Grails:

Desenvolvimento rápido na JVM

Apresentação

Paulo Henrique de Sousa
@paulosousalopes

- Bacharel em Sistemas de Informação
- Desenvolvedor Web (.NET) desde 2008
- Desenvolvedor Java e Grails desde 2009 no TRE-TO
- (...)

Introdução

- JVM – Grem Team 1992;
- Chuva de Frameworks;
- Ruby on Rails (RoR): um banho de água fria…
- JRuby;
- Grails Framework.

Grails?

Grails é um Framework 'Full Stack' de desenvolvimento web que oferece as ferramentas, técnicas e tecnologias presentes em outros frameworks Java, combinando-os com o poder e inovação de uma linguagem de desenvolvimento dinâmica, e as vantagens do CoC (Convention over Configuration);

Grails...

- G2One?
- Springsource?
- VMWare?
- Spring Framework -> Java | Grails -> Groovy Groovy?

Groovy?

Agile and dynamic language for JVM
Sintaxe like Java

Instalação:
gvm install groovy

In the beginning...

...it was cold, dark, an empty space...
...mas no dia 29 de agosto de 2003...
...James Strachan publicou em seu blog:
...“minha idéia inicial é fazer uma pequena linguagem dinâmica, que seja compilada diretamente em classes Java e que tenha toda a produtividade elegante encontrada em Ruby e Python, mas que permita reusar, estender, implementar e testar código Java já existente”;

Groovy

- É uma linguagem OO para a plataforma Java;
- É dinâmica, como Python, Ruby, Perl, e outras;
- É compilada para bytecodes pela JVM;
- Groovy é uma linguagem de script (também);
- Códigos em Java são sintaticamente aceitos;
- Possui recursos como tipagem dinâmica e closures;

Groovy

```
// Exemplo em Java
public class Aluno {
    Integer id;
    String nome;
    public String getNome() { return this.nome; }
    public void setNome(String nome) {this.nome = nome;}
    public Integer getId() { return this.id; }
    public void setId(Integer id) { this.id = id; }
 }
```

Groovy

```
// Exemplo em Groovy
class Aluno { String nome }
```

Groovy

```
//Lendo arquivo em Java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileNotFoundException;
import java.io.IOException;
public class FileTest {
     public static void main(String[] args) {
     try {
          String arquivo = "teste.txt";
          String linha = "";
          StringBuilder sb = new StringBuilder();
          BufferedReader br = new BufferedReader(new FileReader(arquivo));
          while ((linha = br.readLine()) !=null) {sb.append(line);}
          System.out.println("sb.toString() = " + sb.toString()); }
          catch (FileNotFoundException e) {
               System.out.println("e = " + e);
          }
          catch (IOException e) {
               System.out.println("e = " + e);
          }
     }
}
```

Groovy

```
// Lendo arquivo em Groovy

println new File(“teste.txt”).text
```

Groovy - Variáveis

```
x = 1
println x

x = new java.util.Date()
println x

x = -3.1499392
println x

x = false
println x

x = "Hi"
println x
```

Groovy - Lists and Maps

```
myList = [1776, -1, 33, 99, 0, 9287349287 63]

scores = ["Paulo":91, "Marie":"Não tem pontuação", "Petrus":86.84]
```

Groovy - Lopping

```
def x = 0
for ( i in 0..9 ) {
    x += i
}

// iterate over a list
x = 0
for ( i in [0, 1, 2, 3, 4] ) {
    x += i
}

def map = ['abc':1, 'def':2, 'xyz':3]
x = 0
for ( e in map ) {
    x += e.value
}
```

Closures

```
class ClasseBoba {
// bla bla bla pra cima
    def closure = {
        println "Sou uma closure que recebeu como parâmetro o
        valor de ${it}"
    }
// bla bla bla pra baixo
}
```

Closures

Free variables;
Implicit variables;

```
def stringMap = [ "Su" : "Sunday", "Mo" : "Monday", "Tu" : "Tuesday",
                  "We" : "Wednesday", "Th" : "Thursday", "Fr" : "Friday",
                   "Sa" : "Saturday" ];
stringList.each() { print " ${it}" }; println "";
stringMap.each() { key, value -> println "${key} == ${value}" };
```

Closures como argumento

```
def list = ['a','b','c','d']
def newList = []
list.collect( newList ) {
    it.toUpperCase()
}
println newList
```

Iniciando no Grails

```
grails create-app helloworld
cd helloworld
create-controller hello

class HelloController {
    def index() {
        render "Hello World!"
    }
}
```

Tecnologias disponíveis no Grails

- Model: GORM – Grails Object Relational Mapping (Hibernate);
- View: GSP's – Groovy Server Pages Controller: Spring Framework;
- Gant;
- Tomcat Embedded;
- (entre outras).

Estrutura de Diretórios

grails-app - top level directory for Groovy sources
conf - Configuration sources.
controllers - Web controllers - The C in MVC.
domain - The application domain.
i18n - Support for internationalization (i18n).
services - The service layer.
taglib - Tag libraries.
utils - Grails specific utilities.
views - Groovy Server Pages - The V in MVC.
scripts - Gant scripts.
src - Supporting sources
groovy - Other Groovy sources
java - Other Java sources
Test - Unit and integration tests.
web-app – Static files

Comandos do Grails

```
grails help
grails clean
grails compile
grails run-app
grails upgrade
grails create-app
grails create-controller
grails create-domain-class
grails create-filters
grails install-plugin
grails generate-all
(...)
```

Controllers

- Controlam o fluxo da aplicação;
- Possuem actions (ações);
- A action geralmente retorna o processamento para uma GSP de mesmo nome;

GSP's – Groovy Server Pages

- Representam a camada de visualizaçãoo (view);
- São uma evolução das JSP’s;
- Simplificam dificuldades encontradas nas JSP’s, como a utilização de tag libraries;
- Permitea criação de tag libraries personalizadas;

Scaffolding

É um método de meta-programação que permite ao compilador criar o código para o CRUD da aplicacão para uso em runtime;

```
class EncoinfoController {
    static scaffold = true
}
// As actions abaixo serão geradas automaticamente.
// index, show, edit, delete, create, save, update
```

Mapeamento de URLs

```
class UrlMappings {
    static mappings = {
       "/$controller/$action?/$id?(.${format})?"{
            constraints {
               // apply constraints here
            }
        }
        "/"(view:"/index")
        "500"(view:'/error')
    }
}
```

Dynamic Finders

São métodos invocáveis, que podem não existir em Code Level;
Criados em runtime, 'magicamente', pelo GORM;
findBy*
findAllBy*
findAllWhere*

DataSource.groovy

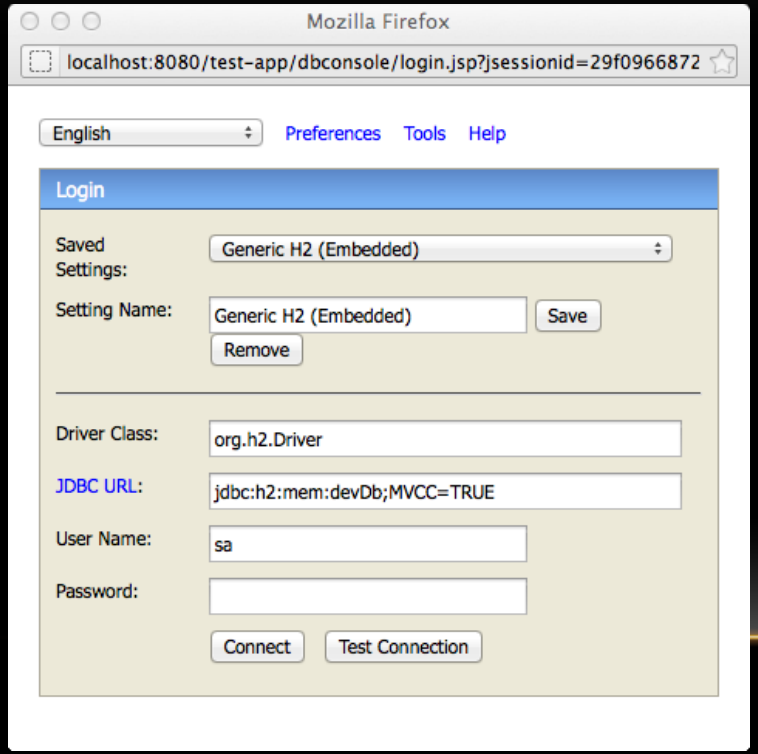
- grails-app/conf/DataSource.groovy
- grails-app/conf/DataSource.groovy Environments:
  - Create-drop: apaga e (re)cria o banco. Deleta os dados existentes;
  - Create: cria o banco se não existir, mas não o modifica caso já exista. Deleta os dados existentes;
  - Update: Cria o banco se não existir, e modifica-o caso exista; banco;
- grails <env> run-ap

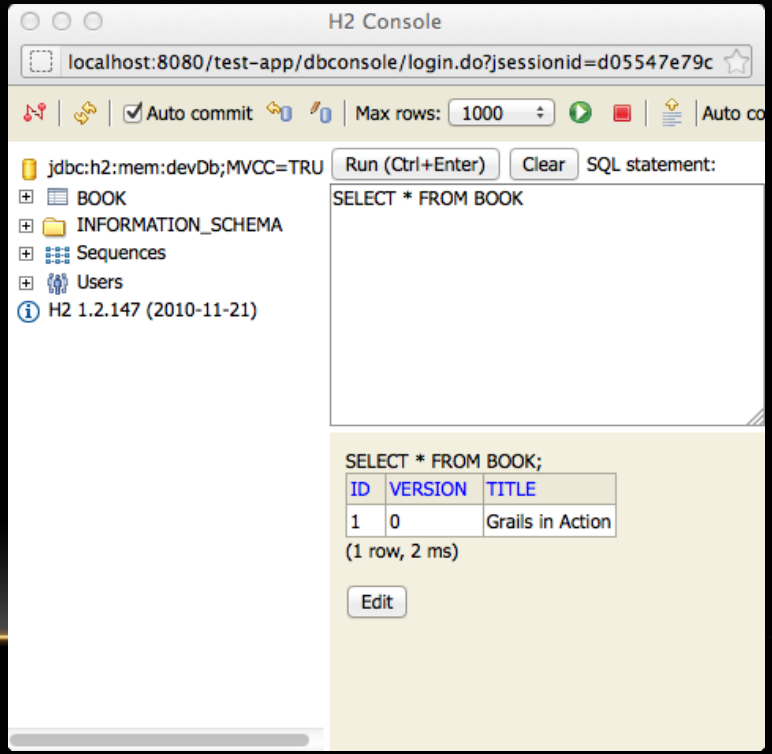
HSQLDB

- Hyperthread SQL Database;
- Serve para criar DB's em memória e em arquivos;
- Facilidade de uso, de backup, etc;

H2 Database ConsoleH2

YourApp:8080/dbconsole





Deploy

Grails war - Grails [env] war

Containers Disponíveis

- Container
- Tomcat
- GlassFish
- Websphere
- JBoss
- Jetty (Padrão)
- Weblogic
- ...