Links

JPA E HIBERNATE

Integração entre CDI e EJB

EJBS

@Stateless

Essa anotação define um **EJB sem estado** (Stateless Session Bean). Um Stateless EJB não mantém informações de estado entre as chamadas de métodos, ou seja, cada requisição é tratada de forma independente das outras.

Detalhes:

- Usado quando o EJB não precisa manter o estado do cliente entre invocações.
- Reútil por múltiplos clientes.
- Ideal para operações que não dependem de dados anteriores, como cálculos ou consultas que não dependem de sessões de usuário.

@Stateful

Essa anotação define um **EJB com estado** (Stateful Session Bean). Um Stateful EJB mantém informações de estado entre as interações com o cliente. O estado é mantido durante toda a sessão do cliente.

Detalhes:

- Usado quando o EJB precisa manter dados específicos entre chamadas consecutivas.
- Um exemplo comum seria um carrinho de compras, onde o estado (itens no carrinho) deve ser mantido durante várias interações.

@Singleton

Define um **EJB Singleton**, ou seja, uma única instância do bean é compartilhada por todos os clientes. É útil para componentes que precisam compartilhar estados entre várias instâncias e sessões de clientes.

Detalhes:

- Usado para gerenciar recursos compartilhados, como cache de dados ou contadores globais.

- O ciclo de vida é controlado pelo contêiner, garantindo que apenas uma instância do bean esteja disponível em toda a aplicação.

@Schedule:

- Define métodos que serão executados automaticamente em horários ou intervalos de tempo específicos. É usado para agendar tarefas repetitivas no EJB.

@Asynchronous:

- Indica que o método será executado de maneira assíncrona. O cliente não precisa esperar a conclusão do método, que será processado em segundo plano.

@Timeout:

- Usada para definir um método que será executado quando um timer programado expirar. Normalmente é usada com a API de temporizadores (TimerService) do EJB.

@TransactionAttribute:

- Define o comportamento transacional de métodos em um bean EJB. Pode ser usada para especificar como o método lida com transações, com valores como REQUIRED, REQUIRES_NEW, MANDATORY, NOT_SUPPORTED, SUPPORTS e NEVER.

@TransactionManagement:

- Especifica se o bean EJB usará **transações gerenciadas pelo contêiner** (TransactionManagementType.CONTAINER) ou **transações gerenciadas pelo próprio bean** (TransactionManagementType.BEAN).

@Lock:

- Usada em **EJB Singleton** para controlar o acesso concorrente aos métodos do bean. Os modos de bloqueio disponíveis são READ (vários clientes podem ler simultaneamente) e WRITE (acesso exclusivo para gravação).

@AccessTimeout:

- Especifica quanto tempo um cliente pode esperar para adquirir um bloqueio em um método de um **Singleton EJB**.

@Startup:

- Usada com **Singleton EJBs** para indicar que o bean deve ser instanciado logo no início da aplicação (startup), e não sob demanda.

@Remote:

- Define que a interface do EJB pode ser acessada remotamente, ou seja, por clientes que estão em uma máquina diferente da que está executando o EJB.

Local:

- Define que a interface do EJB só pode ser acessada localmente, ou seja, dentro da mesma JVM.

JPA E HIBERNATE

Jpa é um conjunto de interfaces e anotações que foram documentadas pela java, pode ser usado através de vários provedores como (hibernate, eclipselink) – Em teoria, podemos facilmente trocar de hibernate po eclipselink, pois ambos implementam as interfaces JPA.

Active Record

Definição: O padrão Active Record combina os dados e a lógica de persistência em uma única classe. Cada objeto da classe representa uma linha de uma tabela no banco de dados. Os métodos da classe são responsáveis por carregar, salvar, atualizar e excluir os dados.

Data Mapper

Definição: O padrão Data Mapper separa a lógica de persistência dos objetos de domínio. As classes de modelo representam a lógica de negócios, enquanto os mapeadores são responsáveis por transferir dados entre os objetos e o banco de dados.

@Named

@ManageBean

Complementar ao @ManagedBean: Antes do CDI, o JSF usava @ManagedBean para expor beans na camada de visão. @Named é a alternativa moderna e recomendada, em conjunto com CDI, trazendo mais flexibilidade e controle sobre o ciclo de vida e escopos.

Integração entre CDI e EJB

Embora CDI e EJB sejam especificações separadas, **eles podem ser usados juntos**. Um bean CDI anotado com @Named pode injetar um EJB para usufruir de funcionalidades como transações, por exemplo:

EJB fornece serviços de back-end como transações, agendamento de tarefas e controle de concorrência