

Tu facturación, así de fácil



Manual de Uso

iZi API v1

El siguiente documento tiene como fin explicar en detalle la forma de uso del Rest API de iZi. Al seguirlo, el cliente debería ser capaz de integrar su software mediante llamadas HTTPS a nuestros servidores.

Servidores y acceso

El Rest API de iZi funciona en dos servidores de manera paralela, uno de producción y uno de pruebas. Para acceder a cualquiera de los dos, deben usarse estas direcciones respectivamente:

Servidor de pruebas

<https://api.test.izifacturacion.com/v1>

Servidor de producción

<https://api.app.izifacturacion.com/v1>

Para poder acceder a los servicios de iZi, deberá realizar llamados HTTPS utilizando las credenciales de su cuenta. Los datos se intercambiarán en formato JSON. El API es stateless, lo que significa que no es necesario mantener una sesión activa con nuestros servidores.

Credenciales

Para acceder al API, se debe contar con el módulo de “Integración a Sistemas” de iZi. Si no cuentas con una suscripción en este momento puedes ponerte en contacto con nosotros a través de hola@izifacturacion.com.

Las llaves se componen de 3 elementos: un ClientId, que es un identificador de la cuenta dentro de nuestro sistema, un PrivateKey, que será utilizado para descryptar datos en nuestros servidores, y un PublicKey, que podrá ser utilizado para encriptar datos en el lado del cliente.

ClientId:

El clientId es un identificador de la cuenta del cliente. El ClientId es Único en todo el sistema.

PrivateKey:

Parte privada de la llave RSA que sirve para encriptar la comunicación entre el sistema del cliente y los servidores de iZi. Esta llave se guarda en el servidor de iZi y es usada para validar las solicitudes del cliente.

PublicKey:

Parte pública de la llave RSA que sirve para encriptar la comunicación entre el sistema del cliente y los servidores de iZi. Esta llave se puede descargar una única vez al momento de generarla, y debe ser almacenada de manera segura en los sistemas del cliente.

Las credenciales para el uso de la aplicación pueden obtenerse a través de la web app, en <https://app.izifacturacion.com>. Ingresa a la aplicación, selecciona la empresa en cuestión e ingresa a la sección de integraciones. Allí busca “Integración a Sistemas”. Aquí encontrarás las llaves si ya se habían generado previamente, o puedes hacer click en el botón de “Generar Llaves” para generar las llaves.

Guarde en un lugar seguro la llave pública, ya que es la única copia. Si se pierde, tendrá que volver a generar las credenciales.

Facturación por Terceros

Nota: este servicio sólo está disponible si contrataste el módulo adicional de “Facturación por Terceros”, que te permite emitir facturas para diferentes clientes y empresas.

Cada cliente que tenga acceso al API de iZi, tendrá la posibilidad de restringir para qué NITs se podrán emitir facturas a través de su cuenta. Esto cumple una doble función de seguridad: por un lado el cliente podrá limitar la emisión de facturas a contribuyentes no autorizados, y además le permitirá configurar la dosificación de facturación una única vez, sin necesidad de enviar estos datos con cada solicitud.

Dentro de la plataforma, en la configuración del cliente se puede definir una lista de contribuyentes asociados para los que el cliente podrá emitir facturas. Los datos a registrar son los siguientes:

Razon Social: del contribuyente asociado.

NIT: del contribuyente asociado.

Logo: del contribuyente asociado.

Correo Electrónico: del contribuyente asociado.

Actividad Económica: con la cual el contribuyente asociado emitirá facturas.

Sucursal: nombre de la sucursal con la cual el contribuyente asociado emitirá facturas, por defecto "Casa Matriz".

Dirección: de la Sucursal/Casa Matriz.

Zona: de la Sucursal/Casa Matriz.

Ciudad: de la Sucursal/Casa Matriz.

Teléfono: de la Sucursal/Casa Matriz.

Autorización de Facturación: obtenida en el portal web de impuestos, en la cuenta del cliente.

Llave de Facturación: obtenida en el portal web de impuestos, en la cuenta del cliente.

Último Número de Factura: el número de la última factura emitida por el cliente (de otra autorización u otro sistema de facturación).

Fecha Límite de Emisión: obtenida en el portal web de impuestos, en la cuenta del cliente.

Leyenda de Facturación: obtenida en el portal web de impuestos, en la cuenta del cliente.

Estado: Permite deshabilitar un contribuyente sin eliminarlo.

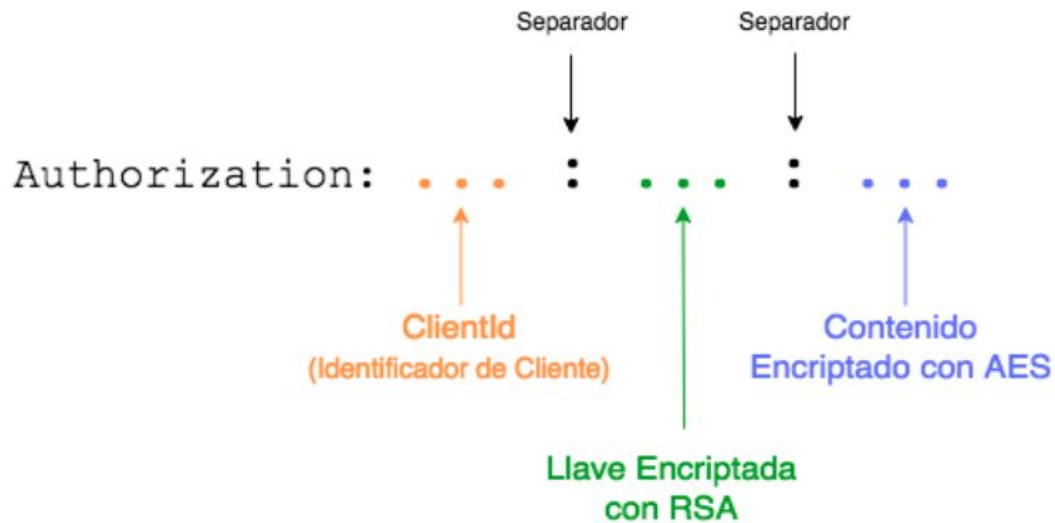
Cada vez que el cliente utilice el API para emitir una factura, nuestro sistema validará que el nit emisor esté dentro de la lista de los contribuyentes autorizados, y utilizará la información configurada allí para generar la factura.

Autorización

Los endpoints del API de iZi utilizan un sistema de encriptación híbrida para asegurar la entrada al sistema, en donde se utilizarán diferentes métodos para encriptar el contenido que se enviará mediante un header HTTP.

El API recibe esta validación con cada uno de los llamados que se realizan, para validar que los datos enviados son efectivamente generados por el dueño de la cuenta. Estos datos vienen encriptados en un header Authorization.

El header Authorization de los llamados al API se compone de 3 partes:



ClientId: El primer elemento del header, el clientId, se consigue a través del portal web de iZi, accediendo a la configuración de la cuenta.

Llave Encriptada con RSA: Esta es una llave definida por el cliente (se recomienda un valor aleatorio), que ha sido encriptada usando el privateKey del cliente.

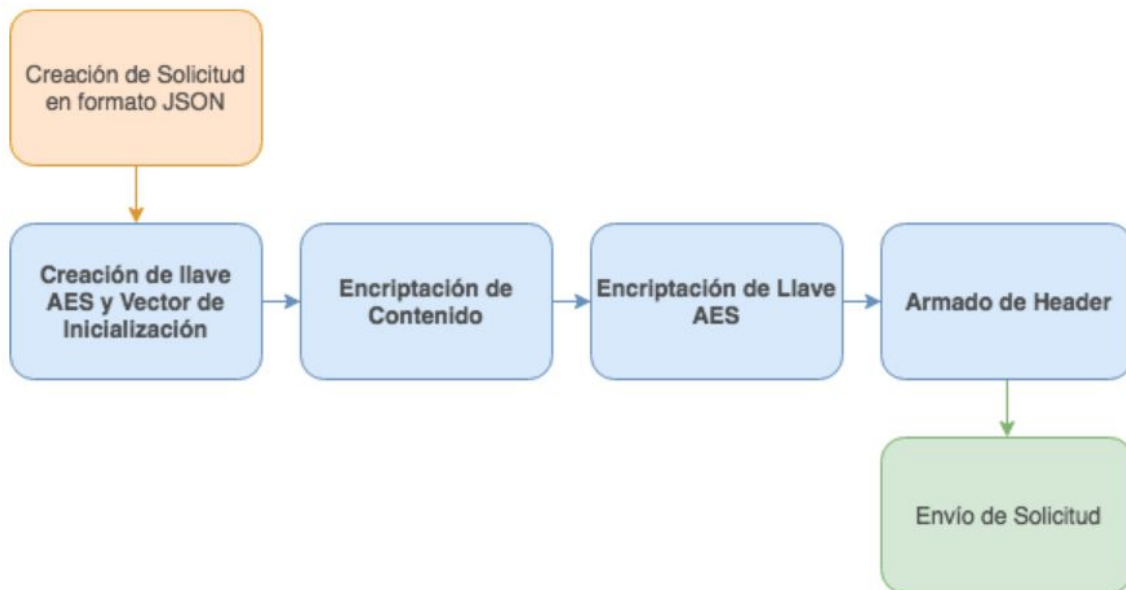
Contenido Encriptado con AES: El contenido que se enviará en el body de la solicitud, encriptado con la llave definida por el cliente.

Un ejemplo de este header podría ser:

```
8abbe332-8b73-45bf-b6df-bkk3364420a6:GP9HjUDcpUH087V0fGIIAsAMTouFYEOtMy1bNe8
GOi90k39zUnEBpkseiyr10Sfmx0/eez1e2P52G53M7zYw7wo6arfSotLlwJKmcA3gDLYDlsZPenm
ekREgVVbBu8wsRnRb9DwC7w+chUCsf8RXrV2MC+TRYh3mDWAAjxh7qsQ=:/jkFYcLeSZsZu5EV4K
aNlyvvnumpumiftqh0KF8oZ8ntSWmuRXF1+Sd2SNJpmRj1lfZFF3J+GlQOiDXsbuNtgqcVKsceVz9S
6N8qxhlP8pTMjiOjp3h+KcAXsXYVdYgKa0v/Dt0QC+IXscVtKjgcBztIcXTbZ/Y/arnrWdaAZxK2
TsCx9N24HDF+BMJ/mq9IUokj/kBek2r13+05t6KvBJQ==
```

Construyendo el header de Authorization

El proceso de construcción de un header de autorización comienza cuando el cliente desea mandar una solicitud al API de iZi, y sigue los siguientes pasos:



Creación de Solicitud en formato JSON

El cliente deberá armar un objeto JSON en el formato descrito en la sección de endpoints, dependiendo del endpoint que desee consultar.

Una vez armado el objeto, se debe convertir en un buffer para poder utilizarlo en el resto del flujo en encriptación.

NodeJS

```
var data = { ... };  
var bufferData = Buffer.from(data, "utf8");
```

Creación de Llave AES y Vector de Inicialización

La llave AES se compone de una cadena de 16 bytes. Se utiliza, al igual que el Vector de Inicialización (iv por sus siglas en inglés), para la encriptación del contenido en el token.

El iv también es una cadena de texto de 8 caracteres, que debe ser generada de manera aleatoria en cada solicitud.

Se recomienda que en cada solicitud se genere una llave AES y un iv, de manera aleatoria para mayor seguridad.

NodeJS

```
var key = crypto.randomBytes(32);  
var iv = crypto.randomBytes(16);
```

Encriptación de Contenido

Utilizando las recién generadas llaves, se procede a encriptar el contenido mediante AES, teniendo en cuenta el vector de inicialización. El contenido encriptado se utilizará en el header de autorización, y posteriormente servirá para validar que los datos de la solicitud sean correctos.

NodeJS

```
var cypher = crypto.createCipheriv('aes-256-cbc', key, iv);  
var cryptedAES = Buffer.concat([cypher.update(bufferData),  
cypher.final()]);
```

Encriptación de la Llave AES

Con el fin de transferir la llave AES de manera segura al servidor (y poder utilizarla así para desencriptar los datos y validarlos) se debe encriptar la llave AES sobre RSA. Para esto se usará la llave pública descargada del sitio web de iZi.

NodeJS

```
var constants = require("constants");
var encryptedKey = crypto.publicEncrypt({"key" : publicKey,
padding : constants.RSA_PKCS1_PADDING}, key);
```

Armado de Header

Finalmente armamos el string que se enviará como header en la solicitud, utilizando las credenciales generadas, así como el clientId obtenido del sitio web de iZi.

NodeJS

```
var auth = clientId + ":" + key.toString('base64') + ":" +
iv.toString('hex') + cryptedAES.toString('base64');
```

Envío de Solicitud

Como paso final enviamos la solicitud al API, organizada de la siguiente manera:

- Header Authorization con el string generado en el anterior paso
- Body de la solicitud con el contenido de la factura

Los servidores de iZi usarán la llave privada asociada a la cuenta para verificar que los valores enviados en el body correspondan a los desencriptados del header.

NodeJS

```
var request = require('request');

request({
  method: 'POST',
  uri: 'https://api.app.izifacturacion.com/v1/facturas',
  body: data,
  headers: {
    'Authorization': auth,
    'Content-Type': 'application/json'
  }
}, callback);
```

Endpoints

La lista de endpoints disponibles se presenta a continuación.

POST /v1/facturas

Este endpoint sirve para crear nuevas facturas. Las solicitudes se validan contra la lista de contribuyentes asociados para evitar facturación no autorizada

Propiedades (Parámetros POST)

emisor* (type: string , pattern: ^[0-9]*\$)

El NIT emisor (debe estar registrado en el portal web). Campo requerido.

comprador* (type: string , pattern: ^[0-9]+)

El NIT de quien realiza la compra. Campo requerido.

razonSocial* (type: string)

El nombre completo o la razón social del comprador.

sucursal (type: number)

El código de sucursal a utilizar, en caso de que se tenga más de una.

actividadEconomica (type: number)

El código de actividad económica a utilizar, en caso de que se tenga más de una.

listaItems* (type: array)

La lista de los artículos con sus precios y cantidades se describen en este campo. La forma de cada posición del arreglo es la siguiente

```
{
    articulo ( type: string ),
    cantidad ( type: number ),
    precioUnitario ( type: number )
}
```

descuentos (type: number)

Descuento sobre el valor total de la factura.

Un ejemplo de los datos JSON a enviarse sería:

```
{
    "emisor": "7777777",
    "razonSocial": "Empresa Prueba",
    "comprador": "4346405",
    "listaItems": [{
```

```
        "articulo": "Artículo de prueba",  
        "cantidad": 13,  
        "precioUnitario": 13  
    } ]  
}
```

Respuesta

200 - OK

Si la solicitud se procesó sin errores, se responderá con un objeto de la factura creada y el código 200.

Errores

Los errores a la solicitud se devuelven como texto plano.

401 - Unauthorized

"Cliente no Encontrado" - Cuando el clientId no corresponde a ningún cliente.

"Token inválido" - Si el header Authorization no se encuentra bien encriptado o no corresponde a la solicitud.

"No existen llaves para el cliente". Cuando el cliente no ha generado sus credenciales a través de la página de iZi.

404 - Not Found

"Cliente no Encontrado" - Cuando el clientId no corresponde a ningún cliente.

"No existen llaves para el cliente" - Cuando el cliente no ha generado sus credenciales a través de la página de iZi.

"Contribuyente autorizado no encontrado" - Si no es posible encontrar a un contribuyente asociado

406 - Not Acceptable

"Contribuyente desactivado" - Si se desactivó al contribuyente autorizado

"Autorización de facturación vencida" - Si la autorización de facturación ya expiró (su fecha límite de emisión ya pasó).

"Sucursal de Facturación Inválida" - En el caso de que esté facturando con su propio NIT como emisor, puede ocurrir este error si tiene configurada más de una sucursal. En este caso, puede enviar el código de sucursal como parámetro o configurarlo en la pestaña de API dentro de la aplicación de iZi. Para obtener el código de sucursal puede acceder a esta misma pestaña dentro de iZi.

"Actividad Económica de Facturación Inválida" - En el caso de que esté facturando con su propio NIT como emisor, puede ocurrir este error si tiene configurada más de una actividad económica. En este caso, puede enviar el código de actividad económica como parámetro o configurarlo en la pestaña de API dentro de la aplicación de iZi. Para obtener el código de sucursal puede acceder a esta misma pestaña dentro de iZi.

"No existe una autorización de facturación válida" - En el caso de que esté facturando con su propio NIT como emisor, puede ocurrir este error si no tiene configurada una autorización de facturación (dosificación) para sucursal y actividad económica especificados.

500 - Internal Server Error

En el caso de ocurrir un error interno de procesamiento, se enviará el detalle del fallo.

POST /v1/cobros

Este endpoint sirve para crear nuevos cobros, que a su vez generan una prefactura asociada al cobro. También es posible realizar la factura de manera previa y mandarla como parámetro, en este caso algunos valores de la solicitud no son necesarios y serán sobreescritos.

Propiedades (Parámetros POST)

descripcion* (type: string)

Una descripción corta para el cobro.

monto* (type: number)

El monto en Bs. para el cobro.

pasarela (type: string, default: CYBERSOURCE)

Pasarela por la cual se realizará el cobro. Puede ser:

- CYBERSOURCE - Pago con tarjetas de crédito y débito
- SIP - Pago con códigos QR
- Khipu - Pago con tarjetas de crédito y débito (deprecated)

Es posible mandar más de una pasarela de la siguiente manera: "CYBERSOURCE,SIP".

correoElectronico* (type: string)

El correo electrónico del pagador. Se enviará el cobro al menos que se especifique lo contrario en `notificarPagador`.

notificarPagador* (type: boolean, default: true)

Booleano para determinar si se notifica o no al pagador vía correo electrónico una vez generado el cobro.

nitFactura* (type: string , pattern: ^[0-9]*\$)

El NIT al cual se emitirá la factura una vez realizado el pago. Requerido si no se especifica una factura.

razonSocialFactura* (type: string)

La razón social a la cual se emitirá la factura una vez realizado el pago. Requerido si no se especifica una factura.

sucursal (type: number)

El código de sucursal a utilizar para la factura, en caso de que se tenga más de una.

actividadEconomica (type: number)

El código de actividad económica a utilizar para la factura, en caso de que se tenga más de una.

order (type: string)

Número de orden del cliente, puede ser el id del cobro en el sistema integrador.

notificacionUrl (type: string)

La URL a la que se realiza la notificación una vez realizado el pago. Se realiza un request tipo GET con los parámetros "result" como resultado de la operación (con valores "success" o "failed"), y "order" en el caso de que haya sido llenado. Un ejemplo sería:

```
notificacionUrl/?order=178&result=success
```

.

Un ejemplo de los datos JSON a enviarse para realizar un cobro sería:

```
{
  "descripcion": "Una compra electrónica",
  "monto": 66.20,
  "pasarela": "CYBERSOURCE",
```

```
"correoElectronico": "algun@mail.com",  
"notificarPagador": false,  
"pasarela": "CYBERSOURCE",  
"nitFactura": "7777777",  
"razonSocial": "Juan Perez",  
"notificacionUrl": "https://misitio.com/pago",  
}
```

Respuesta

200 - OK

Si la solicitud se procesó sin errores, se responderá con un objeto de la factura creada y el código 200.

Errores

Los errores a la solicitud se devuelven como texto plano.

401 - Unauthorized

"Cliente no Encontrado" - Cuando el `clienteId` no corresponde a ningún cliente.

"Token inválido" - Si el header `Authorization` no se encuentra bien encriptado o no corresponde a la solicitud.

"No existen llaves para el cliente". Cuando el cliente no ha generado sus credenciales a través de la página de iZi.

404 - Not Found

"Cliente no Encontrado" - Cuando el `clienteId` no corresponde a ningún cliente.

"No existen llaves para el cliente" - Cuando el cliente no ha generado sus credenciales a través de la página de iZi.

"Contribuyente autorizado no encontrado" - Si no es posible encontrar a un contribuyente asociado

406 - Not Acceptable

POST <https://test.izi.noujau.io/v1/encrypt-test>

Este endpoint está sólo disponible en el ambiente de pruebas, y además tiene la particularidad de que es el único endpoint que no recibe un token de autorización.

Propiedades

data* (type: object)

El objeto que se usará para cualquier otro endpoint, tal como se enviaría.

clientId* (type: string)

El clientId del cliente que realiza la prueba.

key* (type: string)

La llave generada con la cual se espera encriptar los contenidos. Un string utf-8 de 32 caracteres.

iv* (type: string)

El vector de inicialización generado con la cual se espera encriptar los contenidos. Un string utf-8 de 16 caracteres.

Respuesta

200 - OK

Si la solicitud se procesó sin errores, la respuesta será la siguiente:

```
{
  header: El token generado
}
```

Errores

Los errores a la solicitud se devuelven como texto plano.

401 - Unauthorized

"Cliente no Encontrado" - Cuando el clientId no corresponde a ningún cliente.

"No existen llaves para el cliente". Cuando el cliente no ha generado sus credenciales a través de la página de iZi.

Ejemplos de Implementación

PHP - Generación de Llaves

```
<?php

$public_key = "-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDC49KFfNzrCG5DKAdP8dFthNXi
hISCxP89X2pi8Z1EpYpv/msz345csQNNRM/t/NlwGTf7wErPkfRXjL5yYeVrbHAp
xXHKiBE7Y4yuY5To0/lq70QzhfsbUMLE8spvWN0qcHyszowUoJX1pVpnbj3TqS4o
w+Jm2u1oN14xL/MGUwIDAQEB
```

```

-----END PUBLIC KEY-----";

$factura_obj = [
    "emisor" => "321",
    "comprador" => "123",
    "razonSocial" => "razon_social_test",
    "listaItems" => [[
        "articulo" => "Artículo 1",
        "cantidad" => 5,
        "precioUnitario" => 98
    ], [
        "articulo" => "Artículo 2",
        "cantidad" => 4,
        "precioUnitario" => 76
    ], [
        "articulo" => "Artículo 3",
        "cantidad" => 3,
        "precioUnitario" => 54
    ]],
    "descuentos" => 0,
    "tipoCompra" => 1
];

$json_factura = json_encode($factura_obj);

$encryption_method = "aes-256-cbc";

// use this line to generate a 32 characters random string
$secret_key = bin2hex(openssl_random_pseudo_bytes(16));

// use this line as constant secret key
//$secret_key = "secret0key0to0encrypt0under0aes0";

$iv_size = openssl_cipher_iv_length($encryption_method);
$iv = openssl_random_pseudo_bytes($iv_size);

$encrypted_json_factura = @openssl_encrypt ($json_factura,
$encryption_method, $secret_key, 0, $iv);
$encrypted_json_factura = bin2hex($iv) . $encrypted_json_factura;

openssl_public_encrypt($secret_key, $encrypted_secret_key, $public_key);

```

```

$encrypted_secret_key = base64_encode($encrypted_secret_key);

echo "--> Expected output" . PHP_EOL;
echo "--> user_id:encrypted_secret_key:encrypted_json" . PHP_EOL;
echo "clientId" . ":" . $encrypted_secret_key . ":" .
$encrypted_json_factura . PHP_EOL;

?>

```

NodeJS - Generación de Llaves

```

function getAuthorizationHeader (data, IZI_CLIENT_ID, IZI_PUBLIC_KEY) {

    const bufferData = Buffer.from(JSON.stringify(data), 'utf8')

    const key = crypto.randomBytes(32)
    const iv = crypto.randomBytes(16)

    const cypher = crypto.createCipheriv('aes-256-cbc', key, iv)
    const cryptedAES = Buffer.concat([cypher.update(bufferData),
    cypher.final()])

    const encryptedKey = crypto.publicEncrypt({
        'key': IZI_PUBLIC_KEY,
        padding: constants.RSA_PKCS1_PADDING
    }, key);

    return IZI_CLIENT_ID + ':' + encryptedKey.toString('base64') + ':' +
    iv.toString('hex') + cryptedAES.toString('base64')
}

```

Ejemplo de la Solicitud en Curl

```

curl --header "Content-Type: application/json" \
  --header "Authorization:
8abbe332-8b73-45bf-b6df-bkk3364420a6:GP9HjUDcpUH087V0fGIIAsAMTouFYE0Tmy1
bNe8G0i90k39zUnEBpkseiyr10Sfmx0/eez1e2P52G53M7zYw7wo6arfS0tLLwJKmcA3gDLY

```

```

D1sZPenmekREgVVbBu8wsRnRb9DwC7w+chUCsf8RXrV2MC+TRYh3mDWAAjxh7qsQ=:/jkFYc
LeSZsZu5EV4KaNlyvvnumpumiftqh0KF8oZ8ntSWmuRXF1+Sd2SNJpmRj1lfZFF3J+G1Q0iDXs
buNtgqcVKsceVz9S6N8qxh1P8pTMji0jp3h+KcAXsXYVdYgKa0v/Dt0QC+IXscVtKjgcBZtI
cXTbZ/Y/arnrWdaAZxK2TsCx9N24HDF+BMJ/mq9IUokj/kBek2r13+05t6KvBJQ=== " \
--request POST \
--data '{"emisor": "77777777","razonSocial": "Empresa
Prueba","comprador": "4346405","listaItems": [{"articulo": "Artículo de
prueba","cantidad": 13,"precioUnitario": 13}]}' \
https://test.izi.noujau.io/v1/facturas

```