

Obligatorio 1: Mensajería y Web Services

Requerimientos

La empresa de telecomunicaciones INCO Telecommunications está en proceso de implementación de un middleware para la unificación y despacho de órdenes de compra. Dicho middleware será denominado “despachador” y provee una cola de mensajes para la recepción de todas las órdenes de compra generadas por los sistemas de INCO Telecommunications. Es así que una vez que se genera una orden de compra, esta se envía a la cola de mensajes del despachador y este lo redirige al sistema de backoffice correspondiente para su procesamiento.

Una orden de compra está compuesta por información general (p.ej: forma de pago, facturación, información del billing, payment, cliente) e información específica a los ítems de compra (p.ej: celulares, plan de datos, etc), donde cada ítem está asociado a una categoría de producto, la cual determina a qué sistema deberá redirigirse el pedido para su procesamiento. Estos sistemas pueden ser internos a la empresa o externos, provistos por un partner de negocio. Por ejemplo, los ítems de telefonía móvil son procesados por el sistema MobileSys, mientras que los ítems de categoría televisores, son procesados por el partner de negocio Red Strawberry.

En la fase inicial del proyecto se tendrán en cuenta los sistemas internos MobileSys y DataSys. El primero contará con un conjunto de consumidores de tipo event-driven que recibirán los mensajes de una cola de mensajes definida en el despachador, mientras que el segundo creó una nueva tabla donde dejar los ítems de compra. Se deberá insertar una tupla por cada ítem de la orden. Red Strawberry será el partner externo y provee un Web Service SOAP para notificarle las diferentes solicitudes de compra. Este Web Service requiere autenticación http basic y utiliza SSL para mantener la confidencialidad de la información. Asimismo, Red Strawberry solicitó los pedidos de ordenes se empaqueten en grupos de 1000 ítems por pedido.

La comunicación entre el despachador y MobileSys debe garantizar la no pérdida de mensajes, por lo que se deben dar garantías de entrega mediante reintentos y mensajes persistentes (salvados a disco). En caso que MobileSys no pueda procesar un mensaje ante errores inesperados y se hayan agotado los intentos de entrega, los mensajes se deberán enviar a una dead letter queue dentro del despachador. El formato de mensajes de esta comunicación deberá ser XML según el modelo definido en la letra del obligatorio. Se deberá mandar un mensaje por cada ítem de la orden.

La única validación a realizar por el despachador sobre las órdenes de compra es que el monto de la factura se corresponda con la suma del precio de todos los ítems que componen la orden y que la moneda se encuentre dentro de las monedas soportadas. En caso contrario, se debe enviar la orden a un canal de órdenes no válidas para su posterior procesamiento.

Los sistemas generadores de órdenes deberán enviar a la cola de mensajes del despachador, las órdenes de compra en formato XML.

Se pide:

1. Desarrollo de un flujo basado en el framework Spring Integration y patrones de mensajería, que permita implementar el middleware despachador.
2. Desarrollo de una aplicación cliente que permita enviar órdenes de compra al despachador.
3. Desarrollo del Web Service de la empresa Red Strawberry que simule una de las empresas partner de negocio. El Web Service debe ser de tipo One Way.
4. Desarrollo de una aplicación consumidora de mensajes que simule el sistema MobileSys.
5. Implementar todos los sistemas según las estructuras de datos definidas al final del documento.
6. Se deben imprimir en consola las órdenes no validadas por el despachador.

Requerimientos no funcionales

1. Los Web Services deberán implementarse con las librerías JAX-WS. Como servidor de aplicaciones se podrá usar Tomcat o JBoss. Los Web Services deberán loggear todos los pedidos procesados.
2. La aplicación cliente que simula el generador de órdenes deberá desarrollarse utilizando las librerías JMS estándares y no Spring JMS u otra tecnología. Deberá utilizarse ActiveMQ como servidor de colas de mensajes. El formato de estos mensajes deberá ser XML.
3. Deberán existir al menos dos event-driven consumers de la aplicación MobileSys procesando los mensajes de la cola del despachador. El formato de estos mensajes deberá ser XML.
4. El sistema central debe correr sobre Tomcat 8 y estar implementado en Spring Integration 4.3.0
5. El ruteador dentro del despachador deberá implementarse con un ruteador xpath de Spring Integration
6. La transformación de formatos entre el despachador y el sistema MobileSys se deberá implementar con XSLT
7. Se debe utilizar Java 8 para todas las implementaciones
8. Se debe utilizar PostgreSQL como motor de base de datos del sistema DataSys

Entregables:

1. Código fuente de todos los sistemas desarrollados
2. Documentación describiendo la arquitectura del sistema de middleware según los patrones vistos en el curso.

Fecha de entrega: 25 de setiembre hasta las 23:59hrs

Grupos: El trabajo se deberá desarrollar en grupos de cuatro (4) personas

Defensa: Las fechas de las defensas serán el 26/09 y 28/09 en el horario del curso.

Estructura de datos de una orden a enviar al despachador

	Tipo de datos	Valores permitidos
Datos entrada		
Número de orden	Long	N/A
Fecha y hora de creación	Date	N/A
Identificador del cliente	Long	N/A
Forma de pago	String	E = Efectivo, C = Cheque, D = Debito, X = Crédito
Facturación		
Monto	Double	Debe ser el equivalente a la suma de todos precios de los ítems que componen la oferta
Moneda	Integer	858 = Pesos, 840 = Dólares
Cuotas	Integer	1 para forma de pago Efectivo, Cheque o Débito N para forma de pago Crédito
Items*		
Número de ítem	Long	N/A
Categoría	Integer	1 = Datos, 2 = Telefonía Móvil, 3 = Red Strawberry
Id Producto	Long	
Descripción producto	String	
Cantidad	Integer	
Precio	Double	

(*) Lista de ítems

Estructura de datos del Web Service de Red Strawberry

	Tipo de datos	Valores permitidos
Datos entrada		
Identificador del producto	Long	N/A
Cantidad	Date	N/A
Identificador de la transacción	Long	Se debe utilizar la concatenación del id de orden, el carácter ':' y id del ítem. Ejemplo: <ul style="list-style-type: none"> idOrden: 12345 idItem: 9876 Id Transacción: 12345:9876
Fecha y hora	Date	Fecha y hora de la solicitud, al Web Service en el despachador.

Estructura de la tabla del sistema DataSys

Atributo	Tipo de datos	Valores permitidos
Identificador cliente	Long	N/A
Identificador del producto	Long	N/A
Cantidad	Integer	N/A
Id Orden	Long	N/A
Id de ítem	Long	N/A

Datos a enviar al sistema MobileSys

Atributo	Tipo de datos	Valores permitidos
Identificador cliente	String	N/A
Identificador del producto	String	N/A
Fecha y hora de creación de la orden	String	Formato dd/mm/yyyy hh:mm:ss
IdOrdenItem	String	IdOrden concatenado con el id del ítem, todo junto. Ejemplo <ul style="list-style-type: none">• idOrden: 12345• idItem: 9876 IdOrdenItem: 123459876

Referencias

1. Spring integration: <http://projects.spring.io/spring-integration/>
2. Spring Tool Suite: <http://spring.io/tools>
3. Gradle tutorial: <http://spring.io/guides/gs/gradle/>
4. Maven tutorial: <http://spring.io/guides/gs/maven/>
5. JMS tutorial: <http://docs.oracle.com/javaee/6/tutorial/doc/bncdq.html>
6. Web Services tutorial: <http://docs.oracle.com/javaee/6/tutorial/doc/bnayk.html>
7. Getting started with MySQL: http://dev.mysql.com/usingmysql/get_started.html
8. ActiveMQ (AMQ): <http://activemq.apache.org/>
9. Tomcat Web Server: <http://tomcat.apache.org/>
10. JBoss Application Server: <http://jbossas.jboss.org/>
11. PostgreSQL: <https://www.postgresql.org/>