

SFSB SLSB Farkları

Callback'ler

Burak Bayramlı

Bu belge, \LaTeX ile üretilmiştir

EJB Çeşitleri

- Hafızalı (Stateful) Session Bean
- Hazızasız (Stateless) Session Bean
- Entity Bean

Hafızasız (Stateless) Session Bean

- İki çağrı arasında, Bean üzerindeki durum kaybolur
- Hafızasız Bean'ler kütüphane arayüzü (API) gibidirler

Ne Zaman Kullanılmalı

- Bağlanan, ağır müşteri (fat client)
 - ★ Hatırlanması gereken veri çağırının tarafında tutuluyor.
 - ★ Servis tarafı hiçbir şeyi hatırlamaz.

Örnek

```
CustomerService obj = .. // JNDI  
  
obj.recordCustomer('1234', 'ahmet', 'vs');  
...  
obj.recordCustomer('1234', 'veli', 'vs');
```

Dikkat

- Stateless şartlarında, aynı nesne, işi bittikten sonra birkaç değişik kullanıcıya verilebilir.

Hafızalı (Stateful) Session Bean

- İki çağrı arasında, bean üzerindeki durum (state) hatırlanır.
- Küme ortamında bir bean, kümedeki öteki birimlere kopyalanabilir (replication).
- Böylece çökmeden kurtulmak mümkün olur (fail over).

Ne Zaman Kullanılmalı

- Bađlanan, ağır müşteri (fat client) ise, ve elde tutulan referans üzerinde servis tarafında bazı şeyleri hatırlanmasının sağlanması için kullanılmalı.

Örnek

```
AdderService obj = .. // JNDI  
  
obj.increment(10);  
  
obj.increment(20);  
  
System.out.println(obj.getSum()); // Sonuç: 30
```

Callback'ler

Callback'ler

EJB Container (JBoss) ile bean'lerin etkileşiminde haberdar olmak istediğiniz olaylara takılan çengel kodlar callback fonksiyonlarıdır.

Callback'ler

- @PostConstruct
- @PreDestroy
- @PrePassivate
- @PostActivate
- @Init

- @Remove

Hangi Callback Ne Zaman Çağırılır?

@PostConstruct

Bir Bean yaratıldıktan hemen sonra bu şekilde işaretlenen metot çağırılır. Bu annotation, hem stateful hem de stateless bean'ler için geçerli.

@PreDestroy

Bir Bean havuzdan çıkartıldıktan sonra yâni yokedildikten sonra bu şekilde işaretlenen metot çağırılır. Hem stateful hem de stateless bean'ler için geçerli.

@PrePassivate

Eğer bir bean çok uzun süre kullanılmamış ise (idle), container (JBoss) bu bean'i pasifleştirip içeriğini önbelleğe atabilir. Bu işlem olmadan önce `@PrePassivate` ile işaretlenmiş metot çağırılır. Sadece stateful bean için.

@PostActivate

Pasifleştirilmiş bir bean tekrar kullanılıncaya container onu aktif eder. O zaman @PostActivate ile işaretlenmiş metot çağırılır.

@Init

Stateful bean'lerde bean yaratıldıktan sonra çağırılmak üzere kullanılan işarettir. @PostConstruct'tan farklı olarak her bean'de sadece **bir tane** @Init olabilir.

@Remove

Bir EJB'yi elle yoketmek için bir metodu @Remove ile işaretleyebiliriz. Dışarıdan çağırabilmek için aynı metodu `interface` bölümüne de eklemeniz gerekir.

Kullanmak

```
@Stateful
public class xyzBean implements xyz, Serializable
{
    @PrePassivate
    public void xyz()
    {
        ...
    }
    ...
}
```

Mimari Tavsiye

Her kullanıma ilk önce Stateless niyeti ile başlayın.
Mecbur olursanız, Stateful'a geçin.

Görevler

- CounterBean içine biraz önce listelediğimiz her callback için bir fonksiyon eklenecek.
- Test main programı birkaç kez işletilip, callback'lerin ne zaman çağırıldığı görülecek.
- Bean, hem @Stateful hem @Stateless olarak derlenip, testler işletilecek.

Import'lar

```
javax.annotation.PostConstruct;  
javax.annotation.PreDestroy;  
javax.ejb.PrePassivate;  
javax.ejb.PostActivate;  
javax.ejb.Init;  
javax.ejb.Remove;
```