Cómo funciona la API de LibreDTE

La forma de las URL, del cuerpo y de las respuestas de los recursos /api/v2/libredte/, la regla del base64 y los errores.

Anonymous · 05/11/2026

Cómo funciona la API de LibreDTE

La emisión de documentos en API Gateway V2 se hace con los recursos que empiezan con `/api/v2/libredte/`. Todos siguen las mismas reglas, así que si entiendes esta lección vas a poder usar cualquiera de ellos.

La URL

Cada recurso es una **operación** de la biblioteca de LibreDTE, identificada por cuatro partes:

```
POST
https://app.apigateway.cl/api/v2/libredte/{paquete}/{componente}/{worker}/{operación}
```

Por ejemplo, la operación que construye un documento es:

```
POST https://app.apigateway.cl/api/v2/libredte/billing/document/builder/build
                                              ──┬──── ──┬──── ──┬──── ──┬──
                                            paquete componente worker operación
```

Parte	Significado	Valores que usarás en este curso
`paquete`	Área de la biblioteca	`billing` (facturación), `system` (certificado)
`componente`	Tema dentro del área	`document`, `identifier` (folios), `integration` (SII), …
`worker`	Quién hace el trabajo	`builder`, `validator`, `dispatcher`, `renderer`, `sii_dte`, …
`operación`	Qué se hace	`build`, `validateSchema`, `create`, `render`, …

La operación se escribe en camelCase, igual que el método de la biblioteca: `validateSchema`, `sendXmlDocument`. El paquete, el componente y el worker se escriben tal como los nombra la biblioteca, por ejemplo `sii_dte`.

Todas las operaciones usan el método **POST**.

El cuerpo de la solicitud

El cuerpo siempre es un objeto JSON con un único atributo, `parameters`, que contiene los parámetros de la operación con el nombre que declara la documentación:

```
{
    "parameters": {
        "nombre_del_parametro": "valor"
    }
}
```

Por ejemplo, la operación que construye un documento recibe un único parámetro llamado `bag` (la "bolsa" con los datos del documento):

```
{
    "parameters": {
        "bag": {
            "parsedData": { "...": "datos del documento" },
            "caf": "...",
            "certificate": { "...": "certificado digital" }
        }
    }
}
```

Las cabeceras

```
Authorization: Token <tu_token>
Content-Type: application/json
```

El token es el de tu conexión de API Gateway. A diferencia de las consultas al SII de otros cursos, aquí **no** hay bloque `auth` con RUT y clave: lo que se necesita para hablar con el SII (el certificado digital) va dentro de los parámetros de cada operación.

La regla del base64

Los datos que no son JSON viajan codificados en **base64**: los XML (del DTE, del sobre, del CAF), los archivos `.p12` y los PDF. Esto evita problemas de caracteres especiales, porque los XML del SII usan la codificación ISO-8859-1 y no UTF-8.

Dato	Cómo se envía	Cómo vuelve
Datos del documento	JSON	JSON
XML de un DTE, de un sobre o de un CAF	base64	base64

Dato	Cómo se envía	Cómo vuelve
Certificado `.p12`/`.pfx`	base64 (con su contraseña aparte)	(no vuelve)
Certificado en formato PEM	texto PEM, **sin** base64	(no vuelve)
PDF	(no se envía)	base64

Ejemplos para codificar y decodificar en la terminal:

```
# Codificar un archivo en base64, en una sola línea
base64 -i caf.xml | tr -d '\n'

# Decodificar el base64 de una respuesta a un archivo
echo "PD94bWwgdmVyc2lvbj0i..." | base64 -d > dte.xml
```

La respuesta

Si la operación funciona, la respuesta es un JSON con el resultado dentro de `data`:

```
{
    "data": {
        "...": "resultado de la operación"
    }
}
```

Algunas operaciones no entregan nada (por ejemplo, una validación que no encontró problemas). En ese caso la respuesta es `{"data": null}` con código `200`.

Los errores

Si algo falla, la respuesta trae el código HTTP y un mensaje en `detail`:

```
{
    "code": 422,
    "detail": "El folio 500 del documento 88888888-8_T033F000000500 no está disponible
en el rango del CAF CAF33D1H100."
}
```

Código	Qué significa	Qué hacer
`400`	Falta un parámetro o no tiene el tipo esperado.	Revisa los nombres dentro de `parameters`.
`401`	Token inválido o ausente.	Revisa la cabecera `Authorization`.

Código	Qué significa	Qué hacer
`402`	Problema de créditos o de pago de la conexión.	Revisa tu conexión (ver Manejo de errores).
`404`	La operación no existe.	Revisa la URL: las cuatro partes deben coincidir con la documentación.
`422`	La operación se ejecutó, pero **los datos fueron rechazados**: un RUT inválido, un folio fuera del CAF, un XML mal formado, una firma inválida, etc.	Lee `detail`: explica exactamente qué dato corregir.
`500`	Error inesperado.	Reintenta y, si persiste, abre un ticket con la hora y la operación.
`503`	El servicio de LibreDTE no está disponible en ese momento.	Reintenta más tarde.

El 422 es tu amigo

Casi todos los errores de una integración nueva son `422`, y el mensaje de `detail` dice qué está mal. Muéstralo en tus logs completo: te ahorrará mucho tiempo.

Dónde ver cada operación

La documentación de la API tiene una ficha por operación, agrupadas como **API LibreDTE - billing.document**, **API LibreDTE - billing.identifier**, etc. Cada ficha indica:

- Los parámetros y un ejemplo del cuerpo.
- Qué entrega y por qué puede rechazar los datos.
- El costo en créditos.

Última actualización el 05/11/2026