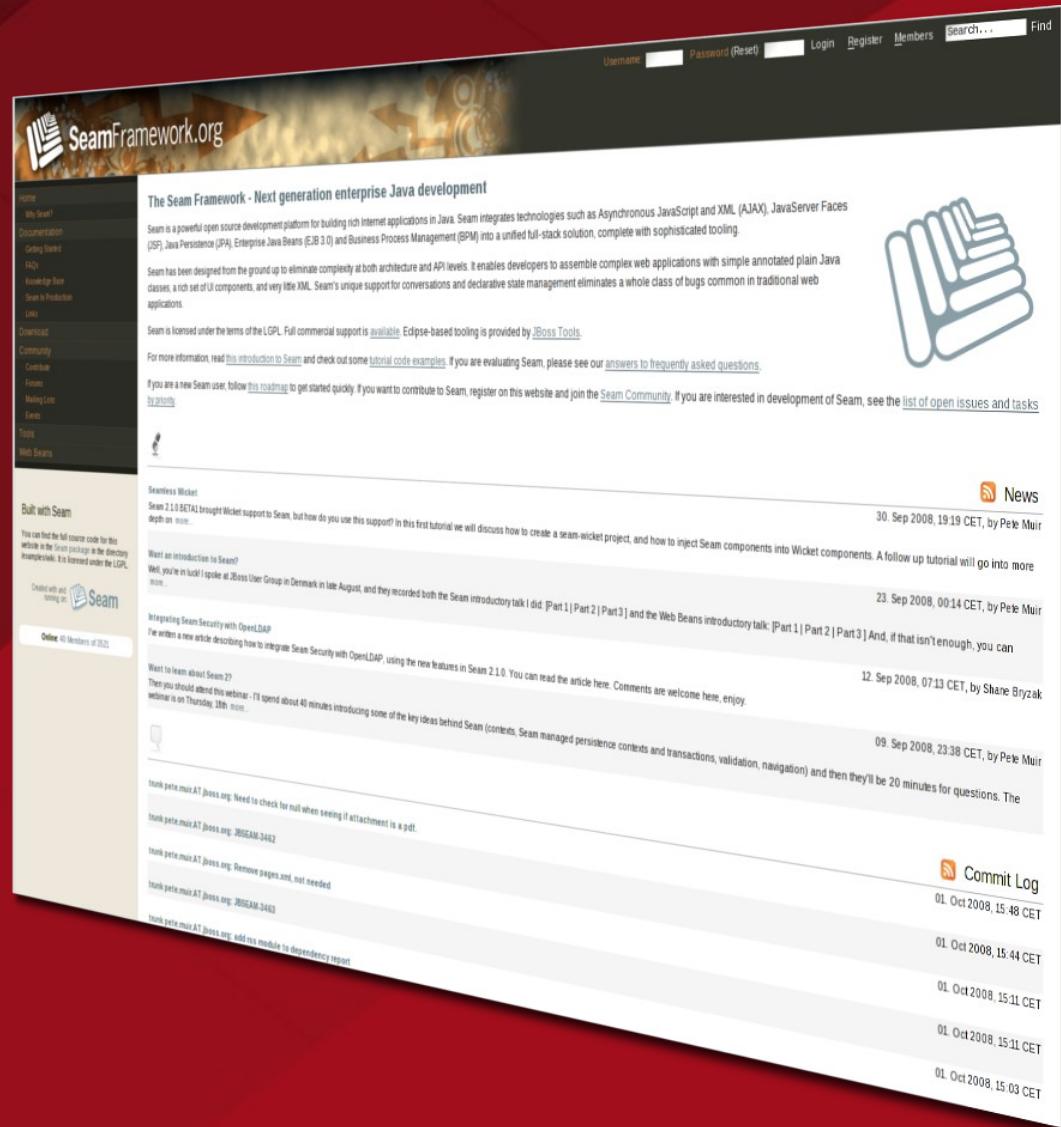


JBoss Seam

Accélérez votre développement web 2.0

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



The Seam Framework - Next generation enterprise Java development

Seam is a powerful open source development platform for building rich internet applications in Java. Seam integrates technologies such as Asynchronous JavaScript and XML (AJAX), JavaServer Faces (JSF), Java Persistence (JP)A, Enterprise Java Beans (EJB 3.0) and Business Process Management (BPM) into a unified full-stack solution, complete with sophisticated tooling.

Seam has been designed from the ground up to eliminate complexity at both architecture and API levels. It enables developers to assemble complex web applications with simple annotated plain Java classes, a rich set of UI components, and very little XML. Seam's unique support for conversations and declarative state management eliminates a whole class of bugs common in traditional web applications.

Seam is licensed under the terms of the GPL. Full commercial support is available. Eclipse-based tooling is provided by JBoss Tools.

For more information, read this [introduction to Seam](#) and check out some [tutorial code examples](#). If you are evaluating Seam, please see our [answers to frequently asked questions](#).

If you are a new Seam user, follow this [roadmap](#) to get started quickly. If you want to contribute to Seam, register on this website and join the [Seam Community](#). If you are interested in development of Seam, see the [list of open issues and tasks](#).

by [petermuir](#)

News

30. Sep 2008, 19:19 CET, by Peter Muir

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...

23. Sep 2008, 00:14 CET, by Peter Muir

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...

12. Sep 2008, 07:13 CET, by Shane Bryzak

Want to learn about Seam 2? Then you should attend this webinar - it's 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...

09. Sep 2008, 23:38 CET, by Peter Muir

I've written a new article describing how to integrate Seam Security with OpenLDAP. You can read the article here. Comments are welcome here; enjoy...

01. Oct 2008, 15:48 CET

thank peter.muir AT jboss.org: Need to check for null when seeing if attachment is a pdf.

01. Oct 2008, 15:44 CET

thank peter.muir AT jboss.org: JBoss Seam 34462

01. Oct 2008, 15:31 CET

thank peter.muir AT jboss.org: Remove pages.xml, not needed

01. Oct 2008, 15:31 CET

thank peter.muir AT jboss.org: JBoss Seam 3462

01. Oct 2008, 15:03 CET

thank peter.muir AT jboss.org: add res module to dependency report

Commit Log

SOMMAIRE

► Introduction

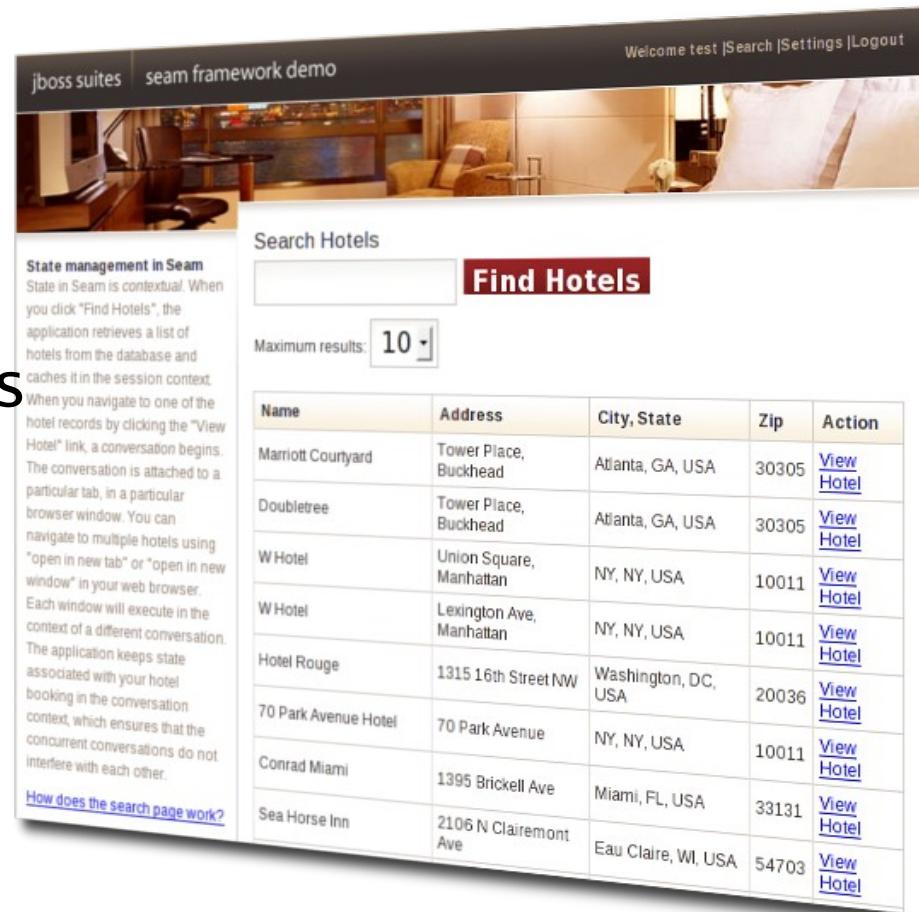
► Atouts Majeurs

► Catalogue de Composants

► Roadmap

► RAD

► Démo



The screenshot shows a web application interface for "jboss suites seam framework demo". The top navigation bar includes links for "Welcome test", "Search", "Settings", and "Logout". Below the header, there's a banner image of a hotel lobby. The main content area has a heading "Search Hotels" with a search input field and a "Find Hotels" button. A dropdown menu labeled "Maximum results: 10" is open. To the right is a table listing ten hotel entries:

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

A note on the left side of the table reads: "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." A blue link "How does the search page work?" is also present.



Sim, SEAM, Sims, SIM ?

Sim



SEAM



Les Sims

SIM



JBoss Seam !



Seam

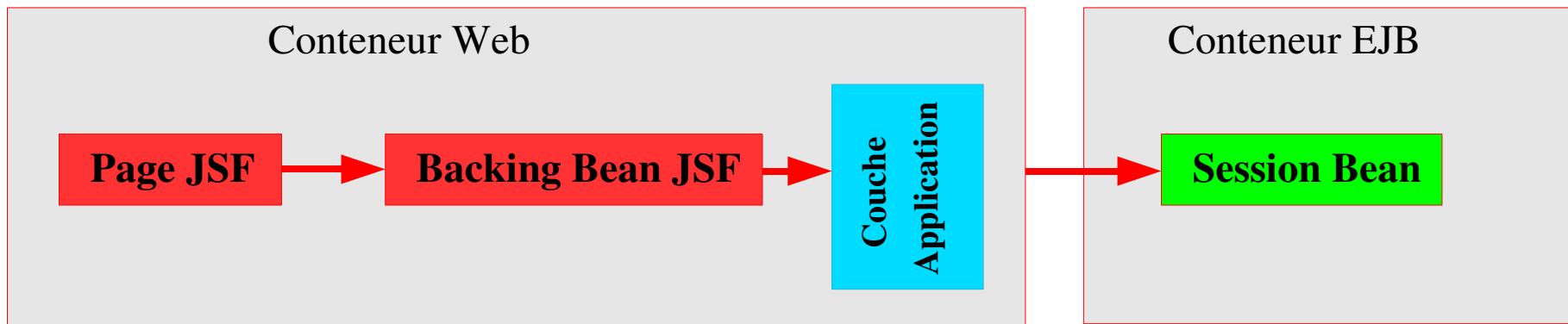
👉 Définition

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

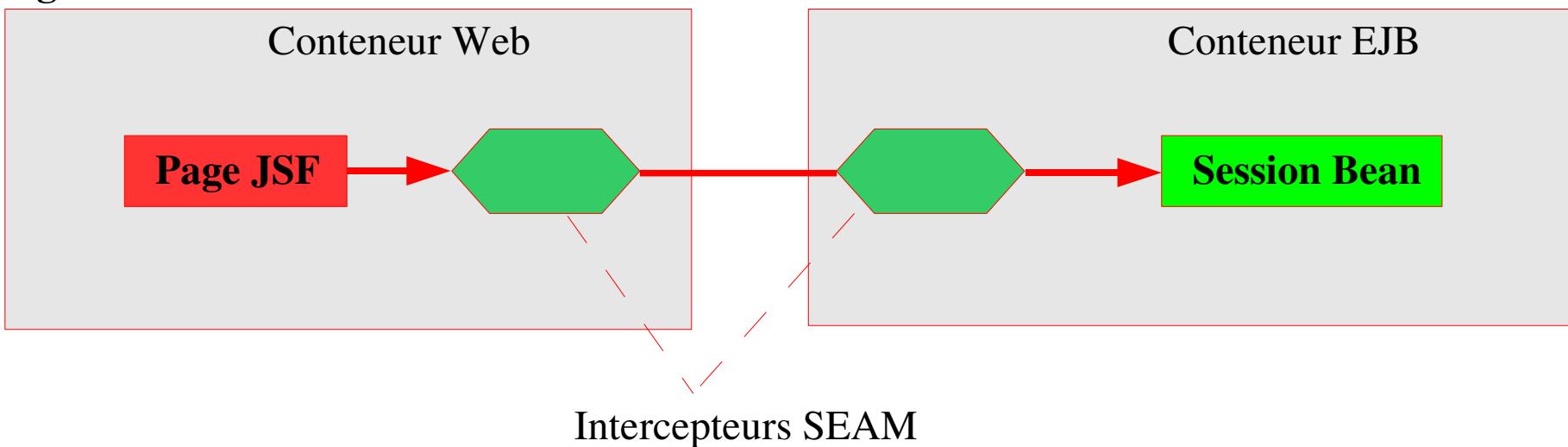
Une Idée de départ

- Prendre le meilleure de JSF et des EJB3 en s'épargnant les composants “intermédiaire”, et la lourdeur des configurations XML.

Integration JSF/EJB3 sans SEAM



Integration JSF/EJB3 avec SEAM





JBoss 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;

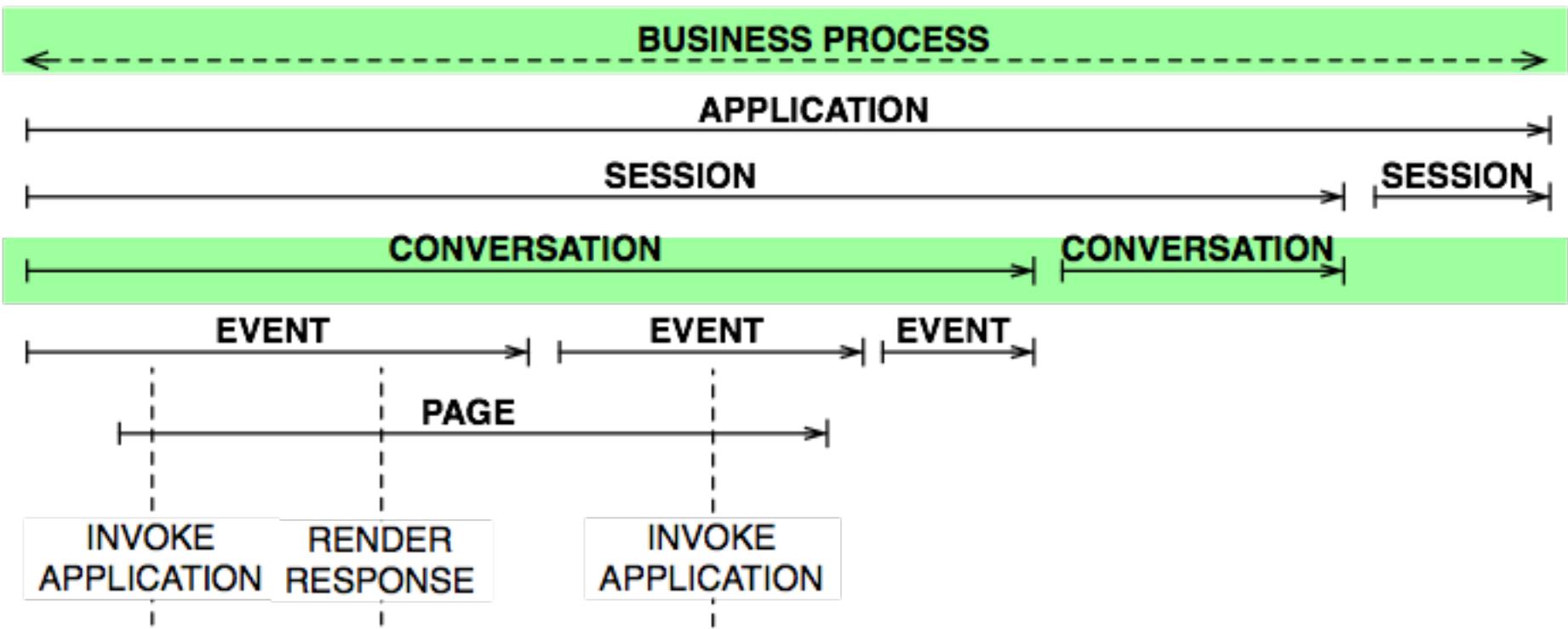
    @Restrict("#{identity.loggedIn}")
    public void hello()
    {
        User user = (User) entityManager
            .createQuery("from User where username = #{identity.username}")
            .getSingleResult();

        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

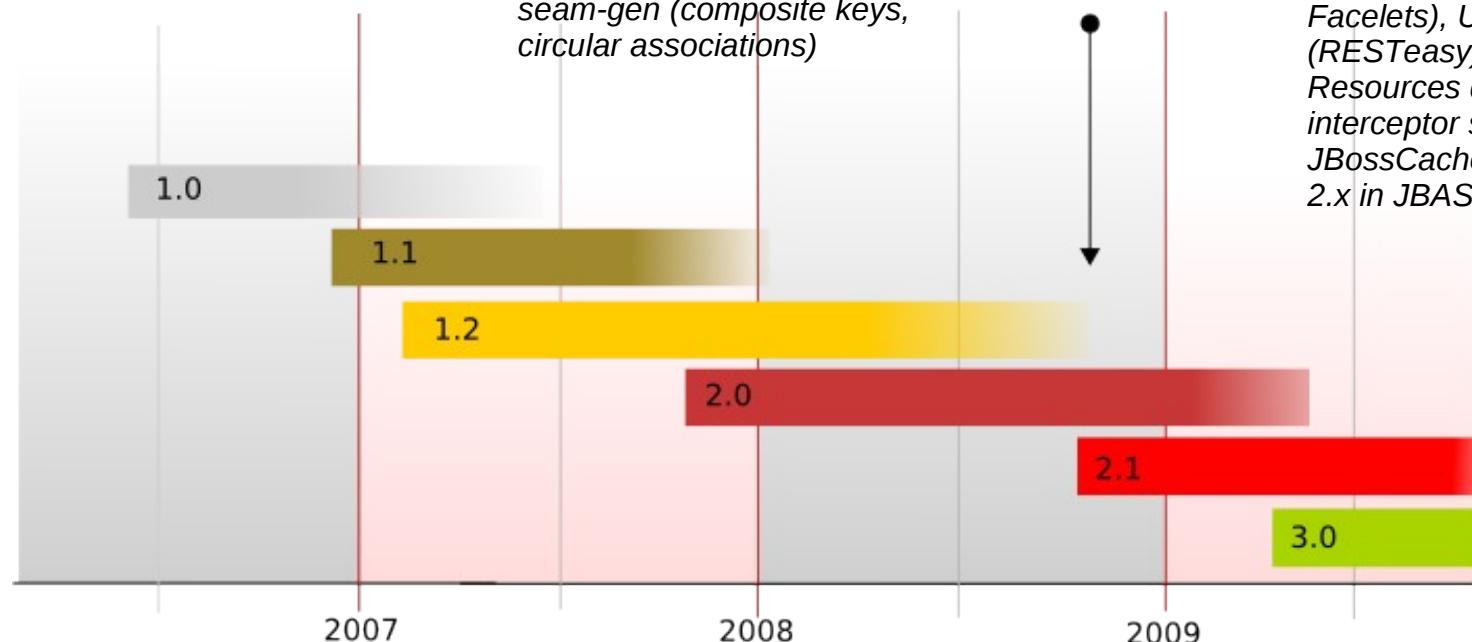
- Evènements
- Validation
- Groovy
- Wicket
- Framework applicatif
- I18n
- Texte formaté
- PDF, XLS, RSS, Email
- Captcha
- Rich JSF Components
(RichFaces + Ajax4JSF)
- GWT
- Spring
- Hibernate Search
- Messages asynchrones
- Caches
- Services web
- Appels distants
- Test unitaire
- Page de debug

Roadmap

 **1.0** (11-06-06)
1.0.1(18-06-06)
Components, contexts,
pageflow, BPM, Rules

 **1.2** (27-02-07)
1.2.1 (27-03-07)
Integration of Spring , Ajax4JSF,
EL (HQL/EJB3-QL), SSL
redirection, pages.xml, Enhance
seam-gen (composite keys,
circular associations)

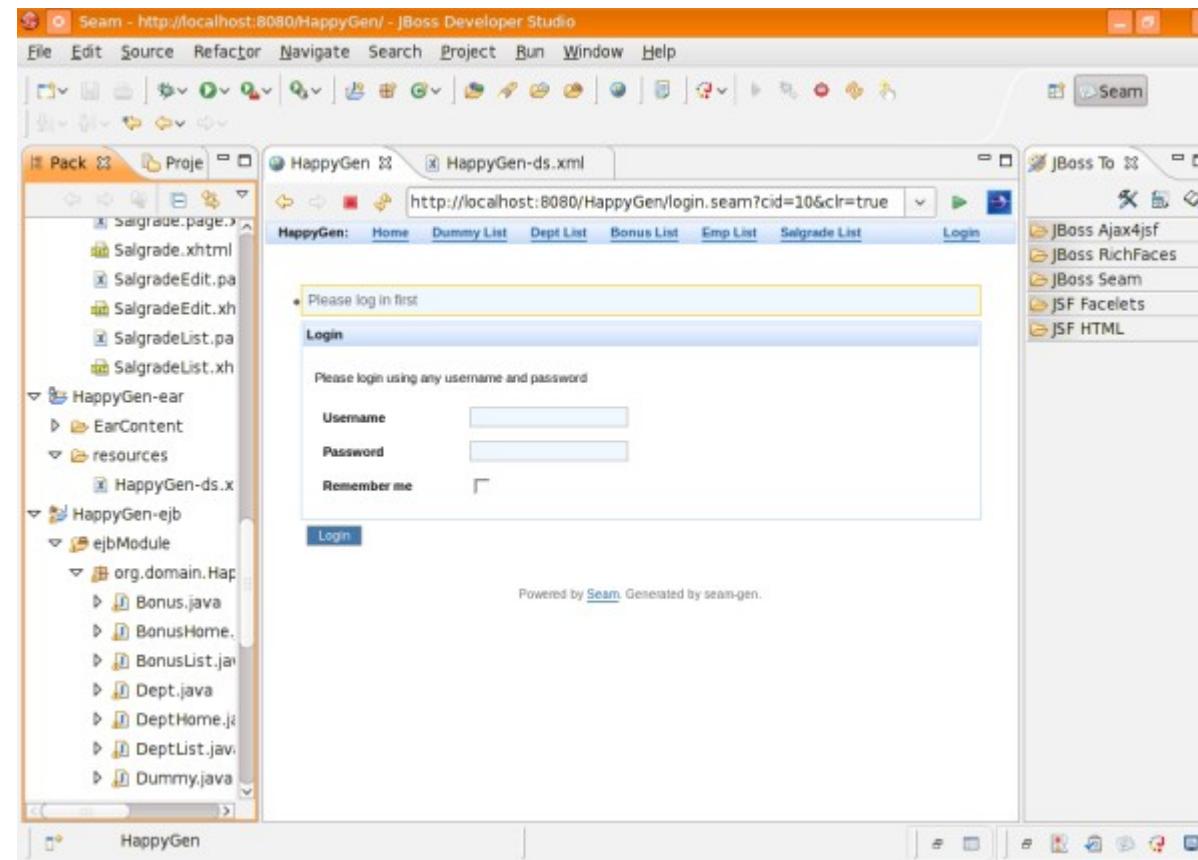
 **2.1** (01-10-08)
2.1.1(...)
Wicket, Identity(JPA, LDAP,
custom), Permission (JPA store),
Reports (XLS, CSV with JSF &
Facelets), URL rewriting, JAX-RS
(RESTeasy), Faster SeamTest,
Resources deployment, custom
interceptor stack, Support
JBossCache or PojoCache (1.x or
2.x in JBAS5)



 **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



Essayez-le !



Liens utiles

- Site internet <http://www.seamframework.org>
- Blog <http://in.relation.to>



In Relation To...

Bean Validation reference implementation and Bootstrap API

02 Oct 2008, 17:21 CET, by Emmanuel Bernard

We have been recently working a lot on the Bean Validation spec (JSR 303) and have two good news for you:

- The bootstrap API proposal is out.
- The reference implementation is available now (and of the blog).

Please have a look at both and give us [feedback](#). It is still time to make it better ;)

Bootstrap API

The primary goal of the bootstrap API is trivial. Provide access to a validation factory capable of building validator instances. These validator instances can then be used to validate the domain model.

Some additional goals have been pursued as well:

- simple to use
- type safe
- extensible and flexible to the container environment (Java EE, Java SE, dependency injection framework, OSGi-esque containers and so on)
- support multiple implementations

Examples

The best way to learn an API is to see it in action.

Everyday bootstrap

The first example shows the most simple way and also the most common:

```
ValidationFactory factory = ValidationProviderBuilderImpl.create();
factory.addFactoryListener(new FactoryListener() {
    public void validate(ValidationContext context) {
        Validator validator = factory.getValidatorFor(context);
        validator.validate(domainObject);
    }
});
```

This creates a thread-safe ValidationFactory object that should be cached. In this example, the default Bean Validator provider is used unless an explicit provider implementation is defined in the Bean Validation XML configuration file.

Container driven bootstrap

The second example defines some of the configuration elements:

```
ValidationFactory factory = ValidationProviderBuilderImpl.create();
factory.addFactoryListener(new FactoryListener() {
    public void validate(ValidationContext context) {
        Validator validator = factory.getValidatorFor(context);
        validator.validate(domainObject);
    }
});
```



JBoss Seam DVD Store

The DVD search screen. Use the search box to the right to search for DVDs. The search result is a conversational Seam component. You can perform multiple searches in multiple tabs and Seam will keep them all separate. The shopping cart, of course, is shared, allowing you to place items in the cart from any of the search screens.

Login

Search for DVDs:

Search For:

Results Per Page:

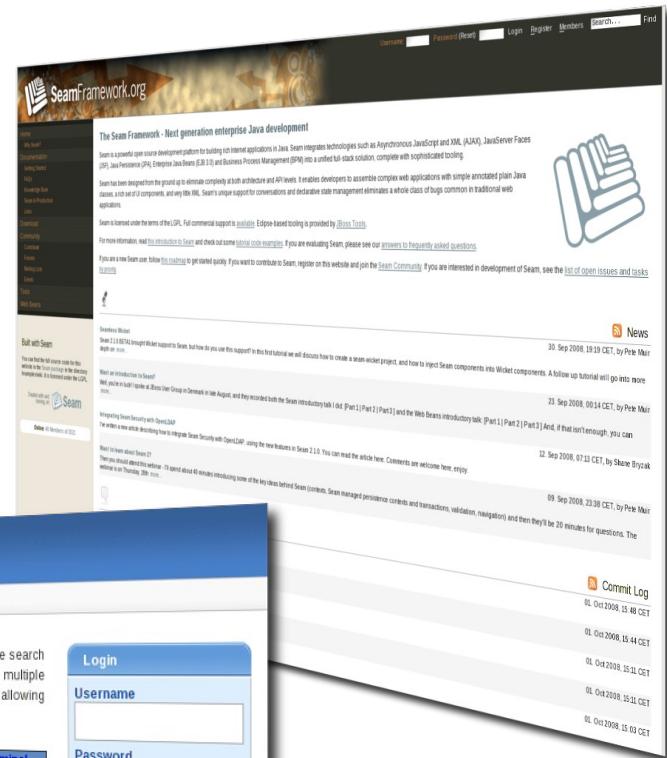
Search

Entries Added

Date	Entries Added
12 Sep 2008	200
11 Sep 2008	100
10 Sep 2008	50
09 Sep 2008	20
08 Sep 2008	10

Tags

- Seam (53)
- Seam News (26)
- JBoss Tools (16)
- JBoss Tools Eclipse (15)
- JBoss Wiki (11)
- Eclipse (8)
- Route (7)
- Web Services (6)
- Web Beans Sneak Peak (5)
- EE5 Wishlist (5)
- Bean Validation Sneak Peak (4)
- Hibernate Search Interoperability (3)



SeamFramework.org

The Seam Framework - Next generation enterprise Java development

Seam is a powerful open source integration platform for building rich Internet applications in Java. Seam integrates technologies such as Asynchronous JavaScript and XML (AJAX), JavaServer Faces (JSF), Java Persistence (JPA), Enterprise JavaBeans (EJB 3.0) and Business Process Management (BPM) into a unified full-stack solution, complete with sophisticated tooling.

Seam has been designed from the ground up to eliminate complexity at both architecture and API levels. It enables developers to assemble complex web applications with simple annotated plain Java classes, a host of J components, and very little XML. Seam's unique support for conversations and declarative state management eliminates a whole class of bugs common in traditional web applications.

Seam is licensed under the terms of the LGPL. Full commercial support is available. Eclipse based tooling is provided by [JBoss Tools](#).

For more information, visit the [introduction to Seam](#) and check out some [sample code examples](#). If you are evaluating Seam, please see our [answers to frequently asked questions](#).

If you are a new Seam user, follow [this tutorial](#) to get started quickly. If you want to contribute to Seam, register on this website and join the [Seam Community](#). If you are interested in development of Seam, see the [list of open issues and tasks](#).

News

- 30 Sep 2008, 19:19 CET, by Pete Muir: [Seam 2.1.0 RC2 brought Wicket support to Seam, but how do you use this support? In this 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.](#)
- 23 Sep 2008, 00:14 CET, by Pete Muir: [Want an introduction to Seam? Well, we're in luck! Update at JBoss User Group in Copenhagen 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 find a new article describing how to integrate Seam Security with OpenSSO!](#)
- 32 Sep 2008, 07:13 CET, by Stéphane Brisabois: [Want to know what Seam is? Then you should check out this article. I'll spend about 40 minutes introducing some of the key ideas behind Seam \(conversations, Seam managed persistence contexts and transactions, validation, navigation\) and then they'll be 20 minutes for questions. The article is available in French.](#)
- 09 Sep 2008, 23:38 CET, by Pete Muir: [The other day we wrote an article describing some of the key ideas behind Seam \(conversations, Seam managed persistence contexts and transactions, validation, navigation\) and then they'll be 20 minutes for questions. The article is available in French.](#)

Commit Log

- 01 Oct 2008, 15:48 CET
- 01 Oct 2008, 15:44 CET
- 01 Oct 2008, 15:31 CET
- 01 Oct 2008, 15:11 CET
- 01 Oct 2008, 15:03 CET

<http://refcardz.dzone.com/refcardz/core-seam>

#31

Get More Refcardz! Visit refcardz.dzone.com

Core Seam
www.dzone.com



Brought to you by...



Core Seam

By Jacob Orshalick

CONTENTS INCLUDE:

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

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.

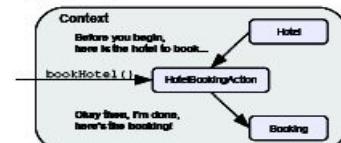
BIJECTION IN A NUTSHELL

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 injecting 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	Type	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.
@AutoCreate	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 close-up photograph of a red fedora hat with a black band. The hat is tilted slightly to the left. The word "Démo" is overlaid in white text on the black band.

Démo