



Creating a Grails Portlet for Luminis 5.x in 10 Easy Steps

Scott D. Umphrey
Senior Programmer/Analyst
(umphrey@ucmo.edu)
University of Central Missouri

2016 MOKA Conference
November 17, 2016



Agenda

- Benefits of Grails
- Drawback of Grails
- Install Grails on Luminis 5.x Server
- Email Checker Portlet
- Create Grails portlet in Luminis in 10 Easy Steps
- Portlet's Default renderView
- Portlet's Default view.jsp
- Edit renderView
- Edit processAction
- Add Database call
- Add No-SQL Database Call
- Caveats



Benefits of Grails

- Convention Over Configuration (Grails in Action)
 - Add a controller class called Shop with an action called order, and Grails will expose it as a URL of /yourapp/shop/order.
 - Place your view files in a directory called /views/shop/order, and Grails will look after linking everything up for you without a single line of configuration.
 - Create a new domain class called Customer, and Grails will automatically create a table called customer in your database.
 - Add some fields to your Customer object, and Grails will automatically create the necessary fields in your customer table on the fly (including the right data types based on the validation constraints you place on them). No SQL required.



Benefits of Grails (cont'd)

- Agile Philosophy
- Frameworks
- Scaffolding and Templetting
- Java Integration
- User Communities
- Productivity Ethos



Drawbacks of Grails

- Huge Learning Curve
- Documentation for Portlets Hard to Find
- Knowledge of many different technologies needed: Java, Spring, Hibernate ...



Install Grails

- Download Grails
- Extract the archive
- Set `GRAILS_HOME`
- Update `PATH` to include `GRAILS_HOME/bin`
- Type “grails” at the command line
- Modify `.cprc` to include `GRAILS_HOME` and `PATH`



Email Checker Portlet

- A student cannot be added to Luminis if the student's email address already exists in Luminis. The Email Checker Portlet allows users to call a stored procedure that checks AD to see if the student's email address is in Luminis



Email Checker Portlet (cont'd)

Email Checker



Please enter Banner ID (700 Number) to search for:

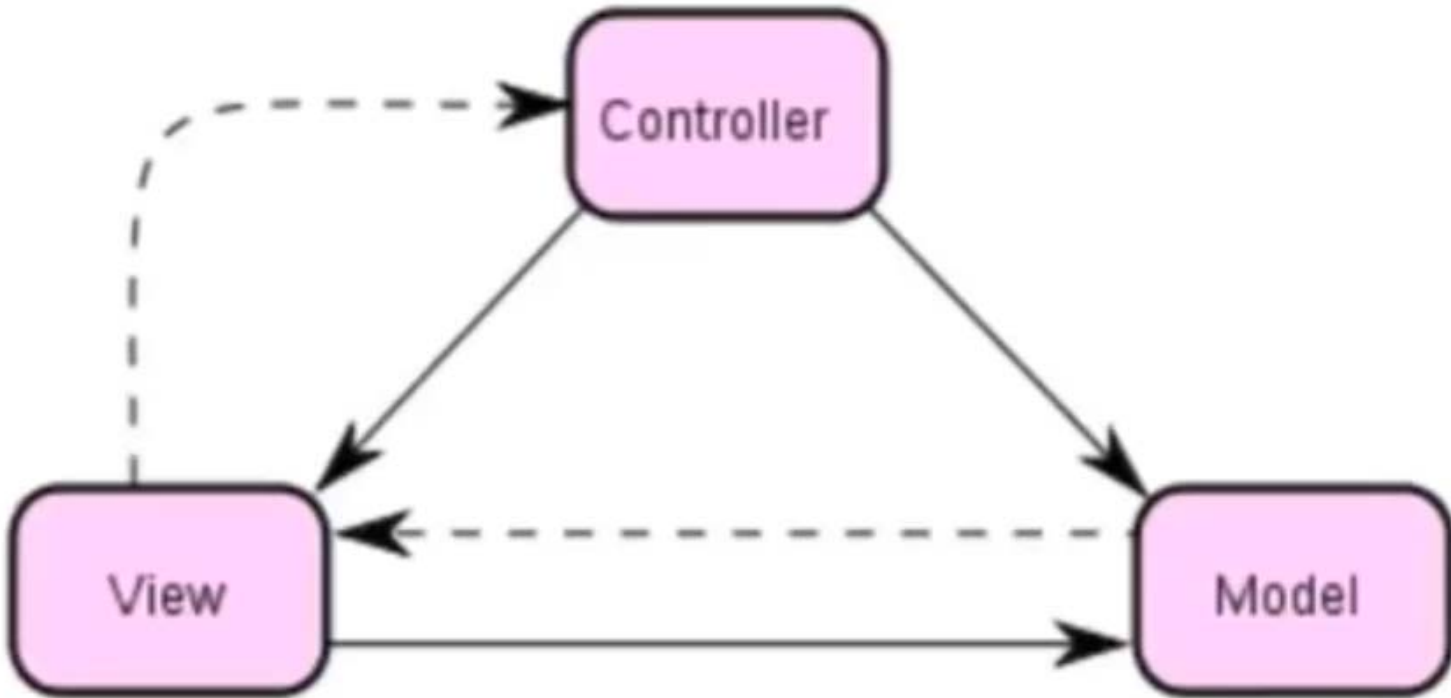
Submit

Cancel

Version 1.1.0



1. Create Project



**Groovy and Grails for Ellucian (Banner) Developers Part
2 of 6 – Youtube - SkillBuilders**



1. Create Project (cont'd)

- Web Application

```
| build.properties
| build.xml
+---build
|   ch2_UserRegistrationPortlet.war
|
+---css
+---images
+---js
+---lib
+---src
|   \---chapter02
|       \---code
|           \---listing
|               User.java
|               UserRegistrationPortlet.java
|
\---WEB-INF (cont'd)
```



1. Create Project (cont'd)

```
\---WEB-INF (cont'd)
| Language-ext.properties
| liferay-display.xml
| liferay-portlet.xml
| portlet.xml
| web.xml
|
+---classes
| +---chapter02
| | \---code
| |   \---listing
| |     User.class
| |     UserRegistrationPortlet.class
| |
| \---content
|   Language-ext.properties
|
+---jsp
|   help.jsp
|   preferences.jsp
|   registrationForm.jsp
|   success.jsp
|
\---lib
   commons-logging.jar
   jstl.jar
   log4j.jar
   standard.jar
```



1. Create Project (cont'd)

- `cd $GRAILS_HOME`
- `grails create-app emailchecker`
- `cd emailchecker`



1. Create Project (cont'd)

- Grails Web Application

```
+---grails-app
| +---conf
| +---controllers      /*C - Controller*/
| +---domain           /*M - Model*/
| +---i18n
| +---services
| +---taglib
| +---utils
| \---views            /*V - View*/
+---lib
+---scripts
+---src
+---target
+---test
\---web-app
```



Config Files

+---conf

- | | ApplicationResources.groovy
- | | BootStrap.groovy
- | | **BuildConfig.groovy**
- | | Config.groovy
- | | **DataSource.groovy**
- | | UrlMappings.groovy



2. Edit BuildConfig.groovy

- Add Entry for Proxy Server – Vortex
System.setProperty("http.proxyHost",
"[server].ucmo.edu")
System.setProperty("http.proxyPort", "port")
- Add dependencies entry
compile "javax.portlet:portlet-api:2.0"
- Add compile entries
compile "org.apache.portlets:portlet-api:0.9.2"
compile "org.apache.portlets:portlet-api-liferay:0.4"



3. Edit DataSource.groovy

- **datasource**

```
pooled = true
```

```
driverClassName = "oracle.jdbc.OracleDriver"
```

```
username = "[username]"
```

```
password = "[password]"
```

```
dialect="org.hibernate.dialect.Oracle9iDialect"
```

- **development-datasource**

```
dbCreate = "" // one of 'create', 'create-drop', 'update', 'validate', "
```

```
url =
```

```
"jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=[server])(PORT=[port]))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=[service name])))"
```




3. Edit DataSource.groovy (contd.)

- test-datasource

```
dbCreate = "" // one of 'create', 'create-drop',  
'update', 'validate', "
```

```
url =
```

```
"jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROT  
OCOL=TCP)(HOST=[server])(PORT=[port]))(CONNECT  
_DATA=(SERVER=DEDICATED)(SERVICE_NAME=[servi  
ce name])))"
```



3. Edit DataSource.groovy (contd.)

- production-dataSource

```
dbCreate = "" // one of 'create', 'create-drop', 'update', 'validate', ""
url = "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=[server
name]))(PORT=[port]))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=[service name])))"
pooled = true
properties {
    validationQuery = "SELECT 1 FROM DUAL"
    testOnBorrow = true
    testWhileIdle = true
    testOnReturn = false
    timeBetweenEvictionRunsMillis = 5000
    minEvictableIdleTimeMillis = 60000
    maxWait = 10000
    maxIdle = 25
    maxActive = 50
    initialSize = 5
}/*End of properties*/
```



4. Install JDBC Driver

- Copy ojdbc.jar to:
~/Development/emailchecker/lib/



5. Clean and Refresh

- grails clean
 - Deletes all compiled resources from the application
- grails refresh-dependencies
 - Refreshes application dependencies and installs any required plugins as necessary



6. Generate Portlet

- `grails create-portlet EmailChecker`
 - +---conf
 - +---controllers
 - +---domain
 - +---i18n
 - +---migrations
 - +---portlets
 - | \---emailchecker
 - | EmailCheckerPortlet.groovy
 - +---services
 - +---taglib
 - +---utils
 - \---views



7. Generate Portlet Views

- grails generate-portlet-views EmailChecker

```
+---conf
+---controllers
+---domain
+---i18n
+---migrations
+---portlets
+---services
+---taglib
+---utils
\---views
|  error.gsp
|  index.gsp
|
+---emailChecker
|  edit.gsp
|  help.gsp
|  view.gsp
|
\---layouts
    main.gsp
```



8. Edit EmailChecker.groovy

- `def title = 'Email Checker'`
- `def description = '''
Email Checker
'''`
- `def displayName = 'Email Checker'`
- `def liferay_display_category = "UCM Custom Portlets"`

Note: By default, the above line is commented out.



9. Package Portlet

- grails [dev | prod] war



10. Deploy WAR File

- Copy

From:

~/Development/emailchecker/target/emailchecker-0.1.war

To:

\$CP_ROOT/products/liferay/liferay-admin/deploy/

\$CP_ROOT/products/liferay/liferay-portal/deploy/



Portlet's Default renderView

```
def renderView = {  
    //TODO Define render phase for 'view' portlet  
    mode.  
    //Return the map of the variables bound to  
    the view,  
    //in this case view.gsp if it exists or render.gsp  
    if not  
        ['mykey':'myvalue']  
}
```



Portlet's Default view.jsp

```
<%@ taglib uri="http://java.sun.com/portlet"  
prefix="portlet" %>
```

```
<div>
```

```
<h1>View Page</h1>
```

The map returned by `renderView` is passed in.

Value of `mykey`: `${mykey}`

```
</div>
```



Edit Portlet's renderView

- Pseudocode

Get message from portletRequest

Place in map []



Edit Portlet's processAction

- Pseudocode

Get Banner ID (700 Number)

| LOOP Through All Validations

| END LOOP

| IF No Errors

| Call Package and receive results

| Set portletResponse.setRenderParameter (to
| pass results back to renderView)

| END

GoTo renderView (automatic)



Add Database Call to actionView

- Obtain Access to dataSource

```
ApplicationContext ctx =
```

```
grailsApplication.mainContext
```

```
def dataSource = ctx.getBean('dataSource')
```



Add Database Call to actionView (contd.)

- Create, Execute, and Return SQL

```
try {
  def sql = new Sql(dataSource)
  def queryGetPersonId = """
SELECT
  person_id ID_Found
FROM
  luminis.lp_person@[database link] per
WHERE
  per.portal_user_id = ?
  """

  def resultGetPersonId = sql.firstRow(queryGetPersonId, [portalUserId])
  personId = resultGetPersonId.ID_Found
  sql.close()
} catch (Exception e) {...}
```



Add No-SQL Database Call (contd.)

- Create Domain Class

grails create-domain-class Spriden



Add No-SQL Database Call

- Edit `grails-app/domain/emailchecker/Spriden.groovy`

```
class Spriden {  
  
    static mapping = {  
  
        table name: 'SPRIDEN', schema: 'SATURN'  
        version false  
        id column: "SPRIDEN_SURROGATE_ID"  
  
        pidm column: "SPRIDEN_PIDM"  
        bannerid column: "SPRIDEN_ID"  
        firstname column: "SPRIDEN_FIRST_NAME"  
        lastname column: "SPRIDEN_LAST_NAME"  
    }  
  
    Integer id  
    Integer pidm  
    String bannerid  
    String firstname  
    String lastname  
  
    static constraints = {}  
}
```



Add No-SQL Database Call (contd.)

- Add to actionView

```
def results1 = Spriden.findAllById(person_Id)
```

```
def pidm = results1.pidm
```

```
def bannerid = results1.bannerid
```

```
def firstname = results1.firstname
```

```
def lastname = results1.lastname
```



Caveats

- Scaffolding Should Not Be Used For Production
- Be careful to change dataSource when Creating WAR file
- Use jconsole to monitor JVM
.../jdk/bin/
- Watch PermGen in production
- Use the following logs to help debugging:
\$CP_ROOT/products/tomcat/tomcat-[admin | portal]/logs/catalina.out
\$CP_ROOT/products/tomcat/tomcat-[admin | portal]/logs/luminis.log



Summary

- Benefits of Grails
- Drawback of Grails
- Installed Grails on Luminis 5.x Server
- Reviewed Email Checker Portlet
- Created Grails portlet in Luminis in 10 Easy Steps
- Viewed Portlet's Default renderView
- Viewed Portlet's Default view.jsp
- Edited renderView
- Edited processAction
- Added Database call
- Added No-SQL Database Call
- Caveats



Resources

- **Grails in Action. Glen Smith and Peter Ledbrock. Manning. 2009.**
- **grails.org**
- **Building Grails Portlets for Luminis and Banner. Ellucian Live. April 9, 2014. Session ID 1529**
- **Liferay in Action. Richard Sezov, Jr. Manning. 2012.**
- **Portlets in Action. Ashish Sarin. Manning. 2012.**
- **Groovy and Grails for Ellucian (Banner Developers) - Part One (MVC)**
<https://www.youtube.com/watch?v=qrPJakbJimE>



Thank You!

Scott D. Umphrey

Senior Programmer/Analyst

(umphrey@ucmo.edu)

University of Central Missouri