



GRAILS

What's new in Grails 2.0?



springsource®

A division of vmware®



About me

- Jeff Brown
- Grails Core Developer
- SpringSource/VMware Engineer
- jbrown@vmware.com
- @jeffscottbrown

The Year in Grails



The Year in Grails

- Grails 1.3
 - Plugins in Dependency DSL, Groovy 1.7, Named Queries etc.



The Year in Grails

- Grails 1.3
 - Plugins in Dependency DSL, Groovy 1.7, Named Queries etc.
- More and more plugins
 - Spring Security Core et al.
 - RabbitMQ
 - Gemfire
 - Resources, etc.



The Year in Grails

- Grails 1.3
 - Plugins in Dependency DSL, Groovy 1.7, Named Queries etc.
- More and more plugins
 - Spring Security Core et al.
 - RabbitMQ
 - Gemfire
 - Resources, etc.
- NoSQL
 - Redis, MongoDB, Riak, etc.



High Profile Sites

Walmart  MP3 Music Downloads

M Northwestern Memorial[®]
Hospital

eHarmony[®]

#hashable^{BETA}

NETFLIX

manymoon

sky

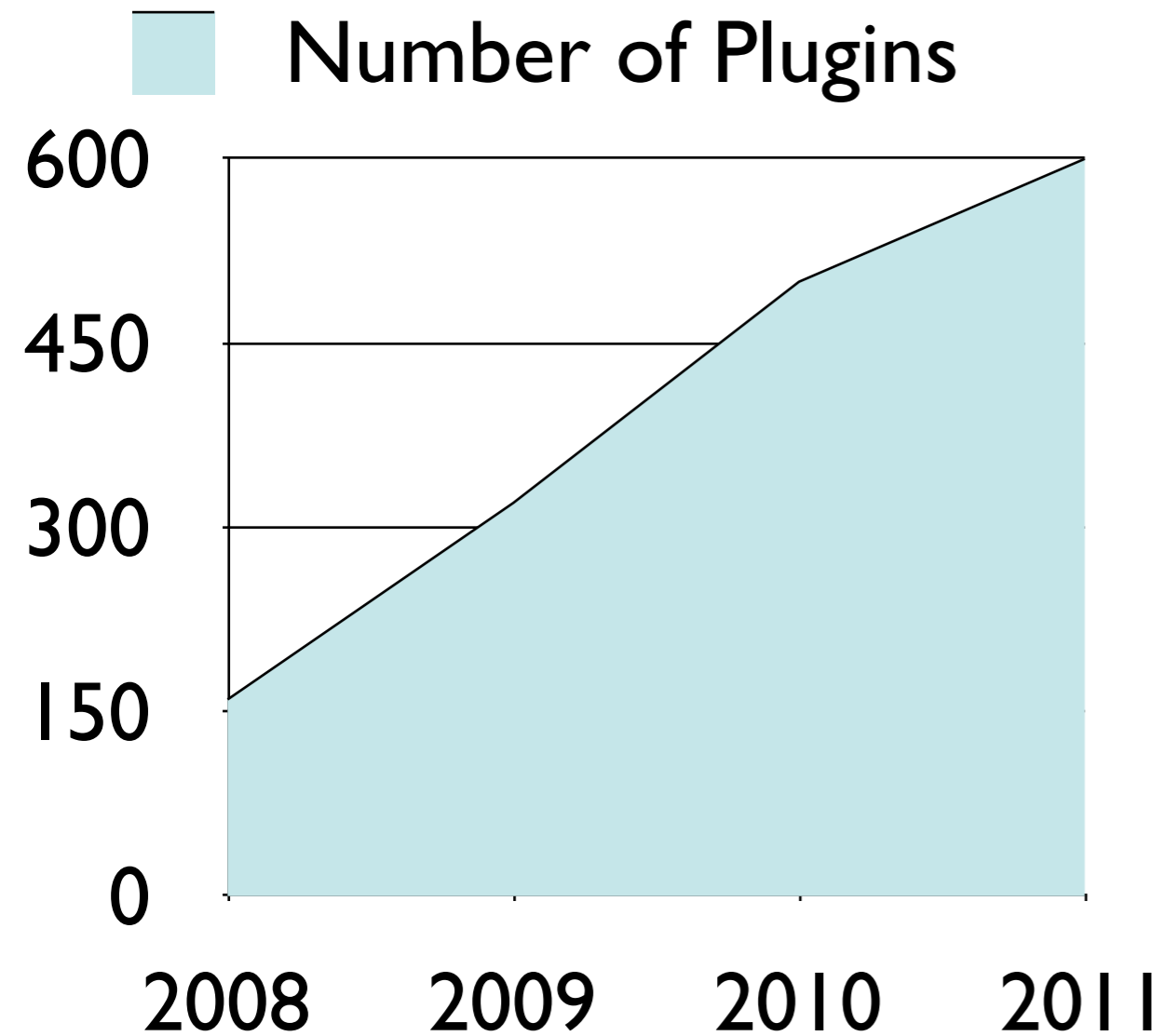
BIGLOTS!



Vodafone Music

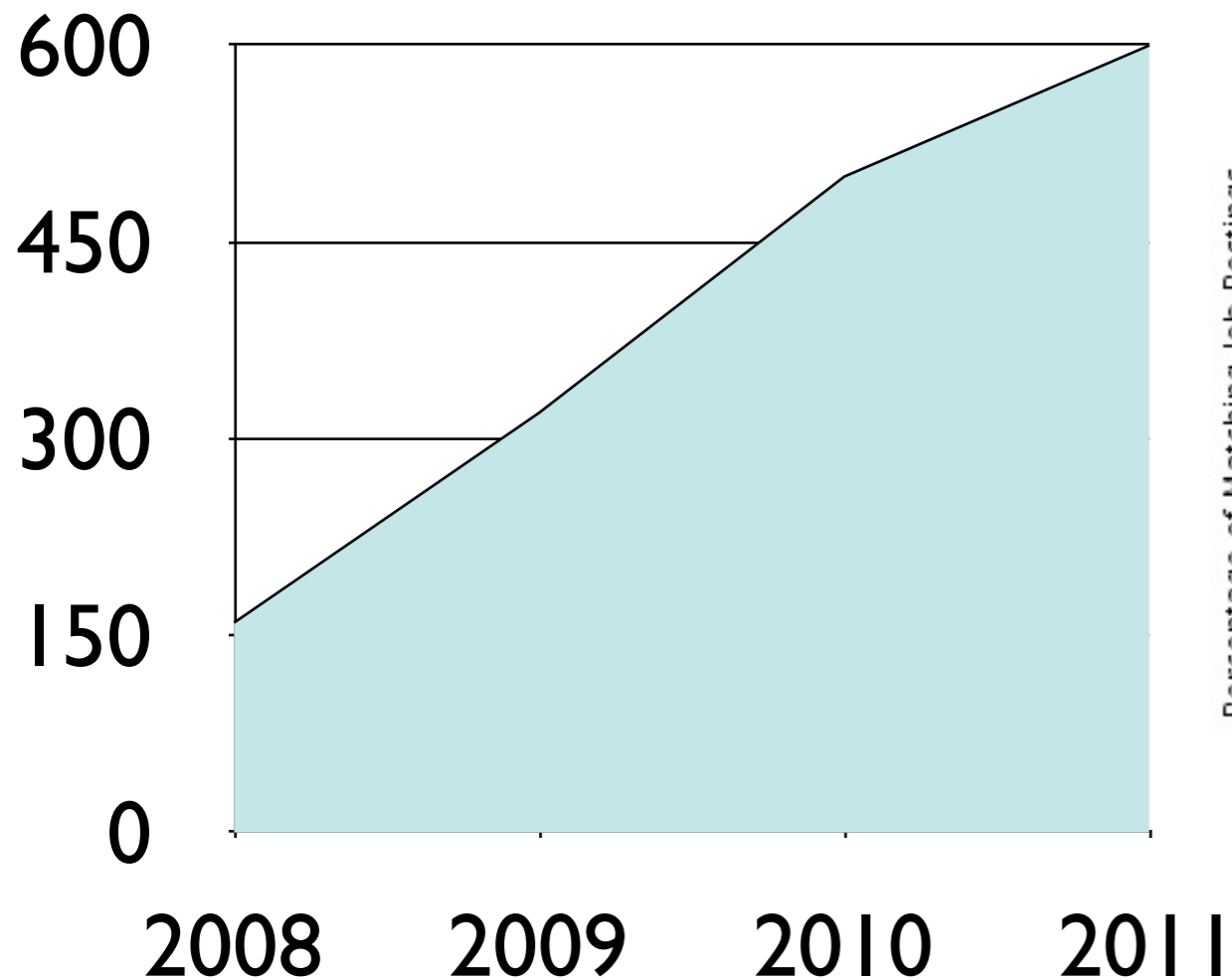
Linked 

Grails Continued Growth



Grails Continued Growth

■ Number of Plugins



What's new in Grails 2.0?

Development Environment Features

New Console UI & Interactive Mode

```
bookstore — java — 74x22
bash java bash bash
| Running 2 unit tests... 1 of 2
| Failure: testFindBook(bookstore.BookControllerTests)
| Assertion failed:
|
| assert model.book.title == "The Shining"
|           |           |           |
|           |           |           false
|           |           The Stand
|           bookstore.Book : 1
|           [book:bookstore.Book : 1]
|
|           at bookstore.BookControllerTests.testFindBook(BookControllerTests.
groovy:23)
| Completed 2 unit tests, 1 failed in 103ms
| Tests FAILED - view reports in target/test-reports
grails>
```

Better Unit Test Template



Unit Test Results - Summary

Executed 12 tests with 4 failures .

test

Executed 12 tests with 4 failures .

✘ BookControllerTests ✔ BookTests

✘ testShow

Executed in 0.12 seconds.


Assertion failed: assert book.sav

```
junit.framework.AssertionFailedError
assert book.save() != null
|      |      |
|      null  false
test.Book : null

at test.BookController1
at TestApp$_run_closure
at TestApp$_run_closure
at TestApp$_run_closure
```

Better Documentation Template

Table of contents | Quick Reference



See the light - agile, industrial strength, rapid web application development made easy

The Grails Framework - Reference Documentation

Authors: Graeme Rocher, Peter Ledbrook, Marc Palmer, Jeff Brown, Luke Daley, Burt Beckwith
Version: 1.4.0.M1

Table of Contents

1. Introduction
2. Getting Started
 - 2.1 Downloading and Installing
 - 2.2 Upgrading from previous versions of Grails
 - 2.3 Creating an Application
 - 2.4 A Hello World Example
 - 2.5 Getting Set-up in an IDE
 - 2.6 Convention over Configuration
 - 2.7 Running an Application

Quick Reference (hide)

- Command Line
- Constraints
- Controllers
- Database Mapping
- Domain Classes
- Plug-ins
- Services
- Servlet API
- Tag Libraries
- Tags

Enhanced Error Reporting



Error 500: Internal Server Error

URI: /bookstore/book/find

Class: groovy.lang.MissingPropertyException

Message: No such property: titl for class: bookstore.BookService

Around line 6 of *grails-app/services/bookstore/BookService.groovy*

```
3: class BookService {
4:
5:     Book findByTitle(String title) {
6:         Book.findByTitle(titl)
7:     }
8: }
```

Around line 10 of *grails-app/controllers/bookstore/BookController.groovy*

```
7:     def bookService
8:     def find() {
9:
10:         def b = bookService.findByTitle(params.title)
11:
12:         [book:b]
```

H2 Console

- Available at <http://localhost:8080/app/dbconsole> in development only!

The screenshot displays the H2 Console interface. At the top, there is a toolbar with icons for undo, redo, and a checked 'Auto commit' box. It also includes a 'Max rows' dropdown set to 1000, a green play button, a red stop button, and an 'Auto complete' dropdown set to 'Normal'. Below the toolbar, the left sidebar shows a tree view of the database 'jdbc:h2:mem:devDb'. The tree includes a 'BOOK' table with columns 'ID', 'VERSION', and 'TITLE', and an 'Indexes' section. Other database objects like 'INFORMATION_SCHEMA', 'Sequences', and 'Users' are also visible. The main area on the right has a 'Run (Ctrl+Enter)' button, a 'Clear' button, and a text input for the 'SQL statement:'. Below this, the executed query 'select * from book;' is shown, followed by a table with columns 'ID', 'VERSION', and 'TITLE'. The result indicates '(no rows, 3 ms)'. An 'Edit' button is located below the result.

Upgraded Libraries



New Automatic Reloading



New Automatic Reloading

- Reloading in run-app works with
 - Typed service references
 - Domain classes
 - src/groovy, src/java

Binary Plugins

- Package pre-compiled plugins into JAR files
- Deployable as standard JARs to Maven repositories
- Declared as JAR dependencies
- Commercial plugins more viable
- No special IDE integration needed

```
$ grails package-plugin --binary
```

Web Features

Methods as Actions and Binding Arguments

- Actions are now declared as public methods

```
def save(String name, int age) {  
    // remaining  
}
```

- Form parameters bound to method arguments

Methods as Actions and Binding Arguments

- Actions are now declared as public methods

```
def save(String name, int age) {  
    // remaining  
}
```

- Form parameters bound to method arguments

```
<g:form name="myForm" action="save">  
    <input name="name" />  
    <input name="age" />  
</g:form>
```

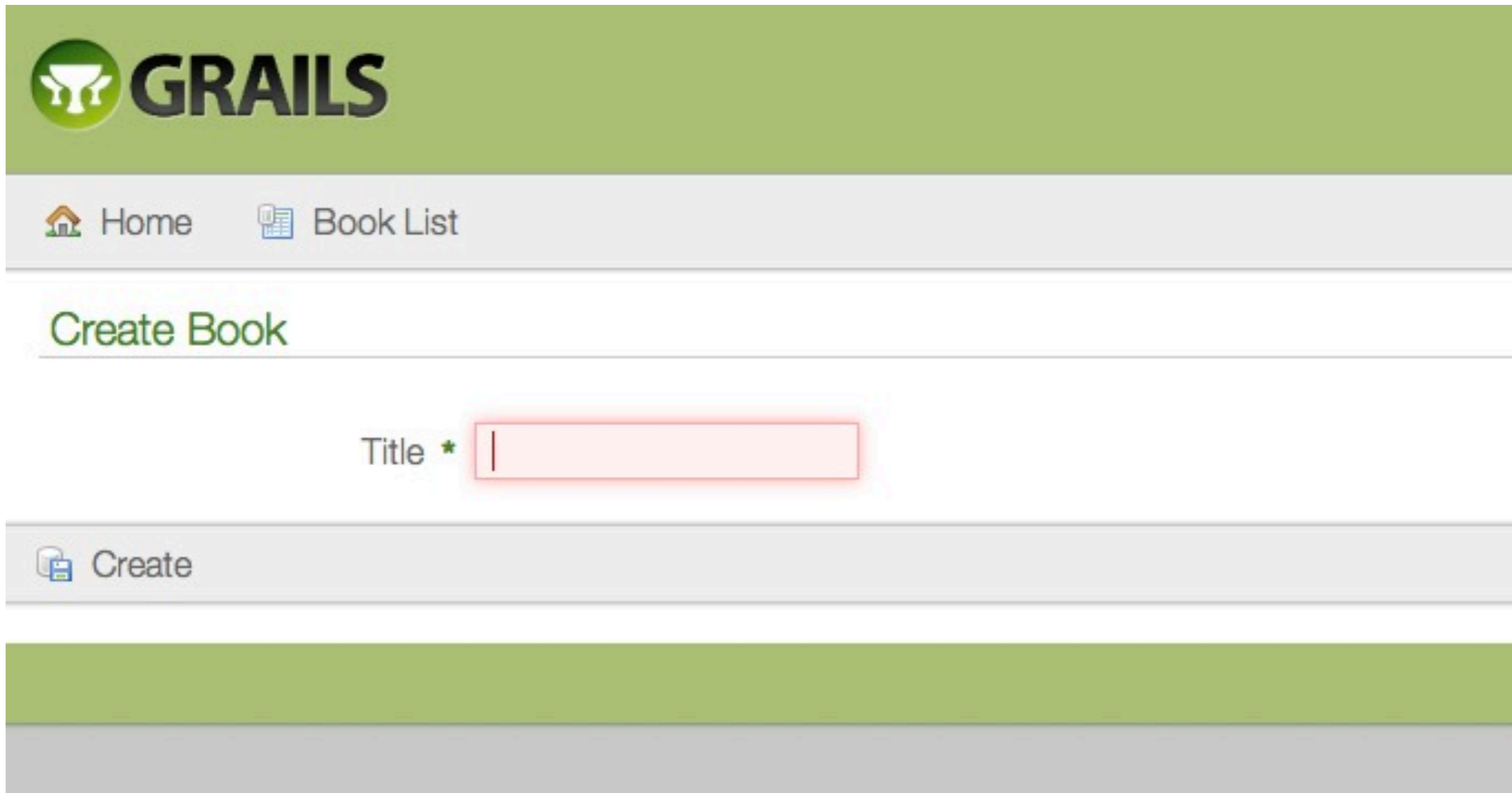

Methods as Actions and Binding Arguments

- Actions are now declared as public methods

```
def save(String name, int age) {  
    // remaining  
}
```

- Form parameters bound to method arguments

HTML5 Scaffolding



The screenshot shows a web application interface for creating a book. At the top left is the Grails logo, which consists of a green circle containing a white icon of a chalice or similar vessel, followed by the word "GRAILS" in a bold, dark grey sans-serif font. Below the logo is a horizontal navigation bar with two items: "Home" with a small house icon and "Book List" with a small document icon. The main content area is titled "Create Book" in a green font. Below the title is a form field for "Title" with a red asterisk indicating it is required. The input field is currently empty and has a light red border. At the bottom of the form is a "Create" button with a small document icon. The interface uses a color palette of green, grey, and white.

New APIs

- Page Rendering

```
PageRenderer renderer
void welcomeUser(User user) {
    def contents = renderer.render(view: "/emails/welcome",
                                   model: [user: user])
    ...
}
```

- Link Generation

```
LinkGenerator generator
def generateLink() {
    generator.link(controller: "book", action: "list")
}
```

Other Web Novelties

- jQuery now the default
- Easy Date Parsing

- Customizable URL formats
- Filter exclusions

Other Web Novelties

- jQuery now the default
- Easy Date Parsing

```
def val =  
    params.date( 'myDate', 'dd-MM-yyyy' )
```

- Customizable URL formats
- Filter exclusions

Other Web Novelties

- jQuery now the default
- Easy Date Parsing

- Customizable URL formats
- Filter exclusions

Persistence Features

GORM API

GORM API



mongoDB

riak



Cassandra



redis



HIBERNATE

GORM API



mongoDB

riak



Cassandra



redis



HIBERNATE

- Plugins should not assume Hibernate!

GORM Plugins

- Redis - <http://grails.org/plugin/redis-gorm>
- MongoDB - <http://grails.org/plugin/mongodb>
- Amazon SimpleDB - <http://grails.org/plugin/simpledb>
- Neo4j - <http://grails.org/plugin/neo4j>
- Riak - <http://grails.org/plugin/riak>
- GORM JPA - <http://grails.org/plugin/gorm-jpa>
- Hibernate - <http://grails.org/plugin/hibernate>

Where Queries

- New, compile-time checked query DSL
- Uses native Groovy operators `==`, `!=`, `>`, `<`, `<=`, `>=` etc
- Aggregate functions supported `avg`, `sum`, `max`, `min` etc.

Where Queries

- New, compile-time checked query DSL

```
def query = Person.where {  
    firstName == "Bart"  
}  
Person bart = query.find()
```

- Uses native Groovy operators ==, !=, >, <, <=, >= etc
- Aggregate functions supported avg, sum, max, min etc.

Where Queries

- New, compile-time checked query DSL

```
def query = Person.where {  
    firstName == "Bart"  
}  
Person bart = query.find()
```

- Uses native Groovy operators ==, !=, >, <, <=, >= etc

```
def query = Person.where {  
    firstName == "Fred" && !(lastName == 'Simpson')  
}
```

- Aggregate functions supported avg, sum, max, min etc.

Where Queries

- New, compile-time checked query DSL

```
def query = Person.where {  
    firstName == "Bart"  
}  
Person bart = query.find()
```

- Uses native Groovy operators ==, !=, >, <, <=, >= etc

```
def query = Person.where {  
    firstName == "Fred" && !(lastName == 'Simpson')  
}
```

- Aggregate functions supported avg, sum, max, min etc.

```
def query = Person.where {  
    age > avg(age)  
}
```


Multiple Data Sources

- Support for defining multiple scoped data sources

```
class ZipCode {  
    String code  
    static mapping = {  
        datasource 'auditing'  
    }  
}
```

- Each data source accessible via static property

Multiple Data Sources

- Support for defining multiple scoped data sources

```
class ZipCode {  
    String code  
    static mapping = {  
        datasource 'auditing'  
    }  
}
```

- Each data source accessible via static property

```
def zipCode = ZipCode.auditing.get(42)
```

Other GORM Improvements

Other GORM Improvements

- Abstract base domain classes
 - These now result in a table

Other GORM Improvements

- Abstract base domain classes
 - These now result in a table
- `findOrCreateWhere()`

Other GORM Improvements

- Abstract base domain classes
 - These now result in a table
- `findOrCreateWhere()`
- `findOrCreateSaveWhere()`

Other GORM Improvements

- Abstract base domain classes
 - These now result in a table
- findOrCreateWhere()
- findOrCreateSaveWhere()

```
def user = User.findByLogin('admin')
if (!user) {
    user = new User(login: 'admin')
    user.save(failOnError: true)
}
```

```
def user = User.findOrCreateSaveWhere(login: 'admin')
```

Better Unit Testing

Unit Testing 1.x

Unit Testing 1.x

- `mockDomain()` had only partial GORM support
 - always lagged changes in GORM

Unit Testing 1.x

- mockDomain() had only partial GORM support
 - always lagged changes in GORM
- Inheritance-based
 - hierarchy duplicated for Spock
 - difficult to extend

Unit Testing 1.x

- `mockDomain()` had only partial GORM support
 - always lagged changes in GORM
- Inheritance-based
 - hierarchy duplicated for Spock
 - difficult to extend
- Weak support for web-related testing
 - controllers
 - tag libraries

The Mixin Approach

```
@TestFor(MyController)
@Mock(Person)
class MyControllerUnitTests {
    void setUp() {
        new Person(...).save()
        new Person(...).save()
    }

    void testIndex() {
        def model = this.controller.index()
        ...
    }
}
```

Spockified

```
@TestFor(MyController)
@Mock(Person)
class MyControllerUnitTests extends Specification {
    void loadPeople() {
        new Person(...).save()
        new Person(...).save()
    }

    def 'Test index action' () {
        given: 'Some people'
            loadPeople()
        when: 'The index action is called'
            def model = this.controller.index()
        then: 'The people variable is in the model'
            model.people != null
    }
}
```

In-Memory GORM

- Full GORM implementation against ConcurrentHashMap
- Based on GORM for NoSQL codebase
- Support for
 - Criteria queries
 - Where queries
 - Dynamic finders
 - Detached criteria

Contributions

Contributions

- 110+ pull requests on grails-core

Contributions

- 110+ pull requests on grails-core
- 60+ pull requests on grails-docs

Contributions

- 110+ pull requests on grails-core
- 60+ pull requests on grails-docs
- More and more plugins

Contributions

- 110+ pull requests on grails-core
- 60+ pull requests on grails-docs
- More and more plugins
- GitHub for the win!
 - grails-core
 - grails-docs
 - grails-website
 - grails-maven
 - and many, many plugins

Grails in the Cloud

Grails in the Cloud



CLOUD FOUNDRY™

Grails in the Cloud



CLOUD FOUNDRY™



Grails in the Cloud



CLOUD FOUNDRY™



Roadmap

MODULARITY

GORM
FOR
REST

NAMESPACES

GORM
FOR
NoSQL

SECURITY
ABSTRACTION

MULTI
PROJECT
BUILDS

MORE ASYNC

PROFILES

Thank you!

Questions?