

What's new in Grails 2.0?



Agenda

- What's new in Grails 2.0?
 - Development environment
 - Web Features
 - Persistence
- What's new in Grails 2.1?
- Roadmap
- Q & A

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.BookControllerTest.testShow  
at TestApp$._run_closure1  
at TestApp$._run_closure1  
at TestApp$._run_closure1  
at TestApp$._run_closure1
```

Better Documentation Template

Table of contents | Quick Reference



GRAILS

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 shows the H2 Database Console interface. At the top, there are several icons: a red key, a yellow gear, a blue checkmark labeled "Auto commit", a blue arrow, a blue square, a green play button, a red square, a blue arrow labeled "Auto complete", a dropdown menu set to "Normal", and a question mark icon. Below the toolbar, the connection URL "jdbc:h2:mem:devDb" is displayed, along with "Run (Ctrl+Enter)", "Clear", and "SQL statement:" input fields.

The left sidebar shows the database schema:

- BOOK**: Contains columns ID, VERSION, and TITLE, and an Indexes section.
- INFORMATION_SCHEMA**
- Sequences**
- Users**

At the bottom left, a status message indicates "H2 1.2.147 (2010-11-21)".

In the main query area, the SQL statement "select * from book;" is entered. The results table has columns ID, VERSION, and TITLE, with no rows present. The time taken is 3 ms. An "Edit" button is available below the results.

Upgraded Libraries



New Automatic Reloading

springsource™

A division of VMware

New Automatic Reloading

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

New Automatic Reloading

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

New Automatic Reloading

- Reloading in run-app works with
 - Typed service references
 - Domain classes
 - src/groovy, src/java
- Any command with -reloading
- Interactive mode and integration tests

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
- 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

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, there is a green header bar with the Grails logo and the word "GRAILS". Below the header, there is a navigation bar with two items: "Home" and "Book List". The main content area has a light gray background. It features a heading "Create Book" in green text. Below the heading is a form field labeled "Title *". The input field is empty and has a red border, indicating it is a required field. At the bottom of the form, there is a button labeled "Create" with a blue icon. The overall design uses a clean, modern look with a color palette of green, white, and gray.

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")
}
```

Servlet 3.0 Async

- Servlet 3.0 async API supported for async response rendering



spring source™

A division of VMware

Servlet 3.0 Async

- Servlet 3.0 async API supported for async response rendering

```
def index() {  
    def ctx = startAsync()  
    ctx.start {  
        new Book(title:"The Stand").save()  
        render template:"books",  
               model:[books: Book.list()]  
        ctx.complete()  
    }  
}
```

Servlet 3.0 Async

- Servlet 3.0 async API supported for async response rendering

Advanced Static Resource Handling

- Integrated resource plugin into core
 - <http://grails.org/plugin/resources>
- Tuning static resources no longer a headache
 - gzip (<http://grails.org/plugin/zipped-resources>)
 - cache (<http://grails.org/plugin/cached-resources>)
 - de-duplication
 - bundling
- New tags to ease integration
 - img
 - external
 - javascript

Bundling Static Resources

springsource™

A division of VMware

Bundling Static Resources

```
modules = {
    core {
        dependsOn 'utils'
        resource url: '/js/core.js', disposition: 'head'
        resource url: '/js/ui.js'
        resource url: '/css/main.css',
        resource url: '/css/branding.css'
        resource url: '/css/print.css', attrs:[media:'print']
    }
    utils {
        dependsOn 'jquery'
        resource url: '/js/utils.js'
    }
}
```

Bundling Static Resources

springsource™

A division of VMware

Zipping and Caching

springsource™

A division of VMware

Zipping and Caching

```
$ grails install-plugin cached-resources  
$ grails install-plugin zipped-resources
```

Zipping and Caching

springsource™

A division of VMware

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>
- Amazon DynamoDB - <http://grails.org/plugin/dynamodb>
- 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
- 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

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)
```

SQL Database Migration

- Install the Database Migration plugin:

```
$ grails install-plugin database-migration
```

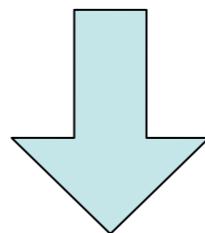
- Official Docs at:
 - <http://grails-plugins.github.com/grails-database-migration/>

SQL Database Migration

Pre-production, Hibernate ‘update’ or ‘create-drop’

SQL Database Migration

Pre-production, Hibernate ‘update’ or ‘create-drop’



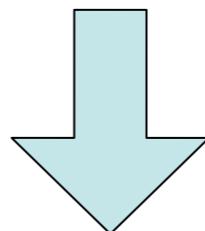
dbm-generate-changelog
dbm-changelog-sync

Change domain model

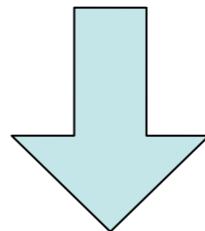
dbm-gorm-diff
dbm-update

SQL Database Migration

Pre-production, Hibernate ‘update’ or ‘create-drop’



**dbm-generate-changelog
dbm-changelog-sync**

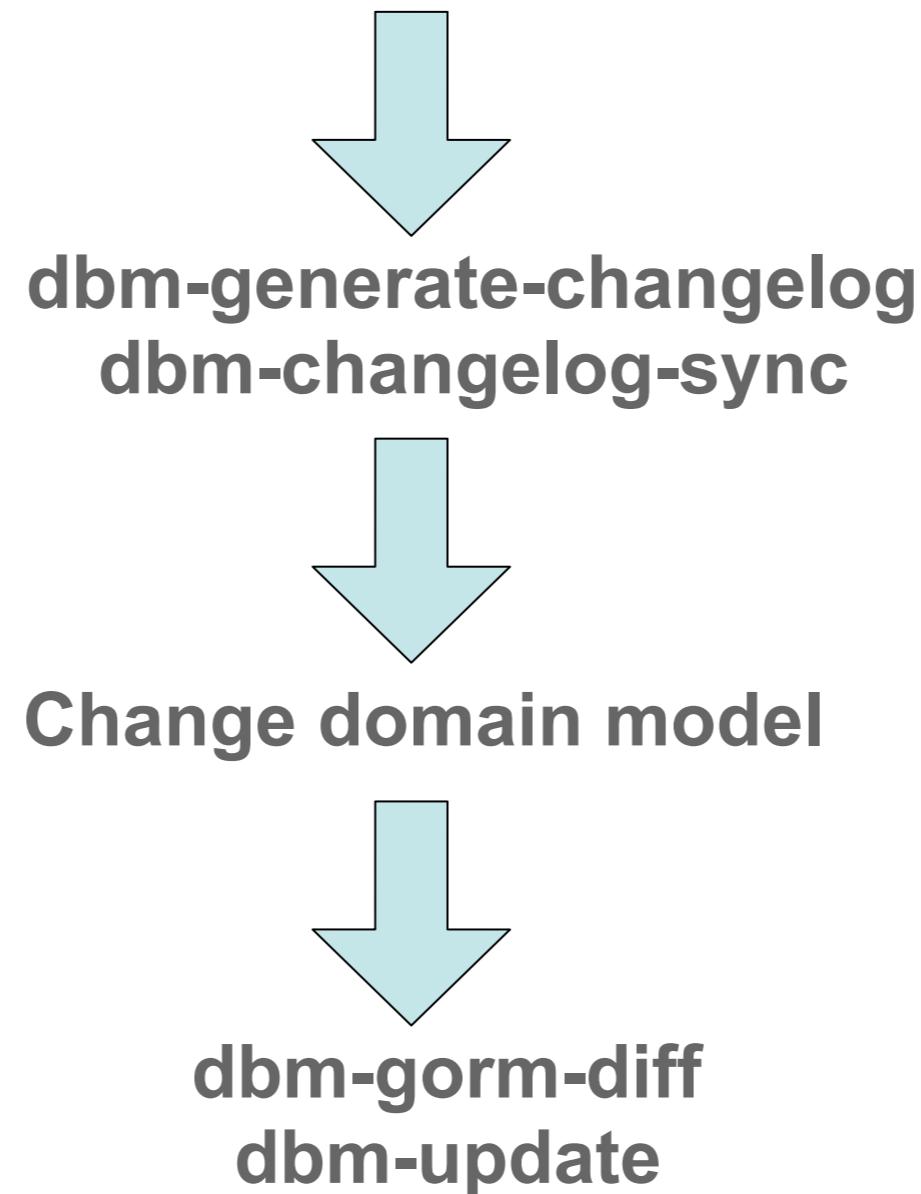


Change domain model

**dbm-gorm-diff
dbm-update**

SQL Database Migration

Pre-production, Hibernate 'update' or 'create-drop'



Other GORM Improvements

Other GORM Improvements

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



springsource®
A division of VMware

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()`
- `findOrSaveWhere()`

Other GORM Improvements

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

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

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

Better Unit Testing



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

Support for testing...

springsource™

A division of VMware

Support for testing...

- Tag libraries

Support for testing...

- Tag libraries
- Command objects

Support for testing...

- Tag libraries
- Command objects
- XML & JSON responses

Support for testing...

- Tag libraries
- Command objects
- XML & JSON responses
- File upload

Support for testing...

- Tag libraries
- Command objects
- XML & JSON responses
- File upload
- View and template rendering

Support for testing...

- Tag libraries
- Command objects
- XML & JSON responses
- File upload
- View and template rendering
- Filters

Support for testing...

- Tag libraries
- Command objects
- XML & JSON responses
- File upload
- View and template rendering
- Filters
- URL mappings

Support for testing...

- Tag libraries
- Command objects
- XML & JSON responses
- File upload
- View and template rendering
- Filters
- URL mappings
- Criteria queries

Support for testing...

- Tag libraries
- Command objects
- XML & JSON responses
- File upload
- View and template rendering
- Filters
- URL mappings
- Criteria queries
- and more!

What's new in Grails 2.1?



Maven Plugin 2.1

- New Maven Compatible Plugin Repository
 - <http://repo.grails.org/grails/plugins>
- Define plugins in your POM:

```
<dependency>
    <groupId>org.grails.plugins</groupId>
    <artifactId>hibernate</artifactId>
    <version>2.1.0.RC2</version>
    <type>zip</type>
    <scope>compile</scope>
</dependency>
```

Maven Plugin 2.1

- New command line commands to ease Maven build setup

```
$ grails create-app myapp  
$ cd myapp  
$ grails create-pom com.mycompany  
$ mvn package
```

- Multi-module builds with “create-multi-project-build”

```
$ grails create-app myapp  
$ grails create-plugin plugin-a  
$ grails create-plugin plugin-b  
$ grails c-m-p-b foo:parent:1.0-SNAPSHOT  
$ mvn install
```

Maven Plugin 2.1

- Forked Grails Command Execution

```
<plugin>
  <groupId>org.grails</groupId>
  <artifactId>grails-maven-plugin</artifactId>
  <version>${grails.version}</version>
  <configuration>
    <fork>true</fork>
  </configuration>
  <extensions>true</extensions>
</plugin>
```

Cache Plugin

- Service Method Caching

```
import grails.plugin.cache.CacheEvict
import grails.plugin.cache.Cacheable

class MessageService {
    @Cacheable('message')
    Message getMessage(String title) {
        println 'Fetching message'
        Message.findByTitle(title)
    }
}
```

Cache Plugin

- Controller Content Caching

```
class TestController {  
    @Cacheable('message')  
    def lookup() {  
        println "called 'lookup'"  
    }  
  
    @CacheEvict(value='message',allEntries=true)  
    def evict() {  
        println "called 'evict'"  
    }  
}
```

Cache Plugin

- Cache Tags

```
<cache:block>
    <!-- Any valid markup -->
</cache:block>

<cache:render
    template="myTemplate"
    model=" [ name: 'Some Value' ] "/>
```

Cache Plugin

- Core Plugin
 - <http://grails.org/plugin/cache>
- Implementations
 - <http://grails.org/plugin/cache-ehcache>
 - <http://grails.org/plugin/cache-redis>
 - <http://grails.org/plugin/cache-gemfire>
- Grails 2.0.x compatible
- Cache Configuration DSL

Other Grails 2.1 Improvements

- Grails Wrapper

```
$ grails wrapper  
$ grailsw run-app
```

- Command Aliases

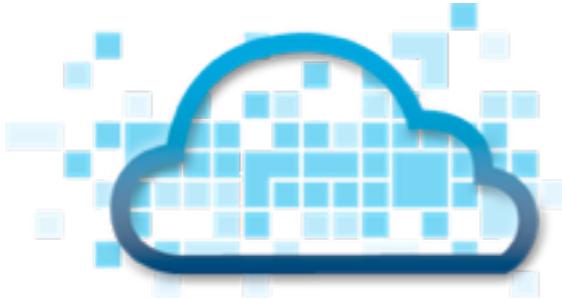
```
$ grails alias rit test-app integration:  
$ grailsw rit
```

Grails in the Cloud

springsource™

A division of VMware

Grails in the Cloud



CLOUD FOUNDRY™

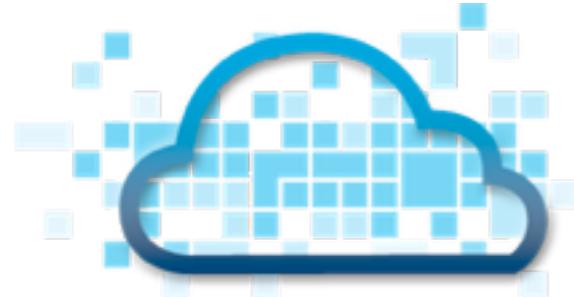
Grails in the Cloud



CLOUD FOUNDRY™



Grails in the Cloud



CLOUD FOUNDRY™



A division of VMware

Roadmap

MODULARITY

GORM
FOR
REST

NAMESPACES

GORM
FOR
NoSQL

SECURITY
ABSTRACTION

MULTI
PROJECT
BUILDS

MORE ASYNC

PROFILES

Roadmap

- **Grails 2.1**
 - Maven
 - Cache Plugin
- **Grails 2.2**
 - Build Isolation
 - Namespace Support
 - Groovy 2
- **Grails 2.3**
 - REST
 - Data Binding
 - GORM for REST
 - Async
- **Grails 3.0**
 - Gradle Integration
 - Application Profiles

Thank you!

Questions?