**What is JMS?**

Java Message Service (JMS) is an application program interface (API) from Sun Microsystems that supports the formal communication known as [messaging](https://whatis.techtarget.com/definition/messaging) between computers in a network.Java message service enables loosely coupled communication between two or more systems. It provides reliable and asynchronous form of communication.

**Need of JMS**

In Java, if a person wants to send a message from one application to another in such a way that both application do not know anything about each other, even they may be deployed in separate continents with no dependency at all. For example, one application A is running in India and another application is running in USA, and B is interested in getting some updates/messages from A – whenever something unique happen on A. There may be N number of such applications who are interested in such updates from A.

In this scenario, java provides it’s best solution in form of JMS – and solve the exactly same problem discussed above.

**Benefits of JMS**

* **Asynchronous**

JMS is asynchronous by default. So to receive a message, the client is not required to send the request. The message will arrive automatically to the client as they become available.

* **Reliable**

JMS provides the facility of assurance that the message will delivered once and only once. You know that duplicate messages create problems. JMS helps you avoiding such problems.

**Message Consumption**

* **Synchronous**

In synchronous message consumption, the subscriber/receiver request the message from the destination by calling the [receive()](https://docs.oracle.com/javaee/7/api/javax/jms/MessageConsumer.html#receive--) method. In the receive() method will block till the message arrives or time out if the message does not arrive within a given time. Just like normal java method calls with some return value.

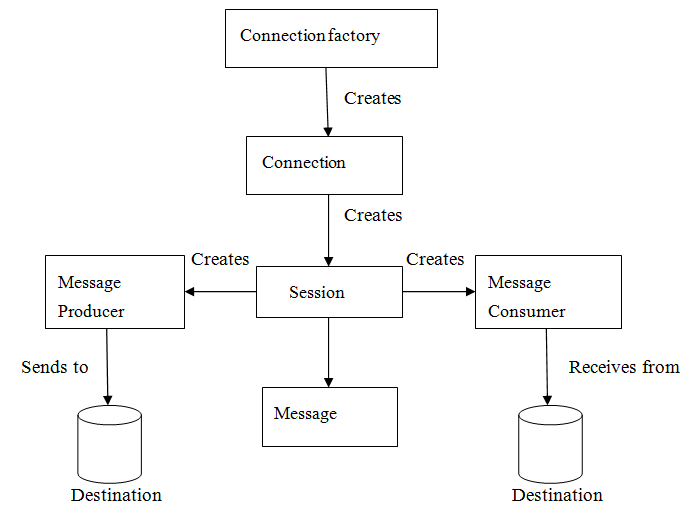
* **Asynchronous**

In asynchronous message consumption, a subscriber can register (or subscribe) as message listener with the consumer. The message listener is same as event listener, whenever the message arrives at the destination the JMS provider will deliver the message by calling the listener’s [onMessage()](https://docs.oracle.com/javaee/7/api/javax/jms/MessageListener.html" \l "onMessage-javax.jms.Message-" \t "_blank) method which will act on the content of the message.

## JMS participating objects

1. Administered objects – Connection Factories and Destination
2. Connections
3. Sessions
4. Message Producers
5. Message Consumers
6. Message Listeners

Reference: <https://howtodoinjava.com/jms/jms-java-message-service-tutorial/>



**JMS providers:**

1. ActiveMQ
2. RabbitMQ
3. IBM MQ
4. Kafka and etc.,