

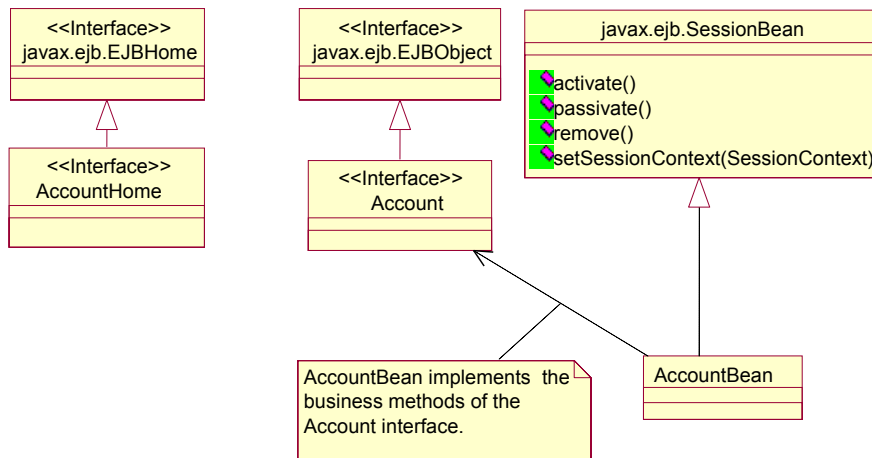
A diagram with a logo on the left consisting of overlapping colored squares (yellow, red, blue) and a black crosshair. To the right of the logo, the text "Session Beans" is written in a blue serif font. A horizontal line extends from the logo area across the width of the diagram. Below this line, there is a bulleted list of three items.

- Modelam o estado não-persistente
 - O estado será perdido entre chamadas de métodos (sessão sem estado) ou chamadas do servidor (sessão com estado)
 - Modelam processos do negócio realizados por um único cliente, envolvendo um ou mais entity bean
- Session beans são perdidos em um crash do servidor

Session Beans

- Estendem as ações do cliente no servidor
 - Simplificam as ações programadas pelo cliente
 - Limitam o número de chamadas distribuídas necessárias entre o cliente e os entity beans
 - Limitam o número de stubs que têm que ser carregados pelo cliente
- Gerenciam ações que podem atravessar vários entity beans ou atingir um escopo externo ao entity bean. Exemplos:
 - Caixa pode autenticar o usuário e transferir fundos entre contas
 - **Statement** pode incluir transações de múltiplas contas

Diagrama de Classe para um Session Bean





Exemplo Session Bean - Home

```
package ejb.bank;
import javax.ejb.*;
import java.rmi.RemoteException;

public interface AccountHome extends EJBHome {
    public Account create() throws CreateException, RemoteException;
}
```



Exemplo Session Bean - Remote

```
package ejb.bank;
import java.rmi.RemoteException;
import javax.ejb.*;

public interface Account extends EJBObject {
    public double balance() throws RemoteException;
}
```



Exemplo Session Bean - Bean

```
package ejb.bank;
import javax.ejb.*;
import java.io.Serializable;
import java.util.*;
import java.rmi.*;

public class AccountBean implements SessionBean {
    private transient SessionContext ctx;

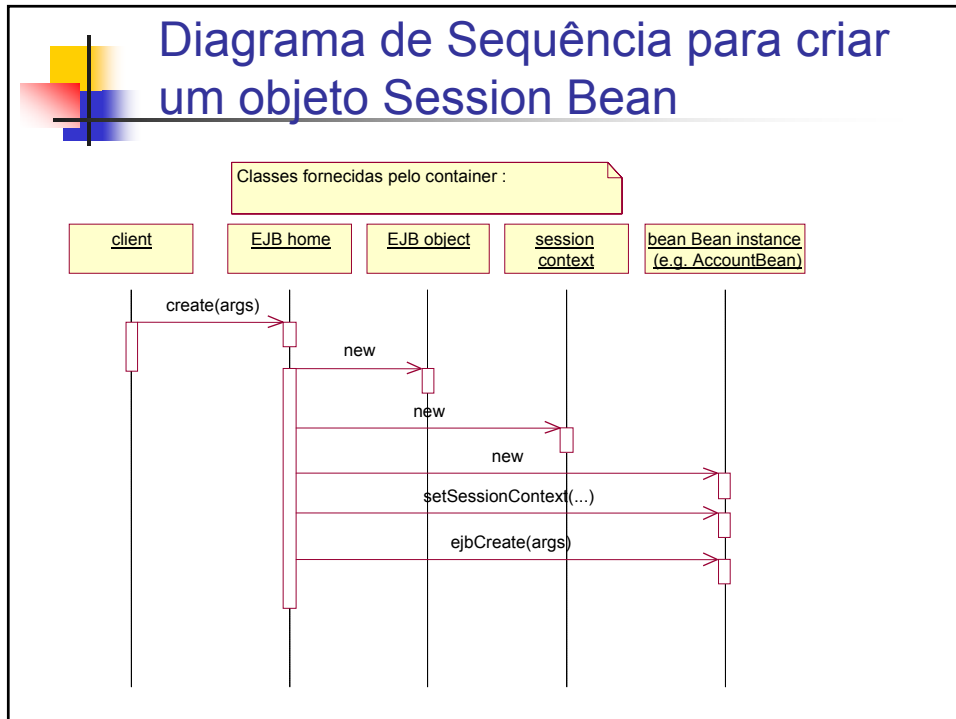
    public void ejbCreate () {}
    public void ejbRemove() {}
    public void setSessionContext(SessionContext ctx) {
        this.ctx = ctx;
    }
}
```



Exemplo Session Bean

```
public void ejbActivate() {}
public void ejbPassivate() {}
/**
 * Lógica de negócio
 */
public String balance() throws RemoteException {
    // lógica de acesso ao BD
    return(value);
}
}
```

Diagrama de Sequência para criar um objeto Session Bean



Session Bean – Primeiro Exemplo

Exemplo de Stateless Session Bean
Hello World



Session Beans com e sem Estado

- Session bean sem estado
 - Não mantém estado conversacional
 - Parece o mesmo para um cliente antes e depois da operação
 - Cada método é independente do outro e a única informação necessária é fornecida nos parâmetros da chamada
 - Consome a menor quantidade de recursos entre todos os tipos de beans



Session Beans com e sem Estado

- Session bean com estado
 - Mantém o estado conversacional entre um cliente e o session bean
 - É semi-persistente
 - Pode ser serializado "fora" e tornado passivo para conservar recursos do sistema ("passivation")
 - Será seralizado "dentro" e ativado quando necessário no futuro
 - Por exemplo:
 - session bean representando um carrinho de compras, utilizado em uma aplicação de um supermercado on-line



Pooling de Objetos

- Containers EJB podem usar pools de objetos
 - Evitar a necessidade de criação de novas instâncias de Beans quando o cliente solicita
 - Flexibilidade de criar instâncias antecipadamente e colocá-las em um pool de objetos prontos
- Quando um cliente invoca um método create () em uma interface home, o container pode obter uma instância do pool, chamar alguns métodos para prepará-la e permitir o seu uso pelo cliente
 - Os métodos invocados dependem do tipo do Bean.
 - Em geral incluem fornecer ao Bean um objeto EJBContext
 - acesso ao ambiente de execução do container (para o bean)
- Quando o cliente chama algum método remove () a instância pode ser recolocada no pool

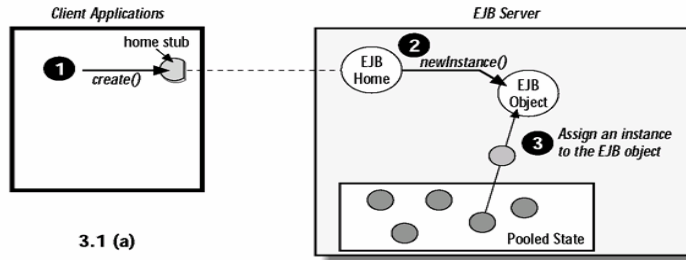


Ativação e “Passivation”

- Processo de “passivation”:
 - Recursos finitos do container podem levar a remoção temporária de objetos Bean para armazenamento secundário
- Processo de ativação:
 - Quando o servidor traz de volta o EJB do armazenamento secundário para a memória
- Somente em Beans que podem possuir estado



Funcionamento do Pooling



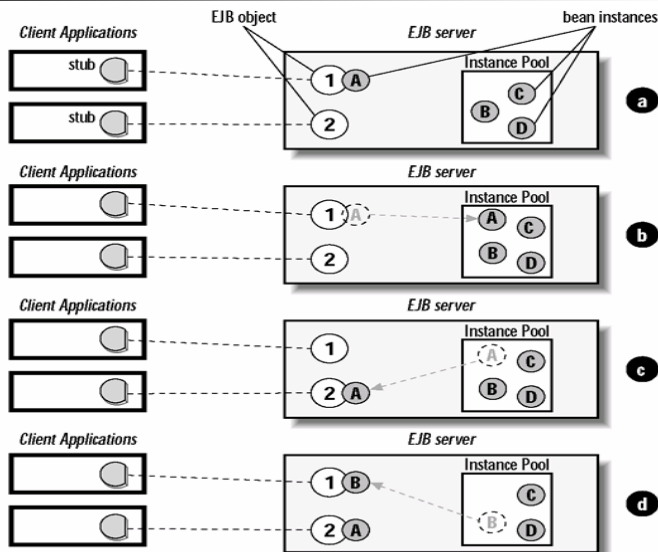
3.1 (a)



3.1 (b)

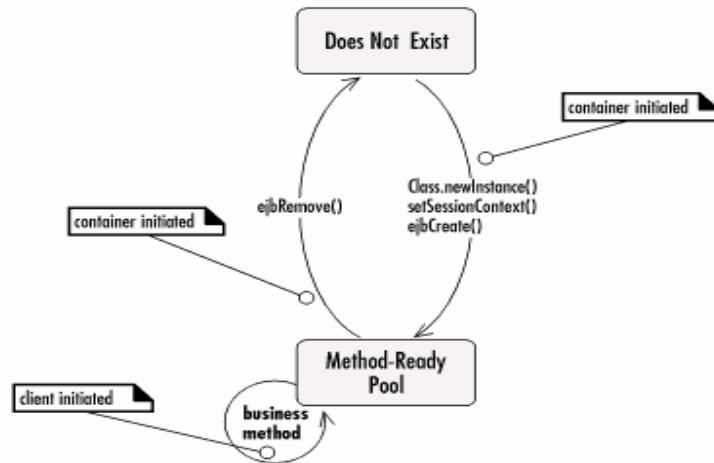


Instance Swapping

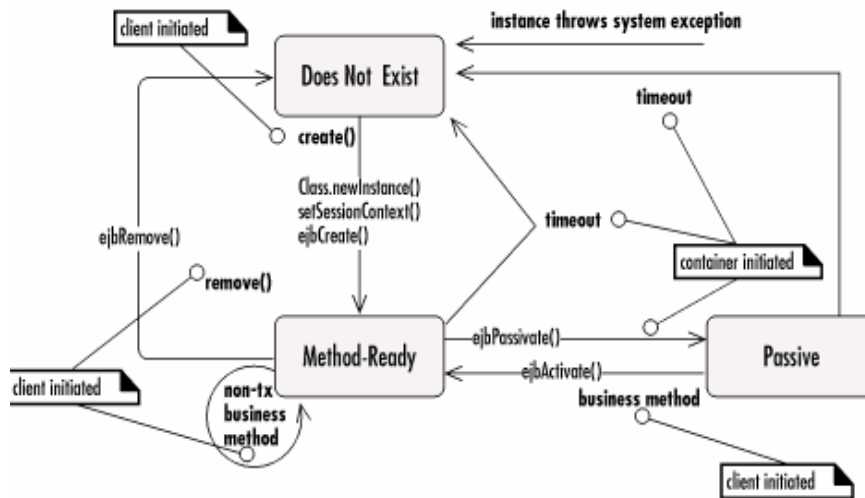




Ciclo de Vida – Session sem Estado



Ciclo de Vida – Session com Estado





Session Bean – Segundo Exemplo

Exemplo de Statefull Session
Bean
Conversor de Temperaturas