



Groovy And Grails

Harshad Oak
Rightrix Solutions



Groovy Basics

- Scripting / Agile / Dynamic ...Language
- Syntax very close to Java
- Meant for Java developers.
- A powerful high level language for the Java “platform”
- Groovy code Compiles to Java bytecode.
- You can get productive quickly



Groovy Basics

- Open source using a BSD / Apache style licence
- Backed by JSR 241
- groovy.codehaus.org

- Harshad Oak



Groovy Installation

- Download binary
- Set GROOVY_HOME
- Add GROOVY_HOME/bin to your PATH
- Set JAVA_HOME
- Try Groovy and GroovyConsole



Groovy With Java

- Use groovyc instead of javac to create bytecode files.
- Run with java
- To embed Groovy you can use the GroovyClassLoader or GroovyScriptEngine
- Groovy integrates with the Bean Scripting Framework, which allows you to embed any scripting engine into your Java code



Groovy Development

- IDE Support
 - Eclipse plugin
 - NetBeans using Coyote (Groovy + Jython)
 - Textpad
 - UltraEdit
 - Crimson Editor
 - IntelliJ IDEA ...

- Harshad Oak



Groovy Development

- Language Constructs
 - List & Maps support
 - XML Capabilities
 - Swing Capabilities
 - Testing
 - Groovy SQL
-
- The power of Java libraries stays with you. It's just easier and faster with Groovy

- Harshad Oak



Closures

- Closures are reusable blocks of code
- One or more program statements enclosed in curly brackets
- Closures do not require a class or a method name



Closures

- Can be passed around like a variable
- The statements within a closure are not executed until the call() is made
- Return is using an explicit return statement or the value of the last executed statement



Strings

- Strings work almost like those in Java
- Expect for Gstring.
 - GStrings can contain expressions in the form `${expression}`



Dynamic Methods

- Adding new methods to an object at runtime
- Intercept non-existent method calls
- Used extensively in Grails
- Meta-Object Protocol

- Harshad Oak



Differences from Java

- == means equals
- Semicolon & return are optional
- Methods and classes are public by default



Groovlets and GSP

- Groovy alternatives for Servlets and JSP
- Will work on any J2EE container.
- Add JAR files
- Tweak web.xml

- Harshad Oak



Groovlets and GSP

- Implicit Variables
 - request ServletRequest
 - response ServletResponse -
 - context ServletContext
 - application ServletContext
 - session getSession(false)
 - out response.getWriter()
 - sout response.getOutputStream()
 - html new MarkupBuilder(out)



Some More Groovy

- Readymade modules for:
 - XML-RPC
 - SOAP
 - Xdoclet like tag processing
 - Working with Google APIs
 - ...



Grails

- Web Framework inspired by Ruby On Rails
- Latest version is 0.3
- Coding by convention paradigm
- Reduce the need for configuration files and other boilerplate code
- File names are important



Grails

- Provides several commands to auto generate common requirements in a web app.
- Based on open source technologies like Spring, Hibernate and SiteMesh
- Grails hides underlying complexity
- Focus is on rapid development and simplicity



Grails Installation

- Extract contents
- Set `GRAILS_HOME` environment variable
- Add `GRAILS_HOME\bin` to the `PATH` variable
- Run the *grails* command from the command prompt to check usage

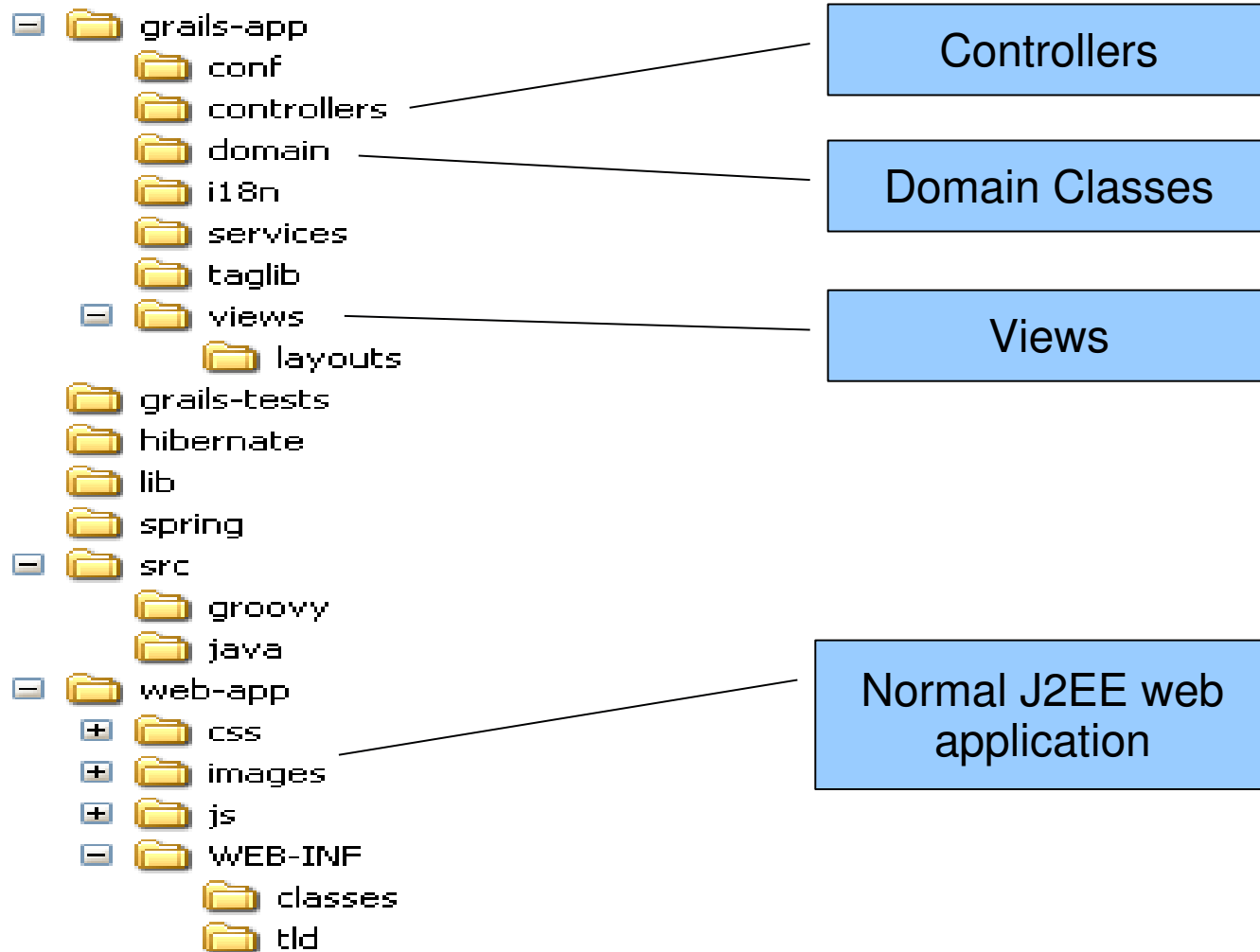


Grails Installation

- Extract contents
- Set `GRAILS_HOME` environment variable
- Add `GRAILS_HOME\bin` to the `PATH` variable
- Run the *grails* command from the command prompt to check usage



Grails Installation



- Harshad Oak



GORM

- Domain Classes are objects that are mapped to the database
- Uses Hibernate 3 underneath
- Link domain classes using relationships
- Provide powerful dynamic methods for CRUD (Create/Read/Update/Delete) operations
- An HSQL database comes built-in with Grails

- Harshad Oak



Grails Development

- Example -Develop a Clothes Manager web application
 - Change database
 - Setup DB
 - Setup ApplicationDataSource
 - Create Domain Classes



Grails Development - Domain Classes

- Create Domain Classes
 - Cabinet.groovy
 - Shirt.groovy
 - Trouser.groovy
- Define relationships



Grails Development -Controllers

- Controllers are responsible for handling the requests in the web application
- Use `grails -generate-all` to generate controllers for our 3 domain classes
- Once controller is done processing a request, it has to delegate to an appropriate view



Grails Development -Views

- Controllers delegate to view based on a convention mechanism
- So list closure in ShirtController will delegate to the view `/grails-app/views/shirt/list.gsp` or `/grails-app/views/shirt/list.jsp`.
- Note that you can use JSPs



Grails Development -Deploy

- Built in Resin server
- Directly run your application by using the grails run-app command
- Lets use Tomcat instead
 - WAR file creation using grails war
 - Deploy the WAR file just like any other J2EE WAR file



Grails Services

- Services are named in the format `XXService.groovy`
- Directory `/grails-app/services`
- Can have Business Logic
- You can call these services from within controllers



Grails Tags

- Range of tags for both GSPs and JSPs
- No configuration
- No updating of TLD descriptors
- Edit `grails-app\taglib\ApplicationTagLib.groovy` or create a new file `XXTagLib.groovy`



Grails Ajax

- Can work with Prototype, Yahoo or Dojo
- Configure the library and then just use the grails tags
- Use render method to send text, markup, JSON or OpenRico responses



Thank You

- Further Reading
 - groovy.codehaus.org
 - grails.org
- Several books in the works
 - Groovy in Action (Manning)
 - Groovy Programming: An Introduction for Java Developers (Morgan Kaufmann)

- Harshad Oak