## Java Message Service

JMS即Java消息服务（Java Message Service）应用程序接口，是一个Java平台中关于面向消息中间件（MOM）的API，用于在两个应用程序之间，或分布式系统中发送消息，进行异步通信。Java消息服务是一个与具体平台无关的API，绝大多数MOM提供商都对JMS提供支持。

JMS允许应用程序组件基于JavaEE平台创建、发送、接收和读取消息。它使分布式通信耦合度更低，消息服务更加可靠以及异步性。

## 在提到JMS时，我们通常会说到一些术语，解释如下：

1. **消息中间件**（JMS Provider) ： 指提供了对JMS协议的第三方组件，比如ActiveMQ就是一个消息中间件，另外比较知名的还有KFA, Rabbit MQ等。
2. **消息模式**：分为点对点（Point to Point，即P2P）和发布/订阅（Pub/Sub)，对应的数据结构分别是队列（Queue)和主题（Topic)
3. 消息（Message): 通信内容的载体，其结构主要分为消息头，属性和消息体，并且根据存储结构的不同分为好几种，后面会详细提到。
4. 消息生产者：产生消息的一方，在P2P模式下，指消息发送者(Sender)，在P/S模式下指消息发布者(Publisher)
5. 消息消费者：接收消息的一方，对应于两种模式分别是消息接收者（Receiver）和消息订阅者(Subscriber)

## JMS基本概念及原理详解

一、基本概念  
　　JMS是java的消息服务，JMS的客户端之间可以通过JMS服务进行异步的消息传输。  
二、体系架构

　　JMS由以下元素组成。

　　1.JMS提供者   
　　连接面向消息中间件的，JMS接口的一个实现。提供者可以是Java平台的JMS实现，也可以是非Java平台的面向消息中间件的适配器。

　　2.JMS客户   
　　生产或消费消息的基于Java的应用程序或对象。

　　3.JMS生产者   
　　创建并发送消息的JMS客户。

　　4.JMS消费者   
　　接收消息的JMS客户。

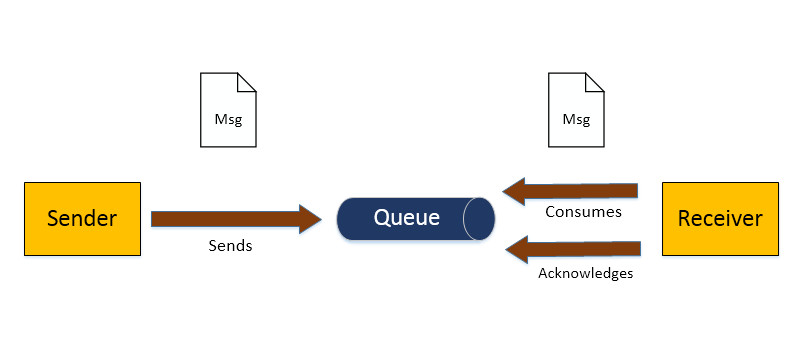
　　5.JMS消息   
　　包括可以在JMS客户之间传递的数据的对象

　　6.JMS队列   
　　一个容纳那些被发送的等待阅读的消息的区域。队列暗示，这些消息将按照顺序发送。一旦一个消息被阅读，该消息将被从队列中移走。

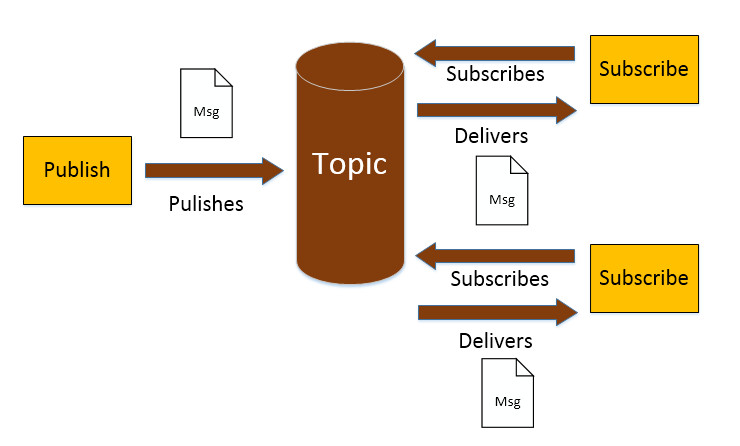
　　7.JMS主题   
　　一种支持发送消息给多个订阅者的机制。

3、JMS消息模型(即点对点和发布订阅模型)

　　　　1、Point-to-Point(P2P)



　　　　2、Publish/Subscribe(Pub/Sub)



## JMS应用程序接口

7.1、ConnectionFactory 接口(连接工厂)  
　　　　创建Connection对象的工厂，根据消息类型的不同，用户将使用队列连接工厂，或者主题连接工厂分，  
　　　　别有QueueConnectionFactory和TopicConnectionFactory两种。可以通过JNDI来查找ConnectionFactory对象。  
　　  
　　7.2、Destination 接口(目标)  
　　　　Destination是一个包装了消息目标标识符的被管对象，消息目标是指消息发布和接收的地点，或者是队列，或者是主题。  
　　　　是消息生产者的消息发送目标或者说消息消费者的消息来源。对于消息生产者来说，它的Destination是某个队列（Queue）或某个主题（Topic）;  
　　　　对于消息消费者来说，它的Destination也是某个队列或主题（即消息来源）。  
　　　　所以，Destination实际上就是两种类型的对象：Queue、Topic可以通过JNDI来查找Destination。

　　7.3、Connection 接口(连接)  
　　　　Connection表示在客户端和JMS系统之间建立的链接（对TCP/IP socket的包装）。  
　　　　Connection可以产生一个或多个Session。跟ConnectionFactory一样，Connection也有两种类型：QueueConnection和TopicConnection。  
  
　　7.4、Session 接口(会话)  
　　　　Session是我们操作消息的接口。表示一个单线程的上下文，用于发送和接收消息。  
　　　　由于会话是单线程的，所以消息是连续的，就是说消息是按照发送的顺序一个一个接收的。  
　　　　可以通过session创建生产者、消费者、消息等。Session提供了事务的功能。当我们需要使用session发送/接收多个消息时，可以将这些发送/接收动作放到一个事务中。  
　　　　同样，也分QueueSession和TopicSession。  
  
　　7.5、MessageProducer 接口(消息的生产者)  
　　　　消息生产者由Session创建，并用于将消息发送到Destination。消费者可以同步地（阻塞模式），或异步（非阻塞）接收队列和主题类型的消息。  
　　　　同样，消息生产者分两种类型：QueueSender和TopicPublisher。可以调用消息生产者的方法（send或publish方法）发送消息。

　　7.6、MessageConsumer 接口(消息消费者)  
　　　　消息消费者由Session创建，用于接收被发送到Destination的消息。两种类型：QueueReceiver和TopicSubscriber。  
　　　　可分别通过session的createReceiver(Queue)或createSubscriber(Topic)来创建。  
　　　　当然，也可以session的creatDurableSubscriber方法来创建持久化的订阅者。

　　7.7、Message 接口（消息）  
　　　　是在消费者和生产者之间传送的对象，也就是说从一个应用程序创送到另一个应用程序。一个消息有三个主要部分：  
　　　　1、消息头（必须）：包含用于识别和为消息寻找路由的操作设置。  
　　　　2、一组消息属性（可选）：包含额外的属性，支持其他提供者和用户的兼容。可以创建定制的字段和过滤器（消息选择器）。  
　　　　3、一个消息体（可选）：允许用户创建五种类型的消息（文本消息，映射消息，字节消息，流消息和对象消息）。消息接口非常灵活，并提供了许多方式来定制消息的内容。

　　　　消息接口非常灵活，并提供了许多方式来定制消息的内容。

　　7.8、MessageListener  
　　　　消息监听器。如果注册了消息监听器，一旦消息到达，将自动调用监听器的onMessage方法。  
　　　　EJB中的MDB（Message-Driven Bean）就是一种MessageListener。