参考：<http://blog.csdn.net/sean_cd/article/details/8725159>

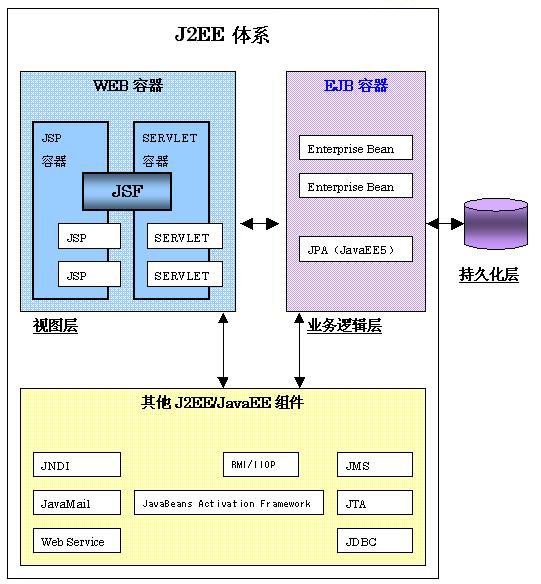
[EJB的体系结构](http://blog.csdn.net/sean_cd/article/details/8725159)

本文对EJB的体系结构作一个全面介绍：

* EJB在J2EE体系结构中的位置
* EJB的环境构成
* EJB的资源配置
* EJB的种类
* EJB组件的构造
* EJB的调用过程
* EJB组件的生命周期

EJB是Enterprise JavaBeans的简称。本文对EJB技术的体系结构作一个全面深入地介绍。

**EJB在J2EE体系结构中的位置**



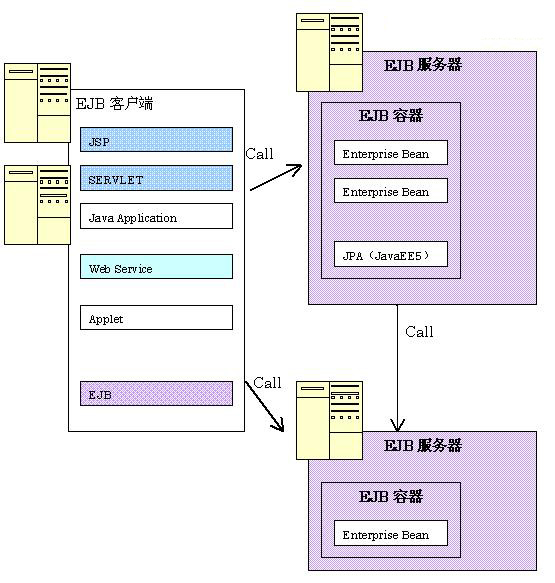
视图层或表现层：提供用户界面；接收用户输入；数据输出。WEB应用的情况，JSP以及SERVLET属于视图层组件。

业务逻辑层：实际的业务逻辑处理。根据视图层传送过来的数据，进行实际的业务逻辑处理（包括数据库的查询，更新等），再把处理后的结果返回给视图层。EJB，以及不使用EJB的情况下担当业务逻辑处理部分的JavaBean等归属于业务逻辑层组件。

数据库层或数据持久化层：多指用于保存业务数据的数据库，也可以是文件等等。

EJB属于J2EE体系结构中的业务逻辑层部分。

**EJB的环境构成（就是容器包着beans，是一个提供业务服务的组件）**



1，EJB组件运行在EJB容器之中，EJB容器是一个EJB引擎，它提供了EJB组件运行的环境，并对EJB组件进行管理。EJB容器一般包含在EJB服务器（或应用服务器）中，EJB服务器可以拥有一到多个EJB容器。比较有名的支持EJB的服务器有Sun One，Interstage，Websphere，Weblogic，JBoss，JRun等。

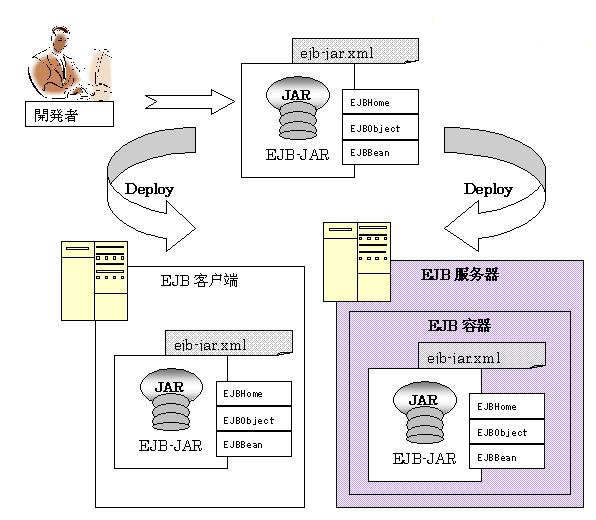
2，调用EJB组件的一方被称为EJB客户端。EJB客户端可以为运行在WEB容器中的JSP，SERVLET；或者一般的Java Application，Applet；或者Web Service；也可以是别的EJB组件。

3，EJB客户端与EJB服务器（me：ejb现在的主要使用是在分布式中，这时客户端和服务端多在不同的机器上；本来在ejb出现的时候，就是将客户端一些业务逻辑放在服务器端，然后客户端请求服务端展现这些结果）可处于同一JVM环境中，也可处于不同计算机的不同JVM环境。

**EJB的资源配置**

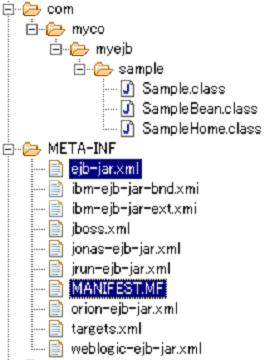
当一个EJB组件被开发完成之后，它需要以JAR文件的形式配置到EJB容器中以备其他客户程序使用；反过来，一个客户程序需要使用某EJB组件，该客户程序需要引用该EJB组件的接口，也就是说，EJB的jar文件需要同时配置到EJB容器和使用该EJB组件的客户端环境中。

如图：



EJB-JAR文件的构成

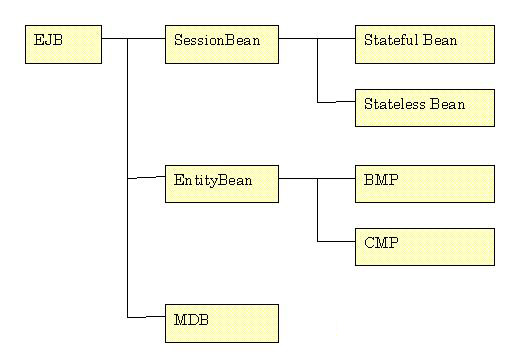
sample-ejb.jar



EJB-JAR文件是一个后缀名为.jar的zip格式文件。它至少需要包含META-INF目录以及其下的ejb-jar.xml文件以及MANIFEST.MF文件。对于不同类型的EJB组件，还需包含它们的Home接口，组件接口，以及组件的实现类。

**EJB组件的种类**

EJB组件有SessionBean（会话Bean），EntityBean（实体Bean），MDB（消息驱动Bean）三种。



SessionBean

主要负责业务逻辑的处理。根据处理时的状态保持与否，SessionBean又被分为：

Stateful SessionBean（有状态SessionBean）

Stateless Bean（无状态SessionBean）

对象的“状态”是指对象的属性值，也就是对象所携带的数据。Stateful SessionBean能维持会话的状态，就是说能为同一客户端保持上一回被调用时的状态，类似于HttpSession；而Stateless SessionBean则不能维持会话状态，也就是说，对Stateless SessionBean来说，每个方法都是独立的，因为对特定客户端不保持会话状态，所以一般也不在Stateless SessionBean里定义成员变量。

EntityBean

负责数据库的访问。通常由SessionBean调用。EntityBean的一个实例所表示的数据通常代表了关系数据库中特定表的特定的一行数据。EJB3开始，EntityBean部分的功能可以通过JPA来实现。

EntityBean的作用：

1，对数据库的操作：查询，数据的更新，删除等

2，数据库表数据的携带

根据对数据库操作（持久化）的不同，EntityBean分为以下2种类型：

1，BMP (Bean-managed persistence：bean管理持久化)：持久化操作（数据库的访问，数据的创建，更新，删除等）由EntityBean的实现（javax.ejb.EntityBean的实现类）自身完成。

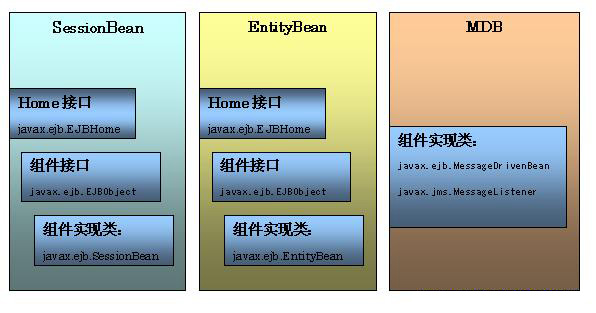
2，CMP (Container-managed persistence：容器管理持久化)：持久化操作由容器根据deploy时的配置信息（ejb-jar.xml）自动完成。

MDB (message-driven bean：消息驱动bean)

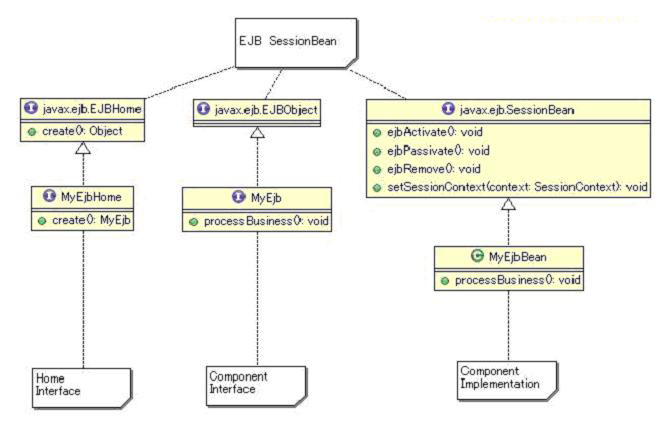
MDB主要用来处理异步消息。客户端调用SessionBean，EntityBean的方法时，会一直处于等待状态，直到被调用的方法处理完毕。但使用MDB，客户端向容器发送一个JMS消息之后，不必等待MDB处理完毕便可直接返回。JMS被发送给由容器管理的JMS消息队列，容器在适当的时候通知MDB的callback方法onMessage加以处理。一个MDB需要实现javax.jms.MessageListener接口与javax.ejb.MessageDrivenBean接口。

**EJB组件的构造**

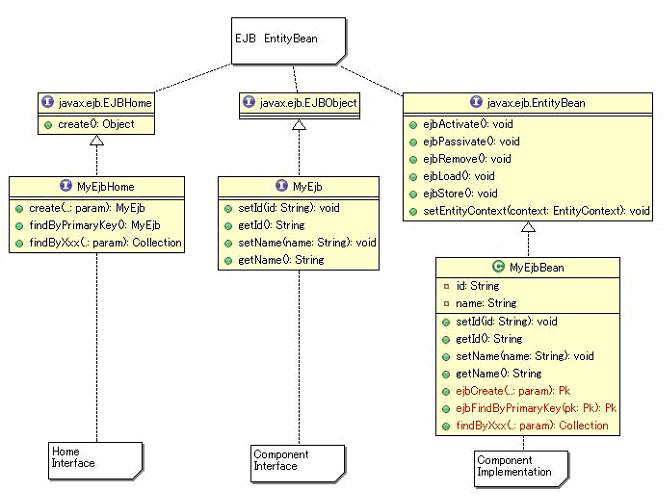
构造图：



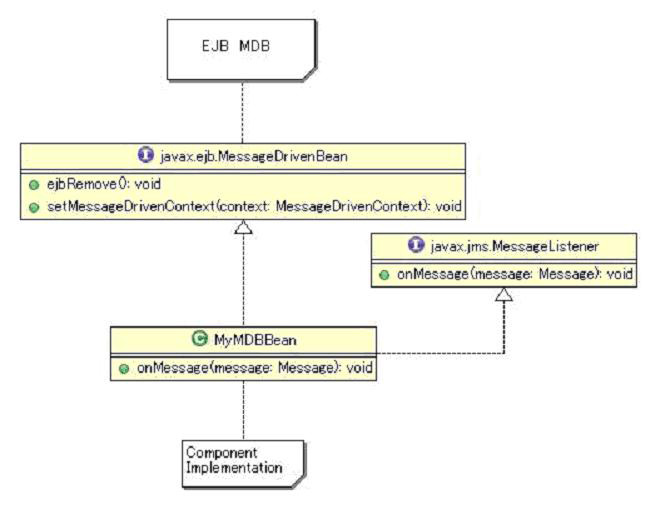
SessionBean的UML-CLASS图



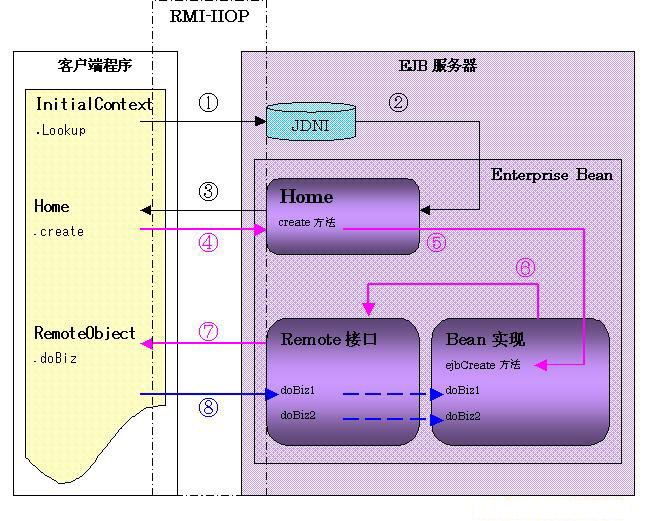
EntityBean的UML-CLASS图



MDB的UML-CLASS图



**EJB的调用过程**



①客户端程序使用lookup方法查找JNDI

②EJB服务器的JDNI服务根据事先登录的信息匹配lookup查询内容，生成Home实例

③客户端得到对Home实例的引用

④客户端调用Home.create方法

⑤创建相对应的Bean实例，调用Bean的ejbCreate方法

⑥生成相对应的Remote Ojbect实例

⑦客户端得到对Remote Ojbect实例的引用

⑧客户端调用业务逻辑方法

客户端与服务器端的通信使用RMI-IIOP通信协议。

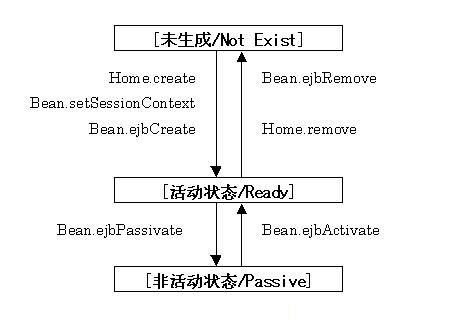
客户端在调用EJB组件时，需要知道EJB组件的位置（包括IP，端口，名称），才能得到对EJB组件对象的引用。JNDI是J2EE提供的一套Java API，是Java Naming and Directory Interface的略称，提供根据名称查找对象的服务。EJB利用JNDI来解决名称问题。JNDI有Service Provider的概念，Provider可以选择LDAP或者RMI。

要使用JNDI，需要先进行登录，然后使用lookup进行查找。

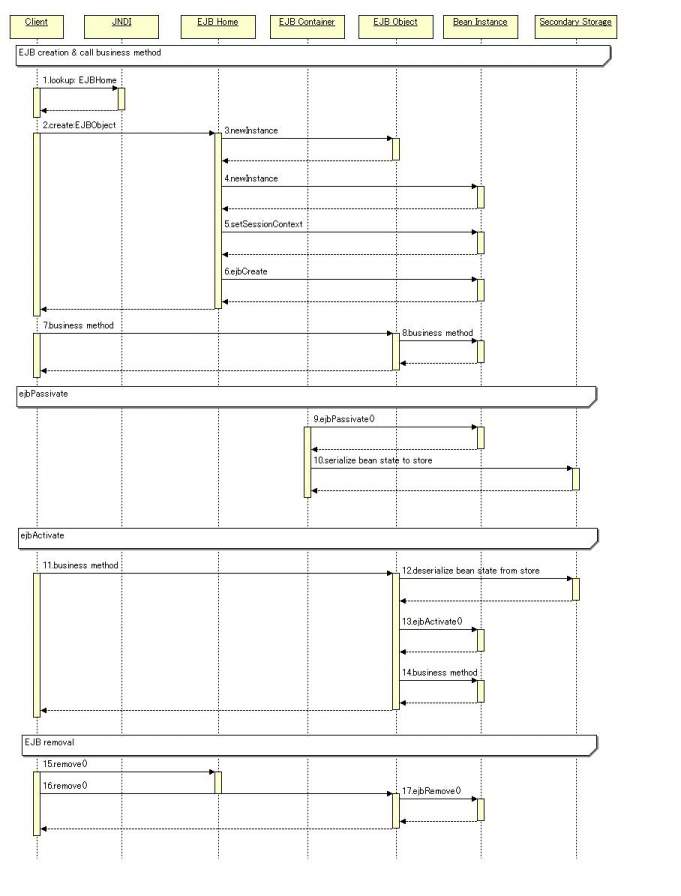
EJB的JNDI的使用，也需要通过配置文件进行登录，不同的EJB服务器有不同的设置方法。具体可参考相关EJB服务器的操作手册。

**EJB组件的生命周期（Lifecycle）**

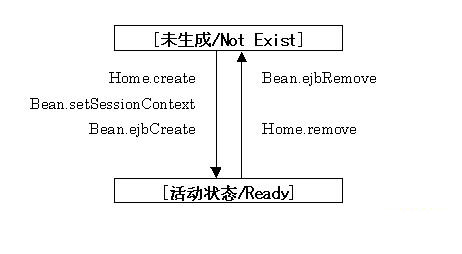
Stateful SessionBean的生命周期



顺序图（sequence图）



Stateless SessionBean的生命周期



顺序图（sequence图）

