EJBs

EJBs

O que são?!

EJBs

Componentes corporativos que executam no lado servidor,
Podendo ser disparados local ou remotamente com todo seu ciclo de vida gerenciado por um container EJB e que implementam regras de negócio, colaborando entre si para entregar valor para o usuário final.

EJBs

Porque usá-los?

- Fácil implementação (POJOs)
- Catálogo de serviços operacionais
- Foco no negócio

EJBs

Quando usar?

Quando a aplicação demanda...

- Escalabilidade
- Segurança
- Controle transacional distribuído e consistente
- E todos os serviços que um container geralmente provê.

EJBs

Tipos de Beans

EJBs

Session Beans

Executam uma ação específica para o cliente.

EJBs

Message Driven Beans (MDBs)

Permitem o processamento assíncrono de mensagens pela aplicação.

Session Beans

Session Beans

Encapsulam os procedimentos e regras de negócio que são executados durante uma sessão

Session Beans

Podem guardar, se necessário, o estado conversacional entre cliente e servidor

Session Beans

Gerenciados pelo container

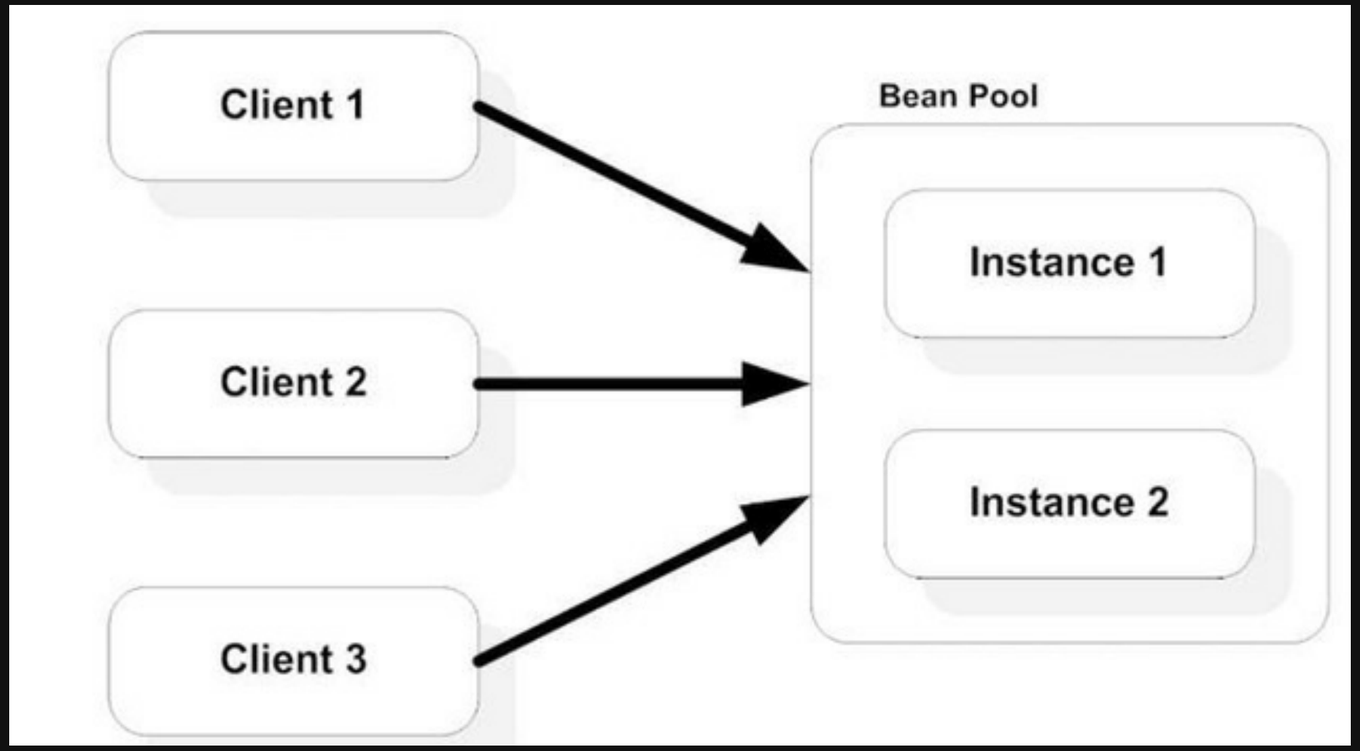
Session Beans

Não são objetos persistentes

Stateless Session Beans

- **Executa procedimento específico sem manter o estado entre chamadas**
  - Ex.: verificar estoque, emprestar livro, debitar conta
- **Thread-safe**
- Podem ser associados a qualquer cliente
- **Anotações**
  - @Stateless
  - @EJB

Stateless Session Beans



Stateless Session Beans

Exemplo

Stateless Session Beans

Servlet

```
@WebServlet("/ReservaMercadoriaServlet")
public class ReservaMercadoriaServlet extends HttpServlet {
    @EJB ControladorSSB controle;
protected void doGet(...) {
    String merc = request.getParameter(...);
    int reservado = controle.obterQuantidade(merc);
    ...
}
```

Stateless Session Bean

```
@Stateless
public class ControladorSSB {
    @Inject private Estoque meuEstoque;
    public int obterQuantidade(String merc) {
        int qtdeEstoque = meuEstoque.obterQuantidade(merc);
        ...
    } }
```

Exercício 2

Atualize o projeto distribuidora para executar uma ação de reserva em um Estoque em memória utilizando EJBs.

Dicas

1. Não será necessário criar projeto ejbclient, nem interfaces no momento
2. No projeto EJB crie uma classe do tipo Estoque que possa ser injetada no EJB. Essa classe terá os dados do estoque
3. Use os parametros do request para indicar a mercadoria e quantidade a ser reservada
4. Utilize as anotações características do bean: @Stateless para o Session e @EJB para injeção no servlet