

Judd Solutions

Development, Mentoring and Training



Agile Development with Groovy and Grails



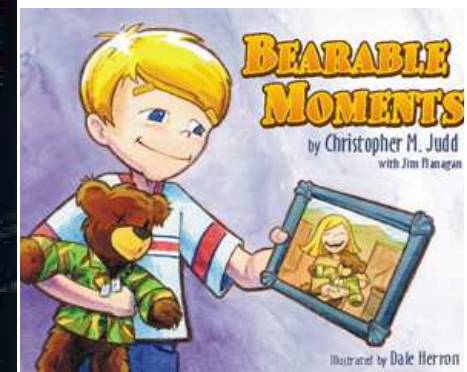
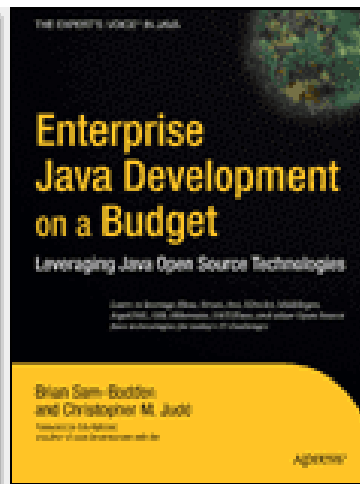
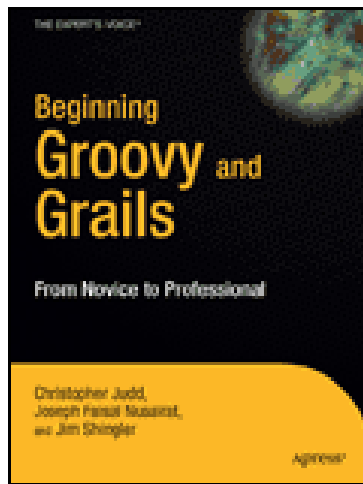
Christopher M. Judd
President/Consultant
Judd Solutions, LLC



Christopher M. Judd



- President/Consultant of Judd Solutions
- Central Ohio Java User Group (COJUG) coordinator
- Creator Open Source projects FallME and Fiddle
- Creator of the Chmod iPhone App

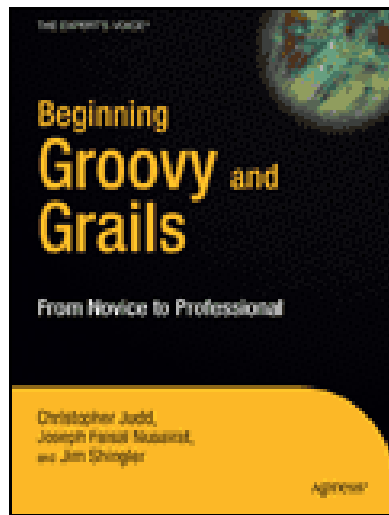




James L. Shingler



- Sr. Consulting Architect for Nationwide
- Principal Consultant for Genuine Solutions
- Co-Creator of Open Source Project FallME
- Co-Author of Beginning Groovy and Grails



Judd Solutions



Agenda



- Java Web Development Landscape
- Groovy and Grails Overview
- Demo



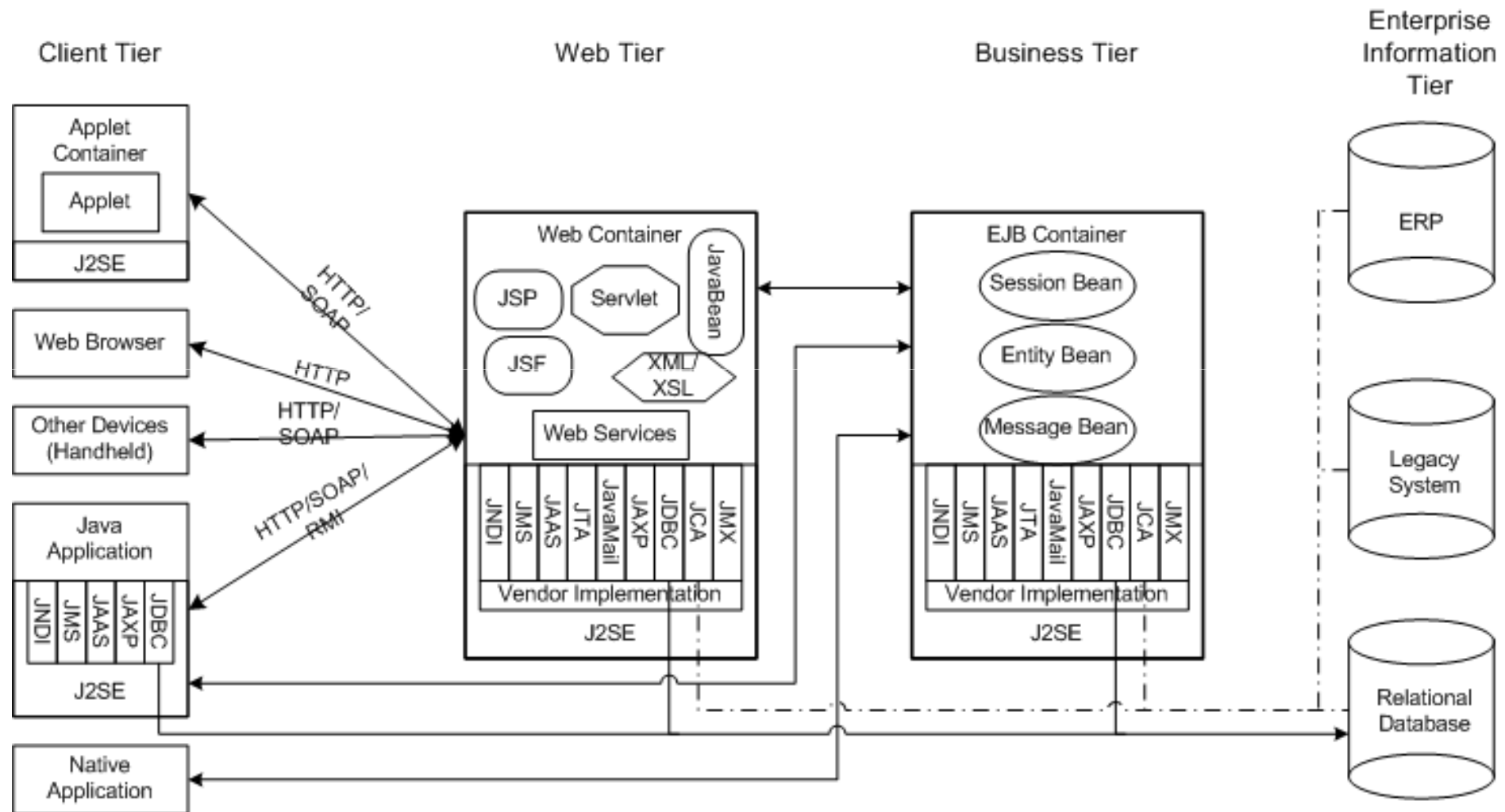
Starting Java Web Projects



- What do you need to start a Java Web Project?
- JDK
- Web Container/App Server  GlassFish Tomcat jboss  WebSphere WebLogic
- Build System  maven
- Application Stack Spring Java EE
- Persistence Framework HIBERNATE JPA JDBC iBATIS
- Web Framework SiteMesh JSF Struts tapestry Spring Web MVC Spring Web Flow
- Unit Test Framework JUnit TestNG EASYMOCK jMock
- AJAX Framework prototype script.aculo.us dojo Y! YUI GWT
- Database HSQL database engine Apache Derby DB2 ORACLE MySQL Microsoft SQL Server
- Scheduling Framework .
- XML Framework
- Logging Framework
- View
- IDE .

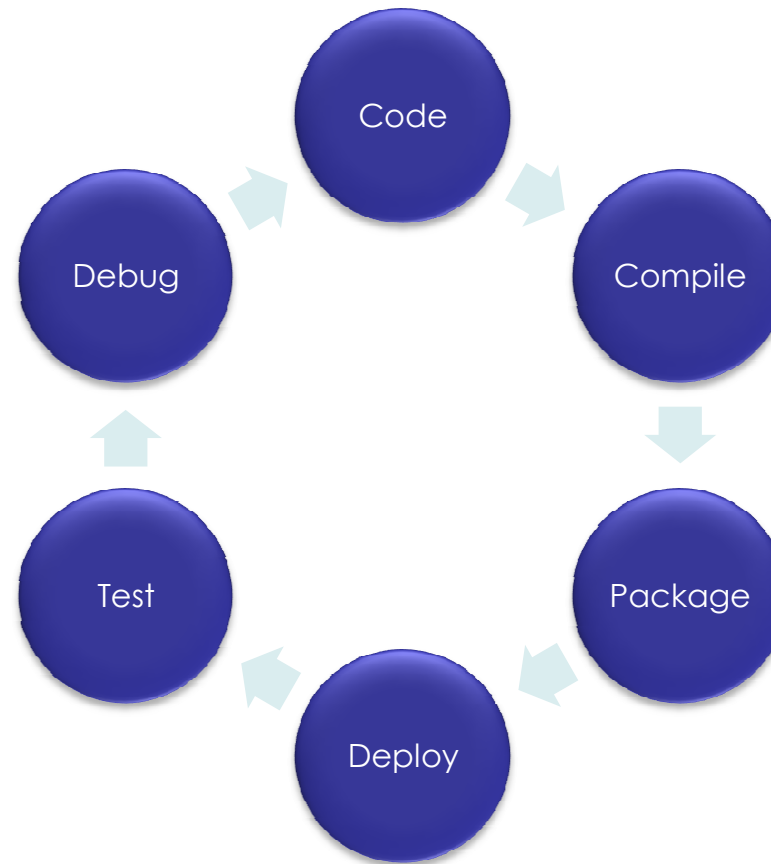


Java EE/J2EE Architecture





Java Development Cycle



Takes too much time and is too complicated!



Conclusion



Java Web and J2EE
development is too
complicated and
not very agile!!!

But we can't or don't want to abandon Java completely.



Solution



&



GRAILS



GRAILS

Judd Solutions



Agenda



- Java Web Development Landscape
- Groovy and Grails Overview
- Demo



Groovy



- Dynamic scripting language for Java VM
- Open Source under Apache 2.0 License
- Current version 1.6
- JSR 241
- Better Java (Java 3.0)
- Advanced features
 - Properties
 - Closures
 - Optional typing
 - Builders
 - Cool operators
 - Operator overloading
 - Meta-programming
 - Groovy JDK (methods to Java library)

groovy = []
groovy << **java**
groovy << **ruby**
groovy << **python**
groovy << **scripting**
groovy << **agile**
groovy << **dynamic**



Language Stack



Domain Specific Language

Domain

Groovy

Application (Dynamic)

Java Language

System (Static)



Grails



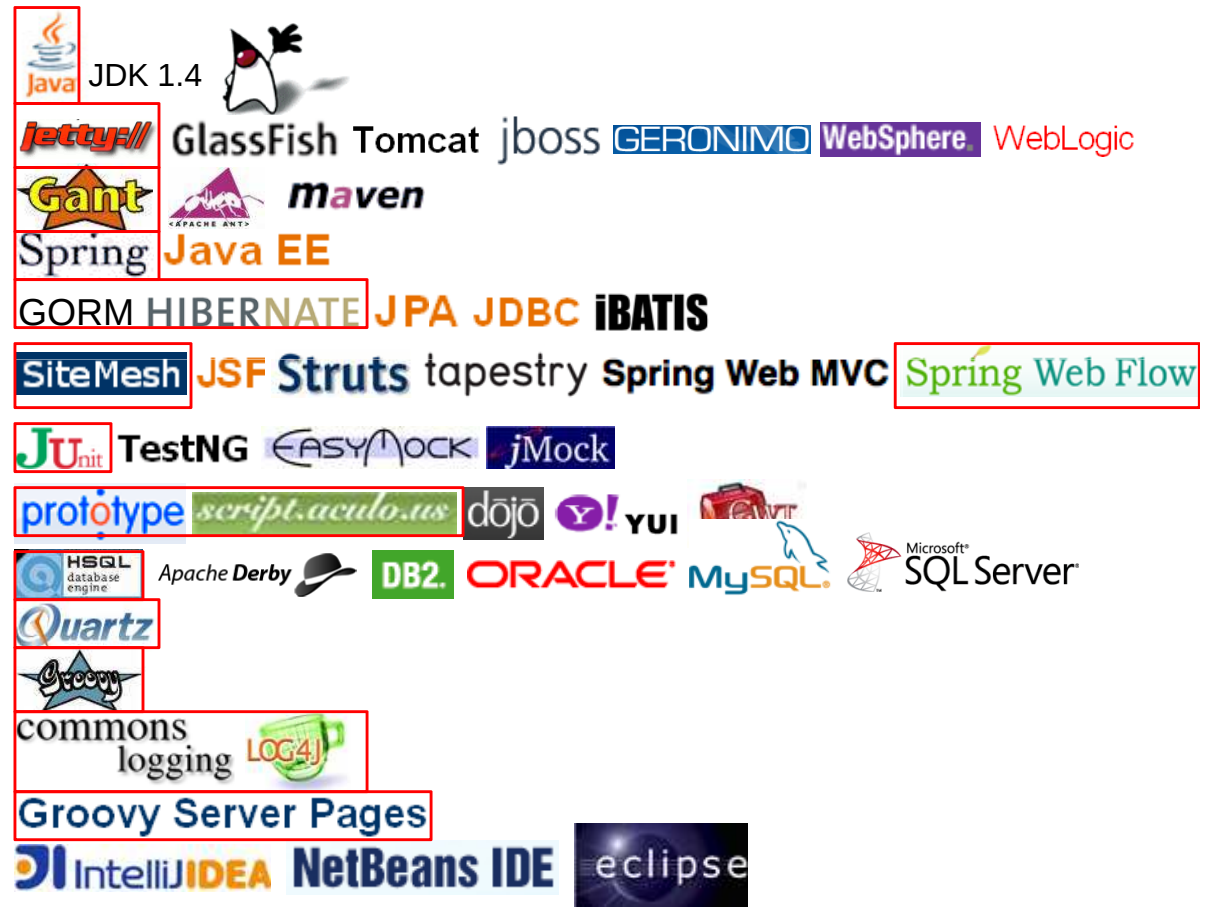
- Web Application Framework
- Development Environment
- Open Source under Apache 2.0 License
- Current version 1.1
- Best Practices
 - Convention over configuration
 - Don't repeat yourself (DRY)
 - Default is what you expect (DIWYE)
 - Agile
 - Scaffolding
 - AJAX
 - Plug-ins
 - Unit testing
 - Web Services



Grails Frameworks

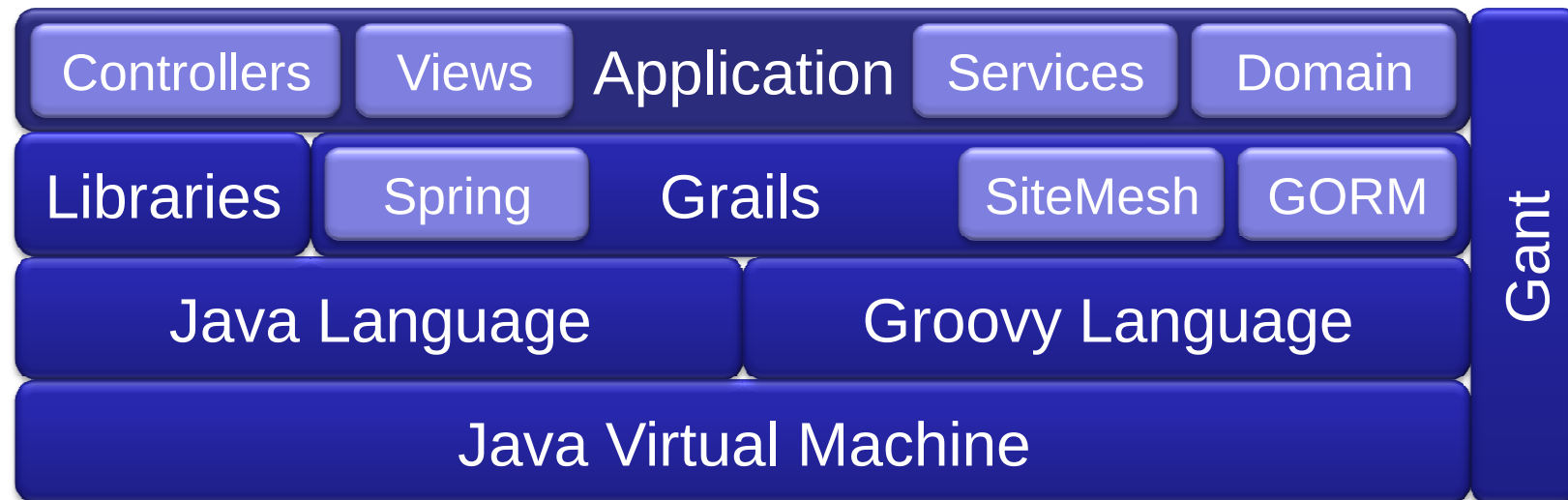


- JDK
- Web Container/App Server
- Build System
- Application Stack
- Persistence Framework
- Web Framework
- Unit Test Framework
- AJAX Framework
- Database
- Scheduling Framework
- XML Framework
- Logging Framework
- View
- IDE



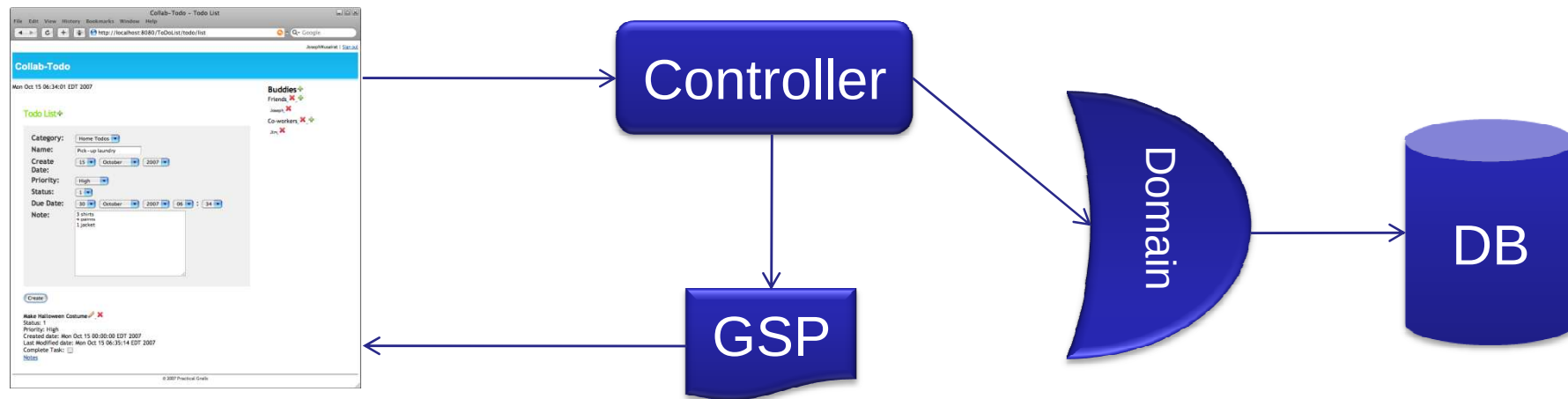


Grails Architecture





MVC Application Design





DEMO



[Home](#) [New Todo](#)

Todo List

Id	Name	Create Date	Priority	Status	Note
1	Buy Beginning Groovy and Grails	2008-07-29 03:30:00.0	high	complete	
2	Todo 1	2008-07-29 03:30:00.0	high	complete	
3	Todo 2	2008-07-29 03:30:00.0	high	complete	

3	Todo 3	2008-07-29 03:30:00.0	high	complete	
5	Todo 5	2008-07-29 03:30:00.0	high	complete	
1	Buy Beginning Groovy and Grails	2008-07-29 03:30:00.0	high	complete	



Post Search

Results:

Ajax Search

Results:

Incremental Ajax Search

Todo 1

Todo 2

Todo 5

Todo 1

bt

Incremental Ajax Search



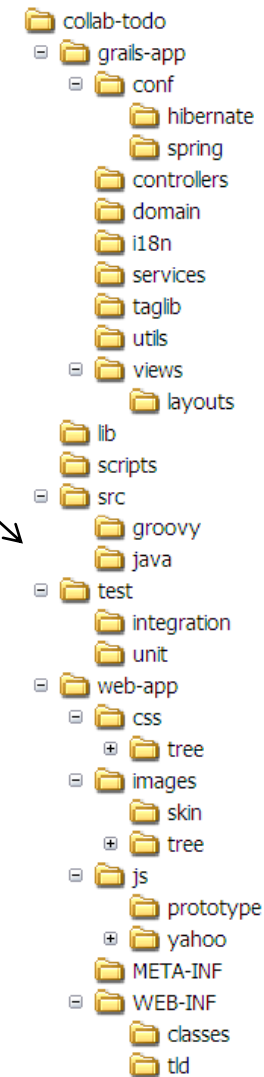
Creating Grails Project



```
grails create-app collab-todo
```



Eclipse Project



Project Structure



Running application



```
grails run-app
```





Creating Domain Class



```
grails create-domain-class todo
```

Domain Class

```
class Todo {  
    static constraints = {  
    }  
}
```

Integration Test

```
import grails.test.*  
  
class TodoTests extends GrailsUnitTestCase {  
    protected void setUp() {  
        super.setUp()  
    }  
  
    protected void tearDown() {  
        super.tearDown()  
    }  
  
    void testSomething() {  
    }  
}
```



Defining Domain Properties



```
class Todo {  
    String name  
    String note  
    Date dateCreated  
    Date dueDate  
    Date completedDate  
    String priority  
    String status  
  
    static constraints = {  
    }  
}
```

Java class
with
getters and setters



Just Another Java Class



groovyc Todo.groovy

javap Todo

Compiled from "Todo.groovy"

```
public class Todo extends java.lang.Object implements groovy.lang.GroovyObject {
    public static final java.lang.Class $ownClass;
    public static java.lang.ref.SoftReference $staticMetaClass;
    transient groovy.lang.MetaClass metaClass;
    public static java.lang.Long __timeStamp;
    public static java.lang.Long __timeStamp__239_neverHappen1219721540048;
    public Todo();
    protected groovy.lang.MetaClass $getStaticMetaClass();
    public groovy.lang.MetaClass getMetaClass();
    public java.lang.Object invokeMethod(java.lang.String, java.lang.Object);
    public java.lang.Object getProperty(java.lang.String);
    public void setProperty(java.lang.String, java.lang.Object);
    static {};
    public java.lang.String getName();
    public void setName(java.lang.String);
    public java.lang.String getNote();
    public void setNote(java.lang.String);
    public java.util.Date getDateCreated();
    public void setDateCreated(java.util.Date);
    public java.util.Date getDueDate();
    public void setDueDate(java.util.Date);
    public java.util.Date getCompletedDate();
    public void setCompletedDate(java.util.Date);
    public java.lang.String getPriority();
    public void setPriority(java.lang.String);
    public java.lang.String getStatus();
    public void setStatus(java.lang.String);
    public void setMetaClass(groovy.lang.MetaClass);
    void super$1$wait();
    java.lang.String super$1$toString();
    void super$1$wait(long);
    void super$1$wait(long, int);
    void super$1$notify();
    void super$1$notifyAll();
    java.lang.Class super$1$getClass();
    boolean super$1$equals(java.lang.Object);
    java.lang.Object super$1$clone();
    int super$1$hashCode();
    void super$1$finalize();
    static java.lang.Class class$(java.lang.String);
}
```



Creating Controller



```
grails create-controller todo
```

Controller

```
class TodoController {  
    def index = { }  
}
```

Integration Test

```
class TodoControllerTests extends  
    ControllerUnitTestCase {  
  
    protected void setUp() {  
        super.setUp()  
    }  
  
    protected void tearDown() {  
        super.tearDown()  
    }  
  
    void testSomething() {  
  
    }  
}
```



Enabling Scaffolding



```
class TodoController {  
  def scaffold = true  
}
```

Generates CRUD Application



Home New Todo

Todo List

Id	Completed Date	Date Created	Due Date	Name	Note
----	----------------	--------------	----------	------	------



Welcome to Grails

Congratulations, you have successfully started your first Grails application! At the moment this is the default page, feel free to modify it to either redirect to a controller or display whatever content you may choose. Below is a list of controllers that are currently deployed in this application, click on each to execute its default action:

- [TodoController](#)



Home Todo List

Create Todo

Completed Date: 26 August 2008 04 : 57

Date Created: 26 August 2008 04 : 57

Due Date: 26 August 2008 04 : 57

Name:

Note:

Priority:

Status:

Create



Adding Constraints



```
class Todo {  
    String name  
    String note  
    Date dateCreated  
    Date dueDate  
    Date completedDate  
    String priority  
    String status  
  
    static constraints = {  
        name(blank:false)  
        priority(inList:["high", "normal"])  
        status(inList:["complete", "in process"])  
        note(maxSize:1000, nullable:true)  
        completedDate(nullable:true)  
        dueDate(nullable:true)  
    }  
}
```

Home Todo List

Create Todo

Property [name] of class [class Todo] cannot be blank

Name:

Priority:

Status:

Note:

Completed Date: :

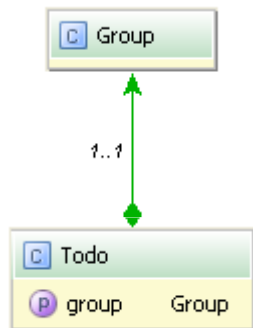
Due Date: :

Date Created: :

Create



Defining Relationships



```
class Todo {  
    String name  
    String note  
    Date dateCreated  
    Date dueDate  
    Date completedDate  
    String priority  
    String status  
}
```

```
static belongsTo = [Group]  
Group group
```

```
static constraints = {  
    name(blank:false)  
    priority(inList:["high", "normal", "low"])  
    status(inList:["complete", "in process"])  
    note(maxSize:1000, nullable:true)  
    completedDate(nullable:true)  
    dueDate(nullable:true)  
}
```

Home Todo List

Create Todo

Name:

Priority:

Status:

Note:

Completed Date: :

Due Date: :

Date Created: :

Group:

Create



Simple Search Action



TodoController search action

```
def search = {
  def todos = Todo.findAllByNameLike("${params.value}%")
  render(view: 'search', model: [value: params.value, todos: todos])
}
```

Annotations: GORM (points to `Todo.findAllByNameLike`), GString (points to `"${params.value}%"`)

search.gsp

```
<html>
<head>
  <meta name="layout" content="main" />
  <title>Todo Find</title>
</head>
<body>
  <h2>Post Search</h2>
  <g:form controller="todo" method="post" >
    <input type='text' name='value' value="${value}" />
    <g:actionSubmit value="Search" />
  </g:form>
  Results:
  <ol>
    <g:each var="todo" in="${todos}">
      <li>${todo?.name}
        <g:if test="${todo.note}">- </g:if> ${todo?.note}
      </li>
    </g:each>
  </ol>
</body>
</html>
```

Annotations: Layout (points to `content="main"`), Safe Navigation Operator (points to `todo?.name` and `todo?.note`)



Post Search

T

Results:

1. Todo 1
2. Todo 2



Ajax Search



TodoController Ajax search action

```
def remoteSearch = {
  def todos = Todo.findAllByNameLike("${params.value}%")
  render(template: 'results', model: [value: params.value, todos: todos])
}
```

_results.gsp

```
<ol>
  <g:each var="todo" in="${todos}">
    <li>${todo?.name}
      <g:if test="${todo.note}">- </g:if> ${todo?.note}
    </li>
  </g:each>
</ol>
```

search.gsp

```
<html>
  <head>
    <meta name="layout" content="main" />
    <title>Todo Find</title>
    <g:javascript library="scriptaculous" />
  </head>
  <body>
    <h2>Ajax Search</h2>
    <g:formRemote name="remoteSearch"
      url="[controller: 'todo', action: 'remoteSearch']"
      update="[success: 'results']" >
      <input type='text' name='value' value="${value}" />
      <g:submitButton name="search" value="Search" />
    </g:formRemote>
```

Results:

```
<div id="results"></div>
```

```
</body>
```

```
</html>
```

Spinner



Post Search

Results:

Ajax Search

Results:

1. Todo 1
2. Todo 2



Incremental Ajax Search



TodoController Incremental Ajax search action

```

def isearch = {
  def todos = Todo.findAllByNameLike("${params.value}%")
  def writer = new StringWriter()
  new groovy.xml.MarkupBuilder(writer).ul {
    todos.each {
      li(id: t.id, "${t.name}")
    }
  }
  render writer.toString()
}

```

Diagram annotations:

- Arrow from `li(id: t.id, "${t.name}")` to `Closure`
- Arrow from `new groovy.xml.MarkupBuilder(writer).ul {` to `Builder`

search.gsp

```

<html>
<head>
  <meta name="layout" content="main" />
  <title>Todo Find</title>
  <g:javascript library="scriptaculous" />
</head>
<body>
  <h2>Incremental Ajax Search</h2>
  <input type="text" id="autocomplete" name="autocomplete_parameter"/>
  <span id="indicator1" style="display: none"></span>
  <div id="autocomplete_choices" class="autocomplete"></div>
  <script type="text/javascript">
    new Ajax.Autocompleter("autocomplete", "autocomplete_choices", "isearch", {paramName: "value", indicator: 'indicator1'});
  </script>
</body>
</html>

```



Post Search

Results:

Ajax Search

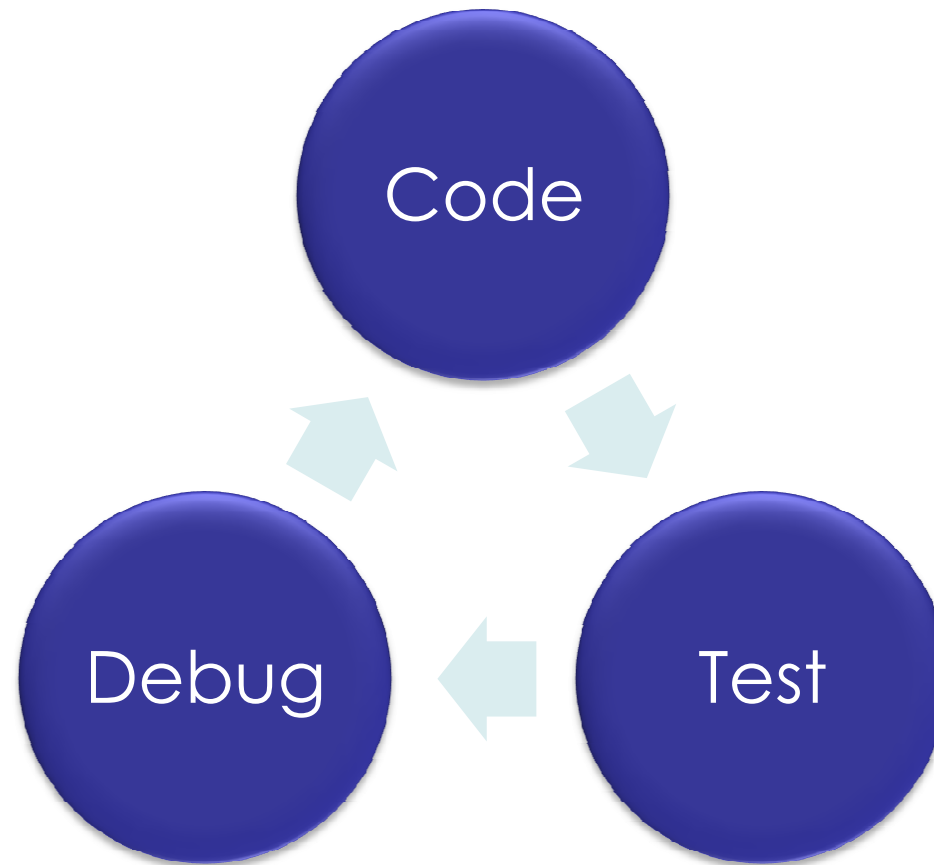
Results:

Incremental Ajax Search

To
Todo 1
Todo 2



Grails Development Cycle





URL Convention



- `http://<server>/<project>/<controller>/<action>/<id>`
- `http://localhost:8080/collab-todo/todo/list`
- `http://localhost:8080/collab-todo/todo/show/1`



Development Convention

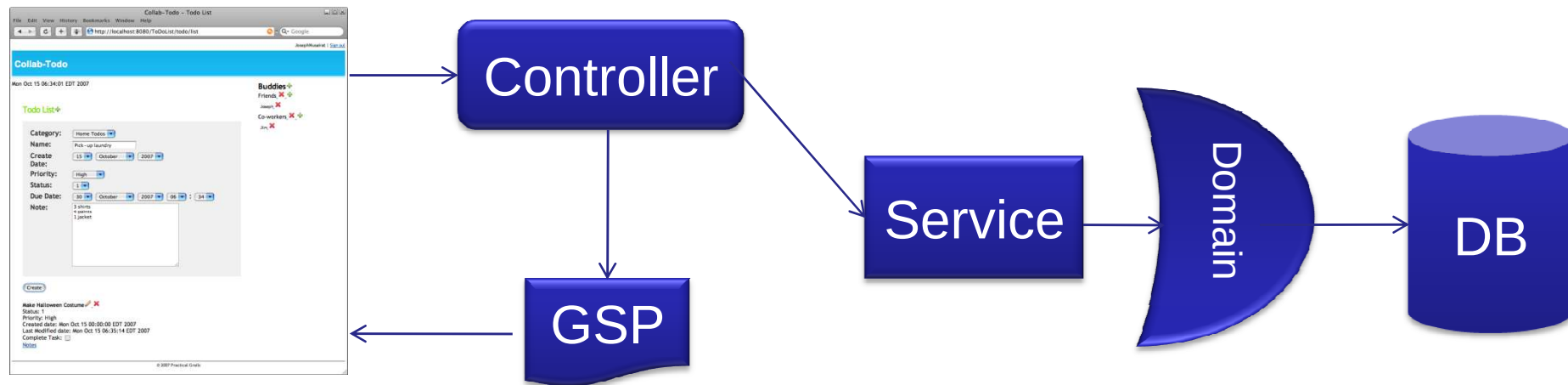


- Grails command-line tool

grails bootstrap	grails init
grails bug-report	grails install-plugin
grails clean	grails install-templates
grails compile	grails list-plugins
grails console	grails package
grails create-app	grails package-plugin
grails create-controller	grails plugin-info
grails create-domain-class	grails release-plugin
grails create-integration-test	grails run-app
grails create-plugin	grails run-app-https
grails create-script	grails run-war
grails create-service	grails set-proxy
grails create-tag-lib	grails set-version
grails create-unit-test	grails shell
grails doc	grails stats
grails generate-all	grails test-app
grails generate-controller	grails upgrade
grails generate-views	grails war
grails help	

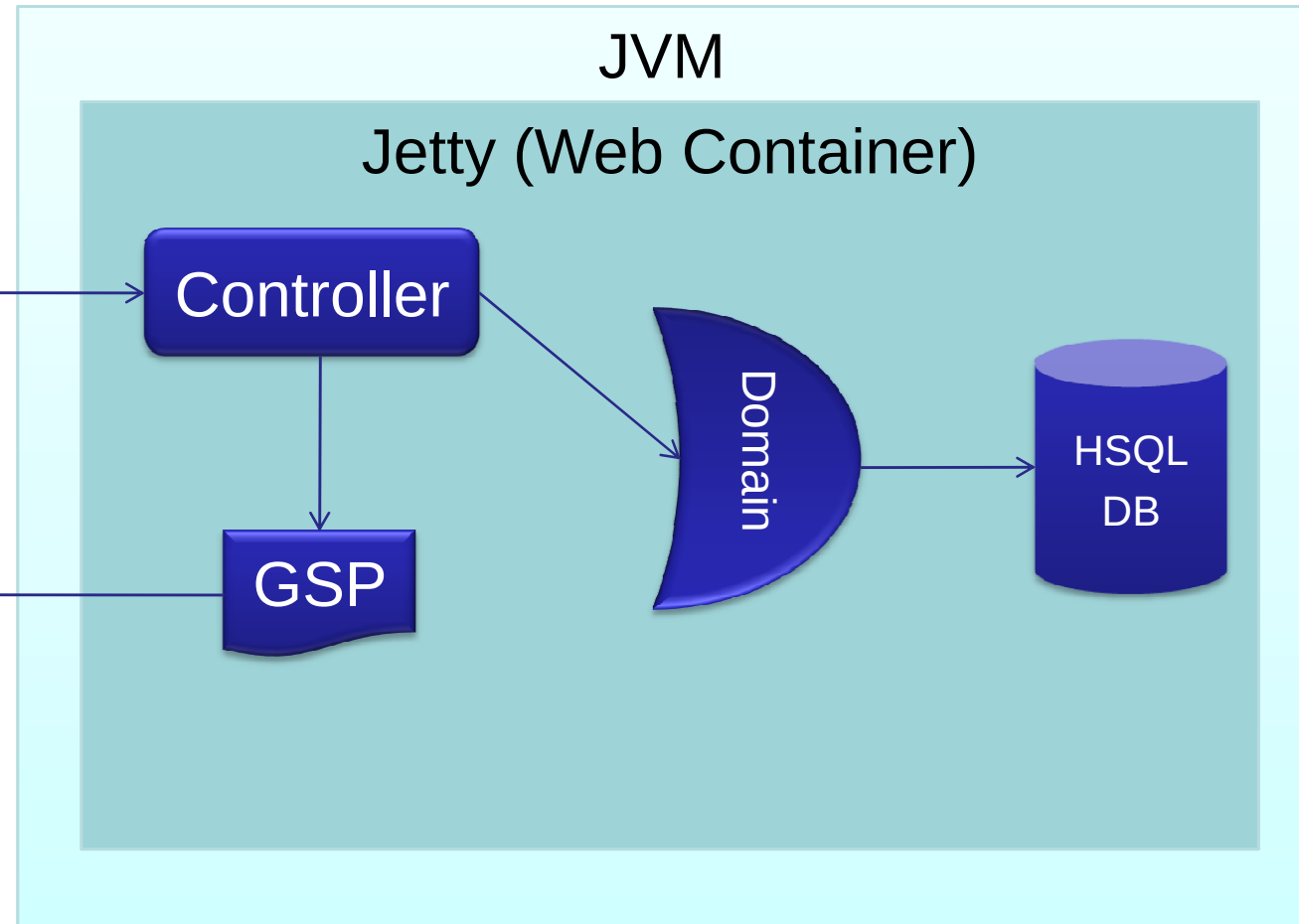
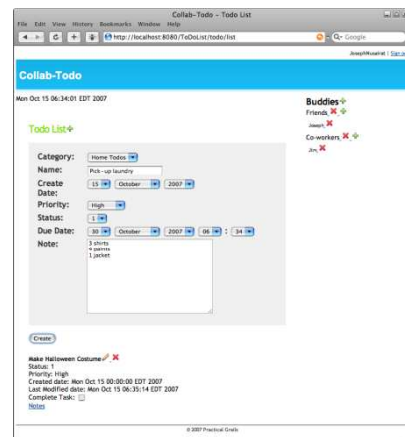


Service Application Design



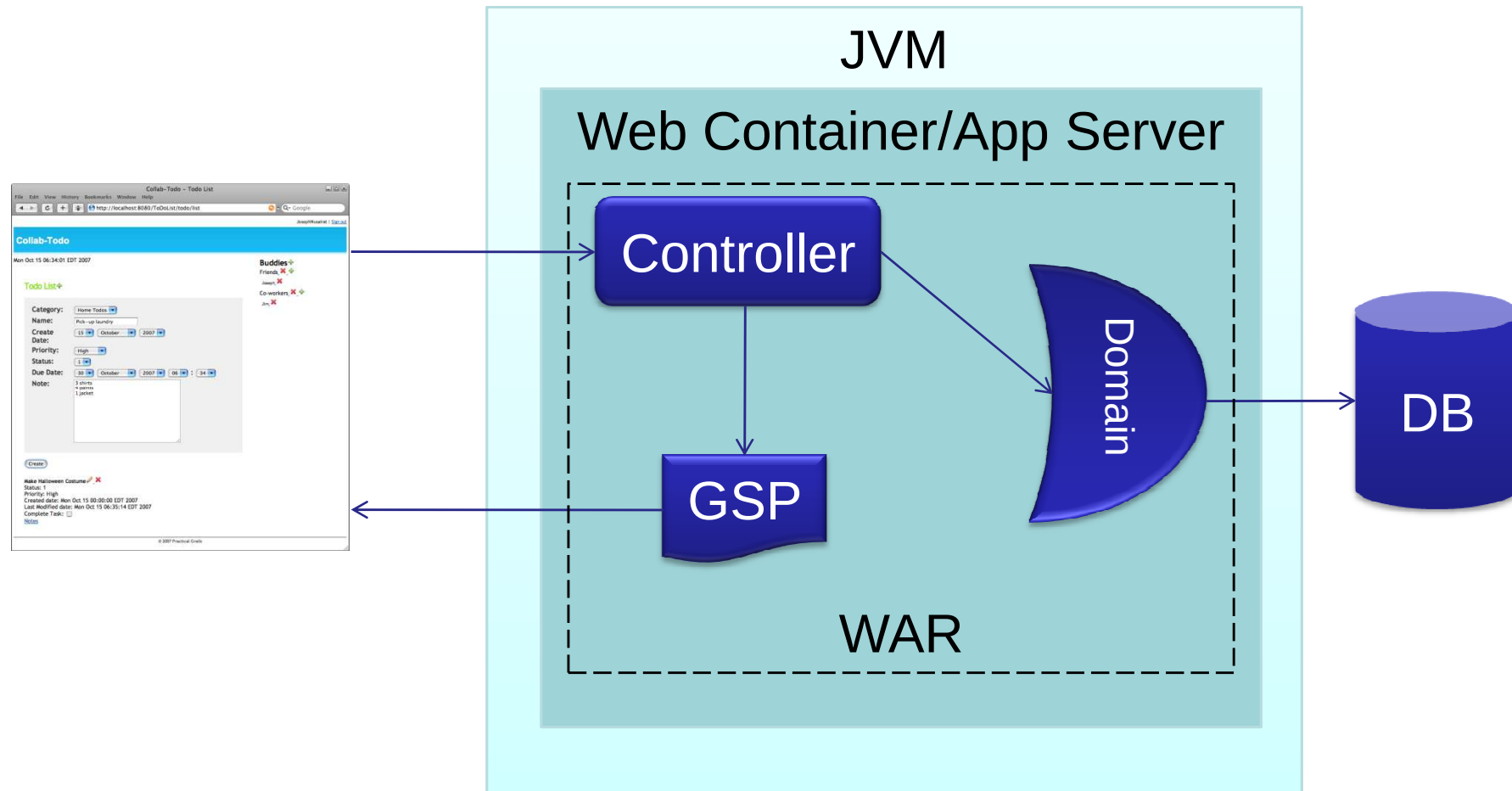


Default Development Environment





Production Environment





Grails vs. Ruby on Rails



- Design for the JVM
- Domain driven
- Flexible
- Supports multiple AJAX frameworks



- Runs on the JVM
- Database driven
- My way or highway
- Single AJAX framework



Conclusion



Java's capability, flexibility, backwards compatibility and community have lead to it's unprecedented success. However, these same things have become it's Achilles heal.

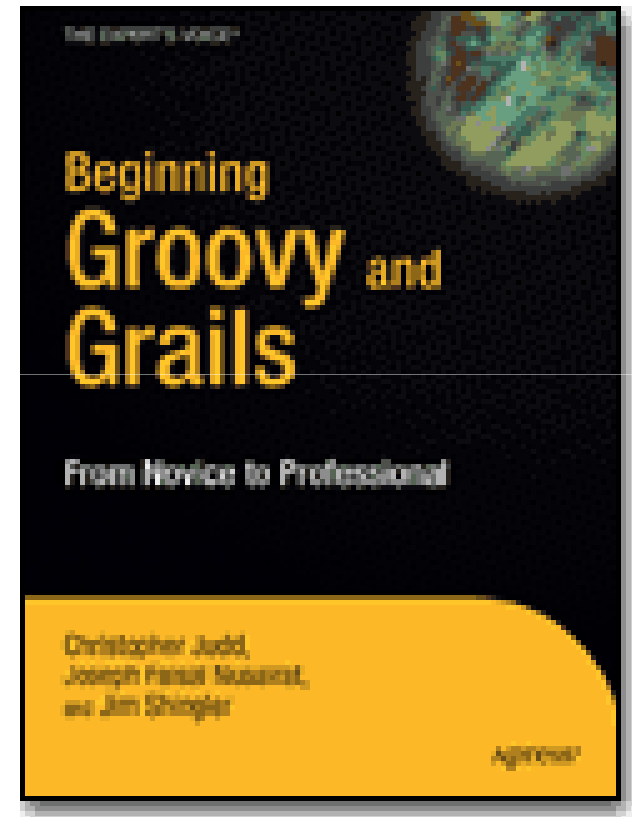
The combination of Groovy and Grails simplifies the Java web platform while making developers more productive and agile.



Beginning Groovy and Grails



1. Introduction to Groovy
2. Groovy Basics
3. More Advanced Groovy
4. Introduction to Grails
5. Building the User Interface
6. Building Domains and Services
7. Security in Grails
8. Web 2.0—Ajax and Friends
9. Web Services
10. Reporting
11. Batch Processing
12. Deploying and Upgrading
13. Alternative Clients

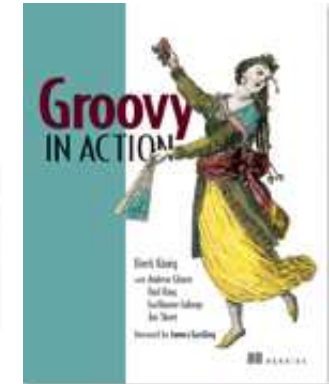
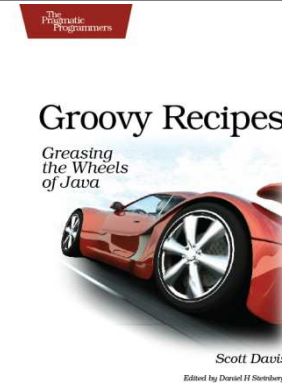




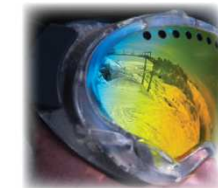
Resources



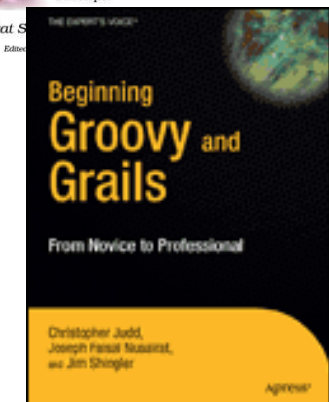
- Groovy
 - groovy.codehaus.org
 - Documentation
 - groovy.codehaus.org/Documentation
 - Groovy JDK (GDK)
 - groovy.codehaus.org/groovy-jdk/
 - Groovy Java Doc
 - groovy.codehaus.org/api/
- Groovy Zone - groovy.dzone.com
- About Groovy - www.aboutgroovy.com
- Groovy Blogs - www.groovyblogs.org
- Mastering Grails Series at IBM Developer Works
 - www.ibm.com/developerworks/java/library/j-grails01158/



Programming
Groovy



Venkat S.
Editor



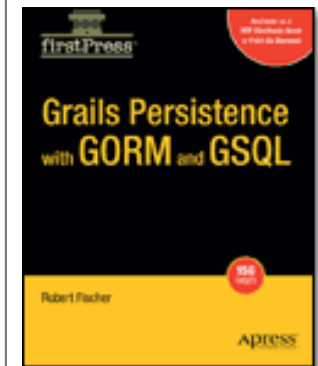
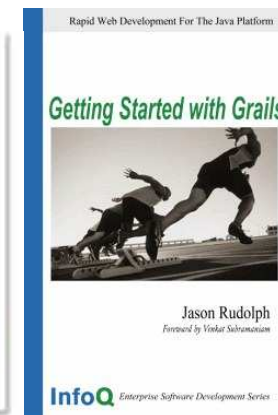
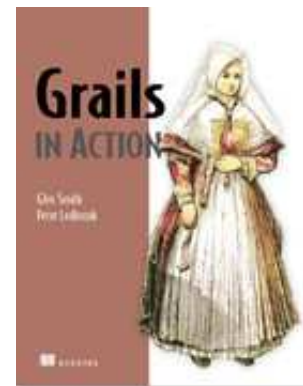
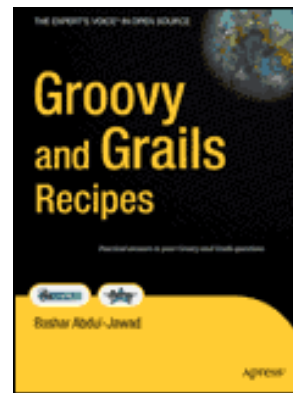
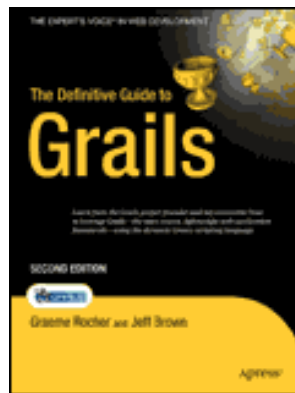
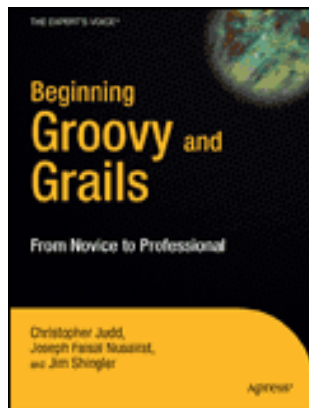
Judd SOLUTIONS



Resources



- Grails
 - grails.codehaus.org or www.grails.org
 - Grails Quick Start
 - grails.codehaus.org/Quick+Start
 - User Guide
 - grails.org/doc/1.1/
- Books

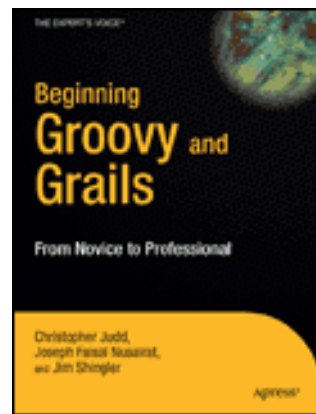




Contact Information



- Website: <http://www.juddsolutions.com>
- Email: cjudd@juddsolutions.com
- Blog: <http://juddsolutions.blogspot.com/>
- Beginning Groovy and Grails
 - <http://www.apress.com/book/view/1430210451>
- Pro Eclipse JST
 - <http://www.apress.com/book/bookDisplay.html?bID=447>
- Enterprise Java Development on a Budget
 - <http://www.apress.com/book/bookDisplay.html?bID=197>





Questions ?

Please fill out your evaluations.