Sistema Integrado Registro Electrónico- SIRE Ventas

SERVICIOS WEB API - SIRE VENTAS

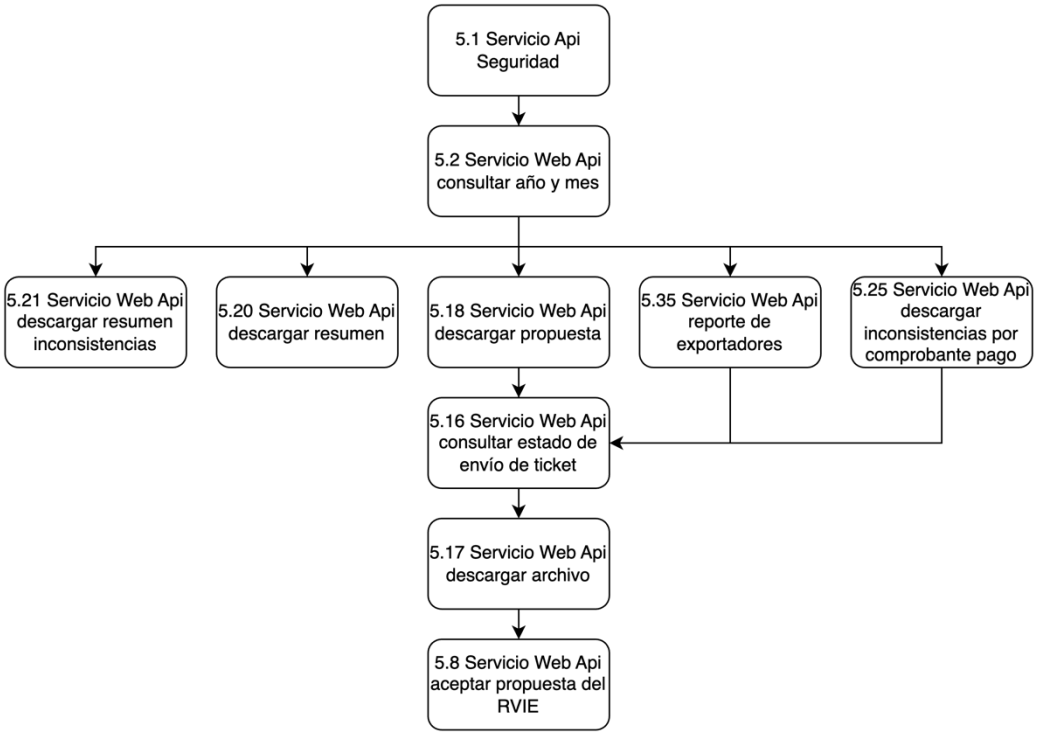
Manual de Usuario

Introducción

Este documento Manual de usuario de los Servicios Web Api - SIRE VENTAS, ha sido diseñado para ser usado como instructivo en la integración de los servicios Web Api expuestos del SIRE Ventas por la SUNAT con los sistemas informáticos de los declarantes, que tienen la necesidad de integrarlos desde sus aplicaciones.

El proyecto SIRE VENTAS que expone los servicios aquí descritos, ha sido desarrollado con la finalidad de facilitar el cumplimiento voluntario de las obligaciones tributarias de los contribuyentes y toma como base al comprobante de pago electrónico para el control del flujo de la transacción del IGV y la información que se genera en cada fase.

El SIRE Ventas una vez autenticado le permite al contribuyente:



Servicios principales:

- Descargar la propuesta **(Servicio: Descargar propuesta)** con el detalle individualizado de los comprobantes y documentos que deberían integrar el registro de ventas que genere, la cual podría ser la propuesta inicial de la SUNAT o aquella que fue actualizada por el contribuyente.

- Aceptar Propuesta **(Servicio: Aceptar propuesta)** permite actualizar el estado del registro libro y Control de procesos para indicar que se está registrando un preliminar a través de la propuesta aceptada.
- Reemplazar propuesta **(Servicio: Reemplazar propuesta)** permite al generador, reemplazar la propuesta SUNAT con lo considerado por el contribuyente mediante el uso de un archivo de formato .txt.

- Registrar preliminar **(Servicio: Registrar preliminar)** permite registrar los comprobantes del preliminar según corresponda al proceso ejecutado por el generador.

Servicios complementarios al proceso:

- Descargar inconsistencias por comprobantes de pago (**Servicio**: **Descargar Inconsistencias por comprobantes**) este servicio WEB API permite descargar las inconsistencias asociadas a los comprobantes que se encuentran en la fase actual de proceso del RVIE, que pueden ser 1-Propuesta o 3-Preliminar.

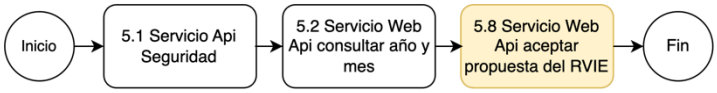
- Consultar el estado del ticket (**Servicio**: **Consultar estado del ticket**) permite al generador consultar el estado del número ticket asociado al proceso que genera el archivo de descarga o carga. Si el estado es "Terminado", devuelve el nombre del archivo generado, si el estado del ticket es diferente, devuelve el estado del ticket.

- Descargar archivo (**Servicio**: **Descargar archivo ticket generado**) permite realizar la descarga de los archivos generados zipeados y particionados guardados en el fileserver.

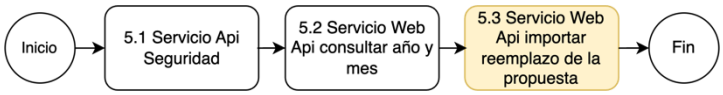
Entre otros servicios, que se detallarán en el presente manual.

La siguiente imagen muestra el flujo mínimo para registrar el preliminar del Registro de Ventas Electrónico de un periodo:

Aceptar propuesta:



Reemplazar propuesta:



El manual comienza con la sección que describe el procedimiento inicial para obtener las credenciales del token, necesarios para hacer uso de los servicios.

Tabla de contenido

Control de cambios del documento 6

I. Guía de Uso 11

1. Servicio prerrequisito 11

2. Secuencia de servicios mínimos para Registrar Preliminar 13

Funcionalidad 1: Aceptar Propuesta 13

Funcionalidad 2: Reemplazar Propuesta 14

Funcionalidad 3: Registrar Preliminar 15

3. Secuencia de servicios interdependientes que completan funcionalidades del SIRE Ventas 16

Funcionalidad 1: Tipo de cambio 16

Funcionalidad 2: Importar comprobantes en propuesta 17

Funcionalidad 3: Importar comprobantes en preliminar 18

Funcionalidad 4: Importar ajustes posteriores 19

Funcionalidad 5: Importar ajustes posteriores de periodos anteriores 19

Funcionalidad 6: Eliminar comprobantes en preliminar 20

Funcionalidad 7: Consultar estado de envío de ticket 20

Funcionalidad 8: Descargar archivo 21

4. Servicios accesorios que pueden ser consumidos en el SIRE Ventas 22

5. Documentación Servicios Web API 24

5.1 Servicio Api Seguridad 24

5.2 Servicio Web Api consultar año y mes 25

5.3 Servicio Web Api importar reemplazo de la propuesta 26

5.4 Servicio Web Api importar nuevos comprobantes propuesta 28

5.5 Servicio Web Api importar nuevos comprobantes preliminar 30

5.6 Servicio Web Api importar ajustes posteriores 31

5.7 Servicio Web Api importar ajustes posteriores de periodos anteriores 33

5.8 Servicio Web Api aceptar propuesta del RVIE 34

5.9 Servicio Web Api registrar preliminar 35

5.10 Servicio Web Api exclusión definitiva de notas de crédito y facturas 37

5.11 Servicio Web Api agregar tipo de cambio masivo 38

5.12 Servicio Web Api editar tipo de cambio individual 39

5.13 Servicio Web Api eliminar comprobante propuesta 41

5.14 Servicio Web Api eliminar comprobante preliminar 42

5.15 Servicio Web Api eliminar preliminar 43

5.16 Servicio Web Api consultar estado de envío de ticket 45

5.17 Servicio Web Api descargar archivo 47
5.18 Servicio Web Api descargar propuesta 49
5.19 Servicio Web Api descargar no incluidos 51
5.20 Servicio Web Api descargar resumen 53
5.21 Servicio Web Api descargar resumen inconsistencias 54
5.22 Servicio Web Api exportar preliminar de registro de Ventas 56
5.23 Servicio Web Api descargar reporte de casillas 58
5.24 Servicio Web Api descargar inconsistencias en registros preliminar registrado 59
5.25 Servicio Web Api descargar inconsistencias por comprobante pago 60
5.26 Servicio Web Api descargar constancia de recepción 62
5.27 Servicio Web Api descargar RVIE por periodo 63
5.28 Servicio Web Api descargar reporte consolidado por periodo 64
5.29 Servicio Web Api descargar ajustes posteriores 66
5.30 Servicio Web Api descargar ajustes posteriores de periodos anteriores 67
5.31 Servicio Web Api descargar reporte inconsistencias por periodo 68
5.32 Servicio Web Api descargar reporte CAR 69
5.33 Servicio Web Api descargar reporte estadístico 70
5.34 Servicio Web Api descargar reporte de cumplimiento 72
5.35 Servicio Web Api reporte de exportadores 73
5.36 Servicio Web Api eliminar preliminar registrado 75
5.37 Servicio Web Api consultar preliminares registrados 76
6. Documentación TUS.IO 78
7. Anexos 79
7.1 Anexo I: Indicador de carga masiva 79
7.2 Anexo II: Tipo de correlativo 80
7.3 Anexo III: Código de estado de envío 80
7.4 Anexo IV: Extension del archivo a descargar 81
7.5 Anexo V: Número de casillas 81
7.6 Anexo VI: Ejemplo cliente TUS JAVA 81
Clase:: TusResponseBody.java 82
Clase:: Http401And403CodeException.java 84
Clase:: Http422CodeException.java 85
Clase:: HttpErrorCodeException.java 86
Clase:: TusClientCustom.java 87
Clase: TusUploaderCustom.java 98
Clase:: Demo.java 108

Control de cambios del documento

N.°	Descripción	Fecha	Versión	Responsable	Motivo de cambio
1	Creación del documento	01/04/2023	1	FSW III	Creación
2	Actualización del documento	31/05/2023	15	FSW	Actualización
3	Actualización del documento	04/08/2023	16	FSW	Actualización
4	Actualización del documento	15/08/2023	17	FSW	Actualización
5	Actualización del documento	16/08/2023	18	FSW	Actualización
6	Actualización del documento	03/11/2023	19	INSI	Actualización
7	Actualización del documento	06/11/2023	20	FSW	Actualización
8	Actualización del documento	09/01/2024	21	INSI	Se agrega nuevo modelo de cliente TUS JAVA que permite recuperar los mensajes de error 422.
9	Actualización del documento	08/03/2024	22	INSI	Se actualiza el servicio 5.36 Servicio Web Api eliminar preliminar registrado -

					VENTAS dice: id-alfanumérico-String Id de registro (Obligatorio) debe decir: id-alfanumérico-String Id de registro (Opcional]) Mensaje Error debe incluir en el punto Mensaje Error : ● 1010 - El libro electronico con los siguientes datos: numero de RUC: XXXXXXXX periodo Tributario: AAAAMM y codigo de Libro:140000 no existe.
10	Actualizacion del Documento	18/03/2024	23	INSI	Se actualiza el servicio 5.36 Servicio Web Api eliminar preliminar registrado – VENTAS Se incluye los siguientes mensajes de Error: CATALOGO_ERROR_2296 CATALOGO_ERROR_2297 Se actualiza el servicio 5.9 Servicio Web Api Registrar preliminares –

					VENTAS Se incluye los siguientes mensajes de Error: CATALOGO_ERROR_2293 CATALOGO_ERROR_2294 CATALOGO_ERROR_2295 Se actualiza el servicio 5.15 Servicio Web Api eliminar reemplazo propuesta - VENTAS Se incluye los siguientes mensajes de Error: CATALOGO_ERROR_2298 CATALOGO_ERROR_2299 CATALOGO_ERROR_2300
11	Actualización de documento	30/07/2024	24	INSI	Se actualiza el servicio 5.13 Servicio Web Api eliminar comprobante propuesta VENTAS Cambiando de método de POST a DELETE para una eliminación de forma masiva
12	Actualización de documento	05/06/2025	25	INSI	Se agrega el servicio 5.37 Servicio Web Api consultar preliminares registrados Se modifica en 5.17 Servicio Web Api descargar

					archivo la descripción general y de los siguientes parámetros: nomArchivoReporte-alfanumérico-String codTipoArchivoReporte-numérico-String codLibro-numérico-String perTributario-alfanumérico-String codProceso-alfanumérico-String numTicket-alfanumérico-String
13	Actualización de documento	04/08/2025	26	INSI	Se agrega el anexo 7.5 Anexo IV: Número de casillas Se actualiza el servicio 5.24 Servicio Web Api descargar inconsistencias en registros preliminar registrado
14	Actualizacion de documento	13/08/2025	27	INSI	Se actualiza el diagrama de Servicios accesorios que pueden ser consumidos en el SIRE Ventas, se agrega el servicio consultar preliminar registrado y eliminar preliminar registrado

15	Actualización de documento	19/08/2025	28	INSI	Se actualiza el servicio 5.37 Servicio Web Api consultar preliminares registrados
19	Actualización de documento	17/02/2026	29	INSI	Se actualiza el servicio 5.9 Servicio Web Api registrar preliminar. Se actualiza el servicio 5.15 Servicio Web Api eliminar preliminar. Se actualiza el servicio 5.36 Servicio Web Api eliminar preliminar registrado.

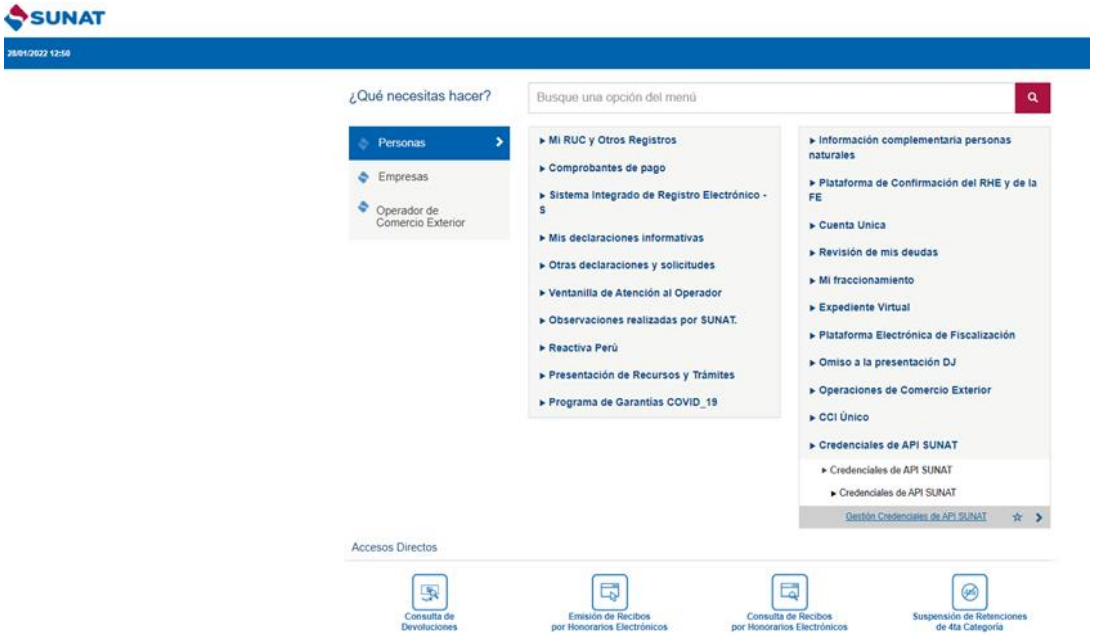
I. Guía de Uso

1. Servicio prerrequisito

a) Diagrama: Esquema gráfico de la secuencia de pasos para llegar a consumir el servicio web, a nivel de proceso para obtener el token

El contribuyente, usuario del sistema SIRE Ventas, que se encuentra obligado a generar el registro de Ventas de manera periódica, debe ingresar al **Portal SOL de la SUNAT** (https://e-menu.sunat.gob.pe/cl-ti-itmenu/MenuInternet.htm) e iniciar sesión con su correspondiente **Clave SOL**.

Ingresar a la opción "EMPRESAS / Credenciales de API SUNAT / Credenciales de API SUNAT/ Credenciales de API SUNAT/ **Gestión Credenciales de API SUNAT**"



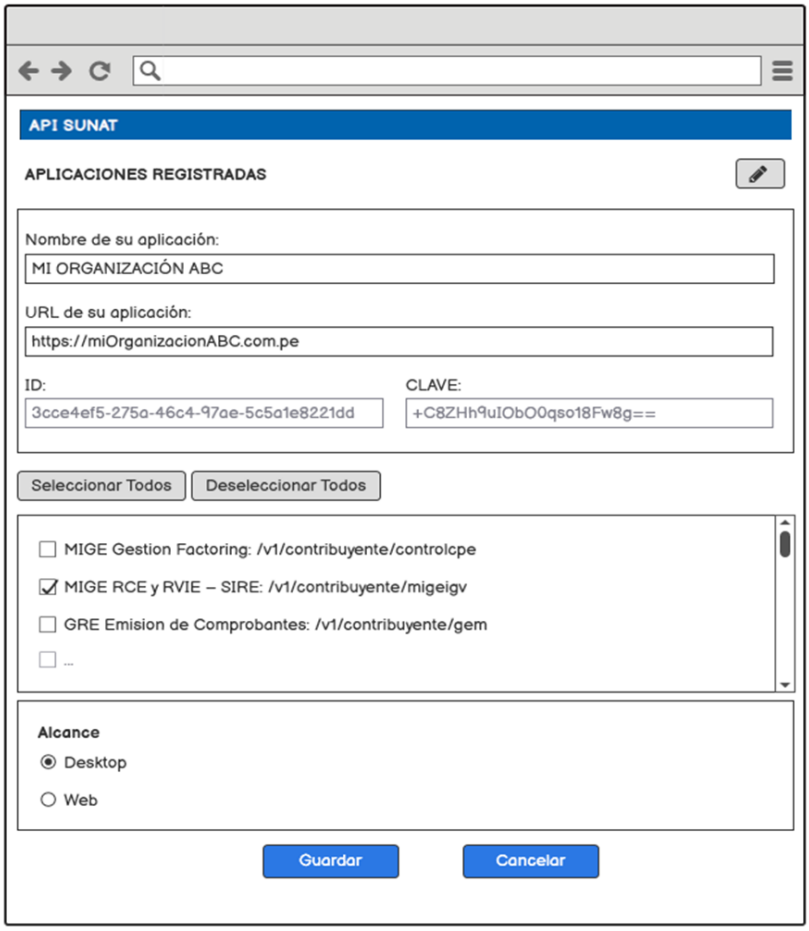
Se muestran las secciones:

- REGISTRE SU APLICACIÓN
  - Nombre de su aplicación: Alfanumérico, obligatorio y con longitud de 50 caracteres. Ejemplo: Mi Organización ABC
  - URL de su aplicación: Alfanumérico, obligatorio y con longitud de 260. Ejemplo: https://miOrganizacionAbc.com

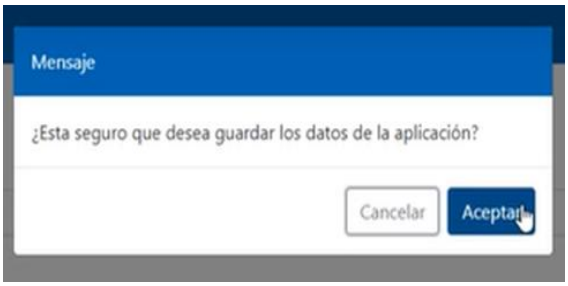
- SECCIÓN LISTADO DE URI'S
  - Botón de acción:
    - Seleccionar Todos, permite seleccionar o marcar todas las casillas que hacen referencia a la URI's
    - Deseleccionar Todos, permite desmarcar todas las casillas que han sido marcadas o seleccionadas.
  - Lista de URI's, muestra un listado de URI´s donde el contribuyente debe seleccionar las URI´s que va consumir para que el sistema le brinde el acceso correspondiente.
  - El contribuyente deberá seleccionar la URI: "MIGE RCE y RVIE - SIRE"

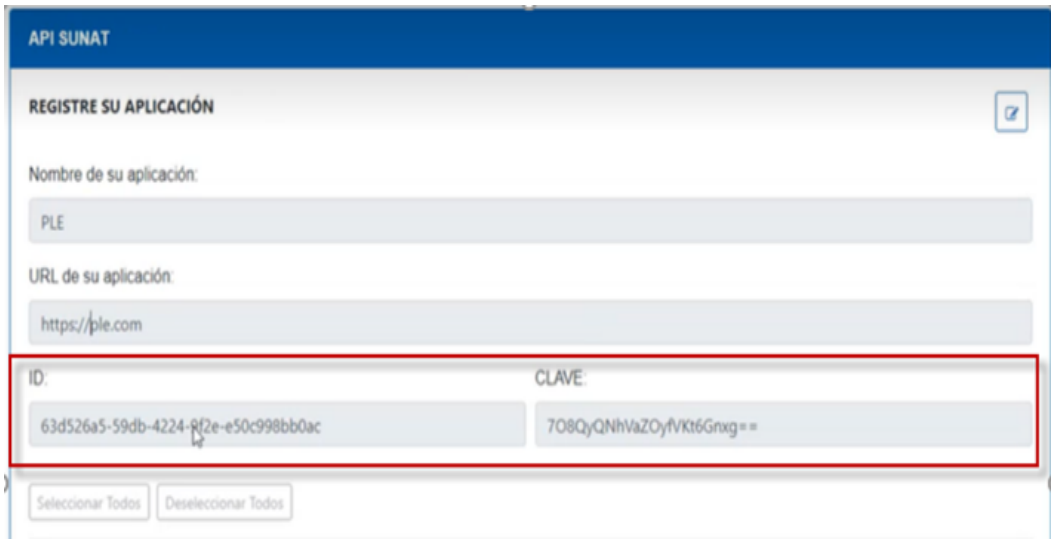
- ALCANCE
    - Desktop, indica que la(s) uri(s) seleccionada(s) van a ser consumidas desde una interfaz de escritorio.
    - Web, indica que la(s) uri(s) seleccionada(s) van a ser consumidas desde una interfaz web.

- BOTÓN DE ACCIÓN
    - Guardar, registrar en la base de datos la información de la empresa del contribuyente, las uris el cual va a consumir y el alcance.
    - Si el contribuyente ya registró su aplicación y las URI´s correspondientes
    - Seleccionar el botón Editar ubicado en la parte superior derecha
    - El usuario del SIRE, debe ingresar los datos solicitados.
    - Una vez completados los datos requeridos debe seleccionar MIGE RCE y RVIE - SIRE y seleccionar como alcance Web.



    - Seleccionar el botón Guardar.
    - Al presionar el botón “ACEPTAR”, el sistema genera las credenciales correspondientes:





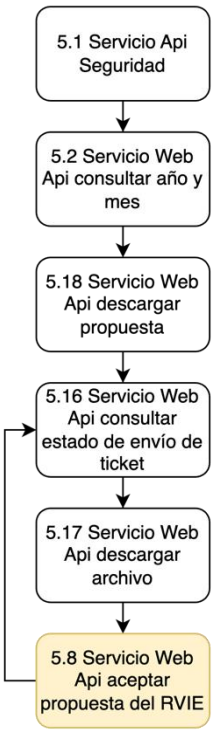
El usuario del SIRE que utiliza los servicios Rest, debe almacenar estos valores para ser utilizado mediante su Sistema de Información.
Una vez que el usuario cuente con los datos del client_id y client_secret además de su cuenta de usuario y clave SOL, podrán generar el token del api-seguridad-SUNAT con la siguiente **url:**
https://api-seguridad.sunat.gob.pe/v1/clientessol/{client_id}/oauth2/token/
Ejemplo:
https://api-seguridad.sunat.gob.pe/v1/clientessol/9cae24a9-10d7-48b0-bee0-e94bd56947e3/oauth2/token/

b) Servicios Necesarios:

- 5.1 Servicio Api Seguridad (ver detalle en el punto 5. Documentación Servicios Web API)

2. Secuencia de servicios mínimos para Registrar Preliminar Funcionalidad 1: Aceptar Propuesta

a) Diagrama: Esquema gráfico de la secuencia de pasos para llegar a consumir el servicio aceptar la propuesta



Nota: Este servicio permite registrar un preliminar del RVIE mediante la aceptación de una propuesta, como resultado se obtiene un ticket asociado al proceso.

Este servicio debe enviar comprobantes de la propuesta en este caso:

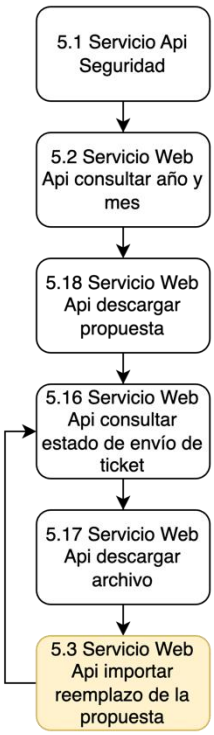
- Se activa el proceso 5 En Generación de registro, etapa: Preliminar registrado
- codTipoRegistro (2 Registro de Ventas)
- Devuelve respuesta (T o F)

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (opcional)
- 5.18 Servicio Web Api descargar propuesta (opcional)
- 5.16 Servicio Web Api consultar estado de envío de ticket (opcional)
- 5.17 Servicio Web Api descargar archivo (opcional)
- 5.8 Servicio Web Api aceptar propuesta del RVIE (necesario)

Funcionalidad 2: Reemplazar Propuesta

a) Diagrama: Esquema gráfico de la secuencia de pasos para llegar a consumir el servicio reemplazar la propuesta



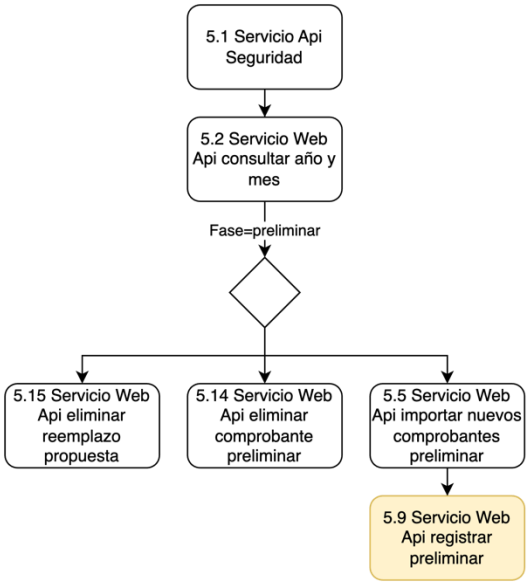
Nota: Servicio web api que permite al generador, reemplazar la propuesta SUNAT con lo considerado por el contribuyente mediante el uso de un archivo de formato .txt zipeado.
Si el estado del generador es "baja definitiva", solo se permitirá actualizar la información correspondiente a los periodos donde estuvo activo o con suspensión temporal (generó y/o fue omiso a la generación del registro), en caso haya generado el registro solo se permitirá la presentación de ajustes posteriores.

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (necesario)
- 5.18 Servicio Web Api descargar propuesta (opcional)
- 5.16 Servicio Web Api consultar estado de envío de ticket (opcional)
- 5.17 Servicio Web Api descargar archivo (opcional)
- 5.3 Servicio Web Api Importar reemplazo de la propuesta (necesario)

Funcionalidad 3: Registrar Preliminar

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio Registrar preliminar



Servicio web api que permitirá al generador registrar el preliminar enviando con esa acción a la opción de generación.
Si el estado del generador es baja definitiva, solo se debe permitir actualizar la información correspondiente a los periodos donde estuvo activo (generó y/o fue omiso a la generación). En caso haya generado el registro solo se permite la presentación de ajustes posteriores.

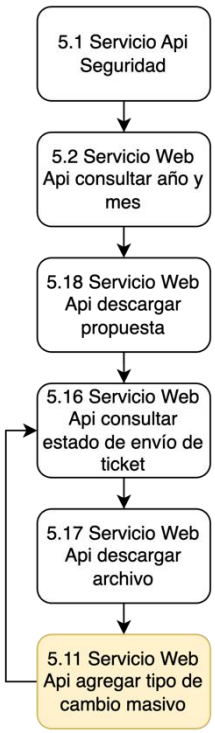
b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (necesario)
- 5.4 Servicio Web Api importar nuevos comprobantes propuesta (opcional)
- 5.13 Servicio Web Api eliminar comprobante propuesta (opcional)
- 5.14 Servicio Web Api eliminar comprobante preliminar (opcional)
- 5.9 Servicio Web Api registrar preliminar (necesario)

3. Secuencia de servicios interdependientes que completan funcionalidades del SIRE Ventas

Funcionalidad 1: Tipo de cambio

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio Tipo de cambio



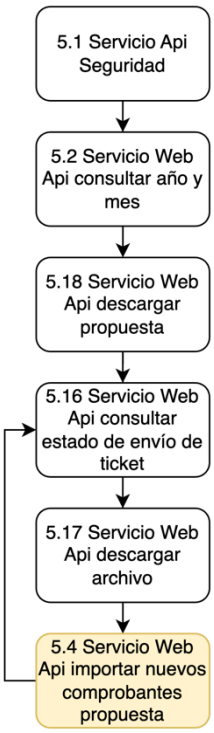
Servicio web api que permite al generador, importar el tipo de cambio en la propuesta.

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (necesario)
- 5.4 Servicio Web Api importar nuevos comprobantes propuesta (opcional)
- 5.11 Servicio Web Api agregar tipo de cambio masivo (necesario)

Funcionalidad 2: Importar comprobantes en propuesta

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio importar comprobantes en propuesta



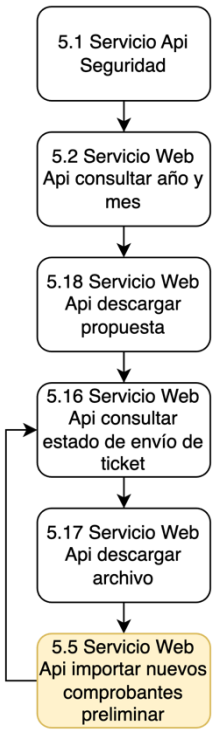
Servicio web api que permite al generador, complementar la propuesta mediante el uso de un archivo de formato .txt zipeado.

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (necesario)
- 5.4 Servicio Web Api importar nuevos comprobantes propuesta (necesario)

Funcionalidad 3: Importar comprobantes en preliminar

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio importar comprobantes en preliminar



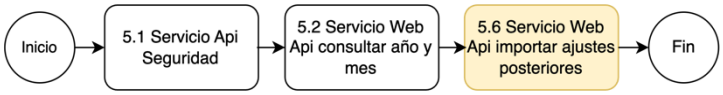
Para poder consumir el servicio importar comprobantes en preliminar, previamente debe haber reemplazado la propuesta.

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (necesario)
- 5.3 Servicio Web Api importar reemplazo de la propuesta (necesario)
- 5.5 Servicio Web Api importar nuevos comprobantes preliminar (necesario)

Funcionalidad 4: Importar ajustes posteriores

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio importar ajustes posteriores



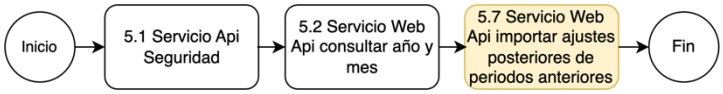
Para poder consumir el servicio importar ajustes posteriores, debe primero haber generado el periodo que desea ajustar.

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (necesario)
- 5.6 Servicio Web Api importar ajustes posteriores (necesario)

Funcionalidad 5: Importar ajustes posteriores de periodos anteriores

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio importar ajustes posteriores de periodos anteriores



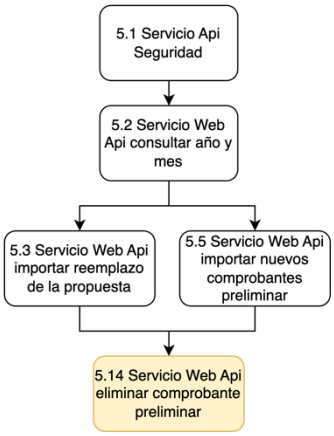
Para poder consumir el servicio importar ajustes posteriores de periodos anteriores, debe hacerlo referenciando al último periodo generado en el SIRE.

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (necesario)
- 5.7 Servicio Web Api importar ajustes posteriores de periodos anteriores (necesario)

Funcionalidad 6: Eliminar comprobantes en preliminar

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio Eliminar comprobantes en preliminar



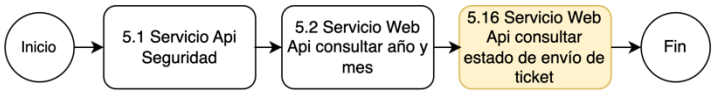
Para poder consumir el servicio eliminar comprobantes preliminar, previamente debe haber reemplazado la propuesta.

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.2 Servicio Web Api consultar año y mes (necesario)
- 5.3 Servicio Web Api importar reemplazo de la propuesta (necesario)
- 5.5 Servicio Web Api importar nuevos comprobantes preliminar (necesario)
- 5.14 Servicio Web Api eliminar comprobante preliminar (necesario)

Funcionalidad 7: Consultar estado de envío de ticket

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio consultar estado de envío de ticket



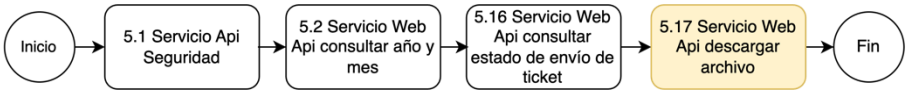
Para poder consumir el servicio consulta de estado de envío de ticket, previamente debe haberse ejecutado al menos un proceso que genere ticket, por ejemplo, aceptar propuesta, reemplazar propuesta, descargar propuesta, entre otros.

b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.8 Servicio Web Api aceptar propuesta del RVIE (al menos 1 debe ejecutarse)
- 5.3 Servicio Web Api importar reemplazo de la propuesta (al menos 1 debe ejecutarse)
- 5.5 Servicio Web Api Importar nuevos comprobantes preliminar (al menos 1 debe ejecutarse)
- 5.6 Servicio Web Api importar ajustes posteriores (al menos 1 debe ejecutarse)
- 5.7 Servicio Web Api importar ajustes posteriores de periodos anteriores (al menos 1 debe ejecutarse)
- 5.18 Servicio Web Api descargar propuesta (al menos 1 debe ejecutarse)
- 5.22 Servicio Web Api exportar preliminar de registro de Ventas (al menos 1 debe ejecutarse)
- 5.23 Servicio Web Api descargar reporte de casillas. (al menos 1 debe ejecutarse)
- 5.24 Servicio Web Api descargar inconsistencias en registros del preliminar registrado. (al menos 1 debe ejecutarse)
- 5.32 Servicio Web Api descargar reporte CAR (al menos 1 debe ejecutarse)
- 5.29 Servicio Web Api descargar ajustes posteriores (al menos 1 debe ejecutarse)
- 5.30 Servicio Web Api descargar ajustes posteriores de periodos anteriores. (al menos 1 debe ejecutarse)
- 5.16 Servicio Web Api consultar estado de envío de ticket. (opcional)
- 5.28 Servicio Web Api descargar reporte consolidado por periodo. (al menos 1 debe ejecutarse)
- 5.27 Servicio Web Api descargar RVIE por periodo. (al menos 1 debe ejecutarse)
- 5.31 Servicio Web Api descargar reporte inconsistencias por periodo. (al menos 1 debe ejecutarse)

Funcionalidad 8: Descargar archivo

a) Diagrama: Esquema gráfico de la ejecución de servicios que mantienen interdependencia para consumir el servicio descargar archivo



Para poder consumir el servicio de descargar archivo", previamente debe haberse ejecutado algún proceso que genere un archivo o más, por ejemplo: "Descargar propuesta", entre otros. Es recomendable verificar el estado del ticket haciendo uso del servicio consultar estado de envío de ticket. El estado debe encontrarse "Terminado".

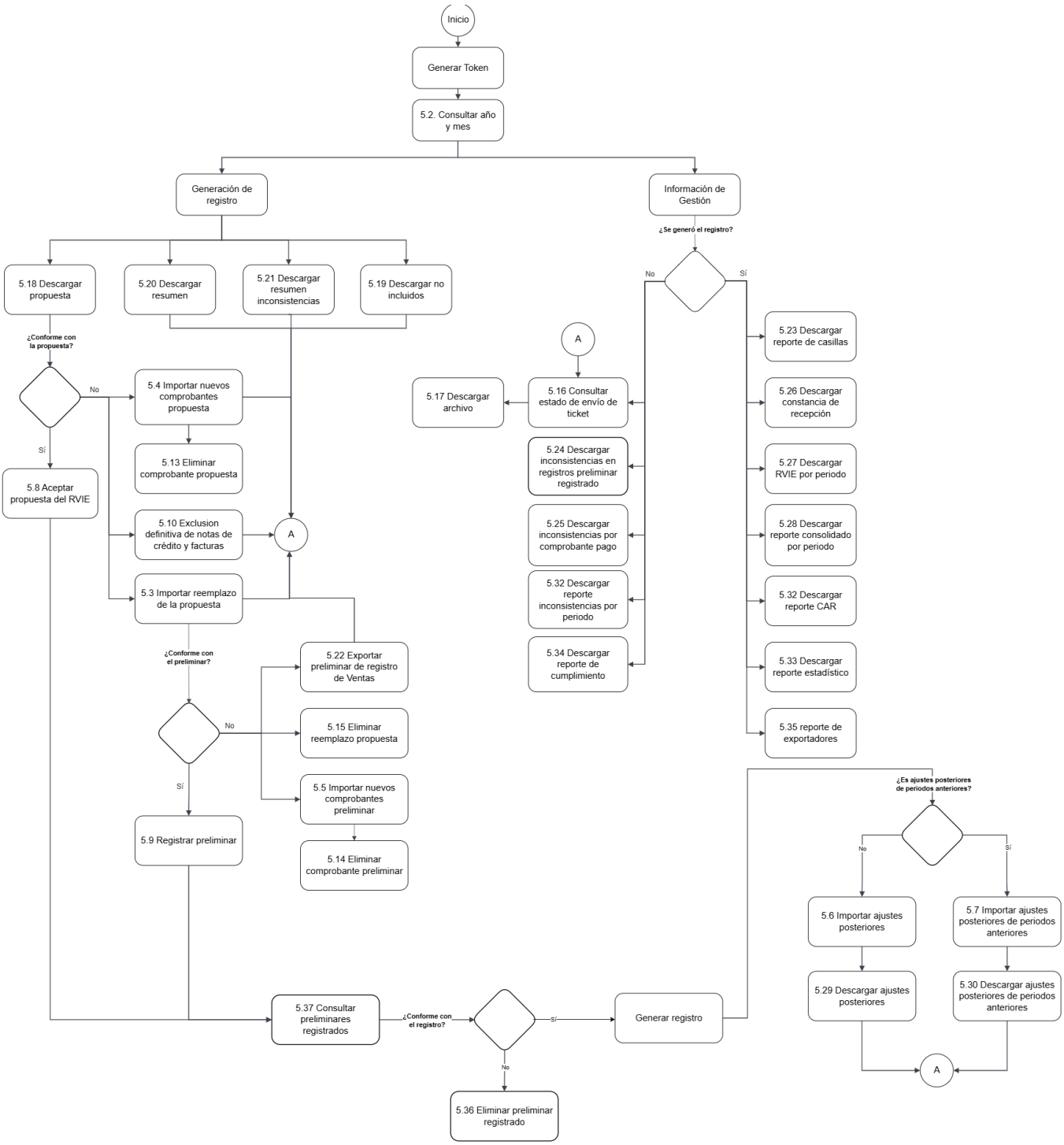
b) Servicios que se pueden invocar:

- 5.1 Servicio Api Seguridad (necesario)
- 5.8 Servicio Web Api aceptar propuesta del RVIE (al menos 1 debe ejecutarse)

- 5.3 Servicio Web Api importar reemplazo de la propuesta (al menos 1 debe ejecutarse)
- 5.5 Servicio Web Api Importar nuevos comprobantes preliminar (al menos 1 debe ejecutarse)
- 5.6 Servicio Web Api importar ajustes posteriores (al menos 1 debe ejecutarse)
- 5.7 Servicio Web Api importar ajustes posteriores de periodos anteriores (al menos 1 debe ejecutarse)
- 5.18 Servicio Web Api descargar propuesta (al menos 1 debe ejecutarse)
- 5.22 Servicio Web Api exportar preliminar de registro de Ventas (al menos 1 debe ejecutarse)
- 5.23 Servicio Web Api descargar reporte de casillas. (al menos 1 debe ejecutarse)
- 5.24 Servicio Web Api descargar inconsistencias en registros del preliminar registrado. (al menos 1 debe ejecutarse)
- 5.32 Servicio Web Api descargar reporte CAR (al menos 1 debe ejecutarse)
- 5.29 Servicio Web Api descargar ajustes posteriores (al menos 1 debe ejecutarse)
- 5.30 Servicio Web Api descargar ajustes posteriores de periodos anteriores. (al menos 1 debe ejecutarse)
- 5.16 Servicio Web Api consultar estado de envío de ticket. (opcional)
- 5.28 Servicio Web Api descargar reporte consolidado por periodo. (al menos 1 debe ejecutarse)
- 5.27 Servicio Web Api descargar RVIE por periodo. (al menos 1 debe ejecutarse)
- 5.31 Servicio Web Api descargar reporte inconsistencias por periodo. (al menos 1 debe ejecutarse)
- 5.33 Servicio Web Api descargar reporte estadístico. (al menos 1 debe ejecutarse)

4. Servicios accesorios que pueden ser consumidos en el SIRE Ventas

a) Diagrama: Esquema gráfico de la secuencia de todos los servicios que SUNAT pone a disposición de los contribuyentes.


Inicio
Generar Token
5.2. Consultar año y mes
Generación de registro
Información de Gestión
¿Se generó el registro?
No
Sí
5.18 Descargar propuesta
5.20 Descargar resumen
5.21 Descargar resumen inconsistencias
5.19 Descargar no incluidos
¿Conforme con la propuesta?
No
Sí
5.4 Importar nuevos comprobantes propuesta
5.13 Eliminar comprobante propuesta
5.8 Aceptar propuesta del RVIE
5.10 Exclusion definitiva de notas de crédito y facturas
A
5.3 Importar reemplazo de la propuesta
¿Conforme con el preliminar?
No
Sí
5.22 Exportar preliminar de registro de Ventas
5.15 Eliminar reemplazo propuesta
5.5 Importar nuevos comprobantes preliminar
5.14 Eliminar comprobante preliminar
5.9 Registrar preliminar
A
5.16 Consultar estado de envío de ticket
5.17 Descargar archivo
5.24 Descargar inconsistencias en registros preliminar registrado
5.25 Descargar inconsistencias por comprobante pago
5.32 Descargar reporte inconsistencias por periodo
5.34 Descargar reporte de cumplimiento
5.23 Descargar reporte de casillas
5.26 Descargar constancia de recepción
5.27 Descargar RVIE por periodo
5.28 Descargar reporte consolidado por periodo
5.32 Descargar reporte CAR
5.33 Descargar reporte estadístico
5.35 reporte de exportadores
5.37 Consultar preliminares registrados
¿Conforme con el registro?
Sí
No
5.36 Eliminar preliminar registrado
Generar registro
¿Es ajustes posteriores de periodos anteriores?
No
Sí
5.6 Importar ajustes posteriores
5.7 Importar ajustes posteriores de periodos anteriores
5.29 Descargar ajustes posteriores
5.30 Descargar ajustes posteriores de periodos anteriores
A

b) Servicios que se pueden invocar (servicios opcionales):

- 5.34 Servicio Web Api descargar reporte de cumplimiento
- 5.26 Servicio Web Api descargar constancia de recepción.
- 5.20 Servicio Web Api descargar resumen.
- 5.21 Servicio Web Api descargar resumen inconsistencias
- 5.23 Servicio Web Api descargar reporte de casillas.
- 5.24 Servicio Web Api descargar inconsistencias en registros preliminar registrado.
- 5.32 Servicio Web Api descargar reporte CAR
- 5.2 Servicio Web Api consultar año y mes
- 5.6 Servicio Web Api importar ajustes Posteriores
- 5.27 Servicio Web Api descargar RVIE por periodo.
- 5.28 Servicio Web Api descargar reporte consolidado por periodo.
- 5.31 Servicio Web Api descargar reporte inconsistencias por periodo.
- 5.33 Servicio Web Api descargar reporte estadístico

5. Documentación Servicios Web API

Importante: los servicios del API SIRE no deben ser consumidos desde un cliente Web, en caso de utilizar un cliente Web se producirá error de CORS. Así mismo los servicios API REST que impliquen el desarrollo de un cliente TUS (Open Protocol for Resumable File Uploads) deben ser desarrollados en el lenguaje JAVA (Ver Anexo 7.5)

5.1 Servicio Api Seguridad

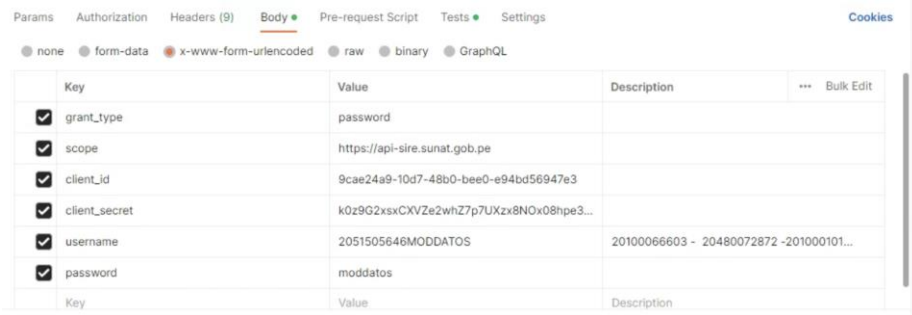
Nombre Web Services	Api Seguridad
Descripción	Permite generar el token para consumo de API's expuestas por SUNAT.
Url	https://api-seguridad.sunat.gob.pe/v1/clientessol/9cae24a9-10d7-48b0-bee0-e94bd56947e3/oauth2/token/
Parámetros[body]	**Descripción:** grant_type: password (credenciales del cliente - usar por defecto: password) scope: https://api-sire.sunat.gob.pe (uri que permitirá el acceso con el token - por defecto: https://api-sire.sunat.gob.pe) client_id: abababababa-9abc-453s-s5s4s4-s457555 (se obtiene desde la opción Credenciales de API SUNAT) client_secret: CSCSSKSJDSKSNSKSKSSKSJDDN (se obtiene desde la opción Credenciales de API SUNAT) username: {RUC} {USUARIO} (RUC y Usuario del generador) password: {CLAVESOL} (Clave sol del generador)
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded (opcional) Método: POST
Evidencias	**URL**

https://api-seguridad.sunat.gob.pe/v1/clientessol/9cae24a9-10d7-48b0-bee0-e94bd56947e3/oauth2/token/

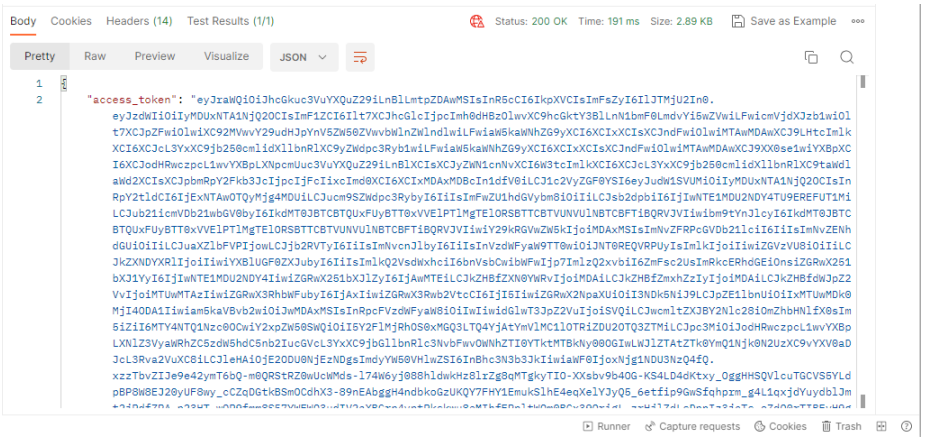
Headers

(No aplica)

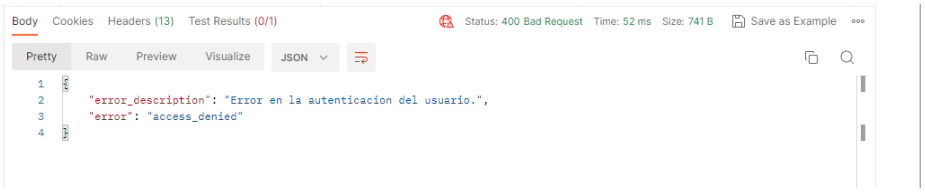
Body



Result OK



Result Fail



5.2 Servicio Web Api consultar año y mes

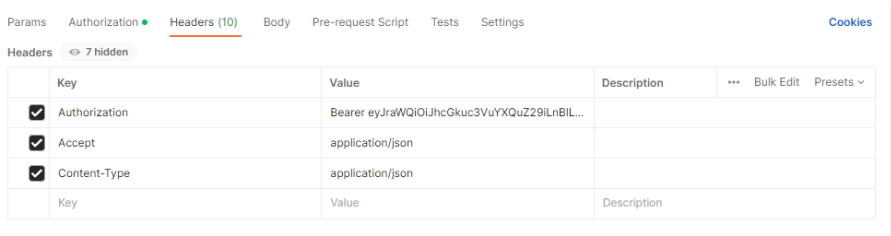
Nombre Web Services	Servicio Web Api que consulta años y meses de RVIE.
Descripción	Permite consultar los periodos (años y meses) habilitados para el contribuyente.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/padron/web/omisos/{codLibro}/periodos
Parámetros[URL]	**Param-formato-tipo**: codLibro-alfanumérico-String **Descripción**: Código de libro: 140000 RVIE (Obligatorio)
Parámetros[body]	No aplica
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Parámetros: Content-Type — valor: application/json Parámetros: Accept — valor: application/json Parámetros: Authorization — valor: Bearer token obtenido de la autenticación Método: GET
Parámetros[salida]	**Parámetros de Salida** / **Descripción**

Object- Object- Object	**Object inicio**
numEjercicio-alfanumérico-String	Año o ejericicio
desEstado-alfanumérico-String	Descripcion del ejercicio
lisPeriodos-array-array	**Array lista de periodos - inicio**
perTributario-alfanumérico-String	Periodo tributario
codEstado-alfanumérico-String	Código del estado del periodo tributario
desEstado-alfanumérico-String	Descripcion del estado del periodo tributario
lisPeriodos-array-array	**Array lista de periodos - fin**
Object- Object- Object	**Object fin**

Evidencias

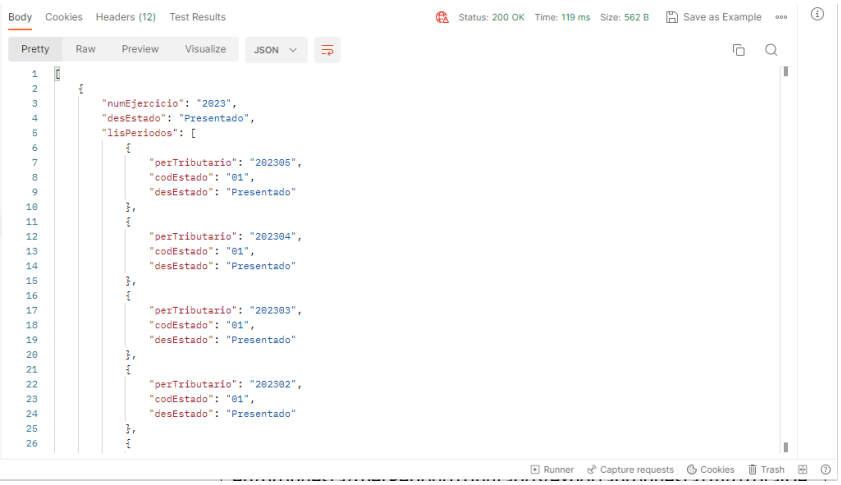
URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/padron/web/omisos/140000/periodos

Headers



Body
(No aplica)

Result OK



Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

Lista de errores 422:

- 1140 - El campo “codLibro” no enviado o es vacío

5.3 Servicio Web Api importar reemplazo de la propuesta

Nombre Web Services	Servicio Web Api importar reemplazo de la propuesta
Descripción	Servicio web api que permite al generador, reemplazar la propuesta SUNAT con lo considerado por el contribuyente mediante el uso de un archivo de formato .txt
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorpropuesta/web/propuesta/upload
Parámetros[body]	No aplica
Parámetros[header]	**Valores:**

Content-type: application/x-www-form-urlencoded

Metadata Cliente TUS

Param-formato-tipo	Descripción
filename-alfanumérico-String	Nombre de archivo (Obligatorio)
filetype-alfanumérico-String	Tipo de archivo (Obligatorio)
perTributario-alfanumérico-String	Periodo tributario (Obligatorio)
codOrigenEnvio-alfanumérico-String	Código de origen de envío: 2 Servicio web (Obligatorio)
codProceso-alfanumérico-String	Código del indicador de carga masiva: 3.Reemplazo de la Propuesta (Ver Anexo I: Indicador de carga masiva) (1 - 97) (Obligatorio)
codTipoCorrelativo-alfanumérico-String	Tipo de correlativo: 01: Tipo envíos masivos (Ver Anexo II: Tipo de correlativo) (Obligatorio)
nomArchivoImportacion-alfanumérico-String	Nombre del archivo utilizado para la importación o nombre de archivo generado, definido en la tabla 6 del Anexo N° 1 de la Resolución de Superintendencia 112-2021/SUNAT, estructuras e información del registro electrónico - RVIE, la estructura es la siguiente: LERRRRRRRRRRRAAAAMM0014040002OIM2.txt. (Obligatorio)
codLibro-alfanumérico-String	Código de libro: 140000 RVIE (Obligatorio)

Parámetros[salida]

Parámetros de Salida	Descripción	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

Evidencias

URL

https://api-sire.sunat.gob.pe /v1/contribuyente/migeigv/libros/rvierce/receptorpropuesta/web/propuesta/upload

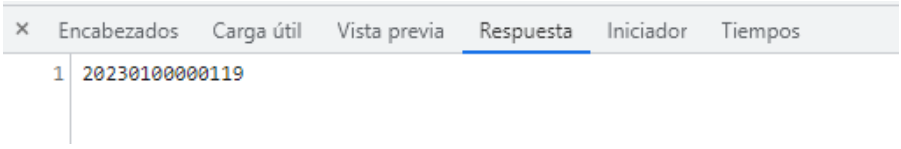
Headers (metadata)

filename TEUyMDEwMDE3NjQ1MDIwMjMwMjAwMTQwNDAwMDIwMTEyLnppcA==,filetype YXBwbGljYXRpb24vemlw,numRuc MjAxMDAxNzY0NTA=,perTributario MjAyMzAy,codOrigenEnvio MQ==,codProceso Mw==,codTipoCorrelativo MQ==,nomArchivoImportacion TEUyMDEwMDE3NjQ1MDIwMjMwMjAwMTQwNDAwMDIwMTEyLnppcA==,codLibro MTQwMDAw

Body

(No aplica)

Result OK



Result Fail

{"cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

{"cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

Lista de errores 422:

- 1001 - El campo “numRuc” no enviado o es vacío
- 1002 - Solo se permite dato numérico de 11 dígitos para el número de RUC.
- 1003 - El RUC ingresado no existe o no es válido
- 1005 - El campo ‘perTributario’ no enviado o es vacio
- 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
- 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
- 1028 - El campo “codOrigenEnvio” no enviado o es vacío
- 1029 - Código tipo de Origen de Envio no permitido o no valido
- 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio
- 1025 - El campo “codProceso” no enviado o es vacío

	• 1026 - Código Proceso no permitido o no valido • 1027 - Solo se permite dato numérico para el codProceso • 1138 - El campo "codProceso" es nulo o vacío • 1139 - Código de Proceso no permitido o no valido • 1048 - Solo se permite dato numérico de 1 dígito para el codTipoOrigen • 1022 - nombre del archivo no enviado o es vacio. • 1024 - El archivo <nombre del archivo txt> fue previamente enviado. • 1044 - Error en la <<Posición - Descripción>> del nombre del archivo plano, favor de corregir • 1348 - La extensión del archivo es diferente a ".zip", por favor corregir • 1346 - El tamaño del archivo comprimido en formato ".zip" debe ser menor o igual a 6GB. • 1350 - El tamaño del archivo mayor a 0 Kb. • 1351 - Se ha producido un error al realizar el envío del archivo, por favor volver a intentar el envío • 1048 - Solo se permite dato numérico de 2 dígitos para el codTipoCorrelativo • 1049 - El campo "codTipoCorrelativo" no enviado o es vacio • 1050 - Código tipo de Correlativo no permitido o no valido • 1140 - El campo "codLibro" no enviado o es vacío
Tecnología	Uso del protocolo TUS.IO (Ver ítem 6. Documentación TUS.IO)

5.4 Servicio Web Api importar nuevos comprobantes propuesta

Nombre Web Services	Servicio Web Api importar nuevos comprobantes en propuesta
Descripción	Servicio web api que permite al generador, agregar nuevos comprobantes que no han sido propuestos por la administración.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorpropuesta/web/propuesta/upload
Parámetros[body]	No aplica
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Metadata Cliente TUS**

Param-formato-tipo	***Descripción***
filename-alfanumérico-String	Nombre de archivo (Obligatorio)
filetype-alfanumérico-String	Tipo de archivo (Obligatorio)
perTributario-alfanumérico-String	Periodo tributario (Obligatorio)
codOrigenEnvio-alfanumérico-String	Código de origen de envío: 2 Servicio web (Obligatorio)
codProceso-alfanumérico-String	Código del indicador de carga masiva: 1.Importar CP (Ver Anexo I: Indicador de carga masiva) (1 - 97) (Obligatorio)
codTipoCorrelativo-alfanumérico-String	Tipo de correlativo: 01: Tipo envíos masivos (Ver Anexo II: Tipo de correlativo) (Obligatorio)
nomArchivoImportacion-alfanumérico-String	Nombre del archivo utilizado para la importación o nombre de archivo generado, definido en la tabla 6 del Anexo N° 1 de la Resolución de Superintendencia 112-2021/SUNAT, estructura e información del archivo texto para complementar la propuesta del RVIE con comprobantes de pago físicos, la estructura es la siguiente: RRRRRRRRRRR-CPF-AAAAMM-Correlativo.txt (Obligatorio)
codLibro-alfanumérico-String	Código de libro: 140000 RVIE (Obligatorio)

Parámetros[salida]	**Parámetros de Salida**	**Descripción**	**Formato**	**Tipo dato**

	numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

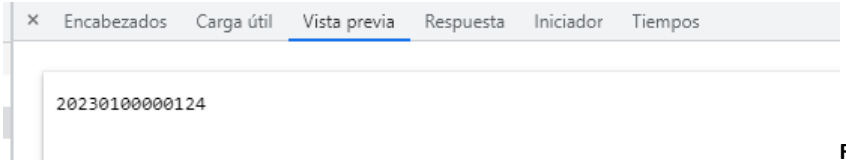
Evidencias

URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorpropuesta/web/propuesta/upload

Headers (metadata)
filename MjAxMDAxNzY0NTAtQ1BGLTIwMjMwMi0wMS56aXA=,filetype YXBwbGljYXRpb24vemlw,numRuc MjAxMDAxNzY0NTA=,perTributario MjAyMzAy,codOrigenEnvio MQ==,codProceso MQ==,codTipoCorrelativo MQ==,nomArchivoImportacion MjAxMDAxNzY0NTAtQ1BGLTIwMjMwMi0wMS56aXA=,codLibro MTQwMDAw

Body
(No aplica)

Result OK



Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

Lista de errores 422:

- 1001 - El campo “numRuc” no enviado o es vacío
- 1002 - Solo se permite dato numérico de 11 dígitos para el número de RUC.
- 1003 - El RUC ingresado no existe o no es válido
- 1005 - El campo ‘perTributario’ no enviado o es vacio
- 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
- 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
- 1028 - El campo “codOrigenEnvio” no enviado o es vacío
- 1029 - Código tipo de Origen de Envio no permitido o no valido
- 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio
- 1025 - El campo “codProceso” no enviado o es vacío
- 1026 - Código Proceso no permitido o no valido
- 1027 - Solo se permite dato numérico para el codProceso
- 1138 - El campo "codProceso" es nulo o vacío
- 1139 - Código de Proceso no permitido o no valido
- 1048 - Solo se permite dato numérico de 1 dígito para el codTipoOrigen
- 1022 - nombre del archivo no enviado o es vacio.
- 1024 - El archivo <nombre del archivo txt> fue previamente enviado.
- 1044 - Error en la <<Posición - Descripción>> del nombre del archivo plano, favor de corregir
- 1348 - La extensión del archivo es diferente a “.zip”, por favor corregir
- 1346 - El tamaño del archivo comprimido en formato “.zip” debe ser menor o igual a 6GB.
- 1350 - El tamaño del archivo mayor a 0 Kb.
- 1351 - Se ha producido un error al realizar el envío del archivo, por favor volver a intentar el envío
- 1048 - Solo se permite dato numérico de 2 dígitos para el codTipoCorrelativo
- 1049 - El campo “codTipoCorrelativo” no enviado o es vacio
- 1050 - Código tipo de Correlativo no permitido o no valido
- 1140 - El campo “codLibro” no enviado o es vacío

Tecnología

Uso del protocolo TUS.IO (Ver ítem 6. Documentación TUS.IO)

5.5 Servicio Web Api importar nuevos comprobantes preliminar

<table>
<tr><td>Nombre Web Services</td><td>Servicio Web Api importar nuevos comprobantes en el preliminar</td></tr>
<tr><td>Descripción</td><td>Permite importar nuevos comprobantes en el preliminar de RVIE.</td></tr>
<tr><td>Url</td><td>https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorpreliminar/web/preliminar/upload</td></tr>
<tr><td>Parámetros[header]</td><td>Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded
Metadata Cliente TUS
<table>
<tr><th>Param-formato-tipo</th><th>Descripción</th></tr>
<tr><td>filename-alfanumérico-String</td><td>Nombre de archivo (Obligatorio)</td></tr>
<tr><td>filetype-alfanumérico-String</td><td>Tipo de archivo (Obligatorio)</td></tr>
<tr><td>perTributario-alfanumérico-String</td><td>Periodo tributario (Obligatorio)</td></tr>
<tr><td>codOrigenEnvio-alfanumérico-String</td><td>Código de origen de envío: 2 Servicio web (Obligatorio)</td></tr>
<tr><td>codProceso-alfanumérico-String</td><td>Código del indicador de carga masiva: 4. Importa CP - Preliminar (Ver Anexo I: Indicador de carga masiva) . (1-97) (Obligatorio)</td></tr>
<tr><td>codTipoCorrelativo-alfanumérico-String</td><td>Tipo de correlativo: 01: Tipo envíos masivos (Ver Anexo II: Tipo de correlativo) (Obligatorio)</td></tr>
<tr><td>nomArchivoImportacion-alfanumérico-String</td><td>Nombre del archivo utilizado para la importación o nombre de archivo generado, definido en la tabla 6 del Anexo N° 1 de la Resolución de Superintendencia 112-2021/SUNAT, estructuras e información del registro electrónico - RVIE, la estructura es la siguiente: LERRRRRRRRRRRAAAAMM0014040002OIM2.txt. (Obligatorio)</td></tr>
<tr><td>codLibro-alfanumérico-String</td><td>Código de libro: 140000 RVIE (Obligatorio)</td></tr>
</table></td></tr>
<tr><td>Parámetros[salida]</td><td>
<table>
<tr><th>Parámetros de Salida</th><th>Descripción</th><th>Formato</th><th>Tipo dato</th></tr>
<tr><td>numTicket</td><td>Número de ticket de envío
[AAAA99999999]
AAAA: Año
99: Tipo de correlativo
99999999: Número correlativo de envío</td><td>alfanumerico</td><td>String</td></tr>
</table></td></tr>
<tr><td>Evidencias</td><td>URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorpreliminar/web/preliminar/upload
Headers (metadata)
filename TEUyMDEwMDE3NjQ1MDIwMjMwMjAwMTQwNDAwMDIwMTEyLnppcA==,filetype YXBwbGljYXRpb24vemlw,numRuc MjAxMDAxNzY0NTA=,perTributario MjAyMzAy,codOrigenEnvio MQ==,codProceso NA==,codTipoCorrelativo MQ==,nomArchivoImportacion TEUyMDEwMDE3NjQ1MDIwMjMwMjAwMTQwNDAwMDIwMTEyLnppcA==,codLibro MTQwMDAw
Body
(No aplica)
Result OK
× Encabezados Carga útil Vista previa Respuesta Iniciador Tiempos
1 20230100000123
Result Fail</td></tr>
</table>

	{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo "numRuc" no enviado o es vacío" }] } **Lista de errores:** ● 1001 - El campo "numRuc" no enviado o es vacío ● 1002 - Solo se permite dato numérico de 11 dígitos para el número de RUC. ● 1003 - El RUC ingresado no existe o no es válido ● 1005 - El campo 'perTributario' no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato 'yyyymm' ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1028 - El campo "codOrigenEnvio" no enviado o es vacío ● 1029 - Código tipo de Origen de Envio no permitido o no valido ● 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio ● 1025 - El campo "codProceso" no enviado o es vacío ● 1026 - Código Proceso no permitido o no valido ● 1027 - Solo se permite dato numérico para el codProceso ● 1138 - El campo "codProceso" es nulo o vacío ● 1139 - Código de Proceso no permitido o no valido ● 1048 - Solo se permite dato numérico de 1 dígito para el codTipoOrigen ● 1022 - nombre del archivo no enviado o es vacio. ● 1024 - El archivo <nombre del archivo txt> fue previamente enviado. ● 1044 - Error en la <<Posición - Descripción>> del nombre del archivo plano, favor de corregir ● 1348 - La extensión del archivo es diferente a ".zip", por favor corregir ● 1346 - El tamaño del archivo comprimido en formato ".zip" debe ser menor o igual a 6GB. ● 1350 - El tamaño del archivo mayor a 0 Kb. ● 1351 - Se ha producido un error al realizar el envío del archivo, por favor volver a intentar el envío ● 1048 - Solo se permite dato numérico de 2 dígitos para el codTipoCorrelativo ● 1049 - El campo "codTipoCorrelativo" no enviado o es vacio ● 1050 - Código tipo de Correlativo no permitido o no valido ● 1140 - El campo "codLibro" no enviado o es vacío
Tecnología	Uso del protocolo TUS.IO (Ver ítem 6. Documentación TUS.IO)

5.6 Servicio Web Api importar ajustes posteriores

Nombre Web Services	Servicio Web Api importar ajustes posteriores
Descripción	Cargar ajustes posteriores SIRE
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorajustesposteriores/web/ajustesposteriores/upload
Parámetros[body]	No Aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Metadata Cliente TUS**

Param-formato-tipo	Descripción
filename-alfanumérico-String	Nombre de archivo (Obligatorio)
filetype-alfanumérico-String	Tipo de archivo (Obligatorio)
perTributario-alfanumérico-String	Periodo tributario (Obligatorio)
codOrigenEnvio-alfanumérico-String	Código de origen de envío: 2 Servicio web (Obligatorio)
codProceso-alfanumérico-String	Código del indicador de carga masiva: 6.Cargar Ajuste posteriores del SIRE (Ver Anexo I: Indicador de carga masiva) (1-97) (Obligatorio)
codTipoCorrelativo-alfanumérico-String	Tipo de correlativo: 01: Tipo envíos masivos (Ver Anexo II: Tipo de correlativo) (Obligatorio)

	nomArchivoImportacion-alfanumérico-String: Nombre del archivo utilizado para la importación o nombre de archivo generado, definido en la tabla 6 del Anexo N° 1 de la Resolución de Superintendencia 112-2021/SUNAT, estructuras e información del registro electrónico - RVIE, la estructura dependerá de la descripción consignada. (Obligatorio) codLibro-alfanumérico-String: Código de libro: 140000 RVIE (Obligatorio)

Parámetros[salida]

Parámetros de Salida	Descripción	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

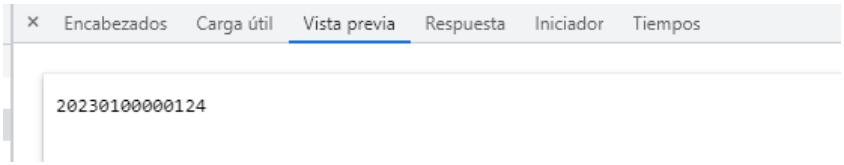
Evidencias

URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorajustesposteriores/web/ajustesposteriores/upload

Headers (metadata)
filename TEUyMDEwMDE3NjQ1MDIwMjMwMjAwMTQwNDAwMDMxMTEwMi56aXA=,filetype YXBwbGljYXRpb24vemlw,numRuc MjAxMDAxNzY0NTA=,perTributario MjAyMzAy,codOrigenEnvio MQ==,codProceso ODc=,codTipoCorrelativo MQ==,nomArchivoImportacion TEUyMDEwMDE3NjQ1MDIwMjMwMjAwMTQwNDAwMDMxMTEwMi56aXA=,codLibro MTQwMDAw

Body
(No aplica)

Result OK



Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

Lista de errores 422:

- 1001 - El campo “numRuc” no enviado o es vacío
- 1002 - Solo se permite dato numérico de 11 dígitos para el número de RUC.
- 1003 - El RUC ingresado no existe o no es válido
- 1005 - El campo ‘perTributario’ no enviado o es vacio
- 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
- 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
- 1028 - El campo “codOrigenEnvio” no enviado o es vacío
- 1029 - Código tipo de Origen de Envio no permitido o no valido
- 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio
- 1025 - El campo “codProceso” no enviado o es vacío
- 1026 - Código Proceso no permitido o no valido
- 1027 - Solo se permite dato numérico para el codProceso
- 1138 - El campo "codProceso" es nulo o vacío
- 1139 - Código de Proceso no permitido o no valido
- 1048 - Solo se permite dato numérico de 1 dígito para el codTipoOrigen
- 1022 - nombre del archivo no enviado o es vacio.
- 1024 - El archivo <nombre del archivo txt> fue previamente enviado.
- 1044 - Error en la <<Posición - Descripción>> del nombre del archivo plano, favor de corregir
- 1348 - La extensión del archivo es diferente a “.zip”, por favor corregir

	• 1346 - El tamaño del archivo comprimido en formato ".zip" debe ser menor o igual a 6GB. • 1350 - El tamaño del archivo mayor a 0 Kb. • 1351 - Se ha producido un error al realizar el envío del archivo, por favor volver a intentar el envío • 1048 - Solo se permite dato numérico de 2 dígitos para el codTipoCorrelativo • 1049 - El campo "codTipoCorrelativo" no enviado o es vacio • 1050 - Código tipo de Correlativo no permitido o no valido • 1140 - El campo "codLibro" no enviado o es vacío
Tecnología	Uso del protocolo TUS.IO (Ver ítem 6. Documentación TUS.IO)

5.7 Servicio Web Api importar ajustes posteriores de periodos anteriores

Nombre Web Services	Servicio Web Api importar ajustes posteriores de periodos anteriores
Descripción	Cargar Ajustes posteriores anteriores de periodos anteriores al SIRE
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorajustesposteriores/web/ajustesposteriores/upload
Parámetros[body]	No Aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Metadata Cliente TUS**

Param-formato-tipo	Descripción
filename-alfanumérico-String	Nombre de archivo (Obligatorio)
filetype-alfanumérico-String	Tipo de archivo (Obligatorio)
perTributario-alfanumérico-String	Periodo tributario (Obligatorio)
codOrigenEnvio-alfanumérico-String	Código de origen de envío: 2 Servicio web (Obligatorio)
codProceso-alfanumérico-String	Código del indicador de carga masiva: 7. Cargar Ajuste posteriores anteriores a la vigencia (Ver Anexo I: Indicador de carga masiva) (1-97) (Obligatorio)
codTipoCorrelativo-alfanumérico-String	Tipo de correlativo: 01: Tipo envíos masivos (Ver Anexo II: Tipo de correlativo) (Obligatorio)
nomArchivoImportacion-alfanumérico-String	Nombre del archivo utilizado para la importación o nombre de archivo generado, definido en la tabla 6 del Anexo N° 1 de la Resolución de Superintendencia 112-2021/SUNAT, estructuras e información del registro electrónico - RVIE, la estructura dependerá de la descripción consignada. (Obligatorio)
codLibro-alfanumérico-String	Código de libro: 140000 RVIE (Obligatorio)

Parámetros[salida]

Parámetros de Salida	Descripción	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

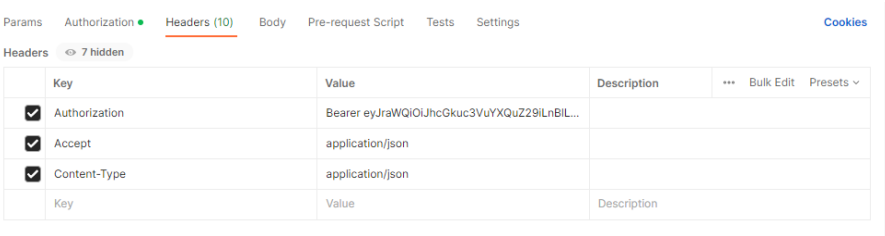
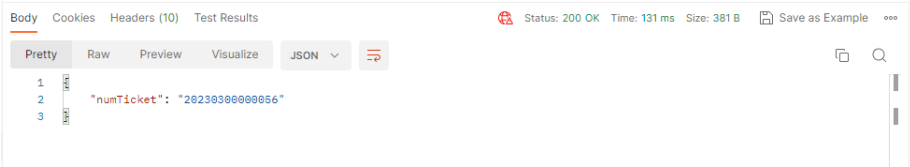
Evidencias

URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/receptorajustesposteriores/web/ajustesposteriores/upload
Headers (metadata)
filename TEUyMDEwMDE3NjQ1MDIwMjMwMzAwMTQwNDAwMDQxMTEyMDIuemlw,filetype YXBwbGljYXRpb24vemlw,numRuc MjAxMDAxNzY0NTA=,perTributario

	MjAyMzAz,codOrigenEnvio MQ==,codProceso ODg=,codTipoCorrelativo MQ==,nomArchivoImportacion TEUyMDEwMDE3NjQ1MDIwMjMwMzAwMTQwNDAwMDQxMTEyMDIuemlw,codLibro MTQwMDAw **Body** (No aplica) **Result OK** × Encabezados Carga útil Vista previa Respuesta Iniciador Tiempos 20230100000124 **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo "numRuc" no enviado o es vacío" }] } **Lista de errores 422:** ● 1001 - El campo "numRuc" no enviado o es vacío ● 1002 - Solo se permite dato numérico de 11 dígitos para el número de RUC. ● 1003 - El RUC ingresado no existe o no es válido ● 1005 - El campo 'perTributario' no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato 'yyyymm' ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1028 - El campo "codOrigenEnvio" no enviado o es vacío ● 1029 - Código tipo de Origen de Envio no permitido o no valido ● 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio ● 1025 - El campo "codProceso" no enviado o es vacío ● 1026 - Código Proceso no permitido o no valido ● 1027 - Solo se permite dato numérico para el codProceso ● 1138 - El campo "codProceso" es nulo o vacío ● 1139 - Código de Proceso no permitido o no valido ● 1048 - Solo se permite dato numérico de 1 dígito para el codTipoOrigen ● 1022 - nombre del archivo no enviado o es vacio. ● 1024 - El archivo <nombre del archivo txt> fue previamente enviado. ● 1044 - Error en la <<Posición - Descripción>> del nombre del archivo plano, favor de corregir ● 1348 - La extensión del archivo es diferente a ".zip", por favor corregir ● 1346 - El tamaño del archivo comprimido en formato ".zip" debe ser menor o igual a 6GB. ● 1350 - El tamaño del archivo mayor a 0 Kb. ● 1351 - Se ha producido un error al realizar el envío del archivo, por favor volver a intentar el envío ● 1048 - Solo se permite dato numérico de 2 dígitos para el codTipoCorrelativo ● 1049 - El campo "codTipoCorrelativo" no enviado o es vacio ● 1050 - Código tipo de Correlativo no permitido o no valido ● 1140 - El campo "codLibro" no enviado o es vacío
Tecnología	Uso del protocolo TUS.IO (Ver ítem 6. Documentación TUS.IO)

5.8 Servicio Web Api aceptar propuesta del RVIE

Nombre Web Services	Servicio Web Api aceptar propuesta del RVIE
Descripción	Actualiza el estado del registro libro y Control de procesos para indicar que se está registrando un preliminar a través de la propuesta aceptada.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/{perTributario}/aceptapropuesta
Parámetros[URL]	**Param-formato-tipo** | **Descripción**

	perTributario-alfanumérico-String | Periodo tributario (Obligatorio)
Parámetros[body]	**Descripción** No aplica
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** **Parámetros** | **Valor** Content-Type | application/json Accept | application/json Authorization | Bearer token obtenido de la autenticación **Método**: POST
Parámetros[salida]	**Parámetros de Salida** | **Descripcion** | **Formato** | **Tipo dato** numTicket | Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío | alfanumerico | String
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/202301/aceptapropuesta **Headers**  **Body** (No aplica) **Result OK**  **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** • 1005 - El campo ‘perTributario’ no enviado o es vacio • 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ • 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual

5.9 Servicio Web Api registrar preliminar

Nombre Web Services	Servicio Web Api Registrar Preliminar
Descripción	Permite registrar los preliminares del registro de Ventas y ajustes posteriores y pueda continuar con la Generación en el portal WEB de SUNAT.
Url	https://api-sire.sunat.gob.pe /v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/{perTributario}/registrapreliminar
Parámetros[URL]	**Param-formato-tipo** | **Descripción** perTributario-alfanumérico-String | Periodo tributario (Obligatorio)

Parámetros[body]	No aplica
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: POST

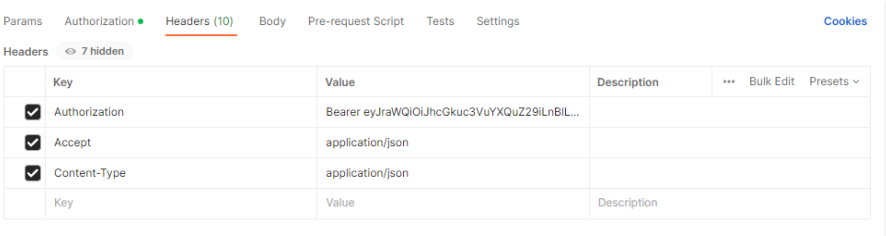
Parámetros[salida]

Parámetros de Salida	Descripción	Formato	Tipo dato
numTicket(Opcional)	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío **Nota**: En caso la petición devuelva número de ticket se debe consultar su estado en 5.16 Servicio Web Api consultar estado de envío de ticket	alfanumerico	String

Evidencias

URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/202302/registrapreliminar

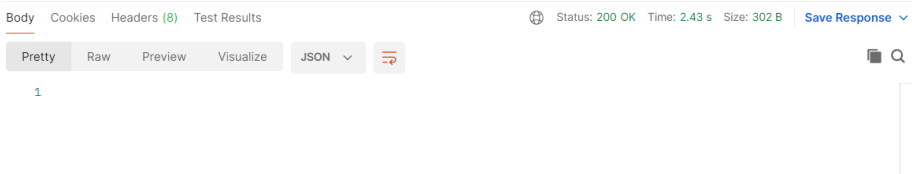
Headers



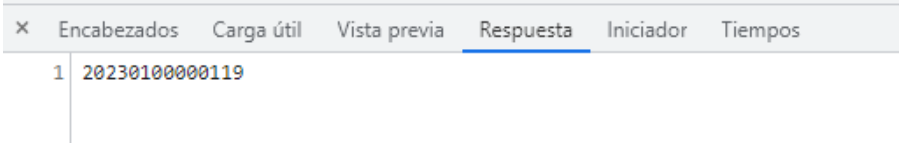
Body
(No aplica)

Result OK

1. Si la respuesta es vacío, el proceso ha terminado correctamente.



2. Si la respuesta es un número de ticket, debe consultar su estado en 5.16 Servicio Web Api consultar estado de envío de ticket, en la cual deberá tener el estado 06 – Terminado.



Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

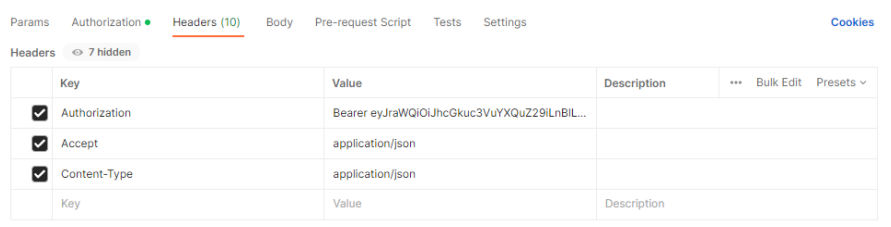
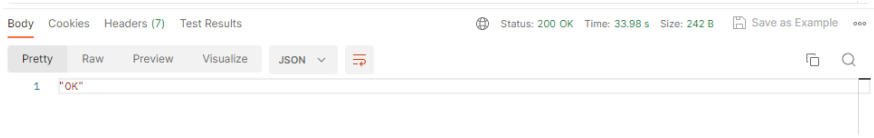
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

Lista de errores 422:

- 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’

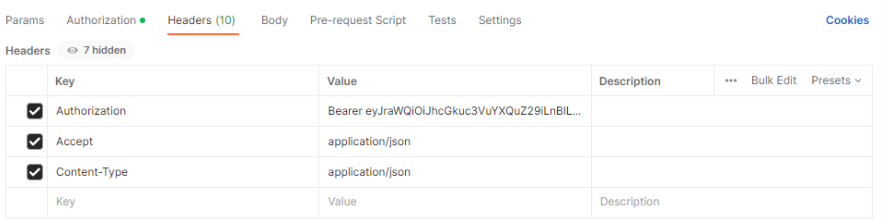
	• 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual • 2293 - No es posible registrar su preliminar, debido a que se encuentra en etapa propuesta, para registrar preliminar primero debe realizar el reemplazo de la propuesta. • 2294 - No es posible registrar su preliminar, debido a que se encuentra en etapa preliminar registrado. • 2295 - No es posible registrar su preliminar, debido a que ya generó su registro desde portal.

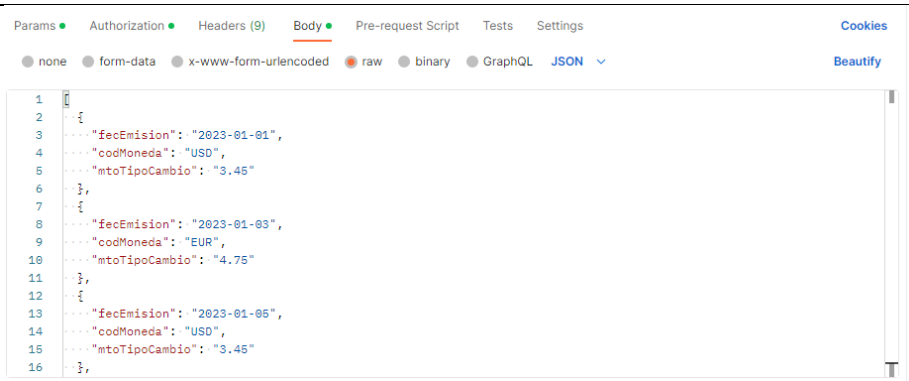
5.10 Servicio Web Api exclusión definitiva de notas de crédito y facturas

Nombre Web Services	Servicio Web Api exclusión definitiva de notas de crédito y facturas
Descripción	Permite la exclusión de las notas de crédito y facturas de manera definitiva e irreversible en la propuesta.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/{perTributario}/retiracomprobante?codCar={codCar}&codSituacion={codSituacion}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio) codCar-alfanumérico-String / Código de Anotación de Registro (CAR SUNAT) (Obligatorio) codSituacion-alfanumérico-String / Código de situación: 0 inactivo, 1 activo (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: POST
Parámetros[salida]	**Parámetros** / **valor** HTTP status / 200 Content-Type / application/json
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/202302/retiracomprobante?codCar=2013729131301FD880000001007&codSituacion=0 **Headers**  **Body** (No aplica) **Result OK**  **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

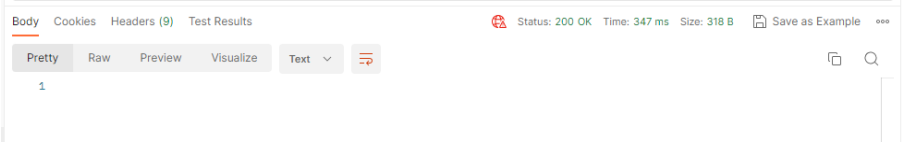
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores:** • 1005 - El campo 'perTributario' no enviado o es vacio • 1006 - Formato de perTributario no cumple con el formato 'yyyymm' • 1135 - El campo “codCar” no enviado o es vacío • 1136 - Solo se permite dato numérico de 27 dígitos para el Codigo CAR. • 1120 - Solo se permite dato numérico de 1 dígito para el codSituacion

5.11 Servicio Web Api agregar tipo de cambio masivo

Nombre Web Services	Servicio Web API agregar tipo de cambio masivo
Descripción	Permite actualizar masivamente todos los tipos de cambio de comprobantes que la administración no encontró tipo de cambio propuesto, de la misma manera los montos propuestos serán actualizados utilizando el o los tipos de cambio ingresados.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/masivo/{perTributario}/guardacomplementomasivo
Parámetros[url]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio)
Parámetros[body]	**Param-formato-tipo** / **Descripción** **Array** / **Array - inicio** fecEmision-dd/mm/yyyy-String / Fecha de emisión del documento (Obligatorio) codMoneda-numérico-String / Códigos de moneda (Obligatorio) mtoTipoCambio-decimal-String / Tipo de cambio de PEN (Soles) a USD (Dólares) (Obligatorio) mtoCambioMonedaExt-decimal-String / Tipo de cambio de moneda extranjera a PEN (Soles) (Opcional: Para conversión de una moneda extranjera a soles.) **Array** / **Array - fin**
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: POST
Parámetros[salida]	**Parámetros** / **Valor** HTTP status / 200 Content-Type / application/json
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/masivo/202301/guardacomplementomasivo **Headers**  **Body**

<table>
<tr><td></td><td>


Result OK



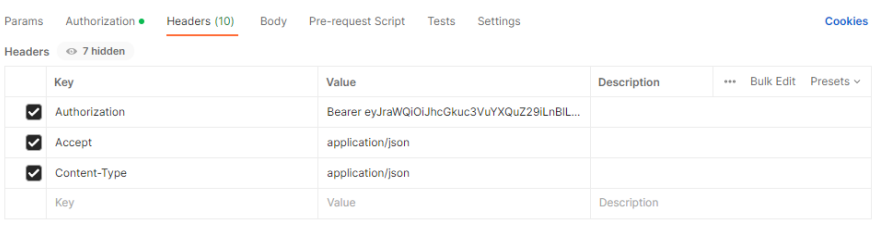
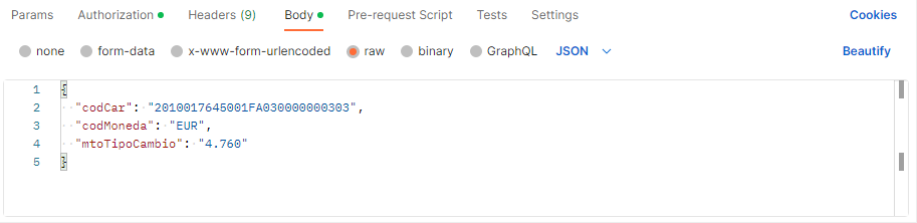
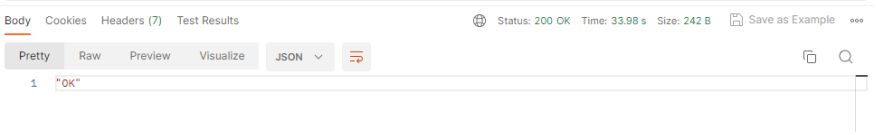
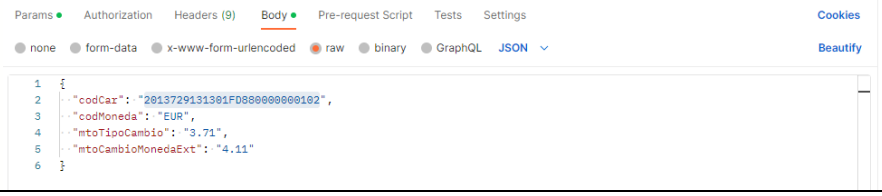
Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
</td></tr>
<tr><td>Mensaje Error</td><td>
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:

1005 - El campo ‘perTributario’ no enviado o es vacio
1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
1141 - Código tipo de moneda no permitido o no valido
1142 - No se permite el tipo de dato para codMoneda
1143 - El campo "codMoneda" es nulo o vacío
1144 - El campo "mtoTipoCambio" es nulo o vacío
1145 - Solo se permite dato numérico y decimal para el mtoTipoCambio

</td></tr>
</table>

5.12 Servicio Web Api editar tipo de cambio individual

<table>
<tr><td>Nombre Web Services</td><td>Servicio Web Api editar tipo de cambio individual</td></tr>
<tr><td>Descripción</td><td>Servicio web api que edita el tipo de cambio individual en propuesta de ventas</td></tr>
<tr><td>Url</td><td>https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/{perTributario}/complementoindividual</td></tr>
<tr><td>Parámetros[URL]</td><td>
<table>
<tr><th>Param-formato-tipo</th><th>Descripción</th></tr>
<tr><td>perTributario-alfanumérico-String</td><td>Periodo tributario (Obligatorio)</td></tr>
</table>
</td></tr>
<tr><td>Parámetros[body]</td><td>
<table>
<tr><th>Param-formato-tipo</th><th>Descripción</th></tr>
<tr><td>Object</td><td>Object - inicio</td></tr>
<tr><td>codCar-alfanumérico-String</td><td>Código de Anotación de Registro (CAR SUNAT) (Obligatorio)</td></tr>
<tr><td>codMoneda-numérico-String</td><td>Códigos de moneda (Obligatorio)</td></tr>
<tr><td>mtoTipoCambio-decimal-String</td><td>Tipo de cambio de PEN (Soles) a USD (Dólares) (Obligatorio)</td></tr>
<tr><td>mtoCambioMonedaExt-decimal-String</td><td>Tipo de cambio de moneda extranjera a PEN (Soles) (Opcional: Para regstrar tipo de cambio de monedas extranjeras distintas a USD)</td></tr>
<tr><td>Object</td><td>Object - fin</td></tr>
</table>
</td></tr>
<tr><td>Parámetros[header]</td><td>
Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded
<table>
<tr><th>Parámetros</th><th>valor</th></tr>
</table>
</td></tr>
</table>

	Content-Type: application/json Accept: application/json Authorization: Bearer token obtenido de la autenticación Método: PUT
Parámetros[salida]	**Parámetros** / **valor** HTTP status: 200 Content-Type: application/json
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/202302/complementoindividual **Headers**  **Body**  **Result OK**  **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." } 
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** ● 1005 - El campo ‘perTributario’ no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1135 - El campo “codCar” no enviado o es vacío ● 1136 - Solo se permite dato numérico de 27 dígitos para el Codigo CAR. ● 1141 - Código tipo de moneda no permitido o no valido ● 1142 - No se permite el tipo de dato para codMoneda ● 1143 - El campo "codMoneda" es nulo o vacío ● 1144 - El campo "mtoTipoCambio" es nulo o vacío ● 1145 - Solo se permite dato numérico y decimal para el mtoTipoCambio

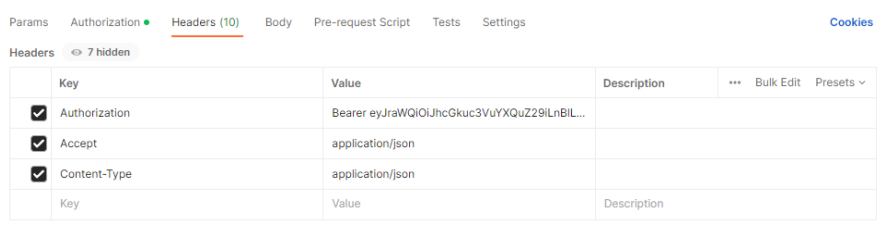
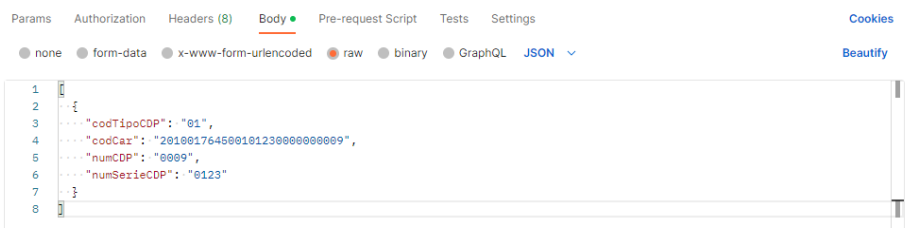
5.13 Servicio Web Api eliminar comprobante propuesta

Nombre Web Services	Servicio Web Api eliminar comprobante de la propuesta
Descripción	Permite eliminar un comprobante de la propuesta que ha sido agregado por el contribuyente
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/{perTributario}/eliminacomprobante
Parámetros[URL]	**Param-formato-tipo**: perTributario-alfanumérico-String **Descripción**: Periodo tributario (Obligatorio)
Parámetros[body]	**Param-formato-tipo** / **Descripción** **Array** / **Array - inicio** numSerieCDP-alfanumérico-String / Número de serie del comprobante de pago o documento (Obligatorio) numCDP-alfanumérico-String / Número del comprobante de pago o documento (Obligatorio) codCar-alfanumérico-String / Código de Anotación de Registro (CAR SUNAT) (Obligatorio) codTipoCDP-alfanumérico-String / Tipo de Comprobante de Pago o Documento. Solo permite los comprobantes de pago 00, 01, 03, 05, 06, 07, 08, 11, 12, 13, 14, 15, 16, 18, 21, 24, 25, 27, 28, 30, 32, 34, 35, 36, 37, 42, 43, 44, 45, 48, 49, 55 y 56 (en tanto haya sido incorporado por importación o agregar en ningún caso se debe eliminar el CP propuesto) (Obligatorio) **Array** / **Array - fin**
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: DELETE
Parámetros[salida]	**Parámetros** / **Valor** HTTP status / 200 Content-Type / application/json
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/202302/eliminacomprobante **Headers** **Body** **Result OK**

	  Result Fail { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo "numRuc" no enviado o es vacío" }] } **Lista de errores 422:** • 1005 - El campo 'perTributario' no enviado o es vacio • 1006 - Formato de perTributario no cumple con el formato 'yyyymm' • 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual • 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado • 1135 - El campo "codCar" no enviado o es vacío • 1136 - Solo se permite dato numérico de 27 dígitos para el Codigo CAR. • 1323- El campo "numSerieCDP" es nulo o vacío • 1011 - El campo "codTipoCDP" no enviado o es vacío • 1012 - El codTipoCDP ingresado no existe o no es válido

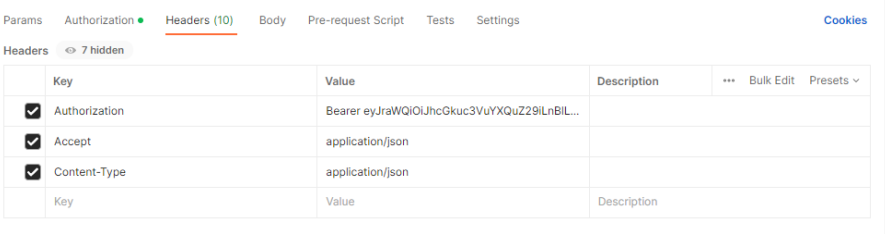
5.14 Servicio Web Api eliminar comprobante preliminar

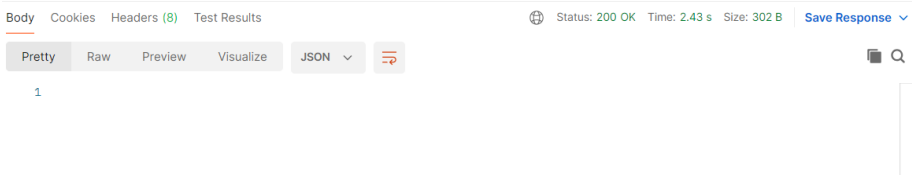
Nombre Web Services	Servicio Web Api eliminar comprobante del preliminar RVIE
Descripción	Permite eliminar un comprobante del preliminar RVIE
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/{perTributario}/comprobantepreliminar
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio)
Parámetros[body]	**Param-formato-tipo** / **Descripción** **Array** / **Array - inicio** codCar-alfanumérico-String / Código de Anotación de Registro (CAR SUNAT) (Obligatorio) codTipoCDP-alfanumérico-String / Tipo de Comprobante de Pago o Documento. Se puede eliminar todos códigos de comprobantes de pago de la tabla 03 del anexo 01, de la Resolución de Superintendencia 112-2021/SUNAT. (Obligatorio) numSerieCDP-alfanumérico-String / Número de serie del comprobante de pago o documento (Obligatorio) numCDP-alfanumérico-String / Número del comprobante de pago o documento(Obligatorio) **Array** / **Array - fin**
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: POST
Parámetros[salida]	**Parámetros** / **Valor** HTTP status / 200 Content-Type / application/json
Evidencias	**URL**

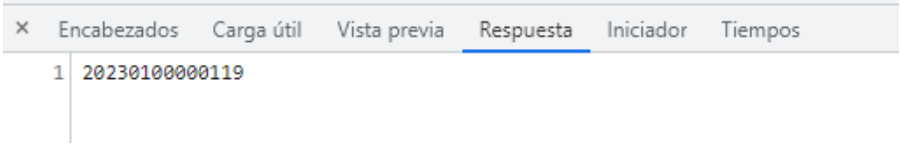
	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/202302/comprobantepreliminar **Headers**  **Body**  **Result OK** **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo "numRuc" no enviado o es vacío" }] } **Lista de errores 422:** • 1005 - El campo 'perTributario' no enviado o es vacio • 1006 - Formato de perTributario no cumple con el formato 'yyyymm' • 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual • 1135 - El campo "codCar" no enviado o es vacío • 1136 - Solo se permite dato numérico de 27 dígitos para el Codigo CAR. • 1323 - El campo "numSerieCDP" es nulo o vacío • 1011 - El campo "codTipoCDP" no enviado o es vacío • 1012 - El codTipoCDP ingresado no existe o no es válido

5.15 Servicio Web Api eliminar preliminar

Nombre Web Services	Servicio Web Api eliminar preliminar
Descripción	Permite eliminar el preliminar y todos los datos del reemplazo de la propuesta
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/{perTributario}/eliminarreemplazo?codLibro={codLibro}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio) codLibro-alfanumérico-String / Código de libro: 140000 RVIE (Obligatorio)
Parámetros[body]	(No aplica)
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: PUT

<table>
<tr><td>Parámetros[salida]</td><td>
<table>
<tr><th>Parámetros de Salida</th><th>Descripción</th><th>Formato</th><th>Tipo dato</th></tr>
<tr><td>numTicket(Opcional)</td><td>Número de ticket de envío [AAAA99999999]
AAAA: Año
99: Tipo de correlativo
99999999: Número correlativo de envío
Nota: En caso la petición devuelva número de ticket se debe consultar su estado en 5.16 Servicio Web Api consultar estado de envío de ticket</td><td>alfanumerico</td><td>String</td></tr>
</table>
</td></tr>
<tr><td>Evidencias</td><td>
URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/202302/eliminarreemplazo?codLibro=140000
Headers


Body
(No aplica)
Result OK
1. Si la respuesta es vacío, el proceso ha terminado correctamente.


2. Si la respuesta es un número de ticket, debe consultar su estado en 5.16 Servicio Web Api consultar estado de envío de ticket, en la cual deberá tener el estado 06 – Terminado.


Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
</td></tr>
<tr><td>Mensaje Error</td><td>
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:
● 1005 - El campo ‘perTributario’ no enviado o es vacio
● 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
● 1140 - El campo “codLibro” no enviado o es vacío
● 2298 - No es posible eliminar su reemplazo ya que aun se encuentra en etapa propuesta, no se verifica ningun reemplazo.
● 2299 - No es posible eliminar su reemplazo, ya que se encuentra en la etapa preliminar registrado, si desea eliminar su preliminar registrado debe utilizar el servicio web eliminar preliminar registrado.
● 2300 - No es posible eliminar su reemplazo, ya que el registro ya fue generado desde portal.
</td></tr>
</table>

5.16 Servicio Web Api consultar estado de envío de ticket

Nombre Web Services	Servicio Web Api consultar estado de envío de ticket.
Descripción	Permite consultar el estado de envío del ticket. Para que funcione el servicio, se necesita haber generado un ticket con reemplazar propuesta, generar propuesta, descargar propuesta, entre otros.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionprocesosmasivos/web/masivo/consultaestadotickets?perIni={perIni}&perFin={perFin}&page={page}&perPage={perPage}&numTicket={numTicket}
Parámetros[URL]	

Param-formato-tipo	Descripción
perIni-alfanumérico-String	Periodo de consulta de documentos de comprobantes del RVIE preliminar Inicio. (Obligatorio)
perFin-alfanumérico-String	Periodo de consulta de documentos de comprobantes del RVIE preliminar Final. (Obligatorio)
page-numerico-int	Número de página. Ejemplo: 1 (Obligatorio)
perPage-numerico-int	Cantidad de tickets por página Ejemplo: 20 (Obligatorio)
codLibro-alfanumérico-String	Código de libro: 140000 RVIE (Obligatorio)
codOrigenEnvio-alfanumérico-String	Código de origen de envío: 2 Servicio web (Obligatorio)

Parámetros[body]: No aplica

Parámetros[header]

Descripción:
Content-type: tipo de contenido a enviar
Valores:

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

Parámetros[salida]

Parámetros de Salida	Descripcion
paginacion-Object	**Object de paginación - inicio**
page- numérico-Int	Ejemplo: 1 (Obligatorio)
perPage- numérico-Int	Ejemplo: 20 (Obligatorio)
totalRegistros- numérico-Int	Total de registros (Obligatorio)
paginacion-Object	**Object de paginación - fin**
registros-array-array	**Array de registros - inicio**
showReportesDescarga-numérico-Integer	Valores 0 y 1 0 - no muestra icono de archivo de texto 1 - muestra ícono de archivo de texto
perTributario-alfanumerico-String	Periodo tributario
numTicket-alfanumerico-String	Número de ticket de envío
fecCargaImportacion-dd/mm/yyyy,'T','hh:ii:ss'-Date	Fecha de la carga del archivo de importación, o fecha de solicitud de generacion de archivo
fecInicioProceso-yyyy-mm-dd-String	Fecha de inicio de proceso
codProceso-alfanumerico-String	Código del indicador de carga masiva. (Ver Anexo I: Indicador de carga masiva) . (1-97)
desProceso-alfanumerico-String	Descripcion del indicador de Carga Masiva. (Ver Anexo I: Indicador de carga masiva)
codEstadoProceso-alfanumerico-String	Código de estado de envio (Ver Anexo III: Código de estado de envío)
desEstadoProceso-alfanumerico-String	Descripción de estado de envio (Ver Anexo III: Código de estado de envío)
nomArchivoImportacion-alfanumérico-String	Nombre del archivo de importación

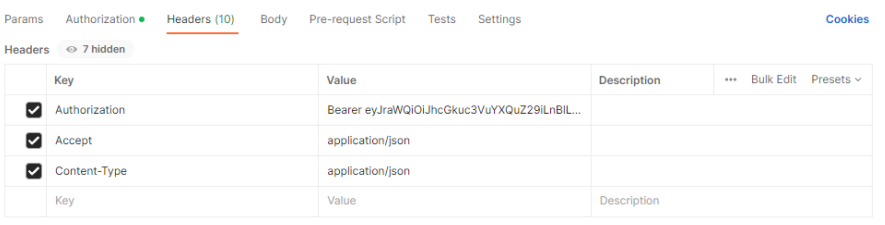
detalleTicket-Object	**Object detalle de ticket - inicio**
numTicket-alfanumerico-String	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío
fecCargaImportacion-dd/mm/yyyy-Date	Fecha de la carga del archivo de importación, o fecha de solicitud de generacion de archivo
horaCargaImportacion-hh:mm:ss'-Date	(DetalleTicket.fecCargaImportacion).- Hora de la carga del archivo de importación, o fecha de solicutud de generacion de archivo
codEstadoEnvio-alfanumérico-String	Código del estado de envío
desEstadoEnvio-alfanumérico-String	Descripción del estado de envío
nomArchivoReporte-alfanumérico-String	Nombre del archivo reporte
cntFilasvalidada-numérico-Integer	Cantidad de filas validadas o total de registros
cntCPError-numérico-Integer	Cantidad de comprobantes con error
cntCPInformados-numérico-Integer	Cantidad de CP informados
detalleTicket-Object	**Object detalle de ticket - fin**
archivoReporte-Array	**Array archivo reporte - inicio**
codTipoAchivoReporte-alfanumerico-String	Código del tipo de archivo de reporte
nomArchivoReporte-alfanumerico-String	Nombre del archivo de reporte
nomArchivoContenido-alfanumerico-String	Nombre del archivo contenido
archivoReporte- Array	**Array archivo reporte - fin**
registros-array-array	**Array de registros - fin**

Evidencias

URL

https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionprocesosmasivos/web/masivo/consultaestadotickets?perIni=202301&perFin=202305&page=1&perPage=20

Headers



Body

(No aplica)

Result OK

<table>
<tr><td></td><td>

<pre>
Body Cookies Headers (12) Test Results 200 OK 551 ms 1.14 KB Save as Example
Pretty Raw Preview Visualize JSON
{
"paginacion": {
"page": 1,
"perPage": 20,
"totalRegistros": 5
},
"registros": [
{…
},
{
"showReporteDescarga": "1",
"perTributario": "202302",
"numTicket": "20230200000156",
"fecCargaImportacion": null,
"fecInicioProceso": "2023-08-14",
"codProceso": "5",
"desProceso": "Generación de Registros",
"codEstadoProceso": "06",
"desEstadoProceso": "Terminado",
"nomArchivoImportacion": "",
"detalleTicket": [
"numTicket": "20230200000156",
"fecCargaImportacion": "2023-08-14",
"horaCargaImportacion": "13:49:07",
"codEstadoEnvio": "06",
"desEstadoEnvio": "Terminado",
"nomArchivoReporte": "LE20563472678202302001404000 3EXP2.txt",
"cntFilasvalidada": 0,
"cntCPError": 0,
"cntCPInformados": 0
],
"archivoReporte": [
{
"codTipoAchivoReporte": "00",
"nomArchivoReporte": "LE20563472678202302001404000 3EXP2.txt",
Console Runner
</pre>

Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
</td></tr>
<tr><td>Mensaje Error</td><td>
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:

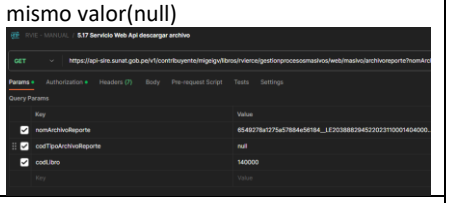
1067 - El campo “perIni” no enviado o es vacío
1068 - Formato de perIni no cumple con el formato “yyyymm”
1069 - El perIni de búsqueda no debe ser mayor a la fecha actual
1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado
1071 - El campo “perFin” no enviado o es vacío
1072 - Formato de perFin no cumple con el formato “yyyymm”
1073 - El perFin de búsqueda no debe ser mayor a la fecha actual
El campo “page” no enviado o es vacío
El campo “page” debe ser numérico mayor a cero
El campo “per_page” no enviado o es vacío
El campo “per_page” debe ser numérico mayor a cero
1052 - Formato no permitido o no valido para el número de Ticket

</td></tr>
</table>

5.17 Servicio Web Api descargar archivo

Nombre Web Services	Servicio Web Api descargar archivo
Descripción	Permite descargar los archivos generados zipeados y particionados guardados en el fileserver. Solo si el resultado del campo “registros [0].codProcesos” del servicio 5.16 Servicio Web Api consultar estado ticket es 3 o 4, se podrá hacer uso de este servicio. De otro modo, no aparecerá el estado de envío del ticket. Además, del servicio 5.16 se utilizaran los siguientes campos: registros[0].codProceso registros[0].detalleTicket registros[0].perTributario registros[0].numTicket registros[0].nomArchivoImportacion
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionprocesosmasivos/web/

	masivo/archivoreporte?nomArchivoReporte={nomArchivoReporte}&codTipoArchivoReporte={codTipoArchivoReporte}&codLibro={codLibro}
Parámetros[URL]	

Param-formato-tipo	Descripción
nomArchivoReporte-alfanumérico-String	Nombre del archivo generado (Parámetro de salida del servicio 5.16 Servicio Web Api consultar estado de envío de ticket: archivoReporte.nomArchivoReporte) (Obligatorio)
codTipoArchivoReporte-numérico-String	Codigo del tipo de archivo (Parámetro de salida del servicio 5.16 Servicio Web Api consultar estado de envío de ticket: archivoReporte.codTipoArchivoReporte) (Obligatorio) Nota: Si el campo codTipoAchivoReporte que devuelve el API 5.16 es null, colocar el mismo valor(null) 
codLibro-numérico-String	Codigo de libro: RVIE 140000 (Obligatorio)
perTributario-alfanumérico-String	Periodo tributario (Parámetro de salida del servicio 5.31 Servicio Web Api consultar estado ticket: perTributario) (Obligatorio)
codProceso-alfanumérico-String	Código del indicador de carga masiva. (Parámetro de salida del servicio 5.31 Servicio Web Api consultar estado ticket: codProceso) (Obligatorio)
numTicket-alfanumérico-String	Número de ticket de envío (Obligatorio)

Parámetros[body]	No aplica
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

Parámetros[salida]

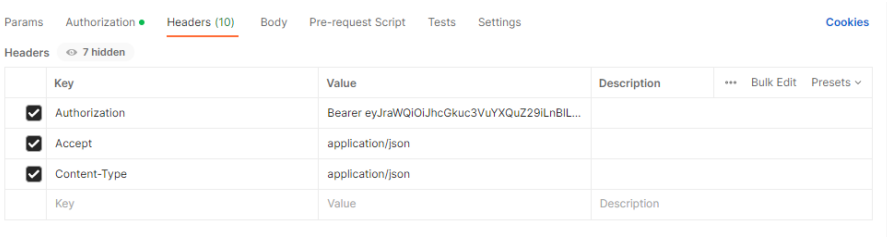
Parámetros de Salida	Descripcion
Buffer-binary-binary	buffer: Arreglo de bits

Evidencias

URL

https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionprocesosmasivos/web/masivo/archivoreporte?nomArchivoReporte=20100176450-CPF-202302-01.zip&codTipoArchivoReporte=01&codLibro=140000

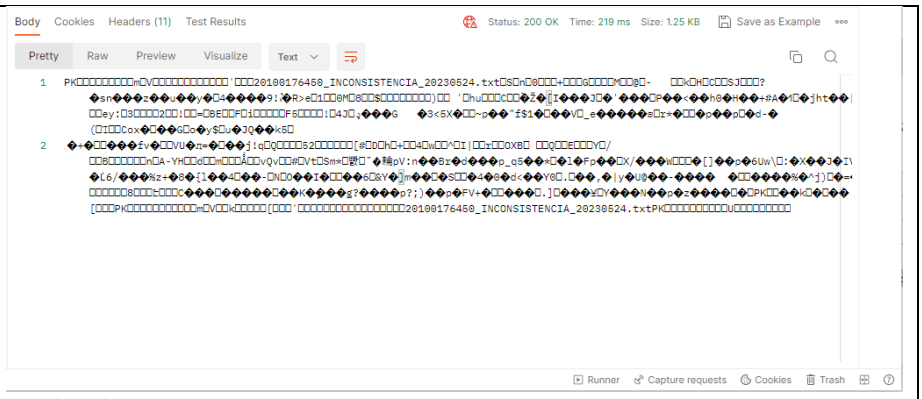
Headers



Body

(No aplica)

Result OK

	 **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** ● 1134 - El campo “nomArchivoReporte” no enviado o es vacío ● 1140 - El campo “codLibro” no enviado o es vacío

5.18 Servicio Web Api descargar propuesta

Nombre Web Services	Servicio Web Api descargar propuesta
Descripción	Permite descargar la propuesta de RVIE. Este servicio generará un ticket para poder utilizar el servicio “5.17 Servicio Web Api descargar archivo” y descargar la propuesta.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/{perTributario}/exportapropuesta?mtoTotalDesde={mtoTotalDesde}&mtoTotalHasta={mtoTotalHasta}&fecDocumentoDesde={fecDocumentoDesde}&fecDocumentoHasta={fecDocumentoHasta}&numRucAdquiriente={numRucAdquiriente}&numCarSunat={numCarSunat}&codTipoCDP={codTipoCDP}&codTipoInconsistencia={codTipoInconsistencia}&codTipoArchivo={codTipoArchivo}
Parámetros[URL]	

Param-formato-tipo	Descripción
perTributario-alfanumérico-String	Periodo tributario (Obligatorio)
codTipoArchivo-numérico-Integer	Código del tipo de archivo (Obligatorio) 0: txt 1: xls
mtoDesde-Numerico-decimal128	Importe total del comprobante de pago. Monto del rango inicial (monto total) (Opcional: Para filtrar comprobante por importe mínimo)
mtoHasta-Numerico-decimal128	Importe total del comprobante de pago. Monto final del rango (monto total) (Opcional: Para filtrar comprobante por importe máximo)
codTipoCDP-alfanumérico-String	Tipo de Comprobante de Pago o Documento. Se puede descargar todos los comprobantes de pado indicados en la tabla 03 del anexo 1 de la Resolución de Superintendencia 112-2021/SUNAT y modificatorias (Opcional: Para hallar solo un tipo de comprobante)
codTipoInconsistencia-numérico-String	Código de tipo de inconsistencia (Opcional: Para inconsistencia específica)
numCarSunat-alfanumerico-String	Numero de identificación del comprobante (Opcional: Para filtrar por comprobante)
fecDocumentoDesde-dd/mm/aaaa-String	Fecha de emision desde (Opcional: Para filtrar por fecha de inicio)

	fecDocumentoHasta-dd/mm/aaaa-String — Fecha de emisión hasta (Opcional: Para filtrar por fecha de fin)
Parámetros[body]	No aplica
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

Parámetros[salida]

Parámetros de Salida	Descripcion	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

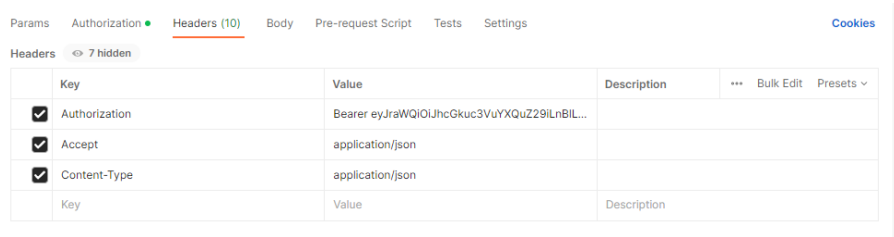
Evidencias

Evidencia 1: ***Existe datos en la propuesta***

URL

https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/202305/exportapropuesta?codTipoArchivo=0

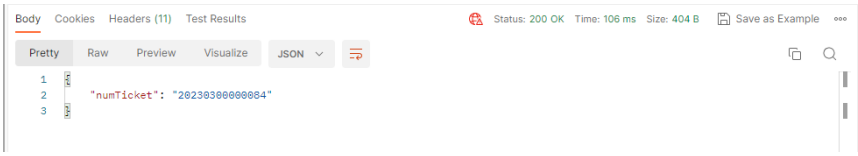
Headers



Body

(No aplica)

Result OK



Result Fail

{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Evidencia 2: ***No existe datos en la propuesta***

URL

https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/propuesta/202301/exportapropuesta?codTipoArchivo=0

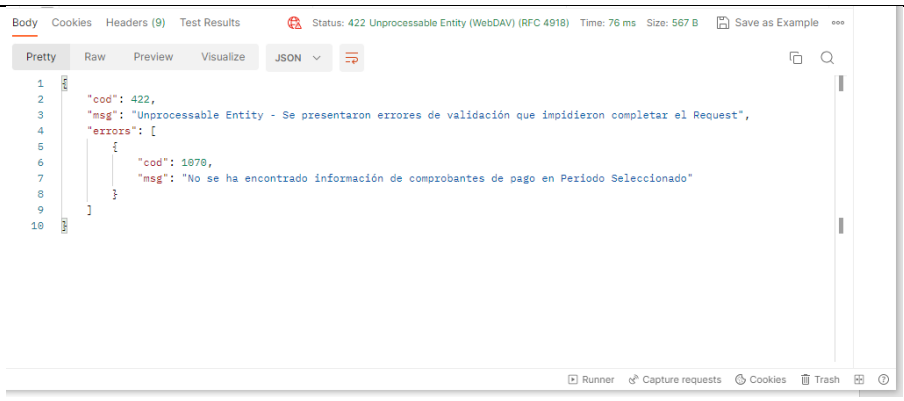
Headers

(No aplica)

Body

(No aplica)

Result Fail 422:

	 **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** ● 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1059 - Código tipo de Archivo no permitido o no valido ● 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo ● 1061 - El campo "codTipoArchivo" es nulo o vacío ● 1112 - El Monto Total Desde debe ser mayor o igual al Monto Total Hasta ● 1113 - Si se realiza busqueda por monto Total, se debe ingresar los campos mtoTotalDesde y mtoTotalHasta ● 1114 - Fecha Documento Desde debe estar dentro del Periodo seleccionado ● 1115 - Debe cumplir con el siguiente formato ‘dd/mm/yyyy’ ● 1116 - Fecha de documento Hasta debe ser mayor o igual al Fecha de documento Desde ● 1117 - Si se realiza busqueda por Fecha Documento, se debe ingresar los campos Fecha Documento Desde, Fecha Documento Hasta ● 1118 - Fecha Documento Hasta debe estar dentro del Periodo seleccionado ● 1104 - El código de tipo de comprobante de pago enviado no es válido ● 1119 - El código de tipo de inconsistencia enviado no es válido

5.19 Servicio Web Api descargar no incluidos

Nombre Web Services	Servicio Web Api descargar no incluidos
Descripción	Permite descargar los comprobantes excluidos. Solo aplicable para CP Excluidos del periodo vigente. Este servicio generará un ticket para poder utilizar el servicio “5.17 Servicio Web Api descargar archivo” y descargar los no incluidos.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/noincluidos/{perTributario}/exportanoincluidos?mtoTotalDesde={mtoTotalDesde}&mtoTotalHasta={mtoTotalHasta}&fecDocumentoDesde={fecDocumentoDesde}&fecDocumentoHasta={fecDocumentoHasta}&numRucAdquiriente={numRucAdquiriente}&numCarSunat={numCarSunat}&codTipoCDP={codTipoCDP}&codTipoInconsistencia={codTipoInconsistencia}&codTipoArchivo={codTipoArchivo}&codOrigenEnvio={codOrigenEnvio}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** codTipoArchivo-numérico-Integer — Código del tipo de archivo - 0: txt - 1: xls (Obligatorio) codOrigenEnvio-numerico-Integer — Código de origen de envio: 2 (Obligatorio)

perTributario Alfanumérico-String	Periodo tributario (Obligatorio)
mtoTotalDesde-decimal-Decimal	Monto total desde (Opcional: Para filtrar por importe mínimo)
mtoTotalHasta-decimal-Decimal	Monto total hasta (Opcional: Para filtrar por importe máximo)
fecDocumentoDesde-dd/mm/yyyy-Date	Fecha de emisión desde (Opcional: Para filtrar por fecha de inicio)
fecDocumentoHasta-dd/mm/yyyy-Date	Fecha de emisión hasta (Opcional: Para filtrar por fecha de fin)
numRucAdquiriente- alfanumérico-String	Numero de ruc adquiriente (Opcional: Para filtrar por adquiriente específico)
numCarSunat-numérico-String	Numero identificador del comprobante (Opcional: Para filtrar por comprobante)
codTipoCDP-numerico-integer	Código de tipo de comprobante (Opcional: Para filtrar por tipo de comprobante)
codTipoInconsistencia-numerico-integer	Código del tipo de inconsistencia (Opcional: Para hallar inconsistencias específicas)

Parámetros[body]

No aplica.

Parámetros[header]

Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

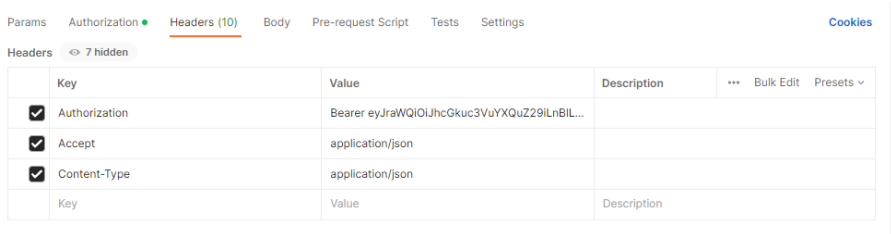
Parámetros[salida]

Parámetros de Salida	Descripcion	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

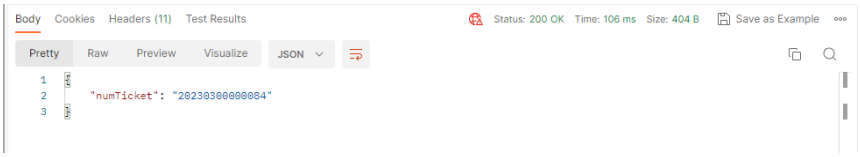
Evidencias

URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/propuesta/web/noincluidos/202301/exportanoincluidos?mtoTotalDesde=&mtoTotalHasta=&fecDocumentoDesde=&fecDocumentoHasta=&numRucAdquiriente=&numCarSunat=&codTipoCDP=&codTipoInconsistencia=&codTipoArchivo=0&codOrigenEnvio=2

Headers



Body
(No aplica)
Result OK

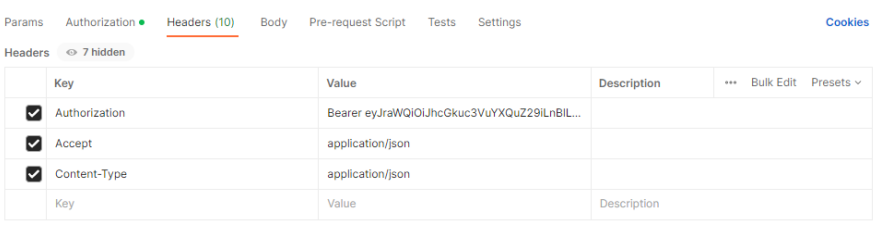
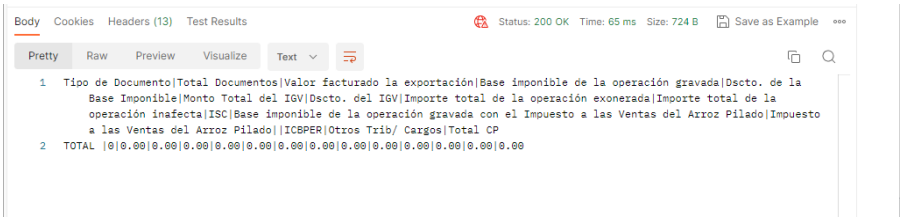


Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** ● 1005 - El campo ‘perTributario’ no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado ● 1112 - El Monto Total Desde debe ser mayor o igual al Monto Total Hasta ● 1113 - Si se realiza busqueda por monto Total, se debe ingresar los campos mtoTotalDesde y mtoTotalHasta ● 1114 - Fecha Documento Desde debe estar dentro del Periodo seleccionado ● 1115 - Debe cumplir con el siguiente formato ‘dd/mm/yyyy’ ● 1116 - Fecha de documento Hasta debe ser mayor o igual al Fecha de documento Desde ● 1117 - Si se realiza busqueda por Fecha Documento, se debe ingresar los campos Fecha Documento Desde, Fecha Documento Hasta ● 1118 - Fecha Documento Hasta debe estar dentro del Periodo seleccionado ● 1104 - El código de tipo de comprobante de pago enviado no es válido ● 1119 - El código de tipo de inconsistencia enviado no es válido ● 1059 - Código tipo de Archivo no permitido o no valido ● 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo ● 1061 - El campo "codTipoArchivo" es nulo o vacío

5.20 Servicio Web Api descargar resumen

Nombre Web Services	Servicio Web Api descargar resumen
Descripción	Permite descargar todos los tipos de resumen, propuesta, incluidos o excluidos, preliminar, RVIE generado, ajustes posteriores
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/resumen/web/resumencomprobantes/{perTributario}/{codTipoResumen}/{codTipoArchivo}/exporta?codLibro={codLibro}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio) codTipoResumen-alfanumérico-String / Código de tipo resumen - 1 Resumen de propuesta - 2 Resumen de preliminar - 3 Resumen no Incluidos (V) o Excluidos(C) - 4 Resumen de registro - 5 Resumen de preliminar registrado - 6 Resumen ajustes posteriores - 7 Resumen no domiciliados (Obligatorio) codTipoArchivo-numérico-Integer / Código del tipo de archivo -0: txt - 1: csv (Obligatorio) codLibro-alfanumérico-String / Código de libro: 140000 RVIE (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: GET
Parámetros[salida]	**Parámetros de Salida** / **Descripcion** Buffer-binary-binary / buffer: Arreglo de bits

Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/resumen/web/resumencomprobantes/202301/1/0/exporta?codLibro=14000 **Headers**  **Body** (No aplica) **Result OK**  **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** ● 1005 - El campo ‘perTributario’ no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ ● 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado ● 1140 - El campo “codLibro” no enviado o es vacío ● 1059 - Código tipo de Archivo no permitido o no valido ● 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo ● 1061 - El campo "codTipoArchivo" es nulo o vacío ● 1056 - Solo se permite dato numérico de 1 dígito para el codTipoResumen ● 1057 - El campo "codTipoResumen" es nulo o vacío

5.21 Servicio Web Api descargar resumen inconsistencias

Nombre Web Services	Servicio Web Api descargar resumen de inconsistencias RVIE
Descripción	Retorna una lista, con el resumen de inconsistencias de los comprobantes de pago de acuerdo con el periodo y tipo de resumen asociado al código enviado, en formato json.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/resumen/web/resumeninconsistencias/{perTributario}?codTipoResumen={codTipoResumen}&codLibro={codLibro}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario -alfanumérico-String / Periodo tributario (Obligatorio) codTipoResumen-numérico-Integer / Código del tipo de resumen. El valor enviado debe ser numérico de un carácter (1, 2, 3 ó 4) (Obligatorio) codLibro-alfanumérico-String / Código de libro: 140000 RVIE (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

Parámetros[salida]

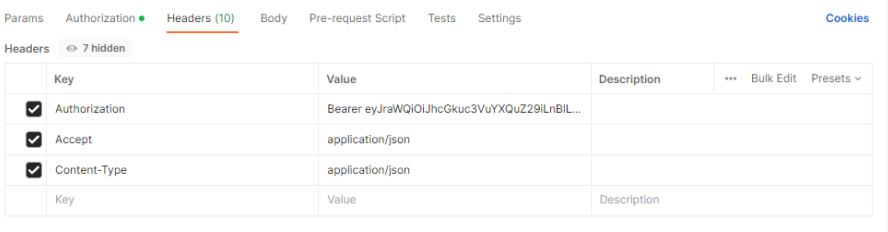
Parámetros de Salida	Descripcion
Object	**Object - inicio**
numRuc-alfanumérico-String	Número de ruc
perTributario-alfanumérico-String	Periodo tributario
codTipoResumen-Numérico-Integer	Código de tipo resumen
cantidad-Object	**Object cantidad - inicio**
porcentajeRelFiscal-Numerico-decimal128	Porcentaje de cantidad de comprobantes con inconsistencias Relacionadas al Crédito Fiscal
porcentajeNoRelFiscal-Numerico-decimal128	Porcentaje de cantidad de comprobantes con inconsistencias No Relacionadas al Crédito Fiscal
porcentajeSinValidaciones-Numerico-decimal128	Porcentaje de cantidad de comprobantes sin inconsistencias
total-Numerico-decimal128	Cantidad total de comprobantes en un determinado periodo
cantidad-Object	**Object cantidad - fin**
monto-Object	**Object monto - inicio**
porcentajeRelFiscal-Numerico-decimal128	Porcentaje de montos de comprobantes con inconsistencias Relacionadas al Crédito
porcentajeNoRelFiscal-Numerico-decimal128	Porcentaje de montos de comprobantes con inconsistencias No Relacionadas al Crédito Fiscal en un determinado periodo
porcentajeSinValidaciones-Numerico-decimal128	Porcentaje de montos de comprobantes sin inconsistencias en un determinado periodo
total-Numerico-decimal128	Monto total de comprobantes en un determinado periodo
monto-Object	**Object monto - fin**
Object	**Object - fin**

Evidencias

URL

https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/resumen/web/resumeninconsistencias/202304?codTipoResumen=1&codLibro=140000

Headers



Body

(No aplica)

Result OK

	Body Cookies Headers (12) Test Results Status: 200 OK Time: 215 ms Size: 563 B Save as Example
	Pretty Raw Preview Visualize JSON
	{
	"numRuc": "20563472678",
	"perPeriodoTributario": "202304",
	"codTipoResumen": "1",
	"cantidad": {
	"porcentajeRelFiscal": 0,
	"porcentajeNoRelFiscal": 0,
	"porcentajeSinValidaciones": 100.00,
	"total": 5
	},
	"monto": {
	"porcentajeRelFiscal": 0,
	"porcentajeNoRelFiscal": 0,
	"porcentajeSinValidaciones": 100.00,
	"total": 799452465.690
	}
	}
	Runner Capture requests Cookies Trash
	Result Fail
	{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** ● 1005 - El campo ‘perTributario’ no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ ● 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado ● 1140 - El campo “codLibro” no enviado o es vacío ● 1056 - Solo se permite dato numérico de 1 dígito para el codTipoResumen ● 1057 - El campo "codTipoResumen" es nulo o vacío

5.22 Servicio Web Api exportar preliminar de registro de Ventas

Nombre Web Services	Servicio Web Api exportar preliminar del registro de Ventas electrónico
Descripción	Permite descargar el preliminar del registro de Ventas eletrónico. Este servicio generará un ticket para poder utilizar el servicio “5.17 Servicio Web Api descargar archivo” y descargar el preliminar.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/{perTributario}/reportepreliminar?mtoTotalDesde={mtoTotalDesde}&mtoTotalHasta={mtoTotalHasta}&fecDocumentoDesde={fecDocumentoDesde}&fecDocumentoHasta={fecDocumentoHasta}&numRucAdquiriente={numRucAdquiriente}&numCarSunat={numCarSunat}&codTipoCDP={codTipoCDP}&codTipoInconsistencia={codTipoInconsistencia}&codTipoArchivo={codTipoArchivo}
Parámetros[URL]	

Param-formato-tipo	Descripción
perTributario-alfanumérico-String	Periodo seleccionado (Obligatorio)
codOrigenEnvio-alfanumérico-String	Código de origen de envío: 2 Servicio web (Obligatorio)
codLibro-alfanumérico-String	Código de libro: 140000 RVIE (Obligatorio)
mtoTotalDesde-decimal-Decimal	Monto desde (Opcional: Para filtrar por monto mínimo)
mtoTotalHasta-decimal-Decimal	Monto hasta (Opcional: Para filtrar por monto máximo)
fecDocumentoDesde-dd/mm/yyyy-Date	Fecha del documento desde (Opcional: Para filtrar por fecha de incio)
fecDocumentoHasta-dd/mm/yyyy-Date	Fecha del documento hasta (Opcional: Para filtrar por fecha de fin)
numRucAdquiriente-alfanumérico-String	Número de RUC Cliente (Opcional: Para filtrar por adquiriente específico)
numCarSunat-alfanumérico-String	CAR SUNAT (Opcional: Para hallar comprobante)
codTipoCDP-alfanumérico-String	Tipo de documento (Opcional: Para filtrar por tipo de comprobante)

	codTipoArchivo-numérico-Integer	Código del tipo de archivo - 0: txt - 1: xls (Obligatorio)
	codTipoInconsistencia-alfanumérico-String	Código de tipo de inconsistencia (Opcional: Para filtrar por inconsistencias específicas)

Parámetros[body]

No aplica

Parámetros[header]

Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

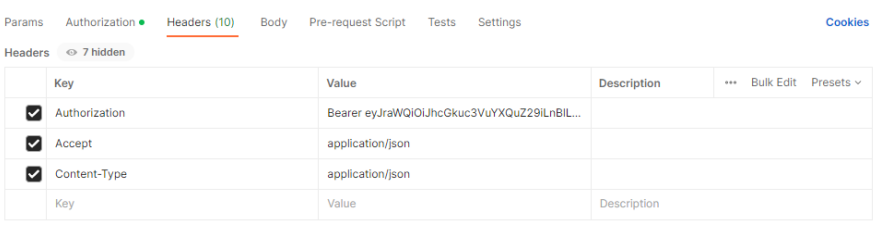
Parámetros[salida]

Parámetros de Salida	Descripcion	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

Evidencias

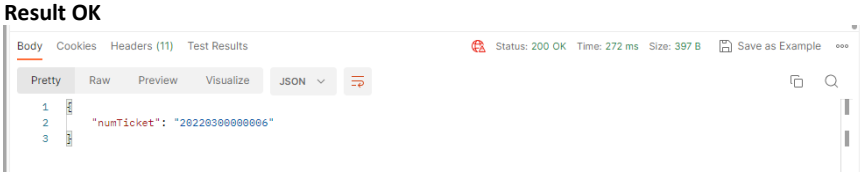
URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/202212/reporteprelimiar?codTipoArchivo=0

Headers



Body
(No aplica)

Result OK



Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

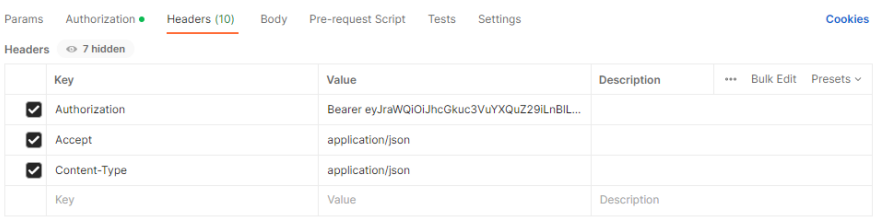
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

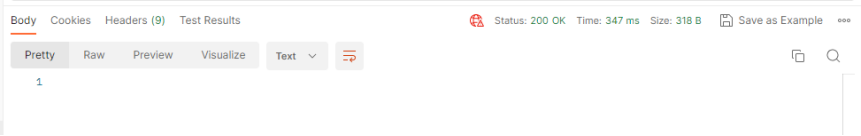
Lista de errores 422:

- 1005 - El campo ‘perTributario’ no enviado o es vacio
- 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
- 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
- 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado
- 1112 - El Monto Total Desde debe ser mayor o igual al Monto Total Hasta
- 1113 - Si se realiza busqueda por monto Total, se debe ingresar los campos mtoTotalDesde y mtoTotalHasta
- 1114 - Fecha Documento Desde debe estar dentro del Periodo seleccionado
- 1115 - Debe cumplir con el siguiente formato ‘dd/mm/yyyy’

	• 1116 - Fecha de documento Hasta debe ser mayor o igual al Fecha de documento Desde • 1117 - Si se realiza busqueda por Fecha Documento, se debe ingresar los campos Fecha Documento Desde, Fecha Documento Hasta • 1118 - Fecha Documento Hasta debe estar dentro del Periodo seleccionado • 1104 - El código de tipo de comprobante de pago enviado no es válido • 1119 - El código de tipo de inconsistencia enviado no es válido • 1059 - Código tipo de Archivo no permitido o no valido • 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo • 1061 - El campo "codTipoArchivo" es nulo o vacío • 1135 - El campo “codCar” no enviado o es vacío • 1136 - Solo se permite dato numérico de 27 dígitos para el Codigo CAR.

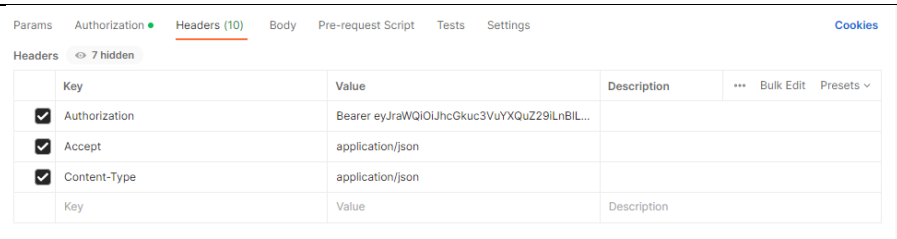
5.23 Servicio Web Api descargar reporte de casillas

Nombre Web Services	Servicio Web Api reporte de casillas
Descripción	Permite descargar todos los tipos de resumen, propuesta, incluidos o excluidos, preliminar, RVIE generado, ajustes posteriores
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/casillas/e/casillaspropuestas/{perTributario}/reporte/{tipoReporte}/{tipoDescarga}
Parámetros [URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio) tipoReporte-alfanumérico-String / Tipo de reporte: 1. Preliminar 2. Comparada (Obligatorio) tipoDescarga-alfanumérico-String / Código de tipo de descarga - 0: Txt - 1: xls - 2: pdf (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: GET
Parámetros[salida]	No aplica
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/casillas/e/casillaspropuestas/202301/reporte/2/txt **Headers**  **Body** (No aplica) **Result OK**

	 **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo "numRuc" no enviado o es vacío" }] } **Lista de errores 422:** ● 1005 - El campo 'perTributario' no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato 'yyyymm' ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado ● 1082 - Código tipo de Descarga no permitido o no valido ● 1333 - El campo "codTipoReporte" es nulo o vacío ● 1334 - El campo "codTipoReporte" solo admite valores: 1 o 2 ● 1009 - Solo se permite dato numérico de 1 dígito para el codTipoDescarga ● 1010 - El campo "codTipoDescarga" es nulo o vacío

5.24 Servicio Web Api descargar inconsistencias en registros preliminar registrado

Nombre Web Services	Servicio Web Api descargar inconsistencias en registros preliminar registrado
Descripción	Permite descargar las inconsistencias de los registros de preliminar registrado
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/casillas/inconsistenciaslibros/{perTributario}/{numCas}/reporteinconsistencia?cntlimite={cntlimite}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio) numCas-alfanumérico-String / Número de casilla (Obligatorio) Ver Anexo V: Numero de Casillas cntlimite-numérico-int / Cantidad de registros para validar el top (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: GET
Parámetros[salida]	**Parámetros de Salida** / **Descripcion** Buffer- binary-binary / buffer: Arreglo de bits
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/casillas/inconsistenciaslibros/202301/10/reporteinconsistencia/txt?cntlimite=10 **Headers**

<table>
<tr>
<td></td>
<td>


Body
(No aplica)
Result OK

Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
</td>
</tr>
<tr>
<td>Mensaje Error</td>
<td>
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:

1005 - El campo ‘perTributario’ no enviado o es vacio
1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado
El campo “numCas” no enviado o es vacío
El campo “cntlimite” no enviado o es vacío

</td>
</tr>
</table>

5.25 Servicio Web Api descargar inconsistencias por comprobante pago

<table>
<tr><td>Nombre Web Services</td><td>Servicio Web Api descargar inconsistencias por comprobantes de pago</td></tr>
<tr><td>Descripción</td><td>Permite exportar las inconsistencias por comprobantes de pago. Este servicio generará un ticket para poder utilizar el servicio “5.17 Servicio Web Api descargar archivo” y descargar las inconsistencias.</td></tr>
<tr><td>Url</td><td>https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/inconsistencias/web/periodoinconsistencias/{perTributario}/exporta?codTipoArchivo={codTipoArchivo}&fecEmisionIni={fecEmisionIni}&fecEmisionFin={fecEmisionFin}&codInconsistencia={codInconsistencia}&numDocAdquiriente={numDocAdquiriente}&codTipoCDP={codTipoCDP}&numSerieCDP={numSerieCDP}&numCDP={numCDP}&s={s}&ord={ord}</td></tr>
<tr><td>Parámetros[URL]</td><td>
<table>
<tr><th>Param-formato-tipo</th><th>Descripción</th></tr>
<tr><td>fecEmisionIni-dd/mm/aaaa-String</td><td>Fecha de emisión del Comprobante de Pago o documento - Inicio (Opcional: Para filtrar por fecha de inicio)</td></tr>
<tr><td>fecEmisionFin-dd/mm/aaaa-String</td><td>Fecha de emisión del Comprobante de Pago o documento - FIN (Opcional: Para filtrar por fecha de fin)</td></tr>
<tr><td>codInconsistencia-alfanumerico-String</td><td>Código de inconsistencia funcional o calculada, ejemplo:
301 - Fecha de emisión del comprobante de pago o fecha de pago del impuesto se anota luego de los doce meses siguientes a la fecha de emisión del comprobante o del pago del impuesto, según corresponda. (Opcional:</td></tr>
</table>
</td></tr>
</table>

	Para filtrar por inconsistencias específicas)
numDocAdquiriente-alfanumérico-String	Número de RUC o Documento de Identidad del adquiriente (Opcional: Para hallar adquiriente específico)
codTipoCDP-alfanumérico-String	Tipo de Comprobante de Pago o Documento. Se puede eliminar todos los códigos de comprobantes de la tabla 03 del anexo N.° 1 de la Resolución de Superintendencia 112-2021/SUNAT y modificatorias. (Opcional: Para filtrar por tipo de comprobante)
numSerieCDP-alfanumérico-String	Número de serie del comprobante de pago o documento. (Opcional: Para filtrar por comprobante)
numCDP-alfanumérico-String	Número del comprobante de pago o documento. Nro Incial (Rango) (Opcional: Para filtrar por comprobante)
perTributario-alfanumerico-String	Periodo tributario (Obligatorio)
codTipoArchivo-numérico-Integer	Código del tipo de archivo - 0: txt - 1: xls (Obligatorio)

Parámetros[body]

No aplica.

Parámetros[header]

Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

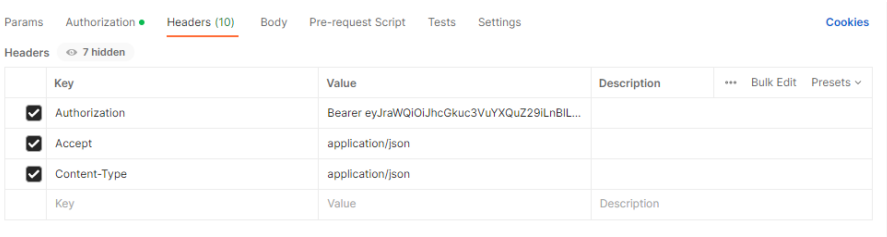
Parámetros[salida]

Parámetros de Salida	Descripción	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

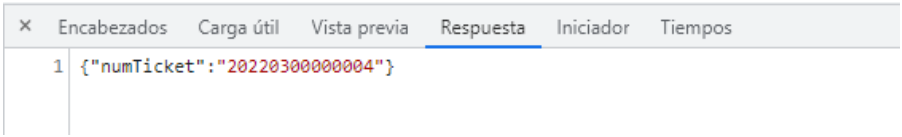
Evidencias

URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/inconsistencias/web/periodoinconsistencias/202212/exporta?codTipoArchivo=0&fecEmisionIni=01/12/2022&fecEmisionFin=31/12/2022&codInconsistencia=&numDocAdquiriente=&codTipoCDP=&numSerieCDP=&numCDP=&s=&ord=

Headers



Body
(No aplica)
Result OK

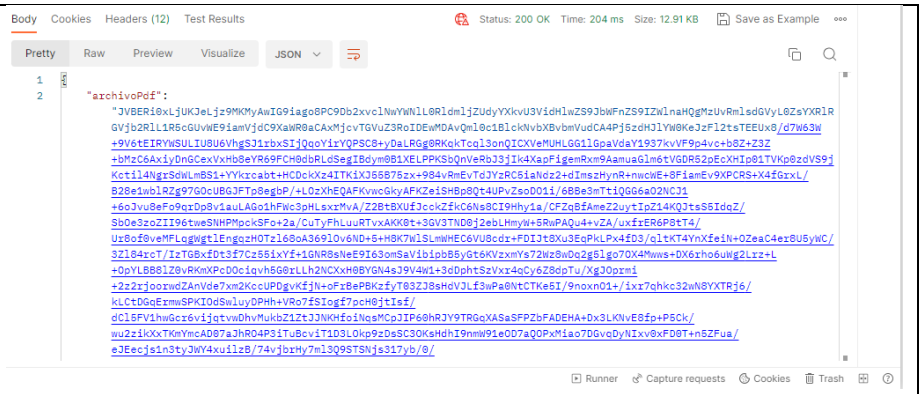


Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** ● 1098 - Formato de fecha de emisión inicial no permitido o no válido para la fecha ● 1100 - Formato de fecha de emisión final no permitido o no válido para la fecha ● 1102 - La Fecha de Emisión Final debe ser mayor o igual a la Fecha de Emisión Inicial ● 1104 - El código de tipo de comprobante de pago enviado no es válido ● 1005 - El campo ‘perTributario’ no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado ● 1059 - Código tipo de Archivo no permitido o no valido ● 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo ● 1061 - El campo "codTipoArchivo" es nulo o vacío

5.26 Servicio Web Api descargar constancia de recepción

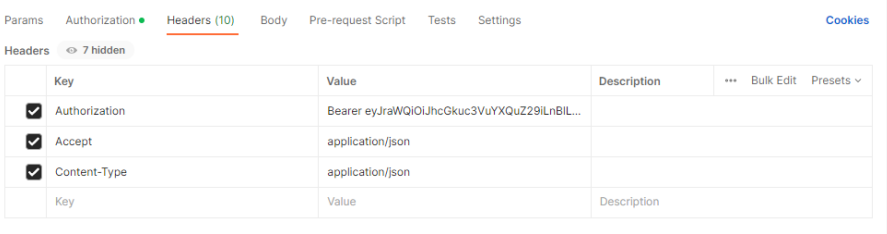
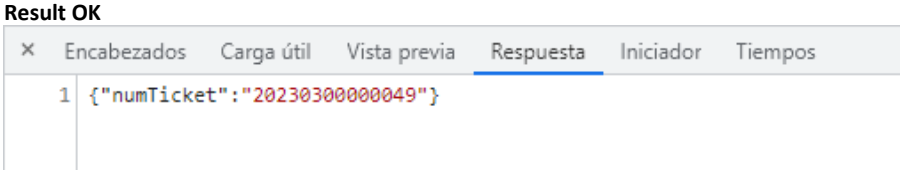
Nombre Web Services	Servicio Web Api descargar constancia de recepción
Descripción	Permite descargar la constancia de recepción.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/constancia/archivo?nomArchivo={nomArchivo}
Parámetros[URL]	**Param-formato-tipo**: nomArchivo-alfanumérico-String **Descripción**: Nombre de archivo (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **valor**: Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: GET
Parámetros[salida]	**Parámetros de Salida**: archivoPdf-Base64-String **Descripcion**: Archivo en base64
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/constancia/archivo?nomArchivo=LE20100176450202212001404000111 12.pdf **Headers** Params / Authorization / Headers (10) / Body / Pre-request Script / Tests / Settings / Cookies Headers 7 hidden Key: Authorization — Value: Bearer eyJraWQiOiJhcGkuc3VuYXQuZ29iLnBIL... — Description: — Key: Accept — Value: application/json — Description: — Key: Content-Type — Value: application/json — Description: — **Body** (No aplica) **Result OK**

<table>
<tr><td></td><td>


Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
</td></tr>
<tr><td>Mensaje Error</td><td>
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:
● 1134 - El campo “nomArchivo” no enviado o es vacío
</td></tr>
</table>

5.27 Servicio Web Api descargar RVIE por periodo

<table>
<tr><td>Nombre Web Services</td><td>Servicio Web Api descargar RVIE por periodo</td></tr>
<tr><td>Descripción</td><td>Permite descargar el reporte consolidado de registro por periodo. Este servicio generará un ticket para poder utilizar el servicio “5.17 Servicio Web Api descargar archivo” y descargar el reporte.</td></tr>
<tr><td>Url</td><td>https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/descarga?perTributario={perTributario}&codOrigenEnvio={codOrigenEnvio}&codTipoArchivo={codTipoArchivo}&codProceso={codProceso}&codLibro={codLibro}</td></tr>
<tr><td>Parámetros[URL]</td><td>
<table>
<tr><th>Param-formato-tipo</th><th>Descripción</th></tr>
<tr><td>perTributario-alfanumérico-String</td><td>Periodo tributario (Obligatorio)</td></tr>
<tr><td>codOrigenEnvio-alfanumérico-String</td><td>Código de origen de envío: 2 Servicio web (Obligatorio)</td></tr>
<tr><td>codTipoArchivo-numérico-Integer</td><td>Código del tipo de archivo
- 0: txt
- 1: excel
(Obligatorio)</td></tr>
<tr><td>codProceso-alfanumérico-String</td><td>Código del indicador de carga masiva. (1-97)
(Ver Anexo I: Indicador de carga masiva)
(Obligatorio)</td></tr>
<tr><td>codLibro-alfanumérico-String</td><td>Código de libro: 140000 RVIE (Obligatorio)</td></tr>
</table>
</td></tr>
<tr><td>Parámetros[body]</td><td>No aplica.</td></tr>
<tr><td>Parámetros[header]</td><td>
Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded
<table>
<tr><th>Parámetros</th><th>valor</th></tr>
<tr><td>Content-Type</td><td>application/json</td></tr>
<tr><td>Accept</td><td>application/json</td></tr>
<tr><td>Authorization</td><td>Bearer token obtenido de la autenticación</td></tr>
</table>
Método: GET
</td></tr>
<tr><td>Parámetros[salida]</td><td>
<table>
<tr><th>Parámetros de Salida</th><th>Descripcion</th><th>Formato</th><th>Tipo dato</th></tr>
<tr><td>numTicket</td><td>Número de ticket de envío [AAAA99999999]
AAAA: Año
99: Tipo de correlativo
99999999: Número correlativo de envío</td><td>alfanumerico</td><td>String</td></tr>
</table>
</td></tr>
</table>

Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/descarga?perTributario=202301&codOrigenEnvio=2&codTipoArchivo=0&codProceso=23&codLibro=140000 **Headers**  Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Headers 7 hidden Key Value Description Bulk Edit Presets Authorization Bearer eyJraWQiOiJhcGkuc3VuYXQuZ29iLnBlL... Accept application/json Content-Type application/json Key Value Description **Body** (No aplica) **Result OK**  Encabezados Carga útil Vista previa Respuesta Iniciador Tiempos 1 {"numTicket":"20230300000049"} **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo "numRuc" no enviado o es vacío" }] } **Lista de errores 422:** ● 1005 - El campo 'perTributario' no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato 'yyyymm' ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado ● 1028 - El campo "codOrigenEnvio" no enviado o es vacío ● 1029 - Código tipo de Origen de Envio no permitido o no valido ● 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio ● 1059 - Código tipo de Archivo no permitido o no valido ● 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo ● 1061 - El campo "codTipoArchivo" es nulo o vacío ● 1138 - El campo "codProceso" es nulo o vacío ● 1139 - Código de Proceso no permitido o no valido ● 1140 - El campo "codLibro" no enviado o es vacío

5.28 Servicio Web Api descargar reporte consolidado por periodo

Nombre Web Services	Servicio Web Api descargar reporte consolidado de registros por período
Descripción	Permite descargar el reporte consolidado de registro por periodo. Este servicio generará un ticket para poder utilizar el servicio "5.17 Servicio Web Api descargar archivo" y descargar el reporte.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/descarga?perTributario={perTributario}&codOrigenEnvio={codOrigenEnvio}&codTipoArchivo={codTipoArchivo}&codProceso={codProceso}&codLibro={codLibro}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio) codOrigenEnvio-alfanumérico-String / Código de origen de envío: 2 Servicio web (Obligatorio) codTipoArchivo-numérico-Integer / Código del tipo de archivo - 0: txt - 1: excel (Obligatorio)

codProceso-alfanumérico-String	Código del indicador de carga masiva. (1-97) (Ver Anexo I: Indicador de carga masiva) (Obligatorio)
codLibro-alfanumérico-String	Código de libro: 140000 RVIE (Obligatorio)

Parámetros[body]

No aplica.

Parámetros[header]

Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

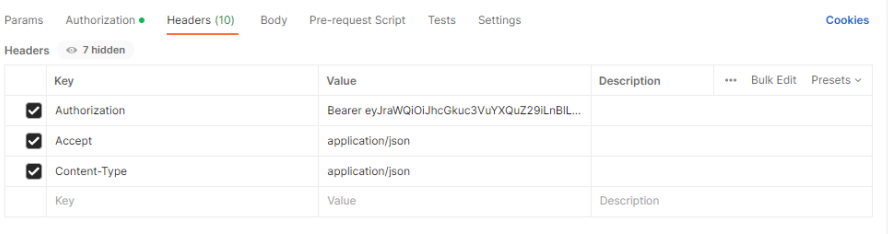
Parámetros[salida]

Parámetros de Salida	Descripcion	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

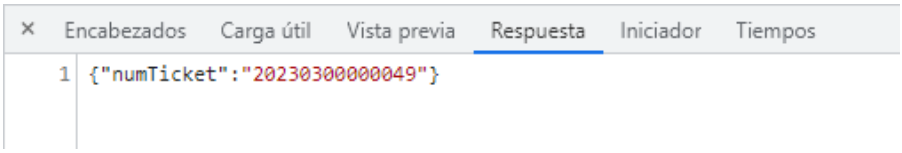
Evidencias

URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/descarga?perTributario=202302&codOrigenEnvio=2&codTipoArchivo=0&codProceso=23&codLibro=140000

Headers



Body
(No aplica)
Result OK



Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

Lista de errores 422:

- 1005 - El campo ‘perTributario’ no enviado o es vacio
- 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
- 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
- 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado
- 1028 - El campo “codOrigenEnvio” no enviado o es vacío
- 1029 - Código tipo de Origen de Envío no permitido o no válido
- 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio
- 1059 - Código tipo de Archivo no permitido o no válido
- 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo
- 1061 - El campo "codTipoArchivo" es nulo o vacío
- 1138 - El campo "codProceso" es nulo o vacío
- 1139 - Código de Proceso no permitido o no valido
- 1140 - El campo “codLibro” no enviado o es vacío

5.29 Servicio Web Api descargar ajustes posteriores

Nombre Web Services	Servicio Web Api descargar ajustes posteriores
Descripción	Permite exportar los ajustes posteriores del RVIE, en caso no se haya cargado ningún archivo entonces descarga la propuesta informativa de ajustes posteriores de la SUNAT. Este servicio generará un ticket para poder utilizar el servicio "5.17 Servicio Web Api descargar archivo" y descargar los ajustes posteriores.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/descarga?perTributario={perTributario}&codOrigenEnvio={codOrigenEnvio}&codTipoArchivo={codTipoArchivo}&codProceso={codProceso}&codLibro={codLibro}
Parámetros[URL]	

Param-formato-tipo	Descripción
perTributario-alfanumérico-String	Periodo tributario (Obligatorio)
codOrigenEnvio-alfanumérico-String	Código de origen de envío: 2 Servicio web (Obligatorio)
codTipoArchivo-numérico-Integer	Código del tipo de archivo - 0: txt - 1: excel (Obligatorio)
codProceso-alfanumérico-String	Código del indicador de carga masiva. (1-97) (Ver Anexo I: Indicador de carga masiva) (Obligatorio)
codLibro-alfanumérico-String	Código de libro: 140000 RVIE (Obligatorio)

Parámetros[body]: No aplica.

Parámetros[header]

Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

Método: GET

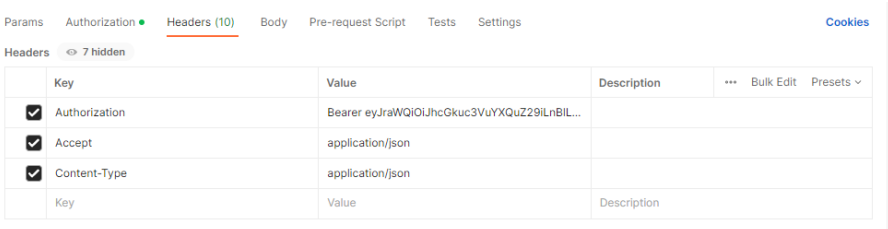
Parámetros[salida]

Parámetros de Salida	Descripcion	Formato	Tipo dato
numTicket	Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío	alfanumerico	String

Evidencias

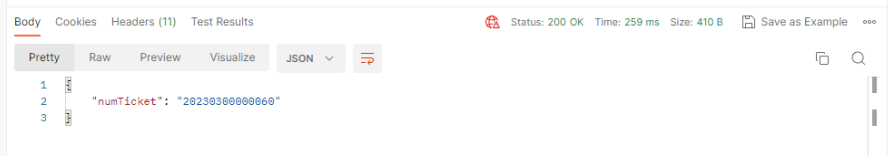
URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/descarga?perTributario=202302&codOrigenEnvio=2&codTipoArchivo=0&codProceso=23&codLibro=140000

Headers



Body
(No aplica)

Result OK



Result Fail

	{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo "numRuc" no enviado o es vacío" }] } **Lista de errores 422:** ● 1005 - El campo 'perTributario' no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato 'yyyymm' ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado ● 1028 - El campo "codOrigenEnvio" no enviado o es vacío ● 1029 - Código tipo de Origen de Envio no permitido o no valido ● 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio ● 1059 - Código tipo de Archivo no permitido o no válido ● 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo ● 1061 - El campo "codTipoArchivo" es nulo o vacío ● 1138 - El campo "codProceso" es nulo o vacío ● 1139 - Código de Proceso no permitido o no valido ● 1140 - El campo "codLibro" no enviado o es vacío

5.30 Servicio Web Api descargar ajustes posteriores de periodos anteriores

Nombre Web Services	Servicio Web Api descargar ajustes posteriores de periodos anteriores
Descripción	Permite exportar los ajustes posteriores de periodos anteriores del RC en caso se hayan cargado ajustes por parte del generador. Este servicio generará un ticket para poder utilizar el servicio "5.17 Servicio Web Api descargar archivo" y descargar los ajustes posteriores.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/descarga?perTributario={perTributario}&codOrigenEnvio={codOrigenEnvio}&codTipoArchivo={codTipoArchivo}&codProceso={codProceso}&codLibro={codLibro}
Parámetros [URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String: Periodo tributario (Obligatorio) codOrigenEnvio-alfanumérico-String: Código de origen de envío: 2 Servicio web (Obligatorio) codTipoArchivo-numérico-Integer: Código del tipo de archivo - 0: txt - 1: xls (Obligatorio) codProceso-alfanumérico-String: Código del indicador de carga masiva (Ver Anexo I: Indicador de carga masiva) (1 - 97) (Obligatorio) codLibro-alfanumérico-String: Código de libro: 140000 RVIE (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **valor** Content-Type: application/json Accept: application/json Authorization: Bearer token obtenido de la autenticación Método: GET
Parámetros[salida]	**Parámetros de Salida** / **Descripcion** / **Formato** / **Tipo dato** numTicket: Número de ticket de envío [AAAA99999999] AAAA: Año / alfanumerico / String

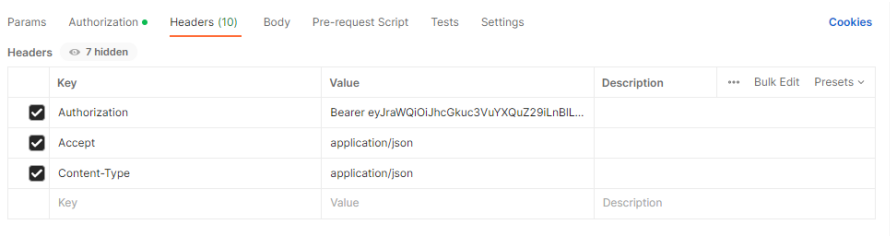
		99: Tipo de correlativo 99999999: Número correlativo de envío		

Evidencias

URL

https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/descarga?perTributario=202302&codOrigenEnvio=2&codTipoArchivo=0&codProceso=23&codLibro=140000

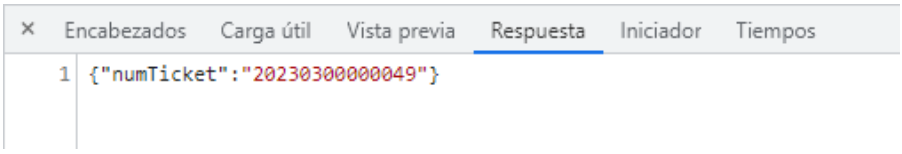
Headers



Body

(No aplica)

Result OK



Result Fail

{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

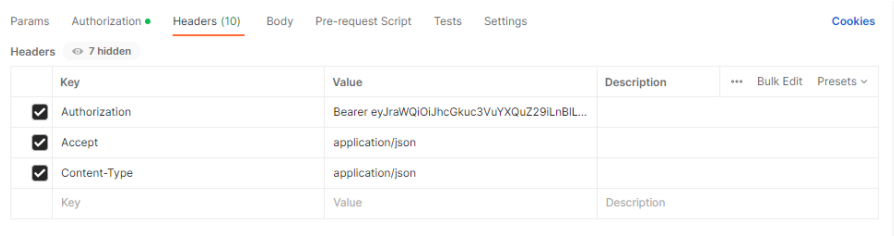
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

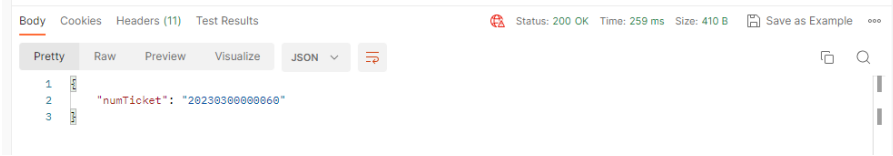
Lista de errores 422:

- 1005 - El campo ‘perTributario’ no enviado o es vacio
- 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
- 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
- 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado
- 1028 - El campo “codOrigenEnvio” no enviado o es vacío
- 1029 - Código tipo de Origen de Envio no permitido o no valido
- 1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio
- 1059 - Código tipo de Archivo no permitido o no válido
- 1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo
- 1061 - El campo "codTipoArchivo" es nulo o vacío
- 1138 - El campo "codProceso" es nulo o vacío
- 1139 - Código de Proceso no permitido o no valido
- 1140 - El campo “codLibro” no enviado o es vacío

5.31 Servicio Web Api descargar reporte inconsistencias por periodo

Nombre Web Services	Servicio Web Api descargar inconsistencias por periodo
Descripción	Permite descargar las inconsistencias por periodo. Este servicio generará un ticket para poder utilizar el servicio “5.17 Servicio Web Api descargar archivo” y descargar las inconsistencias.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/inconsistencias/web/inconsistencia/{codRegistroLibro}/exporta?codOrigenEnvio={codOrigenEnvio}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** codRegistroLibro-alfanumérico-String / Código registro de libro (Obligatorio) codOrigenEnvio-alfanumérico-String / Código de origen de envío: 2 Servicio web (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar

<table>
<tr><td></td><td>
Valores:
Content-type: application/x-www-form-urlencoded
<table>
<tr><th>Parámetros</th><th>valor</th></tr>
<tr><td>Content-Type</td><td>application/json</td></tr>
<tr><td>Accept</td><td>application/json</td></tr>
<tr><td>Authorization</td><td>Bearer token obtenido de la autenticación</td></tr>
</table>
Método: GET
</td></tr>
<tr><td>Parámetros[salida]</td><td>
<table>
<tr><th>Parámetros de Salida</th><th>Descripcion</th><th>Formato</th><th>Tipo dato</th></tr>
<tr><td>numTicket</td><td>Número de ticket de envío [AAAA99999999]
AAAA: Año
99: Tipo de correlativo
99999999: Número correlativo de envío</td><td>alfanumerico</td><td>String</td></tr>
</table>
</td></tr>
<tr><td>Evidencias</td><td>
URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvie/inconsistencias/web/inconsistencia/6463788755ff7605f43b6a0a/exporta?codOrigenEnvio=2
Headers


Body
(No aplica)
Result OK


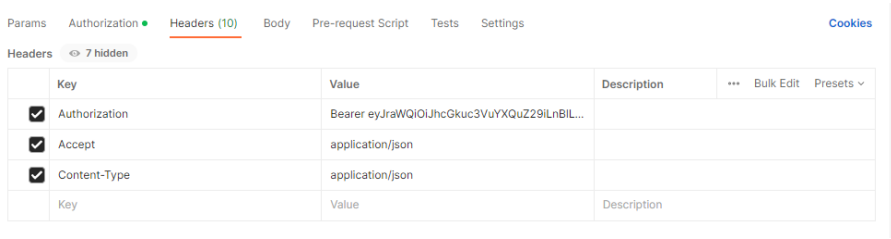
Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
</td></tr>
<tr><td>Mensaje Error</td><td>
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:

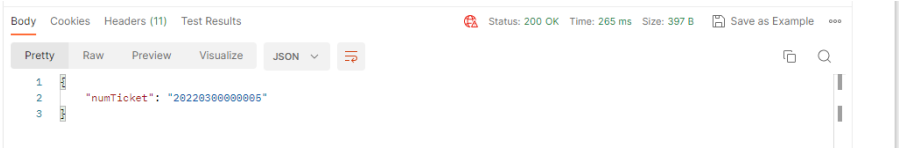
El campo “codRegistroLibro” no enviado o es vacío
1028 - El campo “codOrigenEnvio” no enviado o es vacío
1029 - Código tipo de Origen de Envio no permitido o no valido
1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio

</td></tr>
</table>

5.32 Servicio Web Api descargar reporte CAR

<table>
<tr><td>Nombre Web Services</td><td>Servicio Web Api descargar reporte de CAR</td></tr>
<tr><td>Descripción</td><td>Permite descargar la lista de CAR dependiendo de la fase en que se encuentre.
Este servicio generará un ticket para poder utilizar el servicio “5.17 Servicio Web Api descargar archivo” y descargar la lista de CAR.</td></tr>
<tr><td>Url</td><td>https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/comprobanteslibros/{perTributario}/reportecar?codOrigenEnvio={codOrigenEnvio}&codLibro={codLibro}</td></tr>
<tr><td>Parámetros[URL]</td><td>
<table>
<tr><th>Param-formato-tipo</th><th>Descripción</th></tr>
<tr><td>perTributario-alfanumérico-String</td><td>Periodo tributario (Obligatorio)</td></tr>
<tr><td>codOrigenEnvio-alfanumérico-String</td><td>Código de origen de envío: 2 Servicio web (Obligatorio)</td></tr>
<tr><td>codFase-alfanumérico-String</td><td>Código de fase (Obligatorio)</td></tr>
<tr><td>codLibro-alfanumérico-String</td><td>Código de libro: 140000 RVIE (Obligatorio)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Parámetros[body]</td><td>No aplica.</td></tr>
<tr><td>Parámetros[header]</td><td>Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded
<table>
<tr><th>Parámetros</th><th>valor</th></tr>
<tr><td>Content-Type</td><td>application/json</td></tr>
<tr><td>Accept</td><td>application/json</td></tr>
<tr><td>Authorization</td><td>Bearer token obtenido de la autenticación</td></tr>
</table>
Método: GET</td></tr>
<tr><td>Parámetros[salida]</td><td>
<table>
<tr><th>Parámetros de Salida</th><th>Descripcion</th><th>Formato</th><th>Tipo dato</th></tr>
<tr><td>numTicket</td><td>Número de ticket de envío [AAAA99999999]
AAAA: Año
99: Tipo de correlativo
99999999: Número correlativo de envío</td><td>alfanumerico</td><td>String</td></tr>
</table>
</td></tr>
<tr><td>Evidencias</td><td>URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/comprobanteslibros/202301/reportecar?codOrigenEnvio=2&codLibro=140000
Headers
 

Body
(No aplica)
Result OK
 

Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }</td></tr>
<tr><td>Mensaje Error</td><td>{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:

1005 - El campo ‘perTributario’ no enviado o es vacio
1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
1028 - El campo “codOrigenEnvio” no enviado o es vacío
1029 - Código tipo de Origen de Envio no permitido o no valido
1030 - Solo se permite dato numérico de 1 dígito para el codOrigenEnvio
1140 - El campo “codLibro” no enviado o es vacío
</td></tr>
</table>

5.33 Servicio Web Api descargar reporte estadístico

<table>
<tr><td>Nombre Web Services</td><td>Servicio Web Api descargar reporte estadístico</td></tr>
<tr><td>Descripción</td><td>Permite exportar el resumen estadístico (Razón social, Monto, Porcentaje)</td></tr>
<tr><td>Url</td><td>https://api.sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/estadistica/web/resumenestadistico/exportarvie?numRuc={numRuc}&perTributario={perTributario}&codTipoArchivo={codTipoArchivo}&codTipoReporte={codTipoReporte}&codLibro={codLibro}&fechaIni={fechaIni}&fechaFin={fechaFin}&codTipoCDP={codTipoCDP}</td></tr>
<tr><td>Parámetros[URL]</td><td>
<table>
<tr><th>Param-formato-tipo</th><th>Descripción</th></tr>
<tr><td>numRuc-numérico-int</td><td>Número de RUC del generador (Obligatorio)</td></tr>
</table>
</td></tr>
</table>

perTributario-alfanumérico-String	Periodo tributario (Obligatorio)
fechaIni-yyyy mm dd-date	Fecha emision desde del comprobante de pago (Opcional: Para filtrar por fecha de inicio)
fechaFin-yyyy mm dd-date	Fecha emision hasta del comprobante de pago (Opcional: Para filtrar por fecha de fin)
codTipoDocIdentidadAdquiriente - alfanumérico-String	Código de tipo de documento de indentidad del adquiriente (Opcional: Para filtrar por adquiriente)
numDocIdentidadAdquiriente-alfanumérico-String	Número de documento de identidad del aquiriente (Opcional: Para filtrar por adquiriente)
codTipoCDP-alfanumerico-alfanumerico	Tipo de comprobante (Opcional: Para filtrar por comprobante)
codTipoArchivo-numérico-Integer	Código del tipo de archivo - 0: txt - 1: xls - 2: csv (Obligatorio)
codTipoReporte-numérico-Integer	1. Reporte montos/proveedor 2. Reporte montos/Notas credito y notas de debito (Obligatorio)
codLibro-alfanumerico-String	Codigo del Libro (Obligatorio)

Parámetros[body]

No aplica.

Parámetros[header]

Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded

Parámetros	valor
Content-Type	application/json
Accept	application/json
Authorization	Bearer token obtenido de la autenticación

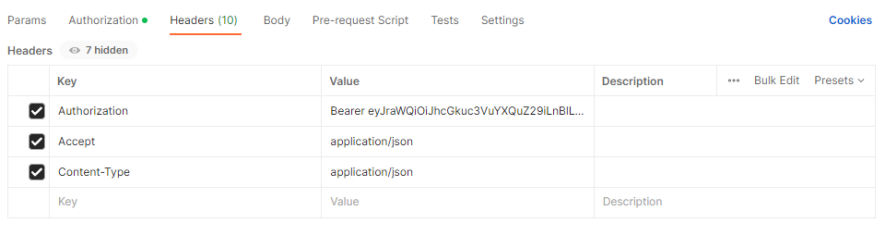
Método: GET

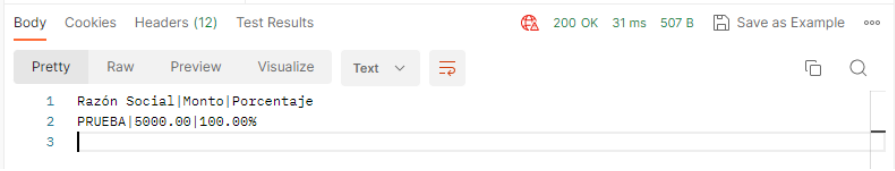
Parámetros[salida]

Parámetros	valor
HTTP status	200
Content-Type	application/json
Content-Disposition	El sistema descarga el archivo con el siguiente formato de nombre: 1-REPORTE MONTOS/ ADQUIRIENTE (estadisticaPorProveedor.<extensión>) Razón Social|Monto|Porcentaje Los constructores SAC|127 000|16% El ingeniero perez|86 999|9% El consorcio unido|75 000|7% 2-REPORTE MONTOS/NOTAS CREDITO Y NOTAS DE DEBITO (estadisticaPorProveedorNotaCreDeb.<extensión>) Razón Social|Monto|Porcentaje Los constructores SAC|12 000|16% El ingeniero perez|8 999|9% El consorcio unido|7 000|7%

Evidencias

URL
https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/estadistica/web/resumenestadistico/exportarvie?numRuc=20195923753&perTributario=202203&codTipoArchivo=0&codTipoReporte=1&codLibro=140000
Headers

<table>
<tr>
<td></td>
<td>


Body
(No aplica)
Result OK


Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
</td>
</tr>
<tr>
<td>Mensaje Error</td>
<td>
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:

1002 –El campo “numRuc” solo se permite dato numérico de 11 dígitos para el número de RUC.
1005 - El campo ‘perTributario’ no enviado o es vacio
1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
1059 - Código tipo de Archivo no permitido o no valido
1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo
1061 - El campo "codTipoArchivo" es nulo o vacío
1333 - El campo "codTipoReporte" es nulo o vacío
1334 - El campo "codTipoReporte" solo admite valores: 1, 2, 3 ó 4
1325 -Fecha de inicio debe estar dentro del Periodo seleccionado
1326 - Debe cumplir con el siguiente formato “dd/mm/yyyy”.
1327 - Fecha Fin debe ser mayor o igual a la Fecha Inicio
1328 - Si se realiza busqueda por Fecha de emision de cp, se debe ingresar los campos: Fecha Inicio, Fecha Fin
1329 - Fecha Fin debe estar dentro del Periodo seleccionado
1330 - Debe cumplir con el siguiente formato “dd/mm/yyyy”.
1331 - Fecha Fin debe ser mayor o igual al Fecha Inicio
1332 - Si se realiza busqueda por Fecha de emision de cp, se debe ingresar los campos: Fecha Fin, Fecha Inicio
1011 - El campo “codTipoCDP” no enviado o es vacío
1059 - Código tipo de Archivo no permitido o no valido
1060 - Solo se permite dato numérico de 1 dígito para el codTipoArchivo
1061 - El campo "codTipoArchivo" es nulo o vacío
1333 - El campo "codTipoReporte" es nulo o vacío
1334 - El campo "codTipoReporte" solo admite valores: 1, 2, 3 ó 4
1140 - El campo “codLibro” no enviado o es vacío
1103 - Solo se permite dato alfanumérico con un tamaño máximo de 15 para el número de documento del adquiriente

</td>
</tr>
</table>

5.34 Servicio Web Api descargar reporte de cumplimiento

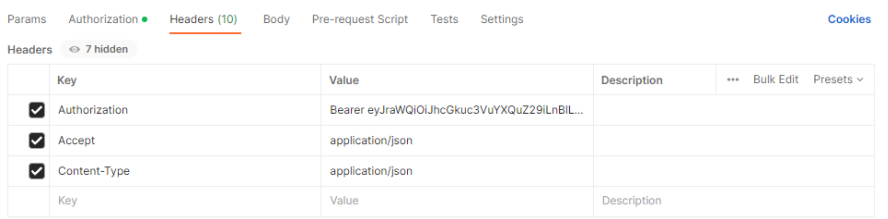
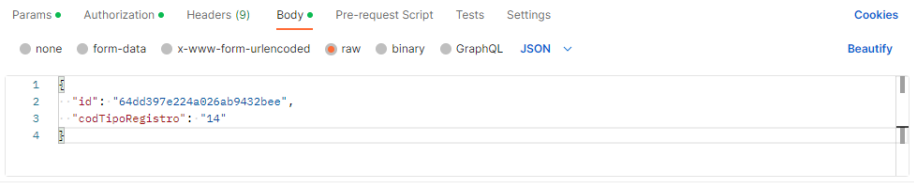
Nombre Web Services	Servicio Web Api descargar reporte de cumplimiento
Descripción	Permite descargar el reporte de cumplimiento.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/cumplimiento/web/omisos/{perTributario}/{codLibro}/consultaReporteCumplimiento/exportardocumento
Parámetros[URL]	**Param-formato-tipo** | **Descripción**

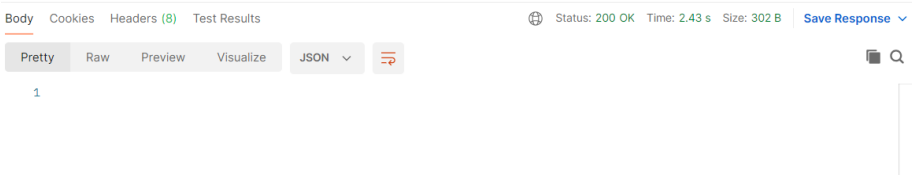
	perTributario-alfanumérico-String: Periodo tributario (Obligatorio) codLibro-alfanumérico-String: Código de libro: 140000 RVIE (Obligatorio)
Parámetros[body]	No aplica
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **Valor**: Content-Type: application/json Accept: application/json Authorization: Bearer token obtenido de la autenticación Método: GET
Parámetros[salida]	**Parámetros de Salida** / **Descripción**: archivoPdf-Base64-String: Base64 representando el archivo nombreArchivoPdf-alfanumérico-String: Nombre de archive de descarga
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/cumplimiento/web/omisos/202212/140000/consultaReporteCumplimiento/exportardocumento **Headers** Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Headers 7 hidden Key: Authorization — Value: Bearer eyJraWQiOiJhcGkuc3VuYXQuZ29iLnBlL... — Description Key: Accept — Value: application/json Key: Content-Type — Value: application/json Key — Value — Description **Body** (No aplica) **Result OK** Body Cookies Headers (12) Test Results — Status: 200 OK Time: 638 ms Size: 17.95 KB — Save as Example Pretty Raw Preview Visualize JSON "data": { "archivoPdf": "JVBERi0xLjUKJeLjz9MKMyAwIG9iago8PC9Db2xvclNwYWNlL0RldmljZUdyYXkvU3VidHlwZS9JbWFnZS9IZWlnaHQgMzUvRmlsdGVyL0ZsY XRlRGVjb2RlL1R5cGUvWE9iamVjdC9XaWR0aCAxMjcvTGVuZ3RoIDEwMDAvQml0c1BlckNvbXBvbmVudCA4Pj5zdHJlYW0KeJzFl2tsTEEUx8/ d7W63W+9V6tEIRYWSULIU8U6VhgSJ1rbxSIjQqoYirYQPSC8+yDaLRGg0RKqkTcql3onQICXVeMUHLGG1lGpaVdaY1937kvVF9p4vc+b8Z+Z3Z +bMzC6AxiyDnGCexVxHb8eYR69FCH0dbRLdSegIBdym0B1XELPPKSbQnVeRbJ3jIk4XapFigemRxm9AamuaGlm6tVGDR52pEcXHIp01TVKp0zd VS9jKctil4NgzSdWLmBS1+YYkzcabt+HCDckXz4ITkiXJ5bB75zx+984vRmEvTdJYzRC6iaNdz2+dImszHynR+nwcWE+8FiamEv9XPCR5 +X4fGrxL/B28e1wblRZg97GOcUBGJFTp8egbP/+LOzXhEQAFKvwcGkyAFKZeiSHBp8Qt4UPvZsoDO1i/6BBe3mTtiQGG6aO2NCJ1 +6oJvu8eFo9qrDp8v1auLAGo1hFWc3pHLsxrMvA/Z2BtBXUfJcckZfkC6Ns8CI9Hhy1a/CFZqBfAmeZ2uytIpZ14KQJtsS5IdqZ/ SbOe3zoZII96tweSNHPMpckSFo+2a/CuTyFhLuuRTvxAKK0t+3GV3TND0j2ebLHmyW+5RwPAQu4+vZA/uxfxER6P8tT4/ Ur8of0veMFLqgWgtlEngqzHOTzl68oA369lOv6ND+5+H8K7WlSLmWHEC6VU8cdz+FDIJt8Xu3EqPkLPx4fD3/qltKT4YnXfeiN +0ZeaC4er8U5yWC/3Zl84rcT/IzTGBxfDt3f7Cz55ixYf+1GNR8sNeE9I63omSaVibipbB5yGt6KVzxmYs72Wz8wDq2g5lgo7OX4Mwws +DX6rho6uWg2Lrz+L+OpYLBB0lZ0vRKmXPcDOciqvh5G0rLLh2NCXxH0BYGN4sJ9V4W1+3dDphtSzVxr4qCy6Z8dpTu/XgJOprmi +2z2rjooxwdZAnVde7xm2KccUPDgvKfjN+oFrBePBKzfyT03ZJ8sHdVJLf3wPa0NtCTKe5I/9noxnO1+/ixx7qhkc32wN8YXTRj6/ kLCtDGqErmwSPKIOdSwluyDPHh+VRo7fSIogf7pcH0jtIsf/ dCl5FV1hwGcr6vijqtvwDhvMukbZ1ZtJJNKHfoiNqsMCpJIP60hRJY9TRGqXASaSFPZbFADEHA+Dx3LKNvE6fp+P5Ck/ wu2zikXxTKmYmcAD07aJhRO4P3iTuBcviT1D3LOkp9zDsSC3OKsHdhI9nmW91eOD7aQOPxMiao7DGvqDyNIxv0xFD0T+n5ZFua/ Runner Capture requests Cookies Trash	
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** • 1005 - El campo ‘perTributario’ no enviado o es vacio • 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ • 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual • 1070 - No se ha encontrado información de comprobantes de pago en Periodo Seleccionado • 1140 - El campo “codLibro” no enviado o es vacío

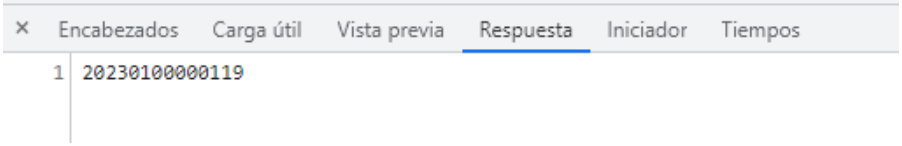
5.35 Servicio Web Api reporte de exportadores

Nombre Web Services	Servicio Web Api reporte de exportadores
Descripción	Permite descargar el reporte de exportadores.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/exportadores/reporte?codRegistroLibro={codRegistroLibro}&perTributario={perTributario}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio) codRegistroLibro-alfanumérico-String / Código registro de libro (Obligatorio)
Parámetros[body]	No aplica.
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: GET
Parámetros[salida]	**Parámetros** / **valor** HTTP status / 200 Content-Type / application/json Content-Disposition / El sistema descarga el archivo con el siguiente formato de nombre: REPORTE DE EXPORTADORES Código de Aduana|Año Dam|Correlativo Dam|Fecha de Embarque|Valor fob $|T.C.|Valor fob s/.|Tipo cdp|Serie cdp|Numero cdp|Fecha cdp | | | |0|0|0| | | | TOTAL EXPORTACIONES: |0
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/exportadores/reporte?codRegistroLibro=6441c029ea85012ac4a77d9b&perTributario=202302 **Headers** Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Headers 7 hidden Key / Value / Description / Bulk Edit Presets Authorization / Bearer eyJraWQiOiJhcGkuc3VuYXQuZ29iLnBIL... / Accept / application/json / Content-Type / application/json / Key / Value / Description **Body** (No aplica) **Result OK** Encabezados Carga útil Vista previa Respuesta Iniciador Tiempos 1 REPORTE DE EXPORTADORES 2 3 RUC: 20100176450 4 Razon social: REPSOL GAS DEL PERU S.A. 5 Periodo: 02/2023 6 Fuente: Aduanas 7 Fecha: 8 9 Código de Aduana|Año Dam|Correlativo Dam|Fecha de Embarque|Valor fob $|T.C.|Valor fob s/.|Tipo cdp|Serie cdp|Numero cdp|Fecha cdp 10 | | | |0|0|0| | | | 11 TOTAL EXPORTACIONES: |0 12 **Result Fail** { "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
Mensaje Error	{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] } **Lista de errores 422:** ● 1005 - El campo ‘perTributario’ no enviado o es vacio ● 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’ ● 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual ● El campo “codRegistroLibro” no enviado o es vacío

5.36 Servicio Web Api eliminar preliminar registrado

Nombre Web Services	Servicio Web Api eliminar el preliminar registrado.
Descripción	Permite eliminar el preliminar registrado. Se utilizan los campos" registros[0].id" y "registros[0].codTipoRegistro" del servicio "5.37 Servicio Web Api consultar preliminares registrados " para hallar los parámetros necesarios y hacer uso de este servicio.
Url	https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/{perTributario}/eliminapreliminar?codLibro={codLibro}
Parámetros[URL]	**Param-formato-tipo** / **Descripción** perTributario-alfanumérico-String / Periodo tributario (Obligatorio) codLibro-alfanumérico-String / Código de libro: 140000 RVIE (Obligatorio)
Parámetros[body]	**Param-formato-tipo** / **Descripción** id-alfanumérico-String / Id de registro (Obligatorio) codTipoRegistro-alfanumérico-String / Código de tipo registro: 14 (Obligatorio)
Parámetros[header]	**Descripción:** Content-type: tipo de contenido a enviar **Valores:** Content-type: application/x-www-form-urlencoded **Parámetros** / **valor** Content-Type / application/json Accept / application/json Authorization / Bearer token obtenido de la autenticación Método: PUT
Parámetros[salida]	**Parámetros de Salida** / **Descripción** / **Formato** / **Tipo dato** numTicket(Opcional) / Número de ticket de envío [AAAA99999999] AAAA: Año 99: Tipo de correlativo 99999999: Número correlativo de envío **Nota**: En caso la petición devuelva número de ticket se debe consultar su estado en 5.16 Servicio Web Api consultar estado de envío de ticket / alfanumerico / String
Evidencias	**URL** https://api-sire.sunat.gob.pe/v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/202304/eliminapreliminar?codLibro=140000 **Headers**  **Body**  **Result OK** 1. Si la respuesta es vacío, el proceso ha terminado correctamente.

<table>
<tr>
<td></td>
<td>


2. Si la respuesta es un número de ticket, debe consultar su estado en 5.16 Servicio Web Api consultar estado de envío de ticket, en la cual deberá tener el estado 06 – Terminado.


Result Fail
{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }
</td>
</tr>
<tr>
<td>Mensaje Error</td>
<td>
{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }
Lista de errores 422:

1005 - El campo ‘perTributario’ no enviado o es vacio
1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
El campo “codRegistroLibro” no enviado o es vacío
1010 - El libro electronico con los siguientes datos: numero de RUC: XXXXXXXX periodo Tributario: AAAAMM y codigo de Libro:140000 no existe.
2296 - No es posible eliminar su preliminar, debido a que aun no registra su preliminar para el periodo/registro ingresado.
2297 - No es posible eliminar su preliminar, debido a que su registro se encuentra generado para el periodo ingresado.

</td>
</tr>
</table>

5.37 Servicio Web Api consultar preliminares registrados

<table>
<tr><td>Nombre Web Services</td><td>Servicio Web Api consultar preliminares registrados</td></tr>
<tr><td>Descripción</td><td>Permite consultar todos los preliminares registrados.</td></tr>
<tr><td>Url</td><td>https://api-sire.sunat.gob.pe /v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/consultapreliminaresregistro?page={page}&perPage={perPage}&perIni={perIni}&perFin={perFin}</td></tr>
<tr><td>Parámetros[URL]</td><td>
<table>
<tr><th>Param-formato-tipo</th><th>Descripción</th></tr>
<tr><td>page-alfanumérico-String</td><td>Número de página (Obligatorio)</td></tr>
<tr><td>perPage-alfanumérico-String</td><td>Cantidad por página (Obligatorio)</td></tr>
<tr><td>perIni-alfanumérico-String</td><td>Periodo de consulta de documentos de comprobantes del RVIE preliminar Inicio. (Obligatorio)</td></tr>
<tr><td>perFin-alfanumérico-String</td><td>Periodo de consulta de documentos de comprobantes del RVIE preliminar Final. (Obligatorio)</td></tr>
</table>
</td></tr>
<tr><td>Parámetros[body]</td><td>No aplica.</td></tr>
<tr><td>Parámetros[header]</td><td>
Descripción:
Content-type: tipo de contenido a enviar
Valores:
Content-type: application/x-www-form-urlencoded
<table>
<tr><th>Parámetros</th><th>valor</th></tr>
<tr><td>Content-Type</td><td>application/json</td></tr>
<tr><td>Accept</td><td>application/json</td></tr>
<tr><td>Authorization</td><td>Bearer token obtenido de la autenticación</td></tr>
</table>
Método: GET
</td></tr>
</table>

Parámetros[salida]

Parámetros	Valor
HTTP status	200
Content-Type	application/json

Evidencias

URL

https://api-sire.sunat.gob.pe /v1/contribuyente/migeigv/libros/rvierce/gestionlibro/web/registroslibros/consultaprelimina resregistro?page=1&perPage=100&perIni=202001&perFin=202505

Headers

Params | Authorization | Headers (10) | Body | Pre-request Script | Tests | Settings | Cookies

Headers | 7 hidden

Key	Value	Description
Authorization	Bearer eyJraWQiOiJhcGkuc3VuYXQuZ29iLnBIL...	
Accept	application/json	
Content-Type	application/json	
Key	Value	Description

Result OK

Body | Cookies | Headers (10) | Test Results | 200 OK | 30 ms | 1.13 KB | Save Response

{} JSON | Preview | Visualize


```
1  {
2      "paginacion": {
3          "page": 1,
4          "perPage": 100,
5          "totalRegistros": 1
6      },
7      "registros": [
8          {
9              "id": "68a4ab1f6a2be29791ca22e3",
10             "numRuc": "10276669431",
11             "nomRazonSocial": "MARIA MIRIAM REQUEJO VALENCIA",
12             "perTributario": "202501",
13             "codTipoRegistro": "14",
14             "desTipoRegistro": "Registro RVIE Sin movimiento",
15             "codLibro": "140000",
16             "indContenido": "0",
17             "desContenido": "Sin Información",
18             "codTipoPreliminar": null,
19             "codEstadoRegistro": "3",
20             "desEstadoRegistro": "En Generacion de Registro",
21             "indInfoPreliminar": "1",
22             "desInfoPreliminar": "Registrado",
23             "numAjustePosterior": null,
```


Result Fail

{ "cod":"500", "msg":"Internal Server Error - Se presento una condicion inesperada que impidio completar el Request", "exc":"java.lang.NullPointerException at ..." }

Mensaje Error

{ "cod":"422", "msg":"Unprocessable Entity - Se presentaron errores de validacion que impidieron completar el Request", "errors":[{ "cod":"1001", "msg":"El campo “numRuc” no enviado o es vacío" }] }

Lista de errores 422:

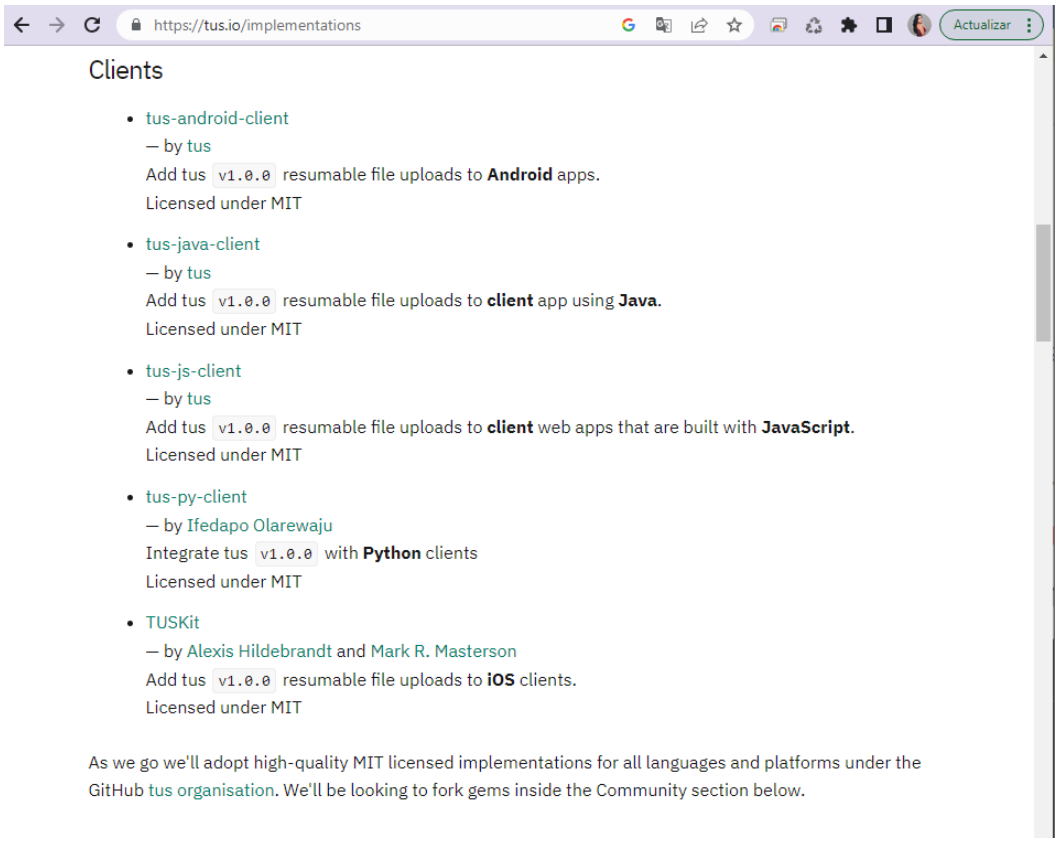
- 1005 - El campo ‘perTributario’ no enviado o es vacio
- 1006 - Formato de perTributario no cumple con el formato ‘yyyymm’
- 1007 - El perTributario de búsqueda no debe ser mayor a la fecha actual
- El campo “codRegistroLibro” no enviado o es vacío
- 1010 - El libro electronico con los siguientes datos: numero de RUC: XXXXXXXX periodo Tributario: AAAAMM y codigo de Libro:140000 no existe.
- 2296 - No es posible eliminar su preliminar, debido a que aun no registra su preliminar para el periodo/registro ingresado.
- 2297 - No es posible eliminar su preliminar, debido a que su registro se encuentra generado para el periodo ingresado.

a. Documentación TUS.IO

Tus.io es un protocolo abierto para carga reanudable basado en HTTP el cual te permite poder implementarlo con cualquier lenguaje de programación, ya sea angular, java, Python, Android, etc. Pero, lo único que necesita es de la configuración de un servicio como servidor y otro como cliente.

Este manual api lo que proporciona es el endpoint servidor en donde los usuarios pueden generar sus peticiones para la importación/carga de archivos reanudables de manera rápida y segura.

Por lo tanto, el usuario que quiera cargar archivos a los servicios api mencionados en el apartado "5. Documentación Servicios Web API", deberá configurar su cliente de acuerdo a su necesidad, por lo que podrá ubicar más detalle de ello en la documentación que es propia del protocolo tus.io en la página web https://tus.io/implementations



Servicios Web Api que funcionan como servidor para la importación de archivos:

- 5.3 Servicio Web Api importar reemplazo de la propuesta
- 5.4 Servicio Web Api importar nuevos comprobantes propuesta
- 5.5 Servicio Web Api importar nuevos comprobantes preliminar
- 5.6 Servicio Web Api importar ajustes posteriores
- 5.7 Servicio Web Api importar ajustes posteriores de periodos anteriores

b. Anexos

7.1 Anexo I: Indicador de carga masiva

Código	Descripción
1	Importar CP - Propuesta
2	Aceptar propuesta
3	Reemplazo de la Propuesta
4	Importa CP - Preliminar
5	Generar libro RVIE
6	Cargar Ajuste posteriores al periodo actual
7	Cargar Ajuste posteriores anteriores a la vigencia
8	Generar registro Ajustes Posterior RVIE
9	Generar registro Ajustes Posterior Anterior RVIE
10	Generar archivo exportar propuesta
11	Generar archivo exportar no incluidos
12	Generar archivo exportar preliminar
13	Generar archivo exportar inconsistencias
14	Generar archivo exportar propuesta ajustes posteriores
15	Generar archivo exportar CAR
16	Generar reporte de observaciones de comparación
17	Generar archivo exportar Libro Venta
18	Generar reporte de ajustes posteriores individual
19	Generar reporte de ajustes posteriores consolidado
20	Generar reporte de ajustes posteriores de periodos anteriores individual
21	Generar reporte de ajustes posteriores de periodos anteriores consolidado
22	Generar reporte consolidado del libro y ajustes
23	Generar reporte Libro RVIE
24	Generar Archivo personalizado Libros RVIE
25	Generar Archivo personalizado Propuesta RVIE
26	Generar Archivo personalizado Ajustes Posteriores RVIE
27	Carga archivo de comparación - validación
28	Generar archivo exportar preliminar registrado
29	Generar archivo exportar preliminar ajustes posteriores registrado
30	Generar reporte de inconsistencias generación del RVIE
31	Generar reporte de inconsistencias ajustes posteriores del RVIE
32	Generar libro RVIE - Archivo exportar Libro Venta
33	Generar libro RVIE - Archivo reporte inconsistencias
34	Generar libro RVIE - Achivo Reporte Exportadores
35	Generar libro RVIE - Archivo Propuesta Casillas
36	Generar Ajustes Posteriroes RVIE - Archivo exportar Ajuste
37	Generar Ajustes Posteriroes RVIE - Archivo reporte inconsistencias
38	Generar Ajustes Posteriores de periodos anteriores RVIE - Archivo exportar Ajuste
39	Aceptar propuesta sin Movimiento
40	Carga Tipo de Cambio
41	Generar reportes Estadisticos
42	Generar Reporte de comparación
43	Descargar Registros Electronicos RVIE
44	Descargar Registros Electronicos RCE
45	Descargar Constancia de Recepción RVIE
46	Descargar Constancia de Recepción RCE
47	Descargar Reporte de Inconcistencias RVIE
48	Descargar Reporte de Inconcistencias RCE
49	Descargar Reporte de Ajustes Posteriores RVIE
50	Descargar Reporte de Ajustes Posteriores RCE
51	Descargar Reporte de Casillas Vista Comparada
52	Descargar Reporte de Inconsistencias de Casillas
53	Generar Reporte de comparación RCE

54	Carga Complementar
55	Carga Incluir Excluir
56	Carga No Domiciliados
57	Carga Comparacion RCE
58	Carga Comparacion RVIE
59	importar CP en Ajustes Posteriores RCE
60	importar CP no domiciliados en Ajustes Posteriores
61	Reemplazo de la Propuesta
62	Generacion de Inconsistencia por Casilla
63	Generación de ventas por Casilla (100. 101)
64	Generacion Inconsistencias en Registros para Casillas
65	Validar Propuesta
66	Validar Preliminar
67	Validar No Domiciliados
68	Reporte de Ajustes posteriores de periodos anteriores del RCE
69	Descarga Consolidada de registros del RCE
70	Descarga RCE
71	Reporte de ajustes posteriores del RVIE
72	Reporte de Ajustes posteriores de periodos anteriores del RVIE
73	Descarga Consolidada de registros del RVIE
74	Descarga RVIE
75	Generación de ventas por Casilla (100. 101)
76	Generacion Inconsistencias en Registros para Casillas
77	Validar Propuesta
78	Validar Preliminar
79	Validar No Domiciliados
80	Generación de archivo personalizado Propuesta RCE).
81	Generación de archivo personalizado Preliminar RCE).
82	Generación de archivo personalizado Preliminar Registrado RCE).
83	Generación de archivo personalizado Registro Compras).
84	Generación de archivo personalizado Ajuste Posterior RCE).
85	Generacion de archivo del libro de Ajustes Posteriores RVIE).
86	Generacion de archivo de inconsistencias de libro de Ajustes Posteriores RVIE).
87	Importar CP en Ajustes Posteriores RVIE.
88	Importar CP en Ajustes Posteriores de periodos anteriores RVIE general.
89	Importar CP en Ajustes Posteriores de periodos anteriores RVIE simplificado.
90	Generación de documentos para Intranet).
91	Exportar detalle propuesta casilla - Registro).
92	Exportar inconsistencias en registro);
93	Importar CP en Ajustes Posteriores RCE de Periodos Anteriores Simplificado
94	Importar CP en Ajustes Posteriores RCE de Periodos Anteriores General
95	Importar CP no domiciliados en Ajustes Posteriores RCE de Periodos Anteriores
96	Generar archivo exportar preliminar - RCE No Domiciliados
97	Exportar comprobantes excluidos

7.2 Anexo II: Tipo de correlativo

Código	Descripción
01	Tipo envíos masivos
02	Número operación de generación RVIE
03	Solicitud de generación de archivo
04	Tipo carga archivo comparación

7.3 Anexo III: Código de estado de envío

Código	Descripción
01	Cargado (solicitado)

02	Validando Archivo (en proceso)
03	Procesado con Errores
04	Procesado sin errores (concluido)
05	En proceso
06	Terminado

7.4 Anexo IV: Extension del archivo a descargar

Código	Descripción
0	txt
1	excel
2	csv

7.5 Anexo V: Número de casillas

Código	Descripción
100	Ventas netas gravadas (Base imponible)
101	Ventas netas gravadas (Tributo)
102	Descuentos concedidos y devolución de ventas (Base imponible)
103	Descuentos concedidos y devolución de ventas (Tributo)
160	Ventas Ley N° 27037 incisos 11.1, 12.1, 12.3 y 12.4 (Base imponible)
161	Ventas Ley N° 27037 incisos 11.1, 12.1, 12.3 y 12.4 (Tributo)
162	Descuentos y Devoluciones Ley N°27037 (Base imponible)
163	Descuentos y Devoluciones Ley N°27037 (Tributo)
106	Exportaciones facturadas en el periodo
127	Exportaciones embarcadas en el período
~~183~~	~~Exportaciones embarcadas de bienes~~
~~186~~	~~Exportaciones facturadas de servicios~~
105	Ventas no gravadas (sin considerar exportaciones)
109	Ventas no gravadas sin efecto en ratio
112	Otras ventas (inciso ii, numeral 6.2 del artículo 6° del Reglamento)
131	Total (Tributo)

7.6 Anexo VI: Ejemplo cliente TUS JAVA

Para poder hacer uso de las API-REST de carga de archivos **SIRE** es necesario tener configurado en su proyecto la librería "**tus-java-client"** en la versión "**0-5-0**", la cual se puede encontrar en el repositorio Maven **https://repo1.maven.org/maven2/io/tus/java/client/tus-java-client/0.5.0/tus-java-client-0.5.0.jar**
Esta librería tiene algunas deficiencias para la gestión de los mensajes de retorno y error.
Por lo cual se tienen que hacer algunas adecuaciones para que se muestren de forma correcta los errores proporcionados por el servicio SIRE.

1. Crear el paquete: **io.tus.java.client** y dentro crear las clases:
   - TusResponseBody.java: transforma los errores a texto que se puede imprimir en la bitácora.
   - Http401And403CodeException.java: Muestra los errores de autenticación o de autorización
   - Http422CodeException.java: Muestra los errores de validación de negocio
   - HttpErrorCodeException.java: Clase de ayuda para la gestión de errores
   - TusClientCustom.java: Cliente tus personalizado que permite ver la gestión de errores.
   - TusUploaderCustom.java: Uploader tus personalizado que permite ver la gestión de errores.

2. Crear la clase: Demo.java (puede usar el paquete que desee): se usará para realizar la operación de carga de archivos del SIRE.

Clase:: TusResponseBody.java

```
/**
 *
 * Copyright 2011-2024 the original author or authors.
 *  *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *  *
 *      https://www.apache.org/licenses/LICENSE-2.0
 *  *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 */
package io.tus.java.client;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.Serializable;
import java.net.HttpURLConnection;

public class TusResponseBody implements Serializable {

  private int responseCode;
  private String responseMessage;
  private String responseBody;

  public TusResponseBody(HttpURLConnection connection)
    throws IOException {
    if (connection != null) {
      responseCode = connection.getResponseCode();
      responseBody = readResponseBody(connection);
      responseMessage = connection.getResponseMessage();
    }
  }

  public int getResponseCode() {
    return responseCode;
  }
```

```java
  public String getResponseBody() {
    return responseBody;
  }

  public String getResponseMessage() {
    return responseMessage;
  }

  private String readResponseBody (
    HttpURLConnection connection
  ) throws IOException {

    ByteArrayOutputStream result = new ByteArrayOutputStream();
    byte[] buffer = new byte[1024];
    int length;
    InputStream inputStream;
    if (connection.getErrorStream() != null) {
      inputStream = connection.getErrorStream();
    } else {
      try {
        inputStream = connection.getInputStream();
      } catch (IOException e) {
        return e.getMessage();
      }
    }

    if (inputStream == null) {
      return "";
    }
    try {
      while ((length = inputStream.read(buffer)) != -1) {
        result.write(buffer, 0, length);
      }
      connection.disconnect();
    } catch (IOException e) {
      return e.getMessage();
    }
    return result.toString("UTF-8");


  }

  @Override
  public String toString() {
    return responseBody.isEmpty()? responseMessage: responseBody;
  }
}
```

Clase:: Http401And403CodeException.java

```java
/**
 *
 * Copyright 2011-2024 the original author or authors.
 *  *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *  *
 *      https://www.apache.org/licenses/LICENSE-2.0
 *  *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 */
package io.tus.java.client;

import java.net.HttpURLConnection;

public class Http401And403CodeException extends HttpErrorCodeException {

  public Http401And403CodeException(
    TusResponseBody response,
    HttpURLConnection connection
  ) {
    super(response, connection);
  }
}
```

Clase:: Http422CodeException.java

```java
/**
 *
 * Copyright 2011-2024 the original author or authors.
 *  *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *  *
 *      https://www.apache.org/licenses/LICENSE-2.0
 *  *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 */
package io.tus.java.client;

import java.io.IOException;
import java.net.HttpURLConnection;

public class Http422CodeException extends HttpErrorCodeException {
  public Http422CodeException(
    TusResponseBody response,
    HttpURLConnection connection
  ) throws IOException {
    super(response, connection);
  }
}
```

Clase:: HttpErrorCodeException.java

```java
/**
 *
 * Copyright 2011-2024 the original author or authors.
 *  *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *  *
 *      https://www.apache.org/licenses/LICENSE-2.0
 *  *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 */
package io.tus.java.client;

import java.net.HttpURLConnection;

public class HttpErrorCodeException extends ProtocolException {

  private static final long serialVersionUID = 0l;

  private final TusResponseBody response;

  public HttpErrorCodeException(
    TusResponseBody response,
    HttpURLConnection connection
  ) {
    super(response.getResponseMessage(), connection);
    this.response = response;
  }

  public TusResponseBody getResponseBody() {
    return response;
  }


}
```

Clase:: TusClientCustom.java

```
/**
 *
 * Copyright 2011-2024 the original author or authors.
 *  *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *  *
 *      https://www.apache.org/licenses/LICENSE-2.0
 *  *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 */
package io.tus.java.client;

import java.net.Proxy;
import org.jetbrains.annotations.NotNull;
import org.jetbrains.annotations.Nullable;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Map;

/**
 * This class is used for creating or resuming uploads.
 */
public class TusClientCustom {
    /**
     * Version of the tus protocol used by the client. The remote server
needs to support this
     * version, too.
     */
    public static final String TUS_VERSION = "1.0.0";

    private URL uploadCreationURL;
    private Proxy proxy;
    private boolean resumingEnabled;
    private boolean removeFingerprintOnSuccessEnabled;
    private TusURLStore urlStore;
    private Map<String, String> headers;
    private int connectTimeout = 5000;
```

```
    /**
     * Create a new tus client.
     */
    public TusClientCustom() {

    }

    /**
     * Set the URL used for creating new uploads. This is required if you
want to initiate new
     * uploads using {@link #createUpload} or {@link #resumeOrCreateUpload}
but is not used if you
     * only resume existing uploads.
     *
     * @param uploadCreationURL Absolute upload creation URL
     */
    public void setUploadCreationURL(URL uploadCreationURL) {
        this.uploadCreationURL = uploadCreationURL;
    }

    /**
     * Get the current upload creation URL.
     *
     * @return Current upload creation URL
     */
    public URL getUploadCreationURL() {
        return uploadCreationURL;
    }

    /**
     * Set the proxy that will be used for all requests.
     *
     * @param proxy Proxy to use
     */
    public void setProxy(Proxy proxy) {
        this.proxy = proxy;
    }

    /**
     * Get the current proxy used for all requests.
     *
     * @return Current proxy
     */
    public Proxy getProxy() {
        return proxy;
    }

    /**
```

```
     * Enable resuming already started uploads. This step is required if you
want to use
     * {@link #resumeUpload(TusUpload)}.
     *
     * @param urlStore Storage used to save and retrieve upload URLs by its
fingerprint.
     */
    public void enableResuming(@NotNull TusURLStore urlStore) {
        resumingEnabled = true;
        this.urlStore = urlStore;
    }

    /**
     * Disable resuming started uploads.
     *
     * @see #enableResuming(TusURLStore)
     */
    public void disableResuming() {
        resumingEnabled = false;
        this.urlStore = null;
    }

    /**
     * Get the current status if resuming.
     *
     * @see #enableResuming(TusURLStore)
     * @see #disableResuming()
     *
     * @return True if resuming has been enabled using {@link
#enableResuming(TusURLStore)}
     */
    public boolean resumingEnabled() {
        return resumingEnabled;
    }

    /**
     * Enable removing fingerprints after a successful upload.
     *
     * @see #disableRemoveFingerprintOnSuccess()
     */
    public void enableRemoveFingerprintOnSuccess() {
        removeFingerprintOnSuccessEnabled = true;
    }

    /**
     * Disable removing fingerprints after a successful upload.
     *
     * @see #enableRemoveFingerprintOnSuccess()
     */
```

```
    public void disableRemoveFingerprintOnSuccess() {
        removeFingerprintOnSuccessEnabled = false;
    }

    /**
     * Get the current status if removing fingerprints after a successful
upload.
     *
     * @see #enableRemoveFingerprintOnSuccess()
     * @see #disableRemoveFingerprintOnSuccess()
     *
     * @return True if resuming has been enabled using {@link
#enableResuming(TusURLStore)}
     */
    public boolean removeFingerprintOnSuccessEnabled() {
        return removeFingerprintOnSuccessEnabled;
    }


    /**
     * Set headers which will be added to every HTTP requestes made by this
TusClient instance.
     * These may to overwrite tus-specific headers, which can be identified
by their Tus-*
     * prefix, and can cause unexpected behavior.
     *
     * @see #getHeaders()
     * @see #prepareConnection(HttpURLConnection)
     *
     * @param headers The map of HTTP headers
     */
    public void setHeaders(@Nullable Map<String, String> headers) {
        this.headers = headers;
    }

    /**
     * Get the HTTP headers which should be contained in every request and
were configured using
     * {@link #setHeaders(Map)}.
     *
     * @see #setHeaders(Map)
     * @see #prepareConnection(HttpURLConnection)
     *
     * @return The map of configured HTTP headers
     */
    @Nullable
    public Map<String, String> getHeaders() {
        return headers;
    }
```

```
    /**
     * Sets the timeout for a Connection.
     * @param timeout in milliseconds
     */
    public void setConnectTimeout(int timeout) {
        connectTimeout = timeout;
    }

    /**
     * Returns the Connection Timeout.
     * @return Timeout in milliseconds.
     */
    public int getConnectTimeout() {
        return connectTimeout;
    }

    /**
     * Create a new upload using the Creation extension. Before calling this
function, an "upload
     * creation URL" must be defined using {@link #setUploadCreationURL(URL)}
or else this
     * function will fail.
     * In order to create the upload a POST request will be issued. The
file's chunks must be
     * uploaded manually using the returned {@link TusUploader} object.
     *
     * @param upload The file for which a new upload will be created
     * @return Use {@link TusUploader} to upload the file's chunks.
     * @throws ProtocolException Thrown if the remote server sent an
unexpected response, e.g.
     * wrong status codes or missing/invalid headers.
     * @throws IOException Thrown if an exception occurs while issuing the
HTTP request.
     */
    public TusUploaderCustom createUpload(@NotNull TusUpload upload) throws
ProtocolException, IOException {
        HttpURLConnection connection = openConnection(uploadCreationURL);
        connection.setRequestMethod("POST");
        prepareConnection(connection);

        String encodedMetadata = upload.getEncodedMetadata();
        if (encodedMetadata.length() > 0) {
            connection.setRequestProperty("Upload-Metadata",
encodedMetadata);
        }

        connection.addRequestProperty("Upload-Length",
Long.toString(upload.getSize()));
```

```
        connection.connect();

        int responseCode = connection.getResponseCode();
        if (responseCode == 401 || responseCode == 403) {
            throw new Http401And403CodeException(new
TusResponseBody(connection), connection);
        }

        if(responseCode == 422) {
            throw new Http422CodeException(new TusResponseBody(connection),
connection);
        }

        if(!(responseCode >= 200 && responseCode < 300)) {
            throw new ProtocolException("unexpected status code (" +
responseCode + ") while creating upload", connection);
        }

        String urlStr = connection.getHeaderField("Location");
        if (urlStr == null || urlStr.length() == 0) {
            throw new ProtocolException("missing upload URL in response for
creating upload", connection);
        }

        // The upload URL must be relative to the URL of the request by which
is was returned,
        // not the upload creation URL. In most cases, there is no difference
between those two
        // but there may be cases in which the POST request is redirected.
        URL uploadURL = new URL(connection.getURL(), urlStr);

        if (resumingEnabled) {
            urlStore.set(upload.getFingerprint(), uploadURL);
        }

        return createUploader(upload, uploadURL, 0L);
    }

    @NotNull
    private HttpURLConnection openConnection(@NotNull URL uploadURL) throws
IOException {
        if (proxy != null) {
            return (HttpURLConnection) uploadURL.openConnection(proxy);
        }
        return (HttpURLConnection) uploadURL.openConnection();
    }

    @NotNull
```

```
    private TusUploaderCustom createUploader(@NotNull TusUpload upload,
@NotNull URL uploadURL, long offset)
        throws IOException {
        TusUploaderCustom uploader = new TusUploaderCustom(this, upload,
uploadURL, upload.getTusInputStream(), offset);
        uploader.setProxy(proxy);
        return uploader;
    }

    /**
     * Try to resume an already started upload. Before call this function,
resuming must be
     * enabled using {@link #enableResuming(TusURLStore)}. This method will
look up the URL for this
     * upload in the {@link TusURLStore} using the upload's fingerprint (see
     * {@link TusUpload#getFingerprint()}). After a successful lookup a HEAD
request will be issued
     * to find the current offset without uploading the file, yet.
     *
     * @param upload The file for which an upload will be resumed
     * @return Use {@link TusUploader} to upload the remaining file's chunks.
     * @throws FingerprintNotFoundException Thrown if no matching fingerprint
has been found in
     * {@link TusURLStore}. Use {@link #createUpload(TusUpload)} to create a
new upload.
     * @throws ResumingNotEnabledException Throw if resuming has not been
enabled using {@link
     * #enableResuming(TusURLStore)}.
     * @throws ProtocolException Thrown if the remote server sent an
unexpected response, e.g.
     * wrong status codes or missing/invalid headers.
     * @throws IOException Thrown if an exception occurs while issuing the
HTTP request.
     */
    public TusUploaderCustom resumeUpload(@NotNull TusUpload upload) throws
            FingerprintNotFoundException, ResumingNotEnabledException,
ProtocolException, IOException {
        if (!resumingEnabled) {
            throw new ResumingNotEnabledException();
        }

        URL uploadURL = urlStore.get(upload.getFingerprint());
        if (uploadURL == null) {
            throw new FingerprintNotFoundException(upload.getFingerprint());
        }

        return beginOrResumeUploadFromURL(upload, uploadURL);
    }
```

```
    /**
     * Begin an upload or alternatively resume it if the upload has already
been started before. In contrast to
     * {@link #createUpload(TusUpload)} and {@link
#resumeOrCreateUpload(TusUpload)} this method will not create a new
     * upload. The user must obtain the upload location URL on their own as
this method will not send the POST request
     * which is normally used to create a new upload.
     * Therefore, this method is only useful if you are uploading to a
service which takes care of creating the tus
     * upload for yourself. One example of such a service is the Vimeo API.
     * When called a HEAD request will be issued to find the current offset
without uploading the file, yet.
     * The uploading can be started by using the returned {@link TusUploader}
object.
     *
     * @param upload The file for which an upload will be resumed
     * @param uploadURL The upload location URL at which has already been
created and this file should be uploaded to.
     * @return Use {@link TusUploader} to upload the remaining file's chunks.
     * @throws ProtocolException Thrown if the remote server sent an
unexpected response, e.g.
     * wrong status codes or missing/invalid headers.
     * @throws IOException Thrown if an exception occurs while issuing the
HTTP request.
     */
    public TusUploaderCustom beginOrResumeUploadFromURL(@NotNull TusUpload
upload, @NotNull URL uploadURL) throws
            ProtocolException, IOException {
        HttpURLConnection connection = openConnection(uploadURL);
        connection.setRequestMethod("HEAD");
        prepareConnection(connection);

        connection.connect();

        int responseCode = connection.getResponseCode();

        if (responseCode == 401 || responseCode == 403) {
            throw new Http401And403CodeException(new
TusResponseBody(connection), connection);
        }

        if(responseCode == 422) {
            throw new Http422CodeException(new TusResponseBody(connection),
connection);
        }

        if(!(responseCode >= 200 && responseCode < 300)) {
```

```
            throw new HttpErrorCodeException(new TusResponseBody(connection),
connection);
        }

        String offsetStr = connection.getHeaderField("Upload-Offset");
        if (offsetStr == null || offsetStr.length() == 0) {
            throw new ProtocolException("missing upload offset in response
for resuming upload", connection);
        }
        long offset = Long.parseLong(offsetStr);

        return createUploader(upload, uploadURL, offset);
    }

    /**
     * Try to resume an upload using {@link #resumeUpload(TusUpload)}. If the
method call throws
     * an {@link ResumingNotEnabledException} or {@link
FingerprintNotFoundException}, a new upload
     * will be created using {@link #createUpload(TusUpload)}.
     *
     * @param upload The file for which an upload will be resumed
     * @throws ProtocolException Thrown if the remote server sent an
unexpected response, e.g.
     * wrong status codes or missing/invalid headers.
     * @throws IOException Thrown if an exception occurs while issuing the
HTTP request.
     * @return {@link TusUploader} instance.
     */
    public TusUploaderCustom resumeOrCreateUpload(@NotNull TusUpload upload)
throws ProtocolException, IOException {
        try {
            return resumeUpload(upload);
        } catch (FingerprintNotFoundException e) {
            return createUpload(upload);
        } catch (ResumingNotEnabledException e) {
            return createUpload(upload);
        } catch (ProtocolException e) {
            // If the attempt to resume returned a 404 Not Found, we
immediately try to create a new
            // one since TusExectuor would not retry this operation.
            HttpURLConnection connection = e.getCausingConnection();
            if (connection != null && connection.getResponseCode() == 404) {
                return createUpload(upload);
            }

            throw e;
        }
    }
```

```
    /**
     * Set headers used for every HTTP request. Currently, this will add the
Tus-Resumable header
     * and any custom header which can be configured using {@link
#setHeaders(Map)},
     *
     * @param connection The connection whose headers will be modified.
     */
    public void prepareConnection(@NotNull HttpURLConnection connection) {
        // Only follow redirects, if the POST methods is preserved. If
http.strictPostRedirect is
        // disabled, a POST request will be transformed into a GET request
which is not wanted by us.

        // CHECKSTYLE:OFF
        // LineLength - Necessary because of length of the link
        // See:https://github.com/openjdk/jdk/blob/jdk7-
b43/jdk/src/share/classes/sun/net/www/protocol/http/HttpURLConnection.java#L2
020-L2035
        // CHECKSTYLE:ON
        connection.setInstanceFollowRedirects(Boolean.getBoolean("http.strict
PostRedirect"));

        connection.setConnectTimeout(connectTimeout);
        connection.addRequestProperty("Tus-Resumable", TUS_VERSION);

        if (headers != null) {
            for (Map.Entry<String, String> entry : headers.entrySet()) {
                connection.addRequestProperty(entry.getKey(),
entry.getValue());
            }
        }
    }

    /**
     * Actions to be performed after a successful upload completion.
     * Manages URL removal from the URL store if remove fingerprint on
success is enabled
     *
     * @param upload that has been finished
     */
    protected void uploadFinished(@NotNull TusUpload upload) {
        if (resumingEnabled && removeFingerprintOnSuccessEnabled) {
            urlStore.remove(upload.getFingerprint());
        }
    }
}
```

Clase: TusUploaderCustom.java

```
/**
 *
 * Copyright 2011-2024 the original author or authors.
 *  *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *  *
 *      https://www.apache.org/licenses/LICENSE-2.0
 *  *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 */
package io.tus.java.client;

import java.io.IOException;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.Proxy;
import java.net.URL;
import java.net.URLConnection;
import java.util.Optional;

/**
 * This class is used for doing the actual upload of the files. Instances are
returned by
 * {@link TusClientCustom#createUpload(TusUpload)}, {@link
TusClientCustom#createUpload(TusUpload)} and
 * {@link TusClientCustom#resumeOrCreateUpload(TusUpload)}.
 * <br>
 * After obtaining an instance you can upload a file by following these
steps:
 * <ol>
 *  <li>Upload a chunk using {@link #uploadChunk()}</li>
 *  <li>Optionally get the new offset ({@link #getOffset()} to calculate the
progress</li>
 *  <li>Repeat step 1 until the {@link #uploadChunk()} returns -1</li>
 *  <li>Close HTTP connection and InputStream using {@link #finish()} to free
resources</li>
 * </ol>
 */
public class TusUploaderCustom {
```

```
    private URL uploadURL;
    private Proxy proxy;
    private TusInputStream input;
    private long offset;
    private TusClientCustom client;
    private TusUpload upload;
    private byte[] buffer;
    private int requestPayloadSize = 10 * 1024 * 1024;
    private int bytesRemainingForRequest;

    private HttpURLConnection connection;
    private OutputStream output;

    /**
     * Begin a new upload request by opening a PATCH request to specified
upload URL. After this
     * method returns a connection will be ready and you can upload chunks of
the file.
     *
     * @param client Used for preparing a request ({@link
TusClient#prepareConnection(HttpURLConnection)}
     * @param upload {@link TusUpload} to be uploaded.
     * @param uploadURL URL to send the request to
     * @param input Stream to read (and seek) from and upload to the remote
server
     * @param offset Offset to read from
     * @throws IOException Thrown if an exception occurs while issuing the
HTTP request.
     */
    public TusUploaderCustom(TusClientCustom client, TusUpload upload, URL
uploadURL, TusInputStream input, long offset)
        throws IOException {
        this.uploadURL = uploadURL;
        this.input = input;
        this.offset = offset;
        this.client = client;
        this.upload = upload;

        input.seekTo(offset);

        setChunkSize(2 * 1024 * 1024);
    }

    private void openConnection() throws IOException, ProtocolException {
        // Only open a connection, if we have none open.
        if (connection != null) {
            return;
        }
```

```
        bytesRemainingForRequest = requestPayloadSize;
        input.mark(requestPayloadSize);

        if (proxy != null) {
            connection = (HttpURLConnection) uploadURL.openConnection(proxy);
        } else {
            connection = (HttpURLConnection) uploadURL.openConnection();
        }
        client.prepareConnection(connection);
        connection.setRequestProperty("Upload-Offset",
Long.toString(offset));
        connection.setRequestProperty("Content-Type",
"application/offset+octet-stream");
        connection.setRequestProperty("Expect", "100-continue");

        try {
            connection.setRequestMethod("PATCH");
            // Check whether we are running on a buggy JRE
        } catch (java.net.ProtocolException pe) {
            connection.setRequestMethod("POST");
            connection.setRequestProperty("X-HTTP-Method-Override", "PATCH");
        }

        connection.setDoOutput(true);
        connection.setChunkedStreamingMode(0);
        try {
            output = connection.getOutputStream();
        } catch (java.net.ProtocolException pe) {
            // If we already have a response code available, our expectation
using the "Expect: 100-
            // continue" header failed and we should handle this response.
            if (connection.getResponseCode() != -1) {
                finish();
            }

            throw pe;
        }
    }

    /**
     * Sets the used chunk size. This number is used by {@link
#uploadChunk()} to indicate how
     * much data is uploaded in a single take. When choosing a value for this
parameter you need to
     * consider that uploadChunk() will only return once the specified number
of bytes has been
     * sent. For slow internet connections this may take a long time. In
addition, a buffer with
     * the chunk size is allocated and kept in memory.
```

```
     *
     * @param size The new chunk size
     */
    public void setChunkSize(int size) {
        buffer = new byte[size];
    }

    /**
     * Returns the current chunk size set using {@link #setChunkSize(int)}.
     *
     * @return Current chunk size
     */
    public int getChunkSize() {
        return buffer.length;
    }

    /**
     * Set the maximum payload size for a single request counted in bytes.
This is useful for splitting
     * bigger uploads into multiple requests. For example, if you have a
resource of 2MB and
     * the payload size set to 1MB, the upload will be transferred by two
requests of 1MB each.
     *
     * The default value for this setting is 10 * 1024 * 1024 bytes (10 MiB).
     *
     * Be aware that setting a low maximum payload size (in the low megabytes
or even less range) will result in
     * decreased performance since more requests need to be used for an
upload. Each request will come with its overhead
     * in terms of longer upload times.
     *
     * Be aware that setting a high maximum payload size may result in a high
memory usage since
     * tus-java-client usually allocates a buffer with the maximum payload
size (this buffer is used
     * to allow retransmission of lost data if necessary). If the client is
running on a memory-
     * constrained device (e.g. mobile app) and the maximum payload size is
too high, it might
     * result in an {@link OutOfMemoryError}.
     *
     * This method must not be called when the uploader has currently an open
connection to the
     * remote server. In general, try to set the payload size before invoking
{@link #uploadChunk()}
     * the first time.
     *
     * @see #getRequestPayloadSize()
```

```
     *
     * @param size Number of bytes for a single payload
     * @throws IllegalStateException Thrown if the uploader currently has a
connection open
     */
    public void setRequestPayloadSize(int size) throws IllegalStateException
{
        if (connection != null) {
            throw new IllegalStateException("payload size for a single
request must not be "
                    + "modified as long as a request is in progress");
        }

        requestPayloadSize = size;
    }

    /**
     * Get the current maximum payload size for a single request.
     *
     * @see #setChunkSize(int)
     *
     * @return Number of bytes for a single payload
     */
    public int getRequestPayloadSize() {
        return requestPayloadSize;
    }

    /**
     * Upload a part of the file by reading a chunk from the InputStream and
writing
     * it to the HTTP request's body. If the number of available bytes is
lower than the chunk's
     * size, all available bytes will be uploaded and nothing more.
     * No new connection will be established when calling this method,
instead the connection opened
     * in the previous calls will be used.
     * The size of the read chunk can be obtained using {@link
#getChunkSize()} and changed
     * using {@link #setChunkSize(int)}.
     * In order to obtain the new offset, use {@link #getOffset()} after this
method returns.
     *
     * @return Number of bytes read and written.
     * @throws IOException  Thrown if an exception occurs while reading from
the source or writing
     *                      to the HTTP request.
     */
    public int uploadChunk() throws IOException, ProtocolException {
        openConnection();
```

```
        int bytesToRead = Math.min(getChunkSize(), bytesRemainingForRequest);

        int bytesRead = input.read(buffer, bytesToRead);
        if (bytesRead == -1) {
            // No bytes were read since the input stream is empty
            return -1;
        }

        // Do not write the entire buffer to the stream since the array will
        // be filled up with 0x00s if the number of read bytes is lower then
        // the chunk's size.
        output.write(buffer, 0, bytesRead);
        output.flush();

        offset += bytesRead;
        bytesRemainingForRequest -= bytesRead;

        if (bytesRemainingForRequest <= 0) {
            finishConnection();
        }

        return bytesRead;
    }

    /**
     * Upload a part of the file by read a chunks specified size from the
InputStream and writing
     * it to the HTTP request's body. If the number of available bytes is
lower than the chunk's
     * size, all available bytes will be uploaded and nothing more.
     * No new connection will be established when calling this method,
instead the connection opened
     * in the previous calls will be used.
     * In order to obtain the new offset, use {@link #getOffset()} after this
method returns.
     *
     * This method ignored the payload size per request, which may be set
using
     * {@link #setRequestPayloadSize(int)}. Please, use {@link
#uploadChunk()} instead.
     *
     * @deprecated This method is inefficient and has been replaced by {@link
#setChunkSize(int)}
     *             and {@link #uploadChunk()} and should not be used anymore.
The reason is, that
     *             this method allocates a new buffer with the supplied chunk
size for each time
```

```
     *             it's called without reusing it. This results in a high
number of memory
     *             allocations and should be avoided. The new methods do not
have this issue.
     *
     * @param chunkSize Maximum number of bytes which will be uploaded. When
choosing a value
     *                  for this parameter you need to consider that the
method call will only
     *                  return once the specified number of bytes have been
sent. For slow
     *                  internet connections this may take a long time.
     * @return Number of bytes read and written.
     * @throws IOException  Thrown if an exception occurs while reading from
the source or writing
     *                      to the HTTP request.
     */
    @Deprecated public int uploadChunk(int chunkSize) throws IOException,
ProtocolException {
        openConnection();

        byte[] buf = new byte[chunkSize];
        int bytesRead = input.read(buf, chunkSize);
        if (bytesRead == -1) {
            // No bytes were read since the input stream is empty
            return -1;
        }

        // Do not write the entire buffer to the stream since the array will
        // be filled up with 0x00s if the number of read bytes is lower then
        // the chunk's size.
        output.write(buf, 0, bytesRead);
        output.flush();

        offset += bytesRead;

        return bytesRead;
    }

    /**
     * Get the current offset for the upload. This is the number of all bytes
uploaded in total and
     * in all requests (not only this one). You can use it in conjunction
with
     * {@link TusUpload#getSize()} to calculate the progress.
     *
     * @return The upload's current offset.
     */
    public long getOffset() {
```

```
        return offset;
    }

    /**
     * This methods returns the destination {@link URL} of the upload.
     * @return The {@link URL} of the upload.
     */
    public URL getUploadURL() {
        return uploadURL;
    }

    /**
     * Set the proxy that will be used when uploading.
     *
     * @param proxy Proxy to use
     */
    public void setProxy(Proxy proxy) {
        this.proxy = proxy;
    }

    /**
     * This methods returns the proxy used when uploading.
     *
     * @return The {@link Proxy} used for the upload or null when not set.
     */
    public Proxy getProxy() {
        return proxy;
    }

    /**
     * Finish the request by closing the HTTP connection and the InputStream.
     * You can call this method even before the entire file has been
uploaded. Use this behavior to
     * enable pausing uploads.
     * This method is equivalent to calling {@code finish(false)}.
     *
     * @throws ProtocolException Thrown if the server sends an unexpected
status
     * code
     * @throws IOException  Thrown if an exception occurs while cleaning up.
     */
    public Optional<TusResponseBody> finish() throws ProtocolException,
IOException {
        return finish(true);
    }

    /**
     * Finish the request by closing the HTTP connection. You can choose
whether to close the InputStream or not.
```

```
     * You can call this method even before the entire file has been
uploaded. Use this behavior to
     * enable pausing uploads.
     * Be aware that it doesn't automatically release local resources if
{@code closeStream == false} and you do
     * not close the InputStream on your own. To be safe use {@link
TusUploader#finish()}.
     * @param closeInputStream Determines whether the InputStream is closed
with the HTTP connection. Not closing the
     *                         Input Stream may be useful for future upload a
future continuation of the upload.
     * @throws ProtocolException Thrown if the server sends an unexpected
status code
     * @throws IOException  Thrown if an exception occurs while cleaning up.
     */
    public Optional<TusResponseBody> finish(boolean closeInputStream) throws
ProtocolException, IOException {
      Optional<TusResponseBody> response = finishConnection();
        if (upload.getSize() == offset) {
            client.uploadFinished(upload);
        }

        // Close the TusInputStream after checking the response and closing
the connection to ensure
        // that we will not need to read from it again in the future.
        if (closeInputStream) {
            input.close();
        }
        return response;
    }

    private Optional<TusResponseBody> finishConnection() throws
ProtocolException, IOException {
        if (output != null) {
            output.close();
        }

        if (connection == null) {
            return Optional.empty();
        }

        int responseCode = connection.getResponseCode();
        TusResponseBody response = new TusResponseBody(connection);
        connection.disconnect();

        if (responseCode == 401 || responseCode == 403) {
            throw new Http401And403CodeException(response, connection);
        }
```

```
        if(responseCode == 422) {
             throw new Http422CodeException(response, connection);
        }

        if (!(responseCode >= 200 && responseCode < 300)) {
            throw new ProtocolException("unexpected status code (" +
responseCode + ") while uploading chunk",
                    connection);
        }

        // TODO detect changes and seek accordingly
        long serverOffset = getHeaderFieldLong(connection, "Upload-Offset");
        if (serverOffset == -1) {
            throw new ProtocolException("response to PATCH request contains
no or invalid Upload-Offset header",
                    connection);
        }
        if (offset != serverOffset) {
            throw new ProtocolException(
                   String.format("response contains different Upload-Offset
value (%d) than expected (%d)",
                       serverOffset,
                        offset),
                   connection);
        }

        connection = null;

        return Optional.of(response);
    }

    private long getHeaderFieldLong(URLConnection connection, String field) {
        String value = connection.getHeaderField(field);
        if (value == null) {
            return -1;
        }

        try {
            return Long.parseLong(value);
        } catch (NumberFormatException e) {
            return -1;
        }
    }
}
```

Clase:: Demo.java

```
package pe.gob.sunat.tecnologia.client.tus.main;

import java.io.File;
import java.io.IOException;
import java.net.URL;
import java.util.HashMap;
import java.util.Map;
import java.util.Optional;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import io.tus.java.client.HttpErrorCodeException;
import io.tus.java.client.ProtocolException;
import io.tus.java.client.TusClientCustom;
import io.tus.java.client.TusExecutor;
import io.tus.java.client.TusResponseBody;
import io.tus.java.client.TusURLMemoryStore;
import io.tus.java.client.TusUpload;
import io.tus.java.client.TusUploaderCustom;


public class Demo {

  private static final String TOKEN = "";

  private static final String HOST_PUBLICA = "https://api-
sire.sunat.gob.pe/v1/contribuyente/migeigv/";
  private static final String END_POINT_PROPUESTA =
"libros/rvierce/receptorpropuesta/web/propuesta/upload";

  private static final Logger log = LoggerFactory.getLogger(Demo.class);


  public static void main(String[] args) {
    try {
      // Create a new TusClient instance
      final TusClientCustom client = new TusClientCustom();


      client.setUploadCreationURL(new URL(HOST_PUBLICA +
END_POINT_PROPUESTA));
      client.enableResuming(new TusURLMemoryStore());
      Map<String, String> headers = new HashMap<>();
      headers.put("authorization", "Bearer " + TOKEN);
      client.setHeaders(headers);
      File archivo = new File(
```

```
        "D:\\migeigv\\archivos\\ajuste_posterior\\LE20100017491202311001404000
03111201.zip"
      );
      final TusUpload upload = new TusUpload(archivo);
      String[] extension = archivo.getName().split("\\.");
      Map<String, String> metaData = new HashMap<>();
      metaData.put("filename", archivo.getName());
      metaData.put("filetype", extension[1]);
      metaData.put("numRuc", "20108745216");
      metaData.put("perTributario", "202304");
      metaData.put("codOrigenEnvio", "3");
      metaData.put("codLibro", "140000");
      metaData.put("codProceso", "1");
      metaData.put("codTipoCorrelativo", "1");
      metaData.put("nomArchivoImportacion", archivo.getName());
      upload.setMetadata(metaData);
      log.info("Starting upload...");
      TusExecutor executor = new TusExecutor() {
        @Override
        protected void makeAttempt() throws ProtocolException, IOException {
          TusUploaderCustom uploader = client.resumeOrCreateUpload(upload);
          uploader.setChunkSize(1024);
          do {
            long totalBytes = upload.getSize();
            long bytesUploaded = uploader.getOffset();
            double progress = (double) bytesUploaded / totalBytes * 100;
            String porcentaje = String.format("%06.2f%%", progress);
            log.info("Upload at {}.", porcentaje);
          } while (uploader.uploadChunk() > -1);


          Optional<TusResponseBody> response = uploader.finish();

          response.ifPresent(resp -> {
            log.info(
              "************** ==============> code: {} message: {}
<================================ *****************",
              resp.getResponseCode(),
              resp.getResponseMessage()
            );
            log.info(
              "************** ==============> Result {} {}
<================================ *****************",
              "20108745216",
              resp.getResponseBody()
            );
          });
        }
      };
```

```
      executor.makeAttempts();
    } catch (HttpErrorCodeException e) {
        log.info(
              "************** ==============> code: {} message: {}
<================================ ****************",
              e.getResponseBody().getResponseCode(),
              e.getResponseBody().getResponseMessage()
            );
      log.error("Error en la peticion: {}", e.getResponseBody(), e);
    } catch (ProtocolException e) {
      log.error("/////// //// ==> Error inesperado en la petición: ", e);
    } catch (IOException e) {
      e.printStackTrace();
    } catch (Exception ex) {

        log.error("ex.getCause(): {}", ex.getMessage(), ex);

    }
  }
}
```