

GRAILS COOKBOOK

A COLLECTION OF TUTORIALS AND EXAMPLES

Beginners Tutorial

- Grails 2 IDE Setup
- Model View Controller (MVC) Pattern
- Playing with Groovy Language
- Starting with Controllers
- Basic Groovy Server Pages (GSP)
- Layout Templates
- Introduction to GORM
- Display Data from the Database
- Form Submission and Saving Data to the Database
- Scaffolding
- Grails Service Layer
- Criteria API
- HQL Queries (executeQuery and executeUpdate)
- Grails 3 Vue.js Hello World Examples
- Grails 3 Vue.js AJAX Table Examples

Example Applications

- Simple Ajax Chat
- Simple Document Management System
- Save Documents To Database
- Download / Export Excel File
- Download / Export ZIP File
- Grails Forum Application

How to Create Great Looking Grails User Interface

If you like my posts, like my Facebook page please :)

GRAILS COOKBOOK

784 likes

Like Page

Be the first of your friends to like this

Links

- Grails Main Site
- Grails Latest Doc
- Grails Wikipedia Page
- Grails Code on GitHub
- Spring IO
- Oracle's Java Site
- Java Tutorials

Groovy Map Tutorial

Feb 06, 2015 Groovy, Tutorial 3 comments

Maps are very useful when coding in Groovy. This tutorial will show the different ways of using a Groovy Map.

Introduction

A map is a collection of (key, value) pairs. It is commonly used for implementing lookups or cache. For example, if it is complicated and takes a lot of time to compute a person's salary, we can have:

```
def salaryMap = [:]
salaryMap['John'] = computeBasicPay('John') + computeDeductions('John')
salaryMap['Peter'] = computeBasicPay('Peter') + computeDeductions('Peter')
salaryMap['Mark'] = computeBasicPay('Mark') + computeDeductions('Mark')
```

We can utilize a map such that computations are performed only once. Once the map is populated, looking up a value is very fast!

```
println "Salary of John is: ${salaryMap['John']}
```

We can lookup the values as often as needed without the complex computation penalty.

In Groovy and Grails, maps can also be used like beans. In fact, this is not uncommon and are often seen in Grails projects. For example:

```
def personDetails = [firstName:'John', lastName:'Doe', age:25]
println "First Name: ${personDetails.firstName}"
println "Last Name: ${personDetails.lastName}"
println "Age: ${personDetails.age}"
```

As you could see, the keys are used like property getter/setter.

Note: A Groovy map is an instance of java.util.Map

Groovy Map Basics

These are the basics things that can be done with a Groovy Map

Declaration

Empty Map

This is how to declare an empty map

```
def emptyMap = [:]
```

The colon inside the square brackets denotes that the object is a map and not a list.

Map Literal Values

This is how to declare a map with initial values

```
def sampleMap = [color:'Blue', shape:'Circle']
```

The keys are String by default when written plainly without quotes (E.g. color instead of 'color' or "color"). The above declaration is the same as:

```
def sampleMap = ['color':'Blue', 'shape':'Circle']
def sampleMap2 = ["color":'Blue', "shape":'Circle']
```

We can use instance of other classes (aside from String) as keys or values

```
def sampleMap = [99:new Long(99), (BigDecimal.valueOf(5)):new Double(5)]
```

Add Values

These are the different ways of adding values to a map.

```
def sampleMap = [:]
sampleMap.put('thickness', 10)
sampleMap['color'] = 'Blue'
sampleMap.weight = 500
sampleMap.'shape' = 'Circle'
sampleMap << [price:150]
```

This map will have equivalent values as this:

```
def sampleMap = [color:'Blue', weight:500, shape:'Circle', thickness:10, price:150]
```

As you could see, we are allowed to:

- `sampleMap.put('thickness', 10)` - use put method because a Groovy map is also an instance of java.util.Map
- `sampleMap['color'] = 'Blue'` - enclose keys inside square brackets
- `sampleMap.weight = 500` - use bean notation
- `sampleMap.'shape'` - also a valid bean notation
- `sampleMap << [price:150]` - add values from another map

Replace Values

Updating values from an existing map is the same as adding values. The statement will just replace the existing value if it already exists in the map. For example:

```
def sampleMap = [color:'Blue', weight:500, shape:'Circle', thickness:10, price:150]
sampleMap['color'] = 'Red'
sampleMap.'shape' = 'Rectangle'
sampleMap.weight = 200
sampleMap.put('thickness', 5)
sampleMap << [price:99]
```

Will have this resulting value:

```
[color:'Red', shape:'Rectangle', weight:200, thickness:5, price:99]
```

Remove Values

Here are some samples on how to remove values from a map.

```
def sampleMap = [color:'Blue', weight:500, shape:'Circle', thickness:10, price:150]
sampleMap.remove('color')
sampleMap = sampleMap - ['shape':'Circle']
sampleMap -= [weight:500]
```

We could use remove or subtraction of map if you know the exact key value pair. The result of the above code is:

```
[thickness:10, price:150]
```

Lookup

Retrieving values from a Groovy Map is straightforward

```
def sampleMap = [color:'Blue', weight:500, shape:'Circle', thickness:10, price:150]
println "The color is: ${sampleMap.get('color')}}"
println "The weight is: ${sampleMap['weight']}"
println "The shape is: ${sampleMap.shape}"
println "The thickness is: ${sampleMap.thickness}"
```

Groovy Map Magic

Here are some Map manipulation tricks that makes Groovy fun to use:

Default Value

We can make a lookup return a default value if the key is not found in the map

```
assert 'Red' == [color:'Red', shape:'blue'].get('color', 'Blue')
assert 'Blue' == [shape:'blue'].get('color', 'Blue')
```

The first statement returns Red because color is in the key set. The second statement returns the default value Blue because color is not in the key set.

collectEntries

We can create a map from the list

```
def wordList = ['Apple', 'Banana', 'Cat']
def wordCountMap = wordList.collectEntries{ [(it):it.length()] }
```

Results to this values:

```
[Apple:5, Banana:6, Cat:3]
```

Count

We can count how many items in the map satisfies a condition

```
def wordList = ['Apple', 'Banana', 'Cat']
def wordCountMap = wordList.collectEntries{ [(it):it.length()] }
def longWords = wordCountMap.count { key, value->
    value >= 4
}
println "${longWords}"
```

This will result to the answer **2** because only 2 words have length greater than or equal to 4.

Union

We can add maps to combine their contents

```
def map1 = [p1:100, p2:200]
def map2 = [p3:300, p4:400]
def map3 = map1 + map2
println "${map3}"
```

This will result to

```
[p1:100, p2:200, p3:300, p4:400]
```

Intersect

We can get the intersection of two maps

```
def map1 = [p1:100, p2:200, p3:300, p4:400, p5:500]
def map2 = [p4:400, p5:500, p6:600, p7:700]
def map3 = map1.intersect(map2)
println "${map3}"
```

This will result to

```
[p4:400, p5:500]
```

Get keys as list

We can get all the keys in a map as a list

```
def map1 = [p1:100, p2:200, p3:300, p4:400, p5:500]
println "${map1.keySet()}"
```

This will print

```
[p1, p2, p3, p4, p5]
```

Get values as list

We can get all the values in a map as a list

```
def map1 = [p1:100, p2:200, p3:300, p4:400, p5:500]
println "${map1.values()}"
```

This will print

```
[100, 200, 300, 400, 500]
```

Min

We can get the lowest entry in the map given an expression inside a closure

```
def map1 = [p1:100, p2:200, p3:300, p4:400, p5:500]
println "${map1.min {it.key}}"
println "${map1.min {it.value}}"
```

Both println will result

```
p1=100
```

Max

We can get the highest entry in the map given an expression inside a closure

```
def map1 = [p1:100, p2:200, p3:300, p4:400, p5:500]
println "${map1.max {it.key}}"
println "${map1.max {it.value}}"
```

Both println will result

```
p5=500
```

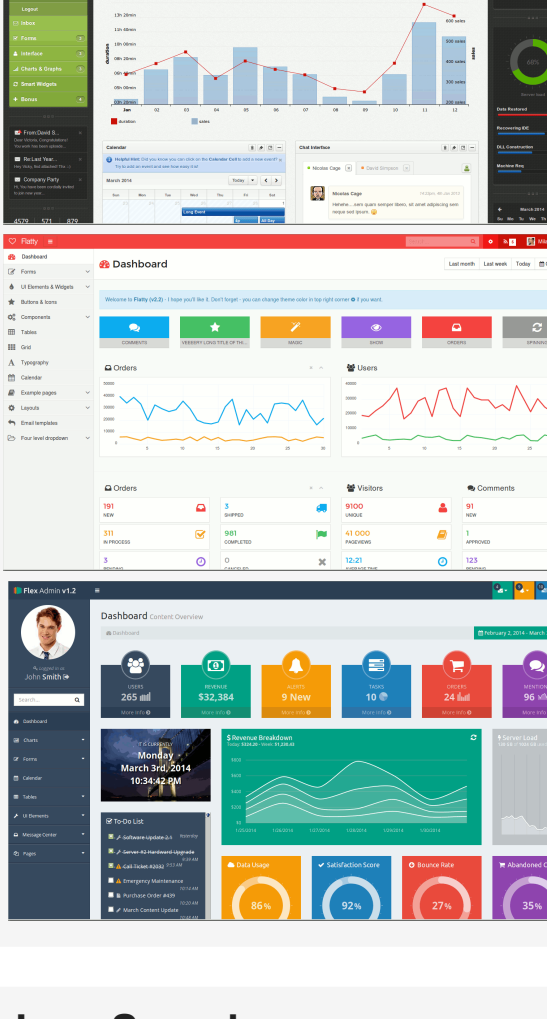
Remarks

There are many more Groovy magic goodness that we can do to a map. Only a few are listed above.

Grails 3 Tutorials

- Grails 3 IDE Setup
- Grails 3 Project Structure Tutorial
- Grails 3 Vue.js Hello World Examples
- Grails 3 Vue.js AJAX Table Examples

Grails Project Themes



Java Samples

- Two Dimensional String Array In Java
- Java Math.Round()
- Byte Array to String In Java
- Double To Integer In Java
- How To Add element to String Array in Java
- String Replace In Java
- Double To String With 2 Decimal Places In Java
- Read Text File in Java
- Maximum Value of Long in Java
- Double To String Without Exponential Scientific Notation in Java
- Iterate Through String Array in Java
- Split String Into Array Of Integers in Java
- Convert Java Float To String With 2 Decimal Place
- Left Pad String in Java
- Left Pad String With Zeros in Java
- Java String Split Newline
- Check If String Array Contains a Value In Java
- Check If String Contains Substring In Java
- Sort String Array in Java
- Float To Integer In Java
- String Array Length in Java
- Convert Java Integer To Fixed Length String
- Java Split String Into ArrayList
- Left Pad String With Spaces in Java
- Java Substring
- Java String Array Initialization
- Maximum Value of Integer in Java
- Java String Split Count
- Java 5 For Loop String Array
- Java String Array Declaration

Plugins

- Barcode (barcode4j) Plugin Example
- Spring Security Core Plugin Example - Annotations
- Spring Security Core Plugin Example - Static URL Map Rules
- Spring Security Core Plugin Example - Dynamic Request Maps
- Mail Plugin Tutorial
- Quartz Plugin Tutorial
- PayPal Plugin Tutorial
- Spring Security Core Plugin - Registering Callback Closures

Examples

- Simple Ajax Chat
- Simple Document Management System
- Save Documents To Database
- Download / Export Excel File
- Download / Export ZIP File
- Grails Forum Application

Groovy SQL

- Groovy SQL Database Connection Examples
- Groovy SQL Create Table Examples
- Groovy SQL Insert Operation Examples
- Groovy SQL Read Operation Examples
- Groovy SQL Update Operation Examples
- Groovy SQL Delete Operation Examples

Snippet

- Improving Performance of Large Batch Insert
- Add MySQL driver jar to a Grails Project using BuildConfig
- How To Check If Environment is Test, Development, or Production in Grails
- How to make Grails use JNDI datasource with Tomcat
- Grails - How To Create Custom Table Index Or Composite Index
- Grails Google Maps Geocoding Example
- Grails Spring Security Core Plugin - Registering Callback Closures
- Grails Example - Upload Excel Data File To Database
- Grails render images on the fly in GSP
- How to make Grails use an external properties file for datasource configuration
- Grails chained select - load data on one dropdown box depending on another
- Grails HQL Join Examples
- Grails HQL Count Examples
- Grails HQL Delete Examples
- Grails HQL Pagination Examples
- Grails HQL Group By Examples
- Grails Criteria Order By Examples
- Grails Criteria Projections Examples
- Grails Criteria Criteria Example
- Grails Select Enum Values

Groovy

- Grails Tutorial for Beginners - Playing with Groovy Language
- Groovy String Tutorial
- Groovy Map Tutorial
- Groovy Each Loop Examples
- Groovy -e Command Line Option
- Groovy String To Integer
- Groovy List To String
- Groovy Sleep Examples
- Groovy For Loop Examples
- Groovy print Examples
- Groovy println Examples
- Groovy Times Loop Examples
- Groovy Array Manipulation Examples
- Groovy File Examples
- Groovy Def Declaration Examples
- Groovy Find
- Groovy FindAll
- Groovy Copy File Examples
- Groovy Array Length
- Groovy Add To Map Examples
- Groovy Substring Examples
- Groovy List Tutorial And Examples
- Groovy Switch Statement Examples
- Groovy XmlSlurper Examples for Parsing XML
- Groovy XmlParser Examples for Parsing XML
- Groovy Create XML Document Examples
- Groovy Edit/Modify XML Examples
- Groovy Sort Information in XML Document
- Groovy Compare String
- Groovy Set Examples
- Groovy Split Examples
- Groovy Range Examples
- Groovy Enum Examples
- Groovy String Tokenize Examples
- Convert String to Enum in Groovy
- Grails Select Enum Values

Tag Reference

- cookie Tag Example
- actionSubmitImage Tag Example
- actionSubmit Tag Example
- checkBox Tag Example
- AJAX Tag - remoteField Example
- localeSelect Tag Example
- timeZoneSelect Tag Example
- country Tag Example
- currencySelect Tag Example
- textArea Tag Example
- textField Tag Example
- unless Tag Example
- if Tag Example
- each Tag Examples
- collect Tag Examples
- join Tag Examples
- findAll Tag Examples
- createLink Tag Example
- createLinkTo Tag Example
- datePicker Tag Example
- else Tag Example
- unless Tag Example
- formatDate Tag Example
- formatBoolean Tag Example
- formatNumber Tag Example
- Select Enum Values