

Introduction à Grails

Alexis Plantin

Sommaire

- Qu'est ce que Grails?
- Guide de démarrage rapide
- Scaffolding

Qu'est ce que Grails?

Qu'est ce que Grails?



- Un projet Open Source démarré en 2005 par Graeme Rocher
- Framework Web (comme RoR, Symfony, Django, ...)
- Basé sur Groovy (Donc... sur la JVM)
- Dernière release: 3.1.11
- Communauté importante

Qu'est ce que Grails?



- Construit avec des socles technologiques reconnus
 - Java
 - Groovy
 - Spring
 - Hibernate
 - ...

Qu'est ce que Grails?



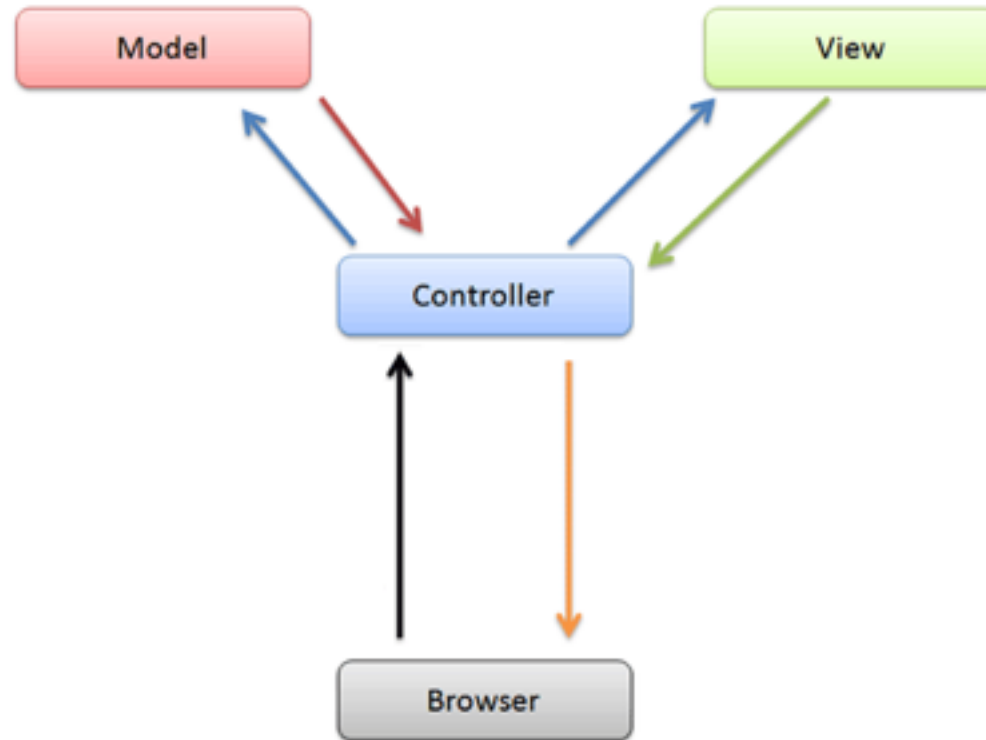
- Une architecture de plugin
- Il y a un plugin pour ça (1000+ plugins)
- Différentes catégories
 - UI
 - Sécurité
 - Reporting
 - Tests
 - ...

Qu'est ce que Grails?



- Out of the box
 - Un système H2 de base de données en mémoire
 - Tomcat comme conteneur de servlets embarqué
 - La possibilité de générer le livrable: le fichier WAR
- Pas besoin d'installer de middlewares

Model - View - Controller



La Philosophie de Grails

- Convention over Configuration
 - Des comportements par défaut
- Don't Repeat Yourself
 - Des mécanismes pour éviter les duplications de code

Guide de démarrage rapide

Grails - Commandes

- Directement depuis le prompt

```
grails [environment]* [target] [arguments]*
```

- Exemple

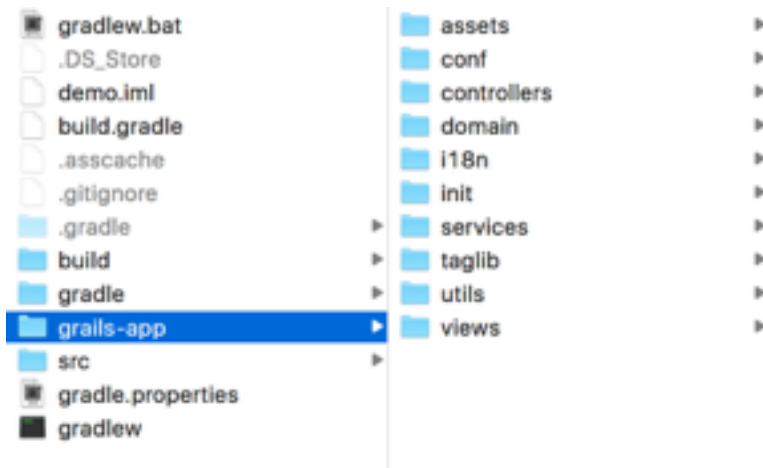
```
grails help create-app
```

```
grails create-app -- Creates a Grails application for the given  
name
```

```
Usage (optionals marked with *):  
  grails [environment]* create-app
```

Créer une application

```
grails create-app wrh
```



Observons l'architecture
de notre projet

Fichiers de configuration

- `/grails-app/conf/application.yml` pour la configuration statique de l'application et la connexion à la base de données
- `/grails-app/conf/logback.groovy` pour la gestion des logs
- `build.gradle` pour la configuration de la construction de l'application (plugins, librairies)
- `/grails-app/controllers/UrlMappings.groovy` pour les réécritures d'URL
- `/grails-app/init/BootStrap.groovy` pour l'exécution de code au démarrage de l'application

Afficher un message

- Créer un controller en ligne de commande

```
grails create-controller wrh.simple
```

- Afficher du texte depuis mon controller

```
package wrh

class SimpleController {

    def index() {
        render 'Hello World!'
    }
}
```

Tester mon code

- Dans le fichier de test unitaire

```
void "test index"() {  
  when:  
    controller.index()  
  
  then:  
    response.text == 'Hello World!'  
}
```



Unit Test Results - All tests

A single test executed without a single error or failure!

[Tests with failure and errors](#)

[Package summary](#)

[Show all tests](#)

SimpleControllerSpec

A single test executed without a single error or failure!

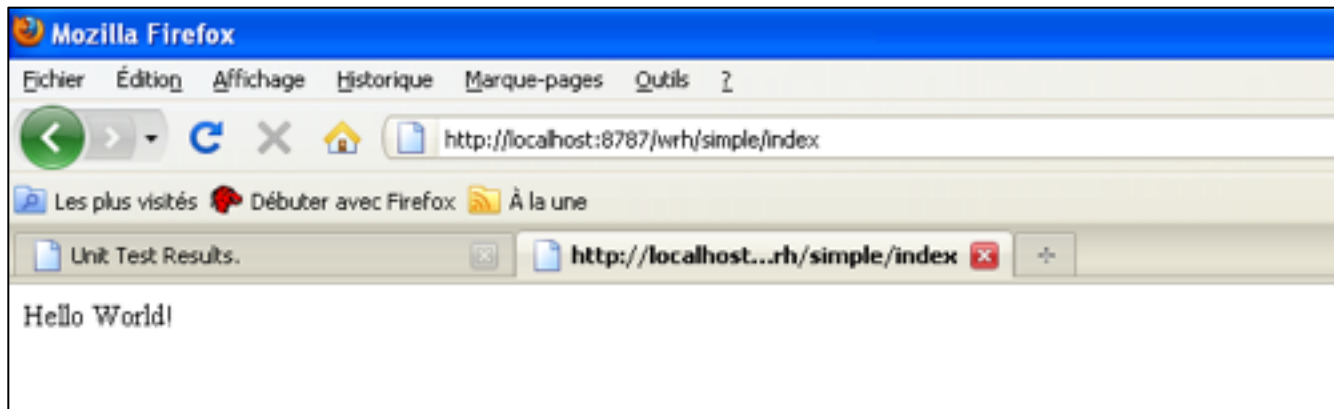


test index

Executed in 3.413 seconds.

Lancer l'application

```
grails run-app
```



Scaffolding

Qu'est ce que le scaffolding?

- La génération du code du controller et des views à partir des models
- Scaffolding dynamique pour la génération au runtime
- Scaffolding statique pour la génération de fichiers statiques

Créer une domain class

```
grails create-domain-class wrh.Product
```

```
package wrh

class Product {

    String name
    Double weight

    static constraints = {
    }
}
```

Scaffolding dynamique

- Activer le scaffolding

```
class ProductController {  
  def scaffold = Product  
}
```

- Ajouter des méthodes

```
class ProductController {  
  def scaffold = Product  
  def hello() {  
    render 'Hello World!'  
  }  
}
```

Scaffolding dynamique

- Modifier des méthodes

```
class ProductController {  
  def scaffold = Product  
  def index() {  
    render 'Hello World!'  
  }  
  def list() {  
    render "$${productInstanceList: Product.list(),  
              productInstanceTotal: Product.count()]]}"  
  }  
}
```

Scaffolding statique

- Générer le Controller

```
rails generate-controller wrh.Product
```

- Générer les Views

```
rails generate-views wrh.Product
```

- Générer l'ensemble

```
rails generate-all wrh.Product
```

Questions?