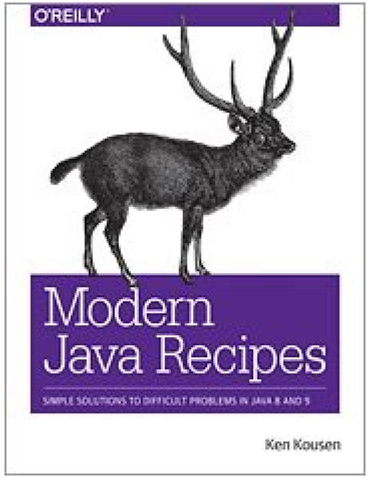
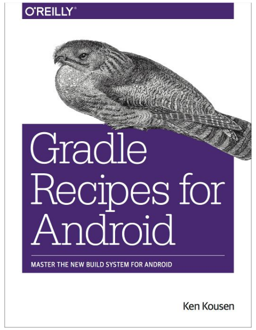
Grails 3

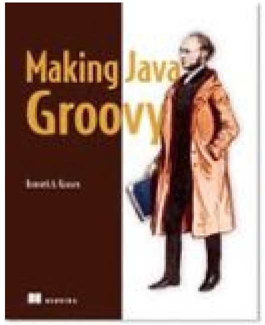
Evolving the Framework

Contact Info

Ken Kousen
Kousen IT, Inc.
ken.kousen@kousenit.com
http://www.kousenit.com
http://kousenit.org (blog)
@kenkousen




O'REILLY
Gradle Recipes for Android
MASTER THE NEW BUILD SYSTEM FOR ANDROID
Ken Kousen

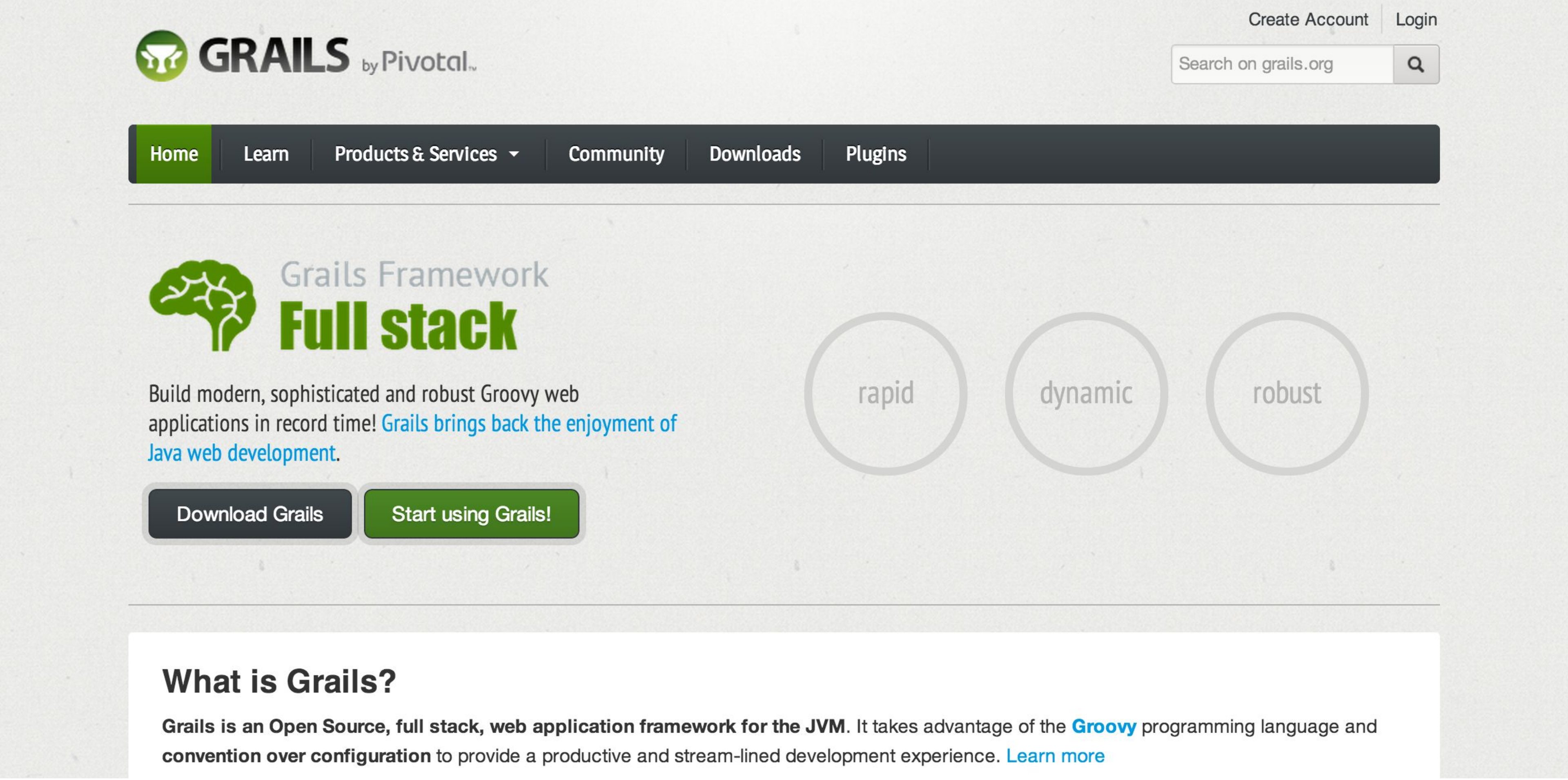


Publications

O'Reilly video courses at Safari Books Online

- Groovy Programming Fundamentals
- Practical Groovy Programming
- Mastering Groovy Programming
- Learning Android
- Practical Android
- Gradle Fundamentals
- Gradle for Android
- Advanced Java Development
- Spring Framework Essentials

Also several on Grails 3



Build modern, sophisticated and robust Groovy web applications in record time! Grails brings back the enjoyment of Java web development.

What is Grails?

Grails is an Open Source, full stack, web application framework for the JVM. It takes advantage of the **Groovy** programming language and **convention over configuration** to provide a productive and stream-lined development experience. Learn more

Grails home page, http://grails.org

Grails

Complete stack framework

Grails

Complete stack framework

    from web server to middleware to DB

Grails

Complete stack framework

- from web server to middleware to DB

Convention over configuration

Installation

Download `grails-x.y.z.zip` and unzip

Installation

Download `grails-x.y.z.zip` and unzip

Set `GRAILS_HOME`

Installation

Download `grails-x.y.z.zip` and unzip

Set `GRAILS_HOME`

Add `bin` folder to path

Installation

No Groovy install required

Installation

No Groovy install required

Grails includes Groovy

Installation

No Groovy install required

Grails includes Groovy

and you can't change the version

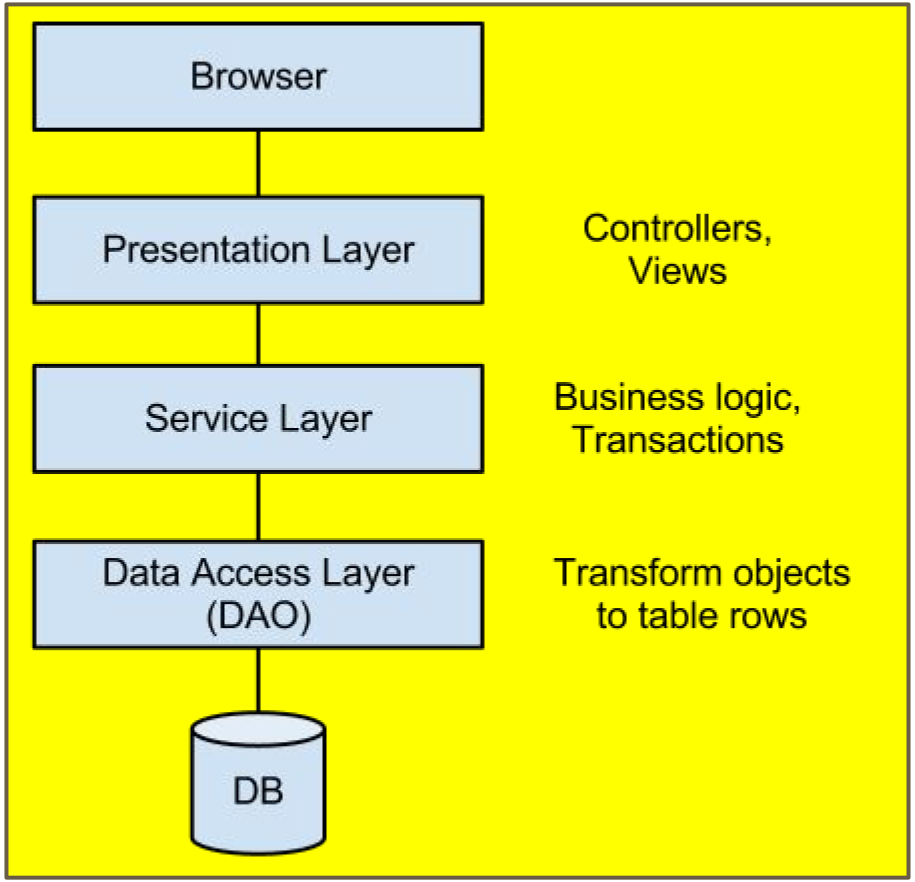
MVC in Grails

domain → persist to DB

controller → map to URL

view → display

service → transactions and business logic



Every Java web app ever

Layered architecture

Presentation layer as expected

Layered architecture

Presentation layer as expected

- controllers and views

Layered architecture

Presentation layer as expected

- controllers and views

Service layer as expected

Layered architecture

Presentation layer as expected

- controllers and views

Service layer as expected

- transactions and business logic

Layered architecture

Presentation layer as expected

- controllers and views

Service layer as expected

- transactions and business logic
- (managed by Spring)

Controllers

URL maps to controller actions

Three ways to finish an action

1. *render*
2. *redirect*
3. *return*

Render

Render $\longrightarrow$ write to output stream

Redirect

Redirect → generate new URL for browser

- Creates new request
- Existing parameters are lost

Return

Return → Add map entries to request

Forward to:

```
grails-app/views/controller/action.gsp
```

Respond

Grails 2.3 introduced *respond*

chooses appropriate output based on

content negotiation

Convention

Default URL Mapping:

```
    http://<server>:<port>
        /controller
            /action
                /id
```

Layered architecture

- Persistence layer is different
    - **Active Record** design pattern

Active Record

DAO methods added to domain classes

```
product.save(), product.delete()

Product.findAllByNameLike("...")
```

`Uses Groovy metaprogramming`

`SQL generated by Hibernate`

Save to database

Three steps in saving an object

1. *binding*
2. *validation*
3. *persistence*

Binding

Populate object from input data

New data binding framework

Validation

Check object properties against constraints

`constraints` closure in domain class

Persistence

`save()` method on domain class

`save()` calls `validate()`

- !valid → save returns null
- valid → save returns object

Testing

Grails uses Spock by default

http://spockframework.org

Tests extend

```
spock.lang.Specification
```

Testing

`@TestFor` annotation

For controllers and services:

instantiates and provides reference

Testing

Tests provide `params` map

Holds request parameters

Mapping Domain

Default relational database

Class name → table name

attributes → column names

constraints may affect schema generation

Existing DB

Can map to existing databases

```
static mapping = {
    table ‘people’
    first column:’first_name’
}
```

GORM

Grails Object Request Mapping

Capt. Kirk struggles with Grails Object Relational Napping

GORM

- Auto-generated methods
  - dynamic finders
  - criteria queries
  - static methods on domain class

dbconsole

Browse database

(development mode only)

http://.../dbconsole

Services

Transactional by default

Use Spring’s `@Transactional`

Gradle Accommodations

New file system locations

`grails-app/conf/BuildConfig.groovy`	→	`build.gradle`
`grails-app/conf/Config.groovy`	→	`grails-app/conf/application.groovy`
`grails-app/conf/BootStrap.groovy`	→	`grails-app/init/BootStrap.groovy`
`src/groovy, src/java`	→	`src/main/groovy`

Gradle Accommodations

More file system changes:

`test/unit`	→	`src/test/groovy`
`test/integration`	→	`src/integration-test/groovy`
`web-app`	→	`src/main/webapp, src/main/resources`

New Files in Grails 3

`build.gradle`	→ Gradle build file
`gradle.properties`	
`grails-app/conf/logback.groovy`	→ new logging system
`grails-app/conf/application.yml`	→ alternative config file
`grails-app/init/<package>`	
`    /Application.groovy`	→ Spring Boot execute app

Removed

No longer needed files

`application.properties`	→ now in gradle.properties
`grails-app/conf/DataSource.groovy`	→ merged into application.yml

API changes

Filters no longer supported → Use Interceptor API instead

Geb plugin installed by default

- New **`create-functional-test`** command

No more Gant (!)

- Everything is Gradle tasks now

Scaffolding

- Dynamic scaffolding removed in 3.0
  - (Restored in 3.0.4)
- Static scaffolding still available
  - Uses the fields plugin

Migration Path

1. Make a new Grails 3 app
2. Copy your 2.* files to the corresponding 3.* locations
   a. Domain classes, controllers, services
   b. Tests to new locations
3. Move BuildConfig.groovy dependencies to build.gradle
4. Move Config.groovy settings to application.yml
5. Move DataSource.groovy settings to application.yml
6. Delete files no longer used

JDBC drivers

Strong preference to use repositories

For drivers not available that way, two alternatives:

1. Add a lib folder, then
   `compile fileTree(dir: 'lib', include: '*.jar')`
2. `Use local repo`
   a. `Push jars to local repo`
      `http://www.mkyong.com/maven/how-to-add-oracle-jdbc-driver-in-your-maven-local-repository/`
   b. `Add mavenRepo to build.gradle`
      ``runtime: "oracle.com:ojdbc6:11.2.0"` ``

Plugins

Good news: Most of the popular ones have been ported

Before you ask:

Spring Security Core now 3.1.1

Packages

Codehaus is now gone

Internal APIs now in org.grails.*

Public facing APIs now in grails.*

Gradle

Grails generates gradlew scripts

- Don't need to install Gradle locally

Can use your own, if Gradle 2.2+

The "grails" command now invokes the bundled "gradle"

Gradle

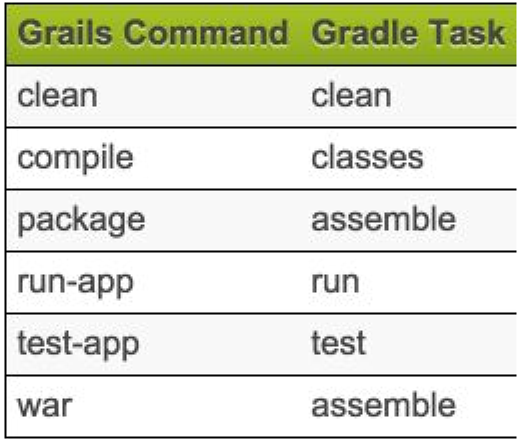
Most Grails dependencies don't have version numbers

```
dependencies {
    compile 'org.springframework.boot:spring-boot-starter-logging'
    compile('org.springframework.boot:spring-boot-starter-actuator')
…
```

Versions set by default to Grails version

```
dependencyManagement {
    imports {
        mavenBom 'org.grails:grails-bom:' + grailsVersion
    }
    applyMavenExclusions false
}
```

Gradle tasks



Grails Command	Gradle Task
clean	clean
compile	classes
package	assemble
run-app	run
test-app	test
war	assemble

Easy to just use the Grails tasks as before

Profiles

Grails default is the "web" profile

Use the `--profile="..."` flag for alternatives

- Profiles hosted on GitHub
  - https://github.com/grails/grails-profile-repository

Not much documentation for them yet

Functional Tests

Grails uses Geb for functional tests

http://www.gebish.org/

- Browser automation
- Uses WebDriver for cross-browser compatibility
- jQuery-like selector syntax
- Page Object model
- Spock integration

See the Book of Geb for details

References

Grails home page: http://grails.org

User Guide: https://grails.org/single-page-documentation.html

Note: https://grails.github.io/grails-doc/latest/ is same,

change "latest" to version you want

Grails API: https://grails.org/api.html

Slack channel: http://slack-signup.grails.org/