Documento de Integración de APIs

Ambas APIs comparten gran similitud en cuanto a funcionalidad y exposición de los datos aunque comparten diversas diferencias.

Se exponen a continuación las tablas de equivalencias de los parámetros de consulta y de las respuestas:

Parámetros de consulta

API A	API B
fecha_inicio (string)	start_date (string)
fecha_fin (string)	end_date (string)

Parámetros de respuestas

API A	API B
facturas (JsonArray)	invoices (JsonArray)
id (string)	invoice_id (string)
cliente (string)	customer (string)
monto (float)	amount_due (float)
fecha_emision (string)	date_issued (string)
estado (enum)	status (enum)

estado (API A)	status (API B)
pagada	paid
no pagada	unpaid

Cabe destacar que los parámetros fecha_inicio/start_date, fecha_fin/end_date y fecha_emision/date_issued representan fechas con un formato **YYYY-MM-DD**.

Propuesta de flujo

El flujo propuesto para la integración de las APIs consistirá de diversas partes.
El flujo propuesto considera su comienzo tras la llamada al endpoint `https://api.sistemaB.com/bills` de la API B, por lo que los parámetros de consulta “start_date” y “end_date” junto con el header “x-api-key” se considera que son proporcionados a la hora de realizar la llamada.

1. Comprobación de parámetros de entrada

En primer lugar se deben comprobar el correcto formato de los parámetros de consulta “start_date” y “end_date”, al igual que la validez de la api key contenida en el header “x-api-key”.

En caso de que alguno de los parámetros de consulta sea incorrecto, se devolverá un mensaje de error con código de estado 400.

En caso de que la api key no haya sido proporcionada se devolverá un mensaje de error con un código de estado 401 y, en caso de que sí haya sido proporcionada pero el usuario no tenga permisos para acceder al endpoint o sea inválida, se devolverá un 403.

2. Obtención del token de autenticación para la API A

Tal y como se describe en el objetivo de la práctica, la API A es la API de la cual la API B obtendrá los datos. Para ello primero es necesario obtener la autorización para acceder a dichos recursos mediante un token obtenido a través de client credentials.
Para esto la API B deberá registrarse con el proveedor de la API A para obtener los datos necesarios (client_id y client_secret) para, posteriormente obtener el token llamando al endpoint de autorización correspondiente de la API A (posiblemente `https://api.sistemaA.com/oauth/token`).

3. Obtención de datos

Tras obtener la autorización para obtener los datos requeridos de la API A, la API B deberá llamar al endpoint `https://api.sistemaA.com/facturas` para obtener los datos requeridos.

Para realizar esta llamada se utilizarán los parámetros de consulta de la API B “start_date” y “end_date” que se corresponden con “fecha_inicio” y “fecha_fin” respectivamente acorde a la tabla de equivalencias entre APIs y el token obtenido mediante el proveedor OAuth 2.0 de la api A.

4. Comprobación de datos obtenidos

Una vez obtenidos los datos de la API A se deberá comprobar que los datos obtenidos cumplen el formato preestablecido y que sean válidos, en caso de que los datos obtenidos no cumplan, la API B deberá devolver un mensaje de error con un código de estado 502.

5. Transformación de datos

Una vez obtenidos y validados los datos de la API A, llega el momento de transformarlos para obtener los datos que la API B debería devolver.
En este caso la transformación es sencilla atendiendo a las tablas de equivalencias mostradas anteriormente, por lo que fácilmente podemos obtener los datos.

6. Devolución de los datos

Tras estos pasos el flujo ha terminado y solo resta devolver los datos al usuario con un código de estado 200.