

Ejercicio “TusLibros.com”

La editorial “TusLibros” desea realizar ventas de manera electrónica, por lo que han decidido abrir dos canales de ventas con sus clientes, uno on-line usando REST, otro batch usando archivos.

Para ambos casos el pago se realizará con tarjeta de crédito la cual debe ser validada con un Merchant Processor externo. El sistema debe guardar todas las compras realizadas por los clientes para luego poder reponer stock, sacar estadísticas de ventas, etc.

En la interfaz REST es muy importante la autenticación de los clientes ya que es una interfaz abierta al público (no hay ningún tipo de control realizado por red como filtros por IP, MAC, etc).

La interfaz batch no necesita esta tipo de autenticación porque los archivos son recibidos por email y una persona se asegura de que son del cliente correcto.

El Merchant Processor cobra por transacción por lo que se debe validar la mayor cantidad de información posible antes de enviarla al mismo.

El Merchant processor ofrece un ambiente de desarrollo para realizar pruebas, pero también cobra por transacción por lo que se debe evitar al máximo hacer pruebas con el mismo al menos que sea necesario (o sea, solo usar el ambiente de desarrollo del merchant processor en el ambiente de integración del sistema que estamos desarrollando).

El Merchant Processor provee una interface Rest para realizar los débitos a las tarjetas. La misma se ejecuta haciendo post a <https://merchant.com/debit> (para producción) y <https://merchanttest.com/debit> (para desarrollo) y recibe los siguientes parámetros:

- 1) creditCardNumber: Número de tarjeta de crédito
- 2) creditCardExpiration: Fecha de expiración con 2 dígitos para el mes y 4 para el año
- 3) creditCardOwner: Nombre del dueño de la tarjeta. Máximo de 30 caracteres
- 4) transactionAmount: Cantidad de la transacción, con un máximo de 15 dígitos para la parte entera y dos dígitos para la decimal utilizando el punto como separador. Siempre se debe generar la parte decimal por más que sea cero.

Por ejemplo:

<https://merchant.com/debit?creditCardNumber=5400000000000001&creditCardExpiration=072011&creditCardOwner=PEPE%20SANCHEZ&transactionAmount=123.50>

Estos datos de ejemplo son los utilizados como datos válidos por el ambiente de desarrollo del Merchant Processor.

Si algún parámetro no tiene el formato correcto devuelve HTTP 400 (Bad request), en caso contrario devuelve HTTP 200 con el siguiente contenido:

- 1) En caso exitoso: 0|TRANSACTION_ID
- 2) En caso de no poder realizar el débito: 1|DESCRIPCION_DE_ERROR
En caso de que la transacción no se pudo realizar se desea pasar la descripción del error al usuario de sistema.

La interfaz on-line REST que ofrecerá el sistema debe permitir crear un carrito (el cual será válido durante 30 minutos luego de la última vez que se realizó alguna operación con él), agregar un libro con su cantidad al carrito ya creado, consultar el contenido de un carrito, hacer el check out y listar las compras de un cliente. Las interfaces son:

- 1) Recurso: /createCart

Parámetros:

clientId: ID del cliente que está creando el carrito

password: Password del cliente que valida que puede operar con TusLibros.com

Output:

En caso de éxito: 0|ID_DEL_CARRITO

En caso de error: 1|DESCRIPCION_DE_ERROR

- 2) Recurso: /addToCart

Parámetros:

cartId: Id del carrito creado con /createCart

bookIsbn: ISBN del libro que se desea agregar. Debe ser un ISBN de la editorial

bookQuantity: Cantidad de libros que se desean agregar. Debe ser ≥ 1 .

Output:

En caso de éxito: 0|OK

En caso de error: 1|DESCRIPCION_DE_ERROR

3) Recurso: /listCart

Parámetros:

cartId: Id del carrito creado con /createCart

Output:

En caso de éxito: 0|ISBN_1|QUANTITY_1|ISBN_2|QUANTITY_2|...|ISBN_N|QUANTITY_N

En caso de error: 1|DESCRIPCION_DE_ERROR

4) Recurso: /checkOutCart

Parámetros:

cartId: Id del carrito creado con /createCart

ccn: Número de tarjeta de credito

cced: Fecha de expiración con 2 digitos para el mes y 4 para el año

cco: Nombre del dueño de la tarjeta.

Output:

En caso de éxito: 0|TRANSACTION_ID

En caso de error: 1|DESCRIPCION_DE_ERROR

5) Recurso: /listPurchases

Parámetros:

clientId: ID del cliente que quiere ver que compras hizo

password: Password del cliente que valida que puede operar con TusLibros.com

Output:

En caso de éxito: 0|ISBN_1|QUANTITY_1|...|ISBN_N|QUANTITY_N|TOTAL_AMOUNT

En caso de error: 1|DESCRIPCION_DE_ERROR

Si el request realizado no cumple con las reglas sintácticas, se debe devolver HTTP 400 (Bad request). Si cumple con la sintaxis se debe devolver HTTP 200 (OK).

Los archivos utilizados como interface batch son:

De entrada: CLIENTE_INPUT_AAAA_MM_DD.csv con el siguiente formato: TipoDeRegistro,RestoDelRegistro

Donde:

1) Tipo de registro 1

a) Significa: Agregar un libro al carrito

b) Formato: ISBN,QUANTITY (Crear el carrito si el mismo no existe)

2) Tipo de registro 2

a) Significa: Realizar el checkout del carrito

b) Formato: numero_de_tarjeta,fecha_de_expiración,nombre_del_dueño.

De Salida: CLIENTE_OUTPUT_AAAA_MM_DD.csv donde cada registro tiene el siguiente formato:

1) Resultado de Transacción: 0 para exito, 1 en caso de error

2) En caso de éxito, id_de_transacción, total_de_transacción

En caso de error: descripción_del_error

Ejemplo:

Archivo TEMATIKA_INPUT_2010_02_01.csv

1,0321146530,2

1,1933988274,1

2,5400000000000001,072011,PEPE SANCHEZ

1,1933988274,3

2,5400000000000002,132012,KENT BECK

Archivo TEMATIKA_OUTPUT_2010_02_01.csv

1,10533,60.53