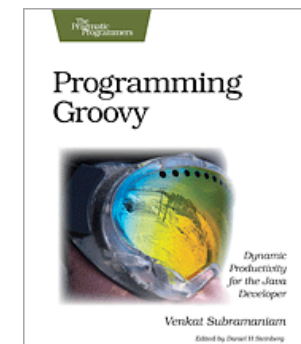


BUILDING DSLs IN GROOVY

```
speaker.identity {  
  name      'Venkat Subramaniam'  
  company   'Agile Developer, Inc.'  
  credentials 'Programmer', 'Author', 'Trainer'  
  blog      'http://agiledeveloper.com/blog'  
  email     'venkats@agiledeveloper.com'  
}
```



Abstract

- * Groovy's dynamic and meta programming capabilities make it a good candidate for implementing DSLs on the JVM. In this presentation we will take a look at a number of examples of DSLs implemented in Groovy.
- * We will discuss the significance of these DSLs, present other ideas for DSLs use case, and take an look at ways to implement them.

Domain Specific Language

* What's DSL?

- A language targeted at a particular type of problem
- Syntax focuses on intended domain/problem
- Unlike a general purpose language, you can't use it for all kinds of stuff (or don't want to)
- Limited in scope and capability
- Hence, they're small and simple

Characteristics of a DSL

* What are some of the qualities of a DSL?

✦ Context Driven

* Extremely context sensitive

✦ Fluent Interfaces

* Provides for a humane, easy to understand interface

Example of Context

```
lst = [1, 2, 3]

lst.with {
  add 4
  add 5
  println "The list contains $size elements"
}
```

Routed to context object lst



Example of Fluency

```
1.upto 5, {  
  println it  
}  
  
3.times { print 'ho ' }
```

Examples of Groovy DSLs

```
//GORM
class State
{
    String twoLetterCode

    static constraints = {
        twoLetterCode size: 2..2, unique: true, blank: false
    }
}

// Translates into validators on server side and
// constraints on client side
```

Examples of Groovy DSLs

- * Gant is Ant without XML
- * Built on top of Ant
- * Pure Groovy syntax to express builds

Examples of Groovy DSLs

```
scenario 'deposit money', {  
  given 'account number 123456789', {}  
  when 'deposit $50', {}  
  then 'account 123456789 balance goes up by $50', {}  
  and  
  then '...', {}  
}
```

- * easyb is a Behavior Driven Design tool to express executable documentation of stories and specs
- * Uses Groovy DSLs and is highly expressive
- * Here's story extracted from above "code"/DSL

```
6 behavior steps executed successfully  
Story: deposit money  
  scenario deposit money  
    given account number 123456789  
    when deposit $50  
    then account 123456789 balance goes up by $50  
    then ...
```

Examples of Groovy DSLs

```
langs = ['C++' : 'Stroustrup', 'Java' : 'Gosling', 'Lisp' : 'McCarthy']  
  
bldr = new groovy.xml.MarkupBuilder()  
  
bldr.languages {  
    langs.each {key, value ->  
        language(name: key) { author(value)}  
    }  
}
```

```
<languages>  
  <language name='C++'>  
    <author>Stroustrup</author>  
  </language>  
  <language name='Java'>  
    <author>Gosling</author>  
  </language>  
  <language name='Lisp'>  
    <author>McCarthy</author>  
  </language>  
</languages>
```

Groovy's XML Builder
makes building XML
tolerable

Groovy Tricks to Build DSL

- * Take advantage of closures and delegates

```
PizzaShop.order {  
  size 'Large'  
  toppings 'Olives', 'Bell Pepper', 'Onions'  
}
```

```
class PizzaShop  
{  
  def static order(closure)  
  {  
    PizzaShop shop = new PizzaShop()  
    shop.with closure // delegates calls from closure to shop  
    // ... do whatever else to complete order here...  
  }  
  def size(theSize) { println "size received $theSize" }  
  def toppings(String[] theToppings) { println "Toppings received $theToppings" }  
}
```

Groovy Tricks to Build DSL

* Take advantage of dynamic methods

```
playersAndScores = [:]

def players(String[] names) process.groovy
{
  names.each {name ->
    playersAndScores[name] = 0
  }
}

def getReportWinner()
{
  def max = -1
  def winner = ''
  playersAndScores.each {name, score ->
    if (score > max)
    {
      max = score
      winner = name
    }
  }

  "winner is $winner with score $max"
}

def methodMissing(String name, args)
{
  playersAndScores[name] = args[0]
  // error checking not shown
}
```

```
players 'Ben', 'George', 'Abe'
George 10
Ben 12
Abe 9
reportWinner scores.dsl
```

```
println new GroovyShell().evaluate(
  new File('process.groovy').text + new File('scores.dsl').text
)
```

winner is Ben with score 12

Groovy Tricks to Build DSL

* Make Use of Categories

```
use(DatesUtil.class)
{
    println 2.days.ago.at(4.30)

    println 20.days.ago.at(16.30)
}
```

Type conversions

2 => int

.days => int

.ago => Calendar

.at => Date

```
class DatesUtil
{
    public static int getDays(Integer self)
    {
        self
    }

    public static Calendar getAgo(Integer self)
    {
        def today = Calendar.instance
        today.add(Calendar.DAY_OF_MONTH, -self)

        today
    }

    public static Date at(Calendar self, Double time)
    {
        def timeDbl = time.doubleValue()
        def hours = (int)timeDbl
        def minutes = (int)((timeDbl - hours) * 100)

        self.set(Calendar.HOUR_OF_DAY, hours)
        self.set(Calendar.MINUTE, minutes)
        self.time
    }
}
```

Groovy Tricks to Build DSL

* Make Use of ExpandoMetaClass

```
Integer.metaClass.getDays = {->
    delegate
}

Integer.metaClass.getAgo = {->
    def today = Calendar.instance
    today.add(Calendar.DAY_OF_MONTH, -delegate)

    today
}

GregorianCalendar.metaClass.at = {Double time ->
    def timeDb1 = time.doubleValue()
    def hours = (int)timeDb1
    def minutes = (int)((timeDb1 - hours) * 100)

    delegate.set(Calendar.HOUR_OF_DAY, hours)
    delegate.set(Calendar.MINUTE, minutes)
    delegate.time
}

println 2.days.ago.at(4.30)

println 20.days.ago.at(16.30)
```

Groovy Tricks to Build DSL

- * Make Use of ExpandoMetaClass

```
// Allows you to enhance hierarchy instead of a specific class  
ExpandoMetaClass.enableGlobally()  
  
//GregorianCalendar.metaClass.at = {Double time ->  
Calendar.metaClass.at = {Double time ->  
...  
...
```

References

- * <http://groovy.codehaus.org>
- * <http://docs.codehaus.org/display/GROOVY/Writing+Domain-Specific+Languages>
- * <http://docs.codehaus.org/display/GROOVY/Gant>
- * <http://www.easyb.org>
- * “Programming Groovy: Dynamic Productivity for the Java Developer,” by Venkat Subramaniam, Pragmatic Bookshelf, 2008.

You can download examples and slides from
<http://www.agiledeveloper.com> - download

Thank You!

Please fill in your session evaluations

You can download examples and slides from
<http://www.agiledeveloper.com> - download