Grails Basics

Sang Shin
JPassion.com
"Learn with Passion!"

Topics

- What is and Why Grails?
- Getting started
- Grails MVC framework
  - Controller, Domain, View
- Scaffolding
- IDE support

What is & Why Grails?

What is Grails?

- MVC action-based framework
  - Leveraging Groovy language – closure, meta-programming, etc
  - Runs over JVM
- Follows the good design principles popularized by Rails
  - Convention over configuration
  - Don't Repeat Yourself (DRY)
- But with tight integration with the Java platform
  - To protect your investment in Java
- Takes advantage of existing and proven Java technologies
  - Spring+Hibernate+SiteMesh+H2

Why Grails?

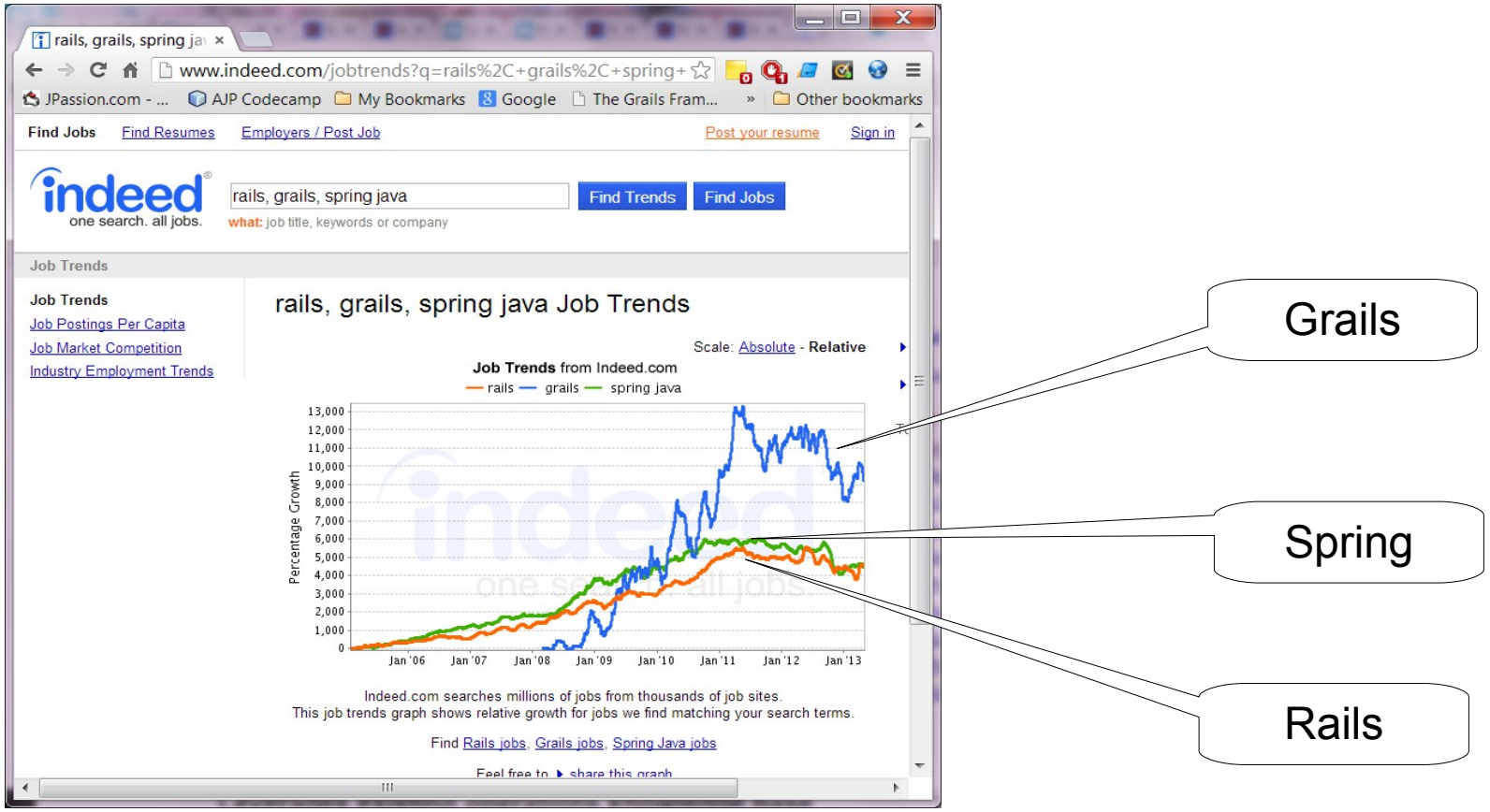
- Productive out of the box with full development stack
  - No configuration
  - No database to configure
  - No servlet container and no app server to install
  - Change and refresh development cycle (no redeploy required)
- Protect your investment in Java
  - Can take advantage of all existing Java libraries
  - Run it over any servlet container and app server
- Viable eco-system
  - Huge collection of plug-in's are available
- Use “beautiful and fun” Groovy language

Grails Productivity Gain Real-Life Stories

- Developer 1 (David C.)
  - Size of the code: Generally less than 1/2 of Java code
  - Clean-ness of the code: Groovy/Grails is easier to write clean code the first time, and it is much more readable
  - Productivity: Usually 2-5X more productive
  - http://groovy.329449.n5.nabble.com/Groovy-productivity-gain-in-comparison-with-Java-td3346539.html
- Developer 2 (Hamlet D)
  - Writing tests in Groovy for existing Java project was a great win

Job Growth Trends

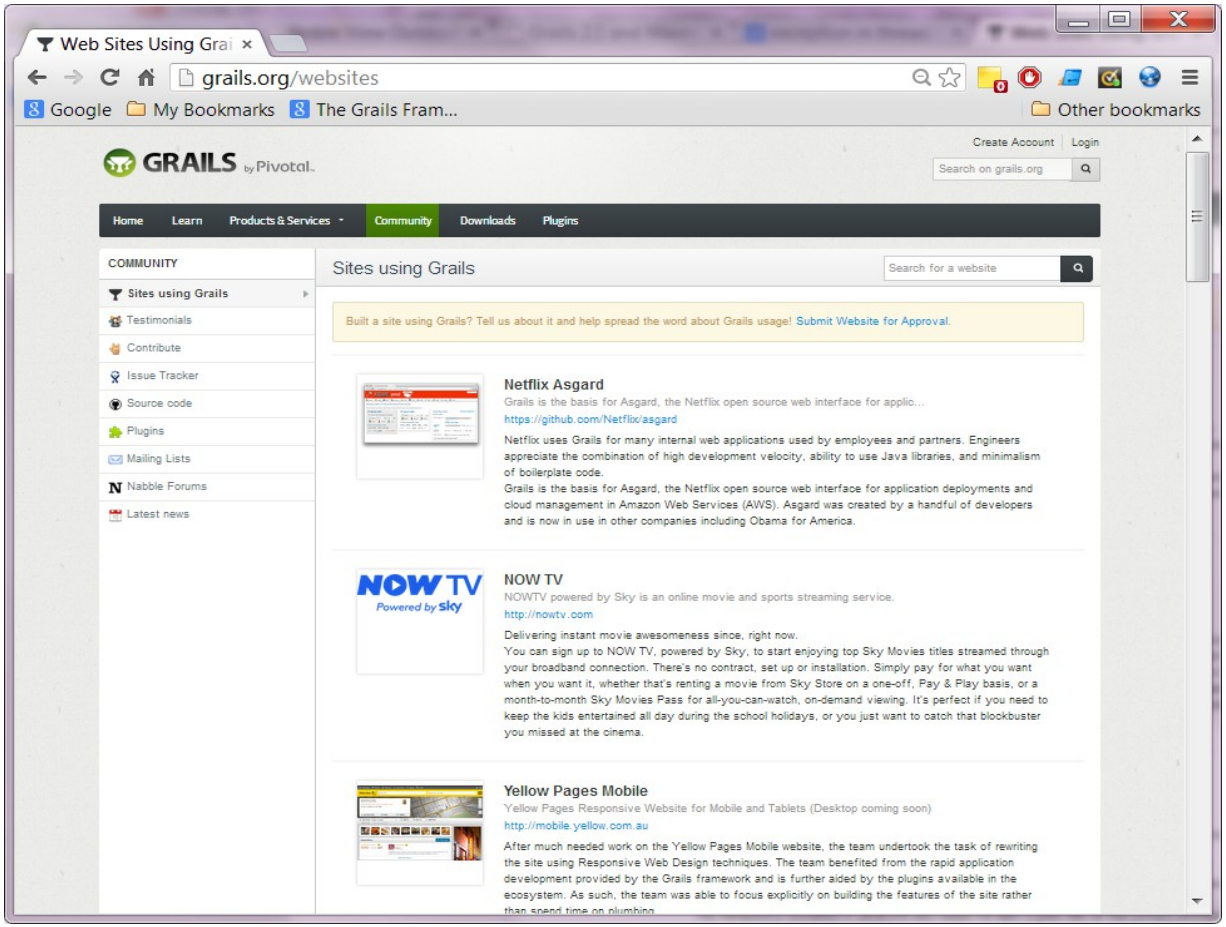
- Grails still has the highest growth trend (as of June, 2013) – absolute jobs are still much smaller than Rails/Spring



http://www.indeed.com/jobtrends?q=rails%2C+grails%2C+spring+java&l=&relative=1

Grails in Real-life

- http://grails.org/websites (Production sites that use Grails)



Pivotal Pulls Groovy/Grails Funding

- http://www.infoq.com/news/2015/01/Pivotal-Pulls-Groovy-Grails-Fund
- How will this impact Groovy/Grails March and beyond?
  - (Guillaume Laforge) We continue to develop Groovy and Grails as usual, we have plenty of work to do and ideas to develop. The risk is that we may not necessarily be able to develop at full speed as usual, but at a more reduced pace
  - (Guillaume Laforge) Thanks to the community's contributions, we know Groovy and Grails will live on and continue to evolve

Getting Started

Getting Started

- *grails create-app helloworld*
  - Create a project directory - “helloworld”, under which common set of directories and files are created
  - Fully functioning app is created
- *grails run-app (executed under “helloworld” directory)*
  - Now you can go to *http://localhost:8080* to see the running app

Directory Structure of Grails App

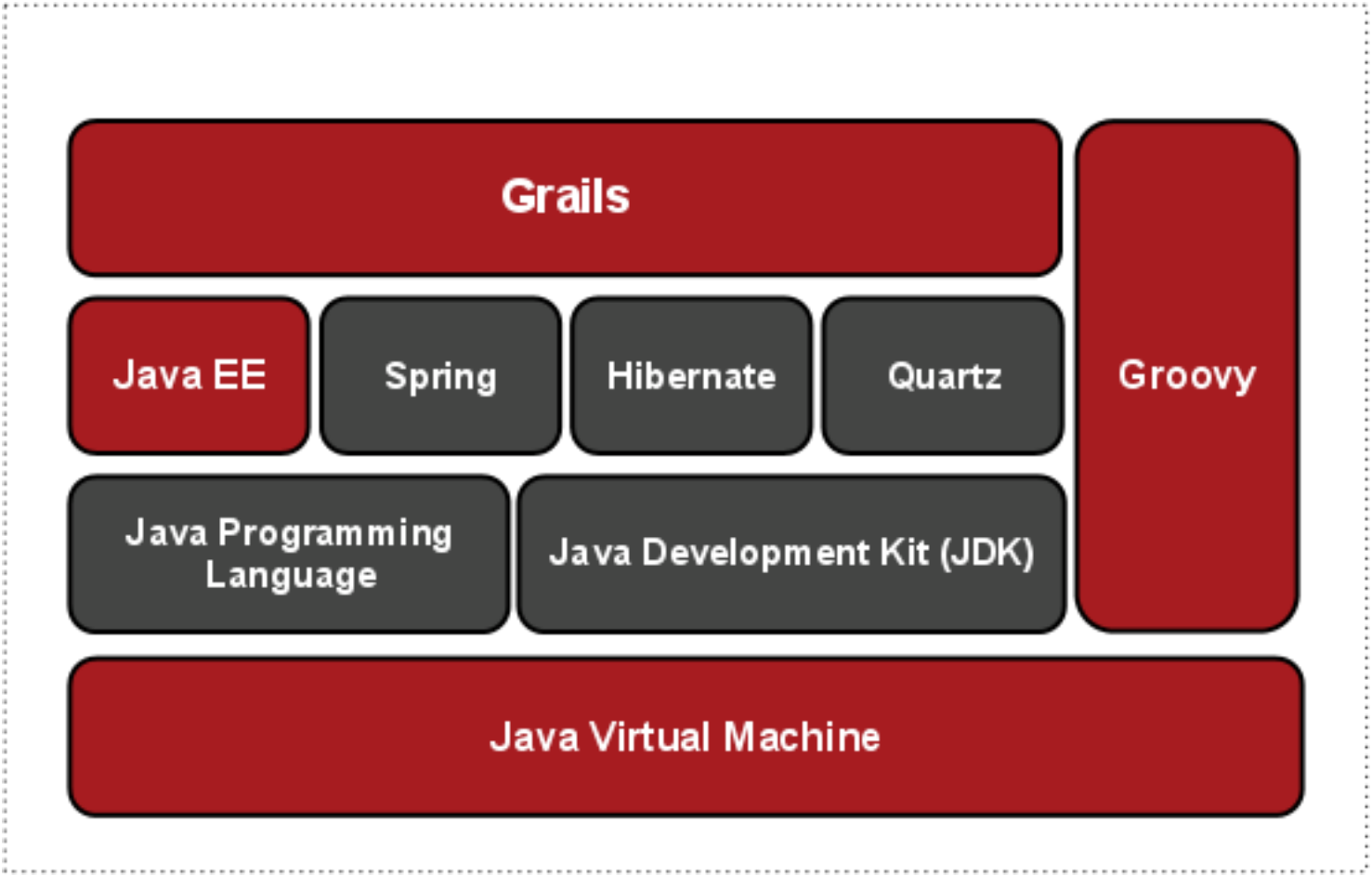
```
%PROJECT_HOME%
    + grails-app
       + conf                  ---> location of configuration artifacts
           + hibernate         ---> optional hibernate config
           + spring            ---> optional spring config
       + controllers           ---> location of controller artifacts
       + domain                ---> location of domain classes
       + i18n                  ---> location of message bundles for i18n
       + services              ---> location of services
       + taglib                ---> location of tag libraries
       + util                  ---> location of special utility classes
       + views                 ---> location of views
           + layouts           ---> location of layouts
   + lib
   + scripts                   ---> scripts
   + src
       + groovy                ---> optional; location for Groovy source files
                                    (of types other than those in grails-app/*)
       + java                  ---> optional; location for Java source files
   + test                      ---> generated test classes
   + web-app
       + WEB-INF
```

Lab:

Exercise 0: Setup
Exercise 1: Build & Run “helloworld” Grails App
5624_grails_basics.zip

Grails MVC Framework

Grails Architecture

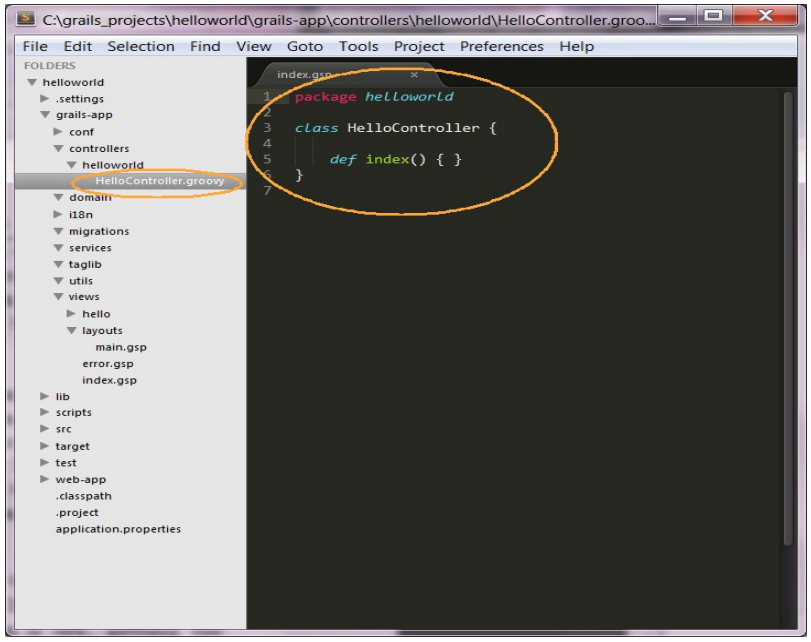


“MVC” of Grails MVC Framework

- Controller (C)
  - Has a set of actions
  - Actions receive client requests
  - Actions might call Services
  - Services perform some tasks creating/manipulating model objects
  - Actions select a view at the end (implicitly or explicitly)
  - Pass model objects to a selected view
- Domain class (M)
  - Typically mapped to the database tables through GORM
- Groovy Server Pages (V)
  - Represents UI of the application
  - Receives user entered data and send it to the controller
  - Receives model objects from the controller for displaying

Create a Controller

- *grails create-controller Hello*
  - grails-app/controllers/helloworld/HelloController.groovy
  - grails-app/views/hello
  - test/unit/helloworld/HelloControllerTests.groovy

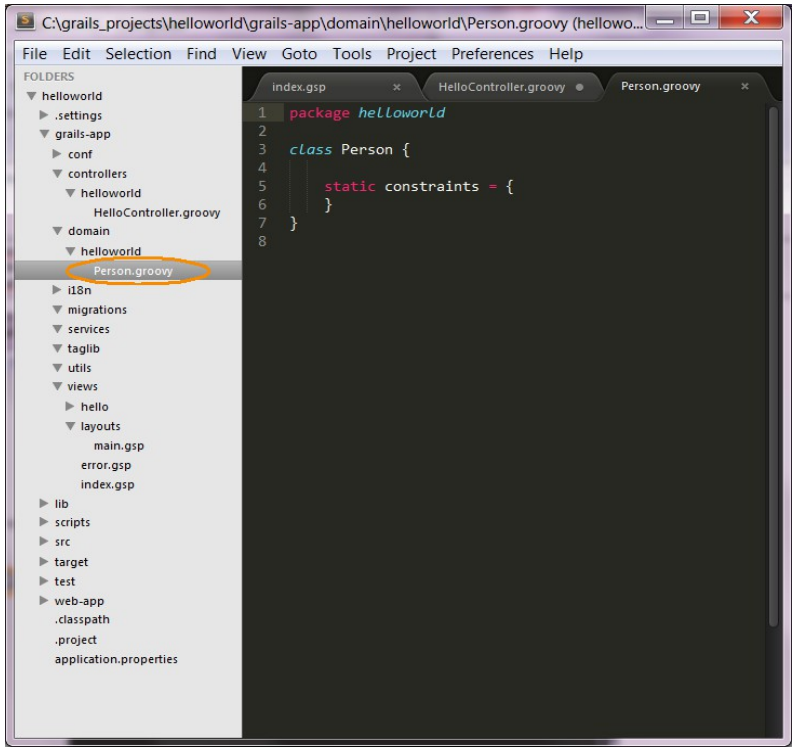


Lab:

Exercise 2: Create a new controller
5624_grails_basics.zip

Create a Domain Class

- *grails create-domain-class Person*
  - grails-app/domain/helloworld/Person.groovy
  - test/unit/helloworld/PersonTests.groovy



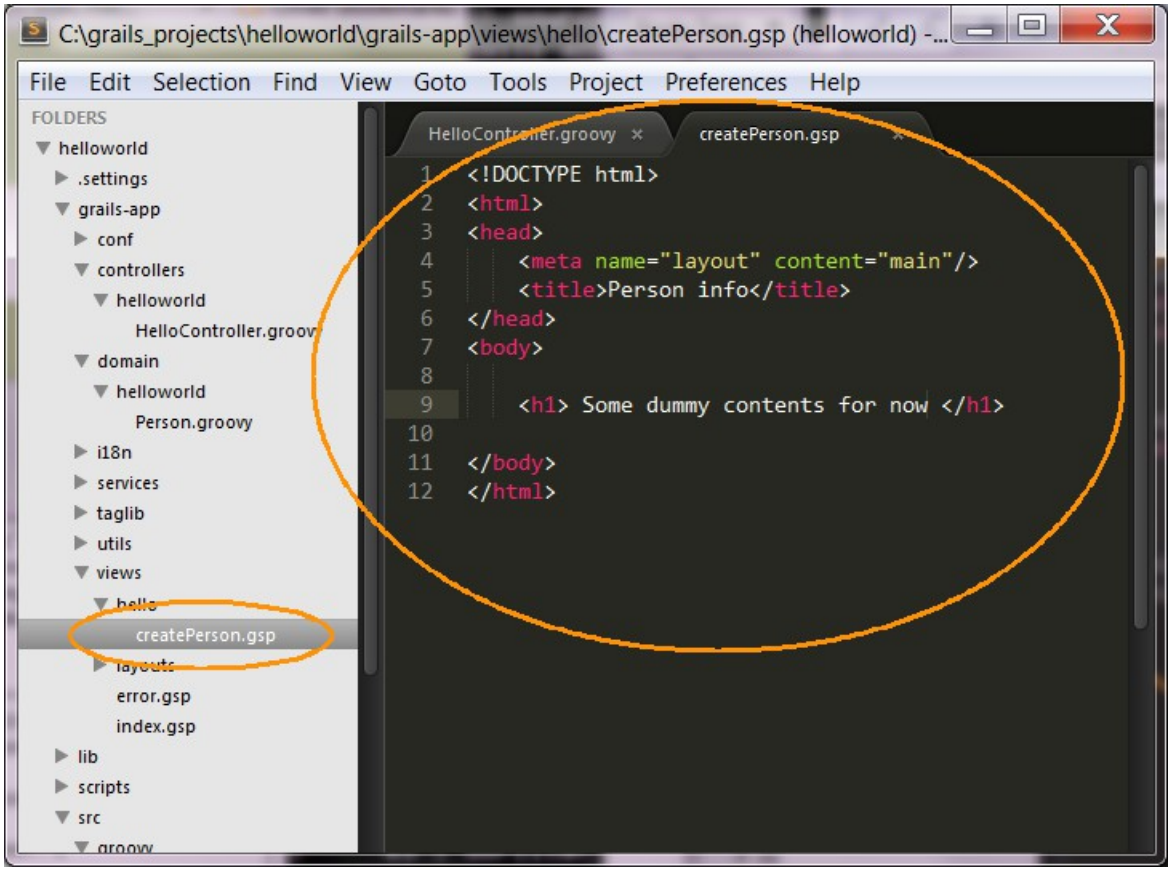
Lab:

Exercise 3: Create a domain class

5624_grails_basics.zip

Create GSP Pages

- Manually create GSP page under *grails-app/views/hello* directory



Lab:

Exercise 4: Create GSP pages
5624_grails_basics.zip

Scaffolding

What Is and Why “Scaffolding”?

- Scaffolding is a quick way to build CRUD-based Grails application against a domain class
  - It generates controller actions and views
- Useful for quick prototyping
- Two ways you can do scaffolding
  - Static scaffolding
  - Dynamic scaffolding

Scaffolding against Domain class

- Dynamic scaffolding: Scaffolding without actual code generation

```
class UserController {
    def scaffold = User
}
class UserController {
    def scaffold = true
}
```

- Static scaffolding: Scaffolding with code generation – if domain class is changed, it has to be regenerated

```
grails generate-all mypackage.User
```

Files Generated Through Scaffolding

- Controller with actions
  - index, create, save, show, edit, update, delete
- View files (created)
  - _form.gsp
  - create.gsp
  - edit.gsp
  - list.gsp
  - show.gsp

Lab:

Exercise 5: Scaffolding without code generation
Exercise 6: Scaffolding with code generation
5624_grails_basics.zip

IDE Support

IDE Support

- All major IDE's supports Groovy/Grails
  - Eclipse, NetBeans, IntelliJ IDEA, STS (SpringSource Tool Suite)
- Grails project creation automatically generates Eclipse project
  - Grails project can be readily importable to Eclipse

Lab:

Exercise 7: Create & Run Grails project with Eclipse
5624_grails_basics.zip

Learn with Passion!
JPassion.com