



OCI | HOME TO GRAILS

Getting started with Grails 3

Colin Harrington

GR8Conf US 2016

whoami > Colin Harrington

- Grails team at OCI
- [@ColinHarrington](#)
- harringtonc@ociweb.com



OCI | HOME TO GRAILS



Grails is a **powerful** web framework, for the Java platform aimed at multiplying developers' productivity thanks to a Convention-over-Configuration, sensible defaults and opinionated APIs. It integrates smoothly with the JVM, allowing you to be immediately productive whilst providing powerful features, including integrated ORM, **Domain-Specific Languages**, runtime and compile-time **meta-programming** and **Asynchronous** programming.

History

Timeline

- Started in 2005
- 0.1 release on March 29, 2006
- Grails 1.0 => February 2008
- Grails 2.0 => 2011
- Grails 3.0 => March 2015

Support

Getting Started

Installation

sdkman

- <http://sdkman.io/>
- `$ curl -s "https://get.sdkman.io" | bash`



Manual Installation

Gradle

Gradle

```
> $ tree -L 1
.
├── build
├── build.gradle
├── gradle
├── gradle.properties
├── gradlew
├── gradlew.bat
├── grails-app
└── src
```

build.gradle

Gradle plugins

Repositories

- Bintray!
- Dependencies
- Grails

buildscript block

Application Profiles

```
grails create-app myapp --profile=rest-api
```

Application Profiles

```
> $ grails list-profiles
```

```
| Available Profiles
```

```
-----
```

- * angular - A profile for creating applications using AngularJS
- * rest-api - Profile for REST API applications
- * base - The base profile extended by other profiles
- * plugin - Profile for plugins designed to work across all profiles
- * web - Profile for Web applications
- * web-plugin - Profile for Plugins designed for Web applications

```
> $ ./gradlew tasks
:tasks
```

```
-----
All tasks runnable from root project
-----
```

Application tasks

```
-----
bootRun - Run the project with support for auto-detecting main class and reloading static resources
```

Build tasks

```
-----
assemble - Assembles the outputs of this project.
bootRepackage - Repackage existing JAR and WAR archives so that they can be executed from the
build - Assembles and tests this project.
buildDependents - Assembles and tests this project and all projects that depend on it.
buildNeeded - Assembles and tests this project and all projects it depends on.
classes - Assembles main classes.
clean - Deletes the build directory.
integrationTestClasses - Assembles integration test classes.
jar - Assembles a jar archive containing the main classes.
testClasses - Assembles test classes.
war - Generates a war archive with all the compiled classes, the web-app content and the libraries
```

Build Setup tasks

```
-----
init - Initializes a new Gradle build. [incubating]
wrapper - Generates Gradle wrapper files. [incubating]
```

Documentation tasks

```
-----
groovydoc - Generates Groovydoc API documentation for the main source code.
javadoc - Generates Javadoc API documentation for the main source code.
```

Help tasks

buildEnvironment - Displays all buildscript dependencies declared in root project 'bookstore'.
components - Displays the components produced by root project 'bookstore'. [incubating]
dependencies - Displays all dependencies declared in root project 'bookstore'.
dependencyInsight - Displays the insight into a specific dependency in root project 'bookstore'.
help - Displays a help message.
model - Displays the configuration model of root project 'bookstore'. [incubating]
projects - Displays the sub-projects of root project 'bookstore'.
properties - Displays the properties of root project 'bookstore'.
tasks - Displays the tasks runnable from root project 'bookstore'.

IDE tasks

cleanEclipse - Cleans all Eclipse files.
cleanIdea - Cleans IDEA project files (IML, IPR)
eclipse - Generates all Eclipse files.
idea - Generates IDEA project files (IML, IPR, IWS)

Verification tasks

check - Runs all checks.
test - Runs the unit tests.

Other tasks

assetClean
assetPluginPackage
cleanIdeaWorkspace
console
dependencyManagement
mergeTestReports
pathingJar
pathingJarCommand
schemaExport

Command line
grails-shell

scaffolding

IDE

Application Structure


```
> $ tree -L 2
.
├── build.gradle
├── gradle
│   └── wrapper
├── gradle.properties
├── gradlew
├── gradlew.bat
├── grails-app
│   ├── assets
│   ├── conf
│   ├── controllers
│   ├── domain
│   ├── i18n
│   ├── init
│   ├── services
│   ├── taglib
│   ├── utils
│   └── views
└── src
    ├── integration-test
    ├── main
    └── test
```

Application.groovy

- Standalone runner
- Configurable

Configuration

application.yml

application.groovy

logback.groovy

resources.groovy

Controllers

MVC

```
class BookController {  
    def list() {  
  
        // do controller logic  
        // create model  
  
        return model  
    }  
}
```

Scopes

- [servletContext](#) - Also known as application scope, this scope lets you share state across the entire web application. The `servletContext` is an instance of [ServletContext](#)
- [session](#) - The session allows associating state with a given user and typically uses cookies to associate a session with a client. The session object is an instance of [HttpSession](#)
- [request](#) - The request object allows the storage of objects for the current request only. The request object is an instance of [HttpServletRequest](#)
- [params](#) - Mutable map of incoming request query string or POST parameters
- [flash](#) - See below

```
class BookController {  
  def find() {  
    def findBy = params["findBy"]  
    def appContext = request["foo"]  
    def loggedUser = session["logged_user"]  
  }  
}
```

data binding

RestController

```
class BookController extends RestfulController {  
    static responseFormats = ['json', 'xml']  
    BookController() {  
        super(Book)  
    }  
}
```

BookController

HTTP Method	URI	Controller Action
GET	/books	index
GET	/books/create	create
POST	/books	save
GET	/books/\${id}	show
GET	/books/\${id}/edit	edit
PUT	/books/\${id}	update
DELETE	/books/\${id}	delete

GORM

GORM is the data access toolkit used by [Grails](#) and provides a rich set of APIs for accessing relational and non-relational data including implementations for Hibernate (SQL), MongoDB, Neo4j, Cassandra, Redis and an in-memory ConcurrentHashMap for testing.

Persistence

Mapping + validation

Associations

- One-to-Many
- hasOne
- many-to-many
- embedded

Dynamic Finders

```
Book.findByTitle("The Hobbit")
```

Advanced Queries

<http://gorm.grails.org/>

- Hibernate
- MongoDB
- Neo4j
- Cassandra
- Redis

UrlMappings

UrlMappings.groovy

Examples

- `"/product"(controller: "product", action: "list")`
- `"/product"(controller: "product")`
- `"/hello"(uri: "/hello.dispatch")`
- `"/books"(resources: 'book')`
- `"/"(view: "/index")`
- Errors
- `"403" (controller: "errors" action: "forbidden")`
- wildcards, exclusions, replacement, variables, etc.

Link Generation

```
<g:link controller="book" action="index">My Link</g:link>
```

Service Layer

@Transactional

Async

```
import static grails.async.Promises.*  
  
...  
  
def p1 = task { 2 * 2 }  
def p2 = task { 4 * 4 }  
def p3 = task { 8 * 8 }  
assert [4,16,64] == waitAll(p1, p2, p3)
```

Grails Events

```
on( "myEvent" ) {  
    println "Event fired!"  
}
```

```
notify "myEvent", "myData"
```

RxGORM

RxJava

Testing

Unit Testing

- `grails test-app`
- Test Mixins
- `@TestFor(BookController)`
`@Mock([Book, Author, BookService])`

Spock

```
import grails.test.mixin.TestFor
import spock.lang.Specification

@TestFor(SimpleController)
class SimpleControllerSpec extends Specification {

    void 'test display'() {
        when:
            controller.display()

        then:
            response.text == 'contents of the template'
    }
}
```

Test Mixins

- `grails.test.mixin.TestFor`
- `grails.test.mixin.Mock`
- `grails.test.mixin.TestMixin`
- `grails.test.mixin.support.GrailsUnitTestMixin`
- `grails.test.mixin.domain.DomainClassUnitTestMixin`
- `grails.test.mixin.services.ServiceUnitTestMixin`
- `grails.test.mixin.web.ControllerUnitTestMixin`
- `grails.test.mixin.web.FiltersUnitTestMixin`
- `grails.test.mixin.web.GroovyPageUnitTestMixin`
- `grails.test.mixin.web.UrlMappingsUnitTestMixin`
- `grails.test.mixin.hibernate.HibernateTestMixin`

Integration testing

Geb

- <http://www.gebish.org/>
- Bullet Two
- Bullet Three

```
import geb.spock.GebReportingSpec

import spock.lang.*

import pages.*

@Stepwise
class PersonCRUDSpec extends GebReportingSpec {

    def "there are no people"() {
        when:
        to ListPage
        then:
        personRows.size() == 0
    }

    def "add a person"() {
        when:
        newPersonButton.click()
        then:
        at CreatePage
    }

    def "enter the details"() {
        when:
        $("#visible").click()
        firstName = "Luke"
        lastName = "Daley"
        createButton.click()
        then:
        at ShowPage
    }
}
```


Interceptors

```
$ grails create-interceptor MyInterceptor
```

```
...
```

```
class MyInterceptor {  
    boolean before() { true }  
  
    boolean after() { true }  
  
    void afterView() {  
        // no-op  
    }  
}
```

Request matching

```
class LoggingInterceptor {  
  LoggingInterceptor() {  
    match(controller:"book", action:"show") // using strings  
    match(controller: ~/(author|publisher)/) // using regex  
  }  
  boolean before() {  
    ...  
  }  
}
```

View Layer

Groovy Server Pages

Controller:

```
render(view: "index")
```

renders `index.gsp` in the `grails-app/views` directory

TagLibs

Layouts

JSON views

Grails Views

- `compile 'org.grails.plugins:views-json:1.0.0'`
- <http://views.grails.org/latest/>
- Statically Compiled
- Fast!
- Markdown views

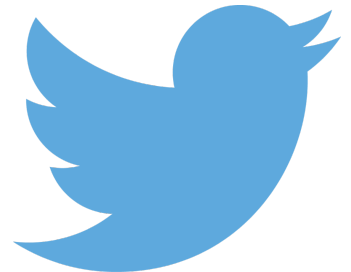
```
json.person {  
    name "bob"  
}
```

```
model {  
    Person person  
}  
json {  
    name person.name  
    age person.age  
}
```



Resources

- grails.org
- User Guide
(RTFM: Read The Fantastic Manual)
- Stack Overflow
- Slack
- Github!
- twitter



Connect with Us

grails.org

slack-signup.grails.org

grailsblog.ociwweb.com

ociweb.com/grails

[@grailsframework](#)

[@objectcomputing](#)



OCI | HOME TO GRAILS

Thank you for attending!

Please visit our table for giveaways
and to meet the team!

ociweb.com/grails
info@ociweb.com



OCI | HOME TO GRAILS