

JMS

cuu duong than cong . com

Nguyễn Văn Khiết

cuu duong than cong . com

Nội dung

- Giới thiệu Java Message Service
- Kiến trúc JMS
- Các đối tượng JMS

Giới thiệu Java Message Service

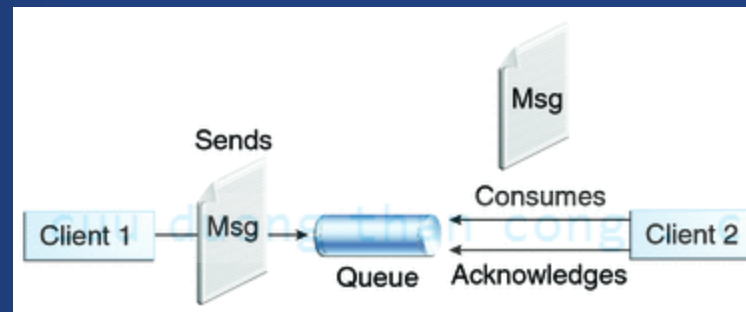
- Java Message Service là một API hỗ trợ giao tiếp dưới hình thức message giữa các máy tính.
- Một cơ chế trao đổi thông tin giữa các hệ thống với nhau, hoặc giữa các module với nhau.
- Cơ chế trao đổi thông tin “loosely coupled”
 - Các hệ thống trao đổi thông tin với nhau
 - Bên gửi không cần biết nhiều về bên nhận, và ngược lại. Hai bên chỉ cần thống nhất định dạng message
 - Bên nhận không cần thiết cùng ở tình trạng sẵn sàng.
 - Có cơ chế đảm bảo không thất lạc message

JMS API

- Java API định nghĩa tập các interface cho phép các chương trình java giao tiếp với các hệ thống truyền/nhận thông điệp
- Định nghĩa cơ chế chung để các ứng dụng tạo, gửi, nhận thông điệp:
 - Bất đồng bộ
 - Tin cậy

Kiến trúc JMS

- Producer/Consumer



- Publisher/Subscriber



Kiến trúc JMS

- Các thành phần JMS
 - JMS provider
 - JMS client
 - Message
 - Administered Objects.

cuu duong than cong . com

Kiến trúc JMS

- Các thành phần JMS
 - JMS provider:
 - Hệ thống implements các interface của JMS
 - Cung cấp cơ chế quản trị message.
 - Phải chọn một JMS provider khi bắt đầu sử dụng JMS

cuu duong than cong . com

Kiến trúc JMS

- Các thành phần JMS
 - JMS client:
 - Các chương trình hoặc component (viết bằng ngôn ngữ Java) sử dụng API của JMS để tạo, gửi, nhận message

cuu duong than cong . com

Kiến trúc JMS

- Các thành phần JMS
 - Message:
 - Đối tượng được trao đổi giữa các JMS client.
 - Gồm các thông tin:
 - JMSDestination
 - JMSCorrelationID
 - JMSMessageID
 - JMSReplyTo
 - JMSPriority
 - JMSDeliveryMode
 - ...

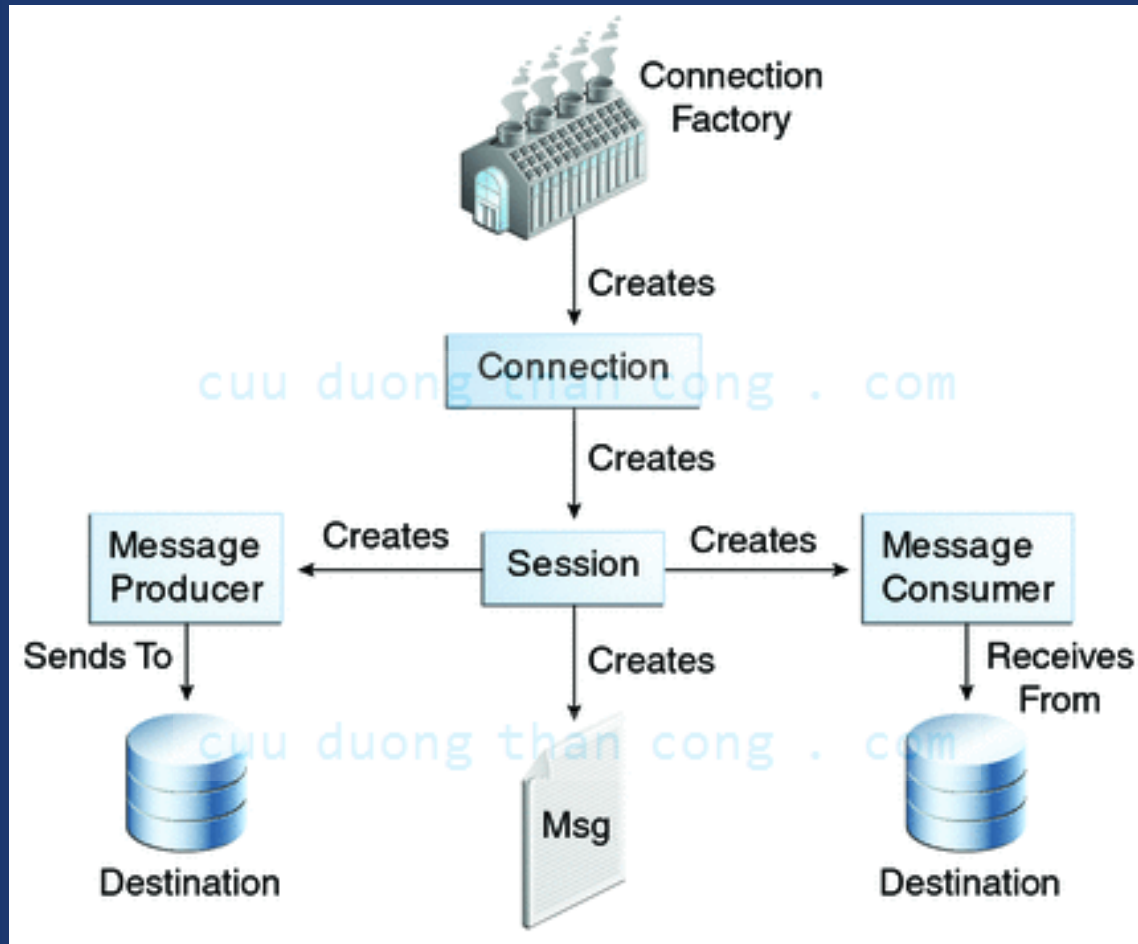
Kiến trúc JMS

- Các thành phần JMS
 - Administered Objects
 - Những objects được cấu hình sẵn (tùy theo provider)
 - Quản trị hệ thống cấu hình trước các đối tượng này, sau đó JMS client truy cập các đối tượng này sử dụng
 - Có 2 loại administered object:
 - ConnectionFactory
 - Destination

Các đối tượng JMS

- Connection factory
- Connection
- Session
- Message producer
- Message consumer
- Message

Các đối tượng JMS



Các đối tượng JMS

- Connection factory
- Connection
- Session
- Message producer
- Message consumer
- Message

Gửi message

```
ActiveMQConnectionFactory connectionFactory = new
    ActiveMQConnectionFactory("tcp://localhost:61616");

Connection connection = connectionFactory.createConnection();
connection.start();
Session session = connection.createSession(false,
    Session.AUTO_ACKNOWLEDGE);
Destination destination = session.createQueue("mytestqueue");
MessageProducer producer = session.createProducer(destination);
producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);

String text = "Hello world! ";
TextMessage message = session.createTextMessage(text);
producer.send(message);
session.close();
connection.close();
```

Nhận message

- Có thể nhận đồng bộ bằng hàm receive của MessageConsumer
- Hoặc nhận bất đồng bộ bằng MessageListener
- Dùng Queue nếu theo mô hình Producer/Consumer
- Dùng Topic nếu theo mô hình Publisher/Subscriber

Nhận message

```
ActiveMQConnectionFactory connectionFactory = new
    ActiveMQConnectionFactory("tcp://localhost:61616");
Connection connection = connectionFactory.createConnection();
    connection.start();
Session session = connection.createSession(false,
    Session.AUTO_ACKNOWLEDGE);
Queue destination = session.createTopic("myqueue");
//Topic destination = session.createTopic("mytopic");

MessageConsumer consumer = session.createConsumer(destination);
consumer.setMessageListener(new MyMessageListener("Topic1"));
```


Gọi lệnh RPC

- Thực hiện gọi hàm từ xa bằng JMS
- Sử dụng các thuộc tính của message:
 - CorrelationID
 - ReplyTo

cuu duong than cong . com