



JBoss Seam

Accélérez votre
développement web 2.0







Malik SAHEB
Senior Solution Architect
Red Hat
malik.saheb@redhat.com

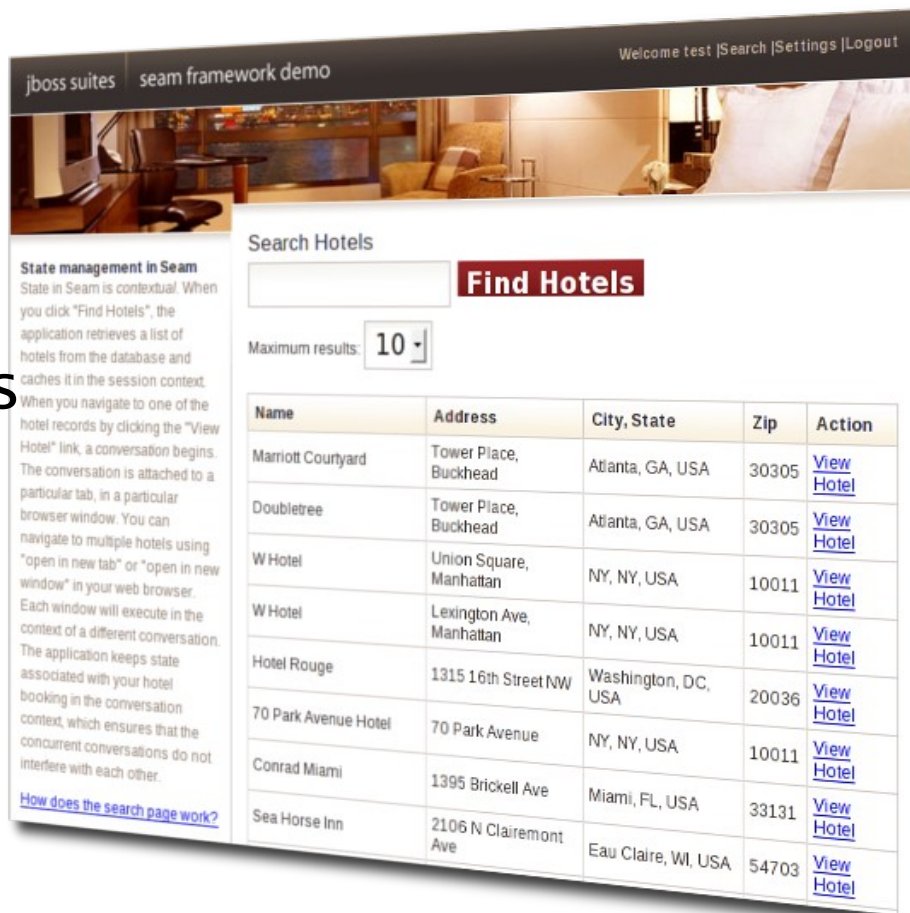
The screenshot shows the SeamFramework.org website. The main content area features a news feed with several entries:

- Seamless Wicket**: Seam 2.1.0.BETA1 brought Wicket support to Seam, but how do you use this support? In this first tutorial we will discuss how to create a seam-wicket project, and how to inject Seam components into Wicket components. A follow up tutorial will go into more depth on this. (30 Sep 2008, 19:19 CET, by Pete Muir)
- Want an introduction to Seam?**: Well, you're in luck! I spoke at JBoss User Group in Denmark in late August, and they recorded both the Seam introductory talk I did (Part 1 | Part 2 | Part 3) and the Web Beans introductory talk: (Part 1 | Part 2 | Part 3) And, if that isn't enough, you can... (23 Sep 2008, 00:14 CET, by Pete Muir)
- Integrating Seam Security with OpenLDAP**: I've written a new article describing how to integrate Seam Security with OpenLDAP, using the new features in Seam 2.1.0. You can read the article here. Comments are welcome here, enjoy. (12 Sep 2008, 07:13 CET, by Shane Bryzak)
- Want to learn about Seam 2?**: There you should attend this webinar - I'll spend about 40 minutes introducing some of the key ideas behind Seam (contexts, Seam managed persistence contexts and transactions, validation, navigation) and then they'll be 20 minutes for questions. The webinar is on Thursday, 18th. (09 Sep 2008, 23:38 CET, by Pete Muir)

Below the news feed is a commit log section:

- 01 Oct 2008, 15:48 CET
- 01 Oct 2008, 15:44 CET
- 01 Oct 2008, 15:11 CET
- 01 Oct 2008, 15:11 CET
- 01 Oct 2008, 15:03 CET

-  Introduction
-  Atouts Majeurs
-  Catalogue de Composants
-  Roadmap
-  RAD
-  Démo



Welcome test | Search | Settings | Logout

jboss suites seam framework demo

Search Hotels

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
W Hotel	Union Square, Manhattan	NY, NY, USA	10011	View Hotel
W Hotel	Lexington Ave, Manhattan	NY, NY, USA	10011	View Hotel
Hotel Rouge	1315 16th Street NW	Washington, DC, USA	20036	View Hotel
70 Park Avenue Hotel	70 Park Avenue	NY, NY, USA	10011	View Hotel
Conrad Miami	1395 Brickell Ave	Miami, FL, USA	33131	View Hotel
Sea Horse Inn	2106 N Clairemont Ave	Eau Claire, WI, USA	54703	View Hotel

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?](#)

Sim, SEAM, Sims, SIM ?

Sim



SEAM



Les Sims

SIM





Seam

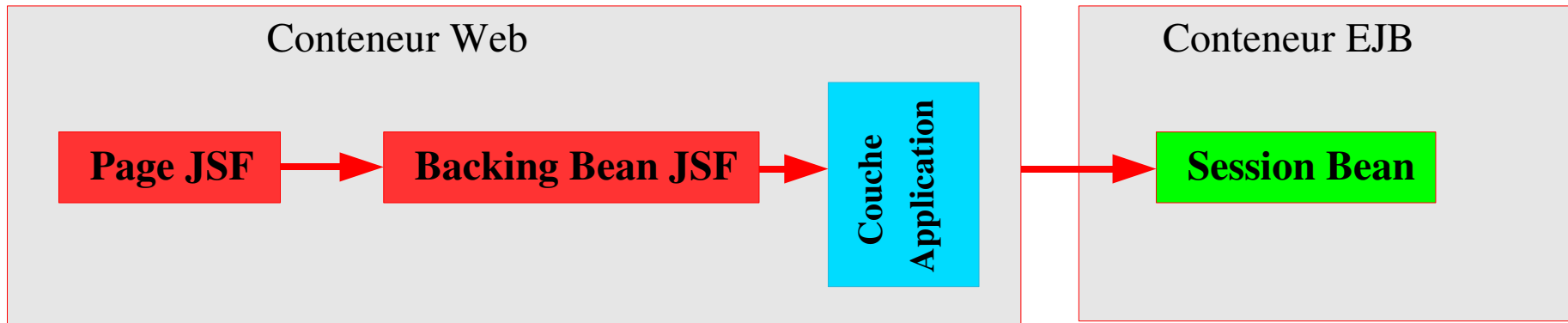
Définition

-  SEAM est un Framework de Développement qui unifie les technologies nécessaires au développement d'applications internet riches (Web 2.0)

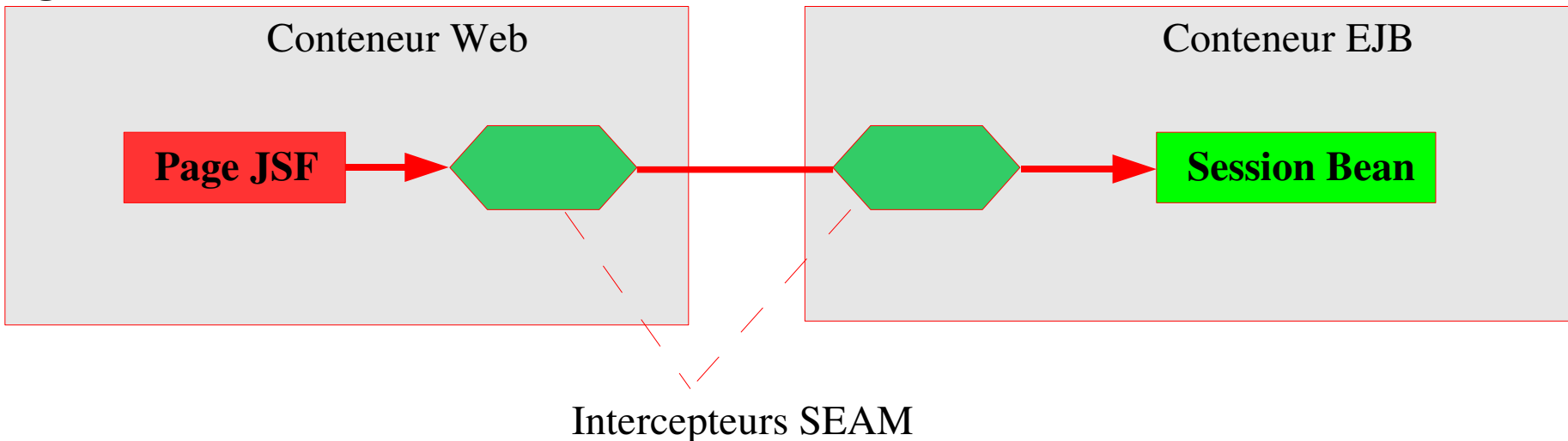
Une Idée de départ

- Prendre le meilleur de JSF et des EJB3 en s'épargnant les composants "intermediaire", et la lourdeur des configurations XML.







Integration JSF/EJB3 sans SEAM



Integration JSF/EJB3 avec SEAM



Signifie “Couture”

-  Intègre les technologies nécessaires au développement d'applications internet riches (Buzz : Web 2.0)
 - AJAX, JSF, JPA, EJB3, Wicket, Spring ...
-  Intègre des technologies utiles au développement d'applications complexes
 - BPM, Rules, Sécurité, PDF, ...
-  Introduit de nouveaux concepts
 - Conversation, Bijection
-  Règle une bonne fois des problèmes récurrents
 - Contextes de persistance étendu (LazyInitializationException), transaction, multi-sessions
http, navigateur : retour arrière, onglets
-  Réduit considérablement la quantité de XML
-  Fournit un ensemble d'outils simplifiant le développement
 - Seam-gen, EL (avec complétion), tests unitaires

Atouts Majeurs

- 👉 Composants
- 👉 Contextes
- 👉 La gestion transparente du contexte de persistance et des transactions
- 👉 Validation
- 👉 Catalogue de solutions prêtes à l'emploi
- 👉 Générateur automatique
- 👉 Documentation, les exemples et la communauté
- 👉 Aussi Testé dans Tomcat, OC4J, Glassfish, Weblogic, Websphere
- 👉 Travail de standardisation des innovations
 - 👉 JPA, Web Beans

Composant Seam

```
@Entity
@Name("item")
public class Item { ... }
```

```
@Stateful
@Name("itemEditor")
public class ItemEditorBean implements ItemEditor { ... }
```

```
@Name("itemEditor")
@Scope(ScopeType.CONVERSATION)
public class ItemEditor { ... }
```

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


Composant Seam

🎩 Accès unifié au composant, via son nom

🎩 Les expressions sont utilisables

- 🎩 Pages JSF
- 🎩 Descripteurs de composants
- 🎩 Descripteurs de pages
- 🎩 Descripteurs de pageflow
- 🎩 Les fichiers de propriétés d'internationalisation

```
@Name("hello")
public class Hello {
    @Logger private Log log;
    @In FacesMessages facesMessages;

    @PersistenceContext EntityManager entityManager;

    @Restrict("#{identity.loggedIn}")
    public void hello()
    {
        User user = (User) entityManager
            .createQuery("from User where username = #{identity.username}")
            .getSingleResult();

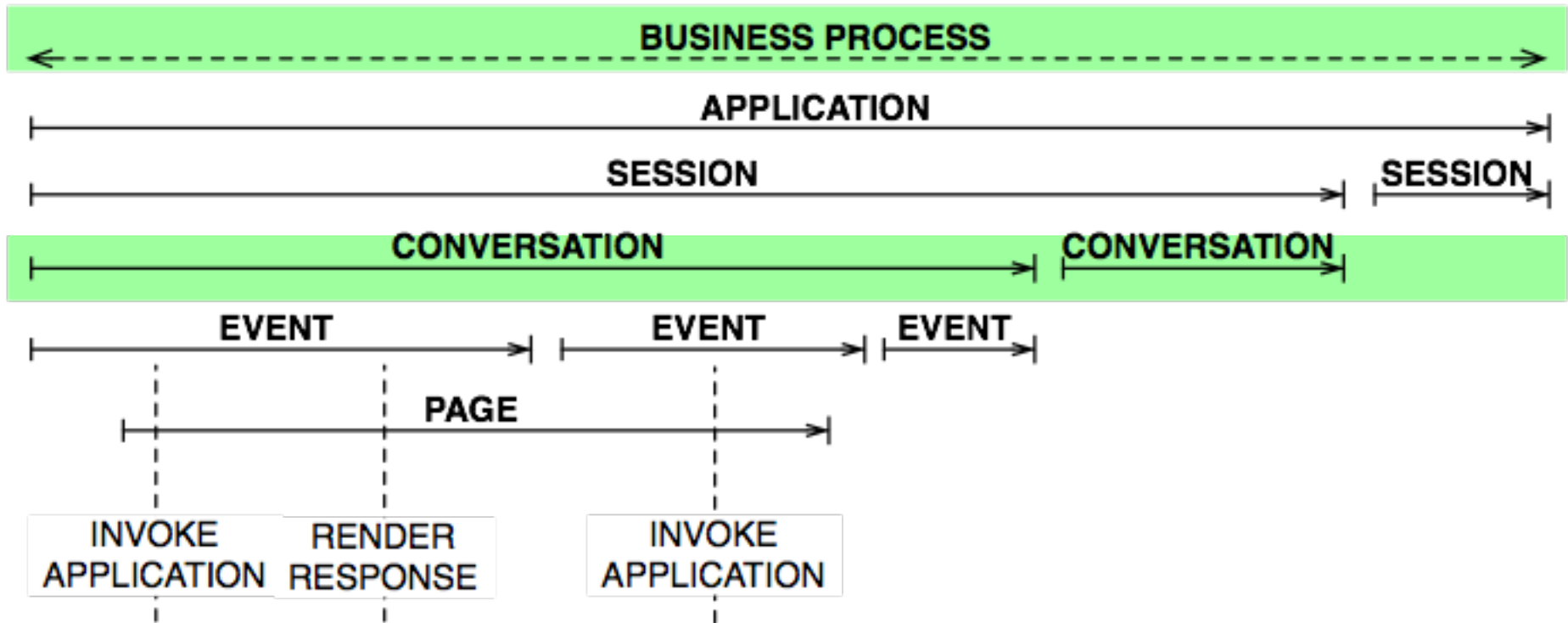
        log.info("hello from #{identity.username}");
        facesMessages.add("Hi #0 #1!", user.getFirstname(), user.getLastname());
    }
}
```

```
<h:form id="helloForm">
    <h:commandButton id="hello" value="hello!"
        action="#{hello.hello}"/>
</h:form>
```

```
<page>
<navigation from-action="#{hello.hello}">
<redirect view-id="/home.xhtml"/>
</navigation>
</page>

<page view-id="/reports.xhtml">
    <restrict>#{s:hasRole('admin')}</restrict>
</page>
```

Contextes

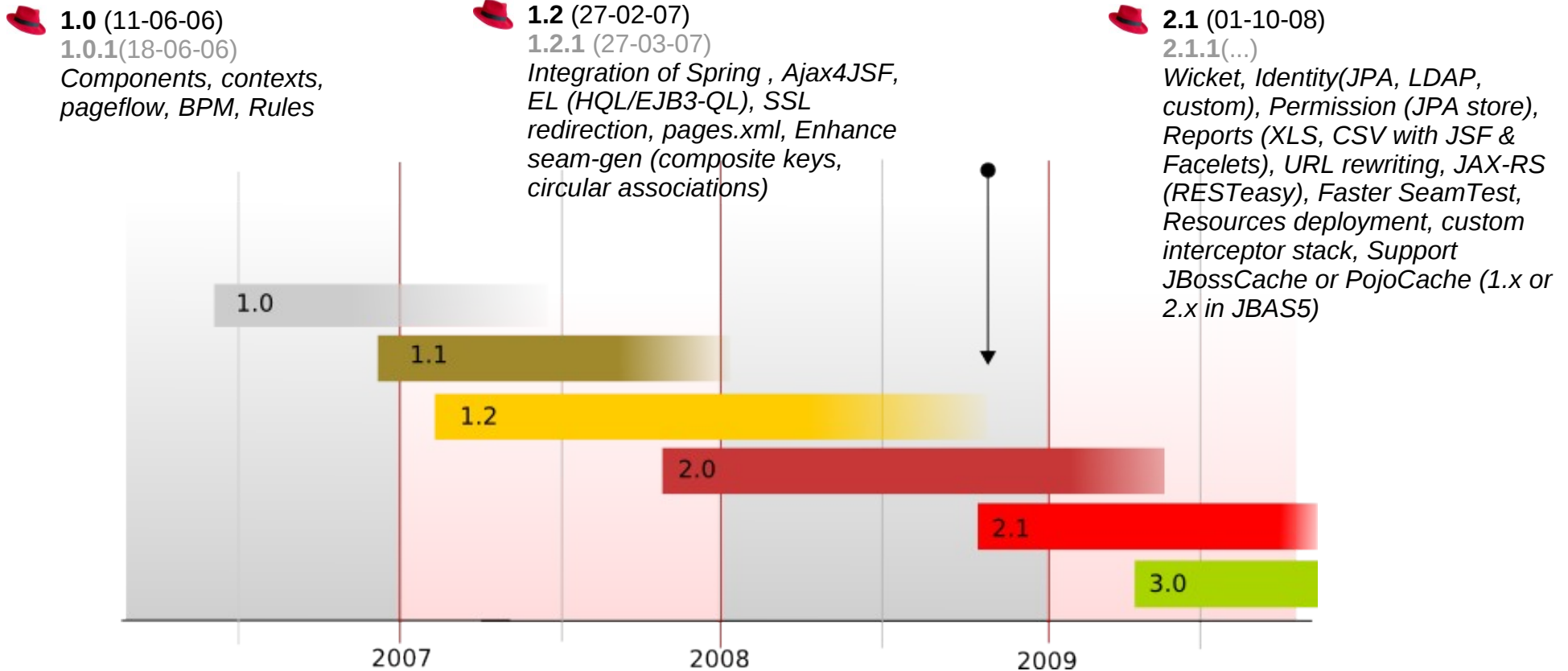



- 🚫 Les contextes application, session, requête et page ne suffisent pas
- 🚫 Contexte de processus métier
 - 🚫 stateful (-> base de données)
 - 🚫 L'utilisateur peut continuer plus tard
- 🚫 Conversation
 - 🚫 Survie à une redirection
 - 🚫 Espaces de travail


Catalogue de Composants


- 🎩 Evènements
- 🎩 Validation
- 🎩 Groovy
- 🎩 Wicket
- 🎩 Framework applicatif
- 🎩 I18n
- 🎩 Texte formaté
- 🎩 PDF, XLS, RSS, Email
- 🎩 Captcha
- 🎩 Rich JSF Components
(RichFaces + Ajax4JSF)
- 🎩 GWT
- 🎩 Spring
- 🎩 Hibernate Search
- 🎩 Messages asynchrones
- 🎩 Caches
- 🎩 Services web
- 🎩 Appels distants
- 🎩 Test unitaire
- 🎩 Page de debug

Roadmap



 **1.1** (13-12-06)
1.1.1 (17-01-07), **1.1.5** (31-01-07), **1.1.6** (07-02-07)
 Asynchronicity, seam-gen, Seam Application Framework, Ajax integration, Clustering, iText

 **2.0** (01-11-07)
2.0.1 (30-01-08), **2.0.2** (15-05-08), **2.0.3CR1** (13-06-08)
 WS, Groovy, Hot deploy, Enhanced JBDS tools, JSF decoupled (GWT sample), JSF 1.2, Enhanced EL (JBoss EL), Maven, Integrations (Quartz, Hibernate Search, JFreeChart), non EE environments

 **3.0** (2009)
 Web Beans, JSF 2, JPA 2



RAD

The screenshot shows the JBoss Developer Studio interface. The main window displays a web browser at the URL `http://localhost:8080/HappyGen/login.seam?cid=10&clr=true`. The browser shows a login page with the following content:

- Message: "Please log in first"
- Section: "Login"
- Text: "Please login using any username and password"
- Form fields: "Username" and "Password" (both empty)
- Checkbox: "Remember me" (unchecked)
- Button: "Login"
- Footer: "Powered by Seam. Generated by seam-gen."

The project explorer on the left shows the following structure:

- HappyGen-ear
 - EarContent
 - resources
 - HappyGen-ds.x
- HappyGen-ejb
 - ejbModule
 - org.domain.Hap
 - Bonus.java
 - BonusHome.
 - BonusList.jar
 - Dept.java
 - DeptHome.je
 - DeptList.jav
 - Dummy.java

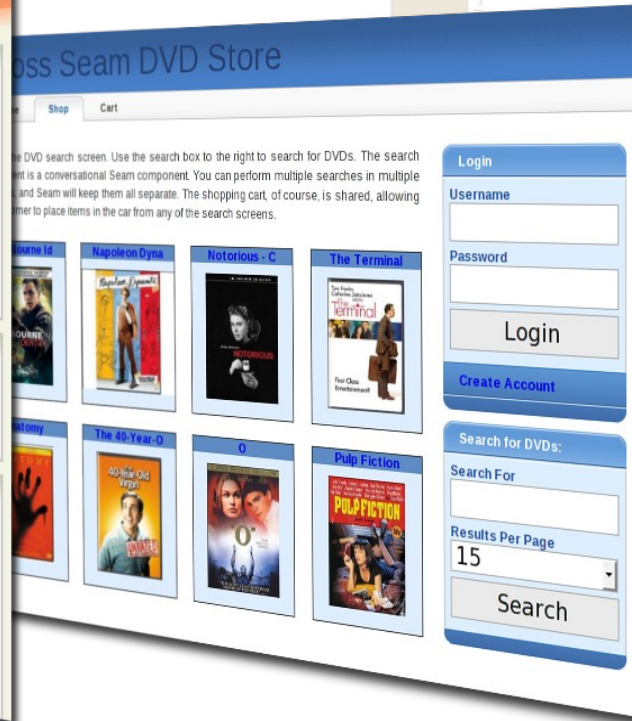


Essayez-le !

Liens utiles

Site internet <http://www.seamframework.org>

Blog <http://in.relation.to>



CONTENTS INCLUDE:

- Bijection in a Nutshell
- Contextual Components
- Common Application Configuration
- Seam Security
- Application Framework
- Hot Tips and More...

Core Seam

By Jacob Orshalick

ABOUT SEAM

Seam is a next generation web framework that integrates standard Java EE technologies with a wide variety of nonstandard technologies into a consistent, unified, programming model. Seam drove the development of the Web Beans specification (JSR-299) and continues to develop innovations that are changing the face of web development as well as Java EE technologies. If you haven't taken a look at Seam, I suggest you do.

As you develop Seam applications, you'll find this quick reference a handy guide for understanding core concepts, configuration, and tool usage. This quick reference is not intended to cover all of what Seam provides, but will cover the most commonly used annotations and XML elements as of Seam 2.1. In addition, this guide will point you to examples distributed with Seam to see real examples of how the configuration options can be used in practice.

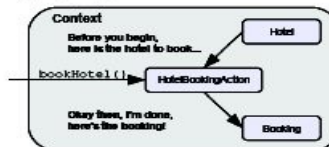
BIJECTION IN A NUTSHELL

Dependency injection is an inversion of control technique that forms the core of modern-day frameworks. Traditionally objects have held the responsibility for obtaining references to the objects they collaborate with. These objects are extroverted as they reach out to get their dependencies. This leads to tight coupling and hard to test code.

Dependency injection allows us to create introverted objects. The objects dependencies are injected by a container or by some external object (e.g. a test class). Bijection is described by the following formula:

$$\text{dependency injection} + \text{context} = \text{bijection}$$

With bijection, when dependencies are injected context counts! Dependencies are injected prior to each component method invocation. In addition, components can contribute to the context by objecting values.



As you can see the HotelBookingAction is scoped to and executes within a context. This behavior allows us to quit worrying about shuffling values into and out of contexts like the HttpSession, allows components to hold state, and unifies the component model across application tiers.

CONTEXTUAL COMPONENTS

When a method is invoked on a component, its dependencies are injected from the current context. Seam performs a context lookup in the following order of precedence: Event Scope, Page Scope, Conversation Scope, Session Scope, Business Scope, Application Scope.

Component Annotations

Component Definition Annotations

In order for your Seam components to take advantage of bijection, you must register them with the Seam container. Registering your component with Seam is as simple as annotating it with @Name. The following annotations will register your component and define its lifecycle.

Annotation	Use	Description
@Name	Type	Declares a Seam component by name. The component is registered with Seam and can be referenced by name through Expression Language (EL), injection, or a context lookup.
@Scope	Type	Defines the scope (or context) the Seam component will be placed into by the container when created.
@Inject	Type	Specifies that a component should be created when being injected if not available in the current context.
@Startup	Type	Indicates that an application scoped component should be created when the application is initialized or that a session component should be created when a session is started. Not valid for any other contexts.
@Install	Type	Declares whether a Seam component should be installed based on availability of dependencies or precedence.
@Role	Type	Defines an additional name and scope associated with the component. The @role annotation allows definition of multiple roles.

Get support for Seam 2

JBoss Enterprise Application Platform now includes Seam 2

- JBoss Enterprise Application Platform pre-integrates JBoss Application Server, Seam, and Hibernate
- Latest feature pack includes support for Seam 2
- Includes caching, clustering, messaging, transactions, and integrated web services stack
- Support for industry-leading Java and newer technologies, including JAX-WS, EJB 3.0, JPA 1.0, JSF 1.2, and JTA 1.1

Download today: jboss.com/download

A red fedora hat with a black band. The word "Démo" is written in white on the black band. The hat is shown from a side profile, tilted slightly. The background is a plain, light color.

Démo