Academia

Punto de partida

Anonymous · 11/09/2026

Tabla de contenidos

Punto de partida

Anonymous · 11/09/2026

Punto de partida

Última actualización el 11/09/2026

Configuración de Proxy

Aprende a configurar un proxy

Anonymous · 11/09/2026

Configuración de Proxy

Un **proxy** es un servidor intermedio entre tu sistema e Internet: el tráfico hacia la plataforma que se consulta sale con la IP del proxy, no la de tu oficina o hosting. Esto es útil porque la plataforma a veces aplica **límites o restricciones** por volumen o por IP; con tu propio proxy reduces la dependencia del tráfico compartido con otros usuarios y tus operaciones pueden mantenerse más estables ante picos o bloqueos generales.

¿Dónde se configura?

En **app.apigateway.cl**, abre **Gestionar conexión** y ve a la pestaña **Conexión**. Ahí registras los datos del proxy. Si tienes varias conexiones, cada una puede tener su propia configuración. Enlace directo: Gestionar conexión

Para montar el proxy en sí, en el tutorial técnico de proxy encontrarás paso a paso cómo instalar **Squid** con Docker, asegurarlo, vincularlo a tu cuenta y probar que las consultas salen por él. Con manejo básico de Docker suele alcanzar.

Última actualización el 11/09/2026

Navegando Swagger UI

Aprende a usar la documentación interactiva (OpenAPI) para explorar y probar la API de API Gateway

Anonymous · 11/09/2026

Navegando Swagger UI

La documentación interactiva (a menudo **Swagger UI**) te permite **ver** las operaciones, **leer** parámetros y respuestas esperadas, y **probar** llamadas reales desde el navegador sin programar al inicio. Si nunca usaste una API, piensa en un **menú de servicios**: cada operación es una **receta** con método (GET, POST…), ruta, datos de entrada y respuesta.

Accediendo a la documentación interactiva

Abre la documentación en apigateway.cl/docs/api, o desde www.apigateway.cl en el menú **Documentación → API (V2)**.

Al ingresar verás el **título** del servicio y su descripción, la **versión** del esquema OpenAPI, el **servidor / URL base** donde se ejecutan las peticiones (habitualmente `https://app.apigateway.cl/api/v2/`) y el listado de **operaciones** agrupadas por categorías.

Estructura de la interfaz

Los endpoints están organizados por categorías:

```
 SII - Consultas
   └── GET /api/v2/sii/contribuyentes/{rut}
   └── POST /api/v2/sii/situacion-tributaria
   └── POST /api/v2/sii/actividades

 LibreDTE - Emisión
   └── POST /api/v2/libredte/dte/documentos/generar
   └── POST /api/v2/libredte/dte/envios/enviar

 Utilidades
   └── GET /api/v2/utilidades/validar-rut
   └── POST /api/v2/utilidades/convertir-moneda
```

Colores y métodos HTTP

Cada método HTTP suele tener un color distintivo en la interfaz:

Método	Color (típico)	Uso sencillo
GET	□ Azul	Pedir información (muchas veces sin cuerpo JSON)
POST	□ Verde	Enviar datos en el cuerpo (por ejemplo credenciales de la plataforma)
PUT	□ Amarillo	Actualizar un recurso completo
DELETE	□ Rojo	Eliminar o anular según el endpoint

Funcionalidad “Try it out”

Importante

Las peticiones que ejecutes desde **Try it out** son **reales**: van a **API Gateway** y, si el endpoint lo requiere, usará la plataforma relacionada con la consulta con los datos que ingreses. Usa credenciales y RUTs con cuidado.

1. Haz clic sobre un endpoint para expandirlo
2. Pulsa **Try it out** (habitualmente arriba a la derecha del bloque)
3. Los campos pasan a ser editables

Completar los datos

Para **GET** con parámetros en la ruta o query, se completan campos como `rut: 76192083-9` o `formato: json`. Para **POST**, se envía un cuerpo JSON, por ejemplo:

```
Path parameter:
rut: 76192083-9

Query parameters (si existen):
formato: json
```

```
{
  "auth": {
    "pass": {
      "rut": "11111111-1",
      "clave": "miclave"
    }
  }
}
```

Recuerda: además del body, la API exige el token de conexión (ver siguiente apartado).

Configurar autorización

Las operaciones están protegidas con un **token por conexión**:

1. Busca **Authorize** (o el esquema de seguridad indicado)
2. Introduce tu token según pida el formulario (a veces solo la clave, a veces el texto completo `Token tu_clave`)
3. Confirma con **Authorize**
4. Un candado cerrado suele indicar que las siguientes pruebas enviarán el token automáticamente

La cabecera HTTP correcta es:

```
Authorization: Token <tu_token>
```

El token lo generas en Gestionar conexión

- Abre **Gestionar conexión**
- Ve a la pestaña **Token API**
- Genera el token

Nota

Si todo está bien, el candado aparece cerrado, puedes ver parte del token oculto y suele haber opción **Logout** en el modal de autorización.

Ejecutar y ver resultados

Consideraciones:

1. Revisa parámetros y JSON
2. Clic en **Execute**
3. La herramienta envía la petición al servidor y muestra el resultado en la sección de **Response**.

La pantalla suele mostrar, en este orden:

1. **Curl command** — comando listo para copiar en terminal
2. **Request URL** — URL completa llamada
3. **Response status** — código HTTP (200 éxito, 401 error de autenticación, etc.)
4. **Response headers** — metadatos (límites, créditos, tiempos; ver guía Tu primera petición)
5. **Response body** — JSON (u otro formato) con los datos o el mensaje de error

Ejemplo de bloque tipo cURL (ilustrativo)

```
curl -X 'GET' \
'https://app.apigateway.cl/api/v2/sii/contribuyentes/situacion_tributaria/tercero/76192083-9' \
  -H 'accept: application/json' \
  -H 'Authorization: Token tu_token_aqui'
```

Ejemplo de cuerpo de respuesta exitosa (ilustrativo)

```
{
  "data": {
      "rut": 76192083,
      "dv": "9",
      "razon_social": "EMPRESA DEMO",
      "inicio_actividades": true
  }
}
```

Última actualización el 11/09/2026

Tu Primera Petición

Realiza tu primera consulta exitosa a la API de API Gateway usando la documentación interactiva

Anonymous · 11/09/2026

Tu Primera Petición

Una **API** es el canal por el que tu programa (o la documentación interactiva) envía **peticiones HTTP** y recibe **respuestas**; casi siempre en **JSON** (texto con llaves `{ }` y listas `[ ]`). Para que el servidor sepa **quién eres**, API Gateway exige un **token** ligado a tu **conexión** (no basta con abrir la página web).

URLs que debes conocer:

Qué necesitas	Dónde
Probar y leer la documentación	www.apigateway.cl/docs/api
Iniciar sesión, token, datos de conexión	app.apigateway.cl
Llamadas reales a la API	Base `https://app.apigateway.cl/api/v2/`

En esta guía el ejemplo es un **GET** a la **situación tributaria de un tercero**: no pide clave del SII en el cuerpo y sirve para practicar **token** y la documentación antes de mezclar autenticación al SII.

Paso 1: Obtener tu token de API

El token es la **llave** que identifica **qué conexión** usa la API; esa conexión define **créditos**, **productos** y **configuración**. Sin token válido, las rutas protegidas suelen responder **401 Unauthorized**.

Para obtener el token de API, debes seguir los siguientes pasos:

1. Entra a **app.apigateway.cl** e inicia sesión.
2. Si tienes varias conexiones, en el **panel** selecciona la conexión con la que vas a integrar.
3. Abre **Gestionar conexión**.
4. Pestaña **Token API**.
5. Si no tienes token: **Generar Token**. Si ya hay uno y quieres otro: **Regenerar Token** (el que reemplazas deja de valer).
6. **Copia y guarda** el token en un lugar seguro.

Importante

Trátalo como una contraseña. No lo subas a repositorios públicos ni lo pegues en capturas. Si se filtra,

regenera el token en la misma pestaña.

Para incluir el token en las solicitudes a la API, debes agregar la siguiente cabecera:

Incluye esta cabecera en **todas** las solicitudes a la API: palabra `Token`, un **espacio** y la clave (sin comillas). Hay un resumen en la introducción a la documentación.

```
Authorization: Token <tu_token>
```

Paso 2: Configurar autorización

Para configurar la autorización en la documentación interactiva, debes seguir los siguientes pasos:

1. Abre **www.apigateway.cl/docs/api**.
2. Busca el botón **Authorize** (o el control de seguridad equivalente).
3. En el campo que indique la herramienta, pega el **token** o el valor completo `Token tu_token`, según lo que pida el formulario.
4. Confirma con **Authorize**.
5. Un candado cerrado suele indicar que las pruebas enviarán el token.

Nota

Si recibes **401**, revisa que no falte la palabra `Token`, que no haya espacios de más al copiar/pegar y que el token corresponda a la conexión activa.

Para verificar que la autorización fue exitosa, deberías ver un candado cerrado, una vista parcial del token y la opción **Logout** en el modal si quieres probar sin token.

Paso 3: Seleccionar el endpoint

Para realizar algunas pruebas, puedes seleccionar el endpoint que desees consultar. En este caso, se seleccionará el endpoint de la **Situación tributaria de un contribuyente**.

En la documentación, bajo **Contribuyentes**, localiza:

```
GET /api/v2/sii/contribuyentes/situacion_tributaria/tercero/{rut}
```

¿Por qué este endpoint? Solo pides el RUT en la ruta, no envías `auth` del SII en el cuerpo y la respuesta es rica en información para validar que todo funciona.

Para revisar la información del endpoint, debes seguir los siguientes pasos:

1. Haz clic sobre la operación.

2. Lee la descripción, parámetros y códigos de respuesta listados.

Paso 4: Configurar la petición

Para activar el modo de prueba, debes seguir los siguientes pasos:

1. **Try it out**
2. Edita el campo del **path** `{rut}`

Para completar los parámetros, debes seguir los siguientes pasos:

```
Path parameter
──────────────
rut: 76192083-9
```

Formato del RUT: sin puntos, con guión y dígito verificador. Válido: `76192083-9` . Evitar: `76.192.083-9` .

Paso 5: Ejecutar la petición

Para enviar la solicitud, debes seguir los siguientes pasos:

1. Comprueba el RUT
2. **Execute**
3. Espera la respuesta bajo el bloque de la operación

Para verificar qué sucede al ejecutar la petición, debes seguir los siguientes pasos:

1. La herramienta arma la URL completa (`https://app.apigateway.cl/api/v2/...`)
2. Añade la cabecera `Authorization: Token …` si configuraste **Authorize**
3. Envía el **GET** al servidor
4. Muestra **código HTTP**, **cabeceras** y **cuerpo**

Paso 6: Interpretar la respuesta

Respuesta exitosa (200 OK)

Ejemplo ilustrativo de cuerpo JSON:

```
{
  "data": {
    "actividades": [
```

```
      {
        "afecta": true,
        "categoria": 1,
        "codigo": "620200",
        "glosa": "ACTIVIDADES DE CONSULTORIA DE INFORMATICA Y DE GESTION DE
INSTALACIONE"
      }
    ],
    "documentos_timbrados": [
      {
        "documento": "Factura Electronica",
        "ultimo_timbraje": 2020
      }
    ],
    "dv": "9",
    "excepcion_dte": false,
    "fecha_inicio_actividades": "2012-06-08",
    "inicio_actividades": true,
    "moneda_extranjera": false,
    "obligacion_dte": true,
    "observaciones": {
      "actividad_esporadica": false,
      "domicilio_inexistente": false,
      "inconcurrente": false,
      "no_habido_domicilio": false,
      "no_ubicado": false,
      "suplantado": false,
      "termino_giro": false,
      "termino_giro_obligatorio": false
    },
    "pro_pyme": true,
    "razon_social": "SASCO SPA",
    "rut": 76192083
  }
}
```


Algunos campos de la respuesta

Campo	Descripción
`rut`	RUT numérico (sin guión) del contribuyente consultado
`dv`	Dígito verificador
`razon_social`	Nombre o razón social
`inicio_actividades`	Si tiene inicio de actividades
`fecha_inicio_actividades`	Fecha de inicio si aplica
`actividades`	Listado de actividades económicas vigentes

Campo	Descripción
`documentos_timbrados`	Documentos timbrados y último año de timbraje
`observaciones`	Banderas de situación (domicilio, término de giro, etc.)

La documentación interactiva puede describir cada campo con más detalle.

Cabeceras de respuesta

Junto al JSON vienen **cabeceras HTTP** con cuota, créditos y tiempos. Los números del ejemplo son **ilustrativos**; los tuyos dependen del plan y del momento.

Fragmento típico

```
content-type: application/json
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 847
X-RateLimit-Reset: 1735689600
X-Stats-Time: 1.234
X-Stats-Memory: 45219840
X-Stats-Credits-Used: 0.010
X-Stats-Credits-Remaining: 1523.450
X-Stats-Started-At: 2026-04-08T15:30:00.123456+00:00
X-Stats-Ended-At: 2026-04-08T15:30:01.357890+00:00
X-Stats-Operations: 1
X-Source-Request-Count: 4
X-Source-Request-Time: 0.85
X-Source-Proxy: 0
X-Source-Captcha-Attempts: 0
```

Tabla rápida

Cabecera	Qué indica
`content-type`	Tipo del cuerpo (habitualmente `application/json`).
`X-RateLimit-Limit`	Tope de peticiones de la ventana que aplica a tu conexión.
`X-RateLimit-Remaining`	Cuántas peticiones te quedan antes de llegar al tope.
`X-RateLimit-Reset`	Instante (segundos Unix, UTC) en que se renueva la ventana del contador.
`X-Stats-Credits-Used`	Créditos consumidos en esta respuesta.
`X-Stats-Credits-Remaining`	Créditos disponibles tras esta operación (según estado al responder).
`X-Stats-Time`	Tiempo de procesamiento en el servidor (segundos, texto).

Cabecera	Qué indica
`X-Stats-Started-At` / `X-Stats-Ended-At`	Inicio y fin del procesamiento.
`X-Stats-Operations`	Operaciones internas contadas en esa petición.
`X-Source-Request-Count`	Solicitudes hacia el SII u orígenes externos para armar la respuesta.
`X-Source-Request-Time`	Tiempo agregado de esas solicitudes externas.
`X-Source-Proxy`	`1` si se usó proxy de conexión hacia el SII, `0` si no.
`X-Source-Captcha-Attempts`	Intentos de captcha en la petición (a menudo `0`).

Error 429 (demasiadas peticiones)

Puede venir **`Retry-After`** (segundos sugeridos de espera); **no** reintentes en bucle al instante.

Cabeceras opcionales de sesión SII

Con **login al SII** en el cuerpo pueden aparecer cabeceras **`X-Auth-*`** o **`X-Auth-Session-Problem`**. En este **GET** de ejemplo (sin credenciales SII) muchas **no** vienen: es normal. Si en otro flujo ves **`X-Auth-Session-Problem: 1`**, revisa Manejo de errores.

Analizando el cURL

La documentación suele generar un comando equivalente:

```
curl -X 'GET' \
'https://app.apigateway.cl/api/v2/sii/contribuyentes/situacion_tributaria/tercero/76192083-9' \
  -H 'accept: application/json' \
  -H 'Authorization: Token TU_TOKEN_AQUI'
```

Qué significa cada parte: `-X 'GET'` es el método de lectura; la URL lleva el RUT en la ruta; `accept: application/json` indica que esperas JSON; `Authorization: Token …` es tu token de la pestaña **Token API**. Para ver **cabeceras** en terminal, usa `curl -i` o `curl -D -`.

Ejercicios adicionales

Probar con un RUT que devuelva pocos datos

Prueba por ejemplo `11111111-2` (dígito verificador incorrecto o sin datos en el SII) y observa cómo cambian campos como `razon_social` o `inicio_actividades`.

Probar sin autorización

1. **Logout** en el modal de autorización.
2. Ejecuta de nuevo el mismo **GET**.
3. Deberías ver **401** y un cuerpo de error (el texto exacto puede variar).

Otro GET simple

En la misma documentación, prueba:

```
GET /api/v2/sii/contribuyentes/actividades_economicas
```

(sin parámetros de ruta; revisa si exige solo el token de conexión).

Última actualización el 11/09/2026

Autenticación SII - Método RUT/Clave

Aprende a autenticarte en el SII usando RUT y contraseña para realizar consultas tributarias

Anonymous · 11/09/2026

Autenticación SII - Método RUT/Clave

Muchas consultas al SII exigen **iniciar sesión**; el método más directo es enviar **RUT y clave tributaria** en el **cuerpo** de la petición (JSON). **API Gateway** actúa de intermediario: usa esas credenciales, obtiene los datos y te responde (casi siempre en **JSON**).

Respecto a Tu primera petición, allí solo usaste el **token de conexión** en la cabecera; aquí además envías el bloque **`auth`** con datos del SII en el cuerpo.

¿Por qué POST si “solo consulto”?

Aunque sea una consulta, muchas operaciones usan **POST** porque las credenciales van en el **cuerpo** (body), no en la URL visible: no quedan en el historial del navegador ni en logs de proxy como query string, y es más seguro que poner claves en parámetros `?clave=...` de un GET.

Caché de sesión (aprox. 2 horas)

API Gateway puede **reutilizar la sesión del SII** alrededor de **dos horas** para responder más rápido en consultas seguidas, reducir inicios de sesión repetidos y usar mejor los recursos. Si algo falla con la sesión, existe el parámetro **`auth_cache`** en la URL; más abajo lo explicamos en **Control de caché de sesión**.

Cabecera obligatoria: token de conexión

Además del JSON con `auth`, **toda** petición lleva el token de tu conexión:

```
Authorization: Token <tu_token>
```

Estructura del objeto `auth.pass`

Para autenticar en el SII usando RUT y clave, debes enviar el siguiente JSON:

```
{
```

```
  "auth": {
    "pass": {
      "rut": "11111111-1",
      "clave": "mi_clave_sii"
    }
  }
}
```

Los componentes del objeto `auth.pass` son:

Campo	Formato	Ejemplo	Descripción
`rut`	Texto	`"76192083-9"`	Sin puntos, con guión
`clave`	Texto	`"miClave123"`	Clave tributaria del SII

Paso a paso en la documentación interactiva

Lista de boletas de honorarios **emitidas** por un emisor en un período. Sirve para comprobar que RUT/clave y token funcionan y que entiendes ruta + cuerpo.

Los datos `emisor` y `periodo` van en la **ruta** (path), no en la query:

```
POST /api/v2/sii/bhe/emitidas/documentos/{emisor}/{periodo}
```

1. Localizar el endpoint

1. Abre **www.apigateway.cl/docs/api**.
2. Busca la sección **API BHE** (Boletas de Honorarios Emitidas) o similar.
3. Localiza el **POST** de documentos emitidos con `emisor` y `periodo` en la ruta.
4. Expande la operación.

2. Revisar la documentación

Lee qué significa cada parámetro de ruta, si hay query opcionales (`pagina`, `formato`, etc.) y el esquema del body.

3. Autorización

Configura **Authorize** con tu **token de conexión** (`Authorization: Token …`), igual que en Navegando Swagger UI.

4. Activar “Try it out”

Habilita la edición de campos.

5. Configurar parámetros

Parámetros de ruta (ejemplo)

```
emisor:   11111111-1
periodo:  202401
```

`periodo` suele ser mes calendario `YYYYMM`; algunos listados permiten día `YYYYMMDD`. Confirma en la descripción del endpoint.

Cuerpo (request body)

```
{
  "auth": {
    "pass": {
      "rut": "11111111-1",
      "clave": "tu_clave_sii"
    }
  }
}
```

Nota

`emisor` en la ruta es el RUT cuyas boletas consultas. El **`rut` dentro de `auth.pass`** es quien **inicia sesión** en el SII. A menudo son el mismo contribuyente; en escenarios con representación pueden diferir según permisos en el SII.

6. Ejecutar

Pulsa **Execute** y revisa código, cabeceras y cuerpo. Las cabeceras de créditos y límites se explican en Tu primera petición.

Interpretando la respuesta

Respuesta exitosa (200 OK)

Ejemplo ilustrativo:

```
{
  "data": {
    "n_boletas": 9,
    "n_paginas": 1,
    "boletas": [
      {
        "numero": 133,
        "rut": 76111111,
        "dv": "9",
        "nombre": "TESTING SPA",
```

```
        "fecha": "2025-07-14",
        "total_honorarios": 1000,
        "retencion_emisor": 0,
        "retencion_receptor": 145,
        "total_liquido": 855,
        "sociedad_profesional": "NO",
        "estado": "S",
        "anulada": "2025-07-14",
        "codigo": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "fecha_emision": "2025-07-14",
        "usuario_emisor": "TESTING USER",
        "email_envio": ""
      }
    ]
  }
}
```


Campos importantes (orientativo)

Campo	Descripción
`numero`	Número de la boleta
`rut` / `dv`	RUT del receptor
`nombre`	Nombre o razón social del receptor
`fecha`	Fecha de la boleta
`total_honorarios`	Total de honorarios
`retencion_emisor` / `retencion_receptor`	Retenciones
`total_liquido`	Monto líquido
`estado`	Estado según glosa del SII en la documentación del endpoint

Manejo de errores

Posibles errores que puedes obtener al autenticar en el SII usando RUT y clave:

Error 401: credenciales incorrectas


```
{
  "code": 401,
  "detail": "..."
}
```


Causas habituales: RUT o clave mal escritos, formato de RUT incorrecto.

Qué hacer: probar en www.sii.cl; usar RUT con guión y sin puntos.

Error 401: sesión SII / volver a autenticar

```
{
  "code": 401,
  "detail": "Se debe volver a autenticar al usuario en la sesión del SII."
}
```

Cabecera de ayuda: puede aparecer `X-Auth-Session-Problem: 1`.

Qué hacer: revisa la sección **Control de caché de sesión** más abajo y la guía Manejo de errores.

Control de caché de sesión

Comportamiento por defecto

Tras un inicio de sesión exitoso en el SII, la sesión puede **reutilizarse** un tiempo (del orden de **2 horas**): **acelera** las siguientes consultas, **reduce** logins repetidos y es el comportamiento recomendado para uso normal.

Forzar una nueva sesión (`auth_cache`)

Si la sesión queda en mal estado (por ejemplo mensajes de reautenticación o cabecera `X-Auth-Session-Problem`), la documentación del endpoint puede indicar que agregues a la URL un parámetro de consulta **`auth_cache`**.

Ejemplo de forma (ajusta la ruta al endpoint real):

```
POST
https://app.apigateway.cl/api/v2/sii/bhe/emitidas/documentos/11111111-1/202401?auth_cache=0
```

Cuándo tiene sentido usarlo: después de **cambiar la clave** del SII; cuando el error indica explícitamente problema de sesión y las repeticiones fallan; o durante **pruebas puntuales** de integración.

Advertencia

No añadas `auth_cache=0` en **cada** petición de producción. Forzar un login nuevo una y otra vez puede provocar bloqueos por demasiados accesos al SII y empeorar el rendimiento.

Los valores exactos aceptados (`0`, `1`) y su efecto están descritos en la **documentación interactiva** de cada operación: revísalos ahí antes de automatizar.

Mejores prácticas

1. **Seguridad**

- **No** escribas la clave del SII fija en código que subas a Git.
- **Usa** variables de entorno o un almacén de secretos.
- **Regenera** el token de API si sospechas filtración.

2. **Rendimiento**

- **Aprovecha** la reutilización de sesión cuando todo funciona bien, **agrupa** consultas cuando sea posible y **observa** `X-RateLimit-Remaining` y créditos en las cabeceras.

3. **Ejemplo de uso seguro (idea general)**

```
// ❌ Evitar: clave en el código fuente
const auth = { pass: { rut: "76192083-9", clave: "miClave123" } };

// ✅ Mejor: leer desde entorno seguro
const auth = {
  pass: {
    rut: process.env.SII_RUT,
    clave: process.env.SII_CLAVE
  }
};
```

Tu aplicación debe además enviar la cabecera `Authorization: Token …` con el token de conexión.

Última actualización el 11/09/2026

Autenticación SII - Firma Electrónica

Aprende a autenticarte en el SII usando tu firma electrónica desde la documentación interactiva

Anonymous · 11/09/2026

Autenticación SII - Firma Electrónica

Con la **firma electrónica** (certificado digital) te autenticas en el SII **sin** enviar tu clave tributaria en el mismo flujo que con RUT/clave. Es útil si quieres evitar manejar la contraseña del SII en integraciones, si el trámite o consulta **exige** certificado según las reglas del SII, o si buscas un esquema más alineado con buenas prácticas de seguridad.

Como en Navegando Swagger UI, llevas **`Authorization: Token …`** en cabecera (token de conexión) y en el **cuerpo** el bloque **`auth`** con el certificado (firma).

Token de conexión (siempre)

Toda petición incluye:

```
Authorization: Token <tu_token>
```

Dos formas de enviar el certificado en el JSON

API Gateway admite **dos variantes** dentro de `auth.cert`:

Método 1: `cert-data` y `pkey-data` (recomendado)

- **cert-data** es el certificado público en **PEM** (bloque `BEGIN CERTIFICATE`);
- **pkey-data** es la llave privada en **PEM** (bloque `BEGIN PRIVATE KEY`).

Método 2: `file-data` y `file-pass`

- **file-data** es el archivo `.p12` / `.pfx` codificado en **Base64**;
- **file-pass** es la contraseña de ese archivo.

Preparación del certificado

Para autenticar con el SII usando firma electrónica, puedes utilizar las siguientes opciones:

Opción A: Herramientas en la aplicación

1. Entra a **app.apigateway.cl**.
2. **Gestionar conexión** → pestaña **Certificado**.
3. Allí puedes obtener o convertir material para usar en **PEM** o seguir las instrucciones en pantalla.

Enlace directo al formulario de formato de certificado: Gestionar conexión

Opción B: OpenSSL en tu computador

```
openssl pkcs12 -in firma.p12 -out firma.pem -nodes
```

Luego **separas** en un editor de texto los bloques:

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
...
-----END PRIVATE KEY-----
```

Uso en la documentación interactiva

Método 1: PEM en el body

1. Localiza un endpoint que requiera autenticación SII.
2. **Authorize** con tu **token de conexión**.
3. **Try it out** y pega un JSON como:

```
{
  "auth": {
    "cert": {
      "cert-data": "-----BEGIN CERTIFICATE-----\nMIIG...\n-----END CERTIFICATE-----",
      "pkey-data": "-----BEGIN PRIVATE KEY-----\nMIIE...\n-----END PRIVATE KEY-----"
    }
  }
}
```

4. **Execute**

Método 2: `.p12` en Base64

1. Genera Base64 del archivo (ejemplo en Mac/Linux):

```
base64 -i firma.p12 | tr -d '\n'
```

2. Body:

```
{
  "auth": {
    "cert": {
      "file-data": "[pega aquí el base64 completo]",
      "file-pass": "contraseña_del_p12"
    }
  }
}
```

Comparación rápida

Aspecto	cert-data / pkey-data	file-data / file-pass
Secretos	Sueles evitar reenviar la contraseña del `.p12` en cada llamada si ya extrajiste PEM	Incluye `file-pass` en la petición
Preparación	Extraer PEM	Más directo si solo tienes `.p12`
Uso típico	Integraciones estables	Pruebas rápidas

Respuesta esperada y errores frecuentes

Si la autenticación es correcta

Suele responder con código **2xx** (a menudo **200**) y un cuerpo con los datos solicitados al SII.

Problemas habituales

PEM mal pegado en JSON (**faltan** los `\n` entre líneas), certificado **vencido** o revocado, o estructura JSON incompleta (falta `auth.cert` o campos requeridos).

Tips importantes

1. Saltos de línea en JSON

 Todo en una sola línea sin `\n`:

```
"cert-data": "-----BEGIN CERTIFICATE-----MIIG...-----END CERTIFICATE-----"
```

 Con `\n` como separador de líneas PEM:

```
"cert-data": "-----BEGIN CERTIFICATE-----\nMIIG...\n-----END CERTIFICATE-----"
```

2. Antes de producción

Confirma la **fecha de vencimiento** del certificado, prueba primero un endpoint sencillo y limita quién puede ver los PEM o `.p12` en tu organización.

3. Seguridad

No subas llaves ni `.p12` a repositorios públicos; usa secretos del entorno o un vault en servidores.

Relación con la sesión y `auth_cache`

Si aparecen errores de sesión o cabeceras como `X-Auth-Session-Problem`, revisa Autenticación SII — RUT y clave (sección **Control de caché de sesión**) y Manejo de errores.

Última actualización el 11/09/2026

Explorando Formatos de Respuesta

Aprende a trabajar con diferentes formatos de respuesta en API Gateway

Anonymous · 11/09/2026

Explorando Formatos de Respuesta

La API no siempre devuelve solo JSON. Según el **endpoint**, puedes pedir otros formatos en el parámetro de consulta `formato` (cuando esté documentado): por ejemplo **HTML** para mostrar en pantalla, **CSV** para bajar tablas a Excel o **XML** para integraciones que lo esperan.

> **No todos los endpoints soportan todos los formatos.** La lista permitida sale en la documentación interactiva de cada operación.

Formatos disponibles (orientación general)

Tabla principal

Formato	Parámetro típico	Descripción	Ideal para
JSON	`formato=json` o valor por defecto	Estructurado, fácil de parsear	Aplicaciones, automatización
HTML	`formato=html`	Página similar a la del SII	Mostrar al usuario en navegador o iframe
CSV	`formato=csv`	Texto separado por columnas	Excel, reportes
XML	`formato=xml`	Estructura tipo SII	Integraciones que consumen XML

Modificar el formato en la documentación interactiva

Paso 1: Localizar el parámetro

1. Abre **www.apigateway.cl/docs/api**.
2. Elige un endpoint que liste el query parameter `formato` (u otro nombre indicado).
3. Lee qué valores admite (enum o descripción).

Paso 2: Probar con “Try it out”

1. **Try it out**
2. Completa ruta, body (si hay) y el campo **`formato`**
3. **Execute**
4. Observa si el **cuerpo** cambia de JSON a HTML, texto CSV, etc.

Ejemplo de URLs (patrón)

Sustituye `...` por la ruta real del endpoint (aparece en la documentación):

```
# JSON (habitual por defecto)
https://app.apigateway.cl/api/v2/sii/.../datos

# HTML
https://app.apigateway.cl/api/v2/sii/.../datos?formato=html

# CSV
https://app.apigateway.cl/api/v2/sii/.../datos?formato=csv
```

Recuerda incluir siempre **`Authorization: Token …`** al probar fuera del navegador.

Trabajar con cada formato

JSON — procesamiento automático

Cuándo usar: backends, scripts, APIs entre sistemas.

Ejemplo ilustrativo:

```
{
  "rut": "11111111-1",
  "razon_social": "EMPRESA DEMO",
  "actividades": [
    {
      "codigo": 620100,
      "descripcion": "ACTIVIDADES DE PROGRAMACION INFORMATICA"
    }
  ]
}
```

HTML — mostrar en pantalla

Cuándo usar: quieres una vista “tipo sitio SII” sin maquetar tú todo.

Características: suele traer estilos; se puede incrustar en **iframe** (valora implicancias de seguridad y cookies según tu caso).

CSV — Excel y análisis

Cuándo usar: exportar a hoja de cálculo.

La **coma** suele ser el valor por defecto; el **punto y coma** es cómodo para Excel en español.

Ejemplo ilustrativo:

```
rut;razon_social;actividad_codigo;actividad_descripcion
11111111-1;EMPRESA DEMO;620100;ACTIVIDADES DE PROGRAMACION INFORMATICA
```

XML — interoperabilidad

Cuándo usar: el sistema destino exige XML o validación con esquemas.

Casos de uso prácticos

Caso 1: Panel para un usuario final

Necesidad: mostrar una pantalla con datos del SII. **Opción:** `formato=html` en un iframe (ajusta URL y token según tu arquitectura; muchas veces el HTML se obtiene vía llamada servidor a servidor, no exponiendo el token en el navegador).

```
<iframe
  title="Consulta SII"
  src="https://app.apigateway.cl/api/v2/sii/.../datos?formato=html"
  width="100%"
  height="600"
></iframe>
```

> En integraciones reales, valorar **autenticación** y **no** incrustar el token en URLs visibles al usuario sin analizar riesgos.

Caso 2: Reporte mensual en Excel

Necesidad: analizar en hoja de cálculo. **Opción:** `formato=csv`.

Caso 3: ERP u otro backend

Necesidad: procesar en servidor. **Opción:** JSON por defecto.

Caso 4: Consumo en XML

Necesidad: otro sistema solo acepta XML. **Opción:** `formato=xml` si el endpoint lo documenta.

Consideraciones importantes

Disponibilidad

Si **no** ves `formato` en los parámetros, ese endpoint probablemente solo devuelve un tipo de respuesta. Respeta el **Content-Type** de la respuesta (`application/json`, `text/html`, etc.).

Buena práctica

En integraciones críticas, fija explícitamente `formato` si quieres evitar sorpresas si cambia el valor por defecto en el futuro.

Última actualización el 11/09/2026

Manejo de Errores

Aprende a identificar y resolver errores al usar API Gateway

Anonymous · 11/09/2026

Manejo de Errores

Los errores son **normales** al integrar; lo clave es saber **dónde mirar**: **código HTTP** (ej. 200, 401, 429), **cuerpo** de la respuesta (mensaje) y **cabeceras** (límites, tiempos, señales de sesión). En la documentación interactiva, con **Try it out**, sueles ver **todo junto**.

Códigos de error principales

Tabla de referencia rápida

Código	Nombre	Significado	Acción habitual
400	Bad Request	Petición inválida o no procesada	Revisar JSON, parámetros y formato
401	Unauthorized	Token de conexión inválido, credenciales SII incorrectas o sesión SII	Token, RUT/clave, certificado o sesión
402	Payment Required	Créditos insuficientes o IP compartida con otra conexión	Cargar créditos o revisar la IP de origen
403	Forbidden	Sin permiso al recurso u operación	Productos de la conexión o estado de cuenta
404	Not Found	Ruta o recurso no encontrado	URL y parámetros
405	Method Not Allowed	Método HTTP incorrecto	GET vs POST, etc.
406	Not Acceptable	Formato de respuesta no aceptado	Parámetro `formato` o cabeceras
409	Conflict	Conflicto con datos en el SII	Revisar datos enviados
410	Gone	Recurso ya no disponible en la API	Actualizar según documentación vigente
423	Locked	Cuenta bloqueada por términos	Revisar términos y soporte
429	Too Many Requests	Límite de uso alcanzado	Esperar, revisar cabeceras y plan

Nota

Con el tiempo pueden añadirse o ajustarse códigos. Diseña tu software para tratar también un **caso por defecto** cuando llegue un código que no esperabas.

Bloqueo de la cuenta

Rige el uso de la plataforma según los términos y condiciones. Entre los factores que pueden afectar están **muchas respuestas 429** por un patrón de uso.

Identificando errores

Dónde está la información

1. **Status code** (arriba en la respuesta de la herramienta o en la primera línea de `curl -i`).
2. **Response body** — mensaje en JSON u otro formato.
3. **Response headers** — límites, `Retry-After`, créditos, señales de sesión (ver Tu primera petición).

Ejemplo visual de error 401 (ilustrativo)

```
Code: 401 Unauthorized

Response body:
{
  "detail": "Token de conexión inválido."
}

Response headers:
content-type: application/json
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 999
```

El texto de `detail` puede variar según idioma y caso.

Interpretando cabeceras de respuesta

Rate limiting (muy frecuente)

Cabecera	Significado	Ejemplo
`X-RateLimit-Limit`	Tope de la ventana que aplica	1000
`X-RateLimit-Remaining`	Peticiones que aún puedes hacer	847
`X-RateLimit-Reset`	Momento (Unix) en que se renueva la ventana	1735689600

En respuestas **exitosas**, mira `X-RateLimit-Remaining` para reducir sorpresas antes de llegar al **429**.

Cuando llegas al límite (429)

Cabecera	Significado	Ejemplo
`Retry-After`	Segundos sugeridos antes de reintentar	3600

Errores comunes y soluciones

Error 401: token de conexión

```
{
  "detail": "Token de conexión inválido."
}
```

O mensajes genéricos de no autenticado.

Solución: usa la cabecera exacta `Authorization: Token <tu_token>` (palabra `Token`, espacio, clave). El token se genera en Gestionar conexión; si **regeneraste** el token, actualiza tu integración.

Error 401: credenciales SII (RUT/clave)

Solución: verificar en www.sii.cl; formato de RUT con guión y sin puntos; JSON `auth.pass` completo.

Error 401: sesión SII / reautenticación

```
{
  "code": 401,
  "detail": "Se debe volver a autenticar al usuario en la sesión del SII."
}
```

Cabecera útil: `X-Auth-Session-Problem: 1`.

Solución inmediata: revisa la sección **Control de caché de sesión** más abajo (parámetro `auth_cache` **solo** cuando corresponda).

Error 429: demasiadas peticiones

```
{
  "detail": "Too Many Attempts."
}
```

Solución:

1. Lee `Retry-After` si viene en cabeceras.
2. Espera esos segundos antes de reintentar en masa.
3. Revisa plan y optimiza frecuencia de llamadas.

Error 402: créditos insuficientes

```
{
  "status": 402,
  "code": "error",
  "detail": "Créditos insuficientes para realizar la consulta. Créditos disponibles:
0.500, costo de esta operación: 1.000"
}
```

Causa: el saldo de la conexión no alcanza para cubrir el costo de la operación solicitada. El mensaje incluye ambos valores, así que sabes exactamente cuánto falta.

Solución: compra créditos en Gestionar conexión. El costo en créditos de cada recurso está en Precios y en la documentación de la API.

Anticípate

La cabecera `X-Stats-Credits-Remaining` viene en las respuestas exitosas con el saldo que te queda. Monitoréala para recargar antes de quedarte sin créditos a mitad de un proceso.

Error 402: IP en uso por otra conexión

```
{
  "status": 402,
  "code": "error",
  "detail": "No está permitido realizar consultas desde la IP 203.0.113.99, ya que se
encuentra en uso por otra conexión. Para consultar desde esta IP debe realizar la
compra de créditos."
}
```

Causa: las conexiones que aún **no han pagado ningún cobro** solo pueden consultar desde una IP que ninguna otra conexión haya utilizado. Si consultas desde una IP que ya está en uso, la petición se rechaza.

Solución: realiza la **compra de créditos**. Una vez registrado el pago, la conexión queda habilitada para consultar desde cualquier IP, incluidas las compartidas.

Alternativa mientras tanto: consultar desde una IP que no esté siendo usada por otra conexión.

Error 400: petición mal formada

Causas: JSON inválido, campos obligatorios faltantes, tipos incorrectos.

En la documentación interactiva: revisa el esquema del body; los editores suelen marcar errores.

Control de caché de sesión (SII)

Si envías **RUT/clave** o **certificado** en el cuerpo, API Gateway puede **reutilizar la sesión del SII** un tiempo (del orden de **2 horas**): es **normal**, ayuda al **rendimiento** y evita **demasiados logins** seguidos al SII.

Comportamiento por defecto

Tras un login exitoso, las siguientes consultas pueden **reaprovechar** la sesión; **no** debes forzar un login nuevo en cada llamada si todo funciona.

Forzar nueva sesión (`auth_cache`)

Ante errores de **reautenticación** o cabecera `X-Auth-Session-Problem: 1`, revisa la **documentación del endpoint**; a veces indican añadir a la URL algo como:

```
?auth_cache=0
```

Ejemplo de idea (la ruta real es la de tu operación):

```
POST
https://app.apigateway.cl/api/v2/sii/bhe/emitidas/documentos/11111111-1/202401?auth_cache=0
```

Cuándo usarlo: tras **cambiar la clave** del SII; cuando el error indica claramente problema de sesión y los reintentos normales fallan; o en **depuración puntual**.

Advertencia

No uses `auth_cache=0` en **todas** las peticiones de producción. Cada forzado puede equivaler a un nuevo inicio de sesión en el SII y aumentar riesgo de bloqueos o lentitud.

Los valores permitidos de `auth_cache` y su significado están en la **documentación interactiva** de cada operación.

Estrategias de prevención

1. Monitoreo de límites

Si `X-RateLimit-Remaining` baja mucho, **espacia** las llamadas y valora **caché** en tu sistema para no

repetir la misma consulta.

2. Sesiones SII

No abuses de `auth_cache=0`. Si aparece `X-Auth-Session-Problem`, trata la sesión antes de insistir a ciegas.

3. Validación previa

Antes de enviar, comprueba **JSON** bien formado, **campos obligatorios** presentes, **tipos** correctos y **token** vigente.

Última actualización el 11/09/2026