



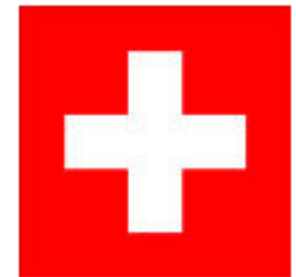
Getting Groovy on the Web and Desktop

Andres Almiray – Canoo Engineering AG

`andres.almiray@canoo.com`

About the speaker

- Ⓜ Java developer since '95
- Ⓜ Open Source believer
- Ⓜ Groovy dev team member since '07
- Ⓜ Griffon co-founder and project lead



canoo

[@aalmiray](http://jroller.com/aalmiray)





- Ⓜ <http://groovy.codehaus.org>
- Ⓜ Dynamic and agile language for the JVM
- Ⓜ Apache License 2.0
- Ⓜ Started in 2003
- Ⓜ We just released 1.8.4 a few weeks ago!

Greeter in Java







```
1 public class Greeter {  
2     private String salutation;  
3     public String setSalutation(String s) {  
4         salutation = s;  
5     }  
6     public String getSalutation() {  
7         return salutation;  
8     }  
9     public String greet(String who) {  
10        return salutation + " " + who;  
11    }  
12    public static void main(String[] args) {  
13        Greeter g = new Greeter();  
14        g.setSalutation("Hello ");  
15        System.out.println(g.greet("Groovy"));  
16    }  
17 }
```

Greeter in Groovy

```
1 public class Greeter {  
2     private String salutation;  
3     public String setSalutation(String s) {  
4         salutation = s;  
5     }  
6     public String getSalutation() {  
7         return salutation;  
8     }  
9     public String greet(String who) {  
10        return salutation + " " + who;  
11    }  
12    public static void main(String[] args) {  
13        Greeter g = new Greeter();  
14        g.setSalutation("Hello ");  
15        System.out.println(g.greet("Groovy"));  
16    }  
17 }
```

Groovier Greeter

```
1 class Greeter {  
2     String salutation  
3     String greet(String who) {  
4         "$salutation $who"  
5     }  
6 }  
7  
8 g = new Greeter(salutation: 'Hello')  
9 println g.greet('Groovy')
```

-  <http://grails.org>
-  Full stack Web development platform
-  Spring, Hibernate, Sitemesh ...
-  Apache License 2.0
-  Started in 2005
-  +600 plugins



- Ⓜ <http://griffon.codehaus.org>
- Ⓜ Grails inspired desktop development platform
- Ⓜ Inspired by JSR 296 and 295 as well
- Ⓜ Apache License 2.0
- Ⓜ +170 plugins
- Ⓜ Started in 2008

Step 1

Building the backend

Domain Classes

```
1 class Author {  
2     static constraints = {  
3         name(blank: false)  
4         lastname(blank: false)  
5     }  
6  
7     String name  
8     String lastname  
9  
10    static hasMany = [books: Book]  
11  
12    String toString() { "$name $lastname" }  
13 }
```

Author Book

Controllers

```
1 import grails.converters.JSON
2
3 class AuthorController {
4     static defaultAction = 'list'
5
6     def list = {
7         render(Author.list(params) as JSON)
8     }
9
10    def show = {
11        def author = Author.get(params.id)
12        if (!author) {
13            redirect(action: 'list')
14        } else {
15            render(author as JSON)
16        }
17    }
18
19    def search = {
20        def list = []
```

Author Book

Configuration

```
1 class UrlMappings {
2     static mappings = {
3         "/author/"(controller: 'author', action: 'list')
4         "/author/search"(controller: 'author', action: 'search')
5         "/author/list"(controller: 'author', action: 'list')
6         "/author/$id"(controller: 'author', action: 'show')
7         "/book/"(controller: 'book', action: 'list')
8         "/book/search"(controller: 'book', action: 'search')
9         "/book/list"(controller: 'book', action: 'list')
10        "/book/$id"(controller: 'book', action: 'show')
11        "/"(view: "/index")
12        "500"(view: '/error')
13    }
14 }
```

UrlMappings

BootStrap

Sample REST query

```
1 curl http://localhost:8080/bookstore/author
2
3 [
4   { "class": "Author", "id": 1,
5     "books": [{"class": "Book", "id": 1}],
6     "lastname": "Paz",
7     "name": "Octavio"},
8   { "class": "Author", "id": 2,
9     "books": [ {"class": "Book", "id": 4},
10               {"class": "Book", "id": 2}],
11     "lastname": "Garcia Marquez",
12     "name": "Gabriel"},
13   { "class": "Author", "id": 3,
14     "books": [ {"class": "Book", "id": 3}],
15     "lastname": "Hofstadter",
16     "name": "Douglas R."
17   }
18 ]
```

Step 2

Building the frontend

Client view

Authors

Books

Search

Name	Lastname
------	----------

Connect

View

```
1 package bookclient
2
3 makeTableTab = { params ->
4     scrollPane(title: params.title) {
5         table {
6             def tf = defaultTableFormat(columnNames: params.columns)
7             eventTableModel(source: params.source, format: tf)
8             installTableComparatorChooser(source: params.source)
9         } } }
10
11 panel(id: 'content', border: lineBorder(color: Color.BLACK, thickne
12     migLayout(layoutConstraints: 'fill')
13     tabbedPane(constraints: 'grow, wrap') {
14         makeTableTab(source: model.authors,
15             title: 'Authors', columns: ['Name', 'Lastname'])
16         makeTableTab(source: model.books,
17             title: 'Books', columns: ['Title'])
18         busyComponent(busy: bind{model.busy}, title: 'Search') {
19             panel {
20                 migLayout(layoutConstraints: 'fill')
21                 textField(columns: 30, text: bind('query', target: model))
```


Model

```
1 package bookclient
2
3 import groovy.beans.Bindable
4 import griffon.transform.PropertyListener
5 import ca.odell.glazedlists.EventList
6 import ca.odell.glazedlists.BasicEventList
7 import ca.odell.glazedlists.SortedList
8 import static griffon.util.GriffonNameUtils.isBlank
9
10 class BookstoreModel {
11     @PropertyListener(enabler)
12     @Bindable String query
13     @Bindable String status = ''
14     @Bindable boolean busy
15     @Bindable boolean enabled = false
16
17     static final AUTHORS = 'author'
18     static final BOOKS = 'book'
19
20     EventList authors = new SortedList(new BasicEventList(),
21         {a, b -> a.lastname <=> b.lastname} as Comparator)
22     EventList books = new SortedList(new BasicEventList(),
```

Controller

```
1 package bookclient
2
3 class BookstoreController {
4     def model
5     def view
6     def bookstoreService
7
8     def search = {
9         execSync {
10             model.busy = true
11             model.status = ''
12             model.results.clear()
13         }
14
15         String where = view.choice.selection.actionCommand
16
17         try {
18             List results = []
19             switch(where) {
20                 case BookstoreModel.AUTHORS:
21                     results = bookstoreService.searchAuthors(model)
22                     break
```

Service

```
1 package bookclient
2
3 class BookstoreService {
4     List searchAuthors(model) {
5         withRest(id: 'bookstoreREST') {
6             def response = get(path: 'author/search',
7                               query: [q: model.query])
8             response.data.inject([]) { list, a ->
9                 author.books.id.collect(list) { bookId ->
10                     def b = model.books.find{it.id == bookId}
11                     [title: b.title, name: a.name, lastname: a.lastname]
12                 }}}
13
14     List searchBooks(model) {
15         withRest(id: 'bookstoreREST') {
16             def response = get(path: 'book/search',
17                               query: [q: model.query])
18             response.data.collect([]) { b ->
19                 def a = model.authors.find{it.id == b.author.id}
20                 [title: b.title, name: a.name, lastname: a.lastname]
21             }}}
22 }
```

Want more Groovy?
Here are some alternatives



- Ⓜ <http://gaelyk.appspot.com>
- Ⓜ Lightweight Groovy toolkit for GAE
- Ⓜ Apache License 2.0
- Ⓜ Started in 2009
- Ⓜ Has an emerging plugin system

Ratpack

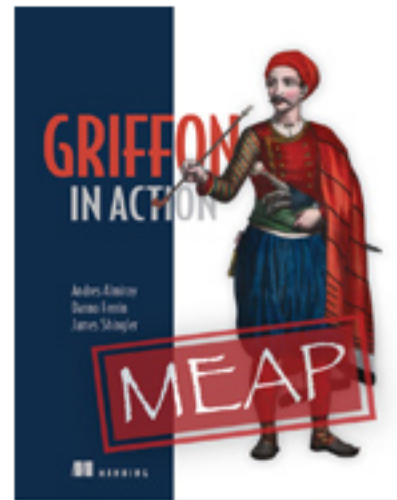
- Ⓜ <https://github.com/bleedingwolf/Ratpack>
- Ⓜ Sinatra inspired web framework
- Ⓜ Apache License 2.0
- Ⓜ Started in 2010
- Ⓜ Still in the early stages

Canoo RIA Suite

- Ⓜ <http://canoo.com/ulc>
- Ⓜ Java based RIA solution
- Ⓜ There's a Grails plugin available
- Ⓜ Combine Groovy & Java as you see fit

<http://people.canoo.com/share>

Thank you!



@aalmiray