



Enterprise Java Beans

Session Bean Stateful

The title 'Session Bean Stateful' is centered at the top. The word 'Session' is partially overlaid by a solid light purple circle. The word 'Bean' is partially overlaid by a hollow light purple circle. The word 'Stateful' is partially overlaid by a solid light purple circle. To the right of the text, there are three more circles: a solid light purple circle, a hollow light purple circle, and another solid light purple circle.

Session Bean Stateful

- Componentes que mantêm o estado dos seus atributos.
- Mantêm os valores de seus atributos entre as várias chamadas de seus métodos de negócio (serviços).

Session Bean Stateful



- Mantém um relacionamento forte com o cliente que o utiliza.
- Se a execução do Bean termina ou se o cliente solicita a remoção da instância para container, a sessão é finalizada e o estado é perdido e numa próxima utilização do Bean os valores serão nulos.

Session Bean Stateful

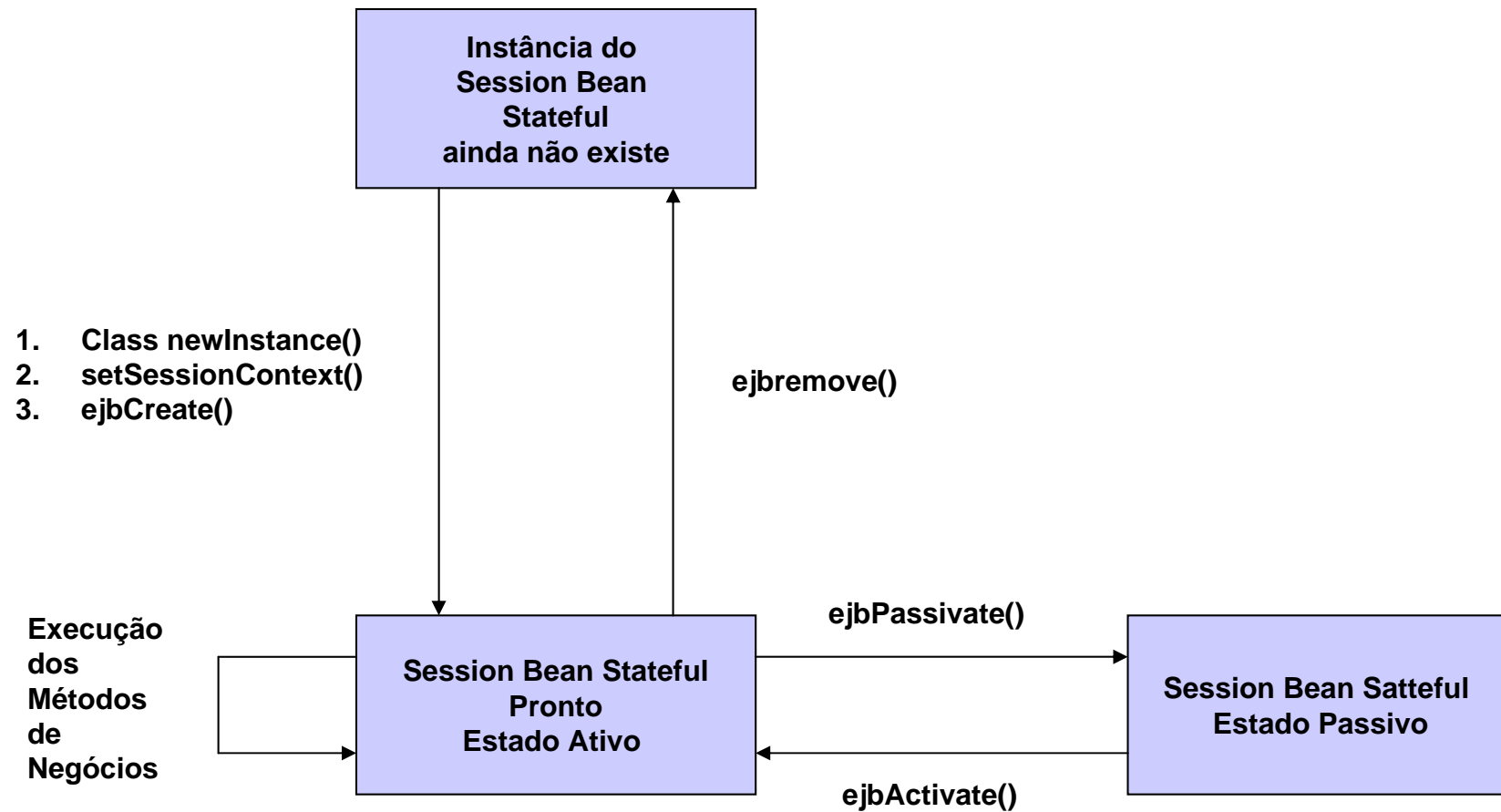


- O cliente pode configurar os valores dos atributos do Bean através dos métodos **set** e obter os valores dos atributos através dos métodos **get**.
- Assim, o Bean pode utilizar esses valores para os métodos de negócio.

Session Bean Stateful

- Ver exemplo escrito.

Session Bean Stateful – Ciclo de Vida



Session Bean Stateful - Estados

- Estado **Ativo** (pronto)
 - É utilizado pelo cliente.
 - O métodos de negócio podem ser executados.
 - Possui recursos alocados pelo container.
- Estado **Passivo**
 - O Bean deixa de ser utilizado por um tempo e fica aguardando novas chamadas de cliente quando está passivo.
 - Os recursos foram liberados pelo container.

Session Bean Stateful



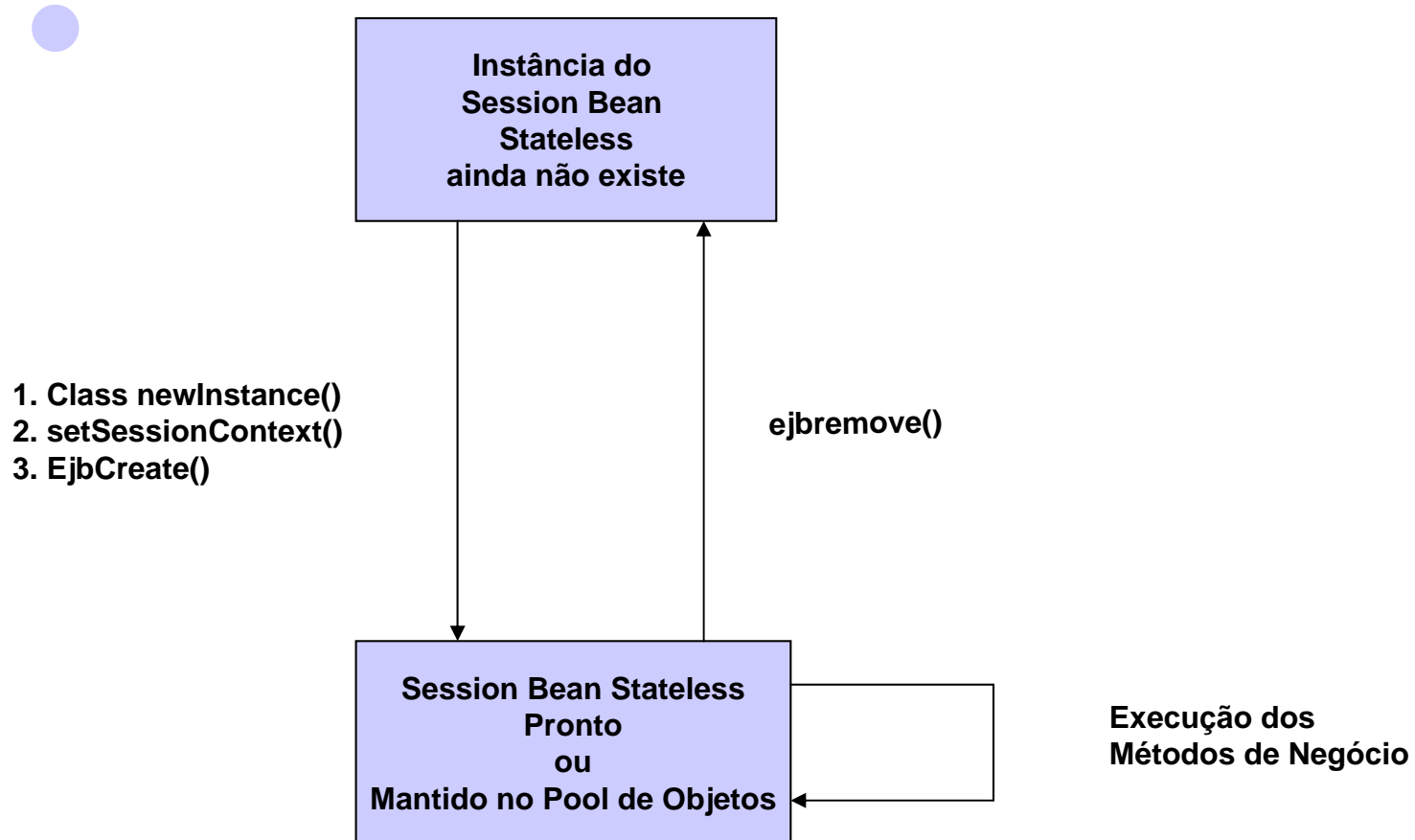
- Note que os métodos

`ejbCreate()`, `ejbPassivate()`,
`ejbActivate()`, `ejbRemove()`,
`ejbSetSessionContext()`

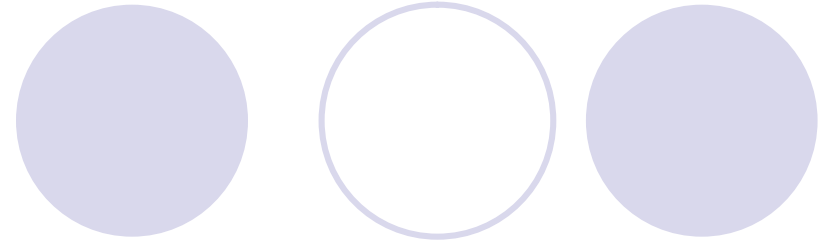
são executados pelo container dentro do ciclo de vida de um Session Bean Stateful.

Session Bean Stateless – Ciclo de Vida

Para você lembrar !



Observações



- Bean significa EJB



Bean - Acesso Remoto

- O acesso remoto é utilizado quando o Bean e o cliente estão em máquinas diferentes (JVMs distintas).
- Se o cliente e o Bean estiverem na mesma máquina (mesma JVM), o acesso remoto será realizado da mesma forma.
- Transparência de Acesso.



Bean - Acesso Remoto

- Devemos implementar as interfaces **Home** e **Remote**.
- A interface **Remote** define os métodos de negócio específicos do Bean.
- A interface **Home** define os métodos do ciclo de vida de um **SessionBean** e também os métodos de busca (create e finders) para um **Entity Beans**.



Bean - Acesso Local

- O cliente e o Bean deve estar na mesma JVM.
- O acesso não é transparente.
- Devemos especificar que o acesso é local.
- É usado para melhorar o desempenho da implementação do negócio.
- O Bean não pode ser acessado remotamente.

Bean - Acesso Local



- Para criar um EJB com acesso local, devemos implementar a interface **Local** e a interface **LocalHome**.
- A interface **Local** define os métodos de negócio específicos do EJB (assim como a interface **Remote**).

Bean - Acesso Local



- A interface **LocalHome** define os métodos do ciclo de vida do Bean (assim como a interface **Home**).
- Para os **Entity Beans** a interface **LocalHome** define também os métodos de busca (finders).

Quando usar um Session Bean

- Quando deseja-se prover serviços, transacionais e seguros, a clientes.
- Um container EJB provê serviços transacionais, seguros e podendo ser com persistência.

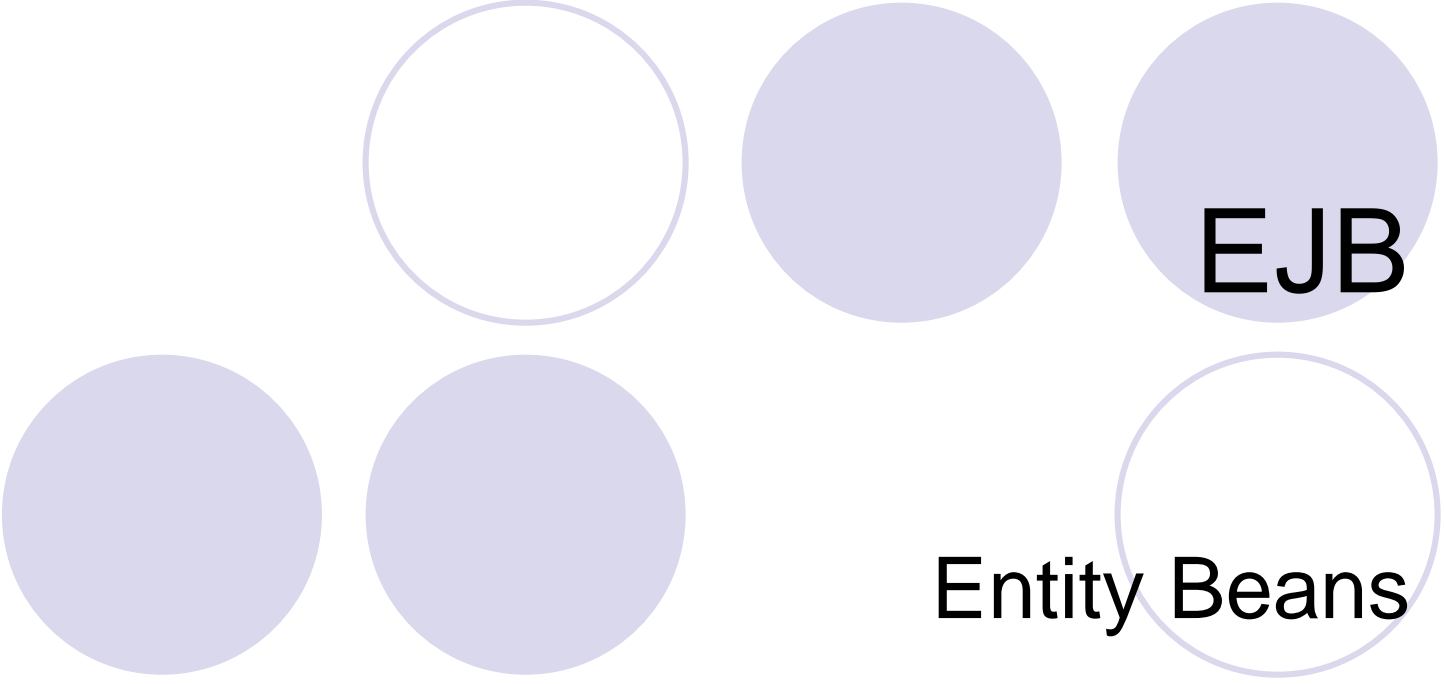


Quando usar um Session Bean Stateless

- Dependendo da aplicação, os Session Beans podem não necessitar guardar os valores dos atributos do Bean, usados na execução de métodos.
- Não estão associados um cliente específico. Implementam comportamento que atendem a necessidade de muitos clientes.
- Exemplo: Validação de um CPF, Consulta de vôos.

Quando usar Session Bean Stateful

- Dependendo da aplicação, os Session Beans podem necessitar guardar os valores dos atributos do Bean, usados durante a execução dos vários métodos de negócio do Bean.
- Exemplo 1: Agente de Viagem que funcione como um proxy para serviços providos por outros Beans.
- Exemplo 2: Um Booking Business Bean que coordena as transações entre vários Beans.





O que são Entity Beans

- São componentes de negócio com mecanismos de persistência de dados.
- O estado pode ser persistido em um banco de dados relacional, arquivos XML ou outros tipos de repositórios de dados.
- O estado é mantido além do tempo de vida da aplicação ou do servidor de aplicação J2EE.



Quando usar um Entity Bean

- Quando seu estado precisa ser persistido, por necessidade da aplicação ou porque o Entity Bean não estava ativo e o servidor foi derrubado. Precisa-se recuperar o estado do Entity Bean.
- Em situações onde deseja-se utilizar os dados do Entity Bean em momentos que seria inviável mantê-los em memória.



Entity Bean - Formas de Persistência

- Persistência gerenciada pelo Entity Bean.
(BMP - Bean Managed Persistence)
- Persistência gerenciada pelo Container.
(CMP – Container Managed Persistence)

BMP - Bean Managed Persistence

- A codificação das chamadas de acesso à base de dados estão na classe que implementa os métodos de negócio do Entity Bean.
- Responsabilidade do desenvolvedor.

BMP - Bean Managed Persistence

- Métodos de **busca** (para encontrar o estado de Entity Beans na base de dados).
- Métodos de **criação, remoção e atualização** de estados persistidos

BMP - Bean Managed Persistence

- Serve para evitar a complexidade de certas operações de persistência, se feitas pelo Container.
- Demanda mais tempo de desenvolvimento.

CMP - Container Managed Persistence

- Mais rapidez.
- Mais facilidade no desenvolvimento de objetos persistentes.
- O desenvolvedor não precisa escrever os comandos para persistir e manipular dados.
- O Container se encarrega de fazer essas operações e retornar os valores desejados.

CMP - Container Managed Persistence

- Ele faz isso para as operações triviais tais como inserção, remoção, atualização e seleção por chave-primária.
- Para operações específicas de consulta a objetos persistidos, se faz necessário o uso de uma linguagem de consulta conhecida por EQL (EJB Query Language).

CMP - Container Managed Persistence

- Essas operações devem ser definidas no arquivo de deploy do componente
(`ejb-jar.xml`)
utilizando EQL para cada operação desejada.
- Neste arquivo define-se os campos que serão persistidos e os relacionamentos entre Entity Beans.