

## Diferenças entre dois tipos de Session Beans

Característica	Stateful Session Bean	Stateless Session Bean
Gerência de estado	<ul style="list-style-type: none"> <li>Gerenciado automaticamente pelo Container (Activation, Passivation)</li> </ul>	<ul style="list-style-type: none"> <li>Não há estado a ser gerenciado</li> <li>Bean pode sair da memória sem salvar qualquer coisa (não há "Passivation")</li> <li>Beans são intercambiáveis e podem ser alocados de um "pool"</li> </ul>
Responsabilidades	<ul style="list-style-type: none"> <li>Estado conversacional pode ser mantido no Bean</li> </ul>	<ul style="list-style-type: none"> <li>O cliente tem que manter qualquer estado necessário</li> </ul>
Performance	<ul style="list-style-type: none"> <li>Mais pesado</li> </ul>	<ul style="list-style-type: none"> <li>Extremamente lightweight</li> </ul>
Exemplo	<ul style="list-style-type: none"> <li>Shopping Cart</li> </ul>	<ul style="list-style-type: none"> <li>Verificação de crédito (Todo o trabalho pode ser feito numa única chamada de método)</li> <li>Mailer Bean para enviar mail confirmando uma compra</li> <li>Bean que valida um ID de empregado</li> <li>Acesso a um Catálogo de Produtos. Embora pareça melhor usar um Entity bean aqui, devido à persistência, lembre que o catálogo não está associado a um cliente particular e que nenhum estado sobre o cliente deve ser mantido. Porém, um estado geral, não específico a um cliente particular pode ser mantido. Podemos usar Stateless Beans para montar uma cache do catálogo à medida que este é acessado. O Bean provê browsing e searching no catálogo. Como o Bean poderá manipular várias linhas do catálogo ao mesmo tempo (no search, p. ex.) e oferecer uma visão compartilhada de informação, o Bean poderá ser compartilhado.</li> </ul>