

Obligatorio 1: Mensajería y Web Services

Requerimientos

ACME Shop es una empresa de comercialización de productos que posee tiendas físicas y virtuales, en particular, un sitio de e-commerce desarrollado hace varios años. Actualmente, el sitio web se comunica con un Web Service SOAP a quien envía las órdenes que generan los clientes. Este Web Service se comunica con cada uno de los sistemas de procesamiento de órdenes en forma sincrónica. Debido a la pandemia, las ventas a través del e-commerce se dispararon aumentando en un 45%. Esto provocó una saturación de los sistemas de la organización, así como la generación de incidentes y demoras en el procesamiento de las órdenes. Es por esto mismo que se desea pasar de un modelo sincrónico a un modelo asíncrono basado en mensajería que les permita soportar un mayor volumen de procesamiento.

El sitio de e-commerce actual se comunica con un Web Service SOAP asegurado con basic authentication y TLS. Se desea modificar la implementación actual que se comunica con los dos sistemas de procesamiento de órdenes vía Web Services sincrónicos, para comunicarse únicamente con una cola de mensajes de tipo punto a punto y deje cada una de las órdenes de compra completa.

Se desarrollará un nuevo middleware de integración que permita procesar las órdenes enviadas a la cola de mensajes, procesarlas y enviarlas a los dos sistemas de backend: 1) electrónica y 2) vestimenta. Este middleware deberá implementar un polling consumer para recibir los mensajes de la cola de mensajes. Una orden de compra puede contener productos de electrónica, vestimenta y ambos a la vez. El middleware debe enviar a cada sistema sólo los productos que procesa. Esto es, las órdenes de productos de vestimenta, al sistema de vestimenta y las órdenes de productos de electrónica, al sistema de electrónica.

Las órdenes de compra generadas por el sitio de e-commerce se encuentran en dólares, pero los sistemas (vestimenta y electrónica) solo permiten el procesamiento de productos en moneda nacional. Por este motivo, el middleware debe tomar la cotización del dólar de una base de datos existente y agregarle a cada producto su valor en moneda nacional. Esta base de datos posee una tabla que dada la fecha retorna la cotización del dólar.

El sistema de electrónica es un viejo sistema Cobol y solo permite una integración mediante archivos, por lo que el middleware debe enviar las órdenes de compra vía archivos de texto en formato Comma Separated Values (CSV). Existirá un archivo por hora de órdenes y cada orden se irá agregando al final del archivo (append).

El sistema de vestimenta cuenta con un Web Services SOAP a quien enviar cada una de las órdenes. Para una mayor eficiencia, el middleware debe agrupar las órdenes de compra y enviarlas en grupos de 100 órdenes. Este Web Service debe tener una interacción de tipo OneWay.

El middleware debe filtrar aquellas órdenes que no se encuentren en el formato correcto o tengan información faltante, para luego ser procesadas manualmente e identificar su error. Estas órdenes se deben enviar a un Invalid Message Channel. Las órdenes con error son procesadas manualmente usando el sistema de soporte de ACME Shop. El sistema de soporte implementa el patrón event driven consumer combinado con el competing consumer para procesar los mensajes.

Toda orden de compra debe ser tomada por el middleware desde la cola de mensajes en un tiempo de 1 hora, en caso contrario y el mensaje no haya sido procesado, este se debe enviar a una dead letter queue para ser procesada manualmente más tarde.

Se pide:

1. Desarrollar un cliente Java consola o web que emule el sitio de e-commerce y envíe al Web Service SOAP las órdenes de los clientes.
2. Desarrollo de un Web Service SOAP de tipo request/response que reciba las órdenes del sitio de e-commerce y las envíe a la cola de mensajes. Los mensajes se deben enviar en formato XML.
3. Desarrollo del middleware utilizando el framework Spring Integration y patrones de mensajería que lleve a cabo la integración solicitada entre los sistemas de ACME Shop: vestimenta, electrónica, soporte.
4. Desarrollo de una aplicación Java que simula al sistema de soporte.
5. Desarrollo del Web Service SOAP que simule el sistema de vestimenta.
6. Implementar todos los sistemas según las estructuras de datos definidas al final del documento.

Requerimientos no funcionales

1. Los Web Services SOAP deberán implementarse con JAX-WS y la librería Apache CXF. Se deberán loggear todos los mensajes procesados.
2. Se debe usar Tomcat como servidor de Web Services
3. La comunicación con las colas de mensajes deberá realizarse usando las librerías JMS 2.0 estándares. No se permite el uso de Spring JMS u otra tecnología similar.
4. Se debe emular el sitio de e-commerce como una aplicación Java de consola o Web. Se debe desarrollar un cliente de Web Services utilizando JAX-WS.
5. Se debe utilizar Artemis como servidor de mensajería.
6. El formato de los mensajes generado por el Web Service y enviado a la cola de mensajes debe ser XML.
7. La transformación de XML a CSV para la integración con el sistema de electrónica debe realizarse con XSLT.
8. Los patrones EIP a utilizar deben ser los provistos por Spring Integration.
9. Se debe utilizar Java 11 para todas las implementaciones.
10. El middleware debe estar implementado en Spring Integration 5.
11. Se debe utilizar Gradle como gestor de dependencias.

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.
3. Documentación con casos de prueba ejecutados sobre el sistema.

Forma de entrega: Para la entrega se usará el Gitlab de Facultad, donde los grupos deberán subir el código al repositorio asignado y documentación de arquitectura y documento con casos de prueba ejecutados. No se tomarán en cuenta entregas pasada la hora de entrega.

Introducción al middleware 2021 – Obligatorio 1

Fecha de entrega: 19 de setiembre hasta las 23:59hrs.

No se aceptarán entregas pasada la fecha/hora indicada.

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

Defensa: Las fechas de las defensas serán el 20 y 23 de setiembre en el horario de 19hrs a 21hrs.

Estructura de datos de los mensajes recibidos por el Web Service del e-commerce

	Tipo de datos	Descripción
Id	long	Identificador de la orden
Order Date	date	Fecha de realizada la orden. Formato YYYY-MM-DDThh:mm:ss
Payment confirmation	String	Identificador generado por el autorizador del pago
Número de cliente	Long	Número de cliente
Dirección de envío		
Calle	String	Nombre de la calle
Número	String	Número de puerta
Localidad	String	Nombre de la localidad
Departamento	String	Nombre del departamento
País	String	Código ISO 3166 del país (Alpha-2)
Dirección de facturación		
Calle	String	Nombre de la calle
Número	String	Número de puerta
Localidad	String	Nombre de la localidad
Departamento	String	Nombre del departamento
País	String	Código ISO 3166 del país (Alpha-2)
Lista de productos		
Id	Long	Identificador de producto
Cantidad	Integer	Cantidad del producto
Precio	double	Precio del producto en dólares
Type	String	Tipo de producto: V = vestimenta; E = electrónica

Estructura de datos del Web Service del sistema de vestimenta

	Tipo de datos	Descripción
Orden de compra e-commerce	Long	Número de orden de compra del e-commerce
Order Date	Datetime	Fecha de realizada la orden. Formato YYYY-MM-DDThh:mm:ss
Dirección de envío		
Calle	String	Nombre de la calle
Número	String	Número de puerta
Localidad	String	Nombre de la localidad
Departamento	String	Nombre del departamento
País	String	Código ISO 3166 del país (Alpha-2)
Lista de productos		
Id	Long	Identificador de producto
Cantidad	Integer	Cantidad del producto
Precio	double	Precio del producto en pesos

Estructura de datos del archivo CSV del sistema de electrónica

El archivo debe contener una línea por cada orden de compra del e-commerce. Cada línea debe incluir los datos de la siguiente tabla.

	Tipo de datos	Descripción
Orden de compra e-commerce	String	Número de orden de compra del e-commerce
Fecha de compra	String	Formato mmddyy
Hora de compra	String	Formato hhmmss
Payment confirmation	String	Identificador generado por el autorizador del pago
Número de cliente	String	Número de cliente
Calle	String	Nombre de la calle de envío
Número	String	Número de puerta de envío
Localidad	String	Nombre de la localidad de envío
Departamento	String	Nombre del departamento de envío
País	String	Código ISO 3166 del país (Alpha-3)
Id	String	Identificador de producto
Cantidad	String	Cantidad del producto
Precio	String	Precio del producto en formato DDDDDDD, donde los últimos dos dígitos son los decimales. Precio en pesos

Estructura de la tabla del sistema de cotización

	Tipo de datos	Descripción
Identificador	String	Identificador del registro. Clave primaria
Fecha	String	Formato: dd/MM/yyyy
Precio de compra	Double	Cotización de compra del dólar
Precio de venta	Double	Cotización de venta del dólar

Referencias

1. Spring integration: <https://spring.io/projects/spring-integration>
2. Spring Tool Suite: <http://spring.io/tools>
3. Eclipse: <https://www.eclipse.org/downloads/>
4. Gradle tutorial:
https://docs.gradle.org/current/samples/sample_building_java_applications.html
5. JMS tutorial: <http://docs.oracle.com/javaee/6/tutorial/doc/bncdq.html>
6. JMS 2.0: <https://www.oracle.com/technetwork/articles/java/jms20-1947669.html>
7. Web Services tutorial: <http://docs.oracle.com/javaee/6/tutorial/doc/bnayk.html>
8. Crear Web Service SOAP con CXF: <https://www.youtube.com/watch?v=pkmZ-xeFqso>
9. Crear un cliente de Web Service SOAP con CXF:
<https://www.youtube.com/watch?v=F1aVpn9G14I>
10. Artemis: <https://activemq.apache.org/components/artemis/>