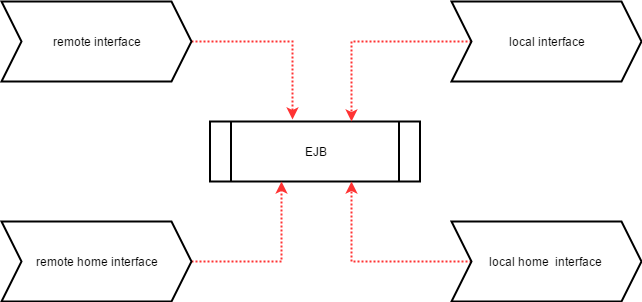
EJB的底层是远程方法调用,本质上的目的是把业务逻辑放到服务器上,然后客户端远程调用服务器上的方法来工作,放在服务器上的那些java类就叫EJB,

EJB容器是一个中间件,他提供事务处理、安全、并发、网络、资源管理、持久存储管理、消息传递、部署时定制等领域无关、业务无关工作.最核心的表现是,他让封装了客户端对远程方法的调用细节.

EJB有三种:

1. 会话bean:实现业务逻辑.分为有状态和无状态,有状态的bean在被客户端调用后能保存会话状态,这使得一个会话bean只能被一个客户端使用,无状态的会话bean不保存会话状态,因此实例之间没有区别.无状态会话bean使用了实例池,有状态会话bean有钝化和激活这种内存管理方法.
2. 实体bean:管理持久化数据的一个对象,会话bean一般通过实体bean与数据库进行交互,他是数据库中信息的对象化表示
3. 消息驱动bean:客户端和服务器之间可以通过JMS机制进行异步消息传送,每个消息驱动bean守候一种消息,实现MessageListener接口中的OnMessage()方法,一旦守候的消息到达,该方法就会被调用.消息驱动bean使用实例池,可以处理大量的JMS消息.



一个EJB实现上述四个接口,这些接口并不需要程序员手动实现其中的接口方法,也不需要在代码里面显式给出继承,他们的实现是由容器进行的.不是所有的bean都必须实现远程接口,例如实体bean可以只有local接口,客户端对实体bean的访问通过会话bean代理.

Remote interface:客户端用于使用远程bean上的业务逻辑的接口,其中的接口方法和bean中的业务逻辑方法一一对应.

Remote home interface:用于控制Bean的生命周期和获取实例,其中有Create,Find,Remove函数,其中Create函数与Bean中的ejbCreate函数相对应,可以重载多个.

Local interface:用来在同一个JVM上的bean之间互相调用,绕开了远程方法调用的网络通信.

Bean的生命周期由容器托管,在周期发生切换时会发生一些”事件”,bean类可以使用对应的回调函数捕获这些事件从而在生命周期发生切换时做一些操作,要注意回调函数不是管理生命周期的,而是在生命周期发生变化时被容器触发的.

不同的bean的生命周期不同: <https://blog.csdn.net/zmh458/article/details/79077726>

EJB部署描述文件