

Academia

Introducción

Anonymous · 11/09/2026

Tabla de contenidos

Introducción	3
Generalidades	4
Autenticación	6
Realizando peticiones	8
Errores	11
Límites de uso	13
Clientes de la API	15

Academia » Integración con la API » Introducción

Introducción

Anonymous · 11/09/2026

Introducción

Última actualización el 11/09/2026

Academia » Integración con la API » Introducción » Generalidades

Generalidades

Generalidades

Anonymous · 11/09/2026

Generalidades

LibreDTE provee una API (interfaz de programación de aplicaciones) que permite a los usuarios interactuar con algunas funcionalidades del sistema directamente desde sus propios programas. Esto permite automatizar procesos como la emisión de documentos tributarios electrónicos (DTE), ver estados, cargar datos de manera masiva, entre otros.

□ Puedes revisar la documentación oficial completa en [este enlace](#).

¿Qué se puede hacer con la API?

Puedes automatizar tareas como:

- Emitir boletas y facturas electrónicas (y todos los DTE).
- Obtener resúmenes de ventas y compras.
- Acceder a datos de clientes, productos, usuarios, etc.
- Consultar intercambios de DTE con proveedores o clientes.
- Recibir notificaciones de pagos.

Entre otras funciones que enontrarás en la [documentación de la API](#).

¿Qué NO se puede hacer con la API?

⚠ La API **no expone todas las funcionalidades** de la plataforma web. Está pensada para integraciones y automatización de lo más usado.

No puedes, por ejemplo:

- Administrar usuarios o permisos desde la API.
- Cambiar la configuración de la empresa (certificados, folios, etc.).
- Modificar plantillas de documentos.

Importante

Se espera que la API se use en conjunto con la plataforma web de LibreDTE.

¿Qué hago si necesito ayuda?

1. Revisa en detalle este curso.
2. Revisa la [especificación técnica de la API](#).
3. Revisa la [documentación adicional de integración con la API](#).
4. Si no encuentras la respuesta que buscas, puedes abrir un [ticket de soporte](#).

Última actualización el 11/09/2026

Academia » Integración con la API » Introducción » Autenticación

Autenticación

Autenticación

Anonymous · 11/09/2026

Autenticación

Para usar la API de LibreDTE necesitas autenticarte con un **hash personal** que obtienes desde tu perfil de usuario. Este hash actúa como contraseña y te identifica en todas tus solicitudes.

¿Dónde obtengo el hash?

Debes ingresar a tu perfil en la plataforma LibreDTE:

🔗 <https://libredte.cl/usuarios/perfil#datos:hashField>

Ahí verás un campo que contiene tu **hash de autenticación**.

API hash

.....



Identificador único del usuario en la API. Para generar uno nuevo borrar este y guardar.

Modo de uso enviando la cabecera de autenticación a la API: **Authorization: Basic BASE64(X:APIHASH)**

¿Cómo se usa el hash?

Se utiliza en la cabecera HTTP `Authorization` con el método de autenticación **Basic Auth**.

- **Usuario:** X (una letra equis mayúscula)
 - **Contraseña:** tu hash de autenticación
-

Ejemplo práctico

Si tu hash es `mihash123`, debes codificar lo siguiente en base64:

```
X:mihash123
```

El resultado es:

```
WDptaWhhc2gxMjM=
```

Y esa cadena la incluyes en la cabecera HTTP de esta forma:

```
Authorization: Basic WDptaWhhc2gxMjM=
```

En PHP (u otros lenguajes)

Ejemplo en PHP para generar la cabecera:

```
$hash = 'mihash123';  
$auth = base64_encode('X:' . $hash);  
$headers = ['Authorization: Basic ' . $auth];
```

Usando el API Key directamente

En tu perfil también encontrarás un campo llamado **API Key**. Este valor ya viene codificado en base64, listo para ser usado directamente.

Solo debes hacer esto:

```
Authorization: Basic APIKEY
```

Donde APIKEY es el valor copiado desde tu perfil.

API key

.....



Modo de uso enviando la cabecera de autenticación a la API: **Authorization: Basic APIKEY**

Última actualización el 11/09/2026

Academia » Integración con la API » Introducción » Realizando peticiones

Realizando peticiones

Realizando peticiones

Anonymous · 11/09/2026

Realizando peticiones

La API de LibreDTE usa 2 verbos HTTP:

- GET para, principalmente, consultar datos.
- POST para enviar datos (crear o ejecutar acciones).

No uses otros verbos HTTP

No uses PUT, PATCH ni DELETE. No están soportados oficialmente.

¿Dónde se envían los parámetros?

Hay **3 ubicaciones posibles** para los parámetros:

1. En el PATH

Se usan para identificar directamente el recurso.

Ejemplo:

```
GET /dte/dte_emitidos/info/33/1234/12345678
```

Significa: buscar DTEs emitidos por el RUT 12345678, tipo 33, con folio 1234, en ambiente configurado.

2. En la URL (query string)

Permiten modificar el comportamiento, formato de salida, filtros, etc.

Ejemplo:

```
GET /dte/dte_emitidos/info/33/1234/12345678?contribuyente_certificacion=1
```

Significa: buscar DTEs emitidos por el RUT 12345678, tipo 33, con folio 1234, en ambiente de certificación.

3. En el cuerpo (POST)

Datos que se envían con una solicitud POST deben ir como JSON.

Ejemplo:

```
POST /dte/dte_emitidos/buscar/12345678
```

```
{
  "cedido": null,
  "dte": 33,
  "fecha": null,
  "fecha_desde": null,
  "fecha_hasta": null,
  "folio": 1234,
  "periodo": null,
  "razon_social": null,
  "receptor": null,
  "receptor_evento": null,
  "sucursal_sii": null,
  "total": null,
  "total_desde": null,
  "total_hasta": null,
  "usuario": null,
  "xml": {
    "Detalle/NmbItem": "abono"
  }
}
```

Significa: buscar DTEs emitidos por el RUT 12345678, tipo 33, con folio 1234 y que el detalle del DTE contenga “abono”.

Cabeceras HTTP necesarias

Todas las solicitudes deben incluir:

```
Content-Type: application/json
Accept: application/json
```

Las únicas excepciones son cuando se solicita un formato de salida diferente a JSON, por ejemplo un PDF.

Parámetros especiales

`_contribuyente_rut`

Permite actuar en nombre del usuario administrador si estás usando el hash de un usuario con permisos delegados.

Ejemplo:

```
?_contribuyente_rut=12345678-9
```

Si usas el hash del usuario administrador, **no necesitas** este parámetro.

`_contribuyente_certificacion`

Permite elegir entre ambiente de **producción** y **certificación (pruebas)**.

- Omitido → Usa el valor por defecto configurado en la plataforma.
- `0` → Producción.
- `1` → Certificación.

Ejemplo:

```
?_contribuyente_certificacion=1
```

Ejemplo completo de solicitud

GET

```
/dte/dte_emitidos/info/33/1234/12345678?_contribuyente_rut=12345678-9&_contribuyente_c  
ertificacion=1
```

Última actualización el 11/09/2026

Errores

Errores

Anonymous · 11/09/2026

Errores

Cuando ocurre un problema al usar la API de LibreDTE, se devuelve:

- Un **código de estado HTTP**.
- Un **mensaje legible en formato JSON** con el detalle del error.

Ejemplo de error típico

Cabecera HTTP:

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json
```

Cuerpo JSON:

```
{
  "error": "Hash de autenticación incorrecto."
}
```

Códigos de error frecuentes

Código	Descripción HTTP	Significado en LibreDTE
400	Bad Request	Petición inválida (faltan campos, JSON mal formado)
401	Unauthorized	Hash incorrecto o faltante
403	Forbidden	No tienes permiso para ese recurso
404	Not Found	Recurso no encontrado
405	Method Not Allowed	Método no soportado (ej. PUT)
406	Not Acceptable	Formato de respuesta inválido

Código	Descripción HTTP	Significado en LibreDTE
410	Gone	El recurso ya no existe
429	Too Many Requests	Límite de uso excedido
500	Internal Server Error	Error inesperado del servidor
503	Service Unavailable	Servicio en mantención

Error clásico: No autorizado

Este error ocurre cuando el hash de autenticación no es válido o no tienes permisos para operar con la empresa solicitada.

```
{
  "error": "No está autorizado a operar con la empresa solicitada."
}
```

Posibles causas:

1. El hash corresponde a un usuario que **no tiene permisos** en la empresa.
2. Estás usando un usuario con permisos delegados y **no incluiste** `_contribuyente_rut`.

Debes revisar que el hash sea correcto y que el usuario tenga permisos para operar con la empresa solicitada mediante la plataforma web de LibreDTE.

□ Solución recomendada: usar el hash del administrador o incluir `_contribuyente_rut` en la URL.

Consejo

Para facilitar el diagnóstico, tu aplicación debería mostrar el mensaje devuelto por la API.

Última actualización el 11/09/2026

Academia » Integración con la API » Introducción » Límites de uso

Límites de uso

Límites de uso

Anonymous · 11/09/2026

Límites de uso

LibreDTE impone límites de uso a su API para proteger la infraestructura y garantizar un rendimiento constante.

¿Qué pasa si excedes el límite?

Recibirás una respuesta HTTP 429 con un cuerpo JSON que indica que has excedido el límite de solicitudes.

Cabecera HTTP:

```
HTTP/1.1 429 Too Many Requests
Content-Type: application/json
```

Cuerpo JSON:

```
{
  "error": "Too Many Attempts"
}
```

Y aparecerán cabeceras adicionales en la respuesta HTTP:

Cabecera	Descripción
X-RateLimit-Limit	Máximo de solicitudes permitidas en el período
X-RateLimit-Remaining	Cuántas solicitudes aún puedes hacer
Retry-After	Segundos que debes esperar para volver a intentarlo
X-RateLimit-Reset	Timestamp (Unix) cuando se reinicia el contador de peticiones

¿Cuánto puedo usar?

En general es suficiente para un uso normal y controlado de tu integración.

Recomendaciones

- No hagas *polling* innecesario.
- Usa caché en tu sistema.
- Maneja errores 429 con lógica de reintento exponencial.

Consejo

Si automatizas llamadas, asegúrate de distribuirlas en el tiempo.

Última actualización el 11/09/2026

Clientes de la API

Clientes de la API

Anonymous · 11/09/2026

Clientes de la API

LibreDTE ofrece **clientes oficiales** y la comunidad ha creado otros **no oficiales** para facilitar el consumo de la API desde distintos lenguajes de programación.

¿Debo usar un cliente?

No. Puedes consumir la API directamente con HTTP, pero un cliente te puede ahorrar trabajo y errores.

Tipos de clientes

Tipo	Soporte	Descripción
Oficial con soporte	☑	Mantenidos por LibreDTE y con soporte en caso de problemas.
Oficial sin soporte	☐	Creados por LibreDTE, pero sin mantenimiento ni soporte.
No oficiales	☐	Hechos por usuarios, útiles como referencia o punto de partida.

Clientes disponibles

Lenguaje	Autor	Soporte	Repositorio
PHP	LibreDTE	☑	libredte-api-client-php
Python	LibreDTE	☑	libredte-api-client-python
Java	LibreDTE	☑	libredte-sdk-java
Perl	LibreDTE	☑	libredte-sdk-perl
C	LibreDTE	☑	libredte-sdk-c

Lenguaje	Autor	Soporte	Repositorio
C++	LibreDTE	□	libredte-sdk-cpp
curl	LibreDTE	□	libredte-sdk-curl
Ruby	@crilam	□	libredte-sdk-ruby
C#	@petermajewski	□	libredte-sdk-c_sharp

¿Y si no hay cliente para mi lenguaje?

No hay problema. Puedes crear uno propio. Solo necesitas:

- Consumir `GET` y `POST` con JSON.
- Autenticación HTTP Basic.

Creación de un nuevo cliente de la API

Si estás interesado en crear o mantener un cliente de la API, puedes hacerlo. Puedes colaborar con LibreDTE para crear uno nuevo, o solicitar acceso a mantener uno existente. Solo necesitas cumplir con algunas condiciones básicas.

¿Qué necesitas?

- Tener conocimientos básicos de HTTP, JSON y autenticación HTTP Basic.
- Conocer el lenguaje en el que quieras crear el cliente.
- Seguir la interfaz estándar que usamos en nuestros clientes oficiales.

Requisitos

1. **Licencia libre:** el código debe estar bajo licencia [LGPL](#).
2. **Interfaz compatible:** debe seguir la interfaz de nuestros clientes oficiales.
3. **Casos de prueba:** al menos los de [facturación](#) deben estar implementados.

Estructura mínima esperada

Orientado a objetos (ejemplo en PHP)

```
class LibreDTE
{
```

```
public function __construct($hash, $url = 'https://libredte.cl')
{
    // Inicializa la conexión.
}

public function post($resource, $data = null)
{
    // Realiza POST a /recurso con datos JSON.
}

public function get($resource)
{
    // Realiza GET a /recurso.
}
}
```

Versión funcional (ejemplo en PHP)

```
function libredte_init($hash, $url = 'https://libredte.cl')
{
    // Retorna estructura para conexión.
}

function libredte_post($libredte, $resource, $data = null)
{
    // Realiza POST.
}

function libredte_get($libredte, $resource)
{
    // Realiza GET.
}
```

¡Con esto has terminado el módulo de Introducción! ☐

Última actualización el 11/09/2026