 years
- Groovy and Grails for 5
- Worked for Shazam, Associated Newspapers, Burberry
- Currently: Consulting at Equal Experts
- Author of GVM (Groovy enVironment Manager)
- **http://wiredforcode.com**
- Twitter: **@marcoVermeulen**

About the Talk

- BDD in a Nutshell
- The Good and Bad of BDD
- Cucumber as solution
- Mini Cucumber Demo
- Grails Cucumber Plugin
- Example Application

BDD Defined

Straight from the horse's mouth:



Dan North
@tastapod

#BDD in a tweet: Using examples at multiple levels to create a shared understanding and surface uncertainty to deliver software that matters

9:10 PM - 26 May 2013

95 RETWEETS 54 FAVORITES

BDD in a Nutshell

Using Examples

at multiple levels

to create a shared understanding

and surface uncertainty

to deliver software

that matters!

So???

- Collaborate on a *Specification* that *all* understand.
- Write in light weight markup called *Gherkin*.
- Make *Executable* Specifications by parsing the markup.
- Watch these new *Pending* unimplemented tests *fail*!
- *Implement* the tests, watching them *fail*.
- Write the *Implementation*, watch it *pass*!

Bad Wrap

An Orphan

Marketing Hype

ATDD (Acceptance Test Driven Development)

Lost it's Zing!!!

Good Vibrations

TDD Evolved

Inspires Collaboration

Behaviour vs Implementation

Living Documentation

Abundance of Tools

Simples!

Abundance of Tools

JBehave

Concordian

Fitnesse

EasyB

Spock?

Jasmine (for JavaScript)

Cucumber

Cucumber and Gradle

Cucumber and Gradle

An Anatomy

- Dependencies
- JUnit Test Runner
- Gherkin
- Step Definitions
- Hooks
- Tags

Cucumber Anatomy

Dependencies

```
repositories {
    mavenCentral()
}

dependencies {
    groovy 'org.codehaus.groovy:groovy:2.1.5'
    testCompile 'junit:junit:4.11'
    testCompile 'info.cukes:cucumber-groovy:1.1.1'
    testCompile 'info.cukes:cucumber-junit:1.1.1'
}
```

`build.gradle`

Cucumber Anatomy

Test Runner

```groovy
import cucumber.api.junit.Cucumber
import org.junit.runner.RunWith

@RunWith(Cucumber)
@Cucumber.Options(
    format=["pretty", "html:build/reports/cucumber"],
    strict=true,
    features=["src/test/cucumber"],
    glue=["src/test/cucumber/steps", "src/test/cucumber/support"],
    tags=["~@manual", "~@review"]
)
public class RunCukesTest {
    //leave me empty!
}
```

`src/test/groovy/RunCukeTests.groovy`

Cucumber Anatomy

Gherkin

```
Feature: Calculate

    Scenario: Add two numbers
        Given the input "2+2"
        When the calculator is run
        Then the output should be "4"

    Scenario: Subtract two numbers
        Given the input "9-4"
        When the calculator is run
        Then the output should be "5"
```

`src/test/cucumber/adding.feature`

Cucumber Anatomy

Step Definitions

```
import static cucumber.api.groovy.EN.*

Given(~'^the input "([^"]*)"$') { String input ->
    //some groovy code
}

When(~'^the calculator is run$') { ->
    //some groovy code
}

Then(~'^the output should be "([^"]*)"$') { String output ->
    //some groovy code
}
```

src/test/cucumber/steps/add_steps.groovy

Cucumber Anatomy

Hooks

```groovy
import static cucumber.api.groovy.Hooks.*

Before(){
    //some groovy code
}

After(){
    //some groovy code
}
```

`src/test/cucumber/support/env.groovy`

Before and After each Scenario

Mini Demo

Grails Cucumber Plugin

by Martin Hauner

Plugin Portal

https://github.com/hauner/grails-cucumber

Grails Cucumber Plugin

Features

- Convention over Configuration
- Easy Configuration
- No Test Runner
- Uses Functional Test phase
- Has Friends! (Geb, Spock, Build Test Data, Fixtures)
- Command Line integration
- Good IDE Support
- Under active development

Grails Cucumber Example

- GVM admin console
- Contrived example!!!
- Walking skeleton
- All moving parts of setup

Grails Cucumber Example

Configuration

```
plugins {
    ...
    test ":cucumber:0.8.0"
    ...
}
```

`grails-app/conf/BuildConfig.groovy`

Configuration

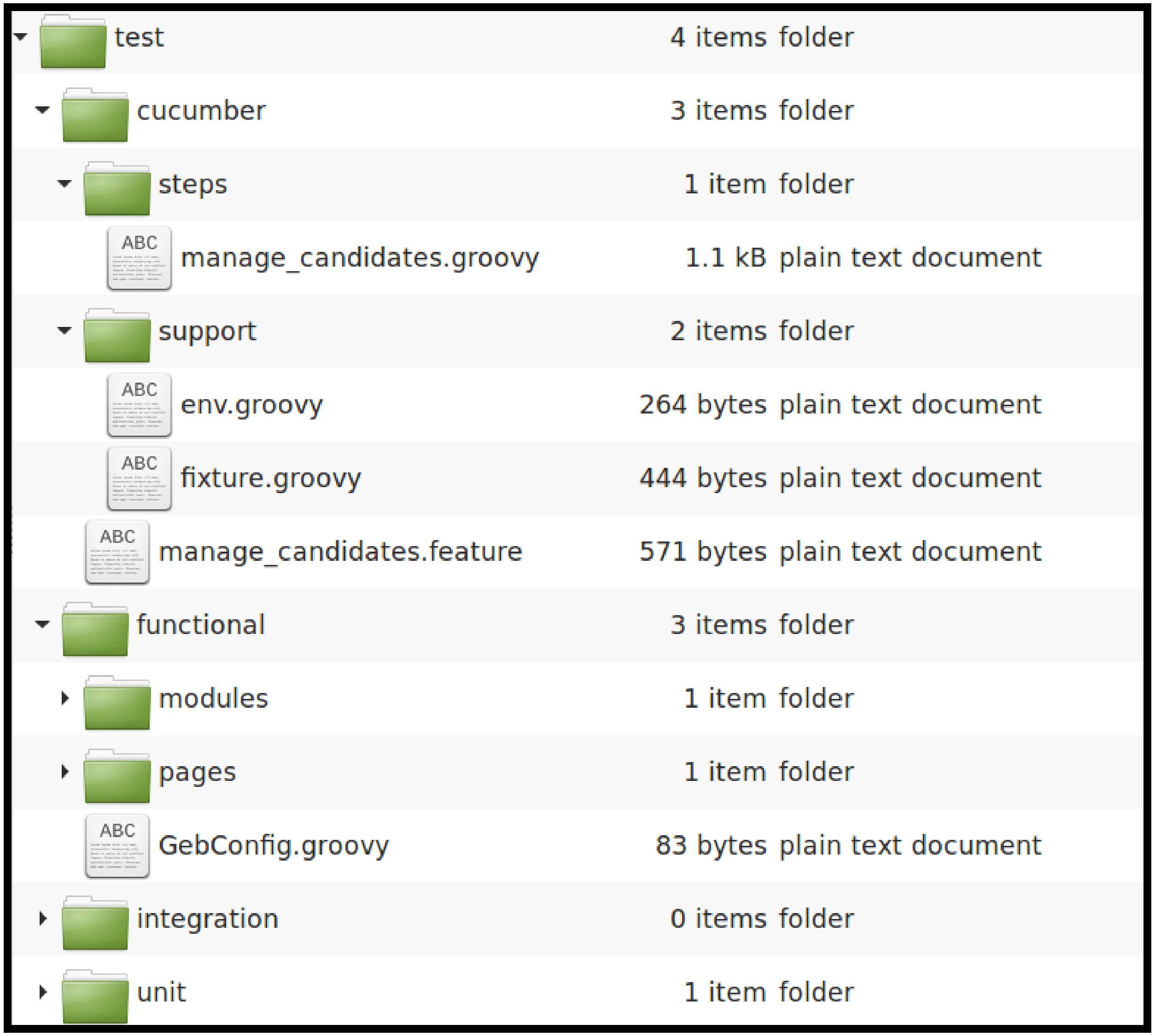
```
cucumber {
    tags = ["~@manual"]
    features = ["test/cucumber"]
    strict = true
    glue = ["test/cucumber/steps", "test/cucumber/support"]
}
```

`grails-app/conf/CucumberConfig.groovy`

Replaces Test Runner ***configuration***.

Grails Cucumber Example

Test Folder Structure



```
▾ test                                    4 items  folder
  ▾ cucumber                              3 items  folder
    ▾ steps                                1 item  folder
        manage_candidates.groovy           1.1 kB  plain text document
    ▾ support                             2 items  folder
        env.groovy                      264 bytes  plain text document
        fixture.groovy                  444 bytes  plain text document
      manage_candidates.feature         571 bytes  plain text document
  ▾ functional                            3 items  folder
    ▸ modules                              1 item  folder
    ▸ pages                                1 item  folder
      GebConfig.groovy                   83 bytes  plain text document
  ▸ integration                           0 items  folder
  ▸ unit                                   1 item  folder
```

Grails Cucumber Example

Gherkin Feature

```
Feature: Manage Candidates

  Scenario: List Candidates
    Given the candidate "Grails" exists with default version "2.2.2"
    And the candidate "Groovy" exists with default version "2.1.4"
    When I visit the Candidate page
    Then I see "Grails" listed
    And I see "Groovy" listed
```

`test/cucumber/manage_candidates.feature`

Grails Cucumber Example

Step Definitions

```
Given(~'^the candidate "([^"]*)" exists with default version "([^"]*)"$'
    def candidate = new Candidate(
        name: name,
        defaultVersion: defaultVersion
    )
    assert candidate.save()
}
When(~'^I visit the Candidate page$') { ->
    to CandidatePage
    at CandidatePage
}
Then(~'^I see "([^"]*)" listed$') { String candidateName ->
    assert page.isCandidateInList(candidateName)
}
```

`test/cucumber/steps/manage_candidates.groovy`

Gorm!

Geb!

Grails Cucumber Example

Domain Class

```
class Candidate {
    String name
    String defaultVersion
}
```

`grails-app/domain/../Candidate.groovy`

Grails Cucumber Example

Geb Configuration

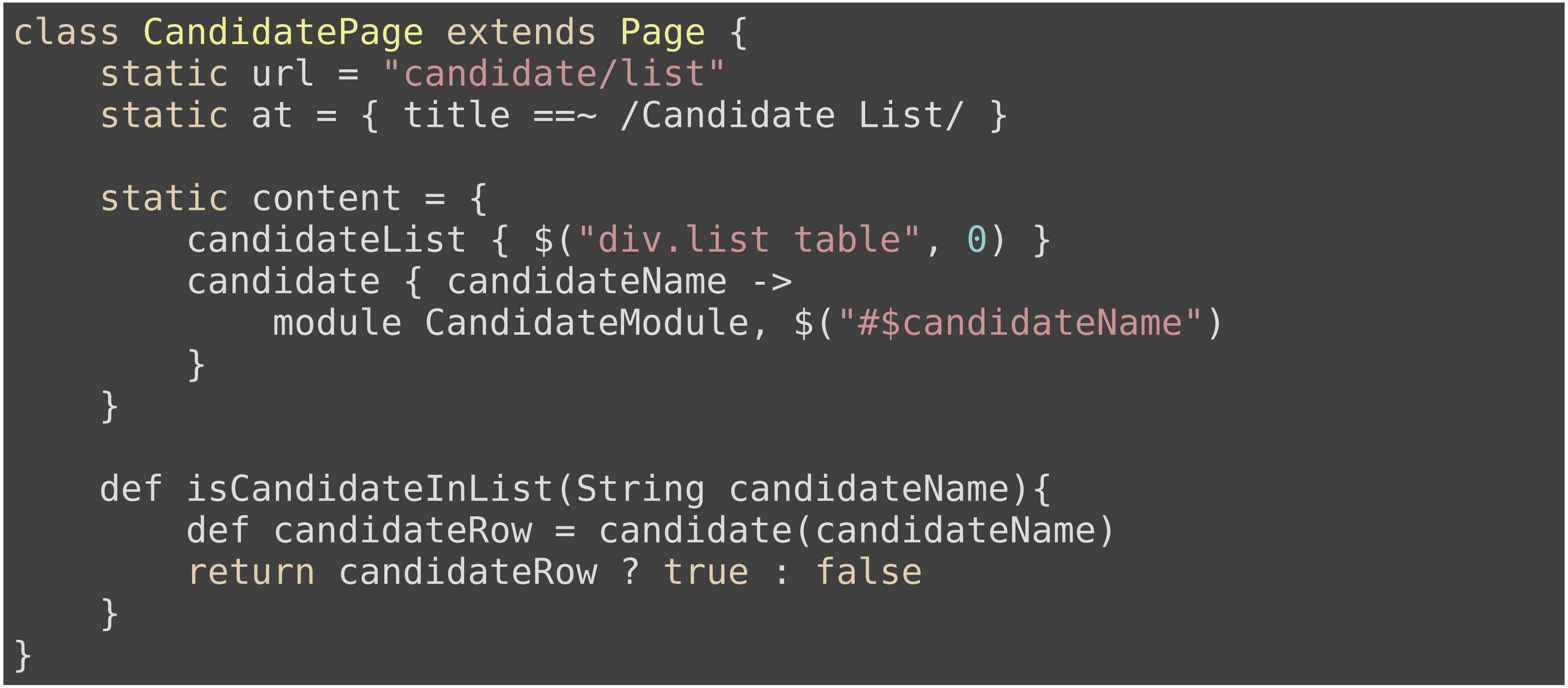
```
import org.openqa.selenium.chrome.ChromeDriver

driver = {
    new ChromeDriver()
}
```

`test/functional/GebConfig.groovy`

- *Remember to* ***download*** *chromedriver!*
- *Place it in root of project folder.*
- *...or use any selenium driver you like.*

Grails Cucumber Example

Geb Page



```
class CandidatePage extends Page {
    static url = "candidate/list"
    static at = { title ==~ /Candidate List/ }

    static content = {
        candidateList { $("div.list table", 0) }
        candidate { candidateName ->
            module CandidateModule, $("#$candidateName")
        }
    }

    def isCandidateInList(String candidateName){
        def candidateRow = candidate(candidateName)
        return candidateRow ? true : false
    }
}
```

test/functional/page/CandidatePage.groovy

Grails Cucumber Example

Geb Module

```groovy
class CandidateModule extends Module{
    static content = {
        name { $("#name") }
        defaultVersion { $("#default") }
    }
}
```

`test/functional/modules/CandidatePage.groovy`

Grails Cucumber Example

Environment Hooks

```
Before () {
    bindingUpdater = new BindingUpdater (binding, new Browser ())
    bindingUpdater.initialize ()
}

After () {
    bindingUpdater.remove ()
}
```

`test/cucumber/support/env.groovy`

- *Use for any long running fixture*
- *Hooks run before and after each scenario*
- *Like @BeforeClass and @AfterClass in JUnit*

Grails Cucumber Example

Hooks: Clean up GORM

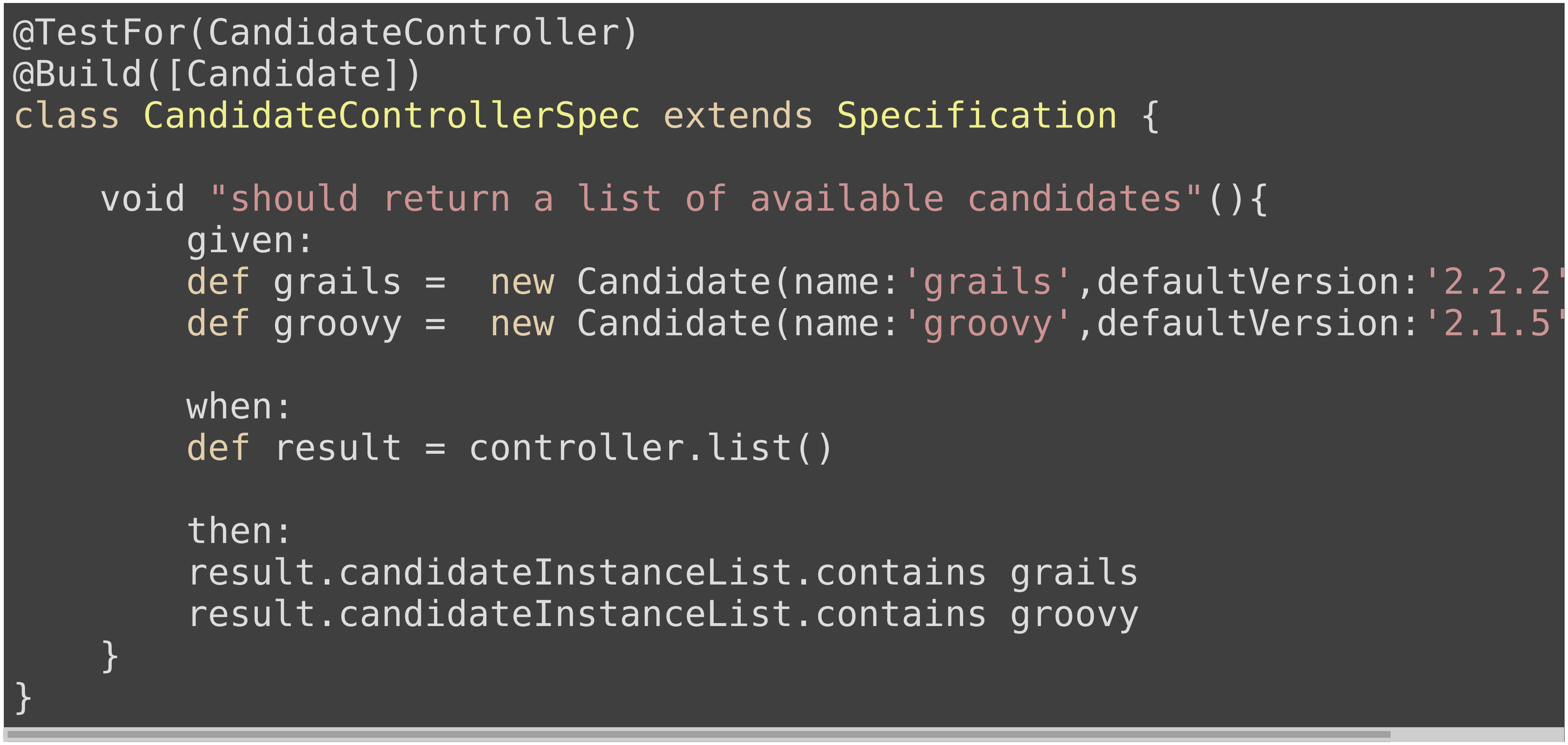
```
After () {

    def sessionFactory = appCtx.getBean("sessionFactory")
    sessionFactory.currentSession.flush()

    def dataSource = appCtx.getBean("dataSource")

    //clean fixtures
    println "Deleting the fixture..."
    def db = new Sql(dataSource)
    db.execute("DELETE FROM CANDIDATE;")

    sessionFactory.currentSession.clear()
}
```

`test/cucumber/support/fixture.groovy`

- *GORM's Hibernate session leaks across Scenarios.*
- *Feels sleezy :-P*

Grails Cucumber Example

Spock Specification



```
@TestFor(CandidateController)
@Build([Candidate])
class CandidateControllerSpec extends Specification {

    void "should return a list of available candidates"(){
        given:
        def grails =  new Candidate(name:'grails',defaultVersion:'2.2.2'
        def groovy =  new Candidate(name:'groovy',defaultVersion:'2.1.5'

        when:
        def result = controller.list()

        then:
        result.candidateInstanceList.contains grails
        result.candidateInstanceList.contains groovy
    }
}
```

Controller

Controller

```
class CandidateController {
    def list(Integer max) {
        params.max = Math.min(max ?: 10, 100)
        [candidateInstanceList: Candidate.list(params),
            candidateInstanceTotal: Candidate.count()]
    }
}
```

Grails Cucumber Example

GSP

```
<g:each in="${candidateInstanceList}"
    status="i" var="candidateInstance">
    <tr>
    <td id="${candidateInstance.name}">
        ${fieldValue(bean: candidateInstance, field: "name")}
    </td>
    <td>
        <g:link action="show" id="${candidateInstance.id}">
        ${fieldValue(bean: candidateInstance, field: "defaultVersion")}
        </g:link>
    </td>
    </tr>
</g:each>
```

Conclusion

- *BDD helps us Collaborate*
- *BDD helps us make software that Matters!*
- *Cucumber JVM and Gradle play nicely*
- *Grails Cucumber plugin Rocks!*
- *BDD is lots of fun!*

Thank You!!!

Q & A