esJUG

Grupo de Usuários de Java do Estado do Espírito Santo

Introduction to JBoss Seam

License for use and distribution

This material is available for non-commercial use and can be derived and/or redistributed, as long as it uses an equivalent license.

Attribution-Noncommercial-Share Alike 3.0 Unported

http://creativecommons.org/licenses/by-nc-sa/3.0/

You are free to share and to adapt this work under the following conditions: (a) You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work); (b) You may not use this work for commercial purposes. (c) If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

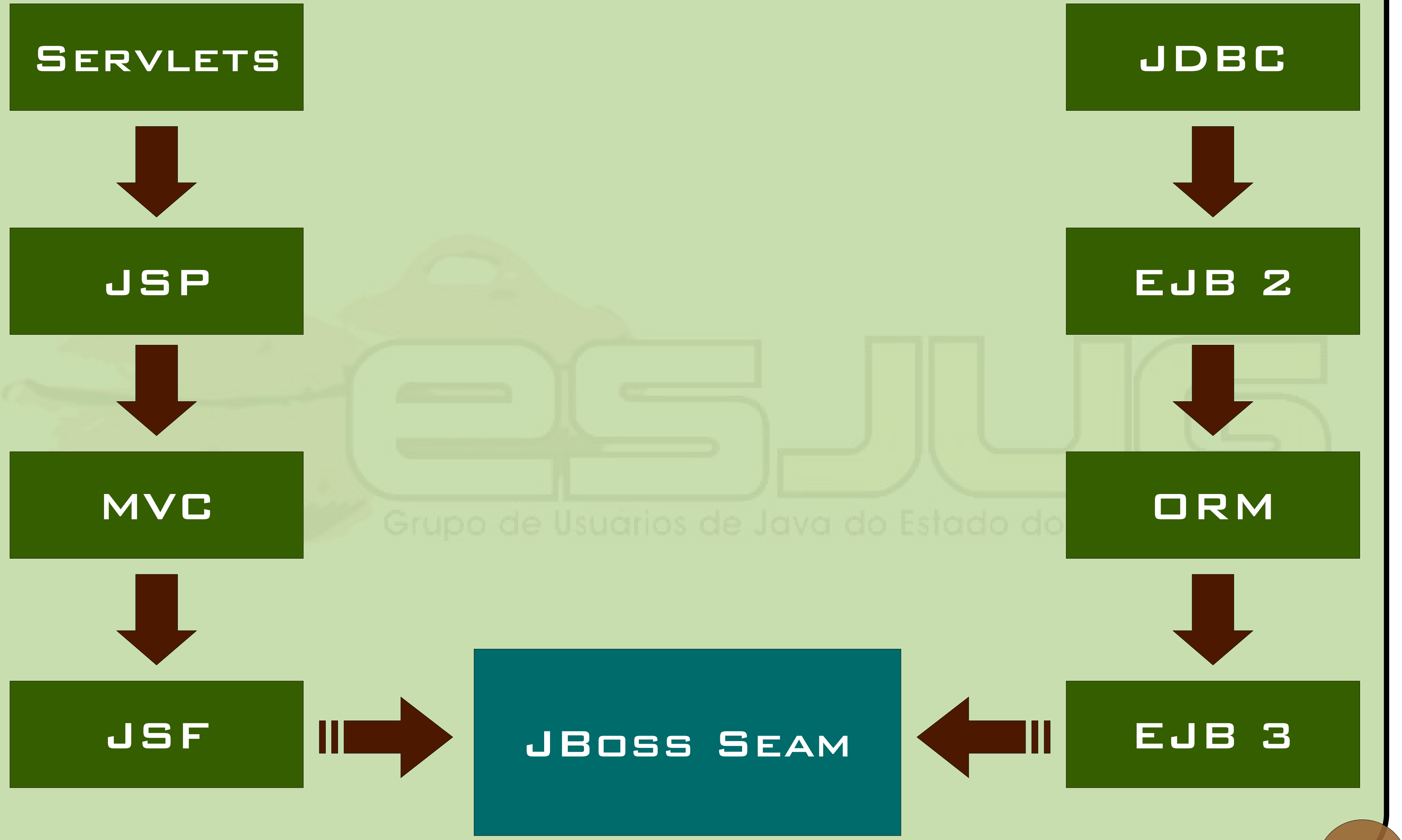
About the author – Vítor Souza

- Education:
  - Computer Science graduate, masters in Software Engineering – Federal University of Espírito Santo (Brazil).
- Java:
  - Developer since 1999;
  - Focus on Web Development;
  - Co-founder and coordinator of ESJUG (Brazil).
- Professional:
  - Substitute teacher at Federal University of ES;
  - Engenho de Software Consulting & Development.
- Contact: vitorsouza@gmail.com

Summary

- Introduction;
- JBoss Seam overview;
- A simple application with JBoss Seam;
- Features to explore...

Technology timeline



Servlets

JSP

MVC

JSF

JDBC

EJB 2

ORM

EJB 3

JBoss Seam

JavaServer Faces

- Similarities with the MVC frameworks;
- Focus on the developer;
- Designed to be used by IDEs;
- Extensible UI components;
- Event handling;
- Simple navigation support.

EJB 3.0

- Part of the Java Enterprise architecture:
  - Security, life cycle management, remote accessibility, persistence, transaction handling, concurrency, clustering, etc.
- Simplified model;
- Less XML and more annotations;
- Entity Beans can be replaced by ORM;
- JNDI can be replaced by DI.

JBoss Seam

- Ideas taken from MVC frameworks, ORM and DI were included in the Java EE 5 “package”;
- Why another framework?
- The piece that is missing in the “package”:
  - JSF and EJB3: different philosophies;
  - Replaces backing beans and the integration plumbing;
  - Use of annotations and contextual components;
  - Many extra features;
  - Productivity with scalability.

JBoss Seam

- General information:
  - Created by Gavin King (creator of Hibernate);
  - Version 1.0 released in 11/06/2006;
  - Current version: 2.0;
  - Websites:

http://www.jboss.com/products/seam

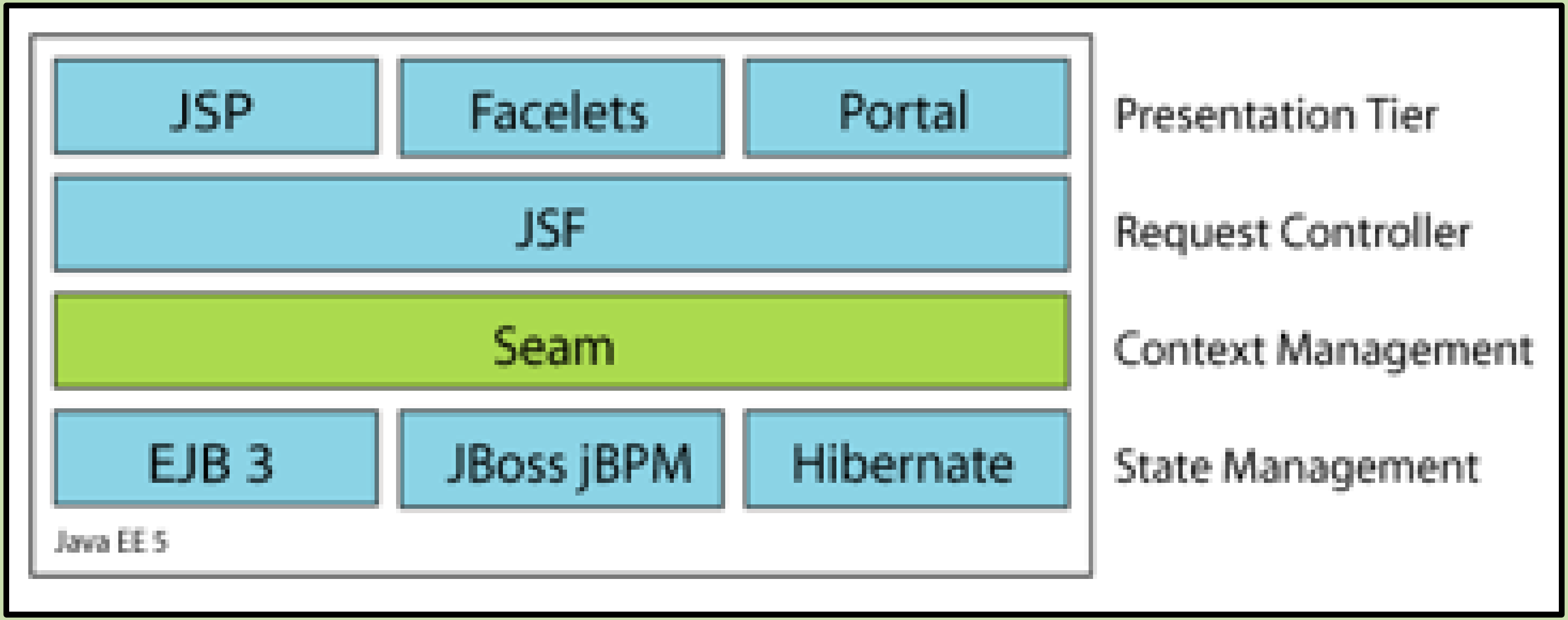
http://labs.jboss.com/jbossseam/gettingstarted

http://labs.jboss.com/jbossseam/download/index.html

Overview

- JSF – EJB3 integration;
- Integrated AJAX and jBPM;
- Uniform component model;
- Declarative state management;
- Bijection;
- Workspace management;
- Use of annotated POJOs;
- Testability.

Seaming things...



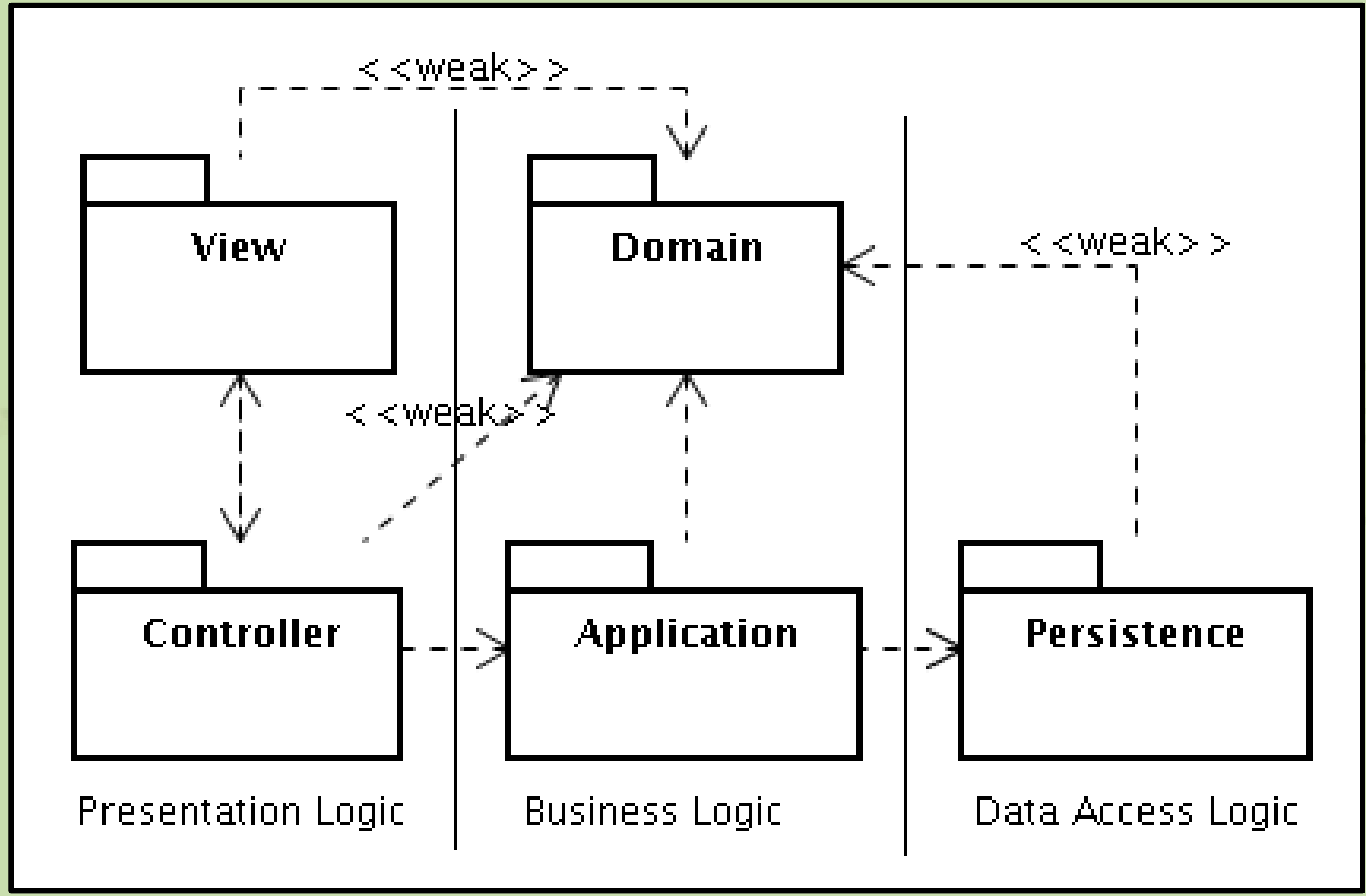
Source: Seam Reference

A simple example

Demonstration

- Tools:
  - JBoss 4.2.1 GA (EJB 3 support);
  - JBoss Seam 2.0.0 GA;
  - Eclipse Europa (3.3.1.1) with Java EE support;
  - ICEFaces 1.6.0.
- Goal:
  - Build a contact list with two simple functionalities:
    - Add contact;
    - View contact.

System architecture



A Seam project on Eclipse

- Create an Enterprise Application Project:
  - EJB module: persistence, domain and application;
  - Web module: controller and view;
  - Client and Connector modules: not used.
- Very specific configuration:
  - It's easier to start from one of Seam's examples;
  - You can also use seam-gen.

The enterprise application project

- EarContent/
  - META-INF/
    - application.xml
    - contactlist-ds.xml
    - jboss-app.xml
  - lib/
    - hsqldb-1.8.jar
    - jboss-el-2.0.jar
  - seam-2.0.jar
- hsqldb/

You will need to adapt this if using another application server

application.xml

```
<application ...>
  <display-name>ContactList</display-name>
  <module>
    <ejb>ContactListEJB.jar</ejb>
  </module>
  <module>
    <web>
      <web-uri>ContactListWeb.war</web-uri>
      <context-root>ContactListWeb</context-root>
    </web>
  </module>
  <module>
    <ejb>seam-2.0.jar</ejb>
  </module>
</application>
```

The JARs under lib/ will be automatically added by JBoss!

jboss-app.xml

```
<jboss-app>
   <loader-repository>
      seam.jboss.org:loader=ContactList
   </loader-repository>
   <module>
     <service>META-INF/contactlist-ds.xml</service>
   </module>
</jboss-app>
```

contactlist-ds.xml

```
<datasources>
   <local-tx-datasource>
      <jndi-name>ContactListDataSource</jndi-name>
      <connection-url>
         jdbc:hsqldb:hsql://localhost/contactlist
      </connection-url>
      <driver-class>org.hsqldb.jdbcDriver</driver-class>
      <user-name>sa</user-name>
      <password></password>
      <min-pool-size>5</min-pool-size>
      <max-pool-size>20</max-pool-size>
      <idle-timeout-minutes>0</idle-timeout-minutes>
      <metadata>
         <type-mapping>Hypersonic SQL</type-mapping>
      </metadata>
   </local-tx-datasource>
</datasources>
```

The EJB project

- ejbModule/
  - net.java.dev.esjug.util.* (utilities)
  - META-INF/
    - ejb-jar.xml
    - persistence.xml
  - seam.properties (empty)

ejb-jar.xml

```
<ejb-jar ...>
   <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>
```

persistence.xml

```
<persistence ...>
   <persistence-unit name="ContactListPersistUnit">
     <provider>
        org.hibernate.ejb.HibernatePersistence
     </provider>
     <jta-data-source>
        java:/ContactListDataSource
     </jta-data-source>
     <properties>
        <property name="hibernate.dialect"
        value="org.hibernate.dialect.HSQLDialect"/>
        <property name="hibernate.hbm2ddl.auto"
        value="update"/>
        <property name="hibernate.show_sql"
        value="true"/>
        <property name="hibernate.format_sql"
        value="true"/>
     </properties>
</persistence-unit></persistence>
```

Domain object support

```
@MappedSuperclass
public abstract class DomainObjectSupport
                                implements DomainObject {
   @Column(nullable = false, length = 40)
   private String uuid;   // + getter/setter

   public DomainObjectSupport() {
      uuid = UUID.randomUUID().toString();
   }
   public boolean equals(Object o) {
      if (! getClass().equals(o.getClass())) return false;
      DomainObjectSupport obj = (DomainObjectSupport)o;
      return uuid.equals(obj.uuid);
   }

   public int hashCode() {
      return uuid.hashCode();
   }
}
```

Persistent object support

```
@MappedSuperclass
public abstract class PersistentObjectSupport<I
extends Serializable, V extends Serializable> extends
DomainObjectSupport implements PersistentObject<I, V>
{
   @Id @GeneratedValue(strategy = GenerationType.AUTO)
   private I id;

   @Version @Column(nullable = false)
   private V version;

   public boolean isPersistent() {
      return (id != null);
   }
}
```

Seam DAO support

```
public abstract class SeamBaseDAO<T extends
PersistentObject<? extends Serializable, ? extends
Serializable>> implements BaseDAO<T> {
   protected abstract EntityManager
                                getEntityManager();

   protected abstract Class<T> getDomainClass();

   @Transactional
   public Collection<T> retrieveAll() {
      Query query = getEntityManager().createQuery(
             "from " + getDomainClass().getName());
      return query.getResultList();
   }

   // retrieveById, save and delete...
}
```

Web Utilities

```
@Name("webUtils")
@Scope(ScopeType.APPLICATION)
public class WebUtils {
   private static final String DEFAULT_DECORATOR =
               "/WEB-INF/decorators/default.xhtml";

   public String getDecoratorPath() {
      return DEFAULT_DECORATOR;
   }
}
```

The Web project

- src/
  - faces.properties
  - messages.properties
- WebContent/
  - resources/
  - WEB-INF/
    - decorators/default.xhtml
    - *Config files*
  - home.xhtml
  - index.jsp

components.xml
faces-config.xml
pages.xml
web.xml

messages.properties

```
link.addNewContact = Add new contact
link.viewContacts = View contact list

text.welcome = Welcome to the Contact List Seam demo.
text.viewContacts = View contacts
text.addContact = Add new contact

form.name = Name
form.birthdate = Birthdate
form.contactType = Contact type
form.contactInformation = Value


...
```

components.xml

```
<components ...>
   <core:init jndi-pattern=
      "ContactList/#{ejbName}/local" debug="true" />

   <core:manager conversation-timeout="120000"
                 concurrent-request-timeout="500"
                 conversation-id-parameter="cid" />

   <core:resource-loader>
      <core:bundle-names>
         <value>messages</value>
         <value>faces</value>
      </core:bundle-names>
   </core:resource-loader>

   <transaction:ejb-transaction />
</components>
```

faces-config.xml

```
<faces-config  ...>
   <application>
      <message-bundle>messages</message-bundle>
      <view-handler>
      com.icesoft.faces.facelets.D2DSeamFaceletViewHandler
      </view-handler>
      <locale-config>
         <default-locale>en</default-locale>
         <supported-locale>en</supported-locale>
      </locale-config>
   </application>
   <managed-bean>
      <managed-bean-name>highlight</managed-bean-name>
      <managed-bean-class>
         com.icesoft.faces.context.effects.Highlight
      </managed-bean-class>
      <managed-bean-scope>application</managed-bean-scope>
   </managed-bean>
</faces-config>
```

pages.xml

```
<pages ...>
</pages>
```

web.xml

```
<!-- JBoss Seam -->
<listener>
   <listener-class>
      org.jboss.seam.servlet.SeamListener
   </listener-class>
</listener>
<servlet>
   <servlet-name>Seam Servlet</servlet-name>
   <servlet-class>
      org.jboss.seam.servlet.SeamResourceServlet
   </servlet-class>
</servlet>
<servlet-mapping>
   <servlet-name>Seam Servlet</servlet-name>
   <url-pattern>/seam/resource/*</url-pattern>
</servlet-mapping>
```

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>

<!-- JSF -->
<context-param>
  <param-name>
    javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>server</param-value>
</context-param>
```

web.xml

```
<context-param>
   <param-name>
      javax.faces.DEFAULT_SUFFIX</param-name>
   <param-value>.xhtml</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>

<!-- Facelets development mode -->
<context-param>
   <param-name>facelets.DEVELOPMENT</param-name>
   <param-value>true</param-value>
</context-param>
```

web.xml

```
<!-- ICEFaces -->
<listener>
  <listener-class>com.icesoft.faces.util.event
  .servlet.ContextEventRepeater</listener-class>
</listener>
<context-param>
  <param-name>
    com.icesoft.faces.actionURLSuffix
  </param-name>
  <param-value>.seam</param-value>
</context-param>
<context-param>
  <param-name>
    com.icesoft.faces.synchronousUpdate
  </param-name>
  <param-value>false</param-value>
</context-param>
```

web.xml

```
<context-param>
   <param-name>
      com.icesoft.faces.doJSFStateManagement
   </param-name>
   <param-value>true</param-value>
</context-param>
<context-param>
   <param-name>
      com.icesoft.faces.standardRequestScope
   </param-name>
   <param-value>true</param-value>
</context-param>
```

web.xml

```
<servlet>
   <servlet-name>Persistent Servlet</servlet-name>
   <servlet-class>
      com.icesoft.faces.webapp
      .xmlhttp.PersistentFacesServlet
   </servlet-class>
   <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
   <servlet-name>Persistent Servlet</servlet-name>
   <url-pattern>*.seam</url-pattern>
</servlet-mapping>
<servlet-mapping>
   <servlet-name>Persistent Servlet</servlet-name>
   <url-pattern>/xmlhttp/*</url-pattern>
</servlet-mapping>
```

web.xml

```
   <servlet>
      <servlet-name>Blocking Servlet</servlet-name>
      <servlet-class>
      com.icesoft.faces.webapp.xmlhttp.BlockingServlet
      </servlet-class>
      <load-on-startup>1</load-on-startup>
   </servlet>
   <servlet-mapping>
      <servlet-name>Blocking Servlet</servlet-name>
      <url-pattern>/block/*</url-pattern>
   </servlet-mapping>

   <!-- Session timeout -->
   <session-config>
      <session-timeout>10</session-timeout>
   </session-config>
```

web.xml

```
<!-- Default files -->
<welcome-file-list>
   <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

Seam components to develop

- Entities (domain classes);
- DAOs (persistence classes);
- Services (application classes);
- Action (controller classes);
- Web pages (view).

Creating an entity

```
@Entity
public class Contact extends
PersistentObjectSupport<Long, Long> implements
Comparable<Contact> {

   @Column(length = 50)
   private String name;

   @Temporal(TemporalType.DATE)
   private Date birthDate;

   @OneToMany(cascade = CascadeType.ALL)
   @Sort(type = SortType.NATURAL)
   private SortedSet<ContactInformation> contactInfo;

   ...
}
```

Creating a DAO

```
@AutoCreate
@Name("contactDAO")
@Scope(ScopeType.APPLICATION)
@Stateless
public class SeamContactDAO extends
SeamBaseDAO<Contact> implements ContactDAO {

   @PersistenceContext
   private EntityManager entityManager;

   ...
}
```

Creating a service

```
@Local
public interface ManageAddressBookService { ... }

@AutoCreate
@Name("manageAddressBookService")
@Scope(ScopeType.APPLICATION)
@Stateless
public class ManageAddressBookServiceBean implements
ManageAddressBookService {

   @In
   private ContactDAO contactDAO;


   ...
}
```

Creating an action

```
@Name("manageAddressBookAction")
@Scope(ScopeType.SESSION)
public class ManageAddressBookAction {

   @In
   private ManageAddressBookService
manageAddressBookService;

   private SortedSet<Contact> contactList;

   private Contact contact = new Contact();

   ...
```

Creating an action

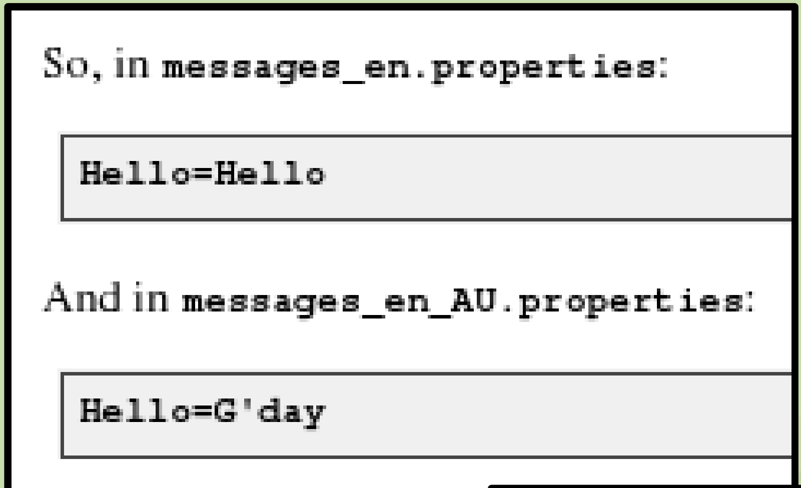
```
   public void addContact() {
      log.info("Adding contact: #0, #1, #2, #3, #4",
contact.getId(), contact.getName(),
contact.getBirthDate(), information.getType(),
information.getValue());

      Contact c =
manageAddressBookService.addContact(contact,
information);
      getContactList().add(c);

FacesMessages.instance().add("#{messages['text.succes
sFullyAddedContact']}", contact.getName());

      contact = new Contact();
      information = new ContactInformation();
   }
}
```

Referring to the action on web pages

```
<ui:composition template="#{webUtils.decoratorPath}">
   <ui:define name="title"><ice:outputText value=
   "#{messages['text.addContact']}" />   </ui:define>

   ...

   <ice:panelGroup><ice:outputText value="#{messages
   ['form.birthdate']}:"/></ice:panelGroup>

   <ice:panelGroup><ice:inputText value=
   "#{manageAddressBookAction.contact.birthDate}" />
   </ice:panelGroup>

   ...
```

Features to experiment

Internationalization

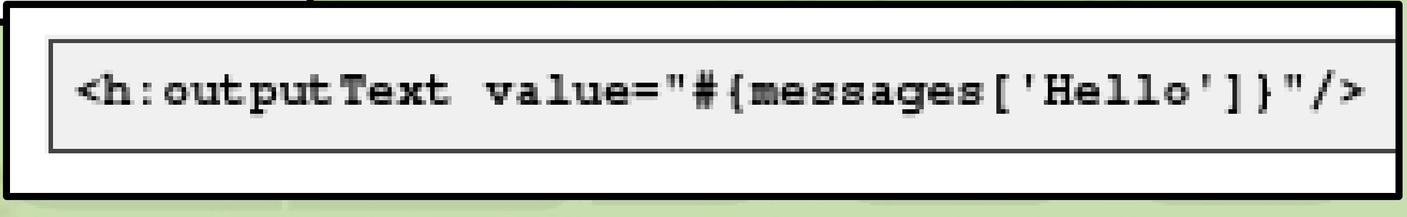


So, in `messages_en.properties`:

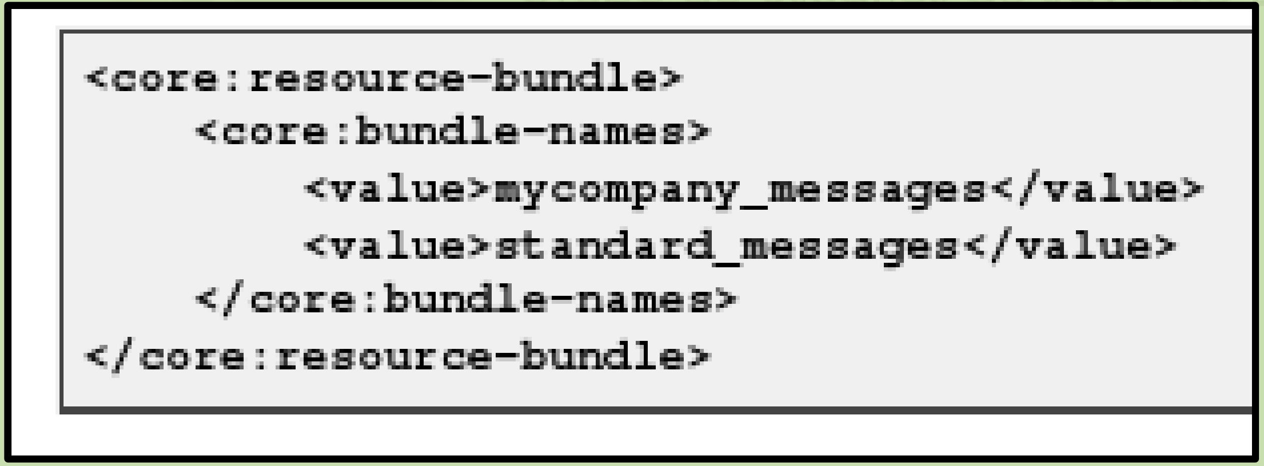
```
Hello=Hello
```

And in `messages_en_AU.properties`:

```
Hello=G'day
```

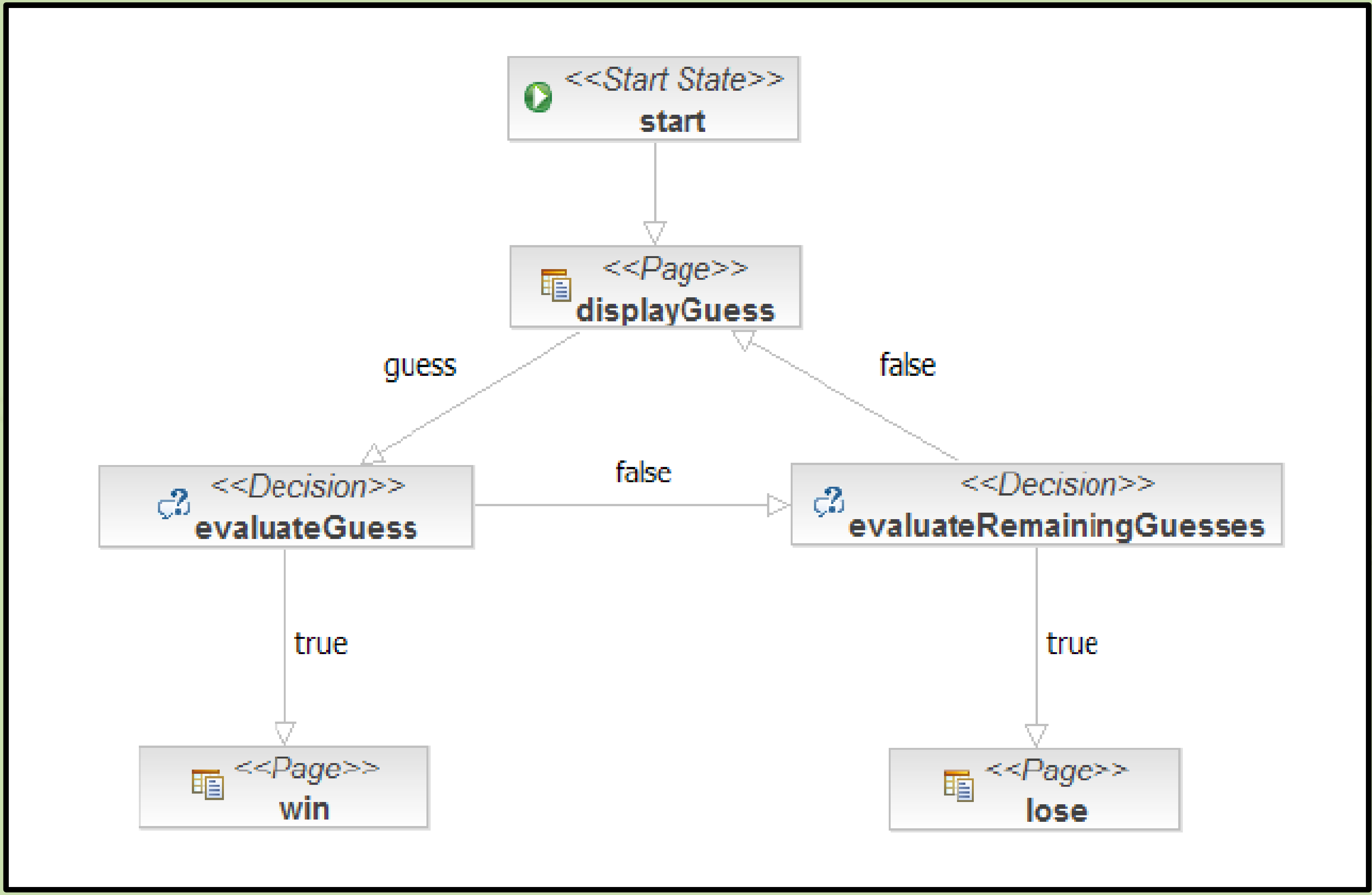


```
<h:outputText value="#{messages['Hello']}"/>
```

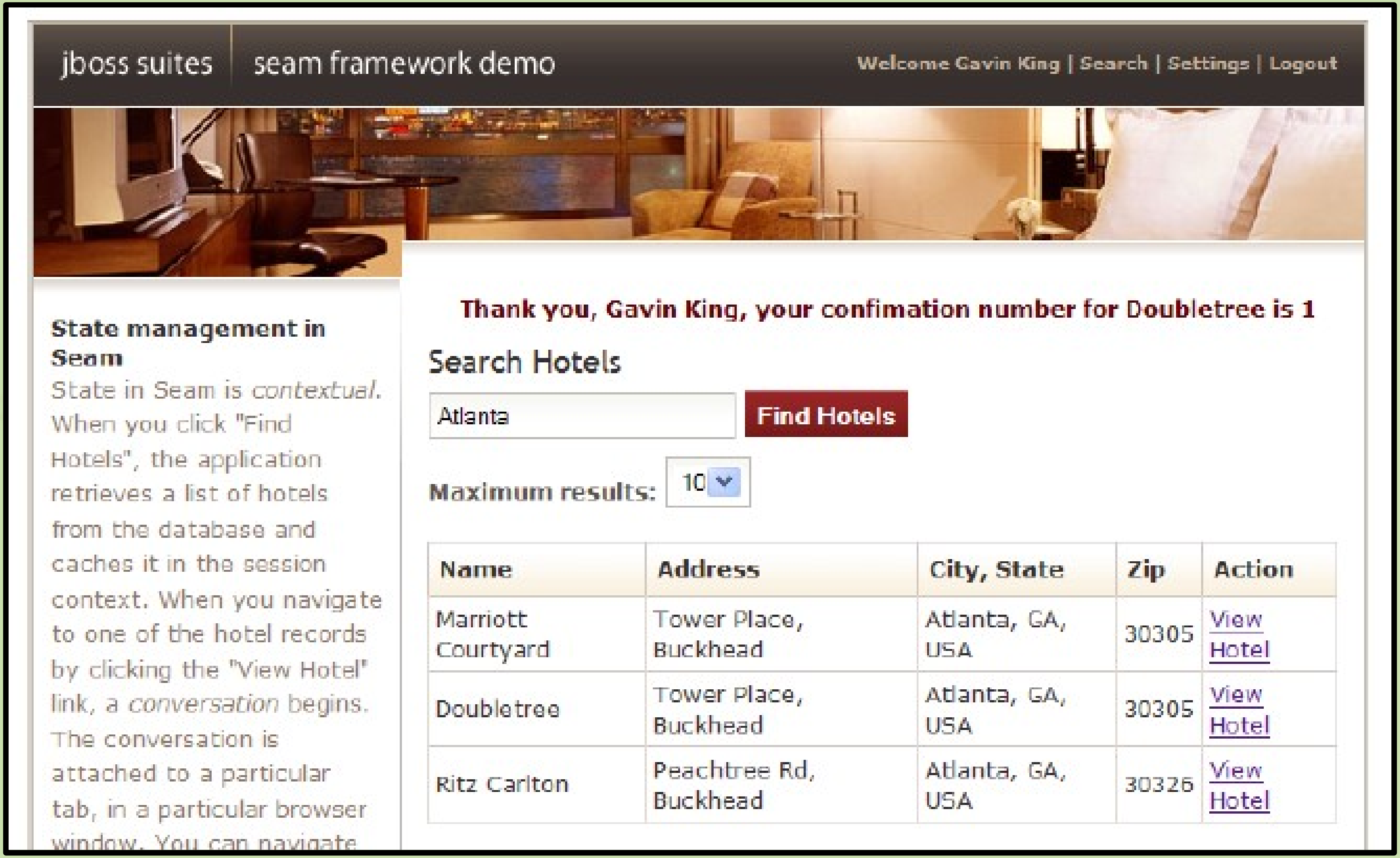


```
<core:resource-bundle>
    <core:bundle-names>
        <value>mycompany_messages</value>
        <value>standard_messages</value>
    </core:bundle-names>
</core:resource-bundle>
```

Pageflow


<<Start State>>
start
<<Page>>
displayGuess
guess
false
<<Decision>>
evaluateGuess
false
<<Decision>>
evaluateRemainingGuesses
true
true
<<Page>>
win
<<Page>>
lose

Conversation and workspace



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

Thank you, Gavin King, your confimation number for Doubletree is 1

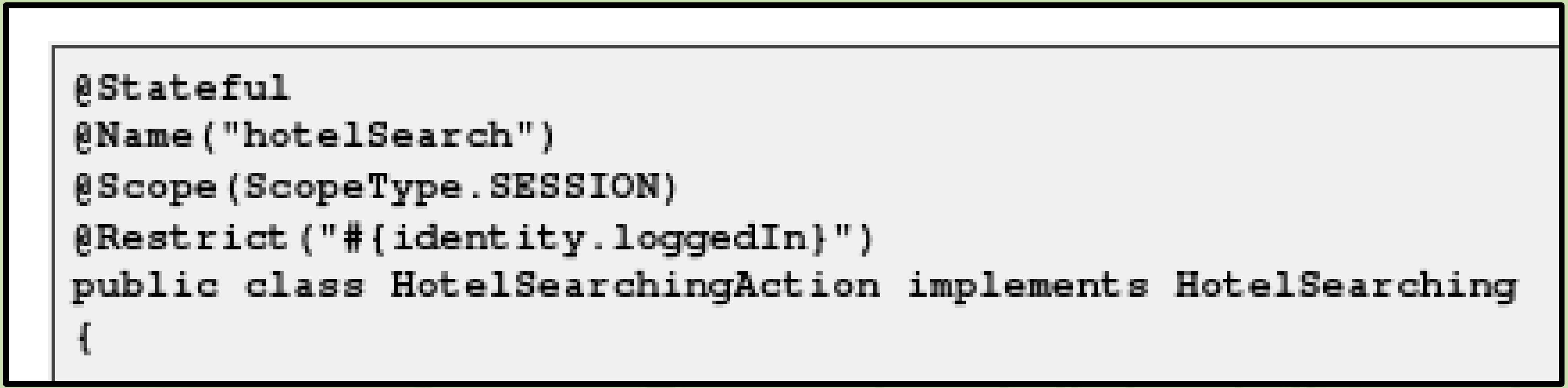
Search Hotels

Atlanta

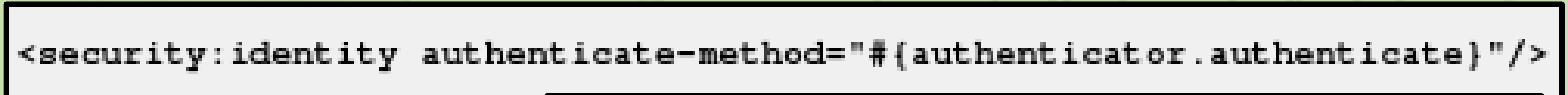
Maximum results: 10

Name	Address	City, State	Zip	Action
Marriott Courtyard	Tower Place, Buckhead	Atlanta, GA, USA	30305	View Hotel
Doubletree	Tower Place, Buckhead	Atlanta, GA, USA	30305	View Hotel
Ritz Carlton	Peachtree Rd, Buckhead	Atlanta, GA, USA	30326	View Hotel

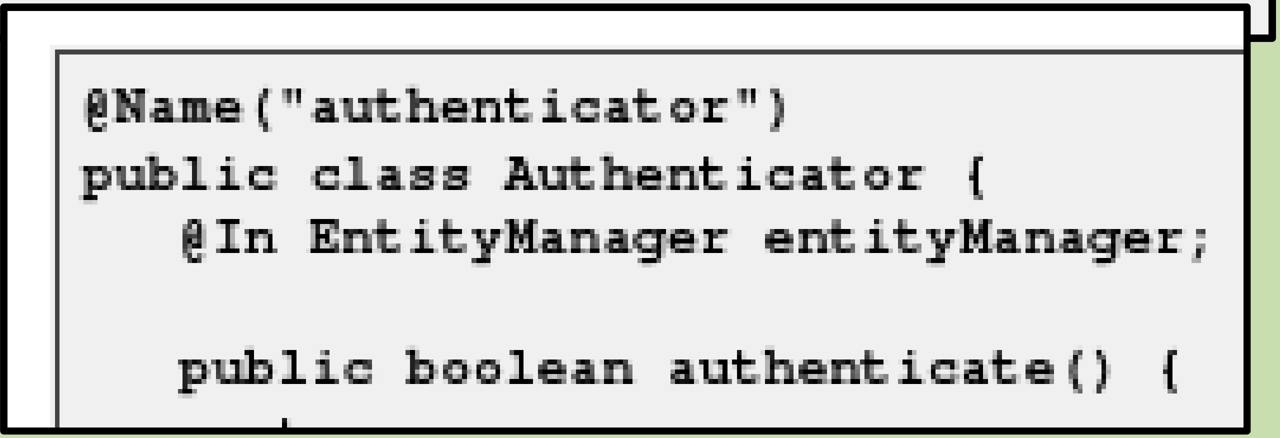
Authentication



```
@Stateful
@Name("hotelSearch")
@Scope(ScopeType.SESSION)
@Restrict("#{identity.loggedIn}")
public class HotelSearchingAction implements HotelSearching
{
```

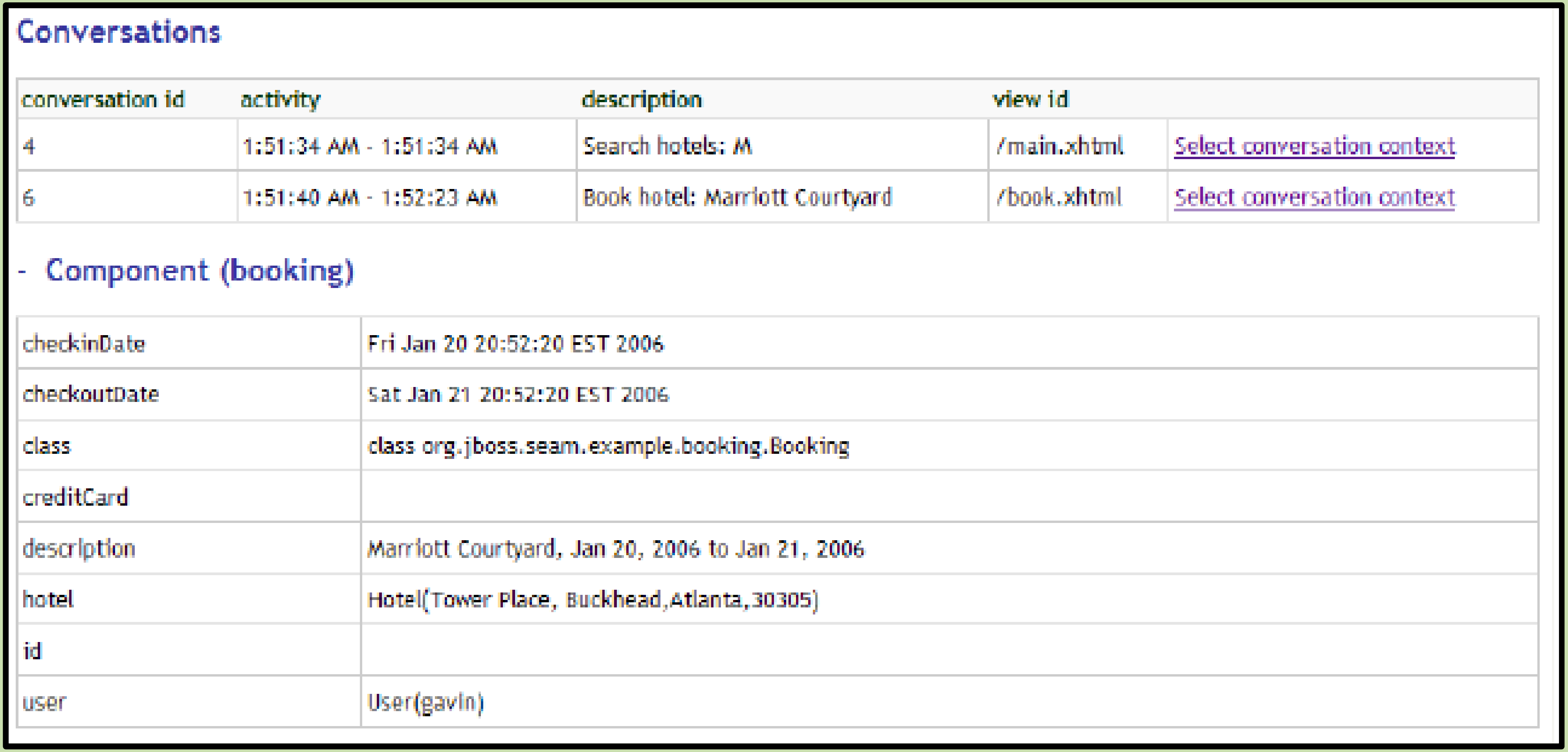


```
<security:identity authenticate-method="#{authenticator.authenticate}"/>
```



```
@Name("authenticator")
public class Authenticator {
   @In EntityManager entityManager;

   public boolean authenticate() {
```

Debugging



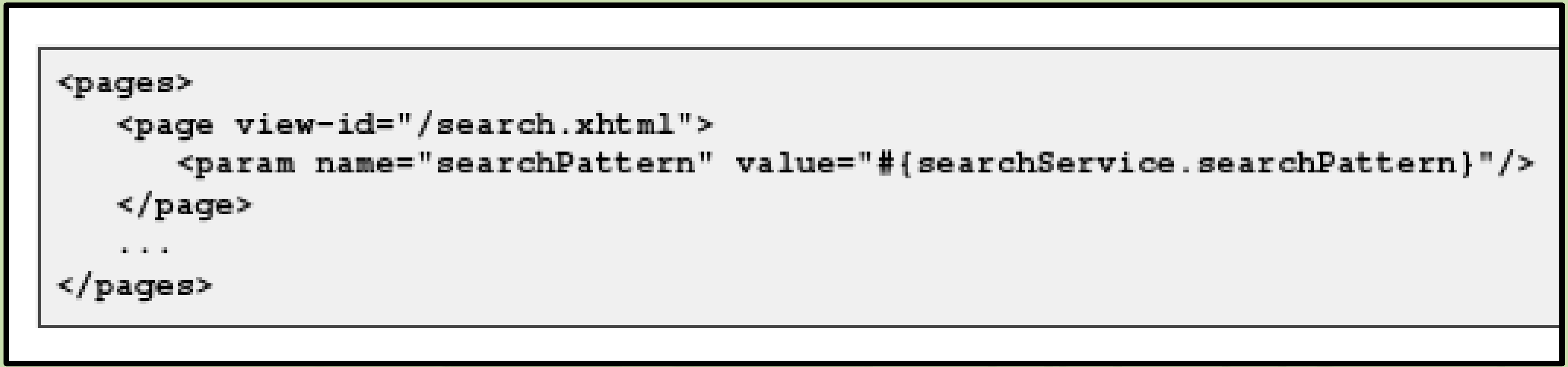
Conversations

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

- 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)

RESTful URLs

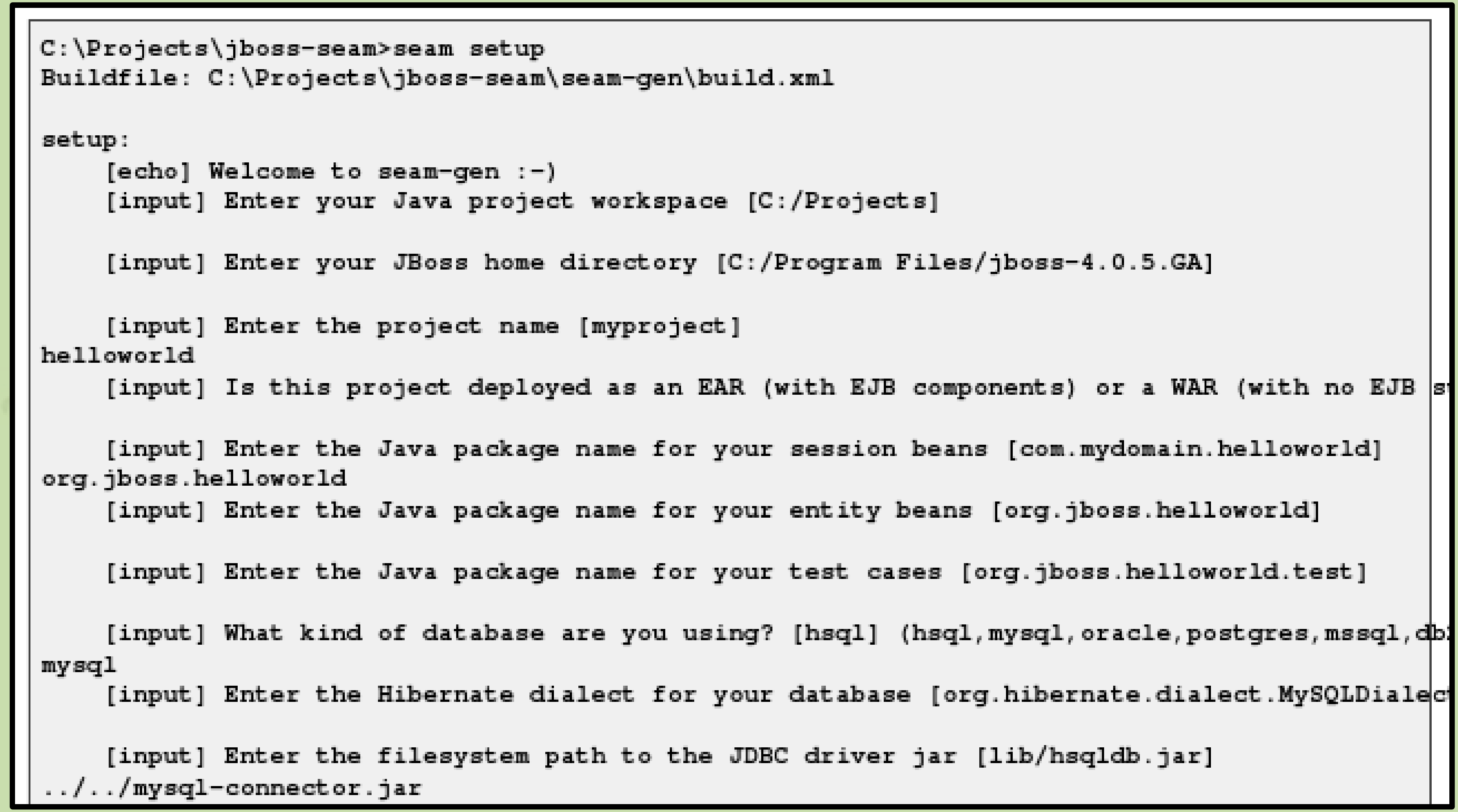


```
<pages>
   <page view-id="/search.xhtml">
      <param name="searchPattern" value="#{searchService.searchPattern}"/>
   </page>
   ...
</pages>
```

JBoss Seam Blog

An example of a RESTful Seam application

seam-gen

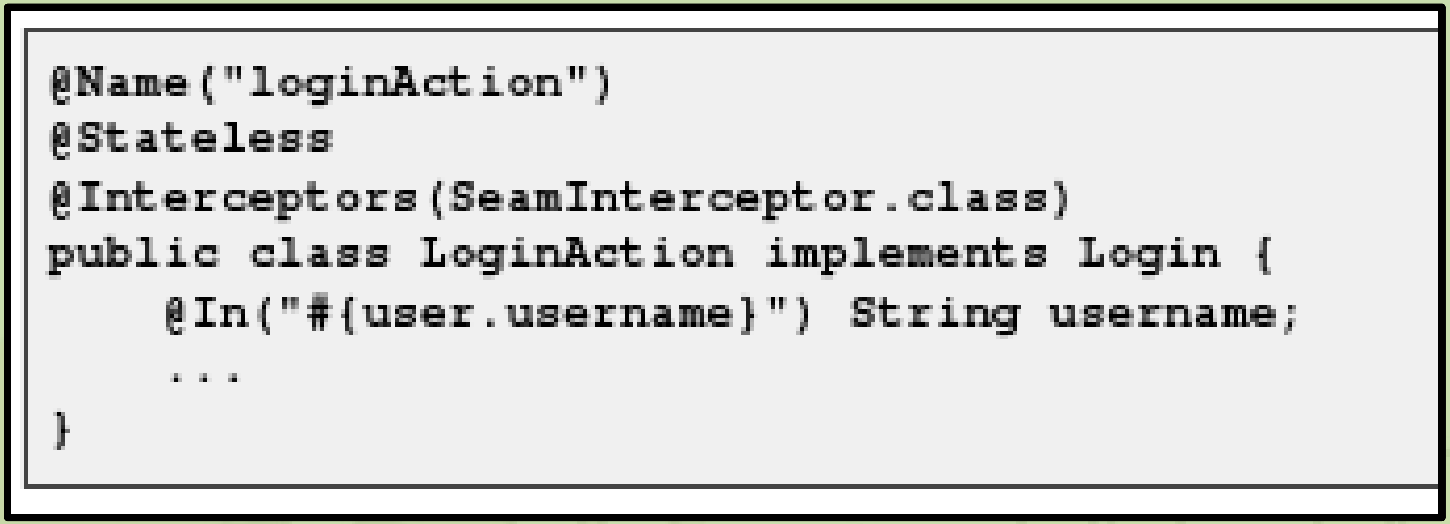


```
C:\Projects\jboss-seam>seam setup
Buildfile: C:\Projects\jboss-seam\seam-gen\build.xml

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

    [input] Enter your JBoss home directory [C:/Program Files/jboss-4.0.5.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 s

    [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,db
mysql
    [input] Enter the Hibernate dialect for your database [org.hibernate.dialect.MySQLDialec

    [input] Enter the filesystem path to the JDBC driver jar [lib/hsqldb.jar]
../../mysql-connector.jar
```

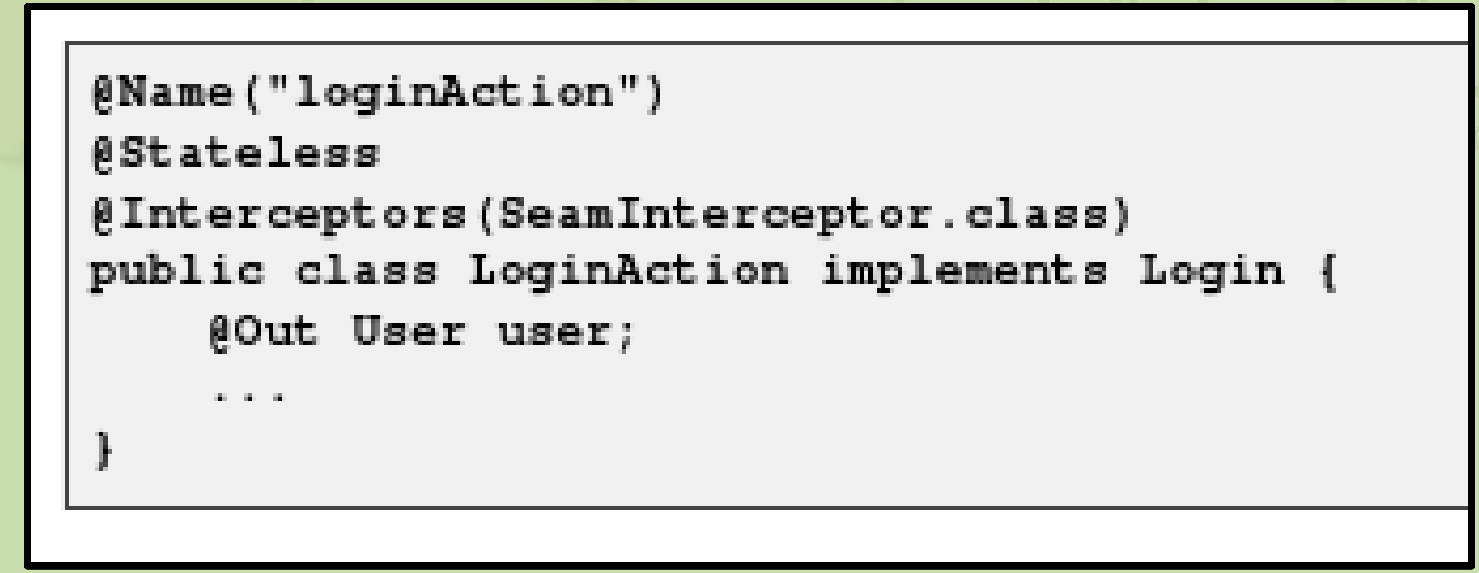
Contexts

- Stateless;
- Event (or request);
- Page;
- Conversation;
- Session;
- Business Process;
- Application.

Bijection

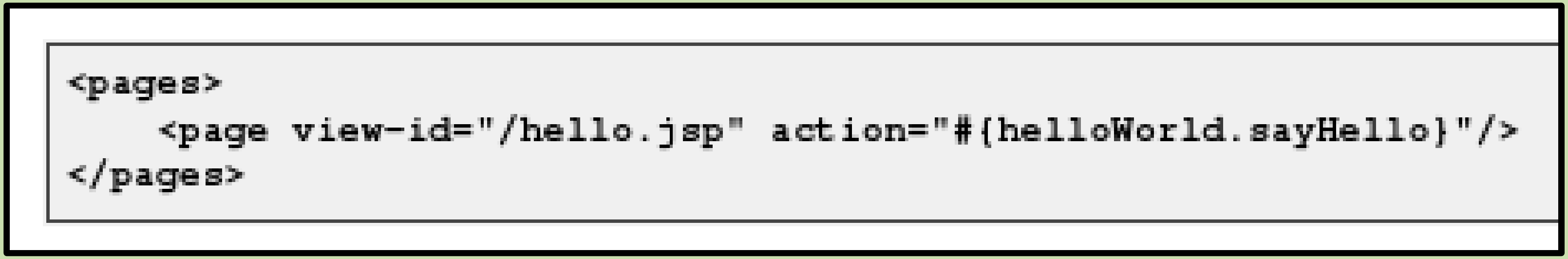


```
@Name("loginAction")
@Stateless
@Interceptors(SeamInterceptor.class)
public class LoginAction implements Login {
    @In("#{user.username}") String username;
    ...
}
```

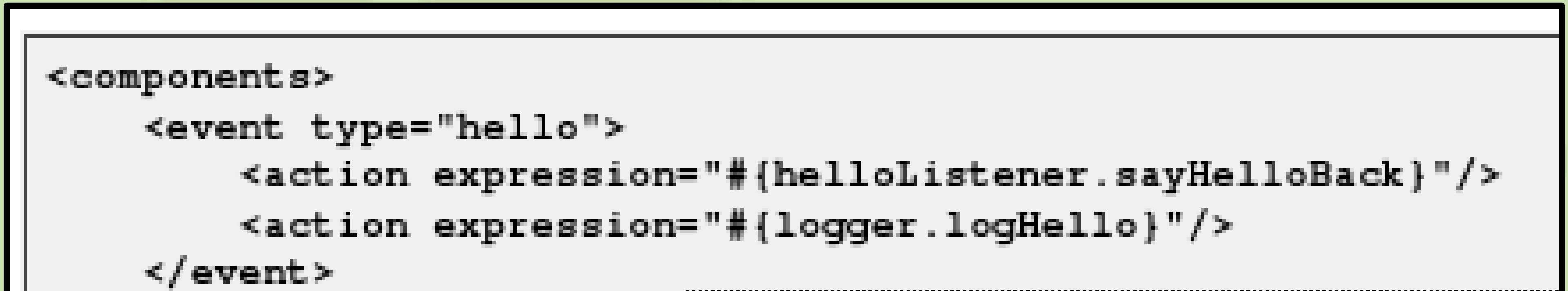


```
@Name("loginAction")
@Stateless
@Interceptors(SeamInterceptor.class)
public class LoginAction implements Login {
    @Out User user;
    ...
}
```

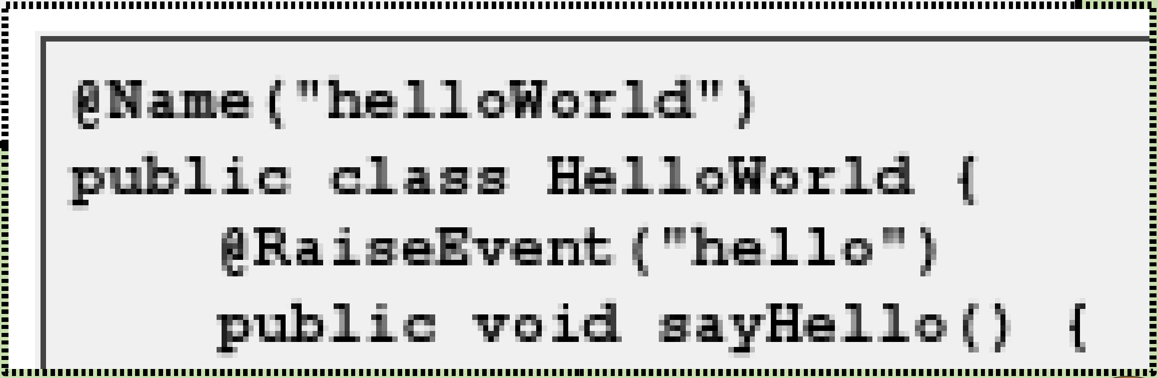
Events



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

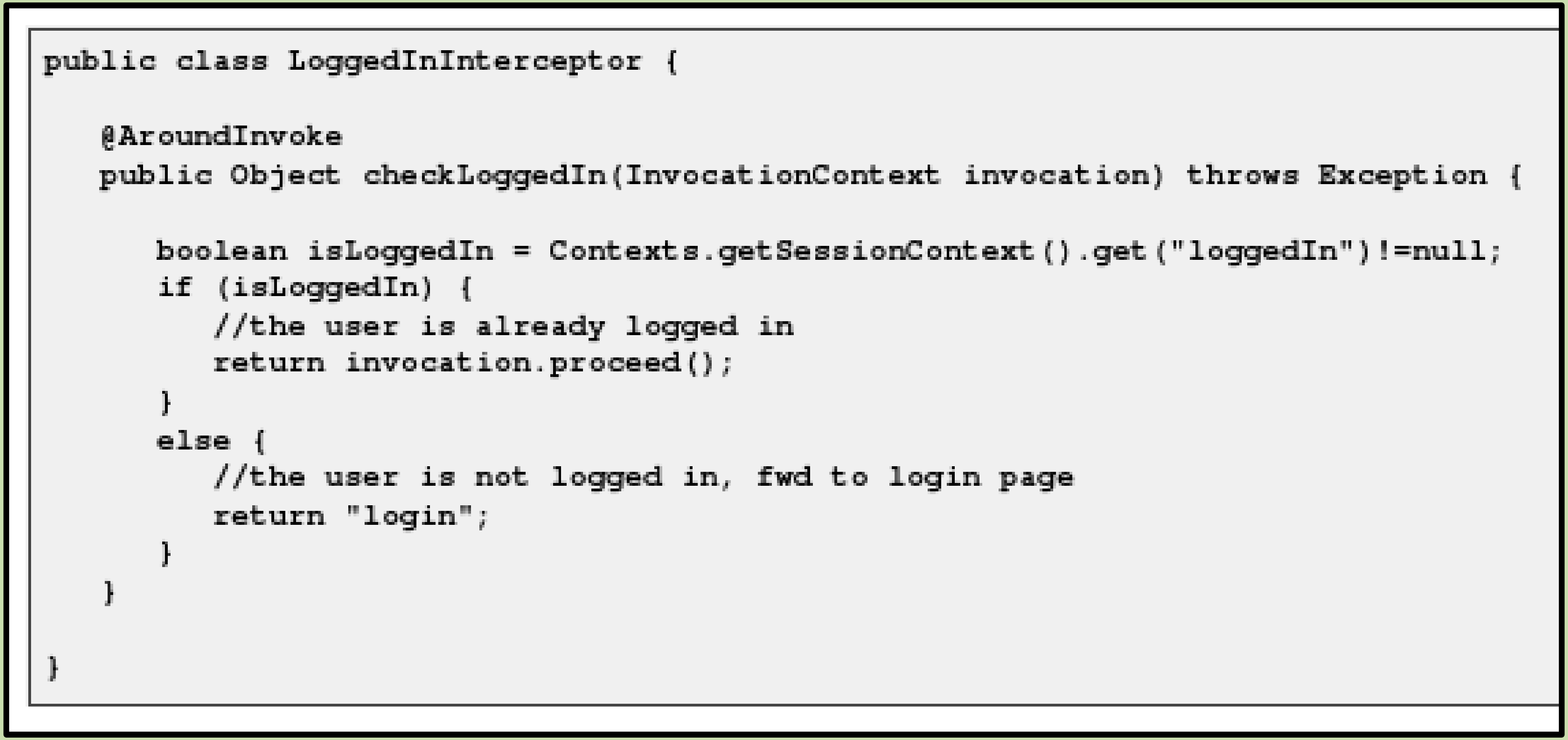


```
<components>
    <event type="hello">
        <action expression="#{helloListener.sayHelloBack}"/>
        <action expression="#{logger.logHello}"/>
    </event>
</components>
```



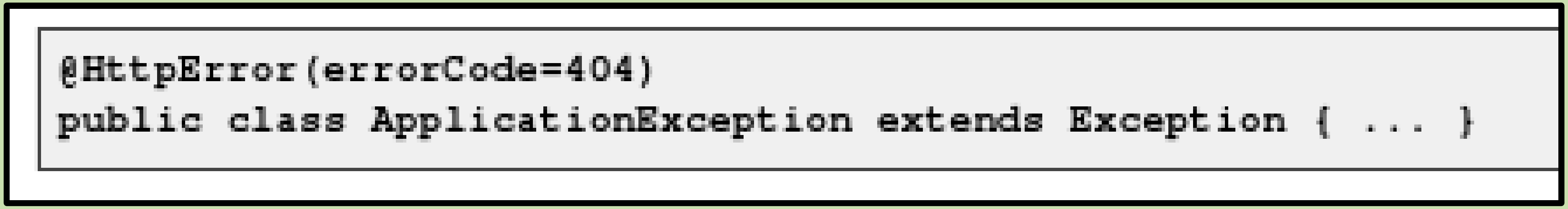
```
@Name("helloWorld")
public class HelloWorld {
     @RaiseEvent("hello")
     public void sayHello() {
```

Interceptors

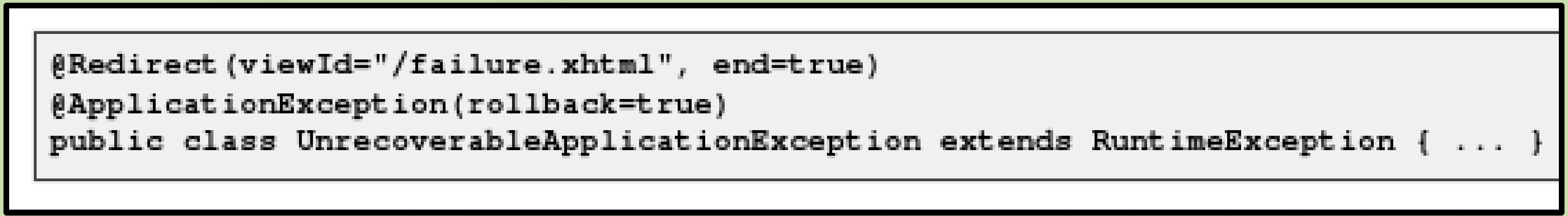


```
public class LoggedInInterceptor {

   @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";
      }
   }

}
```

Exception handling



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



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

New on Seam 2.0

- Web services can be Seam-aware;
- Seam components can be written in Groovy;
- Hot deploy;
- Eclipse support (JBossTools & RH Dev Studio);
- De-coupled from JSF;
- JBoss EL;
- Maven repository;
- Integration to Quartz and JFreeChart;
- Improved support for non-EE environments.

esJUG

Grupo de Usuários de Java do Estado do Espírito Santo

Introduction to JBoss Seam