

# Seam - ##### [###20080327]

Java EE##

Version: 2.0GA

---

# Table of Contents

JBoss Seam##	xi
<b>1. Seam ##</b>	<b>1</b>
1.1. ###	1
1.1.1. #JBoss AS#####	1
1.1.2. #Tomcat#####	1
1.1.3. ###	1
1.2. #####	1
1.2.1. ###	2
1.2.1.1. ##Bean#User.java	3
1.2.1.2. #####Bean#RegisterAction.java	4
1.2.1.3. ##Bean#####Register.java	6
1.2.1.4. Seam#####components.xml	6
1.2.1.5. Web#####web.xml	6
1.2.1.6. JSF###faces-config.xml	7
1.2.1.7. EJB#####ejb-jar.xml	7
1.2.1.8. EJB#####persistence.xml	8
1.2.1.9. ###register.xhtml # registered.xhtml	8
1.2.1.10. EAR#####application.xml	9
1.2.2. ###	10
1.3. Seam#####	10
1.3.1. ###	11
1.3.1.1. ##Bean#Message.java	11
1.3.1.2. #####Bean#MessageManagerBean.java	12
1.3.1.3. ##Bean#####MessageManager.java	13
1.3.1.4. ###messages.jsp	14
1.3.2. ###	14
1.4. Seam#jBPM#####todo list###	15
1.4.1. ###	15
1.4.2. ###	20
1.5. Seam#####	20
1.5.1. ###	20
1.5.2. ###	24
1.6. #####Seam#####	24
1.6.1. ##	24
1.6.2. #####	25
1.6.3. ##Seam####(Conversation)	26
1.6.4. Seam#UI###	31
1.6.5. Seam####	31
1.7. #####Seam#jBPM#####DVD##	32
1.8. ##Seam#Hibernate####Hibernate####	34
1.9. ##RESTful#Seam#####Blog##	34
1.9.1. ##“#”###MVC	35
1.9.2. #####	36
1.9.3. #RESTful#####“#”###"push"-style##MVC	38
<b>2. #Seam-gen##</b>	<b>41</b>
2.1. ###	41
2.2. #####Eclipse##	41
2.3. #####	43

2.4. #####form# .....	44
2.5. ##### .....	45
2.6. #####EAR .....	45
2.7. Seam##### .....	45
2.8. #Jboss 4.0###Seam .....	46
2.8.1. ##JBoss 4.0 .....	46
2.8.2. ##JSF 1.2 RI .....	46
3. ##### .....	47
3.1. Seam### .....	47
3.1.1. Stateless context##### .....	47
3.1.2. Event context##### .....	47
3.1.3. Page context##### .....	48
3.1.4. Conversation context##### .....	48
3.1.5. Session context#Session#### .....	48
3.1.6. Business process context ##### .....	48
3.1.7. Application context##### .....	48
3.1.8. Context variables##### .....	48
3.1.9. Context##### .....	49
3.1.10. ##### .....	49
3.2. Seam ## .....	49
3.2.1. ###Session Bean .....	50
3.2.2. ###Session Bean .....	50
3.2.3. ##Bean .....	50
3.2.4. JavaBeans .....	51
3.2.5. #####Bean .....	51
3.2.6. ## .....	51
3.2.7. ##### .....	51
3.2.8. #####Defining the Component Scope# .....	52
3.2.9. #####Components with multiple roles# .....	52
3.2.10. ##### .....	53
3.3. ##### .....	53
3.4. Lifecycle methods##### .....	55
3.5. #####Conditional installation# .....	55
3.6. ## .....	56
3.7. Mutable###@ReadOnly .....	57
3.8. Factory#Manager## .....	58
4. ##Seam## .....	60
4.1. ##### .....	60
4.2. ## components.xml ##### .....	60
4.3. ##### .....	62
4.4. ##### .....	63
4.5. ##XML#### .....	64
5. ##### .....	67
5.1. Seam## .....	67
5.1.1. ##### .....	67
5.1.1.1. ##### .....	68
5.1.1.2. ## .....	69
5.1.1.3. ##### .....	71
5.1.2. ##### .....	72
5.1.3. ##### .....	73
5.2. Seam ### .....	74
5.3. ##### .....	75

5.3.1. #####	75
5.3.2. ##Seam####	76
5.3.3. #####	76
5.3.4. #XML####	76
5.3.5. #####	77
<b>6. #####</b>	<b>79</b>
6.1. Seam#####	79
6.2. #####	80
6.3. ##GET#####	81
6.4. ##<s:link>##<s:button>	82
6.5. #####	83
6.6. ##“##”###id	83
6.7. #####	84
6.7.1. #####JSF##	84
6.7.2. #####jPDL###	84
6.7.3. #####	85
6.7.4. #####	85
6.7.5. #####	86
6.8. #####JSF####	86
6.9. #####	87
6.9.1. RichFaces Ajax	88
<b>7. #####</b>	<b>89</b>
7.1. Seam#####	89
7.1.1. #####	89
7.1.2. Seam#####	91
7.2. ##jPDL###	92
7.2.1. #####	92
7.2.2. #####	92
7.2.3. #####	93
7.2.4. #####	94
7.2.5. #####	94
7.2.6. #####	94
7.3. Seam#####	95
7.4. ##jPDL#####	95
7.4.1. #####	95
7.4.2. ###Actor id	96
7.4.3. #####	96
7.4.4. #####	96
7.4.5. #####	96
7.4.6. #####	97
<b>8. Seam###/#####</b>	<b>98</b>
8.1. ##	98
8.2. Seam#####	98
8.2.1. ##Seam#####	99
8.2.2. ##Seam#####	99
8.2.3. #####	99
8.3. Seam#####	100
8.3.1. #Seam#####JPA	100
8.3.2. ##Seam###Hibernate##	100
8.3.3. Seam#####	101
8.4. ##JPA “###delegate#”	101
8.5. #EJB-QL/HQL###EL	102

8.6. ##Hibernate##	102
<b>9. Seam##JSF####</b>	104
<b>10. Groovy##</b>	108
10.1. Groovy##	108
10.2. #Groovy##Seam##	108
10.2.1. ##Groovy##	108
10.2.1.1. ##	108
10.2.1.2. Seam##	109
10.2.2. seam-gen	109
10.3. ##	109
10.3.1. ##Groovy##	110
10.3.2. #####.groovy##	110
10.3.3. seam-gen	110
<b>11. Seam#####</b>	111
11.1. ##	111
11.2. Home##	111
11.3. Query##	115
11.4. Controller##	116
<b>12. Seam#JBoss##</b>	118
12.1. ####	118
12.2. #Seam#####	118
12.3. #jBPM#####	119
<b>13. ##</b>	121
13.1. ##	121
13.1.1. #####	121
13.2. ##	121
13.3. ####	121
13.4. ##	122
13.4.1. ##	122
13.4.2. #####	122
13.4.2.1. Identity.addRole()	123
13.4.3. #####	123
13.4.4. #### - ##	124
13.4.5. #####	124
13.4.6. #####	124
13.4.7. HTTP##	125
13.4.7.1. ##Digest###	125
13.4.8. #####	125
13.4.8.1. #####JAAS##	126
13.5. ####	126
13.6. ##	126
13.6.1. ####	126
13.6.2. #####	126
13.6.2.1. @Restrict##	126
13.6.2.2. #####	127
13.6.3. #####	128
13.6.4. ####	128
13.6.5. #####	129
13.6.5.1. ##JPA#####	130
13.6.5.2. ##Hibernate#####	130
13.7. #####	130
13.7.1. ####	130

13.7.2. #####	130
13.7.3. #####	131
13.7.3.1. #####	132
13.8. SSL##	132
13.9. ##Captcha##	132
13.9.1. ##Captcha Servlet	133
13.9.2. ##Captcha###	133
13.9.3. ##Captcha##	133
<b>14. #####</b>	<b>135</b>
14.1. ###	135
14.2. ##	135
14.2.1. ####	135
14.2.2. ####	136
14.2.3. Faces Messages	136
14.3. ##	137
14.4. ##	137
14.5. ##cookie##locale#####	138
<b>15. Seam Text</b>	<b>139</b>
15.1. #####	139
15.2. #####	140
15.3. ##	141
15.4. ##HTML	141
<b>16. iText PDF##</b>	<b>142</b>
16.1. ##PDF##	142
16.1.1. #####	142
16.1.2. #####	143
16.1.3. #####	146
16.1.4. ##	146
16.1.5. ##	147
16.1.6. ##	148
16.1.7. ####	150
16.1.7.1. ###	150
16.1.7.2. #####	150
16.1.8. iText##	151
16.2. ##	151
16.3. #####	157
16.4. #####	158
<b>17. ####</b>	<b>159</b>
17.1. #####	159
17.1.1. ##	159
17.1.2. HTML/Text ####	160
17.1.3. #####	161
17.1.4. #####	161
17.1.5. ##	161
17.1.6. ###	161
17.1.7. #####	162
17.2. ####	162
17.3. ##	162
17.3.1. mailSession	163
17.3.1.1. #JBoss AS###JNDI	163
17.3.1.2. Seam####	163
17.4. Meldware	163

17.5. ##	164
<b>18. #####</b>	166
18.1. ##	166
18.1.1. #####	166
18.1.2. ##Quartz Dispatcher#####	168
18.1.3. #####	170
18.2. Seam#####	170
18.2.1. ##	170
18.2.2. #####	170
18.2.3. #####Bean#####	171
18.2.4. #####	171
<b>19. ##</b>	172
19.1. #Seam###JBossCache	172
19.2. #####	173
<b>20. Web Services</b>	174
20.1. #####	174
20.2. ###Web Services	174
20.2.1. #####	175
20.3. Web Servic##	175
<b>21. Remoting</b>	177
21.1. ##	177
21.2. Seam##	177
21.2.1. Hello World##	177
21.2.2. Seam.Component	178
21.2.2.1. Seam.Component.newInstance()	179
21.2.2.2. Seam.Component.getInstance()	179
21.2.2.3. Seam.Component.getComponentName()	179
21.2.3. Seam.Remoting	180
21.2.3.1. Seam.Remoting.createType()	180
21.2.3.2. Seam.Remoting.getTypeName()	180
21.3. EL#####	180
21.4. #####	180
21.5. ###	181
21.5.1. #####ID	181
21.5.2. #####	181
21.6. #####	181
21.7. #####	181
21.7.1. ## / ## ##	181
21.7.1.1. String	182
21.7.1.2. Number	182
21.7.1.3. Boolean	182
21.7.2. JavaBeans	182
21.7.3. Date#Time	182
21.7.4. Enums #####	182
21.7.5. Collections ##	183
21.7.5.1. Bags	183
21.7.5.2. Maps	183
21.8. ##	183
21.9. #####	183
21.9.1. #####	183
21.9.2. #####	183
21.9.3. #####	184

21.10. #####	184
21.10.1. #####	184
21.10.2. #####	184
21.10.3. #####	185
21.10.4. #####	185
21.11. JMS##	185
21.11.1. ##	185
21.11.2. ##JMS##	185
21.11.3. #####	186
21.11.4. #####	186
<b>22. Seam#Google#Web###(GWT)</b>	187
22.1. ##	187
22.2. #####	187
22.3. #GWT#####Seam##	188
22.4. GWT Ant Targets	189
<b>23. Spring Framework##</b>	190
23.1. #Seam#####Spring Bean#	190
23.2. #Spring Bean###Seam###	191
23.3. #Spring Bean###Seam##	191
23.4. Seam#####Spring Bean	192
23.5. ##Spring PlatformTransactionManagement	192
23.6. #Spring###Seam#####	193
23.7. #Spring###Seam###Hibernate##	194
23.8. ##Seam###Spring#####	194
23.9. ##Spring TaskExecutor#@Asynchronous	194
<b>24. Hibernate Search</b>	196
24.1. ##	196
24.2. ##	196
24.3. ##	197
<b>25. Seam###Seam#####</b>	199
25.1. Seam####	199
25.1.1. #Seam#JSF#servlet####	199
25.1.2. ##Facelets	199
25.1.3. Seam Resource Servlet	200
25.1.4. Seam Servlet###	200
25.1.4.1. #####	200
25.1.4.2. #####	200
25.1.4.3. #####	201
25.1.4.4. #####	201
25.1.4.5. RichFaces	201
25.1.4.6. Identity Logging	202
25.1.4.7. ##Servlet#####	202
25.1.4.8. #####	202
25.1.5. #Seam###EJB####	203
25.1.6. ###	203
25.2. #Java EE 5###Seam	203
25.2.1. ##	204
25.3. #J2EE###Seam	205
25.3.1. #Seam###Hibernate	205
25.3.2. #Seam###JPA	205
25.3.3. ##	206
25.4. #Java SE###Seam#####JBoss	206



25.5. #####JBoss#Java SE###Seam .....	207
25.5.1. #####JBoss .....	207
25.5.2. ## .....	208
25.6. #Seam###jBPM .....	208
25.6.1. ## .....	209
25.7. #Portal###Seam .....	210
25.8. #JBoss AS###SFSB##### .....	210
<b>26. Seam on OC4J .....</b>	<b>211</b>
26.1. jee5/booking ## .....	211
26.1.1. ##### .....	211
26.1.2. OC4J##### .....	211
26.1.3. ##### .....	212
26.1.4. ## jee5/booking ## .....	212
26.2. ##Seam#####OC4J# .....	212
26.3. ##### seam-gen #####OC4J## .....	213
26.3.1. seam-gen#####OC4J##### .....	217
<b>27. Seam## .....</b>	<b>219</b>
27.1. ##### .....	219
27.2. ##### .....	221
27.3. ##### .....	223
27.4. ##### .....	223
27.5. ###J2EE#####Seam JavaBean##### .....	226
27.6. ##### .....	226
27.7. ##Seam Remoting ### .....	227
27.8. ##Seam#####interceptor#### .....	227
27.9. #####asynchronicity#### .....	227
27.10. ##JSF### .....	228
27.10.1. # dataTable ##### .....	228
27.11. ##### .....	229
27.12. #####packing)### .....	229
27.13. ###Servlet##### .....	230
<b>28. ##Seam## .....</b>	<b>231</b>
28.1. ##### .....	231
28.2. ##### .....	231
28.3. ##### .....	232
28.4. ##### .....	234
28.5. #jBPM##### .....	235
28.6. ##### .....	236
28.7. #JMS##### .....	236
28.8. ##### .....	237
28.9. ##### .....	237
28.10. ##### .....	239
28.11. ##### .....	239
<b>29. Seam#JSF## .....</b>	<b>242</b>
29.1. ## .....	242
29.2. ## .....	254
<b>30. ##### .....</b>	<b>255</b>
30.1. ##### .....	255
30.1.1. ## .....	255
30.1.2. ## .....	255
30.1.2.1. #JSP 2.1### .....	255
30.1.2.2. #Java##### MethodExpression .....	255

30.2. #####	256
30.3. ##	256
<b>31. ##Seam####</b>	258
31.1. Seam#####	258
31.2. Seam#####	259
31.2.1. #####Mock##	260
31.3. ####Seam#####	260
31.3.1. ##Mock#####	263
31.3.2. Seam Mail####	263
<b>32. Seam##</b>	265
32.1. jBPM#####	265
32.1.1. #####	265
32.1.2. Pageflow###	265
<b>33. ###</b>	267
33.1. #####	267
33.1.1. Core	267
33.1.2. Ajax4JSF / RichFaces	267
33.1.3. Seam Mail	268
33.1.4. Seam PDF	268
33.1.5. JBoss Rules	269
33.1.6. JBPM	269
33.1.7. GWT	269
33.1.8. Spring	269
33.1.9. Groovy	270
33.2. ##Maven####	270
<b>A. Seam 2.0 #####</b>	272
A.1. ##	272
A.2. ####	272
A.2.1. Seam 1.2.1 #####	272
A.2.2. Seam 2.0 Beta 1 #####	272
A.2.3. Seam 2.0 #####	278

---

# JBoss Seam##

Seam#####Java#####

####“##”

Seam##### Seam#####  
#####Web#####

Seam#####  
#####

####Java EE##J2EE####Seam#####Web#####Web#####  
#####J2EE#####Seam#####—  
#####

#JSF#EJB 3.0##

JSF#EJB 3.0#Java EE5#####EJB3#####  
###JSF##### JSF#EJB3##### ##Java  
EE5#####

Seam#JSF#EJB3#####

##“##”##EJB#Seam#####EJB#####“###”#####  
#####3.0#####EJB###  
EJB#####—#####JavaBean####Seam#####Bean##JSF#####

#####EJB 3.0#####  
#####Java#####Seam#####Seam#####“###”#####EJB#####

##AJAX

Seam#####JSF#AJAX####JBoss RichFaces#ICEfaces#  
#####JavaScript#####AJAX###

Seam#####JavaScript#####JavaScript#####action##  
#####JMS#####AJAX#push#####

#####Seam#####AJAX###

#####

Seam#####jBPM#####jBPM#Seam#####

Seam#####jBPM#####jPDL)#####

JSF#####  
#####jBPM#####Seam#####Seam#####

#####

#EJB#####J2EE#####EJB 3.0#####  
#####--#####context#####  
Seam#####application state## J2EE#####Servlet  
Session#Request###  
#####Session#####Session#####Bug#####  
Seam#####Bug#

#####Seam#####context model#####  
 Seam###Servlet## ————— ##### ————— ##### ————— ##### ————— #####  
 #####Hibernate##JPA###ORM#####  
 Seam##### LazyInitializationException#  
 #####Refresh#####Back##### ###post-  
 then-redirect#####Seam##### #####Web#####

### Bijection#####

Inversion of Control##### ## dependency injection##### #####JSF#EJB3#####“####”##  
 ##### stateless services##### ##### JSF#####  
 #####

Bijection#####IoC#####  
 #####  
 #####

### #####Workspace Management#####

Seam##### workspace  
 management##### Seam#####

### ###XML##

#####Java#####  
 J2EE#####“##”#####Java#####XML##### Java 5  
 #####

EJB3.0 #####“#####”#####JSF#####XML##### Seam###EJB 3.0  
 ##### JSF managed  
 bean(JSF##bean#####XML#####XML####JSF#####

### #####

Seam####POJO##### ##Java  
 Web#####Seam#Seam#####  
 #####JUnit#TestNG#####View#JSP##Facelets#####  
 #####IDE#####Seam#####JBoss Embeddable##EJB###

### #####

#####Java EE##### GET###JSF#####Seam#####  
 Seam#####JCP#####

### Web#####HTML##

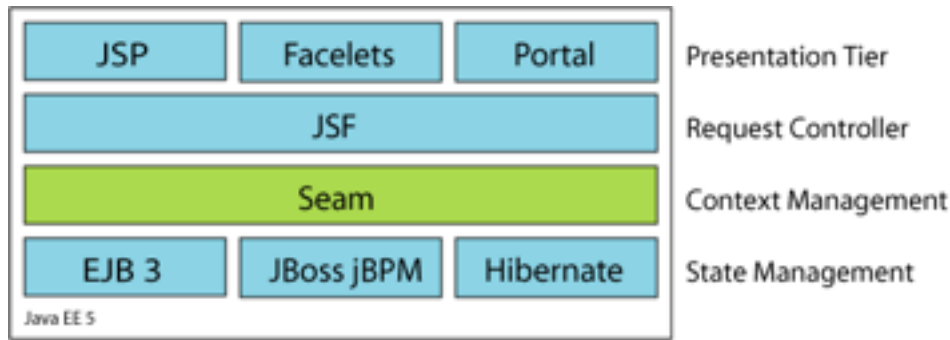
###Web#####Java#####  
 #####Web#####PDF#####wikitext###Web Services##### Seam#####.....

Seam#####JPA#Hibernate 3#####EJB Timer  
 Service#Quartz#####jBPM#####JBoss#####Meldware Mail#####Hibernate  
 Search#Lucene#####JMS#####JBoss Cache#  
 Seam#JAAS#JBoss#####PDF#####wikitext#JSF####  
 Seam#####Web Service#####JavaScript##Google Web Toolkit#####JSF###

### #####

Seam###Java EE#####Tomcat#####EJB 3.0#####  
 #####Seam#####JPA##Hibernate3##### Tomcat###JBoss Embed-

ded#####EJB 3.0 #####



#####Seam#JSF#EJB3#####Java####Web##### # #####

#####Leader#####build###

**Table 1. #####**

##	KB	##	##	##
master.xml	17K		CaoXiaogang	Echo
1. tutorial.xml (1.1-1.4)	130K	20P	seanchan	Jacky
(1.5-1.11)		24P	DigitalSonic	Jacky
2. gettingstarted.xml	21K	6P	seanchan	zaya
3. concepts.xml	56K	16P	CaoXiaogang	kuuyee
4. xml.xml	25K	7P	downpour	Echo
5. events.xml	39K	11P	mochow	xihuyu2000
6. conversations.xml	34K	10P	magice	Echo
7. jbp.xml	32K	10P	##	ronghao
8. persistence.xml	23K	6P	pesome	caoer
9. validation.xml	9K	4P	pesome	DigitalSonic
10. groovy.xml	11K	4P	kuuyee	DigitalSonic
11. framework.xml	20K	7P	alexchang	CaoXiaogang
12. drools.xml	7K	3P	DigitalSonic	shaozhou
13. security.xml	51K	14P	YuLimin	xihuyu2000
14. i18n.xml	14K	4P	YY	DigitalSonic
15. text.xml	7K	3P	DigitalSonic	yeshucheng(###)
16. itext.xml	51K	11P	lyfcdy	Echo
17. mail.xml	26K	7P	chentianyi	yeshucheng(###)
18. jms.xml	11K	5P	YuLimin	caoer
19. cache.xml	11K	3P	crazycy	CaoXiaogang
20. webservices.xml	9K	3P	Echo	YuLimin

##	KB	##	##	##
21. remotng.xml	37K	13P	crazycy	agile_boy
22. gwt.xml	10K	4P	yeshucheng(###)	Echo
23. spring.xml	13K	4P	YY	caoer
24. hsearch.xml	7K	3P	yeshucheng(###)	agile_boy
25. configuration.xml	48K	15P	yeby	kuuyee
26. annotations.xml	64K	14P	caoer	CaoXiaogang
27. components.xml	68K	11P	jiaochar	zaya
28. controls.xml	47K	13P	Echo	YuLimin
29. elenhancements.xml	5K	2P	CaoXiaogang	yeshucheng(###)
30. testing.xml	10K	6P	agile_boy	CaoXiaogang
31. tools.xml	23K	9P	junjzheng	CaoXiaogang
32. oc4j.xml	31K	8P	yeshucheng(###)	YuLimin
33. dependencies.xml	26K	5P	yeshucheng(###)	DigitalSonic

---

# Chapter 1. Seam ##

## 1.1. ###

#####JBoss AS 4.0.5#####EJB 3.0 profile####JBoss AS#####Seam#####

#####

- ##### examples/registration/view #####
- ##### examples/registration/resources #####
- Java##### examples/registration/src ##
- Ant##### examples/registration/build.xml #####

### 1.1.1. #JBoss AS####

#####Ant##### \$ANT\_HOME # \$JAVA\_HOME #####Seam##### build.properties #####JBoss AS 4.0.5##### #####JBoss##### bin/run.sh # bin/run.bat #####JBoss AS#####JDK1.5#####

#####Seam##### examples/registration ### ant deploy #####

#####http://localhost:8080/seam-registration/#

### 1.1.2. #Tomcat#####

#####Ant##### \$ANT\_HOME # \$JAVA\_HOME #####Seam##### build.properties #####Tomcat 6.0#####25.5.1##“#####Jboss”##### (##, SEAM#####Jboss#TOMCAT#####)#

#####Seam##### examples/registration ### ant deploy.tomcat #####

####Tomcat#

#####http://localhost:8080/jboss-seam-registration/#

#####Tomcat#####EJB3####JBoss#####EJB3#####

### 1.1.3. ####

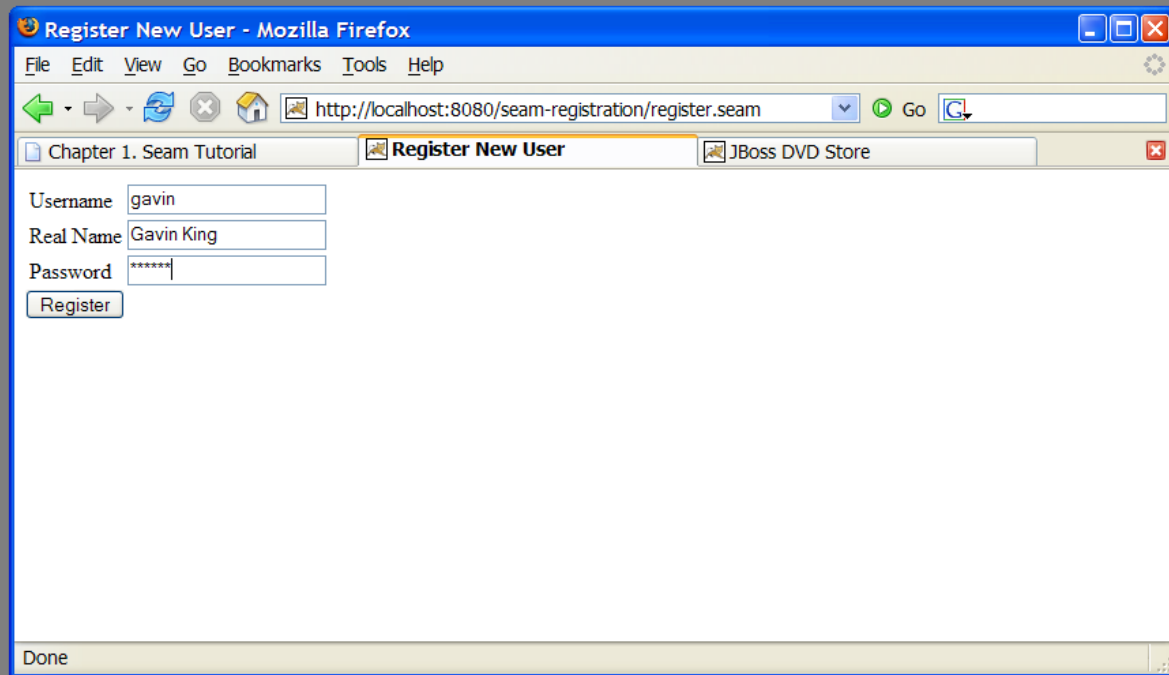
#####TestNG##### examples/registration##### ant  
testexample#####IDE#####TestNG#####

## 1.2. #####

##### Seam##### EJB3  
##Bean##JSF#####Seam#####

####EJB 3.0#####

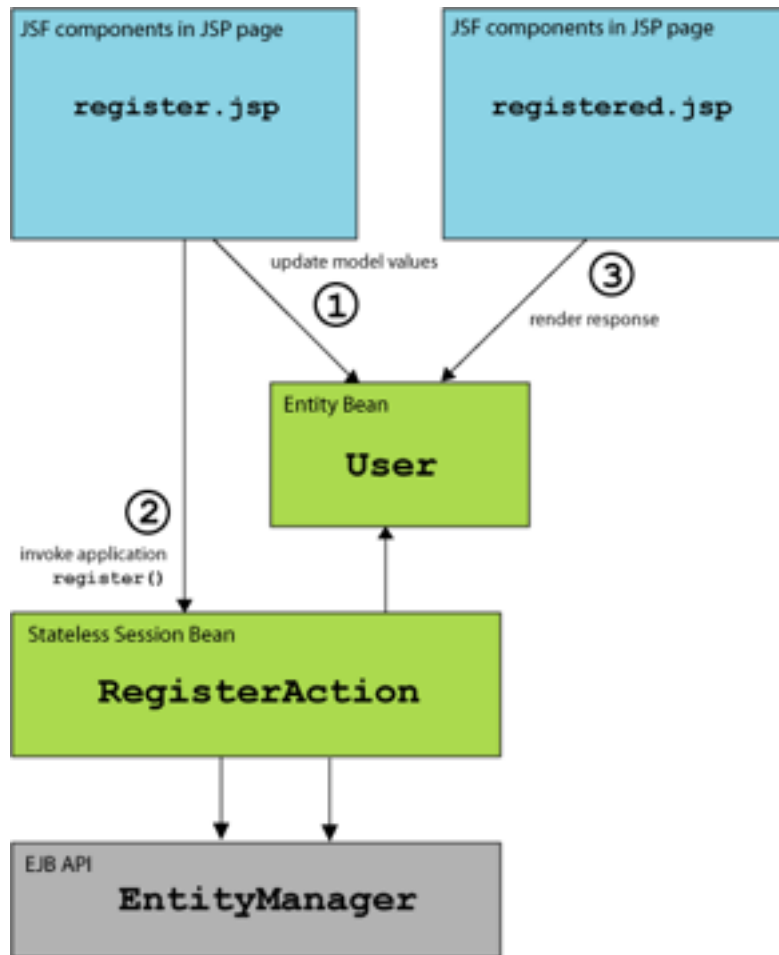
#####user###



### 1.2.1. ####

#####JSP#####Bean#####Bean####





#####“##”###Bean####

#### 1.2.1.1. ##Bean#User.java

####EJB ##Bean##### persistence # validation #####Seam####

#### Example 1.1.

```

@Entity (1)
@Name("user") (2)
@Scope(Session) (3)
@Table(name="users") (4)
public class User implements Serializable
{
    private static final long serialVersionUID = 1881413500711441951L;

    private String username; (5)
    private String password;
    private String name;

    public User(String name, String password, String username)
    {
        this.name = name;
        this.password = password;
        this.username = username;
    }

    public User() {} (6)

    @NotNull @Length(min=5, max=15) (7)
    public String getPassword()
    {

```

```

    return password;
}

public void setPassword(String password)
{
    this.password = password;
}

@NotNull
public String getName()
{
    return name;
}

public void setName(String name)
{
    this.name = name;
}

@Id @NotNull @Length(min=5, max=15)
public String getUsername()
{
    return username;
}

public void setUsername(String username)
{
    this.username = username;
}
}

```

- (1) EJB3#### @Entity ### User #####Bean.
- (2) Seam##### @Name#####Seam#####JSF#####Seam#####  
#####null###Seam##### Seam###JSF##### user #####  
User#
- (3) ##Seam##### @Scope##### User Bean#####
- (4) EJB####@Table #### User #### users ###
- (5) name# password # username  
####Bean#####JSF#####
- (6) EJB#Seam#####
- (7) @NotNull # @Length ###Hibernate Validator##### Seam###Hibernate  
Validator#####Hibernate#####
- (8) ##EJB## @Id #####Bean#####

##### @Name # @Scope #####Seam####

##### User #####JSF####JSF### #####JSP#####Bean#####

#####Bean#####JSF#####Bean#

### 1.2.1.2. #####Bean#RegisterAction.java

#Seam#####Bean###JSF#####JavaBean##

#####JSF#####Bean#####JSF#####Bean#####Bean#####  
##### User Bean##

#####

**Example 1.2.**

```

@Stateless (1)
@Name("register")
public class RegisterAction implements Register
{
    @In (2)
    private User user;

    @PersistenceContext (3)
    private EntityManager em;

    @Logger (4)
    private Log log;

    public String register() (5)
    {
        List existing = em.createQuery(
            "select username from User where username=#{user.username}") (6)
            .getResultList();

        if (existing.size()==0)
        {
            em.persist(user);
            log.info("Registered new user #{user.username}"); (7)
            return "/registered.jsp"; (8)
        }
        else
        {
            FacesMessages.instance().add("User #{user.username} already exists"); (9)
            return null;
        }
    }
}

```

```

(1) EJB#### @Stateless #####Bean#
(2) ## @In#Bean#####Seam#### user #####
(3) EJB#### @PersistenceContext ####EJB#####
(4) Seam# @Logger ##### Log ###
(5) #####EJB3 EntityManager API#####JSF##### Bean#### register()
#####commit##
(6) ###Seam###EJB-QL###JSF EL#####JPA Query #####JPA setParameter() #####
(7) Log API#####
(8) ##JSF#####
#####
####JSF###JSF#####navigation rule# #####JSF##id#####
#####
id#####
#Seam###
#####
#####Seam####JSF##id#####
(9) Seam##### built-in components# ##### # FacesMessages
##### Seam##### instance() #####

##### @Scope#####Seam #####Bean# #####
#####Bean#####

##Bean#####
#####Seam#####

```

```
#####SessionBean#####web##### User #####EntityManager ####
#####J2EE#####J2EE#####Seam##### #
```

### 1.2.1.3. ##Bean#####Register.java

```
#####Bean#####
```

### Example 1.3.

```
@Local
public interface Register
{
    public String register();
}
```

```
###Java#####
```

### 1.2.1.4. Seam#####components.xml

```
#####Java#####XML#####
##Seam#####XML####Seam#####XML####XML#####

##### ## #####Seam##### WEB-INF #### compon-
ents.xml ##### components.xml #####Seam###JNDI###EJB##:
```

### Example 1.4.

```
<components xmlns="http://jboss.com/products/seam/components"
    xmlns:core="http://jboss.com/products/seam/core">
    <core:init jndi-pattern="@jndiPattern@"/>
</components>
```

```
#####Seam##### org.jboss.seam.core.init # jndiPattern
#####@#####ANT#####JNDI#####
```

### 1.2.1.5. Web#####web.xml

```
####WAR#####web#####
```

### Example 1.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

    <!-- Seam -->

    <listener>
        <listener-class>org.jboss.seam.servlet.SeamListener</listener-class>
    </listener>
```

```

<!-- MyFaces -->

<listener>
  <listener-class>
    org.apache.myfaces.webapp.StartupServletContextListener
  </listener-class>
</listener>

<context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>client</param-value>
</context-param>

<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<!-- Faces Servlet Mapping -->
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.seam</url-pattern>
</servlet-mapping>

</web-app>

```

```
# web.xml #####Seam#JSF###Seam#####
```

#### 1.2.1.6. JSF###faces-config.xml

```
#####Seam###JSF#####
```

```
faces-config.xml#SEAM##Facelet#####JSF#Facelet#####
```

#### Example 1.6.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE faces-config
PUBLIC "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.0//EN"
"http://java.sun.com/dtd/web-facesconfig_1_0.dtd">
<faces-config>

  <!-- A phase listener is needed by all Seam applications -->

  <lifecycle>
    <phase-listener>org.jboss.seam.jsf.SeamPhaseListener</phase-listener>
  </lifecycle>

</faces-config>

```

```
#####JSF managed Bean#####managed
Bean#####Seam#####Seam#####faces-config.xml####JSF#####
```

```
##### XML#####jBPM#####Seam### ##process flow# # ####
#####XML####
```

```
#####ID###Action#####
```

#### 1.2.1.7. EJB#####ejb-jar.xml

ejb-jar.xml ### SeamInterceptor #####Bean#####Seam#EJB3####

```
<ejb-jar xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/ ejb-
  version="3.0">

  <interceptors>
    <interceptor>
      <interceptor-class>org.jboss.seam.ejb.SeamInterceptor</interceptor-class>
    </interceptor>
  </interceptors>

  <assembly-descriptor>
    <interceptor-binding>
      <ejb-name>*</ejb-name>
      <interceptor-class>org.jboss.seam.ejb.SeamInterceptor</interceptor-class>
    </interceptor-binding>
  </assembly-descriptor>

</ejb-jar>
```

### 1.2.1.8. EJB#####persistence.xml

persistence.xml #####EJB#####Schema#

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/per
  version="1.0">
  <persistence-unit name="userDatabase">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <jta-data-source>java:/DefaultDS</jta-data-source>
    <properties>
      <property name="hibernate.hbm2ddl.auto" value="create-drop"/>
    </properties>
  </persistence-unit>
</persistence>
```

### 1.2.1.9. ###register.xhtml # registered.xhtml

##Seam#####JSF#####JSP#####  
#####Facelets##

### Example 1.7.

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://jboss.com/products/seam/taglib" prefix="s" %>
<html>
  <head>
    <title>Register New User</title>
  </head>
  <body>
    <f:view>
      <h:form>
        <table border="0">
          <s:validateAll>
            <tr>
              <td>Username</td>
              <td><h:inputText value="#{user.username}"/></td>
            </tr>
            <tr>

```

```

        <td>Real Name</td>
        <td><h:inputText value="#{user.name}"/></td>
    </tr>
    <tr>
        <td>Password</td>
        <td><h:inputSecret value="#{user.password}"/></td>
    </tr>
</s:validateAll>
</table>
<h:messages/>
<h:commandButton type="submit" value="Register" action="#{register.register}"/>
</h:form>
</f:view>
</body>
</html>

```

### <s:validateAll>###Seam#### JSF####JSF#####Bean#####Hibernate#####

### Example 1.8.

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>
<head>
<title>Successfully Registered New User</title>
</head>
<body>
<f:view>
    Welcome, <h:outputText value="#{user.name}"/>,
    you are successfully registered as <h:outputText value="#{user.username}"/>.
</f:view>
</body>
</html>

```

#####JSF###JSP####Seam#####

#### 1.2.1.10. EAR#####application.xml

#####EAR#####

### Example 1.9.

```

<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/
    version="5">

    <display-name>Seam Registration</display-name>

    <module>
        <web>
            <web-uri>jboss-seam-registration.war</web-uri>
            <context-root>/seam-registration</context-root>
        </web>
    </module>
    <module>
        <ejb>jboss-seam-registration.jar</ejb>
    </module>
    <module>
        <java>jboss-seam.jar</java>
    </module>

```

```

    </module>
    <module>
        <java>el-api.jar</java>
    </module>
    <module>
        <java>el-ri.jar</java>
    </module>
</application>

```

#####EAR#####Web##### /seam-registration#

##### ### #####

## 1.2.2. ####

#####JSF##Seam##### user ##### user #####Seam##### Seam#####  
 user#####Seam##### User ##Bean#####JSF#

##### User #####Hibernate##### #####JSF#####JSF##### User ##Bean#####

###JSF##Seam##### register# Seam##### RegisterAction #####Bean#####JSF#### register()  
 #####

Seam#####Seam##### User ###

register() ##### facesmessages #####facesmes-  
 sages #####JSF#####JSF facesmessage #####

#####"/registered.jsp" ##### registered.jsp ## #JSF#####Seam##### user  
 #####Seam##### User #####

## 1.3. Seam#####

##### ##Sean#JSF#####EJB-QL#HQL#####JSF  
 <h:dataTable> #####





### 1.3.1. #####

#####Bean#Message#####Bean MessageListBean ###JSP###

#### 1.3.1.1. ##Bean#Message.java

Message #####title#text#date#time#####

#### Example 1.10.

```
@Entity
@Name("message")
@Scope(EVENT)
public class Message implements Serializable
{
    private Long id;
    private String title;
    private String text;
    private boolean read;
    private Date datetime;

    @Id @GeneratedValue
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
```

```

        this.id = id;
    }

    @NotNull @Length(max=100)
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }

    @NotNull @Lob
    public String getText() {
        return text;
    }
    public void setText(String text) {
        this.text = text;
    }

    @NotNull
    public boolean isRead() {
        return read;
    }
    public void setRead(boolean read) {
        this.read = read;
    }

    @NotNull
    @Basic @Temporal(TemporalType.TIMESTAMP)
    public Date getDatetime() {
        return datetime;
    }
    public void setDatetime(Date datetime) {
        this.datetime = datetime;
    }
}

```

### 1.3.1.2. #####Bean#MessageManagerBean.java

```

#####Bean                                MessageManagerBean                #####
#####

#####MessageManagerBean
##Bean#####JSF#####
#####Seam#####

#####Bean#####

```

### Example 1.11.

```

@Stateful
@Scope(SESSION)
@Name("messageManager")
public class MessageManagerBean implements Serializable, MessageManager
{

    @DataModel                                                    (1)
    private List<Message> messageList;

    @DataModelSelection                                           (2)
    @Out(required=false)                                           (3)
    private Message message;
}

```

```

@PersistenceContext(type=EXTENDED)                                (4)
private EntityManager em;

@Factory("messageList")                                          (5)
public void findMessages()
{
    messageList = em.createQuery("from Message msg order by msg.datetime desc").getResultList();
}

public void select()                                             (6)
{
    message.setRead(true);
}

public void delete()                                             (7)
{
    messageList.remove(message);
    em.remove(message);
    message=null;
}

@Remove @Destroy                                                (8)
public void destroy() {}
}

```

- (1) ## @DataModel ### java.util.List #####JSF##### javax.faces.model.DataModel #### ##JSF  
 <h:dataTable>#####DataModel ##### messageList #####
- (2) @DataModelSelection #####Seam### List #####
- (3) ## @Out ##### Message #####Bean#####  
 #####outjected##### message #####
- (4) #####Bean##EJB3# #####extended persistence context#####Bean#####  
 #####Bean##### EntityManager #####
- (5) #####JSP#####messageList #####@Factory #####Seam### MessageManagerBean #####  
 findMessages() ##### findMessages() ## messages # #####
- (6) select() ##### Message #####
- (7) delete() ##### Message #####
- (8) #####Bean#Seam##### ## ##### @Remove @Destroy  
 #####Seam#####Bean#####

#####Seam##### ##Seam#####

### 1.3.1.3. ##Bean#####MessageManager.java

#####Bean#####

```

@Local
public interface MessageManager
{
    public void findMessages();
    public void select();
    public void delete();
    public void destroy();
}

```

#####

##XML##### components.xml#persistence.xml# web.xml#ejb-jar.xml#faces-con-  
 fig.xml #application.xml #####JSP#

**1.3.1.4. ###messages.jsp**

JSP#####JSF &lt;h:dataTable&gt; #####Seam#####

**Example 1.12.**

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>
<head>
<title>Messages</title>
</head>
<body>
<f:view>
<h:form>
<h2>Message List</h2>
<h:outputText value="No messages to display" rendered="#{messageList.rowCount==0}"/>
<h:dataTable var="msg" value="#{messageList}" rendered="#{messageList.rowCount>0}">
<h:column>
<f:facet name="header">
<h:outputText value="Read"/>
</f:facet>
<h:selectBooleanCheckbox value="#{msg.read}" disabled="true"/>
</h:column>
<h:column>
<f:facet name="header">
<h:outputText value="Title"/>
</f:facet>
<h:commandLink value="#{msg.title}" action="#{messageManager.select}"/>
</h:column>
<h:column>
<f:facet name="header">
<h:outputText value="Date/Time"/>
</f:facet>
<h:outputText value="#{msg.datetime}">
<f:convertDateTime type="both" dateStyle="medium" timeStyle="short"/>
</h:outputText>
</h:column>
<h:column>
<h:commandButton value="Delete" action="#{messageManager.delete}"/>
</h:column>
</h:dataTable>
<h3><h:outputText value="#{message.title}"/></h3>
<div><h:outputText value="#{message.text2}"/></div>
</h:form>
</f:view>
</body>
</html>

```

**1.3.2. ####**

```

##### messages.jsp #####postback##JSF#####GET#####JSP##### mes-
sagelist ##### #####Seam##### findmessages()##### DataModel #####
DataModel ##### <h:dataTable> #####

```

```

##### <h:commandLink> ##JSF### Select() ##### Seam##### messageManager ### message
### ##### Message #####Seam##### Message ### message #### ##EJB##### Mes-
sage #####

```

```

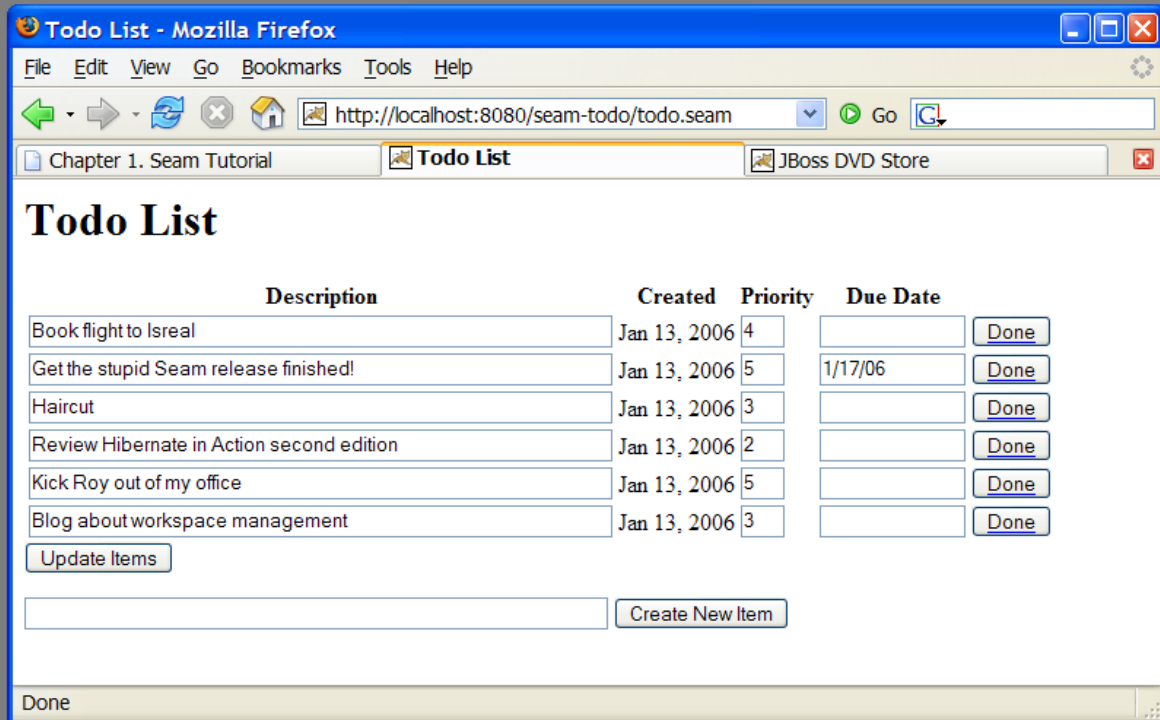
##### <h:commandButton>#JSF### delete() ##### Seam##### messageManager ### mes-
sage ### ##### Message ##### EntityManager ### remove() #####Seam## messageL-

```

ist ##### message ##### ###EJB##### Message #####,#####

## 1.4. Seam#jBPM#####todo list###

jBPM#####jBPM####Seam#####“#####”#####jBPM#####Java###



### 1.4.1. #####

#####jBPM#####process

defini-

tion#####JSP#####JavaBeans#####Bean#####

#### Example 1.13.

```
<process-definition name="todo">

    <start-state name="start">                                     (1)
        <transition to="todo"/>
    </start-state>

    <task-node name="todo">                                       (2)
        <task name="todo" description="#{todoList.description}"> (3)
            <assignment actor-id="#{actor.id}"/>                 (4)
        </task>
        <transition to="done"/>
    </task-node>

    <end-state name="done"/>                                     (5)

</process-definition>
```

```

(1) ## <start-state> ##### todo##
(2) <task-node> #####
(3) <task> ##### Seam## todolist
    #####JavaBeans#####description#
(4) ##### actor
    #Seam#####Seam#####
(5) <end-state>#####

```

```
#####jBossIDE#####
```



```
##### ##### ##### ## #####
```

```
###JavaBean##### login.jsp# ##### actor #####jBPM##id#####
```

### Example 1.14.

```

@Name("login")
public class Login {

    @In
    private Actor actor;

    private String user;

    public String getUser() {
        return user;
    }

    public void setUser(String user) {
        this.user = user;
    }

    public String login()
    {

```

```

        actor.setId(user);
        return "/todo.jsp";
    }
}

```

##### @In ##actor#####Seam### Actor ###

JSP#####

### Example 1.15.

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<html>
<head>
<title>Login</title>
</head>
<body>
<h1>Login</h1>
<f:view>
    <h:form>
        <div>
            <h:inputText value="#{login.user}"/>
            <h:commandButton value="Login" action="#{login.login}"/>
        </div>
    </h:form>
</f:view>
</body>
</html>

```

###JavaBean#####

### Example 1.16.

```

@Name("todoList")
public class TodoList {

    private String description;

    public String getDescription() (1)
    {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    @CreateProcess(definition="todo") (2)
    public void createTodo() {}

    @StartTask @EndTask (3)
    public void done() {}

}

```

- (1) description###JSP#####Seam#####description#  
 (2) Seam# @CreateProcess #####jBPM#####  
 (3) Seam# @StartTask #####@EndTask #####

```
#####@StartTask # @EndTask #####
```

```
##### todo.jsp ##
```

### Example 1.17.

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://jboss.com/products/seam/taglib" prefix="s" %>
<html>
<head>
<title>Todo List</title>
</head>
<body>
<h1>Todo List</h1>
<f:view>
  <h:form id="list">
    <div>
      <h:outputText value="There are no todo items." rendered="#{empty taskInstanceList}"/>
      <h:dataTable value="#{taskInstanceList}" var="task" rendered="#{not empty taskInstanceList}">
        <h:column>
          <f:facet name="header">
            <h:outputText value="Description"/>
          </f:facet>
          <h:inputText value="#{task.description}"/>
        </h:column>
        <h:column>
          <f:facet name="header">
            <h:outputText value="Created"/>
          </f:facet>
          <h:outputText value="#{task.taskMgmtInstance.processInstance.start}">
            <f:convertDateTime type="date"/>
          </h:outputText>
        </h:column>
        <h:column>
          <f:facet name="header">
            <h:outputText value="Priority"/>
          </f:facet>
          <h:inputText value="#{task.priority}" style="width: 30"/>
        </h:column>
        <h:column>
          <f:facet name="header">
            <h:outputText value="Due Date"/>
          </f:facet>
          <h:inputText value="#{task.dueDate}" style="width: 100">
            <f:convertDateTime type="date" dateStyle="short"/>
          </h:inputText>
        </h:column>
        <h:column>
          <s:button value="Done" action="#{todoList.done}" taskInstance="#{task}"/>
        </h:column>
      </h:dataTable>
    </div>
    <div>
      <h:messages/>
    </div>
    <div>
      <h:commandButton value="Update Items" action="update"/>
    </div>
  </h:form>
  <h:form id="new">
    <div>
      <h:inputText value="#{todoList.description}"/>
      <h:commandButton value="Create New Item" action="#{todoList.createTodo}"/>
    </div>
  </h:form>
</f:view>
</body>
```



```
</html>
```

```
#####
```

```
#JSP####Seam#### taskInstanceList #####JSF#####
```

```
<h:form id="list">
  <div>
    <h:outputText value="There are no todo items." rendered="#{empty taskInstanceList}"/>
    <h:dataTable value="#{taskInstanceList}" var="task" rendered="#{not empty taskInstanceList}">
      ...
    </h:dataTable>
  </div>
</h:form>
```

```
#####jBPM# taskinstance #### description#priority#due
date#####
```

```
<h:column>
  <f:facet name="header">
    <h:outputText value="Description"/>
  </f:facet>
  <h:inputText value="#{task.description}"/>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="Created"/>
  </f:facet>
  <h:outputText value="#{task.taskMgmtInstance.processInstance.start}">
    <f:convertDateTime type="date"/>
  </h:outputText>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="Priority"/>
  </f:facet>
  <h:inputText value="#{task.priority}" style="width: 30"/>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="Due Date"/>
  </f:facet>
  <h:inputText value="#{task.dueDate}" style="width: 100">
    <f:convertDateTime type="date" dateStyle="short"/>
  </h:inputText>
</h:column>
```

```
##### @StartTask @EndTask #####id#####Seam#
```

```
<h:column>
  <s:button value="Done" action="#{todoList.done}" taskInstance="#{task}"/>
</h:column>
```

```
#####Seam seam-ui.jar ###JSF <s:button> ####
```

```
#####Seam#jBPM#####
```

```
<h:commandButton value="Update Items" action="update"/>
```

```
##### @CreateProcess#####item##
```

```
<h:form id="new">
  <div>
```

```

<h:inputText value="#{todoList.description}"/>
<h:commandButton value="Create New Item" action="#{todoList.createTodo}"/>
</div>
</h:form>

```

#####jBPM#Seam#####

## 1.4.2. ####

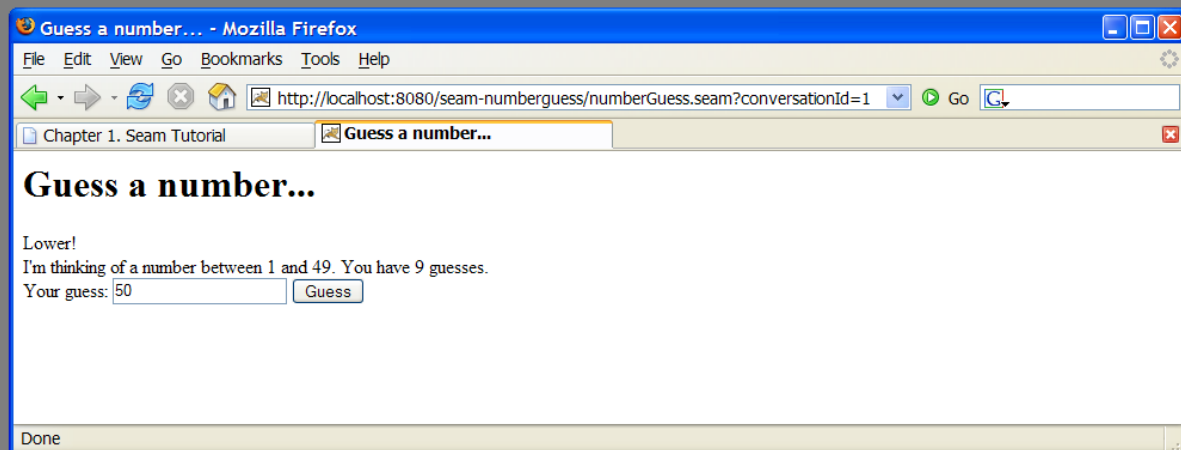
###

## 1.5. Seam#####

#####Seam#####JSF/Seam#####

#####

Seam#####jPDL#####



### 1.5.1. ####

#####JavaBean###JSP#####jPDL#####

### Example 1.18.

```

<pageflow-definition name="numberGuess">

  <start-page name="displayGuess" view-id="/numberGuess.jsp">
    <redirect/>
    <transition name="guess" to="evaluateGuess">
      <action expression="#{numberGuess.guess}" />
    </transition>
  </start-page>

  <decision name="evaluateGuess" expression="#{numberGuess.correctGuess}">
    <transition name="true" to="win"/>
    <transition name="false" to="evaluateRemainingGuesses"/>
  </decision>

  <decision name="evaluateRemainingGuesses" expression="#{numberGuess.lastGuess}">

```

```

    <transition name="true" to="lose"/>
    <transition name="false" to="displayGuess"/>
  </decision>

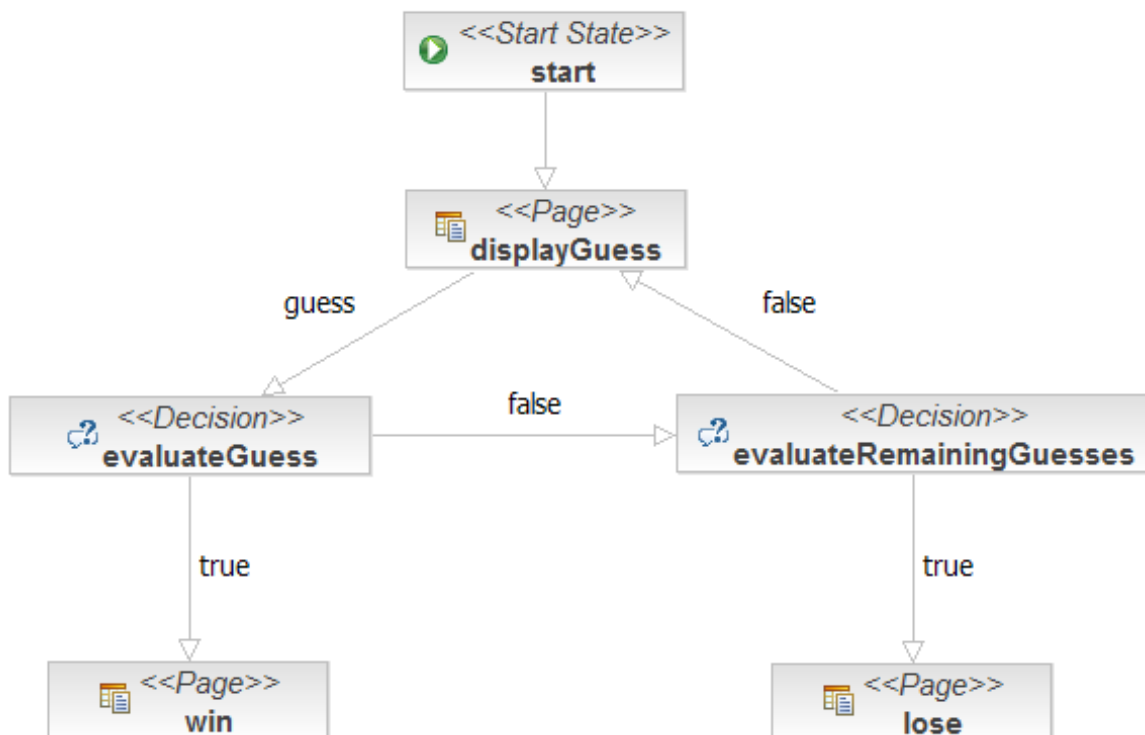
  <page name="win" view-id="/win.jsp">
    <redirect/>
    <end-conversation />
  </page>

  <page name="lose" view-id="/lose.jsp">
    <redirect/>
    <end-conversation />
  </page>
</pageflow-definition>

```

- (1) <page>#####JSF##### view-id###JSF#####view id###  
 redirect####Seam#####post-then-redirect#(#####URL#)
- (2) <transition> #####JSF#####JSF##### ##jBPM#####
- (3) ##### <action> ##JSF#####jBPM##### Seam###
- (4) <decision> #####JSF EL#####

#####JBossIDE#####



#####!

#####numberGuess.jsp#

### Example 1.19.

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<html>

```

```

<head>
<title>Guess a number...</title>
</head>
<body>
<h1>Guess a number...</h1>
<f:view>
    <h:form>
        <h:outputText value="Higher!" rendered="#{numberGuess.randomNumber>numberGuess.currentGuess}" />
        <h:outputText value="Lower!" rendered="#{numberGuess.randomNumber<numberGuess.currentGuess}" />
        <br />
        I'm thinking of a number between <h:outputText value="#{numberGuess.smallest}" /> and
        <h:outputText value="#{numberGuess.biggest}" />. You have
        <h:outputText value="#{numberGuess.remainingGuesses}" /> guesses.
        <br />
        Your guess:
        <h:inputText value="#{numberGuess.currentGuess}" id="guess" required="true">
            <f:validateLongRange
                maximum="#{numberGuess.biggest}"
                minimum="#{numberGuess.smallest}" />
        </h:inputText>
        <h:commandButton type="submit" value="Guess" action="guess" />
        <br />
        <h:message for="guess" style="color: red" />
    </h:form>
</f:view>
</body>
</html>

```

```
##### guess #####
```

```
win.jspx #####
```

### Example 1.20.

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<html>
<head>
<title>You won!</title>
</head>
<body>
<h1>You won!</h1>
<f:view>
    Yes, the answer was <h:outputText value="#{numberGuess.currentGuess}" />.
    It took you <h:outputText value="#{numberGuess.guessCount}" /> guesses.
    Would you like to <a href="numberGuess.seam">play again</a>?
</f:view>
</body>
</html>

```

```
lose.jsp #####JavaBean Seam#####
```

### Example 1.21.

```

@Name("numberGuess")
@Scope(ScopeType.CONVERSATION)
public class NumberGuess {

    private int randomNumber;
    private Integer currentGuess;
    private int biggest;
    private int smallest;

```

```

private int guessCount;
private int maxGuesses;

@Create (1)
@Begin(pageflow="numberGuess") (2)
public void begin()
{
    randomNumber = new Random().nextInt(100);
    guessCount = 0;
    biggest = 100;
    smallest = 1;
}

public void setCurrentGuess(Integer guess)
{
    this.currentGuess = guess;
}

public Integer getCurrentGuess()
{
    return currentGuess;
}

public void guess()
{
    if (currentGuess > randomNumber)
    {
        biggest = currentGuess - 1;
    }
    if (currentGuess < randomNumber)
    {
        smallest = currentGuess + 1;
    }
    guessCount++;
}

public boolean isCorrectGuess()
{
    return currentGuess == randomNumber;
}

public int getBiggest()
{
    return biggest;
}

public int getSmallest()
{
    return smallest;
}

public int getGuessCount()
{
    return guessCount;
}

public boolean isLastGuess()
{
    return guessCount == maxGuesses;
}

public int getRemainingGuesses() {
    return maxGuesses - guessCount;
}

public void setMaxGuesses(int maxGuesses) {
    this.maxGuesses = maxGuesses;
}

public int getMaxGuesses() {
    return maxGuesses;
}

```

```
    }

    public int getRandomNumber() {
        return randomNumber;
    }
}
```

- (1) #####JSP##### numberGuess ###Seam##### @Create #####
- (2) @Begin #####Seam ####(conversation) #####

#####Seam#####

## 1.5.2. ####

TODO

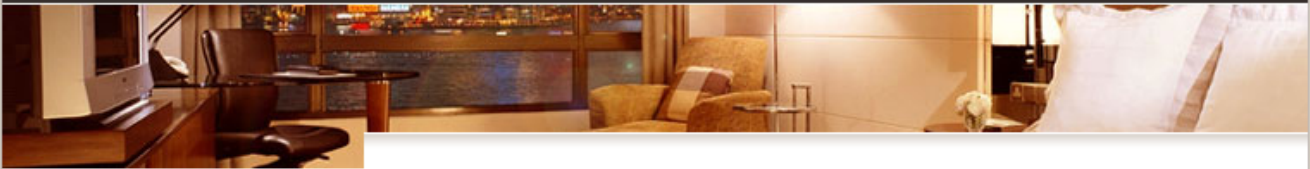
## 1.6. #####Seam#####

### 1.6.1. ##

#####

- ####
- ##
- ##
- ####
- ####
- ####
- ####
- ####
- #####

jboss suites
seam framework demo
Welcome Gavin King | Search | Settings | Logout



### State management in Seam

State in Seam is *contextual*. When you click "Find Hotels", the application retrieves a list of hotels from the database and caches it in the session context. When you navigate to one of the hotel records by clicking the "View Hotel" link, a *conversation* begins. The conversation is attached to a particular tab, in a particular browser window. You can navigate to multiple hotels using "open in new tab" or "open in new window" in your web browser. Each window will execute in the context of a different conversation. The application keeps state associated with your hotel booking in the conversation context, which ensures that the concurrent conversations do not interfere with each other.

[How does the search page work?](#)

**Thank you, Gavin King, your confirmation number for Doubletree is 1**

### Search Hotels

Maximum results:

Name	Address	City, State	Zip	Action
Marriott Courtyard	Tower Place, Buckhead	Atlanta, GA, USA	30305	<a href="#">View Hotel</a>
Doubletree	Tower Place, Buckhead	Atlanta, GA, USA	30305	<a href="#">View Hotel</a>
Ritz Carlton	Peachtree Rd, Buckhead	Atlanta, GA, USA	30326	<a href="#">View Hotel</a>

### Current Hotel Bookings

Name	Address	City, State	Check in date	Check out date	Confirmation number	Action
Doubletree	Tower Place, Buckhead	Atlanta, GA	Apr 16, 2006	Apr 17, 2006	1	<a href="#">Cancel</a>

Created with JBoss EJB 3.0, Seam, MyFaces, and Facelets

```
#####JSF#EJB 3.0#Seam#####Facelets#####JSF#Facelets#Seam#JavaBeans#Hibernate3#
```

```
#####
#####
#####Seam#####web#####Seam#####
#####
```

## 1.6.2. #####

```
#####Section 1.1, "###"##### ht-
```

tp://localhost:8080/seam-booking/ #####

####9####6#Session Bean#####6#Session Bean#####

- BookingListAction #####
- ChangePasswordAction #####
- HotelBookingAction ##### ##### conversation#  
#####
- RegisterAction #####

#####bean###

- Hotel #####Bean
- Booking #####Bean
- User #####Bean

### 1.6.3. ##Seam####(Conversation)

#####  
#####

###Web#####(first class construct)#####  
#####Java#Web#####  
HttpSession#####Request#####

#####  
#####Java#####(##)#####  
#####LRU#####  
#####

##### HttpSession #####session##### Web#####  
#####  
#####session#####  
#####Seam#####

#####

Seam### #####  
#####

#####Session Bean#####Bean#JavaBeans##  
#Java#####Session Bean#####1998#WebFoobar 1.0#####JBoss  
4.0#####Session Bean#####JBoss EJB3#####bean##  
#####Bean##### HttpSession #####Session Bean###Web  
Session#####Bean#####  
#####Seam#####2005##

OK#####



```
##### Seam#session###
#####session#####

#####Ajax4JSF#####JavaScript#####Rich Client####

#####Session#####Session Bean#####Session Bean#
```

### Example 1.22.

```
@Stateful (1)
@Name("hotelSearch")
@Scope(ScopeType.SESSION)
@Restrict("#{identity.loggedIn}") (2)
public class HotelSearchingAction implements HotelSearching
{
    @PersistenceContext
    private EntityManager em;

    private String searchString;
    private int pageSize = 10;
    private int page;

    @DataModel
    private List<Hotel> hotels; (3)

    public String find()
    {
        page = 0;
        queryHotels();
        return "main";
    }

    public String nextPage()
    {
        page++;
        queryHotels();
        return "main";
    }

    private void queryHotels()
    {
        String searchPattern = searchString==null ? "%" : '%' + searchString.toLowerCase().replace('*',
        hotels = em.createQuery("select h from Hotel h where lower(h.name) like :search or lower(h.city
        .setParameter("search", searchPattern)
        .setMaxResults(pageSize)
        .setFirstResult( page * pageSize )
        .getResultList();
    }

    public boolean isNextPageAvailable()
    {
        return hotels!=null && hotels.size()==pageSize;
    }

    public int getPageSize() {
        return pageSize;
    }

    public void setPageSize(int pageSize) {
        this.pageSize = pageSize;
    }

    public String getSearchString()
    {
        return searchString;
    }
}
```

```

public void setSearchString(String searchString)
{
    this.searchString = searchString;
}

@Destroy @Remove
public void destroy() {}

}

```

(4)

- (1) EJB#### @Stateful #####Session Bean#####
- (2) @Restrict#####Seam#####
- (3) @DataModel ##### List ##JSF ListDataModel ##### hotels  
# ListDataModel #####
- (4) EJB#### @Remove #####Session Bean##### #Seam#####Session  
Bean##### @Destroy @Remove ##### ##Seam###Session#####EJB##### @Destroy  
#####Seam##### @Destroy @Remove #####

#####Facelets#####

### Example 1.23.

```

<div class="section">
<h:form>

    <span class="errors">
        <h:messages globalOnly="true"/>
    </span>

    <h1>Search Hotels</h1>
    <fieldset>
        <h:inputText value="#{hotelSearch.searchString}" style="width: 165px;">
            <a:support event="onkeyup" actionListener="#{hotelSearch.find}"
                reRender="searchResults" />
            (1)
        </h:inputText>
        &#160;
        <a:commandButton value="Find Hotels" action="#{hotelSearch.find}"
            styleClass="button" reRender="searchResults"/>
        &#160;
        <a:status>
            (2)
            <f:facet name="start">
                <h:graphicImage value="/img/spinner.gif"/>
            </f:facet>
        </a:status>
        <br/>
        <h:outputLabel for="pageSize">Maximum results:</h:outputLabel>&#160;
        <h:selectOneMenu value="#{hotelSearch.pageSize}" id="pageSize">
            <f:selectItem itemLabel="5" itemValue="5"/>
            <f:selectItem itemLabel="10" itemValue="10"/>
            <f:selectItem itemLabel="20" itemValue="20"/>
        </h:selectOneMenu>
    </fieldset>

</h:form>
</div>

<a:outputPanel id="searchResults">
    (3)
    <div class="section">
        <h:outputText value="No Hotels Found"
            rendered="#{hotels != null and hotels.rowCount==0}"/>
        <h:dataTable value="#{hotels}" var="hot" rendered="#{hotels.rowCount>0}">
            <h:column>
                <f:facet name="header">Name</f:facet>

```

```

        #{hot.name}
    </h:column>
    <h:column>
        <f:facet name="header">Address</f:facet>
        #{hot.address}
    </h:column>
    <h:column>
        <f:facet name="header">City, State</f:facet>
        #{hot.city}, #{hot.state}, #{hot.country}
    </h:column>
    <h:column>
        <f:facet name="header">Zip</f:facet>
        #{hot.zip}
    </h:column>
    <h:column>
        <f:facet name="header">Action</f:facet>
        <s:link value="View Hotel" action="#{hotelBooking.selectHotel(hot)}"/>    (4)
    </h:column>
</h:dataTable>
<s:link value="More results" action="#{hotelSearch.nextPage}"
        rendered="#{hotelSearch.nextPageAvailable}"/>
</div>
</a:outputPanel>

```

- (1) Ajax4JSF# <a:support> #####JSF##### onkeyup ###JavaScript##### XMLHttpRequest ###  
#####reRender #####JSF#####
- (2) Ajax4JSF# <a:status> #####
- (3) Ajax4JSF# <a:outputPanel> #####
- (4) Seam#<s:link> #####JSF#####JavaScript#HTML#### #####JSF# <h:commandLink>  
#####“#####”#“#####”#####  
######{hotelBooking.selectHotel(hot)}#  
#####EL#####Seam#EL#####

##### HotelBookingAction # selectHotel() ##### ## #####

#####Session Bean####Session Bean#####  
#####

### Example 1.24.

```

@Stateful
@Name("hotelBooking")
@Restrict("#{identity.loggedIn}")
public class HotelBookingAction implements HotelBooking
{

    @PersistenceContext(type=EXTENDED)    (1)
    private EntityManager em;

    @In    (2)
    private User user;

    @In(required=false) @Out
    private Hotel hotel;

    @In(required=false)
    @Out(required=false)
    private Booking booking;

    @In
    private FacesMessages facesMessages;

    @In

```

```

private Events events;

@Logger
private Log log;

@Begin (3)
public String selectHotel(Hotel selectedHotel)
{
    hotel = em.merge(selectedHotel);
    return "hotel";
}

public String bookHotel()
{
    booking = new Booking(hotel, user);
    Calendar calendar = Calendar.getInstance();
    booking.setCheckinDate( calendar.getTime() );
    calendar.add(Calendar.DAY_OF_MONTH, 1);
    booking.setCheckoutDate( calendar.getTime() );

    return "book";
}

public String setBookingDetails()
{
    if (booking==null || hotel==null) return "main";
    if ( !booking.getCheckinDate().before( booking.getCheckoutDate() ) )
    {
        facesMessages.add("Check out date must be later than check in date");
        return null;
    }
    else
    {
        return "confirm";
    }
}

@End (4)
public String confirm()
{
    if (booking==null || hotel==null) return "main";
    em.persist(booking);
    facesMessages.add("Thank you, #{user.name}, your confirmation number for #{hotel.name} is #{book:");
    log.info("New booking: #{booking.id} for #{user.username}");
    events.raiseEvent("bookingConfirmed");
    return "confirmed";
}

@End
public String cancel()
{
    return "main";
}

@Destroy @Remove (5)
public void destroy() {}
}

```

- (1) ##bean##EJB3# #####Session Bean#####
- (2) @Out ##### hotel ##### hotel  
#####
- (3) @Begin #####  
@End #####
- (4) @End #####
- (5) ##EJB#####Seam#####

```
HotelBookingAction ##### HttpSession #####
```

```
#####Session#####
#####Seam#####
#####Seam#####
```

#### 1.6.4. Seam#UI###

```
#####WAR##### WEB-INF/lib ##### seam-ui.jar#
#####Seam#JSF##### <s:link> ###
```

```
<s:link value="View Hotel" action="#{hotelBooking.selectHotel}"/>
```

```
### <s:link> #####HTML#####JSF <h:commandLink>
##### <s:link> #####
```

```
#####Seam#Ajax4JSF##### /book.xhtml
#####Seam#JSF#####
```

#### 1.6.5. Seam####

```
WAR##### seam-debug.jar#####jar### WEB-INF/lib #####Facelets#### web.xml ## seam.properties
#####Seam###
```

```
<context-param>
  <param-name>org.jboss.seam.core.init.debug</param-name>
  <param-value>true</param-value>
</context-param>
```

```
#####Seam#####Session###Seam#####Seam### ##### ht-
tp://localhost:8080/seam-booking/debug.seam ###
```

## JBoss Seam Debug Page

This page allows you to view and inspect any component in any Seam context associated with the current session.

### Conversations

conversation id	activity	description	view id	
4	1:51:34 AM - 1:51:34 AM	Search hotels: M	/main.xhtml	<a href="#">Select conversation context</a>
6	1:51:40 AM - 1:52:23 AM	Book hotel: Marriott Courtyard	/book.xhtml	<a href="#">Select conversation context</a>

#### - Component (booking)

checkinDate	Fri Jan 20 20:52:20 EST 2006
checkoutDate	Sat Jan 21 20:52:20 EST 2006
class	class org.jboss.seam.example.booking.Booking
creditCard	
description	Marriott Courtyard, Jan 20, 2006 to Jan 21, 2006
hotel	Hotel(Tower Place, Buckhead,Atlanta,30305)
id	
user	User(gavin)

#### - Conversation Context (6)

<a href="#">booking</a>
<a href="#">conversation</a>
<a href="#">hotel</a>
<a href="#">hotelBooking</a>
<a href="#">hotels</a>

#### - Business Process Context

Empty business process context

#### + Session Context

#### + Application Context

## 1.7. #####Seam#jBPM#####DVD##

DVD#####jBPM#

#####jPDL#####

# JBoss Seam DVD Store Demo

[Search for Movies](#)
[My Orders](#)

## Search Results



Add to cart	Title	Actor	Price
<input type="checkbox"/>	Life is Beautiful	Roberto Benini	\$12.00
<input type="checkbox"/>	Finding Nemo	Albert Brooks	\$22.49
<input type="checkbox"/>	March of the Penguins	Morgan Freeman	\$16.98
<input type="checkbox"/>	Indiana Jones and the Temple of Doom	Harrison Ford	\$19.99
<input type="checkbox"/>	Clear and Present Danger	Harrison Ford	\$19.99
<input type="checkbox"/>	Roman Holiday	Audrey Hepburn	\$12.99
<input type="checkbox"/>	Breakfast at Tiffany's	Audrey Hepburn	\$12.99
<input type="checkbox"/>	Sabrina	Audrey Hepburn	\$12.99
<input type="checkbox"/>	Sabrina	Harrison Ford	\$19.99
<input type="checkbox"/>	Kill Bill Vol. 1	Uma Thurman	\$19.99
<input type="checkbox"/>	Kill Bill Vol. 2	Uma Thurman	\$19.99
<input type="checkbox"/>	Lost in Translation	Bill Murray	\$19.99
<input type="checkbox"/>	Broken Flowers	Bill Murray	\$19.99
<input type="checkbox"/>	Better Off Dead	John Cusak	\$8.99
<input type="checkbox"/>	Grosse Pointe Blank	John Cusak	\$11.99
<input type="checkbox"/>	High Fidelity	John Cusak	\$14.99
<input type="checkbox"/>	Somewhere in Time	Christopher Reeve	\$11.24
<input type="checkbox"/>	Superman - The Movie	Christopher Reeve	\$14.99
<input type="checkbox"/>	Superman II	Christopher Reeve	\$14.99
<input type="checkbox"/>	Superman III	Christopher Reeve	\$14.99

[Update Shopping Cart](#)

Welcome, Harry

Thank you for choosing the DVD Store

[Logout](#)

Search for DVDs:

Title:

Actor:

Category:

Any

Results Per Page:

20

[Search](#)

Shopping Cart

1 Napoleon Dynamite

Total:\$14.06

[Checkout](#)

Done

#####jBPM#####

# JBoss Seam DVD Store Demo

**Manage Orders**

## Order Management

Pending orders are shown here on the order management screen for the store manager to process. Rather than being data-driven, order management is process-driven. A JBoss jBPM process assigns fulfillment tasks to the manager based on the version of the process loaded. The manager can change the version of the process at any time using the admin options box to the right.

- Order process 1 sends orders immediately to shipping, where the manager should ship the order and record the tracking number for the user to see.
- Order process 2 adds an approval step where the manager is first given the chance to approve the order before sending it to shipping. In each case, the status of the order is shown in the customer's order list.
- Order process 3 introduces a decision node. Only orders over \$100.00 need to be accepted. Smaller orders are automatically approved for shipping.

### Task Assignment

Order Id	Order Amount	Customer	Task	
5	\$12.99	user1	ship	<input type="button" value="Assign"/>
7	\$77.70	user2	ship	<input type="button" value="Assign"/>

### Order Acceptance

There are no orders to be accepted.

### Shipping

Order Id	Order Amount	Customer	
6	\$94.95	user1	<input type="button" value="Ship"/>

Done

**Welcome, Albus**

Thank you for choosing the DVD Store

**Statistics**

**Inventory**  
28 sold, 2473 in stock

**Sales**  
\$437.63 from 7 orders

**Admin Options**

**Process Management**

ordermanagement3

TODO

#dvdstore###

## 1.8. ##Seam#Hibernate####Hibernate####

Hibernate#####Hibernate#JavaBeans#####Bean#####

TODO

#hibernate###

## 1.9. ##RESTful#Seam#####Blog##

Seam#####  
#####  
Blog#####Seam#####RESTful#####





Blog#####“#”####"pull"-style##MVC#####

### 1.9.1. ##“#”###MVC

# index.xhtml Facelets#####blog#####

#### Example 1.25.

```
<h:dataTable value="#{blog.recentBlogEntries}" var="blogEntry" rows="3">
  <h:column>
    <div class="blogEntry">
      <h3>#{blogEntry.title}</h3>
      <div>
        <h:outputText escape="false"
          value="#{blogEntry.excerpt==null ? blogEntry.body : blogEntry.excerpt}"/>
      </div>
      <p>
        <h:outputLink value="entry.seam" rendered="#{blogEntry.excerpt!=null}">
          <f:param name="blogEntryId" value="#{blogEntry.id}"/>
          Read more...
        </h:outputLink>
      </p>
      <p>
        [Posted on
        <h:outputText value="#{blogEntry.date}">
          <f:convertDateTime timeZone="#{blog.timeZone}" locale="#{blog.locale}" type="both"/>
        </h:outputText>]
```

```

        &#160;
        <h:outputLink value="entry.seam">[Link]
            <f:param name="blogEntryId" value="#{blogEntry.id}"/>
        </h:outputLink>
    </p>
</div>
</h:column>
</h:dataTable>

```

```

##### <h:dataTable> ##### Blog ##### blog #Seam##"#"###
#####web#####Struts#####

```

### Example 1.26.

```

@Name("blog")
@Scope(ScopeType.STATELESS)
public class BlogService
{
    @In                                                                    (1)
    private EntityManager entityManager;

    @Unwrap                                                                (2)
    public Blog getBlog()
    {
        return (Blog) entityManager.createQuery("from Blog b left join fetch b.blogEntries")
            .setHint("org.hibernate.cacheable", true)
            .getSingleResult();
    }
}

```

- (1) ##### *#Seam#####seam-managed persistence context##*  
 #####Seam#####EJB3###  
 #####Web#####
- (2) @Unwrap ####Seam# Blog ### BlogService ##### ##Seam# #####manager component pattern#

```
#####
```

### 1.9.2. #####

```

Blog##### menu.xhtml #####Facelets## template.xhtml
####

```

### Example 1.27.

```

<div id="search">
    <h:form>
        <h:inputText value="#{searchAction.searchPattern}"/>
        <h:commandButton value="Search" action="/search.xhtml"/>
    </h:form>
</div>

```

```
#####
```

```
#####JSF##id#####Seam#####id#####
```

### Example 1.28.

```
<navigation-rule>
  <navigation-case>
    <from-outcome>searchResults</from-outcome>
    <to-view-id>/search.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
```

```
#####
```

### Example 1.29.

```
<div id="search">
  <h:form>
    <h:inputText value="#{searchAction.searchPattern}"/>
    <h:commandButton value="Search" action="searchResults"/>
  </h:form>
</div>
```

```
#####URL#####
```

```
ht-
```

```
tp://localhost:8080/seam-blog/search.seam?searchPattern=seam# JSF#####Seam#####
WEB-INF/pages.xml #####
```

### Example 1.30.

```
<pages>
  <page view-id="/search.xhtml">
    <param name="searchPattern" value="#{searchService.searchPattern}"/>
  </page>
  ...
</pages>
```

```
###Seam#####      #{searchService.searchPattern}      #####      searchPattern
#####
```

```
##### search.xhtml ###
```

### Example 1.31.

```
<h:dataTable value="#{searchResults}" var="blogEntry">
  <h:column>
    <div>
      <h:outputLink value="entry.seam">
        <f:param name="blogEntryId" value="#{blogEntry.id}"/>
        #{blogEntry.title}
      </h:outputLink>
      posted on
      <h:outputText value="#{blogEntry.date}">
        <f:convertDateTime timeZone="#{blog.timeZone}" locale="#{blog.locale}" type="both"/>
      </h:outputText>
    </div>
  </h:column>
</h:dataTable>
```

```

    </div>
  </h:column>
</h:dataTable>

```

#####“#”###MVC#####

### Example 1.32.

```

@Name("searchService")
public class SearchService
{
    @In
    private EntityManager entityManager;

    private String searchPattern;

    @Factory("searchResults")
    public List<BlogEntry> getSearchResults()
    {
        if (searchPattern==null)
        {
            return null;
        }
        else
        {
            return entityManager.createQuery("select be from BlogEntry be where lower(be.title) like :searchPattern")
                .setParameter( "searchPattern", getSqlSearchPattern() )
                .setMaxResults(100)
                .getResultList();
        }
    }

    private String getSqlSearchPattern()
    {
        return searchPattern==null ? "" : '%' + searchPattern.toLowerCase().replace('*', '%').replace(' ', '%') + '%';
    }

    public String getSearchPattern()
    {
        return searchPattern;
    }

    public void setSearchPattern(String searchPattern)
    {
        this.searchPattern = searchPattern;
    }
}

```

### 1.9.3. #RESTful#####“#”###"push"-style##MVC

#####“#”###MVC###RESTful#####Seam### ##### Blog##### entry.xhtml  
#####“#”###MVC#####

entryAction #####“#”###MVC#####Struts#####action class##

### Example 1.33.

```

@Name("entryAction")

```

```

@Scope(STATELESS)
public class EntryAction
{
    @In(create=true)
    private Blog blog;

    @Out
    private BlogEntry blogEntry;

    public void loadBlogEntry(String id) throws EntryNotFoundException
    {
        blogEntry = blog.getBlogEntry(id);
        if (blogEntry==null) throw new EntryNotFoundException(id);
    }
}

```

# pages.xml #####

### Example 1.34.

```

<pages>
    ...

    <page view-id="/entry.xhtml" action="#{entryAction.loadBlogEntry(blogEntry.id)}">
        <param name="blogEntryId" value="#{blogEntry.id}"/>
    </page>

    <page view-id="/post.xhtml" action="#{loginAction.challenge}"/>

    <page view-id="*" action="#{blog.hitCount.hit}"/>

</pages>

```

##### — ##### #####JSF  
EL##Seam#####JSF#####

# entry.xhtml #####Seam##### blogEntryId##### — blogEntry —  
#####Seam#####

### Example 1.35.

```

<div class="blogEntry">
    <h3>#{blogEntry.title}</h3>
    <div>
        <h:outputText escape="false" value="#{blogEntry.body}"/>
    </div>
    <p>
        [Posted on#{160;
        <h:outputText value="#{blogEntry.date}">
            <f:convertDateTime timezone="#{blog.timeZone}" locale="#{blog.locale}" type="both"/>
        </h:outputText>]
    </p>
</div>

```

#####blog entry##### EntryNotFoundException ### #####404#####505#####

**Example 1.36.**

```
@ApplicationException(rollback=true)
@HttpError(errorCode=HttpServletResponse.SC_NOT_FOUND)
public class EntryNotFoundException extends Exception
{
    EntryNotFoundException(String id)
    {
        super("entry not found: " + id);
    }
}
```

#####

**Example 1.37.**

```
@Name("entryAction")
@Scope(SCOPELESS)
public class EntryAction
{
    @In(create=true)
    private Blog blog;

    @In @Out
    private BlogEntry blogEntry;

    public void loadBlogEntry() throws EntryNotFoundException
    {
        blogEntry = blog.getBlogEntry( blogEntry.getId() );
        if (blogEntry==null) throw new EntryNotFoundException(id);
    }
}
```

```
<pages>
...

<page view-id="/entry.xhtml" action="#{entryAction.loadBlogEntry}">
    <param name="blogEntryId" value="#{blogEntry.id}"/>
</page>

...
</pages>
```

#####

---

## Chapter 2. #Seam-gen##

```
Seam#####Eclipse#####Seam#####

#####Seam#####Ruby#####

#####seam-gen#####JBoss AS#####seam-gen#####J2EE#Java EE 5#####

#####Eclipse###seam-gen#####Eclipse#####
#####Eclipse#####

Seam-gen#Ant###Hibernate#####
```

### 2.1. ####

```
#####JDK 5##JDK6#JBoss AS 4.2#Ant 1.6#####Eclipse#JBoss IDE#TestNG#Eclipse ###
#Eclipse#JBoss Server View##JBoss#####debug####JBoss#####Seam####

JBoss#####WAR#EAR#####JVM#####Bug#####EAR#####PermGen
###Permanent Generation Space## #####perm gen#####JBoss
IDE###JBoss#####VM arguments#####
```

```
-Xms512m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=512
```

```
#####
```

```
-Xms256m -Xmx512m -XX:PermSize=128m -XX:MaxPermSize=256
```

```
#####JBoss##### bin/run.conf #####JVM#####
```

```
##### OutOfMemoryException #####
```

### 2.2. #####Eclipse##

```
#####seam-gen#####JBoss AS#####Eclipse workspace#####
```

```
cd jboss-seam-2.0.x
seam setup
```

```
#####:
```

```
C:\Projects\jboss-seam>seam setup
Buildfile: build.xml

setup:
[echo] Welcome to seam-gen :-)
[input] Enter your Java project workspace [C:/Projects]

[input] Enter your JBoss home directory [C:/Program Files/jboss-4.2.0.GA]

[input] Enter the project name [myproject]
helloworld
[input] Is this project deployed as an EAR (with EJB components) or a WAR (with no EJB support) [
[input] Enter the Java package name for your session beans [com.mydomain.helloworld]
```

```

org.jboss.helloworld
[input] Enter the Java package name for your entity beans [org.jboss.helloworld]

[input] Enter the Java package name for your test cases [org.jboss.helloworld.test]

[input] What kind of database are you using? [hsql] (hsql,mysql,oracle,postgres,mssql,db2,sybase,mysql
mysql
[input] Enter the Hibernate dialect for your database [org.hibernate.dialect.MySQLDialect]

[input] Enter the filesystem path to the JDBC driver jar [lib/hsqldb.jar]
../../mysql-connector.jar
[input] Enter JDBC driver class for your database [com.mysql.jdbc.Driver]

[input] Enter the JDBC URL for your database [jdbc:mysql:///test]

[input] Enter database username [sa]
gavin
[input] Enter database password []

[input] skipping input as property hibernate.default_schema.new has already been set.
[input] Enter the database catalog name (it is OK to leave this blank) []

[input] Are you working with tables that already exist in the database? [n] (y,n,)
Y
[input] Do you want to drop and recreate the database tables and data in import.sql each time you
n
[propertyfile] Creating new property file: C:\Projects\jboss-seam\seam-gen\build.properties
[echo] Installing JDBC driver jar to JBoss server
[echo] Type 'seam new-project' to create the new project

BUILD SUCCESSFUL
Total time: 1 minute 17 seconds
C:\Projects\jboss-seam>

```

#####Enter##

```

#####EAR####WAR#####EAR####EJB 3.0 ###Java EE 5##WAR####EJB 3.0####J2EE#####
##WAR#####EJB3                                profile#####ear#####war#
#####EAR#####WAR###

```

#####

##### seam-gen/build.properties ##### seam setup #####

#####Eclipse workspace#####

```

seam new-project

```

```

C:\Projects\jboss-seam>seam new-project
Buildfile: build.xml

validate-workspace:

validate-project:

copy-lib:
[echo] Copying project jars ...
[copy] Copying 58 files to C:\Projects\helloworld\lib
[copy] Copying 9 files to C:\Projects\helloworld\embedded-ejb

file-copy-war:

file-copy-ear:
[echo] Copying resources needed for EAR deployment to the C:\Projects\helloworld/resources direct

new-project:
[echo] A new Seam project named 'helloworld' was created in the C:\Projects directory

```



```
[echo] Type 'seam explode' and go to http://localhost:8080/helloworld
[echo] Eclipse Users: Add the project into Eclipse using File > New > Project and select General
[echo] NetBeans Users: Open the project in NetBeans
```

```
BUILD SUCCESSFUL
Total time: 7 seconds
C:\Projects\jboss-seam>
```

#####Seam

```
jar#####jar###JDBC###jar#####Eclipse#####Eclipse####Ant#####
##### New -> Project... -> General -> Project -> Next###Project name #####helloworld#####
Finish#####Eclipse#####JBoss AS#####exploded##### Java Project#
```

```
#Eclipse#####JDK##Java SE 5 #Java SE 6##### Project -> Properties -> Java Compiler #####Java SE
5 ###JDK#
```

```
#####Eclipse#### seam explode #####
```

```
# http://localhost:8080/helloworld ##### view/home.xhtml ##### view/layout/template.xhtml
#####Facelets #####Eclipse#####
```

```
#####XML#####Java EE#####
####Seam####90%#####seam-gen#####
```

```
#####jboss-beans.xml#persistence-test.xml # import-test.sql
#####TestNG#HSQLDB##### # import-test.sql #####Schema##### myproject-
dev-ds.xml#persistence-dev.xml # import-dev.sql #####
###schema#####seam-gen##### myproject-prod-ds.xml#persistence-
prod.xml # import-prod.sql #####schema#####
```

## 2.3. #####

```
#####action-style Web#####Java#####action#####Web#####
```

```
seam new-action
```

```
#Seam#####Facelets###Seam###
```

```
C:\Projects\jboss-seam>seam new-action
Buildfile: build.xml

validate-workspace:

validate-project:

action-input:
    [input] Enter the Seam component name
ping
    [input] Enter the local interface name [Ping]

    [input] Enter the bean class name [PingBean]

    [input] Enter the action method name [ping]

    [input] Enter the page name [ping]

setup-filters:

new-action:
```

```
[echo] Creating a new stateless session bean component with an action method
[copy] Copying 1 file to C:\Projects\helloworld\src\action\org\jboss\helloworld
[copy] Copying 1 file to C:\Projects\helloworld\src\action\org\jboss\helloworld
[copy] Copying 1 file to C:\Projects\helloworld\src\action\org\jboss\helloworld\test
[copy] Copying 1 file to C:\Projects\helloworld\src\action\org\jboss\helloworld\test
[copy] Copying 1 file to C:\Projects\helloworld\view
[echo] Type 'seam restart' and go to http://localhost:8080/helloworld/ping.seam
```

```
BUILD SUCCESSFUL
Total time: 13 seconds
C:\Projects\jboss-seam>
```

```
##Seam#####exploded    directory    deployment##  ##seam    restart###Eclipse#####
build.xml ### restart target##### Eclipse##### resources/META-INF/application.xml #####
#####JBoss#
```

```
#####    http://localhost:8080/helloworld/ping.seam    #####    ####    src
#####    ping() #####
```

```
##### PingTest.xml #####Eclipse#TestNG#####    seam    test    #####build#### test
target#####
```

## 2.4. #####form#

```
#####
```

```
seam new-form
```

```
C:\Projects\jboss-seam>seam new-form
Buildfile: C:\Projects\jboss-seam\seam-gen\build.xml

validate-workspace:

validate-project:

action-input:
    [input] Enter the Seam component name
hello
    [input] Enter the local interface name [Hello]

    [input] Enter the bean class name [HelloBean]

    [input] Enter the action method name [hello]

    [input] Enter the page name [hello]

setup-filters:

new-form:
    [echo] Creating a new stateful session bean component with an action method
    [copy] Copying 1 file to C:\Projects\hello\src\com\hello
    [copy] Copying 1 file to C:\Projects\hello\src\com\hello
    [copy] Copying 1 file to C:\Projects\hello\src\com\hello\test
    [copy] Copying 1 file to C:\Projects\hello\view
    [copy] Copying 1 file to C:\Projects\hello\src\com\hello\test
    [echo] Type 'seam restart' and go to http://localhost:8080/hello/hello.seam

BUILD SUCCESSFUL
Total time: 5 seconds
C:\Projects\jboss-seam>
```

```
##### http://localhost:8080/helloworld/hello.seam #####
#####Seam#####Java#####
```

## 2.5. #####

```
##### seam setup #####
```

```
seam generate-entities
```

```
##### http://localhost:8080/helloworld #####
#####
#####Seam-gen#####Seam#####
```

## 2.6. #####EAR

```
#####Java EE##### seam unexplode #####exploded directory)# ##### seam
deploy #####Build##### deploy target####EAR##### seam undeploy ##### undeploy #####EAR#
```

```
##### dev profile #####EAR###persistence-dev.xml#import-dev.sql###myproject-dev-ds.xml
##### #####profile, ##### prod profile#
```

```
seam -Dprofile=prod deploy
```

```
#####profile##### ###persistence-staging.xml#import-staging.sql #
myproject-staging-ds.xml — ##### -Dprofile=staging ###profile#
```

## 2.7. Seam#####

```
#Seam#####exploded##### ##### components.xml
#####Seam#Facelet##debug####:
```

```
<core:init debug="true"/>
```

```
#####web####
```

- ##Facelet##
- ## pages.xml ##

```
###Java##### ##JBoss####EAR#####touch#####EAR#####applica-
tion.xml####WAR#####web.xml ##
```

```
#####/##/#####Seam###JavaBean##### #####JavaBean##### WEB-INF/dev
#####Seam#####WAR#EAR#####
```

```
#####
```

- ###JavaBean#####EJB3 Beans#####

- ##Bean####
- ## components.xml #####
- # WEB-INF/dev #####
- ###Seam#debug##

####Seam-gen##WAR#####src/action#####EAR####

## 2.8. #Jboss 4.0###Seam

Seam 2.0###JavaServer Faces 1.2#####JBoss 4.2###Seam#####JSF 1.2##### Jboss 4.0###Seam#####EJB3#Jboss 4.0#####MyFaces###JSF1.2##### Seam2.0####JBoss 4.0#####

### 2.8.1. ##JBoss 4.0

JBoss 4.0####Seam#####Seam#####JEMS 1.2#####EJB3 profile#  
####EJB3##Seam#####JEMS##### <http://labs.jboss.com/jemsinstaller/downloads#>

### 2.8.2. ##JSF 1.2 RI

JBoss 4.0##### server/default/deploy/jbossweb-tomcat55.sar ### ##### jsf-libs ##### myfaces-api.jar # myfaces-impl.jar### ##### jsf-api.jar#jsf-impl.jar#el-api.jar # el-ri.jar  
#####Seam#lib#####JAR###EL JAR#####Seam 1.2#####

##### conf/web.xml ##### myfaces-impl.jar ### jsf-impl.jar#

---

## Chapter 3. #####

```
Seam##### context#####
#####
#####Se
am#####
component#####EJB#####Bijection#####

#####Seam#####
```

### 3.1. Seam###

```
Seam#####Java
API#####annotation#####

###Seam###:
```

- Stateless context
- Event #or request# context
- Page context
- Conversation context
- Session context
- Business process context
- Application context

```
#####servlet#####conversation context##### business process con-
text##### #Web#####request, session
#application##### session#####
#####Seam#####

#####context#####
```

#### 3.1.1. Stateless context#####

```
#####Session Bean#####
#####
```

#### 3.1.2. Event context#####

```
#####“##”#####Web Request #####
####JSF#####
#####

####RMI##Seam Remoting##Seam#####
```

### 3.1.3. Page context#####

#####Event Listener#####  
#####

### 3.1.4. Conversation context#####

#####Seam#####conversation#####  
#####Servlet#####“#####”#“#####”#“#####”#####  
#####“use case#####”#“user story#####”#####  
  
#####“#####”#####  
#####Bug#  
  
#####  
  
#####Seam###annotation#####  
  
#####tasks#####Seam#####  
  
#####nested#####“#”#####  
  
#####Seam###Servlet                      Session                      #####Seam#####                      conversation  
timeout#####Session#####  
  
#####Seam#####  
  
#####Seam#####

### 3.1.5. Session context#Session####

Session#####session#####Session  
#####  
  
#JSR-168 Portal####Session####Portlet####

### 3.1.6. Business process context #####

#####BPM###jBPM#####  
##### process                      definition                      language#####  
#####annotation#####

### 3.1.7. Application context#####

Application#####Servlet####Servlet#####  
###Seam#####

### 3.1.8. Context variables#####

##### context      variables#####                      #####Servlet####Session#Request      attributes####  
#####Context Variable#####Seam#####Context Variables#

##### Contexts  
##### Context #####

```
User user = (User) Contexts.getSessionContext().get("user");
```

#####

```
Contexts.getSessionContext().set("user", user);
```

#####injection#####outjection#####

### 3.1.9. Context####

##### priority order(#####) #####

- Event context
- Page context
- Conversation context
- Session context
- Business process context
- Application context

##### Contexts.lookupInStatefulContexts() #####JSF#####

### 3.1.10. ####

Servlet#EJB##### Servlet#####  
EJB#####Session Bean,#####

###Web#####OK#

#####AJAX##### Seam#####

Seam Session #####Seam#####Event#####Page##### business  
process##### Seam#Conversation Context### #####  
#####

##Session#####Session#####Seam##### Seam#####Session###Session  
Bean#JavaBean##### #Application Scoped#####Application  
Scoped##### ## #####Session  
Bean#JavaBean##### @Synchronized ###

#####AJAX#####Session#####

## 3.2. Seam ##

Seam###POJO#Plain Old Java Objects#####JavaBean##EJB 3.0 enterprise  
bean#Seam#####EJB#####EJB 3.0#####Seam#####EJB 3.0#####EJB 3.0#####

- EJB 3.0 stateless Session Beans
- EJB 3.0 stateful Session Beans
- EJB 3.0 entity beans
- JavaBeans
- EJB 3.0 message-driven beans

### 3.2.1. ###Session Bean

```
###Session Bean#####Seam#####JSF#action listener#####JSF#####
```

```
#####session bean#####EJB3#####
```

```
###Session Bean#####
```

```
###Session Bean#Seam#####
```

```
Seam###Session Bean##### Component.getInstance() ## @In(create=true) #####JNDI## new #####
```

### 3.2.2. ###Session Bean

```
###Session Bean#####bean##### Session Bean#####Seam###web##### HttpSession #####Seam#####Session Bean#####Seam#####
```

```
###Session Bean#####JSF action listener#####JSF###form###backing bean(##bean,###bean)#####
```

```
#####Session Bean#####Conversation Context#####page#stateless context#
```

```
#Session#####Session Bean#####Seam#####
```

```
Seam###Session Bean##### Component.getInstance() ## @In(create=true) #####JNDI## new #####
```

### 3.2.3. ##Bean

```
##Bean#####Seam#####Entity#####Entity#####Java Code#####Seam#####
```

```
Entity Bean#####Entity Bean#####
```

```
Entity Bean#####JSF#action listener#####JSF#####form#####bean,##### Entity#####Bean#####Session Bean###action listener#####CRUD#####
```

```
#####Entity Bean#####Conversation Context#####Context#
```



```
#####Entity Bean#####Conversation##Session###Seam#####Session Bean#####Entity
Bean#####Seam#####Entity Bean###Seam###

Seam##Bean##### Component.getInstance()#@In(create=true) ##### new #####
```

### 3.2.4. JavaBeans

```
JavaBeans#####Session Bean#####Session
Bean#####EJB 3.0#####

#####EJB#####Seam#Hibernate#####JavaBeans###Session Beans#
#####Conversation#Session ###Seam JavaBean#####Session Bean#####

###JavaBeans###Event Context##

#Session###JavaBeans#####Seam#####

Seam JavaBean##### Component.getInstance()#@In(create=true) ##### new #####
```

### 3.2.5. #####Bean

```
####Bean#####Seam#####Bean###Seam#####Context#####JMS
Queue##Topic####

####Bean#####Seam#####“###”#Session#####Seam###

####Bean#####EJB#####
```

### 3.2.6. ##

```
####Seam##### #JavaBean###Seam#####Entity
Bean##### #Session Bean#####EJB#####
```

```
@Stateless
@Interceptors(SeamInterceptor.class)
public class LoginAction implements Login {
    ...
}
```

```
##### ejb-jar.xml #####
```

```
<interceptors>
  <interceptor>
    <interceptor-class>org.jboss.seam.ejb.SeamInterceptor</interceptor-class>
  </interceptor>
</interceptors>

<assembly-descriptor>
  <interceptor-binding>
    <ejb-name>*/ejb-name>
    <interceptor-class>org.jboss.seam.ejb.SeamInterceptor</interceptor-class>
  </interceptor-binding>
</assembly-descriptor>
```

### 3.2.7. #####

##Seam##### @Name #####

```
@Name("loginAction")
@Stateless
public class LoginAction implements Login {
    ...
}
```

##### seam component name##EJB##### ##Seam#####JSF##Bean  
Name#####

@Name ##### #### ##Seam #####

#####JSF##Seam##### ##### Con-  
texts.getStatelessContext().get("loginAction") ### LoginAction#  
#####Seam#####JSF#####API#####  
#####User#####Session##### currentUser ##### user ###

#####Seam#####

```
@Name("com.jboss.myapp.loginAction")
@Stateless
@Interceptors(SeamInterceptor.class)
public class LoginAction implements Login {
    ...
}
```

#####Java##JSF#####

```
<h:commandButton type="submit" value="Login"
    action="#{com.jboss.myapp.loginAction.login}"/>
```

#####Seam##### components.xml #####

```
<factory name="loginAction" scope="STATELESS" value="#{com.jboss.myapp.loginAction}"/>
```

##Seam#####Seam jar## components.xml #####

### 3.2.8. #####Defining the Component Scope#

##### @Scope #####Seam#####

```
@Name("user")
@Entity
@Scope(SESSION)
public class User {
    ...
}
```

org.jboss.seam.ScopeType#####.

### 3.2.9. #####Components with multiple roles#

##Seam##### User #####Session-  
Scoped#####Conversation-Scoped###@Role #####  
#####Seam## ##

#####@Role#####

```
@Name("user")
@Entity
@Scope(CONVERSATION)
@Role(name="currentUser", scope=SESSION)
public class User {
    ...
}
```

@Roles #####

```
@Name("user")
@Entity
@Scope(CONVERSATION)
@Roles({@Role(name="currentUser", scope=SESSION),
        @Role(name="tempUser", scope=EVENT)})
public class User {
    ...
}
```

### 3.2.10. ###

#####Seam#####Seam#####Seam#####  
 #####Seam##### org.jboss.seam.core ##Seam #####Java#####

###Seam#####instance()####

```
FacesMessages.instance().add("Welcome back, #{user.name}!");
```

### 3.3. ####

Dependency injection##### # inversion of control#####Java  
 #####"##"#####setter#####"##"#####(reference)#####  
 #####  
 ##Seam#####Seam### bijection#####  
 #####injection#####bijection##

- contextual##### - #####
- bidirectional##### - #####outjected#####  
 ####,#####
- dynamic##### - #####Seam#####

#####

@In #####

```
@Name("loginAction")
@Stateless
public class LoginAction implements Login {
    @In User user;
    ...
}
```

####setter###

```

@Name("loginAction")
@Stateless
public class LoginAction implements Login {
    User user;

    @In
    public void setUser(User user) {
        this.user=user;
    }

    ...
}

```

##### Seam##### @In("currentUser")#

#####Seam##### @In(create=true)# #####null####  
@In(required=false)#

##### @In(create=true) ##### @AutoCreate##### create=true#

#####

```

@Name("loginAction")
@Stateless
public class LoginAction implements Login {
    @In("#{user.username}") String username;
    ...
}

```

#####

@Out #####

```

@Name("loginAction")
@Stateless
public class LoginAction implements Login {
    @Out User user;
    ...
}

```

#####getter###

```

@Name("loginAction")
@Stateless
@Interceptors(SeamInterceptor.class)
public class LoginAction implements Login {
    User user;

    @Out
    public User getUser() {
        return user;
    }

    ...
}

```

#####

```

@Name("loginAction")
@Stateless
public class LoginAction implements Login {
    @In @Out User user;
    ...
}

```

}

###

```

@Name("loginAction")
@Stateless
public class LoginAction implements Login {
    User user;

    @In
    public void setUser(User user) {
        this.user=user;
    }

    @Out
    public User getUser() {
        return user;
    }

    ...
}

```

### 3.4. Lifecycle methods#####

Session Bean###Bean Seam#####EJB3.0#####@PostConstruct, @PreDestroy,  
 #####Seam#####JavaBean#####J2EE#####Seam#####  
 @PostConstruct # @PreDestroy.

@Create ###Seam##### @Create ###

@Destroy ###Seam##### @Destroy ###

#####Session Bean## ## ##### @Remove #####Seam###

##### @Startup#####Application##Session##### @Startup  
 #####Seam##### ##startup##### @Startup(depends={...}) ###

### 3.5. #####Conditional installation#

@Install #####

- #####mock out#####
- #####
- #####

@Install ##### precedence##### # dependencies####

#####classpath#####seam#####  
 Seam#####

1. BUILT\_IN — #####,####Seam####
2. FRAMEWORK — #####

3. APPLICATION — #####

4. DEPLOYMENT — #####

5. MOCK — #####mock objects####

##### messageSender #####JMS#####

```
@Name("messageSender")
public class MessageSender {
    public void sendMessage() {
        //do something with JMS
    }
}
```

#####JMS#####stub##### mock #####classpath#####

```
@Name("messageSender")
@Install(precedence=MOCK)
public class MockMessageSender extends MessageSender {
    public void sendMessage() {
        //do nothing!
    }
}
```

#seam#classpath#####precedence ##Seam#####

##### classpath##### #####jar#####

#####classpath#####

@Install

#####Seam#####

## 3.6. ##

#####

```
private static final Log log = LoggerFactory.getLog(CreateOrderAction.class);

public Order createOrder(User user, Product product, int quantity) {
    if ( log.isDebugEnabled() ) {
        log.debug("Creating new order for user: " + user.username() +
            " product: " + product.name()
            + " quantity: " + quantity);
    }
    return new Order(user, product, quantity);
}
```

#####log#####log#####Java##10#####

Seam#####logging API#

```
@Logger private Log log;

public Order createOrder(User user, Product product, int quantity) {
    log.debug("Creating new order for user: #0 product: #1 quantity: #2", user.username(), product.name(), quantity);
    return new Order(user, product, quantity);
}
```

### log #####static#####Entity Bean##### log #####

```
##### if ( log.isDebugEnabled() ) ##### debug() ## ##
#####log#####Seam##### Log#

## User # Product #####Seam#####
```

```
@Logger private Log log;

public Order createOrder(User user, Product product, int quantity) {
    log.debug("Creating new order for user: #{user.username} product: #{product.name} quantity: #0", o
    return new Order(user, product, quantity);
}
```

Seam logging#####log4j###JDK logging###log4j#classpath##Seam#####Seam###JDK logging.

### 3.7. Mutable###@ReadOnly

```
##### HttpSession #####Session##### setAttribute() #####
##Bug#####Bug#####
#####Session#####
```

```
###EJB ###Session Bean#####dirty checking#####EJB #####
#####Seam#####EJB 3.0# #####Session#Conversation####JavaBean#Entity
Bean####Web###Session#####Seam#####
```

```
##Session#Conversation###JavaBean#####Seam##### setAttribute()
##### org.jboss.seam.core.Mutable #####
org.jboss.seam.core.AbstractMutable#####dirty-checking#####
```

```
@Name("account")
public class Account extends AbstractMutable
{
    private BigDecimal balance;

    public void setBalance(BigDecimal balance)
    {
        setDirty(this.balance, balance);
        this.balance = balance;
    }

    public BigDecimal getBalance()
    {
        return balance;
    }

    ...
}
```

```
##### @ReadOnly #####
```

```
@Name("account")
public class Account
{
    private BigDecimal balance;

    public void setBalance(BigDecimal balance)
    {
        this.balance = balance;
    }

    @ReadOnly
```

```

    public BigDecimal getBalance()
    {
        return balance;
    }

    ...

}

```

**#Session#Conversation###Entity**                      **Bean#####Seam#####**                      **setAttribute()**  
**#####Seam#####Session#Conversation###Entity**  
**Bean#####Session Bean##JavaBean###“##”Entity Bean#####**

```

@Stateful
@Name("account")
public class AccountManager extends AbstractMutable
{
    private Account account; // an entity bean

    @Unwrap
    public void getAccount()
    {
        return account;
    }

    ...

}

```

**##,## EntityHome ##Seam#####Seam#####Bean##.**

### 3.8. Factory#Manager##

**#####Seam##### @In ##### #####Seam**  
**#####@Destroy##**  
**##Seam#####Seam#####Seam#####**

*factory component pattern#####Seam##### factory meth-*  
*od##### @Factory#####*  
**#####Seam#####**

```

@Factory(scope=CONVERSATION)
public List<Customer> getCustomerList() {
    return ... ;
}

```

**##### void#####**

```

@DataModel List<Customer> customerList;

@Factory("customerList")
public void initCustomerList() {
    customerList = ... ;
}

```

**##### customerList #####null##### manager com-**  
**ponent pattern##### Seam#####**

**##### @Unwrap ### #####**



```
@Name("customerList")
@Scope(CONVERSATION)
public class CustomerListManager
{
    ...

    @Unwrap
    public List<Customer> getCustomerList() {
        return ... ;
    }
}
```

#####

#####@Unwrap##### @Destroy #####

```
@Name("hens")
@Scope(APPLICATION)
public class HenHouse {

    Set<Hen> hens;

    @In(required=false) Hen hen;

    @Unwrap
    public List<Hen> getHens() {
        if (hens == null) {
            // Setup our hens
        }
        return hens;
    }

    @Observer({"chickBorn", "chickenBoughtAtMarket"})
    public addHen() {
        hens.add(hen);
    }

    @Observer("chickenSoldAtMarket")
    public removeHen() {
        hens.remove(hen);
    }

    @Observer("foxGetsIn")
    public removeAllHens() {
        hens.clear();
    }
    ...
}
```

#####

## Chapter 4. ##Seam##

```
Seam#####XML#####XML###Seam#####
#Java#####Seam##### Seam#####properties#### web.xml
##### components.xml #####
```

### 4.1. #####

```
Seam#####servlet context#####classpath## seam.properties #####

####Seam#####JavaBean#####setter##### com.jboss.myapp.settings #Seam#####
setLocale() #setter##### seam.properties ##### com.jboss.myapp.settings.locale
#####servlet context### #####Seam##### locale #####

Seam##### web.xml ## seam.properties ##
org.jboss.seam.core.manager.conversationTimeout ##### ##Seam#####
org.jboss.seam.core.manager ##### setConversationTimeout() #setter####
```

### 4.2. ## components.xml #####

```
components.xml #####

• ##### @Name ### ##Seam#####

• ##### @Name #####Seam## — #####Seam#####

• ##### # ## @Name #####Seam### ## @Install #####

• #####

components.xml #####

• war ## WEB-INF #####

• jar ## META-INF #####

• ##### @Name ##### jar #####

#####Seam##### seam.properties ##### META-INF/components.xml ##### @Name #####
Seam##### @Install ##### components.xml #####

##### components.xml #####jBPM:
```

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:bpm="http://jboss.com/products/seam/bpm">
  <bpm:jbpm/>
</components>
```

#####

```
<components>
```

```
<component class="org.jboss.seam.bpm.Jbpm" />
</components>
```

#####Seam#####:

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:persistence="http://jboss.com/products/seam/persistence"

  <persistence:managed-persistence-context name="customerDatabase"
    persistence-unit-jndi-name="java:/customerEntityManagerFactory" />

  <persistence:managed-persistence-context name="accountingDatabase"
    persistence-unit-jndi-name="java:/accountingEntityManagerFactory" />

</components>
```

#####:

```
<components>
  <component name="customerDatabase"
    class="org.jboss.seam.persistence.ManagedPersistenceContext">
    <property name="persistenceUnitJndiName">java:/customerEntityManagerFactory</property>
  </component>

  <component name="accountingDatabase"
    class="org.jboss.seam.persistence.ManagedPersistenceContext">
    <property name="persistenceUnitJndiName">java:/accountingEntityManagerFactory</property>
  </component>
</components>
```

#####Seam###session#####

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:persistence="http://jboss.com/products/seam/persistence"

  <persistence:managed-persistence-context name="productDatabase"
    scope="session"
    persistence-unit-jndi-name="java:/productEntityManagerFactory" />

</components>
```

```
<components>

  <component name="productDatabase"
    scope="session"
    class="org.jboss.seam.persistence.ManagedPersistenceContext">
    <property name="persistenceUnitJndiName">java:/productEntityManagerFactory</property>
  </component>

</components>
```

##### auto-create ### ##### @In ##### create=true#

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:persistence="http://jboss.com/products/seam/persistence"

  <persistence:managed-persistence-context name="productDatabase"
    auto-create="true"
    persistence-unit-jndi-name="java:/productEntityManagerFactory" />

</components>
```

```
<components>

  <component name="productDatabase"
```

```

        auto-create="true"
        class="org.jboss.seam.persistence.ManagedPersistenceContext">
        <property name="persistenceUnitJndiName">java:/productEntityManagerFactory</property>
    </component>
</components>

```

<factory> #####context####

```

<components>

    <factory name="contact" method="#{contactManager.loadContact}" scope="CONVERSATION" />

</components>

```

#####Seam#####

```

<components>

    <factory name="user" value="#{actor}" scope="STATELESS" />

</components>

```

#####

```

<components>

    <factory name="contact" value="#{contactManager.contact}" scope="STATELESS" />

</components>

```

auto-create="true" ## <factory> #####

```

<components>

    <factory name="session" value="#{entityManager.delegate}" scope="STATELESS" auto-create="true" />

</components>

```

##### components.xml### Seam#### components.xml ##### @wildcard@  
 ##### Ant#####classpath##### components.properties #####  
 ###Seam#####

## 4.3. #####

#####XML##### components.xml ##### Seam#####  
 com.helloworld.Hello ##### com/helloworld/Hello.component.xml##  
 #####Hibernate##### <components> ## <component> #

#####

```

<components>
    <component class="com.helloworld.Hello" name="hello">
        <property name="name">#{user.name}</property>
    </component>
    <factory name="message" value="#{hello.message}" />
</components>

```

#####

```
<component name="hello">
  <property name="name">#{user.name}</property>
</component>
```

#####

##### com/helloworld/components.xml # com.helloworld ###

## 4.4. #####

String#####

```
org.jboss.seam.core.manager.conversationTimeout 60000
```

```
<core:manager conversation-timeout="60000"/>
```

```
<component name="org.jboss.seam.core.manager">
  <property name="conversationTimeout">60000</property>
</component>
```

####String#####Set#List#

```
org.jboss.seam.bpm.jbpm.processDefinitions order.jpdl.xml, return.jpdl.xml, inventory.jpdl.xml
```

```
<bpm:jbpm>
  <bpm:process-definitions>
    <value>order.jpdl.xml</value>
    <value>return.jpdl.xml</value>
    <value>inventory.jpdl.xml</value>
  </bpm:process-definitions>
</bpm:jbpm>
```

```
<component name="org.jboss.seam.bpm.jbpm">
  <property name="processDefinitions">
    <value>order.jpdl.xml</value>
    <value>return.jpdl.xml</value>
    <value>inventory.jpdl.xml</value>
  </property>
</component>
```

#####String####String#####Map#

```
<component name="issueEditor">
  <property name="issueStatuses">
    <key>open</key> <value>open issue</value>
    <key>resolved</key> <value>issue resolved by developer</value>
    <key>closed</key> <value>resolution accepted by user</value>
  </property>
</component>
```

##### @In #####

#####IoC####JSF##Spring#####

```
<drools:managed-working-memory name="policyPricingWorkingMemory" rule-base="#{policyPricingRules}"/>
```

```
<component name="policyPricingWorkingMemory"
  class="org.jboss.seam.drools.ManagedWorkingMemory">
  <property name="ruleBase">#{policyPricingRules}</property>
```

```
</component>
```

## 4.5. ##XML####

#####XML##### components.xml #####Seam  
Components DTD#

```
<?xml version="1.0" encoding="UTF-8"?>
<components xmlns="http://jboss.com/products/seam/components"
             xsi:schemaLocation="http://jboss.com/products/seam/components http://jboss.com/products/seam/components-dtd.xml"
             >

    <component class="org.jboss.seam.core.init">
        <property name="debug">true</property>
        <property name="jndiPattern">@jndiPattern@</property>
    </component>

</components>
```

#####

#####

```
<?xml version="1.0" encoding="UTF-8"?>
<components xmlns="http://jboss.com/products/seam/components"
             xmlns:core="http://jboss.com/products/seam/core"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="
                http://jboss.com/products/seam/core http://jboss.com/products/seam/core-2.0.xsd
                http://jboss.com/products/seam/components http://jboss.com/products/seam/components-dtd.xml"
             >

    <core:init debug="true" jndi-pattern="@jndiPattern@" />

</components>
```

##Schema#####XML##### Schema#####XML#####  
##### components.xml #####

#####Seam#####? #####Seam##### <component>  
#####Seam#####

##Java##### @Namespace #####XML##### ##### package-info.java  
##### ##seampay####:

```
@Namespace(value="http://jboss.com/products/seam/examples/seampay")
package org.jboss.seam.example.seampay;

import org.jboss.seam.annotations.Namespace;
```

##### components.xml #####

```
<components xmlns="http://jboss.com/products/seam/components"
             xmlns:pay="http://jboss.com/products/seam/examples/seampay"
             ... >

    <pay:payment-home new-instance="#{newPayment}"
                     created-message="Created a new payment to #{newPayment.payee}" />

    <pay:payment name="newPayment"
                 payee="Somebody"
                 account="#{selectedAccount}"
                 payment-date="#{currentDatetime}" />
```

```

        created-date="#{currentDatetime}" />
        ...
    </components>

```

###

```

<components xmlns="http://jboss.com/products/seam/components"
             xmlns:pay="http://jboss.com/products/seam/examples/seampay"
             ... >

    <pay:payment-home>
        <pay:new-instance>"#{newPayment}"</pay:new-instance>
        <pay:created-message>Created a new payment to #{newPayment.payee}</pay:created-message>
    </pay:payment-home>

    <pay:payment name="newPayment">
        <pay:payee>Somebody"</pay:payee>
        <pay:account>#{selectedAccount}</pay:account>
        <pay:payment-date>#{currentDatetime}</pay:payment-date>
        <pay:created-date>#{currentDatetime}</pay:created-date>
    </pay:payment>
    ...
</components>

```

#####<pay:payment-home> ## paymentHome ###

```

package org.jboss.seam.example.seampay;
...
@Name("paymentHome")
public class PaymentController
    extends EntityHome<Payment>
{
    ...
}

```

#####

#####<pay:payment> #### org.jboss.seam.example.seampay ### Payment ## #####Payment  
#####Seam#####

```

package org.jboss.seam.example.seampay;
...
@Entity
public class Payment
    implements Serializable
{
    ...
}

```

#####Schema#

##Seam#####Schema#####Seam##Schema#####

###Seam#####:

- components — <http://jboss.com/products/seam/components>
- core — <http://jboss.com/products/seam/core>
- drools — <http://jboss.com/products/seam/drools>
- framework — <http://jboss.com/products/seam/framework>

- `jms` — <http://jboss.com/products/seam/jms>
- `remoting` — <http://jboss.com/products/seam/remoting>
- `theme` — <http://jboss.com/products/seam/theme>
- `security` — <http://jboss.com/products/seam/security>
- `mail` — <http://jboss.com/products/seam/mail>
- `web` — <http://jboss.com/products/seam/web>
- `pdf` — <http://jboss.com/products/seam/pdf>
- `spring` — <http://jboss.com/products/seam/spring>



## Chapter 5. #####

```
#####Seam#####  
#####JSF#####
```

### 5.1. Seam##

```
Seam##### Seam#####
```

- JSF##
- jBPM#####
- Seam####
- Seam#####
- Seam####

```
#####JSF EL#####Seam####JSF####JSF#####
```

```
<h:commandButton value="Click me!" action="#{helloWorld.sayHello}"/>
```

```
##jBPM#####jBPM#####
```

```
<start-page name="hello" view-id="/hello.jsp">  
  <transition to="hello">  
    <action expression="#{helloWorld.sayHello}"/>  
  </transition>  
</start-page>
```

```
#####JSF###jBPM#####Seam#####
```

#### 5.1.1. ####

```
Seam##### WEB-INF/pages.xml #####JSF##id#####
```

```
<pages>  
  <page view-id="/hello.jsp" action="#{helloWorld.sayHello}"/>  
</pages>
```

```
##### * ## view-id #####ID##
```

```
<pages>  
  <page view-id="/hello/*" action="#{helloWorld.sayHello}"/>  
</pages>
```

```
#####view-id#Seam#####
```

```
#####JSF#####Seam#####
```

```
#####<page>#####id#####JSP#Facelets###  
#####Struts#WebWork#####
```

TODO: translate struts action into page action

#####non-faces#####HTTP GET#####

##### <action> #####

```
<pages>
  <page view-id="/hello.jsp">
    <action execute="#{helloWorld.sayHello}" if="#{not validation.failed}"/>
    <action execute="#{hitCount.increment}"/>
  </page>
</pages>
```

####

##JSF faces#####“##action”#####“###parameters”#####

##GET#####JSF form#####

#####

#####

Seam#####

```
<pages>
  <page view-id="/hello.jsp" action="#{helloWorld.sayHello}">
    <param name="firstName" value="#{person.firstName}"/>
    <param name="lastName" value="#{person.lastName}"/>
  </page>
</pages>
```

<param> #####JSF#####

- ###id###non-faces(GET)#####Seam#####
- ## <s:link> # <s:button> #####request##### <s:link> #####
- ## <redirect/> ###id#####
- #####id#####JSF#####faces### PAGE #####

##### ## ##### /hello.jsp #####/hello.jsp##/hello.jsp##  
#####“##”#####

#####

##### name ##### PAGE #####

```
<pages>
  <page view-id="/hello.jsp" action="#{helloWorld.sayHello}">
    <param name="firstName" />
    <param name="lastName" />
  </page>
</pages>
```

#####CRUD#####“##”#####

- ##### <s:link> ## <s:button> #####
- #####id#####jsf#####faces###PAGE#####

#####“###”#####  
 #####non-faces#####  
 #####POST#GET###  
 #####

####

#####JSF###:

```
<pages>
  <page view-id="/calculator.jsp" action="#{calculator.calculate}">
    <param name="x" value="#{calculator.lhs}"/>
    <param name="y" value="#{calculator.rhs}"/>
    <param name="op" converterId="com.my.calculator.OperatorConverter" value="#{calculator.op}"/>
  </page>
</pages>
```

###

```
<pages>
  <page view-id="/calculator.jsp" action="#{calculator.calculate}">
    <param name="x" value="#{calculator.lhs}"/>
    <param name="y" value="#{calculator.rhs}"/>
    <param name="op" converter="#{operatorConverter}" value="#{calculator.op}"/>
  </page>
</pages>
```

JSF#### required="true" #####

```
<pages>
  <page view-id="/blog.xhtml">
    <param name="date"
      value="#{blog.date}"
      validatorId="com.my.blog.PastDate"
      required="true"/>
  </page>
</pages>
```

###

```
<pages>
  <page view-id="/blog.xhtml">
    <param name="date"
      value="#{blog.date}"
      validator="#{pastDateValidator}"
      required="true"/>
  </page>
</pages>
```

#####Hibernate#####

##### FacesMessage ##### FacesContext ##

##

#####Seam##### faces-config.xml #####JSF#####JSF#####

- #####
- #####
- #####EL#####

#####”##“### pages.xml # faces-config.xml ##### pages.xml ##

##JSF#####

```
<navigation-rule>
  <from-view-id>/editDocument.xhtml</from-view-id>

  <navigation-case>
    <from-action>#{documentEditor.update}</from-action>
    <from-outcome>success</from-outcome>
    <to-view-id>/viewDocument.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
```

#####

```
<page view-id="/editDocument.xhtml">

  <navigation from-action="#{documentEditor.update}">
    <rule if-outcome="success">
      <redirect view-id="/viewDocument.xhtml"/>
    </rule>
  </navigation>

</page>
```

#####JSF##### DocumentEditor ##### ##Seam#####

```
<page view-id="/editDocument.xhtml">

  <navigation from-action="#{documentEditor.update}"
    evaluate="#{documentEditor.errors.size}">
    <rule if-outcome="0">
      <redirect view-id="/viewDocument.xhtml"/>
    </rule>
  </navigation>

</page>
```

#####

```
<page view-id="/editDocument.xhtml">

  <navigation from-action="#{documentEditor.update}">
    <rule if="#{documentEditor.errors.empty}">
      <redirect view-id="/viewDocument.xhtml"/>
    </rule>
  </navigation>

</page>
```

#####

#####

```
<page view-id="/editDocument.xhtml">

    <navigation from-action="#{documentEditor.update}">
        <rule if="#{documentEditor.errors.empty}">
            <end-conversation/>
            <redirect view-id="/viewDocument.xhtml" />
        </rule>
    </navigation>

</page>
```

#####id#####

```
<page view-id="/editDocument.xhtml">

    <navigation from-action="#{documentEditor.update}">
        <rule if="#{documentEditor.errors.empty}">
            <end-conversation/>
            <redirect view-id="/viewDocument.xhtml">
                <param name="documentId" value="#{documentEditor.documentId}" />
            </redirect>
        </rule>
    </navigation>

</page>
```

#JSF##null#####null#####"#####"# #####null##### ### null####

```
<page view-id="/editDocument.xhtml">

    <navigation from-action="#{documentEditor.update}">
        <rule>
            <render view-id="/viewDocument.xhtml" />
        </rule>
    </navigation>

</page>
```

#####null#####

```
<page view-id="/editDocument.xhtml">

    <navigation from-action="#{documentEditor.update}">
        <render view-id="/viewDocument.xhtml" />
    </navigation>

</page>
```

view-id#####JSF EL#####

```
<page view-id="/editDocument.xhtml">

    <navigation if-outcome="success">
        <redirect view-id="/#{userAgent}/displayDocument.xhtml" />
    </navigation>

</page>
```

#####

##### calc/calculator.page.xml  
#####id /calc/calculator.jsp ##### <page> #####id#

```
<page action="#{calculator.calculate}">
```

```

<param name="x" value="#{calculator.lhs}"/>
<param name="y" value="#{calculator.rhs}"/>
<param name="op" converter="#{operatorConverter}" value="#{calculator.op}"/>
</page>

```

## 5.1.2. #####

```

Seam##### Observer/Observable ###
#####Seam### #####

```

```
### components.xml #####
```

```

<components>
  <event type="hello">
    <action execute="#{helloListener.sayHelloBack}"/>
    <action execute="#{logger.logHello}"/>
  </event>
</components>

```

```
####event type #####
```

```
##### components.xml ##### Seam#####
```

```

@Name("helloWorld")
public class HelloWorld {
  public void sayHello() {
    FacesMessages.instance().add("Hello World!");
    Events.instance().raiseEvent("hello");
  }
}

```

```
#####
```

```

@Name("helloWorld")
public class HelloWorld {
  @RaiseEvent("hello")
  public void sayHello() {
    FacesMessages.instance().add("Hello World!");
  }
}

```

```
#####
```

```

@Name("helloListener")
public class HelloListener {
  public void sayHelloBack() {
    FacesMessages.instance().add("Hello to you too!");
  }
}

```

```
### components.xml##### components.xml #####
```

```

@Name("helloListener")
public class HelloListener {
  @Observer("hello")
  public void sayHelloBack() {
    FacesMessages.instance().add("Hello to you too!");
  }
}

```

```
#####
```

```
#Seam#####
```

#####Seam#####

```
@Name("helloWorld")
public class HelloWorld {
    private String name;
    public void sayHello() {
        FacesMessages.instance().add("Hello World, my name is #0.", name);
        Events.instance().raiseEvent("hello", name);
    }
}
```

```
@Name("helloListener")
public class HelloListener {
    @Observer("hello")
    public void sayHelloBack(String name) {
        FacesMessages.instance().add("Hello #0!", name);
    }
}
```

### 5.1.3. #####

Seam#####

- org.jboss.seam.validationFailed — JSF#####
- org.jboss.seam.noConversation — #####
- org.jboss.seam.preSetVariable.<name> — ##### <name> ###
- org.jboss.seam.postSetVariable.<name> — ##### <name> ###
- org.jboss.seam.preRemoveVariable.<name> — ##### <name> ###
- org.jboss.seam.postRemoveVariable.<name> — ##### <name> ###
- org.jboss.seam.preDestroyContext.<SCOPE> — # <SCOPE> #####
- org.jboss.seam.postDestroyContext.<SCOPE> — # <SCOPE> #####
- org.jboss.seam.beginConversation — #####
- org.jboss.seam.endConversation — #####
- org.jboss.seam.beginPageflow.<name> — ##### <name> #####
- org.jboss.seam.endPageflow.<name> — ##### <name> #####
- org.jboss.seam.createProcess.<name> — ##### <name> ###
- org.jboss.seam.endProcess.<name> — ### <name> #####
- org.jboss.seam.initProcess.<name> — ### <name> #####
- org.jboss.seam.initTask.<name> — ### <name> #####
- org.jboss.seam.startTask.<name> — ### <name> #####
- org.jboss.seam.endTask.<name> — ##### <name> ###

- org.jboss.seam.postCreate.<name> — ##### <name> ###
  - org.jboss.seam.preDestroy.<name> — ##### <name> ###
  - org.jboss.seam.beforePhase — #####JSF#####
  - org.jboss.seam.afterPhase — ###JSF#####
  - org.jboss.seam.postInitialization — #Seam#####
  - org.jboss.seam.postAuthenticate.<name> — #####
  - org.jboss.seam.preAuthenticate.<name> — #####
  - org.jboss.seam.notLoggedIn — #####
  - org.jboss.seam.rememberMe — #Seam###cookie#####
  - org.jboss.seam.exceptionHandled.<type> — #Seam#####
  - org.jboss.seam.exceptionHandled — #Seam#####
  - org.jboss.seam.exceptionNotHandled — #####
  - org.jboss.seam.afterTransactionSuccess — ####Seam Application Framework#####
  - org.jboss.seam.afterTransactionSuccess.<name> — ##### <name> #####Seam Application Framework#####
- Seam#####

## 5.2. Seam ###

EJB 3.0###Bean##### ##Bean##### @AroundInvoke###  
 @Interceptors #####Bean#####

```
public class LoggedInInterceptor {

    @AroundInvoke
    public Object checkLoggedIn(InvocationContext invocation) throws Exception {

        boolean isLoggedIn = Contexts.getSessionContext().get("loggedIn")!=null;
        if (isLoggedIn) {
            //the user is already logged in
            return invocation.proceed();
        }
        else {
            //the user is not logged in, fwd to login page
            return "login";
        }
    }

}
```

#####Bean#####Bean @Interceptors(LoggedInInterceptor.class)#  
 #####EJB 3.0###Seam##### @Interceptors ##### @LoggedIn  
 #####

```
@Target (TYPE)
```



```
@Retention(RUNTIME)
@Interceptors(LoggedInInterceptor.class)
public @interface LoggedIn {}
```

##### @LoggedIn #####Bean#####

```
@Stateless
@Name("changePasswordAction")
@LoggedIn
@Interceptors(SeamInterceptor.class)
public class ChangePasswordAction implements ChangePassword {

    ...

    public String changePassword() { ... }

}
```

##### @Interceptor #####

```
@Interceptor(around={BijectionInterceptor.class,
                    ValidationInterceptor.class,
                    ConversationInterceptor.class},
            within=RemoveInterceptor.class)
public class LoggedInInterceptor
{
    ...
}
```

#####“###”#####EJB3#####

```
@Interceptor(type=CLIENT)
public class LoggedInInterceptor
{
    ...
}
```

EJB##### Seam#### @Interceptor(stateless=true)  
#####

Seam#####Seam##### Seam#####

#####JavaBean#####Seam#####EJB3 Bean#####

EJB#####@AroundInvoke##### @PostConstruct#@PreDestroy#@PrePassivate #  
@PostActive# Seam#####EJB3 Bean####JavaBean#####@PreDestroy  
#JavaBean#####

## 5.3. ####

JSF##### Seam#####XML#####  
#####EJB3.0### @ApplicationException #####checked#####

### 5.3.1. #####

EJB#####Bean#####  
##### @ApplicationException(rollback=true)#####checked#####

@ApplicationException #####unchecked##### @ApplicationException #####unchecked####

#####

Seam#Seam JavaBean#####EJB 3.0 #####

#####Seam#####Seam#####JSF#####

#####Seam#####Seam#####

### 5.3.2. ##Seam####

###Seam##### web.xml #####Servlet###:

```
<filter>
  <filter-name>Seam Filter</filter-name>
  <filter-class>org.jboss.seam.servlet.SeamFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>Seam Filter</filter-name>
  <url-pattern>*.seam</url-pattern>
</filter-mapping>
```

##### web.xml #Facelets##### components.xml #####

### 5.3.3. #####

#####Seam#####HTTP

404###

#####Seam#####

```
@HttpError(errorCode=404)
public class ApplicationException extends Exception { ... }
```

#####Seam#####

```
@Redirect(viewId="/failure.xhtml", end=true)
@ApplicationException(rollback=true)
public class UnrecoverableApplicationException extends RuntimeException { ... }
```

#####JSF#####@Redirect ###

#####EL## viewId #####

#####Seam#####

```
@Redirect(viewId="/error.xhtml", message="Unexpected error")
public class SystemException extends RuntimeException { ... }
```

### 5.3.4. #XML####

#####Seam##### pages.xml #####

```
<pages>

  <exception class="javax.persistence.EntityNotFoundException">
    <http-error error-code="404"/>
  </exception>
```

```

<exception class="javax.persistence.PersistenceException">
  <end-conversation/>
  <redirect view-id="/error.xhtml">
    <message>##### Database access failed</message>
  </redirect>
</exception>

<exception>
  <end-conversation/>
  <redirect view-id="/error.xhtml">
    <message>##### Unexpected failure</message>
  </redirect>
</exception>

</pages>

```

#### <exception> ##### pages.xml #####

#####EL## view-id #####

#####EL#####Seam#####

```

...
throw new AuthorizationException("You are not allowed to do this!");

<pages>

  <exception class="org.jboss.seam.security.AuthorizationException">
    <end-conversation/>
    <redirect view-id="/error.xhtml">
      <message severity="WARN">#{org.jboss.seam.handledException.message}</message>
    </redirect>
  </exception>

</pages>

```

org.jboss.seam.handledException  
org.jboss.seam.exception#

#####

#####

### 5.3.5. #####

#####JPA#

```

<exception class="javax.persistence.EntityNotFoundException">
  <redirect view-id="/error.xhtml">
    <message>Not found</message>
  </redirect>
</exception>

<exception class="javax.persistence.OptimisticLockException">
  <end-conversation/>
  <redirect view-id="/error.xhtml">
    <message>##### Another user changed the same data, please try again</message>
  </redirect>
</exception>

```

#####Seam####

```

<exception class="org.jboss.seam.framework.EntityNotFoundException">
  <redirect view-id="/error.xhtml">
    <message>Not found</message>
  </redirect>
</exception>

```

## #####Seam###

```
<exception class="org.jboss.seam.security.AuthorizationException">
  <redirect>
    <message>You don't have permission to do this</message>
  </redirect>
</exception>

<exception class="org.jboss.seam.security.NotLoggedInException">
  <redirect view-id="/login.xhtml">
    <message>Please log in first</message>
  </redirect>
</exception>
```

## #####JSF#

```
<exception class="javax.faces.application.ViewExpiredException">
  <redirect view-id="/error.xhtml">
    <message>##### Your session has timed out, please try again</message>
  </redirect>
</exception>
```

##### ViewExpiredException ### #####no-conversation-view-id # conversation-required #####

## Chapter 6. #####

#####Seam#####

#####Seam“##Conversation”#####

- #####workspace# #####2002#####struts#####
- ##### #####application transaction with optimistic semantics#  
###,##### #Hibernate##### LazyInitializationExcep-  
tion #####Hibernate#####Spring#####J2EE#### #Facade  
#####
- ### #####

#####

### 6.1. Seam#####

#####

- ###JSF#####JSF#####
- #JSF#####Seam##### Seam#####
- ### @Begin #####“#####”###
- ### @End #####“#####”#####
- #JSF#####Seam###“#####”#####
- ##“faces request”###JSF postback##### “faces re-  
quest”###GET#####
- ##JSF#####redirect#####Seam##### — ##### @End(beforeRedirect=true)  
###

Seam####JSF postback#####redirect##### “non-faces request”  
###GET#####

#####“non-faces request”###Seam#####Seam# conversation id #####request###

```
<a href="main.jsf?conversationId=#{conversation.id}">Continue</a>
```

###JSF#####

```
<h:outputLink value="main.jsf">
  <f:param name="conversationId" value="#{conversation.id}" />
  <h:outputText value="Continue" />
</h:outputLink>
```

#####Seam#####

```
<h:outputLink value="main.jsf">
```

```

    <s:conversationId/>
    <h:outputText value="Continue"/>
</h:outputLink>

```

#####postback#####

```

<h:commandLink action="main" value="Exit">
    <f:param name="conversationPropagation" value="none"/>
</h:commandLink>

```

#####Seam#####

```

<h:commandLink action="main" value="Exit">
    <s:conversationPropagation type="none"/>
</h:commandLink>

```

#####

#### conversationPropagation### &lt;s:conversationPropagation&gt; #####

```

<h:commandLink action="main" value="Exit">
    <s:conversationPropagation type="end"/>
</h:commandLink>

```

```

<h:commandLink action="main" value="Select Child">
    <s:conversationPropagation type="nested"/>
</h:commandLink>

```

```

<h:commandLink action="main" value="Select Hotel">
    <s:conversationPropagation type="begin"/>
</h:commandLink>

```

```

<h:commandLink action="main" value="Select Hotel">
    <s:conversationPropagation type="join"/>
</h:commandLink>

```

#####

- #####
- #####

## 6.2. ###

```

##### @Begin(nested=true) #####
##### @End
#####

```

```

#####
#####Seam#####

```

#####

TODO#####

#####

##### @PerNestedConversation ####

## 6.3. ##GET#####

JSF#####action#####JSF##“non-faces request”#####HTTP GET####  
##### <h:outputLink> #####

##### JSF action##### @Begin ##action#####

##### Seam##### @Create  
##### @Factory ###

#####Seam##### pages.xml ##### page action #

```
<pages>
  <page view-id="/messageList.jsp" action="#{messageManager.list}" />
  ...
</pages>
```

##action#####

####action#####Seam####Seam#####JSF#####

##### ## #####action#####

```
<pages>
  <page view-id="/messageList.jsp" action="#{conversation.begin}" />
  ...
</pages>
```

#####JSF#####action##### #{conversation.end} #####

##### &lt;begin-conversation&gt; ###

```
<pages>
  <page view-id="/messageList.jsp">
    <begin-conversation nested="true" pageflow="AddItem" />
  </page>
  ...
</pages>
```

&lt;end-conversation&gt;#####

```
<pages>
  <page view-id="/home.jsp">
    <end-conversation/>
  </page>
  ...
</pages>
```

#####

- # @Begin ## @Create ##
- # @Begin ## @Factory ##
- # @Begin ##Seam##action

- # pages.xml ### <begin-conversation>
- ## #{conversation.begin} ##Seam##action##

## 6.4. ##<s:link>##<s:button>

JSF#####JavaScript#####“#####”##“#####”#####      #####JSF#####  
 <h:outputLink># ## <h:outputLink> #####

- JSF#####action##### <h:outputLink> #####
- #####JSF##### DataModel #####

Seam#####      page      action      #####      ##      ##  
 ##REST#####  
 #####——##Seam#####——#####REST#####  
 @DataModel ## @DataModelSelection #####

#####Seam## <s:link> #####JSF##

#####JSF###id#

```
<s:link view="/login.xhtml" value="Login"/>
```

#####action#####action#####

```
<s:link action="#{login.logout}" value="Logout"/>
```

#####JSF###id#action### ## #####“##”##### ##action#####

```
<s:link view="/loggedOut.xhtml" action="#{login.logout}" value="Logout"/>
```

##### <h:dataTable> ## DataModel #####

```
<s:link view="/hotel.xhtml" action="#{hotelSearch.selectHotel}" value="#{hotel.name}"/>
```

#####

```
<s:link view="/main.xhtml" propagation="none"/>
```

#####

```
<s:link action="#{issueEditor.viewComment}" propagation="nest"/>
```

#####

```
<s:link action="#{documentEditor.getDocument}" propagation="begin"
pageflow="EditDocument"/>
```

#####jBPM##### taskInstance ###

```
<s:link action="#{documentApproval.approveOrReject}" taskInstance="#{task}"/>
```



###DVD Store#####

#####“##”##### <s:button>#

```
<s:button action="#{login.logout}" value="Logout"/>
```

## 6.5. ####

#####action#####JSF#

FacesMessage

#####

#####action#####JSF#####

#####Seam## facesMessages ##### Seam#####

```
@Name("editDocumentAction")
@Stateless
public class EditDocumentBean implements EditDocument {
    @In EntityManager em;
    @In Document document;
    @In FacesMessages facesMessages;

    public String update() {
        em.merge(document);
        facesMessages.add("Document updated");
    }
}
```

##### facesMessages

#####

#####“#####”#####Seam#####

#####faces message#####JSF EL####

```
facesMessages.add("Document #{document.title} was updated");
```

#####

```
<h:messages globalOnly="true"/>
```

## 6.6. ##“##”###id

#####Seam#####id#####id###

#####id#####

```
@Begin(id="#{myConversationIdGenerator.nextId}")
public void editHotel() { ... }
```

#####id#

```
@Begin(id="hotel#{hotel.id}")
public String editHotel() { ... }
```

```
@Begin(id="hotel#{hotelsDataModel.rowData.id}")
public String selectHotel() { ... }
```

```
@Begin(id="entry#{params['blogId']}")
```

```
public String viewBlogEntry() { ... }
```

```
@BeginTask(id="task#{taskInstance.id}")
public String approveDocument() { ... }
```

```
#####id#
#####id##### Seam##### @Begin ###
#####
```

## 6.7. #####

```
#####"##"##### Seam#Java#####
```

- #####id####JSF#Seam#####JPDL##### ## ### #####
- #####JSP#facelets#####

### 6.7.1. #####JSF##

```
####JSF##Seam#####Seam##### view-id ##### ##### pages.xml ##### Seam###
WEB-INF ##### faces-config.xml ###
```

```
<pages>
  <page view-id="/main.xhtml">Search hotels: #{hotelBooking.searchString}</page>
  <page view-id="/hotel.xhtml">View hotel: #{hotel.name}</page>
  <page view-id="/book.xhtml">Book hotel: #{hotel.name}</page>
  <page view-id="/confirm.xhtml">Confirm: #{booking.description}</page>
</pages>
```

```
#####Seam#####
```

### 6.7.2. #####jPDL###

```
####jPDL#####Seam#####jBPM##### ##### view-id ##### <##>
##### <page> #####
```

```
<pageflow-definition name="shopping">

  <start-state name="start">
    <transition to="browse"/>
  </start-state>

  <page name="browse" view-id="/browse.xhtml">
    <description>DVD Search: #{search.searchPattern}</description>
    <transition to="browse"/>
    <transition name="checkout" to="checkout"/>
  </page>

  <page name="checkout" view-id="/checkout.xhtml">
    <description>Purchase: ${cart.total}</description>
    <transition to="checkout"/>
    <transition name="complete" to="complete"/>
  </page>

  <page name="complete" view-id="/complete.xhtml">
    <end-conversation />
  </page>
```

```
</pageflow-definition>
```

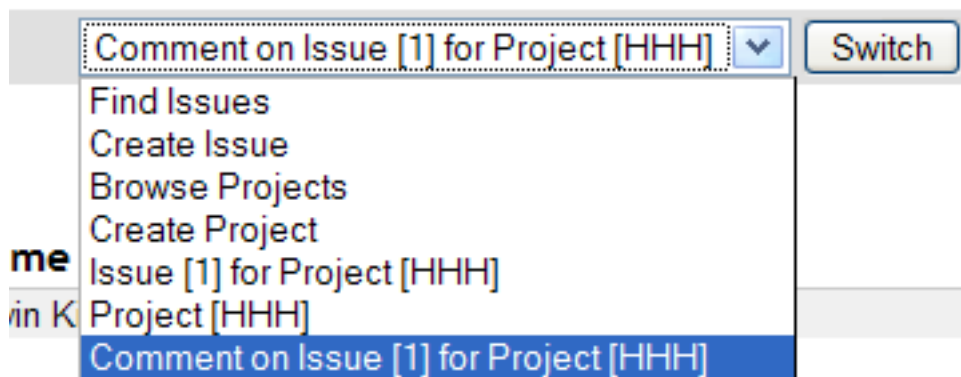
### 6.7.3. #####

###JSP#facelets#####

```
<h:selectOneMenu value="#{switcher.conversationIdOrOutcome}">
  <f:selectItem itemLabel="Find Issues" itemValue="findIssue"/>
  <f:selectItem itemLabel="Create Issue" itemValue="editIssue"/>
  <f:selectItems value="#{switcher.selectItems}" />
</h:selectOneMenu>
<h:commandButton action="#{switcher.select}" value="Switch"/>
```

In this example, we have a menu that includes an item for each conversation, together with two additional items that let the user begin a new conversation. #####

Only conversations with a description will be included in the drop-down menu.



### 6.7.4. #####

#####

```
<h:dataTable value="#{conversationList}" var="entry"
  rendered="#{not empty conversationList}">
  <h:column>
    <f:facet name="header">Workspace</f:facet>
    <h:commandLink action="#{entry.select}" value="#{entry.description}" />
    <h:outputText value="[current]" rendered="#{entry.current}" />
  </h:column>
  <h:column>
    <f:facet name="header">Activity</f:facet>
    <h:outputText value="#{entry.startDatetime}">
      <f:convertDateTime type="time" pattern="hh:mm a" />
    </h:outputText>
    <h:outputText value=" - " />
    <h:outputText value="#{entry.lastDatetime}">
      <f:convertDateTime type="time" pattern="hh:mm a" />
    </h:outputText>
  </h:column>
  <h:column>
    <f:facet name="header">Action</f:facet>
    <h:commandButton action="#{entry.select}" value="#{msg.Switch}" />
    <h:commandButton action="#{entry.destroy}" value="#{msg.Destroy}" />
  </h:column>
</h:dataTable>
```

#####

Workspace	Workspace activity	Action
<a href="#">Comment on Issue [1] for Project [HHH]</a>	01:18 PM - 01:18 PM	<a href="#">Switch</a> <a href="#">Destroy</a>
<a href="#">Issue [1] for Project [HHH]</a>	01:18 PM - 01:18 PM	<a href="#">Switch</a> <a href="#">Destroy</a>
<a href="#">Project [HHH]</a>	01:18 PM - 01:18 PM	<a href="#">Switch</a> <a href="#">Destroy</a>

#####

#####

### 6.7.5. ####

#####

```
<ui:repeat value="#{conversationStack}" var="entry">
  <h:outputText value=" | "/>
  <h:commandLink value="#{entry.description}" action="#{entry.select}"/>
</ui:repeat>
```

[Home](#) | [Find Issues](#) | [Create Issue](#) | [Project \[HHH\]](#) | [Issue \[1\] for Project \[HHH\]](#)

Issue Attributes

### 6.8. #####JSF####

#####JSF#####

###postback#####Seam#####

#####

```
@Name("grid")
@Scope(ScopeType.EVENT)
public class Grid
{
    private HtmlPanelGrid htmlPanelGrid;

    // getters and setters
    ...
}
```

```
@Name("gridEditor")
@Scope(ScopeType.CONVERSATION)
public class GridEditor
{
    @In(required=false)
    private Grid grid;

    ...
}
```

##### uiComponent #####JSF#### ##### UIData ### getRowIndex()

#####

```
<h:dataTable id="lineItemTable" var="lineItem" value="#{orderHome.lineItems}">
```

```

<h:column>
    Row: #{uiComponent['lineItemTable'].rowIndex}
</h:column>
...
</h:dataTable>

```

```
###map#####JSF UI#####
```

## 6.9. #####

```

#    ???    #####Seam#####      #####AJAX#####—#
#####Ajax#####—####RickFaces#####

```

```

#####Seam#####
#####—#####
##,AJAX#####

```

```

##Seam#action#####      #####
#####AJAX#####

```

```
#####components.xml#####ms##
```

```
<core:manager concurrent-request-timeout="500" />
```

```

#####AJAX##——      #####
#AJAX#####

```

```

#####“Poll”####
#####AJAX#####action#####
#####action#####action#####

```

```
#####AJAX#####
```

```
#####“##”#####“Poll##”####
```

```

#####“##”#####AJAX#####——#####
#####      #####
#####onblur#####
#####

```

```
#####
```

```
##“Poll##”##### #######action## @Asynchronous #####
```

```

int total;

// This method is called when an event occurs on the client
// It takes a really long time to execute
@Asynchronous
public void calculateTotal() {
    total = someReallyComplicatedCalculation();
}

// This method is called as the result of the poll
// It's very quick to execute
public int getTotal() {
    return total;
}

```

## 6.9.1. RichFaces Ajax

RichFaces AJAX####Seam####Ajax#####

- eventsQueue — #####  
#####
- #####'###'### ignoreDupResponses — ##### ignoreDupResponses="true" ####  
##### — #####  
  
#####Seam#####
- requestDelay — #####ms##  
#####response#####  
  
#####Seam##### #####action#####
- <a:poll reRender="total" interval="1000" /> — #####

# Chapter 7. #####

JBoss                      jBPM                      #####Java                      SE#EE#####  
jBPM#####Web#####  
#####jPDL#XML#####Eclipse#####  
jPDL#####Web#####SOA#####

Seam#####jBPM#####

- #####jDPL##### Seam#####

- #####jBPM#####  
#####jBPM#####

##### Pageflow Conversation # Task #####  
#####jBPM#####

##Seam###jDPL#####JSF##Seam#####jBPM#  
#####

## 7.1. Seam#####

#Seam#####:

- ##JSF##Seam#### – #####
- ##jPDL – #####

#####

### 7.1.1. #####

#####  
####Action#####

#####JSF#####

```
<navigation-rule>
  <from-view-id>/numberGuess.jsp</from-view-id>

  <navigation-case>
    <from-outcome>guess</from-outcome>
    <to-view-id>/numberGuess.jsp</to-view-id>
    <redirect/>
  </navigation-case>

  <navigation-case>
    <from-outcome>win</from-outcome>
    <to-view-id>/win.jsp</to-view-id>
    <redirect/>
  </navigation-case>

  <navigation-case>
    <from-outcome>lose</from-outcome>
    <to-view-id>/lose.jsp</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
```

```

    </navigation-case>

</navigation-rule>

```

#####Seam#####:

```

<page view-id="/numberGuess.jsp">

    <navigation>
        <rule if-outcome="guess">
            <redirect view-id="/numberGuess.jsp"/>
        </rule>
        <rule if-outcome="win">
            <redirect view-id="/win.jsp"/>
        </rule>
        <rule if-outcome="lose">
            <redirect view-id="/lose.jsp"/>
        </rule>
    </navigation>

</page>

```

#####Action#####View#id#

```

public String guess() {
    if (guess==randomNumber) return "/win.jsp";
    if (++guessCount==maxGuesses) return "/lose.jsp";
    return null;
}

```

#####

```

public String search() {
    return "/searchResults.jsp?searchPattern=#{searchAction.searchPattern}";
}

```

#####

#####jPDL#####Action#####

#####jPDL#####

```

<pageflow-definition name="numberGuess">

    <start-page name="displayGuess" view-id="/numberGuess.jsp">
        <redirect/>
        <transition name="guess" to="evaluateGuess">
            <action expression="#{numberGuess.guess}" />
        </transition>
    </start-page>

    <decision name="evaluateGuess" expression="#{numberGuess.correctGuess}">
        <transition name="true" to="win"/>
        <transition name="false" to="evaluateRemainingGuesses"/>
    </decision>

    <decision name="evaluateRemainingGuesses" expression="#{numberGuess.lastGuess}">
        <transition name="true" to="lose"/>
        <transition name="false" to="displayGuess"/>
    </decision>

    <page name="win" view-id="/win.jsp">
        <redirect/>
        <end-conversation />
    </page>

```

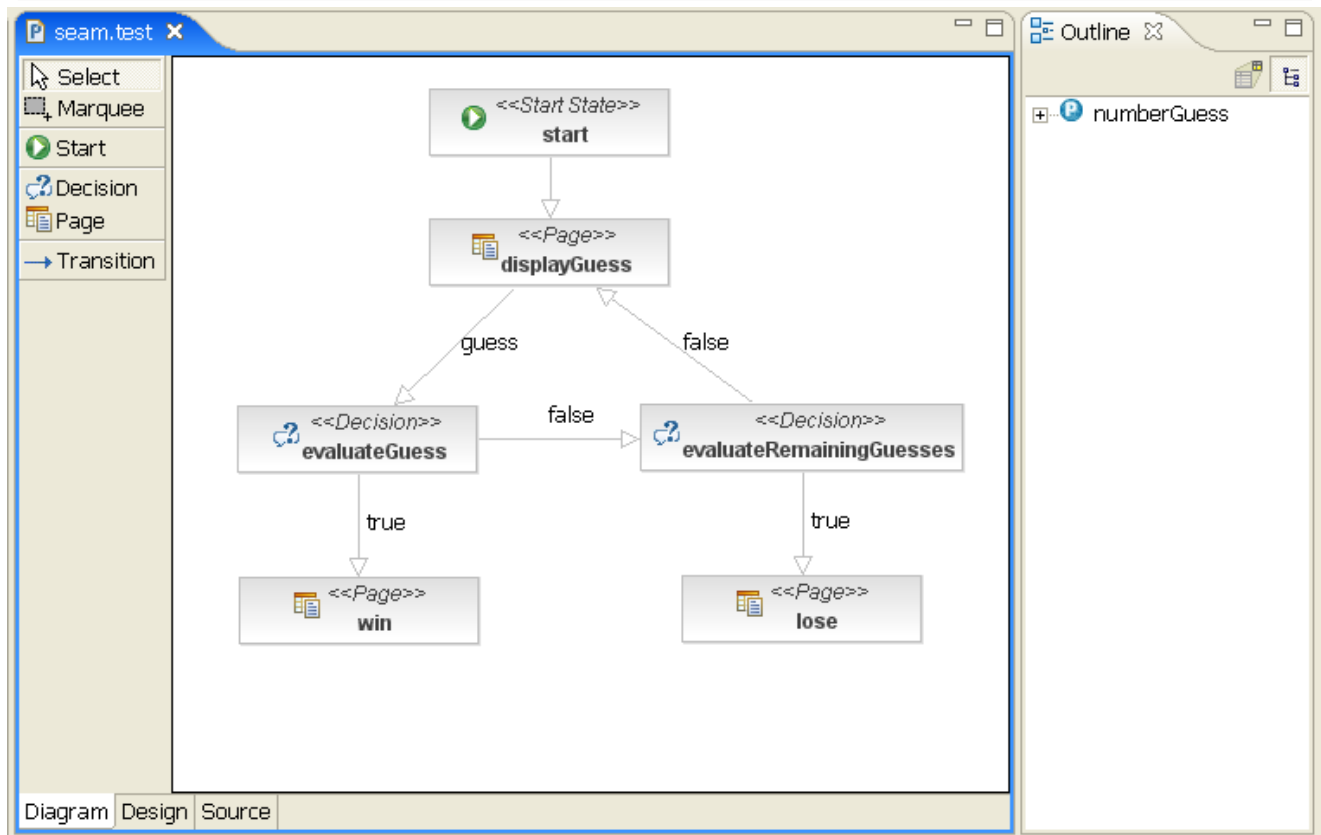


```

<page name="lose" view-id="/lose.jsp">
  <redirect/>
  <end-conversation />
</page>

</pageflow-definition>

```



#####

- JSF/Seam#### ## #####Java#####
- jPDL#####JSP#Java###

```
#####    ###    #####
#####
```

```
#####    ###Seam#####    ##
#####    #####Seam#####
#####
```

#####

## 7.1.2. Seam####

```
###JSF#Seam#####Seam#####
#####Web#####Struts#WebWork    ##    #####    ##    #####EJB###Session
Beans#Spring#####    #####Seam#####Session
Bean#####    ####Action#####    no-conversation-view-id    ####null#####
#####
```

```
#####pages.xml    #####    no-conversation-view-id    ###
```

###Seam#####

```
<page view-id="/checkout.xhtml"
      no-conversation-view-id="/main.xhtml"/>
```

#####

Seam#####Action#####

#####Page####

back="enabled" #####

```
<page name="checkout"
      view-id="/checkout.xhtml"
      back="enabled">
  <redirect/>
  <transition to="checkout"/>
  <transition name="complete" to="complete"/>
</page>
```

##### # checkout ####

##### no-  
conversation-view-id ###

```
<page name="checkout"
      view-id="/checkout.xhtml"
      back="enabled"
      no-conversation-view-id="/main.xhtml">
  <redirect/>
  <transition to="checkout"/>
  <transition name="complete" to="complete"/>
</page>
```

#####

## 7.2. ##jPDL##

### 7.2.1. #####

#####Seam jBPM##### components.xml #####

```
<bpm:jbp>
  <bpm:pageflow-definitions>
    <value>pageflow.jpdl.xml</value>
  </bpm:pageflow-definitions>
</bpm:jbp>
```

#####jBPM#####jPDL#####

### 7.2.2. #####

##### @Begin#@BeginTask # @StartTask #####“##”#####jPDL#####

```
@Begin(pageflow="numberguess")
public void begin() { ... }
```

#####pages.xml#####:

```
<page>
    <begin-conversation pageflow="numberguess" />
</page>
```

```
##### RENDER_RESPONSE ##— ##### @Factory # @Create ### --
##### <start-page> #####
```

```
#####Action#####action#####
<start-state> #####
```

```
<pageflow-definition name="viewEditDocument">

    <start-state name="start">
        <transition name="documentFound" to="displayDocument" />
        <transition name="documentNotFound" to="notFound" />
    </start-state>

    <page name="displayDocument" view-id="/document.jsp">
        <transition name="edit" to="editDocument" />
        <transition name="done" to="main" />
    </page>

    ...

    <page name="notFound" view-id="/404.jsp">
        <end-conversation />
    </page>

</pageflow-definition>
```

### 7.2.3. #####

```
### <page> #####
```

```
<page name="displayGuess" view-id="/numberGuess.jsp">
    <redirect />
    <transition name="guess" to="evaluateGuess">
        <action expression="#{numberGuess.guess}" />
    </transition>
</page>
```

```
### view-id ###JSF ##id# <redirect /> ###JSF##### <redirect /> #####
####post-then-redirect#####
#####Seam#####Seam#####ROR#Ruby On Rails####"flash"####
```

```
#####JSF##### numberGuess.jsp #####Command####Command#####
```

```
<h:commandButton type="submit" value="Guess" action="guess" />
```

```
#####jBPM##### numberGuess ### guess() #####Action#
###jPDL#####Action#####JSF#EL#####Action Handler###Seam#####
JSF#####jBPM#####
```

```
#####Command##### action #####
#####name####Seam#####name###Seam#####
```

```
<h:commandButton type="submit" value="Guess" />
```

```
#####
```

```
<page name="displayGuess" view-id="/numberGuess.jsp">
  <redirect/>
  <transition to="evaluateGuess">
    <action expression="#{numberGuess.guess}" />
  </transition>
</page>
```

#####Action#####Action#####

```
<h:commandButton type="submit" value="Guess" action="#{numberGuess.guess}" />
```

```
<page name="displayGuess" view-id="/numberGuess.jsp">
  <transition name="correctGuess" to="win"/>
  <transition name="incorrectGuess" to="evaluateGuess"/>
</page>
```

#####

## 7.2.4. ####

#####jPDL##### &lt;decision&gt; ###

```
<decision name="evaluateGuess" expression="#{numberGuess.correctGuess}">
  <transition name="true" to="win"/>
  <transition name="false" to="evaluateRemainingGuesses"/>
</decision>
```

###Seam#####JSF EL#####

## 7.2.5. #####

## &lt;end-conversation&gt; ## @End ##### #####

```
<page name="win" view-id="/win.jsp">
  <redirect/>
  <end-conversation/>
</page>
```

#####jBPM# #####transition# #####Seam#####

```
<page name="win" view-id="/win.jsp">
  <redirect/>
  <end-task transition="success"/>
</page>
```

## 7.2.6. #####

##### ## &lt;process-state&gt; #####

```
<process-state name="cheat">
  <sub-process name="cheat"/>
  <transition to="displayGuess"/>
</process-state>
```

```
##### <start-state> ##### <end-state> ##### <process-state>
#####
```

## 7.3. Seam#####

```
##### # #####
Seam#jBPM##### Seam##### BUSINESS_PROCESS
#####jBPM#####
```

```
##### <task-node> ##### <page>
#####
```

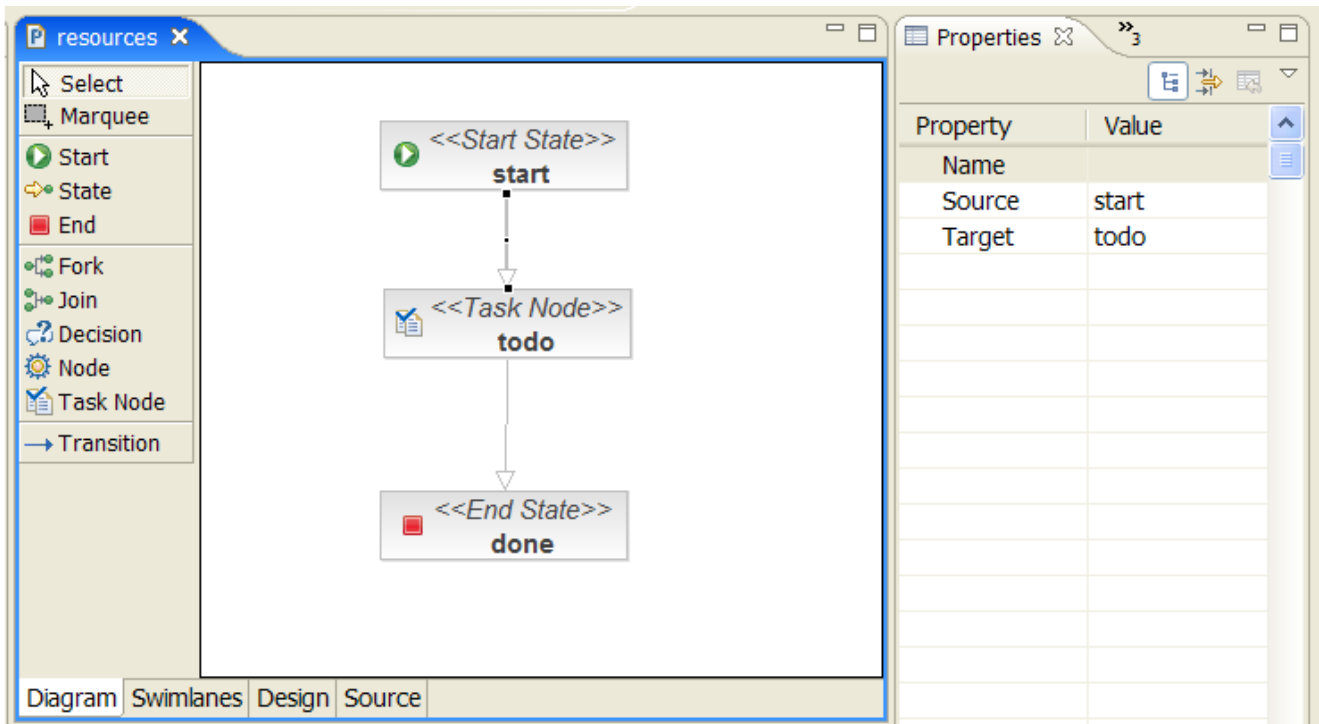
```
<process-definition name="todo">

  <start-state name="start">
    <transition to="todo"/>
  </start-state>

  <task-node name="todo">
    <task name="todo" description="#{todoList.description}">
      <assignment actor-id="#{actor.id}" />
    </task>
    <transition to="done"/>
  </task-node>

  <end-state name="done" />

</process-definition>
```



```
#####jPDL#####jPDL##### <task> #####
<pageflow-definition>#
```

## 7.4. ##jPDL#####

### 7.4.1. #####

```
#####jBPM#####
```

```
<bpm:jbpm>
  <bpm:process-definitions>
    <value>todo.jpdl.xml</value>
  </bpm:process-definitions>
</bpm:jbpm>
```

## 7.4.2. ###Actor id

#####jBPM## actor id # group actor id “##”### #####Seam### actor #####id#

```
@In Actor actor;

public String login() {
    ...
    actor.setId( user.getUserName() );
    actor.getGroupActorIds().addAll( user.getGroupNames() );
    ...
}
```

## 7.4.3. #####

## @CreateProcess #####

```
@CreateProcess(definition="todo")
public void createTodo() { ... }
```

#####pages.xml#####

```
<page>
  <create-process definition="todo" />
</page>
```

## 7.4.4. ####

#####actor id#####Seam###

```
<task name="todo" description="#{todoList.description}">
  <assignment actor-id="#{actor.id}"/>
</task>
```

#####pool actor##

```
<task name="todo" description="#{todoList.description}">
  <assignment pooled-actors="employees"/>
</task>
```

## 7.4.5. ####

#####Seam#####pooledTaskInstanceList #####

```
<h:dataTable value="#{pooledTaskInstanceList}" var="task">
  <h:column>
    <f:facet name="header">Description</f:facet>
    <h:outputText value="#{task.description}"/>
  </h:column>
  <h:column>
    <s:link action="#{pooledTask.assignToCurrentActor}" value="Assign" taskInstance="#{task}"/>
  </h:column>
</h:dataTable>
```

```

    </h:column>
</h:dataTable>

```

#####JSF## <h:commandLink> ### <s:link> ###

```

<h:commandLink action="#{pooledTask.assignToCurrentActor}">
    <f:param name="taskId" value="#{task.id}"/>
</h:commandLink>

```

pooledTask #####

taskInstanceListForType#####

```

<h:dataTable value="#{taskInstanceListForType['todo']}" var="task">
    <h:column>
        <f:facet name="header">Description</f:facet>
        <h:outputText value="#{task.description}"/>
    </h:column>
    <h:column>
        <s:link action="#{todoList.start}" value="Start Work" taskInstance="#{task}"/>
    </h:column>
</h:dataTable>

```

## 7.4.6. #####

##### @StartTask ## @BeginTask #####

```

@StartTask
public String start() { ... }

```

#####pages.xml#####

```

<page>
    <start-task />
</page>

```

#####

##### @EndTask #####Seam#####

```

@EndTask(transition="completed")
public String completed() { ... }

```

#####pages.xml

```

<page>
    <end-task transition="completed" />
</page>

```

#####EL#pages.xml##transition #

###jBPM#####

###jBPM#####jBPM#####

---

## Chapter 8. Seam####/####

Seam#####Java#####Hibernate3##EJB      3.0###Java      Persistence      API#####  
Seam#####web#####ORM####

### 8.1. ##

```
Seam##Hibernate#####Java#####
###Java#####Hibernate#####Seam##### Seam#####
#####
#####(database/JTA)#####
#####Hibernate#####

#####Seam#EJB3.0#####“###”#####
#####Hibernate#      LazyInitializationException      #####
#####

EJB3.0#####bean#####
#####
```

- #####bean#####Web#####
- #####

```
Seam#####(Conversation)#####Session      Bean#####
#####Seam#####
#####      ##Seam###EJB
3.0#####
```

### 8.2. Seam####

```
EJB##Bean#####Bean#####EJB#####
#####JSF#####Bean#####action#####
#####Seam#####
```

```
#####Seam#####Bean#####
```

- #####Web#####Seam##Web#####EJB#####
- #####lazy fetching of associations##

```
#####
```

```
Hibernate##### "Open Session in View" ##### #Hibernate###"Open Session in
View"#####Spring##### LazyInitializationException
###
```

```
##### "Open Session in
View"#####
```

```
Seam###"Open Session in View"#####
```



- #####

- #####

#####Seam####

#####Seam#####Seam#####Seam#####

#####

#####EJB 3.0#####Seam#####Java

EE

5#####Seam#####Seam#####

### 8.2.1. ##Seam####

###JSF#####Seam##### ## ##### components.xml #####

```
<core:init transaction-management-enabled="false"/>
<transaction:no-transaction />
```

### 8.2.2. ##Seam####

Seam#####Seam####JTA#####EJB####EJB####

Seam#####API#####

- javax.persistence.EntityTransaction #####JPA RESOURCE\_LOCAL###

- org.hibernate.Transaction #####Hibernate#####

- org.springframework.transaction.PlatformTransactionManager #####Spring#####

- ####Seam#####

```
#components.xml#####JPA RESOURCE_LOCAL######{entityManager} # persist-
ence:managed-persistence-context ##### Seam#####
```

```
<transaction:entity-transaction entity-manager="#{entityManager}"/>
```

```
###components.xml#####Hibernate######{hibernateSession} #### persist-
ence:managed-hibernate-session #####Seam#####
```

```
<transaction:hibernate-transaction session="#{hibernateSession}"/>
```

#components.xml#####Seam#####

```
<transaction:no-transaction />
```

## ##Spring PlatformTransactionManagement ###Spring#####

### 8.2.3. ####

##### beforeCompletion() # afterCompletion() ###

#####Seam#####Seam##### ###Java EE

5#####components.xml####

<transaction:ejb-transaction/>#####Seam#####Seam#####

## 8.3. Seam#####

#####Java EE 5#####Seam#####Java EE 5  
#####

#####JPA#####Hibernate#####Seam#####  
EntityManager ##### Session #####Seam##### @In #####

Seam#####EJB 3.0#####Seam#####  
Seam#####EJB#####

### 8.3.1. #Seam#####JPA

####Seam##### components.xml ####

```
<persistence:managed-persistence-context name="bookingDatabase"
    auto-create="true"
    persistence-unit-jndi-name=" java:/EntityManagerFactories/bookingData"/>
```

##### bookingDatabase #####Seam#####EntityManagerFactory ##### EntityManager  
##### #JNDI## java:/EntityManagerFactories/bookingData#

#####JNDI#### EntityManagerFactory# #JBoss##### persistence.xml #####

```
<property name="jboss.entity.manager.factory.jndi.name"
    value=" java:/EntityManagerFactories/bookingData"/>
```

##### EntityManager ##

```
@In EntityManager bookingDatabase;
```

#####EJB3##### @TransactionAttribute(REQUIRES\_NEW)#####  
##Seam#####REQUIRES\_NEW#####  
#####REQUIRES\_NEW#####@PersistenceContext#####

### 8.3.2. ##Seam###Hibernate##

Seam###Hibernate sessions##### components.xml #####

```
<persistence:hibernate-session-factory name="hibernateSessionFactory"/>
<persistence:managed-hibernate-session name="bookingDatabase"
    auto-create="true"
    session-factory-jndi-name=" java:/bookingSessionFactory"/>
```

java:/bookingSessionFactory ## hibernate.cfg.xml #####

```
<session-factory name=" java:/bookingSessionFactory">
    <property name="transaction.flush_before_completion">true</property>
    <property name="connection.release_mode">after_statement</property>
    <property name="transaction.manager_lookup_class">org.hibernate.transaction.JBossTransactionManager</property>
    <property name="transaction.factory_class">org.hibernate.transaction.JTATransactionFactory</property>
    <property name="connection.datasource">java:/bookingDatasource</property>
```

```
...
</session-factory>
```

```
###Seam##### hibernate.transaction.flush_before_completion####JTA#####
```

```
#####JavaBean#####Hibernate Session##
```

```
@In Session bookingDatabase;
```

### 8.3.3. Seam#####

```
#####merge()##### LazyInitializa-
tionException ### NonUniqueObjectException ###
```

```
##### Hibernate#EJB 3.0##### @Version #####
```

```
#####
##### JBoss##Sun##Sybase#EJB
3.0#####EJB 3.0##### ##Hibernate##### FlushModeType
#####
```

```
Seam#####FlushModeType.MANUAL###
```

```
#####Hibernate#####
```

```
@In EntityManager em; //a Seam-managed persistence context

@Begin(flushMode=MANUAL)
public void beginClaimWizard() {
    claim = em.find(Claim.class, claimId);
}
```

```
## claim #####claim#####
```

```
public void addPartyToClaim() {
    Party party = ...;
    claim.addParty(party);
}
```

```
#####
```

```
@End
public void commitClaim() {
    em.flush();
}
```

```
#####pages.xml#### flushMode #### MANUAL#####
```

```
<begin-conversation flush-mode="MANUAL" />
```

## 8.4. ##JPA “###delegate#”

```
EntityManager ##### getDelegate() #####API# #####Hibernate#####
org.hibernate.Session# #####
```

```
#####Hibernate#####Seam#####
```

```
@In EntityManager entityManager;

@Create
public void init() {
    ( (Session) entityManager.getDelegate() ).enableFilter("currentVersions");
}
```

#####Java##### components.xml ####

```
<factory name="session"
    scope="STATELESS"
    auto-create="true"
    value="#{entityManager.delegate}" />
```

#####

```
@In Session session;

@Create
public void init() {
    session.enableFilter("currentVersions");
}
```

## 8.5. #EJB-QL/HQL###EL

#####Seam##### @PersistenceContext ##### Seam### EntityManager ## Session  
#####EL#####

```
User user = em.createQuery("from User where username=#{user.username}")
    .getSingleResult();
```

#####

```
User user = em.createQuery("from User where username=:username")
    .setParameter("username", user.getUsername())
    .getSingleResult();
```

#####

```
User user = em.createQuery("from User where username=" + user.getUsername()) //BAD!
    .getSingleResult();
```

#####SQL#####

## 8.6. ##Hibernate###

Hibernate#####

####filter#####

#####Seam#####Seam#####

Seam##### EntityManager ## Hibernate Session  
#####Hibernate#####

```
<persistence:filter name="regionFilter">
    <persistence:name>region</persistence:name>
    <persistence:parameters>
        <key>regionCode</key>
        <value>#{region.code}</value>
```

```
        </persistence:parameters>
    </persistence:filter>

    <persistence:filter name="currentFilter">
        <persistence:name>current</persistence:name>
        <persistence:parameters>
            <key>date</key>
            <value>#{currentDate}</value>
        </persistence:parameters>
    </persistence:filter>

    <persistence:managed-persistence-context name="personDatabase"
        persistence-unit-jndi-name="java:/EntityManagerFactories/personDatabase">
        <core:filters>
            <value>#{regionFilter}</value>
            <value>#{currentFilter}</value>
        </core:filters>
    </persistence:managed-persistence-context>
```

## Chapter 9. Seam##JSF####

###JSF#####:

```
<h:form>
  <h:messages/>

  <div>
    Country:
    <h:inputText value="#{location.country}" required="true">
      <my:validateCountry/>
    </h:inputText>
  </div>

  <div>
    Zip code:
    <h:inputText value="#{location.zip}" required="true">
      <my:validateZip/>
    </h:inputText>
  </div>

  <h:commandButton/>
</h:form>
```

#####DRY#####“validation”#####Schema###  
Seam##Hibernate Validator#####model#####

##### Location #####

```
public class Location {
    private String country;
    private String zip;

    @NotNull
    @Length(max=30)
    public String getCountry() { return country; }
    public void setCountry(String c) { country = c; }

    @NotNull
    @Length(max=6)
    @Pattern("^\d*$")
    public String getZip() { return zip; }
    public void setZip(String z) { zip = z; }
}
```

##### Hibernate Validator #####

```
public class Location {
    private String country;
    private String zip;

    @NotNull
    @Country
    public String getCountry() { return country; }
    public void setCountry(String c) { country = c; }

    @NotNull
    @ZipCode
    public String getZip() { return zip; }
    public void setZip(String z) { zip = z; }
}
```

#####JSF##### <s:validate> #####model#####

```
<h:form>
```

```

<h:messages/>

<div>
    Country:
    <h:inputText value="#{location.country}" required="true">
        <s:validate/>
    </h:inputText>
</div>

<div>
    Zip code:
    <h:inputText value="#{location.zip}" required="true">
        <s:validate/>
    </h:inputText>
</div>

<h:commandButton/>

</h:form>

```

### #model### @NotNull ### ##### required="true"# #####JSF#####

#####model# ## ##### ## #### — #####

##### <s:validateAll>#

```

<h:form>

    <h:messages/>

    <s:validateAll>

        <div>
            Country:
            <h:inputText value="#{location.country}" required="true"/>
        </div>

        <div>
            Zip code:
            <h:inputText value="#{location.zip}" required="true"/>
        </div>

        <h:commandButton/>

    </s:validateAll>

</h:form>

```

##### <s:validate> ### #####

#####

#####JSF#####JSF##### JSF#####  
#####\*##

#####facelets####

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:s="http://jboss.com/products/seam/taglib">

    <div>

        <s:label styleClass="#{invalid?'error':''}">
            <ui:insert name="label"/>
        </s:label>
    </div>

```

```

        <s:span styleClass="required" rendered="#{required}">*</s:span>
    </s:label>

    <span class="#{invalid?'error':''}">
        <h:graphicImage src="img/error.gif" rendered="#{invalid}" />
        <s:validateAll>
            <ui:insert />
        </s:validateAll>
    </span>

    <s:message styleClass="error" />

</div>

</ui:composition>

```

##### <s:decorate> #####

```

<h:form>

    <h:messages globalOnly="true" />

    <s:decorate template="edit.xhtml">
        <ui:define name="label">Country:</ui:define>
        <h:inputText value="#{location.country}" required="true" />
    </s:decorate>

    <s:decorate template="edit.xhtml">
        <ui:define name="label">Zip code:</ui:define>
        <h:inputText value="#{location.zip}" required="true" />
    </s:decorate>

    <h:commandButton />

</h:form>

```

##### RichFaces Ajax #####

```

<h:form>

    <h:messages globalOnly="true" />

    <s:decorate id="countryDecoration" template="edit.xhtml">
        <ui:define name="label">Country:</ui:define>
        <h:inputText value="#{location.country}" required="true">
            <a:support event="onblur" reRender="countryDecoration" bypassUpdates="true" />
        </h:inputText>
    </s:decorate>

    <s:decorate id="zipDecoration" template="edit.xhtml">
        <ui:define name="label">Zip code:</ui:define>
        <h:inputText value="#{location.zip}" required="true">
            <a:support event="onblur" reRender="zipDecoration" bypassUpdates="true" />
        </h:inputText>
    </s:decorate>

    <h:commandButton />

</h:form>

```

#####id#####

Selenium

#####UI#####

#####id##JSF#####

```

<h:form id="form">

    <h:messages globalOnly="true" />

```



```

<s:decorate id="countryDecoration" template="edit.xhtml">
  <ui:define name="label">Country:</ui:define>
  <h:inputText id="country" value="#{location.country}" required="true">
    <a:support event="onblur" reRender="countryDecoration" bypassUpdates="true"/>
  </h:inputText>
</s:decorate>

<s:decorate id="zipDecoration" template="edit.xhtml">
  <ui:define name="label">Zip code:</ui:define>
  <h:inputText id="zip" value="#{location.zip}" required="true">
    <a:support event="onblur" reRender="zipDecoration" bypassUpdates="true"/>
  </h:inputText>
</s:decorate>

<h:commandButton/>

</h:form>

```

#####Seam#Hibernate

Validator#####

#####el#####

```

public class Location {
    private String name;
    private String zip;

    // Getters and setters for name

    @NotNull
    @Length(max=6)
    @ZipCode(message="#{messages['location.zipCode.invalid']}")
    public String getZip() { return zip; }
    public void setZip(String z) { zip = z; }
}

```

```
location.zipCode.invalid = The zip code is not valid for #{location.name}
```

---

# Chapter 10. Groovy##

JBoss Seam#####RAD#####RAD#####  
#####API#####Java API#  
#####API## Groovy [http://groovy.codehaus.org]  
#####Java#####

###JBoss Seam#####Java EE##### JBoss Seam#####  
#####Seam#####Seam#####API#####

## 10.1. Groovy##

Groovy#####Java#####Python#Ruby#Smalltalk#####Groovy#####

- Groovy##Java###Java####Groovy#####
- Groovy####Java###Groovy###Java##Groovy#####Java#####

TODO: ####Groovy#####

## 10.2. #Groovy##Seam##

#####Groovy####Java#####Groovy####Seam####Java##### #####Groovy#Java##

### 10.2.1. ##Groovy##

#####Seam#####annotation####Groovy#####1.1 Beta1### #####Seam#####Groovy#####

#### 10.2.1.1. ##

```
@Entity
@Name("hotel")
class Hotel implements Serializable
{
    @Id @GeneratedValue
    Long id

    @Length(max=50) @NotNull
    String name

    @Length(max=100) @NotNull
    String address

    @Length(max=40) @NotNull
    String city

    @Length(min=2, max=10) @NotNull
    String state

    @Length(min=4, max=6) @NotNull
    String zip

    @Length(min=2, max=40) @NotNull
    String country

    @Column(precision=6, scale=2)
```

```

        BigDecimal price

        @Override
        String toString()
        {
            return "Hotel(${name},${address},${city},${zip})"
        }
    }
}

```

Groovy#####getter/setter#####getter#setter##### Hotel##### hotel.getCity()  
#####Java####getter#setter####Groovy#####

Groovy1.1 Beta1#####Generics##### @\*ToMany #####  
Collection<Entity># ##### Chapter 11, Seam#####  
#####Groovy1.1#####Groovy1.1 Beta2#####

### 10.2.1.2. Seam##

##Groovy##Seam#####Java#####Seam###

```

@Scope(ScopeType.SESSION)
@Name("bookingList")
class BookingListAction implements Serializable
{
    @In EntityManager em
    @In User user
    @DataModel List<Booking> bookings
    @DataModelSelection Booking booking
    @Logger Log log

    @Factory public void getBookings()
    {
        bookings = em.createQuery(''
            select b from Booking b
            where b.user.username = :username
            order by b.checkinDate''
        ).setParameter("username", user.username)
        .getResultList()
    }

    public void cancel()
    {
        log.info("Cancel booking: #{bookingList.booking.id} for #{user.username}")
        Booking cancelled = em.find(Booking.class, booking.id)
        if (cancelled != null) em.remove( cancelled )
        getBookings()
        FacesMessages.instance().add("Booking cancelled for confirmation number #{bookingList.booking
    }
}

```

## 10.2.2. seam-gen

Seam gen#####Groovy#####seam-gen#####Groovy##### Groovy##### .groovy  
#### src/model ##### action#### .groovy #### src/action #####

## 10.3. ##

##Groovy####Java#####3#####

JBoss Seam#####JavaBean Seam##### # .groovy  
#####Seam#GroovyBean Seam #####

### 10.3.1. ##Groovy##

Groovy## # Java###Java##### ####Groovy###Groovy Session Bean##Groovy  
Seam##### groovyc Ant###  
#####Groovy##Java#####Java#####Groovy#Java#####

### 10.3.2. #####.groovy##

JBoss Seam#### .groovy #####/##### ####.groovy##### Section 2.7,  
“Seam#####” #####Groovy###.groovy#### WEB-INF/dev ####  
#####GroovyBean#####  
###.groovy#####Seam#####

- #####JavaBean#GroovyBean####EJB3 Bean
- #####
- ##### WEB-INF/dev #####
- ###Seam Debug##

### 10.3.3. seam-gen

Seam-gen#####Groovy##### .groovy #####WAR###seam-gen####  
src/action #####Java#Groovy#####Groovy#####  
### examples/groovybooking #####Booking Demo#####Groovy#####

# Chapter 11. Seam#####

Seam#####Java#####  
# component.xml #####

###Web#####Hibernate##JPA#####Seam                      Application                      Framework#Seam#####  
#####

#####                      “##”###Seam##                      —  
#####Seam#####“##”#

## 11.1. ##

#####Seam    Application    Framework#####Seam#####    components.xml  
#####    #####    components.xml    #####    Person    #####CRUD###(Create)###(Retrieve)  
###(Update)###(Delete)#####

```
<framework:entity-home name="personHome"
                        entity-class="eg.Person"
                        entity-manager="{personDatabase}">
    <framework:id>#{param.personId}</framework:id>
</framework:entity-home>
```

#####“#XML##”#####

```
@Stateful
@Name("personHome")
public class PersonHome extends EntityHome<Person> implements LocalPersonHome {
    @RequestParameter String personId;
    @In EntityManager personDatabase;

    public Object getId() { return personId; }
    public EntityManager getEntityManager() { return personDatabase; }
}
```

#####

#####Bean#####JavaBean#####JBoss  
AS#####4.2.2.GA#####

###Seam#####CRUD### EntityHome # HibernateEntityHome ##### EntityQuery # Hi-  
bernateEntityQuery#

####Home#Query#####session#event#conversation#####scope#####

Seam#####Seam##### entityManager #####

## 11.2. Home##

Home##### Person ##

```
@Entity
public class Person {
    @Id private Long id;
```

```

private String firstName;
private String lastName;
private Country nationality;

//getters and setters...
}

```

##### personHome ###

```
<framework:entity-home name="personHome" entity-class="eg.Person" />
```

#####

```

@Name("personHome")
public class PersonHome extends EntityHome<Person> {}

```

Home#####persist()#remove()#update() # getInstance()# ##### remove() # update()  
##### setId() #####

#####JSF#####Home#####

```

<h1>Create Person</h1>
<h:form>
  <div>First name: <h:inputText value="#{personHome.instance.firstName}"/></div>
  <div>Last name: <h:inputText value="#{personHome.instance.lastName}"/></div>
  <div>
    <h:commandButton value="Create Person" action="#{personHome.persist}"/>
  </div>
</h:form>

```

#####person ##person##### components.xml #####

```

<factory name="person"
  value="#{personHome.instance}"/>

<framework:entity-home name="personHome"
  entity-class="eg.Person" />

```

##### PersonHome ##### @Factory #####

```

@Name("personHome")
public class PersonHome extends EntityHome<Person> {

  @Factory("person")
  public Person initPerson() { return getInstance(); }

}

```

#####JSF#####

```

<h1>Create Person</h1>
<h:form>
  <div>First name: <h:inputText value="#{person.firstName}"/></div>
  <div>Last name: <h:inputText value="#{person.lastName}"/></div>
  <div>
    <h:commandButton value="Create Person" action="#{personHome.persist}"/>
  </div>
</h:form>

```

##### Person ##### Person ##### Per-  
sonHome#####

```

<pages>
  <page view-id="/editPerson.jsp">
    <param name="personId" value="#{personHome.id}"/>
  </page>
</pages>

```

#####JSF#####

```

<h1>
  <h:outputText rendered="#{!personHome.managed}" value="Create Person"/>
  <h:outputText rendered="#{personHome.managed}" value="Edit Person"/>
</h1>
<h:form>
  <div>First name: <h:inputText value="#{person.firstName}"/></div>
  <div>Last name: <h:inputText value="#{person.lastName}"/></div>
  <div>
    <h:commandButton value="Create Person" action="#{personHome.persist}" rendered="#{!personHome.managed}" />
    <h:commandButton value="Update Person" action="#{personHome.update}" rendered="#{personHome.managed}" />
    <h:commandButton value="Delete Person" action="#{personHome.remove}" rendered="#{personHome.managed}" />
  </div>
</h:form>

```

#####"Create Person"##### personId #####"Edit Person"###

##### Person #####

```

<factory name="person"
  value="#{personHome.instance}"/>

<framework:entity-home name="personHome"
  entity-class="eg.Person"
  new-instance="#{newPerson}"/>

<component name="newPerson"
  class="eg.Person">
  <property name="nationality">#{country}</property>
</component>

```

#####

```

@Name("personHome")
public class PersonHome extends EntityHome<Person> {

    @In Country country;

    @Factory("person")
    public Person initPerson() { return getInstance(); }

    protected Person createInstance() {
        return new Person(country);
    }
}

```

###Country #####Home#####CountryHome#

##### PersonHome #####

```

@Name("personHome")
public class PersonHome extends EntityHome<Person> {

    @In Country country;

    @Factory("person")
    public Person initPerson() { return getInstance(); }
}

```

```

protected Person createInstance() {
    return new Person(country);
}

public void migrate()
{
    getInstance().setCountry(country);
    update();
}
}

```

```

##### persist()#update() # remove() #####Home#####
org.jboss.seam.afterTransactionSuccess ### #####
##### org.jboss.seam.afterTransactionSuccess.<name> ###<name>
#####

```

```
#####Home#####Faces#####
```

```

<factory name="person"
    value="#{personHome.instance}"/>

<framework:entity-home name="personHome"
    entity-class="eg.Person"
    new-instance="#{newPerson}">
    <framework:created-message>New person #{person.firstName} #{person.lastName} created</framework:created-message>
    <framework:deleted-message>Person #{person.firstName} #{person.lastName} deleted</framework:deleted-message>
    <framework:updated-message>Person #{person.firstName} #{person.lastName} updated</framework:updated-message>
</framework:entity-home>

<component name="newPerson"
    class="eg.Person">
    <property name="nationality">#{country}</property>
</component>

```

```
#####
```

```

@Name("personHome")
public class PersonHome extends EntityHome<Person> {

    @In Country country;

    @Factory("person")
    public Person initPerson() { return getInstance(); }

    protected Person createInstance() {
        return new Person(country);
    }

    protected String getCreatedMessage() { return "New person #{person.firstName} #{person.lastName} created"; }
    protected String getUpdatedMessage() { return "Person #{person.firstName} #{person.lastName} updated"; }
    protected String getDeletedMessage() { return "Person #{person.firstName} #{person.lastName} deleted"; }
}

```

```
#####Seam###resource bundle#####bundle## messages ##
```

```

Person_created=New person #{person.firstName} #{person.lastName} created
Person_deleted=Person #{person.firstName} #{person.lastName} deleted
Person_updated=Person #{person.firstName} #{person.lastName} updated

```

```
#####
```



```
##### <s:validateAll> # <s:decorate> #####
```

## 11.3. Query##

```
##### Person #####Query#####
```

```
<framework:entity-query name="people"
    ejbql="select p from Person p"/>
```

```
#####JSF#####
```

```
<h1>List of people</h1>
<h:dataTable value="#{people.resultList}" var="person">
    <h:column>
        <s:link view="/editPerson.jsp" value="#{person.firstName} #{person.lastName}">
            <f:param name="personId" value="#{person.id}"/>
        </s:link>
    </h:column>
</h:dataTable>
```

```
#####
```

```
<framework:entity-query name="people"
    ejbql="select p from Person p"
    order="lastName"
    max-results="20"/>
```

```
#####page#####
```

```
<pages>
    <page view-id="/searchPerson.jsp">
        <param name="firstResult" value="#{people.firstResult}"/>
    </page>
</pages>
```

```
#####JSF#####
```

```
<h1>Search for people</h1>
<h:dataTable value="#{people.resultList}" var="person">
    <h:column>
        <s:link view="/editPerson.jsp" value="#{person.firstName} #{person.lastName}">
            <f:param name="personId" value="#{person.id}"/>
        </s:link>
    </h:column>
</h:dataTable>

<s:link view="/search.xhtml" rendered="#{people.previousExists}" value="First Page">
    <f:param name="firstResult" value="0"/>
</s:link>

<s:link view="/search.xhtml" rendered="#{people.previousExists}" value="Previous Page">
    <f:param name="firstResult" value="#{people.previousFirstResult}"/>
</s:link>

<s:link view="/search.xhtml" rendered="#{people.nextExists}" value="Next Page">
    <f:param name="firstResult" value="#{people.nextFirstResult}"/>
</s:link>

<s:link view="/search.xhtml" rendered="#{people.nextExists}" value="Last Page">
    <f:param name="firstResult" value="#{people.lastFirstResult}"/>
</s:link>
```

#####Query#####“##”#####

```
<component name="examplePerson" class="Person"/>

<framework:entity-query name="people"
    ejbql="select p from Person p"
    order="lastName"
    max-results="20">
    <framework:restrictions>
        <value>lower(firstName) like lower( concat("#{examplePerson.firstName}','') )</value>
        <value>lower(lastName) like lower( concat("#{examplePerson.lastName}','') )</value>
    </framework:restrictions>
</framework:entity-query>
```

##“example”#####

```
<h1>Search for people</h1>
<h:form>
    <div>First name: <h:inputText value="#{examplePerson.firstName}"/></div>
    <div>Last name: <h:inputText value="#{examplePerson.lastName}"/></div>
    <div><h:commandButton value="Search" action="/search.jsp"/></div>
</h:form>

<h:dataTable value="#{people.resultList}" var="person">
    <h:column>
        <s:link view="/editPerson.jsp" value="#{person.firstName} #{person.lastName}">
            <f:param name="personId" value="#{person.id}"/>
        </s:link>
    </h:column>
</h:dataTable>
```

##### org.jboss.seam.afterTransactionSuccess #####

```
<event type="org.jboss.seam.afterTransactionSuccess">
    <action execute="#{people.refresh}" />
</event>
```

##### PersonHome #####

```
<event type="org.jboss.seam.afterTransactionSuccess.Person">
    <action execute="#{people.refresh}" />
</event>
```

#####Query#####

## 11.4. Controller##

Controller ##### EntityController#HibernateEntityController # BusinessProcessController  
 #Seam Application Framework#####  
 #####Seam#####

#####Seam##### RegisterAction#

```
@Stateless
@Name("register")
public class RegisterAction extends EntityController implements Register
{

    @In private User user;

    public String register()
    {
```

```
List existing = createQuery("select u.username from User u where u.username=:username")
    .setParameter("username", user.getUsername())
    .getResultList();

if ( existing.size()==0 )
{
    persist(user);
    info("Registered new user #{user.username}");
    return "/registered.jspx";
}
else
{
    addFacesMessage("User #{user.username} already exists");
    return null;
}
}
```

#####...

---

## Chapter 12. Seam#JBoss##

Seam####Seam###jBPM#####JBoss Rules#Drools#####

### 12.1. ####

##### org.drools.RuleBase #####Seam#####  
#####Seam#####class##### components.xml #####:

```
<drools:rule-base name="policyPricingRules">
  <drools:rule-files>
    <value>policyPricingRules</value>
  </drools:rule-files>
</drools:rule-base>
```

##### .dr1 #####Seam# APPLICATION ### ##### org.drools.RuleBase  
#####

#####Drools DSL#####DSL###

```
<drools:rule-base name="policyPricingRules" dsl-file="policyPricing.dsl">
  <drools:rule-files>
    <value>policyPricingRules</value>
  </drools:rule-files>
</drools:rule-base>
```

#####Drools#####

#####Drools#####BRMS##### components.xml ###:

```
<drools:rule-agent name="insuranceRules"
  configurationFile="/WEB-INF/deployedrules.properties" />
```

#####RulesAgent#####Drools#####

```
newInstance=true
url=http://localhost:8080/drools-jbrms/org.drools.brms.JBRMS/package/org.acme.insurance/fmeyer
localCacheDir=/Users/fernandomeyer/projects/jbossrules/drools-examples/drools-examples-brms/cache
poll=30
name=insuranceconfig
```

#####

```
<drools:rule-agent name="insuranceRules"
  url="http://localhost:8080/drools-jbrms/org.drools.brms.JBRMS/package/org.acme.insu
  local-cache-dir="/Users/fernandomeyer/projects/jbossrules/drools-examples/drools-ex
  poll="30"
  configuration-name="insuranceconfig" />
```

##### org.drools.WorkingMemory ##### WorkingMemory #####fact##

```
<drools:managed-working-memory name="policyPricingWorkingMemory" auto-create="true" rule-base="{poli
```

##### ruleBase ##### policyPricingWorkingMemory #####

### 12.2. #Seam#####

##### WorkingMemory #####Seam#####

```
@In WorkingMemory policyPricingWorkingMemory;

@In Policy policy;
@In Customer customer;

public void pricePolicy() throws FactException
{
    policyPricingWorkingMemory.assertObject(policy);
    policyPricingWorkingMemory.assertObject(customer);
    policyPricingWorkingMemory.fireAllRules();
}
```

## 12.3. #jBPM#####

#####jBPM##### — #####

```
<decision name="approval">

    <handler class="org.jboss.seam.drools.DroolsDecisionHandler">
        <workingMemoryName>orderApprovalRulesWorkingMemory</workingMemoryName>
        <assertObjects>
            <element>#{customer}</element>
            <element>#{order}</element>
            <element>#{order.lineItems}</element>
        </assertObjects>
    </handler>

    <transition name="approved" to="ship">
        <action class="org.jboss.seam.drools.DroolsActionHandler">
            <workingMemoryName>shippingRulesWorkingMemory</workingMemoryName>
            <assertObjects>
                <element>#{customer}</element>
                <element>#{order}</element>
                <element>#{order.lineItems}</element>
            </assertObjects>
        </action>
    </transition>

    <transition name="rejected" to="cancelled"/>

</decision>
```

<assertObjects> #####fact## WorkingMemory #####EL####

#####Seam#####jBPM#####Drools#

```
<task-node name="review">
    <task name="review" description="Review Order">
        <assignment handler="org.jboss.seam.drools.DroolsAssignmentHandler">
            <workingMemoryName>orderApprovalRulesWorkingMemory</workingMemoryName>
            <assertObjects>
                <element>#{actor}</element>
                <element>#{customer}</element>
                <element>#{order}</element>
                <element>#{order.lineItems}</element>
            </assertObjects>
        </assignment>
    </task>
    <transition name="rejected" to="cancelled"/>
    <transition name="approved" to="approved"/>
</task-node>
```

```
#####Drools    global#####jBPM    Assignable    ###assignable##Seam    Decision###decision##
##### decision.setOutcome("result") ##### Assignable #####id#
```

```
package org.jboss.seam.examples.shop

import org.jboss.seam.drools.Decision

global Decision decision

rule "Approve Order For Loyal Customer"
    when
        Customer( loyaltyStatus == "GOLD" )
        Order( totalAmount <= 10000 )
    then
        decision.setOutcome("approved");
    end
```

```
package org.jboss.seam.examples.shop

import org.jbpm.taskmgmt.exe.Assignable

global Assignable assignable

rule "Assign Review For Small Order"
    when
        Order( totalAmount <= 100 )
    then
        assignable.setPooledActors( new String[] {"reviewers"} );
    end
```

---

## Chapter 13. ##

Seam Security API#####Seam#####Seam#####

### 13.1. ##

Seam Security#####

- ##### - #####
- ##### - #####JBoss Rules#####

#### 13.1.1. #####

#####  
#####memory footprint#####  
#####

### 13.2. ##

####Seam Security#####jar##### application.xml ##### #####Seam Security## # #####

- drools-compiler-4.0.0.MR2.jar
- drools-core-4.0.0.MR2.jar
- janino-2.5.7.jar
- antlr-runtime-3.0.jar
- mvel14-1.2beta16.jar

####Web#####jboss-seam-ui.jar #####war####

### 13.3. ####

#####Seam Security##### Identity.setSecurityEnabled(false)  
#####

- Entity Security
- Hibernate Security Interceptor
- Seam Security Interceptor
- Page restrictions

## 13.4. ##

Seam Security#####JAAS#Java Authentication#Authorization Ser-  
vice#####API# #####Seam#####JAAS#####

### 13.4.1. ##

#####JAAS#####SeamLoginModule#####Seam###  
#####Seam#####  
components.xml ### identity ###

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:core="http://jboss.com/products/seam/core"
  xmlns:security="http://jboss.com/products/seam/security"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://jboss.com/products/seam/components http://jboss.com/products/seam/components-2.0.xsd
    http://jboss.com/products/seam/security http://jboss.com/products/seam/security-2.0.xsd"
  >
  <security:identity authenticate-method="#{authenticator.authenticate}"/>
</components>
```

#####Drools#JBoss Rules#jars#####classpath#####

EL### #{authenticator.authenticate} ##### authenticator ### authenticate #####

### 13.4.2. #####

# components.xml ## identity ### authenticate-method #####SeamLoginModule#####  
##### Identity.instance().getUsername() # Identity.  
instance().getPassword() ##### Identity.instance().addRole() ###  
###JavaBean#####

```
@Name("authenticator")
public class Authenticator {
    @In EntityManager entityManager;

    public boolean authenticate() {
        try
        {
            User user = (User) entityManager.createQuery(
                "from User where username = :username and password = :password")
                .setParameter("username", Identity.instance().getUsername())
                .setParameter("password", Identity.instance().getPassword())
                .getSingleResult();

            if (user.getRoles() != null)
            {
                for (UserRole mr : user.getRoles())
                    Identity.instance().addRole(mr.getName());
            }

            return true;
        }
        catch (NoResultException ex)
        {
            return false;
        }
    }
}
```



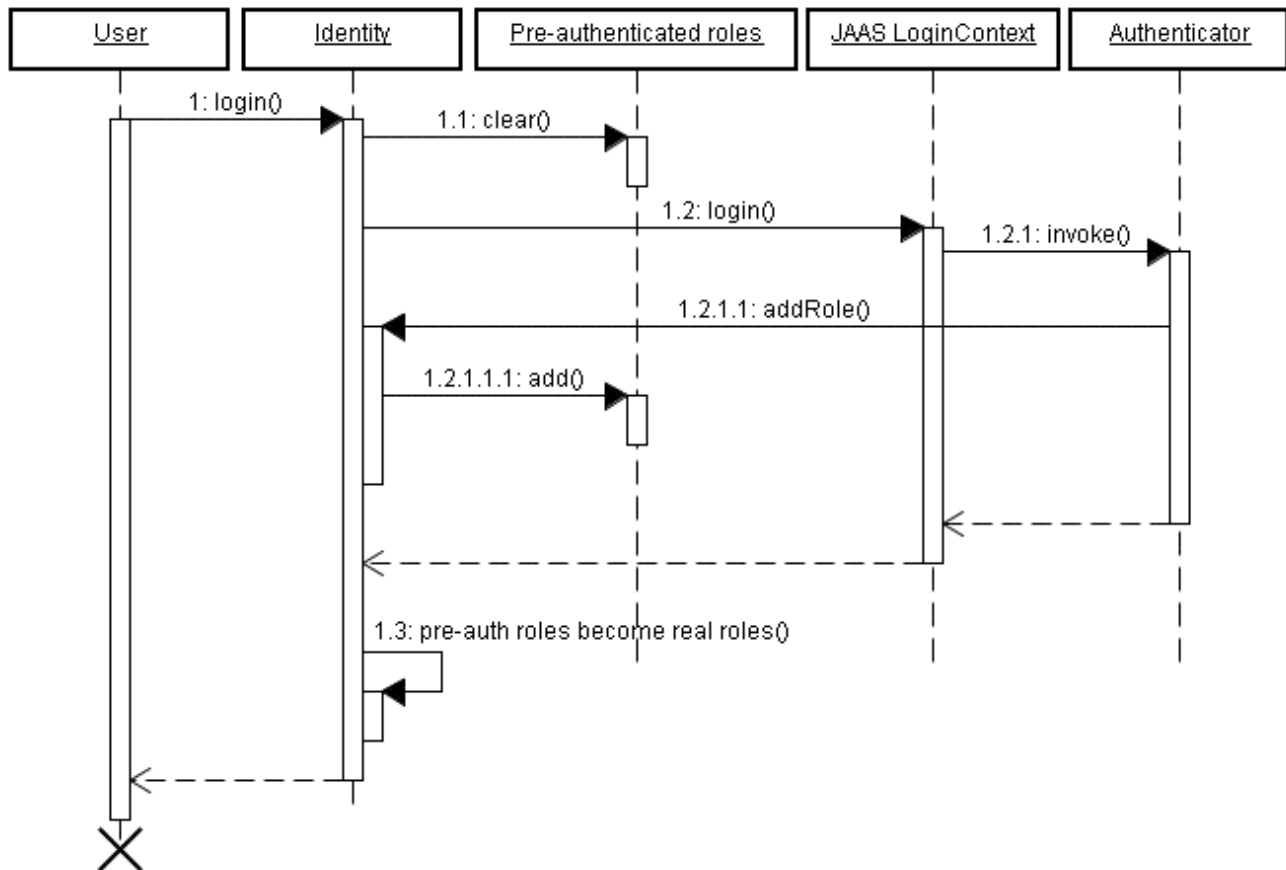
```
}

```

```
#####User # UserRole #####Bean# roles ##### Set
#####“admin”#“user”# ##### NoResultException#####false#####
```

## Identity.addRole()

```
Identity.addRole() ##### addRole() # ## #####
#####“###”##### Identity.hasRole() #####true##
#####
```



## 13.4.3. #####

```
Identity ##### username # password ##### identity
identity.login() #####
```

```

<div>
  <h:outputLabel for="name" value="Username"/>
  <h:inputText id="name" value="#{identity.username}"/>
</div>

<div>
  <h:outputLabel for="password" value="Password"/>
  <h:inputSecret id="password" value="#{identity.password}"/>
</div>

<div>
  <h:commandButton value="Login" action="#{identity.login}"/>
</div>

```

```
##### #{identity.logout} #####
```

### 13.4.4. #### - ##

```
#####
```

- # components.xml #####
- #####
- #####

### 13.4.5. #####

```
##### pages.xml #####“##”#####Security API#####
```

- NotLoggedInException - #####
- AuthorizationException - #####

```
#           NotLoggedInException           #####                      ##
Authorization##### ### pages.xml #####
```

```
<pages>

    ...

    <exception class="org.jboss.seam.security.NotLoggedInException">
        <redirect view-id="/login.xhtml">
            <message>You must be logged in to perform this action #####</message>
        </redirect>
    </exception>

    <exception class="org.jboss.seam.security.AuthorizationException">
        <end-conversation/>
        <redirect view-id="/security_error.xhtml">
            <message>You do not have the necessary security privileges to perform this action.#####</message>
        </redirect>
    </exception>

</pages>
```

```
###Web#####Seam#####
```

### 13.4.6. #####

```
#####id#####Seam#####
```

```
<pages login-view-id="/login.xhtml">

    <page view-id="/members/*" login-required="true"/>

    ...

</pages>
```

```
#####
```

```
#####
components.xml#####
#####
```

```
<event type="org.jboss.seam.notLoggedIn">
    <action execute="#{redirect.captureCurrentView}" />
</event>

<event type="org.jboss.seam.postAuthenticate">
    <action execute="#{redirect.returnToCapturedView}" />
</event>
```

```
##### authenticate() #####
```

### 13.4.7. HTTP##

```
#####Seam##HTTP Basic#HTTP Digest#RFC 2617 ##### authentication-fil-
ter #####components.xml####
```

```
<web:authentication-filter url-pattern="*.seam" auth-type="basic"/>
```

```
####Basic##### auth-type # basic ####Digest## ### auth-type # digest# #####Digest###key #
realm #####
```

```
<web:authentication-filter url-pattern="*.seam" auth-type="digest" key="AA3JK34aSDlkj" realm="My App" />
```

```
key ##### realm #####Realm####
```

### ##Digest##

```
#####Digest##### org.jboss.seam.security.digest.digestauthenticator #### validate-
Password() #####Digest#####
```

```
public boolean authenticate()
{
    try
    {
        User user = (User) entityManager.createQuery(
            "from User where username = :username")
            .setParameter("username", identity.getUsername())
            .getSingleResult();

        return validatePassword(user.getPassword());
    }
    catch (NoResultException ex)
    {
        return false;
    }
}
```

### 13.4.8. #####

```
####Security API#####
```

## #####JAAS##

#####Seam Security API#####JAAS#####JAAS##### components.xml ##### jaasCon-  
figName# #####JBoss AS##### ## #####JBoss AS### UsersRolesLoginModule ##### compon-  
ents.xml #####

```
<security:identity authenticate-method="{authenticator.authenticate}"
    jaas-config-name="other"/>
```

## 13.5. ####

Security API#####Key#### message.properties  
#####Key#####

**Table 13.1. #####Key**

org.jboss.seam.loginS uccessful	#####Security API#####
org.jboss.seam.loginF ailed	#####
org.jboss.seam.NotLog gedIn	#####

## 13.6. ##

Seam Security API#####  
##### components.xml ##### - #####

### 13.6.1. ####

Seam Security API#####  
#####  
#####

#####“admin”#“user”#“customer”##### name:action ##### cus-  
tomer:delete### customer:insert#

### 13.6.2. ####

##### @Restrict #####

#### @Restrict##

Seam##### @Restrict ##### @Restrict #####

```
##### Identity.checkRestriction() ##### @Restrict ####
@Restrict #####
```

```
## @Restrict ### componentName:methodName #####
```

```
@Name("account")
public class AccountAction {
    @Restrict public void delete() {
        ...
    }
}
```

```
##### delete() ##### account:delete# #####
@Restrict("#{s:hasPermission('account','delete',null)}")# #####
```

```
@Restrict @Name("account")
public class AccountAction {
    public void insert() {
        ...
    }
    @Restrict("#{s:hasRole('admin')}")
    public void delete() {
        ...
    }
}
```

```
##### @Restrict ##### @Restrict #####insert() #### ac-
count:insert ##### delete() ##### admin #####
```

```
##### #{s:hasRole()} #### s:hasRole # s:hasPermission ##EL##### Identity
##### #####Security API###EL#####
```

```
####EL####@Restrict #####Seam#####
```

```
@Name("account")
public class AccountAction {
    @In Account selectedAccount;
    @Restrict("#{s:hasPermission('account','modify',selectedAccount)}")
    public void modify() {
        selectedAccount.modify();
    }
}
```

```
##### hasPermission() ##### selectedAccount #### #####Seam#####
Identity ## hasPermission() ### ##### Account #####
```

```
####
```

```
##### @Restrict ##### Identity.checkRestriction() #####
```

```
public void deleteCustomer() {
    Identity.instance().checkRestriction("#{s:hasPermission('customer','delete',selectedCustomer)}");
}
```

```
##### true###
```

- ##### NotLoggedInException #####
- ##### AuthorizationException ###

```
###Java##### hasRole() # hasPermission() #####
```

```
if (!Identity.instance().hasRole("admin"))
    throw new AuthorizationException("Must be admin to perform this action");

if (!Identity.instance().hasPermission("customer", "create", null))
    throw new AuthorizationException("You may not create new customers");
```

### 13.6.3. #####

```
#####Seam Security#####
1#####2#####EL####
```

```
##### identity.isLoggedIn() #####
```

```
<h:form class="loginForm" rendered="#{not identity.loggedIn}">
```

```
##### - ##### manager #####
```

```
<h:outputLink action="#{reports.listManagerReports}" rendered="#{s:hasRole('manager')}">
    Manager Reports
</h:outputLink>
```

```
##### manager #####outputLink##### rendered ##### <s:div> ##
<s:span> ####
```

```
##### h:dataTable ##### s:hasPermission
EL#####dataTable#####
```

```
<h:dataTable value="#{clients}" var="cl">
    <h:column>
        <f:facet name="header">Name</f:facet>
        #{cl.name}
    </h:column>
    <h:column>
        <f:facet name="header">City</f:facet>
        #{cl.city}
    </h:column>
    <h:column>
        <f:facet name="header">Action</f:facet>
        <s:link value="Modify Client" action="#{clientAction.modify}"
            rendered="#{s:hasPermission('client','modify',cl)}/>
        <s:link value="Delete Client" action="#{clientAction.delete}"
            rendered="#{s:hasPermission('client','delete',cl)}/>
    </h:column>
</h:dataTable>
```

### 13.6.4. ####

```
##### pages.xml ##### page ##### <restrict/> ### ##### restrict
#####non-faces#GET##### /viewId.xhtml:render ##### JSF post-
back##### /viewId.xhtml:restore ### #####
```

```
<page view-id="/settings.xhtml">
    <restrict/>
</page>
```

```
##### /settings.xhtml:render ###non-faces##### /settings.xhtml:restore ###faces#####
```

```
<page view-id="/reports.xhtml">
  <restrict>#{s:hasRole('admin')}}</restrict>
</page>
```

#####faces#non-faces##### admin #####

### 13.6.5. ####

Seam Security#####

##### @Restrict ###

```
@Entity
@Name("customer")
@Restrict
public class Customer {
    ...
}
```

##### @Restrict ##### entityName:action ##### #####entityName  
####Seam#####@Name##### action ### read#insert#update ## delete#

#####@Restrict#####

- @PostLoad - ##### read ###
- @PrePersist - ##### insert ###
- @PreUpdate - ##### update ###
- @PreRemove - #####delete###

##### insert #####

```
@PrePersist @Restrict
public void prePersist() {}
```

#####seamspace##### MemberBlog ###  
##### MemberBlog##

```
rule InsertMemberBlog
  no-loop
  activation-group "permissions"
when
  check: PermissionCheck(name == "memberBlog", action == "insert", granted == false)
  Principal(principalName : name)
  MemberBlog(member : member -> (member.getUsername().equals(principalName)))
then
  check.grant();
end;
```

##### memberBlog:insert##### Principal fact#####Blog##### ### Principal  
fact#####“name : name”##### - ## Principal# name ##### name #####  
##### Principal ##### JBoss Rules###

#####Seam Security###JPA#####

**##JPA####**

#EJB3##Bean##### EntityListener ##### META-INF/orm.xml #####

```
<?xml version="1.0" encoding="UTF-8"?>
<entity-mappings xmlns="http://java.sun.com/xml/ns/persistence/orm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence/orm http://java.sun.com/xml/ns/persistence/orm-2.0.xsd"
  version="1.0">

  <persistence-unit-metadata>
    <persistence-unit-defaults>
      <entity-listeners>
        <entity-listener class="org.jboss.seam.security.EntitySecurityListener"/>
      </entity-listeners>
    </persistence-unit-defaults>
  </persistence-unit-metadata>

</entity-mappings>
```

**##Hibernate####**

#####Seam###Hibernate SessionFactory#####

**13.7. #####**

##### Seam#####

**13.7.1. ####**

Security API##### customer:modify ### Seam Security#####JBoss Rules#####

#####1##### 2### - JBoss Rules#####

**13.7.2. #####**

Seam Security##### securityRules # RuleBase ###Seam Security##### ## components.xml

#####

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:core="http://jboss.com/products/seam/core"
  xmlns:security="http://jboss.com/products/seam/security"
  xmlns:drools="http://jboss.com/products/seam/drools"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://jboss.com/products/seam/core http://jboss.com/products/seam/core-2.0.xsd
    http://jboss.com/products/seam/components http://jboss.com/products/seam/components-2.0.xsd
    http://jboss.com/products/seam/drools http://jboss.com/products/seam/drools-2.0.xsd
    http://jboss.com/products/seam/security http://jboss.com/products/seam/security-2.0.xsd"
  >

  <drools:rule-base name="securityRules">
    <drools:rule-files>
      <value>META-INF/security.drl</value>
    </drools:rule-files>
  </drools:rule-base>

</components>
```

##### RuleBase #####



### 13.7.3. #####

```
#####jar### /META-INF ##### security.drl #####
##### components.xml #####
```

```
#####JBoss Rules#####
```

```
package MyApplicationPermissions;

import org.jboss.seam.security.PermissionCheck;
import org.jboss.seam.security.Role;

rule CanUserDeleteCustomers
when
    c: PermissionCheck(name == "customer", action == "delete")
    Role(name == "admin")
then
    c.grant();
end;
```

```
#####JBoss Rules##### # #####
```

```
##### PermissionCheck # Role #####
```

```
#####
CanUserDeleteCustomers#####
```

```
#####LHS(#####RHS##### LHS##### LHS#
when #####RHS#####LHS##### RHS# then ##### end; ####
```

```
#####LHS#####
```

```
c: PermissionCheck(name == "customer", action == "delete")
```

```
##### PermissionCheck #####"customer"### name #####"delete"### action ###
##### hasPermission() ##### PermissionCheck
##### Fact ##### ## PermissionCheck ##### hasPermission("account", "cre-
ate", null)#####"account"# name #####"create"# action # PermissionCheck
#####
```

```
##### fact#####
##### java.security.Principal ### #####
org.jboss.seam.security.Role# ##### ((RuleBasedIdentity) RuleBasedIden-
tity.instance()).getSecurityContext().insert()# #####fact#####
```

```
#####LHS##### c: #### #####LHS#####
```

```
Role(name == "admin")
```

```
##### Role #####"admin"# name# #####fact#####
"##### customer:delete ##### admin #####"
```

```
#####RHS#
```

```
c.grant()
```

```
RHS#Java##### c ### grant() ### ##### PermissionCheck ##### ## Permis-
```

```
sionCheck ### name # action ##### granted ##### false# # PermissionCheck ### grant()
### ## granted ### true#####
```

#####

```
##### PermissionCheck ## action ### #####
```

```
rule CanDoAnythingToCustomersIfYouAreAnAdmin
when
  c: PermissionCheck(name == "customer")
  Role(name == "admin")
then
  c.grant();
end;
```

```
##### admin ##### customer #####
```

## 13.8. SSL##

```
Seam#####HTTPS##### pages.xml ##### scheme #### /login.xhtml
#####HTTPS#
```

```
<page view-id="/login.xhtml" scheme="https">
```

```
##### s:link # s:button JSF##### view #####HTTPS##### /
login.xhtml #####HTTPS#
```

```
<s:link view="/login.xhtml" value="Login"/>
```

```
## ## ##### ## ##### scheme="https"
##HTTP#####HTTPS#####
```

```
##### ##scheme#####HTTPS###
#####scheme#####scheme#
#####HTTPS#####HTTPS#####
scheme##### "*" #####
```

```
<page view-id="*" scheme="http" />
```

```
##### ## #####HTTPS#####scheme#
```

```
#####Seam#####scheme#####HTTP##### components.xml #####
```

```
<core:servlet-session invalidate-on-scheme-change="true"/>
```

```
#####Session ID#####HTTPS##HTTP#####
```

## 13.9. ##Captcha##

```
#####Security API#####blog##### ##Captcha#Completely Auto-
mated Public Turing test to tell Computers and Humans Apart#####
Seam###JCaptcha#####Captcha challenge#####Captcha#####Seam
lib###jcaptcha-* jar##### application.xml #####Java###
```

### 13.9.1. ##Captcha Servlet

#####Seam Resource Servlet#####Captcha challenge##### web.xml #####

```
<servlet>
  <servlet-name>Seam Resource Servlet</servlet-name>
  <servlet-class>org.jboss.seam.servlet.SeamResourceServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>Seam Resource Servlet</servlet-name>
  <url-pattern>/seam/resource/*</url-pattern>
</servlet-mapping>
```

### 13.9.2. ##Captcha###

####Captcha challenge##### Seam####page-scoped###captcha#####captcha#####

```
<div>
  <h:graphicImage value="/seam/resource/captcha?#{captcha.id}"/>
</div>

<div>
  <h:outputLabel for="verifyCaptcha">Enter the above letters</h:outputLabel>
  <h:inputText id="verifyCaptcha" value="#{captcha.response}" required="true">
    <s:validate />
  </h:inputText>
  <div class="validationError"><h:message for="verifyCaptcha"/></div>
</div>

<div>
  <h:commandButton action="#{register.next}" value="Register"/>
</div>
```

#####graphicImage ####Captcha challenge#inputText ##### #####Captcha#####

### 13.9.3. ##Captcha##

Captcha##### ImageCaptchaService #####DefaultManageableImageCaptchaService##### ##### Im-  
ageCaptchaService## components.xml #####

```
<component name="org.jboss.seam.captcha.captchaImage" service="#{customCaptcha.service}"/>
```

service ##### ImageCaptchaService #####Captcha### ##### ImageCaptchaService  
#####JCaptcha### #####seamspace#####

```
@Name("customCaptcha")
public class CustomCaptcha
{
  public ImageCaptchaService getService()
  {
    BasicGimpyEngine customCaptcha = new BasicGimpyEngine();
    GimpyFactory factory = new GimpyFactory(
      new RandomWordGenerator("ABCDEFGHIJKLMNOPQRSTUVWXYZ23456789"),
      new ComposedWordToImage(new RandomFontGenerator(new Integer(15),
        new Integer(15)), new UniColorBackgroundGenerator(new Integer(150),
          new Integer(30)), new RandomTextPaster(new Integer(4),
            new Integer(7), Color.BLACK)));
    GimpyFactory[] factories = {factory};
    customCaptcha.setFactories(factories);

    return new DefaultManageableImageCaptchaService(
```

```
        new FastHashMapCaptchaStore(),  
        customCaptcha,  
        180,  
        120000,  
        75000);  
    }  
}
```

# Chapter 14. #####

Seam#####UI#####

## 14.1. ###

##### java.util.Locale ##### locale ##### locale#Seam  
##JSF#####locale#

- ##HTTP#####locale#####locale# faces-config.xml #####locale#
- ##### faces-config.xml #####locale# #####locale#
- #####locale#

##Seam#####locale### ##### org.jboss.seam.international.localeSelector.language#  
org.jboss.seam.international.localeSelector.country#  
org.jboss.seam.international.localeSelector.variant#####

#####locale#####Seam#####locale#####  
#JSP#Facelet#Form#####

```
<h:selectOneMenu value="#{localeSelector.language}">
  <f:selectItem itemLabel="English" itemValue="en"/>
  <f:selectItem itemLabel="Deutsch" itemValue="de"/>
  <f:selectItem itemLabel="Francais" itemValue="fr"/>
</h:selectOneMenu>
<h:commandButton action="#{localeSelector.select}" value="#{messages['ChangeLanguage']}" />
```

##### faces-config.xml #####locale#####

```
<h:selectOneMenu value="#{localeSelector.localeString}">
  <f:selectItems value="#{localeSelector.supportedLocales}" />
</h:selectOneMenu>
<h:commandButton action="#{localeSelector.select}" value="#{messages['ChangeLanguage']}" />
```

#####Seam#JSF#locale#####

## 14.2. ##

JSF ##### <f:loadBundle /> #####Seam##### Seam# messages  
#####EL#####

### 14.2.1. ####

Seam##### java.util.ResourceBundle ##org.jboss.seam.core.resourceBundle #####  
#####Seam #####messages##### messages.properties#mes-  
sages\_en.properties#messages\_en\_AU.properties ##### WEB-INF/classes #####

##### messages\_en.properties##

Hello=Hello

```
## messages_en_AU.properties##
```

```
Hello=G'day
```

```
#####Seam##### org.jboss.seam.core.resourceLoader.bundleNames #####
#####
```

```
<core:resource-loader>
  <core:bundle-names>
    <value>mycompany_messages</value>
    <value>standard_messages</value>
  </core:bundle-names>
</core:resource-loader>
```

```
#####JSF View id##### / ##### /welcome/hello.jsp
##### welcome/hello_en.properties ##
```

```
##### pages.xml #####
```

```
<page view-id="/welcome/hello.jsp" bundle="HelloMessages"/>
```

```
##### /welcome/hello.jsp ##### HelloMessages.properties #####
```

## 14.2.2. ####

```
####Seam##### <f:loadBundle ... /> #####
```

```
<h:outputText value="#{messages['Hello']}" />
```

```
###
```

```
<h:outputText value="#{messages.Hello}" />
```

```
#####message#####EL####
```

```
Hello=Hello, #{user.firstName} #{user.lastName}
```

```
Hello=G'day, #{user.firstName}
```

```
#####
```

```
@In private Map<String, String> messages;
```

```
@In("#{messages['Hello']}") private String helloMessage;
```

## 14.2.3. Faces Messages

```
facesMessages #####Faces Messages####
```

```
@Name("hello")
@Stateless
public class HelloBean implements Hello {
    @In FacesMessages facesMessages;

    public String sayIt() {
```

```

        facesMessages.addFromResourceBundle("Hello");
    }
}

```

#####locale## Hello, Gavin King ## G'day, Gavin#

## 14.3. ##

```

Seam#####session### java.util.Timezone ##### org.jboss.seam.international.timezone# #####
org.jboss.seam.international.timezoneSelector #####
#####JSF#####UTC #####UTC##### <f:convertDateTime> #####
Seam#####Seam#####Seam### <s:convertDateTime> #####Seam #####

```

## 14.4. ##

Seam#####Theme API####API#####

#####

```

<theme:theme-selector cookie-enabled="true">
    <theme:available-themes>
        <value>default</value>
        <value>accessible</value>
        <value>printable</value>
    </theme:available-themes>
</theme:theme-selector>

```

#####

#####default #####default.properties## default.properties#####

```

css ../screen.css
template /template.xhtml

```

#####CSS#####facelet#####

#####JSP##Facelet#####Facelet#####

```

<link href="#{theme.css}" rel="stylesheet" type="text/css" />

```

#####

```

<link href="#{facesContext.externalContext.requestContextPath}#{theme.css}"
    rel="stylesheet" type="text/css" />

```

#####Facelet##### <ui:composition> #####:

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    template="#{theme.template}">

```

##locale#####

```
<h:selectOneMenu value="#{themeSelector.theme}">
  <f:selectItems value="#{themeSelector.themes}" />
</h:selectOneMenu>
<h:commandButton action="#{themeSelector.select}" value="Select Theme" />
```

## 14.5. ##cookie##locale####

locale#####cookie#####components.xml### cookie-enabled #####

```
<theme:theme-selector cookie-enabled="true">
  <theme:available-themes>
    <value>default</value>
    <value>accessible</value>
    <value>printable</value>
  </theme:available-themes>
</theme:theme-selector>

<international:locale-selector cookie-enabled="true" />
```



---

# Chapter 15. Seam Text

```
#####Wiki#####Blog##### Seam### <s:formattedText/> #####  
Seam Text ##### Seam Text#####ANTLR#####ANTLR#####
```

## 15.1. #####

```
#####
```

```
It's easy to make *bold text*, /italic text/, |monospace|,  
~deleted text~, superscripts or underlines.
```

```
##### <s:formattedText/> #####HTML#
```

```
<p>  
It's easy to make <b>bold text</b>, <i>italic text</i>, <tt>monospace</tt>  
<del>deleted text</del>, super<sup>scripts</sup> or <u>underlines</u>.  
</p>
```

```
##### + #####
```

```
+This is a big heading  
You /must/ have some text following a heading!  
  
++This is a smaller heading  
This is the first paragraph. We can split it across multiple  
lines, but we must end it with a blank line.  
  
This is the second paragraph.
```

```
#####HTML####
```

```
<h1>This is a big heading</h1>  
<p>  
You <i>must</i> have some text following a heading!  
</p>  
  
<h2>This is a smaller heading</h2>  
<p>  
This is the first paragraph. We can split it across multiple  
lines, but we must end it with a blank line.  
</p>  
  
<p>  
This is the second paragraph.  
</p>
```

```
# # ##### = ###
```

```
An ordered list:  
  
#first item  
#second item  
#and even the /third/ item  
  
An unordered list:  
  
=an item  
=another item
```

```

<p>
An ordered list:
</p>

<ol>
<li>first item</li>
<li>second item</li>
<li>and even the <i>third</i> item</li>
</ol>

<p>
An unordered list:
</p>

<ul>
<li>an item</li>
<li>another item</li>
</ul>

```

#####

```

The other guy said:

"Nyeah nyeah-nee
/nyeah/ nyeah!"

But what do you think he means by "nyeah-nee"?

```

```

<p>
The other guy said:
</p>

<q>Nyeah nyeah-nee
<i>nyeah</i> nyeah!</q>

<p>
But what do you think he means by <q>nyeah-nee</q>?
</p>

```

## 15.2. #####

# \*#| # # ##### <#> # & ###HTML##### \####

```

You can write down equations like 2\*3\=6 and HTML tags
like \<body> using the escape character: \\.

```

```

<p>
You can write down equations like 2*3=6 and HTML tags
like &lt;body> using the escape character: \.
</p>

```

#####`#####

```

My code doesn't work:

`for (int i=0; i<100; i--)
{
    doSomething();
}`

Any ideas?

```

```
<p>
My code doesn't work:
</p>

<pre>for (int i=0; i<100; i--)
{
    doSomething();
}</pre>

<p>
Any ideas?
</p>
```

#####

```
This is a |<tag attribute="value"/>| example.
```

#####

## 15.3. ##

#####

```
Go to the Seam website at [=>http://jboss.com/products/seam].
```

#####

```
Go to [the Seam website=>http://jboss.com/products/seam].
```

#####Seam Text#####Wiki###

## 15.4. ##HTML

#####HTML#####

```
You might want to link to <a href="http://jboss.com/products/seam">something
cool</a>, or even include an image: 
```

#####

```
<table>
  <tr><td>First name:</td><td>Gavin</td></tr>
  <tr><td>Last name:</td><td>King</td></tr>
</table>
```

#####

# Chapter 16. iText PDF##

Seam#####iText#####Seam#iText#####PDF#####RTF#####

## 16.1. ##PDF##

iText### jboss-seam-pdf.jar ### ##JAR#####PDF####iText  
JSF#####DocumentStore### #####PDF##### jboss-seam-pdf.jar #iText JAR  
##### WEB-INF/lib #### ##Seam#iText#####

Seam iText#####Facelets#####JSP#####seam-ui##

## examples/itext #####PDF#####PDF#####

### 16.1.1. #####

<p>&lt;p:document&gt;</p>	<p>##</p> <p>#####facelets##### http://jboss.com/products/seam/pdf #####</p> <p>##### document ##### document</p> <p>###Seam#####HTML#####</p> <p>##</p> <ul style="list-style-type: none"><li>• type — ##### ####PDF#RTF # HTML ### Seam###PDF#####PDF#####</li><li>• ##### LETTER #A4# ##### com.lowagie.text.PageSize #### #####pageSize#####"612 792"#LETTER#####</li><li>• orientation — ##### #### portrait # landscape#####</li><li>• margins — #####</li><li>• marginMirroring — #####</li></ul> <p>Metadata##</p> <ul style="list-style-type: none"><li>• ##</li><li>• ##</li><li>• ###</li><li>• ##</li><li>• ###</li></ul> <p>##</p> <div data-bbox="491 2101 1434 2145"><p>&lt;p:document xmlns:p="http://jboss.com/products/seam/pdf"&gt;</p></div>
---------------------------	--

	<pre> The document goes here.##### &lt;/p:document&gt; </pre>
--	---

## 16.1.2. #####

#####UI#####HTML#####PDF### ###Seam#####UI#####PDF### #  
 <p:image> # <p:paragraph> ##### # <p:font> #####

<p:paragraph>	<pre> ##  #####  ##  • firstLineIndent ###  • extraParagraphSpace #####  • leading ##  • multipliedLeading ###  • spacingBefore — #####  • spacingAfter — #####  • indentationLeft ###  • indentationRight ###  • keepTogether ###  ##  &lt;p:paragraph alignment="justify"&gt;   This is a simple document.  It isn't very fancy. ##### &lt;/p:paragraph&gt; </pre>
---------------	--

<p:text>	<pre> ##  text #####JSF#####HTML##### out- putText ###  ##  • value — #####  ##  &lt;p:paragraph&gt; </pre>
----------	---

	<pre> The item costs #####&lt;p:text value="{product.price}"&gt;   &lt;f:convertNumber type="currency" currencySymbol="\$"/&gt; &lt;/p:text&gt; &lt;/p:paragraph&gt; </pre>
--	---

<p:font>	<pre> ##  font#####  ##  • name — ##### COURIER#HELVETICA# TIMES-ROMAN#SYMBOL ## ZAPFDING-   BATS#  • size — ####  • style — ##### NORMAL#BOLD# ITALIC#OBLIQUE#   UNDERLINE#LINE-THROUGH  • encoding — #####  ##  &lt;p:font family="courier" style="bold" size="24"&gt;   &lt;p:paragraph&gt;My Title&lt;/p:paragraph&gt; &lt;/p:font&gt; </pre>
----------	---

<p:newPage>	<pre> ##  p:newPage #####  ##  &lt;p:newPage /&gt; </pre>
-------------	---

<p:image>	<pre> ##  p:image ##### ##value ###classpath##Web#####  ##### imageData ##### java.awt.Image #####  ##  • value — #####  • rotation — #####  • height — ##### </pre>
-----------	--

	<ul style="list-style-type: none"> <li>• width — #####</li> <li>• alignment — ##### Section 16.1.7.2, “#####”#</li> <li>• alt — #####</li> <li>• indentationLeft ###</li> <li>• indentationRight ###</li> <li>• spacingBefore — #####</li> <li>• spacingAfter — #####</li> <li>• widthPercentage #####</li> <li>• initialRotation ###</li> <li>• dpi ##</li> <li>• scalePercent##### — ##### #####X###Y#####</li> <li>• wrap</li> <li>• underlying ###</li> </ul> <p>##</p> <div>&lt;p:image value="/jboss.jpg" /&gt;</div> <div>&lt;p:image value="#{images.chart}" /&gt;</div>
--	--

<p:anchor>	<p>##</p> <p>p:anchor #####. #####</p> <p>##</p> <ul style="list-style-type: none"> <li>• name — #####</li> <li>• reference — #####"###"###"link1" # link1 ##### name #####URL###</li> </ul> <p>##</p> <div> <p>&lt;p:listItem&gt;&lt;p:anchor reference="#reason1"&gt;Reason 1&lt;/p:anchor&gt;&lt;/p:listItem&gt;</p> <p>...</p> <p>&lt;p:paragraph&gt;</p> <p>    &lt;p:anchor name="reason1"&gt;It's the quickest way to get "rich"&lt;/p:anchor&gt;</p> <p>    ...</p> <p>&lt;/p:paragraph&gt;</p> </div>
------------	--

### 16.1.3. #####

<p:header>	##
<p:footer>	<p>p:header # p:footer</p> <p>#####</p> <p>##</p> <ul style="list-style-type: none"> <li>• alignment — ##### Section 16.1.7.2, “#####”#</li> <li>• backgroundColor — ##### Section 16.1.7.1, “###”#</li> <li>• borderColor — ##### borderColorLeft#borderColorRight#borderColorTop # borderColorBottom##### Section 16.1.7.1, “###”#</li> <li>• borderWidth — ##### borderWidthLeft#borderWidthRight#borderWidthTop # borderWidthBottom####</li> </ul> <p>##</p> <pre>&lt;p:facet name="header"&gt;   &lt;p:font size="12"&gt;     &lt;p:footer borderWidthTop="1" borderColorTop="blue"       borderWidthBottom="0" alignment="center"&gt;       Why Seam? [&lt;p:pageNumber /&gt;]     &lt;/p:footer&gt;   &lt;/p:font&gt; &lt;/f:facet&gt;</pre>

<p:pageNumber>	<p>##</p> <p>## p:pageNumber #####</p> <p>##</p> <pre>&lt;p:footer borderWidthTop="1" borderColorTop="blue"   borderWidthBottom="0" alignment="center"&gt;   Why Seam? [&lt;p:pageNumber /&gt;] &lt;/p:footer&gt;</pre>
----------------	---

### 16.1.4. ##

<p:chapter>	##
<p:section>	<p>#####p:chapter # p:section #####</p> <p>##### PDF#####</p> <p>##</p>



	<ul style="list-style-type: none"> <li>• alignment — ##### Section 16.1.7.2, “#####”#</li> <li>• number — #####</li> <li>• numberDepth — ##### #####3###3.1.4##### #####2#####1.4#</li> </ul> <p>##</p> <pre> &lt;p:document xmlns:p="http://jboss.com/products/seam/pdf"             title="Hello"&gt;      &lt;p:chapter number="1"&gt;         &lt;p:title&gt;&lt;p:paragraph&gt;Hello&lt;/p:paragraph&gt;&lt;/p:title&gt;         &lt;p:paragraph&gt;Hello #{user.name}!&lt;/p:paragraph&gt;     &lt;/p:chapter&gt;      &lt;p:chapter number="2"&gt;         &lt;p:title&gt;&lt;p:paragraph&gt;Goodbye&lt;/p:paragraph&gt;&lt;/p:title&gt;         &lt;p:paragraph&gt;Goodbye #{user.name}.&lt;/p:paragraph&gt;     &lt;/p:chapter&gt;  &lt;/p:document&gt; </pre>
--	---

<p:header>	<p>##</p> <p>##### p:title #####</p> <p>p:paragraph ###</p>
------------	---

## 16.1.5. ##

## p:list # p:listItem ##### ui:repeat  
#####Seam#####

```

<p:document xmlns:p="http://jboss.com/products/seam/pdf"
            xmlns:ui="http://java.sun.com/jsf/facelets"
            title="Hello">
    <p:list style="numbered">
        <ui:repeat value="#{documents}" var="doc">
            <p:listItem>#{doc.name}</p:listItem>
        </ui:repeat>
    </p:list>
</p:document>

```

<p:list>	<p>##</p> <ul style="list-style-type: none"> <li>• style — ##### NUMBERED#LETTERED# GREEK#ROMAN# ZAPFDINGBATS#ZAPFDINGBATS_NUMBER# #####</li> <li>• listSymbol — #####</li> <li>• indent — #####</li> </ul>
----------	---

	<ul style="list-style-type: none"> <li>• lowerCase — #####</li> <li>• charNumber — ##ZAPFDINGBATS#####</li> <li>• numberType — ##ZAPFDINGBATS_NUMBER#####</li> </ul> <p>##</p> <pre>&lt;p:list style="numbered"&gt;   &lt;ui:repeat value="#{documents}" var="doc"&gt;     &lt;p:listItem&gt;#{doc.name}&lt;/p:listItem&gt;   &lt;/ui:repeat&gt; &lt;/p:list&gt;</pre>
--	--

<p:listItem>	<p>##</p> <p>p:listItem #####</p> <p>##</p> <ul style="list-style-type: none"> <li>• alignment — ##### Section 16.1.7.2, “#####”#</li> <li>• alignment — ##### Section 16.1.7.2, “#####” #</li> <li>• indentationLeft — #####</li> <li>• indentationRight — #####</li> <li>• listSymbol — #####</li> </ul> <p>##</p> <p>...</p>
--------------	---

### 16.1.6. ##

##### p:table # p:cell ##### #####3#####3#####  
#####

<p:table>	<p>##</p> <p>p:table #####</p> <p>##</p> <ul style="list-style-type: none"> <li>• columns — #####</li> <li>• widths — ##### ##widths="2 1"#####3#####</li> </ul>
-----------	--

	<ul style="list-style-type: none"> <li>• headerRows — #####</li> <li>• footerRows — ##### headerRows ## #####2#####1##### headerRows #####3#footerRows #####1</li> <li>• widthPercentage — #####</li> <li>• horizontalAlignment — ##### Section 16.1.7.2, “#####”#</li> <li>• skipFirstHeader</li> <li>• runDirection</li> <li>• lockedWidth</li> <li>• splitRows</li> <li>• spacingBefore — #####</li> <li>• spacingAfter — #####</li> <li>• extendLastRow</li> <li>• headersInEvent</li> <li>• splitLate</li> <li>• keepTogether</li> </ul> <p>##</p> <pre>&lt;p:table columns="3" headerRows="1"&gt;   &lt;p:cell&gt;name&lt;/p:cell&gt;   &lt;p:cell&gt;owner&lt;/p:cell&gt;   &lt;p:cell&gt;size&lt;/p:cell&gt;   &lt;ui:repeat value="#{documents}" var="doc"&gt;     &lt;p:cell&gt;#{doc.name}&lt;/p:cell&gt;     &lt;p:cell&gt;#{doc.user.name}&lt;/p:cell&gt;     &lt;p:cell&gt;#{doc.size}&lt;/p:cell&gt;   &lt;/ui:repeat&gt; &lt;/p:table&gt;</pre>
--	---

<p:cell>	<p>##</p> <p>p:cell #####</p> <p>##</p> <ul style="list-style-type: none"> <li>• colspan — ##### colspan ###1#####</li> <li>• horizontalAlignment — ##### Section 16.1.7.2, “#####”#</li> <li>• verticalAlignment — ##### Section 16.1.7.2, “#####”#</li> <li>• padding — ##### paddingLeft#paddingRight# padding-</li> </ul>
----------	---

	<p>Top # paddingBottom#</p> <ul style="list-style-type: none"> <li>• useBorderPadding</li> <li>• leading</li> <li>• multipliedLeading</li> <li>• indent ##</li> <li>• verticalAlignment ####</li> <li>• extraParagraphSpace #####</li> <li>• fixedHeight ####</li> <li>• noWrap</li> <li>• minimumHeight ####</li> <li>• followingIndent ####</li> <li>• rightIndent ###</li> <li>• spaceCharRatio #####</li> <li>• runDirection ####</li> <li>• arabicOptions #####</li> <li>• useAscender ##</li> <li>• grayFill ####</li> <li>• rotation ###</li> </ul> <p>##</p> <div>&lt;p:cell&gt;...&lt;/p:cell&gt;</div>
--	--

### 16.1.7. ####

#####

#### 16.1.7.1. ###

Seam #####white#gray# lightgray#darkgray#black# red#pink#yellow#  
green#magenta#cyan # blue#

#### 16.1.7.2. #####

#####Seam PDF##### left#right#center# justify # justifyall#  
#####top#middle#bottom # baseline#

## 16.1.8. iText##

#####

#####URL##PDF###/seam-doc.seam# #####PDF##### /myDocument.pdf####  
#####PDF#####\*.pdf#####Seam Servlet####DocumentStoreServlet#

```
<filter>
  <filter-name>Seam Servlet Filter</filter-name>
  <filter-class>org.jboss.seam.servlet.SeamServletFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>Seam Servlet Filter</filter-name>
  <url-pattern>*.pdf</url-pattern>
</filter-mapping>

<servlet>
  <servlet-name>Document Store Servlet</servlet-name>
  <servlet-class>org.jboss.seam.pdf.DocumentStoreServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>Document Store Servlet</servlet-name>
  <url-pattern>*.pdf</url-pattern>
</servlet-mapping>
```

##### useExtensions #####

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:pdf="http://jboss.com/products/seam/pdf">
  <pdf:documentStore useExtensions="true" />
</components>
```

##### documentStore# errorPage #####

```
<pdf:documentStore useExtensions="true" errorPage="/pdfMissing.seam" />
```

## 16.2. ##

<p>&lt;p:barchart&gt;</p>	<p>##</p> <p>#####</p> <p>##</p> <ul style="list-style-type: none"> <li>• borderVisible — #####</li> <li>• borderPaint — #####</li> <li>• borderBackgroundPaint — #####</li> <li>• borderStroke —</li> <li>• domainAxisLabel — #####</li> <li>• domainAxisPaint — #####</li> </ul>
---------------------------	--

- domainGridlinesVisible— #####
- domainGridlinePaint— #####
- domainGridlineStroke — #####
- height — #####
- width — #####
- is3D — #####3D###2D#####
- legend — #####
- legendItemPaint— #####
- legendItemBackgroundPaint— #####
- orientation — ##### vertical##### horizontal#
- plotBackgroundPaint— #####
- plotBackgroundAlpha— #####alpha#####  
####0#####1#####
- plotForegroundAlpha— #####alpha#####  
####0#####1#####
- plotOutlinePaint— #####
- plotOutlineStroke — #####
- rangeAxisLabel — #####
- rangeAxisPaint — #####
- rangeGridlinesVisible— #####
- rangeGridlinePaint— #####
- rangeGridlineStroke — #####
- title — #####
- titlePaint— #####
- titleBackgroundPaint— #####
- width — #####

##

```

<p:barchart title="Bar Chart" legend="true"
  width="500" height="500">
  <p:series key="Last Year">
    <p:data columnKey="Joe" value="100" />
    <p:data columnKey="Bob" value="120" />
  </p:series>
</p:barchart>

```

	<pre>         &lt;/p:series&gt;          &lt;p:series key="This Year"&gt;           &lt;p:data columnKey="Joe" value="125" /&gt;           &lt;p:data columnKey="Bob" value="115" /&gt;         &lt;/p:series&gt;       &lt;/p:barchart&gt; </pre>
--	--

<p:linechart>	<pre> ##  #####  ##  • borderVisible — #####  • borderPaint — #####  • borderBackgroundPaint — #####  • borderStroke —  • domainAxisLabel — #####  • domainAxisPaint — #####  • domainGridlinesVisible — #####  • domainGridlinePaint — #####  • domainGridlineStroke — #####  • height — #####  • width — #####  • is3D — #####3D###2D#####  • legend — #####  • legendItemPaint — #####  • legendItemBackgroundPaint — #####  • orientation — #####vertical(##)# ##### horizontal#  • plotBackgroundPaint — #####  • plotBackgroundAlpha — #####alpha#####   #####0#####1#####  • plotForegroundAlpha — #####alpha#####   #####0#####1#####  • plotOutlinePaint — ##### </pre>
---------------	--

	<ul style="list-style-type: none"> <li>• plotOutlineStroke — #####</li> <li>• rangeAxisLabel — #####</li> <li>• rangeAxisPaint — #####</li> <li>• rangeGridlinesVisible — #####</li> <li>• rangeGridlinePaint — #####</li> <li>• rangeGridlineStroke — #####</li> <li>• title — #####</li> <li>• titlePaint — #####</li> <li>• titleBackgroundPaint — #####</li> <li>• width — #####</li> </ul> <p>##</p> <pre> &lt;p:linechart title="Line Chart"   width="500" height="500"&gt;   &lt;p:series key="Prices"&gt;     &lt;p:data columnKey="2003" value="7.36" /&gt;     &lt;p:data columnKey="2004" value="11.50" /&gt;     &lt;p:data columnKey="2005" value="34.625" /&gt;     &lt;p:data columnKey="2006" value="76.30" /&gt;     &lt;p:data columnKey="2007" value="85.05" /&gt;   &lt;/p:series&gt; &lt;/p:linechart&gt; </pre>
--	---

<p:piechart>	<p>##</p> <p>#####</p> <p>##</p> <ul style="list-style-type: none"> <li>• title ##</li> <li>• label ##</li> <li>• legend ##</li> <li>• is3D ##3D</li> <li>• labelLinkMargin #####</li> <li>• labelLinkPaint #####</li> <li>• labelLinkStroke #####</li> <li>• labelLinksVisible #####</li> </ul>
--------------	--



	<ul style="list-style-type: none"> <li>• labelOutlinePaint #####</li> <li>• labelOutlineStroke #####</li> <li>• labelShadowPaint #####</li> <li>• labelPaint ####</li> <li>• labelGap ####</li> <li>• labelBackgroundPaint #####</li> <li>• startAngle ####</li> <li>• circular ##</li> <li>• direction ##</li> <li>• sectionOutlinePaint #####</li> <li>• sectionOutlineStroke #####</li> <li>• sectionOutlinesVisible #####</li> <li>• baseSectionOutlinePaint #####</li> <li>• baseSectionPaint #####</li> <li>• baseSectionOutlineStroke #####</li> </ul> <p>##</p>
--	---

<p:series>	<p>##</p> <p>#####series#####</p> <p>##</p> <ul style="list-style-type: none"> <li>• key — ####</li> <li>• seriesPaint — #####</li> <li>• seriesOutlinePaint — #####</li> <li>• seriesOutlineStroke — #####</li> <li>• seriesVisible — #####</li> <li>• seriesVisibleInLegend — #####</li> </ul> <p>##</p> <p>&lt;p:series key="data1"&gt;</p>
------------	--

	<pre> &lt;ui:repeat value="#{data.pieData1}" var="item"&gt;   &lt;p:data columnKey="#{item.name}" value="#{item.value}" /&gt; &lt;/ui:repeat&gt; &lt;/p:series&gt; </pre>
--	---

<p:data>	<pre> ##  #####  ##  • key — #####  • series — #####&lt;p:series&gt;#####  • value — #####  • explodedPercent — #####  • sectionOutlinePaint — #####  • sectionOutlineStroke — #####  • sectionPaint — #####  ## </pre> <pre> &lt;p:data key="foo" value="20" sectionPaint="#111111"   explodedPercent=".2" /&gt; &lt;p:data key="bar" value="30" sectionPaint="#333333" /&gt; &lt;p:data key="baz" value="40" sectionPaint="#555555"   sectionOutlineStroke="my-dot-style" /&gt; </pre>
----------	--

<p:color>	<pre> ##  #####  ##  • color — ##### Section 16.1.7.1, “###”  • color2 — #####  • point — #####  • point2 — #####  ## </pre> <pre> &lt;p:color id="foo" color="#0ff00f"/&gt; &lt;p:color id="bar" color="#ff00ff" color2="#00ff00"   point="50 50" point2="300 300"/&gt; </pre>
-----------	---

<code>&lt;p:stroke&gt;</code>	<pre>##  #####  ##  • width — #####  • cap — ##### butt#round # square  • join — ##### miter, round # bevel  • miterLimit — #####  • dash — #####  • dashPhase — #####  ##  &lt;p:stroke id="dot2" width="2" cap="round" join="bevel" dash="2 3" /&gt;</pre>
-------------------------------	--

## 16.3. #####

...

<code>&lt;p:barcode&gt;</code>	<pre>##  ##  • type ##  • code ##  • xpos ###  • ypos ###  • rotDegrees ####  • barHeight ##  • textSize ####  • minBarWidth #####  • barMultiplier  ##</pre>
--------------------------------	---

## 16.4. #####

##iText#####

- iText ## [<http://www.lowagie.com/iText/>]
- iText in Action [<http://www.manning.com/lowagie/>]

# Chapter 17. #####

Seam#####

##### jboss-seam-mail.jar ##### ##JAR#####mail JSF##### mailSession #####

examples/mail#####email#####

#####Seam#####mail##### Section 31.3.2, “Seam Mail#####”#

## 17.1. #####

#####Seam Mail##### — #####facelet##Facelet#####JSF#####

```
<m:message xmlns="http://www.w3.org/1999/xhtml"
  xmlns:m="http://jboss.com/products/seam/mail"
  xmlns:h="http://java.sun.com/jsf/html">

  <m:from name="Peter" address="peter@example.com" />
  <m:to name="{#{person.firstname} #{person.lastname}}">#{person.address}</m:to>
  <m:subject>Try out Seam!</m:subject>

  <m:body>
    <p><h:outputText value="Dear #{person.firstname}" />,</p>
    <p>You can try out Seam by visiting
    <a href="http://labs.jboss.com/jbossseam">http://labs.jboss.com/jbossseam</a>.</p>
    <p>Regards,</p>
    <p>Pete</p>
  </m:body>

</m:message>
```

<m:message> #####Seam##### ### <m:message> ##### <m:from> #####  
<m:to> #####facelet#####EL### <m:subject> ###

<m:body> #####JSF#####HTML#####

#####email##### # m:message #####mailSession #####  
#####Seam#####

```
@In(create=true)
private Renderer renderer;

public void send() {
    try {
        renderer.render("/simple.xhtml");
        facesMessages.add("Email sent successfully");
    }
    catch (Exception e) {
        facesMessages.add("Email sending failed: " + e.getMessage());
    }
}
```

#####email#####

### 17.1.1. ##

Seam#####Java###

##### jboss-seam-mail.jar#

```
<m:attachment value="/WEB-INF/lib/jboss-seam-mail.jar"/>
```

Seam####classpath##### jboss-seam-mail.jar ##### file-  
Name #####

```
<m:attachment value="/WEB-INF/lib/jboss-seam-mail.jar" fileName="this-is-so-cool.jar"/>
```

##### java.io.File#java.net.URL#

```
<m:attachment value="#{numbers}"/>
```

#### byte[] ## java.io.InputStream#

```
<m:attachment value="#{person.photo}" contentType="image/png"/>
```

##### byte[] # java.io.InputStream# #####MIME#####

#####Seam####PDF#####JSF##### &lt;m:attachment&gt; #####

```
<m:attachment fileName="tiny.pdf">
  <p:document>
    A very tiny PDF
  </p:document>
</m:attachment>
```

##### &lt;ui:repeat&gt;#

```
<ui:repeat value="#{people}" var="person">
  <m:attachment value="#{person.photo}" contentType="image/jpeg" fileName="#{person.firstname}_{person.lastname}.jpg" />
</ui:repeat>
```

#####

```
<m:attachment
  value="#{person.photo}"
  contentType="image/jpeg"
  fileName="#{person.firstname}_{person.lastname}.jpg"
  status="personPhoto"
  disposition="inline" />

```

##### cid:#{...} #####

#####IETF#####src#####Content-ID#####— #####

#####

## 17.1.2. HTML/Text ####

#####HTML#####

```
<m:body>
  <f:facet name="alternative">Sorry, your email reader can't show our fancy email,
  please go to http://labs.jboss.com/jbossseam to explore Seam.</f:facet>
</m:body>
```

### 17.1.3. #####

##### <ui:repeat> ####

```
<ui:repeat value="#{allUsers}" var="user">
  <m:to name="#{user.firstname} #{user.lastname}" address="#{user.emailAddress}" />
</ui:repeat>
```

### 17.1.4. ####

##### <ui:repeat> ####

```
<ui:repeat value="#{people}" var="p">
  <m:message>
    <m:from name="#{person.firstname} #{person.lastname}">#{person.address}</m:from>
    <m:to name="#{p.firstname}">#{p.address}</m:to>
    ...
  </m:message>
</ui:repeat>
```

### 17.1.5. ##

#####facelets#####Seam#mail#####

### template.xhtml ###

```
<m:message>
  <m:from name="Seam" address="do-not-reply@jboss.com" />
  <m:to name="#{person.firstname} #{person.lastname}">#{person.address}</m:to>
  <m:subject>#{subject}</m:subject>
  <m:body>
    <html>
      <body>
        <ui:insert name="body">This is the default body, specified by the template.</ui:insert>
      </body>
    </html>
  </m:body>
</m:message>
```

### templating.xhtml ###

```
<ui:param name="subject" value="Templating with Seam Mail"/>
<ui:define name="body">
  <p>This example demonstrates that you can easily use <i>facelets templating</i> in email!</p>
</ui:define>
```

#####facelet#####jar##### WEB-INF/lib ### # ###Seam Mail# web.xml ##  
.taglib.xml #####Seam Mail#####JSF#Servlet##### web.xml #####

#####Facelets#JSF#####Renderer##### # #####

### 17.1.6. ###

Seam#####JSF#####

```
<m:message charset="UTF-8">
  ...
</m:message>
```

#####facelets#####

```
<?xml version="1.0" encoding="UTF-8"?>
```

### 17.1.7. #####

#####Seam##### Section 17.5, “##”## #####

```
<m:message xmlns:m="http://jboss.com/products/seam/mail"
  importance="low"
  requestReadReceipt="true"/>
```

##### &lt;m:header&gt; #####

```
<m:header name="X-Sent-From" value="JBoss Seam"/>
```

## 17.2. ####

```
#####EJB#####MDB#####Bean:Message Driven Bean##### JBoss##JCA### —mail-ra.rar
####JBoss##### Seam### mail-ra.rar###Seam## mail #### mail-
ra.rar ##### $JBoss_HOME/server/default/deploy ##### JBoss#####
```

#####

```
@MessageDriven(activationConfig={
    @ActivationConfigProperty(propertyName="mailServer", propertyValue="localhost"),
    @ActivationConfigProperty(propertyName="mailFolder", propertyValue="INBOX"),
    @ActivationConfigProperty(propertyName="storeProtocol", propertyValue="pop3"),
    @ActivationConfigProperty(propertyName="userName", propertyValue="seam"),
    @ActivationConfigProperty(propertyName="password", propertyValue="seam")
})
@ResourceAdapter("mail-ra.rar")
@Name("mailListener")
public class MailListenerMDB implements MailListener {

    @In(create=true)
    private OrderProcessor orderProcessor;

    public void onMessage(Message message) {
        // Process the message
        orderProcessor.process(message.getSubject());
    }

}
```

##### onMessage(Message message) ##### Seam#####MDB#####

### http://wiki.jboss.org/wiki/Wiki.jsp?page=InboundJavaMail # mail-ra.rar #####

#####JBoss##### mail-ra.rar#####

## 17.3. ##

```
##### jboss-seam-mail.jar ### WEB-INF/lib ##### JBoss
AS####Seam##### JavaMail#API#####JavaMail API####JBoss
AS####API##### lib/mail.jar ##Seam#####Java Activation Framework##### lib/ac-
```



tivation.jar ##Seam####

Seam#Email####Facelets#####JSP#####seam-ui##

mailSession ####JavaMail###'###SMTP#####

### 17.3.1. mailSession

#####JEE#####JNDI#####JavaMail Session#####Seam####Session#

# Section 28.8, “#####” #####

#### 17.3.1.1. #JBoss AS###JNDI

JBossAS deploy/mail-service.xml ##JavaMail#####JNDI# ##### http://wiki.jboss.org/wiki/Wiki.jsp?page=JavaMail

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:core="http://jboss.com/products/seam/core"
  xmlns:mail="http://jboss.com/products/seam/mail">

  <mail:mail-session session-jndi-name="java:/Mail"/>

</components>
```

#####Seam#JNDI#### java:/Mail #####Session##

#### 17.3.1.2. Seam####

##### components.xml ##### Seam## smtp.example.com ##SMTP####

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:core="http://jboss.com/products/seam/core"
  xmlns:mail="http://jboss.com/products/seam/mail">

  <mail:mail-session host="smtp.example.com"/>

</components>
```

## 17.4. Meldware

Seam#####Meldware### buni.org [http://buni.org]#####

Meldware### SMTP#POP3#IMAP#WebMail#####

#####JEE#####Seam#####JBoss##

#Seam#####Meldware#####mail/buni-meldware##### # #####

#####Meldware#####vanilla### ##### meldware #####

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:core="http://jboss.com/products/seam/core"
  xmlns:mail="http://jboss.com/products/seam/mail">

  <mail:mail-session host="smtp.example.com"/>

  <mail:meldware>
    <mail:users>
      <value>#{duke}</value>
      <value>#{root}</value>
    </mail:users>

  </mail:meldware>
```

```

    </mail:meldware>

    <mail:meldware-user name="duke" username="duke" password="duke">
        <mail:aliases>
            <value>duke@jboss.org</value>
            <value>duke@jboss.com</value>
        </mail:aliases>
        <mail:meldware-user name="root" username="root" password="root" administrator="true" />
    </components>

```

##### duke ### root #####

## 17.5. ##

##### http://jboss.com/products/seam/mail ##### ##### message  
 ###message###Seam#####

###facelets##### JSF#####stylesheets#javascript#####  
 urlBase#

<m:message>

#####

- importance — #####
- precedence — #####
- requestReadReceipt — ###false##### From: ###
- urlBase — ##### requestContextPath ##### <h:graphicImage> ####

<m:from>

#####

- name — #####
- address — #####

<m:replyTo>

#####

- address — #####

<m:to>

#####<m:to>### #####<ui:repeat>####

- name — #####
- address — #####

<m:cc>

#####<m:cc>#####<ui:repeat>####

- name — #####
- address — #####

<m:bcc>

#####<m:bcc>### #####<ui:repeat>###

- name — #####
- address — #####

<m:header>

#####X-Sent-From: JBoss Seam##

- name — #####X-Sent-From##
- value — #####JBoss Seam##

<m:attachment>

#####

- value — #####
  - String — #classpath### String #####
  - java.io.File —#### File ###EL####
  - java.net.URL — ####URL###EL####
  - java.io.InputStream — #### InputStream ###EL#### #####fileName # contentType #####
  - byte[] — #### byte[] ###EL#### #####fileName # contentType #####
- #####
- ##### <p:document> ##### fileName #####
- #####JSF#####HTML#####fileName#####
- fileName — #####
- contentType —#####MIME###

<m:subject>

#####

<m:body>

##### alternative facet# #####HTML#####html#####

- type — #### plain#####HTML###

## Chapter 18. #####

Seam#####Web#####Java EE#####JMS#  
#Seam#####QoS#####Seam##Seam#####JMS#####

#####JMS#####Seam##### *dispatchers* ###

- java.util.concurrent.ScheduledThreadPoolExecutor ####
- EJB Timer Service ### EJB 3.0 ###
- Quartz

### 18.1. ##

##### ## ScheduledThreadPoolExecutor  
###dispatcher##### EJB 3.0##### compon-  
ents.xml ##

```
<async:timer-service-dispatcher/>
```

#####EJB##### Timer#####Seam#####  
#####EJB 3.0#####

#####Quartz##### ##Quartz#JAR## lib #####EAR#### application.xml  
#####Java### ##### components.xml ####Quartz Dispatcher#

```
<async:quartz-dispatcher/>
```

Seam#API##### ScheduledThreadPoolExecutor #Seam API##EJB3 Timer#Quartz Scheduler #####  
##### components.xml #####”#####plug and play#“#

#### 18.1.1. ####

#####  
#####AJAX#####AJAX#####

##EJB#####

```
@Local
public interface PaymentHandler
{
    @Asynchronous
    public void processPayment(Payment payment);
}
```

###JavaBean#####

#####Bean#####

```
@Stateless
@Name("paymentHandler")
public class PaymentHandlerBean implements PaymentHandler
{
```

```

    public void processPayment(Payment payment)
    {
        //do some work!
    }
}

```

#####

```

@Stateful
@Name("paymentAction")
public class CreatePaymentAction
{
    @In(create=true) PaymentHandler paymentHandler;
    @In Bill bill;

    public String pay()
    {
        paymentHandler.processPayment( new Payment(bill) );
        return "success";
    }
}

```

##### ##### ### ###

##### @Duration#@Expiration # @IntervalDuration#####

```

@Local
public interface PaymentHandler
{
    @Asynchronous
    public void processScheduledPayment(Payment payment, @Expiration Date date);

    @Asynchronous
    public void processRecurringPayment(Payment payment, @Expiration Date date, @IntervalDuration Long
}

```

```

@Stateful
@Name("paymentAction")
public class CreatePaymentAction
{
    @In(create=true) PaymentHandler paymentHandler;
    @In Bill bill;

    public String schedulePayment()
    {
        paymentHandler.processScheduledPayment( new Payment(bill), bill.getDueDate() );
        return "success";
    }

    public String scheduleRecurringPayment()
    {
        paymentHandler.processRecurringPayment( new Payment(bill), bill.getDueDate(), ONE_MONTH );
        return "success";
    }
}

```

##### Timer #####EJB3 Dispatcher## The Timer #####ScheduledThread-  
PoolExecutor#####JDK###Future###Quartz Dispatcher###QuartzTriggerHandle#####

```

@Local
public interface PaymentHandler
{
    @Asynchronous
    public Timer processScheduledPayment(Payment payment, @Expiration Date date);
}

```

```

@Stateless
@Name("paymentHandler")
public class PaymentHandlerBean implements PaymentHandler
{
    @In Timer timer;

    public Timer processScheduledPayment(Payment payment, @Expiration Date date)
    {
        //do some work!

        return timer; // #####
    }
}

```

```

@Stateful
@Name("paymentAction")
public class CreatePaymentAction
{
    @In(create=true) PaymentHandler paymentHandler;
    @In Bill bill;

    public String schedulePayment()
    {
        Timer timer = paymentHandler.processScheduledPayment( new Payment(bill), bill.getDueDate() );
        return "success";
    }
}

```

#####

### 18.1.2. ##Quartz Dispatcher####

Quartz dispatcher##### @Asynchronous#@Duration#@Expiration # @IntervalDuration  
#####Quartz dispatcher#####

@FinalExpiration #####

```

// Defines the method in the "processor" component
@Asynchronous
public QuartzTriggerHandle schedulePayment(@Expiration Date when,
                                           @IntervalDuration Long interval,
                                           @FinalExpiration Date endDate,
                                           Payment payment)
{
    // do the repeating or long running task until endDate
}

... ..

// Schedule the task in the business logic processing code
// Starts now, repeats every hour, and ends on May 10th, 2010
Calendar cal = Calendar.getInstance ();
cal.set (2010, Calendar.MAY, 10);
processor.schedulePayment(new Date(), 60*60*1000, cal.getTime(), payment);

```

##### QuartzTriggerHandle ##### QuartzTriggerHandle  
#####

```

QuartzTriggerHandle handle =
    processor.schedulePayment(payment.getPaymentDate(),
                             payment.getPaymentCron(),
                             payment);

```

```

payment.setQuartzTriggerHandle( handle );
// Save payment to DB

// later ...

// Retrieve payment from DB
// Cancel the remaining scheduled tasks
payment.getQuartzTriggerHandle().cancel();

```

@IntervalCron #####Unix cron#####2:10pm#2:44pm###

```

// Define the method
@Asynchronous
public QuartzTriggerHandle schedulePayment(@Expiration Date when,
                                           @IntervalCron String cron,
                                           Payment payment)
{
    // do the repeating or long running task
}

... ..

// Schedule the task in the business logic processing code
QuartzTriggerHandle handle =
    processor.schedulePayment(new Date(), "0 10,44 14 ? 3 WED", payment);

```

@IntervalBusinessDay #####”#n#Business Day“### #####2#business day#14:00###  
#####business day####2010#####

```

// Define the method
@Asynchronous
public QuartzTriggerHandle schedulePayment(@Expiration Date when,
                                           @IntervalBusinessDay NthBusinessDay nth,
                                           Payment payment)
{
    // do the repeating or long running task
}

... ..

// Schedule the task in the business logic processing code
QuartzTriggerHandle handle =
    processor.schedulePayment(new Date(),
        new NthBusinessDay(2, "14:00", WEEKLY), payment);

```

NthBusinessDay ##### additionalHolidays #####

```

public class NthBusinessDay implements Serializable
{
    int n;
    String fireAtTime;
    List <Date> additionalHolidays;
    BusinessDayIntervalType interval;
    boolean excludeWeekends;
    boolean excludeUsFederalHolidays;

    public enum BusinessDayIntervalType { WEEKLY, MONTHLY, YEARLY }

    public NthBusinessDay ()
    {
        n = 1;
        fireAtTime = "12:00";
        additionalHolidays = new ArrayList <Date> ();
        interval = BusinessDayIntervalType.WEEKLY;
    }
}

```

```

        excludeWeekends = true;
        excludeUsFederalHolidays = true;
    }
    ... ..
}

```

```

@IntervalDuration#@IntervalCron    #    @IntervalNthBusinessDay    #####    #####
RuntimeException#

```

### 18.1.3. #####

```

##### Events ## raiseAsynchronousEvent() ##### 
raiseTimedEvent() ##### schedule ## #####dispatcher#####dispatcher#### TimerSchedule##
#####

```

## 18.2. Seam####

```

Seam#JMS#####Seam####Seam#####

```

### 18.2.1. ##

```

#####JMS#####Seam#####Seam#####Topic#####Queue#####Seam##### QueueCon-
nectionFactory ### TopicConnectionFactory#

```

```

Seam####          UIL2ConnectionFactory#####JBossMQ#####          #####JMS#####
seam.properties#web.xml          #          components.xml          #####          queueConnec-
tion.queueConnectionFactoryJndiName # topicConnection.topicConnectionFactoryJndiName#

```

```

##### components.xml #####Topic#####Queue#####Seam### TopicPublisher # QueueSender#

```

```

<jms:managed-topic-publisher name="stockTickerPublisher" auto-create="true" topic-jndi-name="topic/st
<jms:managed-queue-sender name="paymentQueueSender" auto-create="true" queue-jndi-name="queue/payment

```

### 18.2.2. #####

```

#####JMS TopicPublisher # TopicSession #####

```

```

@In
private TopicPublisher stockTickerPublisher;
@In
private TopicSession topicSession;

public void publish(StockPrice price) {
    try
    {
        stockTickerPublisher.publish( topicSession.createObjectMessage(price) );
    }
    catch (Exception ex)
    {
        throw new RuntimeException(ex);
    }
}

```

```

####Queue####

```



```
@In
private QueueSender paymentQueueSender;
@In
private QueueSession queueSession;

public void publish(Payment payment) {
    try
    {
        paymentQueueSender.send( queueSession.createObjectMessage(payment) );
    }
    catch (Exception ex)
    {
        throw new RuntimeException(ex);
    }
}
```

### 18.2.3. #####Bean####

#####EJB3####Bean##### Bean#####Seam#####Seam###

### 18.2.4. #####

Seam Remoting#####JavaScript#####JMS###Topic#####

## Chapter 19. ##

```
#####PHP/Ruby#####share                                noth-
ing#####“#####
##(Share          nothing          except          for          the          data-
base)”#####——#####
```

```
#####/#####
```

```
#####Cache#####Seam#####
```

- #####
- #####ORM#####Hibernate#####JPA#####  
#####  
#####Seam#####
- Seam#####
- #####Seam#####EJB#####Bean#####  
#####Seam#####Seam#####  
#####
- #####Seam#####
- #####Seam# pojoCache  
#####JBossCache#####Seam#####JBossCache#####
- ###Seam#####JSF#####rendered frag-  
ments###ORM#####

```
#####ORM##### pojoCache
#####JBossCache##### <s:cache>#####page fragment####
```

### 19.1. #Seam###JBossCache

```
### pojoCache ##### org.jboss.cache.aop.PojoCache
#####Java#####Java#####JBossCache#####
#####
```

```
##### pojoCache#####JBossCache jar####classpath##### treecache.xml
#####cache###JBossCache#####JBossCache#####
```

```
# examples/blog/resources/treecache.xml ##### treecache.xml #####
```

```
#Seam#EAR#####JBossCache jar#####EAR#####EAR#lib##### jboss-cache.jar #
jgroups.jar#
```

```
#####Seam###
```

```
@Name("chatroom")
public class Chatroom {
    @In PojoCache pojoCache;
```

```

public void join(String username) {
    try
    {
        Set<String> userList = (Set<String>) pojoCache.get("chatroom", "userList");
        if (userList==null)
        {
            userList = new HashSet<String>();
            pojoCache.put("chatroom", "userList", userList);
        }
        userList.put(username);
    }
    catch (CacheException ce)
    {
        throw new RuntimeException(ce);
    }
}
}

```

#####JBossCache##### components.xml:

```
<core:pojo-cache name="myCache" cfg-resource-name="myown/cache.xml" />
```

## 19.2. #####

##JBossCache##### <s:cache> #####Seam#####JSF#####<s:cache> #####  
 pojoCache#####jar####EAR#####

<s:cache>#####blog#####blog#

```

<s:cache key="recentEntries-#{blog.id}" region="welcomePageFragments">
  <h:dataTable value="#{blog.recentEntries}" var="blogEntry">
    <h:column>
      <h3>#{blogEntry.title}</h3>
      <div>
        <s:formattedText value="#{blogEntry.body}" />
      </div>
    </h:column>
  </h:dataTable>
</s:cache>

```

key#####blog#####region

#####JBossCache#####

###<s:cache>#####

```

public void post() {
    ...
    entityManager.persist(blogEntry);
    pojoCache.remove("welcomePageFragments", "recentEntries-" + blog.getId() );
}

```

#####JbossCache#####

# Chapter 20. Web Services

Seam#JBossWS#####JEE Web Services####Seam#####Web Services# #####Web Services#Sema#####

## 20.1. #####

####Seam##Web Service#####Seam#####SOAP####  
org.jboss.seam.webservice.SOAPRequesHandler ### SOAPHandler #####Web  
Service#####Seam##### standard-jaxws-endpoint-config.xml  
#####Web Service## jar ### META-INF #### #####SOAP#####

```
<jaxws-config xmlns="urn:jboss:jaxws-config:2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:javaee="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="urn:jboss:jaxws-config:2.0 jaxws-config_2_0.xsd">
  <endpoint-config>
    <config-name>Seam WebService Endpoint</config-name>
    <pre-handler-chains>
      <javaee:handler-chain>
        <javaee:protocol-bindings>##SOAP11_HTTP</javaee:protocol-bindings>
        <javaee:handler>
          <javaee:handler-name>SOAP Request Handler</javaee:handler-name>
          <javaee:handler-class>org.jboss.seam.webservice.SOAPRequestHandler</javaee:handler-cla
        </javaee:handler>
      </javaee:handler-chain>
    </pre-handler-chains>
  </endpoint-config>
</jaxws-config>
```

## 20.2. ###Web Services

#####Web Service##### Seam#####SOAP#####SOAP#####ID#####  
#####ID#Web Service#####

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:seam="http://seambay.example.seam.jboss.org/">
  <soapenv:Header>
    <seam:conversationId xmlns:seam='http://www.jboss.org/seam/webservice'>2</seam:conversationId>
  </soapenv:Header>
  <soapenv:Body>
    <seam:confirmAuction/>
  </soapenv:Body>
</soapenv:Envelope>
```

#####SOAP#####ID#####2##SOAP##### conversationId #####Web  
Services#####Web Service#####Web Services#####ID#####

#####conversationId ##### http://www.jboss.org/seam/webservice  
#####Seam#####ID#####

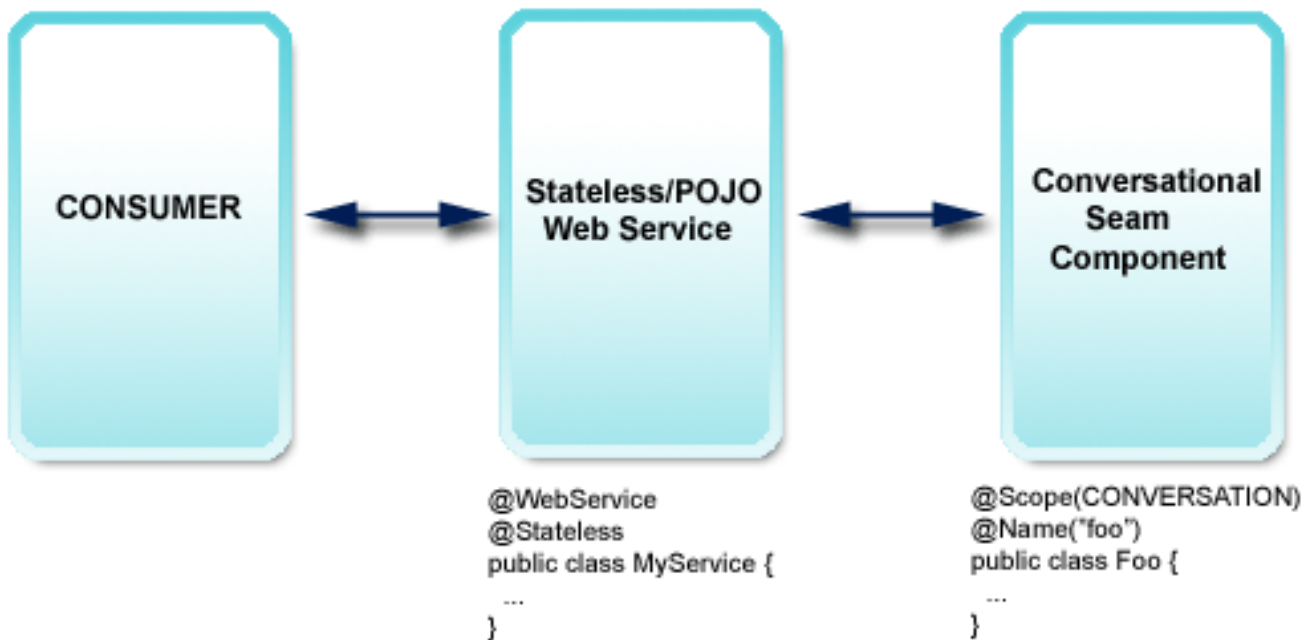
```
<env:Envelope xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
  <env:Header>
    <seam:conversationId xmlns:seam='http://www.jboss.org/seam/webservice'>2</seam:conversationId>
  </env:Header>
  <env:Body>
    <confirmAuctionResponse xmlns="http://seambay.example.seam.jboss.org/">
```

```
</env:Body>
</env:Envelope>
```

```
##### conversationId ###
```

## 20.2.1. ####

```
##Web Services#####Bean##POJO#####Web Services####Web
Service#####Seam#####facade##
```



```
##Web Service#####Bean##### @Name#####Seam### #####Seam#####Web
Service#####
```

## 20.3. Web Servic##

```
#####Web Service#### #####Seam# /examples #####seamBay#####
#####Web Service#####Web Service###
```

```

@Stateless
@WebService(name = "AuctionService", serviceName = "AuctionService")
public class AuctionService implements AuctionServiceRemote
{
    @WebMethod
    public boolean login(String username, String password)
    {
        Identity.instance().setUsername(username);
        Identity.instance().setPassword(password);
        Identity.instance().login();
        return Identity.instance().isLoggedIn();
    }

    // snip
}

```

```
#####Web Service#####Bean###JSR-181##### javax.jws ###JWS##### @WebService
#####Web Service## login() ##### @WebMethod #####Web Service### @WebService #####
```

name # serviceName #####

#####Web Service#####Web Service#####Web Service#####Bean#####  
#####AuctionServiceRemote ##### login() ##### @WebMethod#

#####Web Service#####Seam### Identity ### login() ### #####Web  
Service#####Seam### #####Web Services#####

#####Web Service##### AuctionAction.createAuction() #####

```
@WebMethod
public void createAuction(String title, String description, int categoryId)
{
    AuctionAction action = (AuctionAction) Component.getInstance(AuctionAction.class, true);
    action.createAuction();
    action.setDetails(title, description, categoryId);
}
```

##### AuctionAction #####

```
@Begin
public void createAuction()
{
    auction = new Auction();
    auction.setAccount(authenticatedAccount);
    auction.setStatus(Auction.STATUS_UNLISTED);
    durationDays = DEFAULT_AUCTION_DURATION;
}
```

#####Web Services#####Seam#####

# Chapter 21. Remoting

```
Seam##AJAX##Web#####
#####AJAX#####AJAX#Web#####Seam
Remoting#####
```

## 21.1. ##

```
####Remoting##### web.xml #####Seam Resource Servlet#
```

```
<servlet>
  <servlet-name>Seam Resource Servlet</servlet-name>
  <servlet-class>org.jboss.seam.servlet.SeamResourceServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>Seam Resource Servlet</servlet-name>
  <url-pattern>/seam/resource/*</url-pattern>
</servlet-mapping>
```

```
#####Web#####JavaScript#####
```

```
<script type="text/javascript" src="seam/resource/remoting/resource/remote.js"></script>
```

```
#####
##### @Name("customerAction") #####Bean#####
```

```
<script type="text/javascript"
      src="seam/resource/remoting/interface.js?customerAction"></script>
```

```
#####
```

```
<script type="text/javascript"
      src="seam/resource/remoting/interface.js?customerAction&accountAction"></script>
```

```
##### s:remote #####JavaScript#####
```

```
<s:remote include="customerAction,accountAction"/>
```

## 21.2. Seam##

```
##### Seam JavaScript##### ##JavaScript### remote.js #####
#####Seam.Component #####Seam.Remoting #####
```

### 21.2.1. Hello World##

```
##### Seam ##### helloAction ###Seam###
```

```
@Stateless
@Name("helloAction")
public class HelloAction implements HelloLocal {
  public String sayHello(String name) {
```

```
        return "Hello, " + name;
    }
}
```

##### ——— @WebRemote #####

```
@Local
public interface HelloLocal {
    @WebRemote
    public String sayHello(String name);
}
```

#####Web## ——— #####

```
<s:remote include="helloAction"/>
```

#####

```
<button onclick="javascript:sayHello()">Say Hello</button>
```

#####

```
<script type="text/javascript">
//

function sayHello() {
    var name = prompt("What is your name?");
    Seam.Component.getInstance("helloAction").sayHello(name, sayHelloCallback);
}

function sayHelloCallback(result) {
    alert(result);
}

// ]]&gt;
&lt;/script&gt;</pre>
</div>
<div data-bbox="81 590 770 604" data-label="Text">
<p>#####hello#####</p>
</div>
<div data-bbox="81 606 765 622" data-label="Text">
<p>#####Seam# /examples/remoting/helloworld #####Hello World#####</p>
</div>
<div data-bbox="81 637 762 652" data-label="Text">
<p>##### JavaScript#####</p>
</div>
<div data-bbox="81 654 410 668" data-label="Text">
<p>#####</p>
</div>
<div data-bbox="98 686 765 699" data-label="Text">
<pre>Seam.Component.getInstance("helloAction").sayHello(name, sayHelloCallback);</pre>
</div>
<div data-bbox="81 719 923 733" data-label="Text">
<p>##### Seam.Component.getInstance("helloAction")##### helloAction #####</p>
</div>
<div data-bbox="81 735 684 750" data-label="Text">
<p>##### sayHello(name, sayHelloCallback); #####</p>
</div>
<div data-bbox="81 766 923 781" data-label="Text">
<p>##### sayHello ##### name ##### sayHelloCallback ##### sayHello #####</p>
</div>
<div data-bbox="81 782 923 798" data-label="Text">
<p>#####Seam Remoting#####JavaScript### sayHelloCallback ###</p>
</div>
<div data-bbox="81 799 575 814" data-label="Text">
<p>##### void #####</p>
</div>
<div data-bbox="81 829 580 844" data-label="Text">
<p>## sayHelloCallback #####</p>
</div>
<div data-bbox="81 868 361 889" data-label="Section-Header">
<h2>21.2.2. Seam.Component</h2>
</div>
<div data-bbox="81 904 923 938" data-label="Text">
<pre>Seam.Component JavaScript#####Seam##### newInstance() #
getInstance()##### newInstance() ##### getInstance() #####</pre>
</div>
<div data-bbox="81 963 230 979" data-label="Page-Footer">JBoss Seam 2.0GA</div>
<div data-bbox="884 963 923 979" data-label="Page-Footer">178</div>
```



**Seam.Component.newInstance()**

```
##newInstance()#####JavaBean#####
```

```
#####Seam#####
```

```
@Name("customer")
@Entity
public class Customer implements Serializable
{
    private Integer customerId;
    private String firstName;
    private String lastName;

    @Column public Integer getCustomerId() {
        return customerId;
    }

    public void setCustomerId(Integer customerId) {
        this.customerId = customerId;
    }

    @Column public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    @Column public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

```
#####Customer#####
```

```
var customer = Seam.Component.newInstance("customer");
```

```
#####customer#####
```

```
customer.setFirstName("John");
// Or you can set the fields directly
customer.lastName = "Smith";
```

**Seam.Component.getInstance()**

```
##                               #####Seam##Bean#####
#####
```

```
##### customer##### ##### customerAction ### saveCustomer() ###
```

```
Seam.Component.getInstance("customerAction").saveCustomer(customer);
```

**Seam.Component.getComponentName()**

```
##### null#
```

```
if (Seam.Component.getComponentName(instance) == "customer")
```

```

    alert("Customer");
    else if (Seam.Component.getComponentName(instance) == "staff")
    alert("Staff member");

```

### 21.2.3. Seam.Remoting

Seam Remoting##### Seam.Remoting ####

#### Seam.Remoting.createType()

```

#####Seam###JavaBean##### ## createType()
#####Java#####

```

```

var widget = Seam.Remoting.createType("com.acme.widgets.MyWidget");

```

#### Seam.Remoting.getTypeName()

```

Seam.Remoting.getTypeName() # Seam.Component.getComponentName() #####
##### null#####Java###

```

## 21.3. EL####

```

Seam Remoting##EL#####
Seam.Remoting.eval()###EL#####
#####EL#####

```

```

function customersCallback(customers) {
    for (var i = 0; i < customers.length; i++) {
        alert("Got customer: " + customers[i].getName());
    }
}

Seam.Remoting.eval("#{customers}", customersCallback);

```

```

##### #{customers} ##Seam### #####Customer##### customersCallback() ###
##### s:remote#####JavaScript##### customer ##### cus-
tomer ###

```

```

<s:remote include="customer"/>

```

## 21.4. #####

##### seam/resource/remoting/interface.js #####

####s:remote##

```

<script type="text/javascript"
    src="seam/resource/remoting/interface.js?customerAction"></script>

```

```

<s:remote include="customerAction"/>

```

```
#####Seam Remoting#####

#####bean#####

#####Seam#####Bean#####JavaBean#####
#####JavaBean#####Bean#####Bean#####@WebRemote#
#####EJB#####Remoting###JavaBean#####Bean#
```

## 21.5. ###

Seam Remoting#####ID#####

### 21.5.1. #####ID

```
#####Seam Remoting#####ID# #####
Seam.Remoting.getContext().getConversationId() #####ID# #####
Seam.Remoting.getContext().setConversationId() #####ID#

####ID### Seam.Remoting.getContext().setConversationId() #####
#####ID#####ID#
#####
```

### 21.5.2. #####

```
#####ID#
#####JavaScript#####ID#####ID#
```

```
Seam.Remoting.getContext().setConversationId( #{conversation.id} );
```

## 21.6. ####

Seam Remoting#####

```
Seam.Remoting.startBatch() #####
#####Seam.Remoting.executeBatch() #####
#####

##### startBatch() ##### Seam.Remoting.cancelBatch()
#####

##### /examples/remoting/chatroom#
```

## 21.7. #####

### 21.7.1. ## / ## ##

```
#####
```

## String

```
#####Javascript#####
```

## Number

```
##Java##### Byte#Double#Float#Integer#Long
# Short #####
```

## Boolean

```
Boolean#####JavaScript Boolean#####Java Boolean#
```

## 21.7.2. JavaBeans

```
#####JavaBean###Seam#####JavaBean##### Seam#####
Seam.Component.newInstance()##### Seam.Remoting.createType()#####
```

```
#####
```

```
@Name("myAction")
public class MyAction implements MyActionLocal {
    public void doSomethingWithObject(Object obj) {
        // code
    }
}
```

```
##### myWidget ##### myAction ##### myWidget#####
##### MyWidget#
```

```
<s:remote include="myAction,myWidget"/>
```

```
##### Seam.Component.newInstance("myWidget") ## myWidget ##### myAc-
tion.doSomethingWithObject() ##
```

## 21.7.3. Date#Time

```
#####Javascript##### java.util.Date #####
java.sql.Date # java.sql.Timestamp##
```

## 21.7.4. Enums ####

```
#####Enum####String#####Enum#####Enum#####
```

```
@Name("paintAction")
public class paintAction implements paintLocal {
    public enum Color {red, green, blue, yellow, orange, purple};

    public void paint(Color color) {
        // code
    }
}
```

```
#### paint() #####color### red####red#####
```

```
Seam.Component.getInstance("paintAction").paint("red");
```

```
##### —— #####Enum#####Enum#####
```

## 21.7.5. Collections ##

### Bags

```
Bags#####arrays#collections#lists#sets#####Maps —— #####JavaScript array#
#####JavaScript array#####JavaScript array#
#####Seam Remoting#####Bag#####
```

### Maps

```
JavaScript####Map#####Seam Remoting#####Map### ##### Seam.Remoting.Map#####Map#####
```

```
var map = new Seam.Remoting.Map();
```

```
##Javascript#####Map#####size()#isEmpty()#keySet()#values()#get(key)#put(key, value)#re-
move(key) # contains(key)# #####Java#####Java##### keySet() # values()#
#####JavaScript###key##value###JavaScript Array#####
```

## 21.8. ##

```
#####Bug##### #####JavaScript#####
setDebug() ###
```

```
Seam.Remoting.setDebug(true);
```

```
####components.xml####:
```

```
<remoting:remoting debug="true"/>
```

```
##### setDebug(false)# ##### Seam.Remoting.log(message)#
```

## 21.9. ####

```
#####
```

### 21.9.1. ####

```
#####“Please Wait...”##### Seam.Remoting.loadingMessage ##:
```

```
Seam.Remoting.loadingMessage = "Loading...";
```

### 21.9.2. #####

```
##### displayLoadingMessage() # hideLoadingMessage() #####
```

```
// don't display the loading indicator
Seam.Remoting.displayLoadingMessage = function() {};
Seam.Remoting.hideLoadingMessage = function() {};
```

### 21.9.3. #####

```
##### displayLoadingMessage() # hideLoadingMessage()#
```

```
Seam.Remoting.displayLoadingMessage = function() {
    // Write code here to display the indicator
};

Seam.Remoting.hideLoadingMessage = function() {
    // Write code here to hide the indicator
};
```

### 21.10. #####

```
#####XML#####JavaScript###
#####Javabean#####
#####
#####
```

```
Seam      Remoting##### @WebRemote      #### exclude      ###
#####.#####
```

```
##### Widget #####
```

```
@Name("widget")
public class Widget
{
    private String value;
    private String secret;
    private Widget child;
    private Map<String,Widget> widgetMap;
    private List<Widget> widgetList;

    // getters and setters for all fields
}
```

#### 21.10.1. #####

```
##### Widget ##### secret #####
```

```
@WebRemote(exclude = {"secret"})
public Widget getWidget();
```

```
#"secret"##### secret ##### widget ##### child ##### widget#
##### child #####.#####
```

```
@WebRemote(exclude = {"child.secret"})
public Widget getWidget();
```

#### 21.10.2. #####

```
##### Map #####List#Set#Array #### ##### Widget ### wid-
getList ##### Widget ##### Widget # secret #####
```

```
@WebRemote(exclude = {"widgetList.secret"})
public Widget getWidget();
```

```
##### Map ##### # Map ##### [key] ### Map ##### [value] ##### widgetMap
##### secret ###
```

```
@WebRemote(exclude = {"widgetMap[value].secret"})
public Widget getWidget();
```

### 21.10.3. #####

```
#####
#####Seam#####Seam#####[]#####
```

```
@WebRemote(exclude = {"[widget].secret"})
public Widget getWidget();
```

### 21.10.4. ####

```
#####
```

```
@WebRemote(exclude = {"widgetList.secret", "widgetMap[value].secret"})
public Widget getWidget();
```

## 21.11. JMS##

```
Seam Remoting#####JMS#####JMS#####
```

### 21.11.1. ##

```
#####JMS#####Seam Remoting##### ##seam.properties#web.xml ## components.xml #
org.jboss.seam.remoting.messaging.subscriptionRegistry.allowedTopics #####
```

```
<remoting:remoting poll-timeout="5" poll-interval="1"/>
```

### 21.11.2. ##JMS##

```
#####JMS###
```

```
function subscriptionCallback(message)
{
    if (message instanceof Seam.Remoting.TextMessage)
        alert("Received message: " + message.getText());
}

Seam.Remoting.subscribe("topicName", subscriptionCallback);
```

```
Seam.Remoting.subscribe()#####JMS#####
```

```
#####Text###Object### ##### instanceof #####
Seam.Remoting.TextMessage ## Seam.Remoting.ObjectMessage# TextMessage # text ##### ObjectMes-
sage # object ##### getObject() #####
```

### 21.11.3. ####

```
##### Seam.Remoting.unsubscribe()#####
```

```
Seam.Remoting.unsubscribe("topicName");
```

### 21.11.4. #####

```
##### Seam.Remoting.pollInterval#####10#
```

```
##### Seam.Remoting.pollTimeout#####
#####0#####
```

```
##### pollTimeout #####
#####
```

```
####components.xml#####JavaScript#####
#####
```

```
##components.xml##
```

```
<remoting:remoting poll-timeout="5" poll-interval="1"/>
```

```
##JavaScript##
```

```
// Only wait 1 second between receiving a poll response and sending the next poll request.
Seam.Remoting.pollInterval = 1;

// Wait up to 5 seconds on the server for new messages
Seam.Remoting.pollTimeout = 5;
```



---

## Chapter 22. Seam#Google#Web###(GWT)

```
#####Google      Web      Toolkit#####GWT#####AJAX#####Seam#####GWT
widget###Seam#####
```

```
####GWT#####GWT##      -      #####      http://code.google.com/webtoolkit/      ###
#####GWT#####
```

### 22.1. ##

```
#Seam#####GWT#####Seam###servlet##### Chapter 25, Seam###Seam##### #
```

### 22.2. #####

```
#####GWT#####Seam#####GWT##
com.google.gwt.user.client.rpc.RemoteService #
```

```
public interface MyService extends RemoteService
{
    public String askIt(String question);
}
```

```
##### AsyncCallback ###
```

```
public interface MyServiceAsync extends RemoteService
{
    public void askIt(String question, AsyncCallback callback);
}
```

```
##### MyServiceAsync #####GWT#####
```

```
#####Seam###
```

```
@Name("org.jboss.seam.example.remoting.gwt.client.MyService")
public class ServiceImpl implements MyService
{
    @WebRemote
    public String askIt(String question)
    {
        if (!validate(question))
        {
            throw new IllegalStateException("Hey, this shouldn't happen, I checked on the client, " +
                "but its always good to double check.");
        }
        return "42. Its the real question that you seek now.";
    }

    public boolean validate(String q)
    {
        ValidationUtility util = new ValidationUtility();
        return util.isValid(q);
    }
}
```

```
#####GWT##### @WebRemote #####Web#####
```

## 22.3. #GWT#####Seam##

#####stub#####

```
private MyServiceAsync getService()
{
    String endpointURL = GWT.getModuleBaseURL() + "seam/resource/gwt";

    MyServiceAsync svc = (MyServiceAsync) GWT.create(MyService.class);
    ((ServiceDefTarget) svc).setServiceEntryPoint(endpointURL);
    return svc;
}
```

#####label#####text input#####button#####

```
public class AskQuestionWidget extends Composite
{
    private AbsolutePanel panel = new AbsolutePanel();

    public AskQuestionWidget()
    {
        Label lbl = new Label("OK, what do you want to know?");
        panel.add(lbl);
        final TextBox box = new TextBox();
        box.setText("What is the meaning of life?");
        panel.add(box);
        Button ok = new Button("Ask");
        ok.addClickListener(new ClickListener()
        {
            public void onClick(Widget w)
            {
                ValidationUtility valid = new ValidationUtility();
                if (!valid.isValid(box.getText()))
                {
                    Window.alert("A question has to end with a '?'");
                }
                else
                {
                    askServer(box.getText());
                }
            }
        });
        panel.add(ok);

        initWidget(panel);
    }

    private void askServer(String text)
    {
        getService().askIt(text, new AsyncCallback()
        {
            public void onFailure(Throwable t)
            {
                Window.alert(t.getMessage());
            }

            public void onSuccess(Object data)
            {
                Window.alert((String) data);
            }
        });
    }

    ...
}
```

```
##### askServer() ##### askServer()
##### getService() ##### askIt() #####
```

## HelloWorld

This is an example of a host page for the HelloWorld application. You can attach a Web Toolkit module to any HTML page you like, making it easy to add bits of AJAX functionality to existing pages without starting from scratch.

OK, what do you want to know?

What is the meaning of life

```
#####Seam#### examples/remoting/gwt #####
```

## 22.4. GWT Ant Targets

```
##GWT#####JavaScript#####Ant#####GWT#####GUI#####
#####Ant classpath###Ant###jar#####GWT#####
```

```
#####Ant#####Ant#####
```

```
<taskdef uri="antlib:de.samaflost.gwttasks"
         resource="de/samaflost/gwttasks/antlib.xml"
         classpath="./lib/gwttasks.jar"/>

<property file="build.properties"/>
```

```
#### build.properties #####
```

```
gwt.home=/gwt_home_dir
```

```
#####GWT#####Target#
```

```
<!-- the following are are handy utilities for doing GWT development.
      To use GWT, you will of course need to download GWT seperately -->
<target name="gwt-compile">
  <!-- in this case, we are "re homing" the gwt generated stuff, so in this case
        we can only have one GWT module - we are doing this deliberately to keep the URL short -->
  <delete>
    <fileset dir="view"/>
  </delete>
  <gwt:compile outDir="build/gwt"
    gwtHome="${gwt.home}"
    classBase="${gwt.module.name}"
    sourceclasspath="src"/>
  <copy todir="view">
    <fileset dir="build/gwt/${gwt.module.name}"/>
  </copy>
</target>
```

This target when called will compile the GWT application, and copy it to the specified directory (which would be in the webapp part of your war - remember GWT generates HTML and Javascript artifacts). You never edit the resulting code that gwt-compile generates - you always edit in the GWT source directory.

```
####Target#####GWT#####war# webapp ### —— ##GWT##HTML#Javascript####
##### gwt-compile ##### —— #####GWT#####
```

```
##GWT##### —— #####GWT#####
#####GWT ——
#####
```

---

## Chapter 23. Spring Framework##

Spring#####Spring#####Seam#####Spring##Seam#####Conversation##Seam#####  
##

####Spring#####jboss-seam-ioc#####seam-spring#####

Seam#Spring#####

- #Seam#####Spring Bean#
- #Spring Bean###Seam###
- #Spring Bean###Seam##
- ##Spring Bean#####Seam#####
- ##Seam#####Spring WebApplicationContext
- ##Spring PlatformTransactionManagement
- #Spring# OpenEntityManagerInViewFilter # OpenSessionInViewFilter #####Seam#####
- ### @Asynchronous ###Spring TaskExecutors

### 23.1. #Seam####Spring Bean#

##Seam#####Spring Bean##### <seam:instance/> #####  
Bean#####Seam#####

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:seam="http://jboss.com/products/seam/spring-seam"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
    http://jboss.com/products/seam/spring-seam
    http://jboss.com/products/seam/spring-seam-2.0.xsd">
```

#####Seam#####Spring Bean###

```
<bean id="someSpringBean" class="SomeSpringBeanClass" scope="prototype">
  <property name="someProperty">
    <seam:instance name="someComponent" />
  </property>
</bean>
```

###EL#####

```
<bean id="someSpringBean" class="SomeSpringBeanClass" scope="prototype">
  <property name="someProperty">
    <seam:instance name="#{someExpression}" />
  </property>
</bean>
```

Seam#####Spring Bean id####Spring Bean##

```
<seam:instance name="someComponent" id="someSeamComponentInstance"/>

<bean id="someSpringBean" class="SomeSpringBeanClass" scope="prototype">
  <property name="someProperty" ref="someSeamComponentInstance">
</bean>
```

###

Seam#####Spring#####Seam#####bijection####Spring#####Spring  
 Bean#####Bean#####Bean#####Seam# CONVERSATION  
 #####Spring Bean#####Bean##### scope  
 impedance##Seam#####Spring#####Seam#####  
 <seam:instance/> #####Seam###

```
<seam:instance id="seamManagedEM" name="someManagedEMComponent" proxy="true"/>

<bean id="someSpringBean" class="SomeSpringBeanClass">
  <property name="entityManager" ref="seamManagedEM">
</bean>
```

#####Spring Bean###Seam#####Seam#####Spring# OpenEntityMa-  
 nagerInView ##### #Spring###Seam#####

## 23.2. #Spring Bean###Seam###

#Spring Bean###Seam#####

- ##EL#####Spring Bean
- #Spring Bean###Seam##

#####Spring Bean#####EL####

Spring#DelegatingVariableResolver#Spring####JSF#####VariableResolver #####Spring Bean##Bean  
 id#EL##### faces-config.xml ### DelegatingVariableResolver#

```
<application>
  <variable-resolver>
    org.springframework.web.jsf.DelegatingVariableResolver
  </variable-resolver>
</application>
```

##### @In ### Spring Bean#

```
@In("#{bookingService}")
private BookingService bookingService;
```

Spring Bean#EL#####Seam###EL#####Spring Bean#####working memory as-  
 sertions###...

## 23.3. #Spring Bean###Seam##

<seam:component/> #####Spring Bean#####Seam#####Seam###Bean#####

```
<seam:component /> #####
```

```
<bean id="someSpringBean" class="SomeSpringBeanClass" scope="prototype">
  <seam:component />
</bean>
```

```
#####, <seam:component /> ###Bean##### STATELESS# #Seam##### FactoryBean
##Spring Bean#####Bean#####class #####Seam#####
```

```
#####Spring Bean#####Seam##### <seam:component /> # scope ##### ## #Seam##Spring
Bean##### prototype #####Spring Bean#####
```

## 23.4. Seam####Spring Bean

```
Seam#####Spring 2.0#####Seam#####Seam#####Spring
Bean#####Spring#####Statefulness#####Session#Conversation####Spring
ng Bean#####
```

```
###Spring#Bean Factory##### <seam:configure-scopes />####Seam#####Spring
Bean#####Spring Bean#####Seam#####bean### scope #####Seam####
```

```
<!-- Only needs to be specified once per bean factory-->
<seam:configure-scopes />

...

<bean id="someSpringBean" class="SomeSpringBeanClass" scope="seam.CONVERSATION" />
```

```
##### configure-scopes ### prefix #####seam##
```

```
#####Seam###Spring Bean#####Spring Bean#####
<seam:instance />#####Spring#####Bean#####
<aop:scoped-proxy />####Seam###Spring Bean# # ##
<aop:scoped-proxy />#####Seam###Spring Bean##### <seam:instance />#
```

```
<bean id="someSpringBean" class="SomeSpringBeanClass" scope="seam.CONVERSATION" />

...

<bean id="someSingleton">
  <property name="someSeamScopedSpringBean">
    <seam:instance name="someSpringBean" proxy="true" />
  </property>
</bean>
```

## 23.5. ##Spring PlatformTransactionManagement

```
Spring#####API#JPA#Hibernate#JDO#JTA#####WebSphere#WebLogic#####Tran
sactionManagers#####Spring#####Java
EE#####REQUIRES_NEW#NOT_SUPPORTED#####Spring##
[http://static.springframework.org/spring/docs/2.0.x/reference/transaction.html]#
```

```
####Seam###SpringTransaction#####Spring###
```

```
<spring:spring-transaction platform-transaction-manager="#{transactionManager}" />
```

```
spring:spring-transaction#####Spring#####
```

## 23.6. #Spring###Seam#####

```
Seam#####conversation scope#####EntityManager#####
LazyInitializationException #####Spring#####Web#####OpenEntityManagerInViewFilter#####Spring#####Spring##JPA#####Seam#####PersistenceAnnotationBeanPostProcessor#JpaTemplate####
```

```
Seam###Spring####JPA####Seam#####Spring#####
```

```
#####
```

- ##Spring#####Seam#####
- ##Web#####Seam#####Quartz####
- #####Seam#####Spring#####

```
Spring#####EntityManagerFactory#####EntityManager###Seam#####EntityManagerFactory#####Seam#####
```

```
<bean id="seamEntityManagerFactory" class="org.jboss.seam.ioc.spring.SeamManagedEntityManagerFactoryBean"
    <property name="persistenceContextName" value="entityManager"/>
</bean>
```

'persistenceContext-

```
Name'#Seam#####EntityManagerFactory####Seam#####unitName#####'entityManager'#####unitName#####persistenceUnitName#####
```

```
<bean id="seamEntityManagerFactory" class="org.jboss.seam.ioc.spring.SeamManagedEntityManagerFactoryBean"
    <property name="persistenceContextName" value="entityManager"/>
    <property name="persistenceUnitName" value="bookingDatabase:extended"/>
</bean>
```

```
##EntityManagerFactory####Spring#####Spring# PersistenceAnnotationBeanPostProcessor#
```

```
<bean class="org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor"/>
```

```
#####Spring#####EntityManagerFactory#####Seam##### PersistenceAnnotationBeanPostProcessor #####persistenceUnitName##### defaultPersistenceUnitName #####
```

```
applicationContext.xml #####
```

```
<bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalEntityManagerFactoryBean">
    <property name="persistenceUnitName" value="bookingDatabase"/>
</bean>
<bean id="seamEntityManagerFactory" class="org.jboss.seam.ioc.spring.SeamManagedEntityManagerFactoryBean"
    <property name="persistenceContextName" value="entityManager"/>
    <property name="persistenceUnitName" value="bookingDatabase:extended"/>
</bean>
<bean class="org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor">
    <property name="defaultPersistenceUnitName" value="bookingDatabase:extended"/>
</bean>
```

component.xml #####

```
<persistence:managed-persistence-context name="entityManager"
    auto-create="true" entity-manager-factory="{entityManagerFactory}" />
```

JpaTemplate # JpaDaoSupport #####

```
<bean id="bookingService" class="org.jboss.seam.example.spring.BookingService">
    <property name="entityManagerFactory" ref="seamEntityManagerFactory" />
</bean>
```

## 23.7. #Spring###Seam###Hibernate##

Seam#Spring#####Spring#####Seam###Hibernate###Hibernate Session#### JPA## ###

#Spring#JPA#####Spring#####Spring#####EntityManagerFactory#####EntityManager###S  
eam Session#####SessionFactory#####Seam###Hibernate#####

```
<bean id="seamSessionFactory" class="org.jboss.seam.ioc.spring.SeamManagedSessionFactoryBean">
    <property name="sessionName" value="hibernateSession" />
</bean>
```

'sessionName'#persistence:managed-hibernate-session

#####SessionFactory#####Spring##### SessionFactory.getCurrentInstance() #####

SeamManagedSessionFactory # getCurrentInstance() ###

## 23.8. ##Seam###Spring#####

#####Spring#ContextLoaderListener #####Spring ApplicationContext#####

- Spring#ApplicationContext##### SeamListener ##
- ##Seam#####Spring ApplicationContext####

#####Spring#####Spring      ApplicationContext#Seam#### components.xml      ###  
<spring:context-loader/> ##### config-locations #####Spring#####  
components.xml ##### <spring:config-locations/> #####

```
<components xmlns="http://jboss.com/products/seam/components"
    xmlns:spring="http://jboss.com/products/seam/spring"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://jboss.com/products/seam/components
        http://jboss.com/products/seam/components-2.0.xsd
        http://jboss.com/products/seam/spring
        http://jboss.com/products/seam/spring-2.0.xsd">

    <spring:context-loader context-locations="/WEB-INF/applicationContext.xml"/>

</components>
```

## 23.9. ##Spring TaskExecutor#@Asynchronous

Spring##### TaskExecutor ##### @Asynchronous #####Spring      Seam#####



##### SpringTaskExecutorDispatchor #####taskExecutor#Spring Bean#

```
<spring:task-executor-dispatcher task-executor="#{springThreadPoolTaskExecutor}"/>
```

##Spring# TaskExecutor #####Seam Dispatcher #####

```
<!-- Install a ThreadPoolDispatcher to handle scheduled asynchronous event -->
<core:thread-pool-dispatcher name="threadPoolDispatcher"/>

<!-- Install the SpringDispatcher as default -->
<spring:task-executor-dispatcher task-executor="#{springThreadPoolTaskExecutor}" schedule-dispatcher=
```

# Chapter 24. Hibernate Search

## 24.1. ##

```
#Apache Lucene™ #####
##### Hibernate
Search##### Hiberante Search#####Apache Luncene#

Hibernate Search#####JPA#Hibernate#####JBoss Seam#####Hibernate Search####

###Hibernate Search##### Hibernate Search documentation
[http://www.hibernate.org/hib_docs/search/reference/en/html_single/]
```

## 24.2. ##

```
Hibernate Search#### META-INF/persistence.xml ##### hibernate.cfg.xml #####
```

```
#####Hibernate Search#####
```

```
<persistence-unit name="sample">
  <jta-data-source>java:/DefaultDS</jta-data-source>
  <properties>
    [...]
    <!-- use a file system based index -->
    <property name="hibernate.search.default.directory_provider"
      value="org.hibernate.search.store.FSDirectoryProvider"/>
    <!-- directory where the indexes will be stored -->
    <property name="hibernate.search.default.indexBase"
      value="/Users/prod/apps/dvdstore/dvdindexes"/>
  </properties>
</persistence-unit>
```

```
#####Hibernate Annotations##EntityManager 3.2.x#####JBoss AS 4.2.GA#####
```

```
<persistence-unit name="sample">
  <jta-data-source>java:/DefaultDS</jta-data-source>
  <properties>
    [...]
    <!-- use a file system based index -->
    <property name="hibernate.search.default.directory_provider"
      value="org.hibernate.search.store.FSDirectoryProvider"/>
    <!-- directory where the indexes will be stored -->
    <property name="hibernate.search.default.indexBase"
      value="/Users/prod/apps/dvdstore/dvdindexes"/>

    <property name="hibernate.ejb.event.post-insert"
      value="org.hibernate.search.event.FullTextIndexEventListener"/>
    <property name="hibernate.ejb.event.post-update"
      value="org.hibernate.search.event.FullTextIndexEventListener"/>
    <property name="hibernate.ejb.event.post-delete"
      value="org.hibernate.search.event.FullTextIndexEventListener"/>

  </properties>
</persistence-unit>
```

### Note

```
#####Hibernate Annotation##EntityManager 3.3.x#####
```

#####jars#####

- hibernate-search.jar
- hibernate-commons-annotations.jar
- lucene-core.jar

## Note

#####EAR##### application.xml ##

## 24.3. ##

Hibernate Search#####Lucene##### reference documentation  
[http://www.hibernate.org/hib\_docs/search/reference/en/html\_single/] #####

Hibernate Search####JPA/Hibernate#API#####HQL##Criteria#####API#####  
FullTextSession API #Hibernate Session #####

##Hibernate Search####JBoss Seam## FullTextSession #

```
@Stateful
@Name("search")
public class FullTextSearchAction implements FullTextSearch, Serializable
{
    @In
    FullTextSession session;

    public void search(String searchString) {
        org.apache.lucene.query.Query luceneQuery = getLuceneQuery();
        org.hibernate.Query query = session.createFullTextQuery(luceneQuery, Product.class);
        searchResults = query
            .setMaxResults(pageSize + 1)
            .setFirstResult(pageSize * currentPage)
            .list();
    }
    [...]
}
```

## Note

FullTextSession ### org.hibernate.Session#####Hibernate Session####

####Java Persistence API#####

```
@Name("search")
public class FullTextSearchAction implements FullTextSearch, Serializable
{
    @In
    FullTextEntityManager em;

    public void search(String searchString) {
        org.apache.lucene.query.Query luceneQuery = getLuceneQuery();
        javax.persistence.Query query = em.createFullTextQuery(luceneQuery, Product.class);
        searchResults = query
            .setMaxResults(pageSize + 1)
            .setFirstResult(pageSize * currentPage)
            .getResultList();
    }
}
```

```

    }
    [...]
}

```

```

##Hibernate Search####FulltextEntityManager #### FullTextEntityManager ### EntityManager
##### ## FullTextSession ### Session#

```

```

### EJB 3.0 Session ## Message Driven Bean ##### @PersistenceContext ###Bean## ##### Full-
TextEntityManager ##### EntityManager ### ##### FullTextEntityManager #####

```

```

@Stateful
@Name("search")
public class FullTextSearchAction implements FullTextSearch, Serializable
{
    @PersistenceContext
    EntityManager em;

    public void search(String searchString) {
        org.apache.lucene.query.Query luceneQuery = getLuceneQuery();
        FullTextEntityManager ftEm = (FullTextEntityManager) em;
        javax.persistence.Query query = ftEm.createFullTextQuery(luceneQuery, Product.class);
        searchResults = query
            .setMaxResults(pageSize + 1)
            .setFirstResult(pageSize * currentPage)
            .getResultList();
    }
    [...]
}

```

## Caution

```

#####Seam####Hibernate Search##### Search.createFullTextSession #####

```

```

##Hibernate Search#####JBoss Seam#####DVDStore##Blog###

```

---

## Chapter 25. Seam####Seam#####

#####Seam####JSF####Servlet#####XML#####  
#####

### 25.1. Seam####

#####Seam#JSF#####

#### 25.1.1. #Seam#JSF#servlet####

#####Faces Servlet#

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.seam</url-pattern>
</servlet-mapping>
```

#####URL#####

###Seam##### web.xml #####

```
<listener>
  <listener-class>org.jboss.seam.servlet.SeamListener</listener-class>
</listener>
```

##listener####Seam#####

##JSF#####Seam##### web.xml  
#####

```
<context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>client</param-value>
</context-param>
```

#### 25.1.2. ##Facelets

#####Facelets##JSP##### faces-config.xml #####

```
<application>
  <view-handler>com.sun.facelets.FaceletViewHandler</view-handler>
</application>
```

##### web.xml ##

```
<context-param>
  <param-name>javax.faces.DEFAULT_SUFFIX</param-name>
  <param-value>.xhtml</param-value>
</context-param>
```

### 25.1.3. Seam Resource Servlet

Seam Resource Servlet#####Seam Remoting#captchas####“##”#####JSF UI##### ##Seam Resource Servlet##### web.xml #####

```
<servlet>
  <servlet-name>Seam Resource Servlet</servlet-name>
  <servlet-class>org.jboss.seam.servlet.SeamResourceServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>Seam Resource Servlet</servlet-name>
  <url-pattern>/seam/resource/*</url-pattern>
</servlet-mapping>
```

### 25.1.4. Seam Servlet###

Seam#####Servlet#####

#####Seam#####Servlet#####Seam##### ##### web.xml #####

```
<filter>
  <filter-name>Seam Filter</filter-name>
  <filter-class>org.jboss.seam.servlet.SeamFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>Seam Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Seam#### ## ## web.xml #####

##### disabled="true"

#####

####

##### pages.xml ##### ##### ##Java  
EE###Web##### ##Servlet###Tomcat#####

##### <web:exception-filter> ## components.xml  
#####

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:web="http://jboss.com/products/seam/web">

  <web:exception-filter url-pattern="*.seam"/>

</components>
```

- url-pattern — #####

#####

#####Seam#####Seam#####

##### components.xml #####

```
<web:redirect-filter url-pattern="*.seam"/>
```

- url-pattern — #####

#####

###Seam###JSF#####/#####RFC-2388##### components.xml  
#####

```
<web:multipart-filter create-temp-files="true"
    max-request-size="1000000"
    url-pattern="*.seam"/>
```

- create-temp-files — ##### true#####  
##### false#
- max-request-size — ##### Content-Length #####0#####
- url-pattern — #####

####

#####

##### components.xml #####

```
<web:character-encoding-filter encoding="UTF-16"
    override-client="true"
    url-pattern="*.seam"/>
```

- encoding — #####
- override-client — ##### true##### encoding #####  
false##### false
- url-pattern — #####

## RichFaces

#####Ajax4jsf#Seam#####Ajax4jsf#### web.xml ###Ajax4jsf###

#####RichFaces jar#####RichFaces Ajax####

##### components.xml #####RichFaces#####

```
<web:ajax4jsf-filter force-parser="true"
    enable-cache="true"
    log4j-init-file="custom-log4j.xml"
    url-pattern="*.seam"/>
```

- force-parser — ##Richfaces#XML#####JSF### ## false####AJAX#####XML#  
##force-parser # false #####AJAX#####

- enable-cache — #####JavaScript#CSS#Images###  
#####JavaScript#CSS#####true#####
- log4j-init-file — ##### per-application ### #####web#####log4j.xml#####
- url-pattern — #####

## Identity Logging

#####log4j#####%X{username}#####

##### components.xml ### <web:logging-filter> #####

```
<components xmlns="http://jboss.com/products/seam/components"
  xmlns:web="http://jboss.com/products/seam/web">
  <web:logging-filter url-pattern="*.seam"/>
</components>
```

- url-pattern — Used to specify which requests the filter is active for. The default is all requests. url-pattern — #####

## ##Servlet#####

#####Servlet###JSF

Servlet#####JSF#####Seam#####Servlet#####Servlet#####Servlet#####Seam###

#####Servlet#Seam#####Seam##### ##### ###JSF FacesSer-  
vlet# Seam#JSF#####phase listener##

##### components.xml #####

```
<web:context-filter url-pattern="/media/*"/>
```

- url-pattern — #####URL#####

##### conversationId #####id# #####ID#####

#####id#####Seam###id##### conversation #####

## #####

Seam##### ##### web.xml #####servlet#####  
#####Seam##### @Filter #####Seam##### javax.servlet.Filter##

```
@Startup
@Scope(APPLICATION)
@Name("org.jboss.seam.web.multipartFilter")
@BypassInterceptors
@Filter(within="org.jboss.seam.web.ajax4jsfFilter")
public class MultipartFilter extends AbstractFilter {
```

## @Startup #####Seam#####bijection#####@BypassInterceptors##  
#####Ajax4jsf#####@Filter(within="org.jboss.seam.web.ajax4jsfFilter")#####



## 25.1.5. #Seam###EJB###

##### SeamInterceptor #####Seam### ##### ejb-jar.xml #####

```
<interceptors>
  <interceptor>
    <interceptor-class>org.jboss.seam.ejb.SeamInterceptor</interceptor-class>
  </interceptor>
</interceptors>

<assembly-descriptor>
  <interceptor-binding>
    <ejb-name>*/</ejb-name>
    <interceptor-class>org.jboss.seam.ejb.SeamInterceptor</interceptor-class>
  </interceptor-binding>
</assembly-descriptor>
```

Seam#####JNDI#####Bean#####Bean Seam##### @JndiName###  
#####Seam#####EJB#####JNDI###  
#####EJB3#####JNDI##### components.xml #####

##JBoss AS#####

```
<core:init jndi-name="myEarName/#{ejbName}/local" />
```

myEarName####Bean##EAR####

###EAR#####JBoss#EJB3#####

```
<core:init jndi-name="#{ejbName}/local" />
```

##### GlassFish#####EJB#####JNDI###  
#####:-)

###EJB3##### Events ##### components.xml #####Seam#####:

```
<transaction:ejb-transaction/>
```

## 25.1.6. ###

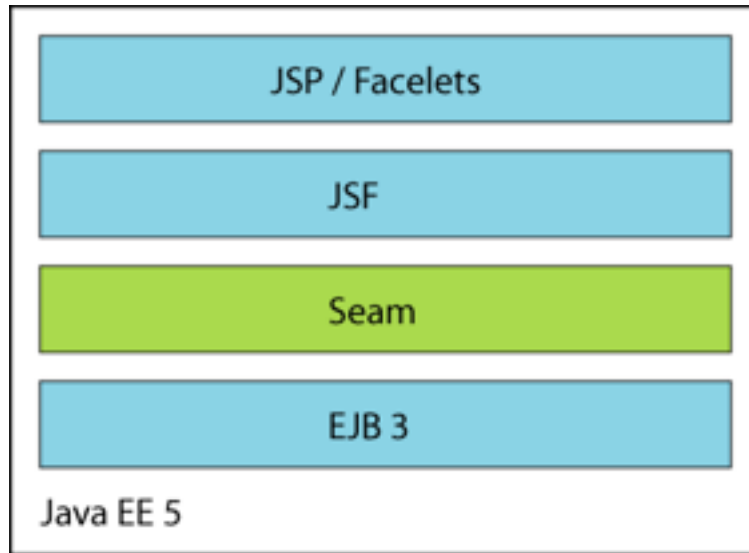
#####Seam##### seam.properties#META-INF/seam.properties #META-INF/components.xml### #####Seam### seam.properties  
#####Seam###

###Web#####WAR##### WEB-INF/classes #####Seam##### seam.properties###

#####Seam##### seam.properties #####

##### Java#####  
##### components.xml #####

## 25.2. #Java EE 5###Seam



#####Java EE 5#####Seam#####

## 25.2.1. ##

#####EAR#####

```

my-application.ear/
  jboss-seam.jar
  lib/
    jboss-el.jar
  META-INF/
    MANIFEST.MF
    application.xml
  my-application.war/
    META-INF/
      MANIFEST.MF
    WEB-INF/
      web.xml
      components.xml
      faces-config.xml
    lib/
      jsf-facelets.jar
      jboss-seam-ui.jar
    login.jsp
    register.jsp
    ...
  my-application.jar/
    META-INF/
      MANIFEST.MF
      persistence.xml
    seam.properties
    org/
      jboss/
        myapplication/
          User.class
          Login.class
          LoginBean.class
          Register.class
          RegisterBean.class
          ...
  
```

### META-INF/application.xml ## jboss-seam.jar #####EJB### jboss-el.jar

#####EAR#lib#####EAR#classpath###

#####jBPM#Drools#####EAR#lib#####jar###

#####Facelets#####WAR# WEB-INF/lib ##### jsf-facelets.jar

#####Seam#####Seam#####WAR# WEB-INF/lib ##### jboss-seam-ui.jar#  
#####PDF##Email##### WEB-INF/lib ##### jboss-seam-pdf.jar # jboss-seam-mail.jar#

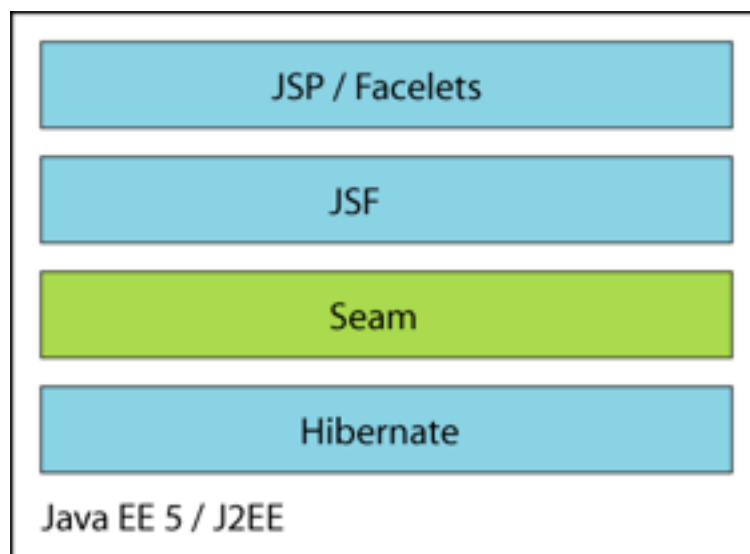
#####Seam#####Facelets#####WAR# WEB-INF/lib #####jboss-seam-debug.jar

Seam#####EJB 3.0#Java EE#####

#####

## 25.3. #J2EE###Seam

Seam#####EJB 3.0#####Hibernate3##JPA##EJB 3.0#####JavaBeans####Bean#  
#####bean#####EJB 3.0#####Seam#####



Seam JavaBean#####Bean##### ## ## JTA UserTransaction #####Seam#  
@Transactional ##### #####JavaBean####Hibernate#####Seam#####

Seam#####Hibernate3#JavaBeans###EJB3#####JPA#JavaBeans####  
#####J2EE#####

### 25.3.1. #Seam###Hibernate

#####Seam##### hibernate.cfg.xml ##### Hibernate SessionFactory#

```
<persistence:hibernate-session-factory name="hibernateSessionFactory" />
```

#####Seam###Hibernate Session##### *managed session*#

```
<persistence:managed-hibernate-session name="hibernateSessionFactory"
    session-factory="#{hibernateSessionFactory}" />
```

### 25.3.2. #Seam###JPA

#####Seam##### persistence.xml ##### JPA EntityManagerFactory#

```
<persistence:entity-manager-factory name="entityManagerFactory" />
```

#####Seam###JPA EntityManager##### *managed persistence context*

```
<persistence:managed-persistence-context name="entityManager"
    entity-manager-factory="#{entityManagerFactory}" />
```

### 25.3.3. ##

#####WAR###

```
my-application.war/
  META-INF/
    MANIFEST.MF
  WEB-INF/
    web.xml
    components.xml
    faces-config.xml
    lib/
      jboss-seam.jar
      jboss-seam-ui.jar
      jboss-el.jar
      jsf-facelets.jar
      hibernate3.jar
      hibernate-annotations.jar
      hibernate-validator.jar
      ...
    my-application.jar/
      META-INF/
        MANIFEST.MF
      seam.properties
      hibernate.cfg.xml
      org/
        jboss/
          myapplication/
            User.class
            Login.class
            Register.class
            ...
  login.jsp
  register.jsp
  ...
```

#####Hibernate#####EE#####Tomcat#TestNG#####

## 25.4. #Java SE###Seam#####JBoss

#####EE#####Seam#####Seam#####JTA###

#####JPA#####Seam###JPA##### EntityTransaction#####

```
<transaction:entity-transaction entity-manager="#{entityManager}" />
```

#####Hibernate#####Seam#####Hibernate##API#

```
<transaction:hibernate-transaction session="#{session}" />
```

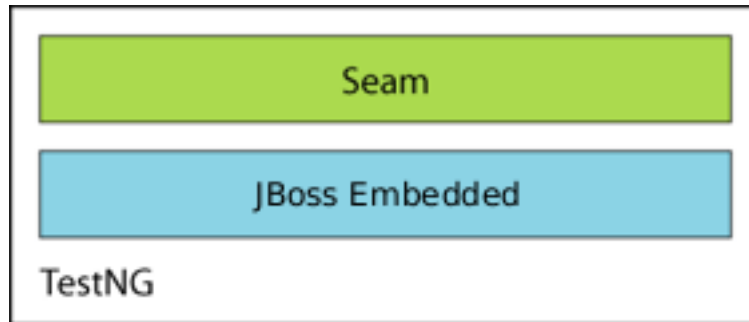
#####

#####JBoss###EE#API#

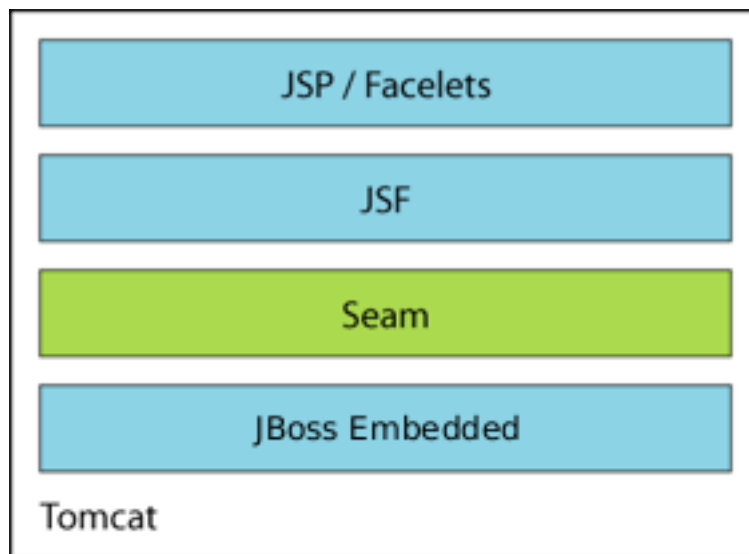
## 25.5. #####JBoss#Java SE###Seam

####JBoss#####Java EE 5#####EJB3#####

Seam#booking##### SeamTest #####JBoss####TestNG#####



booking#####Tomcat####



### 25.5.1. #####JBoss

```
###Seam#####Tomcat#####JBoss###Tomcat##          #####JBoss###          ##
[http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=228977]
#####JBoss###Tomcat 6#####JBoss#JAR#####Tomcat##
```

- #####JBoss# bootstrap # lib ##### jndi.properties #####Tomcat# lib ####
- #Tomcat# lib ##### annotations-api.jar ###

#####JBoss#####

- #####JBoss listener### conf/server.xml #####listener####

```
<Listener className="org.jboss.embedded.tomcat.EmbeddedJBossBootstrapListener" />
```

- #####listener# conf/context.xml #####WAR#####

```
<Listener className="org.jboss.embedded.tomcat.WebinfScanner" />
```

#####Tomcat#####JBoss Wiki## [http://wiki.jboss.org/wiki/Wiki.jsp?page=EmbeddedAndTomcat]#

## 25.5.2. ##

###Servlet###Tomcat###WAR#####

```
my-application.war/
  META-INF/
    MANIFEST.MF
  WEB-INF/
    web.xml
    components.xml
    faces-config.xml
    lib/
      jboss-seam.jar
      jboss-seam-ui.jar
      jboss-el.jar
      jsf-facelets.jar
      jsf-api.jar
      jsf-impl.jar
      ...
    my-application.jar/
      META-INF/
        MANIFEST.MF
        persistence.xml
      seam.properties
      org/
        jboss/
          myapplication/
            User.class
            Login.class
            LoginBean.class
            Register.class
            RegisterBean.class
            ...
      login.jsp
      register.jsp
      ...
```

#####Seam##### ant deploy.tomcat ###Tomcat##

## 25.6. #Seam###jBPM

Seam#jBPM#####jBPM##### # components.xml #####

```
<bpm:jbpm>
  <bpm:pageflow-definitions>
    <value>createDocument.jpdl.xml</value>
    <value>editDocument.jpdl.xml</value>
    <value>approveDocument.jpdl.xml</value>
  </bpm:pageflow-definitions>
  <bpm:process-definitions>
    <value>documentLifecycle.jpdl.xml</value>
  </bpm:process-definitions>
</bpm:jbpm>
```

#####jBPM#####jBPM#####Hibernate### Seam#DVD Store  
Demo####Seam##### jbpm.cfg.xml # hibernate.cfg.xml #####

```
<jbpm-configuration>

  <jbpm-context>
    <service name="persistence">
```

```

    <factory>
      <bean class="org.jbpm.persistence.db.DbPersistenceServiceFactory">
        <field name="isTransactionEnabled"><false/></field>
      </bean>
    </factory>
  </service>
  <service name="tx" factory="org.jbpm.tx.TxServiceFactory" />
  <service name="message" factory="org.jbpm.msg.db.DbMessageServiceFactory" />
  <service name="scheduler" factory="org.jbpm.scheduler.db.DbSchedulerServiceFactory" />
  <service name="logging" factory="org.jbpm.logging.db.DbLoggingServiceFactory" />
  <service name="authentication"
    factory="org.jbpm.security.authentication.DefaultAuthenticationServiceFactory" />
</jbpm-context>

</jbpm-configuration>

```

#####JBPM#####Seam#EJB3####JTA###

## 25.6.1. ##

##jBPM#####

#Seam#####EAR##### ##EAR#####

```

my-application.ear/
  jboss-seam.jar
  lib/
    jboss-el.jar
    jbpm-3.1.jar
  META-INF/
    MANIFEST.MF
    application.xml
  my-application.war/
    META-INF/
      MANIFEST.MF
    WEB-INF/
      web.xml
      components.xml
      faces-config.xml
    lib/
      jsf-facelets.jar
      jboss-seam-ui.jar
    login.jsp
    register.jsp
    ...
  my-application.jar/
    META-INF/
      MANIFEST.MF
      persistence.xml
      seam.properties
    org/
      jboss/
        myapplication/
          User.class
          Login.class
          LoginBean.class
          Register.class
          RegisterBean.class
          ...
    jbpm.cfg.xml
    hibernate.cfg.xml
    createDocument.jpdl.xml
    editDocument.jpdl.xml
    approveDocument.jpdl.xml
    documentLifecycle.jpdl.xml

```

## 25.7. #Portal###Seam

```
##Seam#####portlet#####portlet####portlet.xml#####Java EE##### ## examples/portal
#####booking demo#####JBoss Portal#####
```

## 25.8. #JBoss AS###SFSB#####

```
#####Bean#####HTTP#####SFSB###HTTP#####
JBoss#####30#####Bean##### server/default/conf/standardjboss.xml##### default#####
###SFSB##### LRUStatefulContextCachePolicy ##### max-bean-life #####
```

```
<container-cache-conf>
  <cache-policy>org.jboss.ejb.plugins.LRUStatefulContextCachePolicy</cache-policy>
  <cache-policy-conf>
    <min-capacity>50</min-capacity>
    <max-capacity>1000000</max-capacity>
    <remover-period>1800</remover-period>

    <!-- SFSB timeout in seconds; 1800 seconds == 30 minutes -->
    <max-bean-life>1800</max-bean-life>

    <overager-period>300</overager-period>
    <max-bean-age>600</max-bean-age>
    <resizer-period>400</resizer-period>
    <max-cache-miss-period>60</max-cache-miss-period>
    <min-cache-miss-period>1</min-cache-miss-period>
    <cache-load-factor>0.75</cache-load-factor>
  </cache-policy-conf>
</container-cache-conf>
```

```
###JBoss 4.0.x# server/default/deploy/jbossweb-tomcat55.sar/conf/web.xml #####JBoss 4.2.x#
server/default/deploy/jboss-web.deployer/conf/web.xml #####HTTP#####
#####Web#####
```

```
<session-config>
  <!-- HTTP Session timeout, in minutes -->
  <session-timeout>30</session-timeout>
</session-config>
```

```
##### web.xml #####
```



---

## Chapter 26. Seam on OC4J

OC4J###Java#Oracle###11g#####“###”#####Oracle#JEE 5##### Seam#####Hotel Book-  
ing#####seam-gen##### Seam#RichFaces Ajax####Seam Security  
###Drools##Facelets#Hibernate###JPA#

#####OC4J 11g#####OC4J 10g##### <http://www.oracle.com/technology/tech/java/oc4j/11/> ##OC4J  
11g#

### 26.1. jee5/booking ##

jee5/booking #####JBoss AS#####Glassfish#####OC4J##

#### 26.1.1. #####

#####OC4J#####

- jboss-seam.jar — #####EJB3#####Seam#####CMT#####EJB3#####Bean##
- jboss-el.jar
- jboss-seam-ui.jar — Seam#JSF Control###Apache#commons-beanutils
- jboss-seam-debug.jar
- jsf-facelets.jar
- richfaces-api.jar — ###Apache commons-digester#commons-beanutils
- richfaces-impl.jar # richfaces-ui.jar — ##Apache commons-digester#commons-beanutils

#### 26.1.2. OC4J#####

- Hibernate — #####Hibernate##JPA#####OC4J####TopLink Essentials#

#####Hibernate##JPA#####jar##hibernate3.jar#hibernate-annotations.jar#hibernate-en-  
#

titymanager.jar#####jboss-common.jar#jboss-archive-browsing.jar#ejb3-persistence.jar#  
#####Seam####hibernate/lib#####jar##

- thirdparty-all.jar — #Seam#####javassist#

#####Seam###JBoss AS##Glassfish##### Seam  
Text#####ANTLR#####OC4J#####“###”classloading#####

- antlr-2.7.6.jar — Seam Text#####
- jbpj-jpdl.jar — Seam#JBPM#####
- Drools — Seam Security#####Drools##Seam Security##### Drools#5#jar### - drools-

```
core-4.0.0.jar# drools-compiler-4.0.0.jar# janino-2.5.7.jar# mvel14-1.2rc1.jar # antlr-
runtime-3.0.jar#####Drools###
```

### 26.1.3. #####

```
#####
```

```
web.xml
```

```
#### web.xml #####EJB#####JEE 5##### # ##OC4J#Glassfish#
```

```
<ejb-local-ref>
  <ejb-ref-name>
    jboss-seam-jee5/AuthenticatorAction/local
  </ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home/>
  <local>
    org.jboss.seam.example.booking.Authenticator
  </local>
  <ejb-link>AuthenticatorAction</ejb-link>
</ejb-local-ref>
```

```
persistence.xml
```

```
#####JPA#####Hibernate###OC4J#####ANTLR#####OC4J#####
#
```

```
<property
  name="hibernate.query.factory_class"
  value="org.hibernate.hql.classic.ClassicQueryTranslatorFactory" />
<property
  name="hibernate.transaction.manager_lookup_class"
  value="org.hibernate.transaction.OrionTransactionManagerLookup" />
```

### 26.1.4. ## jee5/booking ##

```
1. #####
```

- build.xml — ###OC4J#####
- resources/META-INF/persistence.xml — ###Glassfish#####OC4J###

```
2. #### ant ###demo##### dist/jboss-seam-jee5.ear
```

```
3. # hsqldb.jar ###OC4J## cp ../../seam-gen/lib/hsqldb.jar $ORACLE_HOME/j2ee/home/applib/
###OC4J#####HSQLDB##
```

## 26.2. ##Seam#####OC4J#

```
#####JEE 5#####OC4J##### #####hsqldb#### jee5/booking ###
#####
```

```
1. #####OC4J
```

2. ##### \$JAVA\_HOME # \$ORACLE\_HOME ##### #\$ORACLE\_HOME#####OC4J#####  
#####OC4J#####OC4J##### Readme.txt #

3. ##OC4J### \$ORACLE\_HOME/j2ee/home/config/data-sources.xml### <data-sources> ###

```
<managed-data-source
  connection-pool-name="jee5-connection-pool"
  jndi-name="jdbc/__default"
  name="jee5-managed-data-source" />
<connection-pool name="jee5-connection-pool">
  <connection-factory
    factory-class="org.hsqldb.jdbcDriver"
    user="sa"
    password=""
    url="jdbc:hsqldb:." />
</connection-pool>
```

# persistence.xml ##jndi-name ### jta-data-source#

4. ## \$ORACLE\_HOME/j2ee/home/config/server.xml ## <application-server>###

```
<application name="jboss-seam-jee5"
  path="../../home/applications/jboss-seam-jee5.ear"
  parent="default"
  start="true" />
```

#####

5. ## \$ORACLE\_HOME/j2ee/home/config/default-web-site.xml## <web-site> ###

```
<web-app application="jboss-seam-jee5"
  name="jboss-seam-jee5"
  load-on-startup="true"
  root="/seam-jee5" />
```

root #####Web#####

6. #####OC4J#cp dist/jboss-seam-jee5.ear \$ORACLE\_HOME/j2ee/home/applications/

7. ##OC4J#\$ORACLE\_HOME/bin/oc4j -start

#####OC4J#####

8. # http://localhost:8888/seam-jee5 #####

9. ##### CTRL-C #####

## 26.3. ##### seam-gen #####OC4J##

#####IDE # seam-gen #####Eclipse#Netbeans#

##### seam-gen ##### seam-gen ##Hibernate Tools #####Scheam#####JPA###Bean#

#####Seam#####JSF#CRUD#####

#####MySQL#####SQL#####MySQL#####

#####Seam##### ./seam setup#

```
> ./seam setup
```

```
Buildfile: build.xml
```

```
setup:
```

```
[echo] Welcome to seam-gen :-)
[input] Enter your Java project workspace (the directory that contains your Seam projects) [/home/
[input] Enter your JBoss home directory [/home/pmuir/java/jboss-4.2.1.GA] [/home/pmuir/java/jboss
[input] Enter the project name [oc4j-example] [oc4j-example]
[input] Is this project deployed as an EAR (with EJB components) or a WAR (with no EJB support) [
[input] Enter the Java package name for your session beans [org.jboss.seam.tutorial.oc4j.action] [
[input] Enter the Java package name for your entity beans [org.jboss.seam.tutorial.oc4j.model] [o
[input] Enter the Java package name for your test cases [org.jboss.seam.tutorial.oc4j.test] [org.
[input] What kind of database are you using? [mysql] (hsql, [mysql], oracle, postgres, mssql, db
[input] Enter the Hibernate dialect for your database [org.hibernate.dialect.MySQLDialect] [org.h
[input] Enter the filesystem path to the JDBC driver jar [lib/mysql.jar] [lib/mysql.jar]
[input] Enter JDBC driver class for your database [com.mysql.jdbc.Driver] [com.mysql.jdbc.Driver]
[input] Enter the JDBC URL for your database [jdbc:mysql:///oc4j] [jdbc:mysql:///oc4j]
[input] Enter database username [user] [user]
[input] Enter database password [password] [password]
[input] skipping input as property hibernate.default_schema.new has already been set.
[input] Enter the database catalog name (it is OK to leave this blank) [] []
[input] Are you working with tables that already exist in the database? [y] ([y], n, )
[input] Do you want to drop and recreate the database tables and data in import.sql each time you
[propertyfile] Updating property file: /home/pmuir/workspace/jboss-seam/seam-gen/build.properties
[echo] Installing JDBC driver jar to JBoss server
[echo] Type 'seam new-project' to create the new project
```

```
BUILD SUCCESSFUL
```

```
## ./seam new-project ##### cd #####
```

```
## ./seam generate-entities #####Seam#####
```

```
#####
```

```
resources/META-INF/persistence-dev.xml
```

- ## jta-data-source # jdbc/\_\_\_oc4jExample ### data-sources.xml ##### jndi-name #
- #####

```
<property name="hibernate.query.factory_class"
  value="org.hibernate.hql.classic.ClassicQueryTranslatorFactory" />
<property name="hibernate.transaction.manager_lookup_class"
  value="org.hibernate.transaction.OrionTransactionManagerLookup" />
<property name="hibernate.transaction.flush_before_completion"
  value="true"/>
<property name="hibernate.cache.provider_class"
  value="org.hibernate.cache.HashtableCacheProvider"/>
```

- **##JBoss AS###EntityManagerFactory###:**

```
<property
name="jboss.entity.manager.factory.jndi.name"
value="java:/oc4j-exampleEntityManagerFactory">
```

- **#####OC4J##### persistence-prod.xml #**

resources/META-INF/jboss-app.xml

**#####JBoss AS##jboss-app.xml ###JBoss AS###classloading####**

resources/\*-ds.xml

**#####JBoss AS# ##JBoss AS#####OC4J##### data-sources.xml  
####**

resources/WEB-INF/components.xml

- **##CMT#container managed transaction#####—### # ## <transaction:ejb-transaction /> ###  
##### xmlns:transaction="http://jboss.com/products/seam/transaction"**
- **# jndi-pattern ### java:comp/env/oc4j-example/#{ejbName}/local**
- **#####Seam#MPC#Managed Persistence Context#####  
#####OC4J###JNDI###EntityManagerFactory###Seam#####**

```
<persistence:entity-manager-factory
auto-create="true"
name="oc4jEntityManagerFactory"
persistence-unit-name="oc4j-example" />
```

**#####Seam##### managed-persistence-context#**

```
<persistence:managed-persistence-context
name="entityManager"
auto-create="true"
entity-manager-factory="#{oc4jEntityManagerFactory}" />
```

resources/WEB-INF/web.xml

**#####EJB#####Seam#####**

```
<ejb-local-ref>
<ejb-ref-name>
oc4j-example/EjbSynchronizations/local
</ejb-ref-name>
<ejb-ref-type>Session</ejb-ref-type>
<local>
org.jboss.seam.transaction.LocalEjbSynchronizations
</local>
<ejb-link>EjbSynchronizations</ejb-link>
</ejb-local-ref>
```

build.xml

**#####target#####OC4J#####**

**#####**

- Hibernate —

- **#Seam####**      hibernate/lib      **#####jar#####**      oc4j-example/lib      **####**      cp  
     ../jboss-seam/hibernate/lib/\*.jar lib/

- **## build.xml#####ear## # #####**

```
<include name="lib/hibernate-annotations.jar" />
<include name="lib/hibernate-entitymanager.jar" />
<include name="lib/hibernate3.jar" />
<include name="ejb3-peristence.jar" />
<include name="lib/jboss-archive-browsing.jar" />
<include name="lib/jboss-common.jar" />
```

- thirdparty-all.jar — **## build.xml ### # #####**

```
<include name="lib/thirdparty-all.jar" />
```

- antlr-2.7.6.jar — **## build.xml ### # #####:**

```
<include name="lib/antlr-*.jar" />
```

- **#####Drools###Seam**      **Security#####Eclipse**      **JDT#####JBoss**  
**AS#####OC4J#classloading#####**

•

```
cp ../jboss-seam/seam-gen/lib/org.eclipse.jdt.core*.jar lib/
```

- **## build.xml #####ear###**

```
<include name="lib/org.eclipse.jdt.core*.jar" />
```

**#####**

```
<fileset dir="${basedir}">
  <!-- other libraries added by seam-gen -->
  <include name="lib/hibernate-annotations.jar" />
  <include name="lib/hibernate-entitymanager.jar" />
  <include name="lib/hibernate3.jar" />
  <include name="lib/jboss-archive-browsing.jar" />
  <include name="lib/jboss-common.jar" />
  <include name="lib/thirdparty-all.jar" />
  <include name="lib/antlr-*.jar" />
  <include name="lib/org.eclipse.jdt.core*.jar" />
</fileset>
```

**#####**      User      **#####Seam**      **Security#**      **#####username**      **##**      password      **##**      User      **###**  
**#####Bean###OC4J##EJB3#####**

1.

- **## @Stateless ###**
- **##### AuthenticatorAction**
- **##### Authenticator ##### AuthenticatorAction #####EJB3#####Bean##### ##**  
     @Local **#####AuthenticatorAction ## authenticate #####**

```
@Name("authenticator") @Stateless public class
```

```
AuthenticatorAction implements Authenticator {
```

```
@Local public interface Authenticator {
    public boolean authenticate();
}
```

## 2. ## @PersistenceContext #####EntityManager#

```
@PersistenceContext private EntityManager entityManager;
```

## 3. ##authenticate#

```
public boolean authenticate() {
    List <User> users = entityManager.createQuery("select u from User u where
    u.username = #{identity.username} and
    u.password = #{identity.password}") .getResultList();
    if (users.size() == 1) {
        identity.addRole("admin");
        return true;
    } else {
        return false;
    }
}
```

## 4. ### web.xml ###EJB3###

```
<ejb-local-ref>
  <ejb-ref-name>
    oc4j-example/AuthenticatorAction/local
  </ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local>
    org.jboss.seam.tutorial.oc4j.action.Authenticator
  </local>
  <ejb-link>AuthenticatorAction</ejb-link>
</ejb-local-ref>
```

```
#####
```

## 26.3.1. seam-gen#####OC4J#####

```
#####
```

```
$ORACLE_HOME/j2ee/home/config/data-sources.xml
```

```
<managed-data-source
  connection-pool-name="oc4j-example-connection-pool"
  jndi-name="jdbc/___oc4jExample"
  name="oc4j-example-managed-data-source" />
<connection-pool
  name="oc4j-example-connection-pool">
  <connection-factory
    factory-class="com.mysql.jdbc.Driver"
    user="username"
    password="password"
    url="jdbc:mysql:///oc4j" />
  </connection-factory>
</connection-pool>
```

```
$ORACLE_HOME/j2ee/home/config/server.xml
```

```
<application name="oc4j-example"
  path="../../home/applications/oc4j-example.ear"
  parent="default"
  start="true" />
```

\$ORACLE\_HOME/j2ee/home/config/default-web-site.xml

```
<web-app application="oc4j-example"
  name="oc4j-example"
  load-on-startup="true"
  root="/oc4j-example" />
```



---

## Chapter 27. Seam##

##Seam#####Seam#####EJB3.0#####  
Validator#####Seam#####

##### org.jboss.seam.annotations #####

### 27.1. #####

#####Seam#####component#####

@Name

```
@Name( " componentName" )
```

#####Seam#####Seam#####

@Scope

```
@Scope( ScopeType.CONVERSATION)
```

##### ScopeType ###EVENT, PAGE, CONVERSATION, SESSION, BUSINESS\_PROCESS, AP-  
PLICATION, STATELESS#

##### bean#### STATELESS# ##Entity Bean#Stateful Session Bean####  
CONVERSATION# ##JavaBean#### EVENT#

@Role

```
@Role( name=" roleName" , scope=ScopeType.SESSION)
```

####Seam##### @Name/@Scope #####“####”#### @Role #####

- name — #####
- scope — #####

@Roles

```
@Roles( {  
    @Role( name=" user" , scope=ScopeType.CONVERSATION) ,  
    @Role( name=" currentUser" , scope=ScopeType.SESSION)  
} )
```

#####

@BypassInterceptors

```
@BypassInterceptors
```

#####

@JndiName

```
@JndiName( "my/jndi/name" )
```

```
Seam##EJB###JNDI##      #####JNDI##Seam####      org.jboss.seam.core.init.jndiPattern
###JNDI###
```

```
@Conversational
```

```
@Conversational
```

```
#####
```

```
@Startup
```

```
@Scope( APPLICATION ) @Startup( depends="org.jboss.seam.bpm.jbpm" )
```

```
####Application Scope#####JNDI#####
```

```
@Scope( SESSION ) @Startup
```

```
####Session Scope####Session#####
```

- depends — #####

```
@Install
```

```
@Install( false )
```

```
#####@Install#####
```

```
@Install( dependencies="org.jboss.seam.bpm.jbpm" )
```

```
#####
```

```
@Install( genericDependencies=ManagedQueueSender.class )
```

```
#####
```

```
@Install( classDependencies="org.hibernate.Session" )
```

```
#####classpath#####
```

```
@Install( precedence=BUILT_IN )
```

```
#####
```

- BUILT\_IN — #####Seam#####
- FRAMEWORK — ####Seam#####
- APPLICATION — #####
- DEPLOYMENT — #####
- MOCK — ####mock#####

@Synchronized

```
@Synchronized(timeout=1000)
```

```
#####Seam#####
```

@ReadOnly

```
@ReadOnly
```

```
##JavaBean#####
```

@AutoCreate

```
@AutoCreate
```

```
##### create=true#
```

## 27.2. #####

```
#####
```

@In

```
@In
```

```
#####null#####
```

```
@In(required=false)
```

```
#####null#
```

```
@In(create=true)
```

```
#####null###Seam#####
```

```
@In(value="contextVariableName")
```

```
#####
```

```
@In(value="#{customer.addresses['shipping']}")
```

```
#####JSF EL#####
```

- value — #####JSF EL##### #{...} ####
- create — #####Seam#####false#
- required — #####Seam#####

@Out

```
@Out
```

#####Seam#####null#####

```
@Out(required=false)
```

#####Seam#####null#

```
@Out(scope=ScopeType.SESSION)
```

#####Seam#####scope#

#####scope##### @Out #####scope##### EVENTscope ##

```
@Out(value="contextVariableName")
```

#####

- value — #####
- required — #####null#Seam#####

#####

```
@In(create=true) @Out private User currentUser;
```

##### #####manager component# #####Seam#####class#####getter#####

@Unwrap

```
@Unwrap
```

#####getter#####

##### #####factory component# #####Seam#####  
#####faces#request####response#####

@Factory

```
@Factory("processInstance") public void createProcessInstance() { ... }
```

##### void ####

```
@Factory("processInstance", scope=CONVERSATION) public ProcessInstance createProcessInstance() { ... }
```

#####Seam##### #####scope#### @Factory  
#####scope##### EVENT #####

- value — #####getter#####JavaBean####
- scope — ##Seam#####
- autoCreate — #####@In###create=true#

##### ##Log##

@Logger

```
@Logger("categoryName")
```

```
## org.jboss.seam.log.Log #####Entity Bean#####static#
```

- value — ####category#####

```
#####request####
```

@RequestParameter

```
@RequestParameter("parameterName")
```

```
#request#####
```

- value — ##request#####

## 27.3. #####

```
#####class#####
```

@Create

```
@Create
```

```
#####Seam#####JavaBean#Stateful Session Bean##create###
```

@Destroy

```
@Destroy
```

```
#####JavaBean#Stateful Session Bean##destroy###
```

```
Destroy#####Seam #####destroy#####
```

@Observer

```
@Observer("somethingChanged")
```

```
#####component-driven#####
```

```
@Observer(value="somethingChanged", create=false)
```

```
##### create#false#####create####true#
```

## 27.4. #####

```
#####declarative conversation demarcation#####Seam#####Action Listener
```

## Method##

```
##Web##### @Begin
####'##'#####long-running conversation##
```

```
@Begin
```

```
@Begin
```

```
#####
```

```
@Begin(join=true)
```

```
#####
```

```
@Begin(nested=true)
```

```
#####      ##nested#      #####      #####      @End
#####
```

```
@Begin(pageflow="process definition name")
```

```
#####pageflow##jBPM#####
```

```
@Begin(flushMode=FlushModeType.MANUAL)
```

```
####Seam#####flush###flushMode=FlushModeType.MANUAL      ##      #####atomic
conversations##### flush()#####
```

- join — ##### ##true##### false##### ##false#### nested=true  
#####
- nested — #####
- flushMode — #####Seam###Hibernate Session#JPA#####flush###
- pageflow — # org.jboss.seam.bpm.jbpm.pageflowDefinitions #####jBPM#####

```
@End
```

```
@End
```

```
#####
```

- beforeRedirect — #####destory#  
##beforeRedirect=true#####request#####

```
@StartTask
```

```
@StartTask
```

```
###jBPM#####request#####jBMP#####
#####business process context#####
```

```
jBPM# TaskInstance #request context## taskInstance ##### jBPM# ProcessInstance #request
context## processInstance ##### @In ####
```

- taskIdParameter — ###task id#request##### ##"taskId"#####Seam taskList JSF component#####
- flushMode — #####Seam###Hibernate Session#JPA#####flush###

@BeginTask

```
@BeginTask
```

```
#####jBPM#####request#####jBPM#####
#####business process context#####
```

```
jBPM# org.jbpm.taskmgmt.exe.TaskInstance #request context## taskInstance ##### jBPM#
org.jbpm.graph.exe.ProcessInstance #request context## processInstance #####
```

- taskIdParameter — ###task id#request##### ##"taskId"#####Seam taskList JSF component#####
- flushMode — #####Seam###Hibernate Session#JPA#####flush###

@EndTask

```
@EndTask
```

```
""###jBPM#####jBPM ###transition## ##### transition #####
Transition.setName() #####transition#
```

```
@EndTask(transition="transitionName")
```

```
####jBPM###
```

- transition — #####jBPM#####transition#
- beforeRedirect — #####destory# ## beforeRedirect=true  
#####request#####

@CreateProcess

```
@CreateProcess(definition="process definition name")
```

```
#####jBPM##### ProcessInstance ##### processInstance #####
```

- definition — ## org.jboss.seam.bpm.jbpm.processDefinitions ###jBPM #####

@ResumeProcess

```
@ResumeProcess(processIdParameter="processId")
```

```
#####jBPM #####context#ProcessInstance ##### processInstance
#####
```

- processIdParameter — #####id#request##### "processId".

@Transition

```
@Transition("cancel")
```

#####jBPM#####

## 27.5. ###J2EE#####Seam JavaBean#####

Seam#####JTA###

@Transactional

```
@Transactional
```

##JavaBean#####Session

Bean#####

#####

#####EJB3.0##### @TransactionAttribute#

@ApplicationException

```
@Transactional
```

TDB

@Interceptors

```
@Transactional
```

TDB

#####JavaBean Seam#####EJB3.0##### @TransactionAttribute ###

## 27.6. #####

#####Seam#####Seam#####

@Redirect

```
@Redirect(viewId="error.jsp")
```

#####id#

- viewId — #####JSF##id#####EL#
- message — #####
- end — ##### false#



@HttpError

```
@HttpError(errorCode=404)
```

#####HTTP###

- errorCode — HTTP#####500#
- message — #HTTP#####
- end — ##### false#

## 27.7. ##Seam Remoting ###

Seam Remoting####Bean#####

@WebRemote

```
@WebRemote(exclude="path.to.exclude")
```

#####JavaScript#### exclude #####Remoting#####

## 27.8. ##Seam####interceptor####

#####Seam#####

###EJB3.0#####EJB#####

@Interceptor

```
@Interceptor(stateless=true)
```

#####Seam#####

```
@Interceptor(type=CLIENT)
```

#####“###”#####EJB#####

```
@Interceptor(around={SomeInterceptor.class, OtherInterceptor.class})
```

#####

```
@Interceptor(within={SomeInterceptor.class, OtherInterceptor.class})
```

#####

## 27.9. #####asynchronicity####

#####

```
@Asynchronous public void scheduleAlert(Alert alert, @Expiration Date date) { ... }
```

```
@Asynchronous public Timer scheduleAlerts(Alert alert,
                                           @Expiration Date date,
                                           @IntervalDuration long interval) { ... }
```

@Asynchronous

```
@Asynchronous
```

```
#####
```

@Duration

```
@Duration
```

```
#####
```

@Expiration

```
@Expiration
```

```
#####
```

@IntervalDuration

```
@IntervalDuration
```

```
#####
```

## 27.10. ##JSF###

```
#####JSF#####
```

@Converter

```
####Seam####JSF####JSF converter)# #####Seam#####javax.faces.convert.Converter###
```

- id — JSF###id#####
- forClass — #####

@Validator

```
####Seam####JSF####JSF Validator## #####Seam##### javax.faces.validator.Validator
###
```

- id — JSF###id#####

### 27.10.1. # dataTable #####

```
#####Stateful Session Bean#####attributes)####
```

@DataModel

```
@DataModel("variableName")
```

```
##### List#Map#Set # Object[] #####JSF DataModel #####scope##### STATELESS##EVENT
scope ## ### Map ##DataModel ##### Map.Entry#
```

- value — #####
- scope — #scope=ScopeType.PAGE ##### DataModel ## PAGE #####

@DataModelSelection

```
@DataModelSelection
```

```
#JSF DataModel##### @DataModel ##### DataModel #####
##### @DataModel ##### @DataModelSelection #####
```

```
##### @DataModel ##### PAGE scope#####DataModel Selection,####DataModel#####
@DataModel #####getter####Seam#####setter####Business API####
```

- value — ##### @DataModel#####

@DataModelSelectionIndex

```
@DataModelSelectionIndex
```

```
##JSF DataModel #####map key## ##### @DataModel ##### DataModel
##### @DataModel ##### @DataModelSelection #####
```

- value — ##### @DataModel#####

## 27.11. #####

```
#####list##### @DataModel # @DataModelSelection #####
```

@DataBinderClass

```
@DataBinderClass(DataModelBinder.class)
```

```
#####
```

@DataSelectorClass

```
@DataSelectorClass(DataModelSelector.class)
```

```
#####dataselection###
```

## 27.12. #####packing)###

#####Java##

@Namespace

```
@Namespace(value="http://jboss.com/products/seam/example/seampay")
```

##### components.xml #####XML#####

```
@Namespace(value="http://jboss.com/products/seam/core", prefix="org.jboss.seam.core")
```

#####XML##### init  
#XML##### org.jboss.seam.core.init ####

## 27.13. ###Servlet#####

#####Seam###Servlet#####

@Filter

#### @Filter ###Seam##### javax.servlet.Filter #####Servlet  
####Filter#####Seam##Filter###

•

```
@Filter(around={"seamComponent", "otherSeamComponent"})
```

#####

•

```
@Filter(within={"seamComponent", "otherSeamComponent"})
```

#####

---

## Chapter 28. ##Seam##

```
#####Seam##### components.xml #####
#####components.xml #####

##### @Name #####

##### auto-create="true"#####
create=true#
```

### 28.1. #####

```
#####Seam#####:
```

```
@In private Context sessionContext;
```

```
org.jboss.seam.core.contexts
    #####Seam##### org.jboss.seam.core.contexts.sessionContext['user']#

org.jboss.seam.faces.facesContext
    FacesContext #####Seam####

#####
```

### 28.2. ####

```
#####
```

```
org.jboss.seam.faces.facesMessages
    ##faces#####
```

- add(FacesMessage facesMessage) — #####faces#####
- add(String messageTemplate) — #####faces#####EL#####
- add(Severity severity, String messageTemplate) —  
#####faces#####EL#####
- addFromResourceBundle(String key) —  
#####faces#####Seam#####EL####
- addFromResourceBundle(Severity severity, String key) —  
#####faces#####Seam#####EL####
- clear() — #####

```
org.jboss.seam.faces.redirect
    #####API#####
```

- `redirect.viewId` — #####JSF##ID#
- `redirect.conversationPropagationEnabled` — #####
- `redirect.parameters` — #####
- `execute()` — #####
- `captureCurrentRequest()` — ####ID###GET##### `execute()`#

`org.jboss.seam.faces.httpError`  
#####HTTP###API#

`org.jboss.seam.core.events`  
#####API# ##### @Observer ##### components.xml #####

- `raiseEvent(String type)` — #####
- `raiseAsynchronousEvent(String type)` — #####EJB3#####
- `raiseTimedEvent(String type, ...)` — #####EJB3#####
- `addListener(String type, String methodBinding)` — #####

`org.jboss.seam.core.interpolator`  
#####JSF EL####API#

- `interpolate(String template)` — ##JSF EL#### #{...}#####

`org.jboss.seam.core.expressions`  
#####API#

- `createValueBinding(String expression)` — #####
- `createMethodBinding(String expression)` — #####

`org.jboss.seam.core.pojoCache`  
JBoss ##PojoCache #####

- `pojoCache.cfgResourceName` — ##### treecache.xml#

#####

## 28.3. #####

#####Seam#####

`org.jboss.seam.core.locale`  
Seam locale#####

`org.jboss.seam.international.timezone`  
Seam#####

org.jboss.seam.core.resourceBundle

Seam#####stateless#####JAVA#####

org.jboss.seam.core.resourceLoader

#####

- resourceLoader.bundleNames — #####Java#####Seam##### messages#

org.jboss.seam.international.localeSelector

##locale#####

- select() — #####locale#
- localeSelector.locale — ### java.util.Locale#
- localeSelector.localeString — locale#####
- localeSelector.language — ##locale###
- localeSelector.country — ##locale###
- localeSelector.variant — ##locale###
- localeSelector.supportedLocales — ## SelectItems #####locales#### jsf-config.xml####
- localeSelector.cookieEnabled — ##locale#####

org.jboss.seam.international.timezoneSelector

#####

- select() — #####locale#####
- timezoneSelector.timezone — ### java.util.TimeZone#
- timezoneSelector.timeZoneId — #####
- timezoneSelector.cookieEnabled — #####

org.jboss.seam.international.messages

#####Seam#####

org.jboss.seam.theme.themeSelector

#####

- select() — #####
- theme.availableThemes — #####
- themeSelector.theme — #####
- themeSelector.themes — ##SelectItems #####
- themeSelector.cookieEnabled — #####

```
org.jboss.seam.theme.theme
#####
```

```
#####
```

## 28.4. #####

```
#####
```

```
org.jboss.seam.core.conversation
#####Seam#####API#
```

- getId() — #####ID
- isNested() — #####
- isLongRunning() — #####
- getId() — #####ID
- getParentId() — #####ID
- getRootId() — #####ID
- setTimeout(int timeout) — #####
- setViewId(String outcome) — #####breadcrumbs#####ID#
- setDescription(String description) — #####breadcrumbs##
- redirect() — #####ID#####
- leave() — #####
- begin() — #####@Begin)#
- beginPageflow(String pageflowName) — #####@Begin(pageflow="..."))#
- end() — ##### @End)#
- pop() — #####
- root() — #####
- changeFlushMode(FlushModeType flushMode) — #####

```
org.jboss.seam.core.conversationList
#####
```

```
org.jboss.seam.core.conversationStack
#####breadcrumbs#####
```

```
org.jboss.seam.faces.switcher
#####
```



#####

## 28.5. #jBPM####

#####jBPM####

```
org.jboss.seam.pageflow.pageflow
  #####Seam####API#
```

- `IsInProcess()`##### `true`
- `getProcessInstance()` — #####jBPM `ProcessInstance` #####
- `begin(String pageflowName)` — #####
- `reposition(String nodeName)` — #####

```
org.jboss.seam.bpm.actor
  #####Session###jBPM#####API
```

- `setId(String actorId)` — #####jBPM####
- `getGroupActorIds()` — #####jBPM#####`Set`#

```
org.jboss.seam.bpm.transition
  #####jBPM#####API#
```

- `setName(String transitionName)` — ##### `@EndTask` #####jBPM#####

```
org.jboss.seam.bpm.businessProcess
  #####API#
```

- `businessProcess.taskId` — #####
- `businessProcess.processId` — #####
- `businessProcess.hasCurrentTask()` — #####
- `businessProcess.hasCurrentProcess()` — #####
- `createProcess(String name)` — #####
- `startTask()` — #####
- `endTask(String transitionName)` — #####
- `resumeTask(Long id)` — #####
- `resumeProcess(Long id)` — #####
- `transition(String transitionName)` — #####

```

org.jboss.seam.bpm.taskInstance
  jBPM TaskInstance #####

org.jboss.seam.bpm.processInstance
  jBPM ProcessInstance #####

org.jboss.seam.bpm.jBPMContext
  ##### jBPMContext #####

org.jboss.seam.bpm.taskInstanceList
  jBPM#####

org.jboss.seam.bpm.pooledTaskInstanceList
  jBPM#####

org.jboss.seam.bpm.taskInstanceListForType
  jBPM#####

org.jboss.seam.bpm.pooledTask
  #####

org.jboss.seam.bpm.processInstanceFinder
  #####

org.jboss.seam.bpm.processInstanceList
  #####

## org.jboss.seam.core.jBPM#####

```

## 28.6. #####

```
#####Web#####
```

```

org.jboss.seam.web.userPrincipal
  #####(##)Principal#####

org.jboss.seam.web.isUserInRole
  #JSF#####                                <h:commandButton                                value="edit"
  rendered="#{isUserInRole['admin']}" />#

```

## 28.7. #JMS####

```
##### TopicPublisherS ##### QueueSenderS #####
```

```

org.jboss.seam.jms.queueSession
  JMS QueueSession #####

org.jboss.seam.jms.topicSession
  JMS TopicSession #####

```

## 28.8. #####

#####Seam#####

org.jboss.seam.mail.mailSession

JavaMail Session ##### Session#####JNDI##### sessionJndiName#####  
#####host #####

- org.jboss.seam.mail.mailSession.host — ##SMTP#####
- org.jboss.seam.mail.mailSession.port — ##SMTP#####
- org.jboss.seam.mail.mailSession.username — #####SMTP#####
- org.jboss.seam.mail.mailSession.password — #####SMTP#####
- org.jboss.seam.mail.mailSession.debug — ##JavaMail debug#####
- org.jboss.seam.mail.mailSession.ssl — ###SMTP#SSL#####465#  
org.jboss.seam.mail.mailSession.tls — ###true#####TLS##
- org.jboss.seam.mail.mailSession.sessionJndiName —  
#JNDI###javax.mail.Session#####

## 28.9. ####

##### components.xml ##### install="true" #####

org.jboss.seam.core.init

#Seam#####

- org.jboss.seam.core.init.jndiPattern — JNDI#####Bean#
- org.jboss.seam.core.init.debug — ## Seam #####
- org.jboss.seam.core.init.clientSideConversations — #### true#Seam##### HttpSession #####
- org.jboss.seam.core.init.userTransactionName — ###JTA UserTransaction #####JNDI##

org.jboss.seam.core.manager

#####Seam#####

- org.jboss.seam.core.manager.conversationTimeout — #####
- org.jboss.seam.core.manager.concurrentRequestTimeout —  
#####
- org.jboss.seam.core.manager.conversationIdParameter — #####ID#####  
conversationId#

- org.jboss.seam.core.manager.conversationIsLongRunningParameter  
#####conversationIsLongRunning#

org.jboss.seam.navigation.pages  
#####Seam#####

- org.jboss.seam.core.pages.noConversationViewId — #####ID#####
- org.jboss.seam.navigation.pages.loginViewId — #####ID#####
- org.jboss.seam.navigation.pages.httpPort — http#####
- org.jboss.seam.navigation.pages.httpsPort — https#####
- org.jboss.seam.navigation.pages.resources — ##### pages.xml ##### WEB-INF/pages.xml#

org.jboss.seam.bpm.jBPM  
## jBPMConfiguration #jBPM##### org.jboss.seam.bpm.jBPM#

- org.jboss.seam.core.jBPM.processDefinitions — ####jPDL#####orchestration##
- org.jboss.seam.core.jBPM.pageflowDefinitions — #####jPDL#####orchestration##

org.jboss.seam.core.conversationEntries  
###Session#####

org.jboss.seam.faces.facesPage  
#####

org.jboss.seam.persistence.persistenceContexts  
#####

org.jboss.seam.jms.queueConnection  
####JMS QueueConnection##### QueueSender#####

- org.jboss.seam.jms.queueConnection.queueConnectionFactoryJndiName — JMS QueueConnectionFactory#####JNDI##### UIL2ConnectionFactory#

org.jboss.seam.jms.topicConnection  
####JMS TopicConnection##### TopicPublisher #####

- org.jboss.seam.jms.topicConnection.topicConnectionFactoryJndiName — ##JMS TopicConnectionFactory #####JNDI##### UIL2ConnectionFactory

org.jboss.seam.persistence.persistenceProvider  
#####JPA#####

org.jboss.seam.core.validators  
##Hibernate Validator## ClassValidator#

org.jboss.seam.faces.validation

#####

org.jboss.seam.debug.introspector

##Seam####

org.jboss.seam.debug.contexts

##Seam####

org.jboss.seam.exception.exceptions

#####

org.jboss.seam.transaction.transaction

#####JTA#####API#

org.jboss.seam.faces.safeActions

#####action expression#####URL#####

## 28.10. ####

#####

org.jboss.seam.async.dispatcher

#####Bean#

org.jboss.seam.core.image

#####

org.jboss.seam.core.pojoCache

PojoCache#####

org.jboss.seam.core.uiComponent

#####Component ID####UIComponents#####

## 28.11. ####

##Seam#####Seam##### components.xml#####Seam###

```

<component name="bookingDatabase"
    class="org.jboss.seam.persistence.ManagedPersistenceContext">
    <property name="persistenceUnitJndiName">java:/comp/emf/bookingPersistence</property>
</component>

<component name="userDatabase"
    class="org.jboss.seam.persistence.ManagedPersistenceContext">
    <property name="persistenceUnitJndiName">java:/comp/emf/userPersistence</property>
</component>

```

###Seam### bookingDatabase # userDatabase#

&lt;entityManager&gt; , org.jboss.seam.persistence.ManagedPersistenceContext

##### EntityManager ##### # EntityManager #####

- `<entityManager>.entityManagerFactory` — ##### EntityManagerFactory ####  
  
`<entityManager>.persistenceUnitJndiName` — #####JNDI#####  
`java: /<managedPersistenceContext>#`

`<entityManagerFactory>` , org.jboss.seam.persistence.EntityManagerFactory  
 ####JPA EntityManagerFactory#####EJB3.0#####

- `entityManagerFactory.persistenceUnitName` — #####

###API JavaDoc#####

`<session>` , org.jboss.seam.persistence.ManagedSession  
 #####Hibernate Session#

- `<session>.sessionFactory` — ##### SessionFactory ####  
  
`<session>.sessionFactoryJndiName` — Session ###JNDI##### `java: /<managedSession>#`

`<sessionFactory>` , org.jboss.seam.persistence.HibernateSessionFactory  
 ##Hibernate SessionFactory#####

- `<sessionFactory>.cfgResourceName` — ##### hibernate.cfg.xml#

###API JavaDoc#####

`<managedQueueSender>` , org.jboss.seam.jms.ManagedQueueSender  
 #####JMS QueueSender#####

- `<managedQueueSender>.queueJndiName` — JMS###JNDI###

`<managedTopicPublisher>` , org.jboss.seam.jms.ManagedTopicPublisher  
 #####JMS TopicPublisher#####

- `<managedTopicPublisher>.topicJndiName` — JMS###JNDI###

`<managedWorkingMemory>` , org.jboss.seam.drools.ManagedWorkingMemory  
 #####Drools WorkingMemory#

- `<managedWorkingMemory>.ruleBase` — ##### RuleBase ####

`<ruleBase>` , org.jboss.seam.drools.RuleBase  
 #####Drools RuleBase#####

- `<ruleBase>.ruleFiles` — #####Drools#####

`<ruleBase>.dslFile` — (Drools)DSL###

`<entityHome>` , org.jboss.seam.framework.EntityHome

`<hibernateEntityHome>` , org.jboss.seam.framework.HibernateEntityHome

`<entityQuery> ,org.jboss.seam.framework.EntityQuery`

`<hibernateEntityQuery> ,org.jboss.seam.framework.HibernateEntityQuery`

# Chapter 29. Seam#JSF##

Seam#####Seam#JSF#####JSF#####JBoss Ajax4jsf#JBoss  
Richfaces#Apache MyFaces Trinidad#####Seam#####Tomahawk#####Seam#

## 29.1. ##

#####tagsd#####“s”#####Facelets##

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:s="http://jboss.com/products/seam/taglib">
```

ui#####

Table 29.1. Seam#JSF####

<div>&lt;s:button&gt;</div>	<div>##  #####  ##  <ul style="list-style-type: none"><li>• value — ##</li><li>• action — #####</li><li>• view — ###JSF view id#</li><li>• fragment — ###fragment####</li><li>• disabled — #####</li><li>• propagation — #####begin#join#nest#none ## end#</li><li>• pageflow — ##### #### propagation="begin" ## propaga- tion="join" #####</li></ul> ## <div>&lt;s:button id="cancel" value="Cancel"           action="#{hotelBooking.cancel}"/&gt;</div></div>
<div>&lt;s:cache&gt;</div>	<div>##  ##JBoss#Cache##### ## &lt;s:cache&gt; ##### pojoCache #####JBoss Cache####  ##  <ul style="list-style-type: none"><li>• key — ##### key="Document-#{document.id}"#</li></ul></div>



	<ul style="list-style-type: none"> <li>enabled — #####</li> <li>region — #####JBoss Cache#####</li> </ul> <p><i>Usage ##</i></p> <pre>&lt;s:cache key="entry-#{blogEntry.id}" region="pageFragments"&gt;   &lt;div class="blogEntry"&gt;     &lt;h3&gt;#{blogEntry.title}&lt;/h3&gt;     &lt;div&gt;       &lt;s:formattedText value="#{blogEntry.body}" /&gt;     &lt;/div&gt;     &lt;p&gt;       [Posted on&amp;#160;       &lt;h:outputText value="#{blogEntry.date}"&gt;         &lt;f:convertDateTime timezone="#{blog.timeZone}" locale="#{blog.locale}"           type="both" /&gt;       &lt;/h:outputText&gt;]     &lt;/p&gt;   &lt;/div&gt; &lt;/s:cache&gt;</pre>
<s:conversationId>	<p>##</p> <p>###id#####JSF#####Facelets#</p> <p>##</p> <p>##</p>
<s:conversationPropagation>	<p>##</p> <p>#####JSF#####Facelets#</p> <p>##</p> <ul style="list-style-type: none"> <li>propagation — #####begin#join#nest#none ## end#</li> <li>pageflow — ##### #### propagation="begin" ## propagation="join" #####</li> </ul> <p>##</p> <pre>&lt;h:commandButton value="Apply" action="#{personHome.update}"&gt;   &lt;s:conversationPropagation type="join" /&gt; &lt;/h:commandButton&gt;</pre>
<s:convertDateTime>	<p>##</p> <p>#Seam#timezone#####</p> <p>##</p> <p>##</p> <p>##</p> <pre>&lt;h:outputText value="#{item.orderDate}"&gt;</pre>

	<pre> &lt;s:convertDateTime type="both" dateStyle="full"/&gt; &lt;/h:outputText&gt; </pre>
<s:convertEntity>	<pre> ##  #####  ##### @Id #####  ##  ##  ##  ##### &lt;s:convertEntity /&gt; ## Seam##### ###Section 8.2, "Seam#####"#  #### ##### entityManager##### components.xml #####  &lt;component name="org.jboss.seam.ui.EntityConverter"&gt;   &lt;property name="entityManager"&gt;#{em}&lt;/property&gt; &lt;/component&gt;  #####componets.xml#####  &lt;component name="myEntityConverter" class="org.jboss.seam.ui.converter.Entity   &lt;property name="entityManager"&gt;#{em}&lt;/property&gt; &lt;/component&gt;  &lt;h:selectOneMenu value="#{person.continent}"&gt;   &lt;s:selectItems value="#{continents.resultList}" var="continent"     label="#{continent.name}" /&gt;   &lt;f:converter converterId="myEntityConverter" /&gt; &lt;/h:selectOneMenu&gt;  ##  &lt;h:selectOneMenu value="#{person.continent}" required="true"&gt;   &lt;s:selectItems value="#{continents.resultList}" var="continent"     label="#{continent.name}"     noSelectionLabel="Please Select..." /&gt;   &lt;s:convertEntity /&gt; &lt;/h:selectOneMenu&gt; </pre>
<s:convertEnum>	<pre> ##  #####enum#####  ##  ##  ##  &lt;h:selectOneMenu value="#{person.honorific}"&gt;   &lt;s:selectItems value="#{honorifics}" var="honorific"     label="#{honorific.label}"     noSelectionLabel="Please select" /&gt;   &lt;s:convertEnum /&gt; </pre>

	<pre>&lt;/h:selectOneMenu&gt;</pre>
<s:decorate>	<pre>##  ##### required="true" ##"##"##JSF####  ##  • template — #####Facelets###  #{invalid}      #      #{required}      ###      s:decorate      #### ######{required} ##### true# #####, #{invalid} ### true#  ##</pre> <pre>&lt;s:decorate template="edit.xhtml"&gt;   &lt;ui:define name="label"&gt;Country:&lt;/ui:define&gt;   &lt;h:inputText value="#{location.country}" required="true"/&gt; &lt;/s:decorate&gt;</pre> <pre>&lt;ui:composition xmlns="http://www.w3.org/1999/xhtml"   xmlns:ui="http://java.sun.com/jsf/facelets"   xmlns:h="http://java.sun.com/jsf/html"   xmlns:f="http://java.sun.com/jsf/core"   xmlns:s="http://jboss.com/products/seam/taglib"&gt;    &lt;div&gt;      &lt;s:label styleClass="#{invalid?'error':''}"&gt;       &lt;ui:insert name="label"/&gt;       &lt;s:span styleClass="required" rendered="#{required}"&gt;*&lt;/s:span&gt;     &lt;/s:label&gt;      &lt;span class="#{invalid?'error':''}"&gt;       &lt;s:validateAll&gt;         &lt;ui:insert/&gt;       &lt;/s:validateAll&gt;     &lt;/span&gt;      &lt;s:message styleClass="error"/&gt;    &lt;/div&gt;  &lt;/ui:composition&gt;</pre>
<s:div>	<pre>##  ####HTML&lt;div&gt;#  ##  ##  ##</pre> <pre>&lt;s:div rendered="#{selectedMember == null}"&gt;   Sorry, but this member does not exist. &lt;/s:div&gt;</pre>

<s:enumItem>	<pre>##  ###enum##### SelectItem#  ##  • enumValue — #enum#####  • label — ### SelectItem #####  ##  &lt;h:selectOneRadio id="radioList"                   layout="lineDirection"                   value="#{newPayment.paymentFrequency}"&gt;   &lt;s:convertEnum /&gt;   &lt;s:enumItem enumValue="ONCE" label="Only Once" /&gt;   &lt;s:enumItem enumValue="EVERY_MINUTE" label="Every Minute" /&gt;   &lt;s:enumItem enumValue="HOURLY" label="Every Hour" /&gt;   &lt;s:enumItem enumValue="DAILY" label="Every Day" /&gt;   &lt;s:enumItem enumValue="WEEKLY" label="Every Week" /&gt; &lt;/h:selectOneRadio&gt;</pre>
<s:fileUpload>	<pre>##  #####form### multipart/form-data #####  &lt;h:form enctype="multipart/form-data"&gt;  ##### web.xml ###Seam Multipart Servlet####  &lt;filter&gt;   &lt;filter-name&gt;Seam Filter&lt;/filter-name&gt;   &lt;filter-class&gt;org.jboss.seam.servlet.SeamFilter&lt;/filter-class&gt; &lt;/filter&gt;  &lt;filter-mapping&gt;   &lt;filter-name&gt;Seam Filter&lt;/filter-name&gt;   &lt;url-pattern&gt;/*&lt;/url-pattern&gt; &lt;/filter-mapping&gt;  ##  ##### components.xml #####  • createTempFiles — #####true#####  • maxRequestSize — #####  #####  &lt;component class="org.jboss.seam.web.MultipartFilter"&gt;   &lt;property name="createTempFiles"&gt;true&lt;/property&gt;   &lt;property name="maxRequestSize"&gt;1000000&lt;/property&gt; &lt;/component&gt;  ##</pre>

	<ul style="list-style-type: none"> <li>• data — ##### byte[] ## InputStream #####</li> <li>• contentType — #####</li> <li>• fileName — #####</li> <li>• fileSize — #####</li> <li>• accept — ##### ## "images/ png,images/jpg"#"images/"#</li> <li>• style — #####CSS###</li> <li>• styleClass — #####</li> </ul> <p>##</p> <pre>&lt;s:fileUpload id="picture" data="#{register.picture}"     accept="image/png"     contentType="#{register.pictureContentType}" /&gt;</pre>
<s:formattedText>	<p>##</p> <p>## Seam Text#####Wiki#####Seam Text###</p> <p>##</p> <ul style="list-style-type: none"> <li>• value — #####EL####</li> </ul> <p>##</p> <pre>&lt;s:formattedText value="#{blog.text}" /&gt;</pre> <p>##</p> <div> <div>Please type your comment</div> <div> <pre>+Lorem ipsum *Lorem ipsum* /dolor sit amet/, [consectetur adipiscing elit]. -Suspendisse a risus- ^quis lorem pharetra viverra^. _Fusce in ipsum. Nam et turpis id arcu lobortis dapibus_. ++Curabitur et sem vel quam #venenatis mattis. #Nulla hendrerit orci ut massa. "Donec condimentum, libero in iaculis hendrerit, risus dolor congue nulla, non accumsan ante risus et ipsum. "Suspendisse dui. Maecenas lorem. Maecenas sit amet purus nec metus sodales sagittis. Phasellus varius lacus nec velit."</pre> </div> </div> <div> <div>Preview</div> <div> <h2>Lorem ipsum</h2> <p><b>Lorem ipsum dolor sit amet</b>, consectetur adipiscing elit. -Suspendisse a risus- quis lorem pharetra viverra, <u>Fusce in ipsum. Nam et turpis id arcu lobortis dapibus.</u></p> <h3>Curabitur et sem vel quam</h3> <ol style="list-style-type: none"> <li>1. venenatis mattis.</li> <li>2. Nulla hendrerit orci ut massa.</li> <li>3. Donec condimentum,</li> </ol> <ul style="list-style-type: none"> <li>• libero in iaculis hendrerit,</li> <li>• risus dolor congue nulla,</li> <li>• non accumsan ante risus et ipsum.</li> </ul> <p>"Suspendisse dui. Maecenas lorem. Maecenas sit amet purus nec metus sodales sagittis. Phasellus varius lacus nec velit."</p> </div> </div>

<pre>&lt;s:validateFormattedText&gt;</pre>	<pre>##  #####Seam Text  ##  ##</pre>
<pre>&lt;s:fragment&gt;</pre>	<pre>##  #####  ##  ##  ##  <div data-bbox="491 775 1437 882" style="border: 1px solid black; padding: 5px;"> <pre>&lt;s:fragment rendered="#{auction.highBidder ne null}"&gt;   Current bid: &lt;/s:fragment&gt;</pre> </div></pre>
<pre>&lt;s:graphicImage&gt;</pre>	<pre>##  #####Seam Component##### &lt;h:graphicImage&gt;#####  ## &lt;h:graphicImage&gt; #####  ##  • value — ##### String##classpath#### byte[]#java.io.File#java.io.InputStream ## java.net.URL# ##### image/png#image/jpeg # image/gif#  • fileName — ##### #####  ##  #####Seam#####  &lt;s:transformImageSize&gt; s:transformImageSize  • width — #####  • height — #####  • maintainRatio — ### true##### ## width/height#####  • factor — #####  &lt;s:transformImageBlur&gt;</pre>

	<ul style="list-style-type: none"> <li>radius — #####</li> </ul> <pre>&lt;s:transformImageType&gt;</pre> <ul style="list-style-type: none"> <li>contentType — ##### image/jpeg ## image/png</li> </ul> <pre>##### — #####                                ##                                # org.jboss.seam.ui.graphicImage.ImageTransform # UIComponent# # ap- plyTransform() ##### image.getBufferedImage() ##### im- age.setBufferedImage() #####</pre> <pre>##</pre> <pre>&lt;s:graphicImage rendered="#{auction.image ne null}"                 value="#{auction.image.data}"&gt;   &lt;s:transformImageSize width="200" maintainRatio="true"/&gt; &lt;/s:graphicImage&gt;</pre>
<pre>&lt;s:link&gt;</pre>	<pre>##</pre> <pre>#####</pre> <pre>##</pre> <ul style="list-style-type: none"> <li>value — ###</li> <li>action — #####</li> <li>view — ###JSF view id#</li> <li>fragment — ###fragment####</li> <li>disabled — #####</li> <li>propagation — #####begin#join#nest#none ## end#</li> <li>pageflow — ##### ##### propagation="begin" ## propaga- tion="join"#####</li> </ul> <pre>##</pre> <pre>&lt;s:link id="register" view="/register.xhtml"         value="Register New User"/&gt;</pre>
<pre>&lt;s:message&gt;</pre>	<pre>##</pre> <pre>”##“#####JSF####</pre> <pre>##</pre> <pre>##</pre> <pre>##</pre> <pre>&lt;f:facet name="afterInvalidField"&gt;</pre>

	<pre> &lt;s:span&gt;   &amp;#160;Error:&amp;#160; &lt;s:message/&gt; &lt;/s:span&gt; &lt;/f:facet&gt; </pre>
<s:label>	<pre> ##  #####JSF#####HTML  &lt;label&gt;  #####JSF#####  #### &lt;s:decorate&gt; ###  Attributes ##  • style — #####  • styleClass — #####  ##  &lt;s:label styleClass="label"&gt;   Country: &lt;/s:label&gt; &lt;h:inputText value="#{location.country}" required="true"/&gt; </pre>
<s:remote>	<pre> ##  #Seam Remoting#####JavaScript###stub##  ##  • include — #####Seam Remoting JavaScript ##### Chapter 21, Remoting#  ##  &lt;s:remote include="customerAction,accountAction,com.acme.MyBean" /&gt; </pre>
<s:selectDate>	<pre> ##  ##### # &lt;rich:calendar /&gt; ###  #####selectDate #####HTML#####  #####  ##  ##CSS##### ###Seam booking demo###CSS#### date.css #####seam-gen### #####CSS#####  ##  • for — #####id#  • dateFormat — #####  • startYear — ##### </pre>



- endYear — #####
  - firstDayOfWeek — #####0 = Sunday, 6 = Saturday##  
#####
- ##

```
<div class="row">
  <h:outputLabel for="dob">Date of birth<em>*</em></h:outputLabel>
  <h:inputText id="dob" value="#{user.dob}" required="true">
    <s:convertDateTime pattern="MM/dd/yyyy"/>
  </h:inputText>
  <s:selectDate for="dob" startYear="1910" endYear="2007">
    
  </s:selectDate>
  <div class="validationError"><h:message for="dob"/></div>
</div>
```

##

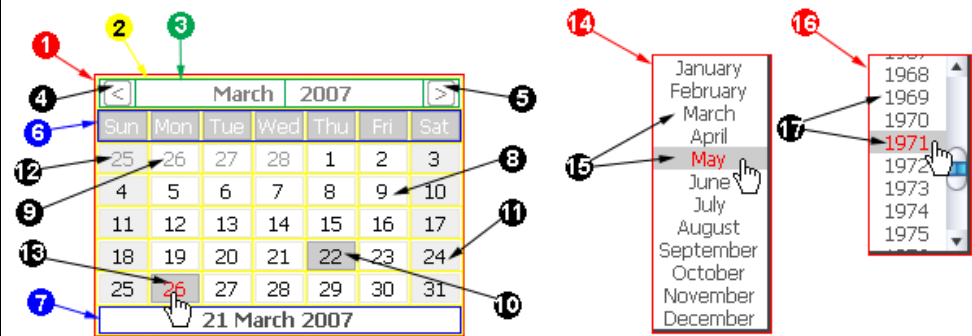


CSS##

#####selectDate#####CSS###

- seam-date — ##### div# (1) ##### table# (2)
- seam-date-header — #####tr#####td## (3)
- seam-date-header-prevMonth —  
##'####'#####td##### (4)
- seam-date-header-nextMonth —  
#####'####'#####td##### (5)
- seam-date-headerDays — #####header##tr##### (6)
- seam-date-footer — #####footer##tr##### (7)
- seam-date-inMonth — #####(td)## (8)

- seam-date-outMonth — #####td#### (9)
- seam-date-selected — #####td### (10)
- seam-date-dayOff-inMonth —  
#####”##“##### (11)
- seam-date-dayOff-outMonth —  
##### (12)
- seam-date-hover — #####td#### (13)
- seam-date-monthNames — ##### div ### (14)
- seam-date-monthNameLink — #####anchor#a### (15)
- seam-date-years — #####div### (16)
- seam-date-yearLink — #####anchor#a#### (17)



<s:selectItems>

##

###List#Set#DataModel##Array##### List<SelectItem> #

##

- value — ##EL##### List<SelectItem> ####
- var — #####
- label — ## SelectItem ##### var ###
- disabled — ###true#SelectItem ##### var ###
- noSelectionLabel — #####  
required="true"#####
- hideNoSelectionLabel — ###true#####noSelectionLabel #####

##

```
<h:selectOneMenu value="#{person.age}"
                  converter="#{converters.ageConverter}">
  <s:selectItems value="#{ages}" var="age" label="#{age}" />
</h:selectOneMenu>
```

<code>&lt;s:span&gt;</code>	<pre>##  ####HTML#&lt;span&gt;#  ##  ##  ##  &lt;s:span styleClass="required" rendered="#{required}"&gt;*&lt;/s:span&gt;</pre>
<code>&lt;s:taskId&gt;</code>	<pre>##  ##### #{task} #####id#####JSF#####Facelets#  ##  ##</pre>
<code>&lt;s:validate&gt;</code>	<pre>##  ##### Hibernate Validator #####JSF####  ##  ##  ##  &lt;h:inputText id="userName" required="true"             value="#{customer.userName}"&gt;     &lt;s:validate /&gt; &lt;/h:inputText&gt; &lt;h:message for="userName" styleClass="error" /&gt;</pre>
<code>&lt;s:validateAll&gt;</code> <code>s:validateAll</code>	<pre>##  ##### Hibernate Validator #####JSF####  ##  ##  ##  &lt;s:validateAll&gt; &lt;div class="entry"&gt;     &lt;h:outputLabel for="username"&gt;Username:&lt;/h:outputLabel&gt;     &lt;h:inputText id="username" value="#{user.username}"                 required="true"/&gt;     &lt;h:message for="username" styleClass="error" /&gt; &lt;/div&gt; &lt;div class="entry"&gt;     &lt;h:outputLabel for="password"&gt;Password:&lt;/h:outputLabel&gt;     &lt;h:inputSecret id="password" value="#{user.password}"                 required="true"/&gt;     &lt;h:message for="password" styleClass="error" /&gt; &lt;/div&gt;</pre>

```

<div class="entry">
  <h:outputLabel for="verify">Verify Password:</h:outputLabel>
  <h:inputSecret id="verify" value="#{register.verify}"
    required="true"/>
  <h:message for="verify" styleClass="error" />
</div>
</s:validateAll>

```

## 29.2. ##

#####Seam####JSF#####Seam#####

@Converter

```

@Name("fooConverter")
@BypassInterceptors
@Converter
public class FooConverter implements Converter {

    @In EntityManager entityManager;

    @Transactional
    public Object getAsObject(FacesContext context, UIComponent cmp, String value) {
        EntityManager entityManager = (EntityManager) Component.getInstance("entityManager");
        entityManager.joinTransaction();
        // Do the conversion
    }

    public String getAsString(FacesContext context, UIComponent cmp, Object value) {
        // Do the conversion
    }

}

```

#Seam#####JSF#####JTA####JPA EntityManager####

@Validator

```

@Name("barValidator")
@BypassInterceptors
@Validator
public class BarValidator implements Validator {

    @In FooController fooController;

    public void validate(FacesContext context, UIComponent cmp, Object value)
        throws ValidatorException {
        FooController fooController = (FooController) Component.getInstance("fooController");
        return fooController.validate(value);
    }

}

```

#Seam#####JSF#####Seam#####

---

## Chapter 30. #####

Seam#####EL####JBoss EL#JBoss EL#####EL#####

### 30.1. #####

##EL#####Java#####JSF#####Seam#####EL##### Seam### any  
#####JSF#####

```
<h:commandButton action="#{hotelBooking.bookHotel(hotel)}" value="Book Hotel"/>
```

#### 30.1.1. ##

#####

```
<h:commandButton action="#{hotelBooking.bookHotel(hotel, user)}" value="Book Hotel"/>
```

## hotel # user ##### bookHotel() ##### @In #####

#####

```
<h:commandButton action="#{hotelBooking.bookHotel(hotel.id, user.username)}"
value="Book Hotel"/>
```

### #####hotel.id # user.username#  
#####Seam##action#####  
##### hotel # user #####action##### null ###

#####:

```
<h:commandLink action="#{printer.println('Hello world!')}" value="Hello"/>
```

```
<h:commandLink action="#{printer.println('Hello again')}" value="Hello"/>
```

#####action#####

```
<s:link value="Cancel" action="#{hotelBooking.cancel()}" />
```

#### 30.1.2. ##

#####

##### 30.1.2.1. #JSP 2.1###

#####JSP 2.1#####JSF 1.2#####Facelets#####JSP 2.0#####

##### 30.1.2.2. #Java#### MethodExpression

#####MethodExpression # MethodBinding #####JSF####  
#####JSF#####

- **#Java##### MethodExpression#####**
- **##### methodExpression.getMethodInfo().getParamTypes() ##### #####**  
**MethodExpression##### getParamTypes()#**

**#####Java##### MethodExpression #####**

## 30.2. #####

**##EL#####JavaBean#####** **#{person.name}** **##** **getName()** **#####**  
**##### length() #####**

```
#{person.name.length() }
```

**#####**

```
#{searchResults.size() }
```

**#### form #{obj.property} ##### #{obj.getProperty()}#**

**##### productsByColorMethod ##**

```
#{controller.productsByColor('blue')}
```

## 30.3. ##

**JBoss EL#####Facelets##JSP#####xhtml##JSP####** **#####JBoss**  
**EL###projection#####**

**#####list, set#####**

```
#{company.departments}
```

**#####department#list#####department name#####JBoss EL#####**

```
#{company.departments.{d|d.name}}
```

**#####{ }##### d.name #####department#### d ##department#####**  
**#####String#####**

**#####department#####**

```
#{company.departments.{d|d.size()}}
```

**#####department#####**

```
#{company.departments.{d|d.employees.{emp|emp.lastName}}}
```

**#####department##employee####**

```
#{company.departments.{d|d.employees}}
```

#####department##employee#####

```
# {company.departments.{d|d.employees.{e|e}}}
```

---

## Chapter 31. ##Seam####

```
####Seam##### #####unit test# #####Seam##### #####integration test#  
#####Java#####
```

```
#####
```

### 31.1. Seam#####

```
###Seam#####Java###POJO#####Seam#####Seam#  
#####
```

```
#####Seam###
```

```
@Stateless  
@Scope(EVENT)  
@Name("register")  
public class RegisterAction implements Register  
{  
    private User user;  
    private EntityManager em;  
  
    @In  
    public void setUser(User user) {  
        this.user = user;  
    }  
  
    @PersistenceContext  
    public void setBookingDatabase(EntityManager em) {  
        this.em = em;  
    }  
  
    public String register()  
    {  
        List existing = em.createQuery("select username from User where username=:username")  
            .setParameter("username", user.getUsername())  
            .getResultList();  
        if (existing.size()==0)  
        {  
            em.persist(user);  
            return "success";  
        }  
        else  
        {  
            return null;  
        }  
    }  
}
```

```
#####TestNG#####
```

```
public class RegisterActionTest  
{  
  
    @Test  
    public testRegisterAction()  
    {  
        EntityManager em = getEntityManagerFactory().createEntityManager();  
        em.getTransaction().begin();  
  
        User gavin = new User();  
        gavin.setName("Gavin King");  
    }  
}
```



```

        gavin.setUserName("lovthafew");
        gavin.setPassword("secret");

        RegisterAction action = new RegisterAction();
        action.setUser(gavin);
        action.setBookingDatabase(em);

        assert "success".equals( action.register() );

        em.getTransaction().commit();
        em.close();
    }

    private EntityManagerFactory emf;

    public EntityManagerFactory getEntityManagerFactory()
    {
        return emf;
    }

    @BeforeClass
    public void init()
    {
        emf = Persistence.createEntityManagerFactory("myResourceLocalEntityManager");
    }

    @AfterClass
    public void destroy()
    {
        emf.close();
    }
}

```

Java Persistence API###Java SE#Java EE#### — #####Java EE#####Java SE##### persistence.xml #####

Seam#####

## 31.2. Seam#####

#####  
#####  
#####

##Seam#####Seam#####JBoss#####JDK 1.5#####JDK 1.6#####

```

public class RegisterTest extends SeamTest
{
    @Test
    public void testRegisterComponent() throws Exception
    {
        new ComponentTest() {
            protected void testComponents() throws Exception
            {
                setValue("#{user.username}", "lovthafew");
                setValue("#{user.name}", "Gavin King");
                setValue("#{user.password}", "secret");
                assert invokeMethod("#{register.register}").equals("success");
                assert getValue("#{user.username}").equals("lovthafew");
                assert getValue("#{user.name}").equals("Gavin King");
            }
        }
    }
}

```

```

        assert getValue("#{user.password}").equals("secret");
    }

    }.run();
}

...
}

```

### 31.2.1. #####Mock##

####Seam##### Seam#####facade####

```

@Name("paymentProcessor")
public class PaymentProcessor {
    public boolean processPayment(Payment payment) { .... }
}

```

#####Mock##

```

@Name("paymentProcessor")
@Install(precedence=MOCK)
public class MockPaymentProcessor extends PaymentProcessor {
    public void processPayment(Payment payment) {
        return true;
    }
}

```

##

MOCK

#####Seam#####classpath##Mock#####Mock#####

## 31.3. ####Seam#####

#####assertion##

#####Web#####

#####JSF####SeamTest ##### #####scripted# ###  
 #####Seam#####JSF####

#####

#####JSP#####

```

<html>
<head>
<title>Register New User</title>
</head>
<body>
<f:view>
<h:form>
<table border="0">
<tr>
<td>Username</td>
<td><h:inputText value="#{user.username}"/></td>
</tr>
<tr>
<td>Real Name</td>
<td><h:inputText value="#{user.name}"/></td>

```

```

        </tr>
        <tr>
            <td>Password</td>
            <td><h:inputSecret value="#{user.password}" /></td>
        </tr>
    </table>
    <h:messages/>
    <h:commandButton type="submit" value="Register" action="#{register.register}" />
</h:form>
</f:view>
</body>
</html>

```

#####TestNG#####JSF#####

```

public class RegisterTest extends SeamTest
{
    @Test
    public void testRegister() throws Exception
    {
        new FacesRequest() {
            @Override
            protected void processValidations() throws Exception
            {
                validateValue("#{user.username}", "lovthafew");
                validateValue("#{user.name}", "Gavin King");
                validateValue("#{user.password}", "secret");
                assert !isValidationFailure();
            }

            @Override
            protected void updateModelValues() throws Exception
            {
                setValue("#{user.username}", "lovthafew");
                setValue("#{user.name}", "Gavin King");
                setValue("#{user.password}", "secret");
            }

            @Override
            protected void invokeApplication()
            {
                assert invokeMethod("#{register.register}").equals("success");
            }

            @Override
            protected void renderResponse()
            {
                assert getValue("#{user.username}").equals("lovthafew");
                assert getValue("#{user.name}").equals("Gavin King");
                assert getValue("#{user.password}").equals("secret");
            }
        }.run();
    }
    ...
}

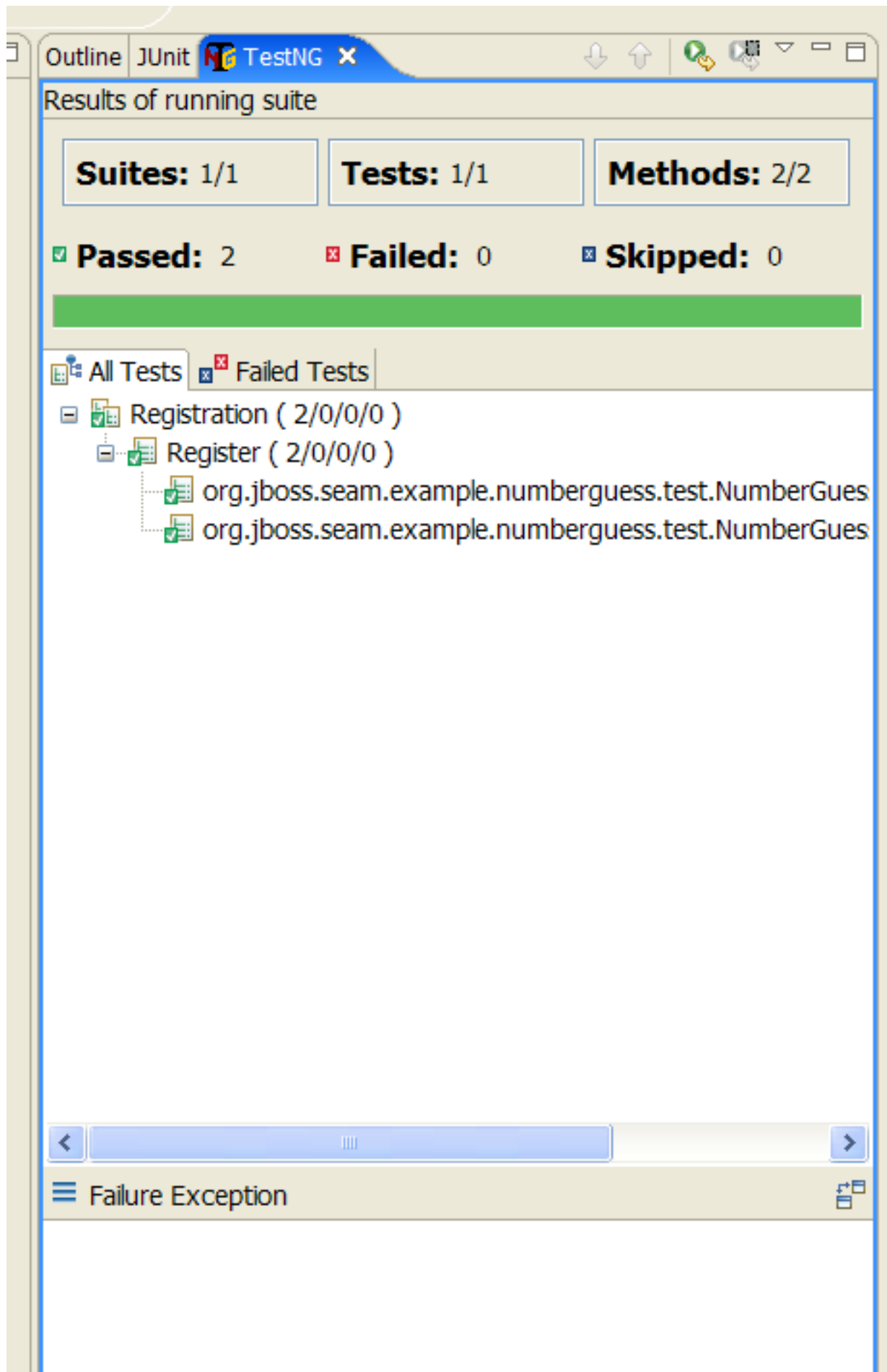
```

```

##### SeamTest#####Seam##### SeamTest.FacesRequest
#####JSF##### SeamTest.NonFacesRequest ###GET#####
####JSF#####JSF#####

```

####Seam#####Ant##Eclipse#TestNG#####



### 31.3.1. ##Mock#####

#####DBUnit##Seam#####DBUnitSeamTest###SeamTest#

#####DBUnit#

```
<dataset>

  <ARTIST
    id="1"
    dtype="Band"
    name="Pink Floyd" />

  <DISC
    id="1"
    name="Dark Side of the Moon"
    artist_id="1" />

</dataset>
```

##### prepareDBUnitOperations() ###Seam:

```
protected void prepareDBUnitOperations() {
    beforeTestOperations.add(
        new DataSetOperation("my/datasets/BaseData.xml")
    );
}
```

##### DataSetOperation ##### DatabaseOperation.CLEAN\_INSERT# ##### @Test  
##### BaseData.xml ##### BaseData.xml #####

##### afterTestOperations #####

##### datasourceJndiName #TestNG#####DBUnit#####

```
<parameter name="datasourceJndiName" value="java:/seamdiscsDatasource"/>
```

### 31.3.2. Seam Mail####

#####

####Seam Mail#####

```
public class MailTest extends SeamTest {

    @Test
    public void testSimpleMessage() throws Exception {

        new FacesRequest() {

            @Override
            protected void updateModelValues() throws Exception {
                setValue("#{person.firstname}", "Pete");
                setValue("#{person.lastname}", "Muir");
                setValue("#{person.address}", "test@example.com");
            }

            @Override
            protected void invokeApplication() throws Exception {
                MimeMessage renderedMessage = getRenderedMailMessage("/simple.xhtml");
            }
        };
    }
}
```

```
        assert renderedMessage.getAllRecipients().length == 1;
        InternetAddress to = (InternetAddress) renderedMessage.getAllRecipients()[0];
        assert to.getAddress().equals("test@example.com");
    }

    }.run();
}
}
```

##### FacesRequest# # invokeApplication #####viewId### getRenderedMailMes-  
sage(viewId); #### #####JSF#####

#####JSF#####

# Chapter 32. Seam##

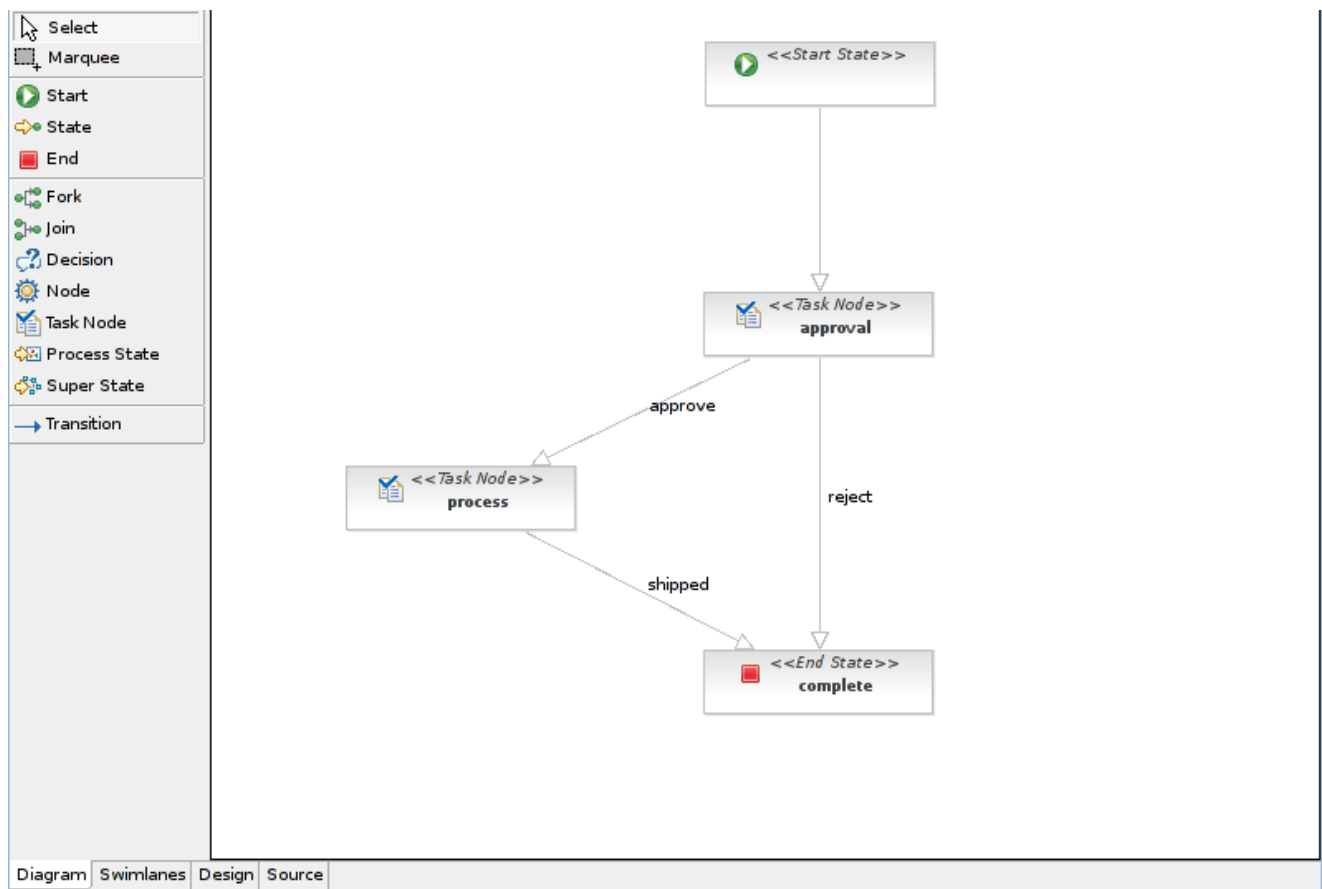
## 32.1. jBPM#####

jBPM#####JBoss  
IDE#####jBPM###http://docs.jboss.com/jbpm/v3/gpd/#####

Eclipse

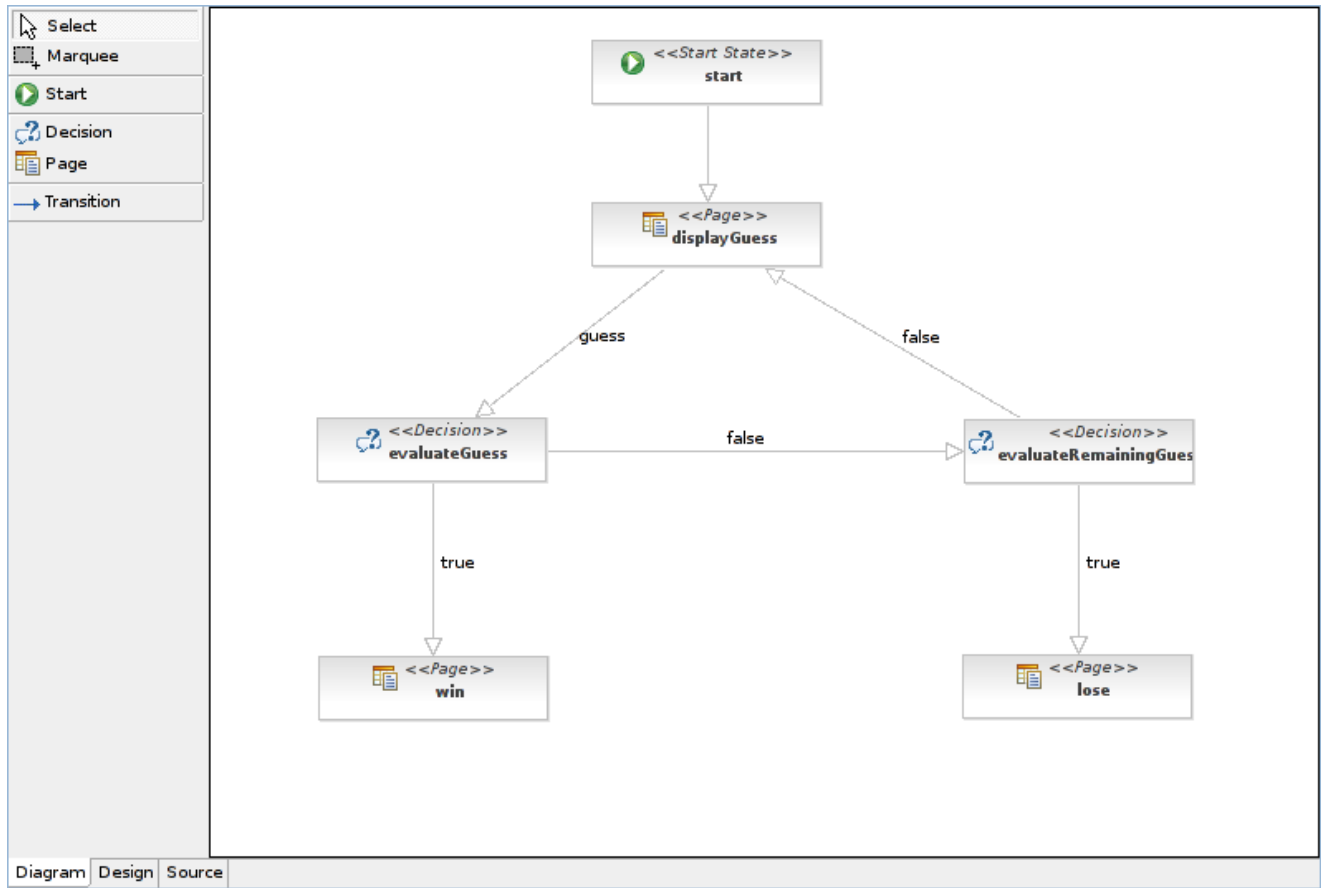
### 32.1.1. #####

#####



### 32.1.2. Pageflow###

#####Pageflow#####





## Chapter 33. ###

### 33.1. #####

#####Seam##### ear #####ear###/lib#### war #####war### /  
WEB-INF/lib #### ##Scope#all#runtime#provided#JBoss AS 4.2##

##### /build/root.pom.xml Maven POM #####

#### 33.1.1. Core

Table 33.1.

##	##	##	##
commons-codec.jar	runtime	ear	Seam Security##Digest#####
jboss-seam.jar	all	ear	Seam#####
jboss-seam-debug.jar	runtime	war	#####Seam#####
jboss-seam-ioc.jar	runtime	war	#Spring#####
jboss-seam-pdf.jar	runtime	war	###Seam#PDF#####
jboss-seam-remoting.jar	runtime	war	###Seam Remoting####
jboss-seam-ui.jar	runtime	war	###Seam JSF#####
jsf-api.jar	provided		JSF API
jsf-impl.jar	provided		JSF####Reference Implementation
jsf-facelets.jar	runtime	war	Facelets
urlrewrite.jar	war	no	URL Rewrite#
jcaptcha-all.jar	ear	no	####Captcha#
quartz.jar	ear	yes	#####Seam#Quartz#####

#### 33.1.2. Ajax4JSF / RichFaces

**Table 33.2. RichFaces###**

##	##	##	##
richfaces-api.jar	all	ear	##RichFaces#####API#### #####
richfaces-impl.jar	runtime	war	##RichFaces####
richfaces-ui.jar	runtime	war	##RichFaces#####UI##

### 33.1.3. Seam Mail

**Table 33.3. Seam Mail###**

##	##	##	##
activation.jar	runtime	ear	#####
mail.jar	runtime	ear	#####
mail-ra.jar	compile only		##### mail-ra.rar#####
jboss-seam-mail.jar	runtime	war	Seam Mail

### 33.1.4. Seam PDF

**Table 33.4. Seam PDF###**

##	##	##	##
itext.jar	runtime	war	PDF##
jfreechart.jar	runtime	war	####
jcommon.jar	runtime	war	JFreeChart#####
jboss-seam-pdf.jar	runtime	war	Seam PDF####

### 33.1.5. JBoss Rules

The JBoss Rules#####Seam#drools/lib#####

**Table 33.5. JBoss Rules###**

##	##	##	##
antlr-runtime.jar	runtime	ear	ANTLR Runtime##
core.jar	runtime	ear	Eclipse JDT
drools-compiler.jar	runtime	ear	
drools-core.jar	runtime	ear	
janino.jar	runtime	ear	
mvel.jar	runtime	ear	

### 33.1.6. JBPM

**Table 33.6. JBPM###**

##	##	##	##
jbpm-jpdl.jar	runtime	ear	

### 33.1.7. GWT

#####Seam#####Google Web Toolkit(GWT)#####

**Table 33.7. GWT###**

##	##	##	##
gwt-servlet.jar	runtime	war	GWT Servlet#

### 33.1.8. Spring

#####Seam#####Spring#####

**Table 33.8. Spring Framework ###**

##	##	##	##
spring.jar	runtime	ear	Spring Framework##

### 33.1.9. Groovy

#####Seam#####Groovy#####

**Table 33.9. Groovy###**

##	##	##	##
groovy-all.jar	runtime	ear	Groovy#

## 33.2. ##Maven####

Maven#####Seam#####Maven Ant Task###Maven#Ant#####Maven#####

#####Maven#####POM#

Seam##### <http://repository.jboss.org/maven2> ##### <http://snapshots.jboss.org/maven2>.

##Seam#####Maven#####

```
<dependency>
  <groupId>org.jboss.seam</groupId>
  <artifactId>jboss-seam</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.jboss.seam</groupId>
  <artifactId>jboss-seam-ui</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.jboss.seam</groupId>
  <artifactId>jboss-seam-pdf</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.jboss.seam</groupId>
  <artifactId>jboss-seam-remoting</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.jboss.seam</groupId>
  <artifactId>jboss-seam-ioc</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.jboss.seam</groupId>
  <artifactId>jboss-seam-ioc</artifactId>
</dependency>
```

#### #####POM#####Seam#JPA##Hibernate#####Hibernate Validator#

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.jboss.seam.example</groupId>
  <artifactId>my-project</artifactId>
  <version>1.0</version>
  <name>My Seam Project</name>
  <packaging>jar</packaging>
  <repositories>
    <repository>
      <id>repository.jboss.org</id>
      <name>JBoss Repository</name>
      <url>http://repository.jboss.org/maven2</url>
    </repository>
  </repositories>

  <dependencies>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-validator</artifactId>
      <version>3.0.0.GA</version>
    </dependency>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-annotations</artifactId>
      <version>3.3.0.ga</version>
    </dependency>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-entitymanager</artifactId>
      <version>3.3.1.ga</version>
    </dependency>

    <dependency>
      <groupId>org.jboss.seam</groupId>
      <artifactId>jboss-seam</artifactId>
      <version>2.0.0.GA</version>
    </dependency>

  </dependencies>
</project>
```

---

# Appendix A. Seam 2.0 #####

## A.1. ##

Seam#####Seam Framework#####Gavin  
#####  
#####  
King#####http://www.redsaga.com##Seam  
Framework#####LGPL#####LGPL#####  
#####

## A.2. ####

### A.2.1. Seam 1.2.1 #####

##2.0##### 1.2.1 #####

### A.2.2. Seam 2.0 Beta 1 #####

2007#09#05#

#####

##Spring#####http://spring.jactiongroup.net/viewtopic.php?t=3521

##Yanger####XXX#####Gavin King#Spring#####

JavaEye#http://yulimin.javaeye.com/blog/120769

Matrix#ht-  
tp://www.matrix.org.cn/thread.shtml?topicId=14ba3958-5bbc-11dc-b06d-09b637715141&forumId=17

CSDN#http://community.csdn.net/Expert/TopicView1.asp?id=5745111

Dev2Dev#http://dev2dev.bea.com.cn/bbs/thread.jspa?forumID=64104&threadID=43900&tstart=0

CJSDN#http://www.cjsdn.net/post/view?bid=20&id=191168&sty=1&tpg=1&age=0

#####

#####wiki####(http://wiki.redsaga.com/confluence/display/SeamRef)#

2007#09#07#

#####Robbin#JavaEye#####  
#####

##### # #####

#####GTalk#####

#####BTW#####11'6#####

####1.2.1#####

#####

**2007#09#09#**

#####Wiki#####

#####Wiki#####

**2007#09#11#**

6. conversations.xml 34K 10P seanchan working

7. jbp.xml 32K 10P ## #-85% 2007-04-15

17. mail.xml 26K 7P Xiaogang Cao Working

27. components.xml 68K 11P jiaochar Working

28. controls.xml 47K 13P blackwing Working

30. testing.xml 10K 6P geshe Working

31. tools.xml 23K 9P geshe Working

#####

## yeshucheng #####

#####

#####

#####

**2007#09#13#**

#####

#####

yeshucheng

#####

###

#####mail.xml#testing.xml#####Review#####CVS####

**2007#09#15#**

#####

#####zaya#####

##### gettingstarted.xml #####

#####

**2007#09#17#**

#####cn-2.0.0B1#####

#####build##### CHM #####build#####

#####

## richard ##### groovy.xml #####

**2007#09#19#**##### groovy.xml #####  
itext.xml #####

#####

**2007#09#21#**

groovy.xml ##### concepts.xml

#####kuuyee#####conversations.xml#configuration.xml

#####  
#####

#####

#### yaza #####

**2007#09#23#**

#####80%#####20%### webservices.xml controls.xml tools.xml

#####itext.xml#####Xu

MingMing#####20%#####

**2007#09#26#**

#####

#####CVS####

#####

20. webservices.xml 9K mazhao### Working 2007-09-07

31. tools.xml 23K 9P junjzheng### 2007-09-12 28. controls.xml 47K 13P huwenjiemaster### 2007-09-13

#####

**2007#09#28#**

#### DigitalSonic ####

**2007#09#29#**

mazhao #####junjzheng #####huwenjiemaster #####

**2007#09#30#**



#####

#####

16. itext.xml 51K 11P topquan###

20. webservices.xml 9K mazhao###

28. controls.xml 47K 13P xukai##

31. tools.xml 23K 9P junjzheng###

#####

#####

**2007#10#07#**

##### Share #####

##### DigitalSonic ##### 3 ##### 3 #####yeshucheng ##### 1 #####1  
#####

**2007#10#09#**

##### yeshucheng ##### 1 #####

**2007#10#11#**

Hi##### ##### continuing ... ##

**2007#10#13#**

##### continuing ... ##

**2007#10#15#**

#####

##### webservices.xml ##### itext.xml  
###20%#####  
#####

**2007#10#19#**

#####

**2007#10#21#**

#####

#####

#####  
#####

#####

#####

**2007#10#24#**

####

1#####

2#####

3#####20%####

4#####

5#####Seam 2.0.0##RC2#####

# 05.10.2007: Seam 2.0.0.CR2 has been released.

# 18.09.2007: Seam 2.0.0.CR1 has been released.

# 27.06.2007: Seam 2.0.0.BETA1 has been released.

###Seam#####CR2#####

**2007#10#25#**

#####

#####

#####

#####99%##

**2007#10#27#**

#####

**2007#10#29#**

xukai#####

**2007#10#29#**

#####

#####

#####

**Table A.1. #####**

####	#####	##		
yeshucheng	4			
CaoXiaogang	3			

####	#####	##		
DigitalSonic	3			
crazycy	2			
Echo	2	#		
seanchan	2			
YuLimin	2			
YY	2	#		
agile_boy	1			
alexchang	1			
caoer	1			
chentianyi	1			
downpour	1			
jiaochar	1			
junjzheng	1			
kuuyee	1			
lyfcdy	1			
magice	1			
mochow	1			
pesome	1			
yeby	1			
##	1			

**2007#10#31#**

#####

#####prajack#####

#####

#####

**2007#11#05#**

#####Seam

2.0.0

GA#####xml#####

**2007#11#07#**

#####

**2007#11#09#**

#####

**2007#11#11#**

####2007.11.11 11:11:11#####1111111111#####

**2007#11#13#**

#####

#####

**2007#11#14#**

#####

#####Leader#####

### **A.2.3. Seam 2.0 #####**

**2007#11#15#**

#####

####http://wiki.redsaga.com/confluence/pages/viewpage.action?pageId=1751

#####CVS##### cn2.0.0GA #####

**2007#11#17#**

#####

**2007#12#22#**

#####

**2007#12#24#**

#####MM#####"#####"#####

**2007#12#30#**

#####RC#####

**2008#03#27#**

#####Seam 2.0#####

#####RC#####

<http://yulimin.javaeye.com/blog/151917>

#####Seam 2.0#####

#####RC#####

#####

##### T-Shirt #####

<http://wiki.redsaga.com/confluence/pages/viewpage.action?pageId=2791>

#####

#####

<http://wiki.redsaga.com/confluence/display/SeamRef/Home>