Seam

Placeholder for the seam project

General

The Seam project from JBoss is a very new (at the time of writing it was in beta stage) and very interesting new backend project for JSF.

It is closer to Shale than to a component library, because it tries to rely on the backend only.

It solves many issues raised by the current implementation of web centric applications while trying to be as easy as possible for solving those issues. (Something raised and achieved by the also excellent Ruby Rails project)

What is Seam and what it is not

- Seam as such is no component library, its focus is entirely on the backend
- Seam as such is not a direct competitor to Spring, however there are some overlapping areas, but Spring tries to be a general framework which adds additional value to any application not just web applications, Seam at the current state sees itself as an extension to JSF which should make application programming within the JSF realm and within the realm of EJB 3.0 as easy as possible. You surely will see comparisons in the near future, because in the past there was some kind of competition between the JBoss people and the people behind the Spring framework, but to the authors opinion both comparisons are not really valid and probable, it has not been tested, a mixture of both frameworks is/will be possible.
- Seam does not program programs for you automatically. But you can probably expect toolvendors supporting it in the long run, due to the support of EJB 3.0 in projects like Eclipse JSR220 which blend perfectly into the Seam infrastructure

Advantages and disadavantages

Due to its beta status, little is know about its advantages and disadvantages to the author of this page, however after reading the docs and trying out the demo following seems to be possible:

Advantages

- It seems to have a very clear solution to the scoping problem (like x:saveState and shale Dialog), the Seam approach focuses around being

able to define EJB Session beans with a small number of annotations for a specific stateful scope instead of being stateless or stateful. Also as it seems it solves the problems many of those dialog/scoping systems have with double opened windows.

- It seems to integrated some web application security aspects directly into the scoping layer. As it looks like, seam tries to solve

the problem of trying to hack into certain pages by altering the URL by simply covering that on the scoping level.

- It seems to have basic AOP, the Seam paper indicates that some kind of interceptor mechanism was integrated. The AOP however is by far

not as extensive as the mechanisms used by Spring or AspectJ

- It also has some kind of runtime object wiring system, like spring and JSF have (one with Inversion of object control, the other one

with backend beans), the Seam object wiring system is called inject and outject and is tightly integrated with JSFs Backend Bean system.

- Tries to dock on standards only: With being tightly integrated into JSF and EJB 3.0 it is tightly integrated into an existing and an upcoming standard.

Disadvantages

- Heavy reliance on EJB one way or the other: Having to use a dedicated EJB container often is not really what developers like, due to the extra burden and higher startup times those containers have. Seam tries to resolve this problem by allowing the installment of a JBoss ejb3.0 embedded engine. This engine however has lots of dependencies into third party libraries, which might cause version problems in projects.
- Beta status, this is a temporary problem, the project currently still is beta, but that will change hopefully soon
- The inject outject metaphor is sort of confusing, compared to other Inversion of Object Control mechanism, whether this metaphor will hold within the eyes of the developers, only time will tell
- Unknown entity, Seam is due to its new age an unknown entity within the realm of J2EE only time will tell if it will get wider adoption.

Useful Links

*Seam Project Page
*JBoss
*Spring Framework Project Page