Paso 2: Validar antes de enviar (opcional)

Cómo revisar el esquema, los datos y la firma de un DTE antes de enviarlo al SII.

Anonymous · 05/11/2026

Paso 2: Validar antes de enviar (opcional)

Este paso es opcional, pero muy recomendable mientras desarrollas: detecta en segundos los errores que el SII te diría recién al procesar el envío. Hay tres validaciones, y las tres reciben el mismo dato: el XML que construiste en el paso 1.

Las validaciones son **las mismas para todos los documentos, incluidas las boletas**: la API reconoce el tipo de documento y, si es una boleta, la revisa con el esquema de boletas del SII.

Operación	Qué revisa
`billing/document/validator/validateSchema`	Que el XML cumpla el **esquema oficial del SII** (campos, tipos de datos, largos, orden).
`billing/document/validator/validate`	Que los **datos** del documento sean coherentes (por ejemplo, que una nota tenga referencia).
`billing/document/validator/validateSignature`	Que la **firma** del XML sea válida.

Qué dato usar del paso 1

De la respuesta de `builder/build` solo necesitas el campo **`document_xml`**:

```
{
    "data": {
        "document_id": "88888888-8_T033F000000001",
        "document_xml":
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iSVNPLTg4NTktMSI/Pgo8RFRF...",
        "...": "los demás campos de la respuesta"
    }
}
```

Copia ese valor **tal cual**, en base64 y sin decodificarlo, en el parámetro `source` de cada validación:

```
{
    "parameters": {
        "source": "<el document_xml del paso 1>"
```

```
    }
}
```

Valida el mismo XML que vas a enviar

No vuelvas a construir el documento para validarlo: cada construcción genera un XML con otra firma. Valida el `document_xml` que guardaste, que es el mismo que usarás en el paso 3.

Validar el esquema

```
curl -X POST
https://app.apigateway.cl/api/v2/libredte/billing/document/validator/validateSchema \
  -H "Authorization: Token $API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"parameters": {"source":
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iSVNPLTg4NTktMSI/Pgo8RFRF..."}}'
```

Si el XML es válido, la respuesta es `200` con el documento leído como JSON dentro de `data`. Si no lo es, es un `422` con el motivo en `detail`.

Validar los datos

```
curl -X POST
https://app.apigateway.cl/api/v2/libredte/billing/document/validator/validate \
  -H "Authorization: Token $API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"parameters": {"source":
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iSVNPLTg4NTktMSI/Pgo8RFRF..."}}'
```

Si no hay problemas, la respuesta es `{"data": null}` con código `200`.

Validar la firma

```
curl -X POST
https://app.apigateway.cl/api/v2/libredte/billing/document/validator/validateSignature
\
  -H "Authorization: Token $API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"parameters": {"source":
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iSVNPLTg4NTktMSI/Pgo8RFRF..."}}'
```

La respuesta siempre es `200`, y el resultado está en `data.valid`. Si la firma es válida, `valid` es `true` y `signature` trae los datos de la firma, incluido quién firmó (`signer`):

```
{
    "data": {
        "valid": true,
        "error": null,
        "signature": {
            "reference": "LibreDTE_88888888-8_T033F000000000001",
            "digestValue": "YMnSHe2YViJQh/GdUuLB66wOM3M=",
            "signer": {
                "id": "11222333-9",
                "name": "Daniel Bot",
                "to": "2027-10-29T11:38:23",
                "isActive": true,
                "...": "..."
            },
            "...": "..."
        }
    }
}
```

Si no es válida, `valid` es `false` y el motivo está en `error.message`. Por ejemplo, si el XML se modificó después de firmarlo:

```
{
    "data": {
        "valid": false,
        "error": {
            "message": "The DigestValue that comes in the XML
\"76PnCAaCZb2XiWjKywMj4K3Qvko=\" for the reference
\"LibreDTE_88888888-8_T033F000000000001\" does not match the calculated value when
validating \"KRgsXYrcUs+tmoE7Tg9UiOgWiSQ=\". The data of the reference could have been
manipulated after being signed.",
            "code": 0,
            "errors": []
        },
        "signature": { "...": "..." }
    }
}
```

Revisa siempre `valid`: el código `200` solo indica que la validación se hizo, no que la firma sea válida.

Para no romper la firma, guarda y envía el `document_xml` tal como lo entrega la API.

¿Cuándo validar?

Etapa	Recomendación
Desarrollo y pruebas	Valida el esquema de **todos** los documentos.

Etapa	Recomendación
Producción	Valida al menos el esquema. Cuesta pocos créditos y te evita rechazos del SII.

El sobre también se valida

En el siguiente paso armarás el sobre. Para el sobre existen las mismas validaciones: `billing/document/dispatcher/validate`, `billing/document/dispatcher/validateSchema` y `billing/document/dispatcher/validateSignature`. Esta última responde una lista, con un resultado por cada firma del sobre: la de cada documento y la del sobre.

La validación no reemplaza al SII

Estas validaciones revisan el formato y los datos. Hay reglas que solo el SII puede verificar, como que el folio esté autorizado o que el certificado esté autorizado para la empresa. La respuesta final siempre la da el SII en el paso 5.

Última actualización el 05/11/2026