ARTIGO: 51
Dica em Groovy - Transformar um java.util.Map para parâmetros http GET
Olá pessoas
Frequentemente precisamos construir uma string para representar parâmetros para enviar em uma requisição http GET, por exemplo:

http://maps.googleapis.com/maps/api/geocode/json?address=Bento+Gonçalves&sensor=false

Aqui enviamos 2 parâmetros: address e sensor

Em Groovy, muitas vezes temos esses atributos em um java.util.Map (talvez params em Grails) ou em um Objeto. Poderíamos ter essa classe que representa esses atributos:

Groovy Code
```
class GoogleMapsAddress {
    String address
    Boolean sensor
    String inutil
}
```

Agora, imagine que precisamos fazer uma requisição para a api do google maps com os atributos de um objeto dessa classe enviados como parâmetros na url. Podemos sair concatenando Strings, mas isso é muito chato de se fazer e propenso a erros. O Groovy disponibiliza vários recursos que em conjunto possibilitam fazer essa tarefa em somente uma linha:

Groovy Code
```
// url da api
String apiUrl = 'http://maps.googleapis.com/maps/api/geocode/json?'

// críamos a instância que representa os atributos da requisição GET
def googleMapsAddress = new GoogleMapsAddress(address:'Bento Gonçalves', sensor: false)

// transformamos o objeto em parâmetros com essa linha
// lembre que groovy.lang.Object.properties é um Map representando os atributos da classe, googleMapsAddress.properties é um Map
def getParams = googleMapsAddress.properties.collect{ it }.join( '&' )

// críamos a requisição completa com os parâmetros concatenados
def requestUrl = "${apiUrl}${getParams}"
```

Teremos essa url:

Groovy Code
```
http://maps.googleapis.com/maps/api/geocode/json?class=class GoogleMapsAddress&address=Bento Gonçalves&sensor=false&inutil=null
```

Quase lá. Não queremos atributos 'null', como o atributo 'inutil' no exemplo acima, e nem o atributo chamado 'class' que vêm de brinde no map properties de todas as classes Groovy, podemos fazer um pouco melhor. A linha de código que coleta os atributos ficaria assim:

Groovy Code
```
def getParams = googleMapsAddress.properties.findAll{ k, v -> v != null && !['class'].contains(k)}.collect{it}.join('&')
```

Agora utilizamos o método findAll para filtrar resultados, teremos:

Groovy Code
```
http://maps.googleapis.com/maps/api/geocode/json?address=Bento Gonçalves&sensor=false
```

E parece que temos o que queríamos. Não importa a quantidade de atributos da classe, todos os atributos serão transformados. No nosso exemplo temos 3, mas poderia ser 30 e não precisaríamos alterar nada no nosso código para lidar com esses outros atributos.

Espero que esse dica tenha sido útil pra alguém.
Salute!!

Até a próxima!