



DOCUMENTACION PARA INTEGRADORES

Guía de integración Factly API

Conexión de sistemas externos con la facturación electrónica DGII por medio de Factly: autenticación, ambientes, payloads, estados, webhooks, Postman y paso a producción.

Version

v1 inicial

Fecha

2026-06-10

Audiencia

Clientes e integradores

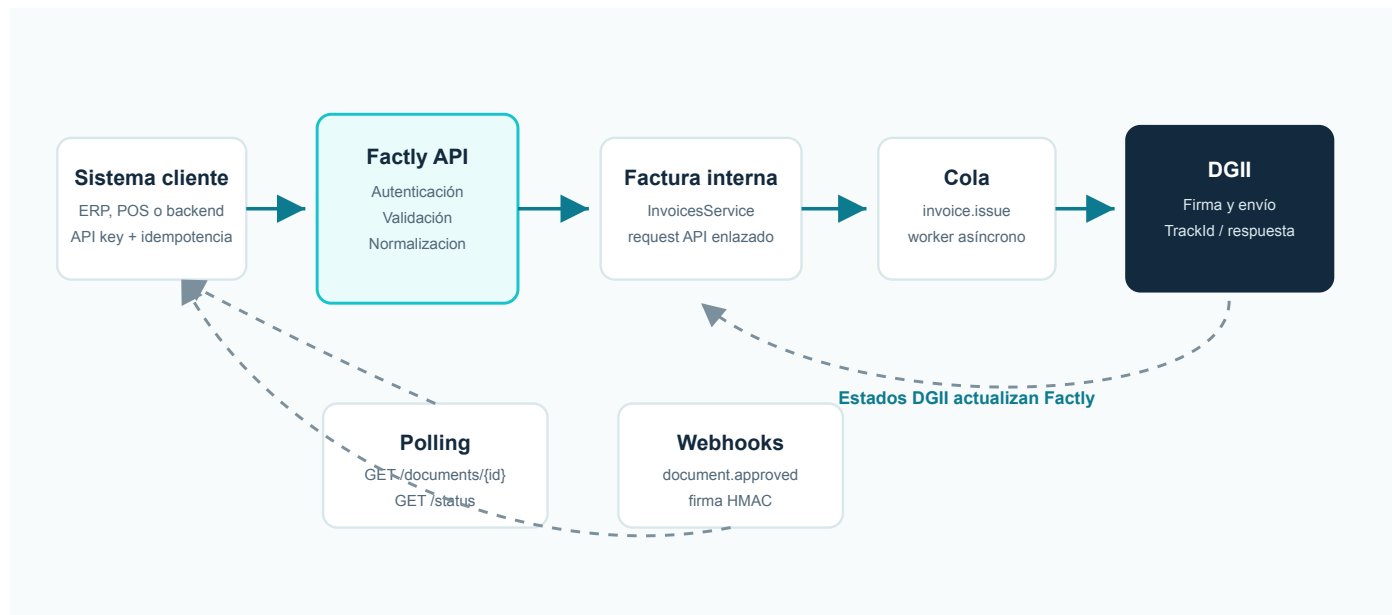
Factly - API para facturación electrónica en República Dominicana. Documento preparado para revision comercial y técnica.

Contenido

- 1** Antes de empezar
- 2** URLs base separadas
- 3** Ambientes
- 4** Configuración del tenant en Factly
- 5** Autenticación
- 6** Headers requeridos
- 7** Flujo recomendado de integración
- 8** Endpoints disponibles
- 9** Crear un documento JSON
- 10** Campos del payload JSON
- 11** Tipos de documento
- 12** Notas de crédito y débito
- 13** Respuesta de creacion
- 14** Estados
- 15** Consultar estado
- 16** XML directo
- 17** Descargar XML y PDF
- 18** Uso y cuotas
- 19** Webhooks
- 20** Errores
- 21** Rate limits y trazabilidad
- 22** Postman
- 23** OpenAPI
- 24** Criterios para pasar a producción
- 25** Fuentes de buenas practicas usadas

Factly API permite que un ERP, POS, ecommerce o sistema interno envíe documentos a Factly para que Factly cree la factura, genere el XML e-CF, lo firme, lo envíe a DGII cuando aplique y devuelva estado, XML, PDF y eventos.

Esta guía está escrita para clientes integradores. Los detalles internos de arquitectura viven en [docs/factly-api-user-manual.md](#).



1. Antes de empezar

Solicita a Factly:

- API key de **sandbox**.
- API key de **production**, solo cuando el tenant este listo para producción.
- URL base del ambiente.
- Confirmación del modo de sandbox:
 - **Sandbox interno Factly**, para probar contrato e idempotencia sin DGII.
 - **QA DGII**, para probar firma, XML y envío real a DGII de pruebas.
- Confirmación de formatos habilitados: **JSON** o **XML**.
- Lista de eventos webhook habilitados, si usaras callbacks.
- Secret compartido de webhook, si usaras firma HMAC. Factly lo guarda protegido y no debe tratarse como valor recuperable.

Prepara en tu sistema:

- Generación de un **externalId** único por documento.
- Generación de una llave **Idempotency-Key** única por intento lógico de creación.
- Manejo asincrónico de estados: polling o webhooks.

- Registro del `id` devuelto por Factly y del `externalId` propio.
- Manejo de errores DGII devueltos en `validationErrors` y `dgii.messages`.

2. URLs base separadas

Factly mantiene sandbox y producción como rutas separadas y siempre disponibles para el tenant cuando el módulo API está activo.

Sandbox:

```
https://app.factly.do/api/public/v1/sandbox
```

Producción:

```
https://app.factly.do/api/public/v1/production
```

Local o QA interno, si Factly te lo indica:

```
http://app_factly/api/public/v1/sandbox  
http://app_factly/api/public/v1/production
```

Ejemplos:

```
POST https://app.factly.do/api/public/v1/sandbox/documents  
GET https://app.factly.do/api/public/v1/production/usage
```

Los catálogos compartidos usan la ruta general:

```
GET https://app.factly.do/api/public/v1/catalogs/document-types
```

Importante: no cambies solo la URL al pasar de sandbox a producción. También debes usar la API key de ese ambiente.

3. Ambientes

Ambiente en URL	Uso	Salida a DGII
sandbox con modo interno	Probar autenticación, contrato JSON, idempotencia, respuestas y webhooks	No
sandbox con modo QA DGII	Probar XML, firma y envío contra DGII de pruebas	Si, a DGII de pruebas
production	Emitir documentos reales	Si, a DGII producción

Importante: las API keys son separadas por ambiente. Una llave de **sandbox** no funciona en **production**.

4. Configuración del tenant en Factly

La configuración operativa se realiza desde:

platform-admin -> Tenants -> Gestionar -> API

Configuración requerida antes de entregar una API key



Checklist administrativo:

1. Activar el modulo API en el plan o como override del tenant.
2. Definir si el tenant puede usar **sandbox**.
3. Elegir el modo de sandbox: interno o **QA DGII**.
4. Habilitar formatos. Para integraciones nuevas usa **JSON**.

5. Configurar cuotas, modelo comercial y política de excedente.
6. Generar una API key por ambiente.
7. Configurar webhooks por ambiente, si aplica.
8. Para `QA DGII` o `production`, confirmar certificado `.p12`, fase DGII, secuencias e-CF y ambiente del servidor.

Desde el panel del tenant solo se controla si puede usar sandbox. El modo de sandbox, formatos y cuotas se administran desde plataforma.

Seguridad operativa:

- La API key se muestra una sola vez al generarse.
- Las listas administrativas muestran prefijo, estado y ultimo uso, pero nunca muestran el token completo.
- Si el nombre de una llave parece contener un secreto, Factly lo redacta en listados y muestra un nombre seguro.
- El secret de webhook se almacena protegido; conserva el valor original en tu gestor de secretos.

5. Autenticación

Puedes enviar la API key de cualquiera de estas dos formas:

```
Authorization: Bearer TU_API_KEY
```

```
X-API-Key: TU_API_KEY
```

Recomendacion: usa `Authorization: Bearer`.

Nunca expongas la API key en frontend, apps moviles sin backend o repositorios. Factly solo muestra la llave una vez al generarla.

6. Headers requeridos

Para crear documentos JSON:

```
Authorization: Bearer TU_API_KEY
Content-Type: application/json
Idempotency-Key: pedido-1001-v1
```

Idempotency-Key evita duplicados. Si repites la misma llave para el mismo tenant y ambiente, Factly devuelve el mismo request en vez de crear otro documento.

7. Flujo recomendado de integración

1. Llama `GET /v1/catalogs/document-types` para confirmar tipos disponibles.
2. Llama `GET /v1/sandbox/usage` para probar autenticación.
3. Crea un documento con `POST /v1/sandbox/documents`.
4. Guarda el `id` de Factly y tu `externalId`.
5. Consulta `GET /v1/sandbox/documents/{id}/status`.
6. Descarga XML y PDF cuando estén disponibles.
7. Configura webhook y prueba `POST /v1/sandbox/webhooks/test`.
8. Si usaras DGII real, repite en `sandbox` con modo `QA DGII`.
9. Pasa a `production` solo después de validar datos fiscales, secuencias, certificado, errores y reconciliación.

8. Endpoints disponibles

Método	Endpoint	Uso
GET	<code>/v1/catalogs/document-types</code>	Lista tipos de documento soportados
GET	<code>/v1/{environment}/usage</code>	Consulta uso y cuota del ciclo
POST	<code>/v1/{environment}/documents</code>	Crea documento desde JSON
POST	<code>/v1/{environment}/documents/xml</code>	Crea documento XML directo, solo sandbox interno
GET	<code>/v1/{environment}/documents/{id}</code>	Consulta documento completo
GET	<code>/v1/{environment}/documents/{id}/status</code>	Consulta estado resumido
GET	<code>/v1/{environment}/documents/{id}/xml</code>	Descarga XML
GET	<code>/v1/{environment}/documents/{id}/pdf</code>	Descarga PDF preliminar o final
POST	<code>/v1/{environment}/documents/{id}/retry</code>	Reintenta emisión cuando aplique
POST	<code>/v1/{environment}/webhooks/test</code>	Dispara evento de prueba

`environment` solo acepta `sandbox` o `production`.

9. Crear un documento JSON

Endpoint:

```
POST /v1/{environment}/documents
```

Ejemplo cURL para sandbox interno:

```
curl -sS -X POST "https://app.factly.do/api/public/v1/sandbox/documents" \
-H "Authorization: Bearer TU_API_KEY_SANDBOX" \
-H "Content-Type: application/json" \
-H "Idempotency-Key: pedido-1001-v1" \
-d '{
  "externalId": "pedido-1001",
  "documentType": "32",
  "issueDate": "2026-05-18",
  "billingType": "CONTADO",
  "currency": "DOP",
  "customerName": "Consumidor final",
  "customerDocument": "",
  "customerDocumentType": "",
  "lines": [
    {
      "description": "Producto de prueba API",
      "qty": 1,
      "unitPrice": 1000,
      "taxRate": 18,
      "discountAmount": 0,
      "unidadMedida": "43"
    }
  ]
}'
```

Ejemplo para QA DGII o producción:

```
{
  "externalId": "pedido-1001",
  "documentType": "31",
  "encf": "E310000000018",
  "issueDate": "2026-05-18",
  "fechaVencimientoSecuencia": "2026-12-31",
  "billingType": "CREDITO",
  "creditDays": 30,
  "currency": "DOP",
  "customerName": "Cliente Ejemplo SRL",
  "customerDocument": "101123456",
  "customerDocumentType": "rnc",
  "customerEmail": "facturas@cliente-ejemplo.com",
  "lines": [
    {
      "description": "Servicio de integración",
      "qty": 1,
      "unitPrice": 3500,
      "taxRate": 18,
      "discountAmount": 0,
      "unidadMedida": "43"
    }
  ]
}
```

Notas DGII:

- En flujo real, envía **encf**.
- Envía **fechaVencimientoSecuencia** con la fecha correcta de la secuencia DGII del tenant.
- Para notas de crédito o débito, envía los campos de referencia.
- En **E32** menor a RD\$250,000, Factly puede usar flujo **RFCE**.

10. Campos del payload JSON

Campo	Requerido	Descripción
<code>externalId</code>	Si	ID único del documento en tu sistema
<code>documentType</code>	Si	Tipo e-CF: <code>31</code> , <code>32</code> , <code>33</code> , <code>34</code> , etc.
<code>issueDate</code>	No	Fecha del documento en formato <code>YYYY-MM-DD</code> ; si falta, Factly usa la fecha actual
<code>encf</code>	En flujo real	e-NCF asignado al documento
<code>fechaVencimientoSecuencia</code>	Recomendado en flujo real	Fecha de vencimiento de la secuencia DGII
<code>billingType</code>	No	<code>CONTADO</code> o <code>CREDITO</code>
<code>creditDays</code>	No	Días de crédito
<code>currency</code>	No	Moneda, normalmente <code>DOP</code>
<code>notes</code>	No	Notas internas o visibles segun configuración
<code>customerName</code>	Segun tipo	Nombre o razon social del receptor
<code>customerDocument</code>	Segun tipo	RNC, cedula u otro documento
<code>customerDocumentType</code>	Segun tipo	Por ejemplo <code>rnc</code>
<code>customerEmail</code>	No	Email para reconciliacion o envío
<code>referenceEncf</code>	Para notas	e-NCF del documento afectado
<code>referenceIssueDate</code>	Para notas	Fecha del documento afectado
<code>referenceModificationCode</code>	Para notas	Código DGII de modificación
<code>lines</code>	Si	Arreglo de lineas

Campos por línea:

Campo	Requerido	Descripción
<code>description</code>	Si	Descripción del producto o servicio
<code>qty</code>	Si	Cantidad, mayor que cero
<code>unitPrice</code>	Si	Precio unitario
<code>taxRate</code>	No	Tasa ITBIS, por defecto <code>18</code> si se omite
<code>discountAmount</code>	No	Descuento de la línea
<code>unidadMedida</code>	No	Código DGII de unidad; default comun: <code>43</code>

Variantes aceptadas por el API:

- `external_id` por `externalId`
- `eNCF` o `e_ncf` por `encf`
- `fecha_vencimiento_secuencia` por `fechaVencimientoSecuencia`
- `items` por `lines`
- `quantity` por `qty`
- `unit_price` o `price` por `unitPrice`
- `tax_rate` por `taxRate`
- `discount_amount` por `discountAmount`
- `unitMeasureCode` o `unit_measure_code` por `unidadMedida`

11. Tipos de documento

Código	Documento
31	Factura de Crédito Fiscal Electrónica
32	Factura de Consumo Electrónica
33	Nota de Débito Electrónica
34	Nota de Crédito Electrónica
41	Compras Electronico
43	Gastos Menores Electronico
44	Regimenes Especiales Electronico
45	Gubernamental Electronico
46	Exportacion Electronico
47	Pagos al Exterior Electronico

Los tipos más usados por integradores son 31 , 32 , 33 y 34 .

12. Notas de crédito y débito

Para 33 y 34 , envía referencia al comprobante original:

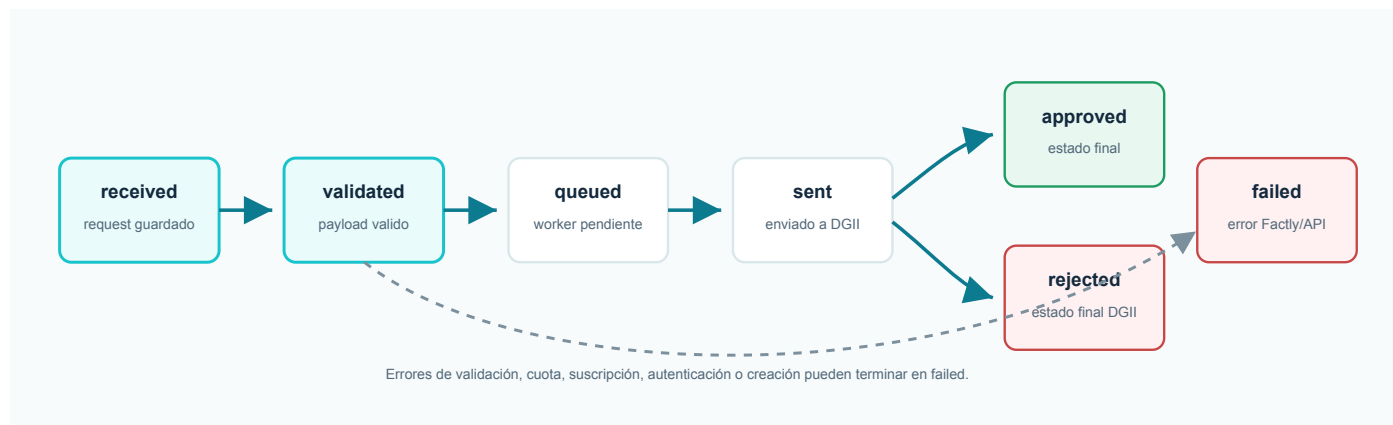
```
{
  "externalId": "nc-4001",
  "documentType": "34",
  "encf": "E340000000001",
  "issueDate": "2026-05-18",
  "fechaVencimientoSecuencia": "2026-12-31",
  "customerName": "Cliente Ejemplo SRL",
  "customerDocument": "101123456",
  "customerDocumentType": "rnc",
  "referenceEncf": "E3100000000017",
  "referenceIssueDate": "2026-05-01",
  "referenceModificationCode": "1",
  "lines": [
    {
      "description": "Devolucion parcial",
      "qty": 1,
      "unitPrice": 1000,
      "taxRate": 18,
      "discountAmount": 0,
      "unidadMedida": "43"
    }
  ]
}
```

13. Respuesta de creacion

Ejemplo simplificado:

```
{
  "id": 12,
  "externalId": "pedido-1001",
  "environment": "sandbox",
  "requestFormat": "json",
  "status": "queued",
  "invoiceId": 80,
  "receivedAt": "2026-05-18 10:30:00",
  "completedAt": "",
  "links": {
    "self": "/api/public/v1/sandbox/documents/12",
    "status": "/api/public/v1/sandbox/documents/12/status",
    "pdf": "/api/public/v1/sandbox/documents/12/pdf",
    "xml": "/api/public/v1/sandbox/documents/12/xml"
  },
  "document": {
    "invoiceId": 80,
    "internalNumber": "FAC-00000050",
    "encf": "E310000000018",
    "dgiiStatus": "Enviado",
    "status": "SENT",
    "issueDate": "2026-05-18",
    "total": 4130,
    "currency": "DOP",
    "pdfFinalAvailable": false,
    "xmlPreviewAvailable": true
  }
}
```

14. Estados



Estados del request API:

Estado	Significado
received	Factly recibio y guardo el request
validated	El payload paso validaciones basicas
queued	El documento fue creado y queda pendiente del worker
sent	El documento fue enviado a DGII o esta en seguimiento
approved	Flujo cerrado como aprobado
rejected	Flujo cerrado como rechazado
failed	Error de Factly/API antes de cerrar como documento DGII valido

No asumas que la respuesta de creacion siempre es final. En flujos reales, usa polling o webhooks.

15. Consultar estado

```
curl -sS "https://app.factly.do/api/public/v1/sandbox/documents/12/status" \
-H "Authorization: Bearer TU_API_KEY_SANDBOX"
```

Ejemplo con rechazo DGII:

```
{
  "id": 12,
  "externalId": "pedido-1001",
  "environment": "sandbox",
  "status": "rejected",
  "invoiceId": 80,
  "document": {
    "invoiceId": 80,
    "encf": "E310000000019",
    "dgiiStatus": "Rechazado",
    "status": "REJECTED",
    "total": 4130,
    "currency": "DOP"
  },
  "dgii": {
    "transmissionStatus": "REJECTED",
    "trackId": "...",
    "environment": "testecf",
    "stateText": "Rechazado",
    "messages": [
      {
        "codigo": "145",
        "valor": "Fecha de vencimiento de secuencia invalida."
      }
    ]
  },
  "validationErrors": [
    "145: Fecha de vencimiento de secuencia invalida."
  ]
}
```

16. XML directo

Endpoint:

```
POST /v1/{environment}/documents/xml
```

Estado actual del producto:

- Disponible solo en `sandbox` con modo `Sandbox interno Factly`.
- Requiere que el tenant tenga formato XML habilitado.

- No se debe usar como ruta de producción todavía.
- Para producción, usa JSON y deja que Factly genere, firme y envíe el XML e-CF desde la tubería central.

17. Descargar XML y PDF

XML:

```
GET /v1/{environment}/documents/{id}/xml
```

PDF:

```
GET /v1/{environment}/documents/{id}/pdf
```

Comportamiento:

- Si el documento ya fue aprobado y existe RI/PDF final, Factly devuelve el final.
- Si no, Factly puede devolver PDF preliminar.
- Si no hay XML real disponible, Factly puede devolver un preview XML.

18. Uso y cuotas

```
GET /v1/{environment}/usage
```

Respuesta:

```
{
  "environment": "production",
  "cycleStart": "2026-05-01",
  "cycleEnd": "2026-05-31",
  "documentsIncluded": 1000,
  "documentsUsed": 235,
  "documentsOverage": 0,
  "amountEstimated": 0,
  "billingStrategy": "monthly_quota",
  "overagePolicy": "block"
}
```

Si `overagePolicy` es `block` y se agota la cuota, los `POST` pueden fallar con `API_QUOTA_EXCEEDED`.

19. Webhooks

Los webhooks evitan hacer polling constante.

Eventos disponibles:

- `document.received`
- `document.validated`
- `document.queued`
- `document.sent`
- `document.approved`
- `document.rejected`
- `document.failed`

Headers enviados:

```
Content-Type: application/json
X-Factly-Event: document.approved
X-Factly-Environment: sandbox
X-Factly-Timestamp: 2026-05-18T10:30:00
X-Factly-Signature: firma_hex
```

La firma se calcula así:

```
HMAC_SHA256_HEX(timestamp + "." + bodyJson, secret)
```

Factly usa el secret original configurado para calcular la firma. En la administración, el secret puede mostrarse solo como configurado o enmascarado; no dependas de poder recuperarlo desde Factly.

Ejemplo body:

```
{
  "event": "document.approved",
  "environment": "sandbox",
  "timestamp": "2026-05-18T10:30:00",
  "data": {
    "id": 12,
    "externalId": "pedido-1001",
    "status": "approved",
    "invoiceId": 80
  }
}
```

Verificación en Node.js:

```
import crypto from "node:crypto";

function verifyFactlyWebhook({ timestamp, rawBody, signature, secret }) {
  const expected = crypto
    .createHmac("sha256", secret)
    .update(`${timestamp}.${rawBody}`)
    .digest("hex");

  return crypto.timingSafeEqual(
    Buffer.from(expected, "hex"),
    Buffer.from(signature, "hex")
  );
}
```

Verificación en PHP:

```
<?php
$timestamp = $_SERVER["HTTP_X_FACTLY_TIMESTAMP"] ?? "";
$signature = $_SERVER["HTTP_X_FACTLY_SIGNATURE"] ?? "";
$rawBody = file_get_contents("php://input");
$secret = getenv("FACTLY_WEBHOOK_SECRET");

$expected = hash_hmac("sha256", $timestamp . "." . $rawBody, $secret);

if (!hash_equals($expected, $signature)) {
    http_response_code(401);
    exit("invalid signature");
}

http_response_code(204);
```

Buenas practicas:

- Responde `2xx` rápido.
- Verifica firma antes de procesar.
- Deduplica usando `event`, `data.id`, `externalId` y estado.
- Registra el body completo para auditoria.
- Reintenta en tu lado solo procesos internos; no crees documentos sin nueva `Idempotency-Key`.

20. Errores

Forma general:

```
{
  "code": "API_DOCUMENT_VALIDATION_ERROR",
  "message": "El payload del documento no es valido.",
  "details": {
    "errors": [
      "lines[1].qty debe ser mayor que cero"
    ]
  }
}
```

Errores frecuentes:

Código	HTTP	Causa
<code>API_AUTH_MISSING_TOKEN</code>	401	No enviaste API key
<code>API_AUTH_INVALID_TOKEN</code>	401	API key invalida
<code>API_AUTH_INACTIVE_CREDENTIAL</code>	403	Llave revocada o inactiva
<code>API_ENVIRONMENT_MISMATCH</code>	403	Llave de otro ambiente
<code>API_ENVIRONMENT_INVALID</code>	400	Ambiente distinto de <code>sandbox</code> o <code>production</code>
<code>API_SANDBOX_DISABLED</code>	403	Sandbox desactivado para el tenant
<code>API_MODULE_DISABLED</code>	403	Modulo API no incluido
<code>API_JSON_DISABLED</code>	403	Formato JSON no habilitado para el tenant
<code>API_XML_DISABLED</code>	403	Formato XML no habilitado para el tenant
<code>API_IDEMPOTENCY_REQUIRED</code>	400	Falta <code>Idempotency-Key</code>
<code>API_EXTERNAL_ID_REQUIRED</code>	400	Falta <code>externalId</code>
<code>API_EXTERNAL_ID_EXISTS</code>	409	Ya existe un documento con ese <code>externalId</code>
<code>API_DOCUMENT_VALIDATION_ERROR</code>	422	Payload invalido
<code>API_ENCF_REQUIRED</code>	422	Falta <code>encf</code> en flujo real
<code>API_QUOTA_EXCEEDED</code>	402	Cuota agotada
<code>API_XML_REQUIRED</code>	400	Falta contenido XML en el endpoint XML
<code>API_XML_UNSUPPORTED_ENVIRONMENT</code>	409	XML directo fuera de sandbox interno
<code>API_DOCUMENT_NOT_FOUND</code>	404	Documento no encontrado para tenant/ambiente
<code>API_DOCUMENT_RETRY_NOT_AVAILABLE</code>	409	El documento no tiene factura enlazada para reintento
<code>API_DOCUMENT_PDF_UNAVAILABLE</code>	409	El documento aún no tiene PDF disponible
<code>API_DOCUMENT_XML_UNAVAILABLE</code>	409	El documento aún no tiene XML disponible
<code>API_TENANT_USER_REQUIRED</code>	409	El tenant no tiene usuario activo para crear documentos API
<code>SUBSCRIPTION_PAST_DUE</code>	403	Cuenta en mora, API en solo lectura
<code>RATE_LIMIT_EXCEEDED</code>	429	Exceso de solicitudes

Errores DGII pueden llegar en:

- `validationErrors`
- `dgii.errorCode`

- `dgii.errorMessage`
- `dgii.messages`

21. Rate limits y trazabilidad

Factly puede devolver:

```
X-Request-Id: 9d4f...  
X-RateLimit-Limit: 240  
X-RateLimit-Remaining: 239  
X-RateLimit-Reset: 60
```

Guarda `X-Request-Id` en tus logs. Ayuda a soporte a rastrear una llamada específica.

22. Postman

Archivos incluidos:

- `docs/postman/Factly-API-Minimal.postman_collection.json`
- `docs/postman/Factly-API-Minimal.postman_environment.json`

Orden recomendado:

1. Importa coleccion y environment.
2. Completa `base_url`, `api_environment` y `api_key`.
3. Si pruebas `QA DGII` o `production`, completa `encf` y `sequence_expiration_date`.
4. Ejecuta `Catálogos – Tipos de documento`.
5. Ejecuta `Uso actual`.
6. Ejecuta `Crear documento JSON`.
7. Ejecuta `Consultar estado`.
8. Prueba XML, PDF y webhook.

23. OpenAPI

La especificacion inicial esta en:

```
docs/openapi/factly-api.v1.openapi.json
```

Puedes cargarla en Redoc, Swagger UI, Stoplight o herramientas equivalentes para generar una referencia interactiva.

24. Criterios para pasar a producción

Antes de usar `production`, valida:

- El tenant esta activo.
- La suscripción no esta en mora.
- El modulo API esta incluido.
- Existe API key de `production`.
- El certificado `.p12` esta cargado y vigente.
- El tenant completo certificación y producción DGII fue activada.
- `DGII_ENV` del servidor apunta a producción.
- Tu sistema envia `encf` y `fechaVencimientoSecuencia` correctos.
- Probaste al menos un `E31`, `E32`, `E33` o `E34` segun tu caso.
- Tu sistema maneja `queued`, `sent`, `approved`, `rejected` y `failed`.
- Los webhooks se verifican por HMAC o tienes polling estable.
- Tienes reconciliacion por `externalId`, `id` Factly, `invoiceId`, `encf` y `trackId`.

25. Fuentes de buenas practicas usadas

- OpenAPI Initiative: <https://www.openapis.org/faq>
- Swagger - Authentication: https://swagger.io/docs/specification/v3_0/authentication/
- Google Developer Documentation Style Guide: <https://developers.google.com/style/highlights>
- Google API reference guidance: <https://developers.google.com/style/api-reference-comments>
- Stoplight API documentation guide: <https://stoplight.io/api-documentation-guide>
- Redoc: <https://github.com/Redocly/redoc>